

**Міністерство освіти і науки України**  
**Луцький національний технічний університет**



# **НЕЙРОННІ МЕРЕЖІ**

Методичні вказівки до виконання самостійної роботи  
для здобувачів  
першого (бакалаврського) рівня вищої освіти  
освітньої програми «Штучний інтелект та аналіз масивів даних»  
спеціальності Прикладна математика  
денної форми навчання

Луцьк 2025

УДК 004.032.26(076)

Н15

Електронна копія друкованого видання передана для внесення в репозитарій ЛНТУ

Директор бібліотеки \_\_\_\_\_ Н. Поліщук

Рекомендовано до видання вченою радою факультету архітектури, будівництва та дизайну ЛНТУ,

протокол № \_\_\_ від «\_\_\_» \_\_\_\_\_ 2025 року.

Голова вченої ради факультету ФАБД \_\_\_\_\_ О. Андрійчук

Розглянуто і схвалено на засіданні кафедри прикладної математики та механіки ЛНТУ,

протокол № \_\_\_ від «\_\_\_» \_\_\_\_\_ 2025 року.

Завідувачка кафедри прикладної математики та механіки \_\_\_\_\_ О. Мікуліч

Укладач: \_\_\_\_\_ О. Приходько, кандидат технічних наук, доцент кафедри прикладної математики та механіки ЛНТУ

Рецензент: \_\_\_\_\_ доктор технічних наук, професор О. Мікуліч

Відповідальний за випуск: \_\_\_\_\_ О. Мікуліч, доктор технічних наук, професор, завідувачка кафедри прикладної математики та механіки ЛНТУ

**Нейронні мережі:** методичні вказівки до виконання самостійної роботи для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Штучний інтелект та аналіз масивів даних» спеціальності «Прикладна математика» денної форми навчання / уклад. О. Приходько  
**Н 15** Луцьк: ЛНТУ, 2025. – 24 с.

Методичні вказівки містять розгорнуті рекомендації до виконання самостійної роботи з навчальної дисципліни «Нейронні мережі». Видання побудовано у форматі «дорожньої карти», що включає інструкції до теоретичного вивчення матеріалу, опис практичних та індивідуальних завдань, перелік контрольних запитань і рекомендовану літературу.

Призначено для студентів денної форми навчання спеціальності 113 «Прикладна математика» за освітньою програмою «Штучний інтелект та аналіз масивів даних»

## Зміст

ВСТУП .....	4
Тема 1. Основні теоретичні відомості про нейронні мережі.....	5
Тема 2. Класифікація нейронних мереж.....	7
Тема 3. Навчання нейронних мереж .....	9
Тема 4. Багатозв'язні нейронні мережі (MLP) .....	11
Тема 5. Згорткові нейронні мережі (CNN) .....	12
Тема 6. Рекурентні нейронні мережі (RNN).....	14
Тема 7. Автоенкодери та зменшення розмірності .....	16
Тема 8. Генеративні моделі.....	18
ПІСЛЯМОВА ТА АВТОРСЬКА ПРИМІТКА .....	21
Список рекомендованої літератури.....	22

## ВСТУП

Дисципліна «Нейронні мережі» займає центральне місце у підготовці фахівців за освітньою програмою «Штучний інтелект та аналіз масивів даних» (спеціальність 113 «Прикладна математика»).

Важливість цього предмету важко переоцінити. Це, по суті, остання фундаментальна дисципліна, яка фокусується саме на архітектурі та внутрішній механіці моделей машинного навчання перед переходом до дисциплін, орієнтованих на роботу з даними глобального масштабу (Big Data, Data Mining), що чекають на вас на старших курсах. Якщо Data Mining вчить, *де* шукати закономірності у великих масивах, то «Нейронні мережі» вчать створювати інструменти, *як* ці закономірності знаходити, класифікувати та навіть генерувати нову інформацію. Без глибокого розуміння принципів побудови нейромереж ви ризикуєте залишитись лише користувачем готових бібліотек, нездатним адаптувати архітектуру під специфічні прикладні задачі.

Ці методичні вказівки до самостійної роботи створені не як сухий перелік вимог, а як дорожня карта (roadmap) вашого професійного розвитку. Обсяг матеріалу в галузі Deep Learning зростає експоненційно, і студенту легко загубитися у морі статей та фреймворків. Мета цієї методички – структурувати ваш час та увагу, вказавши на критично важливі вузли (теми), без яких неможливе становлення фахівця. Тут ви не знайдете готових відповідей чи переписаних сторінок з підручників. Натомість тут сформульовано вектори пошуку, акценти для аналізу та практичні кейси, які допоможуть трансформувати теоретичні знання у практичні навички. Використовуйте цей посібник як навігатор: він підкаже, *що* шукати, *на що* звернути увагу при читанні рекомендованої літератури та *як* перевірити глибину власного розуміння.

