

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»**

**МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних занять з дисципліни
«Методи штучного інтелекту»
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальностями
113 – Прикладна математика
124 – Системний аналіз**

Затверджено
на засіданні кафедри ПМІТ
Протокол № 1 від 30.08.2024 р.

Одеса: Одеська політехніка, 2025

Методичні рекомендації до лабораторних занять з дисципліни «Методи штучного інтелекту» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 124 – Системний аналіз (освітньо-професійна програма: «Системний аналіз, управління і аналітика» – 2022) та за спеціальністю 113 – Прикладна математика (освітньо-професійна програма: «Математичне забезпечення комп’ютерних систем» – 2022)/ Укл.: *М. В. Полякова*. – Одеса: Одеська політехніка, 2025. – 37 с.

Укладач: **Полякова М. В.**, докт. техн. наук, доц.

Методичні вказівки містять теоретичні відомості, які необхідні і достатні для роботи на лабораторних заняттях з дисципліни «Методи штучного інтелекту».

Призначаються для студентів денної форми навчання.

ЗМІСТ

Вступ	4
Лабораторне заняття 1. Застосування згорткової нейронної мережі для класифікації рукописних цифр	5
Лабораторне заняття 2. Класифікація зображень обличь, що містять певні емоції, за допомогою багатопланового перцептрона	8
Лабораторне заняття 3. Порівняння результатів класифікації зображень обличь, що містять певні емоції	13
Лабораторне заняття 4. Прогнозування часових рядів із використанням рекурентної нейронної мережі	17
Лабораторне заняття 5. Передавальне навчання	20
Лабораторне заняття 6. Семантична сегментація зображень за допомогою згорткової нейронної мережі типу "енкодер-декодер"	24
Рекомендована література	28

Вступ

Дослідження та осмислення штучного інтелекту є актуальним напрямком вітчизняної та зарубіжної науки. Системи штучного інтелекту активно впроваджуються у різні сфери суспільного життя, удосконалюють технологічні процеси та дозволяють вирішувати складні інтелектуальні завдання. Визначення методів та систем штучного інтелекту дозволяє більш ґрунтовно поглянути на сутнісні характеристики вказаного наукового напрямку та сформулювати концептуальні моделі і прийоми щодо реалізації отриманих знань на практиці.

Метою лабораторних занять з дисципліни «Методи штучного інтелекту» є опанування студентами сукупності теоретичних, методичних питань і практичного досвіду з основ нейронних мереж на основі використання сучасних програмних засобів..

Лабораторне заняття 1

Застосування згорткової нейронної мережі для класифікації рукописних цифр

Мета заняття: розпізнавання рукописних цифр за допомогою нейронних мереж із застосуванням Dropout для підвищення точності та стійкості моделей.

Зміст заняття: побудова програми класифікації рукописних цифр за допомогою згорткової нейронної мережі на Python та набуття практичних навичок використання Dropout.

Теоретичні основи

Dropout - це техніка регуляризації, яка випадково "вимикає" частину нейронів під час тренування, що допомагає уникнути перенавчання мережі. Для розпізнавання рукописних цифр зазвичай використовують згорткові нейронні мережі, які добре виявляють просторові залежності та ієрархічні структури в зображеннях. Dropout додається у шари нейронної мережі (часто у повнозв'язні шари після згорткових), щоб під час тренування випадково ігнорувати частину нейронів, що зменшує залежність від окремих ознак і покращує узагальнення моделі. Застосування dropout дозволяє підвищити точність розпізнавання, зменшуючи ризик перенавчання, особливо на невеликих або середніх за розміром датасетах, таких як MNIST.

У якості прикладу застосування слід зазначити, що після побудови згорткових нейронних мереж для розпізнавання рукописних цифр (наприклад, на базі MNIST) було продемонстровано, що використання dropout у поєднанні зі згортковими шарами дає точність близько 96-97%. Архітектура мережі включає кілька згорткових шарів із активацією, шари підвибірки (pooling), а потім кілька повнозв'язних шарів із застосуванням dropout для регуляризації. Dropout зазвичай налаштовують із ймовірністю "вимкнення" нейронів від 0.2 до 0.5, залежно від розміру мережі та обсягу даних.

Перевагами використання Dropout у задачі розпізнавання рукописних цифр є зменшення перенавчання та підвищення узагальнюючої здатності моделі; покращення стабільності та надійності класифікації; можливість тренування глибших мереж без значного ризику втрати узагальнення.

Під час виконання роботи рекомендується використовувати бібліотеки Python, такі як TensorFlow або Keras, які мають вбудовану підтримку dropout56. Для початку можна взяти просту згорткову нейронну мережу з кількома шарами Conv2D, MaxPooling, Dense, після чого додати dropout-шари після повнозв'язних шарів. Необхідно враховувати, що Dropout випадково вимикає частину нейронів під час навчання, що допомагає уникнути перенавчання і покращує узагальнення моделі.

