

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА ПРИКЛАДНОЇ ФІЗИКИ

«На правах рукопису»

УДК 535.373.1,546.47

«До захисту допущено»

В. о. завідувача кафедри

Г. Є. Монастирський

(підпис)

(ініціали, прізвище)

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-науковою програмою «Прикладна фізика»

зі спеціальності 105 Прикладна фізика та наноматеріали

(код і назва)

на тему: **Обернена задача розсіювання електромагнітного поля на металевих наночастинках**

Виконав: студент 2 курсу, групи ФФ-11мн

Поляцко Антоній Костянтинович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник доцент, к.ф.-м.н. Гільчук А. В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант 2, 3 доцент, д.ф.-м.н. Сахненко Н. К.

(номер розділу)

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент доцент, к.ф.-м.н. Бойко І. В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент

(підпис)

Київ – 2023

Національний Технічний Університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
 Фізико-технічний інститут
 Кафедра прикладної фізики

Рівень вищої освіти — другий (магістр)

Освітньо наукова програма «Прикладна фізика»

Спеціальність 105 Прикладна фізика та наноматеріали

«ЗАТВЕРДЖЕНО»

Завідувач кафедри

Монастирський. Г. Є.

(підпис)

(ініціали, прізвище)

« ____ » _____ 2021 р.

ЗАВДАННЯ

на магістерську дисертацію
здобувачу ступеня магістра

Поляцку Антонію Костянтиновичу

(прізвище, ім'я, по батькові)

(підпис)

1. Тема роботи: Обернена задача розсіювання електромагнітного поля на металевих наночастинках,
науковий керівник роботи _____ доцент, к.ф.-м.н. Гільчук А. В. _____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
- затверджені наказом по університету від « 11 » квітня 2023 р № 1482-с
2. Термін подання студентом роботи « 16 » травня 2023 р.
3. Об'єкт дослідження: локалізований поверхневий плазмонний резонанс в металевих наночастинках
4. Предмет дослідження: металеві наночастинки
5. Перелік завдань, які потрібно розробити: Змоделювати пререріз розсіювання на багаточарових наночастинках, згенерувати дані для навчання нейромереж, натренерувати нейромережі для вирішення оберненої задачі
6. Орієнтовний перелік ілюстративного матеріалу: презентація
7. Орієнтовний перелік публікацій: Розв'язання оберненої задачі дифракції на багаточарових металевих наночастинках з використанням перерізу розсіювання за допомогою нейромереж

8 Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
2	доцент, д.ф.-м.н. Сахненко Н. К.		
3	доцент, д.ф.-м.н. Сахненко Н. К.		

9 Дата видачі завдання « 1 » вересня 2021 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів проекту	Примітка
1.	Опрацювання літератури за темою	1.09.2021 - 30.12.2021	
2.	Написання літературного огляду	01.01.2022 - 01.03.2022	
3.	Вирішення рівняння Гельмгольца та моделювання результатів	02.03.2022 - 01.12.2022	
4.	Написання другої частини дипломної роботи	01.12.2022 - 31.03.2023	
5.	Підготовка тез	15.04.2023 - 05.05.2021	
5.	Виступ на конференції	12.05.2021	
6.	Написання третьої частини дипломної роботи	1.04.2023 - 14.05.2023	
7.	Написання висновків	14.05.2023 - 15.05.2023	
8.	Подання роботи на рецензування	16.05.2023	
9.	Підготовка доповіді та презентації	18.05.2023 - 22.05.2023	
10.	Подання роботи до захисту	23.05.2023	

Студент

_____ А. К. Поляцко
(підпис) (ініціали, прізвище)

Науковий керівник роботи

_____ А. В. Гільчук
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Пояснювальна записка дипломної роботи за обсягом становить 53 сторінки, містить 6 таблиці та 25 рисунки. Для дослідження було використано 22 бібліографічних найменувань.

Застосування нейромереж для розв'язання обернених задач розсіювання електромагнітного поля на металевих наночастинках виявляється перспективним напрямком досліджень з кількох причин. Такий підхід може дозволити ефективно моделювати і передбачати поведінку електромагнітного поля, що розсіюється наночастинками з високою точністю та швидкістю. Це важливо для розуміння та оптимізації властивостей наноматеріалів у різних областях застосування, таких як електроніка, оптика, сенсорика та енергетика. Також застосування нейромереж для обернених задач розсіювання може вирішити деякі ключові проблеми, пов'язані зі складністю обчислень та великим обсягом даних.

В результаті даної роботи було показано що нейромережі є досить перспективною технологією для моделювання розсіювання електромагнітного поля для наночастинок та показано приклади нейромереж які можуть знаходити структуру металевих багаточарових наночастинок. Найкращою архітектурою для вирішення даних задач виявилась нейромережа з LSTM шаром, яка показала в середньому точність $\approx 93\%$.

Ключові слова: нейронні мережі, зворотня задача розсіювання, поверхневий плазмонний резонанс, локалізований, наночастинки, переріз розсіювання.

SUMMARY

The diploma work explanatory message includes 53 pages of the text, 6 table and 25 illustrations. At the problem modern state analysis, overall 22 references were used.

The application of neural networks for solving inverse scattering problems of electromagnetic fields on metal nanoparticles is a promising research direction for several reasons. This approach can effectively model and predict the behavior of the scattered electromagnetic field with high accuracy and speed. This is crucial for understanding and optimizing the properties of nanomaterials in various fields of application, such as electronics, optics, sensing, and energy. Furthermore, the use of neural networks for inverse scattering problems can address key challenges related to computational complexity and large data volumes. By employing neural networks, particularly deep neural networks, it becomes possible to achieve faster and more efficient solutions for these problems, thereby reducing the time and costs involved in researching and developing new materials and devices.

In the course of this study, it was demonstrated that neural networks are a promising technology for modeling the scattering of electromagnetic fields by nanoparticles, and examples of neural networks capable of identifying the structures of multilayered metal nanoparticles were presented. The best architecture for addressing these tasks was found to be a neural network with an LSTM layer, which achieved an average accuracy of approximately 93

Key words: neural networks, inverse scattering problem, surface plasmon resonance, localized, nanoparticles, scattering cross section.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.	5
ВСТУП	6
РОЗДІЛ 1. Огляд літератури	8
1.1. Нейронні мережі	8
1.1.1. Історія розвитку нейронних мереж	8
1.1.2. Архітектури нейромереж	11
1.2. Огляд перерізу розсіювання для металевих наночастинок	24
1.2.1. Теоретичні підходи до визначення перерізу розсіювання	24
1.2.2. Вплив параметрів наночастинок на переріз розсіювання	24
1.2.3. Експериментальні методи вимірювання перерізу розсіювання	25
1.2.4. Діелектрична проникність металів	25
1.3. Проблематика та постановка задачі.	27
1.4. Висновки до першого розділу	32
РОЗДІЛ 2. Матеріали та методи	33
2.1. Розсіювання плоскої хвилі на сфері.	33
2.1.1. Кутова частина рівняння	34
2.1.2. Радіальна частина рівняння	35
2.1.3. Перехід до сферичних функцій	36
2.1.4. Граничні умови	37
2.2. Розсіювання на багат шаровій сфері	37
2.3. Переріз розсіювання для наночастинок	39
2.4. Нейромережа для знаходження матеріалів наночастинок	39
2.5. Нейромережа для знаходження радіусів наночастинок	42
2.6. Нейромережі для знаходження діелектричної проникності матеріалів.	43
2.7. Висновки до другого розділу	44
РОЗДІЛ 3. Результати досліджень	45
3.1. Отримані перерізи розсіювання та генерація даних	45

	4
3.2. Порівняння нейронних мереж для класифікації даних	46
3.3. Порівняння нейронних мереж для знаходження радіусів шарів наночастинок	47
3.4. Результати тренування нейромереж для знаходження діелектри- чної проникності	48
3.5. Висновки до третього розділу	49
ВИСНОВКИ.	51
ПЕРЕЛІК ПОСИЛАНЬ	52

ШНМ — штучна нейронна мережа

LTSM — довга короткочасна пам'ять

RRN, РНМ — рекурентна нейронна мережа

MAE — середня абсолютна похибка

MSE — середня квадратична помилка

MAPE — Середня абсолютна похибка у відсотках

Розсіювання електромагнітного поля на металевих наночастинках є актуальним об'єктом дослідження в сучасній науці та технології. Властивості цих наноматеріалів визначають їх потенційні застосування в різних галузях, включаючи електроніку, оптику, сенсорику та енергетику. Розуміння та моделювання процесів розсіювання електромагнітного поля на металевих наночастинках є важливим кроком для розробки нових матеріалів та пристроїв з покращеними функціональними характеристиками.

Однак, вирішення обернених задач розсіювання, тобто встановлення параметрів наночастинок на основі вимірів розсіяного поля, є викликом через їх складність та обчислювальну складність. Традиційні методи моделювання, такі як метод скінченних елементів та методи дискретних диференцій, можуть бути обчислювально витратними та часово затратними.

У цій роботі розглядається перспектива використання нейромереж для розв'язання обернених задач розсіювання електромагнітного поля на металевих наночастинках. Нейромережі, зокрема глибокі нейронні мережі, відомі своєю здатністю вирішувати складні завдання з обробки інформації та передбачення на основі великого обсягу даних. Використання цих мереж може відкрити нові можливості для точного моделювання та передбачення розсіювання електромагнітного поля на металевих наночастинках.

Основною метою даної роботи є дослідження ефективності та потенціалу нейромереж у вирішенні обернених задач розсіювання на металевих наночастинках. Зокрема, досліджується можливість використання нейромереж для встановлення структури та властивостей металевих багат шарових наночастинок на основі змодельованих даних про розсіяне електромагнітне поле.

Були поставлені такі завдання:

1. Вирішити рівняння Гельмгольца для задачі дифракції на багат шаровій наночастинці.
2. Змодельувати переріз розсіювання для багат шарової наночастинок та згенерувати дані для тренування нейромереж.

3. Натренувати нейромережі для знаходження матеріалів та радіусів шарів для двошарових та трьох-шарових наночастинок.

4. Оцінити точність роботи нейронної мережі для визначення структури наночастинок.

ОГЛЯД ЛІТЕРАТУРИ

1.1. Нейронні мережі

1.1.1. Історія розвитку нейронних мереж

Наука про штучні нейронні мережі зробила свою першу значну появу в 1940-х роках. Дослідники, які намагалися імітувати функції людського мозку, розробили фізичні моделі (згодом симуляції за допомогою програм) біологічних нейронів та їхніх взаємозв'язків. Оскільки нейробіологи поглиблювали знання про нервову систему людини, ці перші моделі вважалися все більш рудиментарними підходами. Тим не менш, деякі з результатів, отриманих у ці перші часи, були вражаючими, що спонукало до майбутніх досліджень і розробок складних і потужних штучних нейронних мереж [16].

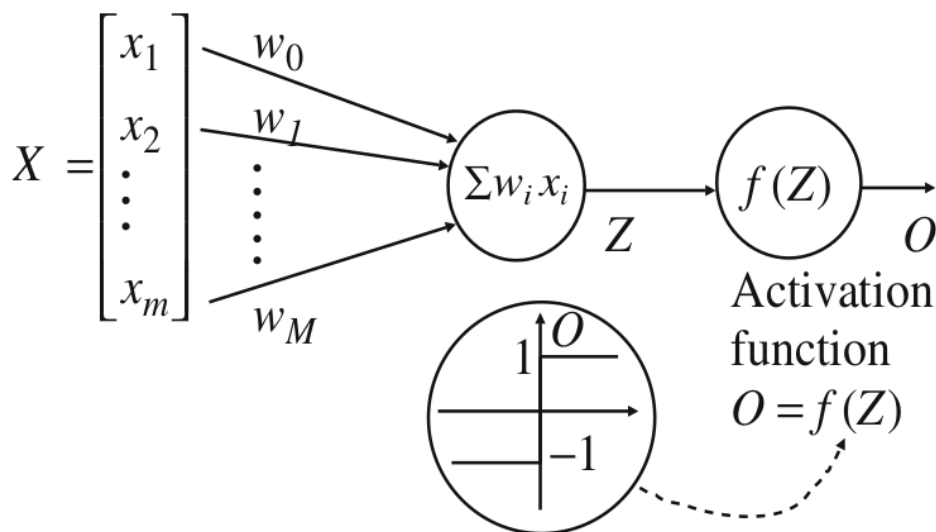


Рис. 1.1. Штучна модель біологічного нейрона. Як можна помітити, зв'язок між входом і виходом відповідає нелінійній функції, яка називається функцією активації. У першій моделі, показаній на цьому малюнку, активація є жорсткою пороговою функцією [16]

Мак Калох і Піттс, у 1943 та 1947, опублікували перші систематичні дослідження штучних нейронних мереж. Воно з'явилося на основі обчислювальної моделі нервової діяльності клітин нервової системи людини [16].

Більша частина їхньої роботи зосереджена на поведінці простого нейрона, чия математична чи обчислювальна модель показана на малюнку 1. Усередині штучного нейрона створюється сума кожного вхідного сигналу x_i , помножена на масштабний коефіцієнт (або вагу w_i). Вхідні дані емулюють збудження, отримані біологічними нейронами. Вагові коефіцієнти представляють силу синаптичного з'єднання: позитивний ваговий коефіцієнт означає збудливий ефект, а негативний — гальмівний ефект. Якщо результат суми перевищує певне порогове значення або зміщення (представлене вагою w_0), комірka активується, надаючи позитивне значення (зазвичай $+1$); у протилежному випадку вихід представляє від'ємне значення (зазвичай -1) або нуль. Отже, це двійковий вихід. В загальному, модель слідує нейробіологічній поведінці: нервові клітини виробляють нелінійні відповіді, коли їм надається збудження певним входом. Зокрема, McCulloch і Pitts запропонували функцію активації, яка представляє нелінійність моделі, яка називається жорсткою пороговою функцією [16].

Хоча ця перша модель може виконувати лише дуже прості завдання, потенціал нейронних систем по суті полягає у взаємозв'язку між нейронами для формування мереж. Цей взаємозв'язок зазвичай влаштований у вигляді шарів вузлів (штучних нейронів). Цей вид нейронних мереж називається багат шаровим перцептроном (MLP) [16].

Модель адаптивного лінійного елемента (ADALINE), що відноситься приблизно до того ж часу, просто повертала саме значення $f(x)$ для передбачення цільового числа (Widrow and Hoff, 1960) і також могла навчатися передбаченню чисел на даних [10].

Ці прості алгоритми навчання помітно вплинули на сучасний ландшафт машинного навчання. Алгоритм, використаний для підбору ваг у моделі ADALINE, був окремим випадком алгоритму стохастичного градієнтного спуску. Його трохи модифіковані варіанти і досі залишаються основними алгоритмами моделей глибокого навчання [10].

1980-і роки піднялася друга хвиля досліджень нейронних мереж, викликана головним чином рухом під назвою коннекціонізм, або паралельна розпо-

ділена обробка. Коннекціонізм виник у контексті когнітивістики – міждисциплінарного підходу до розуміння процесу пізнання, що поєднує кілька різних рівнів аналізу. На початку 1980-х років більшість когнітивістів вивчали моделі прийняття рішень шляхом маніпулювання символами (symbolic reasoning). Незважаючи на популярність символічних моделей, важко було пояснити, як мозок міг би реалізувати їх за допомогою нейронів [10].

Ще одним великим досягненням коннекціоністів стали успішне використання зворотного поширення для навчання глибоких нейронних мереж із внутрішніми уявленнями та популяризація алгоритму зворотного поширення. Його популярність то зростала, то зменшувалась, але на даний момент це переважний підхід до навчання глибоких моделей [10].

1990-ті роки стали часом важливих досягнень у моделюванні послідовностей за допомогою нейронних мереж. У роботах Hochreiter (1991) та Bengio et al. (1994) сформульовано низку фундаментальних математичних труднощів моделювання довгих послідовностей. У роботі Hochreiter and Schmidhuber (1997) введено поняття мереж із довгою короткостроковою пам'яттю (long short-term memory – LSTM) на вирішення деяких труднощів. Сьогодні LSTM-мережі широко використовують у багатьох завданнях моделювання послідовностей, зокрема для обробки природних мов у Google [10].

На теперішній час нейронні мережі використовуються для вирішення багатьох типів задач. Деякі з найбільш популярних є:

1) **Класифікація.** У задачах цього типу програма повинна відповісти, який з k категорій належить певний приклад. Для вирішення цього завдання алгоритм навчання зазвичай просять породити функцію $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$. Існують й інші варіанти завдання класифікації, наприклад коли f – розподіл ймовірності приналежності до класів. Прикладом завдання класифікації є розпізнавання об'єктів, коли на вхід подається зображення (зазвичай представлене безліччю яскравості пікселів), а на виході виходить числовий код, який ідентифікує присутній у зображенні об'єкт [10].