## Тема 1. Основні теоретичні відомості про нейронні мережі

Вивчення даної теми слід розпочати з усвідомлення того факту, що штучні нейронні мережі (ШНМ) є спробою математичного моделювання принципів роботи людського мозку. Тому першим кроком у вашій самостійній роботі має стати опрацювання будови **біологічного нейрона**. Вам необхідно розібратися в призначенні його основних складових: тіла клітини (соми), дендритів (вхідних каналів), аксона (вихідного каналу) та синапсів (місць з'єднання). Зверніть увагу на електрохімічну природу передачі сигналу та поняття «порогу збудження», оскільки саме ці біологічні аспекти лягли в основу математичної моделі.

Наступним етапом є перехід від біології до математики. Ви повинні детально розглянути **формальну модель нейрона** (часто звану моделлю МакКаллока-Піттса або розширену модель Розенблатта). У процесі опрацювання літератури ви маєте чітко співставити біологічні елементи з їхніми математичними аналогами: дендрити відповідають вектору вхідних сигналів ( $x$ ), синаптична сила – ваговим коефіцієнтам ( $w$ ), тіло клітини – блоку суматора, а поріг збудження реалізується через функцію активації та зміщення ( $bias$ ).

Особливу увагу приділіть процесу перетворення інформації всередині штучного нейрона. Ви повинні вміти не лише записати формулу зваженої суми, але й пояснити фізичний зміст вагових коефіцієнтів як пам'яті нейрона. Важливо зрозуміти, що навчання нейронної мережі – це, по суті, процес налаштування саме цих ваг. Також обов'язково розгляньте роль функції активації, яка додає нелінійність у систему, дозволяючи мережі вирішувати складні задачі, недоступні для лінійних моделей.

Після розбору моделі нейрона перейдіть до **порівняльного аналізу**. Сформууйте для себе чітке уявлення про відмінності між живою та штучною системами. Зосередьтеся на таких аспектах, як швидкість поширення сигналу (у мозку вона повільніша, ніж у процесорах), принцип обробки даних (паралельний у мозку проти послідовного у класичних архітектурах фон Неймана), енергоефективність та здатність до самонавчання. У цьому ж контексті опрацюйте відмінності між поняттями «штучний інтелект» (Artificial

Intelligence) та «природний інтелект», звертаючи увагу на те, що сучасні ШНМ відносяться до так званого «слабкого ШІ» (Weak AI), який спеціалізується на вирішенні конкретних вузькопрофільних задач.

Завершальним теоретичним блоком є огляд **сучасних сфер застосування**. Не намагайтеся запам'ятати всі можливі галузі, натомість зосередьтеся на розумінні того, *які саме* типи задач найкраще вирішуються нейромережами (класифікація, регресія, кластеризація, генерація контенту). Ознайомтеся з прикладами використання ШНМ у медицині (діагностика), фінансах (скоринг, трейдинг), комп'ютерному зорі та обробці природної мови.

### **Вказівки до виконання індивідуальних завдань**

У межах цієї теми передбачено виконання двох завдань. По-перше, вам необхідно побудувати схему математичної моделі нейрона. Схема повинна бути детальною: з позначенням входів, ваг, блоку сумування, функції активації, порогового значення (зсуву) та виходу. Кожен елемент схеми супроводжуйте коротким текстовим поясненням його функції. По-друге, підготуйте короткий реферат (2–3 сторінки) щодо практичного застосування нейромереж. Рекомендується обрати одну конкретну галузь, яка вас цікавить (наприклад, агросектор, кібербезпека або ігрова індустрія), і навести конкретні кейси, уникаючи загальних фраз про «великі перспективи».

### **Контрольні запитання**

1. Які компоненти біологічного нейрона є прототипами входів, ваг та аксона штучного нейрона?
2. У чому полягає математична сутність процесу зваженого сумування сигналів?
3. Яку роль відіграє функція активації (передавальна функція) у моделі штучного нейрона?
4. Що таке зміщення (bias) і як воно впливає на результат роботи нейрона?

5. Назвіть фундаментальні відмінності в принципах обробки інформації біологічним мозком та традиційним комп'ютером.
6. Чому лінійна модель нейрона (без нелінійної функції активації) має обмежені можливості?
7. Які класи задач є найбільш придатними для вирішення за допомогою нейронних мереж, а де їх застосування є недоцільним?

**Рекомендована література:** [1, 6, 10, 11]

## **Тема 2. Класифікація нейронних мереж**

Робота над цією темою має сформувати у вас системне бачення "зоопарку" нейронних мереж. Не намагайтеся відразу вивчити математику кожного типу – ваше завдання на цьому етапі зрозуміти *логіку поділу*.