Завдання для виконання роботи

1. Реалізувати нейронну мережу для класифікації рукописних цифр на основі набору даних MNIST. Набір MNIST є стандартним для таких завдань, містить 60 000 зображень для навчання і 10 000 для тестування рукописних цифр.
2. Впровадити механізм Dropout у мережу для запобігання перенавчанню та підвищення узагальнюючої здатності моделі.
3. Навчити мережу на тренувальному наборі даних, контролюючи точність і втрати під час навчання.
4. Провести тестування моделі на тестовому наборі даних для оцінки якості розпізнавання.
5. Візуалізувати результати роботи мережі: приклади розпізнаних цифр, ймовірності класифікації.

6. Проаналізувати вплив параметра Dropout на продуктивність моделі (наприклад, порівняти результати з різними значеннями dropout).
7. Описати архітектуру мережі (кількість шарів, типи шарів, функції активації).

Контрольні питання

1. Що таке dropout у нейронних мережах і яку роль він відіграє у навчанні моделей?
2. Як dropout допомагає уникнути перенавчання (overfitting) при розпізнаванні рукописних цифр?
3. Які параметри навчання нейронної мережі можна змінювати для покращення якості розпізнавання?
4. Чому важливо перемішувати (shuffle) вхідні дані під час навчання?
5. Які метрики використовуються для оцінки якості розпізнавання рукописних цифр?
6. Як працює механізм випадкового виключення нейронів (dropout) під час тестування моделі?

Лабораторне заняття 2

Класифікація зображень обличь, що містять певні емоції, за допомогою багатошарового перцептрона

Мета заняття: розв'язання задачі класифікації зображень обличь, що містять певні емоції, за допомогою багатошарового перцептрона

Зміст заняття: вивчення функцій бібліотеки scikit-learn для класифікації зображень обличь, що містять певні емоції, за допомогою багатошарового перцептрона на Python та набуття практичних навичок його використання.

Теоретичні основи

Дані, що пропонуються для розв'язання задачі складаються з 48x48 піксельних рисунків у градаціях сірого. Кожний файл містить центровані зображення обличь, що виражають певні емоції. Завдання полягає в класифікації кожного обличчя на основі емоцій до однієї із семи категорій (0 = злість, 1 = огида, 2 = страх, 3 = щастя, 4 = сум, 5 = здивування, 6 = нейтральний вираз).

Файл train.csv містить два стовпці, "емоція" та "пікселі". Стовпець "емоція" містить числовий код від 0 до 6 включно для емоції, яка присутня на зображенні. Стовпець "пікселі" містить рядок пікселів для кожного зображення, які містять безпосередньо обличчя. Вміст цього рядка — розділені пробілами значення пікселів у зростаючому порядку рядків. Файл test.csv містить лише стовпець "пікселі", завдання моделі класифікації — спрогнозувати стовпець емоцій. Навчальний набір складається з 28709 прикладів.

Завдання для виконання роботи

1. Отримати дані для виконання роботи за посиланням <https://www.kaggle.com/datasets/aadityasinghal/facial-expression-dataset>. Навести опис даних у звіті.
2. Спроектувати нейронну мережу багатошаровий перцептрон для класифікації зображень обличь за емоціями. Приклад наведено за посиланням <https://www.kaggle.com/avk256/face-emotion-classification-with-mlp>. Навести архітектуру мережі у звіті.
3. Розбити вибірку на навчальну та тестову у співвідношенні 0.8, 0.2 та 0.9, 0.1. Виконати навчання нейронної мережі на навчальних даних.
4. Використовуючи функції confusion_matrix та score, оцінити якість класифікації на навчальних даних. Обґрунтувати отримані результати.
5. Виконати узагальнення (валідацію) нейронної мережі на тестових даних.

6. Використовуючи функції `confusion_matrix` та `score`, оцінити якість класифікації на тестових даних в залежності від кількості прихованих шарів нейронної мережі. Обґрунтувати отримані результати.

Контрольні питання

1. Як створити множину зображень обличч для класифікації?
2. Як спроектувати нейронну мережу багатошаровий перцептрон для класифікації зображень обличч?
3. Як обрати кількість шарів та кількість нейронів у шарах для багатошарового перцептрона?
4. Як інтерпретувати рішення задачі класифікації зображень обличч?

Лабораторне заняття №4. Прогнозування часових рядів із використанням рекурентної нейронної мережі.

Мета заняття: розв'язання задачі прогнозування часового ряду за допомогою нейронної мережі LSTM.

Лабораторне заняття 3

Порівняння результатів класифікації зображень обличч, що містять певні емоції

Мета заняття: вивчення функцій бібліотек Keras, PyTorch та TensorFlow для класифікації зображень обличч, що містять певні емоції.

Зміст заняття: класифікація зображень обличч, що містять певні емоції, за допомогою згорткової нейронної мережі на Python та набуття практичних навичок їх використання.