2) **Регресія.** У цьому завданні програма має передбачити числове значення за вхідними даними. Для її вирішення алгоритм навчання просять поро-

дити функцію $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Регресія відрізняється від класифікації форматом виходу. Прикладом може бути прогнозування суми претензії, яку претендуватиме застрахований, чи прогнозування майбутньої вартості цінних паперів. Такі прогнози застосовуються також у алгоритмічному трейдингу [10].

3) **Транскрипція.** У завданнях такого типу системі машинного навчання пропонується проаналізувати неструктуроване представлення деяких даних та перетворити його на дискретну текстову форму. Наприклад, програма розпізнавання тексту показує фотографію тексту, а вона повинна повернути текст у вигляді послідовності символів [10].

4) **Машинний переклад.** У цьому завдання вхідні дані – послідовність символів однією мовою, а програма має перетворити їх у послідовність символів іншою мовою. Зазвичай такі системи застосовуються до природних мов, скажімо, для перекладу з англійської на французьку [10].

5) **Виявлення аномалій.** У цій задачі програма аналізує безліч подій чи об'єктів і позначає деякі їх як нетипові. Прикладом може бути виявлення шахрайства з кредитними картками. Завдяки моделюванню звичок купівлі компанія-емітент кредитних карток може виявити аномальне використання картки. Покупки людини, яка вкрала вашу карту або інформацію про неї, часто характеризуються не таким розподілом ймовірності, як ваші. Виявивши аномальну покупку, компанія зможе запобігти шахрайству, заблокувавши рахунок. Огляд методів виявлення аномалій наведено у роботі Chandola et al. (2009) [10].

1.1.2. Архітектури нейромереж

1.1.2.1. Повнозв'язна нейронна мережа (Dense)

Нейрон – основна одиниця мозку. Невеликий його фрагмент, розміром приблизно з рисове зернятко, містить понад 10 тисяч нейронів, кожен з яких у середньому формує близько 6000 зв'язків з іншими такими клітинами⁵. Саме ця громіздка біологічна мережа дозволяє нам сприймати світ довкола. У цьому

розділі наше завдання скористатися природною структурою для створення моделей машинного навчання, які вирішують завдання аналогічно. По суті нейрон оптимізований для отримання інформації від «колег», її унікальної обробки та пересилання результатів в інші клітини. Нейрон отримує вхідну інформацію про дендрити — структури, що нагадують антени. Кожен із вхідних зв'язків динамічно посилюється або послаблюється на підставі частоти використання, і сила з'єднань визначає внесок вхідного елемента інформації в те, що нейрон видасть на виході. Вхідні дані оцінюються на основі цієї сили та поєднуються в клітинному тілі. Результат трансформується в новий сигнал, який поширюється клітинним аксоном до інших нейронів [4].

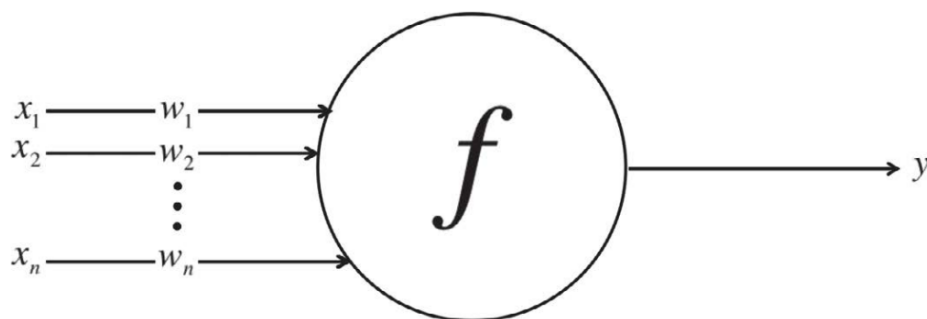


Рис. 1.2. Схема роботи нейрона у штучній нейромережі [4]

Ми можемо перетворити функціональне розуміння роботи нейронів у нашому мозку на штучну модель на комп'ютері. Остання описана на рис.1.2, де застосований підхід, вперше введений в 1943 Уоррен Маккаллоу і Уолтером Піттсом. Як і біологічні нейрони, штучний одержує деякий обсяг вхідних даних — $x_1, x_2, \dots, x_n$ кожен елемент яких множиться на певне значення ваги — $w_1, w_2, \dots, w_n$. Ці значення підсумовуються, даючи логіт нейрону: $z = \sum_{i=0}^n w_i x_i$. Часто він включає також зміщення (константа тут не показана). Логіт проходить через функцію активації f утворюючи вихідне значення $y = f(z)$. Це значення може бути передано до інших нейронів [4].

Представимо вхідні дані нейрона як вектор $x = [x_1, x_2, \dots, x_n]$, а ваги нейрона як $w = [w_1, w_2, \dots, w_n]$. Тепер вихідні дані нейрона можна виразити як $y = f(xw + b)$, де b - зміщення. Ми здатні обчислити вихідні дані із скалярного твору вхідного вектора на вектор ваг, додавши зміщення та отримавши логіт, а

потім застосувавши функцію активації. Це здається тривіальним, але уявлення нейронів у вигляді низки векторних операцій є дуже важливим [4].

Поодинокі нейрони могутніші за лінійні персептрони, але не здатні вирішити складні проблеми навчання. Тому наш мозок складається з багатьох нейронів. Наприклад, за допомогою одного з них неможливо розрізнити цифри, написані від руки. І щоб вирішувати складніші завдання, нам потрібні моделі машинного навчання [4].

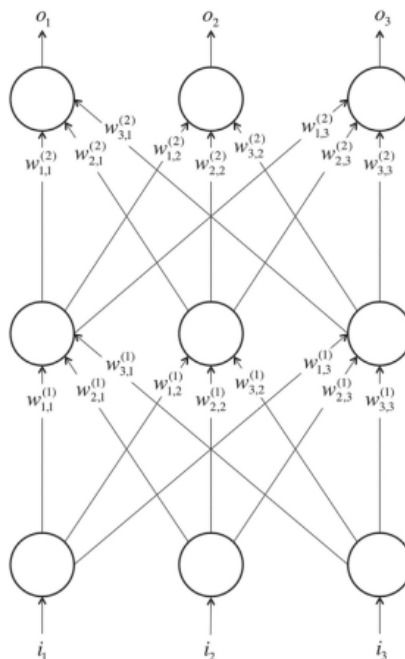


Рис. 1.3. Простий приклад нейромережі з прямим поширенням сигналу з трьома шарами (вхідний, прихований, вихідний) та трьома нейронами на кожен шар [4]

Нейрони у людському мозку розташовані шарами. Його кора, що здебільшого відповідає за інтелект, складається з шести шарів. Інформація перетікає по них, поки сенсорні дані не перетворюються на концептуальне розуміння. Наприклад, нижній шар візуальної зони кори отримує необроблені візуальні дані від очей. Ця інформація перетворюється в кожному наступному шарі і передається далі, поки на шостому шарі ми не укладаємо, що бачимо кішку, банку газування або літак [4].

На основі цих ідей ми можемо створити штучну нейромережу. Вона виникає, коли ми починаємо з'єднувати нейрони один з одним, з вхідними даними та вихідними вузлами, які відповідають відповідям мережі на завдання, що вивчається. На рис.1.3 показаний найпростіший приклад штучної нейромережі,

схожій по архітектурі з тією, що була описана в 1943 в роботі Маккаллоу і Піттса. У нижній шар надходять вхідні дані. Верхній (вихідні вузли) обчислює відповідь. Середній шар (шари) k нейронів називається прихованим, і тут $w_{i,j}^{(k)}$ - вага з'єднання i -го нейрона в k -му шарі з j -м нейроном в $(k + 1)$ -му шарі. Ці ваги утворюють вектор параметрів θ і, як і раніше, наша здатність вирішувати задачу за допомогою нейромереж залежить від знаходження оптимальних значень для θ [4].

Функції активації нейронів

Функція активації $f(z)$ — це функція через яку проходить логіт на виході з нейрона. Таких функцій є досить багато, кожна з яких краще підходить в тій чи іншій ситуації.

Найпростішою функцією активації є лінійна — $f(z) = az + b$. Вона застосовується в коли потрібно порахувати наприклад просту комбінацію признаков. Наприклад, нейрон, який намагається підрахувати вартість страви в кафе швидкого обслуговування, буде лінійним, $a = 1$ і $b = 0$ [4].

Обчислення з лінійними нейронами прості, але мають серйозні обмеження. Нескладно довести, що будь-яка нейромережа з прямим поширенням сигналу, що складається тільки з таких нейронів, може бути представлена як мережа без прихованих шарів. Це проблема: як ми вже говорили, саме приховані шари дозволяють пізнавати важливі властивості вхідних даних. Щоб навчитися розуміти складні відносини, потрібно використовувати нейрони з певною нелінійністю [4].

Окрім цієї функції активації є інші: *ReLU*, сигмоїдальна функція, гіперболічний тангенс та *softmax*.

У функція активації *ReLU* (Rectified Linear Unit) (рис. 1.4) визначається як $f(z) = \max\{0, z\}$ [10].

Ці блоки легко оптимізувати, тому що вони дуже схожі на лінійні. Різниця тільки в тому, що блок лінійної ректифікації в половині своєї області визначення виводить 0. Тому похідна блок лінійної ректифікації залишається великою усюди, де блок активний. Градієнти не лише великі, а ще й узгоджені. Друга

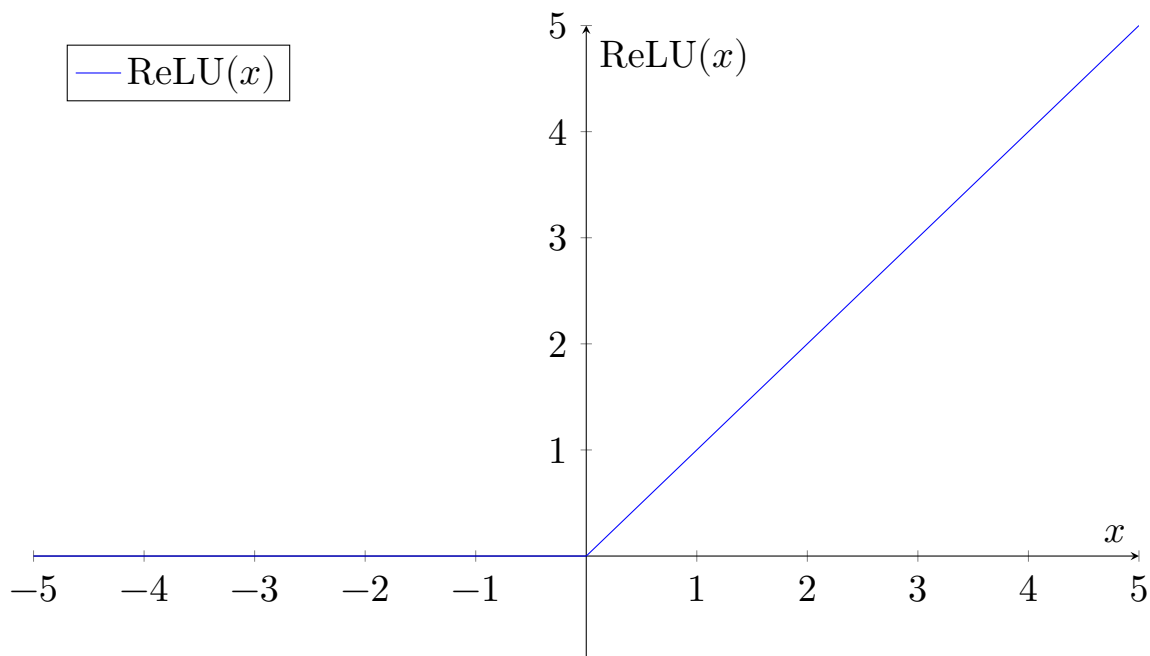


Рис. 1.4. Графік $ReLU$ функції [4]

похідна операції ректифікації всюди дорівнює нулю, а перша похідна дорівнює 1 усюди, де блок активний. Це означає, що напрям градієнта набагато корисніше для навчання, ніж у випадку, коли функція активації піддається ефектам другого порядку [10].

Недоліком блоків лінійної ректифікації є неможливість навчити їх методами градієнтами на прикладах, для яких функція активації блоку дорівнює нулю. Різні узагальнення гарантують, що градієнт є у будь-якій точці [10].

Блоки лінійної ректифікації стали використовуватися порівняно недавно, а раніше у більшості нейронних мереж у ролі функції активації застосовувалася логістична сигмоїда або гіперболічний тангенс [10].

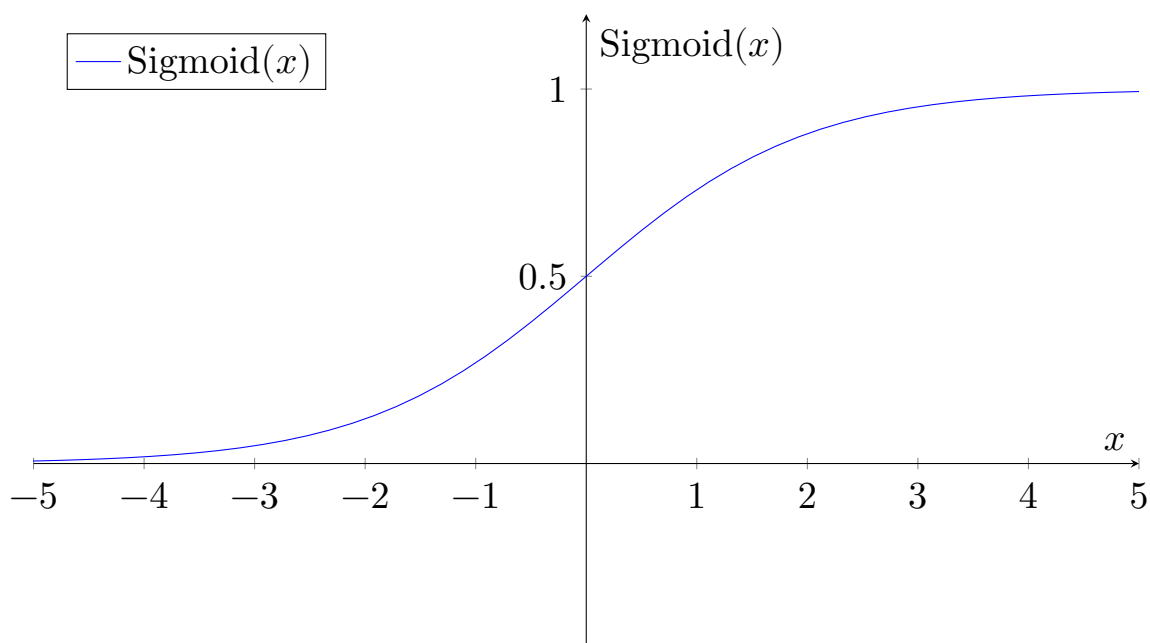
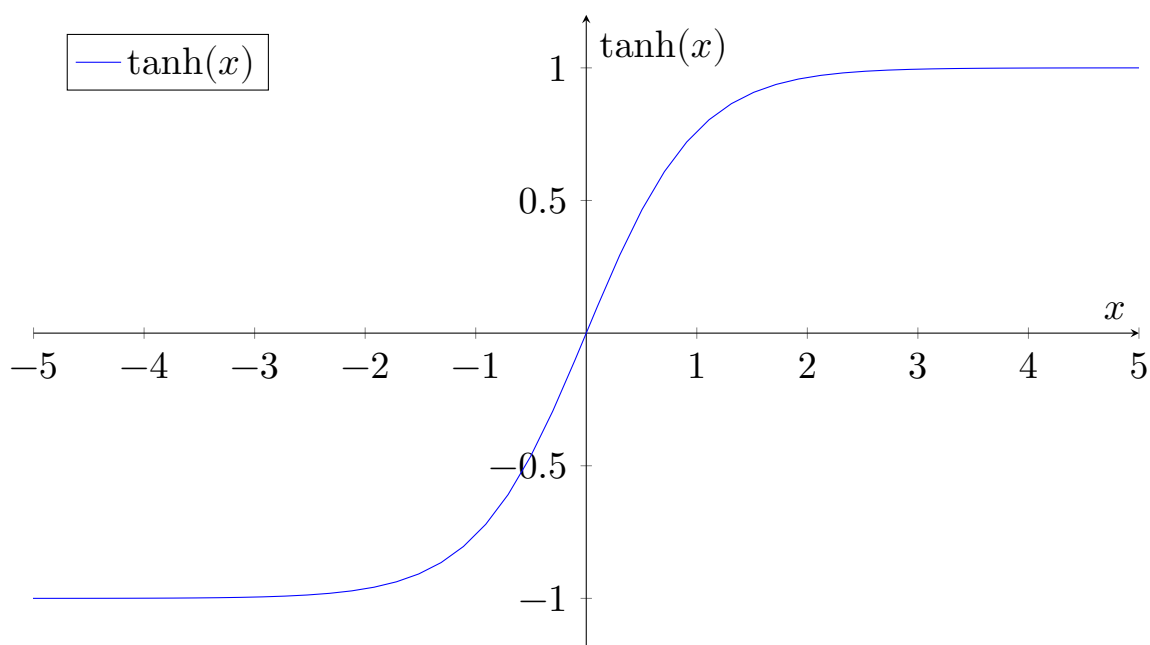
Формули для сигмоїдальної функції (рис. 1.5) та гіперболічного тангенсу (рис. 1.6) представлені нижче:

$$f(z) = \frac{1}{1 + e^{-z}} \quad (1.1)$$

$$f(z) = \tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1}$$

Ці функції активації тісно пов'язані виразом:

$$\tanh(z) = 2\sigma(2z) - 1 \quad (1.2)$$

Рис. 1.5. Графік *Sigmoid* функції [4]Рис. 1.6. Графік *tanh* функції [4]

Зазвичай вони використовуються коли стоїть задача класифікації якоїсь величини, що передбачає ймовірність того, що бінарна величина дорівнює 1. На відміну від кусково-лінійних, сигмоїдальні блоки близькі до асимптоти в більшій частині своєї області визначення – наближаються до високого значення, коли z прагне до нескінченності, і до низького, коли z прагне мінус нескінченності. Високу чутливість вони мають тільки в околиці нуля. Через насичення сигмоїдальних блоків градієнтне навчання дуже утруднене. Тому використання їх як прихованих блоків у мережах прямого поширення нині не рекомендується. Застосування ж як вихідні блоки сумісне з навчанням

градієнтними методами, якщо функція вартості компенсується насиченням сигмоїди у вихідному шарі [10].