Розпочніть з класифікації за **типом навчання**. Ви повинні чітко розмежовувати навчання з учителем (Supervised Learning), де є розмічені дані, навчання без учителя (Unsupervised Learning), де мережа сама шукає закономірності, та навчання з підкріпленням (Reinforcement Learning). Для кожного типу знайдіть і зафіксуйте 2-3 типові задачі (наприклад, прогнозування ціни – це вчитель, а кластеризація клієнтів – без учителя).

Далі перейдіть до класифікації за **архітектурою зв'язків**. Це фундамент для розуміння подальших тем. Розберіть відмінність між мережами прямого поширення (Feedforward), де сигнал йде лише від входу до виходу, та рекурентними мережами (Recurrent), де є зворотні зв'язки.

Наступний крок – оглядове знайомство з конкретними архітектурами. Ваша мета – зрозуміти *спеціалізацію* кожної.

- **Одношаровий перцептрон:** розберіть, чому він історично важливий, але не може вирішити задачу XOR (проблема лінійної сепарабельності).
- **Багатошарові мережі (MLP):** зрозумійте, як додавання прихованих шарів вирішує проблеми перцептрона.

- **Згорткові мережі (CNN):** запам'ятайте, що це стандарт для роботи з зображеннями (інваріантність до зміщення).
- **Рекурентні мережі (RNN):** зафіксуйте їхню роль у роботі з послідовностями (текст, звук, часові ряди).
- **Автоенкодер та генеративні моделі:** зверніть увагу на їхню здатність стискати дані або створювати нові.

### **Вказівки до виконання індивідуальних завдань**

У практичній частині вам потрібно систематизувати здобуті знання. По-перше, складіть **порівняльну таблицю**. У рядках вкажіть типи мереж (Перцептрон, MLP, CNN, RNN, GAN тощо), а у стовпцях – критерії: тип вхідних даних (числа, картинка, текст), наявність вчителя, основна сфера застосування, переваги та недоліки. Це стане вашою "шпаргалкою" на майбутнє. По-друге, підготуйте презентацію на тему вибору архітектури. Не робіть її теоретичною. Уявіть, що ви – архітектор системи, якому принесли задачу (наприклад, розпізнавання облич на прохідній). Поясніть логіку: чому тут потрібна саме CNN, а не MLP чи RNN? Розгляньте 3-4 різні кейси.

### **Контрольні запитання**

1. У чому полягає принципова відмінність між навчанням з учителем та без учителя?
2. Чому одношаровий перцептрон не може вирішити задачу "виключне АБО" (XOR)?
3. Яка архітектурна особливість дозволяє рекурентним мережам "пам'ятати" попередні дані?
4. Для яких типів даних найдоцільніше використовувати згорткові нейронні мережі (CNN)?
5. Назвіть основну ідею автоенкодерів: що відбувається з даними на вході та виході?

6. Який тип нейронних мереж використовується для генерації фотореалістичних зображень?
7. Як класифікуються нейронні мережі за напрямком поширення сигналу?

**Рекомендована література:** [1, 3, 5, 6, 10]

### Тема 3. Навчання нейронних мереж

Це одна з найважливіших тем курсу, оскільки навіть ідеально спроектована архітектура не працюватиме без правильного навчання.

Почніть з опрацювання поняття **функції втрат (Loss Function)**. Ви маєте усвідомити, що навчання – це процес мінімізації помилки. Розберіться, які функції втрат використовуються для регресії (MSE), а які – для класифікації (Cross-Entropy).

Далі детально вивчіть **функції активації**. Не просто завчіть їхні назви, а зрозумійте їхні властивості.

- *Sigmoid / Tanh*: чому вони зараз використовуються рідше (проблема зникнення градієнта)?
- *ReLU (та її варіації Leaky ReLU, ELU)*: чому це стандарт де-факто для глибоких мереж?
- *Softmax*: чому вона ставиться лише на вихідному шарі задач класифікації?

Центральна частина теми – **методи оптимізації**. Основою є градієнтний спуск (Gradient Descent). Уявіть цей процес як спуск з гори в тумані. Вивчіть еволюцію оптимізаторів: від класичного SGD до Momentum (додає інерцію), AdaGrad (адаптує крок навчання) та сучасних RMSProp і Adam. Ви повинні розуміти, чому Adam є універсальним вибором для більшості задач. Окремо розберіть поняття **гіперпараметрів** (learning rate, batch size, кількість epoch) та їхній вплив на швидкість і якість навчання.

Завершіть тему вивченням проблеми **перенавчання (overfitting)**. Це ситуація, коли мережа "зазубрила" приклади, але не розуміє суті. Опрацюйте

методи боротьби з цим: Dropout (випадкове виключення нейронів), L1/L2 регуляризація, раннє припинення (Early Stopping).