Теоретичні основи

Згорткова нейронна мережа (CNN) - спеціальна архітектура штучних нейронних мереж, запропонована Яном Лекуном в 1988 році і націлена на ефективне розпізнавання образів, входить до складу технологій глибокого навчання. Використовує деякі особливості зорової кори, в якій були відкриті так звані прості клітини, що реагують на прямі лінії під різними кутами, і складні клітини, реакція яких пов'язана з активацією певного набору простих клітин. Таким чином, ідея згорткових нейронних мереж полягає в чергуванні згорткових шарів і шарів підвибірки. Структура мережі - односпрямована (без зворотних зв'язків), принципово багатошарова. Для навчання використовуються стандартні методи, найчастіше метод зворотного поширення помилки. Функція активації нейронів (передавальна функція) - будь-яка, за вибором дослідника.

Назву архітектура мережі отримала через наявність операції згортки, суть якої в тому, що кожен фрагмент зображення множиться на матрицю (ядро) згортки поелементно, а результат підсумовується і записується в аналогічну позицію вихідного зображення.

В згортковій нейронній мережі в операції згортки використовується лише обмежена матриця ваг невеликого розміру, яку «рухають» по всьому оброблюваному шару (на самому початку - безпосередньо по вхідному зображенню), формуючи після кожного зсуву сигнал активації для нейрона наступного шару з аналогічною позицією. Тобто для різних нейронів вихідного шару використовуються одна і та ж матриця ваг, яку також називають ядром згортки. Її інтерпретують як графічне кодування якої-небудь ознаки, наприклад, наявність похилої лінії під певним кутом. Тоді наступний шар, що вийшов в результаті операції згортки з такою матрицею ваг, показує наявність даної ознаки в оброблюваному шарі і її координати, формуючи так звану карту ознак (англ. feature map). Природно, в згортковій нейронній мережі набір ваг не один, а ціла гама, що кодує елементи зображення (наприклад лінії і дуги під різними кутами). При цьому такі ядра згортки не закладаються дослідником заздалегідь, а

формується самостійно шляхом навчання мережі класичним методом зворотного поширення помилки. Прохід кожним набором ваг формує свій власний примірник карти ознак, роблячи нейронну мережу багатоканальною (багато незалежних карт ознак на одному шарі). Також слід зазначити, що при переборі шару матрицею ваг її пересувають зазвичай не на повний крок (розмір цієї матриці), а на невелику відстань. Так, наприклад, при розмірності матриці ваг 5×5 її зрушують на один або два нейрона (пікселя) замість п'яти, щоб не «переступити» шукану ознаку.

Операція підвибірки (англ. pooling), виконує зменшення розмірності сформованих карт ознак. У даній архітектурі мережі вважається, що інформація про факт наявності шуканої ознаки важливіше точного знання його координат, тому з кількох сусідніх нейронів карти ознак вибирається максимальний і приймається за один нейрон ущільненої карти ознак меншої розмірності. За рахунок цієї операції, крім прискорення подальших обчислень, мережа стає більш інваріантною до масштабу вхідного зображення.

Після початкового шару (вхідного зображення) сигнал проходить серію згорткових шарів, в яких чергується власне згортка і підвибірка. Чергування шарів дозволяє складати «карти ознак» з карт ознак, на кожному наступному шарі карта зменшується в розмірі, але збільшується кількість каналів. На практиці це означає здатність розпізнавання складних ієрархій ознак. Зазвичай після проходження декількох шарів карта ознак вироджується в вектор або навіть скаляр, але таких карт ознак стають сотні. На виході згорткових шарів мережі встановлюють кілька шарів повнозв'язної нейронної мережі, на вхід яких подаються кінцеві карти ознак.

Завдання для виконання роботи

1. Отримати дані для виконання роботи за посиланням <https://www.kaggle.com/datasets/aadityasinghal/facial-expression-dataset>.
2. Спроекувати згорткову нейронну мережу для класифікації зображень обличчя за емоціями. Приклад наведено за посиланням <https://www.kaggle.com/avk256/face-emotion-classification-with-mlp>. Навести архітектуру мережі у звіті.
3. Розбити вибірку на навчальну та тестову у співвідношенні 0.8, 0.2 та 0.9, 0.1. Виконати навчання нейронної мережі на навчальних даних.
4. Використовуючи функції `confusion_matrix` та `score`, оцінити якість класифікації на навчальних даних. Обґрунтувати отримані результати.
5. Виконати узагальнення (валідацію) нейронної мережі на тестових даних.
6. Використовуючи функції `confusion_matrix` та `score`, оцінити якість класифікації на тестових даних. Обґрунтувати отримані результати.
7. Порівняти якість класифікації зображень обличчя, що отримано за допомогою багатшарового перцептрона у попередній роботі з якістю класифікації тих самих зображень за допомогою згорткової нейронної мережі.

