

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Адаптивні активаційні функції глибинних нейронних мереж для сегментації
тривимірних медичних зображень
(тема)

Виконав:
студент 2 курсу, групи СШМ-20-2
Скорік В.А.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник проф. Бодянський Є.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
(код і повна назва)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Скорік Вікторії Анатоліївни
(прізвище, ім'я, по батькові)

1. Тема роботи Адаптивні активаційні функції глибинних нейронних мереж для сегментації тривимірних медичних зображень

затверджена наказом університету від 24 березня 20 22 р. № 414Ст

2. Термін подання студентом роботи до екзаменаційної комісії 11 травня 20 22 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та наукових проектів щодо розробки та дослідження сегментації медичних зображень, методи глибокого навчання для вирішення задач комп'ютерного зору в медицині, адаптивних активаційних функцій, набір даних BraTS20

4. Перелік питань, що потрібно опрацювати в роботі Аналіз предметної галузі та постановка задачі, традиційні методи сегментації медичних зображень, методи глибокого навчання для багатовимірних даних, дослідження функцій активації та їх адаптивних варіантів, вибір метрики для оцінки моделей, вибір технологій для програмної реалізації, реалізація, проведення експериментів, аналіз результатів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1.1 – Приклад відмінності анатомічних структур, Рисунок 1.2 – Графік бінарної ступінчастої функції, Рисунок 1.3 – Графік лінійної функції, Рисунок 1.4 – Ілюстрація простої нейронної мережі з трьома шарами та одним нейроном у кожному шарі, Рисунок 1.5 – Графік сигмоїди та її похідної, Рисунок 1.6 – Графік Tanh та його похідної, Рисунок 1.7 – Графік ReLU та його похідної, Рисунок 1.8 – Графік LReLU та його похідної, Рисунок 1.9 – Графік PReLU (a=0.3) та його похідної, Рисунок 1.9 – Графік PReLU (a=0.3) та його похідної, Рисунок 1.11 – Графік Softplus та його похідної, Рисунок 1.12 – Графік SoftMax, Рисунок 1.13 – зворотний хід для двошарової нейронної мережі з одним нейроном у кожному шарі, Рисунок 2.1 – Ядро 2D Convolution (ліворуч) та його вихід (праворуч), Рисунок 2.2 – Набір ядер 2D Convolution, Рисунок 2.3 – Набір ядер 3D Convolution, Рисунок 2.4 – приклад архітектури 3D U-Net

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу | |
|----------------------|--|---|------|
| | | підпис | дата |
| | | | |
| | | | |

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|----|--|---------------------------------|----------|
| 1 | Аналіз предметної галузі | 17.01.22 – 01.02.22 | Виконано |
| 2 | Вивчення проблеми, що потребує рішення | 01.02.22 – 08.02.22 | Виконано |
| 3 | Формування задачі дослідження | 08.02.22 – 22.02.22 | Виконано |
| 4 | Аналіз підходів, що існують | 22.02.22 – 09.02.22 | Виконано |
| 5 | Концептуальне проектування методу рішення задачі | 09.02.22 – 03.03.22 | Виконано |
| 6 | Вибір та підготовка вхідних даних | 03.03.22 – 10.03.22 | Виконано |
| 7 | Вибір архітектури | 10.03.22 – 12.03.22 | Виконано |
| 8 | Реалізація програмного коду | 12.03.22 – 23.03.22 | Виконано |
| 9 | Навчання нейронних мереж | 23.03.22 – 01.04.22 | Виконано |
| 10 | Оформлення пояснювальної записки | 01.04.22 – 25.04.22 | Виконано |
| | | | |
| | | | |

Дата видачі завдання 28 березня 20 22 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис) _____
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 95 с., 44 рис., 37 формул, 2 дод., 20 джерел.

АДАПТИВНА, БАГАТОВИМІРНІ МЕДИЧНІ ЗОБРАЖЕННЯ, ГЛИБИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, СЕМАНТИЧНА СЕГМЕНТАЦІЯ, ШТУЧНА НЕЙРОННА МЕРЕЖА, ШТУЧНИЙ ІНТЕЛЕКТ, 3D ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, ELU, PReLU, RELU, SOFTPLUS

Об'єкт дослідження – застосування адаптивних ELU та PReLU зі згортковою нейронною мережею для семантичної сегментації багатовимірних медичних зображень.

Предмет дослідження – адаптивні активаційні функції з тривимірною архітектурою U-Net.

Мета роботи – програмна реалізація AdELU, AdPReLU та вбудова їх в U-Net з тривимірними згортковими шарами для вирішення завдання семантичної сегментації на тривимірних медичних даних. Порівняння ефективності застосування адаптивних активаційних функцій для медичних даних в порівнянні зі статичними активаційними функціями.

Методи дослідження – аналіз активаційних функцій, що існують, вивчення предметної галузі, вирішення практичних завдань і проведення порівняльного аналізу.

Програмно реалізовані U-Net, EdELU та AdPReLU. Проведено дослідний аналіз набору даних, який складається з мультиінституціональних передопераційних МРТ-сканів і фокусується на сегментації внутрішньо неоднорідних пухлин мозку – BraTS20. Навчені нейронні мережі та проведено порівняльний аналіз.

ABSTRACT

Explanatory note: 95 pp., 44 fig., 37 formuls, 2 annex., 20 sources.

ADAPTIVE, ARTIFICIAL INTELLIGENCE, ARTIFICIAL NEURAL NETWORK, DEEP LEARNING, ELU, MULTIDIMENSIONAL MEDICAL IMAGES, PRELU, RELU, SOFTPLUS, SEMANTIC SEGMENTATION, COMPUTER VISION, 3D CONVOLUTIONAL NEURAL NETWORK

The object of the study is the application of adaptive ELU and PReLU with convolutional neural network for semantic segmentation of multidimensional medical images.

The subject of the study is adaptive activation functions with a three-dimensional U-Net architecture.

The purpose of the work is the software implementation of AdELU, AdPReLU and their embedding into U-Net with three-dimensional convolutional layers to solve the problem of semantic segmentation on three-dimensional medical data. To compare the effectiveness of adaptive activation functions for medical data with static activation functions.

Research methods – analysis of existing activation functions, study of the field of study, practical problem solving and comparative analysis.

Implemented U-Net, EdELU and AdPReLU. Experimental analysis of a dataset consisting of multi-institutional preoperative MRI scans and focusing on the segmentation of internally heterogeneous (in appearance, shape and histology) brain tumors – BraTS20. Neural networks are trained, and a comparative analysis is done.

ЗМІСТ

| | |
|--|----|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів | 7 |
| Вступ..... | 8 |
| 1 Аналіз предметної галузі та формалізована постановка задачі | 9 |
| 1.1 Комп'ютерний зір у медицині..... | 10 |
| 1.1.1 Класичний комп'ютерний зір | 10 |
| 1.1.2 Deep Learning у комп'ютерному зорі | 11 |
| 1.2 Семантична сегментація | 12 |
| 1.2.1 Труднощі медичної сегментації..... | 13 |
| 1.2.2 Метрики якості семантичної сегментації..... | 14 |
| 1.3 Функції активації..... | 15 |
| 1.3.1 Sigmoid та Tanh..... | 18 |
| 1.3.2 ReLU, Leaky ReLU та Parametric ReLU | 21 |
| 1.3.3 Exponential Linear Units | 25 |
| 1.3.4 Softplus | 27 |
| 1.3.5 Softmax та Hardmax | 28 |
| 1.4 Проблема градієнтів, що затухають і вибухають | 30 |
| 1.5 Постановка завдання дослідження | 33 |
| 2 Адаптивні активаційні функції в задачі семантичної сегментації | 35 |
| 2.1 Нейрона мережева архітектура для сегментації | 35 |
| 2.1.1 Дво- та тривимірний згортка | 35 |
| 2.1.2 3D U-Net..... | 38 |
| 2.2 Пропозиція адаптивних варіантів активаційних функцій..... | 39 |
| 2.2.1 Адаптивний PReLU | 40 |
| 2.2.2 Адаптивний ELU | 41 |
| 2.2.3 Адаптивний SoftPlus..... | 43 |
| 3 Програмна реалізація, обробка даних та аналіз результатів..... | 44 |
| 3.1 Опис бібліотек та платформ | 44 |
| 3.1.1 Python | 44 |

| | |
|--|----|
| 3.1.2 PyTorch..... | 44 |
| 3.1.3 NiBabel | 45 |
| 3.1.4 Seaborn..... | 46 |
| 3.1.5 NumPy | 46 |
| 3.2 Дослідження даних..... | 47 |
| 3.2.1 МРТ-модальності..... | 47 |
| 3.2.2 Первина обробка даних | 50 |
| 3.3 Опис умов експериментів | 51 |
| 3.4 Аналіз експериментів | 52 |
| 3.4.1 Метрики | 52 |
| 3.4.2 Поведінка параметрів активаційних функцій під час навчання . | 61 |
| 3.5 Подальший розвиток | 68 |
| Висновки..... | 69 |
| Перелік джерел посилання | 70 |
| Додаток А Вихідний код програми | 72 |
| Додаток Б Відомість кваліфікаційної роботи | 95 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- Карта ознака – feature map;
- Конструювання ознак – feature engineering;
- КТ – комп'ютерна томографія;
- МРТ – магнітно-резонансна томографія;
- Правила диференціювання складної функції – chain rule;
- Підвищення дискретизації – upsampling;
- УЗД – ультразвукова діагностика;
- Шар розгортки – convolution up layer;
- Швидкість відновлення ваг – learning rate;
- ШІ – штучний інтелект;
- ШНМ – штучна нейронна мережа;
- CNN – Convolutional Neural Network – згорткових нейронних мереж;
- DSC – Dice Score – Індекс Соренсена;
- Max pooling – розрахувати максимальне значення для кожного фрагмента карти ознака.

ВСТУП

Високоякісна візуалізація скорочує кількість необхідних медичних процедур (наприклад, хірургічне втручання, якщо використовувати МРТ або УЗД) для постановки діагнозу та прискорює процес прийняття медичних рішень. З прогресом в області комп'ютерного зору поліпшується і медична візуалізація. Це означає, що її переваги будуть продовжувати збільшуватися в майбутньому в міру того, як все більше дослідників в області комп'ютерного зору і медичних фахівців будуть об'єднувати свої зусилля для розвитку медичної візуалізації. Нині штучні нейронні мережі широко використовуються в технологіях комп'ютерного зору для розв'язання задач медицини, завдяки їх можливості налаштовувати параметри в процесі обробки інформації та їх універсальним апроксимаційним здібностям. Найбільш широко використовуються багат шарові перцептрони, вузлами-нейронами яких зазвичай є елементарні перцептрони Розенблата із сигмоїдальними функціями активації. На основі класичних багат шарових перцептронів було створено глибокі нейронні мережі. Це призвело до підвищення ефективності обробки зображень і почало нову еру комп'ютерного зору. Однак існують значні обчислювальні проблеми, пов'язані з градієнтом, що зникає і вибухає, які пов'язані з формою сигмоїдальних функцій. Тому як функції активації часто використовуються функції родини Rectified Linear Unit. У цю родину входять такі функції, як випрямлена лінійна одиниця (ReLU); параметрична випрямлена лінійна одиниця (PReLU); експонентна лінійна одиниця (ELU) та багато інших. Всі перераховані функції активації мають свої переваги та обмеження, до того, всі вони мають один загальний головний мінус – вони не адаптивні. Адаптивна функція активації має кращі можливості навчання, ніж традиційна (фіксована активація), оскільки вона значно покращує швидкість збіжності, особливо на ранніх стадіях навчання, а також точність вирішення задачі [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ФОРМАЛІЗОВАНА ПОСТАНОВКА ЗАДАЧІ

Методи сегментації медичних зображень знаходять межі об'єкта на зображенні і призначають семантичні ярлики пікселям в 2D зображеннях або вокселям в 3D/4D зображеннях.

Ефективні та дієві методи сегментації медичних зображень дають безліч важливих клінічних переваг [2]:

– сканери. Комп'ютерна томографія (КТ) і Магнітно-резонансна томографія (MRI) мають безліч можливих конфігурацій і протоколів зображення. Створення сканерами узгоджених і відтворюваних зображень високої якості для різних пацієнтів можливо тільки якщо сканування персоналізоване по відношенню до пацієнта. Так само, як і висока сканувальна пропускну здатність є питанням зменшення вартості процедури сканування, так і захист пацієнта від небажаної радіації від КТ сканера є однією з головних проблем. Ідеальна діагностика сканером повинна персоналізувати зображення цільового регіону конкретного пацієнта таким чином, щоб не опромінювати пацієнта більше, ніж потрібно, але і не втратити цінну інформацію. Тому ефективне визначення розташування та меж органів в попередніх зображеннях дозволяє персоналізувати сканування, і при цьому зменшує час, вартість і одержувану дозу опромінення пацієнта при скануванні;

– читання медичних зображень для діагностики, терапії та планування операцій. Під час читання медичного зображення з метою виявлення уражень (хвороби) в конкретних органах і частинах тіла, рентгенологу необхідно відзначити потрібний регіон, де була виявлена хвороба, і скласти звіт. Сегментація медичних зображень дозволяє автоматизувати процес виявлення і визначення ураженого регіону, що прискорює час визначення діагнозу і зменшує вартість діагностики;

– моделювання і кількісний вимір. Вимірювання об'єму органів є

важливим для кількісної діагностики хвороби. Визначення цільового органна та його кількісний вимір забирають багато часу в лікаря, особливо при роботі з багатовимірними зображеннями. Сегментація зображень дозволяє автоматизувати кількісні вимірювання і надає матеріал для моделювання.

1.1 Комп'ютерний зір у медицині

Технології комп'ютерного зору аналізують медичні зображення, що підвищує точність та швидкість діагностики захворювань. Комп'ютерний зір використовується для пошуку пухлин на зображеннях МРТ, пневмонії на рентгенівських знімках, визначення позиції серцевих клапанів пацієнтів на знімках УЗД, допомагає приймати рішення про необхідність переливання крові, проводять первинну діагностику деяких захворювань, технології комп'ютерного зору є критичними для роботи роботів-хірургів та інше.

Алгоритми нейронних мереж, а якщо точніше, підклас архітектур збільшення роздільної здатності (алгоритми супер резольюшина), покращують якість знімків рентгена, КТ, МРТ, прибирають зайві шуми та спотворення, що дозволяє зменшити частку опромінення пацієнта, т.к. час перебування пацієнта на апараті стає коротшим.

Варіація медичних зображень відрізняється від зображень, які роблять за допомогою фотокамери. Медичні зображення, як правило, представляють собою об'ємні зображення (3D) та іноді мають додатковий часовий вимір (4D) і/або декілька каналів (4-5D) (наприклад, багатомодальні МРТ зображення). Це потребує нішеві підходи комп'ютерного зору.