Сигмоїдальні функції активації все ж таки застосовуються, але не в мережах прямого поширення. До рекурентних мереж, багатьом імовірнісним моделям і деяким автокодировщикам пред'являються додаткові вимоги, що виключають використання кусково-лінійних функцій активації і роблять сигмоїдальні блоки більш відповідними, незважаючи на проблеми насичення [10].

В задачах класифікації коли у нас є n класів використовувати функції активації для прикладу сигмоїди можна, але є не дуже доречним. Часто вихідний вектор краще представити таким чином щоб він був розподілом ймовірностей набору взаємовиключних значень. Тоді нам потрібно представити вектор таким чином щоб він показував на скільки ми впевнені що даний об'єкт належить до того чи іншого класу. Для прикладу у нас задача класифікації для n об'єктів, тоді логічно було б припустити що сумарна ймовірність то що наш об'єкт належить до одного і того ж класу дорівнює одиниці [10]. Це можна записати наступним чином:

$$\sum_0^n p_i = 1 \quad (1.3)$$

Тут p_i це ймовірність того що даний об'єкт належить до i -го класу.

Для цього використовується спеціальний вихідний шар, який називається шаром з м'яким максимумом (softmax). На відміну від інших типів, вихідні дані нейрона у шарі з м'яким максимумом залежать від вихідних даних решти нейронів у ньому [10].

При сильному передбаченні одне із значень вектора буде близько до 1, решта — до 0. При слабкому — залишиться кілька можливих значень, кожне з яких характеризується своїм рівнем ймовірності [10].

1.1.2.2. Рекурентні нейронні мережі

Рекурентні нейронні мережі, або RNM (Rumelhart et al., 1986a), - це сімейство нейронних мереж для обробки послідовних даних. Рекурентна ней-

ронна мережа призначена для обробки послідовності значень $x^{(1)}, \dots, x^{(\tau)}$. Дана мережа масштабується на набагато довгі послідовності, ніж було б практично можливо для неспеціалізованих нейронних мереж. Більшість рекурентних мереж здатні також обробляти послідовності змінної довжини [10].

Найпростіша рекурентна мережа являє собою граф обчислень (рис. 1.7). Це формальний спосіб описати структуру безлічі обчислень, наприклад, необхідних для відображення входів і параметрів на виходи і втрату.

Наприклад, розглянемо класичну форму такої системи:

$$s^{(t+1)} = f(s^{(t)}, \theta) \quad (1.4)$$

де $s^{(t)}$ є станом системи.

Рівняння 1.4 є рекурентне, тому що стан s в момент t посилається на той самий стан в момент $t - 1$ [10].

Для кінцевого числа тимчасових кроків граф можна розгорнути, застосувавши це визначення $\tau - 1$ раз. Наприклад, якщо розгорнути вираз 1.4 для $\tau = 3$ кроків, то отримаємо:

$$s^{(3)} = f(s^{(2)}, \theta) = f(f(s^{(1)}, \theta), \theta) \quad (1.5)$$

Після такої розгортки шляхом повторного застосування визначення виходить вираз, що не містить рекурсії. Такий вираз можна уявити традиційним ациклічним орієнтованим графом обчислень. Розгорнутий граф обчислень показано на рис. 1.7 [10].

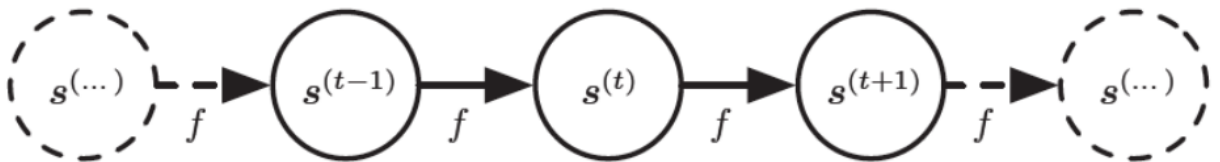


Рис. 1.7. Класична динамічна система, що описується виразом 1.4, як розгорнутого графа обчислень. Кожна вершина представляє стан у певний момент t , а функція f відображає стан у момент t на стан у момент $t + 1$. Одні й самі параметри використовуються на всіх тимчасових кроках [10]

Дякуючи механізмам розгорнення графів та розділення параметрів, ми можемо перейти до проектування різноманітних рекурентних нейронних мереж.

Рекурентна нейронна мережа на рис. 10.3 та рівняння (10.8) універсальні в тому сенсі, що будь-яка функція, що обчислюється машиною Тьюринга, може бути обчислена і такою рекурентною мережею кінцевого розміру. Результат можна прочитати з РНМ після виконання числа кроків, що асимптотично лінійно залежить від числа тимчасових кроків машини Тьюринга і від довжини вхідної послідовності (Siegelmann and Sontag, 1991; Siegelmann, 1995; Siegelmann and Sontag, 1995; Hyotyniemi, 1996). Функції, обчислювані машиною Тьюринга, дискретні, і тому ці результати відносяться до точної реалізації функції, а не до апроксимацій. Коли РНМ використовується як машина Тьюрінга, вона приймає на вході двійкову послідовність, а її виходи можна дискретизувати для отримання двійкового результату. У таких припущеннях можна обчислити будь-яку функцію за допомогою однієї конкретної РНМ кінцевого розміру (у роботі Siegelmann and Sontag [1995] використовується РНМ з 886 блоками). «Входом» машини Тьюринга є специфікація функції, що обчислюється, тому тієї ж мережі, яка моделює машину Тьюринга, достатньо для вирішення всіх завдань. Теоретична РНМ, що використовується в доказі, вміє моделювати необмежений стек, представляючи його активації та ваги раціональними числами необмеженої точності.

Виведемо рівняння прямого поширення для РНМ, зображеної на рис. 1.8. На цьому малюнку не показано конкретну функцію активації для прихованих блоків. Припускати, що це гіперболічний тангенс. Крім того, на малюнку точно не вказана форма виходу та функції втрат. Припускати, що вихід дискретний, як би РНМ застосовувалася для передбачення слів чи символів. Природний спосіб уявлення дискретних величин – розглядати вихід як не-нормовані логарифмічні можливості кожного можливого значення дискретної величини. Тоді на етапі постобробки можна застосувати операцію softmax і отримати вектор $\hat{y}$ нормованих ймовірностей. Пряме поширення починається

із завдання початкового стану $h^{(0)}$. Потім для кожного тимчасового кроку від $t = 1$ до $t = \tau$ застосовуємо наступні рівняння оновлення:

$$\begin{aligned}
 a^{(t)} &= b + Wh^{(t-1)} + Ux^{(t)} \\
 h^{(t)} &= \tanh(a^{(t)}) \\
 o^{(t)} &= c + Vh^{(t)} \\
 y^{(t)} &= \text{softmax}(o^{(t)})
 \end{aligned}
 \tag{1.6}$$

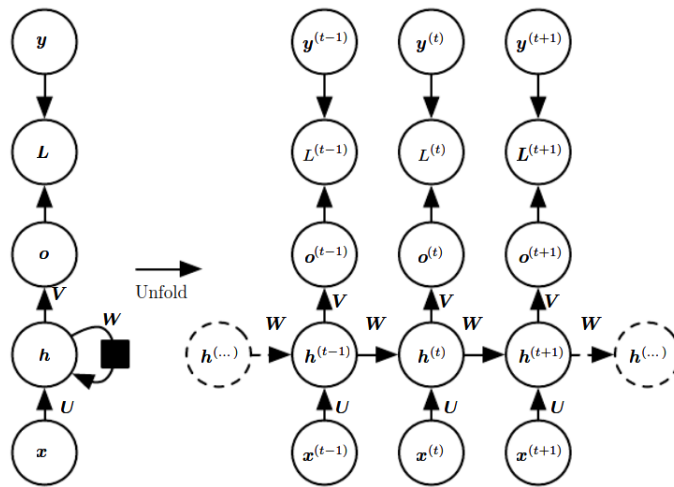


Рис. 1.8. Граф обчислень втрат під час навчання рекурентної мережі, яка відображає вхідну послідовність значень x відповідну вихідну послідовність значень o . Функція втрат L вимірює, наскільки далеко кожен елемент o віддалений від відповідної мітки y . У разі застосування до виходів функції softmax можна припускати, що o ненормовані логарифмічні ймовірності. У середині функція L обчислює $\hat{y} = \text{softmax}(o)$ і порівнює цю величину із міткою y . У РНМ є зв'язки між вхідним і прихованим шарами, параметризовані матрицею ваг U , рекурентні зв'язки між прихованими блоками, параметризовані матрицею ваг W , і зв'язки між прихованим і вихідним шарами, параметризовані матрицею ваг V . Рівняння 1.6 визначає пряме поширення цієї моделі. (Зліва) РНМ та її функція втрат, подані у вигляді рекурентних зв'язків. (Справа) Те саме у вигляді розгорнутого в часі графа обчислень, в якому кожна вершина асоційована з одним моментом часу [10]

Тут параметрами є вектори зміщення b і c , а також матриці ваг U , V і W відповідно для зв'язків між вхідним та прихованим шарами, між прихованим та вихідним шарами та між прихованими блоками. Це приклад рекурентної мережі, яка відображає вхідну послідовність на вихідну тієї ж довжини. Повна втрата для даної вхідної послідовності x разом із послідовністю міток y

дорівнює сумі втрат за всіма тимчасовими кроками. Наприклад, якщо $L(t)$ – негативна логарифмічна правдоподібність $y^{(t)}$ за умови $x^{(1)}, \dots, x^{(t)}$, то:

$$L(\{x^{(1)}, \dots, x^{(\tau)}\}, \{y^{(1)}, \dots, y^{(\tau)}\}) = \sum_t L^{(t)} = - \sum_t \log p_{model}(y^{(\tau)} | \{x^{(1)}, \dots, x^{(\tau)}\}) \quad (1.7)$$

Тут $p_{model}(y(t)|x(1), \dots, x(t))$ отримується з компоненту $y(t)$ вектора моделі $\hat{y}(t)$. Обчислення градієнта цієї функції відносно параметрів є витратною операцією. Для обчислення градієнта потрібно здійснити пряме поширення, рухаючись зліва направо по розгорнутому графіку на рис. 1.8, а потім зворотне поширення, рухаючись справа наліво по тому самому графіку. Час роботи має порядок $O(\tau)$, і його неможливо зменшити шляхом паралелізації, оскільки граф прямого поширення фундаментально послідовний: кожен часовий крок можна обчислити лише після завершення попереднього. Стани, обчислені під час прямого проходу, необхідно зберігати для подальшого використання під час зворотного проходу, тому обсяг використовуваної пам'яті також має порядок $O(\tau)$. Алгоритм зворотного поширення, застосований до розгорнутого графіку і маючи складність $O(\tau)$, називається зворотнім поширенням у часі (back-propagation through time - BPTT). Таким чином, мережа з рекурентними зв'язками між прихованими блоками є потужною, але дорогою для навчання [10].

Але у даній моделі виникає проблема зникаючого градієнта (vanishing gradient problem) є одним з викликів, що виникають при навчанні рекурентних нейронних мереж (RNN) у глибокому навчанні. Ця проблема полягає в тому, що градієнт (похідна) функції втрати по вагам мережі дуже швидко зменшується з кожним часовим кроком під час зворотного поширення [4; 10].

У RNN градієнт передається від кроку до кроку, із одного прихованого стану до іншого. Проблема полягає в тому, що при многокрокових RNN, особливо з великою кількістю часових кроків, градієнт може швидко зменшуватися з кожним кроком. Це стає особливо помітним при використанні функції активації, яка має маленькі похідні, такі як сигмоїда або гіперболічний тангенс [4; 10].

Коли градієнт зникає, навчання моделі стає складним, оскільки ваги не оновлюються належним чином, і модель не може вивчити відповідні залежності в даних. Це особливо проблематично в задачах, де довготривалі залежності між вхідними та вихідними сигналами важливі, наприклад, у випадку обробки природної мови або машинного перекладу [4; 10].

1.1.2.3. LSTM як ефективна рекурентна архітектура нейромереж

Для боротьби з проблемою зникаючого градієнта Зепп Хохрайтер та Юрген Шмідхубер запровадили архітектуру довгої короткострокової пам'яті (LSTM). Основний її принцип такий: мережа створюється для надійного перенесення важливої інформації на багато кроків у майбутнє. Ці міркування спричинили створення архітектури, показаної на рис. 1.9 [10].

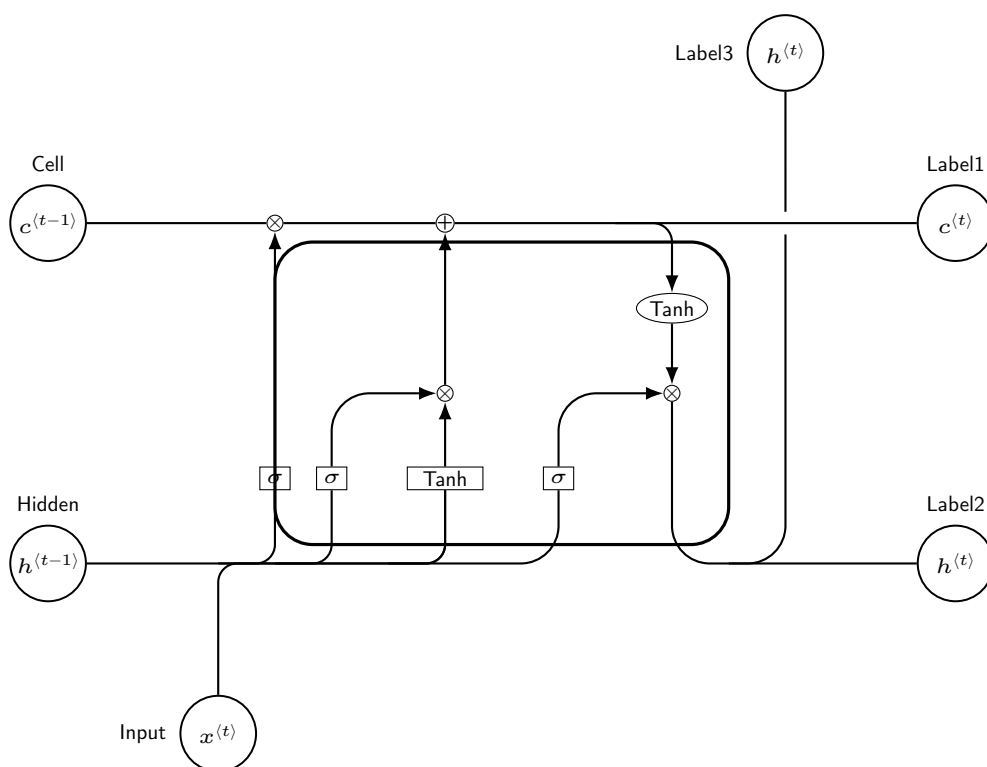


Рис. 1.9. Архітектура нейрона LSTM [10]

Як зрозуміло з малюнка, нейрон LSTM складається з кількох ключових компонентів. Один із них — осередок пам'яті. Вона містить важливу інформацію, яку засвоїла з часом, а мережа покликана ефективно зберігати в ній цю корисну інформацію протягом кількох кроків. На кожному кроці нейрон LSTM змінює комірку пам'яті, забезпечуючи її новою інформацією три етапи.

