

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова праця  
на правах рукопису

**ЧОПОРОВА ОКСАНА ВОЛОДИМИРІВНА**

УДК: 004.896.032.26:539.3(043.5)

**ДИСЕРТАЦІЯ**

**ПРОГНОЗУВАННЯ ЗОН РУЙНУВАННЯ В ІНЖЕНЕРНИХ  
КОНСТРУКЦІЯХ З ВИКОРИСТАННЯМ КОМП'ЮТЕРНОГО НАВЧАННЯ**

Спеціальність: 122 Комп'ютерні науки

Галузь знань: 12 Інформаційні технології

Подається на здобуття наукового ступеня **доктора філософії**

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_ О. В. Чопорова

Науковий керівник: Лісняк Андрій Олександрович,  
кандидат фізико-математичних наук,  
доцент

Запоріжжя 2021

## АНОТАЦІЯ

**Чопорова О. В. Прогнозування зон руйнування в інженерних конструкціях з використанням комп'ютерного навчання.** – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки». – Запорізький національний університет Міністерства освіти і науки України, Запоріжжя, 2021.

У вступі обґрунтовано актуальність теми дисертаційної роботи, сформульовано мету, завдання, об'єкт, предмет дослідження, розкрито наукову новизну та практичну значущість отриманих результатів, визначено особистий внесок здобувача, апробацію результатів дослідження та їх висвітлення у наукових публікаціях. Приводяться відомості щодо структури та обсягу дисертаційної роботи.

У першому розділі «Аналіз предметної області моделювання тонкостінних конструкцій методами машинного навчання» розглянуто та проаналізовано сучасний стан досліджень за тематикою роботи. Зроблено огляд публікацій, проаналізовано нейромережеві методи моделювання та прогнозування у задачах механіки. Розглянуто публікації із застосуванням генетичних алгоритмів для структурної оптимізації нейронних мереж. Проаналізовано генетичні алгоритми у задачах моделювання напружено-деформованого стану конструкцій. З аналізу публікацій зроблено висновок про актуальність розробки нейромережевих методів моделювання мереж для визначення напружено-деформованого стану та інтенсивності напружень за Мізесом пластин та оболонок, а також прогнозування зон руйнування в інженерних конструкціях. Підкреслено, що нейромережеві моделі є одним із найбільш адаптивних методів прогнозування, однак використання цих методів на практиці вимагає налаштування великої кількості параметрів нейронних мереж. Одним із підходів до автоматичного вибору структури нейромереж та налаштування їхніх гіперпараметрів є генетичні алгоритми.

Другий розділ присвячено розробці нейромережевому методу для прогнозування напружено-деформованого стану плоских пластин. Зокрема, квадратної, круглої та кільцевої пластин, а також квадратної пластини з круглим вирізом та з різними умовами закріплення. Розглянуто особливості генерації навчальної вибірки, архітектури нейронної мережі та обчислювальний експеримент для кожного випадку форми пластини. Розглянуто випадки нейромережевого моделювання пластин зі сталими умовами закріплення (наприклад, затиснених), а також особливості архітектури нейронних мереж для моделювання пластин з довільними крайовими умовами. Досліджено вплив кількості нейронів, шарів нейронів і функцій активації на точність моделювання. Для генерації навчальної вибірки пластин з отвором довільного розміру та розташування розроблено скінченно-елементну модель. Параметри навчальних вибірок обрано таким чином, щоб моделі включали переважну більшість типових металевих пластин.

Розділ 3 «Використання машинного навчання для прогнозування напружено-деформованого стану оболонок» сконцентровано на розробці нейромережевих методів для прогнозування напружено-деформованого стану циліндричної оболонки, а також комбінації циліндричної та конічної оболонок. Для кожного випадку навчальну вибірку згенеровано на базі скінченно-елементної моделі оболонки. Розроблена архітектури нейронних мереж, а також проведено обчислювальні експерименти. Проілюстровано точність прогнозування нейромережевих моделей.

Четвертий розділ «Використання генетичного алгоритму для оптимізації архітектури нейронної мережі прогнозування напружено-деформованого стану конструкції» присвячено особливостям застосування генетичного алгоритму для оптимізації архітектури нейронної мережі. Розглянуто кодування архітектури нейронної мережі, функцій кросоверу та мутації. Проведено обчислювальний експеримент з генетичної оптимізації архітектури нейронної мережі для моделювання напружено-деформованого стану прямокутної пластини. Також приділено увагу генетичній оптимізації

архітектури нейронної мережі для моделювання напружено-деформованого стану циліндричної оболонки.

У висновках наведено наукову новизну роботи, її практичну значимість і перспективи подальшого розвитку.

У дисертаційній роботі отримано такі наукові результати:

- уперше розроблено нейромережевий метод для визначення напружено-деформованого стану пластин з довільними умовами закріплення, який дозволяє визначити максимальний прогин, а також максимальне значення інтенсивності напружень за Мізесом;

- уперше розроблено нейромережевий метод для визначення напружено-деформованого стану циліндричної оболонки та комбінації циліндричної та конічної оболонок, який дозволив визначити прогин та інтенсивність напружень за Мізесом;

- уперше розроблено нейромережевий метод на базі архітектури «автокодувальник» для визначення можливих зон руйнування у квадратних пластинах з отвором шляхом генерації ймовірних картин розподілу напружень;

- удосконалено нейромережеві методи моделювання напружено-деформованого стану тонкостінних конструкцій шляхом розробки генетичного алгоритму оптимізації архітектури нейронної мережі, що дало змогу підвищити точність прогнозування.

Програмну реалізацію розроблених методів виконано мовою програмування Python з використанням бібліотек scikit-learn, numpy, Pandas і Keras.

Отримані результати дослідження впроваджено в навчальний процес при вивченні дисциплін «Емпіричні методи програмної інженерії», «Засоби машинного навчання», «Нейронні мережі» та виконанні кваліфікаційних робіт студентами спеціальності 121 «Інженерія програмного забезпечення» Запорізького національного університету.

**Ключові слова:** мережа штучних нейронів, напружено-деформований стан, пластина, оболонка, прогнозування, генетичний алгоритм, популяція, мутація, схрещування, функція активації.

## ABSTRACT

Choporova, O. V. **Damage zones forecasting for engineering constructions using machine learning.** PhD Thesis. Manuscript. Qualifying scientific.

Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy. Study program: 122 – Computer Science. Zaporizhzhia National University of the Ministry of Education and Science of Ukraine, Zaporizhzhia, 2021.

The introduction substantiates the relevance of the work, formulates the purpose, objectives, object, subject of research, reveals the scientific novelty and practical significance of the results, provides data on testing the work and implementation of the results in practice, the number of publications.

In the first chapter the current state of research on the subject of work is considered and analyzed. A review of publications is made, neural network methods of modeling and predicting in mechanics problems are analyzed. Publications on the application of genetic algorithms for structural optimization of neural networks are considered. Genetic algorithms in the problems of modeling the stress-strain state of structures are analyzed. From the analysis of publications, it is concluded that there is no instrumental system for modeling neural networks to determine the stress-strain state and stress intensity according to Mises of plates and shells, and predicting failure zones in engineering structures. Underlined, that neural network models are one of the most adaptive methods of forecasting, but the use of these methods in practice requires the adjustment of a large number of parameters of neural networks.

The second chapter is devoted to the development of a neural network method for predicting the stress-strain state of flat plates. In particular, square, round and ring plates, and a square plate with a circle cut-out and with different boundary conditions. Features of training sample generation, neural network architecture and computational experiment for each case of plate shape are considered. Cases of neural network modeling of a layer with constant fixing conditions (for example,

clamped), as well as features of neural network architecture for modeling plates with arbitrary boundary conditions are considered. The influence of the number of neurons, layers of neurons and activation functions on the accuracy of modeling has been studied. A finite element model has been developed to generate a training sample of plates with a cut-out of arbitrary size and location. The parameters of the training samples were chosen so that the models included the vast majority of typical metal plates.

Chapter 3 "Using machine learning to predict the stress-strain state of shells" focuses on the development of neural network methods for predicting the stress-strain state of a cylindrical shell, and a combination of cylindrical and conical shells. For each case, the training sample was generated on the basis of a finite-element model. The architecture of neural networks is developed, and also computational experiments are carried out. The accuracy of forecasting neural network models is illustrated.

The fourth chapter "Using of a genetic algorithm to optimize the architecture of the neural network for predicting the stress-strain state of the structure" is devoted to the features of the application of genetic algorithms for the optimization of the neural network architecture. The coding of neural network architecture, crossover functions and mutations is considered. A computational experiment on genetic optimization of neural network architecture for modeling the stress-strain state of a rectangular plate is performed. Attention is also paid to the genetic optimization of the neural network architecture for modeling the stress-strain state of a cylindrical shell.

The conclusions present the scientific novelty of the work, its practical significance and prospects for further development.

The following results were obtained:

- for the first time a neural network method was developed to determine the stress-strain state of plates with arbitrary boundary conditions, which allows to determine the maximum deflection, and the maximum value of the stress intensity according to Mises;

- for the first time a neural network method was developed to determine the stress-strain state of a cylindrical shell and a combination of cylindrical and conical shells, which allowed to determine the deflection and intensity of stresses according to Mises;
- for the first time developed a neural network method based on the architecture "autocoder" to determine possible fracture zones in square plates with a cut-out by generating probable patterns of stress distribution;
- improved neural network methods for modeling the stress-strain state of structures by developing a genetic algorithm for optimizing the architecture of the neural network, which allowed to increase the accuracy of prediction.

The software implementation of the developed methods is performed in the Python programming language using the scikit-learn, numpy, Pandas and Keras libraries.

The obtained research results are introduced into the educational process in the study of disciplines «Empirical methods of software engineering», «Machine learning tools», «Neural networks» and qualification work of students majoring in 121 «Software Engineering» Zaporizhzhia National University.

**Keywords:** network of artificial neurons, stress-strain state, plate, shell, prediction, genetic algorithm, population, mutation, crossing, activation function.

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

### Наукові праці, в яких опубліковано основні наукові результати дисертації:

*Наукові публікації, що висвітлюють основний зміст дисертаційної роботи:*

1) **Choporova O. V.**, Choporov S. V., Lisniak A. A. Implementation of genetic algorithm for neural network optimization for making predictions of the stress-strain state of a rectangular plate with a circular cut out. In the book: *Mathematical and Computer Modelling of Engineering Systems*. Riga: Baltija Publishing. 2020 P. 1-18. URL: <http://www.baltijapublishing.lv/omp/index.php/bp/catalog/view/89/2215/4790-1>. DOI: 10.30525/978-9934-26-019-3-1.

2) **Чопорова О. В.**, Чопоров С. В., Лісняк А. О. Використання машинного навчання для прогнозування напружено-деформованого стану квадратної пластинки. *Вісник Херсонського національного технічного університету*. 2019. № 2(69). С. 192-201. (Index Copernicus).

3) Lisnyak A. O., **Choporova O. V.** Using machine learning to predict the stress-strain state of a rectangular plate with a circular cut-out. *CEUR Workshop Proceedings*. 2020. Vol. 2791. P. 1–6. ISSN: 1613-0073. URL: [4\) \*\*Чопорова О. В.\*\*, Чопоров С. В., Лісняк А. О. Використання генетичного алгоритму для оптимізації параметрів нейронної мережі при прогнозуванні напружено-деформованого стану квадратної пластинки. \*Вісник Херсонського національного технічного університету\*. 2020. Т. 3 № 2.1. С. 290–300. \(Index Copernicus\).](https://www.scopus.com/record/display.uri?eid=2-s2.0-85098720535&origin=resultslist&sort=plf-f&src=s&st1=&st2=&sid=80dfe74813393a8626e1696d73fee20a&sp; sot=b&sdt=b&sl=112&s=TITLE+%28Using+machine+learning+to+predict+the+stress-strain+state+of+a+rectangular+pl. (Scopus)</a>.</p>
</div>
<div data-bbox=)

5) **Чопорова О. В.**, Кривохата А. Г. Оптимізація згорткових нейронних мереж та їх ансамблів. *Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки*. 2019. № 1. С. 107–115. (Index Copernicus).

*Наукові публікації, що висвітлюють апробацію дисертаційної роботи:*

6) **Чопорова О. В.** Використання машинного навчання для прогнозування поведінки інженерних конструкцій. *Сучасні проблеми машинобудування*. Харків: Інститут проблем машинобудування ім. А. М. Підгорного, 2019. С. 19.

7) **Чопорова О. В.**, Чопоров С. В., Лісняк А. О. Модель прогнозування папружено-деформованого стану прямокутної пластини з круглим вирізом. *Інформаційні системи та технології (ICT-2019): VIII Міжнародна науково-технічна конференція*. Коблево-Харків: ХНУРЕ, 2019. С. 97–99. URL: [http://istconf.nure.ua/archive/ist\\_2019.pdf](http://istconf.nure.ua/archive/ist_2019.pdf).

8) **Чопорова О. В.**, Чопоров С. В., Лісняк А. О. Реалізація генетичного алгоритму для оптимізації нейронної мережі при прогнозуванні напружено-деформованого стану прямокутної пластини з круглим вирізом. *Інформаційні системи та технології (ICT-2020): 9-та Міжнародної науково-технічної конференції*. Коблево-Харків: ХНУРЕ, 2020. С. 303–306. URL: [http://istconf.nure.ua/archive/ist\\_2020.pdf](http://istconf.nure.ua/archive/ist_2020.pdf).

9) **Чопорова О. В.**, Лісняк А. О. Аналіз можливостей мови програмування Python у машинному навчанні. Збірка тез доповідей Десятої Всеукраїнської, сімнадцятої регіональної наукової конференції молодих дослідників "Актуальні проблеми математики та інформатики". Запоріжжя: Видавничий дім "Гельветика", 2019. С. 57–59.

## ЗМІСТ

АНОТАЦІЯ.....	2
ABSTRACT.....	5
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА.....	8
ЗМІСТ .....	10
ВСТУП.....	13
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ МОДЕЛЮВАННЯ ТОНКОСТІННИХ КОНСТРУКЦІЇ МЕТОДАМИ МАШИННОГО НАВЧАННЯ.....	18
1.1 Постановка задачі прогнозування поведінки тонкостінних конструкцій .....	18
1.2 Аналіз нейромережових методів моделювання та прогнозування у задачах механіки .....	19
1.3 Аналіз генетичних методів у задачах моделювання напружено- деформованого стану конструкцій.....	25
1.4 Висновки до розділу 1 .....	32
2 ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ НАПРУЖЕНО-ДЕФОРМОВАНОГО СТАНУ ПЛАСТИН .....	34
2.1 Нейромережеве моделювання напружено-деформованого стану плоских об'єктів.....	34
2.2 Особливості генерації вибірки методом скінченних елементів .....	43
2.3 Метод нейромережевого моделювання напружено-деформованого стану квадратної пластини .....	49
2.3.1 Генерація навчальної вибірки квадратної пластини .....	50
2.3.2 Архітектура нейронної мережі.....	51

	11
2.3.3 Обчислювальний експеримент.....	52
2.4 Нейромережевий метод моделювання напружено-деформованого стану круглої пластини.....	56
2.4.1 Генерація навчальної вибірки круглої пластини.....	58
2.4.2 Архітектура нейронної мережі.....	61
2.4.3 Обчислювальний експеримент.....	62
2.4.4 Моделювання довільних умов закріплення на прикладі круглої пластини .....	65
2.5 Нейромережевий метод моделювання напружено-деформованого стану кільцевої пластини.....	70
2.5.1 Генерація навчальної вибірки кільцевої пластини .....	70
2.5.2 Архітектура нейронної мережі.....	74
2.5.3 Обчислювальний експеримент.....	76
2.6 Врахування у нейромережевій моделі довільних умов закріплення на прикладі прямокутної пластинки з отвором.....	77
2.7 Розробка нейромережевого методу на базі архітектури «автокодувальник» для визначення можливих зон руйнування у квадратних пластинах з отвором.....	83
2.8 Висновки до розділу 2 .....	87
3 ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ НАПРУЖЕНО-ДЕФОРМОВАНОГО СТАНУ ОБОЛОНОК .....	89
3.1 Генерація даних методом скінченних елементів .....	89
3.2 Нейромережевий метод моделювання напружено-деформованого стану циліндричної оболонки з довільними умовами закріплення.....	93
3.2.1 Архітектура нейронної мережі.....	99

	12
3.2.2 Обчислювальний експеримент.....	99
3.3 Нейромережевий метод моделювання напружено-деформованого стану комбінації циліндр-конус оболонок з довільними умовами закріплення .....	101
3.3.1 Архітектура нейронної мережі для прогнозування напружено-деформованого стану комбінації циліндр-конус.....	107
3.3.2 Обчислювальний експеримент.....	108
3.4 Висновки до розділу 3 .....	109
4 ЗАСТОСУВАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ У ЗАДАЧАХ ПРОГНОЗУВАННЯ.....	110
4.1 Особливості застосування генетичних алгоритмів для оптимізації архітектури нейронної мережі.....	110
4.2 Генетична оптимізація архітектури нейронної мережі для моделювання напружено-деформованого стану прямокутної пластини .....	114
4.2.1 Обчислювальний експеримент.....	116
4.3 Генетична оптимізація архітектури нейронної мережі для моделювання напружено-деформованого циліндричної оболонки.....	120
4.3.1 Обчислювальний експеримент.....	122
4.4 Висновки до розділу 4 .....	123
ВИСНОВКИ .....	124
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	126
ДОДАТОК А .....	140
Акт впровадження у навчальний процес Запорізького національного університету .....	140

## ВСТУП

Дисертаційну роботу присвячено вирішенню актуальної науково-практичної задачі – розробці штучних нейронних мереж для прогнозування зон руйнування в інженерних конструкціях.

Нейронні мережі успішно використовуються для вирішення різних прикладних задач. Вони використовуються у широкому спектрі додатків, таких як розпізнавання образів, прогнозування, задачі управління та інші. Одним із важливих напрямлень практичного використання штучних нейронних мереж є задачі прогнозування. До задач прогнозування відносяться задачі прогнозування цін на сировину та інші товари, задачі визначення властивостей нових матеріалів по їх структурі, прогнозування навантажень енергетичних систем, а також можна віднести і задачі прогнозування зон руйнування в інженерних конструкціях. Узагальнюючи області використання штучних нейронних мереж, можна стверджувати, що штучні нейронні мережі можуть використовуватися при вирішенні будь-яких задач, для вирішення яких відсутня адекватна математична модель, в тому числі недостатньо даних для ефективного використання методів їх аналізу. Таким чином, використання штучних нейронних мереж у задачах прогнозування є актуальним.

**Актуальність теми дисертаційного дослідження.** Автоматизоване проектування об'єктів сучасної техніки пов'язане з необхідністю швидкого прогнозування їх стану або поведінки у залежності від геометричних і/або фізико-механічних параметрів. Класичні методи математичного моделювання дозволяють робити такі прогнози з необхідною точністю. Проте підготовка адекватних математичних моделей та проведення відповідних обчислювальних експериментів можуть бути тривалими за часом. Можливою альтернативою є розробка методів і моделей машинного навчання, які дозволять робити швидкі оцінки необхідних параметрів стану об'єкту. З практичної точки зору моделі машинного навчання для прогнозування значень

параметрів стану конструкції можуть слугувати як інтерактивні асистенти у процесі проектування.

У розвиток теорії та практики розробки нейромережових моделей суттєвий внесок зробили такі вчені: Безсонов О. О., Бодянський Є. В., Зайченко Ю. П., Згуровський М. З., Нестеренко Б. Б., Новотарський М. А., Олійник О. О., Руденко О. Г., Синєглазов В. М., Субботін С. О., Тимошук П. В., Чумаченко О. І., Ямпольський Л. С., Bengio Y., Goodfellow I., Hinton G., LeCun Y., Ng A., Rosenblatt F., Schmidhuber J. та ін.

Істотний внесок у розвиток еволюційних методів та їх використання в задачах дискретної оптимізації зробили такі науковці, як-от: Безсонов О. О., Козін І. В., Олійник А. О., Пічугіна О. С., Перепелиця В. О., Субботін С. О., Скобцов Ю. А., Holland J., Fogel L., Rechenberg I., Schwefel H-P. та ін.

Отже, актуальною задачею є розробка методів та підходів до автоматичної оптимізації нейронних мереж із метою підвищення точності прогнозування зон руйнування в інженерних конструкціях.

**Мета і задачі дослідження** – розробка штучних нейронних мереж для прогнозування напружено-деформованого стану пластин, оболонок та комбінації оболонок із ізотропного матеріалу з довільними умовами закріплення, які знаходяться під дією рівномірно розподіленого по всій поверхні поперечного тиску.

Для досягнення мети в роботі, на основі систематичного огляду та аналізу літературних джерел, що розкривають сучасний стан досліджень в обраній області, поставлені та вирішені наступні завдання:

- розробити методи формування вибірки для навчання та тестування моделей;
- розробити нейромережові методи прогнозування максимального прогину пластин та оболонок;
- розробити нейромережові методи прогнозування максимальних значень інтенсивності напружень за Мізесом у пластинах та оболонках;

- розробити метод генетичної оптимізації архітектури нейронної мережі для моделювання напружено-деформованого стану тонкостінних конструкцій;

- розробити нейромережевий метод на базі архітектури «автокодувальник» для визначення можливих зон руйнування у квадратних пластинах з отвором шляхом генерації ймовірних картин розподілу напружень.

**Об’єкт дослідження** – є моделі та методи комп’ютерного моделювання напружено-деформованого стану пластин і оболонок.

**Предмет дослідження** – методи і моделі машинного навчання для регресійного аналізу при комп’ютерному дослідженні напружено-деформованого стану пластин і оболонок.

**Наукова новизна одержаних результатів:**

- уперше розроблено нейромережевий метод для визначення напружено-деформованого стану пластин з довільними умовами закріплення, який дозволяє визначити максимальний прогин, а також максимальне значення інтенсивності напружень за Мізесом;

- уперше розроблено нейромережевий метод для визначення напружено-деформованого стану циліндричної оболонки та комбінації циліндричної та конічної оболонок, який дозволив визначити прогин та інтенсивність напружень за Мізесом;

- удосконалено нейромережеві методи моделювання напружено-деформованого стану тонкостінних конструкцій шляхом розробки генетичного алгоритму оптимізації архітектури нейронної мережі;

- уперше розроблено нейромережевий метод на базі архітектури «автокодувальник» для визначення можливих зон руйнування у квадратних пластинах з отвором шляхом генерації ймовірних картин розподілу напружень.

Програмну реалізацію розроблених методів виконано мовою програмування Python з використанням бібліотек scikit-learn, numpy, Pandas і Keras.

### **Методи дослідження:**

Досягнення мети та виконання завдань дисертаційного дослідження здійснено за допомогою використання методів аналітичної геометрії, математичного моделювання, теорії нейронних мереж, генетичних алгоритмів.

Отримані результати дослідження впроваджено в навчальний процес при вивченні дисциплін «Емпіричні методи програмної інженерії», «Засоби машинного навчання», «Нейронні мережі» та виконанні кваліфікаційних робіт студентами спеціальності 121 «Інженерія програмного забезпечення» Запорізького національного університету.

**Апробація результатів дисертації.** Основні результати, що отримані у дисертаційному дослідженні доповідались на конференціях:

1. «Сучасні проблеми машинобудування» конференція молодих вчених та спеціалістів (м. Харків, 15-18 квітня 2019);
2. Десята Всеукраїнська, сімнадцята регіональна наукова конференція молодих дослідників «Актуальні проблеми математики та інформатики» (м. Запоріжжя, 25-26 квітня 2019);
3. 8-ї Міжнародна науково-технічна конференція «Інформаційні системи та технології (ICT-2019)» (м. Коблево-Харків, 9-14 вересня 2019);
4. 9-та Міжнародна науково-технічна конференція «Інформаційні системи та технології (ICT-2020)» (м. Харків, 17-20 листопада 2020);
5. 16<sup>th</sup> International Conference on ICT in Education, Research and Industrial Applications, PhD Symposium (м. Харків, 06-10 жовтня 2020).

**Публікації.** За результатами досліджень опубліковано п'ять наукових робіт, що висвітлюють основний зміст дисертації, серед яких одна робота – розділ колективної монографії, виданої англійською мовою у Європейському Союзі [112]; одну статтю опубліковано у міжнародному періодичному виданні, що проіндексовано у базі даних Scopus [34]; три статті опубліковано у наукових виданнях України, що включено до Переліку наукових фахових видань України, в яких можуть публікуватися результати дисертаційних робіт

на здобуття наукових ступенів доктора наук, кандидата наук та ступеня доктора філософії [113 – 115]; також опубліковано чотири роботи апробаційного характеру [116 – 119].

**Структура і обсяг дисертації.** Дисертаційна робота складається з анотації, вступу, чотирьох розділів, висновків, списку використаних джерел, додатку. Загальний обсяг роботи складає 140 сторінок. Робота містить 64 рисунки, 2 таблиці, список використаних джерел із 122 найменувань та 1 додаток.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ МОДЕЛЮВАННЯ ТОНКОСТІННИХ КОНСТРУКЦІЇ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

## 1.1 Постановка задачі прогнозування поведінки тонкостінних конструкцій

Необхідність в дослідженні поведінки інженерних конструкцій виникає в різноманітних галузях таких як будівництво, автомобілебудування, ракетобудування, і ін. Машинне навчання дозволяє отримати моделі для миттєвої (щодо, наприклад, методу скінченних елементів) оцінки параметрів стану об'єкта. Таким чином, розробка методів машинного навчання для прогнозування поведінки інженерних конструкцій є актуальною.

Класичні методи математичного моделювання дозволяють робити прогнози з необхідною точністю, але підготовка математичних моделей і проведення відповідних обчислювальних експериментів можуть бути тривалими за часом. Можливо альтернативою є розробка методів і моделей машинного навчання, які дозволяють робити швидкі оцінки необхідних параметрів стану об'єкта. Методи машинного навчання дозволяють адаптувати свою структуру шляхом обробки вибірки даних. Вони дозволяють розширити можливості математичного моделювання за допомогою обліку результатів обчислювальних і натурних експериментів. З практичної точки зору моделі машинного навчання для прогнозування значень параметрів стану конструкції можуть служити інтерактивними асистентами в процесі проектування.

В авіаційній техніці і кораблебудуванні поширене використання тонкостінних конструкцій. Такі конструкції можна моделювати пластинами та оболонками. Моделі, засновані на мережах штучних нейронів повинні

враховувати геометричні і фізико-механічні параметри тіла, а також умови закріплення його граней.

Завдання прогнозування параметрів стану об'єкта по його геометричним і фізико-механічними параметрами можна віднести до класу задач регресії. Для прогнозування напружено-деформованого стану пластин та оболонок необхідно враховувати наступні параметри: ширину, висоту, модуль Юнга, ставлення Пуассона, навантаження. Також необхідно враховувати умови закріплення: пластина або оболонка навантажена поперечно розподіленим навантаженням на одиницю площі; на краях можлива довільна комбінація таких крайових умов: затиснення (заборонені переміщення і обертання), шарнірне закріплення (заборонені переміщення, обертання дозволені) або вільні. Можлива будь-яка комбінація граничних умов, виключаючи ситуації, які можуть обумовлювати відсутність рішення: з чотирма вільними ребрами або одним шарнірно закріпленим ребром і трьома вільними ребрами.

## **1.2 Аналіз нейромережових методів моделювання та прогнозування у задачах механіки**

Області застосування нейронних мереж досить різноманітні – це розпізнавання тексту і мови, семантичний пошук, експертні системи і системи підтримки прийняття рішень, передбачення курсів акцій, системи безпеки, аналіз текстів, прогнозування інженерних конструкцій у різних галузях таких як авіабудування, рекетобудування, автомобілебудування, та ін.

В останні кілька років спостерігається вибух інтересу до нейронних мереж, які успішно застосовуються в самих різних областях – бізнесі, медицині, техніці, геології, фізики. Нейронні мережі увійшли в практику

всюди, де потрібно вирішувати завдання прогнозування, класифікації або управління завдяки багатим можливостям та простоті у використанні.

У різноманітних галузях, таких як будівництво, автомобілебудування, ракетобудування тощо, виникає потреба у дослідженні поведінки інженерних конструкцій. Проведення натурних експериментів є вартісними та трудомісткими. Машинне навчання дозволяє їх замінити серією комп'ютерних експериментів. Отже, розробка методів машинного навчання для прогнозування поведінки інженерних конструкцій є актуальною.

Перший практичний додаток штучних нейронних мереж відноситься до 1950 років та пов'язаний з винаходом Розенблаттом перцептрона та відповідного правила його навчання [46]. Розенблатт показав його здатність до навчання та вирішенню задач класифікації. Видров та Хофф [55] запропонували інший навчальний алгоритм для налаштування адаптивних лінійних нейронних мереж. Навчальний алгоритм Видрова—Хоффа використовується і в наш час. Велика кількість задач, для вирішення яких використовуються нейронні мережі, можуть розглядатися як окремі випадки наступних основних проблем: класифікація об'єктів, апроксимація функції по кінцевому набору її значень, оптимізація, побудова відношень на множині об'єктів, нейромережева реалізація алгоритмів обчислювальної математики.

На початку 2000-х років почався розвиток глибокого навчання згорткових мереж на основі навчання з вчителем без попереднього навчання без вчителя. У роботі [39] вперше було описане застосування алгоритма зворотнього розповсюдження помилки для навчання глибокої нейронної мережі з архітектурою, що подібна неокогнітрону та кресцептрону, що складається з шарів згортки та максимального вибору. Така архітектура нейронних мереж активно використовується і до нашого часу [54, 40, 47, 7, 30].

Великий внесок у розвиток згорткових нейронних мереж внесла пропозиція використовувати напівлінійну функцію активації (rectified linear unit) [17, 18], що задається у наступному вигляді:  $ReLU(x) = \max(0, x)$ . Така

функція активації дозволяє позбавитися від проблеми зникаючого градієнту, так як при додатньому значення сигналу не відбувається його змін, на відміну від сигмоїдальної функції активації. Крім того, напівлінійна функція активації дозволяє скоротити час навчання нейронної мережі [13]. Нейрони, вихідний сигнал яких від'ємний, не приймають участі у розрахунках, а для інших потрібно виконати лише лінійні обчислення.

У роботі [14] було проведено дослідження впливу методів навчальної ініціалізації ваг та функції активації на розповсюдження сигналу у мережі як в прямому так і в зворотньому напрямках. Відповідно до дослідження, використання використання логістичної сигмоїдальної функції разом з початковою ініціалізацією ваг погано підходить для створення глибоких нейронних мереж, тому що призводить до швидкого насичення. Функція активації гіперболічний тангенс не має такої проблеми через свою симетричність (середнє значення функції 0, а область значень –  $(-1, 1)$ ). У роботі [26] автори запропонували використовувати у нейронних мережах спеціальні шари пакетної нормалізації (batch normalization), що дозволяють підвищити якість навчання нейронної мережі. У [33] встановлено, що алгоритм зворотнього розповсюдження помилки сходиться швидше, якщо вхідні дані нормалізовані. Однак при розповсюдженні сигналу по нейронній мережі його маточікування та дисперсія міняються, причому інколи дуже значно, що негативно сказується на процесі навчання.

Компанія Google у 2014 році запропонувала нову архітектуру згорткових нейронних мереж Inception [51]. У цій архітектурі мережа будується з набору блоків, що містять комбінацію операцій згортки та підвибірки різної розмірності. Такий підхід дозволяє уникнути перенавчання, а також знизити кількість параметрів мережі, які потрібно навчати, що знижує час навчання мережі. На основі запропонованої архітектури була створена та навчена мережа GoogLeNet [51], що складається з 22 шарів. Мережа використовується для задач комп'ютерного зору. Мережа використовується для задач комп'ютерного зору. Згодом Google запропонувала ще декілька

варіантів блоків архітектури Inception та мереж на її основі [52, 53]. Збільшення кількості шарів у нейронних мережах, навіть з використанням напівлінійних функцій активації, нормалізації початкових значень ваг та шарів нормалізації, не завжди призводить до покращення якості навчання. При цьому причина не в проблемі зникаючого градієнту, а у збільшенні помилки навчання мережі при збільшенні кількості шарів [22, 23]. Для вирішення цієї проблеми компанія Microsoft у 2015 році запропонувала нову архітектуру та підхід до навчання нейронних мереж – залишкове навчання (residual learning). Архітектура використовує той факт, що нейронну мережу завжди можна зробити більш глибокою без зниження якості роботи шляхом додавання декількох шарів, які не змінюють сигнал.

В наш час створено велика кількість програмних систем для навчання глибоких нейронних мереж [6, 32, 27]. Серед найбільш популярних з них можна відмітити Caffe, Theano, TensorFlow, Torch и CNTK. TensorFlow [1] створена компанією Google у 2015 році та включає системи ефективної роботи з тензорами та потокової обробки даних на графі. У доповненні до описаних вище систем можна відмітити також бібліотеку Keras [8], яка представляє зручний та простий у використанні програмний інтерфейс для навчання глибоких нейронних мереж. Keras не є самостійною системою, а працює поверх Theano, TensorFlow або CNTK. У 2016 році Keras включили у склад TensorFlow.

Методи машинного навчання дозволяють адаптувати свою структуру шляхом обробки вибірки даних. Вони розширюють можливості математичного моделювання як за допомогою врахування результатів обчислювальних експериментів, так і натурні експерименти. Перший крок в машинному навчанні для прогнозування поведінки в інженерних конструкціях – збір даних. Якість і кількість даних суттєво впливають на результат моделювання. Вихідними даними можуть бути результати натурних та обчислювальних експериментів. Результати всіх експериментів можна комбінувати в єдину вибірку для уточнення методів[85].

Зі зростанням популярності штучних нейронних мереж збільшується кількість досліджень щодо розробки моделей та застосування їх можливостей у задачах механіки деформованого твердого тіла. Наприклад, у роботах [2, 9, 37] досліджено можливості машинного навчання для розв'язання задач механіки руйнування. Зокрема, у роботі [2] для навчання нейронної мережі прогнозувати можливі зони руйнування балок використано дані 64 обчислювальних експериментів та 3 натурних. У роботі [9] розроблено нейронну мережу, яка базується на фільтрі Калмана (англ. Kalman filter), для прогнозування руйнування швидкісної дороги на мосту шляхом обробки даних про температуру та коливання. У роботі [37] для прогнозування детектування руйнування за даними про вібрації розроблено модель на основі самоорганізаційної карти Кохонена (англ. Self-organizing map). У роботі [3] досліджено можливості нейронних мереж для прогнозування максимальних переміщень у залізничних балках. Модель нейронної мережі побудовано як функцію двох змінних: фрикційного параметру та швидкості навантаження. Для навчання використано 663 точки, що дозволило отримати максимальну похибку відносно скінченно-елементної моделі у 5,4%. Модель нейронної мережі для прогнозування критичного навантаження (у контексті втрати стійкості) двотаврової балки з отворами розроблено у роботі [4]. Розроблена модель враховує сім геометричних параметрів і дозволяє обчислити критичне навантаження з точністю у межах від 3,7% до 0,4% за час, менший однієї мілісекунди. Спосіб підвищення точності та ефективності обчислення функцій обмежень у задачах оптимального проектування конструкцій на базі апроксимації похибки розв'язання системи диференціальних рівнянь запропоновано у роботі [77]. Модель нейронної мережі використовується для обчислення похибки аналітико-чисельного алгоритму обчислення довговічності балочної конструкції відносно скінченно-елементного рішення. Покроковий алгоритм донавчання нейронної моделі, який дозволяє розв'язувати задачі динаміки будівельних конструкцій, розроблено у роботі [86]. Розроблений алгоритм дозволяє прогнозувати напружно-деформований

стан конструкції, яка знаходиться під дією тривалого або короткочасного навантаження.

В [60, 61] за допомогою традиційного підходу будуються нейромережеві моделі, що дозволяють вирішувати прямі і зворотні задачі будівельної механіки, теорії пружності і теорії оболонок на прикладах розрахунку циліндричних резервуарів, сферичних оболонок і плит, наведені підходи до вирішення завдань оптимізації. Результати нейромережевого рішення в [60, 61] (похибка становить від 1 до 5%) можна пояснити тим, що нейромоделі для цих завдань є одновимірною і однопараметричною, що істотно спрощує рішення. Крім того, функція рішення є гладкою і безперервною, для якої можна легко підібрати функції активації. Робота [38], присвячена використанню нейромережевих методів екстраполяції при розрахунку і проектуванні конструкцій, крім вихідної інформації використовуються корисні відомості про додаткову інформацію в невеликій суміжній зоні і здійснюється донавчання мережі з урахуванням нових даних, після чого один раз виконується нейромережева екстраполяція. Для вирішення завдань механіки, теорії пластин і оболонок використовуються крокові нейромережеві технології з донавчанням моделі, але без зміни її структури [63, 64].

Нейромережеві технології рішень, в тому числі нейропрогнозування, нейрооптимізація, нейроуправління, повинні ґрунтуватися не тільки на можливості навчання нейромоделі на початкових даних, але і на модифікації її на додаткових, що надходять в процесі нових даних, на здатності самонавчання, самоорганізації, самовдосконалення. Таке багатокрокове донавчання на відміну від традиційного підходу призводить до підвищення точності рішення і достовірності результатів, що підтверджено численними прикладами розв'язання задач з області математики, механіки, теорії пластин і оболонок, будівельних конструкцій [63-68]. Зазвичай натурні випробування закінчуються розробкою отриманих результатів. В [89, 62, 90] показано, що на їх базі за допомогою крокової нейротехнології прогнозування можна зробити теоретичний прогноз подальшої роботи конструкції при різних видах

навантаження, аж до руйнування. У результаті кроковий нейропрогноз дає можливість суттєво знизити терміни випробувань і заощадити кошти на проведення робіт. На базі крокових нейромережових технологій в [63, 64] реалізований новий підхід до вирішення проблем оптимізації на основі гібридних нейромережових програм, що з'єднують кращі якості традиційного програмування та нейромережової апроксимації. Встановлено універсальність методики нейромережової оптимізації.

### **1.3 Аналіз генетичних методів у задачах моделювання напружено-деформованого стану конструкцій**

Перші публікації щодо використання принципів біологічної еволюції для розв'язання оптимізаційних задач з'явилися ще у 1960-х роках. А в 1975 році вийшла знакова робота Дж. Холланда «Адаптація в природних і штучних системах» [24], де власне і був запропонований перший генетичний алгоритм.

Генетичні алгоритми ґрунтуються на принципах біологічної еволюції і генетики. Для їх опису активно використовують біологічні терміни [75]. Особина – потенційний розв'язок задачі; популяція – набір особин; нащадок – як правило, покращена копія потенційного розв'язку (батька); придатність – зазвичай, якісна характеристика розв'язку. Хромосома – закодована структура даних особини у вигляді масиву фіксованої довжини. В найпростішому випадку – бінарний рядок фіксованої довжини. Ген – елемент цього масиву.

Загалом робота генетичного алгоритму полягає в одночасній обробці елементів множини альтернативних розв'язків і складається з наступних етапів [75]:

1. Генерація початкової популяції;
2. Обчислення значень функції активації кожної її хромосоми;

3. Застосування генетичних операторів (селекції, кросоверу, мутації) поки не буде виконано правило завершення генетичного алгоритму;
4. Формування нової популяції і перехід до п.2.

Теорію еволюційних, зокрема, генетичних алгоритмів на даний час добре розроблено, наприклад, у роботах [78, 97, 103, 102, 56], в яких формалізовано основні поняття та описано приклади застосувань до задач дискретної оптимізації. Оскільки генетичні алгоритми є метаевристичними, досягнення глобального оптимуму неможливо гарантувати, однак ці підходи добре зарекомендували себе у низці задач дискретної математики. У роботі [55] досліджено можливість застосування не тільки генетичних, а й інших евристичних алгоритмів для пошуку оптимальної архітектури нейронних мереж.

Можливі різні варіанти реалізації стандартної схеми генетичних алгоритмів, які відрізняються операторами схрещування, мутації та селекції, процедурами вибору батьків для схрещування, формою представлення хромосом (двійкове чи дійсне кодування) і т. ін. [30, 17]. Вибір того або іншого варіанту реалізації стандартних складових при розробці ГА для розв'язання конкретної задачі залежить від ряду факторів: особливостей предметної області, формалізації задачі, структури даних та ін.

Ефективність роботи генетичного алгоритму істотно залежить від способу кодування генів, складу початкової популяції, використовуваних генетичних операторів, параметрів генетичного алгоритму, таких як розмір популяції, кількості відібраних хромосом при селекції і для кросоверу, імовірності використання генетичних операторів. Центральне місце в генетичних алгоритмах належить генетичним операторам, з яких найбільш важливими є селекція (відбір), кросовер (схрещування) і мутація.

Генетичні оператори працюють аналогічно оператору в класичному математичному розумінні, відображаючи одну множину об'єктів на іншу, і представляють один з кроків генетичного алгоритму. У роботах [24, 21, 31, 48] описано велику кількість різних операторів.