Контрольні питання

1. Що уявляє собою згорткова нейронна мережа?
2. Як спроекувати згорткову нейронну мережу для класифікації зображень обличчя?
3. Які типи шарів містить згорткова нейронна мережа для класифікації зображень обличчя?
4. Як отримати ознаки зображень за допомогою згорткової нейронної мережі?

Лабораторне заняття 4

Прогнозування часових рядів із використанням рекурентної нейронної мережі

Мета заняття: створіння рекурентної нейронної мережі на Python і набуття практичних навичок її використання для аналізу настроїв.

Зміст заняття: розв'язання задачі прогнозування часового ряду за допомогою нейронної мережі LSTM.

Теоретичні основи

Послідовність значень ціни актива за рівновіддалені проміжки часу може розглядатися як часовий ряд. Часовий ряд – це сукупність значень якогось показника за кілька послідовних моментів часу. Позначимо числовий ряд як $R(t) = \{R(1), R(2), \dots, R(T)\}$, де $R(1), R(2), \dots, R(T)$ – значення числового ряду, які доступні в дискретні моменти часу $t=1, 2, \dots, T$; T – останній момент часу, для якого відоме значення часового ряду. Потрібно визначити значення $R(T), R(T+1), \dots, R(T+P)$ часового ряду в моменти часу $T, T+1, \dots, T+P$, де P – час попередження. Таким чином, якщо часовий ряд $R(t)$ складається з T значень ціни актива, то завдання прогнозування полягає в екстраполяції часового ряду на моменти часу $T+1, T+2, \dots, T+P$. Для обчислення значень часового ряду в майбутні моменти часу необхідно знайти функціональну залежність між минулими та майбутніми значеннями часового ряду шляхом мінімізації середньоквадратичного відхилення $\varepsilon(t)^2$ істинного значення від прогнозованого.

Рекурентні нейронні мережі можуть навчатися в послідовному режимі і враховувати дані, що знову надійшли на обробку. Це дозволяє підвищити якість прогнозування значень ціни актива.

Визначення архітектури нейронної мережі

Для прогнозування значень ціни актива можна використовувати рекурентну нейронну мережу LSTM-10, що складається з 10 послідовних блоків LSTM, що приймає на вхід 10 значень часового ряду і видає наступне 11-те значення як прогноз. Архітектура кожного блоку LSTM складається з чотирьох взаємодіючих шарів (рис. 5.1).

Формування навчальних множин

Для навчання нейронної мережі необхідно сформувати з масиву даних навчальні приклади, кожен із яких містить вхід і вихід, який мережа повинна навчитися видавати у відповідь даний вхід.

Сукупність навчальних прикладів створюється з масиву даних з інформацією про значення у попередні моменти часу за допомогою методу ковзного вікна. Для формування навчального прикладу на масив вихідних даних накладається часове вікно. Значення, які попадають у це вікно, формують навчальний приклад. Наступний приклад формується шляхом зсуву вікна на один часовий інтервал уперед. Сформовані приклади поділяються на дві підмножини: навчальну та тестову. Перша підмножина використовується для безпосереднього навчання мережі, друга - для навчання не використовується і застосовується для оцінки можливості мережі працювати з незнайомими даними. Зазвичай кількість прикладів навчальної множини відноситься до кількості прикладів тестової множини як 2:1 або 3:1.

Завдання для виконання роботи

1. Побудувати базову лінію, використовуючи дані про курс валют та акцій із сайту <https://www.investing.com/currencies/> згідно з Вашим індивідуальним варіантом (табл. 5.1). Базова лінія має містити щонайменше 100 спостережень.