Спочатку він повинен визначити, яку частину попередньої інформації слід зберігати за допомогою вентиля забуття [10].

Основна ідея проста. Тензор стану пам'яті з попереднього кроку насичений інформацією, але її частина може бути застарілою, і її слід стерти. Ми з'ясовуємо, які елементи тензора є релевантними, а які вже ні, обчислюючи двійковий тензор (що складається з нулів та одиниць), який ми множимо на попередній стан. Якщо відповідне місце в двійковому тензорі містить 1, це означає, що місце комірки пам'яті, як і раніше, значуще і його потрібно зберегти. Якщо ж у цьому місці 0, воно втратило значимість і його слід забути [10].

Ми апроксимуємо цей двійковий тензор, з'єднавши вхідні дані цього кроку та вихідні дані нейрона LSTM з попереднього та наклавши на отриманий тензор сигмоїдний шар (sigmoid). Останній на виході дає значення, яке зазвичай дуже близько до 0 або 1 (єдиний виняток - якщо вхідне значення близько до 0). Вихідні дані сигмоїдного шару – гарне наближення двійкового тензора, чим можна скористатися при побудові вентиля забуття. Зрозумівши яку інформацію від колишніх станів слід зберегти, а яку забути ми переходимо до тієї, яку потрібно додати в пам'ять. Ця частина нейрона LSTM називається вентилям запису. Вона поділяється на дві основні частини. Перша визначає, яку інформацію ми хочемо додати до стану. Це обчислюється у шарі $\tanh$ шляхом створення проміжного тензора. Другий компонент визначає, які частини цього тензора ми хочемо ввести у новий стан, а які викинути та не записувати. Для цього ми апроксимуємо двійковий вектор із нулів та одиниць за допомогою тієї ж стратегії (сигмоїдного шару), що й для вентиля забуття. Потім ми примножуємо двійковий вектор на проміжний тензор і додаємо отриманий результат, створюючи новий вектор стану для LSTM [10].

Нарешті, на кожному етапі часу нейрон LSTM повинен видавати дані. Можна сприймати вектор стану як вихідні дані, але нейрон LSTM покликаний забезпечити більшу гнучкість, передаючи вихід тензор - інтерпретацію чи зовнішню «комунікацію» те, що містить вектор стану. Ми використовуємо структуру, майже ідентичну реалізованої для вентиля запису: шар $\tanh$ породжує проміжний тензор від вектора стану; сигмоїдний шар створює маску

двійкового тензора на основі поточного введення та попереднього виведення; проміжний тензор множиться на двійковий тензор, що дає кінцеві вихідні дані [10].

1.2. Огляд перерізу розсіювання для металевих наночастинок

Переріз розсіювання є важливою характеристикою для вивчення і використання металевих наночастинок у різних дисциплінах, включаючи нанофотоніку, плазмоніку та нанооптику. Цей параметр визначає ефективну площу поглинання або розсіювання світла наночастиною і є результатом взаємодії світла зі структурою наночастинки.

1.2.1. Теоретичні підходи до визначення перерізу розсіювання

Для визначення перерізу розсіювання існують різні теоретичні підходи. Один з найбільш поширених методів - це теорія Міє, яка базується на розкладі поля, що виникає при взаємодії світла з наночастиною у ряд Фур'є [3]. Цей підхід дозволяє визначити переріз розсіювання для різних режимів розсіювання, таких як електричне, магнітне або мультипольне розсіювання.

1.2.2. Вплив параметрів наночастинок на переріз розсіювання

Переріз розсіювання металевих наночастинок значно залежить від їх геометрії та фізичних властивостей. Наприклад, розмір, форма, композиція та структура наночастинок можуть суттєво впливати на їхні оптичні характеристики та переріз розсіювання [17]. Багато досліджень було проведено для вивчення впливу цих параметрів на розсіювання світла наночастинами та розробки стратегій для контролю цього явища.

1.2.3. Експериментальні методи вимірювання перерізу розсіювання

Для вимірювання перерізу розсіювання металевих наночастинок використовуються різні експериментальні методи. Серед них можна виділити спектроскопію розсіювання, яка дозволяє визначити оптичні властивості наночастинок, включаючи їхній переріз розсіювання [13]. Інші методи включають мікроскопію розсіювання та оптичну маніпуляцію наночастинок для дослідження їхнього розсіювання світла.

1.2.4. Діелектрична проникність металів

Розглянемо діелектричні властивості трьох металів: золота Au, срібла Ag та міді Cu. Діелектрична проникність металів є важливою характеристикою, яка визначає їх взаємодію з електромагнітним випромінюванням у широкому діапазоні частот.

Одна з найпоширеніших моделей, яку використовують для визначення діелектричної проникності металів в широкому діапазоні частот, є модель, запропонована Johnson та Christy [12]. Ця модель базується на експериментальних даних і надає достатньо точний опис діелектричної проникності Au, Ag та Cu. Вона враховує явища розсіювання світла, поглинання та взаємодії з електромагнітним полем на молекулярному рівні.

Крім того, існують також другорядні моделі, які детальніше описують певні аспекти діелектричної проникності металів. Наприклад, модель Друде-Лоренца, що базується на класичній теорії дробового відображення світла металами, використовується для опису розсіювання світла та взаємодії з електронами в металі. Ця модель дозволяє квантово-механічно описати процеси в металевих системах, зокрема взаємодію з електромагнітним полем.

Також існують моделі, які враховують нелінійні ефекти в металевих системах, такі як високоінтенсивне оптичне випромінювання та нелінійна оптична

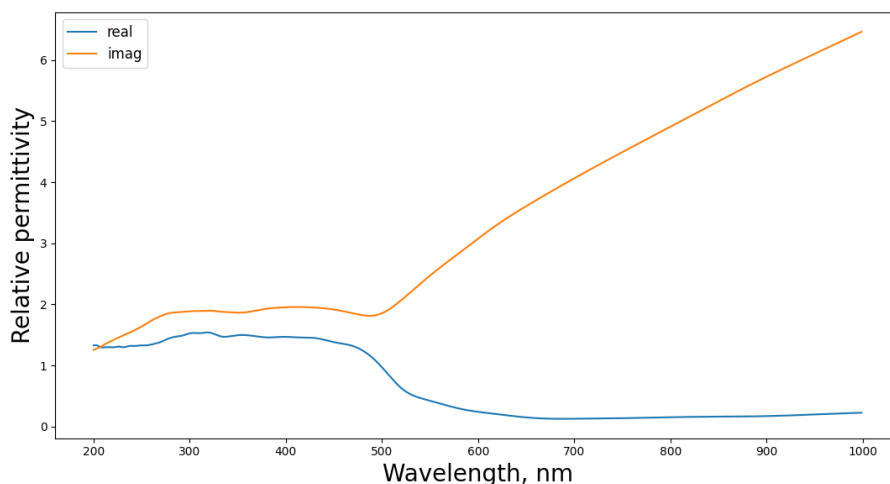


Рис. 1.10. Графік залежності діелектричної проникності від довжини падучої хвилі для Au [12]

відповідь. Ці моделі базуються на квантовій теорії розсіювання та враховують вплив взаємодії електронів в металі під впливом сильного електромагнітного поля.

В даній роботі для розрахунків діелектричної проникності металів Au, Ag та Cu використані дані, отримані з класичної статті [12]. Дані цих авторів є широко використовуваними в галузі плазмоніки та нанооптики для моделювання взаємодії світла з металевими наночастинками та структурами.

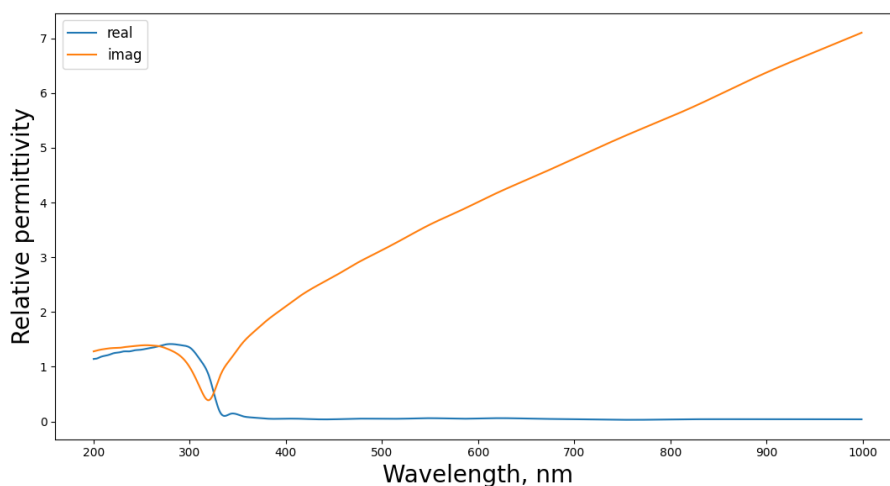


Рис. 1.11. Графік залежності діелектричної проникності від довжини падучої хвилі для Ag [12]

У статті [12] наведено значення діелектричної проникності металів Au, Ag та Cu в широкому діапазоні частот. Ці дані дозволяють враховувати вплив

дисперсійних ефектів та резонансних явищ при моделюванні взаємодії світла з металевими наночастинками.

Отримані графіки діелектричної проникності металів Au — рис. 1.10, Ag — рис. 1.11 та Cu — рис. 1.12.

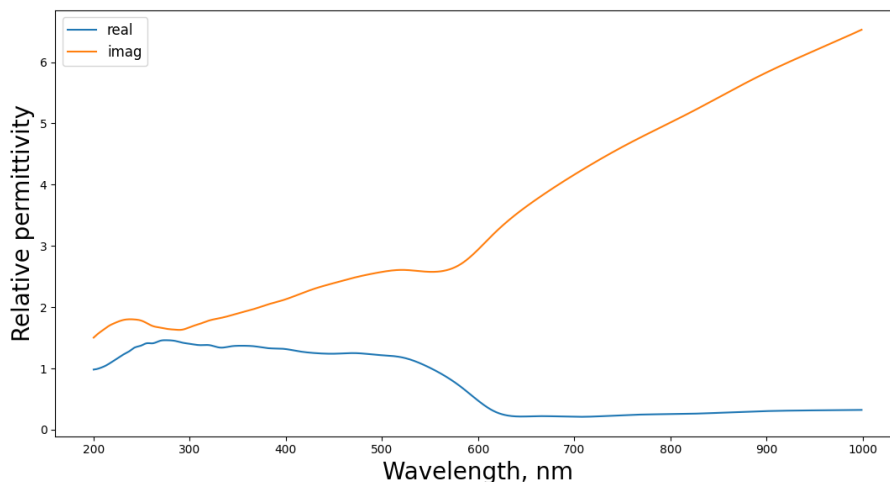


Рис. 1.12. Графік залежності діелектричної проникності від довжини падачої хвилі для Cu [12]

1.3. Проблематика та постановка задачі

ШНМ довели свою ефективність у моделюванні різних розподілів даних, вони здатні вивчати складні взаємозв'язки та закономірності, у тому числі і нелінійні. ШНМ також здатні до узагальнення, а саме робити передбачення на нових даних, яких вони раніше не бачили, застосовуючи вивчені закономірності. Їхній потенціал у обробці складних багатовимірних розподілів даних, а також можливість до адаптування до різних типів завдань, від опрацювання зображень і природної мови, до розробки персоніфікованих медичних діагностичних інструментів, робить їх новим дослідницьким інструментом у також прикладній електродинаміці.

У статті [1] автори надають комплексний огляд поточного стану досліджень щодо використання ШНМ у моделюванні антен. Автори підкреслюють переваги їхнього використання, такі як їх здатність до оптимізації параметрів антен швидше та точніше, ніж традиційні методи, особливо для випадку скла-

дних нелінійних залежностей. Також обговорюються деякі обмеження, такі як необхідність великої кількості даних.

Автори статті [2] надають огляд досліджень у галузі фотоніки та фотонних матеріалів, наголошують на потенціалі ШНМ у дизайні, оптимізації та контролі фотонних пристроїв та матеріалів. Також вони наводять загальний покроковий алгоритм застосування методів машинного навчання у таких задачах:

1) Збір даних: дані можуть бути отримані експериментально за допомогою вимірювання певних властивостей, або теоретично, як результат певних обчислень;

2) представлення даних: форма, у якій представлені необроблені дані часто впливає на алгоритм навчання, цей етап особливо важливий для класичних алгоритмів машинного навчання, на цьому етапі відбувається знаходження, для нейронних мереж зазвичай цей крок не потрібен, зазвичай достатньо нормалізації даних;

3) вибір моделі та її архітектури, навчання та оцінка моделей.

Алгоритми машинного навчання, зокрема ШНМ, використовуються для з'єднання інформації про структуру з інформацією про відгук (спектр, ширина забороненої зони, тощо), рис. 1.13.

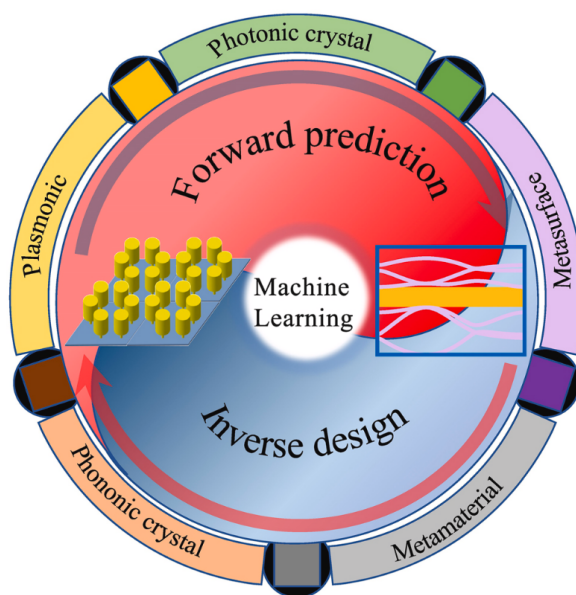


Рис. 1.13. Схематична діаграма застосування алгоритмів машинного навчання в фотоніці [11]

ШНМ стали дуже потужним інструментом саме інверсного проектування [6], тобто знаходження матеріалів та геометричних властивостей структур

з точки зору бажаних функціональностей. Після навчання ШНМ можуть вирішувати такі задачі набагато швидше їхніх більш традиційних аналогів, таких як генетичні алгоритми [8] та суміжний метод [9].

В статті [15] автори розглянули як пряму так і обернену задачу для багатошарової наночастинки, де чергуються шари діоксиду титану (TiO_2) і кремнію (silica). Розглянуті матеріали суто діелектричні без домішок металів. Показано, що нейромережа досить точно відновлює спектр, на якому не навчалась рис. 1.14. Вона з великою точністю прогнозує положення високодобротних резонансів (гострих піків). Також на рис. 1.14 зображені найближчі приклади із навчального датасета. Графік показує, що мережа не просто інтерполює або усереднює дані, а знаходить певні закономірності, що пов'язують геометрію структури і спектральний відгук.