Використання оператора селекції зберігається деяка кількість хромосом з кращими значеннями придатності на кожній ітерації генетичного алгоритму. Найчастіше використані механізми відбору [48] – елітна селекція, яка полягає у виборі заданої кількості хромосом поточної популяції з кращими значеннями придатності. Пропорційна селекція (Fitness-Proportionate Selection), або селекція на основі рулетки (Roulette Wheel Selection), де спочатку обчислюється придатність кожної хромосоми в популяції. Далі кожній хромосомі популяції ставиться у відповідність сектор колеса рулетки, пропорційний величині її пристосованості. Тоді при повороті колеса рулетки хромосома з більшою пристосованістю матиме велику імовірність вибору для подальшого схрещування. Турнірна селекція (Tournament selection). Ця селекція полягає у випадковому виборі з популяції хромосом і відборі з цієї групи в нову популяцію однієї хромосоми з найкращим значенням придатності. Так повторюється певну кількість разів. Турнірна селекція стала основним методом селекції в генетичних алгоритмах з різних причин. По-перше, вона нечутлива до особливостей обчислення придатності. По-друге, дуже проста, не потребує попередньої обробки і може бути легко розпаралелена. По-третє, вона налаштовується: розмір турніру визначає тиск селекції.

Оператор кросоверу породжує нащадків шляхом обміну частинами батьківських хромосом. При цьому зазвичай кросовер застосовується з заданою імовірністю до пари хромосом-батьків з поточної популяції, утворюючи дві хромосоми – нащадки, проте існують й інші варіанти [21, 72]. Одноточковий кросовер (One-point crossover). З популяції випадково вибираються дві батьківські хромосоми, і довільним чином генерується точка розриву. Далі батьківські хромосоми обмінюються частинами, утворюючи двох нащадків. Двоточковий кросовер (Two-point crossover) відрізняється від одноточкового тим, що випадковим чином генерується не одна, а дві точки розриву, і хромосоми обмінюються ділянками між цими двома точками [12, 31]. Рівномірний кросовер (Uniform crossover). Випадково створюється

хромосома-маска – рядок з нулів і одиниць, довжина якої дорівнює довжині хромосоми. Два нащадки утворюються за таким правилом – одиниця в  $i$ -й позиції маски означає, що  $i$ -й елемент першого батька слід розташувати на  $i$ -му місці першого нащадку, а нуль означає, що  $i$ -й елемент другого батька слід розташувати на  $i$ -му місці першого нащадку. Вважаючи першого батька другим, а другого – першим, за описаним правилом отримують другого нащадку. Маска може бути одна на всіх або своя для кожної пари батьків [12, 48]. Роль мутації полягає в запобіганні передчасної збіжності генетичного алгоритму і його “застрягання” в локальному оптимумі. Оператор мутації вносить випадкові зміни до генів хромосоми, додаючи новий генетичний матеріал в популяцію. Вибір процедури залежить, перш за все, від обраного способу кодування. Цей оператор може застосовуватися з заданою ймовірністю до всієї хромосоми або до кожного її гена. Існує декілька видів операторів мутації. Бітова мутація [24, 72]. Кожен ген хромосоми мутує із заданою малою ймовірністю. Багатоточкова мутація [24, 31]. У хромосомі змінюються значення довільно вибраних генів, кількість яких може бути заданою або випадковою, але не перевищує довжини хромосоми. У разі, коли в хромосомі змінюється тільки один випадково вибраний ген, отримуємо одноточкову мутацію [72, 48]. Слід зазначити, що у багатьох задачах одноточкова мутація не здатна вивести генетичний алгоритм з локального оптимуму.

Генетичні алгоритми – це потужний і гнучкий інструмент розв’язання складних оптимізаційних задач. Ідея генетичного алгоритму полягає у комп’ютерній організації еволюційного процесу створення, модифікації та відбору кращих розв’язків. Перевагою генетичних алгоритмів є пошук найкращих розв’язків, який здійснюється не з єдиної точки, як у більшості традиційних методів оптимізації, а з цілої множини точок – популяції особин.

Сьогодні генетичні алгоритми широко використовуються для розв’язання багатьох задач, про що свідчить велика бібліографія [79, 101, 70, 98-100].

Одним із сучасних підходів до моделювання складних нелінійних систем є блок-орієнтований підхід, в якому система розділяється на декілька певним чином з'єднаних підсистем, серед яких є, принаймні, одна нелінійна і одна лінійна підсистеми [12]. При моделюванні системи, що складається з двох послідовно з'єднаних підсистем, залежно від того, яка підсистема (лінійна або нелінійна) приймає вхідні сигнали, розрізняють моделі Гаммерштейна і Вінера. Структурно-параметрична ідентифікація таких систем є непростим завданням і часто потребує застосування методів глобальної оптимізації, зокрема генетичного алгоритму. Наприклад, у [59] здійснюється ідентифікація моделі Гаммерштейна, яка складається з одного вхідного статичного нелінійного блоку і динамічної лінійної стаціонарної підсистеми. У цій моделі невідома нелінійна статична частина, яку слід оцінити, представлена мережею радіально базисних функцій, структура якої оптимізується із застосуванням генетичного алгоритму. В [41] розглянуто ідентифікацію моделі Вінера, в якій лінійна динамічна частина передуює нелінійній статичній частині, представленій зворотною кусково-лінійною функцією, параметри якої оцінюються алгоритмом генетичного алгоритму. В [58] моделюється сила пальця на основі даних сигналів поверхневої електроміографії для покращення управління протезом руки. Сила пальця і дані сигналів генеруються, якщо суб'єкт виконує випадкові рухи безіменним пальцем для імітації різних рівнів сили. Подальша обробка сигналів виконується з використанням просторової фільтрації. Деякі маски просторового фільтру для фільтрації сигналів оптимізуються з використанням генетичного алгоритму.

Успішне застосування генетичного алгоритму в онлайн-ідентифікації для прямої оцінки полюсів і нулів динамічної системи стає можливим, оскільки генетичний алгоритм не потребує лінійності за цими параметрами [42]. Генетичний алгоритм використовується для ідентифікації параметрів лінійних і нелінійних систем для мінімізації помилки апроксимації за наявності шуму [43]. Під час моделювання було виявлено, що конвергенція

процесу дуже чутлива до механізму селекції і таких параметрів генетичного алгоритму, як розмір популяції, імовірність мутації і кросоверу. Для подолання цієї проблеми деякі з цих параметрів розглянуто як конструктивні параметри генетичного алгоритму, закодовані в хромосомі, які оптимізуються шляхом еволюційного процесу.

У роботі [71] для знаходження наближеного розв'язку нелінійної крайової задачі у вигляді аналітичного виразу пропонується підхід, що ґрунтується на застосуванні генетичних алгоритмів для визначення оптимальних значень параметрів вказаного виразу.

Можливі різні варіанти реалізації стандартної схеми генетичних алгоритмів, які відрізняються операторами схрещування, мутації та селекції, процедурами вибору батьків для схрещування, формою представлення хромосом і т. ін. [98, 99]. Вибір того або іншого варіанту реалізації стандартних складових при розробці генетичного алгоритму для розв'язання конкретної задачі залежить від ряду факторів: особливостей предметної області, формалізації задачі, структури даних та ін.

Одним із актуальних питань застосувань нейронних мереж є їхня структурна оптимізація, тобто, вибір оптимальної кількості шарів, нейронів, функцій активації тощо. Така оптимізація може проводитися як вручну, за умови відносно невеликої кількості параметрів, так і в автоматичному режимі. Генетичні алгоритми досить давно використовуються для задач структурної оптимізації нейронних мереж [45], однак, вони менше застосовувались для підбору параметрів ансамблів мереж.

Теорію еволюційних, зокрема, генетичних алгоритмів на даний час добре розроблено, наприклад, у роботі [78, 97] формалізовано основні поняття та приклади застосувань до задач дискретної оптимізації.

Покроковий алгоритм донавчання нейронної моделі, який дозволяє розв'язувати задачі динаміки будівельних конструкцій, розроблено у роботі [91]. Розроблений алгоритм дозволяє прогнозувати напружено-деформований

стан конструкції, яка знаходиться під дією тривалого або короткочасного навантаження.

Для розв'язання задач проектування також використовують генетичні алгоритми. Наприклад, у роботі [91] для оптимального проектування будівельних конструкцій. У роботі [81] запропоновано метод розв'язання нелінійних крайових задач для звичайних диференціальних рівнянь другого порядку, який застосовує генетичні алгоритми для знаходження оптимальних значень параметрів наближених розв'язань. Проте його практичне застосування пов'язане з необхідністю емпіричного підбору деяких параметрів та великою кількістю ітерацій для отримання прийняттого результату.

Еволюційні алгоритми є основою сучасних евристичних комп'ютерних технологій оптимізації, навчання, моделювання, проектування й управління у найширшому значенні цих понять. Еволюційні алгоритми ґрунтуються на глибокій аналогії між біологічним генетичним кодом і комп'ютерним двійковим кодом. Ця аналогія дає змогу формалізувати біологічний еволюційний процес і застосувати ідею про природний добір для розв'язання найскладніших проблем у галузі природних і комп'ютерних наук, промисловості та бізнесу [20]. Під час побудови системи штучного інтелекту за допомогою еволюційного підходу основна увага зосереджена на побудові початкової моделі, і правилах, за якими вона може змінюватися (еволюціонувати).

Питання ефективного застосування генетичних алгоритмів є актуальним, адже можливості їх застосування є досить широкими. Так, у роботі [76] було запропоновано підхід до використання генетичних алгоритмів при обробці експериментальних даних, для розв'язання оберненої задачі еліпсометрії. Показані переваги застосування саме генетичного алгоритму перед методами чисельного розв'язку, та методом крокової мінімізації функціоналу. В роботі [104] було розглянуто застосування генетичного алгоритму у задачах прогнозування, а саме застосування алгоритму під назвою

LGAP. Даний алгоритм призначений для вилучення закономірностей (знань) з табличних даних і використання цих знань для прогнозування майбутніх подій. Було встановлено, що даний алгоритм також доцільно використовувати для вирішення таких прикладних задач, як прогноз результатів діяльності підприємств, прогноз врожайності, прогноз динаміки стану пацієнтів. А в роботі [82] доведені переваги застосування генетичного алгоритму в задачах прогнозування забруднення довкілля, використовуючи менший об'єм необхідної інформації, при цьому не значно втрачаючи в якості і точності прогнозу. В роботі [108] було розглянуто використання генетичних алгоритмів в задачах пофарбування графів. Було встановлено, що генетичні алгоритми дають змогу розфарбувати граф навіть з великою кількістю вершин (наприклад, пофарбувати граф  $G$ , що має 25 вершин, тобто матриця  $5 \times 5$ ).

У роботі [81] для оптимального проектування будівельних конструкцій використовують генетичні алгоритми. У роботі [20] запропоновано метод розв'язання нелінійних крайових задач для звичайних диференціальних рівнянь другого порядку, який застосовує генетичні алгоритми для знаходження оптимальних значень параметрів наближених розв'язань. Проте його практичне застосування пов'язане з необхідністю емпіричного підбору деяких параметрів та великою кількістю ітерацій для отримання прийняттого результату.

#### **1.4 Висновки до розділу 1**

У даному розділі було розглянуто та проаналізовано сучасний стан наявних досліджень у напрямку роботи.

З аналізу літературних джерел можна зробити висновок, що задача прогнозування зон руйнування в інженерних конструкціях з використанням комп'ютерного навчання є досить актуальною.

На основі методології, можна припустити, що метод скінчених елементів може поєднуватися з розробкою нейронних мереж, що дасть кращі результати не лише для прогнозування зон руйнування в інженерних конструкціях, а також для багатьох інших системи, що використовуються в техніці.

Генетичні алгоритми мають широку область застосування, та можуть вирішувати будь-які оптимізаційні задачі. Особливу ефективність генетичні алгоритми показують в вирішенні задач, де повний перебір є надто довгим, або неможливим. Вони характеризуються високою швидкістю роботи (особливо для комбінаторних завдань); хоча генетичні алгоритми не є максимально точним методом розв'язання завдань. Проте, генетичні алгоритми є ефективним і перспективним засобом розв'язку оптимізаційних задач, а особливо тих, де необхідно знайти глобальний екстремум деякої функції.

Генетичні алгоритми є потужним засобом розв'язання різноманітних складних задач глобальної оптимізації і моделювання, які часто або не можуть бути розв'язані іншими методами, або розв'язуються ними з незадовільною точністю та/або швидкістю. Ефективність генетичного алгоритму істотно залежить від того, наскільки якісно буде проаналізовано проблему, що безпосередньо впливає на подальший вибір виду генетичного алгоритму, його операторів, способу кодування потенційних розв'язків та інших характеристик. Їх можна застосовувати самостійно, адаптуючись під конкретне завдання, або удосконалювати існуючі методи розв'язання, утворюючи гібридні системи. Генетичні алгоритми мають велику перспективу і потребують подальшого удосконалення, розроблення і більш загального теоретичного обґрунтування.

Отже, переважну більшість розглянутих публікацій орієнтовано на розробці моделей нейронних мереж для прогнозування. Практично на даний момент відсутній підхід для прогнозування зон руйнування в інженерних конструкціях.

## 2 ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ НАПРУЖЕНО-ДЕФОРМОВАНОГО СТАНУ ПЛАСТИН

### 2.1 Нейромережеве моделювання напружено-деформованого стану плоских об'єктів

Необхідність дослідження поведінки інженерних конструкцій виникає у різних галузях таких як будівництво, автобудівництво, ракетне будівництво та інші. Машинне навчання дозволяє отримати моделі для миттєвої (відносно, наприклад, методу скінчених елементів) оцінки параметрів стану об'єкту. Таким чином, розробка методів машинного навчання для прогнозування поведінки інженерних конструкцій є актуальною.

Класичні методи математичного моделювання дозволяють робити прогнозування з потрібною точністю, але підготовка математичних моделей та проведення відповідних обчислювальних експериментів можуть бути довготривалими за часом. Можливо альтернативою є розробка методів і моделей машинного навчання, які дозволяють робити швидкі оцінки необхідних параметрів стану об'єкту. Методи машинного навчання дозволяють адаптувати свою структуру шляхом обробки вибірки даних. Вони дозволяють розширити можливості математичного моделювання за допомогою врахування результатів обчислювальних та натурних експериментів. З практичної точки зору моделі машинного навчання для прогнозування значень параметрів стану конструкцій можуть слугувати інтерактивними асистентами в процесі проектування.

В авіаційній техніці та суднобудівництві розповсюджено використання призматичних тіл, які також можуть мати отвори, у яких один розмір (товщина) значно менший двох останніх. Такі тіла можна моделювати

пластинами та оболонками. Задача прогнозування зон руйнування в інженерних конструкціях полягає у тому, щоб визначити наружено-деформований стан під заданим навантаженням. Моделі, основані на мережах штучних нейронів повинні враховувати геометричні та фізико-механічні параметри тіла, а також умови закріплення. Задачі прогнозування параметрів стану об'єкта за його геометричними та фізико-механічними параметрами можна віднести до класу задач регресії.

Схематично нейронна мережа складається з шару вхідних нейронів, вихідного шару та декількох внутрішніх прихованих шарів. Типова схема нейронної мережі має вигляд (рис. 2.1):

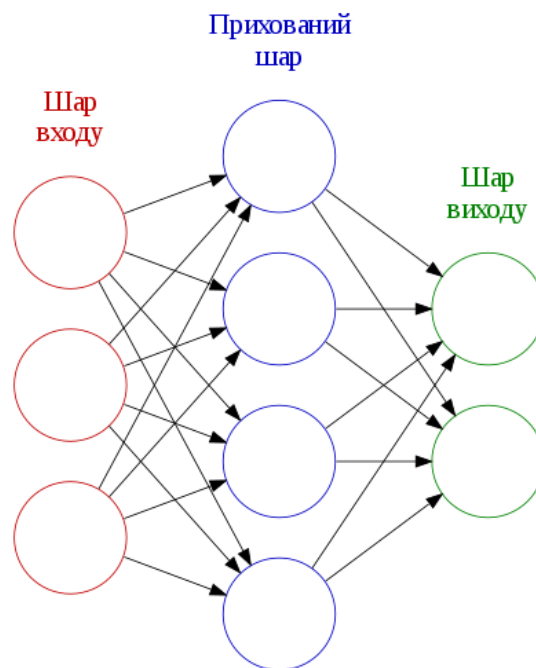


Рисунок 2.1 – Загальна схема нейронної мережі

Де вхідний шар отримує геометричні та механічні характеристики. Внутрішні шари дозволяють моделювати нелінійні зв'язки між вхідними параметрами. Вихідний шар містить кількість нейронів, що відповідає кількості невідомих.

Штучний нейрон являє собою вузол штучної нейронної мережі, що є спрощеною моделлю природного нейрона. Математично нейрон

представляється у вигляді функції. Нейрон проводить розрахунок зваженої суми значень на своїх входах, додає зміщення і визначає, слід виключити дане значення або використовувати далі [69].

Функція, у вигляді якої представляються нейрони – називається функцією активації. Функція активації визначає вихідне значення нейрона в залежності від результату зваженої суми входів і порогового значення.

Головна властивість функції активації – її нелінійність [83]. Якщо використовувати лінійну функцію, тоді можливо вирішувати лише певний тип задач, де залежність між вхідними та вихідними даними описується лінійною функцією, а також збільшення кількості прихованих шарів не підвищує ефективність моделі, так як композиція лінійних функцій – це лінійна функція.

На практиці неможливо точно вказати межу, після якої відбувається активація нейрону. Для визначення моменту активації нейрону використовують функції активації. Функція перевіряє значення нейрону, та визначає повинні зовнішні зв'язки розглядати його як активований або його можна ігнорувати.

Деякі властивості функції активації включають:

1. нелінійна – коли передавальна функція нелінійна, то, як доведено, двошарова нейронна мережа є універсальною апроксимацією функцій[118]. Тотожна передавальна функція не має такої властивості. Коли декілька шарів використовують тотожну передавальну функцію, тоді вся мережа еквівалентна одношаровій моделі;

2. неперервна диференційовність – ця властивість бажана *property is desirable* (RELU не є неперервно диференційовною і має неоднозначне рішення для оптимізації заснованій на градієнті) для використання методів оптимізації заснованих на градієнті. Передавальна функція двійковий крок не диференційовна у 0, але диференційовна в усіх інших значення, що є проблемою для методів заснованих на градієнті[119];

3. область визначення;

4. монотонність;

5. гладка функція з монотонною похідною;
6. наближення до тотожної функції  $f(x) = x$  в початку координат.

Функції активації можуть бути різного виду [106]:

- сигмоїдна;
- ReLU;
- Exponential linear unit (ELU);
- Scaled exponential linear unit (SELU);
- SoftExponential
- тангенс гіперболічний.

Задача функції активації – допомогти прийняти локальне рішення у кожному з нейронів. Наприклад, функція типу сигмоїда відображає значення на виході з нейрону у щось більше чи менше нуля. Сигмоїда – гладка монотонно зростаюча нелінійна функція:

$$A = \frac{1}{1+e^{-x}}. \quad (2.1)$$

Сигмоїда не лінійна за своєю природою, а результатом комбінацій набору таких функцій має нелінійний характер, що дозволяє будувати мережу з декількох шарів. Головною перевагою даної функції є те, що вона не являється бінарною. Для сигмоїди також характерний гладкий градієнт (рис 2.2) [83, 11].

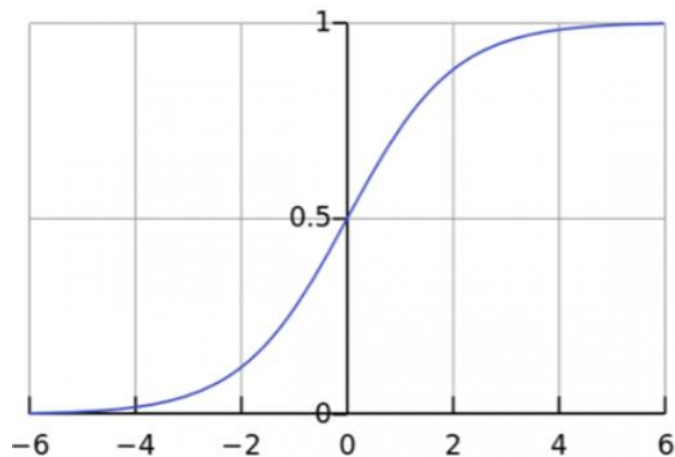


Рисунок 2.2 – Загальний вигляд сигмоїди

В діапазоні значень  $X(-2; 2)$  значення  $Y$  швидко змінюються. Це означає, що будь-яка невелика зміна значень  $X$  в цій області сприяє значним змінам  $Y$ . Така поведінка функції вказує на те, що  $Y$  має тенденцію прагнути до одного з країв діапазону. Інша перевага сигмоїди над лінійною функцією є в тому, що маємо фіксований діапазон значень функції –  $[0; 1]$  [83]. Така властивість сигмоїди дуже корисна, так як не призводить до помилок у випадку великих значень порогу активації. На даний момент сигмоїда є однією з найбільш часто вживаних активаційних функцій у нейромережах. Але вона має певні недоліки. При наближенні до границь діапазону, значення  $Y$  починають слабо реагувати на зміни  $X$ . це означає, що градієнт в таких областях має невеликі значення. В такому випадку значення градієнту дуже мале та прагне до 0. Нейромережа перестає навчатися або робить це дуже повільно. Але існує безліч варіантів для вирішення цієї проблеми. Незважаючи на ці недоліки дана функція дуже популярна та часто використовується для вирішення задач класифікації [28].

Ще однією функцією активації є гіперболічний тангенс (рис. 2.3) [29]:

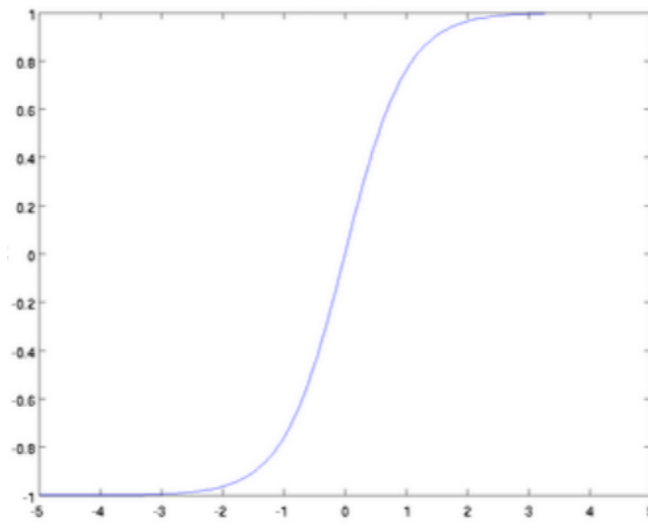


Рисунок 2.3 – Загальний вигляд графіку гіперболічного тангенсу

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1. \quad (2.2)$$

Гіперболічний тангенс є скорегованою сигмоїдною функцією:

$$\tanh(x) = 2 \operatorname{sigmoid}(2x) - 1. \quad (2.3)$$

Дана функція має ті ж характеристики, що і сигмоїда, розглянута вище. Вона також має нелінійний характер і добре підходить для комбінації шарів, а діапазон значень функції знаходиться у діапазоні  $(-1; 1)$ . Вузкий діапазон значень запобігає перезавантаженню активаційної функції. Але варто відмітити, що градієнт даної функції більше ніж у сигмоїди. Рішення вибору функції залежить від вимог до амплітуди градієнту. Також, як і в сигмоїді, гіперболічному тангенсу присуща проблема зникнення градієнту [29]. Тангенс також є дуже популярною та часто вживаною функцією активації.

За останні роки великої популярності набула функція активації ReLU (rectified linear unit, рис. 2.4). Її похідна дорівнює або одиниці, або нулю, і тому не може статися розростання або загасання градієнтів. Більш того, використання даної функції приводить до проріджування ваг. ReLU має наступну формулу [83]:

$$\sigma(x) = \max(0, x). \quad (2.4)$$

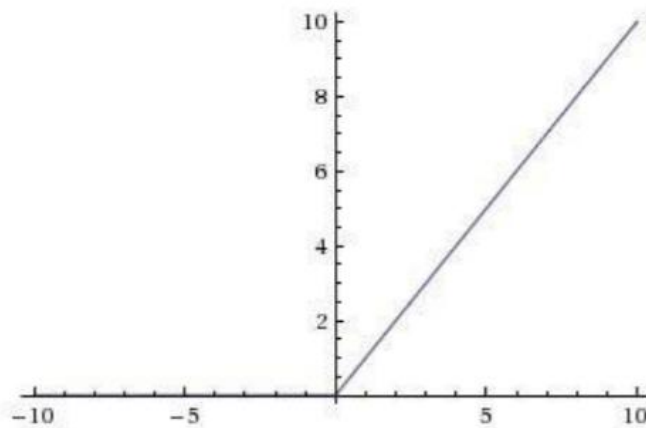


Рисунок 2.4 – Загальний вигляд функція ReLU

Переваги функції ReLU в наступному [83]:

1. Сигмоїда і гіперболічний тангенс потребує для виконання операцій велику кількість системних ресурсів, таких як піднесення до степеню, що при великій кількості шарів та нейронів сповільнюють процес навчання, в той час як ReLU може бути реалізована за допомогою простого порогового перетворення матриці активацій в нулі.

2. Застосування ReLU істотно підвищує швидкість збіжності стохастичного градієнтного спуску в порівнянні з сигмоїдою і гіперболічним тангенсом. Вважається, що це обумовлено лінійним характером і відсутністю насичення даної функції.

Але функції ReLU не завжди достатньо надійні і в процесі навчання. Наприклад, великий градієнт, що проходить через ReLU, може привести до такого оновлення ваг, що даний нейрон ніколи більше не активується. Якщо це станеться, то, починаючи з даного моменту, градієнт, що проходить через цей нейрон, завжди буде дорівнює нулю.

В даний час існує декілька різновидів даної функції [83]:

1. ReLU з втратами є спробою вирішити проблему, яка пов'язана з втратами звичайної являє собою одну зі спроб вирішити виходу з ладу звичайних. ReLU на інтервалі на своєму виході дає 0, а LReLU на цьому інтервалі має невелике від'ємне значення (кутовий коефіцієнт близько 0,01). Тобто функція для LReLU має вигляд при  $x < 0$  і  $f(x)$  при  $x \geq 0$ , де  $\alpha$  - мала константа. Деякі дослідники повідомляють що успішно застосували дану функцію, але результати не завжди стабільні.

2. Для ReLU з параметрами – кутовий коефіцієнт не задається на початку, а обраховується на основі даних які отримуються під час. Процес зворотного поширення помилки і оновлення для PReLU досить простий і подібний до процесу для традиційних ReLU.

Кожна функція активації має свої властивості, переваги та недоліки. Жодна з функцій не є універсальною, неможливо однозначно сказати в якому випадку краще використовувати слід використовувати ReLU, сигмоїду або

тангенс гіперболічний неможливо. Знаючи деякі характеристики функції, яку потрібно апроксимувати, потрібно вибрати таку активаційну функцію, яка апроксимує максимально точно та приведе до швидкого навчання. Необхідно вибрати ту функцію, з якою процес навчання та збіжність будуть найшвидшими.

Існує певна кількість оптимізаційних алгоритмів нейронних мереж [115]: SGD (стохастичний градієнт); Adam (метод адаптивної оцінки моментів); L-BFGS (квазіньютонівський алгоритм Бройде – Флетчера – Гольдфарба – Шанно з обмеженим використанням пам'яті), Adagrad, Adadelat, Adamax, Momentum, Nesterov momentum, RMSProp. Найбільш використовуваним методом навчання нейронних мереж є алгоритм зворотного поширення помилки (Backpropagation algorithm), в якому мінімізація цільової функції проводиться методом пакетного градієнтного спуску (Batch gradient descent) [115].

Adagrad – це алгоритм для оптимізації на основі градієнта, який виконує адаптацію швидкості навчання до параметрів, виконуючи менші оновлення (тобто низькі швидкості навчання) для параметрів, пов'язаних з ознаками, які часто зустрічаються і більші оновлення (тобто високі швидкості навчання) для параметрів, пов'язаних з рідкісними ознаками. З цієї причини цей оптимізатор добре підходить для роботи з розрідженими даними. Однією з головних переваг Adagrad є те, що він виключає необхідність ручного налаштування швидкості навчання. Основним недоліком Adagrad є його накопичення у знаменнику квадратичних градієнтів: оскільки кожен доданий термін є позитивним, накопичена сума продовжує зростати під час навчання. Це в свою чергу призводить до того, що швидкість навчання скорочується і в кінцевому підсумку стає нескінченно малою, і тоді алгоритм більше не може отримати додаткові знання.

RMSProp – модифікований метод AdaGrad, в якому накопичення градієнта замінене на експоненціально зважене ковзне середнє, тобто ігнорує значення після певного числа ітерацій.

AdaDelta – розширений метод AdaGrad, який дозволяє усунути проблеми розпаду швидкості навчання. Замість того, щоб накопичувати всі попередні градієнти, Adadelata обмежує простір накопичених градієнтів деякими фіксованими розмірами. Замість неефективного зберігання попередніх градієнтів у квадраті, сума градієнтів рекурсивно визначається як середнє затухання всіх минулих квадратних градієнтів [116].

Adam базується на основі адаптивної оцінки моменту. Адаптивна оцінка моменту (Adam) – новий метод, який обчислює адаптивні швидкості навчання для кожного параметра.

В даний час адаптивні алгоритми часто є методами за замовчуванням в багатьох модулях для машинного навчання (наприклад, Scikitlearn, Keras). Однак існують дослідження [116], які підтверджують низьку продуктивність даних методів (Adam, RMSProp, Adagrad) на наборах даних з великою кількістю параметрів, в порівнянні з методом стохастичного градієнтного спуску. Даний недолік пояснюється тим, що алгоритми з адаптивною швидкістю навчання більш схильні до перенавчання, внаслідок чого їх похибка на тестовій вибірці може істотно підвищуватися.

Множина для навчання нейронної мережі може бути отримана декількома способами:

1. з аналітичних формул;
2. проведенням серії скінчено елементних експериментів;
3. експериментальним способом;
4. змішаним (гібридний підхід), наприклад при недостатній кількості експериментальних даних додатково генеруються дані методом скінчених елементів.

Навчання нейронних мереж зазвичай проводиться з певною кількістю епох. Епоха – одна ітерація в процесі навчання, що включає пред'явлення всіх прикладів з навчальної множини і, можливо, перевірку якості навчання на контрольному безлічі [117]. Процес навчання здійснюється на навчальній вибірці. Навчальна вибірка включає вхідні значення і відповідні їм вихідні

значення набору даних. В ході навчання нейронна мережа знаходить якісь залежності вихідних полів від вхідних. Таким чином, перед нами постає питання – які вхідні поля (ознаки) необхідно використовувати. Спочатку вибір здійснюється евристичний, далі кількість входів може бути змінено.

Для прогнозування поведінки плоских об'єктів розробляється модель, яка має декілька шарів нейронів: шари вхідних нейронів, нейрони вхідних шарів поєднуються з прихованими шарами нейронів, потім зовнішній шар нейронів з функцією активації формує регресійну оцінку вектору, що описує напружено-деформований стан об'єкту [118]. Процес навчання відбувається на навчальній вибірці, що складається з  $N$  кількості епох. Навчальна вибірка включає вхідні значення і відповідні їм вихідні значення набору даних. У процесі навчання нейронна мережа знаходить залежності вихідних полів від вхідних.

Для машинного навчання вихідну множину значень розділяють на дві частини: навчальну і тестову. Навчальна частина містить значення вхідних параметрів результатів, які використовують алгоритм для побудови дерев. Тестова частина, зазвичай, менша за навчальну, її значення використовуються для перевірки (у випадку регресійного аналізу) точності навчання. Точність регресійного аналізу оцінюють шляхом обчислення середнього квадратичного відхилення значень, обчислених для тестової частини у результаті машинного навчання, від вихідних значень.

## **2.2 Особливості генерації вибірки методом скінченних елементів**

В багатьох випадках, коли форма пластин відмінна від тривіальної, для визначення компонентів напружено-деформованого стану можна використовувати метод скінченних елементів на базі теорії Тимошенко-Міндліна другого порядку.

Для логічного конструювання неявних функцій, що відповідають складним областям, використовують системи R-функцій [102]. Одна з найпоширеніших на практиці систем R-функцій – система  $\mathfrak{R}_0$  має вигляд [102]:

$$\begin{cases} \neg x \equiv -x, \\ x_1 \wedge x_2 \equiv x_1 + x_2 - \sqrt{x_1^2 + x_2^2}, \\ x_1 \vee x_2 \equiv x_1 + x_2 + \sqrt{x_1^2 + x_2^2}, \end{cases} \quad (2.5)$$

де  $x_1$  і  $x_2$  – значення неявних функцій.

Ця система визначає логічні заперечення, кон'юнкцію та диз'юнкцію, операндами яких виступають неявні функції, що відповідають більш простим логічним примітивам.

Геометричну модель прямокутної пластини можна подати за допомогою R-функції у вигляді формули:

$$rectangle(x, y, w, h) = \left[1 - \left(\frac{2}{w}x\right)^2\right] \wedge \left[1 - \left(\frac{2}{h}y\right)^2\right], w > 0, h > 0, \quad (2.6)$$

де  $w$  – ширина,

$h$  – висота.

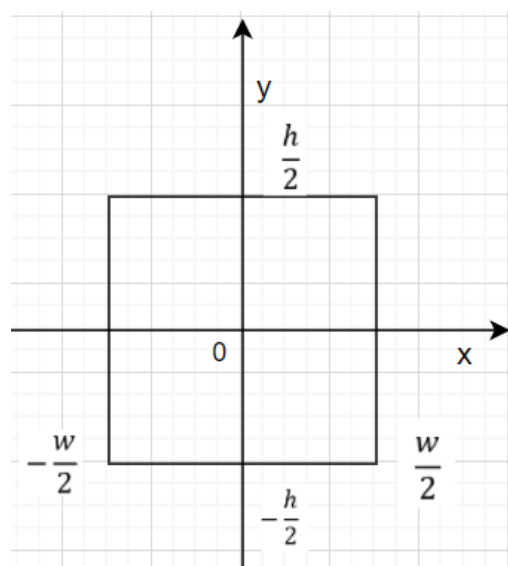


Рисунок 2.4 – Модель квадратної пластини

Аналогічним способом можна представити геометричну модель прямокутної пластини з круглим отвором (рис. 2.5):

$$rectangle(x, y, w, h, x_0, y_0, r) = \left[1 - \left(\frac{2}{w}x\right)^2\right] \wedge \left[1 - \left(\frac{2}{h}y\right)^2\right] \wedge \left[1 - \frac{(x-x_0)^2 + (y-y_0)^2}{r^2}\right], \quad w > 0, h > 0, r > 0, \quad (2.7)$$

де  $w$  – ширина,

$h$  – висота,

$r$  – радіус.

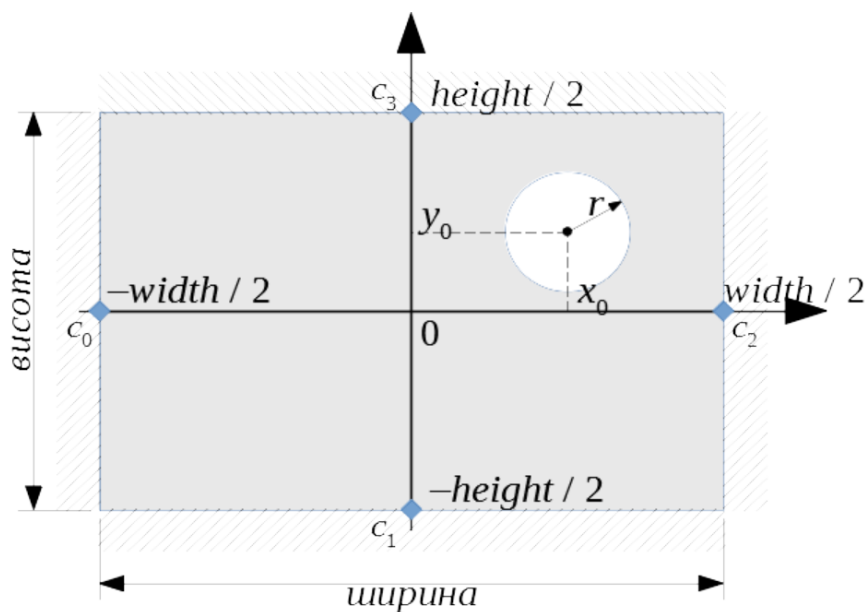


Рисунок 2.5 – Схема прямокутної пластини з круглим отвором

Також можна представити геометричну модель кільцевої пластини (рис. 2.6):

$$ring(x, y, R, r) = \left(1 - \frac{x^2 + y^2}{R^2}\right) \wedge \neg \left(1 - \frac{x^2 + y^2}{r^2}\right), \quad R > 0, r > 0, R > r. \quad (2.8)$$

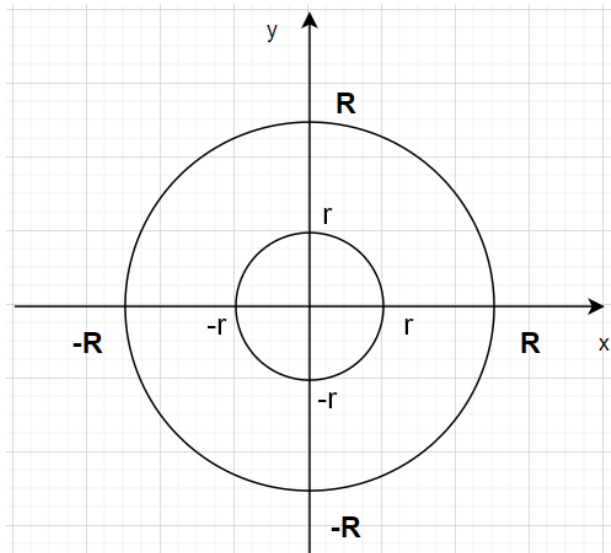


Рисунок 2.6 – Модель кільцевої пластини

Двовимірні та тривимірні скінченні елементи є досить ефективними в аналізі напружено-деформованого стану суцільних об'єктів. тривимірні скінченні елементи можна використовувати при розрахунку пластин, зменшуючи їх розмір у напрямку товщини. Але при їх використанні виникають певні труднощі. Наявність трьох ступенів свободи в кожному вузлі приводить до великих коефіцієнтів жорсткості для переміщень по товщині оболонки, що може бути причиною поганої обумовленістю системи рівнянь, якщо товщина пластини мала у порівнянні з іншими розмірами скінченних елементів [106]. Також використання декількох вузлів по товщині є досить неекономічним. Тому на практиці для дослідження напружено деформованого стану пластин використовують спеціальні скінченні елементи – елементи пластин. Теорія зсувних деформацій для пластин є розвитком положень теорії Тимошенко-Міндліна. Використовуючи стандартні гіпотези про малість переміщень та деформацій, а також про нестискуваність та збереження нормалей серединної поверхні, дана теорія враховує поперечні зсувні деформації [110]. Поле переміщень визначається відношенням:

$$\begin{cases} u(x, y, z) = z\theta_x(x, y), \\ v(x, y, z) = z\theta_y(x, y), \\ w(x, y, z) = w_0(x, y), \end{cases} \quad (2.7)$$

де  $w_0(x, y)$  – прогин точок серединної поверхні пластини;

$\theta_x(x, y)$  та  $\theta_y(x, y)$  – повороти нормалі відносно осей  $Ox$ ,  $Oy$ , відповідно.

Значення  $(w_0, \theta_x, \theta_y)$  – узагальнені переміщення.

Вектор деформацій визначається диференціюванням переміщень:

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} = z \begin{Bmatrix} \frac{\partial \theta_x}{\partial x} \\ \frac{\partial \theta_y}{\partial y} \\ \frac{\partial \theta_x}{\partial y} + \frac{\partial \theta_y}{\partial x} \end{Bmatrix}. \quad (2.8)$$

Вектор поперечних зсувних деформацій представлений у вигляді:

$$\{\gamma\} = \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \\ \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial w}{\partial x} + \theta_x \\ \frac{\partial w}{\partial y} + \theta_y \end{Bmatrix}. \quad (2.9)$$

Співвідношення між напруженнями та деформаціями для ізотропного матеріалу можуть бути представлені як:

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = [D]\{\varepsilon\}, \quad (2.10)$$

$$\{\tau\} = \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = [G]\{\gamma\}, \quad (2.11)$$

де  $E$  – модуль Юнга;

$\nu$  – коефіцієнт Пуассона;

$G$  – модуль зсуву.