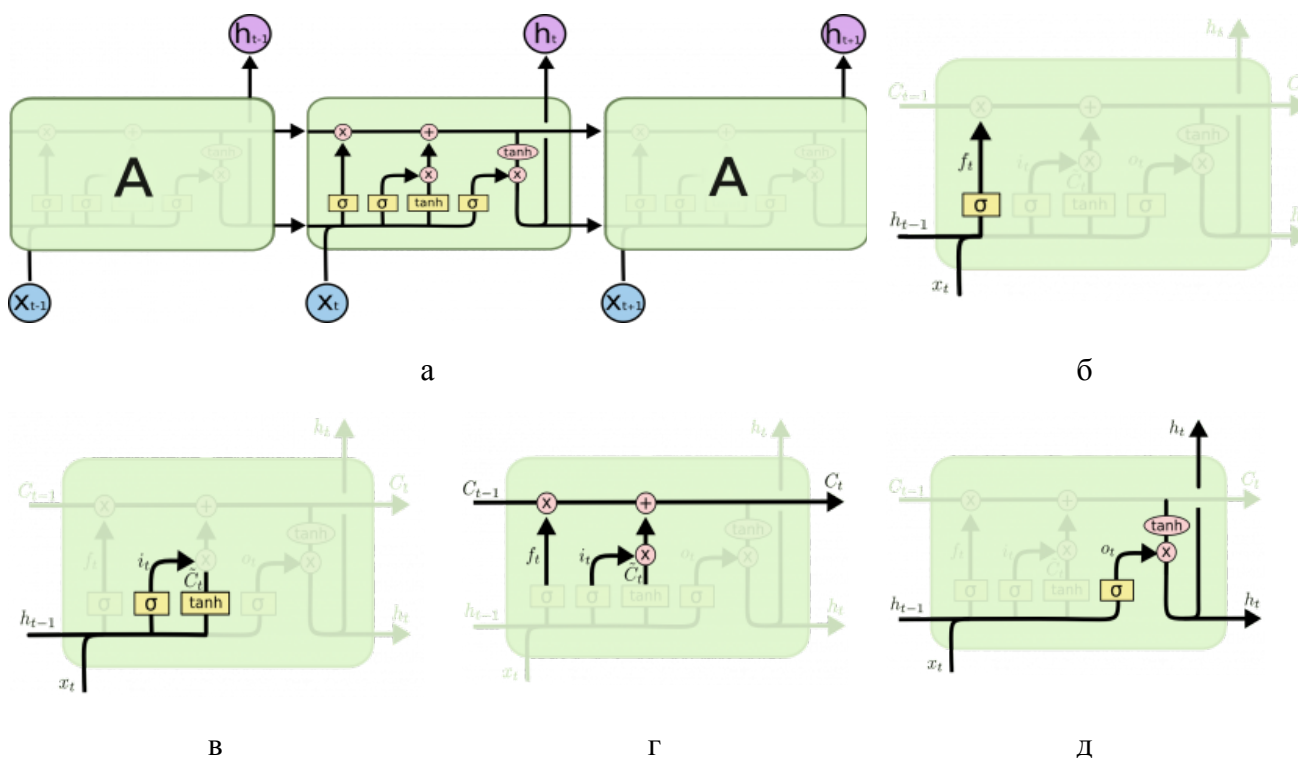


Рисунок 5.1 - Послідовні блоки мережі LSTM (а), шар забування (б), шар збереження (в), обчислення нового стану (г), вихідний шар (д).

Таблиця 4.1 - Варіанти завдань

№ варіанта	Посилання на сайт
1	https://www.investing.com/currencies/gbp-usd-historical-data
2	https://www.investing.com/currencies/eur-usd-historical-data
3	https://www.investing.com/currencies/usd-jpy-historical-data
4	https://www.investing.com/currencies/usd-chf-historical-data
5	https://www.investing.com/currencies/aud-usd-historical-data
6	https://www.investing.com/currencies/usd-cad-historical-data
7	https://www.investing.com/currencies/nzd-usd-historical-data
8	https://www.investing.com/currencies/usd-zar-historical-data
9	https://www.investing.com/currencies/usd-try-historical-data
10	https://www.investing.com/currencies/usd-mxn-historical-data
11	https://www.investing.com/currencies/usd-pln-historical-data
12	https://www.investing.com/currencies/usd-sek-historical-data
13	https://www.investing.com/currencies/usd-sgd-historical-data
14	https://www.investing.com/currencies/usd-dkk-historical-data
15	https://www.investing.com/currencies/usd-nok-historical-data
16	https://www.investing.com/currencies/usd-ils-historical-data
17	https://www.investing.com/currencies/usd-huf-historical-data
18	https://www.investing.com/currencies/usd-czk-historical-data
19	https://www.investing.com/currencies/usd-thb-historical-data

2. Побудувати навчальну та тестову вибірки. Для цього часовий ряд розбити на 2 частини: дані, що використовуються для навчання мережі, та дані для перевірки.

3. Побудувати нейронну мережу LSTM для прогнозування.
4. Провести навчання мережі на навчальній множині, навести графік вихідного ряду та спрогнозованого, обчислити середньоквадратичну помилку прогнозування.
5. Провести перевірку роботи мережі на тестовій вибірці, навести графік вхідного ряду та спрогнозованого, обчислити середньоквадратичну помилку прогнозування.

Контрольні питання

1. Що таке часовий ряд?
2. За допомогою якої нейронної мережі прогнозується часовий ряд?
3. Як сформувати навчальну вибірку?
4. Як сформувати тестову вибірку?

Лабораторне заняття 5

Передавальне навчання

Мета заняття: створіння рекурентної нейронної мережі на Python і набуття практичних навичок її використання для аналізу настроїв.

Зміст заняття: аналіз настроїв за висловлюваннями рекурентною нейронною мережею.

Теоретичні основи

Передавальне навчання (англ. Transfer Learning) – це технологія навчання нейронних мереж, яка дозволяє використовувати попередньо навчені мережі для розв’язання нових задач. Наприклад, поширені архітектури, що використовуються у комп’ютерному зорі та попередньо навчені на наборі даних ImageNet, можна додатково навчити для розпізнавання нових типів зображень.

Фреймворк Keras містить такі стандартні архітектури (рис. 5.1).

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572

Рисунок 5.1 - Стандартні архітектури нейромереж Keras

Процес передавання навчання складається з таких етапів.