### **Вказівки до виконання практичних завдань**

Перше завдання – аналітичне. Побудуйте графіки функцій активації (Sigmoid, Tanh, ReLU) та, що важливо, графіки їхніх похідних. Проаналізуйте діапазони значень. Зверніть увагу, де похідна стає близькою до нуля – це ключ до розуміння проблеми зникнення градієнта. Друге завдання – порівняльне. Вам потрібно дослідити роботу оптимізаторів Adam та RMSProp. Якщо маєте навички програмування (Python), запустіть просту модель на обох оптимізаторах і порівняйте графіки падіння помилки (Loss). Якщо ні – зробіть теоретичний огляд: в яких випадках один може переважати інший, як вони працюють з розрідженими даними та як швидко збігаються.

### **Контрольні запитання**

1. У чому полягає геометричний зміст градієнта функції помилки?
2. Чому функція ReLU краще підходить для глибоких мереж, ніж сигмоїда?
3. Яке призначення функції Softmax і чому сума її вихідних значень завжди дорівнює одиниці?
4. Поясніть сутність проблеми зникнення градієнта (Vanishing Gradient Problem).
5. Як параметр "швидкість навчання" (learning rate) впливає на процес збіжності алгоритму?
6. У чому полягає основна відмінність оптимізатора Adam від звичайного стохастичного градієнтного спуску?
7. Як метод регуляризації Dropout допомагає уникнути перенавчання мережі?

**Рекомендована література:** [2, 3, 10, 11, 13]

## Тема 4. Багатозв'язні нейронні мережі (MLP)

У цій темі ви маєте заглибитися в архітектуру **багат шарового перцептрона (Multilayer Perceptron – MLP)**, який є класичним прикладом повнозв'язної нейронної мережі. Ваше завдання – зрозуміти, як поєднання простих нейронів у шари створює систему, здатну апроксимувати будь-яку неперервну функцію (теорема цибенка).

Розпочніть з вивчення структури MLP: вхідний шар, приховані шари та вихідний шар. Ви повинні усвідомити роль **прихованих шарів (hidden layers)**. Уявіть їх як етапи переробки інформації, де кожен наступний шар формує все більш складні абстрактні ознаки на основі даних попереднього шару. Важливо навчитися розрізняти **параметри** мережі (ваги та зміщення, які мережа вивчає сама) та **гіперпараметри** (кількість шарів, кількість нейронів у шарі, які ви задаєте вручну перед початком навчання).

Центральним елементом вашої роботи над цією темою має стати детальне вивчення **алгоритму зворотного поширення помилки (Backpropagation)**. Це математичний «двигун» сучасного ШІ. Не лякайтеся формул, а спробуйте зрозуміти логіку процесу:

1. **Прямий прохід (Forward Pass):** дані йдуть від входу до виходу, мережа робить передбачення.
2. **Обчислення помилки:** порівняння передбачення з реальністю.
3. **Зворотний прохід (Backward Pass):** помилка «розподіляється» назад по мережі від виходу до входу. Тут ключовим є **ланцюгове правило (Chain Rule)** диференціювання. Ви маєте зрозуміти, що ми шукаємо часткову похідну помилки по кожній вазі, щоб знати, в який бік і наскільки цю вагу підкрутити.

### Вказівки до виконання практичних завдань

Перше завдання – графічне. Вам необхідно накреслити схему MLP для конкретної задачі класифікації (наприклад, розпізнавання рукописних цифр MNIST, де на вході 784 нейрони, а на виході – 10). Чітко покажіть зв'язки «кожен

з кожним» між сусідніми шарами, щоб відчуті масштаб кількості параметрів. Друге завдання – описове. Опишіть алгоритм Backpropagation своїми словами, не використовуючи складних формул, але розбивши процес на чіткі кроки. Уявіть, що ви пояснюєте це колезі: «Спочатку ми подаємо сигнал... потім рахуємо різницю... потім, використовуючи похідні, дізнаємося внесок кожного нейрона в помилку...». Це допоможе закріпити розуміння механізму навчання.

### **Контрольні запитання**

1. Чому додавання прихованих шарів дозволяє мережі вирішувати нелінійні задачі?
2. У чому різниця між параметрами моделі та гіперпараметрами? Наведіть приклади обох.
3. Поясніть сутність «ланцюгового правила» (Chain Rule) в контексті навчання нейромережі.
4. Що відбувається під час «прямого проходу» (Forward Pass)?
5. Чому для навчання багатошарових мереж не можна використовувати просте правило навчання перцептрона Розенблатта?
6. Як кількість нейронів у прихованому шарі впливає на здатність мережі до узагальнення (underfitting vs overfitting)?
7. Що таке епоха навчання і чим вона відрізняється від ітерації?