Математичну модель пластинки довільної форми можна отримати з використанням теорії Тимошенко-Міндліна (теорія зсувних деформацій

першого порядку). Енергія деформацій скінченного елемента об'єму  $V_e$ , серединна площина якого збігається з  $xOy$ , має вигляд:

$$U_p = \frac{1}{2} d_e^T \int_{V_e} (z^2 B_f^T D_f B_f) dV_e d_e + \left( \frac{K}{2} d_e^T \int_{V_e} (B_c^T D_c B_c) \right) dV_e d_e, \quad (2.12)$$

де  $d_e^T = \{\omega_1 \theta_{x1} \theta_{y1} \dots \omega_m \theta_{xm} \theta_{ym}\}$  – вектор узагальнених переміщень ( $\omega$  – прогин серединної площини;

$\theta_x, \theta_y$  – обертання нормалі серединної площини відносно відповідних осей координат);

$B_f$  і  $B_c$  – матриці зв'язків між переміщеннями та деформаціями;

$D_f$  і  $D_c$  – матриці зв'язків між напруженнями та деформаціями; параметр  $K$  – коефіцієнт, що враховую нерівномірність розподілу дотичних напружень у перерізі (для ізотропного матеріалу  $K = \frac{5}{6}$ );

$m$  – кількість вузлів у скінченному елементі.

Матриці  $B_f$  і  $B_c$  для ізотропного матеріалу мають вигляд:

$$B_f = \begin{bmatrix} 0 & \frac{\partial N_1}{\partial x} & 0 & 0 & \frac{\partial N_m}{\partial x} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial y} & \dots & 0 & \frac{\partial N_m}{\partial y} \\ 0 & \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_m}{\partial y} & \frac{\partial N_m}{\partial x} \end{bmatrix}, \quad (2.13)$$

$$B_c = \begin{bmatrix} \frac{\partial N_1}{\partial x} & N_1 & 0 & \frac{\partial N_m}{\partial x} & N_m & 0 \\ \frac{\partial N_1}{\partial y} & 0 & N_1 & \frac{\partial N_m}{\partial y} & 0 & N_m \end{bmatrix}, \quad (2.14)$$

де  $N_i$  – функція форми  $i$ -го вузла скінченного елемента.

Матриці  $D_f$  і  $D_c$  відповідають плосконапруженому стану та мають вигляд:

$$D_f = \frac{E}{1-\vartheta^2} \begin{bmatrix} 1 & \vartheta & 0 \\ \vartheta & 1 & 0 \\ 0 & 0 & \frac{1-\vartheta}{2} \end{bmatrix}, D_c = \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix}, \quad (2.15)$$

де  $E$  – модуль Юнга;

$\vartheta$  – коефіцієнт Пуассона;

$G$  – модуль зсуву.

Після інтегрування по товщині отримаємо таку формулу для матриці жорсткості:

$$K_{ep} = \frac{h^3}{12} \int_{\Omega_e} (B_f^T D_f B_f) d\Omega_e + Kh \int_{\Omega_e} (B_c^T D_c B_c) d\Omega_e. \quad (2.16)$$

### 2.3 Метод нейромережевого моделювання напружено-деформованого стану квадратної пластини

Дана квадратна пластинка з ізотропного матеріалу, яка знаходиться під дією рівномірно розподіленого по поверхні поперечного тиску. Для розробки нейронної мережі для прогнозування напружено-деформованого стану введені наступні параметри з обмеженнями:

- $a \in [0.14; 1,6]$  (м);
- $h \in [0.01; 0.1]$  (м);
- $E \in [50000; 300000]$  (МПа);
- $\vartheta \in [0; 0,45]$ ;
- $q \in [0.01; 0.1]$  (МПа),

де  $a$  – довжина пластини;

$h$  – товщина пластини;

$E$  – модуль пружності матеріалу;

$\vartheta$  – коефіцієнт Пуассона;

$q$  – поперечна сила.

### 2.3.1 Генерація навчальної вибірки квадратної пластини

Загальний алгоритм для генерації навчальної вибірки має наступний вигляд:

**algorithm** *generation\_square\_plate*

**input:**

$n$  (кількість значень у вибірці)

**output:**

$S$  (вибірка для навчання)

**begin**

$S \leftarrow \emptyset$

**for**  $i \leftarrow 1$  to  $n$  **do**

**begin**

$a \leftarrow \text{random}(0,14; 1,6)$

$h \leftarrow \text{random}(0,01; 0,1)$

$q \leftarrow \text{random}(0,01; 0,1)$

$E \leftarrow \text{random}(50000; 300000)$

$\vartheta \leftarrow \text{random}(0; 0,45)$

$S \leftarrow S \cup (a, h, E, \vartheta, q)$

**end**

**end**

Покроковий алгоритм побудови вибірки для навчання та тестування такий:

1. згенерувати випадкові значення параметрів  $a$ ,  $h$ ,  $E$ ,  $\vartheta$  і  $q$  у діапазонах можливих значень. Діапазон можливих значень необхідно обрати виходячи із

фізичного смислу та обмежень теорій і методів, що використані для побудови математичних моделей.

2. обчислити значення максимального прогину  $\omega_{max}$  за допомогою формули (2.17):

$$\omega_{max} = 0,00126qa^4 \frac{12(1-\vartheta^2)}{Eh^3}, \quad (2.17)$$

де  $q$  – значення тиску;

$a$  – розмір сторони пластинки;

$h$  – її товщина;

$\vartheta$  – коефіцієнт Пуассона;

$E$  – модуль Юнга.

3. Якщо  $10^{-5} < \omega_{max}$ , то обчислити компоненти напружено-деформованого стану за допомогою методу скінченних елементів на базі формули (2.16) для локальної матриці жорсткості; перейти до наступного кроку. Інакше перейти на крок 1.

4. Зберегти значення параметрів  $a$ ,  $h$ ,  $E$ ,  $\vartheta$  і  $q$ , максимального прогину  $\omega_{max}$  і компоненти напружено-деформованого стану у csv-файл.

### 2.3.2 Архітектура нейронної мережі

У випадку прогнозування максимального прогину пластинки отримаємо модель (рис. 2.7, а), яка має зовнішній шар з 5 нейронами для обробки вхідних параметрів і 1 нейроном для обчислення прогину. Для прогнозування прогину та, наприклад, інтенсивності напружень у пластинці мережа штучних нейронів повинна у вихідному шарі мати два нейрони (рис. 2.7, б). Отже, у функції

створення моделі зміниться лише значення одного параметру: кількості вихідних нейронів – `output_neurons=2`.

Всі кроки підготовки даних, за виключенням першого, описані вище, залишаються без змін. На першому кроці зміниться формування векторів вхідних і вихідних параметрів. Як і раніше вхідними будуть  $a$ ,  $h$ ,  $E$ ,  $v$  і  $q$ . Вихідних буде два параметри: обчислені за допомогою методу скінченних елементів максимальні прогин та інтенсивність напружень:

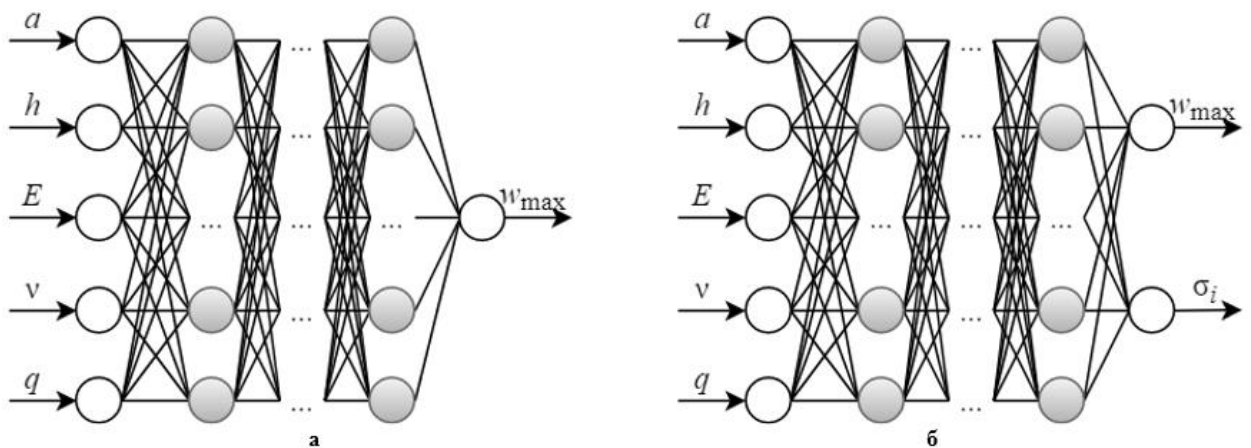


Рисунок 2.7 – Модель нейронної мережі для прогнозування максимального прогину квадратної пластинки

### 2.3.3 Обчислювальний експеримент

У результаті навчання з 700 епох було досягнуто значення середньої квадратичної похибки. В ході обчислювального експерименту було порівняно вплив алгоритмів оптимізації на швидкість навчання.

На рис. 2.6 порівнюється середнє квадратичне відхилення при навчанні з використанням різних алгоритмів обчислення ваг у нейронній мережі при мінімізації похибки.

Можна побачити, що найменше середнє квадратичне відхилення отримано при використанні алгоритму Adam (скорочено від Adaptive Moment Estimation) [88], який базується на аналізі середнього градієнтів і квадратів градієнтів штрафної функції.

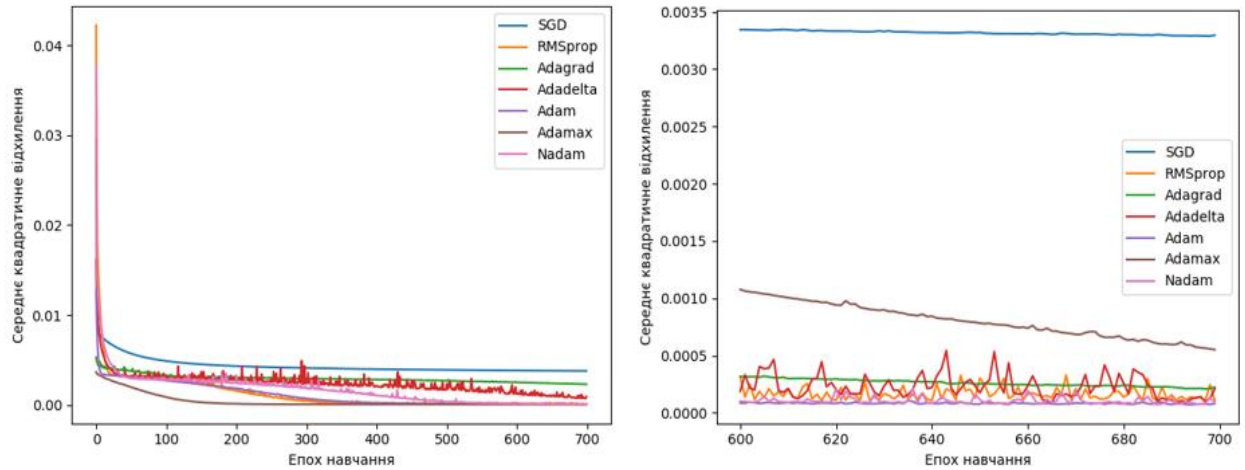


Рисунок 2.8 – Порівняння алгоритмів оптимізації нейронної мережі

Під час обчислювального експерименту було перевірено вплив функції активації на тих самих вихідних даних. Обчислювальний експеримент (рис. 2.8) показав, що при обчисленні максимального прогину у пластинці найменше середнє квадратичне відхилення на тестовій вибірці отримано при використанні гіперболічного тангенсу.

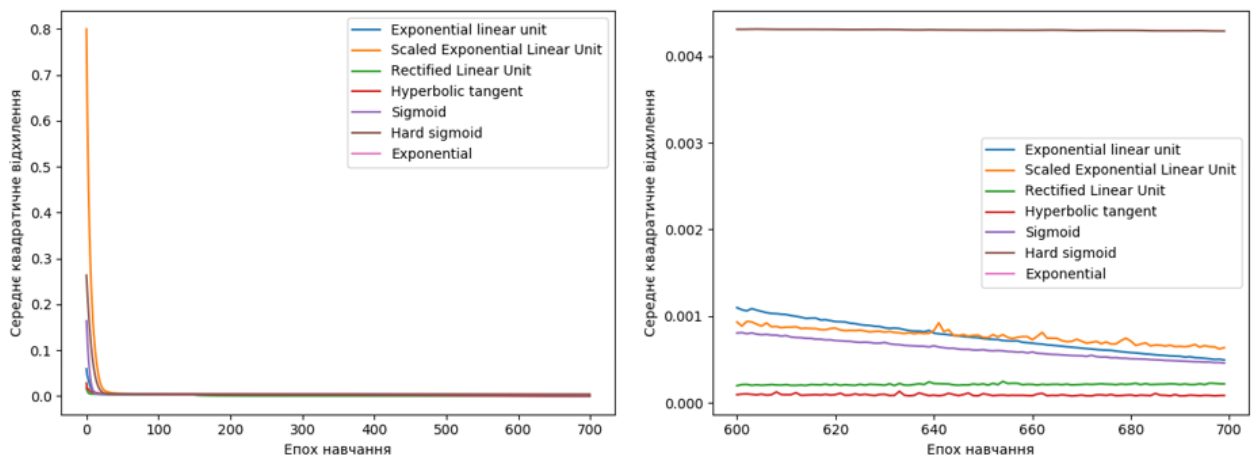


Рис. 2.9 – Порівняння функцій активації у нейронній мережі

Обчислювальні експерименти показали, що найменше середнє квадратичне відхилення від значень тестової підмножини буде при 1 або 3 внутрішніх шарах мережі (рис. 2.9, а). Водночас, якщо використати один внутрішній шар, то найменше середнє квадратичне відхилення буде, якщо використовувати 6 нейронів у цьому шарі (рис. 2.9, б).

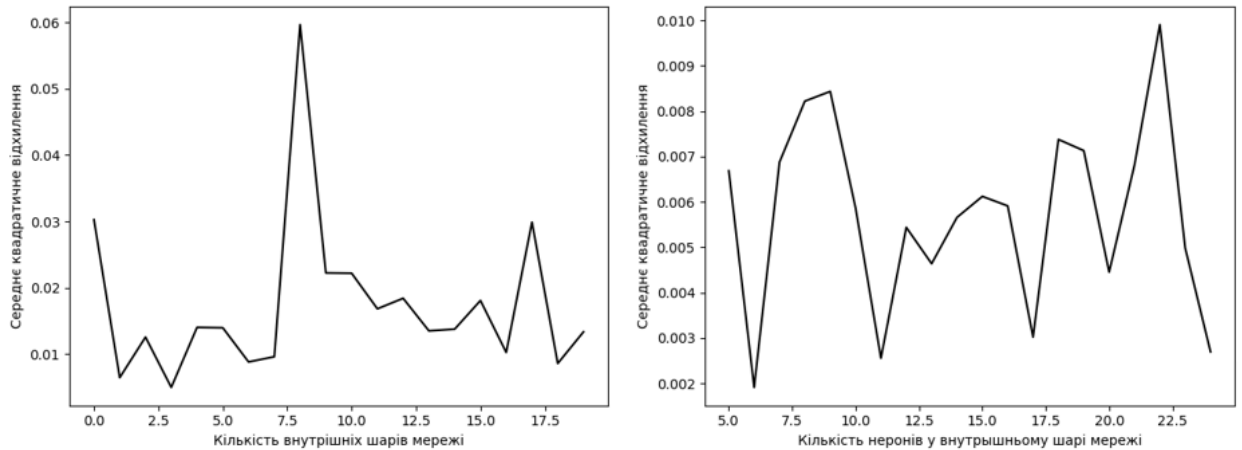


Рис. 2.9 – Вплив кількості внутрішніх шарів і нейронів у внутрішніх шарах

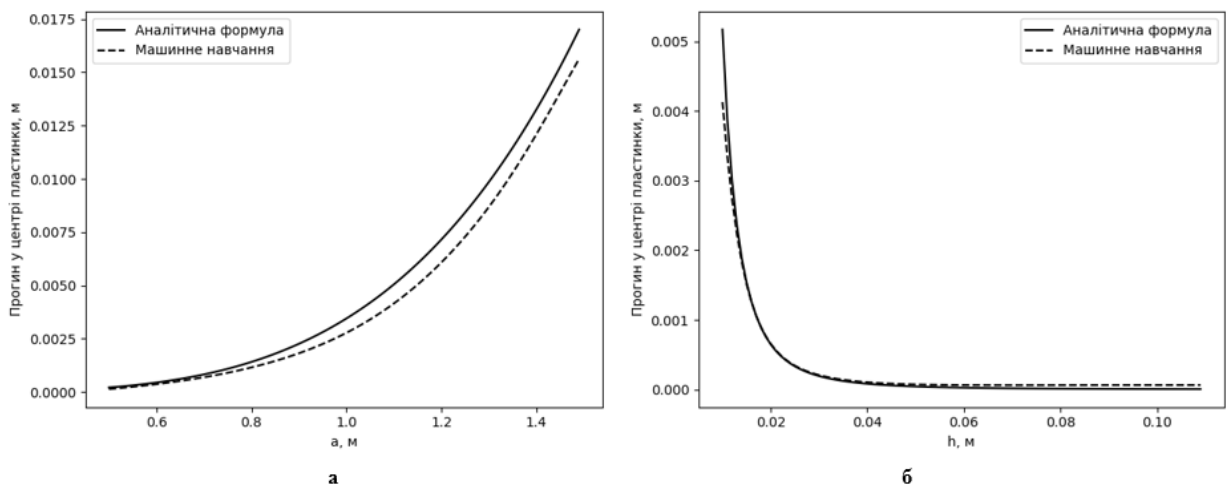


Рис. 2.10 – Порівняння аналітичного розв’язку з результатами машинного навчання

На рис. 2.10 порівняно значення прогину, отримані за допомогою аналітичної формули, зі значеннями, отриманими за допомогою нейронної мережі, при різних значеннях довжини сторони пластинки (рис. 2.10, а) та

різних значеннях її товщини (рис. 2.10, б). Можна побачити, що похибка відносно аналітичного розв'язку у межах 20%.

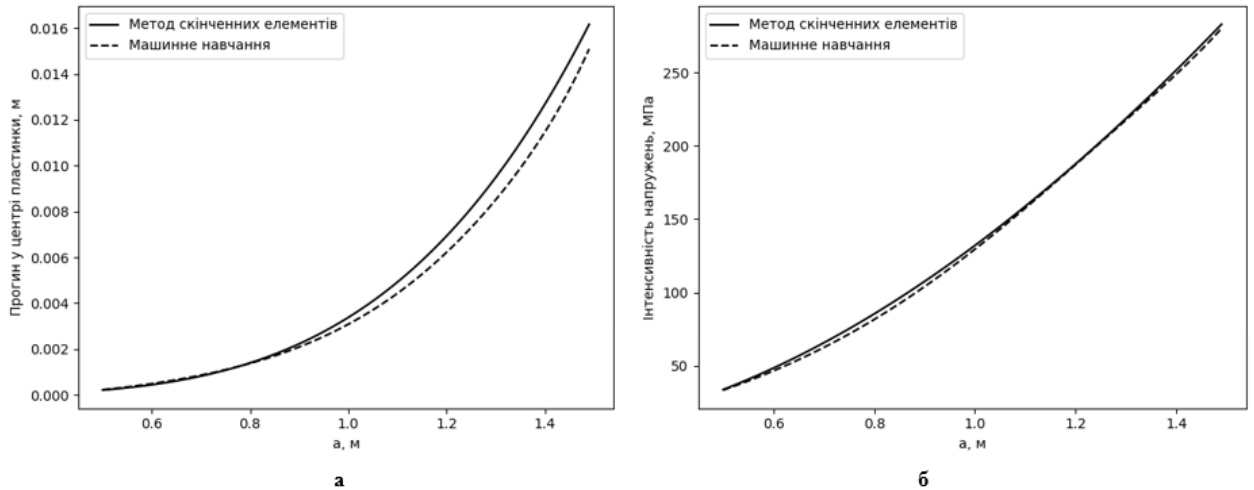


Рисунок 2.11 – Порівняння скінченно-елементного розв'язку з результатами машинного навчання: прогин та інтенсивність напружень у залежності від довжини сторони пластинки

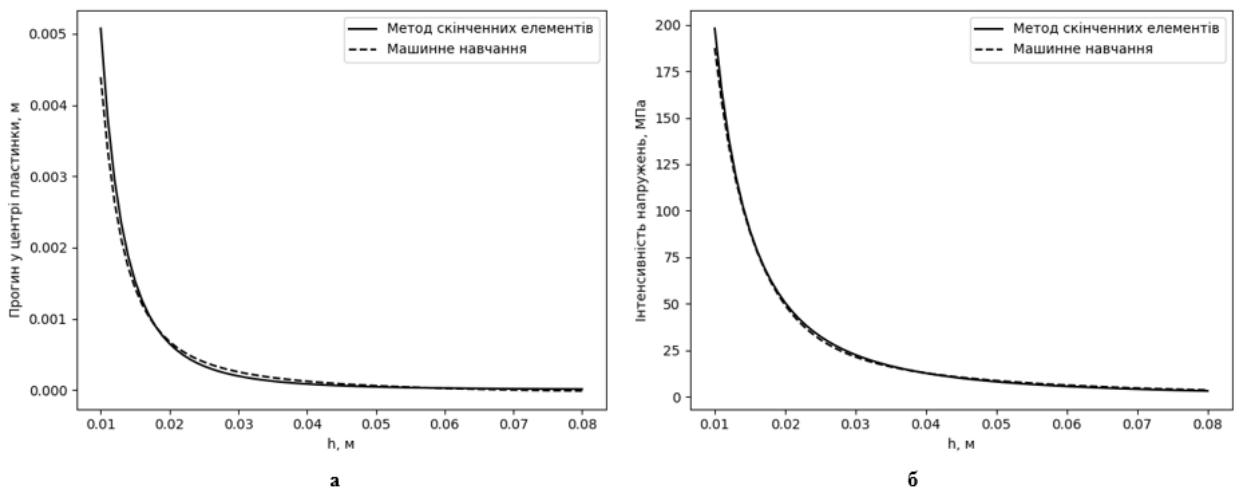


Рисунок 2.12 – Порівняння скінченно-елементного розв'язку з результатами машинного навчання: прогин та інтенсивність напружень у залежності від товщини пластинки

На рис. 2.11 порівняно значення прогину (рис. 2.11, а) та інтенсивності напружень (рис. 2.11, б), отримані за допомогою методу скінченних елементів, зі значеннями, отриманими за допомогою нейронної мережі, у залежності від

довжини сторони пластинки. Водночас на рис. 2.12 порівняно значення прогину (рис. 2.12, а) та інтенсивності напружень (рис. 2.12, б), отримані за допомогою методу скінченних елементів, зі значеннями, отриманими за допомогою нейронної мережі, у залежності від товщини пластинки.

Отже, штучні нейронні мережі дозволяють роботи прогноз одночасно декількох характеристик напружено-деформованого стану конструкції.

## 2.4 Нейромережевий метод моделювання напружено-деформованого стану круглої пластини

Розглянемо круглу пластину з параметрами:  $a$  – радіус,  $h$  – товщина,  $E$  – модуль пружності матеріалу,  $\nu$  – коефіцієнт Пуассона,  $q$  – поперечна сила. Якщо навантаження, що діє на круглу пластинку розподілене по ній симетрично відносно осі, що перпендикулярна до площини пластини та проходить через її центр, тоді вигнута поверхня, в яку перетвориться серединна площина пластини, також буде симетричною [110].

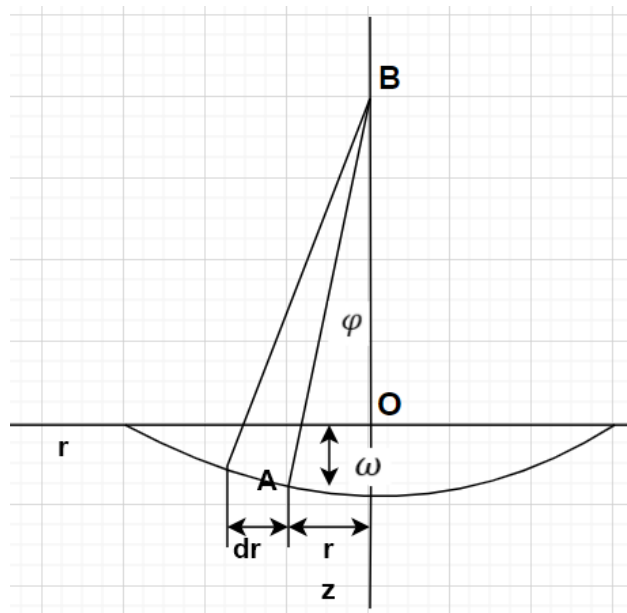


Рисунок 2.13 – Модель прогину круглої пластини

В усіх точках, рівновіддалених від центру пластини, прогини будуть однакові, тому їх можна розглядати в одному діаметральному перетині, що проходить через вісь симетрії (рис. 2.12)

Відповідно до [110] початок координат  $O$  знаходиться в центрі невингнутої пластини, через  $r$  позначено радіальну відстань точок, що лежать у серединній площині, а через  $\omega$  – їх прогин донизу.

Рівняння (2.18) використовується для опису прогину круглої пластини, що поперечно навантажена:

$$\frac{d}{dr} \left[ \frac{1}{r} \frac{d}{dr} \left( r \frac{d\omega}{dr} \right) \right] = \frac{Q}{D}, \quad (2.18)$$

де  $Q$  – перерізуюча сила, що приходить на одиницю довжини циліндричного перерізу радіуса  $r$ ;

$r$  – радіус;

$D$  – жорсткість пластини при вигині, рівняння (2.19):

$$D = \frac{Eh^3}{12(1-\vartheta^2)}, \quad (2.19)$$

де  $E$  – модуль пружності матеріалу,

$\vartheta$  – коефіцієнт Пуассона,

$h$  - товщина.

У випадку рівномірного навантаження  $q$  модель (2.18) отримує наступний вигляд:

$$\frac{d}{dr} \left[ \frac{1}{r} \frac{d}{dr} \left( r \frac{d\omega}{dr} \right) \right] = \frac{qr}{2D}. \quad (2.20)$$

Якщо пластина затиснена по контуру, тоді прогин можна представити у наступному вигляді:

$$\omega = \frac{q}{64D} (a^2 - r^2)^2, \quad (2.21)$$

де  $a$  – радіус,

$r$  – відстань від центру до поточної точки.

Найбільший прогин спостерігається в центрі пластини, тоді його величина буде рівною:

$$\omega_{max} = \frac{qa^4}{64D}. \quad (2.22)$$

Найбільша напруга спостерігається на контурі пластини і буде дорівнювати:

$$(\sigma_r)_{max} = \frac{3qa^2}{4h^2}. \quad (2.23)$$

#### 2.4.1 Генерація навчальної вибірки круглої пластини

Для визначення напружено-деформованого стану круглої пластини необхідні значення таких параметрів як:  $a$  – радіус пластини,  $h$  - товщина пластини,  $E$  – модуль пружності матеріалу,  $\vartheta$  – коефіцієнт Пуассона,  $q$  – поперечна сила.

Параметри пластинки генеруються випадковим чином з такими обмеженнями:

$$a \in [0.1; 4] \text{ (м);}$$

$$h \in [0.01; 0.05] \text{ (м);}$$

$$E \in [50000; 300000] \text{ (МПа);}$$

$$\vartheta \in [0; 0.45];$$

$$q \in [0.01; 0.1] \text{ (МПа).}$$

Алгоритм генерації даних круглої пластини представлено нижче:

**algorithm generation\_circular\_plate**

**input:**

n (кількість значень у вибірці)

**output:**

S (вибірка для навчання)

**begin**

S ← ∅

**for** i ← 1 to n **do**

**begin**

a ← *random* (0,1; 4)

h ← *random*(0,01; 0,05)

q ← *random*(0,01; 0,1)

E ← *random*(50000; 300000)

ϑ ← *random*(0; 0,45)

$$D \leftarrow \frac{Eh^3}{12(1-\vartheta^2)}$$

$$\omega_{max} \leftarrow \frac{qa^4}{64D}$$

$$\sigma_{max} \leftarrow \frac{3qa^2}{4h^2}$$

S ← S ∪ (a, h, E, ϑ, q, ω<sub>max</sub>, σ<sub>max</sub>)

**end**

**end**

де  $D$  – жорсткість пластини,

$\omega_{max}$  – максимальний прогин,

$\sigma_{max}$  – інтенсивність напружень.

В результаті генерації 10000 даних отримаємо таблицю з результатами по кожному параметру в залежності від заданих обмежень. На таблиці 2.1 зображено перші п'ять рядків згенерованих даних.

Таблиця 2.1 – Приклад генерації параметрів

	$a$	$h$	$E$	$\nu$	$q$	$\omega_{max}$	$\sigma_{max}$
0	2.618162	0.036304	200924.515528	0.371103	0.068699	0.054285	267.972256
1	2.633158	0.072611	173247.308759	0.300470	0.075827	0.009375	74.787976
2	1.167727	0.052212	284928.588053	0.014165	0.064196	0.000552	24.083265
3	0.231626	0.008641	288239.353767	0.240053	0.064966	0.000178	35.013128
4	2.405824	0.064774	149395.212444	0.012892	0.030405	0.004703	31.458769

У табл. 2.1 введено такі позначення:  $\omega_{max}$  – прогин пластини, а  $\sigma_{max}$  – інтенсивність напружень.

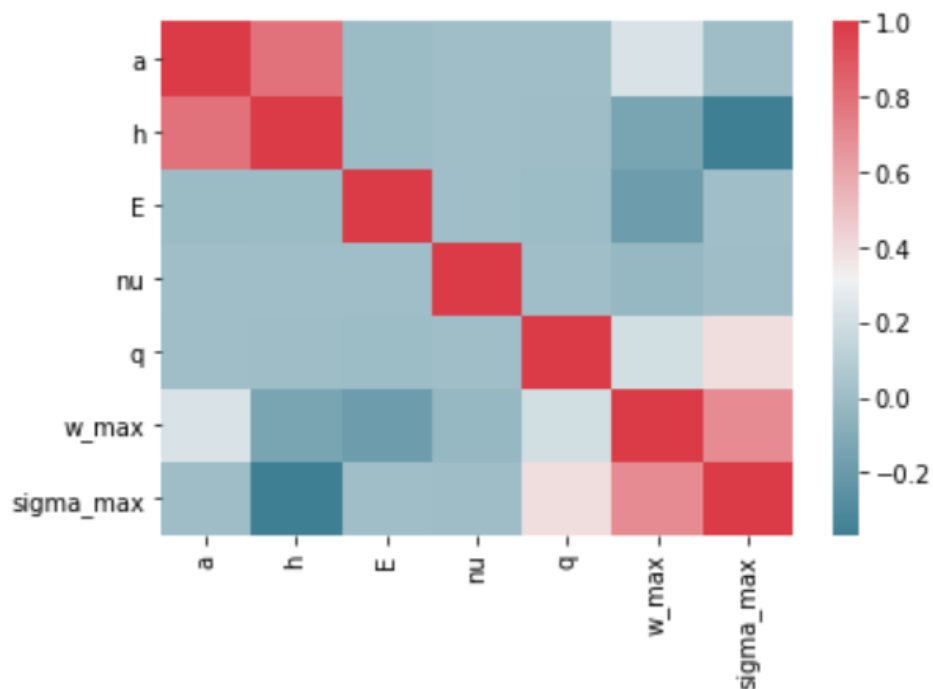


Рисунок 2.14 – Графік кореляції параметрів круглої пластини

У результаті генерації даних отримаємо графік кореляції, який зображено на рисунку 2.14. З графіку можна визначити взаємозв'язок параметрів круглої пластини, а саме: радіусу, товщини, модуль пружності, коефіцієнт Пуассона та поперечну силу. З графіку можна побачити, що зміни значень одних параметрів відповідає змінам значень інших параметрів.

### 2.4.2 Архітектура нейронної мережі

Для прогнозування прогину та інтенсивності напружень круглої пластини було розроблено дві штучні нейронні мережі. Структура нейронної мережі для визначення максимального прогину має наступний вигляд (рис. 2.15). Перший шар зовнішній і складається з 5 параметрів, що були використані під час генерації даних. Внутрішні шари моделі – 5 штук по 30 нейронів у кожному. Останній шар (також зовнішній) має кількість нейронів, яка дорівнює кількості вихідних значень, тобто 1 нейрон для обчислення прогину.

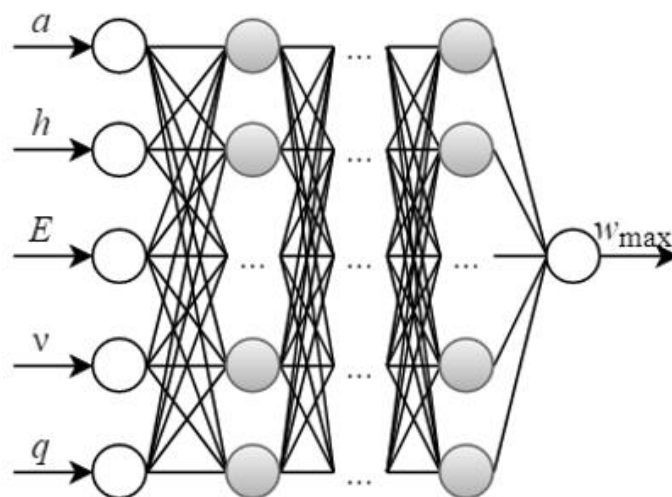


Рисунок 2.15 – Модель нейронної мережі для визначення максимального прогину

Структура нейронної мережі для прогнозування інтенсивності напружень  $\sigma_{max}$  у круглій пластині має шість вхідних параметрів у зовнішньому шарі:  $a$ ,  $h$ ,  $E$ ,  $q$ ,  $\vartheta$ ,  $\omega_{max}$  (рис. 2.16). П'ять внутрішніх прихованих шарів по 30 нейронів у кожному, та зовнішній вихідний шар з одним нейроном для прогнозування інтенсивності напружень.

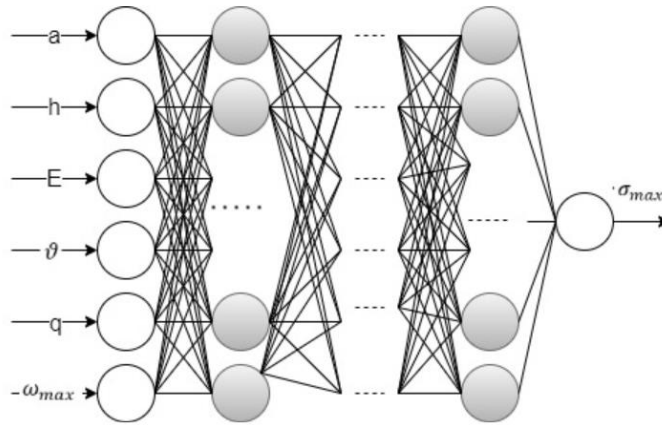


Рисунок 2.16 – Модель нейронної мережі для прогнозування інтенсивності напружень

### 2.4.3 Обчислювальний експеримент

Під час обчислювального експерименту для визначення максимального прогину та напруження круглої пластини було використано у створеній нейромережі різну кількість внутрішніх шарів з різною кількістю нейронів в них. Експеримент показав, що найкращі результати можна отримати використовуючи п'ять внутрішніх шарів з 30 нейронами. На рисунку 2.17 зображено як змінювались показники нейромережі в залежності від кількості нейронів.

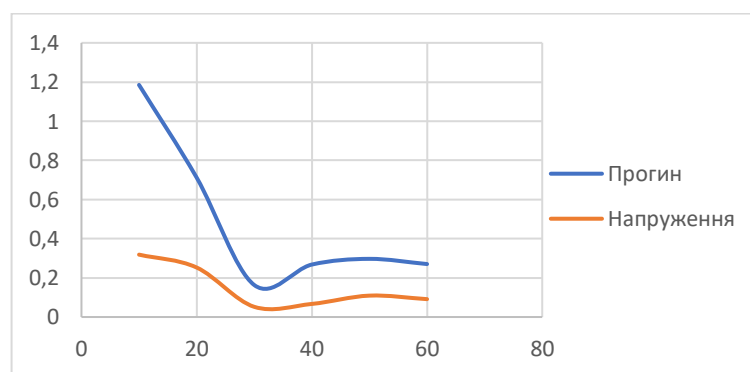


Рисунок 2.17 – Порівняння значень нейромережі в залежності від кількості нейронів у внутрішніх шарах

Експериментальні дані були порівняні з даними отриманими аналітично (табл. 2.2).

Таблиця 2.2 – Порівняння значень отримані аналітичною формулою з машинним навчанням

Значення аналітичної формули	Кількість нейронів					
	10	20	30	40	50	60
0,002673	0,005841	0,00077	0,002236	0,003389	0,003467	0,003396
	1,185185	0,711934	0,163487	0,267864	0,297045	0,270483
93,75	123,61	70,053	88,87	100,0174	103,996	102,308
	0,318507	0,252768	0,052053	0,066852	0,109291	0,091285

У результаті навчання 2000 епох досягнуто найкраще значення середньоквадратичної похибки (рис. 2.18).

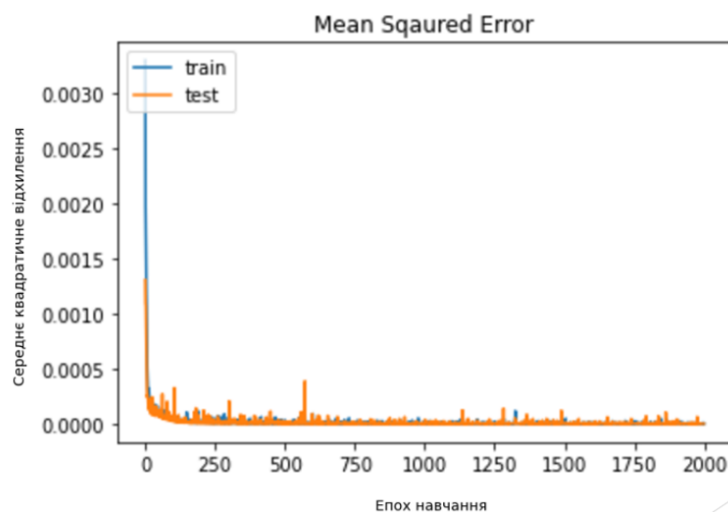


Рисунок 2.18 – Графік середньоквадратичної похибки у нейромережевій моделі

На рис. 2.19 порівняно значення прогину, отримані за допомогою аналітичної формули, зі значеннями, отриманими за допомогою нейронної мережі.

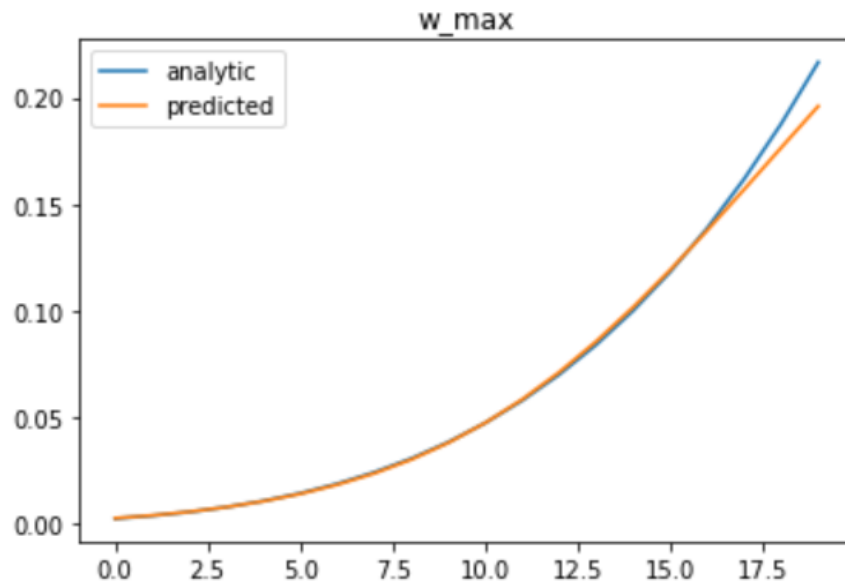


Рисунок 2.19 – Порівняння значень прогину, отримані за допомогою аналітичної формули, зі значеннями, отриманими за допомогою нейронної мережі

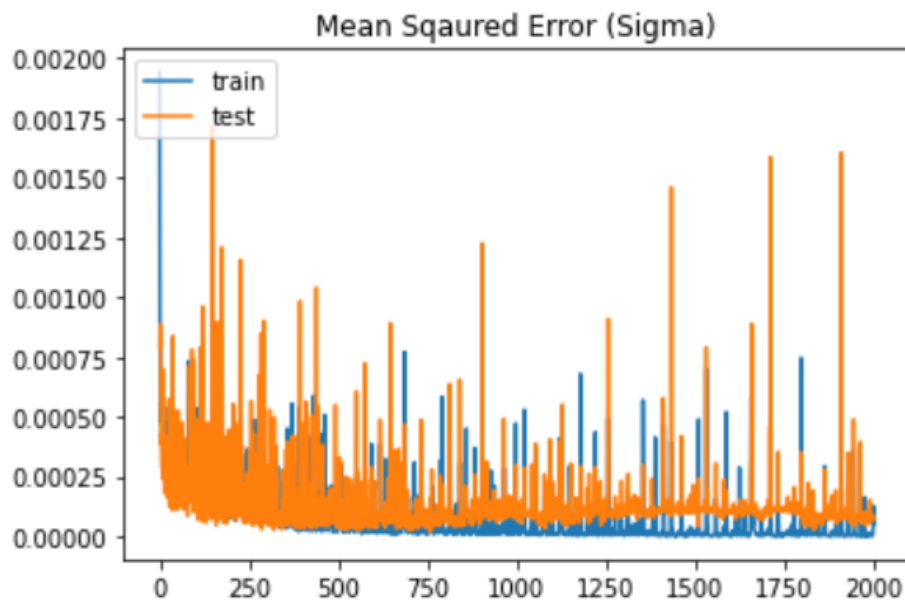


Рисунок 2.20 – Середньоквадратична похибка інтенсивності напружень

У результаті 2000 епох отримали середньоквадратичну похибку інтенсивності напружень (рис. 2.20).