1.1.1 Класичний комп'ютерний зір

Методи комп'ютерного зору давно застосовують для автоматичного аналізу медичних зображень. Найчастіше «класичні» методи комп'ютерного

зору застосовують, коли потрібно отримати деяку кількісну інформацію про зображення (пов'язану з кольором, контурами, траєкторією, кількістю об'єктів і так далі). Більшість методів та алгоритмів цієї групи спочатку витягують із зображення корисні характеристики, після працюють з ними для вирішення завдання [3].

Існують такі основні підходи класичного комп'ютерного зору:

а) аналіз контурів та градієнтів. Методи контурного аналізу в здебільшого базуються на контури зображення. Найпоширеніші контурні методи – це методи математичної морфології, що дозволяють ефективно вирішувати такі прості задачі, як відокремлення об'єкта від фону, створення масок об'єктів тощо;

б) порівняння з шаблоном. Пошук за шаблоном – найчастіше використовуваний метод класичного комп'ютерного зору. Існує кілька його підвидів, наприклад, при пошуку простої відповідності початкове зображення сканується покроково, і при кожному скануванні відбувається розрахунок частки збігу даної ділянки з шаблоном відносно обраної метрики;

в) зіставлення за характеристиками (зазвичай ключові точки). Існує кілька класичних алгоритмів (наприклад, SIFT, SURF), які:

1) вибирають найбільш «важливі» точки на зображенні, оцінюючи їхнє положення, контраст, із сусідніми точками тощо;

2) створюють «дескриптори» ключових точок, тобто ідентифікатори досить стійкі до основних типів перетворень: зміна освітлення, зміна масштабу, зміна відстані до об'єкта, поворот об'єкта;

3) дозволяють порівнювати дескриптори на різних зображеннях і тим самим зіставляти їх фрагменти. [3]

1.1.2 Deep Learning у комп'ютерному зорі

З розвитком глибокого навчання і збільшенням доступних

обчислювальних потужностей багато інших методів машинного навчання були замінені. Причиною стало те, що глибоке навчання дозволяє уникнути конструювання ознак, тим самим усуває критичне джерело помилок.

1.2 Семантична сегментація

Семантична сегментація зображення – це попиксельна класифікація зображення. Під час такої класифікації пікселів, зображення розбивають на області зі схожими властивостями.

Мета – для зображення розміром $W \times H$ (2D зображення) або $W \times H \times D$ (3D зображення) згенерувати матриці $W \times H$ і $W \times H \times D$ відповідно, в яких кожна компонента це прогнозований ID класу для відповідних пікселів зображення.

Семантична сегментація відрізняється від детектування об'єктів, тим що не передбачає ніяких кордонів навколо об'єктів. Ми не розрізняємо різні екземпляри одного і того ж об'єкта. Наприклад, на зображенні може бути кілька дерев, і всі вони будуть мати одну спаяну маску.

Сегментація медичного зображення – це процес автоматичного або напівавтоматичного визначення меж в 2D або 3D зображенні. Основною складністю сегментації медичних зображень є висока варіабельність медичних зображень. Крім того, для створення медичних зображень використовують безліч різних методів (рентген, КТ, МРТ та інші).

Можливі застосування:

- автоматичне вимірювання органів;
- підрахунок клітин;
- моделювання;
- результат сегментації може бути використаний для отримання

подальших діагностичних знань.

Відтоді як була запропонована повнозв'язна згорткова нейронна мережа (FCN), автоматичні методи сегментації зображень досягли великих

успіхів. Повнозв'язна згорткова нейронна мережа (FCN) була першою нейронною мережею зі згортками, яка трансформувала завдання класифікації в задачу сегментації з внутрішньомережевим підвищенням дискретизації (upsampling) і втратами попіксельно (pixelwise loss) [4]. Методи сегментації медичних зображень можна розділити на дві категорії: двовимірні та тривимірні методи залежно від розміру вхідних даних. Архітектура U-Net є найпопулярнішою FCN для сегментації медичних зображень. [5]

1.2.1 Труднощі медичної сегментації

Головна проблема медичної сегментації полягає в тому, що органи, частини тіла мають значну мінливість форм і зовнішнього вигляду, які викликані безліччю факторів [2]:

- шум. Медичне обладнання генерує шум / артефакти, властиві його фізичному датчику і процесу формування зображення. Ступінь артефакту залежить від модальності та конфігурації зображення. Наприклад, в той час, як високодозова КТ дає зображення з меншою кількістю артефактів, низькодозова КТ дає значно зашумлене зображення. Так само металеві імпланти можуть створювати шум;

- м'які тканини. Анатомічні структури, такі як внутрішні органи, є м'якими тканинами з подібними властивостями. Вони можуть торкатися один одного і створюють ледве помітні кордони між органами, що ускладнює сегментацію без перекриття;

- відмінності пацієнтів і патології. У пацієнтів різна статура: товстий або худий, високий або низький, дорослий або дитина і так далі. В результаті анатомічні структури також мають різні форми. Патологія може привести до появи сильно деформованих анатомічних структур або навіть до відсутності деяких з них (рис. 1.1). Це значно ускладнює статистичне моделювання;

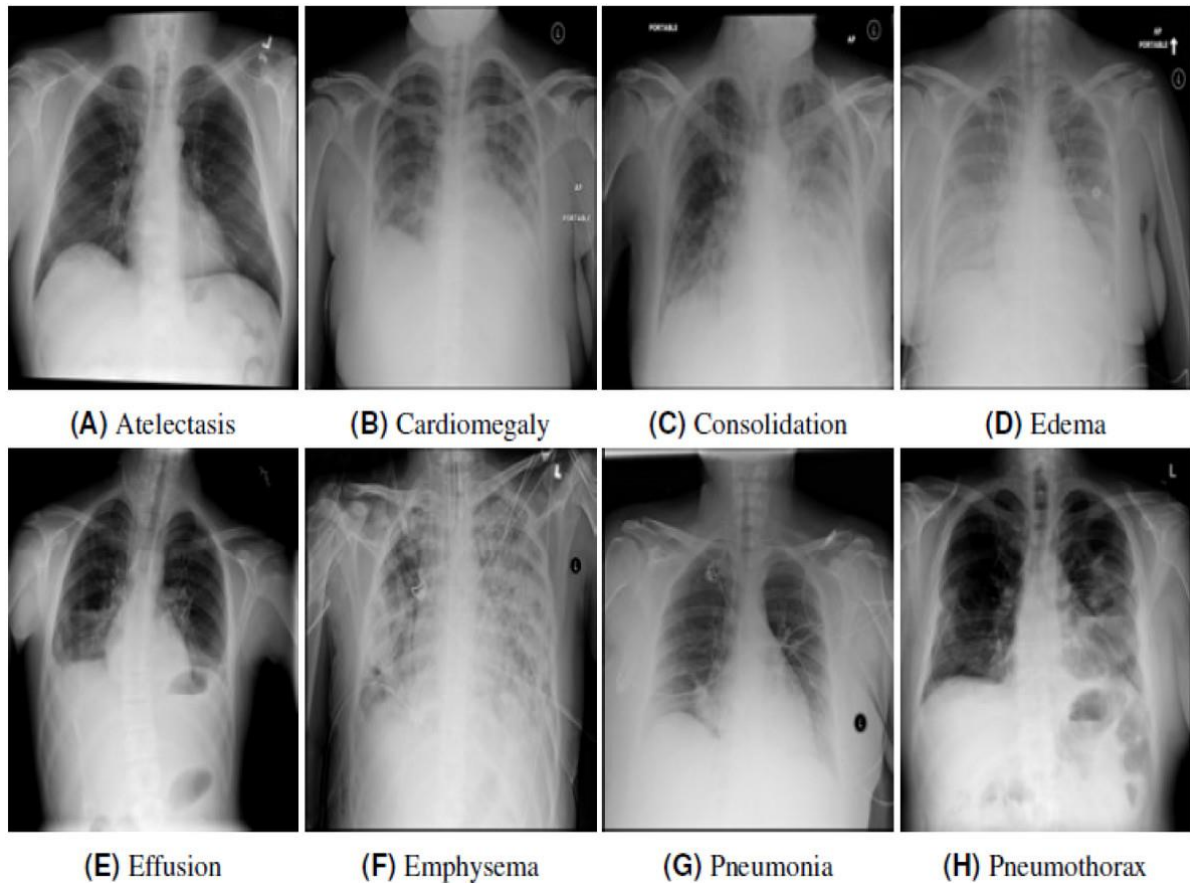


Рисунок 1.1 – Приклад відмінності анатомічних структур [6]

– рух. Не тільки довільні рухи пацієнтів сприяють деформації анатомічних форм на зображенні, але так само і рух від дихання, ковтання, серцевого циклу, циркуляції крові та спинномозкової рідини.

1.2.2 Метрики якості семантичної сегментації

Коефіцієнт подібності Dice Score та перетин над об'єднанням (IoU) є двома основними метриками для оцінки ефективності методів сегментації, Dice та IoU є мірами перекриття між згенерованою та цільовою масками. Dice Score та IoU визначаються таким чином, відповідно:

$$Dice = \frac{2 \times TP}{2 \times TP + FP + FN} = \frac{2|A \cap B|}{|A| + |B|} \quad (1.1)$$

$$IoU = \frac{TP}{2 \times TP \times FP \times FN} = \frac{2|A \cap B|}{|A| + |B|} \quad (1.2)$$

де TP – істинно позитивно класифіковані пікселі;

FP – хибно позитивно класифіковані пікселі;

FN – хибно негативно класифіковані пікселі;

$|A \cap B|$ – спільні елементи між множинами A і B;

$|A|$, $|B|$ – потужність множин A і B відповідно.

Для оцінки якості згенерованих масок $|A \cap B|$ апроксимують як поелементне множення між цільовою та згенерованою масками, а потім підсумовують результуючу маску.

1.3 Функції активації

Найбільш популярні функції активації нейронних мереж: ступінчаста, лінійна та нелінійна.

Бінарна ступінчаста функція (рис. 1.2) залежить від порогового значення, яке визначає нейрон буде активовано чи ні. Вхідна інформація, яка подається на функцію активації, порівнюється із заданим порогом; якщо значення входу проходить поріг, то нейрон активується, в інакшому випадку він деактивується, тобто його вихід не передається до наступного шару.

Математично двійкову ступінчасту функцію можна представити як:

$$f(x) = \begin{cases} x < 0, 0 \\ x \geq 0, 1 \end{cases} \quad (1.3)$$

де x – значення з виходу шару (внутрішній сигнал активації).

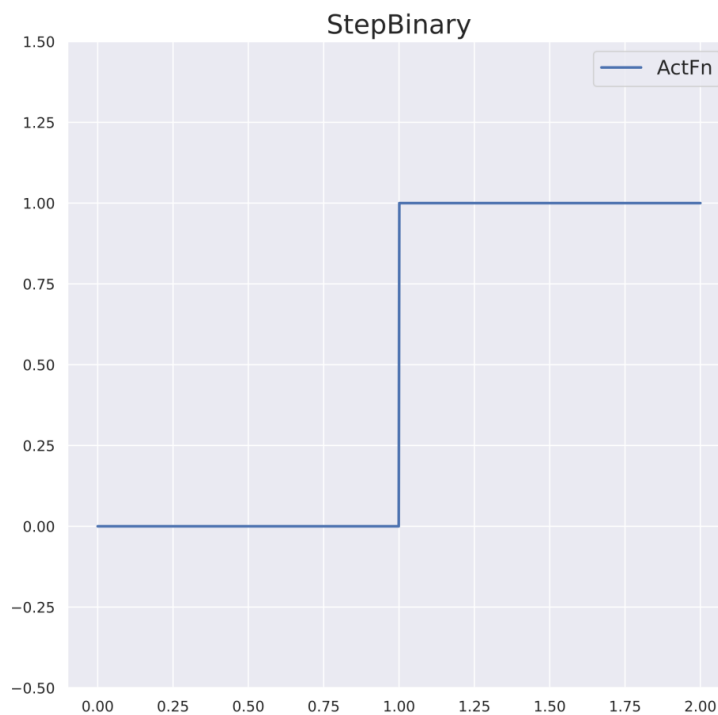


Рисунок 1.2 – Графік бінарної ступінчастої функції

У лінійної функції (рис. 1.3) активація пропорційна входу.

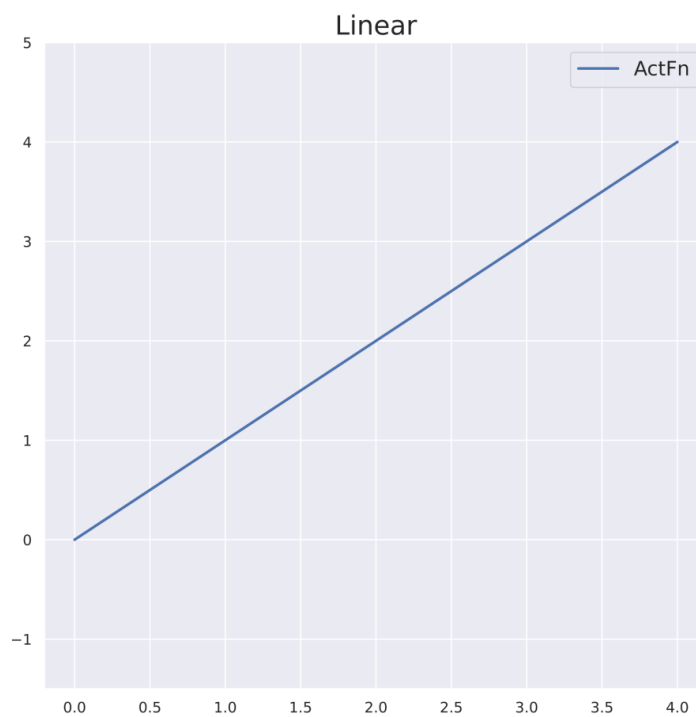


Рисунок 1.3 – Графік лінійної функції

Математично лінійну функцію можна представити як:

$$f(x) = x, \quad (1.4)$$

де x – значення з виходу шару (внутрішній сигнал активації).

У лінійної функції активації є критичний мінус: з нею нема сенсу використовувати зворотне поширення, оскільки похідна функції є константою, тобто, похідна її функції не пов'язана з входом. Усі шари нейронної мережі зіллються в один, якщо використовувати лінійну функцію активації (рис. 1.4).



Рисунок 1.4 – Ілюстрація простої нейронної мережі з трьома шарами та одним нейроном у кожному шарі

$$\begin{aligned}
 y &= h2 W3 + b3 = (h1 W2 + b2) W3 + \\
 &+ b3 = h1 W2 W3 + b2 W3 + \\
 + b3 &= (x W1 + b1)W2 W3 + b2 W3 + b3 = x W1 W2 W3 + \\
 &+ b1 W2 W3 + b2 W3 + b3 = x W' + b'.
 \end{aligned} \quad (1.5)$$

Як видно з прикладу з трьома шарами та лінійної активаційної функції: комбінація кількох лінійних перетворень можна замінена одним перетворенням, а комбінація кількох зміщень – лише одним зміщенням; ця поведінка не зміниться навіть якщо додати будь-яку кількість шарів. Отже, можна замінити таку нейронну мережу одношаровою нейронною мережею.