**Рекомендована література:** [1, 2, 3, 5, 10, 11]

## **Тема 5. Згорткові нейронні мережі (CNN)**

Ця тема переносить вас у світ комп'ютерного зору. Головне питання, на яке ви маєте знайти відповідь: *чому звичайні MLP погано працюють із зображеннями?* (Підказка: втрата просторової структури при витягуванні картинки в вектор та величезна кількість wag).

Основа вивчення – архітектура **CNN (Convolutional Neural Networks)**. Розберіть роботу **згорткового шару (Convolutional Layer)**. Ви повинні зрозуміти, що таке **ядро згортки (kernel/filter)** і як воно «ковзає» по зображенню, формуючи карти ознак (feature maps). Усвідомте, що, на відміну від MLP, ваги тут – це значення в самому ядрі, і вони є спільними для всього зображення (parameter sharing), що критично зменшує їх кількість.

Детально опрацюйте параметри згортки:

- **Stride (крок):** як швидко фільтр рухається (через 1 піксель чи через 2).
- **Padding (доповнення):** навіщо ми додаємо нулі по краях зображення (щоб зберегти розмірність).

Наступний важливий блок – **шари субдискретизації (Pooling)**. Розберіть різницю між Max Pooling (вибір найяскравішої ознаки) та Average Pooling (усереднення). Зрозумійте, як це допомагає боротися зі зміщенням об'єкта на картинці та зменшує обчислювальне навантаження.

Завершіть тему розумінням того, як CNN збирається в єдине ціле: серія блоків «Conv + ReLU + Pool» виділяє ознаки (від ліній до складних форм), а в кінці стоїть шар **Flatten** і класичний MLP для фінальної класифікації.

### **Вказівки до виконання практичних завдань**

Це завдання суто математичне і вкрай важливе для розуміння архітектури. Вам потрібно навчитися розраховувати розмірність вихідного тензора. Використовуйте формулу:  $O = \frac{I - K + 2P}{S} + 1$ , де I – розмір входу, K – розмір ядра, P – паддінг, S – страйд.

Прорахуйте кілька прикладів вручну. Також поясніть письмово різницю між Max Pooling та Average Pooling: у яких випадках важливо зберегти «максимальну реакцію» (наявність ознаки), а в яких – загальний фон?

### **Контрольні запитання**

1. Чому для обробки зображень використовують згорткові мережі, а не повнозв'язні?

2. Що таке «карта ознак» (feature map) і як вона утворюється?
3. Як працює механізм спільних ваг (parameter sharing) і яку перевагу він надає?
4. Поясніть призначення операції Padding (доповнення нулями).
5. Як впливає збільшення кроку згортки (Stride) на розмір вихідного зображення?
6. У чому полягає функція шару Pooling і чому Max Pooling використовується частіше за Average Pooling?
7. Яку роль відіграє шар Flatten перед фінальною класифікацією?

**Рекомендована література:** [3, 6, 8, 10, 12, 13]

## Тема 6. Рекурентні нейронні мережі (RNN)

При вивченні цієї теми ви повинні змінити парадигму мислення: якщо раніше ми працювали зі статичними даними (картинка, набір параметрів), то тепер переходимо до **послідовностей** (текст, аудіо, котирування акцій, відео). Головне питання, яке ви маєте вирішити: *як мережа може враховувати контекст минулого?*

Розпочніть з архітектури **простої RNN (Vanilla RNN)**. Зрозумійте ключову відмінність від MLP: наявність зворотного зв'язку, де вихід нейрона на поточному кроці подається йому ж на вхід на наступному кроці. Вивчіть поняття **прихованого стану (hidden state)** – це і є «пам'ять» мережі. Щоб зрозуміти, як це працює, вам потрібно опанувати процедуру **розгортання в часі (Unrolling)**: уявіть RNN не як цикл, а як ланцюжок однакових нейромереж, де кожна передає інформацію наступній.

Наступний критично важливий блок – розуміння проблем навчання. Оскільки при навчанні використовується алгоритм ВРТТ (Backpropagation Through Time), ви неминуче стикнетесь з проблемою **зникаючого (vanishing)** або **вибухового (exploding)** градієнта. Ви повинні чітко розуміти фізику цього

процесу: при багаторазовому множенні чисел, менших за одиницю (похідні активаційних функцій), градієнт прямує до нуля, і мережа «забуває» початок довгої фрази.

Це логічно підведе вас до вивчення вдосконалених архітектур – **LSTM (Long Short-Term Memory)** та **GRU (Gated Recurrent Unit)**. Не намагайтеся завчити формули напам'ять, головне – зрозуміти логіку **гейтів (вентилів)**.

В LSTM розберіть роль трьох елементів: *Forget Gate* (що забути), *Input Gate* (що запам'ятати) та *Output Gate* (що видати). Зверніть увагу на наявність окремої "клітинної пам'яті" (cell state), яка діє як швидкісна магістраль для градієнта.