#### 2.4.4 Моделювання довільних умов закріплення на прикладі круглої пластини

Розглянемо круглу пластину з параметрами:  $a$  – радіус,  $h$  – товщина,  $E$  – модуль пружності матеріалу,  $\nu$  – коефіцієнт Пуассона,  $q$  – поперечна сила. Усі краї пластини в даному випадку вільні.

Параметри пластинки генеруються випадковим чином з наступними обмеженнями:

$$a \in [0.1; 4] \text{ (м);}$$

$$h \in [0.01; 0.05] \text{ (м);}$$

$$E \in [50000; 300000] \text{ (МПа);}$$

$$\nu \in [0; 0.45];$$

$$q \in [0.01; 0.1] \text{ (МПа)}.$$

Алгоритм генерації даних з урахуванням довільних умов закріплення представлено нижче:

**algorithm generation\_circular\_plate\_free**

**input:**

$n$  (кількість значень у вибірці)

**output:**

$S$  (вибірка для навчання)

**begin**

$S \leftarrow \emptyset$

**for**  $i \leftarrow 1$  to  $n$  **do**

**begin**

$a \leftarrow \text{random}(0,1; 4)$

$h \leftarrow \text{random}(0,01; 0,05)$

$q \leftarrow \text{random}(0,01; 0,1)$

$E \leftarrow \text{random}(50000; 300000)$

$\vartheta \leftarrow \text{random}(0; 0,45)$

$$D \leftarrow \frac{Eh^3}{12(1-\vartheta^2)}$$

$$\omega_{max} \leftarrow \frac{q(5+\vartheta)a^4}{64D(1+\vartheta)}$$

$$\sigma_{max} \leftarrow \frac{3(5+\vartheta)qa^2}{8h^2}$$

$S \leftarrow S \cup (a, h, E, \vartheta, q, \omega_{max}, \sigma_{max})$

**end**

**end**

де  $D$  – жорсткість пластини,

$\omega_{max}$  – максимальний прогин,

$\sigma_{max}$  – інтенсивність напружень.

У результаті генерації даних отримаємо графік кореляції – взаємозв'язок параметрів круглої пластини (рис. 2.21, 2.22).

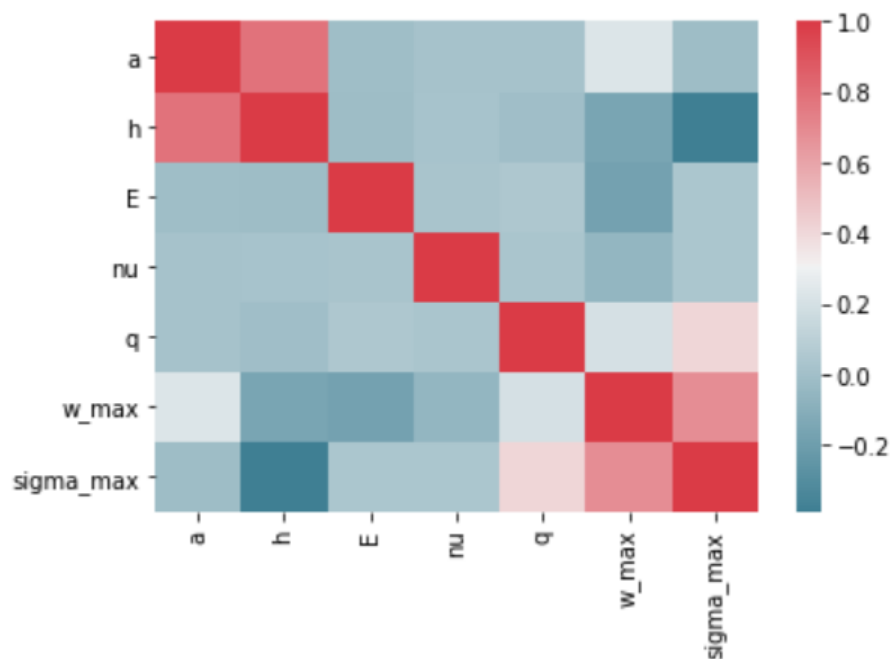


Рисунок 2.21 – Графік кореляції круглої пластини

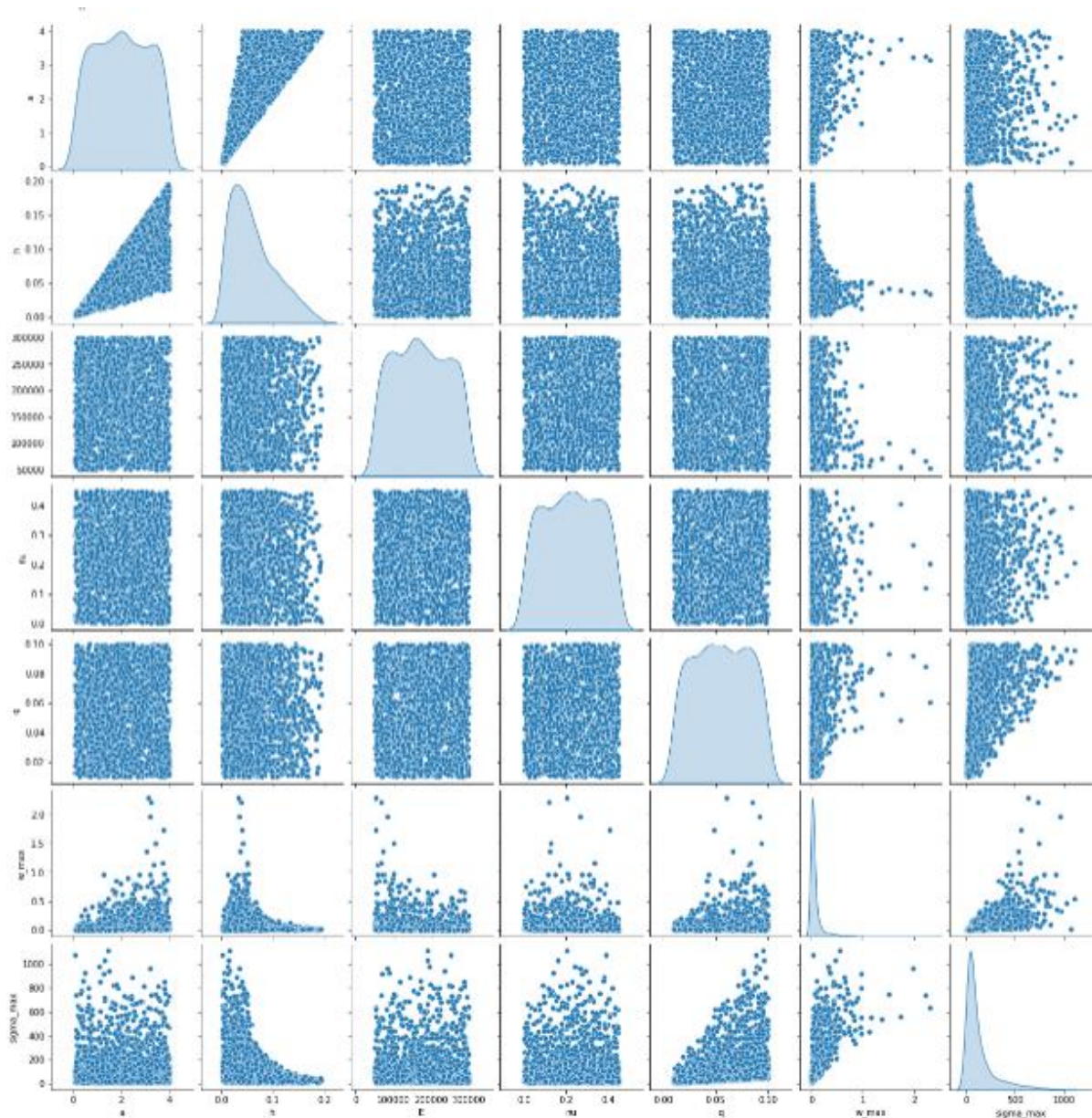


Рисунок 2.22 – Кореляційна залежність параметрів

Для прогнозування прогину та інтенсивності напружень круглої пластини з вільними краями було розроблено дві штучні нейронні мережі. Структура першої нейронної мережі для визначення максимального прогину має наступний вигляд (рис. 2.23). Перший шар зовнішній і складається з 5 нейронів, що були використані під час генерації даних. П'ять внутрішніх шарів по 30 нейронів у кожному. Останній шар (також зовнішній) має кількість нейронів, яка дорівнює кількості вихідних значень, тобто 1 нейрон для обчислення прогину.

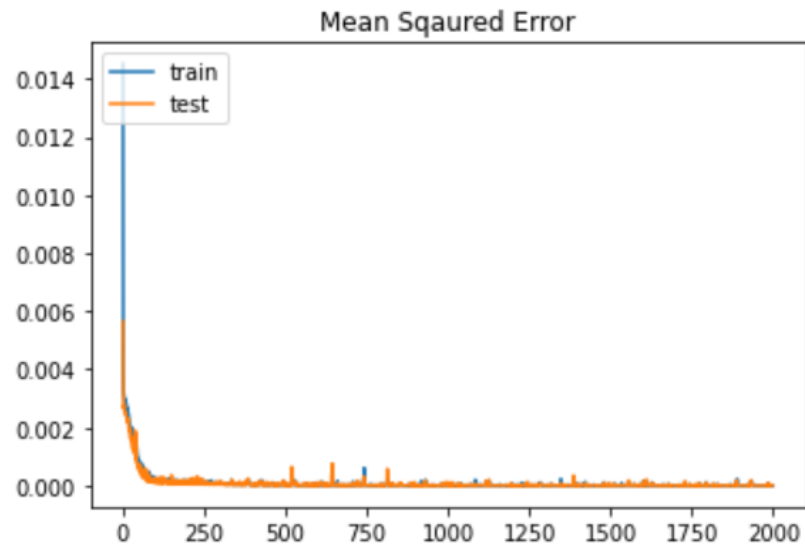


Рисунок 2.23 – Середньоквадратична похибка прогину

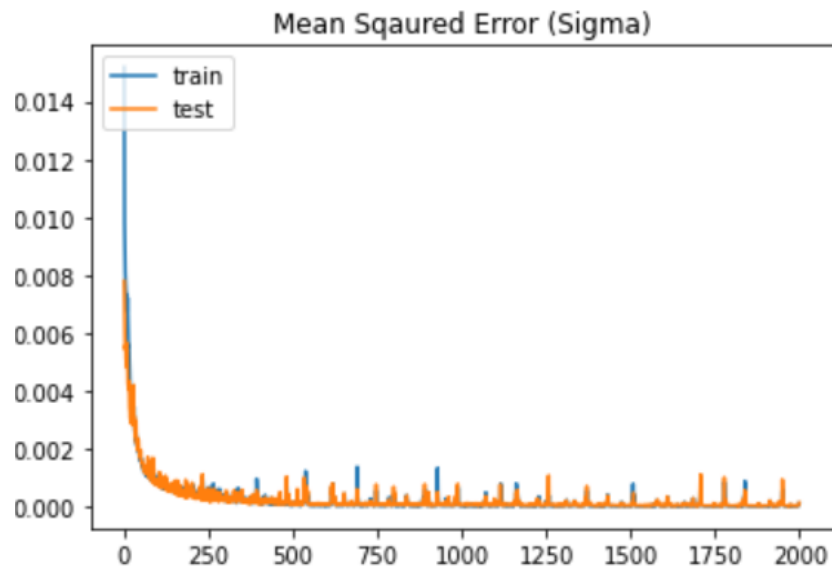


Рисунок 2.24 – Середньоквадратична похибка інтенсивності напружень

Структура другої нейронної мережі для визначення інтенсивності напружень складається з вхідного шару з 6 нейронів, 5 внутрішніх прихованих шарів по 30 нейронів у кожному, та зовнішній шар з 1 нейроном на виході.

На рисунках 2.23, 2.24 показано середньоквадратичну похибку тренувальної та тестової множин штучної нейронної мережі, навчання проведено з 2000 епох.

У ході обчислювального експерименту також було встановлено, що показники тренувальної та тестової множин для визначення максимального прогину та інтенсивності напружень відносно товщини пластини дають достатньо точні результати (рис. 2.25, 2.26)

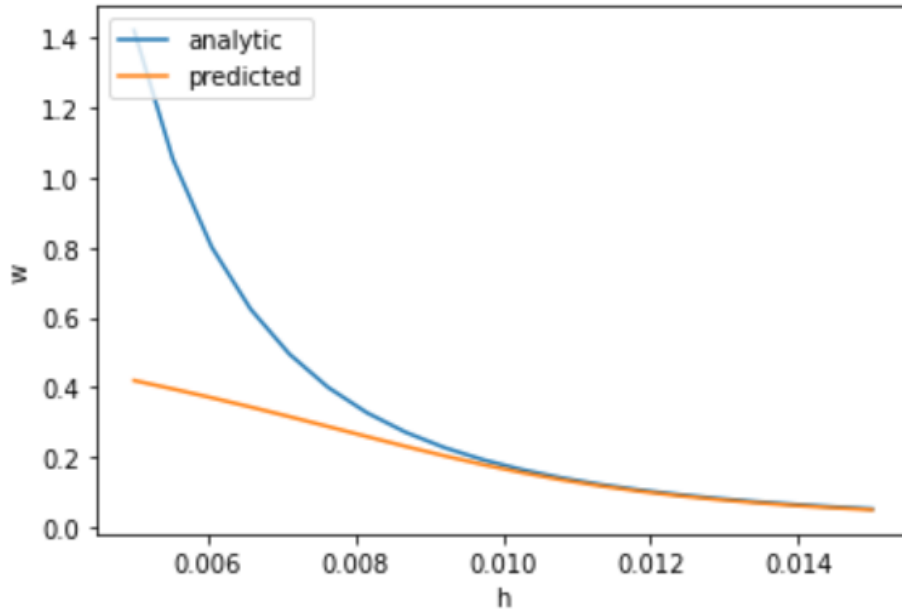


Рисунок 2.25 – Графік визначення прогину

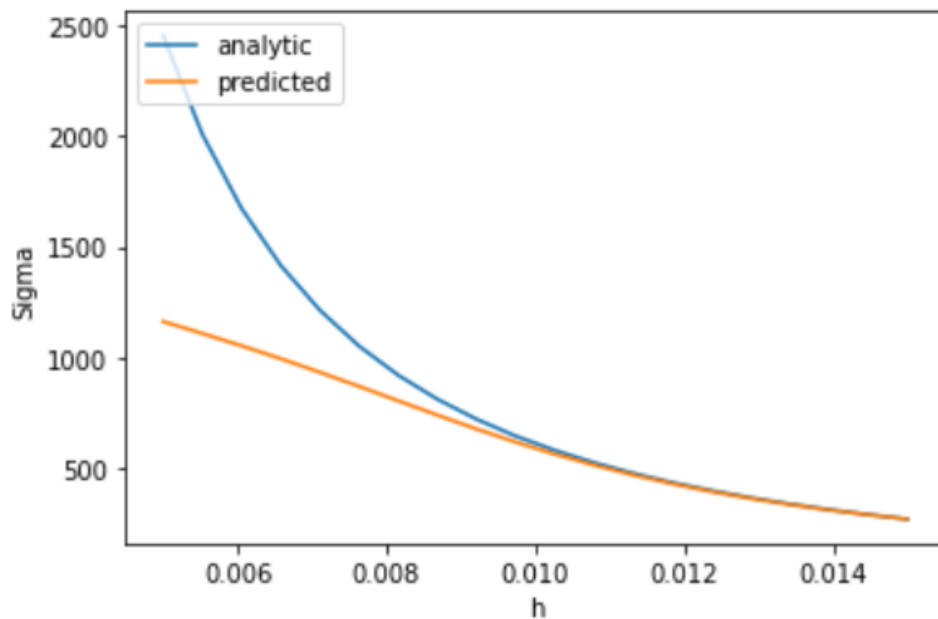


Рисунок 2.26 – Графік визначення інтенсивності напружень

## 2.5 Нейромережевий метод моделювання напружено-деформованого стану кільцевої пластини

Розглянемо кільцеву пластину з параметрами:  $radius_o$  – зовнішній радіус,  $radius_i$  – радіус внутрішній,  $h$  - товщина,  $E$  – модуль пружності матеріалу,  $\nu$  – коефіцієнт Пуассона,  $q$  – поперечна сила.

Дана кільцева пластинка з ізотропного матеріалу, яка знаходиться під дією рівномірно розподіленого по поверхні поперечного тиску. Для розробки нейронної мережі для прогнозування напружено-деформованого стану введені наступні параметри:

- $radius_o \in (0.1; 5)(\text{м})$ ;
- $radius_i \in [0.2; 0.8] \cdot radius_o (\text{м})$ ;
- $h \in \left[\frac{radius_o}{100}; \frac{radius_o}{20}\right] (\text{м})$  [116];
- $E \in [50000; 300000] (\text{МПа})$ ;
- $\nu \in [0; 0,45]$ ;
- $q \in [0.001; 0.1] (\text{МПа})$ .

### 2.5.1 Генерація навчальної вибірки кільцевої пластини

Для визначення напружено-деформованого стану круглої пластини необхідні значення таких параметрів як:  $radius_o$  та  $radius_i$  – зовнішній та внутрішній радіуси кільцевої пластини,  $h$  – товщина пластини,  $E$  – модуль пружності матеріалу,  $\nu$  – коефіцієнт Пуассона,  $q$  – поперечна сила.

Параметри пластинки генеруються випадковим чином. Алгоритм генерації даних представлено нижче.

**algorithm generation\_ring\_plate****input:**

$n$  (кількість значень у вибірці)

**output:**

$S$  (вибірка для навчання)

**begin**

$S \leftarrow \emptyset$

**for**  $i \leftarrow 1$  to  $n$  **do**

**begin**

$radius_o \leftarrow random(0.1, 5)$

$radius_i \leftarrow random(0.2; 0.8) \cdot radius_o$

$h \leftarrow random\left(\frac{radius_o}{100}; \frac{radius_o}{20}\right)$

$q \leftarrow random(0,001; 0,1)$

$E \leftarrow random(50000; 300000)$

$\vartheta \leftarrow random(0; 0,45)$

$S \leftarrow S \cup (radius_o, radius_i, h, E, \vartheta, q)$

**end****end**

Графік кореляції для кільцевої пластини представлено нижче на рисунку 2.27.

Приклад напружено-деформованого стану кільцевої пластини з навчальної вибірки, згенерованої методом скінченних елементів. Наприклад, якщо:

$$radius_o = 2.5;$$

$$radius_i = 0.8;$$

$$q = 0.05;$$

$$E = 203200;$$

$$\vartheta = 0.27;$$

$$h = 0.01;$$

то можна отримати дискретну модель з 45344 вузлів та 44684 елементів, а у результаті скінченно-елементного аналізу розподіл прогину та напружень у вигляді рисунків (рис. 2.28 – 2.31).

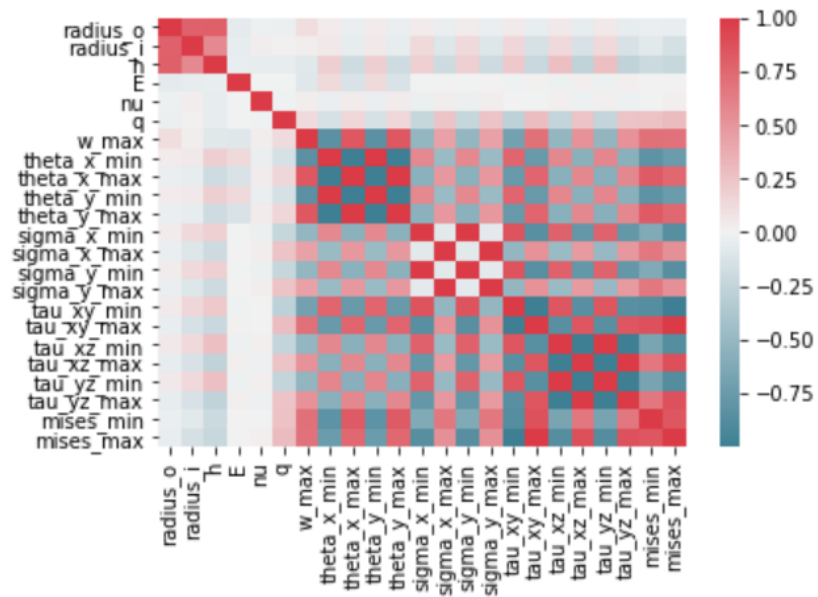


Рисунок 2.27 – графік кореляції кільцевої пластини

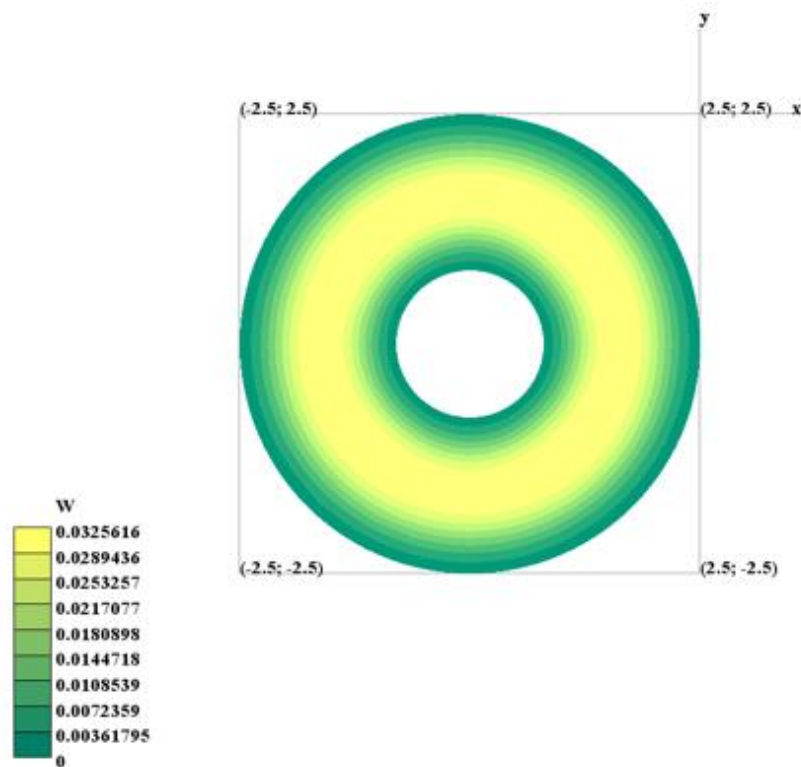


Рисунок 2.28 – Зміна значень прогину  $w$  кільцевої пластини

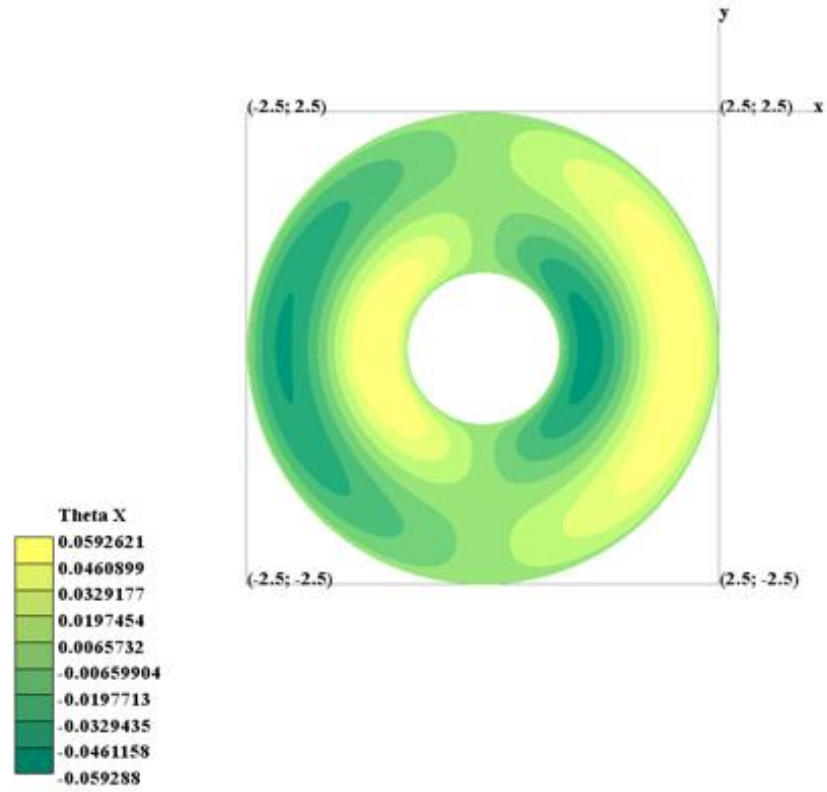


Рисунок 2.29 – Зміна значень  $\theta_x$  на заданому інтервалі

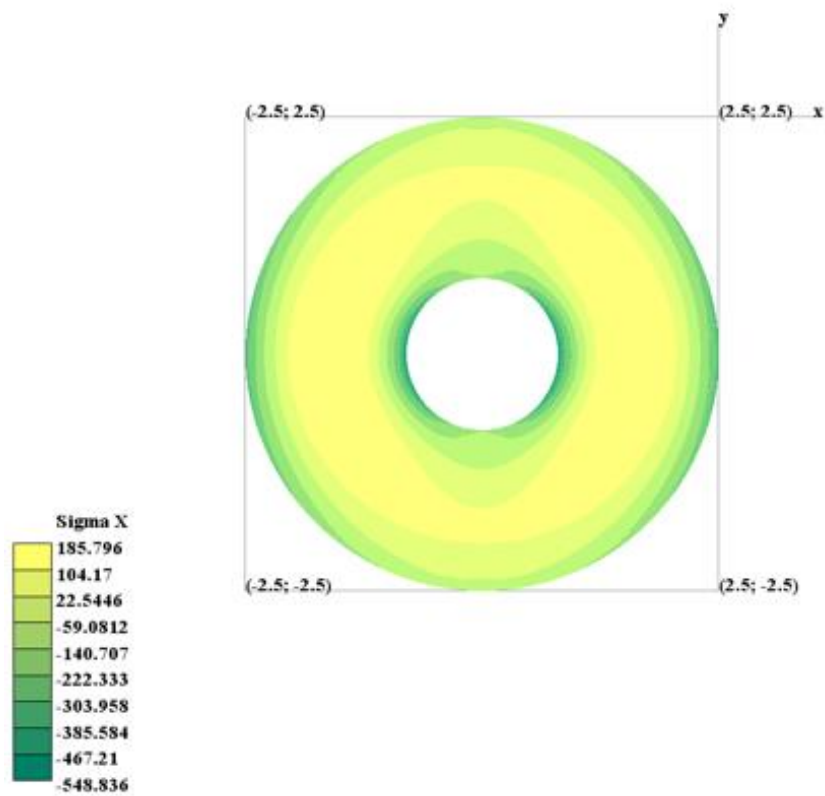


Рисунок 2.30 – Напруження  $\sigma_x$  у кільцевій пластині

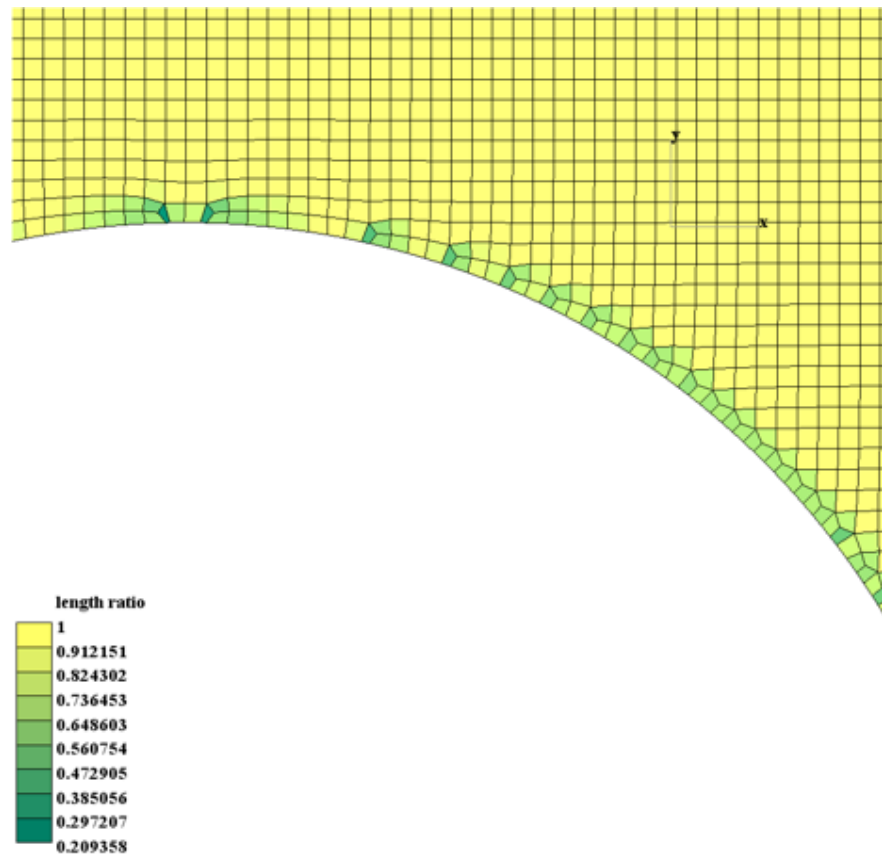


Рисунок 2.31 – Дискретна модель кільцевої пластини

## 2.5.2 Архітектура нейронної мережі

Для прогнозування прогину та інтенсивності напружень кільцевої пластини було розроблено дві штучні нейронні мережі.

Структура нейронної мережі для визначення максимального прогину має наступний вигляд (рис. 2.32). Перший шар зовнішній поділено на дві частини – числові дані та дані категорій. До числових даних відносяться 6 параметрів пластини, тобто даний шар має 6 нейронів. До категоріальних даних відносяться умови закріплення пластини. Ця гілка містить 12 параметрів, тобто 12 нейронів. Далі для гілки з числовими даними було експериментально виявлено, що для найкращого навчання необхідно 4

прихованих шарів по 30 нейронів у кожному. Для гілки з даними категорій було розроблено 2 прихованих шарів по 60 нейронів у кожному. Наступним кроком є об'єднання двох гілок в один шар, який містить 90 нейронів. Для навчання було розроблено ще 4 прихованих шарів по 90 нейронів у кожному. Останній шар (також зовнішній) має кількість нейронів, яка дорівнює кількості вихідних значень, тобто 1 нейрон для обчислення прогину.

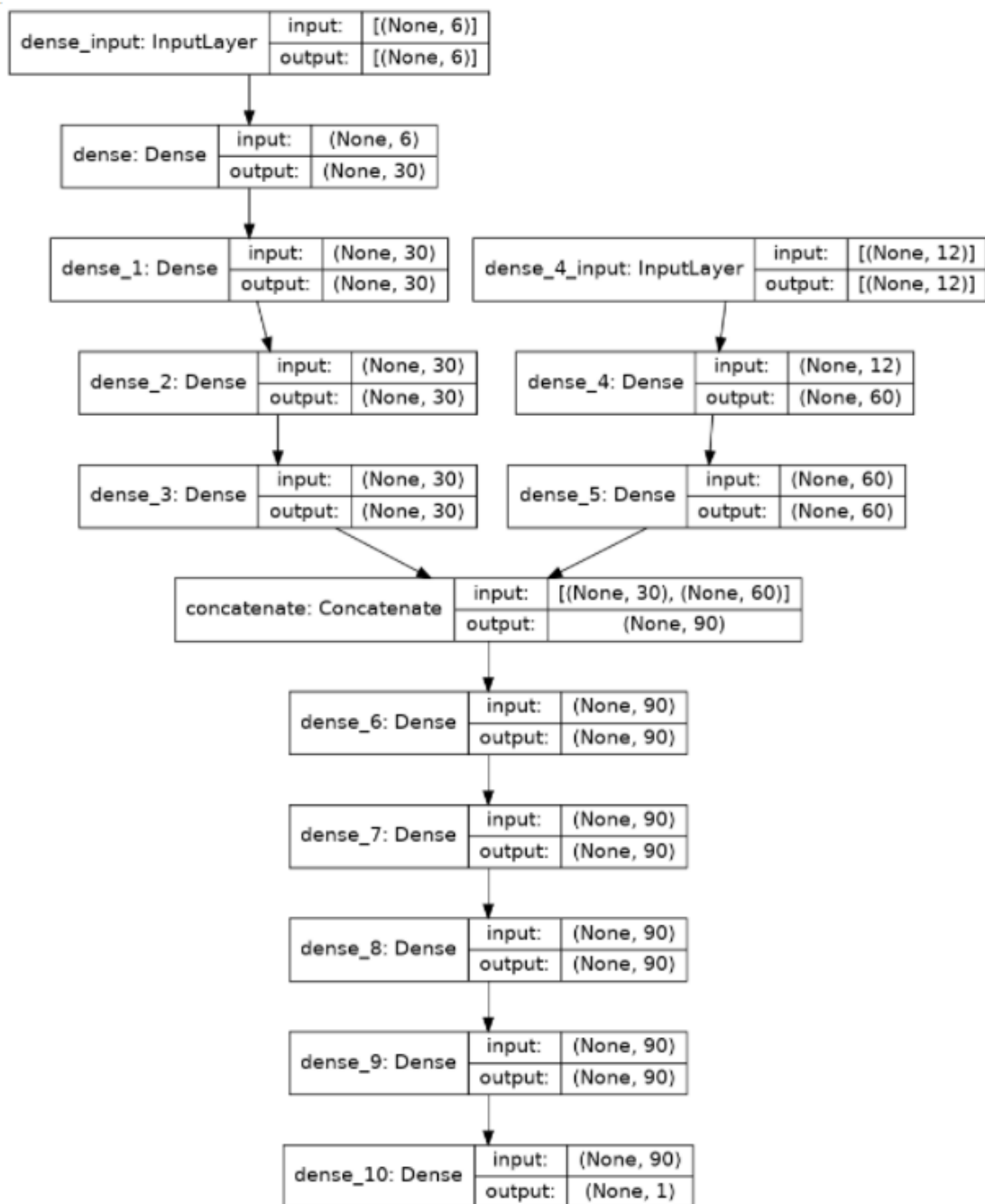


Рисунок 2.32 – Модель нейронної мережі для визначення максимального прогину кільцевої пластини

Структура штучної нейронної мережі для визначення інтенсивності напружень має таку ж структуру, як і для визначення прогину, тільки в даному випадку вхідний шар поділено на дві гілки: числову з 7 нейронами та категоріальну з 12 нейронами. Числова гілка має 4 внутрішні приховані шари, категоріальна – 1 внутрішній шар. Після об'єднання отримаємо 4 внутрішніх шарів з 19 нейронами у кожному. Зовнішній шар має 1 нейрон для визначення інтенсивності напружень.

### 2.5.3 Обчислювальний експеримент

У результаті обчислювального експерименту, проведеного з 1000 епох, отримали середньоквадратичну похибку (рис. 2.33), а також значення інтенсивності напружень (рис. 2.34).

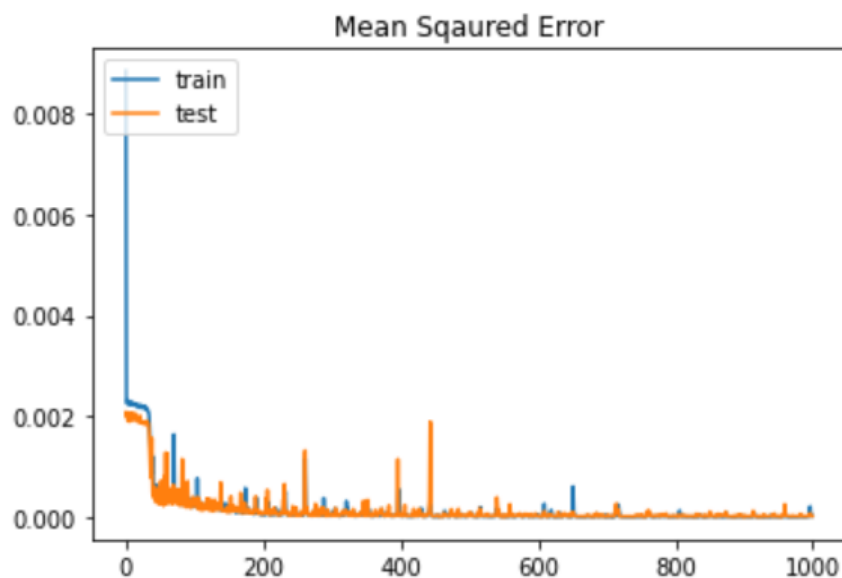


Рисунок 2.33 – Середньоквадратична похибка обчислювального експерименту

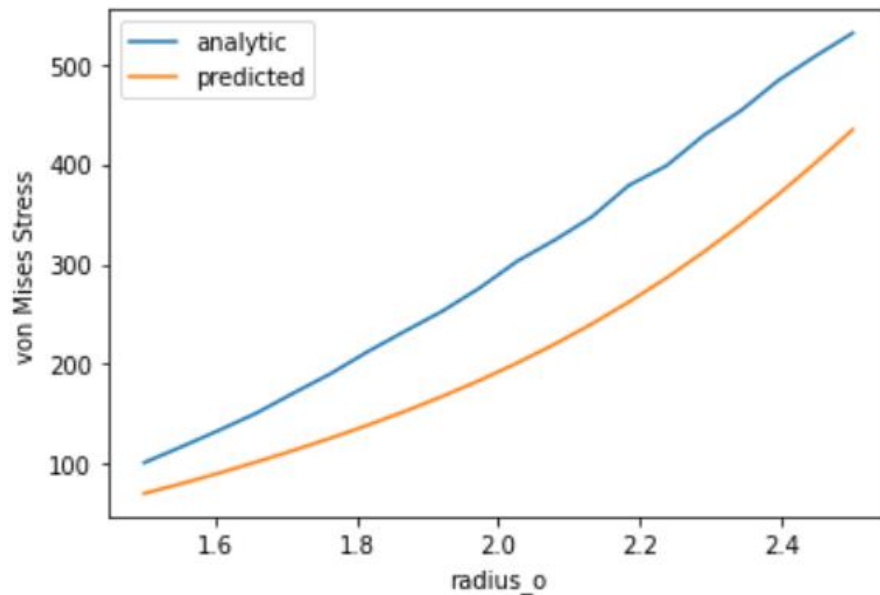


Рисунок 2.34 – Порівняння значень аналітичного розрахунку та машинного навчання інтенсивності напружень

## 2.6 Врахування у нейромережевій моделі довільних умов закріплення на прикладі прямокутної пластинки з отвором

Для реалізації нейронної мережі для оптимізації параметрів для прогнозування прогину квадратної пластини з круглим вирізом входні дані були розділені на дві частини – числові дані та дані категорій. До числових даних відноситься 9 параметрів пластини, що генеруються випадковим чином. Пластина має певні параметри такі як: ширина  $w$ , довжина  $h$ , радіус вирізу  $r$  з центром у точці  $(x_0, y_0)$ , товщину  $h$ , модуль Юнга  $E$  та коефіцієнт Пуассона  $\nu$ . На пластину діє поперечне навантаження  $q$ .

Генерація даних відбувається з використанням методу скінченних елементів. Параметри платини генеруються випадковим чином дотримуючись певних умов:

$$w \in [0,1; 10] \text{ (м)};$$

$$h \in [0,1; 10] \text{ (м)};$$

$r \in \left[3s; \frac{1}{2}a - 3s\right]$  (м), де  $s$  – розмір фонові комірки у сітці (для рівномірної сітки можна вважати  $s = \frac{\max(w,h)}{n}$ ;  $n^2$  – кількість комірок),  $a = \min(w, h)$ ;

$x_0 \in \left[0; \frac{1}{2}a - r - 3s\right]$  (м);

$y_0 \in \left[\frac{1}{2a} - r - 3s\right]$  (м);

$h \in \left[\frac{1}{80}a; \frac{1}{5}a\right]$  (м);

$E \in [50000; 300000]$  (МПа);

$\vartheta \in [0; 0,45]$ ;

$q \in [0,01; 0,1]$  (МПа).