Тобто, додавання шарів не збільшує апроксимуючу здатність нейронної мережі з лінійними активаційними функціями.

Дані реального світу майже завжди нелінійні. Абсолютна більшість відносин між даними реального світу настільки складні, що для їхнього уявлення нам потрібні дуже складні нелінійні функції. Функція активації додає нелінійності до нейронної мережі. Використовуючи відповідну нелінійну функцію активації, ми можемо допомогти нейронним мережам зрозуміти ці нелінійні відносини. Доведено, що нейронні мережі є універсальними апроксиматорами функцій [7]. Це означає, що при достатній кількості навчальних даних і часу навчання нейронна мережа може згенерувати наближену функцію для представлення будь-якої складного взаємозв'язку в даних.

Для апроксимації нелінійних функцій потрібні нелінійні функції активації, більшість проблем реального світу дуже складні й нелінійні.

Нелінійні функції активації вирішують такі обмеження лінійних функцій активації:

- вони допускають зворотне поширення, оскільки похідна їх функцій буде пов'язана з входом;
- вони з кожним шаром підвищують апроксимуючу здатність нейронної мережі, оскільки вихід тепер буде нелінійною комбінацією вхідних даних, що пройшли через кілька шарів.

1.3.1 Sigmoid та Tanh

Сигмоїда приймає будь-яке дійсне значення як вхідний сигнал і видає значення в діапазоні $[0, 1]$.

Чим більше вхідне значення, тим ближче вихідне значення до 1, тоді, чим менше вхідне значення, тим ближче вихідне значення до 0 (рис. 1.5).

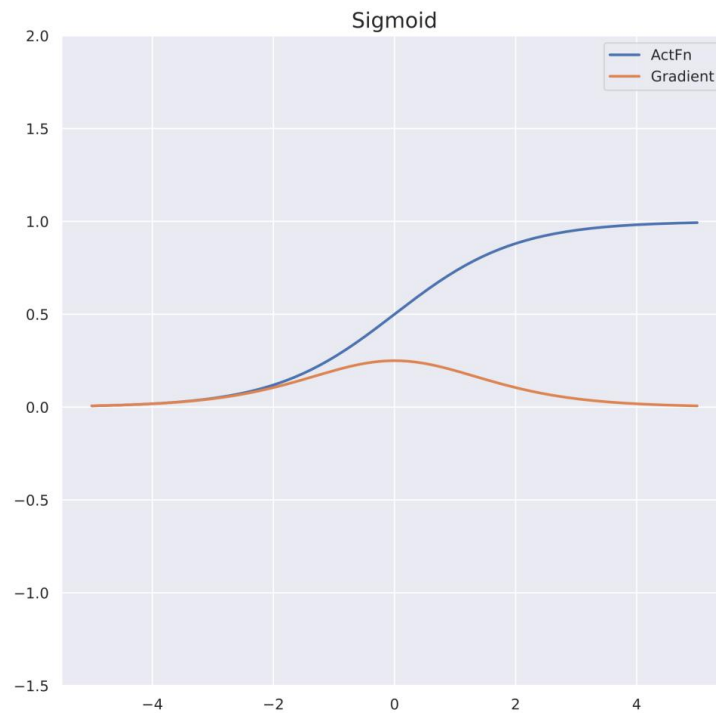


Рисунок 1.5 – Графік сигмоїди та її похідної

Математично сигмоїду можна виразити як:

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (1.6)$$

де x – значення отримане після виконання лінійного перетворення з використанням ваг та зміщень (внутрішній сигнал активації).

Часто сигмоїду використовують, коли потрібно спрогнозувати ймовірність будь-якої події. Також плюсом сигмоїди є те, що вона забезпечує плавний градієнт, тобто запобігає стрибкам у вихідних значеннях.

Похідна сигмоїди:

$$\sigma'(x) = \sigma(x) \times (1 - \sigma(x)). \quad (1.7)$$

Як видно з рисунка (рис. 1.6), значення градієнта значні лише для

діапазону від -3 до 3, а поза цим діапазоном графік наближається до нуля. Це означає, що для значень більше ніж 3 або менше ніж -3 функція буде мати дуже маленькі градієнти. Коли значення градієнта наближається до нуля, нейронна мережа перестане навчатися, інакше кажучи, страждає від проблеми градієнта, що зникає (див. розділ 1.4).

Функція активації \tanh схожа на сигмоїдну функцію (рис. 1.6), і так само як і сигмоїда має S-подібну форму з різницею в діапазоні вихідних значень $[-1, 1]$. У \tanh чим більше вхідне значення, тим ближче вихідне значення буде до 1, чим менше вхідне значення, тим ближче вихідне значення буде до -1.

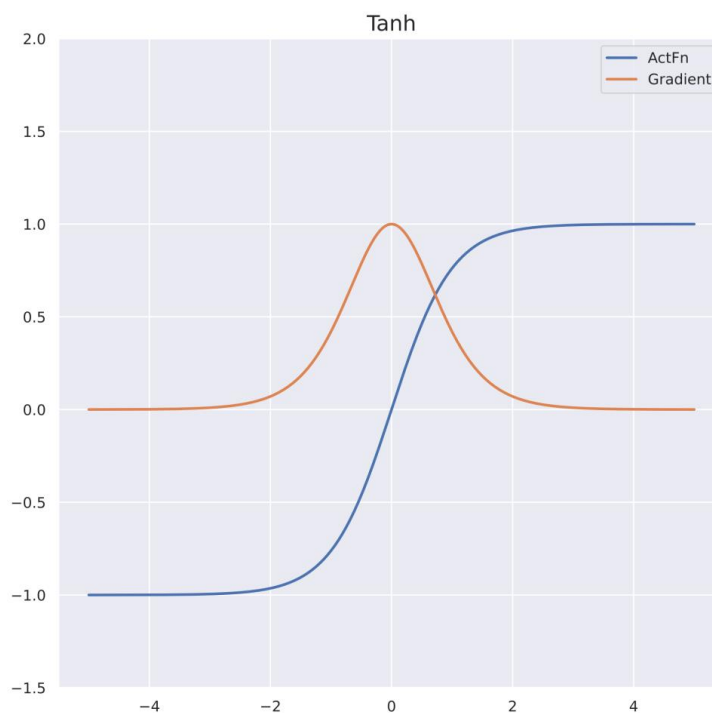


Рисунок 1.6 – Графік Tanh та його похідної

Tanh зазвичай використовується у прихованих шарах нейронної мережі, оскільки її значення лежать у діапазоні від -1 до 1; тоді середнє значення виходів прихованого шару виходить близьким чи рівним 0. Це допомагає центрувати дані, отже, спрощує навчання наступного шару.

Так само іноді Tanh використовують як активаційну функцію вихідного шару в задачах сентименту аналізу. Вихід активаційної функції Tanh центрований щодо нуля, що дає можливість розбити вихідні значення на негативні, нейтральні та позитивні.

Математично Tanh можна виразити як:

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}, \quad (1.8)$$

де x – значення отримане після виконання лінійного перетворення з використанням ваг та зміщень (внутрішній сигнал активації).

Похідна tanh:

$$f'(x) = \tanh^2(x). \quad (1.9)$$

Як видно з рисунка (рис 1.7) похідної tanh функції, як і сигмоїдна функція активації, також схильна до проблеми градієнта, зникає.

1.3.2 ReLU, Leaky ReLU та Parametric ReLU

На відміну від сигмоїди та tanh ReLU не схильний до проблеми градієнта, що затухає. Також відмінною особливістю ReLU є те, що на виході ми отримуємо розріджену результуючу матрицю обчислень, через те, що ReLU (рис. 1.7) не активує всі нейрони одночасно. Нейрони будуть активовані тільки в тому випадку, якщо вихід лінійного перетворення більше за 0. З обчислювальної точки зору функція ReLU набагато ефективніша у порівнянні з сигмоїдною та tanh, через те, що активує лише певні нейрони. Розрідженість призводить до створення простих моделей, які менше схильні до перенавчання.

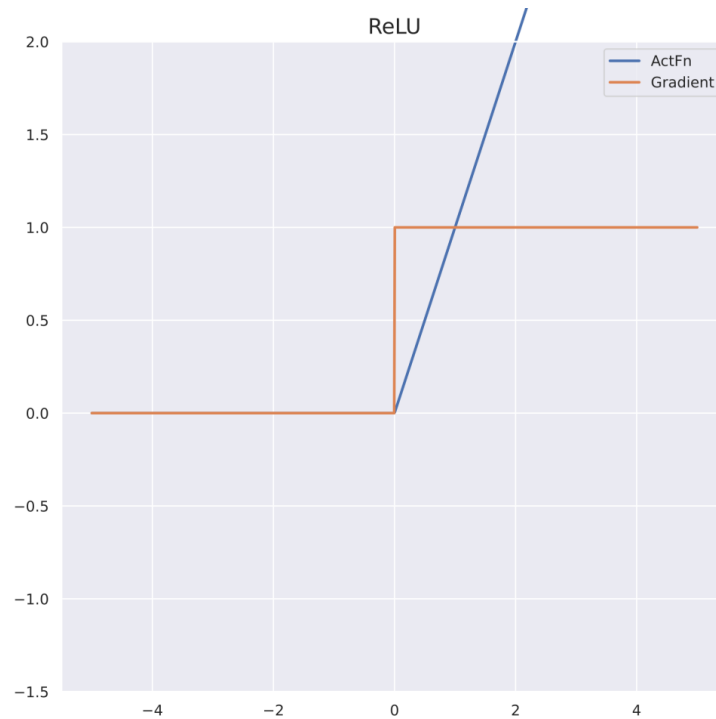


Рисунок 1.7 – Графік ReLU та його похідної

Математично ReLU можна виразити як:

$$f(x) = \max(0, x), \quad (1.10)$$

де x – значення отримане після виконання лінійного перетворення з використанням ваг та зміщень (внутрішній сигнал активації).

Похідна ReLU:

$$f'(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (1.11)$$

ReLU схильний до так званої проблеми «мертвого нейрона». У негативній частині графіка похідної значення градієнта дорівнює нулю. Тому в процесі зворотного поширення ваги деяких нейронів не оновлюються. Що призводить до утворення мертвих нейронів, які

втрачають свою здатність до активації, що знижує здатність моделі до навчання.

Leaky ReLU є покращеною версією ReLU. Переваги Leaky ReLU такі ж, як і у ReLU, крім того, оскільки він дозволяє здійснювати зворотне поширення навіть для негативних вхідних значень (рис. 1.8) через наявність невеликого кута нахилу в цьому діапазоні, дана модифікація спрямована на розв'язання проблеми нейрона, що помирає.

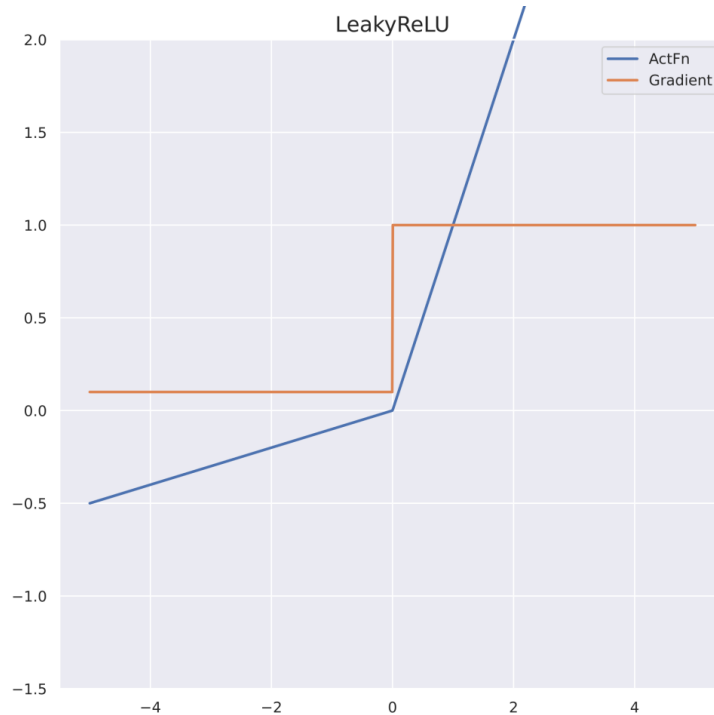


Рисунок 1.8 – Графік LReLU та його похідної

Математично LReLU можна виразити як:

$$f(x) = \max(0.1x, x), \quad (1.12)$$

де x – значення отримане після виконання лінійного перетворення з використанням ваг та зміщень (внутрішній сигнал активації).

Похідна LReLU:

$$f'(x) = \begin{cases} 0.01, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (1.13)$$

Обмеження, з якими стикається ця функція, включають:

- для негативних вхідних значень вихід нейрона може бути не консистентним;
- градієнт для негативних значень досить маленький, що збільшує час навчання.

Параметричний ReLU (PReLU) – це тип Leaky ReLU, який замість фіксованого нахилу, наприклад, значенням 0,01 (рис. 1.9) робить його параметром.

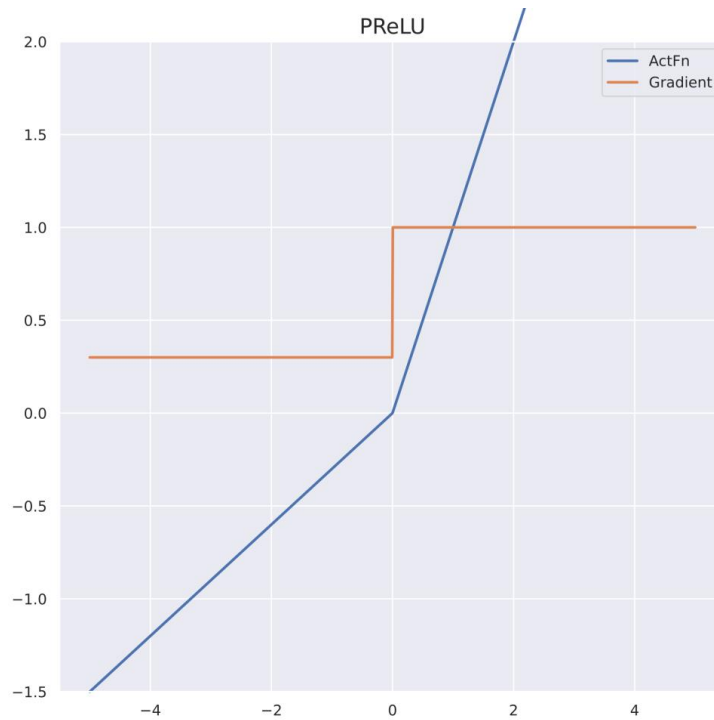


Рисунок 1.9 – Графік PReLU (a=0.3) та його похідної

Математично PReLU можна виразити як:

$$f(x) = \max(ax, x), \quad (1.14)$$

де x – значення отримане після виконання лінійного перетворення з використанням ваг та зміщень (внутрішній сигнал активації);

α – кут нахилу для негативних значень.