В GRU подивіться, як архітектуру спростили, об'єднавши деякі гейти, і чому це часто працює не гірше, але швидше.

Завершіть тему ознайомленням з **двонаправленими мережами (Bidirectional RNN)**, які читають вхідні дані одночасно з початку і з кінця (корисно для перекладу текстів), та поняттям Deep RNN (багатошарові рекурентні мережі).

### **Вказівки до виконання практичних завдань**

Перше завдання – схематичне. Намалуйте часову розгортку (unrolled architecture) простої RNN для вхідної послідовності довжиною 3-4 кроки. Позначте вектори входу ( $x$ ), прихованого стану ( $h$ ) та виходу ( $y$ ), а також покажіть, що вагові коефіцієнти ( $W$ ) однакові на кожному кроці. Друге завдання – аналітичне. Порівняйте LSTM та GRU. Складіть короткий опис: у яких випадках краще обрати простішу GRU (менше даних, потрібна швидкість), а коли варто брати потужнішу LSTM.

### **Контрольні запитання**

1. Чому звичайні мережі прямого поширення (Feedforward) непридатні для аналізу зв'язного тексту?
2. Що таке прихований стан (hidden state) і як він оновлюється на кожному кроці часу?

3. Поясніть сутність алгоритму Backpropagation Through Time (BPTT).
4. Чому виникає проблема зникнення градієнта в довгих послідовностях?
5. Яку функцію виконує «вентиль забування» (forget gate) в архітектурі LSTM?
6. У чому полягає основна архітектурна відмінність GRU від LSTM?
7. Навіщо використовуються двонаправлені (Bidirectional) RNN?

**Рекомендована література:** [3, 4, 6, 10, 12]

## Тема 7. Автоенкодері та зменшення розмірності

Тут ми переходимо до навчання без учителя (Unsupervised Learning). Головна ідея, яку ви маєте засвоїти: мережа може навчатися сама на собі, якщо поставити їй завдання: *«відтвори на виході те, що отримала на вході»*.

Почніть з вивчення базової архітектури **автоенкодера**. Вона складається з двох частин: **Енкодер**, який стискає вхідні дані, та **Декодер**, який їх відновлює. Ключовий елемент – це **пляшкове горло (Bottleneck)** або латентний простір. Ви повинні зрозуміти: якщо шар посередині менший за вхідний шар, мережа змушена відкидати шум і вчитися кодувати лише найважливіші ознаки. Це і є нелінійний аналог методу головних компонент (PCA).

Далі розгляньте різні модифікації автоенкодерів, щоб зрозуміти їхню практичну користь:

- **Denoising Autoencoder (DAE):** вивчіть принцип його навчання – на вхід подається «зашумлена» картинка, а порівнюється з «чистим» оригіналом. Це вчить мережу ігнорувати шум.
- **Sparse Autoencoder:** зрозумійте, як додавання обмежень на активність нейронів змушує мережу вивчати унікальні ознаки.
- **Variational Autoencoder (VAE):** це місток до наступної теми. Тут важливо усвідомити, що VAE кодує вхід не в фіксовані числа, а в *розподіл*

*ймовірностей* (середнє значення та дисперсію), що дозволяє генерувати нові плавні варіації даних.

Окрему увагу приділіть застосуванню автоенкодерів для **виявлення аномалій**. Логіка проста: якщо мережа навчилася добре відновлювати «нормальні» дані, то на «аномальних» даних вона зробить велику помилку відновлення (Reconstruction Error). Ця помилка і є індикатором аномалії.

### **Вказівки до виконання практичних завдань**

Перше завдання – пояснити принцип компресії. Опишіть, чому стиснення даних за допомогою автоенкодера є "втратним" (lossy compression) і від чого залежить якість відновлення. Друге завдання – прикладне. Наведіть конкретний алгоритм дій для виявлення банківського шахрайства за допомогою автоенкодера: на яких даних навчаємо (тільки легальні транзакції), що подаємо на вхід під час експлуатації і як інтерпретуємо вихідну помилку (якщо помилка > порогу, то це підозра на шахрайство).

### **Контрольні запитання**

1. Яка функція втрат (Loss Function) зазвичай використовується для навчання автоенкодерів?
2. Чому розмір прихованого шару (bottleneck) має бути меншим за розмір вхідного шару?
3. Чим автоенкодер відрізняється від архітектур для класифікації (наприклад, CNN)?
4. Як працює шумопоглинаючий автоенкодер (Denoising Autoencoder)?
5. Поясніть принцип використання автоенкодерів для виявлення аномалій.
6. У чому принципова різниця між звичайним автоенкодером та варіаційним (VAE)?
7. Чи може автоенкодер ідеально відтворити вхідні дані? Чому?