Дані категорій – всі можливі поєднання граничних умов, що також генеруються випадковим чином. Можлива будь-яка комбінація граничних умов, виключаючи ситуації з чотирма вільними ребрами або одним підтримуваним ребром та трьома вільними. Позначимо граничні умови на краю  $x = -w/2$  як  $c_0$ , на краю  $y = -h/2$  як  $c_1$ , на краю  $x = w/2$  як  $c_2$  та на краю  $y = h/2$  як  $c_3$ . Якщо ми також позначимо вільний край на 0, підтримуваний край на 1, затиснутий край на 2, то отримаємо таке обмеження:  $c_1 + c_2 + c_3 + c_4 \geq 2$ .

Приклад напружено-деформованого стану прямокутної пластинки з отвором з навчальної вибірки. Наприклад, якщо жорстко затиснена прямокутна пластинка має такі параметри:

$$w = 0.9005274341208861,$$

$$h = 0.7638958718627692,$$

$$r = 0.15896447113567758,$$

$$x_0 = -0.08917247917477059,$$

$$y_0 = 0.15737602391689418,$$

$$h = 0.037682979653901806,$$

$$E = 76875.65085943788,$$

$$\vartheta = 0.32642319321166724,$$

$$q = 0.08670016849576496$$

то буде згенеровано дискретну модель з 48198 вузлів та 47560 елементів, що дозволить отримати розподіли компонентів НДС у вигляді рисунків. (рис. 2.35 – 2.39)

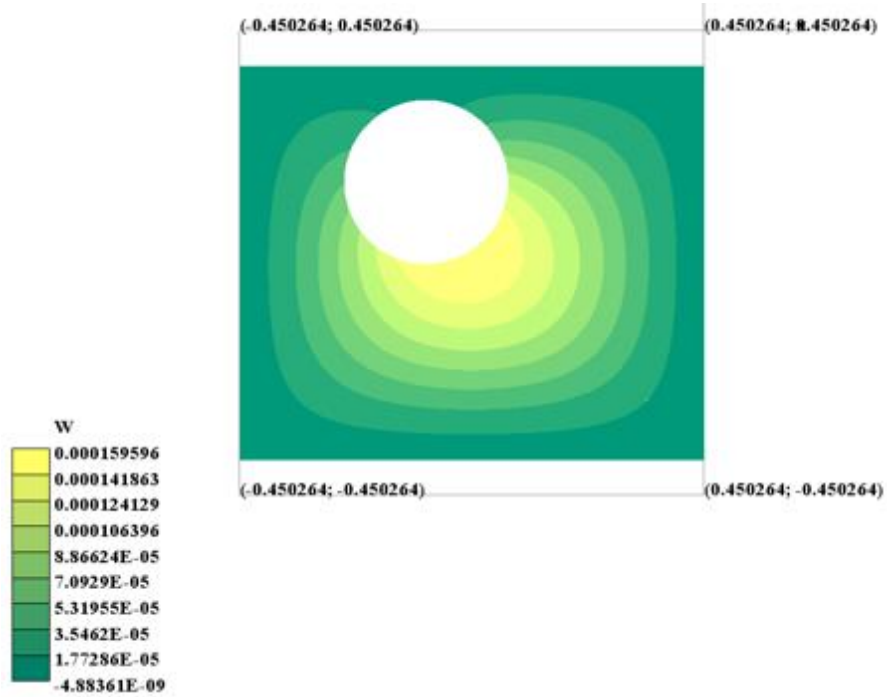


Рисунок 2.35 – Зміна значень  $w$  прямокутній пластині з отвором

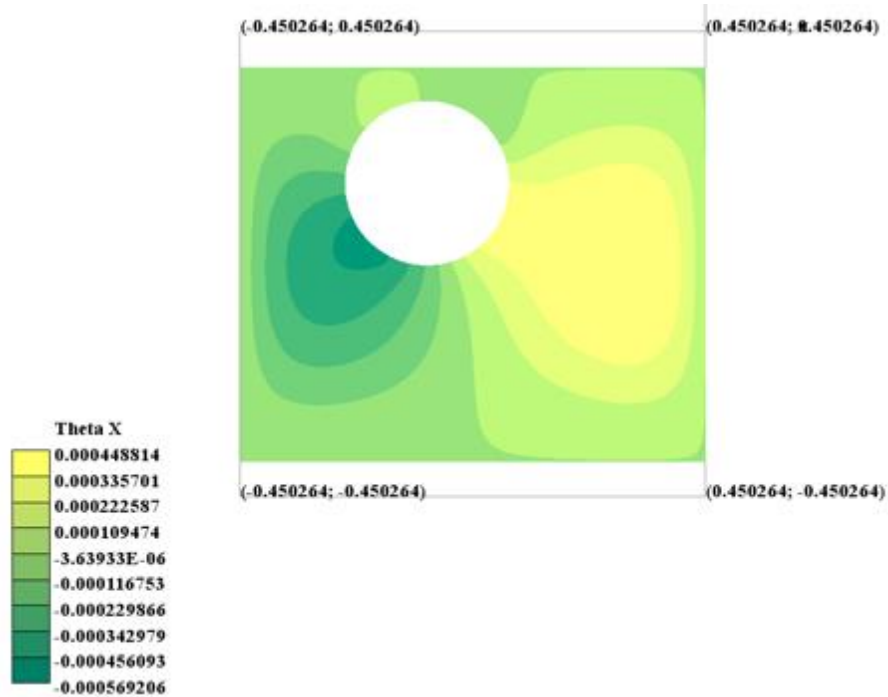


Рисунок 2.36 – Зміна значень  $\theta_x$  прямокутній пластині з отвором

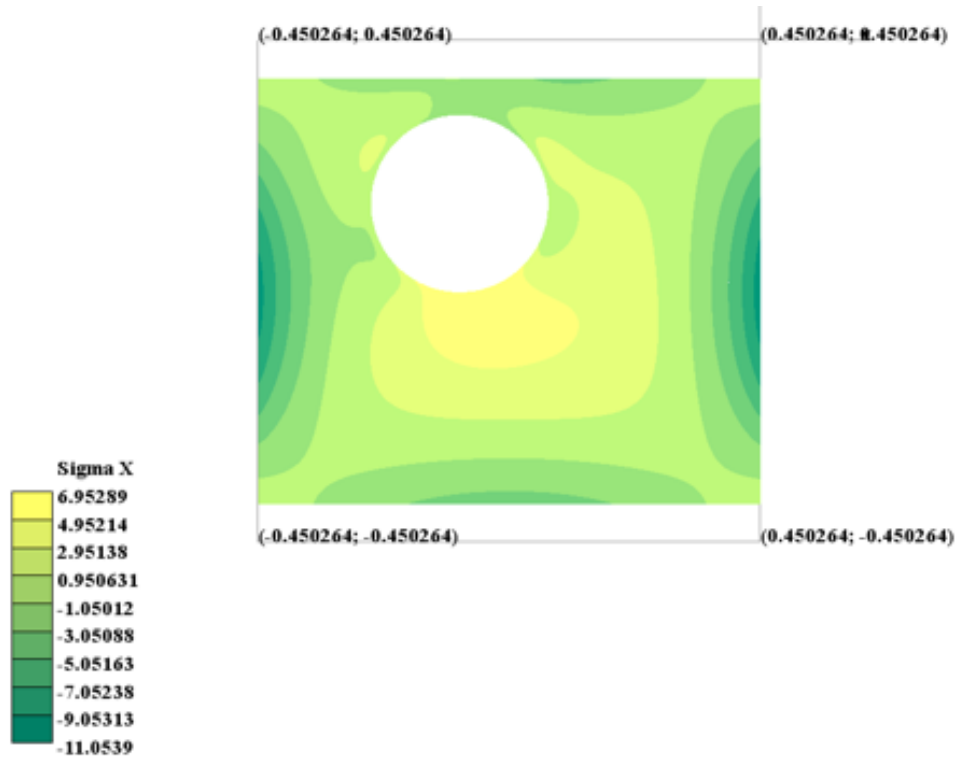


Рисунок 2.37 – Зміна значень  $\sigma_x$  прямокутній пластині з отвором

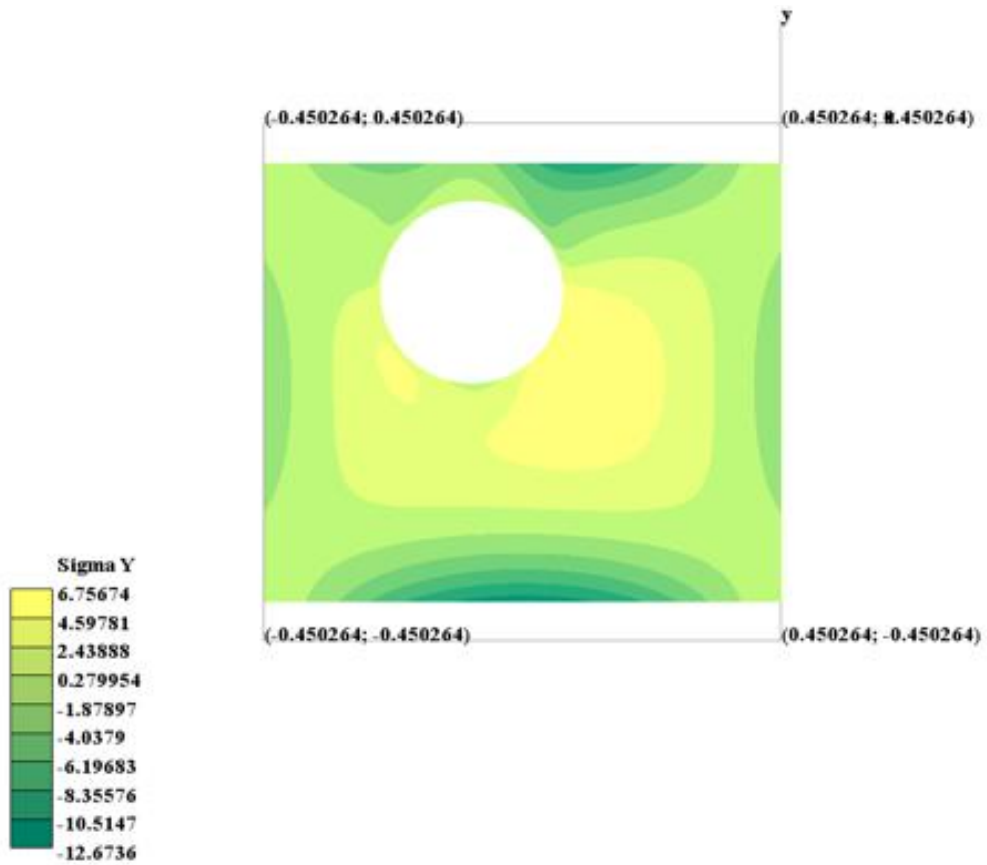


Рисунок 2.38 – Зміна значень  $\sigma_y$  прямокутній пластині з отвором

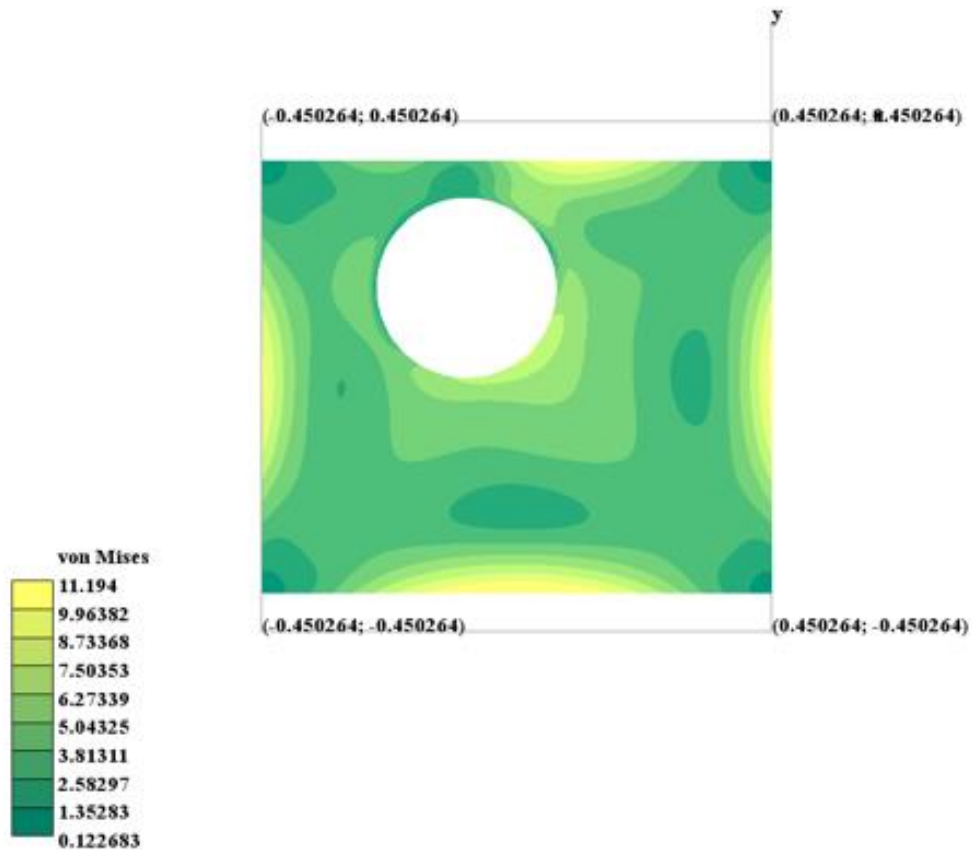


Рисунок 2.39 – Інтенсивність напружень за Мізесом

Для прогнозування поведінки пластинки розроблено модель, яка має такі шари нейронів. По-перше, шари вхідних нейронів (окремо для числових параметрів і умов закріплення). Нейрони вхідних шарів поєднуються з прихованим шаром нейронів. Далі зовнішній шар нейронів з лінійною функцією активації формує регресійну оцінку вектору, що описує напружено-деформований стан пластинки. Внутрішні шари мають певну кількість нейронів, кожен з яких зв'язаний з внутрішньою функцією активації. Останній шар (також зовнішній) має кількість нейронів, яка дорівнює кількості вихідних значень. Дані категорій містять 76 значень (рис. 2.40).

Випадково згенерований набір даних складається з 11000 обчислювальних експериментів. Приблизно 7500 значень залишилося після проведення чистки даних. Чистка даних відбувається за умови, що максимальний прогин повинен бути більшим ніж  $10^{-5}$  метрів або менше ніж 0,2 від товщини пластини.

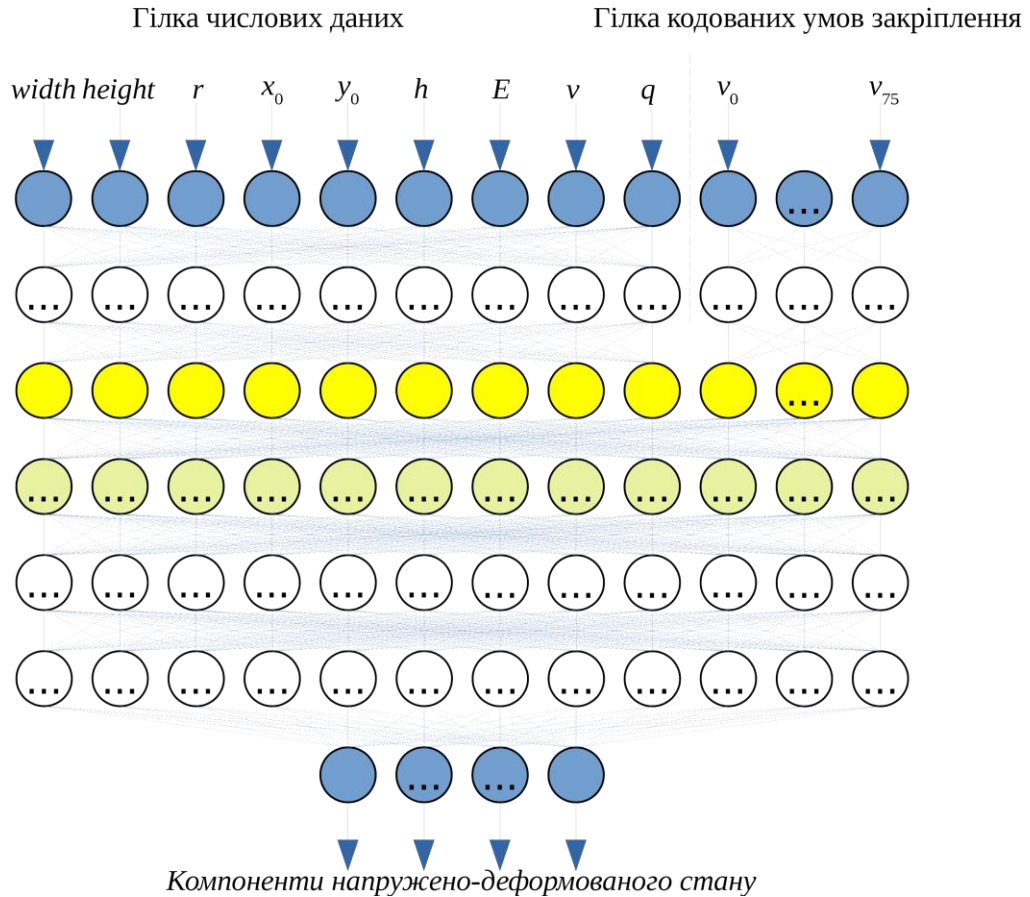


Рисунок 2.40 – Модель розробленої нейронної мережі

Вхідний шар був поділений на дві частини, які складаються з чисельних даних та даних категорій. Кожен вхідний шар з'єднується з прихованими внутрішніми шарами. Внутрішній шар об'єднує вхідні шари для обробки числових та категоріальних даних. Останній шар об'єднаний з додатковим прихованим шаром. В результаті, вихідний шар з'єднується з останнім прихованим шаром.

Для вхідного шару та прихованих шарів розробленої моделі нейронної мережі було використано функцію активації тангенс гіперболічний. Вихідний шар використовує лінійну функцію активації.

Кількість нейронів у вхідних шарах дорівнює кількості параметрів вхідної множини. Наприклад, для обробки чисельних даних було використано 8 нейронів, а для обробки даних категорій було використано 76 нейронів. Кількість нейронів у прихованих шарах змінюється з метою отримання

мінімуму функції витрат. На рисунку 2.41 зображено функцію витрат для обох множин – тренувальної та тестової.

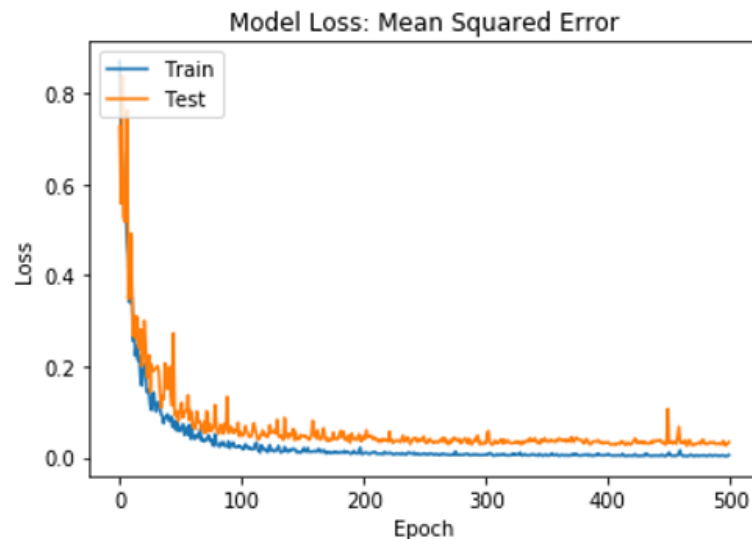


Рисунок 2.41 – Середньоквадратична похибка моделі нейронної мережі для визначення максимального прогину та інтенсивності напружень з урахуванням умов закріплення

## **2.7 Розробка нейромережевого методу на базі архітектури «автокодувальник» для визначення можливих зон руйнування у квадратних пластинах з отвором**

Для розробки штучної нейронної мережі такого типу були використані 25 зображень квадратної пластини з круглим отвором. Розміри пластини залишалися незмінними, змінювалися координати вирізу та його радіус.

Модель розробленої нейронної мережі для кодувальника та декодувальника має наступний вигляд – вхідний шар отримує зображення розміром  $N \times M$ .

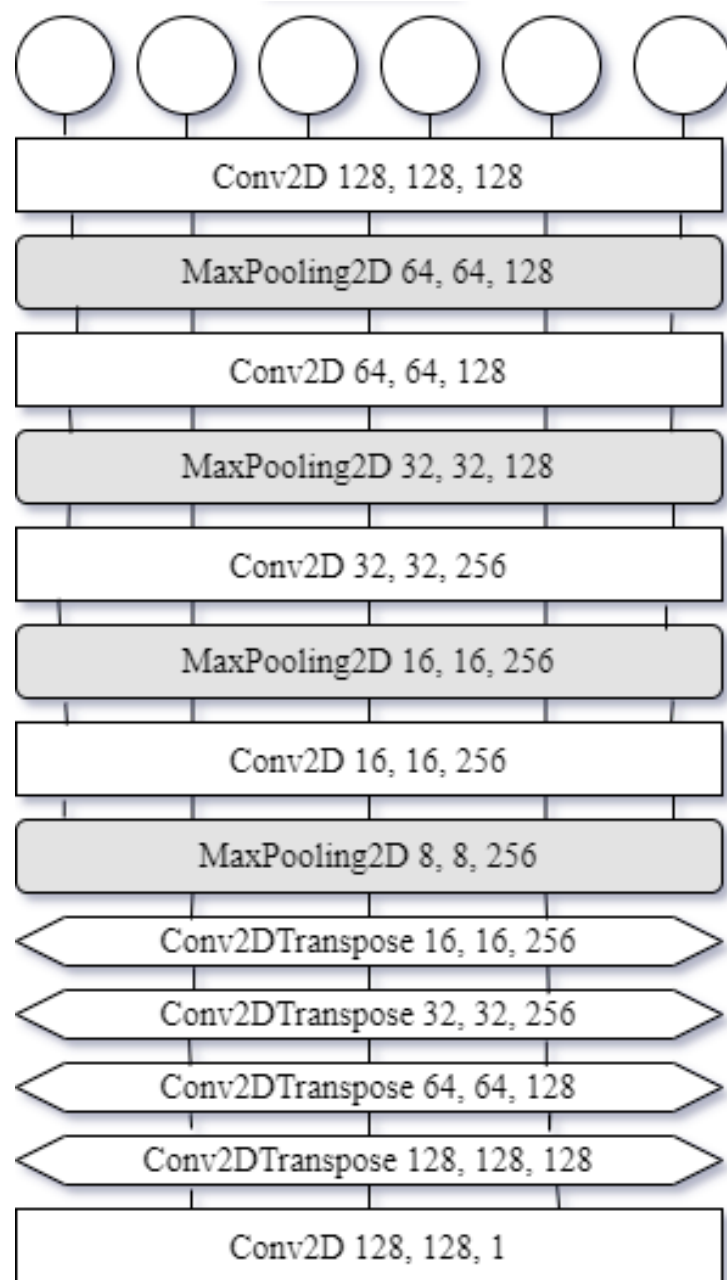


Рисунок 2.42 – Розроблена модель нейронної мережі

Далі відбувається чергування шарів: згортковий шар (Conv2D) з фільтрами  $F_1$  з ядром  $x_1 \times y_1$ , шар субдискретизації (MaxPooling2D) з ядром  $k_1 \times l_1$ , на якому відбувається зменшення розмірів вхідного зображення  $\frac{N}{k_1} \times \frac{M}{l_1}$ , наступний шар – згортковий з фільтрами  $F_2$  з ядром  $x_2 \times y_2$ , шар субдискретизації з ядром  $k_2 \times l_2$ , після цього зображення має розміри  $\frac{N}{k_1 k_2} \times \frac{M}{l_1 l_2}$ , і т. д. Кількість необхідних шарів можна визначити

експериментальним способом. Наступним етапом є кодування декодувальника, тобто розробка шарів транспонованої згортки (Conv2DTranspose). Кількість шарів декодувальника повинна відповідати кількості шарів кодувальника. У шарах транспонованої згортки відбувається зворотній процес – накладаються фільтри і розміри зображення збільшуються, тобто на виході отримуємо зображення розміром  $N \times M$ .

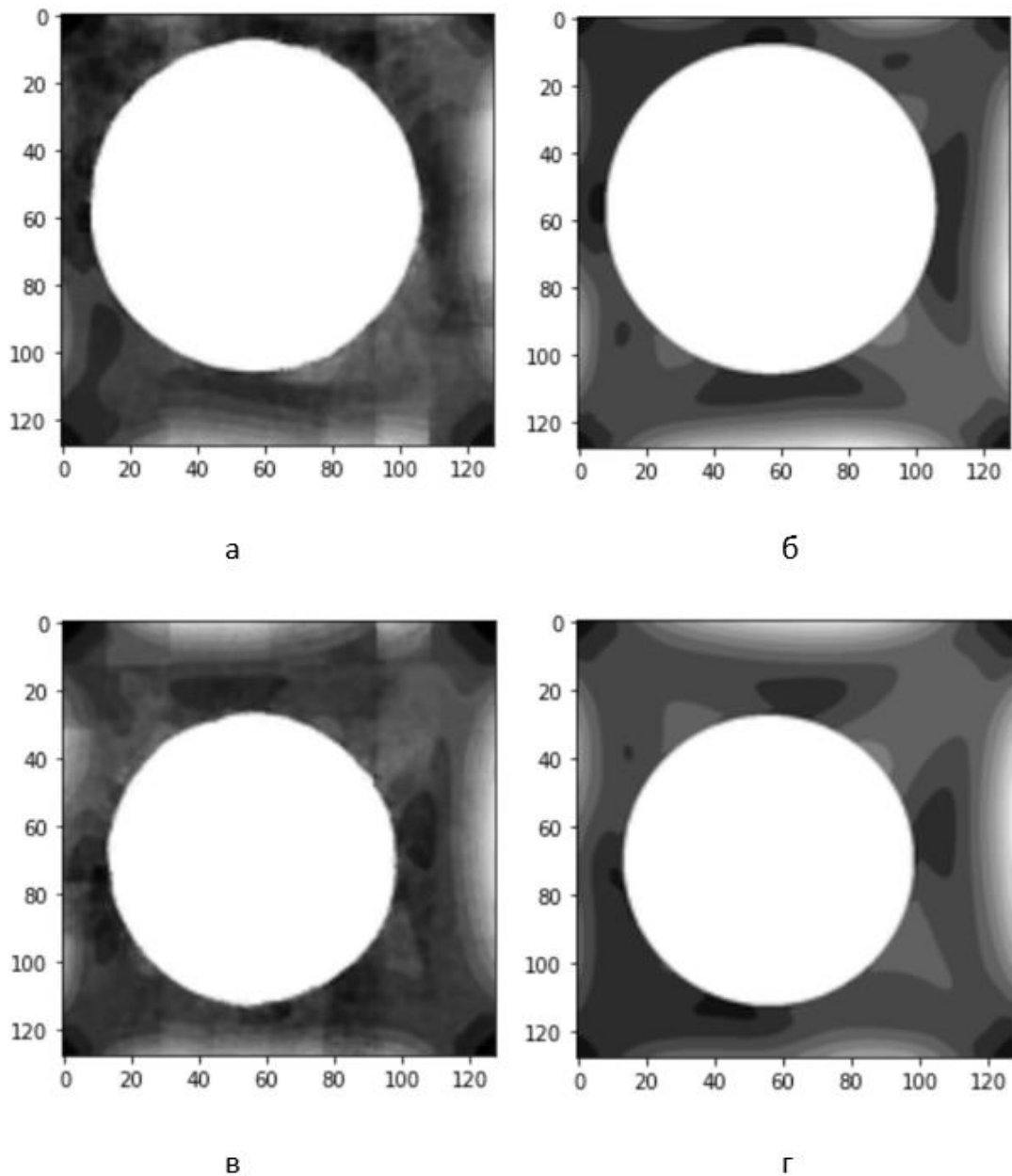


Рисунок 2.43 – Порівняння значень отриманих методом скінченних елементів та з використанням машинного навчання (а, в – машинне навчання, б, г – метод скінченних елементів)

Під час обчислювального експерименту була розроблена модель штучної нейронної мережі (рис. 2.42), яка на вході отримує зображення розміром  $128 \times 128$ . Далі чергуються внутрішні шари згортки Conv2D з шарами субдискретизації MaxPooling2D, у яких на зображення накладаються фільтри з ядром певного розміру. Кількість шарів Conv2D дорівнює 4. В останньому шарі кодувальника зображення має розмір  $8 \times 8$ .

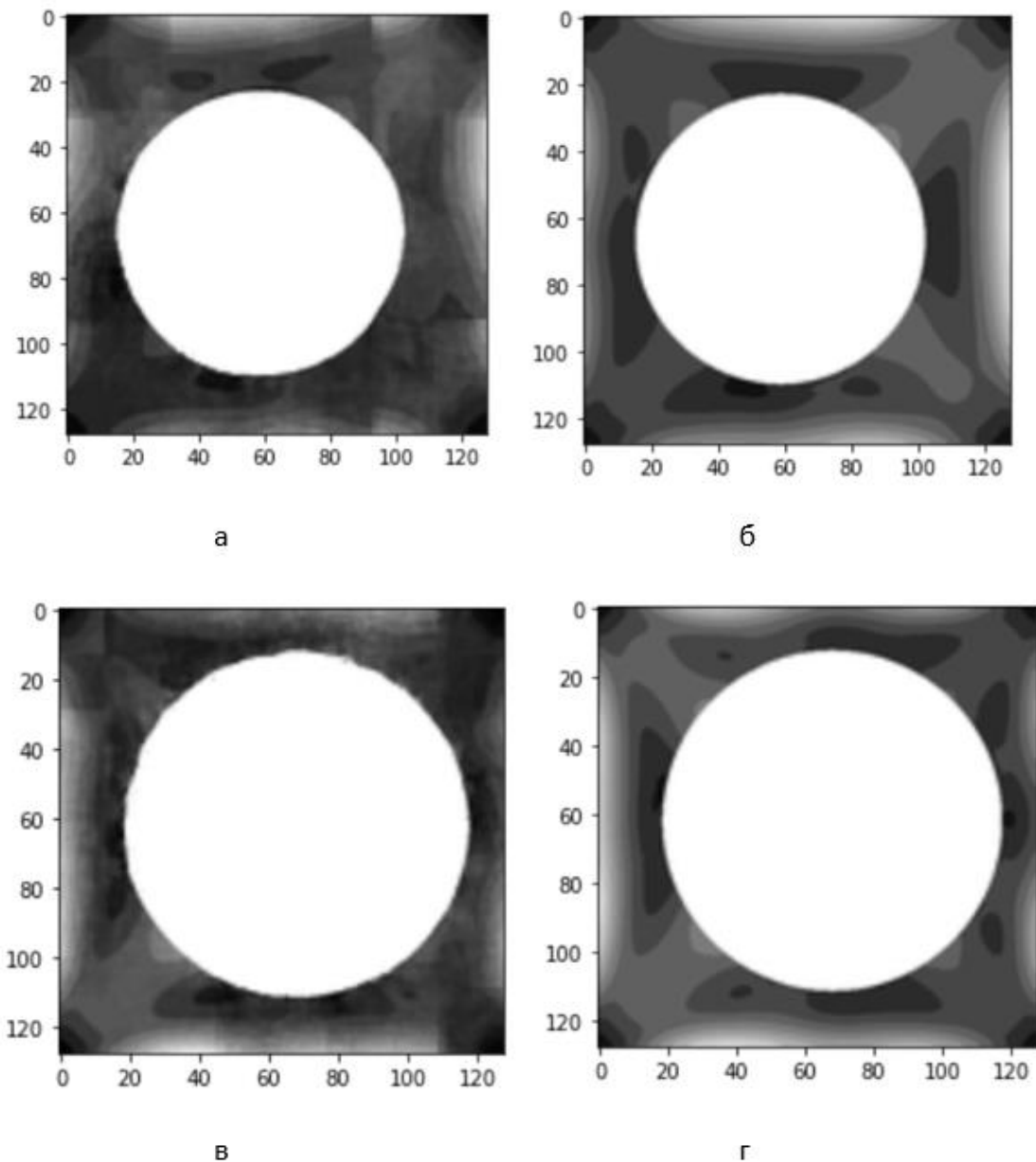


Рисунок 2.44 – Порівняння значень отриманих методом скінченних елементів та з використанням машинного навчання (а, в – машинне навчання, б, г – метод скінченних елементів)

Функцією активації у кодувальнику вибрано ReLU. Далі йдуть 4 шари декодувальника Conv2DTranspose, в яких розмір зображення збільшується поетапно з  $8 \times 8$  до  $128 \times 128$ . Тобто на виході отримуємо зображення, розмір якого співпадає з розміром зображення на вході.

Під час проведення обчислювального експерименту з 2000 епох з такими параметрами:

$$a = 1 \text{ (м);}$$

$$E = 203200 \text{ (МПа)}$$

$$h = 0,01 \text{ (м);}$$

$$q = 0,05 \text{ (МПа);}$$

$$\vartheta = 0,27,$$

отримали зображення квадратної пластини з круглим отвором, на яких світлим кольором позначено напружено-деформовані зони. При порівнянні з результатами, отриманими методом скінченних елементів, можна зробити висовки, що вони практично співпадають (рис. 2.43 – 2.44).

## 2.8 Висновки до розділу 2

У даному розділі запропоновано неромержеві моделі для прогнозування напружено-деформованого стану квадратної, круглої, кільцевої та прямокутної з отвором пластин.

Розроблені алгоритми генерації даних для пластин. Проведено обчислювальні експерименти з базовими архітектурами нейромереж. Проаналізовано вплив функцій активації та алгоритмів активації на штучні нейронні мережі. Проведені дослідження на вплив кількості нейронів та прихованих шарів на нейромержеве прогнозування напружено-деформованого стану пластин.

Розроблено нейронну мережу для прогнозування напружено-деформованого стану прямокутної пластини з круглим отвором та довільними умовами закріплення.

Виявлено, що розроблені нейронні мережі дозволяють прогнозувати значення напружено-деформованого стану з похибкою приблизно 10% відносно аналітичного рішення.

Розроблено модель штучної згорткової нейронної мережі автокодувальника для прогнозування напружено-деформованого стану квадратної пластини з круглим отвором. Виявлено, що результати практично співпадають з результатами отриманими методом скінченних елементів. Переваги даної нейронної мережі у швидкості прогнозування у порівнянні з методом скінченних елементів.

### **3 ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ НАПРУЖЕНО-ДЕФОРМОВАНОГО СТАНУ ОБОЛОНОК**

В авіаційній техніці та космічних конструкціях широко поширене використання об'єктів циліндричної та конічної форми з невеликою товщиною. Такі об'єкти можна моделювати за допомогою оболонок. Тому одним із етапів прогнозування зон руйнування в інженерних конструкціях є прогнозування напружено-деформованого стану оболонок.

#### **3.1 Генерація даних методом скінченних елементів**

Дво- і тривимірні ізопараметричні елементи заслужено вважають вельми ефективними в аналізі напружено-деформованого стану суцільних тіл. Здавалося б, що ці елементи можна безпосередньо застосовувати при розрахунку оболонкових конструкцій, зменшуючи їх розмір в напрямку товщини оболонки. Проте у загальному випадку застосування таких скінченних елементів для моделювання оболонок пов'язане з певними труднощами [108].

По-перше, наявність трьох ступенів свободи в кожному вузлі призводить до великих коефіцієнтів жорсткості для переміщень по товщині оболонки [108], що може бути чинником поганої обумовленості системи рівнянь, якщо товщина оболонки є малою величиною в порівнянні з іншими розмірами скінченного елемента. По-друге, використання декількох вузлів по товщині оболонки є надлишковим. Це пояснює відомий факт, що в ряді випадків для товстих оболонок нормалі до серединних поверхонь можна вважати прямими.

Можливим способом вирішення цих питань є розробка моделі напружено-деформованого стану оболонкової конструкції, в якій оболонку представляють як сукупність плоских елементів. Водночас передбачають, що поведінка неперервної криволінійної поверхні досить точно характеризує поведінку поверхні, складеної з малих плоских елементів. З фізичних міркувань випливає, що зі зменшенням розмірів елемента рішення повинно сходитися і, як свідчить досвід, така збіжність дійсно спостерігається [110].

Нехай діють стандартні гіпотези про малість переміщень і деформацій, а також про відсутність стиснення і збереження нормалей серединної поверхні [110]. При моделюванні оболонок для кожного скінченного елемента локальну систему координат обрано так, щоб площина  $xOy$  збігалася з площиною елемента, вісь  $Oz$  була паралельна нормалі, а початок координат поєднувався з першим вузлом.

Деформований стан скінченного елемента для кожного  $i$ -го вузла однозначно описують переміщеннями  $u_i$  та  $v_i$ , які виникають під дією мембранних сил, прогином  $w_i$  та трьома кутами повороту нормалі:  $\theta_x$ ,  $\theta_y$  і  $\theta_z$ . Водночас потрібно врахувати дві обставини. По-перше, переміщення, викликані мембранними силами, та деформації згину не залежать одини від одного. По-друге, кут повороту  $\theta_z$ , який у локальній системі координат не входить до числа параметрів, що визначають напружено-деформований стан, введено як фіктивний поворот нормалі навколо своєї вісі для забезпечення ансамблю елементів. Водночас вважають, що деформації згину відповідають положенням теорії Рейсснера-Міндліна [110] (теорія зсувних деформацій другого порядку для тонких і середніх пластин і оболонок). У результаті поле передбачуваних переміщень визначається такими співвідношеннями:

$$\begin{cases} u(x, y, z) = u(x, y) + z\theta_x(x, y), \\ v(x, y, z) = v(x, y) + z\theta_y(x, y), \\ w(x, y, z) = w(x, y). \end{cases} \quad (3.1)$$

Вектор узагальнених переміщень у вузлах скінченного елемента набуває наступного вигляду:

$$\{d_e\}^T = \{u_1 v_1 w_1 \theta_{x1} \theta_{y1} \theta_{z1} \cdots u_{n_e} v_{n_e} w_{n_e} \theta_{xn_e} \theta_{yn_e} \theta_{zn_e}\}, \quad (3.2)$$

де  $n_e$  – кількість вузлів в елементі.

Вектор плоских деформацій можна визначити диференціюванням функцій переміщень:

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \{\varepsilon_m\} + z\{\varepsilon_b\} = \begin{Bmatrix} \varepsilon_x^{(m)} \\ \varepsilon_y^{(m)} \\ \gamma_{xy}^{(m)} \end{Bmatrix} + z \begin{Bmatrix} \varepsilon_x^{(b)} \\ \varepsilon_y^{(b)} \\ \gamma_{xy}^{(b)} \end{Bmatrix} = \begin{Bmatrix} u_{,x} \\ v_{,y} \\ u_{,y} + v_{,x} \end{Bmatrix} + z \begin{Bmatrix} \theta_{x,x} \\ \theta_{y,y} \\ \theta_{x,y} + \theta_{y,x} \end{Bmatrix}, \quad (3.3)$$

де  $\{\varepsilon_m\}$  – вектор мембранних деформацій;

$\{\varepsilon_b\}$  – вектор деформацій згину;

$$u_{,x} = \frac{\partial u}{\partial x}, \quad u_{,y} = \frac{\partial u}{\partial y}, \quad v_{,x} = \frac{\partial v}{\partial x}, \quad v_{,y} = \frac{\partial v}{\partial y}, \quad \theta_{x,x} = \frac{\partial \theta_x}{\partial x}, \quad \theta_{x,y} = \frac{\partial \theta_x}{\partial y}, \quad \theta_{y,x} = \frac{\partial \theta_y}{\partial x} \text{ і}$$

$$\theta_{y,y} = \frac{\partial \theta_y}{\partial y} \text{ – похідні переміщень.}$$

Вектор деформацій зсуву можна отримати у такому вигляді:

$$\{\gamma\} = \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{Bmatrix} w_{,x} + \theta_x \\ w_{,y} + \theta_y \end{Bmatrix}, \quad (3.4)$$

$$\text{де } w_{,x} = \frac{\partial w}{\partial x} \text{ і } w_{,y} = \frac{\partial w}{\partial y} \text{ – похідні прогину.}$$

Співвідношення між напруженнями та деформаціями для ізотропного матеріалу можна записати у такому вигляді:

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [D]\varepsilon = \frac{E}{1-\mu^2} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}, \quad (3.5)$$

$$\{\boldsymbol{\tau}\} = \begin{Bmatrix} \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = [\mathbf{G}]\{\boldsymbol{\gamma}\} = \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix}, \quad (3.6)$$

де  $E$  – модуль Юнга;

$\mu$  – коефіцієнт Пуассона;

$G = \frac{E}{2(1+\mu)}$  – модуль зсуву.