Похідна PReLU:

$$f'(x) = \begin{cases} \alpha, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (1.15)$$

Обмеження цієї функції в тому, що вона додає ще один гіперпараметр, який треба підбирати.

1.3.3 Exponential Linear Units

Подібно до Leaky ReLU, ELU має невеликий нахил для негативних значень, їх відмінності полягає в тому, що замість прямої лінії ELU має логарифмічну криву (рис. 1.10).

ELU є гідною альтернативою ReLU завдяки наступним перевагам:

- в той час як ReLU різко згладжується, ELU робить це повільно, доки його вихід не стане рівним $-\alpha$;
- уникає проблеми мертвого нейрона ReLU, вводячи логарифмічну криву для негативних значень входу.

ELU не позбавлена своїх обмежень:

- вона збільшує час обчислень через включення експоненційної операції;
- не відбувається навчання коефіцієнта α ;
- схильна до проблеми градієнта, що вибухає.

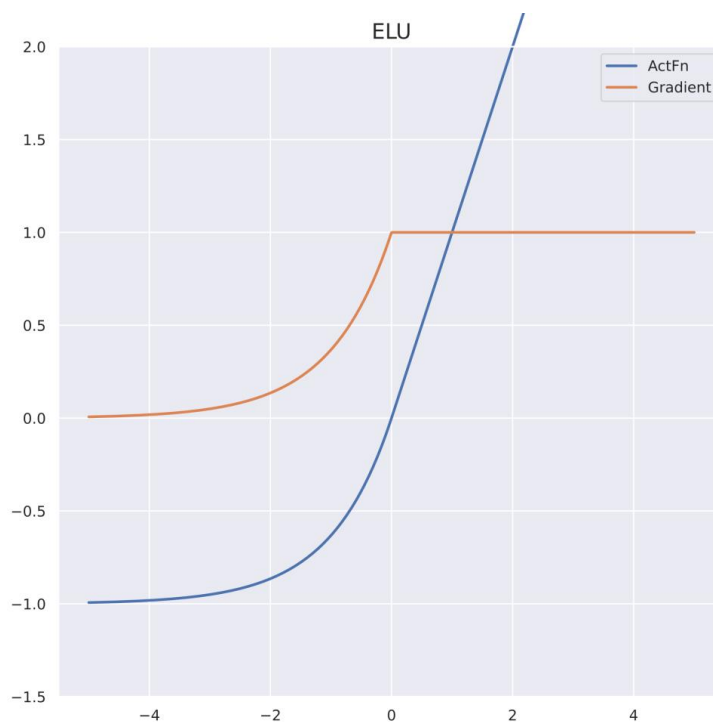


Рисунок 1.10 – Графік ELU та її похідної

Математично ELU можна виразити як:

$$f(x) = \begin{cases} \alpha(e^x - 1), & x < 0 \\ x, & x \geq 0 \end{cases} \quad (1.16)$$

де x – значення отримане після виконання лінійного перетворення з використанням ваг та зміщень (внутрішній сигнал активації);

α – гіперпараметр, який потрібно налаштувати.

Похідна ELU:

$$f'(x) = \begin{cases} f(x) + \alpha, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (1.17)$$

1.3.4 Softplus

Softplus можна розглядати як гладку версію ReLU (рис. 1.11).

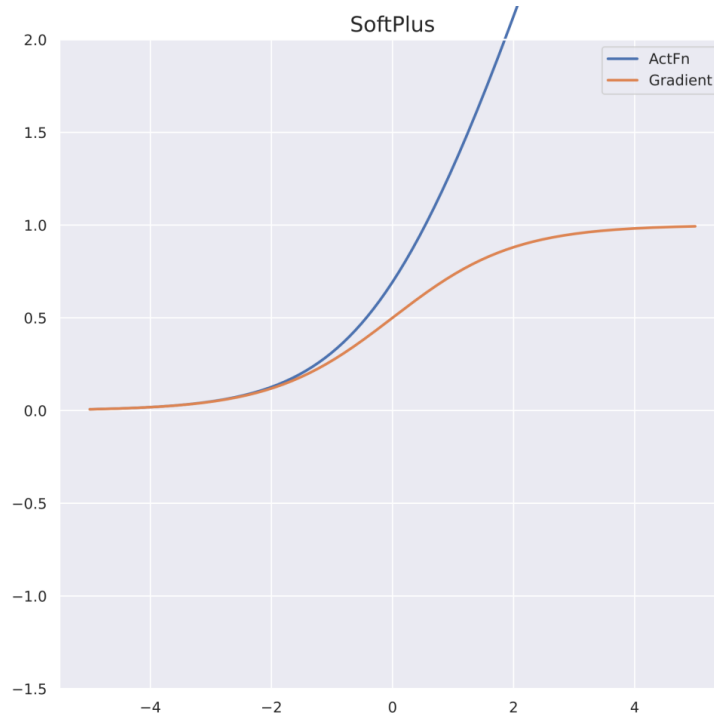


Рисунок 1.11 – Графік Softplus та його похідної

$$f(x) = \ln(1 + e^x). \quad (1.18)$$

Також можна контролювати різкість функції:

$$f(x) = \frac{\ln(1 + e^{kx})}{k}, \quad (1.19)$$

де k – параметр різкості;

x – значення отримане після виконання лінійного перетворення з використанням ваг та зміщень (внутрішній сигнал активації).

Softplus тісно пов'язана із сигмоїдною функцією. При $x \rightarrow -\infty$ ці дві

функції стають ідентичними:

$$f'(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}} \quad (1.20)$$

І ReLU, і Softplus багато в чому схожі, за винятком околиці 0, Softplus має привабливу диференційовність через гладкість функції градієнта. Водночас, обчислювати ReLU та її похідну значно ефективніше, ніж функцію Softplus. Оскільки Softplus має у своєму вираженні $\exp(x)$ і $\ln(x)$, ця функція може мати справу з проблемами чисельної стійкості (numerical stabilit). Ця проблема виникає у $\log(x)$, коли x близький до нуля, в $\exp(x)$, коли x близький до нуля, або x занадто великий. [8]

1.3.5 Softmax та Hardmax

Вихід сигмоїдної функції знаходиться в діапазоні від 0 до 1, що можна представити як ймовірність деякої події, але сигмоїди мають обмеження, коли. Сигмоїдна активація підходить для задач бінарної класифікації, де вихід інтерпретується як біноміальний розподіл ймовірності, разом з тим сигмоїда не підходить для задач мультикласової класифікації, де класи є взаємозаперечними, де потрібний мультиноміальний розподіл ймовірності.

Softmax (рис. 1.12) обчислює відносні ймовірності, інакше кажучи, сума ймовірностей всіх класів дорівнюватиме 1.

Найчастіше вона використовується як функція активації для останнього шару нейронної мережі у разі багатокласової класифікації.

Функція Softmax описується як комбінація кількох сигмоїд.

Математично її можна висловити так:

$$\sigma(y = j|z_i)_i = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}}, \quad (1.21)$$

де i, j – індекс поточного класу, $i, j=1, \dots, n$;

z_i – вихід попереднього шару;

e^{z_i} – експонента параметра;

n – кількість класів у задачі мультикласової класифікації;

$\sum_{j=1}^n e^{z_j}$ – сума експонентного параметра.

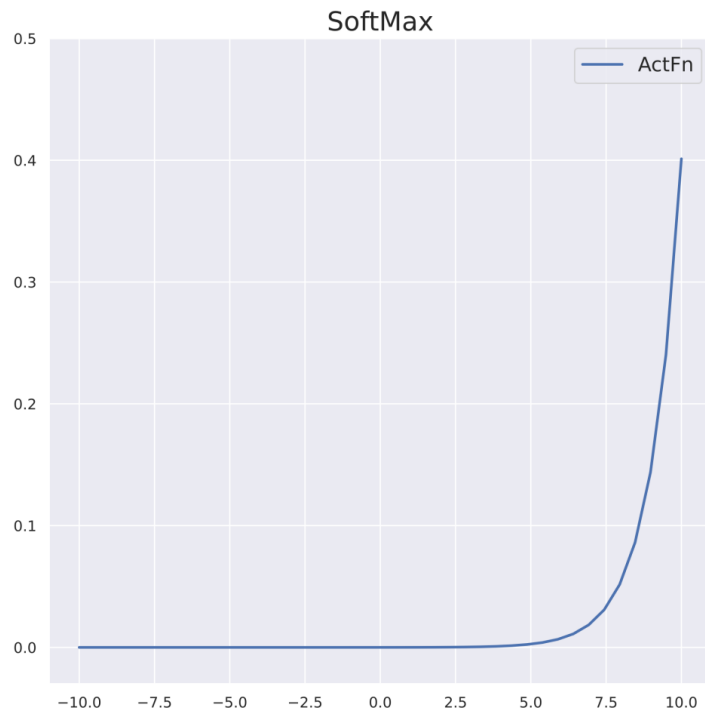


Рисунок 1.12 – Графік SoftMax

Сигмоїдну функцію активації можна використовувати як функції активації для завдань мультикласової класифікації, де класи не є взаємовиключними. Такі завдання часто називають класифікацією за декількома мітками (multi-label), а не мультикласовою класифікацією. У такому разі сума ймовірностей кожного класу може не дорівнювати 1.

Hardmax, як і softmax, використовують як активаційну функцію в останньому шарі. На відміну від softmax, hardmax все значення, крім максимального, обнуляє, максимальне нормалізується до 1. Як правило, модель машинного навчання навчають за допомогою softmax, але при

використанні моделі для прогнозування (інференсу) шар з softmax замінюється шаром з hardmax.

1.4 Проблема градієнтів, що згасають і вибухають

Проблема градієнта, що згасає та вибухає, зачіпає градієнти нелінійних функцій активації, які обчислюються в процесі зворотного поширення.

Під час зворотного розповсюдження ми рухаємося від виходу мережі до першого шару, обчислюємо похідну функції вартості J відносно ваг у кожному шарі. Оновлюємо ваги відносно обчислених похідних на кожному кроці. Обчислення похідних та зворотне поширення здійснюється за правилами диференціювання складної функції.

Беремо як приклад нейронну мережу з двома шарами, у кожному з яких один нейрон (рис. 1.13), активаційна функція – сигмоїда (через природу своїх похідних функції сигмоїда та \tanh є причинами загасання градієнта).

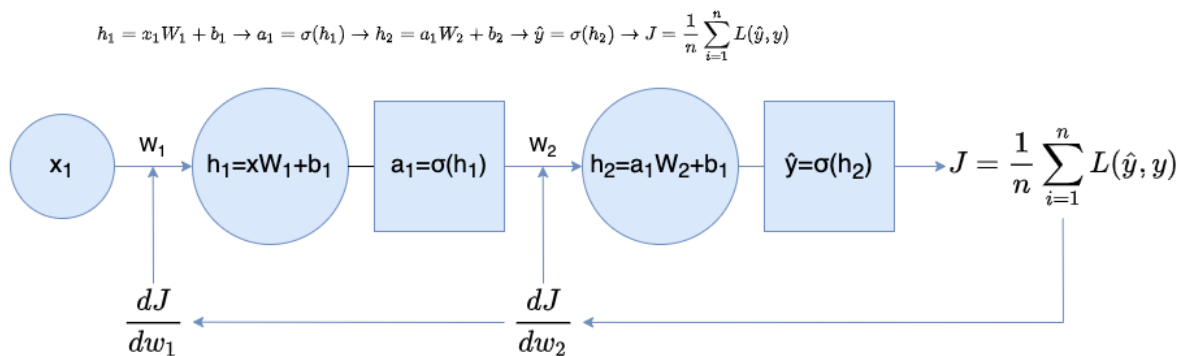


Рисунок 1.13 – зворотний хід для двошарової нейронної мережі з одним нейроном у кожному шарі

Щоб отримати градієнт ваги w_2 , спочатку потрібно обчислити

похідну функції вартості щодо спрогнозованого нейронною мережею \hat{y} .

\hat{y} – вихід функції активації в останньому шарі, яка приймає h_2 як вхідний сигнал, при цьому h_2 є добуток ваги w_2 та виходу попереднього шару a_2 . Похідну \hat{y} щодо h_2 :

$$\frac{d\bar{y}}{dh_2} = \sigma(h_2)(1 - \sigma(h_2)) = \frac{1}{1+e^{z_2}} \left(1 - \frac{1}{1+e^{z_2}}\right). \quad (1.22)$$

Візьмемо, що значення z_2 дорівнює 5. Отримуємо:

$$\frac{d\bar{y}}{dh_2} = \sigma(5)(1 - \sigma(5)) \approx 0.0066. \quad (1.23)$$

Також обчислюємо похідну h_2 відносно w_2 :

$$\frac{dh_2}{dw_2}. \quad (1.24)$$

Розрахуємо градієнт для коригування ваги w_2 , використовуючи правила диференціювання складної функції з'єднуємо (1), (3) та (4):

$$\frac{dJ}{dw_2} = \frac{dJ}{d\bar{y}} \frac{d\bar{y}}{dh_2} \frac{dh_2}{dw_2}. \quad (1.25)$$

В (2) було пораховано похідну функції активації у другому шарі при $h_2=5$, цей же градієнт є другим градієнтом у ланцюжку (4). За аналогією з (4) отримуємо вираз для пошуку градієнта для коригування ваги w_1 :

$$\frac{dJ}{dw_1} = \frac{dJ}{d\bar{y}} \frac{d\bar{y}}{dh_2} \frac{dh_2}{da_2} \frac{da_2}{dh_1} \frac{dh_1}{dw_1}. \quad (1.26)$$

Тепер нам слід оцінити похідні двох функцій активації. Оскільки похідні функцій активації сигмоїду в межах від 0 до 0,25, градієнт для

коригування w_1 матиме тенденцію «стиратися».

Таким чином градієнт стирається на кожному наступному шарі під час зворотного поширення. Коли градієнт для оновлення ваг дуже малий, час навчання займає багато часу, що в гіршому випадку може повністю зупинити навчання нейронної мережі.

Градієнт, що загасає під час навчання нейронної мережі можна визначити за такими ознаками:

- модель метрики покращується дуже повільно на етапі навчання або навчання на ранніх етапах припиняється зовсім, що означає, що будь-яке подальше навчання не покращує модель;

- ваги, розташовані ближче до вихідного шару моделі, будуть сильніше змінюватися, тоді як ваги, розташовані ближче до вхідного шару, змінюватимуться незначно (якщо взагалі будуть);

- ваги моделі зменшуються експонентно.