**Рекомендована література:** [3, 5, 10, 11, 12]

## Тема 8. Генеративні моделі

Вивчення цієї теми вимагає від вас розуміння фундаментальної різниці між двома підходами в машинному навчанні. Раніше ви вивчали **дискримінативні моделі**, мета яких – провести межу між класами (відрізнити кота від собаки). Тепер ви переходите до **генеративних моделей**, мета яких – зрозуміти, як виглядає розподіл даних, щоб створити *нового* кота, якого ніколи не існувало.

Центральне місце у вашій роботі має посісти архітектура **GAN (Generative Adversarial Networks)**. Уявіть її як гру або змагання двох нейромереж:

1. **Генератор (Generator):** його задача – взяти на вхід випадковий шумовий вектор і перетворити його на зображення, яке максимально схоже на реальне. Це «фальшивомонетник».
2. **Дискримінатор (Discriminator):** його задача – отримати зображення і сказати, чи воно справжнє (з реального датасету), чи підроблене (від Генератора). Це «поліція».

Ваше завдання – розібратися в механіці їхнього спільного навчання. Це так звана **мінімаксна гра (Minimax Game)**: Генератор намагається максимізувати помилку Дискримінатора, а Дискримінатор намагається її мінімізувати. Зверніть увагу на складність цього процесу: якщо одна мережа стане значно сильнішою за іншу занадто швидко, навчання зупиниться.

Окремо опрацюйте типові проблеми навчання GAN, зокрема **Mode Collapse**. Це ситуація, коли Генератор знаходить одне зображення, яке добре обманює Дискримінатор, і починає генерувати тільки його (наприклад, малює лише один тип взуття, ігноруючи інші).

Також проведіть порівняння GAN з **Варіаційними автоенкодерами (VAE)**, які ви розглядали в попередній темі. Запам'ятайте головну відмінність: VAE дають більш розмиті, але плавні зображення і мають чітку математичну базу, тоді як GAN дають дуже чіткі, фотореалістичні картинки, але є нестабільними в навчанні.

На завершення обов'язково розгляньте **етичні аспекти**. Технологія Deepfake, породжена генеративними мережами, створює серйозні виклики для інформаційної безпеки. Ви повинні розуміти не тільки як створити фейк, але і яку відповідальність це накладає на розробника.

### **Вказівки до виконання індивідуальних завдань**

Перше завдання – структурне. Побудуйте детальну схему GAN. На ній має бути чітко видно потоки даних:

- Вхід Генератора (вектор шуму  $z$ ).
- Вихід Генератора (фейкове зображення  $G(z)$ ).
- Вхід Дискримінатора (мікс реальних зображень  $x$  та фейкових  $G(z)$ ).
- Вихід Дискримінатора (ймовірність: реальне/фейк).

*Важливо:* покажіть стрілками, як проходить помилка (Backpropagation). Помилка Дискримінатора оновлює і його ваги, і ваги Генератора (через градієнт), тоді як Генератор не має доступу до реальних даних, а вчиться лише на "критиці" Дискримінатора.

Друге завдання – дослідницьке (доповідь). Підготуйте матеріал на тему етичних ризиків. Не обмежуйтеся загальними фразами. Розгляньте конкретні загрози: підробка голосу для шахрайства, використання діпфейків у політиці, проблема авторського права на згенеровані твори (кому належить картина – нейромережі чи автору коду?).

### **Контрольні запитання**

1. У чому полягає принципова відмінність між дискримінативними та генеративними моделями?
2. Яка роль випадкового шуму (noise vector) на вході Генератора?
3. Чому процес навчання GAN називають «грою з нульовою сумою» або мінімаксною грою?
4. Поясніть сутність явища «Mode Collapse» (колапс мод). Чому це погано?

5. Чому зображення, згенеровані VAE, часто виглядають більш розмитими порівняно з GAN?
6. Яким чином Дискримінатор допомагає навчатися Генератору, якщо Генератор ніколи не бачить реальних даних?
7. Які основні соціальні та етичні загрози несе безконтрольне використання технології Deepfake?

**Рекомендована література:** [3, 10, 11]

## ПІСЛЯМОВА ТА АВТОРСЬКА ПРИМІТКА

На завершення варто зазначити важливий нюанс щодо створення цього навчального посібника. Дана методична розробка є живим прикладом синергії людського досвіду та новітніх технологій штучного інтелекту.

Текст методичних вказівок був згенерований великою мовною моделлю (LLM), після чого пройшов етап верифікації, редагування та авторської вичитки викладачем. Такий підхід було обрано невипадково. Це наочна демонстрація можливостей тих самих генеративних нейронних мереж (зокрема архітектури Transformer та механізмів уваги), вивченню яких присвячено останні теми цього курсу.