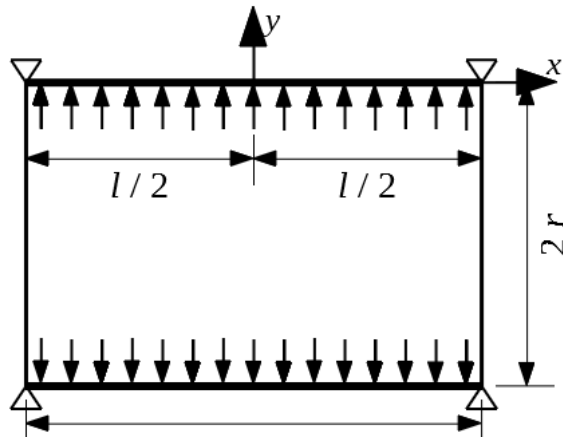


Рисунок 3.1 – Схема обчислювального експерименту

Для обчислення прогину циліндричної оболонки з вільно обпертими торцями (рис. 3.1), яка знаходиться під дією рівномірного внутрішнього тиску можна використати аналітичну формулу:

$$w(x) = \frac{-pl^4}{64D\alpha^4} \left( 1 - \frac{2 \sin \alpha \sinh \alpha}{\cos(2\alpha) + \cosh(2\alpha)} \sin(\beta x) \sinh(\beta x) - \frac{2 \cos \alpha \cosh \alpha}{\cos(2\alpha) + \cosh(2\alpha)} \cos(\beta x) \cos(\beta x) \right), \quad (3.7)$$

де  $\alpha = \frac{\beta l}{2}$ ;

$$\beta = \sqrt[4]{\frac{3(1-\nu^2)}{r^2 h^2}};$$

$$D = \frac{Eh^3}{12(1-\nu^2)};$$

$l$  – довжина оболонки;

$h$  – товщина оболонки;  
 $r$  – радіус оболонки;  
 $E$  – модуль Юнга;  
 $\nu$  – коефіцієнт Пуассона;  
 $p$  – значення внутрішнього тиску.

### 3.2 Нейромережевий метод моделювання напружено-деформованого стану циліндричної оболонки з довільними умовами закріплення

Дана циліндрична оболонка із ізотропного матеріалу з наступними параметрами:

$l \in [0.1; 5]$  (м);  
 $r \in [0.1; 5]$  (м);  
 $h \in [1/100 a; 1/20 a]$  (м), де  $a = \min(l, 2\pi r)$ ;  
 $E \in [50000; 300000]$  (МПа);  
 $\nu \in [0; 0,45]$ ;  
 $q \in [0.01; 0.1]$  (МПа).

Можливі умови закріплення генеруються випадково, також умови закріплення є даними категорій. Можливі наступні випадки умов закріплення: вільні краї, підтримувані краї або зафіксовані краї. Можлива будь-яка комбінація, виключаючи ситуації з двома вільними краями.

Алгоритм генерації даних представлено нижче:

```

function generation( $N$ )
begin
 $D \leftarrow \emptyset$ 
 $ic \leftarrow 0$ 
while  $ic < N$  do
  
```

**begin**

$l \leftarrow \text{random}(0.1,5)$

$r \leftarrow \text{random}(0.1,5)$

$a \leftarrow \min(1, 2\pi r)$

$q \leftarrow \text{random}(0.001,0.1)$

$E \leftarrow \text{random}(50000,300000)$

$\vartheta \leftarrow \text{random}(0,0.45)$

$h \leftarrow \text{random}\left(\frac{a}{100}, \frac{a}{20}\right)$

$\text{mesh} \leftarrow \text{CylinderQuads}(m, \frac{l_n}{\pi r}, r, l)$

**foreach**  $C\_0$  *in*  $[0; 2]$  **do**

**begin**

**foreach**  $C\_1$  *in*  $[0; 2]$  **do**

**begin**

**if**  $C\_1 + C\_1 \geq 1$  **then**

**begin**

$f \leftarrow \left( q \cdot \cos\left(\text{atan}\frac{z}{x}\right), 0, q \cdot \sin\left(\text{atan}\frac{z}{x}\right) \right)$

**function**  $\text{left}(x, y, z)$

**begin**

**if**  $C\_0 = 2$  **and**  $y = 0$  **then return** *ALL*

**else if**  $C\_0 = 1$  **and**  $y = 0$  **then**

**return** *FIRST|SECOND|THIRD*

**return** *FREE*

**end**

**function**  $\text{right}(x, y, z)$

**begin**

**if**  $C\_1 = 2$  **and**  $y = l$  **then return** *ALL*

**else if**  $C\_1 = 1$  **and**  $y = l$  **then**

**return** *FIRST|SECOND|THIRD*

```

return FREE

end

D ← D ∪ MindlinShell(mesh, h, E, ν, left, right, f)
ic ← ic + 1

end

end

end

end

return D

```

де *MindlinShell* – викликає функцію методу скінченних елементів для аналізу оболонок;

*random(a, b)* – функція, яка генерує дані категорій на інтервалі [a; b];

*N* – кількість експериментів.

Приклад напружено-деформованого стану циліндричної оболонки з навчальної вибірки:

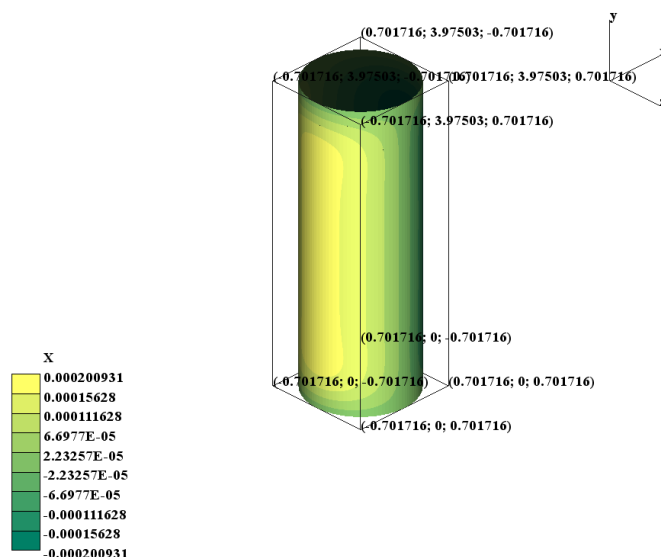


Рисунок 3.2 – Зміна значень  $x$  циліндричної оболонки

Якщо параметри циліндричної оболонки такі:

$$r = 0.7017159727867693,$$

$$l = 3.9750300840707498,$$

$$h = 0.14249240126137896,$$

$$E = 160421.2962090969,$$

$$\vartheta = 0.13400800911476837,$$

$$q = 0.06746708249859512,$$

то у результаті скінченно-елементного аналізу з використанням дискретної моделі, що включає 18100 вузлів, 18000 елементів, картини напружено-деформованого стану матимуть вигляд (рис. 3.2 – 3.8):

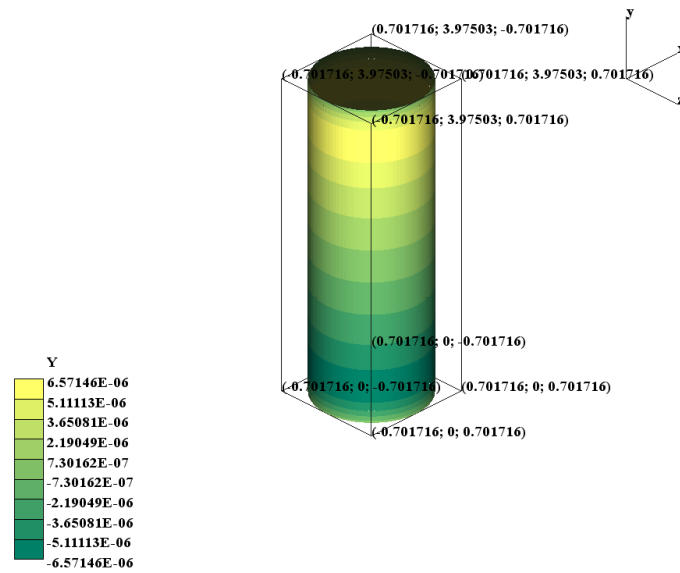


Рисунок 3.3 – Зміна значень у циліндричній оболонки

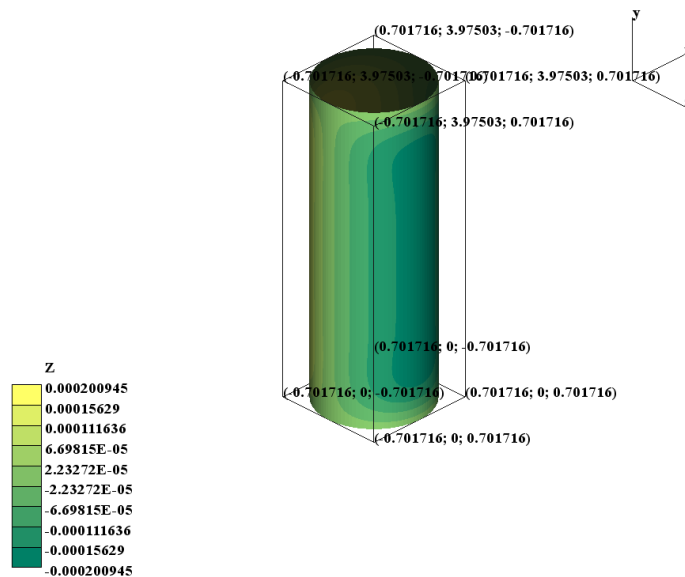


Рисунок 3.4 – Зміна значень z з циліндричній оболонки

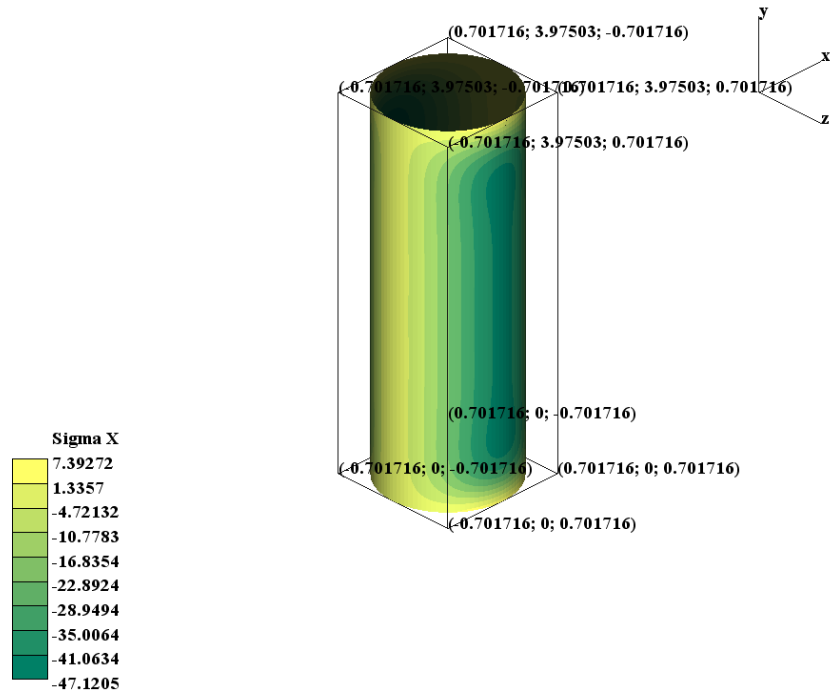


Рисунок 3.5 – Зміна значень  $\sigma_x$  циліндричної оболонки

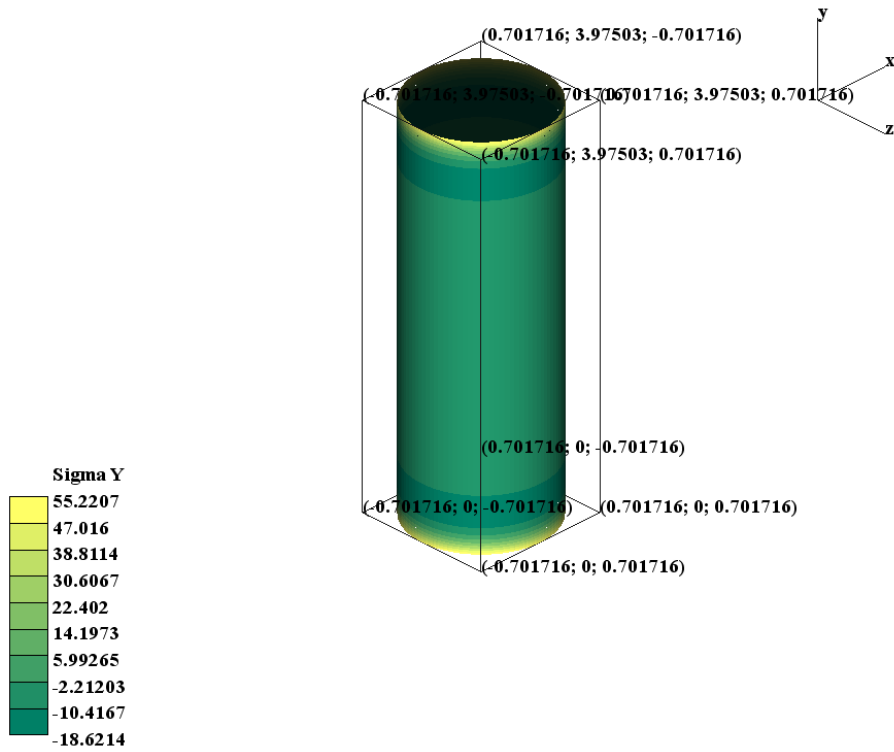


Рисунок 3.6 – Зміна значень  $\sigma_y$  циліндричної оболонки

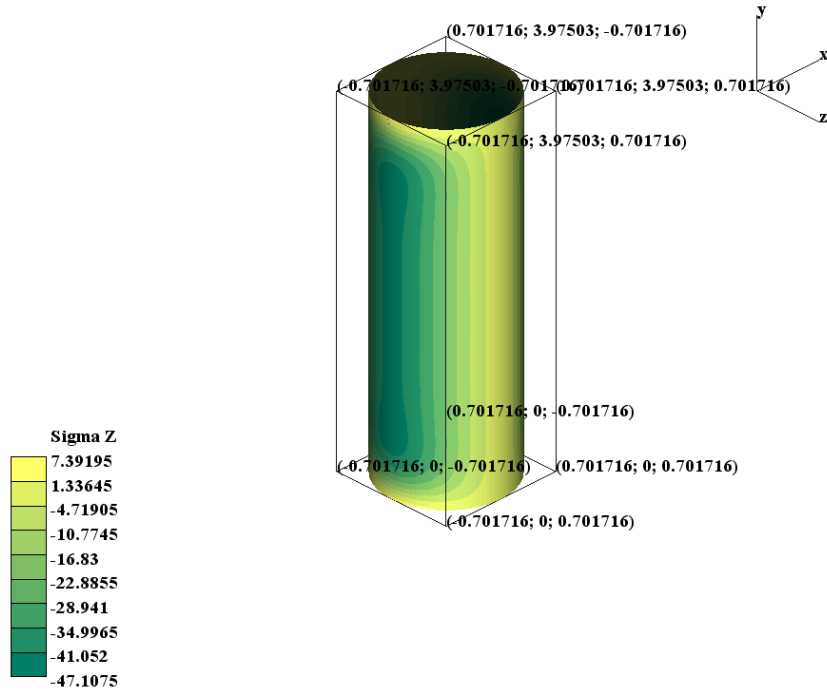


Рисунок 3.7 – Зміна значень  $\sigma_z$  циліндричної оболонки

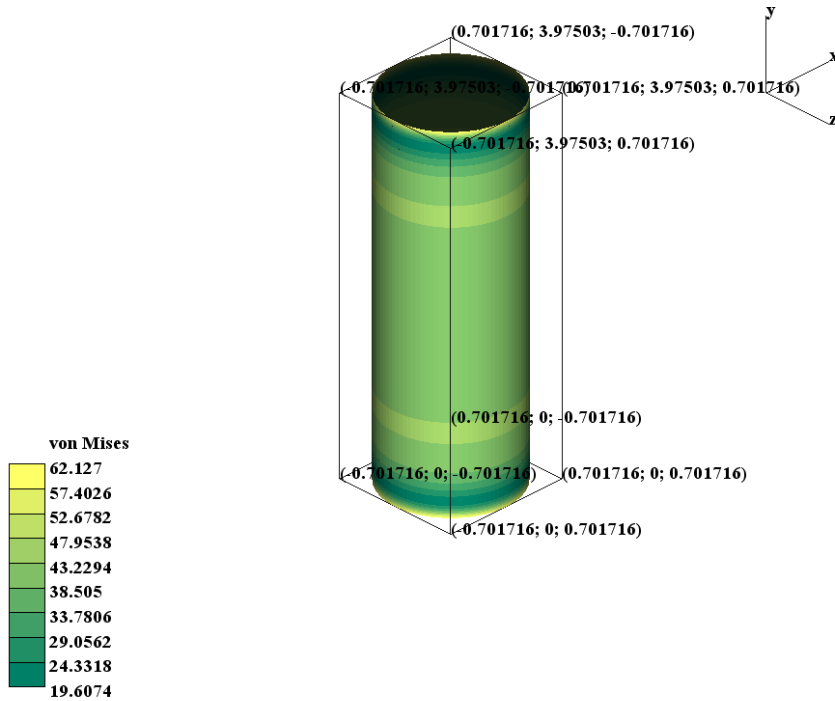


Рисунок 3.8 – Інтенсивність напружень за Мізесом

### 3.2.1 Архітектура нейронної мережі

Модель штучної нейронної мережі складається з двох гілок вхідних нейронів (рис. 3.9): чисельні дані та дані категорій. Кожна гілка може складатися з деяких прихованих шарів нейронів. Потім гілки об'єднуються в один шар. В результаті останній прихований шар об'єднується з останнім – зовнішнім вихідним шаром.

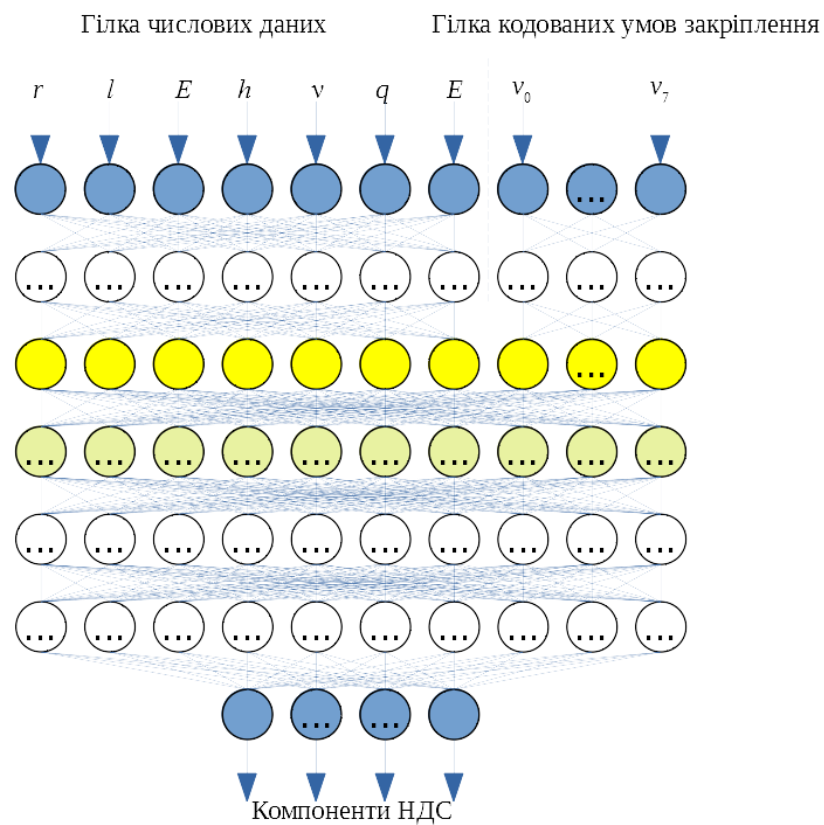


Рисунок 3.9 – Модель розробленої нейронної мережі

### 3.2.2 Обчислювальний експеримент

Обчислювальний експеримент був проведений на 2000 епохах. Рисунки 3.10, 3.11 показують середньоквадратичну похибку для тестування та тренування та абсолютну відсоткову похибку.

Отже, розроблена штучна нейронна мережа дозволяє прогнозувати напружено-деформований стан циліндричної оболонки із ізотропного матеріалу. Розроблений вигляд штучної нейронної мережі дозволяє тренувати її з кращою збіжністю. Абсолютна відсоткова похибка становить не більше 20%.

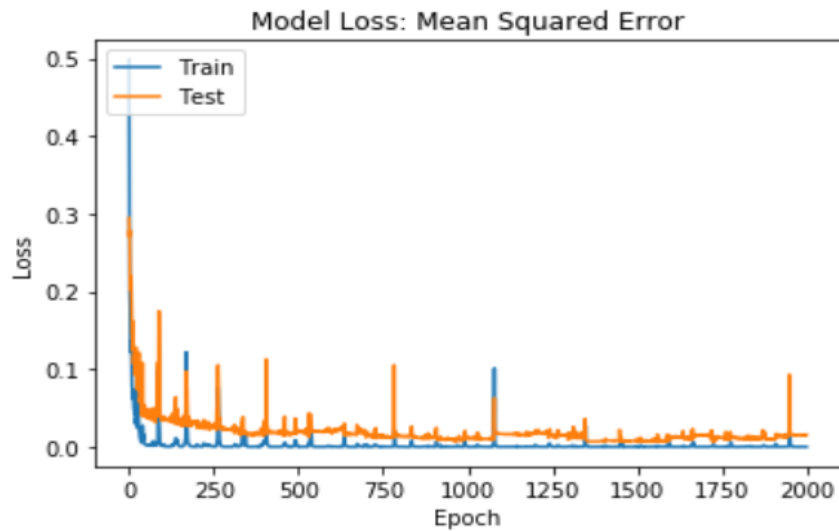


Рисунок 3.10 – Середньоквадратична похибка моделі

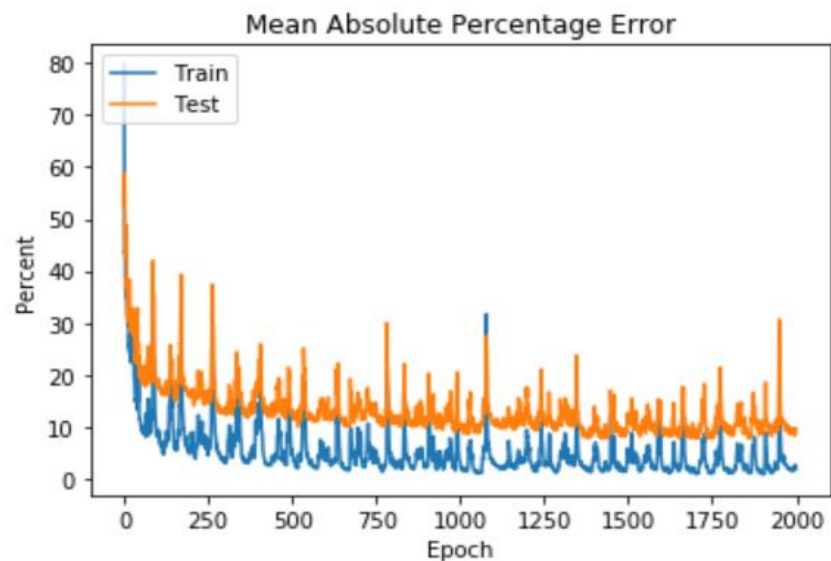


Рисунок 3.11 – Абсолютна відсоткова похибка

### 3.3 Нейромережевий метод моделювання напружено-деформованого стану комбінації циліндр-конус оболонок з довільними умовами закріплення

Розглянемо випадок об'єднання циліндричної та конічної оболонок із ізотропного матеріалу з наступними параметрами:

$$r_0 \in [1; 3] \text{ (м)};$$

$$r_1 \in [0.1; 0.5] \text{ (м)};$$

$$l_{cone} \in [2; 4] \text{ (м)};$$

$$l_{cylinder} \in [5; 7] \text{ (м)};$$

$$h \in \left[ \frac{r_0}{100}; \frac{r_0}{20} \right] \text{ (м) [116]};$$

$$E \in [50000; 300000] \text{ (МПа)};$$

$$\vartheta \in [0; 0.45];$$

$$q \in [0.001; 0.1] \text{ (МПа)}.$$

Можливі умови закріплення генеруються випадково, також умови закріплення є даними категорій. Можливі наступні випадки умов закріплення: вільні краї, підтримувані краї або зафіксовані краї. Можлива будь-яка комбінація, виключаючи ситуації з двома вільними краями.

Приклад напружено-деформованого стану оболонки-комбінації циліндр-конус з навчальної вибірки. Наприклад, якщо:

$$r_0 = 1.8991808043792844;$$

$$r_1 = 0.213331504072994;$$

$$l_{cone} = 3.858336561359465;$$

$$l_{cylinder} = 6.0348932249471545;$$

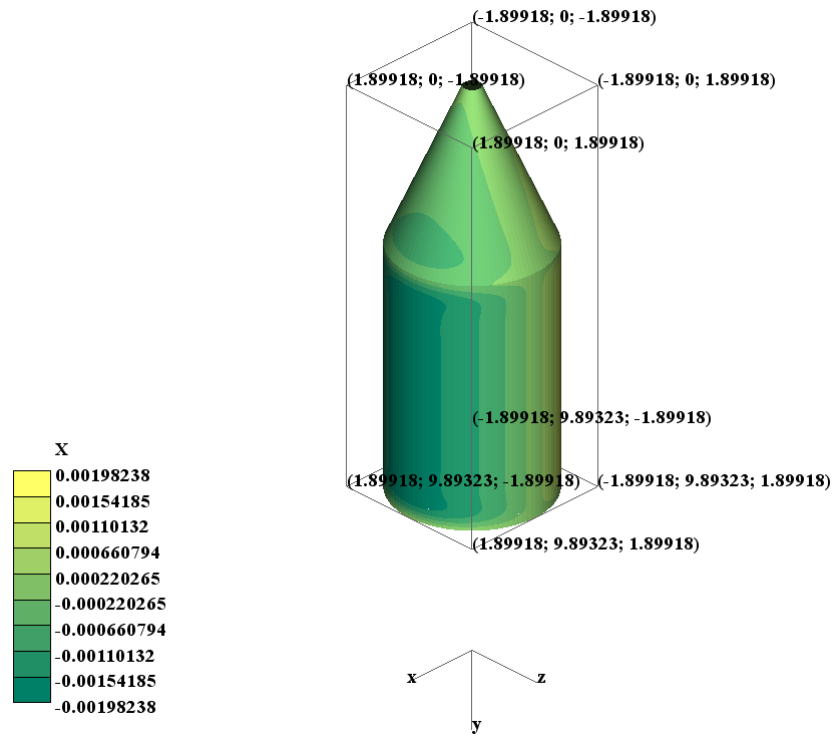
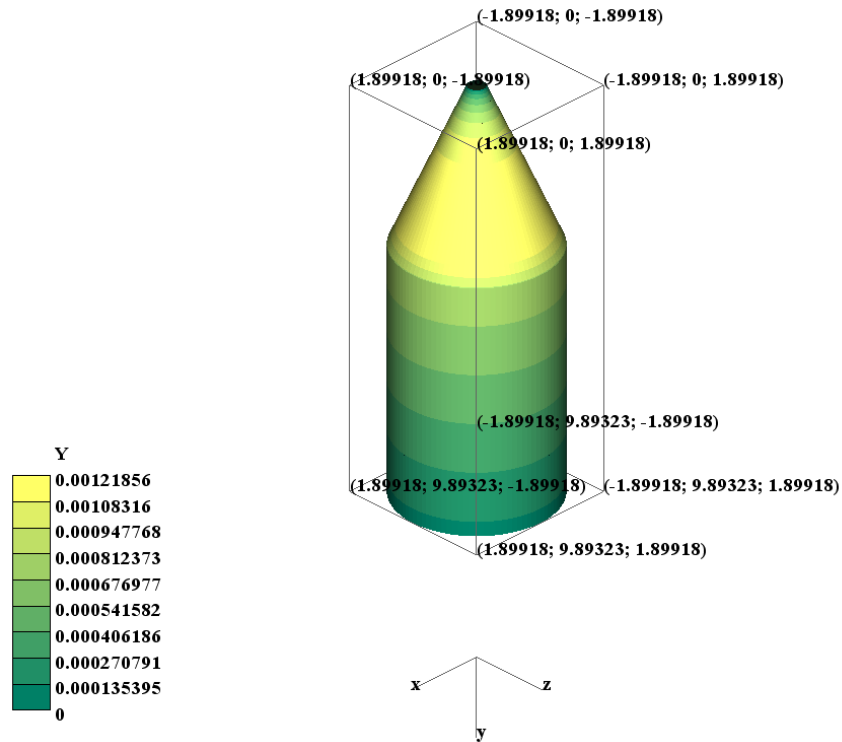
$$h = 0.020135312849976893;$$

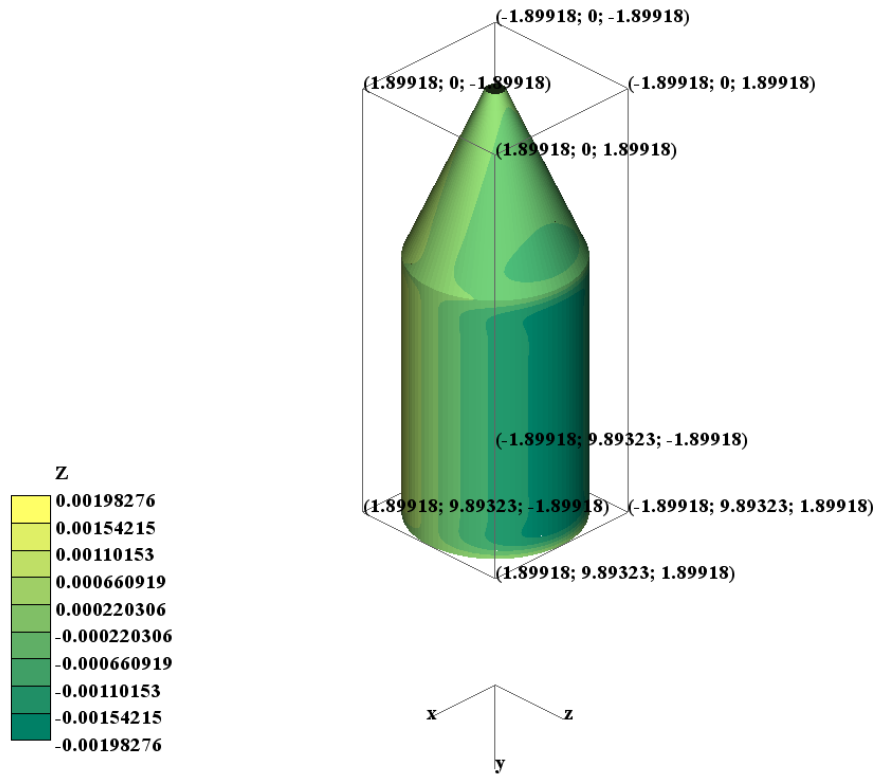
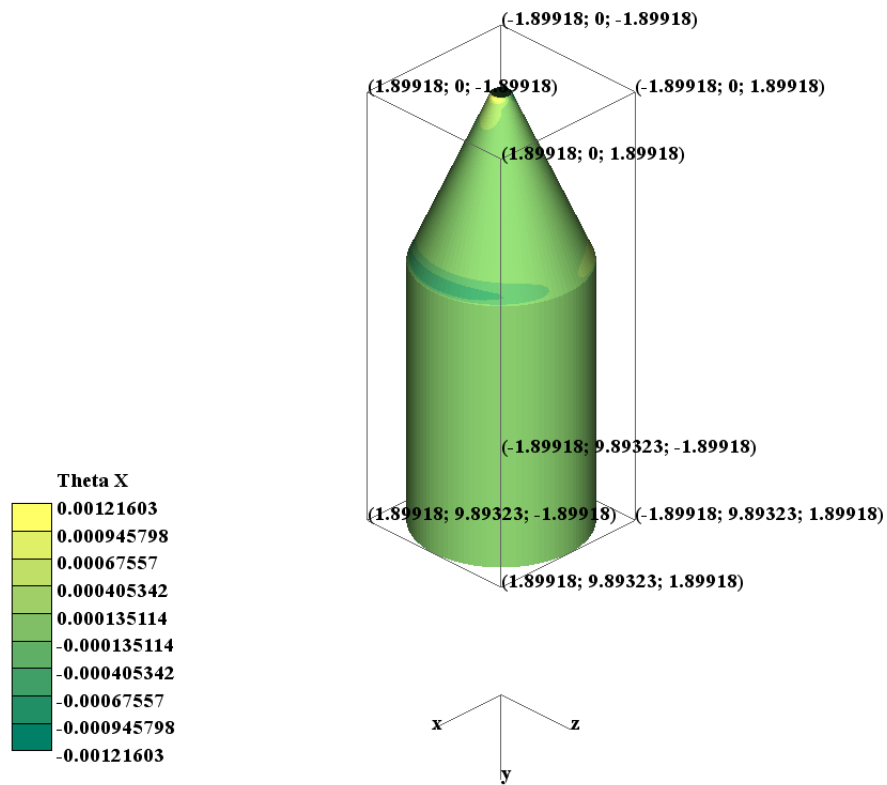
$$E = 288684.7013258375;$$

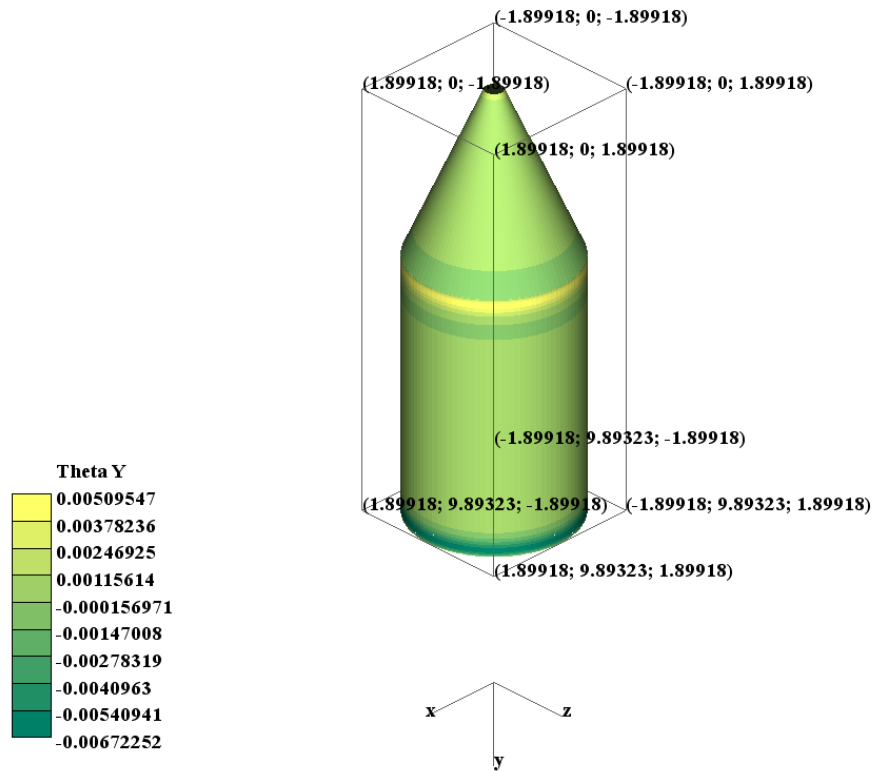
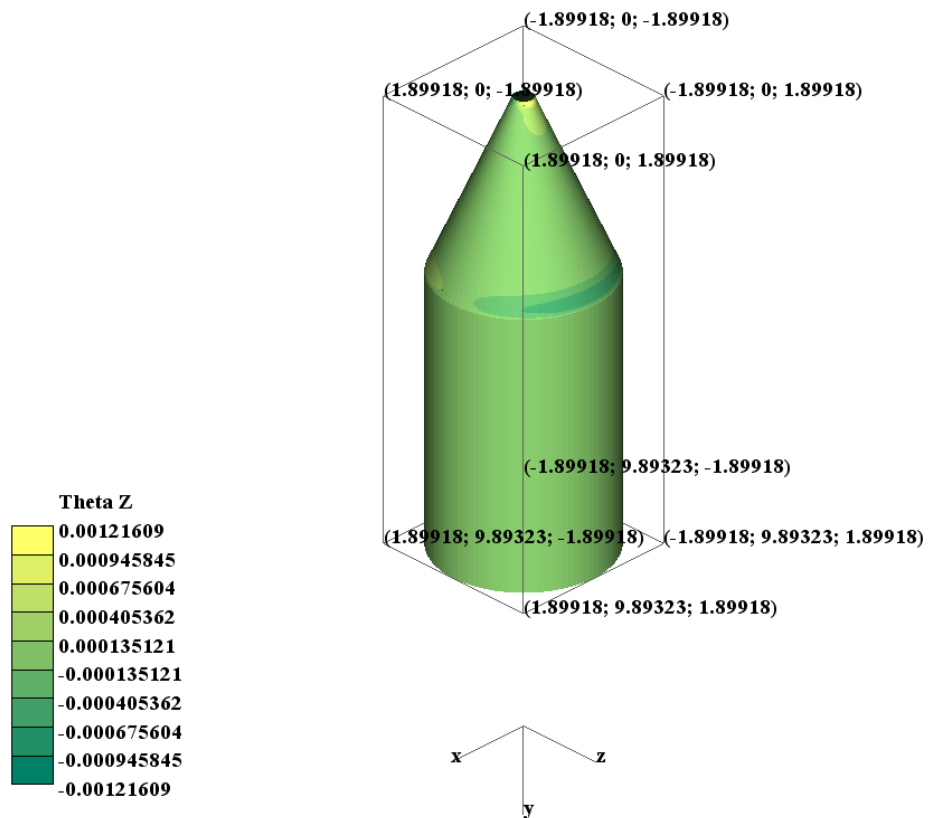
$$\vartheta = 0.16687570314388722;$$

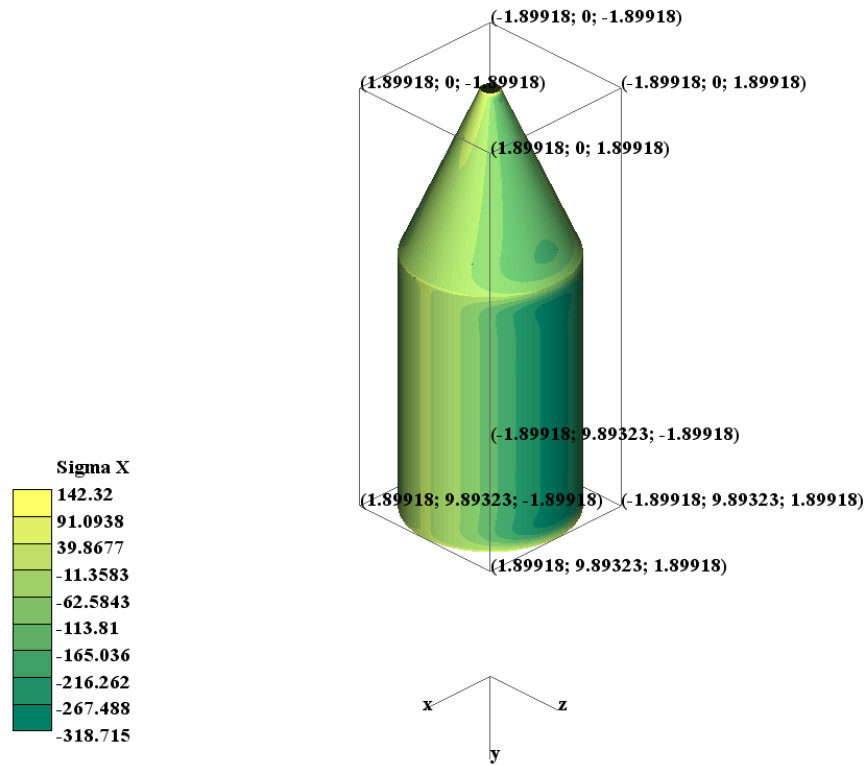
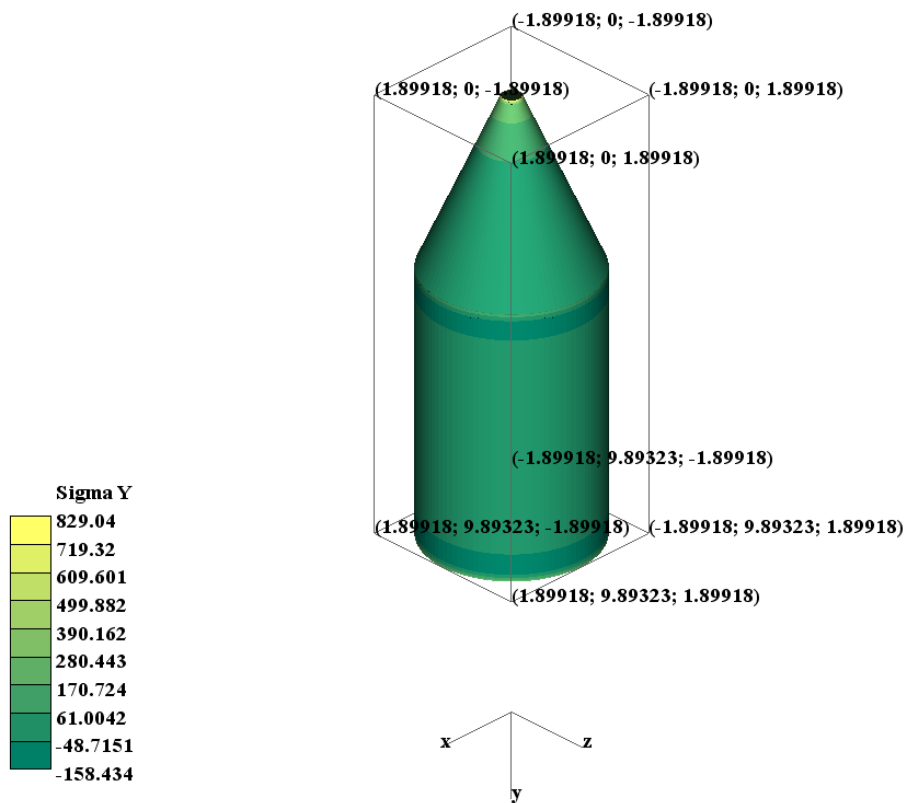
$$q = -0.04492475583637134.$$