Градієнт, що вибухає є повною протилежністю градієнту, що загасає. Він виникає, коли похідні стають дедалі більше у міру просування з кожним шаром у процесі зворотного поширення. Ця проблема виникає через ваги, а не через функцію активації. Через високі значення ваг похідні також будуть вищими, отже, кожна нова оновлена вага буде сильно відрізнятися від попередньої, і градієнт ніколи не зійдеться. В результаті він може коливатись навколо мінімумів і ніколи не прийде до точки глобального мінімуму.

Градієнт, що вибухає, під час навчання нейронної мережі можна визначити за такими ознаками:

- у модель невеликий прогрес під час навчання;

- модель має різкі великі зміни функції втрат під час кожного оновлення;

- під час навчання функція втрат дорівнює NaN.

При зіткненні з цими проблемами, для підтвердження гіпотези про градієнт, що вибухнув, є деякі більш прозорі ознаки, наприклад:

- ваги моделі зростають експоненційно і стають дуже великими під час навчання моделі;
- ваги моделі зростають експоненційно і стають дуже великими під час навчання моделі;
- похідні константи.

1.5 Постановка завдання дослідження

Головним завданням магістерського проекту полягає в дослідженні методів сегментації медичних багатовимірних зображень за допомогою нейронних мереж, дослідження варіантів комбінування плюсів деяких активаційних функцій зі спробою уникнення їх мінусів, реалізація та дослідження поведінки адаптивних варіантів LReLU, ELU та Softplus, дослідження поведінки активаційних функцій під час навчання сегментатора.

Для реалізації поставленої мети було встановлено такий порядок дослідження:

- а) визначення проблеми;
- б) дослідження проблем медичної візуалізації та застосування алгоритмів сегментації на них;
- в) опрацювати літературу та інтернет-джерела по темі алгоритм навчання та оптимізації нейронних мереж;
- г) вибір набору даних;
- г) обробка вибраного набору даних;
 - 1) аналіз набору даних;
 - 2) первинна обробка набору даних;
 - 3) реалізація дата генераторів для навчання нейронних мереж;
- д) програмна реалізація 3D U-Net архітектур;
- е) визначення функцій активації на яких буде проведено дослідження;

- є) програмна реалізація модуля візуалізації для полегшення інтерпретації результатів навчання нейронних мереж;
- ж) підбір гіперпараметрів;
- з) реалізація AdELU, AdReLU, AdSoftplus;
- и) навчання 3D U-Net архітектури з вибраними функціями активації;
- і) дослідження поведінки функції активації під час навчання;
- ї) порівняльний аналіз результатів;

2 АДАПТИВНІ АКТИВАЦІЙНІ ФУНКЦІЇ В ЗАДАЧІ СЕМАНТИЧНОЇ СЕГМЕНТАЦІЇ

2.1 Нейрона мережева архітектура для сегментації

2.1.1 Дво- та тривимірна згортка

В основі двомірної згортки лежить ідея ядра, яке являє собою матрицю ваг. Це ядро «ковзає» по двовимірним вхідним даним та виконує поелементне множення з тією частиною вхідного сигналу, на якій воно цієї миті знаходиться, і потім підсумовує результати в один вихідний піксель (рис. 2.1).

2D Convolution називається двовимірним тому, що його ядро ковзає уздовж двовимірних даних.

Перевага використання CNN полягає в тому, що він може витягувати просторові характеристики з даних, наприклад, CNN може виявляти границі, розподіл кольорів і т.д. Ця його властивість і стало причиною ширшого застосування CNN в роботі з зображеннями.

| | | | | |
|-------|-------|-------|---|---|
| 3_0 | 3_1 | 2_2 | 1 | 0 |
| 0_2 | 0_2 | 1_0 | 3 | 1 |
| 3_0 | 1_1 | 2_2 | 2 | 3 |
| 2 | 0 | 0 | 2 | 2 |
| 2 | 0 | 0 | 0 | 1 |

| | | |
|------|------|------|
| 12.0 | 12.0 | 17.0 |
| 10.0 | 17.0 | 19.0 |
| 9.0 | 6.0 | 14.0 |

Рисунок 2.1 – Ядро 2D Convolution (ліворуч) та його вихід (праворуч)

Термін «фільтр» призначений для тривимірних структур з декількох ядер, зібраних разом. Для одноканальної 2D згортки фільтр – це те ж саме, що і ядро. Але 3D-фільтр це структура, що складається з набору ядер (рис. 2.2). Кожне ядро унікальне і витягує різні аспекти з вхідного каналу.

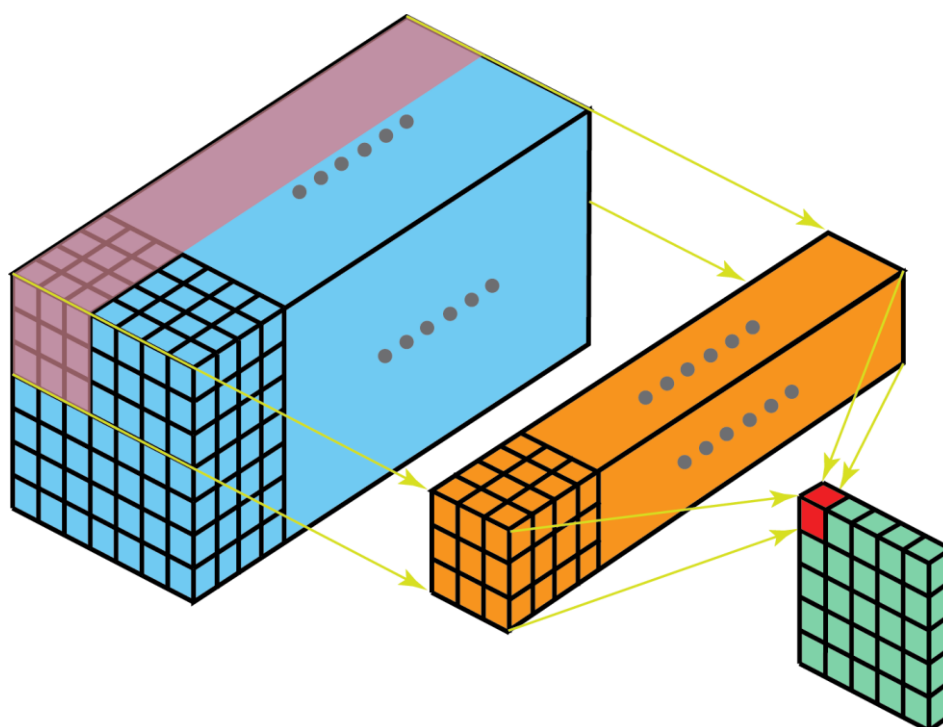


Рисунок 2.2 – Набір ядер 2D Convolution

Процес конволюції з 3D-фільтрами – це переміщення матриці тривимірного фільтра через вхідний шар. 3D-фільтр переміщається тільки в 2-х напрямках, по висоті і ширині зображення (тому така операція називається 2D-згорткою, хоча 3D-фільтр використовується для обробки об'ємних 3D-даних). Важливо відзначити те, що вхідний шар і фільтр мають однакову глибину: кількість каналів дорівнює кількості ядер.

2D Convolution є згорткою для тривимірних даних, при цьому ця операція називається двовимірною згорткою. У випадку двовимірної згортки на тривимірних даних глибина фільтра дорівнює глибині вхідного

шару, і 3D-фільтр переміщується тільки в 2-х напрямках (висота та ширина зображення). Результатом такої операції є 2D-зображення (тільки з 1-м каналом).

3D Convolution є узагальненням двовимірної згортки. У тривимірній згортки глибина фільтра менше глибини вхідного шару (розмір ядра менший за розмір каналу). Тому 3D-фільтр може рухатися у всіх 3-х напрямках (висота, ширина, канал зображення) (рис. 2.3). В результаті через поелементне множення і додавання на виході маємо одне число. Оскільки фільтр ковзає по тривимірному простору, вихідні числа також розташовуються в тривимірному просторі. В результаті виходять 3D-дані [9].

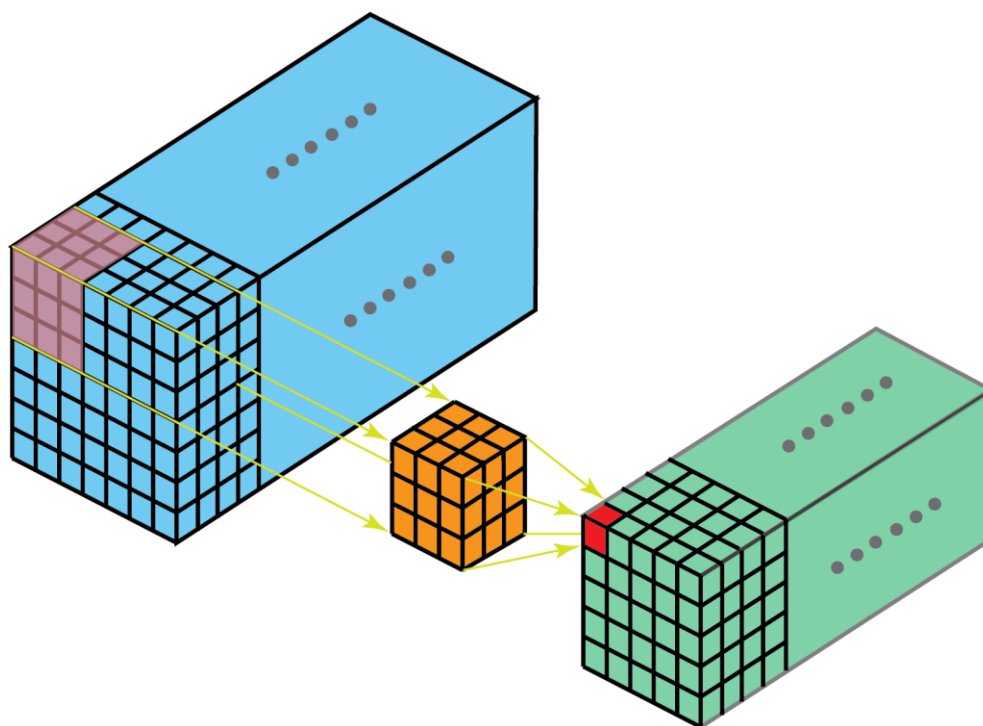


Рисунок 2.3 – Набір ядер 3D Convolution

У той час як двовимірні згортки кодують просторові відносини об'єктів у двовимірному просторі, тривимірні згортки можуть описувати просторові відносини об'єктів в тривимірному просторі. Здатність

навчатися тривимірним взаємозв'язкам корисна для таких завдань як тривимірна сегментація/реконструкція біомедичних зображень, наприклад, КТ і МРТ.

2.1.2 3D U-Net

U-Net складається з блоку, що звужує (енкодера), та блоку, що розширює (декодера). Енкодер є типовою архітектурою CNN. Він складається з багаторазового застосування згорток, за кожною з яких слідує ReLU і max pooling зі страйдом. На кожному наступному етапі подвоюється кількість ядер. Кожен крок у декодері складається з підвищення дискретизації карти ознак із подальшою деконволюцією, яка вдвічі зменшує кількість ядер. Між останнім блоком енкодера та першим блоком декодера є вузьке місце – згортка розміром 1×1 ; для конкатенації блоків енкодера та декодера та забезпечення нелінійного зв'язку між ними.

Архітектура 3D U-Net (рис. 2.4) тривимірна модифікація U-Net.

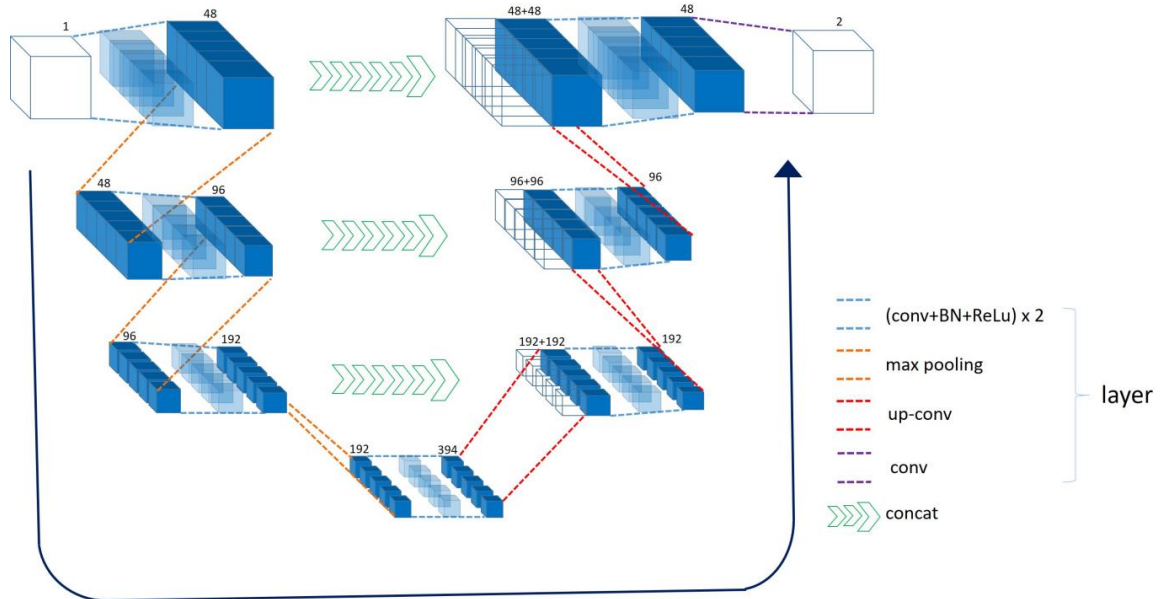


Рисунок 2.4 – приклад архітектури 3D U-Net [11]

Відмінність 3D U-Net полягає в тому, що блоці, що звужує, (енкодері) по два шара згортки з ядрами розміру $3 \times 3 \times 3$, за якими слідує функція ReLU, потім йде max pooling розміром $2 \times 2 \times 2$ з кроком 2 у кожному вимірі. У блоці розширення (декодері) спочатку йдуть шари розгортки з ядром розмір якого $2 \times 2 \times 2$ та кроками по два в кожному із вимірів, потім слідує по два шара згортки з ядром $3 \times 3 \times 3$, за якими слідує функція активації ReLU. У вузькому місці шар згортки $1 \times 1 \times 1$ зменшує кількість вихідних каналів до кількості ярликів [10].

2.2 Пропозиція адаптивних варіантів активаційних функцій

Для уникнення проблеми так званого градієнта, що зникає та вибухають, в архітектурах сімейства U-Net використовують сімейство активаційних функцій – rectified unit. Однак ці функції не задовольняють умовам класичної універсальної теореми апроксимації через свою необмежену вертикальну ось [7, 12], тому для забезпечення необхідної якості апроксимації необхідно збільшити кількість прихованих шарів, що веде до збільшення обчислювальної складності мережі та зниження швидкості процесу навчання. Відповідно, доцільно ввести до розгляду адаптивні активаційні функції, параметри яких можна налаштовувати в процесі навчання, як це роблять ваги нейронної мережі.