1. Видалення верхніх Dense шарів.
2. Фіксація ваг шарів, що залишились.
3. Додавання нових Dense шарів з потрібною кількістю нейронів на виході останнього шару.

4. Навчання на новому наборі даних.

На рис. 5.2 зображено процес підключення деяких стандартних архітектур, видалення верхніх шарів, які відповідають за класифікацію та фіксацію ваг шарів, що залишились.

```
model_xception = Xception(weights='imagenet', include_top=False, input_shape=input_shape)
model_vgg19 = VGG19(weights='imagenet', include_top=False, input_shape=input_shape)
model_resnet = ResNet50(weights='imagenet', include_top=False, input_shape=input_shape)
model_mobilenet = MobileNet(weights='imagenet', include_top=False, input_shape=input_shape)
model_xception.trainable = False
model_vgg19.trainable = False
model_resnet.trainable = False
model_mobilenet.trainable = False
model_vgg19.summary()
```

Рисунок 5.2 - Видалення та фіксація шарів стандартних архітектур нейромереж

Процес додавання Dense шарів для навчання на нових даних зображено на рис. 5.3.

```
flat1 = model_vgg19.layers[-1].output
flat1 = GlobalAveragePooling2D()(flat1)
class1 = layers.Dense(1024, activation='relu')(flat1)
output = layers.Dense(1, activation='sigmoid')(class1)
model = Model(inputs=model_vgg19.inputs, outputs=output)
model.summary()
```

Рисунок 5.3 - Додавання нових шарів для додаткового навчання

Завдання для виконання роботи

1. Отримати дані для виконання роботи за посиланням Dogs vs. Cats | Kaggle.
2. Спроекувати згорткову нейронну мережу для класифікації зображень. Приклад наведено за посиланням Cats vs. dogs | Augmentation | Transfer learning. Навести архітектуру мережі у звіті.
3. Виконати навчання нейронної мережі на навчальних даних. Використовуючи функції `confusion_matrix` та `score`, оцінити якість класифікації на навчальних даних. Обґрунтувати отримані результати.
4. Виконати узагальнення (валідацію) нейронної мережі на тестових даних. Використовуючи функції `confusion_matrix` та `score`, оцінити якість класифікації на тестових даних. Обґрунтувати отримані результати.

Контрольні питання

1. Що уявляє собою передавальне навчання?
2. З яких етапів складається процес передавання навчання?
3. Які архітектури нейронних мереж з бібліотеки Keras можуть бути використані для передавання навчання під час класифікації зображень?
4. Як додати нові шари у мережу для додаткового навчання ?

Лабораторне заняття 6

Семантична сегментація зображень за допомогою згорткової нейронної мережі типу "енкодер-декодер"

Мета заняття: побудова згорткової нейронної мережі типу "енкодер-декодер" на Python та набуття практичних навичок її використання.

Зміст заняття: вивчення функцій бібліотеки Keras та реалізація згорткової нейронної мережі типу "енкодер-декодер" засобами фреймворку Keras.

Теоретичні основи

Семантична сегментація — це процес визначення кожного пікселя на зображенні належним до певного класу. Вона не лише визначає клас об'єктів на зображенні, але і точно визначає їх межі на піксельному рівні.

Згорткові нейронні мережі (ЗНМ) відіграють ключову роль у семантичній сегментації. Вони використовуються для визначення ознак на зображенні та допомагають в ідентифікації та сегментації об'єктів. Згорткові шари у ЗНМ ефективно виявляють локальні та глобальні ознаки на зображенні, що допомагає у точному визначенні об'єктів та їх меж.

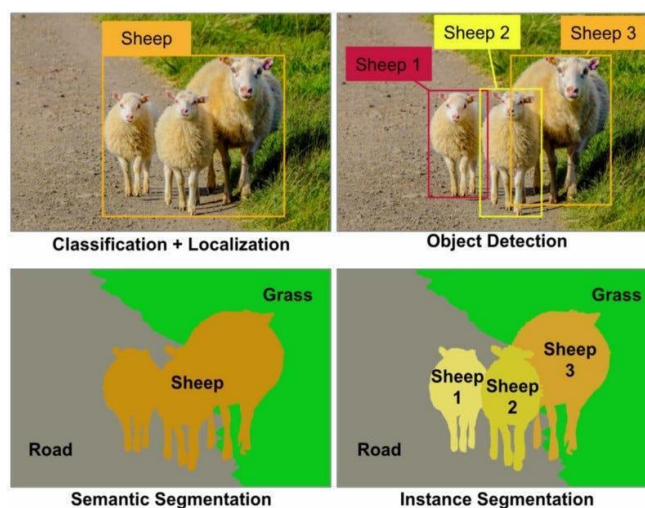


Рисунок 6.1 - Приклад того, як відрізняються між собою задачі сегментації від задач класифікації та виявлення об'єктів.