у результаті скінченни-елементного аналізу на дискретній моделі з 26700 вузлів та 26600 елементів отримали наступні результати (рис. 3.12-3.21)

Рисунок 3.12 – Зміна значень по  $x$  в об'єднанні оболонокРисунок 3.13 – Зміна значень по  $y$  в об'єднанні оболонок

Рисунок 3.14 – Зміна значень по  $Z$  в об'єднанні оболонокРисунок 3.15 – Зміна значень по  $\theta_x$  в об'єднанні оболонок

Рисунок 3.16 – Зміна значень по  $\theta_y$  в об'єднанні оболонокРисунок 3.17 – Зміна значень по  $\theta_z$  в об'єднанні оболонок

Рисунок 3.18 – Зміна значень по  $\sigma_x$  в об'єднанні оболонокРисунок 3.19 – Зміна значень по  $\sigma_y$  в об'єднанні оболонок

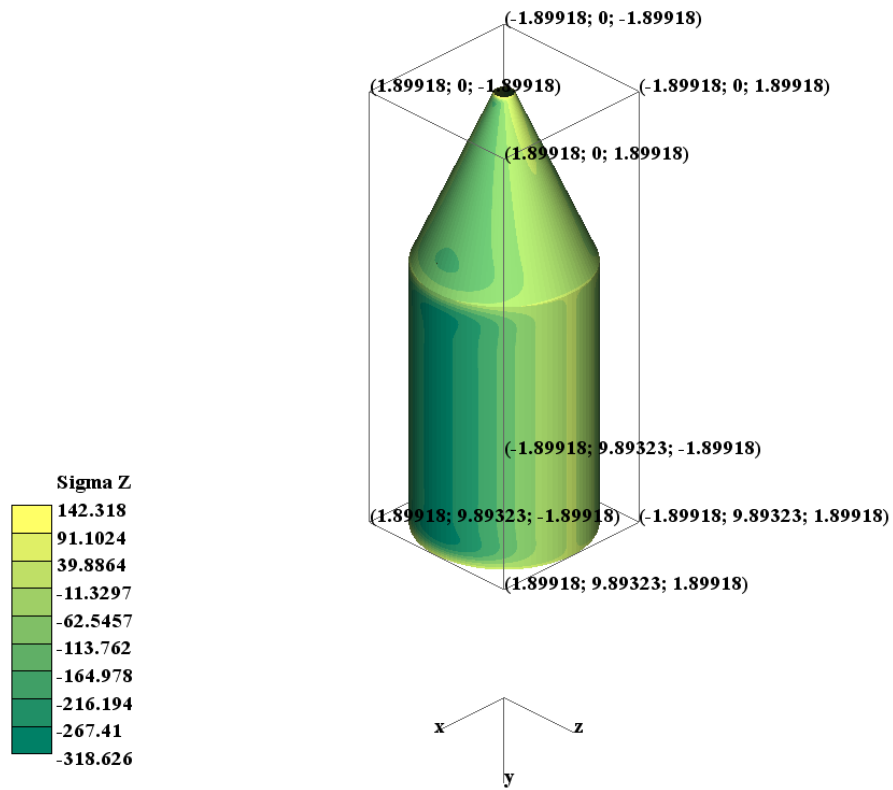
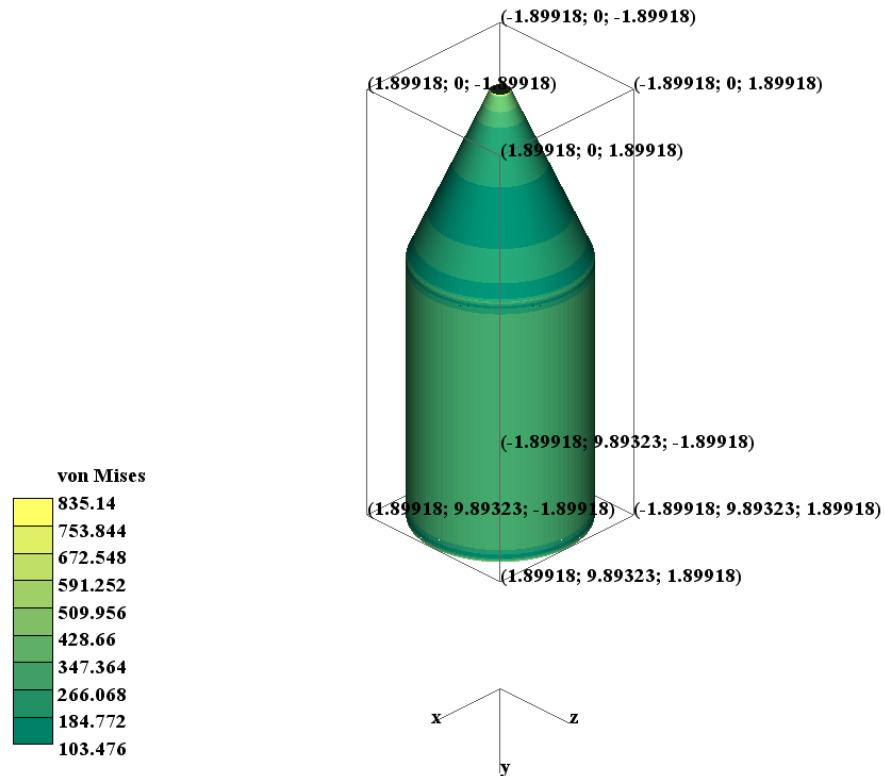
Рисунок 3.20 – Зміна значень по  $\sigma_z$  в об'єднанні оболонок

Рисунок 3.21 – Інтенсивність напружень на Мізесом

### 3.3.1 Архітектура нейронної мережі для прогнозування напружено-деформованого стану комбінації циліндр-конус

Розроблена для даного випадку модель нейронної мережі має наступний вигляд (рис. 3.22):

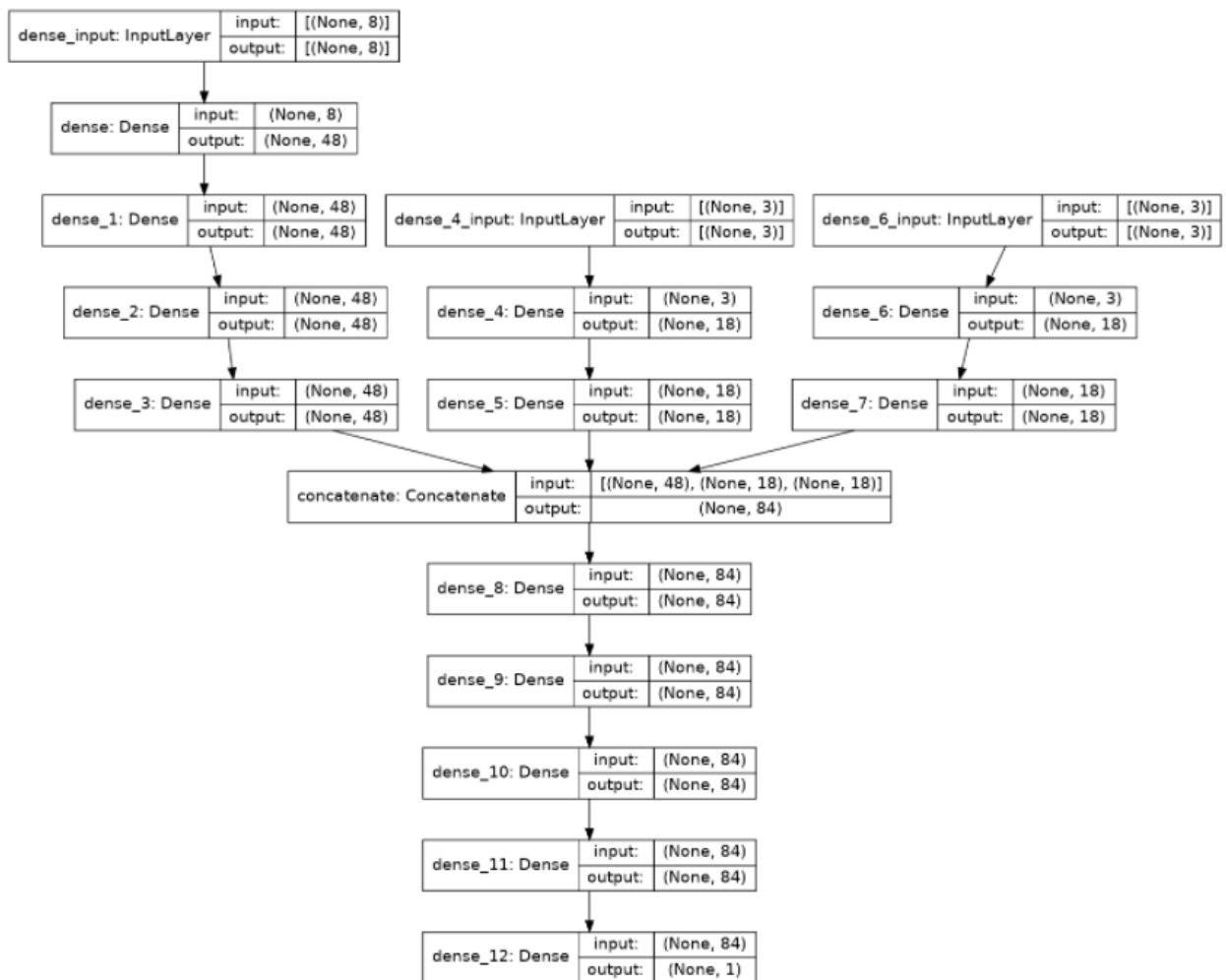


Рисунок 3.22 – Модель нейронної мережі

Вхідний шар поділений на 3 гілки. До першої гілки відносяться числові дані, тобто вхідний зовнішній шар містить 8 нейронів. Друга та третя гілки – дані категорій. Дані категорій – умови закріплення країв оболонки, тобто можливі випадки вільних країв, затиснутих або підтримуваних країв. У другій та третій гілках вхідні шари містять по 3 нейрони. Кожна гілка має приховані внутрішні шари. У першій – 4 приховані

шари по 48 нейронів у кожній. У другій та третій – по 2 шари з 18 нейронами. Потім відбувається об'єднання трьох гілок і отримуємо внутрішній шар з 84 нейронами. Нейронна мережа має ще 4 внутрішні приховані шари по 84 нейрони у кожному. На виході маємо 1 зовнішній шар з 1 нейроном.

### 3.3.2 Обчислювальний експеримент

Обчислювальний експеримент проведений на 1000 епохах показав наступні результати (рис. 3.6):

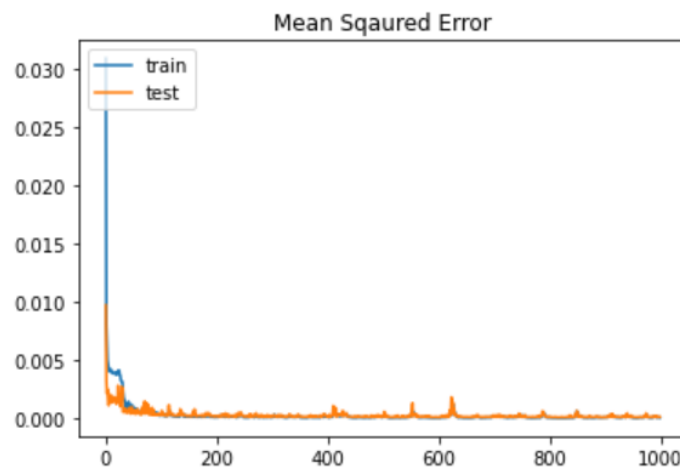


Рисунок 3.6 – Середньоквадратична похибка

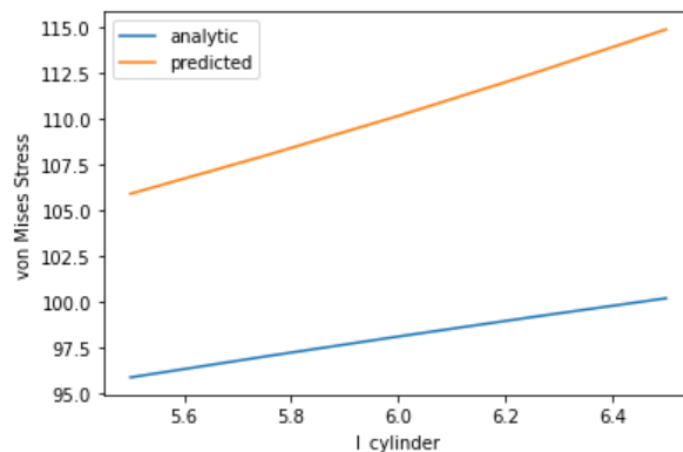


Рисунок 3.7 – Значення інтенсивності напружень за Мізесом

### 3.4 Висновки до розділу 3

У даному розділі запропоновано нейромережеві моделі для прогнозування напружено-деформованого стану циліндричної оболонки та об'єднання циліндричної та конічної оболонок.

Розроблені алгоритми генерації даних для оболонок. Проведено обчислювальні експерименти з базовими архітектурами нейромереж. Проаналізовано вплив функцій активації та алгоритмів активації на штучні нейронні мережі. Проведені дослідження на вплив кількості нейронів та прихованих шарів на нейромережеве прогнозування напружено-деформованого стану оболонок.

Розроблено нейронну мережу для прогнозування напружено-деформованого стану циліндричної оболонки з довільними умовами закріплення, а також випадку об'єднання циліндричної та конічної оболонок з довільними умовами закріплення.

Виявлено, що розроблені нейронні мережі дозволяють прогнозувати значення напружено-деформованого стану з похибкою приблизно 10% відносно аналітичного рішення.

## 4 ЗАСТОСУВАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ У ЗАДАЧАХ ПРОГНОЗУВАННЯ

### 4.1 Особливості застосування генетичних алгоритмів для оптимізації архітектури нейронної мережі

В практичних задачах саме структурна та параметрична оптимізація вимагає суттєвих часових затрат та експертних знань предметної області. Одним з методів, що дозволяє уникнути такого ручного налаштування роботи системи є генетичні алгоритми.

Для розробки генетичного алгоритму для оптимізації гіперпараметрів моделі необхідно кодувати їх у цілочисельну хромосому  $V = (v_0, v_1, \dots, v_{14})$ :

- Кількість нейронів у вхідному шарі на гілці з чисельними даними ( $v_0$ );
- Функція активації вхідного шару з числовими даними ( $v_1$ );
- Кількість прихованих шарів на гілці з числовими даними ( $v_2$ );
- Кількість нейронів у прихованих шарах з числовими даними ( $v_3$ );
- Функція активації у прихованих шарах з числовими даними ( $v_4$ );
- Кількість нейронів у вхідному шарі на гілці з даними категорій ( $v_5$ );
- Функція активації у вхідному шарі на гілці з даними категорій ( $v_6$ );
- Кількість прихованих шарів на гілці з даними категорій ( $v_7$ );
- Кількість нейронів у прихованих шарах на гілці з даними категорій ( $v_8$ );
- Функція активації у прихованих шарах на гілці з даними категорій ( $v_9$ );
- Кількість нейронів в об'єднаних шарах ( $v_{10}$ );
- Функція активації у прихованих об'єднаних шарах ( $v_{11}$ );

- Кількість прихованих шарів після об'єднання двох гілок з даними ( $v_{12}$ );
- Кількість нейронів у прихованих шарах після об'єднання ( $v_{13}$ );
- Функція активації у прихованих шарах ( $v_{14}$ );

Модель має 9 вхідних параметрів на гілці з числовими даними та 76 параметрів на гілці з даними категорій. Можна припустити, що гілки мають шари з кількістю нейронів, що дорівнює кількості вхідних параметрів; максимальна кількість прихованих шарів може бути 10; максимальна кількість нейронів може бути обмежена мінімальним значенням помноженим на 3; мінімальна кількість нейронів у прихованих шарах після злиття може бути встановлена на рівні 25% від загальної кількості вхідних сигналів.

#### **algorithm genetic**

**begin**

$epoch \leftarrow 0$

initialize( $P_{epoch}$ )

**while not stop-criterion do**

**begin**

$F_{epoch} \leftarrow \text{fitness}(P_{epoch})$

$P_{epoch} \leftarrow P_{epoch}[\text{argsort}(F_{epoch})]$

$epoch \leftarrow epoch + 1$

$P_{epoch} \leftarrow \text{crossover}(P_{epoch-1})$

$P_{epoch} \leftarrow \text{mutate}(P_{epoch})$

**end**

**end**

Найпоширенішими функціями активації є тангенс гіперболічний, сигмоїда та ReLU. Можна позначити функції активації так: 0 – tanh, 1 –

sigmoid, 2 – relu. Отже, наступні обмеження на значення хромосом можуть бути використані для формування популяції:  $v_0, v_3 \in [n_f; 3n_f]$ ,  $v_1, v_4, v_6, v_9, v_{11}, v_{14} \in [0; 2]$ ,  $v_2, v_7, v_{12} \in [0; 10]$ ,  $v_3, v_8 \in [n_c; 3n_c]$ ,  $v_{10} \in [n_f + n_c; 3(n_f + n_c)]$ ,  $v_{13} \in [\frac{n_f+n_c}{4}; 3(n_f + n_c)]$ , де  $n_f = 9$  та  $n_c = 76$ .

Схема генетичного алгоритму представлена вище.

Функція втрат моделі використовується як вихід фітнес функції. Фітнес функція будує модель, тренує її за допомогою набору даних і повертає значення функції втрат для тренованої моделі. Алгоритм фітнес функції показаний нижче:

```

function fitness (V)
begin
    f ← dense(units = V0, activation = V1)(F)
    for i ← 0 to V2 do
        begin
            f ← dense(units = V3, activation = V4)(f)
        end
        c ← dense(units = V5, activation = V6)(C)
        for i ← 0 to V7 do
            begin
                c ← dense(units = V8, activation = V9)(c)
            end
        m ← dense(units = V10, activation = V11)(f ∪ c)
        for i ← 0 to V12 do
            begin
                m ← dense(units = V13, activation = V14)(m)
            end
        model ← dense(units = 0, activation = linear)(m)
    train(model)

```

```
return mse( $m$ )
```

Шар ( $units, activation$ )( $I$ ) позначає повністю пов'язаний шар, де  $units$  кількість нейронів,  $activation$  це функція активації,  $I$  вхідний сигнал у шарі;  $F$  вхідний сигнал шару з числовими даними та  $C$  вхідний сигнал шару з даними категорій;  $O$  позначає кількість вихідних сигналів;  $linear$  позначає лінійну функцію активації; функція  $mse(m)$  середньоквадратична похибка в тренувальній моделі.

Кросовер поєднує гени популяції. Кросовер відбирає хромосому з найкращою фізичною формою та поєднує її з іншими хромосомами. Це випадково відібрані гени кожної хромосоми за генами найкращої хромосоми. Алгоритм кросоверу представлений нижче:

```
function crossover( $P$ )
```

```
begin
```

```
  for  $i \leftarrow 1$  to  $N-1$  do
```

```
    begin
```

```
       $idx \leftarrow \text{choise}(N, \text{swap})$ 
```

```
       $P_i[idx] \leftarrow P_0[idx]$ 
```

```
    end
```

```
return  $P$ 
```

де,  $\text{choise}(N, \text{swap})$  генерує список  $\text{swap}$  обміну випадковими цілими числами без дублікатів, кожне ціле число є значенням з інтервалу  $[0; N - 1]$ ;  $N$  розмір популяції;  $\text{swap}$  позначає кількість генів, взятих з найкращої хромосоми.

Щоб уникнути передчасного зближення ми використовуємо рівномірну мутацію. Мутація поєднує найгірші  $M$  хромосоми з новими випадковими хромосомами, подібними до кросоверу. В алгоритмі хромосома () генерує нову випадкову хромосому.

```
function mutate( $P$ )
```

```
begin
```

```

for  $i \leftarrow N - M - 1$  to  $N - 1$  do
begin
     $idx \leftarrow \text{choise}(N, \text{swap})$ 
     $C \leftarrow \text{chromosome}()$ 
     $P_i[idx] \leftarrow C[idx]$ 
end

return  $P$ 

```

## 4.2 Генетична оптимізація архітектури нейронної мережі для моделювання напружено-деформованого стану прямокутної пластини

Дана прямокутна пластинка, яка має наступні параметри з обмеженнями:

- $width \in [0.1; 10]$  (м);
- $height \in [0.1; 10]$  (м);
- $r \in [3s; \frac{1}{2}a - 3s]$  (м), де  $s$  - це розмір фонові комірки в процедурі обміну даними ( $s = \frac{\max(width, height)}{n}$ ,  $n^2$  кількість комірок),  $a = \min(width, height)$ ;
- $x_0 \in [0; \frac{1}{2}a - r - 3s]$  (м);
- $y_0 \in [0; \frac{1}{2}a - r - 3s]$  (м);
- $h \in [\frac{1}{100}a; \frac{1}{20}a]$  (м);
- $E \in [50000; 300000]$  (МПа);
- $\nu \in [0; 0,45]$ ;
- $q \in [0.01; 0.1]$  (МПа).

Граничні умови застосовуються до країв пластини. У числових експериментах граничні умови також генеруються випадковим чином. Для

кожного краю пластини може бути застосована одна з наступних граничних умов: вільний край, підтримуваний край, затиснутий (фіксований) край. Отже, граничні умови можуть бути оброблені як дані категорій. Будь-яка комбінація граничних умов можлива за винятком комбінацій чотирьох вільних ребер або одного підтримуваного ребра та трьох вільних ребер.

Нехай  $c_0$  гранична умова, що відноситься до ребра  $x = -\frac{width}{2}$ ,  $c_1$  гранична умова, що відноситься до ребра  $y = -\frac{height}{2}$ ,  $c_2$  гранична умова, що відноситься до ребра  $x = \frac{width}{2}$ , та  $c_3$  гранична умова, що відноситься до ребра  $y = \frac{height}{2}$  (рис. 4.1). Якщо позначити вільний край як 0, підтримуваний край як 1, затиснутий край як 2, тоді отримаємо наступну умову:  $c_1 + c_2 + c_3 + c_4 \geq 2$ . Граничні умови включають 76 можливих комбінацій

Множина даних складається з 40,000 випадково згенерованих записів. Ми виключаємо записи, які не відповідають вимогам лінійної (якщо прогин не відповідає інтервалу від  $10^{-5}width$  до  $10^{-5}$  до  $\frac{1}{5}width$ ).

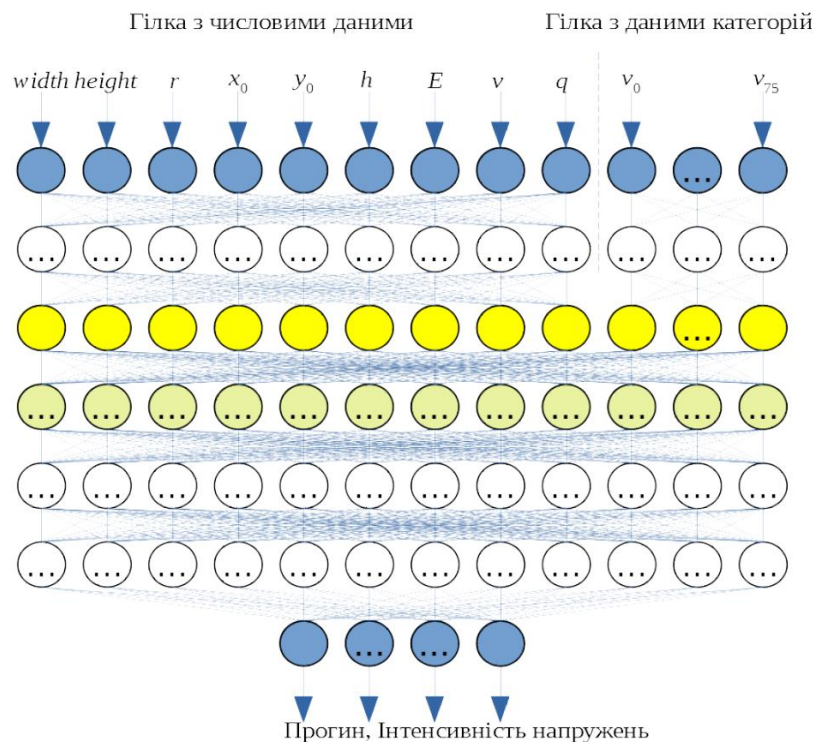


Рисунок 4.2 – Модель штучної нейронної мережі

Розроблена модель штучної нейронної мережі включає дві гілки вхідних нейронів (рис. 4.2), в одній знаходяться числові значення, а в іншій – дані категорій. Кожна гілка може включати декілька прихованих шарів нейронів. Потім гілки об'єднуються, об'єднуючи вихідні приховані шари для чисельної та категоріальної обробки даних. Довільна кількість прихованих щільних шарів може слідувати за шаром для об'єданого виводу.

#### 4.2.1 Обчислювальний експеримент

В обчислювальному експерименті було використано популяцію з 10 хромосом з 15 генами. Встановлено  $M = 7$ . Алгоритм зупиняється якщо найкращі 3 хромосом не змінювались хоча б у 50 популяціях.

Алгоритм зупиняється після 100 популяцій. Можна зробити висновок, що генетичний алгоритм значно зменшує середню квадратичну помилку в моделі (рис 4.3).

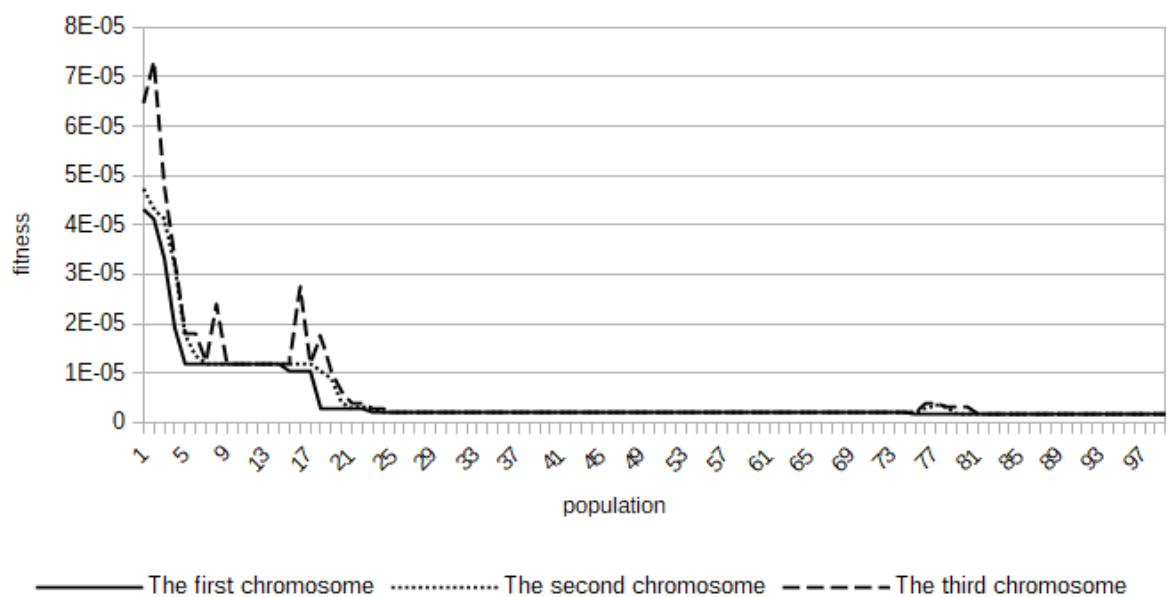


Рисунок 4.3 – Три найкращі хромосоми

Рисунки 4.4 – 4.7 показують як змінювалась кількість нейронів у різних шарах. Рисунок 4.8 – відображає популяції, що використовують аналіз основних компонентів для зменшення розмірів.

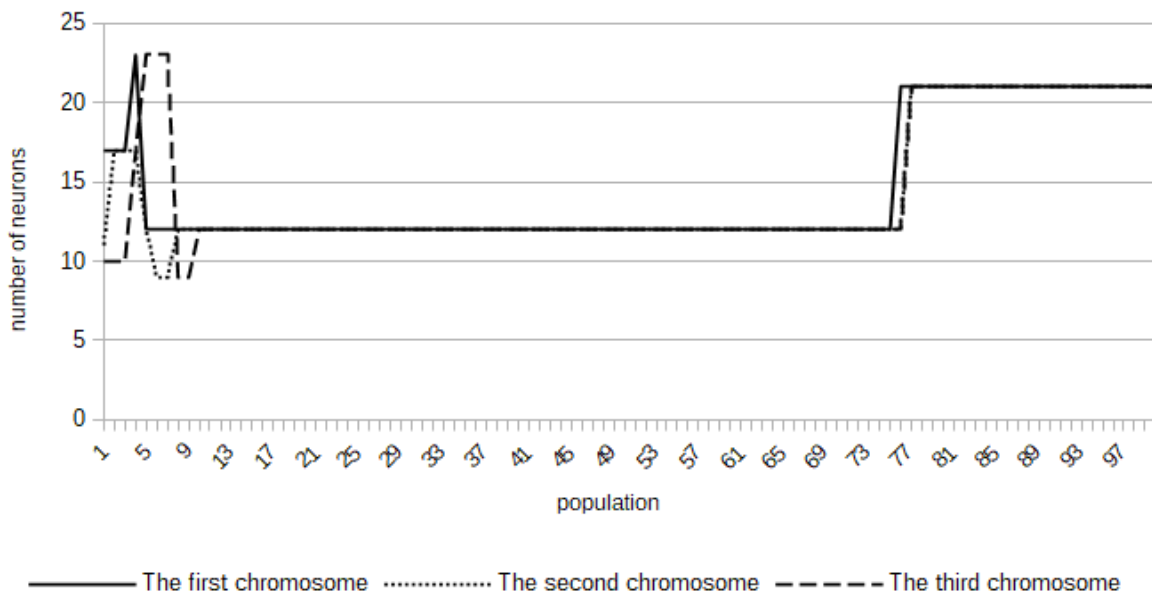


Рисунок 4.4 – Кількість нейронів у вхідному шарі на гілці з числовими даними

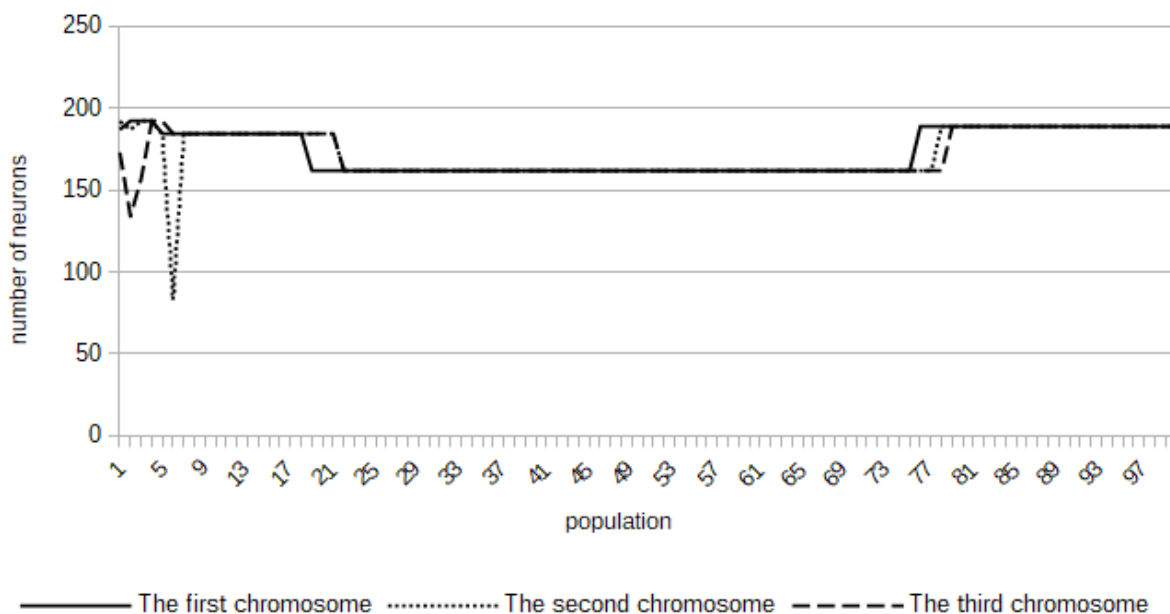


Рисунок 4.5 – Кількість нейронів у вхідному шарі гілки з даними категорій

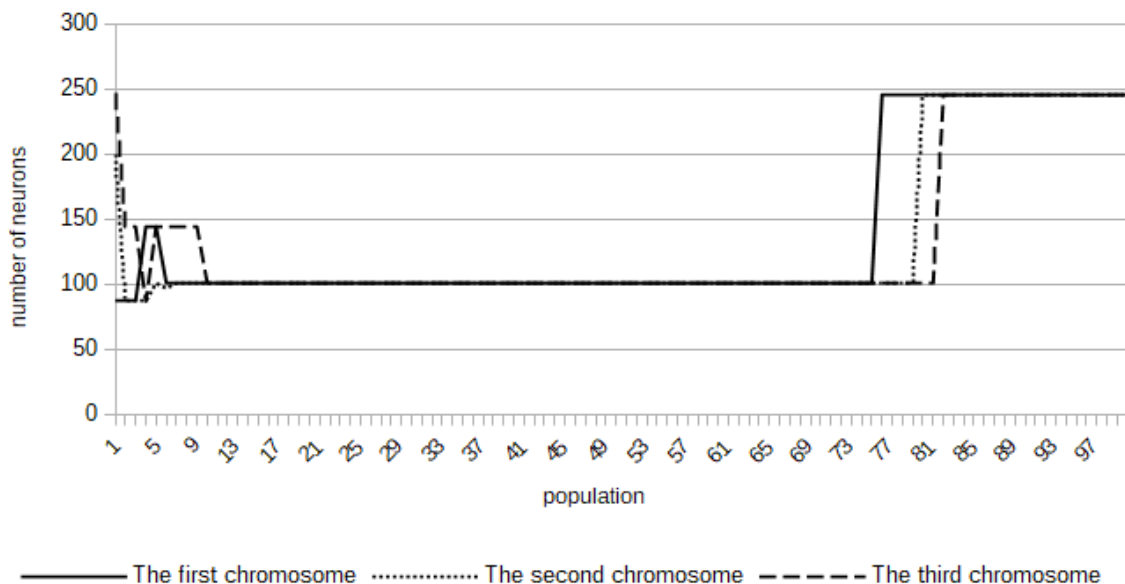


Рисунок 4.6 – Кількість нейронів в об'єднаному шарі

Найкращі хромосоми – (21, 0, 0, 22, 2, 189, 2, 0, 139, 1, 245, 2, 3, 192, 2). Хромосома визначає модель з 21 нейроном на вхідному шарі, з функцією активації гіперболічний тангенсі та 0 прихованими шарами у гілці числових даних; 189 нейронів на вхідному рівні, функція активації ReLU та 0 прихованих шарів у гілці категоріальних даних; 245 нейронів та функція активації ReLU у шарі злиття; 3 прихованих шари з 192 нейронами та функцією активації ReLU після шару злиття (рис. 4.9).

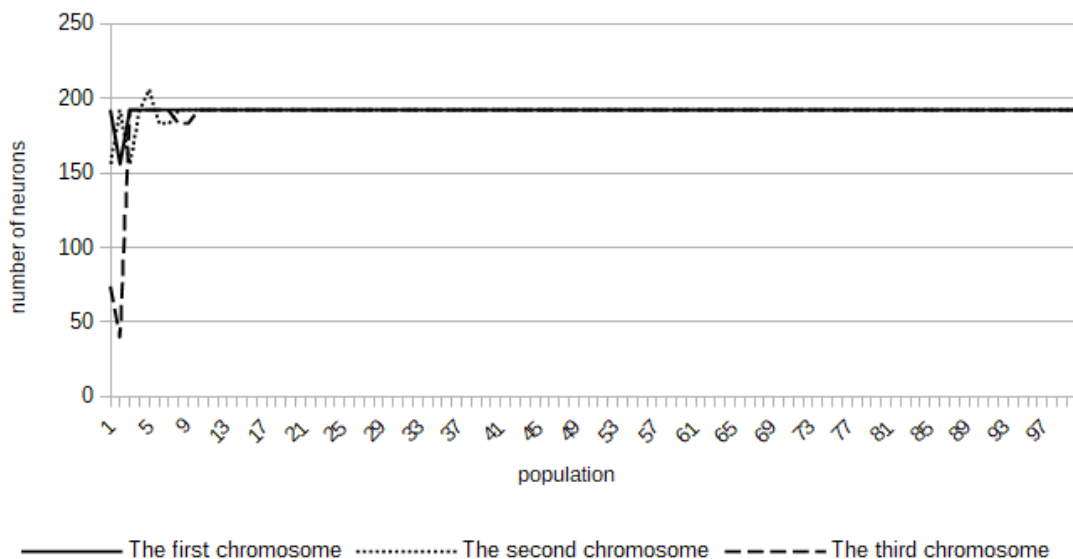


Рисунок 4.7 – Кількість нейронів у прихованих шарах після шару злиття

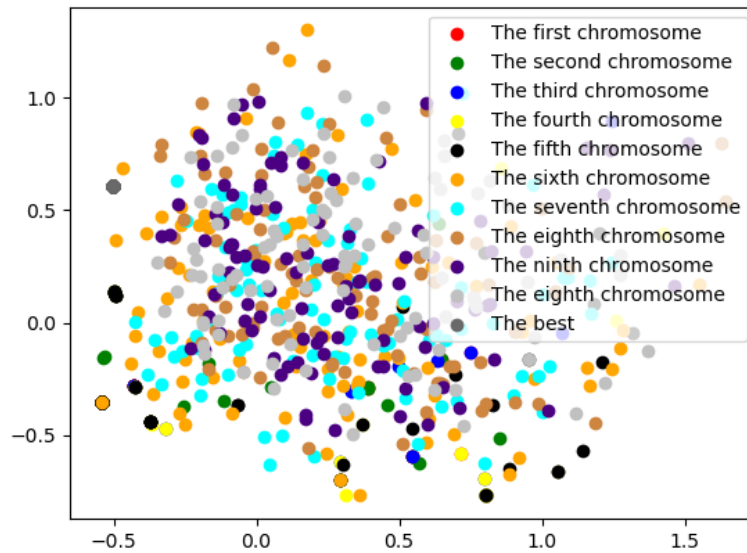


Рисунок 4.8 – Показники популяції

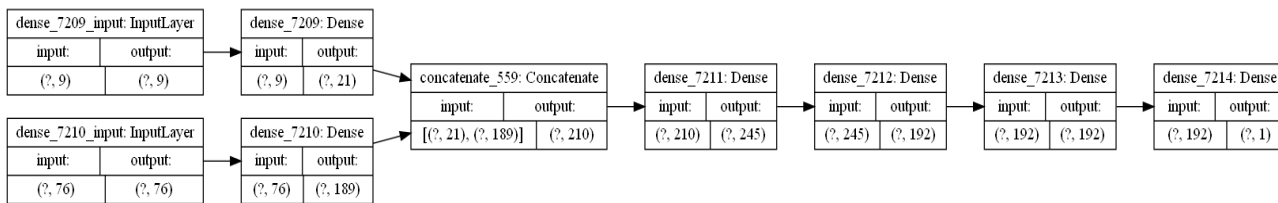


Рисунок 4.9 – Найкраща розроблена модель

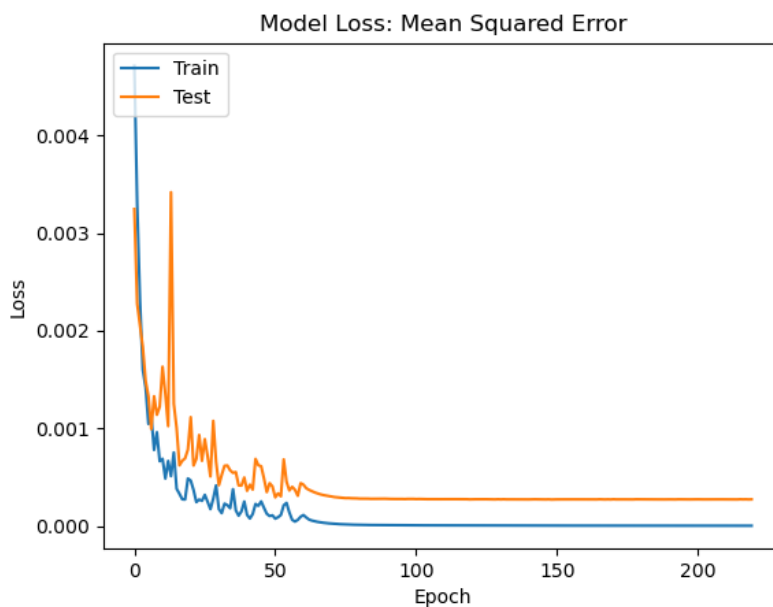


Рисунок 4.10 – Середньоквадратична похибка найкращої моделі

Рисунок 4.10 показує середньоквадратичну похибку найкращої моделі. Рисунок 4.11 – середня абсолютна відсоткова помилка.