Досліджуваними модифікаціями є адаптивні функції активації, які дозволяють зменшити обсяг навчального набору даних та збільшити швидкість навчання нейронної мережі, зважаючи на те, що параметри цих функцій налаштовуються паралельно із синаптичними вагами DNN в онлайн-режимі. Запропонований підхід може бути застосований у глибоких згорткових нейронних мережах (CNN) в умовах малого обсягу навчальних наборів даних.

2.2.1 Адаптивний PReLU

Введемо до розгляду адаптивну модифікацію PReLU у вигляді [12]:

$$f(x) = \begin{cases} a^L u(k), & \text{якщо } u(k) \leq 0 \\ a^R u(k), & \text{якщо } u(k) > 0 \end{cases} \quad (2.1)$$

де $u(k)$ – значення з виходу шару (внутрішній сигнал активації);

a^L – адаптивний параметр, який визначає швидкість насичення PReLU при $x \leq 0$;

a^R – адаптивний параметр, який визначає швидкість насичення PReLU при $x > 0$.

Навчання нейрона з AdPReLU має такий вигляд:

Якщо $w_j^T(k-1)x(k) > 0$

$$a_j^R(k) = a_j^R(k-1) + (r_a^R(k))^{-1} \times (y_j(k) - a_j^R(k-1)u_j(k))u_j(k) \quad (2.2)$$

$$r_a^R(k) = \beta r_a^R(k-1) + u_j^2(k), \quad (2.3)$$

$$w_j(k) = w_j(k-1) + (r^R(k))^{-1} \times (y_j(k) - a_j^R(k)w_j^T(k-1)x(k))a_j^R(k)x(k), \quad (2.4)$$

$$r^R(k) = \beta r^R(k-1) + (a_j^R(k))^2 \|x(k)\|^2, \quad (2.5)$$

де a^R – адаптивний параметр, який визначає швидкість насичення PReLU при $x > 0$;

$u_j(k)$ значення з виходу шару (внутрішній сигнал активації) j -го нейрона в період часу $k=1, 2, \dots, N$;

w_j вектор синаптичних ваг;

$0 \leq \beta \leq 1$ – коефіцієнт забування.

Інакше:

$$a_j^L(k) = a_j^L(k-1) + (r_a^L(k))^{-1} \times (y_j(k) - a_j^L(k-1)u_j(k))u_j(k), \quad (2.6)$$

$$r_a^L(k) = \beta r_a^L(k-1) + u_j^2(k), \quad (2.7)$$

$$w_j(k) = w_j(k-1) + (r^L(k))^{-1} \times (y_j(k) - a_j^L(k)w_j^T(k-1)x(k))a_j^L(k)x(k), \quad (2.8)$$

$$r^L(k) = \beta r^L(k-1) + (a_j^L(k))^2 \|x(k)\|^2, \quad (2.9)$$

де a^L – адаптивний параметр, який визначає швидкість насичення PReLU при $x \leq 0$;

$u_j(k)$ значення з виходу шару (коефіцієнт забування) j -го коефіцієнт забування $k=1, 2, \dots, N$;

w_j вектор синаптичних ваг;

$0 \leq \beta \leq 1$ – коефіцієнт забування.

2.2.2 Адаптивний ELU

Аналогічно до AdPReLU введено до розгляду адаптивну модифікацію ELU у вигляді

$$f(x) = \begin{cases} a^L(e^{u(k)} - 1), & \text{якщо } u(k) < 0 \\ a^R u(k), & u(k) \geq 0 \end{cases} \quad (2.10)$$

де $u(k)$ – значення з виходу шару (внутрішній сигнал активації);

a^L – адаптивний параметр, який визначає швидкість насичення AdELU при $x < 0$;

a^R – адаптивний параметр, який визначає швидкість насичення AdELU при $x \geq 0$.

Розглянемо процес налаштування AdELU з одним нейроном. Якщо $u(k) < 0$:

$$a^L(k) = a^L(k-1) + \eta(k)(y(k) - a^L(k-1)(e^{u(k)} - 1))(e^{u(k)} - 1), \quad (2.11)$$

де a^L – адаптивний параметр, який визначає швидкість насичення ELU при $x < 0$;

$u(k)$ – значення з виходу шару (внутрішній сигнал активації);

$\eta(k)$ – параметр швидкості навчання (параметр кроку).

Інакше:

$$a^R(k) = a^R(k-1) + \eta(k) \left(y(k) - a^R(k-1)(e^{u(k)} - 1) \right) \times (e^{u(k)} - 1), \quad (2.12)$$

де a^R – адаптивний параметр, який визначає швидкість насичення ELU при $x < 0$;

$u(k)$ – значення з виходу шару (внутрішній сигнал активації);

$\eta(k)$ – параметр швидкості навчання (параметр кроку).

2.2.3 Адаптивний SoftPlus

Також введемо адаптивний варіант SoftPlus:

$$f(x) = \frac{\ln(1+e^{\alpha kx})}{\alpha k}, \quad (2.13)$$

де k – параметр різкості та адаптивний параметр SoftPlus.

На відміну від AdPReLU та AdELU AdSoftPlus має один адаптивний параметр – k . З збільшенням цього параметра вид функції SoftPlus буде наближатися до виду функції ReLU

3 ПРОГРАМНА РЕАЛІЗАЦІЯ, ОБРОБКА ДАНИХ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Опис бібліотек та платформ

У цьому розділі описані основні бібліотеки та сервіси, які були використані для реалізації програмного забезпечення цього дослідження в рамках магістерської кваліфікаційної роботи.

3.1.1 Python

Python – це високорівнева, інтерпретована, інтерактивна та об'єктно-орієнтована мова програмування:

- Python інтерпретується – код Python обробляється під час виконання програми інтерпретатором. Програму не потрібно компілювати перед виконанням;

- Python інтерактивний – можна слідувати за підказками Python і безпосередньо взаємодіяти з інтерпретатором у процесі написання програми;

- Python є об'єктно-орієнтованим – Python підтримує об'єктно-орієнтовану парадигму програмування.

Синтаксис Python розроблений таким чином, щоб забезпечити легкість читання, наближеного до легкості читання англійської мови. У ньому часто використовуються англійські слова як службові «символи», тоді як в інших мовах використовується пунктуаційні знаки, й у ньому менше синтаксичних конструкцій, ніж в інших мовами [13].

3.1.2 PyTorch

PyTorch – бібліотека машинного навчання з відкритим кодом для

Python. Спочатку PyTorch був розроблений дослідницькою групою штучного інтелекту Facebook, а також Uber для ймовірнісного програмування.

Розробники PyTorch переробили та реалізували Torch на Python, використовуючи при цьому ті ж бібліотеки на C для бекенд-коду з метою зберегти ефективність виконання Python. Вони також зберегли апаратне прискорення на базі GPU, а також можливість розширюваності, які були реалізовані в Torch на базі Lua.

Основні особливості PyTorch [14]:

- простий інтерфейс – PyTorch пропонує інтуїтивно зрозумілий у використанні API, що робить простим в експлуатації;
- так як інтерфейс PyTorch реалізований на Python, це дозволяє модулі реалізовані за допомогою PyTorch інтегрувати зі стеком data science Python'a;
- PyTorch підтримує динамічні обчислювальні графи. Таким чином користувач може змінювати їх під час виконання. Це дуже корисно, коли розробник не має уявлення, скільки пам'яті потрібно для створення моделі нейронної мережі.

3.1.3 NiBabel

NiBabel – це бібліотека, яка надає доступ для читання/запису деяких поширених форматів медичних та нейровізуальних файлів. NiBabel підтримує колекцію форматів файлів нейровізуалізації, що постійно зростає. Кожен формат файлу має свої особливості та нюанси, про які необхідно взяти до уваги, щоб отримати з нього максимальну користь. Для цього NiBabel пропонує як високорівневий незалежний від формату доступ до нейророзображень, так і API з різними рівнями специфічного для доступу до всієї доступної інформації в конкретному форматі файлу [15].

Скани в наборі даних BraTS20 мають формат .nii.gz. Ця бібліотека

була використана для читання сканів з BraTS20, а доступ до матриці переходів дозволив виконати первинну обробку даних.

3.1.4 Seaborn

Seaborn – бібліотека візуалізації даних на Python. Вона побудована на базі matplotlib і тісно інтегрована із структурами даних pandas.

Seaborn допомагає досліджувати та інтерпретувати дані. Його орієнтований на набір даних декларативний API дозволяє зосередитись на тому, що означають різні елементи графіків, а не на деталях того, як їх малювати [16].

У цій кваліфікаційній роботі Seaborn використовується для реалізації модуля візуалізації, який використовується для аналізу даних та результатів екскрементів.

3.1.5 NumPy

NumPy – пакет для наукових обчислень з Python. Деякі з переваг NumPy:

- математичні операції над об'єктами ndarray в NumPy виконуються в 50 разів швидше, ніж ітерації над списками в Python з використанням циклів. Підвищення ефективності в першу чергу пов'язано з тим, що NumPy зберігає елементи масиву в упорядкованому в єдиному місці пам'яті, шляхом того, що всі елементи мають однаковий тип, усуває надмірність;
- містить вбудовані корисні функції для розв'язання задач лінійної алгебри, перетворення масивів та матричної математики;
- компактний синтаксис, для виконання більшості математичних операцій потрібно менше рядків коду, ніж під час роботи зі списками Python.

3.2 Дослідження даних

Як навчальні дані для експериментів був обраний набір даних BRaTS20. Дані BRaTS20 являють собою обширні мультиінституційні передопераційні мультимодальні МРТ-скани гліобластоми (GBM/HGG) і гліоми низького класу (LGG) з патологоанатомічно підтвердженим діагнозом.

3.2.1 МРТ-модальності

МРТ-апарати мають різні конфігурації та роблять зображення з різною інтенсивністю. Наприклад, коли використовуються різні конфігурації МРТ (для прикладу, напруженість магнітного поля) і протокол збірки (дозвіл вокселя, сила градієнта, значення поля зору і т.д.), така варіативність робить виявлення пухлини важким завданням.

МРТ може показувати різні контрасти тканин через різні послідовності імпульсів, тому МРТ широко використовують для візуалізації людського мозку (рис. 3.2). Модальності МРТ комбінують для отримання мультимодальних зображень, що дає більше інформації про нерегулярну форму пухлини, яку важко локалізувати за допомогою одного методу. Ці методи включають T1-зважений МРТ (T1), T1-зважений МРТ з підвищенням контрастності (T1c), T2-зважений МРТ (T2) і T2-зважений МРТ з відновленням інверсії рідини, що ослаблена (T2-Flair). Ці мультимодальні дані містять інформацію, яка може бути використана для сегментації пухлини зі значним поліпшенням продуктивності. На рисунку 3.1 показаний осьовий зріз з чотирьох стандартних послідовностей для пацієнта з гліомою, включаючи області пухлини, намальовані вручну.

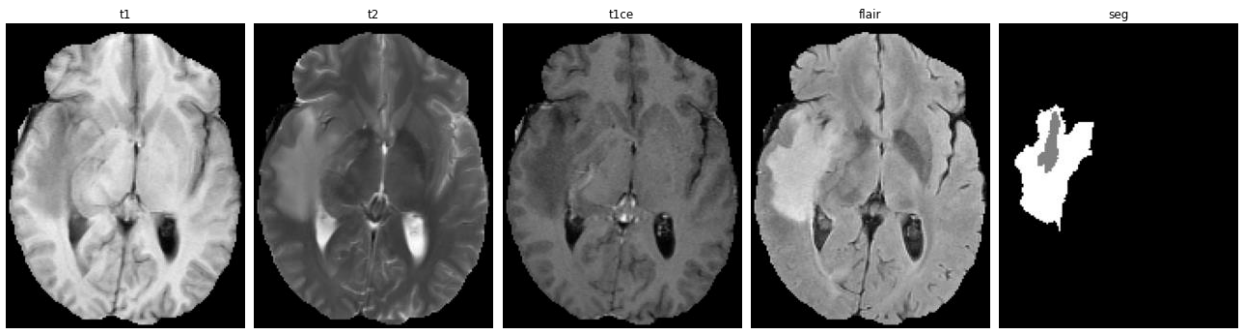


Рисунок 3.1 – МРТ-зріз пацієнта з гліомою

Кожна з модальностей має такі властивості:

- T1-зважування найбільш часто використовують для структурного аналізу; воно також дозволяє легко анотувати здорові тканини;
- на зображеннях з посиленням контрастом T1CE межі пухлини виглядають яскравіше, оскільки контрастна речовина накопичується там через порушення бар'єру кров-мозок в області проліферативної пухлини. У цій послідовності можна легко відрізнити некротичну і активну пухлинну області;
- на T2-зваженому МРТ область набряку і навколишня пухлина виглядає яскравіше;
- FLAIR – це спеціальна послідовність, яка допомагає відокремити область набряку від спинномозкової рідини.