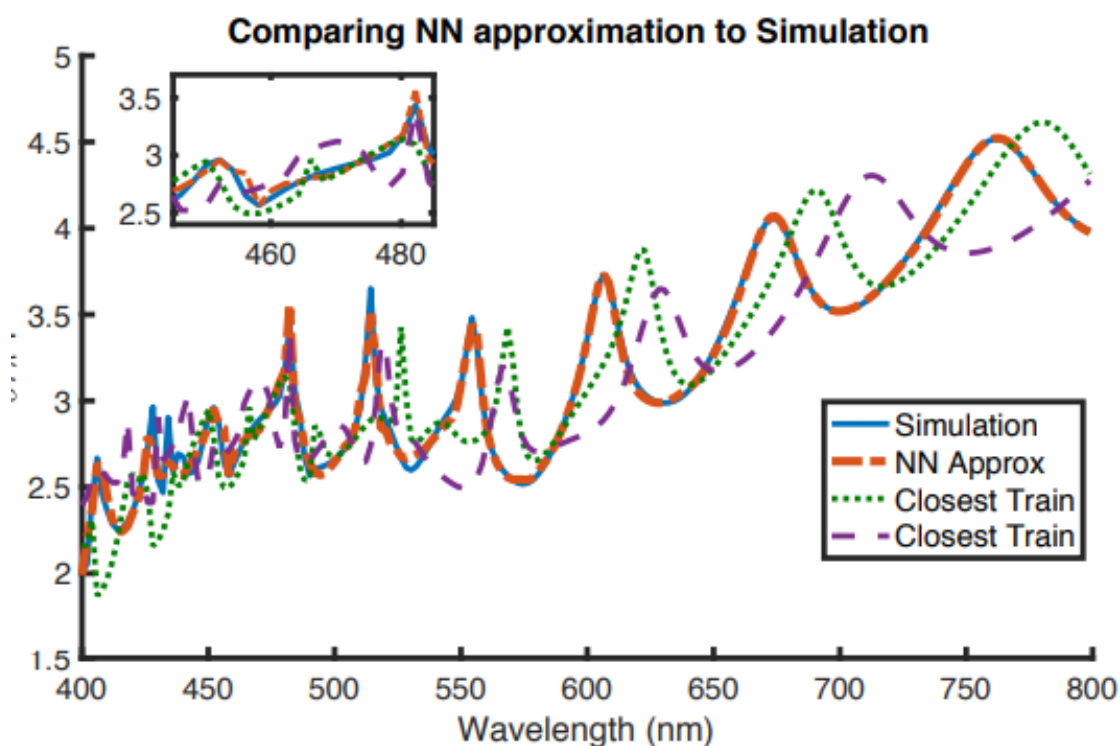


Рис. 1.14. Порівняння спектрального відгуку згенерованого ШНМ, очікуваного, а також найближчих спектральних відгуків що містились у вихідному наборі даних [15]

Для обернених задач досліджують різні архітектури і підходи. Так найбільш простим і унтуїтивно зрозумілим є підхід, в якому на вхід мережі досліджуваний електромагнітний відгук системи (наприклад спектр), а на вихід – параметри структури (геометрія, матеріал тощо). Також використовуються

так звані тандемні мережі [22]. У цьому випадку спочатку навчають модель на пряму задачу, потім навчену модель комбінують з моделлю зворотнього дизайну, навчають пару, тримаючи ваги першої незмінними. При навчанні мінімізується функція помилки, яка являє собою різницю між прогнозованим відгуком і цільовим відгуком.

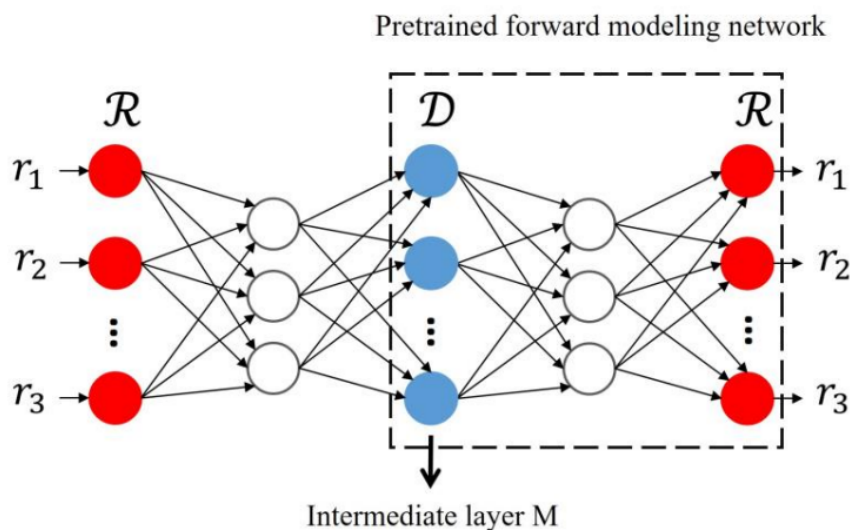


Рис. 1.15. Схематичний рисунок тандемної ШММ для оберненої задачі [22]

У статті [20] автори застосували «двонаправлений» багатошаровий персеPTRон (Рис. 1.16, а) для моделювання «H-shaped» плазмонних наноструктур (Рис. 1.16, b, c). НШМ складається з двох нейромереж, одна з яких передбачає геометрію по спектру (зворотній шлях), а інша передбачає спектр (прямий шлях). Геометрія, передбачена першою мережею, подається на вхід другій мережі. Для генерації датасета автори використовували COMSOL Multiphysics модуль.

Металеві наноструктури в оптичному діапазоні привертають до себе значну увагу насамперед можливістю сильної локалізації світла на субхвильовому рівні за рахунок збудження поверхневих і локалізованих плазмонів. Дослідження таких структур обумовлене можливостями їхнього використання як оптичних антен [14; 21], нанолазерів [19], біосенсорів [5; 7]. Локалізований поверхневий плазмонний резонанс виникає на поверхні наночастинки (зазвичай золотої або срібної) і він сильно залежить від розміру, форми та складу наночастинки [18].

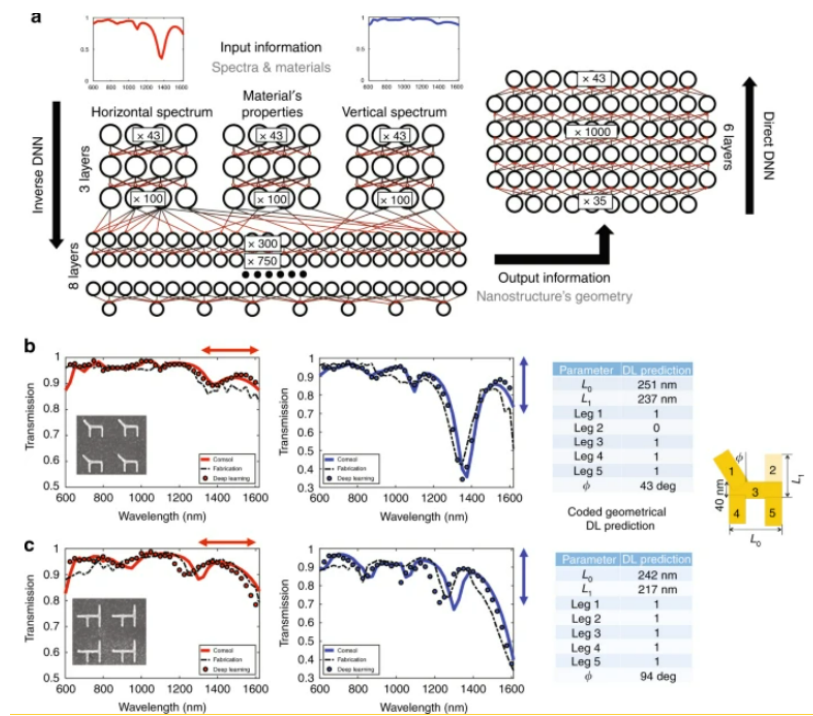


Рис. 1.16. Мережна архітектура глибокого навчання та результати **a**. Результати пошуку глибокого навчання для двох різних наноструктур золота, які були виготовлені **b, c** [20]

На Рис. 1.17 наведені спектри екстинції (чорний), поглинання (червоний) та розсіювання (синій) для срібних наночастинок різних форм.

Спектральні характеристики таких структур можуть бути скориговані підбором розміру, форми, у разі постійної форми (наприклад сферичної) додаванням шарів із різних матеріалів.

Поверхневі плазмони існують вздовж границі металу з діелектриком, за умови що діелектрична проникність одного з середовищ від'ємна. Оскільки метали, і, особливо, шляхетні метали, мають малу уявну частину комплексної діелектричної проникності і велику за модулем негативну дійсну частину, то саме на межі такого металу та діелектрика існують сильно локалізовані поверхневі плазмони.

Також важливо враховувати дисперсію показника заломлення металів в оптичному діапазоні. У цій роботі залежності величин діелектричних проникностей від довжини хвилі брались із експериментальної роботи [12].

В даній роботі задачею дослідження буде знаходження структури багатослоєвої сферичної наночастинок (радіус шарів та матеріал кожного з них), за допомогою нейромереж та порівняти різні нейромережі для виконання цієї задачі.

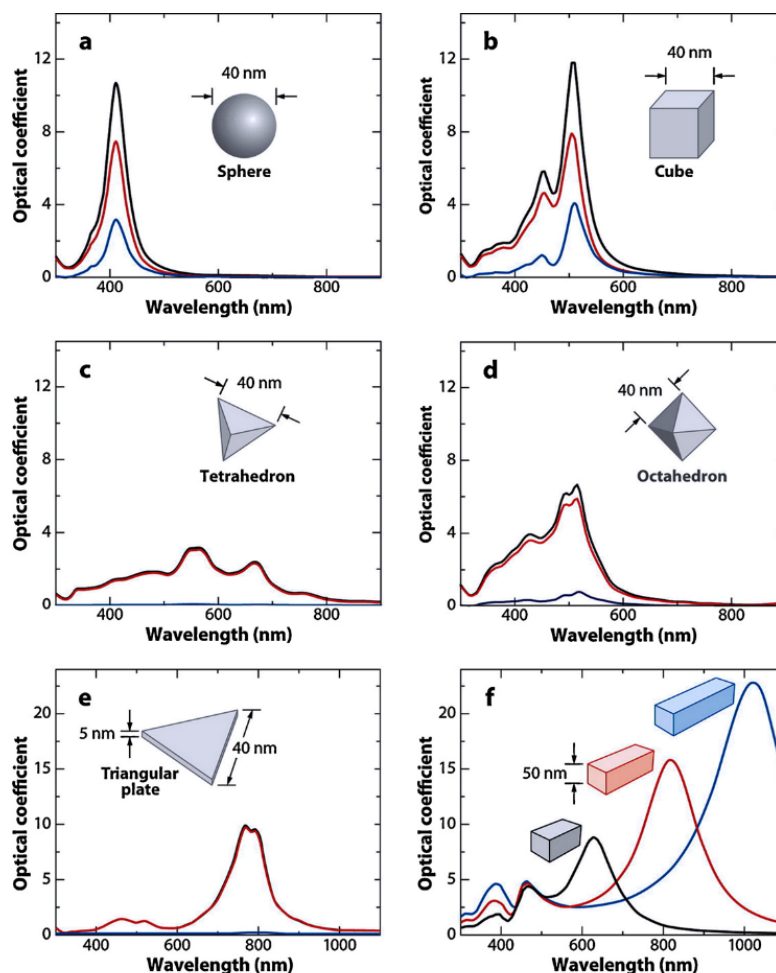


Рис. 1.17. наведені спектри екстинції (чорний), поглинання (червоний) та розсіювання (синій) для срібних наночастинок різних форм [18]

1.4. Висновки до першого розділу

В літературному огляді були розглянуті:

1. Основні принципи роботи та рекурентні архітектури нейромереж
2. Залежність діелектричної проникності металів від довжини падаючої хвилі та властивості перерізу розсіювання
3. Методи вирішення подібних задач за допомогою нейромереж
4. Метою даної роботи є знаходження структури двошарових та трьохшарових наночастинок за допомогою нейромереж.

МАТЕРІАЛИ ТА МЕТОДИ

2.1. Розсіювання плоскої хвилі на сфері

Розсіювання хвиль на сферичних об'єктах є ключовою проблемою в різних галузях науки та технологій, таких як оптика, акустика та радіолокація. Для опису цього феномену часто використовується рівняння Гельмгольца, яке є одним з основних рівнянь для хвильових процесів. Це рівняння дозволяє описати поширення хвиль у тривимірному просторі та їх взаємодію з розсіювачем. У цьому розділі ми розглядаємо розв'язок рівняння Гельмгольца для розсіювання на сфері та досліджуємо його вплив на форму та розподіл поля.

Рівняння Гельмгольца

$$\Delta U - \varepsilon_2 k^2 U = 0, r > a \quad (2.1)$$

$$\Delta U - \varepsilon_1 k^2 U = 0, 0 < r < a \quad (2.2)$$

Тут U відповідає або E_z або H_z , $\Delta = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \varphi^2}$ - оператор Лапласа.

Підставимо оператор Лапласа в рівняння 2.2:

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial U}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial U}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 U}{\partial \varphi^2} + \varepsilon k^2 U = 0 \quad (2.3)$$

Виконаємо розділення змінних

$$U(r, \theta, \varphi) = R(r)\Theta(\theta)\Phi(\varphi) \quad (2.4)$$

З 2.4 та 2.3 помножимо на $\frac{r^2}{R\Theta\Phi}$ маємо:

$$\frac{1}{R} \frac{d}{dr} \left(r^2 \frac{dR}{dr} \right) + \frac{1}{\Theta \sin \theta} \frac{d}{d\theta} \left(\sin \theta \frac{d\Theta}{d\theta} \right) + \frac{1}{\Phi \sin^2 \theta} \frac{d^2 \Phi}{d\varphi^2} + \varepsilon k^2 r^2 = 0 \quad (2.5)$$

2.1.1. Кутова частина рівняння

Якщо домножити 2.5 на $\sin^2 \theta$, то останній доданок буде залежати тільки від φ , а решта від r та θ .

Тому цей доданок є константою, яку позначимо через $-m^2$. Отримуємо рівняння:

$$\frac{1}{\Phi} \frac{d^2 \Phi}{d\varphi^2} \quad (2.6)$$

Причому має відбуватись $\Phi(\varphi) = \Phi(\varphi + 2\pi)$, звідки слідує:

$$\Phi(\varphi) = e^{in\varphi} \quad (2.7)$$

Підставимо 2.7 в 2.5:

$$\frac{1}{R} \frac{d}{dr} \left(r^2 \frac{dR}{dr} \right) + \frac{1}{\Theta \sin \theta} \frac{d}{d\theta} \left(\sin \theta \frac{d\Theta}{d\theta} \right) - \frac{m^2}{\sin^2 \theta} + \varepsilon k^2 r^2 = 0 \quad (2.8)$$

В рівнянні 2.8 другий та третій доданки є функціями тільки від θ , а тому є константою яку позначимо $l(l+1)$.

$$\frac{1}{\Theta \sin \theta} \frac{d}{d\theta} \left(\sin \theta \frac{d\Theta}{d\theta} \right) - \frac{m^2}{\sin^2 \theta} = -l(l+1) \quad (2.9)$$

Звідки

$$\frac{1}{\Theta \sin \theta} \frac{d}{d\theta} \left(\sin \theta \frac{d\Theta}{d\theta} \right) + \left(l(l+1) - \frac{m^2}{\sin^2 \theta} \right) \Theta = 0 \quad (2.10)$$

Зробимо заміну $x = \cos \theta$, тоді з 2.10:

$$\frac{d}{dx} \left[(1-x^2) \frac{d\Theta(x)}{dx} \right] + \left(l(l+1) - \frac{m^2}{1-x^2} \right) \Theta(x) = 0 \quad (2.11)$$

Рівняння 2.11 задає приєднані поліноми Лежандра

$$\Theta(x) = P_l^m(x) \quad (2.12)$$

Де $l = 0, 1, 2, \dots$, $m \in [-l, l]$. Добуток функцій Φ та Θ є сферичними гармоніками:

$$Y_l^m(\theta, \varphi) = C_l^m P_l^m(\cos \theta) e^{im\varphi} \quad (2.13)$$

Константу C_l^m можна отримати з умови нормування:

$$\int_0^{2\pi} d\varphi \int_0^\pi d\theta \sin \theta (Y_l^m(\theta, \varphi)^*) Y_{l'}^{m'}(\theta, \varphi) = \delta_{l,l'} \delta_{m,m'} \quad (2.14)$$

2.1.2. Радіальна частина рівняння

З рівнянь 2.8 та 2.10 маємо:

$$\frac{d}{dr} \left(r^2 \frac{dR}{dr} \right) + [\varepsilon k^2 r^2 - l(l-1)] R = 0 \quad (2.15)$$

або

$$r^2 \frac{d^2 R}{dr^2} + 2r \frac{dR}{dr} + [\varepsilon k^2 r^2 - l(l-1)] R = 0 \quad (2.16)$$

Введемо функцію $Z(r)$ для якої:

$$R(r) = \frac{Z(r)}{(kr)^{1/2} \varepsilon^{1/4}} \quad (2.17)$$

З 2.16 та 2.17 маємо:

$$r^2 \frac{d^2 Z}{dr^2} + r \frac{dZ}{dr} + \left[\varepsilon k^2 r^2 - l(l + \frac{1}{2})^2 \right] Z = 0 \quad (2.18)$$

Це рівняння Бесселя порядку $l + \frac{1}{2}$, розв'язками якого є функції Бесселя та Неймана: $J_{l+\frac{1}{2}}(\sqrt{\varepsilon}kr)$ і $N_{l+\frac{1}{2}}(\sqrt{\varepsilon}kr)$.

Остаточний розв'язок рівняння Гейльмгольца:

$$U(r, \theta, \varphi) = \sum_k \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{1}{(kr)^{1/2} \varepsilon^{1/4}} \left(a_{klm} J_{l+\frac{1}{2}}(\sqrt{\varepsilon} kr) + b_{klm} N_{l+\frac{1}{2}}(\sqrt{\varepsilon} kr) \right) Y_l^m(\theta, \varphi) \quad (2.19)$$

При $r \rightarrow \infty \Rightarrow N_{l+\frac{1}{2}} \rightarrow 0$ тому в області $r < a : b_{klm} = 0$. Також у середовищі у нас є падаюча хвиля.