Той факт, що нейромережа здатна структурувати складний технічний матеріал, виділяти головне та формулювати педагогічно виважені завдання, найкраще підтверджує актуальність спеціальності, яку ви здобуваєте. Ви вивчаєте технології, які вже сьогодні здатні асистувати у створенні освітнього контенту, написанні коду та вирішенні креативних задач. Сподіваємось, це стане для вас додатковою мотивацією розібратися у тому, як саме працює цей "магічний чорний ящик", щоб у майбутньому не лише користуватися подібними системами, а й створювати нові, ще досконаліші архітектури.

## Список рекомендованої літератури

1. Нейронні мережі для обробки інформації : архітектури, навчання, застосування : монографія / О. Г. Руденко, О. О. Безсонов, С. П. Євсєєв, О. Б. Ахієзер, С. О. Руденко. – Львів : «Новий Світ-2000», 2025. – 680 с. Принципи штучних нейронних мереж та їх застосування: навч. посіб. /
2. Павло Тимощук, Михайло Лобур; М-во освіти і науки України, Нац. ун-т «Львів. політехніка». – Львів : Вид-во Львів. політехніки, 2020. – 292 с. Python: алгоритмізація та програмування : навчальний посібник / В. А. Висоцька, О. В. Оборська. – 2-ге видання, стереотипне. – Львів : «Новий Світ-2000», 2025. – 514 с. ISBN 978-617-7519-74-3
3. Литвин В. В. Глибинне навчання: навч. посіб. / В. В. Литвин, Р. М. Пелешак, В. А. Висоцька; М-во освіти і науки України, Нац. ун-т «Львів. політехніка». – Львів : Вид-во Львів. політехніки, 2021. – 264 с.: іл
4. Навчання з підкріпленням в автономних інтелектуальних системах: навч. посібник / О. Ю. Бочкар'єв. – Львів: Видавець Марченко Т.В., 2024. – 125с
5. Машинне навчання : навчальний посібник / Т. М. Басюк, В. В. Литвин, Л. М. Захарія, Н. Е. Кунанець ; за науковою редакцією д.т.н., проф., В. В. Пасічника. – 3-тє видання, стереотипне. – Львів : «Новий Світ – 2000», 2025. – 330 с. ISBN 978-617-7519-42-2
6. Штучний інтелект. Нейромережева обробка інформації : архітектури, навчання, застосування : навчальний посібник у 2-х ч. : Ч. 1 / О. Г. Руденко, О. О. Безсонов, С. П. Євсєєв, О. Б. Ахієзер, Ю. І. Зайцев ; за заг. ред. С. П. Євсєєва. – Харків : НТУ «ХПІ», – Львів : «Новий Світ-2000», 2025. – 426 с.
7. Булгакова О. С. Методи та системи штучного інтелекту: теорія та практика : навч. посібник / О. С. Булгакова, В. В. Зосімов, В. О. Поздєєв. – Херсон : Олді-плюс, 2020. – 356 с. – ISBN 978-966-289-364-9
8. Системи штучного інтелекту. Навчальний посібник / Н. Б. Шаховська, Р. М. Камінський, О. Б. Вовк. Львів : Видавництво Львівської політехніки, 2018. 392 с.

9. Prykhodko O.S. Using a neural network to increase the course stability of UAVs in the absence of GNSS / O.S. Prykhodko, V.M. Matviychuk // Наукові нотатки ЛНТУ - 2024. – №66. - С.283-286
10. Deep Learning Ian Goodfellow, Yoshua Bengio, Aaron Courville. Електронна книга: <https://www.deeplearningbook.org/>
11. Jentzen, A., Kuckuck, B., & von Wurstemberger, P. (2023). Mathematical Introduction to Deep Learning: Methods, Implementations, and Theory. Електронний ресурс: <https://arxiv.org/abs/2310.20360>. Альтернативне посилання для вільного завантаження: <https://opentechbook.com/book/mathematical-introduction-to-deep-learning/>
12. Dive into Deep Learning. Interactive deep learning book with code, math, and discussions. Електронний ресурс: <https://d2l.ai/>
13. Petersen, P., & Zech, J. (2024). Mathematical theory of deep learning. Електронний ресурс: <https://arxiv.org/abs/2407.18384>

**Нейронні мережі:** методичні вказівки до виконання самостійної роботи для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Штучний інтелект та аналіз масивів даних» спеціальності «Прикладна математика» денної форми навчання / уклад. О. Приходько Луцьк: ЛНТУ, 2025. – 24 с.

Комп'ютерний набір  
Редактор

О. Приходько  
О. Приходько

Підп. до друку « \_\_\_\_ » \_\_\_\_\_ 2025 р. Формат 60x84/16. Папір офс.  
Гарн. Таймс. Ум. друк. арк. 2,25.  
Тираж 50 прим.

Інформаційно-видавничий відділ  
Луцького національного технічного університету  
43018, м. Луцьк, вул. Львівська, 75  
Друк – ІВВ ЛНТУ