Як приклади для розв'язання цього завдання розглянемо кілька популярних архітектур, а саме U-Net та Fully Convolutional Network (FCN).

Архітектура U-Net складається з частин «стискання» (encoder) і «розширення» (decoder), що формують форму U. Encoder використовується для визначення ознак зображення, тоді як decoder використовує ці ознаки для побудови сегментованого зображення, тобто передбачає клас для кожного пікселя на зображенні. Особливістю U-Net є наявність «мостів» між encoder і decoder, що дозволяє передавати контекстну інформацію в decoder.

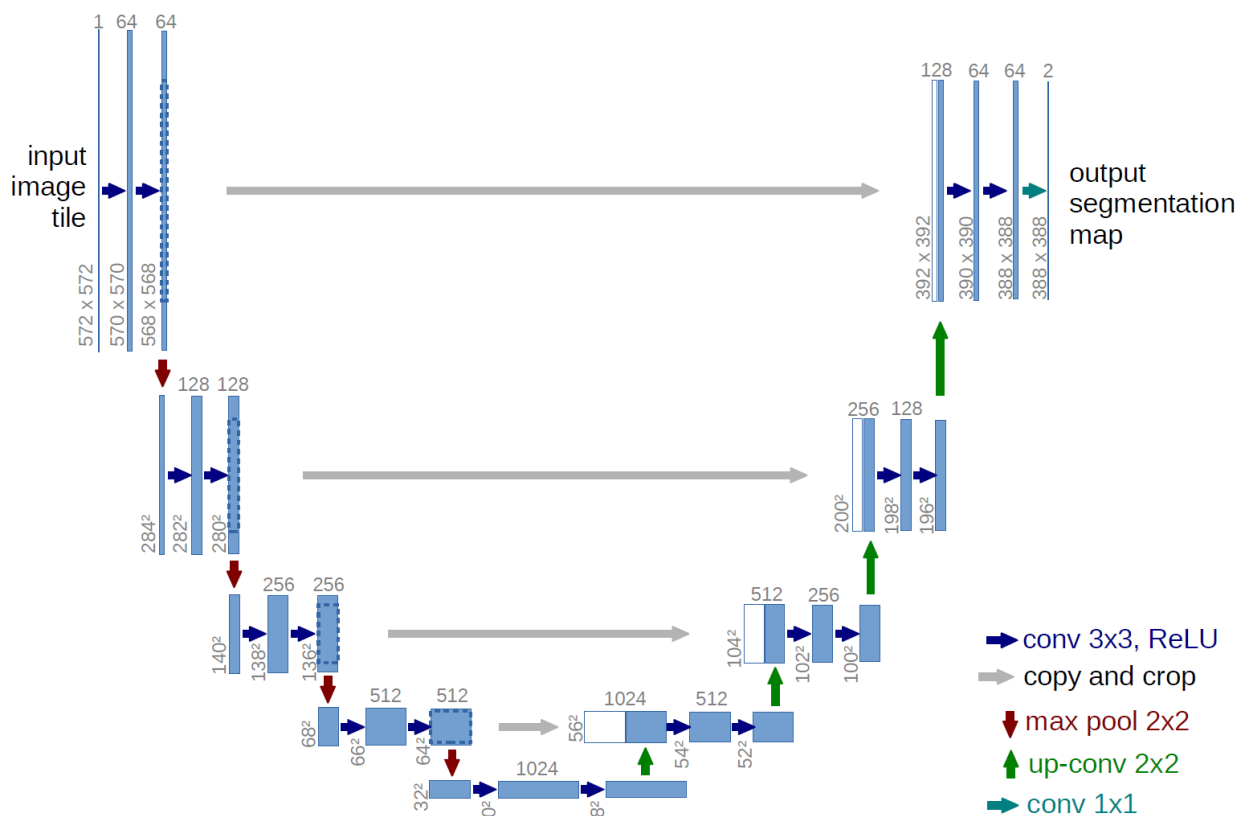


Рисунок 6.2 - Згорткові шари в U-Net передають інформацію не тільки для всіх наступних шарів, але й для тих шарів, які знаходяться на протилежному «кінці» моделі. Тобто перший шар передає інформацію останньому, другий — передостанньому, тощо.

На відміну від традиційних ЗНМ, які використовують повністю з'єднані шари для класифікації, FCN перетворює ці шари в згорткові, що дозволяє обробляти зображення будь-якого розміру і виводити семантичну мапу, яка відповідає розміру вхідного зображення.