Отже:

$$U(r, \theta, \varphi) = \sum_k \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{1}{(kr)^{1/2} \varepsilon_1^{1/4}} a_{klm}^{(2)} J_{l+\frac{1}{2}}(\sqrt{\varepsilon_1} kr) Y_l^m(\theta, \varphi), \text{ якщо } r < a \quad (2.20)$$

$$U(r, \theta, \varphi) = \sum_k \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{1}{(kr)^{1/2} \varepsilon_1^{1/4}} (a_{klm}^{(1)} J_{l+\frac{1}{2}}(\sqrt{\varepsilon_2} kr) + b_{klm}^{(1)} N_{l+\frac{1}{2}}(\sqrt{\varepsilon_2} kr) + \varepsilon_2^{1/4} J_l(kr)) Y_l^m(\theta, \varphi), \text{ якщо } r > a \quad (2.21)$$

2.1.3. Перехід до сферичних функцій

При вирішенні рівняння Гельмгольца в сферичних координатах ми можемо замінити функції Бесселя та Неймана сферичними функціями, зв'язок між якими має вид:

$$S_- J_l(z) = \sqrt{\frac{\pi}{2z}} J_{l+\frac{1}{2}}(z) \quad (2.22)$$

$$S_- Y_l(z) = \sqrt{\frac{\pi}{2z}} N_{l+\frac{1}{2}}(z) \quad (2.23)$$

Після підстанови сферичних функцій розв'язки рівняння будуть мати вид:

$$U(r, \theta, \varphi) = \sqrt{\frac{2}{\pi}} \sum_k \sum_{l=0}^{\infty} \sum_{m=-l}^l a_{klm}^{(2)} S_- J_l(\sqrt{\varepsilon_1} kr) Y_l^m(\theta, \varphi), \text{ якщо } r < a \quad (2.24)$$

$$U(r, \theta, \varphi) = \sqrt{\frac{2}{\pi}} \sum_k \sum_{l=0}^{\infty} \sum_{m=-l}^l \left(a_{klm}^{(1)} S_{-} J_l(\sqrt{\varepsilon_2} kr) + b_{klm}^{(1)} S_{-} Y_l(\sqrt{\varepsilon_2} kr) + J_l(kr) \right) Y_l^m(\theta, \varphi), \text{ якщо } r > a \quad (2.25)$$

2.1.4. Граничні умови

$$\begin{aligned} U|_{r \rightarrow a-0} &= U|_{r \rightarrow a+0} \\ n_2 \frac{\partial U}{\partial r}|_{r \rightarrow a-0} &= n_1 \frac{\partial U}{\partial r}|_{r \rightarrow a+0} \end{aligned} \quad (2.26)$$

$$\begin{aligned} a_{klm}^{(2)} S_{-} J_l(\sqrt{\varepsilon_1} ka) + J_l(kr) &= a_{klm}^{(1)} S_{-} J_l(\sqrt{\varepsilon_2} ka) + b_{klm}^{(1)} S_{-} Y_l(\sqrt{\varepsilon_2} ka) \\ a_{klm}^{(2)} J_l'(\sqrt{\varepsilon_1} ka) + J_l'(kr) &= \left(a_{klm}^{(1)} S_{-} J_l'(\sqrt{\varepsilon_2} ka) + b_{klm}^{(1)} S_{-} Y_l'(\sqrt{\varepsilon_2} ka) \right) \end{aligned} \quad (2.27)$$

Приймемо амплітуду падаючої хвилі за 1 та поділимо 2 рівняння на $a_{klm}^{(2)}$.
Тоді:

$$\begin{aligned} S_{-} J_l(\sqrt{\varepsilon_1} ka) + J_l(kr) &= A_{klm} S_{-} J_l(\sqrt{\varepsilon_2} ka) + B_{klm} S_{-} Y_l(\sqrt{\varepsilon_2} ka) \\ S_{-} J_l'(\sqrt{\varepsilon_1} ka) + J_l'(kr) &= A_{klm} S_{-} J_l'(\sqrt{\varepsilon_2} ka) + B_{klm} S_{-} Y_l'(\sqrt{\varepsilon_2} ka) \end{aligned} \quad (2.28)$$

Звідки можна знайти A_{klm} та B_{klm} .

2.2. Розсіювання на багатошаровій сфері

Розв'язок рівняння Гельмгольца для багатошарової сфери можна записати наступним чином

$$\begin{aligned}
U(r, \theta, \varphi) &= \sqrt{\frac{2}{\pi}} \sum_k \sum_{l=0}^{\infty} \sum_{m=-l}^l S_- J_l(\sqrt{\varepsilon_1} kr) Y_l^m(\theta, \varphi), \text{ якщо } r < a_0 \\
U(r, \theta, \varphi) &= \sqrt{\frac{2}{\pi}} \sum_k \sum_{l=0}^{\infty} \sum_{m=-l}^l \left(A_{klm}^{(1)} S_- J_l(\sqrt{\varepsilon_2} kr) + B_{klm}^{(1)} S_- Y_l(\sqrt{\varepsilon_2} kr) \right) \\
&\quad Y_l^m(\theta, \varphi), \text{ якщо } a_0 < r < a_1 \\
&\quad \dots \\
U(r, \theta, \varphi) &= \sqrt{\frac{2}{\pi}} \sum_k \sum_{l=0}^{\infty} \sum_{m=-l}^l \left(A_{klm}^{(n)} S_- J_l(\sqrt{\varepsilon_n} kr) + B_{klm}^{(n)} S_- Y_l(\sqrt{\varepsilon_n} kr) + J_l(kr) \right) \\
&\quad Y_l^m(\theta, \varphi), \text{ якщо } a_n < r
\end{aligned} \tag{2.29}$$

Тоді систему для знаходження коефіцієнтів (2.28) буде мати наступний вигляд:

$$\begin{aligned}
S_- J_l(\sqrt{\varepsilon_1} ka_0) &= A_{klm}^0 S_- J_l(\sqrt{\varepsilon_2} ka_0) + B_{klm}^0 S_- Y_l(\sqrt{\varepsilon_2} ka_0) \\
\frac{n_1}{n_2} S_- J'_l(\sqrt{\varepsilon_1} ka_0) &= A_{klm}^0 S_- J'_l(\sqrt{\varepsilon_2} ka_0) + B_{klm}^0 S_- Y'_l(\sqrt{\varepsilon_2} ka_0) \\
A_{klm}^0 S_- J_l(\sqrt{\varepsilon_2} ka_1) + B_{klm}^0 S_- Y_l(\sqrt{\varepsilon_1} ka_1) &= A_{klm}^1 S_- J_l(\sqrt{\varepsilon_3} ka_1) + \\
&\quad B_{klm}^1 S_- Y_l(\sqrt{\varepsilon_3} ka_1) \\
\frac{n_2}{n_3} (A_{klm}^0 S_- J'_l(\sqrt{\varepsilon_2} ka) + B_{klm}^0 S_- Y'_l(\sqrt{\varepsilon_2} ka)) &= A_{klm}^1 S_- J'_l(\sqrt{\varepsilon_3} ka) + \\
&\quad B_{klm}^1 S_- Y'_l(\sqrt{\varepsilon_3} ka) \\
&\quad \dots \\
A_{klm}^n S_- J_l(\sqrt{\varepsilon_n} ka_1) + B_{klm}^n S_- Y_l(\sqrt{\varepsilon_n} ka_1) &= A_{klm}^{n+1} S_- J_l(\sqrt{\varepsilon_{n+1}} ka_1) + \\
&\quad B_{klm}^{n+1} S_- Y_l(\sqrt{\varepsilon_{n+1}} ka_1) + J_l(kr) \\
\frac{n_n}{n_{n+1}} (A_{klm}^n S_- J'_l(\sqrt{\varepsilon_n} ka) + B_{klm}^n S_- Y'_l(\sqrt{\varepsilon_n} ka)) &= A_{klm}^{n+1} S_- J'_l(\sqrt{\varepsilon_{n+1}} ka) + \\
&\quad B_{klm}^{n+1} S_- Y'_l(\sqrt{\varepsilon_{n+1}} ka) + J'_l(kr)
\end{aligned} \tag{2.30}$$

Дану систему можна вирішити за допомогою метода Крамера.

2.3. Переріз розсіювання для наночастинок

Маючи коефіцієнти розв'язку рівняння Гельмгольца, ми можемо знайти переріз розсіювання для наночастинки:

$$C_{SCA} = \frac{2\pi}{k^2} \sum_{l=0}^{\infty} (2l+1) (|a_n|^2 + |b_n|^2) \quad (2.31)$$

тут a_n та b_n коефіцієнти розв'язку рівняння в середовищі.

2.4. Нейромережа для знаходження матеріалів наночастинок

Для написання нейромережі була використана бібліотека Keras, яка дозволяє досить швидко створювати нейромережі та підтримує багато різних архітектур.

Для вирішення було побудовано 3 нейромережі:

- 1) повнозв'язна нейронна мережа
- 2) мережа з простим рекурентним шаром
- 3) мережа з LSTM шаром

Архітектура нейромережі показана на рис. 2.1.

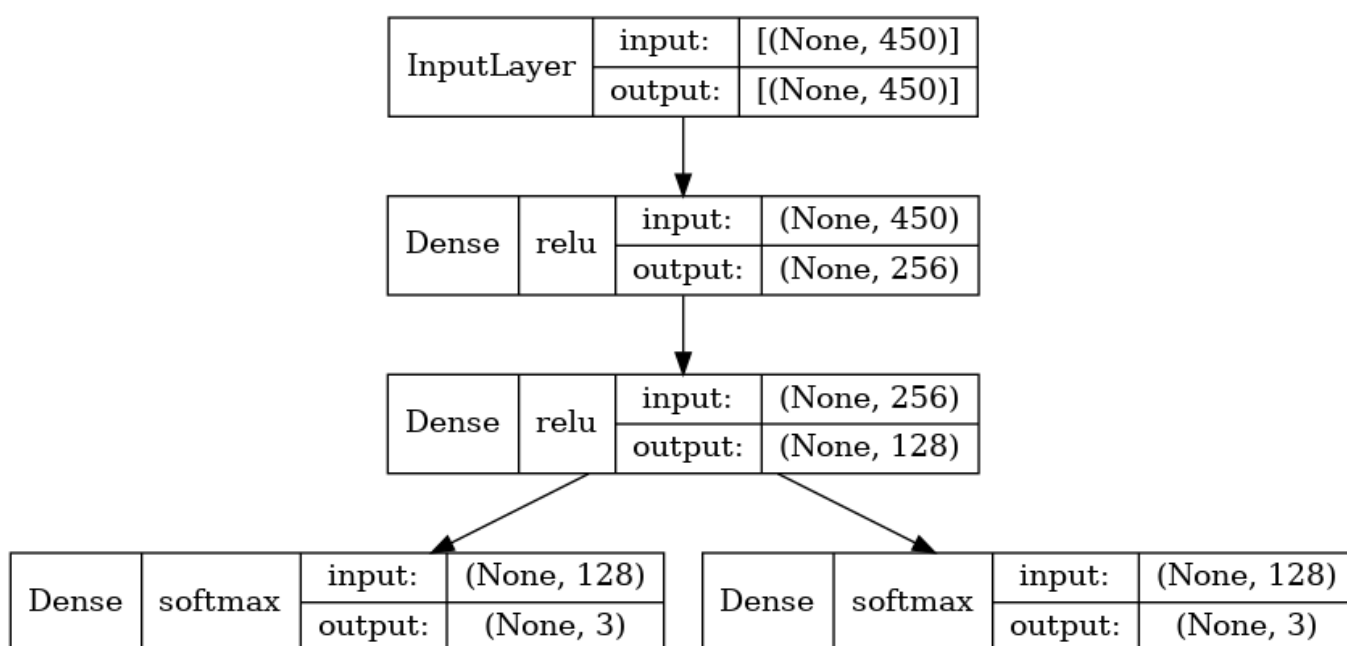


Рис. 2.1. Архітектура повнозв'язної нейронної мережі

Як видно з рис. 2.1 нейромережа мала 1 вхідний шар на який подавався сам переріз розсіювання, 1 скритий шар та 2 шари на виході оскільки ми маємо двошарову наносферу.

Для тренування нейронної мережі вхідні дані були представлені наступним чином. На вхід подавався переріз розсіювання. А дані для на виході представлялись у вигляді як показано в табл. 2.1.

EPS1_Ag	EPS1_Au	EPS1_Cu	EPS2_Ag	EPS2_Au	EPS2_Cu
0	1	0	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0
0	0	1	0	1	0

Таблиця 2.1

Приклад цільових даних для знаходження матеріалів двошарової наночастинки

В даному випадку ми використали OneHotEncoding для представлення вихідних даних та хочемо щоб нейронна мережа дала нам ймовірності який матеріал знаходиться в першому шарі (EPS1) та в другому шарі (EPS2).

Дві наступні нейронні мережі мають наступну структуру. Друга нейронна мережа представлена на рис. 2.2, а третя на рис. 2.3.

У всіх трьох випадках за функцію втрат була взята *categoricalcrossentropy*, яка зазвичай використовується для задач класифікації.

$$H(p, q) = - \sum_i p(i) \log(q(i)) \quad (2.32)$$

тут:

- 1) $p(i)$ - ймовірність i -го класу в розподілі p ,
- 2) $q(i)$ - ймовірність i -го класу в розподілі q ,
- 3) $H(p, q)$ - категоріальна крос-ентропія між p та q .

Ця функція приймає два вектора: y_true (істинні мітки) і y_pred (прогнозовані ймовірності). Вона розраховує крос-ентропію шляхом обчислення суми виразу $-y_true * \log(y_pred)$ для кожного елементу і потім обчислює загальну суму цих значень [4; 10].

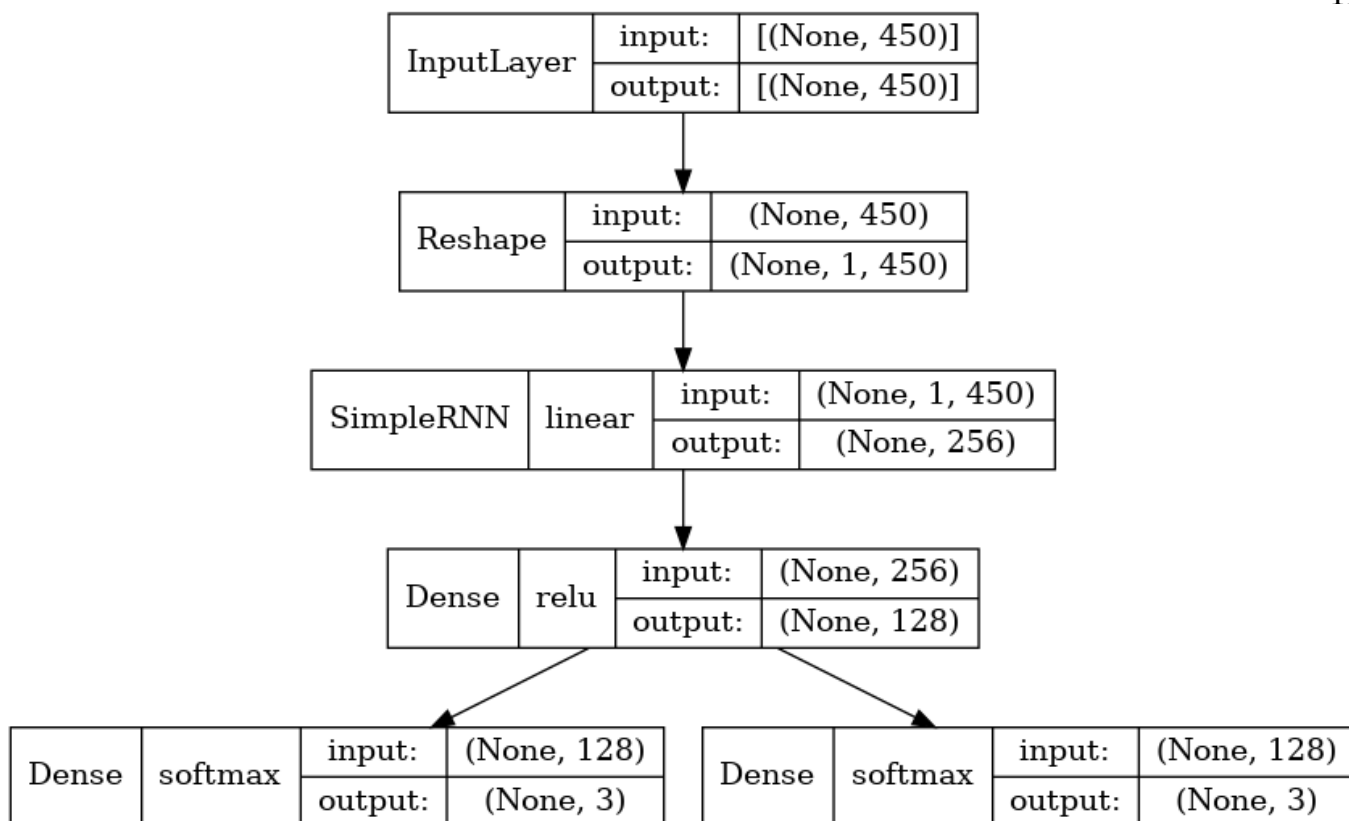


Рис. 2.2. Архітектура рекурентної нейронної мережі з використанням SimpleRNN шару