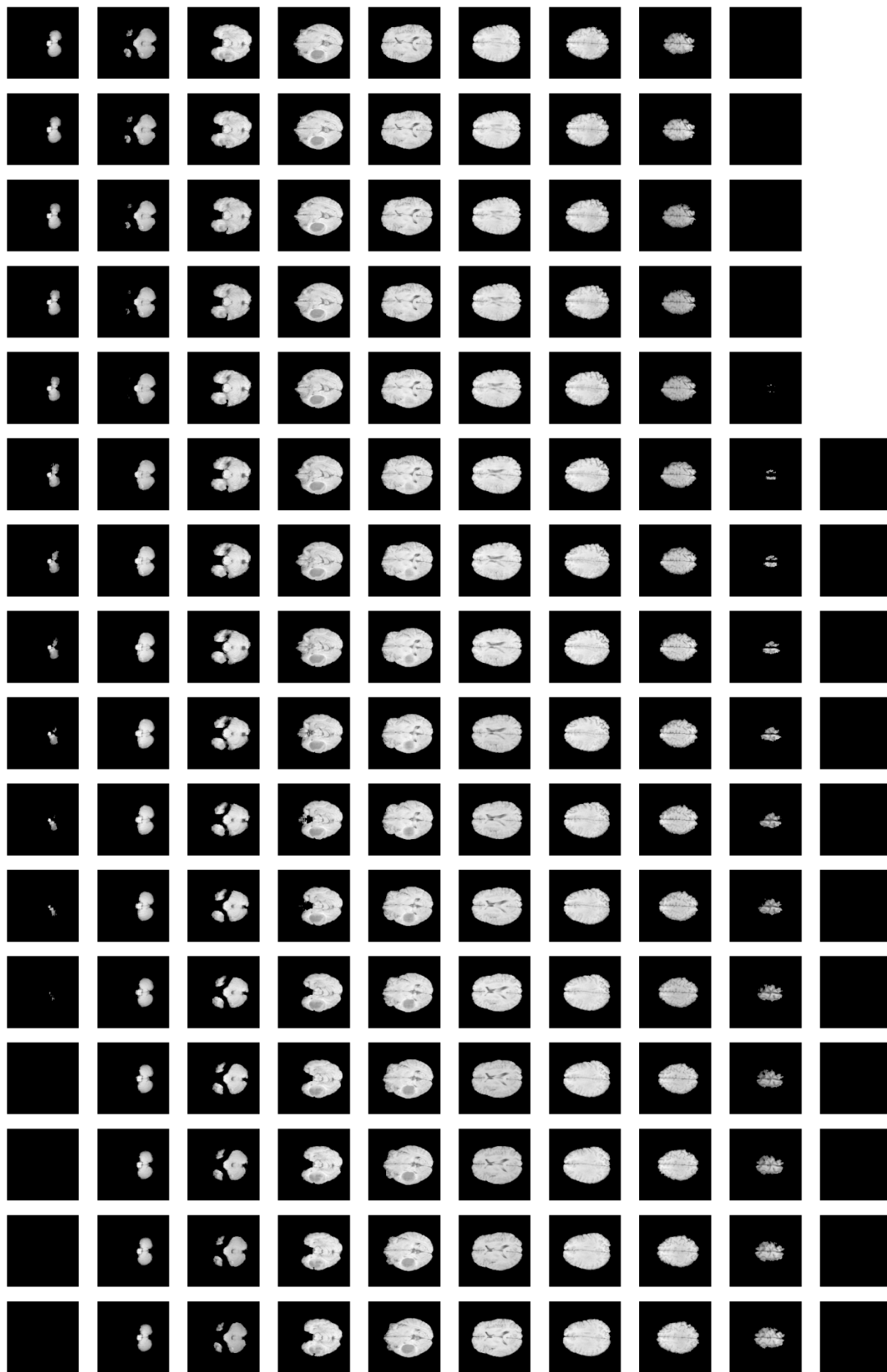


Рисунок 3.2 – Послідовність сканів пацієнта з набору даних BraTS20 модальністю T1

3.2.2 Первина обробка даних

Хоча відсутність нормалізованої шкали інтенсивності МРТ не прямо впливає на клінічну діагностику лікарями, це має негативний ефект методів автоматичної обробки зображень, такими як сегментація, класифікація, метод кількісної оцінки, які сильно залежать від інформації про інтенсивність (рис. 3.3). Ця проблема, зокрема, є більш помітною у великомасштабних багатосторонніх нейровізуалізаційних дослідженнях, що включають значну кількість суб'єктів, що скануються за допомогою різних типів сканерів і параметрів сканування (що схоже з набір даних BRATS, зображення, для якого були зроблені з різними налаштуваннями та в різних лікарнях).

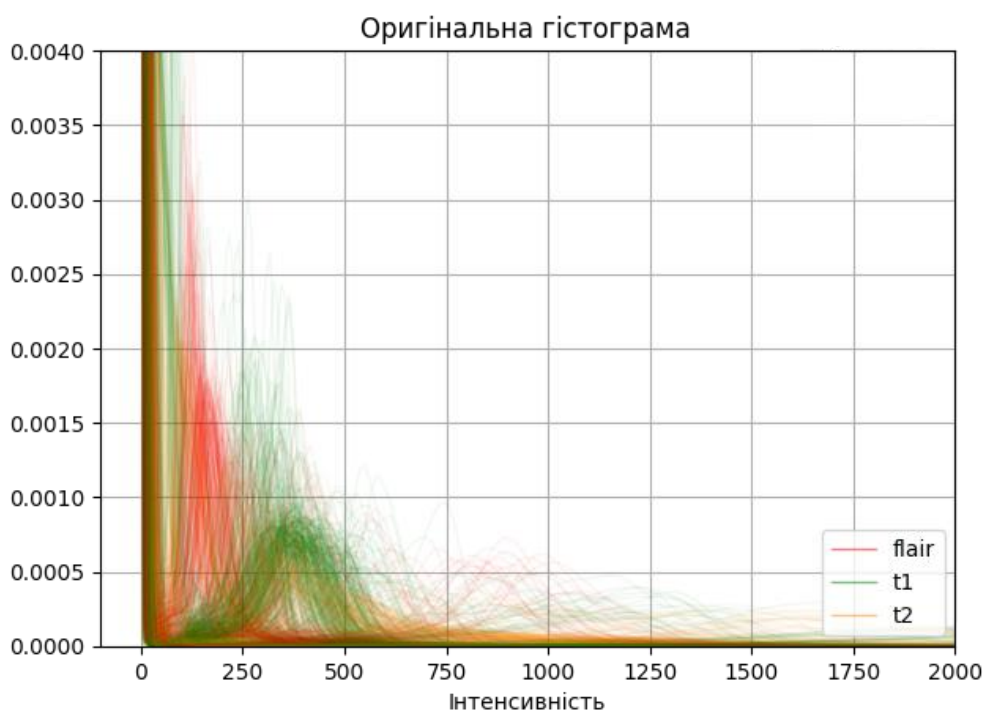


Рисунок 3.3 – Інтенсивність оригінальний даних BraTS20

Нормалізація інтенсивності зображень має вирішальне значення для вивчення прогресування захворювання в багатьох клінічних дослідженнях.

Однак зображення з різних сканерів або з різними параметрами отримання можуть мати великі відмінності в інтенсивності, що впливає на результати аналізу зображень. Тому нормалізація інтенсивності МРТ-сканів, спрямована на корекцію варіацій, яка залежить від сканера, необхідна точного аналізу МРТ [17].

Застосуємо z-стандартизацію (рис. 3.4) для полегшення навчання нейронної мережі. Z-стандартизація означає, що наші вихідні дані матимуть нульову середню та одиничну дисперсію.

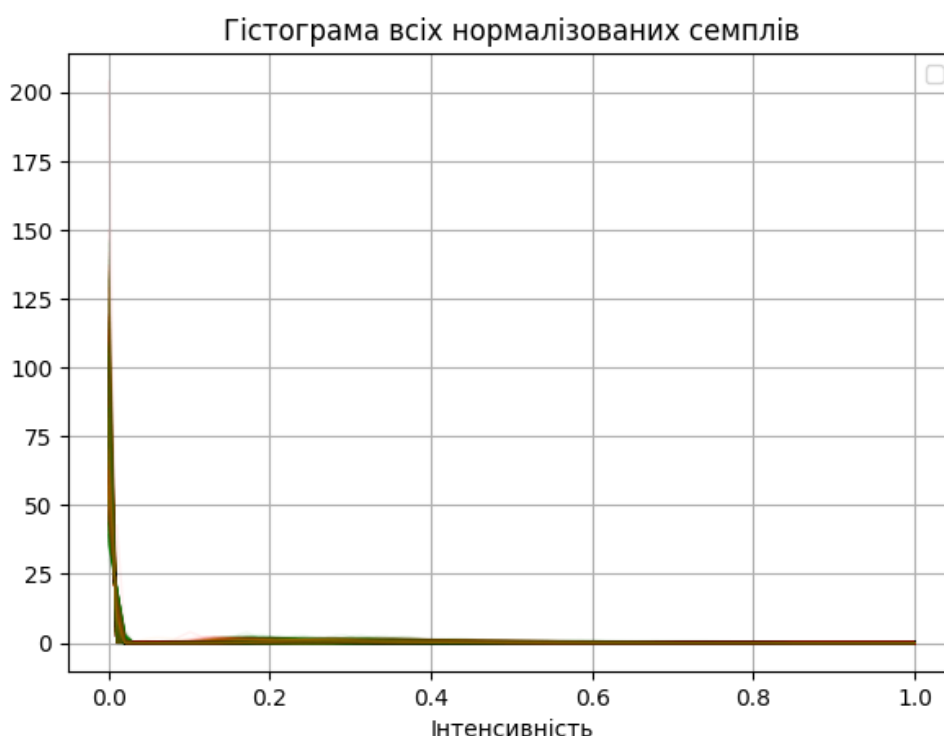


Рисунок 3.4 – Інтенсивність даних після z-стандартизації

Так само розмір всіх сканів був стиснутий до $78 \times 60 \times 60$ з метою економії RAM та зростання швидкості навчання.

3.3 Опис умов експериментів

Для проведення експериментів були обрані такі параметри:

- на вхід нейронні мережі приймають скани розмірністю [4, 60, 60, 78];
- один семпл складається з таких модальностей T1, T1CE, T2, Flair;
- на виході очікується розмірністю [3, 60, 60, 78], де 3 це кількість вимірювань, кожен з яких відповідає за один з класів: GD-пухлина, що розширюється, перитуморальний набряк (ED), некротичне і не розширюється ядро пухлини (NCR/ NET);
 - розмір батча – 2;
 - 10 епох;
 - як алгоритм оптимізації було обрано комбінацію Dice Loss + Cross Entropy;
 - learning rate=0.0005;
 - learning rate множиться на 0.1 кожного разу, як функція втрат валідаційного набору даних не покращую своє мінімальне значення протягом 2-х епох;
 - початкові параметри для AdPReLU та AdELU $\alpha=0.1$, $\alpha_r=1$;
 - початкові параметри AdSoftplus $k=1$.

3.4 Аналіз експериментів

3.4.1 Метрики

За результатами навчання 3D U-Net з різними активаційними функціями найвищі показники метрик Dice та Jaccard (рис. 3.5 – 3.13) має модель навчена з AdPReLU, потім йде AdELU, за нею AdSoftPlus, і тільки після того запропонований, як частина архітектури, в оригінальній статті 3D U-Net не адаптивний ReLU.

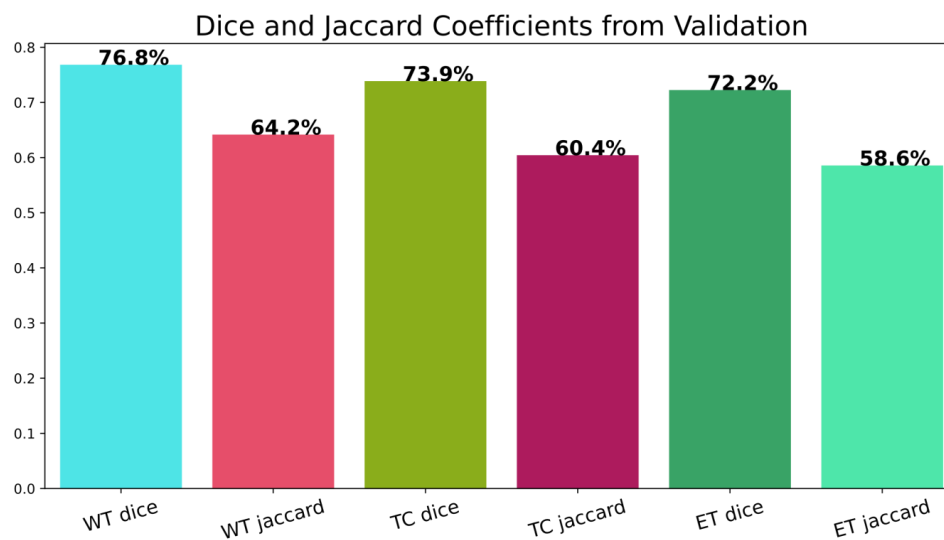


Рисунок 3.5 – Значення метрик Жассард та Дісе для моделі з активаційною функцією ReLU

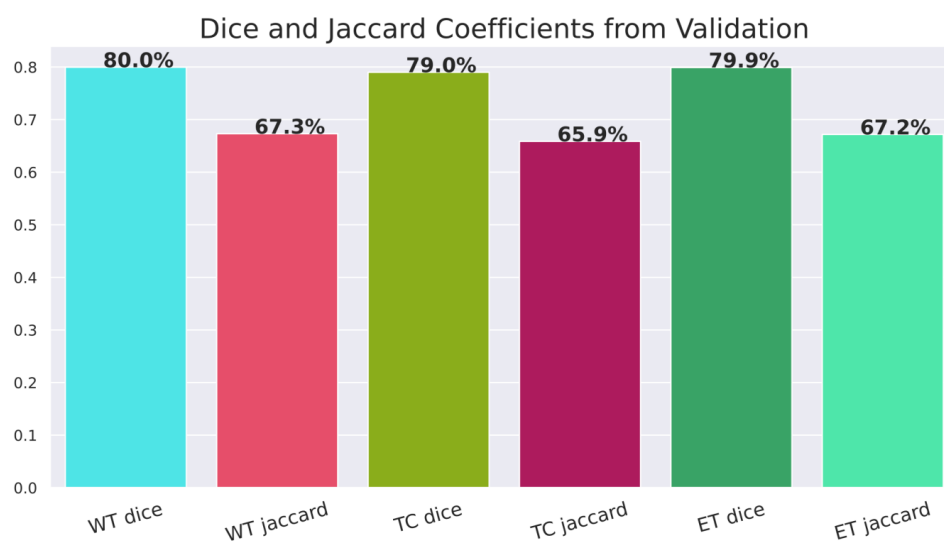


Рисунок 3.6 – Значення метрик Жассард та Дісе для моделі з активаційною функцією AdPreLU