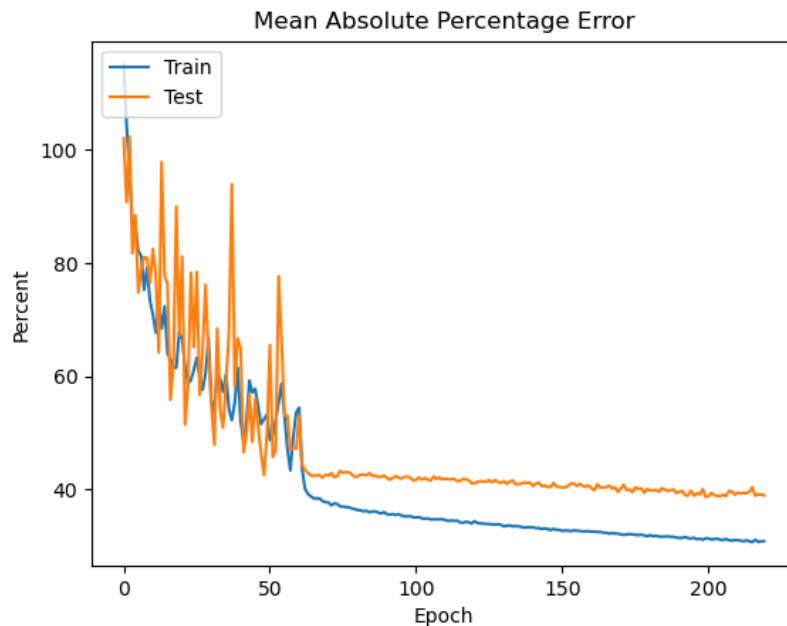


Рисунок 4.11 – Середня абсолютна відсоткова помилка

### 4.3 Генетична оптимізація архітектури нейронної мережі для моделювання напружено-деформованого циліндричної оболонки

Дана циліндрична оболонка з параметрами  $l$ ,  $r$ ,  $h$ ,  $E$ ,  $\vartheta$ ,  $q$ . Множина даних згенерована за допомогою методу скінченних елементів. Параметри циліндричної оболонки генеруються випадковим способом з певними обмеженнями:

$$l \in [0.1; 5] \text{ (м)};$$

$$r \in [0.1; 5] \text{ (м)};$$

$h \in [1/100 a; 1/20 a] \text{ (м)}$ , де  $a = \min(l, 2\pi r)$  "Передбачається, що товщина оболонки розраховується як  $\frac{1}{20} - \frac{1}{100}$  від її ширини";

$$E \in [50000; 300000] \text{ (MPa)};$$

$$\vartheta \in [0; 0,45];$$

$$q \in [0.01; 0.1] \text{ (MPa)}.$$

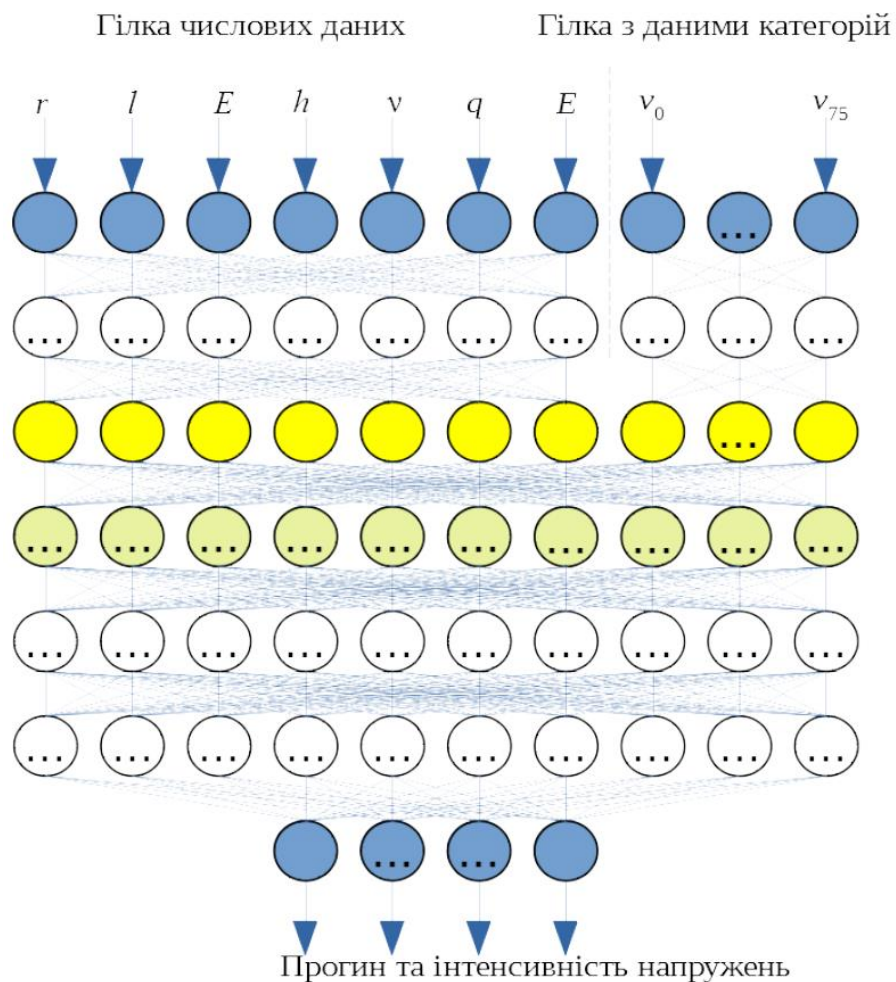


Рисунок 4.12 – Модень штучної нейронної мережі

Граничні умови також як і у випадку з прямокутною пластиною з отвором генеруються випадково. Умови закріплення – дані категорій. Можливими умовами закріплення є: вільні краї, підперті краї, затиснуті краї. Будь-яка комбінація умов закріплення країв циліндричної оболонки можлива, виключаючи ситуації з двома вільними краями.

Генерація даних відбувається алгоритмами описаними вище у другому та третьому розділах.

Для прогнозування поведінки оболонки розроблено модель, яка має декілька шарів нейронів (рис. 4.12). Шари вхідних нейронів (окремо для

числових параметрів і умов закріплення). Нейрони вхідних шарів поєднуються з прихованим шаром нейронів. Далі зовнішній шар нейронів з лінійною функцією активації формує регресійну оцінку вектору, що описує напружено-деформований стан оболонки. Внутрішні шари мають певну кількість нейронів, кожен з яких зв'язаний з внутрішньою функцією активації. Останній шар (також зовнішній) має кількість нейронів, яка дорівнює кількості вихідних значень.

### 4.3.1 Обчислювальний експеримент

Як і у випадку з прямокутною пластиною, розглянутою у цьому розділі, в обчислювальному експерименті було використано популяцію з 10 хромосом з 15 генами. Алгоритм зупиняється якщо найкращі 3 хромосоми не змінювались хоча б у 50 популяціях. Згенерована популяція векторів визначає структуру нейромережі. Для кожного вектора знаходиться фітнес-функція, яка є середньоквадратичним відхиленням нейромережі від рішення. Наступним кроком є сортування отриманих значень функції у порядку убутання, таким чином, що на першому місці буде знаходитись найкращий отриманий результат, а на останньому – найгірший.

Далі відбувається операція схрещування (crossover) членів популяції, основна задача якої – створити нову популяцію з вже існуючої.

Алгоритм зупинився після 200 популяцій. Можна зробити висновок, що генетичний алгоритм значно зменшує середню квадратичну похибку в моделі.

Обчислювальний експеримент показав такі ж результати, як і у випадку з прямокутною пластиною з отвором, зображені на рис. 4.10 – 4.11, тому що використовувались такі ж алгоритми, різниця була тільки у вхідних параметрах.

#### 4.4 Висновки до розділу 4

Отже, генетичні алгоритми досить успішно поєднуються з локальними алгоритмами пошуку або локальними методами машинного навчання. Генетичний алгоритм дозволяє зменшити середню квадратичну похибку в нейронних мережах, перевіряючи її архітектуру.

Також можна зробити висновки, що гени змінюються повільно, тому існує певна передчасна конвергенція. Отже, процедуру кросоверу або мутації слід вдосконалити.

Отримана нейронна мережа дозволяє оцінити напружено-деформований стан прямокутної пластини з круглим вирізом та циліндричної оболонки.

## ВИСНОВКИ

Отже, розроблені в дисертаційній роботі методи і моделі машинного навчання дозволяють робити швидкі оцінки параметрів стану об'єкту, а саме визначення напружено-деформованого стану. Під час розв'язання задач дисертаційного дослідження були розроблені штучні нейронні мережі для визначення максимального прогину та інтенсивності напружень за Мізесом у пластинах та оболонках. Також були розроблені алгоритми формування вибірки для навчання та тестування моделей. Досліджено можливості застосування генетичного алгоритму для оптимізації нейронної мережі регресійного аналізу і прогнозування максимального прогину пластин та оболонок. Виявлено, що алгоритми дозволяють прогнозувати значення прогину з похибкою приблизно 10% відносно аналітичного рішення.

У дисертаційній роботі отримані такі результати:

- виконано аналіз сучасного стану задачі прогнозування зон руйнування в інженерних конструкціях. Встановлено, що нейромережеві моделі є одним з найбільш універсальним методом прогнозування зон руйнування.

- уперше розроблено нейромережевий метод для визначення напружено-деформованого стану пластин з довільними умовами закріплення, який дозволяє визначити максимальний прогин, а також максимальне значення інтенсивності напружень за Мізесом;

- уперше розроблено нейромережевий метод для визначення напружено-деформованого стану циліндричної оболонки та комбінації циліндричної та конічної оболонок, який дозволив визначити прогин та інтенсивність напружень за Мізесом;

- уперше розроблено метод генетичної оптимізацію архітектури нейронної мережі для моделювання напружено-деформованого стану

прямокутної пластини та циліндричної оболонки, що дало змогу підвищити точність прогнозування;

– уперше запропоновано підхід для визначення можливих зон руйнування у квадратних пластинах з отвором.

Програмну реалізацію наведеної інструментальної системи написано мовою програмування Python. Використано такі бібліотеки Python: scikit-learn, numpy, pandas і Keras.

Основною перевагою штучної нейронної мережі є швидкість прогнозування: обчислення необхідних характеристик у порівнянні з методом скінченних елементів відбувається майже миттєво (мілісекунди). Отже, «натреновані» штучні нейронні мережі можуть слугувати як інтерактивні допоможники у процесі проектування. Перспективи подальших досліджень пов'язані з розробкою штучних нейронних мереж, які дозволять прогнозувати напружено-деформований стан за кресленням або зображенням оболонкових конструкцій з використанням машинного зору і алгоритмів класифікації.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Abadi M., Agarwal A., Barham P. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16) (Savannah, GA, USA, November, 2 – 4, 2016). Savannah, GA, USA, 2016. P. 265 – 283.
2. Abambres M., Marcy M., Doz G. Potential of Neural Networks for Structural Damage Localization engrXiv. 2018. URL: <https://engrxiv.org/rghpf/> DOI: 10.31224/osf.io/rghpf.
3. Abambres M., Corrêa R., A. P. da Costa, F. Simões F. Potential of neural networks for maximum displacement predictions in railway beams on frictionally damped foundations engrXiv. 2019. URL: <https://engrxiv.org/m3b7j/> DOI: 10.31224/osf.io/m3b7j
4. Abambres M., Rajana K., Tsavdaridis K., Ribeiro T. Neural Network-based formula for the buckling load prediction of I-section cellular steel beams engrXiv. 2018. URL: <https://engrxiv.org/wg7hd/> DOI: 10.31224/osf.io/wg7hd
5. Abu-Arqub Omar, Abo-Hammour Zaer, Momani Shaher. Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems. Appl. Math. Inf.Sci. 2014. Vol. 8, N 1. P. 235 – 248.
6. Bahrapour S., Ramakrishnan N., Schott L., et al. Comparative Study of Deep Learning Software Frameworks. Available at: <https://arxiv.org/abs/1511.06435>
7. Boominathan L., Kruthiventi S.S., Babu R.V. CrowdNet: A Deep Convolutional Network for Dense Crowd Counting. Proceedings of the 2016 ACM on Multimedia Conference (Amsterdam, The Netherlands, October 15 – 19, 2016), Amsterdam, The Netherlands, 2016. P. 640 – 644. DOI: 10.1145/2964284.2967300.
8. Chollet. F., et al. Keras. 2015. Available at: <https://github.com/fchollet/keras>

9. C. Jin, S. Jang, X. Damage detection of a highway bridge under severe temperature changes using extended Kalman filter trained neural network *Journal of Civil Structural Health Monitoring*. 2016. Vol. 6, Iss. 3, P. 545 – 560.
10. Cybenko, G.V. Approximation by Superpositions of a Sigmoidal function. Y van Schuppen, Jan H. *Mathematics of Control, Signals, and Systems*. Springer International. 2006. P. 303–314.
11. Du K.-L., Swamy M. N. S. *Neural Networks and Statistical Learning*. London: Springer, 2014. DOI: 10.1007/978-1-4471-5571-3
12. Duong V., Stubberud A.R. System identification by genetic algorithm, *Proc. IEEE Aerospace Conf., Big Sky, Montana, 2002, 5, N 3, P. 2331–2337*.
13. Glorot X., Bordes A., Bengio Y. Deep Sparse Rectifier Neural Networks. *Journal of Machine Learning Research*. 2011. vol. 15. P. 315 – 323.
14. Glorot X., Bengio Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks : proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10) (Sardinia, Italy, May 13 – 15, 2010). Society for Artificial Intelligence and Statistics. Sardinia, Italy 2010. P. 249 – 256.
15. Goldberg D.E. *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, 1989, P. 432.
16. Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Joshua (2014). «Generative Adversarial Networks». 2014
17. Hahnloser R.H.R., Sarpeshkar R., Mahowald M.A., et al. Digital Selection and Analogue Amplification Coexist in a Cortex-Inspired Silicon Circuit. *Nature*. 2000. vol. 405. P. 947 – 951. DOI: 10.1038/35016072.
18. Hahnloser R.H.R., Seung H.S., Slotine J.J. Permitted and Forbidden Sets in Symmetric Threshold-Linear Networks. *Neural Computation*. 2003. vol. 15, no. 3. P. 621 – 638. DOI: 10.1162/089976603321192103.

19. Hany Sallam, Carlo S. Regazzoni, Ihab Talkhan, and Amir Atiya. Evolving neural networks ensembles nnes. IAPR Workshop on Cognitive Information Processing, P. 142-147
20. Hany Sallam, Carlo S. Regazzoni, Ihab Talkhan, and Amir Atiya. Evolving neural networks ensembles nnes. IAPR Workshop on Cognitive Information Processing, P. 142-147
21. Hatanaka T., Uosaki K., Koga M. Evolutionary Computation Approach to Wiener Model Identification, Proc. of the 2002 Cong. on Evolutionary Computation, CEC'2002, Honolulu, Hawaii: IEEE, 2002, P. 914–919.
22. He K., Zhang X., Ren S., et al. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (Las Vegas, NV, USA, 27 – 30 June 2016), Las Vegas, NV, USA, 2016. P. 770 – 778. DOI: 10.1109/CVPR.2016.90.
23. He K., Sun J. Convolutional Neural Networks at Constrained Time Cost. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Boston, MA, USA, June 07 – 12, 2015), Boston, MA, USA, 2015. P. 5353–5360. DOI: 10.1109/CVPR.2015.7299173.
24. Holland J.H. Adaptation in natural and artificial systems. Ann Arbor : Univ. of Michigan Press, 1975. P. 219.
25. Hui Wang, Qing-Hua Qin. Methods of Fundamental Solutions in Solid Mechanics, 2019
26. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. JMLR Workshop and Conference Proceedings : proceedings of the 32nd International Conference on Machine Learning (Lille, France, July 06 – 11, 2015), Lille, France, 2015. vol. 37. P. 448 – 456.
27. Jia Y., Shelhamer E., Donahue J., et al. Caffe: Convolutional Architecture for Fast Feature Embedding. Proceedings of the 22nd ACM International Conference on Multimedia (Orlando, FL, USA, November 03 – 07, 2014), Orlando, FL, USA 2014. P. 675–678. DOI: 10.1145/2647868.2654889.

28. Keras. URL: <https://www.tensorflow.org/guide/keras> (дата звернення 23.05.2021)
29. Kingma D. P. Ba J. L. Adam: A Method for Stochastic Optimization. URL: <https://arxiv.org/abs/1412.6980> (дата звернення 14.03.2021)
30. Kinnikar A., Husain M., Meena S.M. Face Recognition Using Gabor Filter And Convolutional Neural Network : proceedings of the International Conference on Informatics and Analytics (Pondicherry, India, August 25 – 26, 2016), Pondicherry, India, 2016. P. 113:1 – 113:4. DOI: 10.1145/2980258.2982104.
31. Kristinsson K., Dumont G. System identification and control using genetic algorithms, IEEE Transactions on Systems, Man and Cybernetics, 1992, 22, N 5, P. 1033–1046.
32. Kruchinin D., Dolotov E., Korniyakov K. et al. Comparison of Deep Learning Libraries on the Problem of Handwritten Digit Classification. Analysis of Images, Social Networks and Texts. Communications in Computer and Information Science. 2015. vol. 542. P. 399 – 411. DOI: 10.1007/978-3-319-26123-2\_38.
33. LeCun Y., Bottou L., Orr G.B. Efficient BackProp. : Tricks of the Trade. 1998. P. 9 – 50. DOI: 10.1007/3-540-49430-8\_2.
34. Lisnyak A. O., Choporova O. V. Using machine learning to predict the stress-strain state of a rectangular plate with a circular cut-out. CEUR Workshop Proceedings. 2020. T. 2791. C. 1-6. URL: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85098720535&origin=resultslist&sort=plf-f&src=s&st1=&st2=&sid=80dfe74813393a8626e1696d73fee20a&sot=b&sdt=b&sl=112&s=TITLE+%28Using+machine+learning+to+predict+the+stress-strain+state+of+a+rectangular+pl.> (SCOPUS).
35. Ljung L. System Identification – Theory for the User, 2nd Edition, Prentice-Hall, Upper Saddle River, N J, 1999, 607 p
36. Nanda J., R. Narayanan R. Application of genetic algorithm to economic load dispatch with lineflow constraints. International journal of electrical power & energy systems. 2002. Vol. 24, N 9. C. 723–729.

37. Onur Avci P. O., Abdeljaber A. O. Self-Organizing Maps for Structural Damage Detection: A Novel Unsupervised Vibration-Based Algorithm. *Journal of Performance of Constructed Facilities*. 2016. Vol. 30, Iss. 3. P. 1–11.
38. Peter Z. Szewczyk Neural Network Based Extrapolation Strategies in Structural Analysis and Design, 1999. p.238 – 255.
39. Ranzato M.A., Huang F.J., Boureau Y.L., et al. Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (Minneapolis, MN, USA, 17 – 22 June 2007)*, Minneapolis, MN, USA, 2007. P. 1 – 8. DOI: 10.1109/cvpr.2007.383157.
40. Razavian A.Sh., Azizpour H., Sullivan J., et al. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (Washington, DC, USA, June 23–28, 2014)*, Washington, DC, USA, 2014. P. 512–519. DOI: 10.1109/cvprw.2014.131.
41. Reddy J. N. *Mechanics of Laminated Composite Plates and Shells. Theory and Analysis* / J. N. Reddy. – Boca Raton, London, New York, Washington, D.C.: CRC Press, 2003. – 854 p.
42. Reddy J. N. *Mechanics of Laminated Composite Plates and Shells. Theory and Analysis*. Boca Raton, London, New York, Washington, D.C. : CRC Press, 2003. 854 p.
43. Real E., Liang C., So D. R., Le Q. V. AutoML-Zero: Evolving Machine Learning Algorithms From Scratch. Preprint arXiv.org, 2020. 23 p. URL: <https://arxiv.org/abs/2003.03384> (дата звернення: 01.08.2020).
44. Reissner E. The effect of transverse shear deformation on the bending of elastic plates. *ASME Journal of Applied Mechanics*. 1945. Vol. 12. P. 68–77.
45. Reeves C.R., Rowe J.E. Genetic algorithms: principles and perspectives. *A Guide to GA Theory*, Kluwer Acad. Publ., 2003, P. 332. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*. Vol. 65, 1958. P. 386 – 408.

46. Ruo Chen W., Zhe X. A Pedestrian and Vehicle Rapid Identification Model Based on Convolutional Neural Network. Proceedings of the 7th International Conference on Internet Multimedia Computing and Service (ICIMCS '15) (Zhangjiajie, China, August 19–21, 2015), Zhangjiajie, China, 2015. P. 32. DOI: 10.1145/2808492.2808524.
47. Sebastian A., Kumar P., Schoen M.P. Spatial filter masks optimization using genetic algorithm and modeling dynamic behavior of sEMG and finger force signals, *Int. J. of Circuits, Systems, and Signal Processing*, 2011, 5, N 6, P. 597–608.
48. Symone G. Soares, Carlos H. Antunes, Rui Arajo. A Genetic Algorithm for Designing Neural Network Ensembles. Proceedings of the 14th annual conference on Genetic and evolutionary computation. P. 681-688.
49. Snyman, Jan. *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer Science & Business Media. ISBN 978-0-387-24348-1.
50. Seppo Linnainmaa. The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Helsinki, 1970 P. 6-7.
51. Szegedy C., Liu W, Jia Y. et al. Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA, USA, June 7 – 12, 2015), Boston, MA, USA, 2015. P. 1–9. DOI: 10.1109/CVPR.2015.7298594.
52. Szegedy C., Vanhoucke V., Ioffe S., et al. Rethinking the Inception Architecture for Computer Vision : proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition* (Seattle, WA, USA, Jun 27 – 30, 2016), Seattle, WA, USA, 2016. P. 2818 – 2826. DOI: 10.1109/cvpr.2016.308.
53. Szegedy C., Ioffe S., Vanhoucke V., et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning : proceedings of the *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)* (San Francisco, California, USA, February 4 – 9, 2017), San Francisco, California, USA, 2017. P. 4278 – 4284.

54. Viren J., Sebastian S. Natural Image Denoising with Convolutional Networks. Advances in Neural Information : processing Systems (NIPS) 21. 2009. P. 769–776.
55. Widrow B., Hoff M. E. Adaptive switching circuits. 1960 IRE WESCON Convention Record. Part 4. New York: IRE, 1960. P. 96 – 104.
56. Wirsansky E. Hands-On Genetic Algorithms with Python. Birmingham : Packt Publishing Ltd, 2020. P. 334.
57. Wright A. Genetic algorithms for real parameter optimization. Foundations of Genetic Algorithms. 1991. Vol. 1. P. 205 – 218.
58. Zienkiewicz O. C., Taylor R. L.. The finite element method, Fifth Edition, Volume 2: Solid Mechanics. London: Butterworth-Heinemann, 2000. 460 P.
59. Zaccone G., Karim Md. R., Menshawy A. Deep learning with TensorFlow. Birmingham : Packt Publishing Ltd, 2018. 767 p.
60. Абовский Н.П. Нейросетевые модели в задачах строительной механики. Известия вузов. Строительство. 2000. № 7. С.6 – 14.
61. Абовский Н.П., Белобородова Т.В., Максимова О.М. Нейросетевое моделирование в задачах теории пластин и оболочек. Известия вузов. Строительство. 2001. № 9. С.9 – 15.
62. Abovskiy N.P., Maximova O.M. Neuro-Prognosis Based on Step Model with Teaching for Natural Tests Results of Building Structures. j. Optical Memory & Neural Networks (Information Optics), 2007, P. 40 – 46.
63. Абовский Н.П. Нейроуправляемые конструкции и системы: учеб. пособие для вузов; Кн.13 научной серии «Нейрокомпьютеры и их применение» под ред. А.И. Галушкина. Москва : Радиотехника, 2003. С. 368.
64. Абовский Н.П. Нейросетевые технологии в задачах оптимизации, прогнозирования и управления: научное издание. Красноярск : КрасГАСА. 2003. С. 176.
65. Абовский Н.П., Максимова О.М., Белобородова Т.В. Нейросетевые подходы к расчетным и проектировочным задачам

строительной механики. Нейрокомпьютеры, разработка и применение. 2001. № 9. С.14 – 30.

66. Абовский Н.П., Максимова О.М., Светашков П.А. Эволюционная модель нейросетевого прогнозирования. V Всеросс. конф. «Нейроинформатика – 2002»: сб. науч. тр. Москва : МИФИ, 2003. С.207 – 214.

67. Абовский Н.П., Максимова О.М. Эффективный нейросетевой метод пошагового прогнозирования и его применение к многомерным задачам. Нейрокомпьютеры, разработка и применение. 2003. С.112 – 126.

68. Абовский Н.П., Максимова О.М., Светашков П.А. Развитие и применение нейроинформатики к задачам прогнозирования и оптимизации. Вестник Красноярского Регионально-го Отделения САН ВШ. Красноярск, 2005. С.5–23.

69. Белозерский Л. А. Введение в системы автоматического распознавания. Киев : Наукова думка, 2005. С. 434.

70. Вакал Л.П. Генетичні алгоритми для чебишовської апроксимації. Комп'ютерні засоби, мережі та системи. 2013. № 12. С. 20–26.

71. Вакал Л. П. Генетичні алгоритми як інструмент розв'язання нелінійних крайових задач. Комп'ютерні засоби, мережі та системи. 2015, № 14

72. Гаращенко Ф.Г., Мороз О.Г. Ідентифікація нелінійної системи Гаммерштейна за допомогою генетичного алгоритму, Вісник КНУ ім.Т. Шевченка. Серія: фізико-математичні науки, 2012, № 1, С. 145–150.

73. Глибовець М.М., Гулаєва Н.М. Еволюційні алгоритми, Київ : НаУКМА, 2013, С. 828.

74. Гудфелов І., Бенгіо Й., Коурвілле А. «Машинне навчання» MIT Press, 2016. URL: <http://www.deeplearningbook.org>. (дата звернення 10.03.2021).

75. Гурейчик В.М. Генетические алгоритмы и их применение, Таганрог : Таганрогского РТО, 2002, С. 244.

76. Демиденко М.Г., Проценко С.І, Проценко О.Б., Шифолович П., Федченко О.В. Використання генетичного алгоритму для розв'язання

оберненої задачі еліпсометрії. Вісник СумДУ. Серія “ Фізика, математика, механіка”, № 1’ 2008.

77. Зеленцов Д. Г. Нейронные сети как средство повышения точности и эффективности решения задач оптимизации. Комп’ютерне моделювання: аналіз, управління оптимізація. 2018. № 2. С. 18 – 26.

78. Козин И. В. Эволюционные модели в дискретной оптимизации. Запорожье : ЗНУ, 2019. С. 204.

79. Клепиков В.Б., Сергеев С.А., Махотило К.В. и др. Применение методов нейронных сетей и генетических алгоритмов в решении задач управления электроприводами. Электротехника. 1999. № 5. С. 2–6.

80. Круглов В.В., Борисов В. В. Искусственные нейронные сети. Теория и практика. – Москва : Горячая линия - Телеком, 2001. С. 382.

81. Лесовик Р. В. Оптимальное проектирование строительных конструкций на основе генетического алгоритма. Строительная механика инженерных конструкций и сооружений. 2010. С. 20–24.

82. Лисенко В. П., Решетюк В. М., Штепа В. М. Системи штучного інтелекту: нечітка логіка, нейронні мережі, нечіткі нейронні мережі, генетичний алгоритм : монографія. Київ : НУБіП України, 2014. 332 с.

83. Литвинов, М. И. Применение нейронной сети с функцией активации вида  $\sin(x)/x$  и  $|\sin(x)/x|$  для экстраполяции стохастической функции (на примере продаж компании). Новые информационные технологии в автоматизированных системах. 2009. URL: <https://cyberleninka.ru/article/n/primenenie-neyronnoy-seti-s-funktsiey-aktivatsii-vida-sin-x-x-i-sin-x-x-dlya-ekstrapolyatsii-stohasticheskoy-funktsii-na-primere-prodazh> (дата звернення 14.03.2021)

84. Ліла В.Б. Алгоритм та програмна реалізація адаптивного метода навчання штучних нейронних мереж. Інженерний вестн. Дона, 2012.

85. Максимова О. М. Развитие и применение нейросетевых технологий для задач механики и строительных конструкций. Вестник ИрГТУ. Красноярск, 2013. №8 (79). С. 81.

86. Мазуров В. Д. Математические методы распознавания образов : уч. пособие, 2-е изд. Екатеринбург : Урал. ун-т, 2010. 101 с. Максимова О.М. Использование нейросетевых технологий для повышения точности в задачах экстраполяции и интерполяции. VI Всеросс. науч.-техн. конф. «Нейроинформатика – 2004»: сб. науч. тр. в 2 ч. Москва : МИФИ, 2004. Ч.1. С.190 – 200.
87. Максимова О.М. Нейропрогнозирование как эволюционный интеллектуальный процесс. Международная конференция «Искусственный интеллект – 2007». Научно-теоретический журнал. Донецк : МОНІНАН, 2007. №4. С.623 – 634.
88. Максимова О.М. Нейропрогнозирование с использованием пакетов программ моделирования нейронных сетей при исследовании строительных конструкций. Нейрокомпьютеры, разработка и применение. 2007. №9. С.426–439.
89. Максимова О.М. Разработка и применение нейросетевой технологии прогнозирования к задачам строительной механики и конструкций. Международный конгресс «Наука и инновации в строительстве» SIB – 2008: сб. науч. тр. Воронеж, 2008. С.146 – 151.
90. Максимова О. М. Развитие и применение нейросетевых технологий для задач механики и строительных конструкций. Вестник ИрГТУ. Красноярск, 2013. № 8 (79). С. 81–88.
91. Максимова О.М., Дюмин М.В. Обоснование развития и применения нейросетевой технологии в решении задач геодинамики. Научная сессия МИФИ-2009. XI Всеросс. науч.-техн. конф. «Нейроинформатика – 2009»: сб. науч. тр. в 3 ч. Москва: МИФИ, 2009. Ч.1. С.9.
92. Максимова О.М. Создание и применение нейросетевой технологии для прогнозирования в строительных конструкциях и строительной механике. Фундаментальные и прикладные проблемы науки: сб. науч. тр. I Международного симпозиума. Москва : РАН, 2010. Т. 2. С.3 – 24.
93. Максимова О.М. Нейросетевые технологии прогнозирования для задач динамики строительных конструкций. Научная сессия НИЯУ МИФИ-

2012. XIV Всеросс. науч.-техн. конф. «Нейроинформатика – 2012»: сб. науч. тр. в 3 ч. Москва : МИФИ, 2012. Ч.2. С.71–82.

94. Мороз, О.Г. Аналіз застосування генетичних алгоритмів в задачах глобальної оптимізації. 2018. URI: <http://dspace.nbuiv.gov.ua/handle/123456789/144134> (дата звернення 10.03.2021)

95. Неклюдов І., Клепиков В., Корда В., Шепелєв А., Немашкало О., Юрченко Л., Тутубалін А., Корда Л., Шляхов М. Еволюційні алгоритми у природних науках. Вісник НАН України, 2005.

96. Олійник А. О., Субботін С. О., Олійник О. О. Еволюційні обчислення та програмування. Запоріжжя : ЗНТУ, 2010. 324 с.

97. Панченко Т.В. Генетические алгоритмы. Астрахань: Издательский дом «Астраханский университет», 2007. С. 87.

98. Подлазова А.В. Генетические алгоритмы на примерах задач раскроя. Проблемы управления. 2008. № 2. С. 57–63.

99. Погорілий С.Д., Білоус Р.Б. Генетичний алгоритм розв'язання задачі маршрутизації в мережах. Проблеми програмування. 2010. № 2–3 Спец. вип. С. 171–178.

100. Пейгин С.В., Перио Ж., Тимченко С.В. Применение генетических алгоритмов для оптимизации формы тела по тепловому потоку. Мат. моделирование. 1998. Т. 10, № 9. С. 111–122.

101. Рвачев В.Л. Шейко Т.И. Введение в теорию R-функций. Проблемы машиностроения. – 2001. – Т. 4, No 1–2. – С. 46-58.

102. Рутковская Р. Д. Нейронные сети, генетические алгоритмы и нечеткие системы. Москва : Горячая линия-Телеком, 2007. С. 383.

103. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы. Москва : Горячая линия-Телеком, 2007. 383 с.

104. Сергеева О.П. Применение генетических алгоритмов для распознавания изображений. Искусственный интеллект. 2002. № 4. С. 516–520.

105. Сигмоида. URL: <http://www.wikiwand.com/ru/%D0%A1%D0%B8%D0%B3%D0%BC%D0%BE%D0%B8%D0%B4%D0%B0>  
(дата звернення 14.03.2021)
106. Скобцов Ю. А. Метаэвристики : монография. Донецк : «Ноулидж» (Донецкое отделение), 2013. С. 426
107. Скрильник І.І. Пофарбування графів за допомогою генетичного алгоритму. Вісник ТДТУ. 2010. Том 15. № 1. С.194-203.
108. Смолянинова Л.Г. Нейроуправляемые конструкции. Учет свойств реального объекта. Нейрокомпьютеры. Разработка, применение. 2001. № 9. С.39 – 49.
109. Тимошенко С.П., Войновский-Кригер С. Пластинки и оболочки. 1966 URL: [pnu.edu.ru/media/filer\\_public/2013/04/10/5-5\\_timoshenko\\_1966.pdf](http://pnu.edu.ru/media/filer_public/2013/04/10/5-5_timoshenko_1966.pdf)
110. Функции активации в нейронных сетях. URL: <http://www.aiportal.ru/articles/neural-networks/activation-function.html> (дата звернення 12.03.2021)
111. Чопорова О. В., Чопоров С. В., Лісняк А. О. Implementation of genetic algorithm for neural network optimization for making predictions of the stress-strain state of a rectangular plate with a circular cut out. В кн.: Mathematical and Computer Modelling of Engineering Systems Riga: Baltija Publishing. 2020 С. 1-18. URL: <http://www.baltijapublishing.lv/omp/index.php/bp/catalog/view/89/2215/4790-1>.
112. Чопорова О. В., Кривохата А. Г. Оптимізація згорткових нейронних мереж та їх ансамблів. Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки.. 2019. № 1. С. 107-115. (Index Copernicus).
113. Чопорова О. В., Чопоров С. В., Лісняк А. О. Використання машинного навчання для прогнозування напружено-деформованого стану квадратної пластинки. Вісник Херсонського національного технічного університету. 2019. № 2(69). С. 192-201. URL: <https://mkmm.org.ua/upload/%D0%92%D1%96%D1%81%D0%BD%D0%B8%D>

0%BA%20%D0%A5%D0%9D%D0%A2%D0%A3%20%D1%87%D0%B0%D1%81%D1%82%D0%B8%D0%BD%D0%B0%203.pdf.

114. Чопорова О. В., Чопоров С. В., Лісняк А. О. Використання генетичного алгоритму для оптимізації параметрів нейронної мережі при прогнозуванні напружено-деформованого стану квадратної пластинки. Вісник Херсонського національного технічного університету. 2020. Т.3 № 2.1. С. 290-300. (Index Copernicus)

115. Чопорова О. В., Кривохата А. Г., Лісняк А. О. Прогнозирование поведения квадратной пластины с помощью машинного обучения. Тезисы докладов. Научно-практическая конференция "Современные расчётно-экспериментальные методы определения характеристик ракетно-космической техники" Дніпро: Дніпро, 2019. С. 56-57.

116. Чопорова О. В., Чопоров С. В., Лісняк А. О. Модель прогнозування напружено-деформованого стану прямокутної пластини з круглим вирізом. «Інформаційні системи та технології» ІСТ-2019 Коблеве-Харків: ХНУРЕ, 2019. № 8 С. 97-99. URL: [http://istconf.nure.ua/archive/ist\\_2019.pdf](http://istconf.nure.ua/archive/ist_2019.pdf).

117. Чопорова О. В. Використання машинного навчання для прогнозування поведінки інженерних конструкцій. Сучасні проблеми машинобудування Харків: Інститут проблем машинобуд ім. А.П.Підгорного, 2019. С. 19.

118. Чопорова О. В., Лісняк А. О. Аналіз можливостей мови програмування Python у машинному навчанні. Збірка тез доповідей Десятої Всеукраїнської, сімнадцятої регіональної наукової конференції молодих дослідників "Актуальні проблеми математики та інформатики" Запоріжжя: Видавничий дім "Гельветика", 2019. С. 57-59.

119. Чопорова О. В., Лісняк А. О. Використання машинного навчання для прогнозування поведінки квадратої пластини. VI міжнародна науково-практична конференція Інформаційні технології та взаємодії Київ: КНУ ім.Тараса Шеченка, 2019. С. 263-266.

120. Чопоров С.В., Лисняк А.А., Гоменюк С.И. Дискретизация геометрических областей, заданных R-функциями, на четырехугольные конечные элементы. Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2009. № 1. С. 199-207.

121. Ясницкий Л.Н., Гладкий С.Л., Никитенко И.И. и др. Искусственный интеллект в решении краевых задач проектной инженерии. URL: <http://www.permai.ru/files/projects/P03.pdf>.

122. Яцук Ю.В. Построение расчетных сеток для решения уравнений математической физики методом «объемов Безье» Вестник Астраханского государственного технического университета. Серия: Морская техника и технология. 2009. № 1. С. 116-120.

## ДОДАТОК А

**Акт впровадження у навчальний процес Запорізького національного  
університету**



**ДОВІДКА**

**про використання результатів дисертаційної роботи  
Чопорової Оксани Володимирівни, «Прогнозування зон руйнування  
в інженерних конструкціях з використанням комп'ютерного навчання»,  
виконаної у Запорізькому національному університеті**

Комісія у складі:

Голова комісії: доцент кафедри програмної інженерії, кандидат фізико-математичних наук, доцент Горбенко В. І.

Члени комісії: доцент кафедри програмної інженерії, к.ф.-м.н. Кудін О. В.;  
доцент кафедри програмної інженерії, к.т.н., доцент Мухін В. В.

Засвідчує, що результати дисертаційної роботи: нейромережевий метод для визначення напружено-деформованого стану пластин з довільними умовами закріплення, який дозволяє визначити максимальний прогин, нейромережевий метод для визначення напружено-деформованого стану циліндричної оболонки та комбінації циліндричної та конічної оболонок, метод генетичної оптимізації архітектури нейронної мережі для моделювання напружено-деформованого стану прямокутної пластини та циліндричної оболонки, використані у навчальній діяльності кафедри програмної інженерії Запорізького національного університету при викладанні дисциплін «Емпіричні методи програмної інженерії», «Засоби машинного навчання», «Нейронні мережі» для здобувачів ступеня вищої освіти бакалавр спеціальності 121 інженерія програмного забезпечення, а також у підготовці бакалаврських і магістерських кваліфікаційних робіт.

Голова комісії: Горбенко Віталій Іванович

Члени комісії: Кудін Олексій Володимирович

Мухін Віталій Вікторович