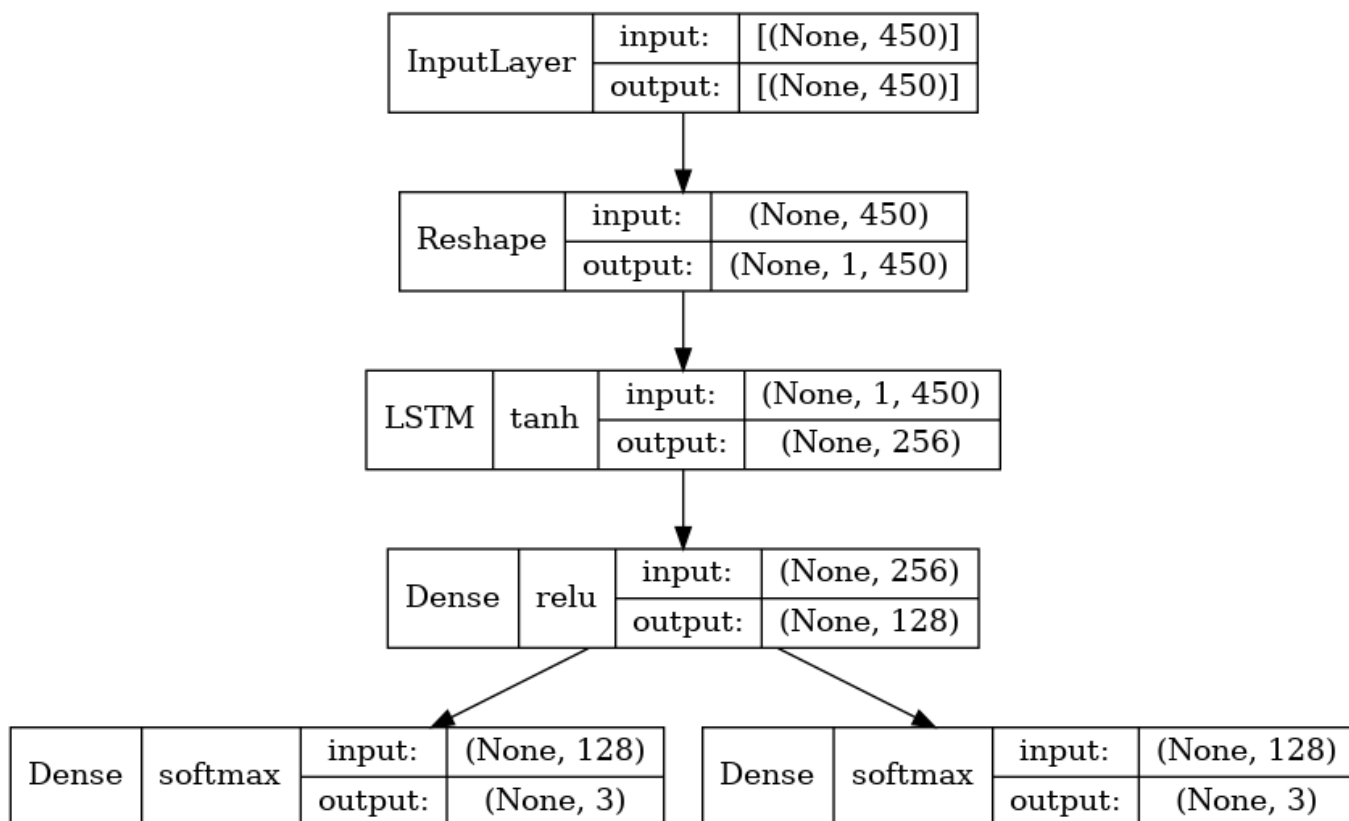


Рис. 2.3. Архітектура рекурентної нейронної мережі з використанням LSTM шару

Також були інші способи для побудови моделей. Для прикладу передбачати саму комбінацію матеріалів, наприклад "Ag|Au" або "Au|Cu", але в такому

варіанті при збільшені кількості матеріалів, або ж кількості шарів кількість вихідних міток класу досить стрімко зростає. Що досить сильно впливає на використання ресурсів для тренування нейромережі.

2.5. Нейромережа для знаходження радіусів наночастинок

Для знаходження радіусів частинок було використано такі ж самі архітектури нейронних мереж. Різниця була тільки у вихідному шарі. У даному випадку виходом був дох-вимірний тензор, який відповідав за відповідні радіуси частинки.

В даному випадку у якості функції втрат було використано MAE (Середня абсолютна похибка):

$$MAE = \frac{1}{n} \sum_{i=0}^n |y_i - \hat{y}| \quad (2.33)$$

тут:

- 1) n - кількість даних,
- 2) y_i - спостережене значення,
- 3) $\hat{y}$ - прогнозоване значення.

Також було застосовано у якості функції втрат MSE (Середнє квадратичне відхилення):

$$MSE = \frac{1}{n} \sum_{i=0}^n (y_i^2 - \hat{y}^2) \quad (2.34)$$

але при тренуванні нейромережі використовуючи дану функції, нейронна мережа видавала більш неточні результати чим з функцією втрат MAE .

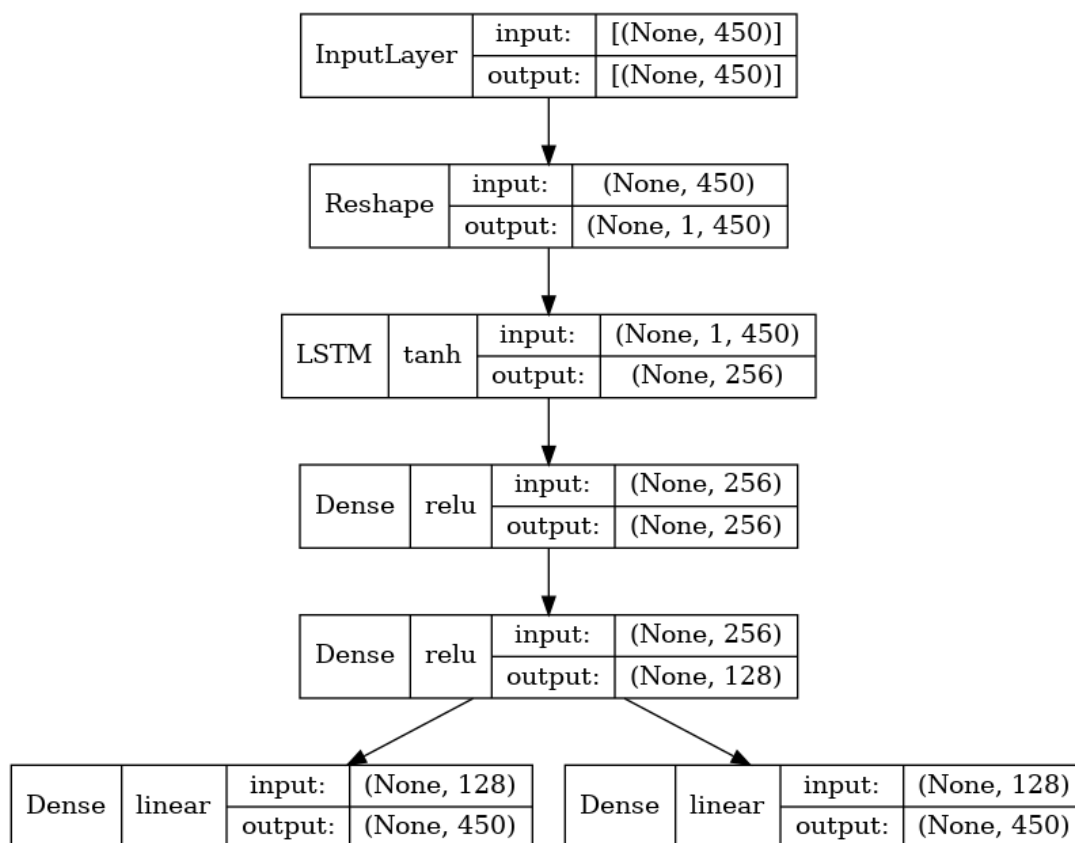


Рис. 2.4. Архітектура рекурентної нейронної мережі з використанням LSTM шару для знаходження діелектричної проникності матеріалу

2.6. Нейромережі для знаходження діелектричної проникності матеріалів

Можлива ситуація коли у нас є переріз розсіювання частинки з матеріалом на якому ми не навчали нейромережу. У такому випадку ми ніяк не зможемо знайти структуру наночастинки. Для вирішення такої ситуації можна поспробувати навчити нейромережу видавати не мітки матеріалів для тої чи іншої частинки, а те що може охарактеризувати сам матеріал. Для прикладу це може бути сама залежність діелектричної проникності від довжини падаючої хвилі.

У такому випадку нейромережа буде виглядати наступним чином(рис. 2.4).

Різниця між цією моделлю та минулими полягає тільки у вихідному шарі.

В даному випадку для виходів доцільніше застосувати функцію втрат *MSE*.

2.7. Висновки до другого розділу

В другому розділі було розглянуто:

1. Розв'язок рівняння Гельмгольца та метод для обчислення перерізу розсіювання
2. Методи побудови нейромереж для знаходження матеріалів наночастинок.
3. Методи побудови нейромереж для знаходження радіусів наночастинок.

РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

3.1. Отримані перерізи розсіювання та генерація даних

За допомогою теорії описаної в розділі 2 отримані графіки для двошарової(рис. 3.1) та тьох-шарової(рис. 3.2) сфер, які наведені нижче.

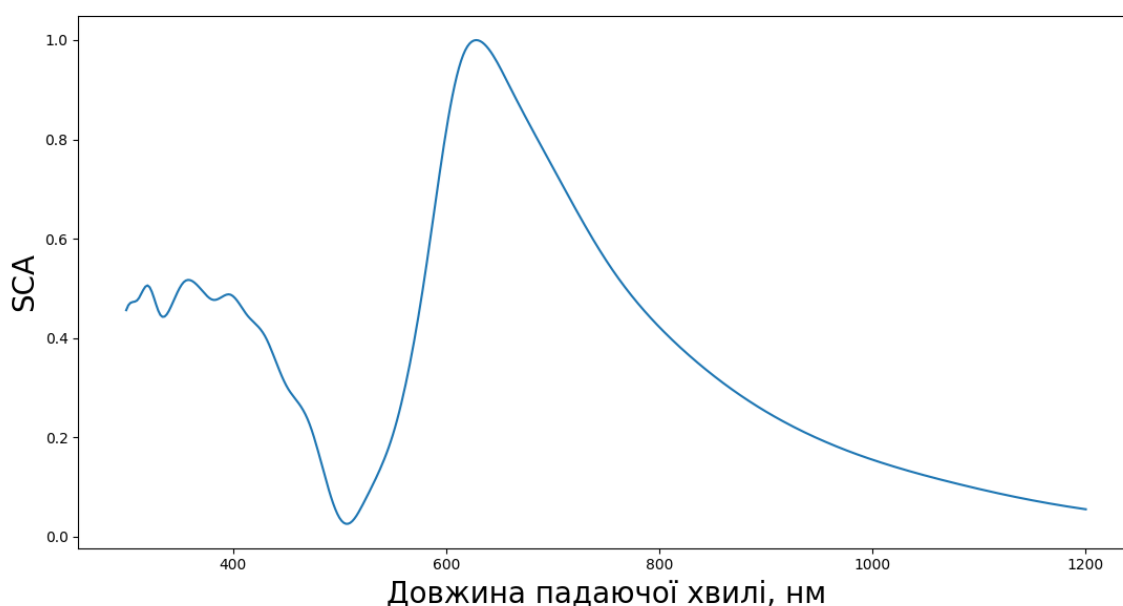


Рис. 3.1. Переріз розсіювання для дво-шарової частинки з Au та Cu, відповідні радіуси 100нм та 150нм

Для навчання нейромереж бралось 3 матеріали — Au, Cu та Ag. Самі дані генерувались наступним чином:

- 1) випадково генерувалось 2 радіуси для наносфери від 30нм до 200нм;
- 2) випадковим чином вибиралось 2 різних матеріали;
- 3) для згенерованої наночастинки вираховувався переріз розсіювання в межах від 300нм до 1200нм з кроком в 2нм.

Таким чином для двошарової наночастинки вдалось згенерувати приблизно 2300 даних.

Отримані дані випадковим чином розбивались на 2 вибірки: для тренування нейромереж (80%) та для її тестування(20%).

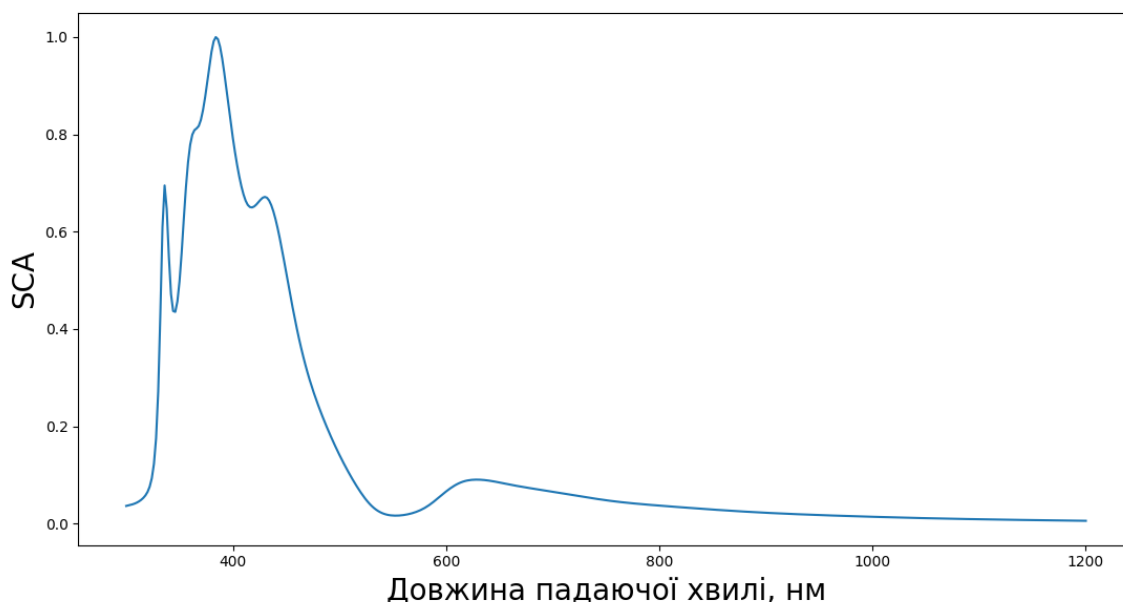


Рис. 3.2. Переріз розсіювання для тьох-шарової частинки з Au, Cu та Ag, відповідні радіуси 70нм, 110нм та 150нм

Дані для тьох-шарової наночастинки дані генерувались таким ж чином.

Також для двошарової наночастинки були згенеровані інші дані. В даному випадку радіуси для наночастинок брались в межах від 200нм до 250нм. Вони використовувались для перевірки готової моделі, для того щоб зрозуміти наскільки якісно модель змогла узагальнити результати.

3.2. Порівняння нейронних мереж для класифікації даних

Для двошарових наночастинок на згенерованих даних було натреновано 3 нейромережі. Усі нейромережі тренувались 100 епох. В процесі тренування за допомогою методу *keras.callbacks.ModelCheckpoint* зберігався найкращий варіант кожної з моделей, який і був використаний для тестування.

У табл. 3.1 приведено результати тестувань моделей для двох шарової наночастинки.

Також досить важливим моментом у тренування нейромереж є те наскільки модель змогла узагальнити отримані дані, а не просто їх запам'ятати. Результати тестування нейромереж на другому наборі даних приведені у табл. 3.2.