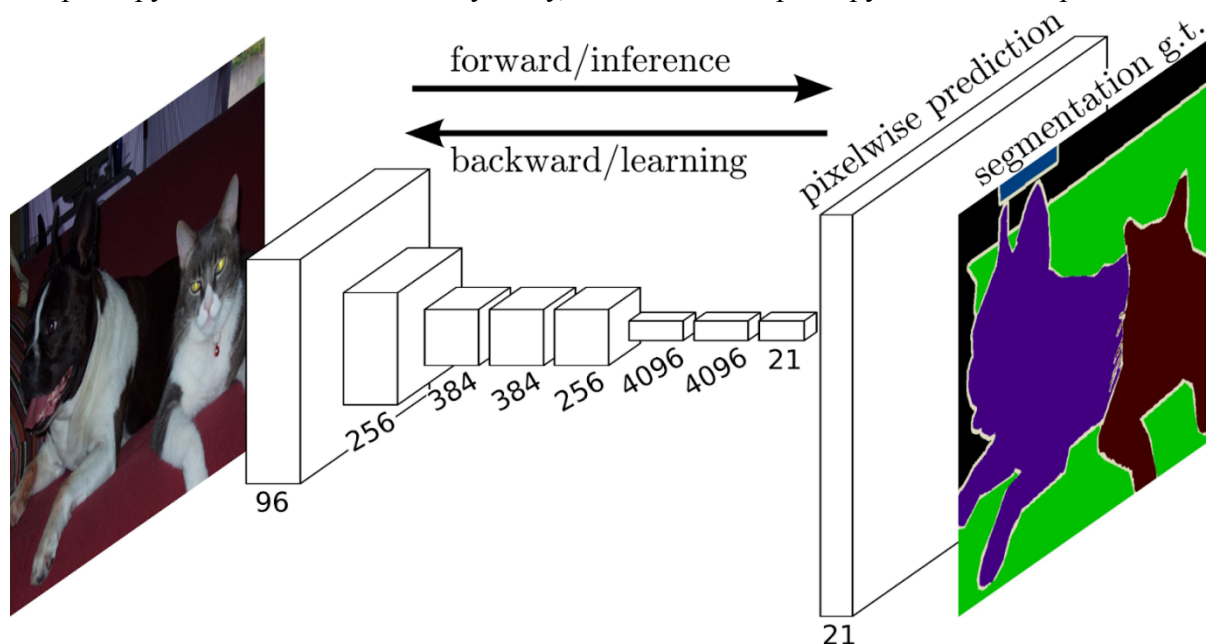


Рис. 6.3. FCN в ролі передостаннього шару використовує фільтри, що за розміром зівставні з розміром оригінального зображення.

Завдання для виконання роботи

1. Отримати дані для виконання роботи за посиланням <https://www.robots.ox.ac.uk/~vgg/data/pets/>.
2. Спроекувати згорткову нейронну мережу для класифікації зображень. Приклад наведено за посиланням <https://www.kaggle.com/code/ammarnassanalhajali/image-segmentation-with-a-u-net-and-kerashttps://www.robots.ox.ac.uk/~vgg/data/pets/>. Навести архітектуру мережі у звіті.
3. Виконати навчання нейронної мережі на навчальних даних. Навести графіки залежності функції втрат для навчальної вибірки та валідаційної множини. Обґрунтувати отримані результати.
4. Виконати узагальнення нейронної мережі на тестових даних. Навести графіки залежності функції втрат для тестової вибірки та валідаційної множини. Обґрунтувати отримані результати.
5. Навести приклади початкових та сегментованих зображень.

Контрольні питання

1. Що уявляє собою семантична сегментація зображень?
2. З яких етапів складається процес кодування-декодування зображень?
3. Які архітектури нейронних мереж з бібліотеки Keras можуть бути використані для семантичної сегментації зображень?
4. Як під час вирішення практичних завдань оцінити якість сегментації зображень?

Рекомендована література

1. Субботін С. О. , Нейронні мережі : теорія та практика: навч. посіб. [Текст] / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с. ISBN 978-966-995-189-2
2. Основи нейромережних технологій. Вибрані розділи: Класифікаційні властивості персептронів : навч. посіб./ П.О. Яганов. – Київ : КПІ ім. Ігоря Сікорського, 2021. – 44 с.
3. Технології нейронних мереж і нечіткого моделювання в системах управління : підручник / [авт. кол.: Г. І. Канюк, Б. І. Кузнецов, Т. Ю. Василець та ін.]. — Харків : Мадрид, 2020. — 306 с.
4. Ткаченко Р. О. Нейромережеві засоби штучного інтелекту : навч. посіб. / Р. О. Ткаченко, П. Р. Ткаченко, І. В. Ізонін ; Нац. ун-т "Львівська політехніка". – Львів : Вид-во Львівської політехніки, 2017. – 206 с.
5. Литвин В.В., Висоцька В.А., Пелещак Р.М. Глибинне навчання.. – Львів : Вид-во Львівської політехніки, 2021. – 264 с.
6. Троцько В.В. Методи штучного інтелекту: навчально-методичний і практичний посібник. – Київ: Університет економіки та права «КРОК», 2020 – 86 с.

Навчальне видання

Методичні рекомендації до лабораторних занять з дисципліни «Методи штучного інтелекту» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 124 – Системний аналіз та за спеціальністю 113 – Прикладна математика

Укладач: Марина Вячеславівна Полякова