Результати тестування нейромереж для класифікації матеріалів для двох-шарової наносфери на першому наборі даних

	$accuracy_1$	$accuracy_2$
Dense	92.56%	91.96%
SimpleRNN	88.65%	91.25%
LSTM	94.56%	92.20%

Таблиця 3.2

Результати тестування нейромереж для двох-шарової наносфери на другому наборі даних

	$accuracy_1$	$accuracy_2$
Dense	82.11%	88.88%
SimpleRNN	83.48%	88.22%
LSTM	93.33%	91.24%

Як видно з таблиць, найкращий результат показала нейромережа з LSTM шаром.

У випадку трьох шарової сфери, нейромережі показали гірші результати чим для двошарової наночастинки. Це означає що для наночастинок у яких більш чим 2 шари виділити якусь залежність стає досить проблематично для дуже простих нейромереж. В таких випадках потрібно будувати більш складні архітектури.

Таблиця 3.3

Результати тестування нейромереж для трьох-шарової наносфери

	$accuracy_1$	$accuracy_2$	$accuracy_3$
Dense	77.69%	67.88%	66.61%
SimpleRNN	61.87%	50.32%	62.18%
LSTM	69.46%	61.39%	65.66%

3.3. Порівняння нейронних мереж для знаходження радіусів шарів наночастинок

Процес навчання даних нейромереж був таким ж самим як для класифікації матеріалів.

Результати тестування нейромереж для двошарової наночастинки приведені в табл. 3.4 для першого набору даних, в табл. 3.5 для другого набору даних.

Таблиця 3.4

Результати тестування нейромереж для двох-шарової наносфери на першому наборі даних

	mape
Dense	8.0657%
SimpleRNN	9.0348%
LSTM	5.8197%

Таблиця 3.5

Результати тестування нейромереж для двох-шарової наносфери на другому наборі даних

	mape
Dense	9.2769%
SimpleRNN	8.1074%
LSTM	6.3251%

Як видно з наведених таблиць, найкращі результати показала знову ж таки нейромережа з LSTM шаром.

3.4. Результати тренування нейромереж для знаходження діелектричної проникності

У даному випадку була взята тільки нейромережа з LSTM шаром.

Нейромережу тренували тільки на двох матеріалах Au та Cu. А для оцінки результату було використано Ag.

Для тестування було взято дві частинки з Au та Ag і відповідними радіусами шарів 100 та 150nm. Інша частинка була з Cu та Au.

На рис. 3.3 зображено результат роботи нейромережі для першої частинки. Як бачимо вона досить гарно визначила матеріали.

На рис. 3.4 зображено результат роботи нейромережі для другої наночастинки. Як видно з малюнку нейромережа не змогла визначити жодного матеріалу.

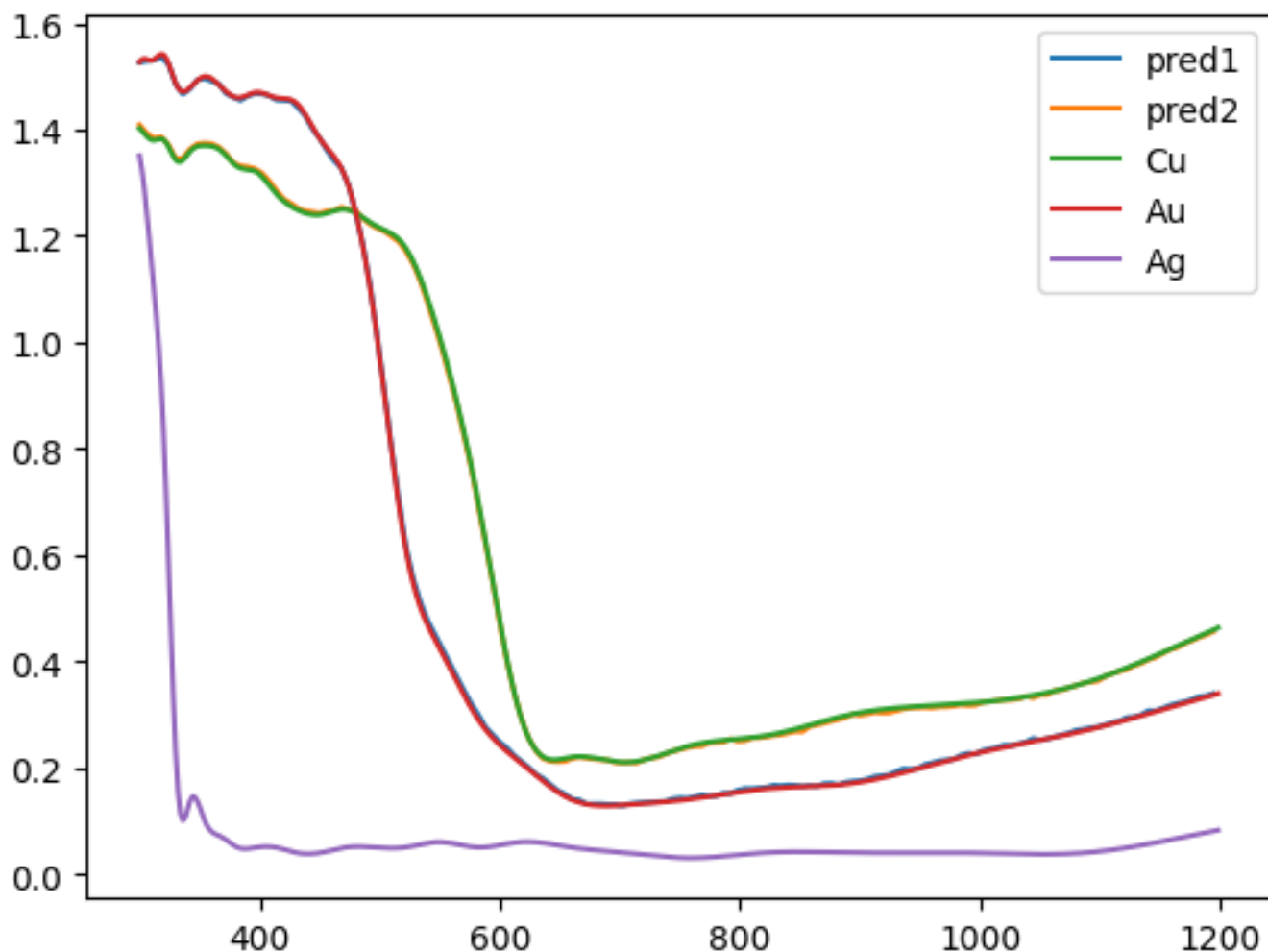


Рис. 3.3. Графік передбачення нейронної мережі для першої наночастинки

Даний результат свідчить про те що ми отримали нейромережу яка просто запам'ятовує результат. Це було очікувано оскільки ми взяли тільки два матеріали.

У такому випадку було б краще взяти більшу кількість матеріалів для навчання, а також збільшити нейромережу щоб вона змогла не просто запам'ятовувати вихідні результати, а знайти методи для узагальнення отриманих знань.

3.5. Висновки до третього розділу

В третьому розділі:

1. Змодельовано переріз розсіювання на багат шаровій наночастинці
2. Натреновано та порівняно три типи нейромереж для визначення матеріалу кожного шару та його радіуса для двошарових та тьох-шарових на-

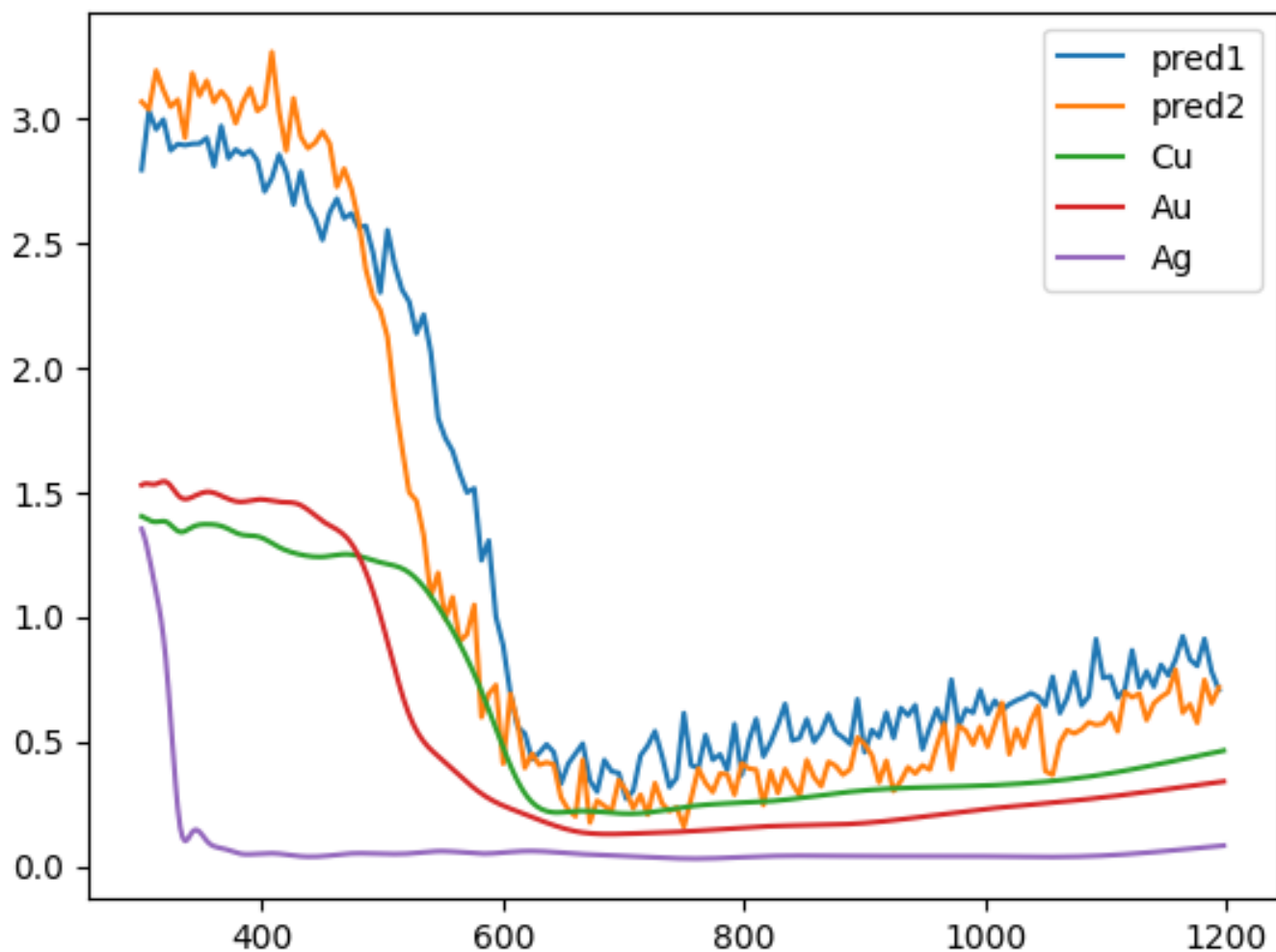


Рис. 3.4. Графік передбачення нейронної мережі для другої наночастинки наночастинок. В обох випадках найкращою виявилась нейромережа яка включає в себе LSTM архітектуру.

1. Одержано розв'язок рівняння Гельмгольца для задачі дифракції на сферичних багатошарових наночастинках.

2. Змодельовано переріз розсіювання на сферичних багатошарових наночастинках. Для навчання нейромереж у випадку двошарової наночастинки було згенеровано 2 набори даних. Перший набір містив 2300 згенерованих перерізів розсіювання частинок розміром від 30 до 200 нм. Другий набір даних містив 110 згенерованих перерізів розсіювання які використовувались для додаткової перевірки нейромережі.

3. Побудовано три нейромережі для знаходження матеріалів та радіусів шарів для двошарових та трьох-шарових наночастинок. Перша архітектура містила тільки повноз'язні шари. У двох наступних було використано рекурентну архітектуру SimpleRNN та LSTM відповідно.

4. Для кожної архітектури проведена оцінка точності визначення структури наночастинок.

Для першої нейронної мережі були отримані такі результати: похибка у визначенні радіусів $\approx 9\%$, точність для знаходження 1 шару $\approx 82\%$, другого — $\approx 89\%$.

Для другої нейромережі були отримані приблизно такі ж результати: похибка у визначенні радіусів $\approx 8\%$, точність для знаходження 1 шару $\approx 83\%$, другого — $\approx 88\%$.

Найкращою архітектурою виявилась нейромережа з використанням LSTM шару, яка змогла визначити радіуси шарів з похибкою $\approx 6\%$ та матеріал першого і другого шарів з точністю $\approx 94\%$ та $\approx 92\%$ відповідно.

1. A Review on Optimization Techniques of Antennas Using AI and ML / DL Algorithms / T. Gayatri [та ін.] // International Journal of Advances in Microwave Technology (IJAMT). — 2022. — Т. 7, № 2. — С. 288—295.
2. Artificial Intelligence for Photonics and Photonic Materials / D. Piccinotti [та ін.] // Reports on Progress in Physics. — 2021. — Т. 84. — С. 012401.
3. Bohren C., Huffman D. R. Absorption and Scattering of Light by Small Particles / за ред. C. Bohren, D. R. Huffman. — Wiley Science Paperback Series, 1998.
4. Buduma N., Locascio N. Deep Learning Basics: Creating Algorithms for Artificial Intelligence of the Next Generation. — O'Reilly Media, 2017.
5. De A., Kalita D. Bio-Fabricated Gold and Silver Nanoparticle Based Plasmonic Sensors for Detection of Environmental Pollutants: An Overview // Critical reviews in analytical chemistry. — 2021. — Бep. — Т. 53. — С. 1—17. — DOI: 10.1080/10408347.2021.1970507.
6. Deep learning in nano-photonics: inverse design and beyond / P. Wiecha [та ін.] // Photonic Research. — 2021. — Т. 9, № 5. — B182—B200.
7. Development of nanoparticle-based optical sensors for pathogenic bacterial detection / T. Mocan [та ін.] // Journal of Nanobiotechnology. — 2017. — Бep. — Т. 15. — DOI: 10.1186/s12951-017-0260-y.
8. Froemming N., Henkelman G. Optimizing core-shell nanoparticle catalysts with a genetic algorithm // Journal of Chemical Physics. — 2009. — Т. 131. — С. 234103.
9. Giles M., Pierce N. An introduction to the adjoint approach to design // Flow, Turbulence and Combustion. — 2000. — Т. 65. — С. 393—415.
10. Goodfellow I., Bengio Y., Courville A. Deep Learning. — MIT Press, 2016.
11. Intelligent on-demand design of phononic metamaterials / Y. Jin [та ін.] // Nanophotonics. — 2022. — Т. 11, № 3. — С. 439—460.

12. Johnson P. B., Christy R. W. Optical Constants of the Noble Metals // *Phys. Rev. B.* — 1972. — Груд. — Т. 6, вып. 12. — С. 4370—4379. — DOI: 10.1103/PhysRevB.6.4370.
13. Maier S. *Plasmonics: Fundamentals And Applications* //. — 01.2007. — С. 245.
14. Multimode hybrid gold-silicon nanoantennas for tailored nanoscale optical confinement / С. McPolin [та ін.] // *Nanophotonics.* — 2023. — Трав. — DOI: 10.1515/nanoph-2023-0105.
15. Nanophotonic particle simulation and inverse design using artificial neural networks / J. Peurifoy [та ін.] // *Science Advances.* — 2018. — Т. 4. — eaar4206.
16. Neural Networks Historical Review / D. Andina [та ін.] //. — 02.2007. — ISBN 978-0-387-37450-5. — DOI: 10.1007/0-387-37452-3_2.
17. Novotny L., Hecht B. *Principles of Nano-Optics.* — Cambridge University Press, 2006. — DOI: 10.1017/CB09780511813535.
18. Petryayeva E., Krull U. Localized Surface Plasmon Resonance: Nanostructures, bioassays and Biosensing—A Review // *Analytica chimica acta.* — 2011. — Листоп. — Т. 706. — С. 8—24. — DOI: 10.1016/j.aca.2011.08.020.
19. Plasmonic nano-lasers / Y. Yin [та ін.] // *Nano Energy.* — 2012. — Січ. — Т. 1. — С. 25—41. — DOI: 10.1016/j.nanoen.2011.09.002.
20. Plasmonic nanostructure design and characterization via Deep Learning / I. Malkiel [та ін.] // *Light: Science Applications.* — 2018. — Т. 7. — С. 60. — DOI: 10.1038/s41377-018-0060-7.
21. Silver Nanoparticles as Nano Antenna for TiO₂ Activation and its Application in DSSC for Enhanced Performance / E. Danladi [та ін.] // *Journal of Theoretical and Applied Physics.* — 2019. — Трав. — Т. 1. — С. 88—98.
22. Training Deep Neural Networks for the Inverse Design of Nanophotonic Structures / D. Liu [та ін.] // *ACS Photonics.* — 2018. — Т. 5. — С. 1365—1369.