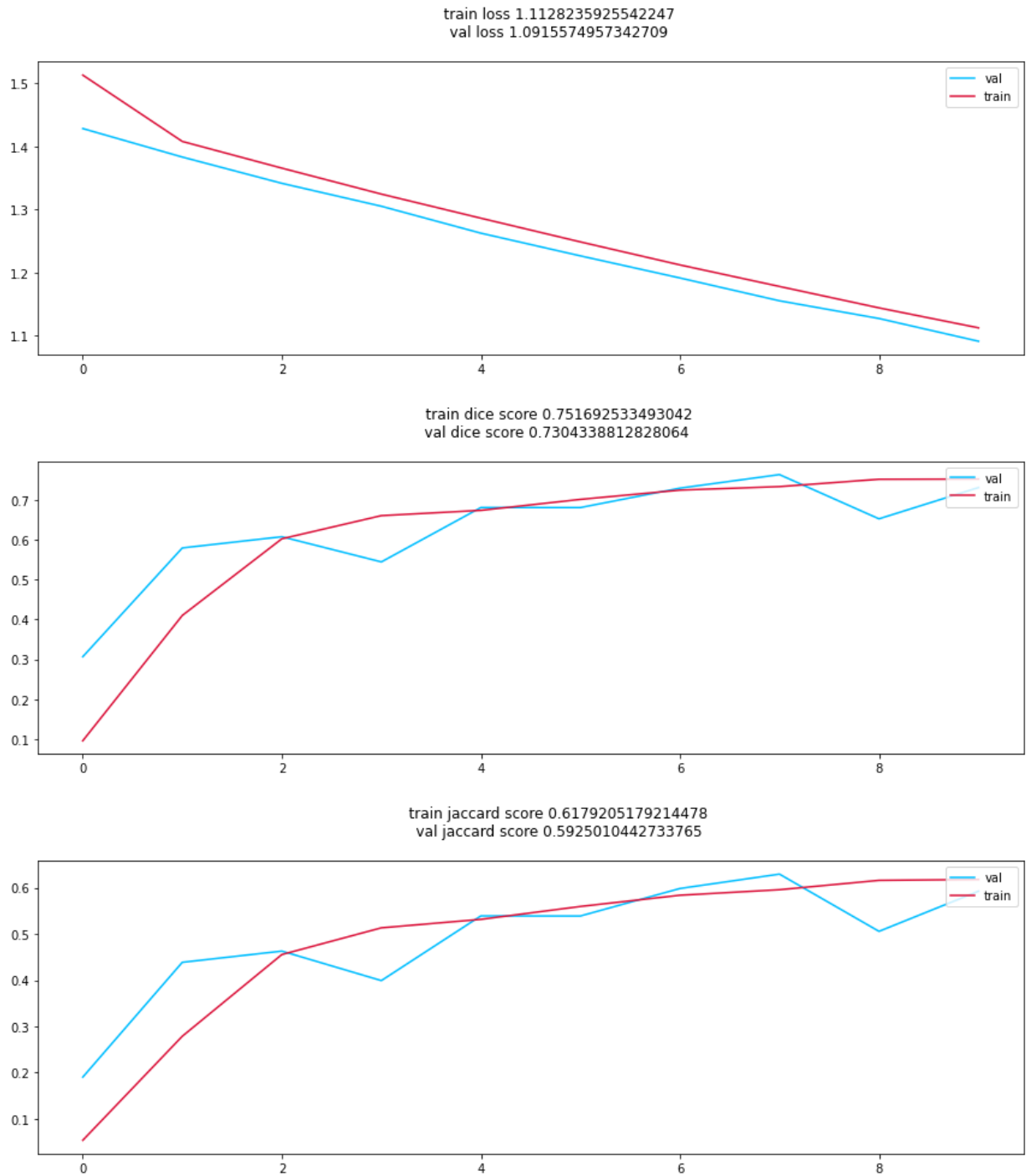


Рисунок 3.7 – Графік зміна функції втрат и метрик під час навчання моделі з активаційною функцією ReLU

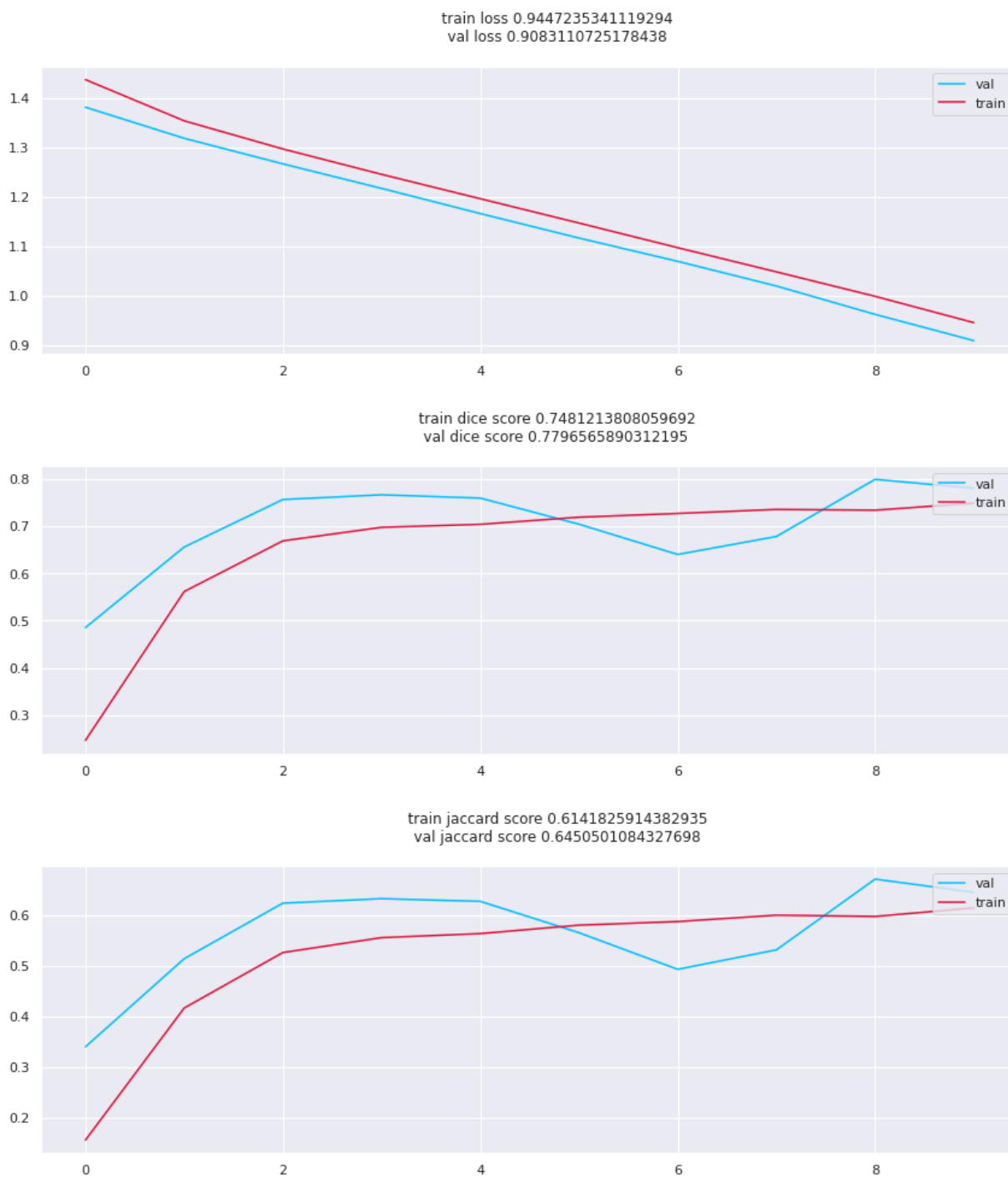


Рисунок 3.8 – Графік зміна функції втрат и метрик під час навчання моделі з активаційною функцією AdPReLU

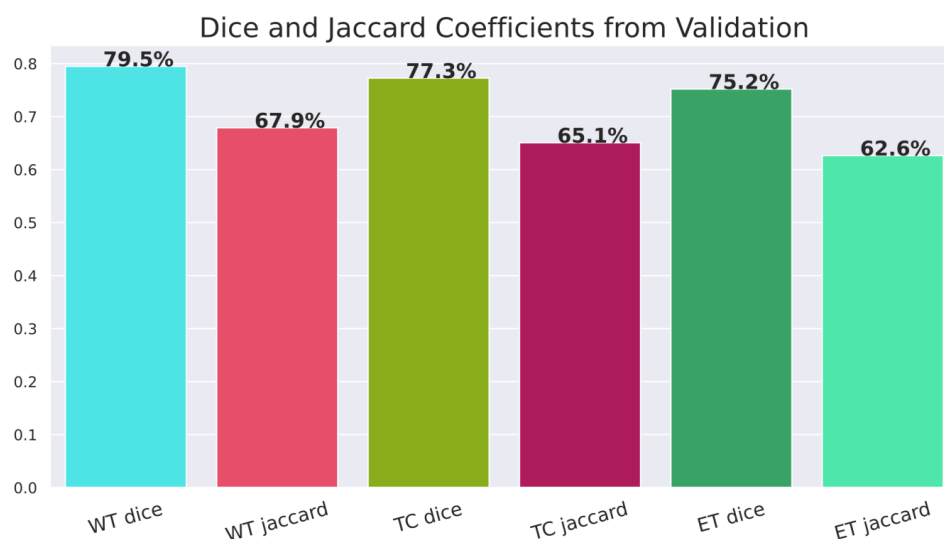


Рисунок 3.9 – Значення метрик Жассард та Дісе для моделі з активаційною функцією AdELU

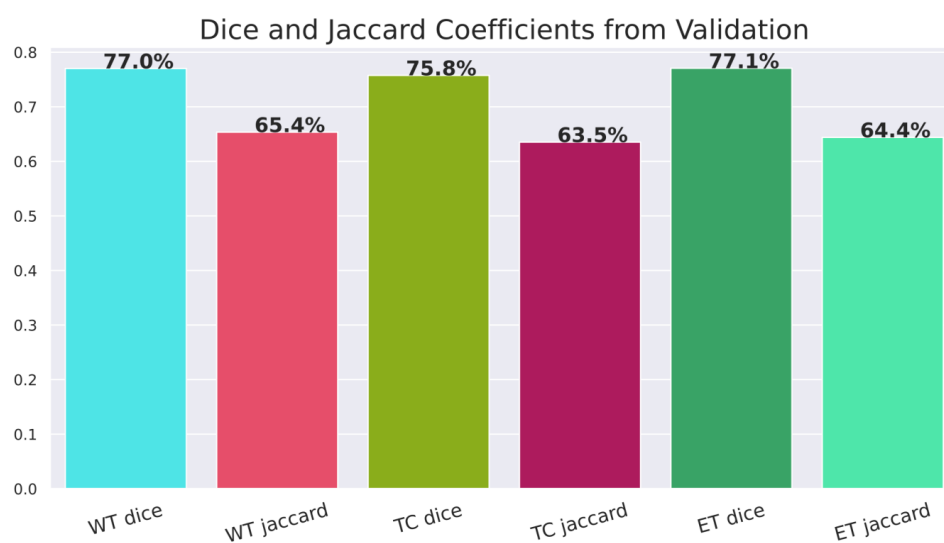


Рисунок 3.10 – Значення метрик Жассард та Дісе для моделі з активаційною функцією AdSoftPlus

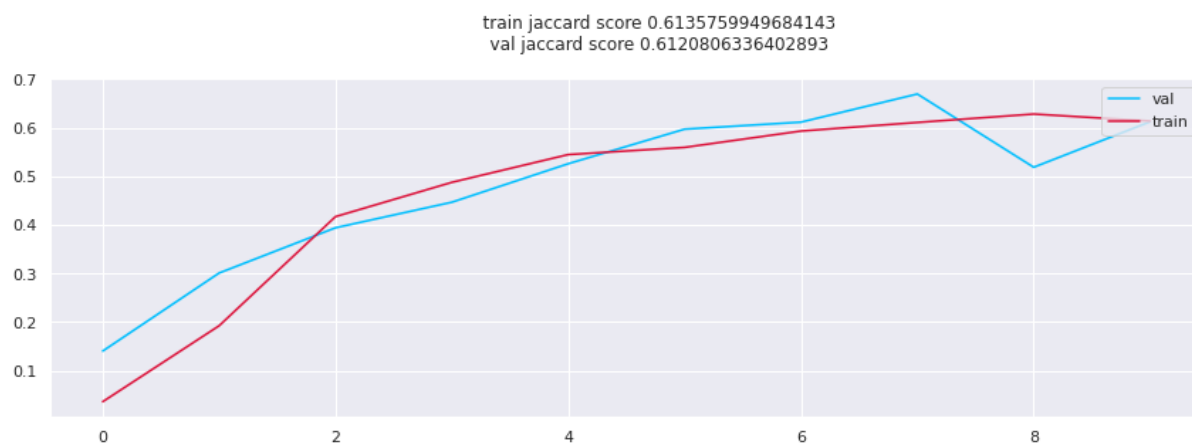
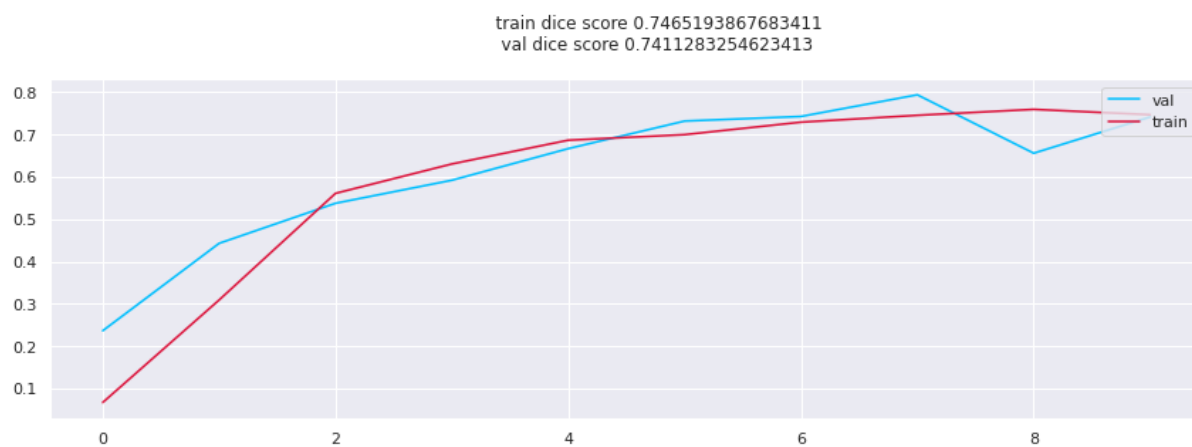
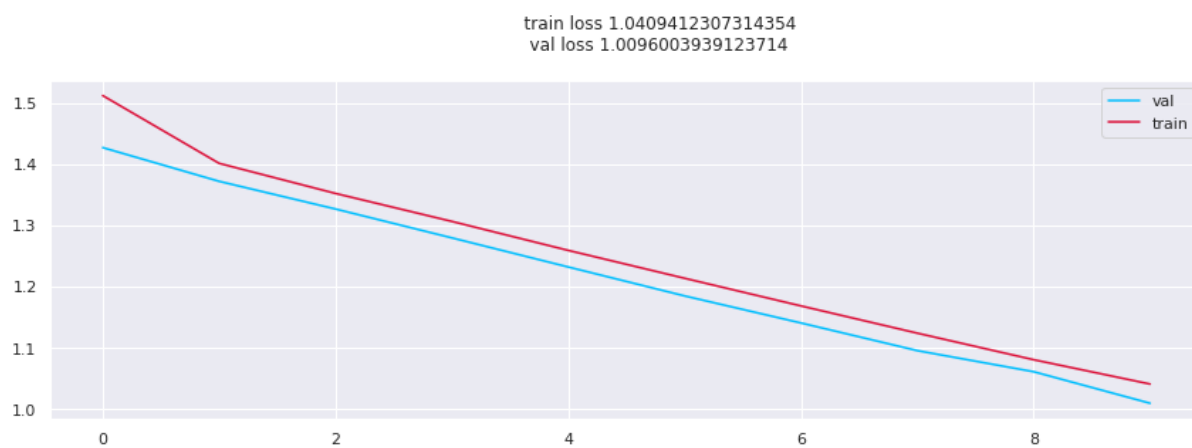


Рисунок 3.11 – Графік зміна функції втрат и метрик під час навчання моделі з активаційною функцією AdELU

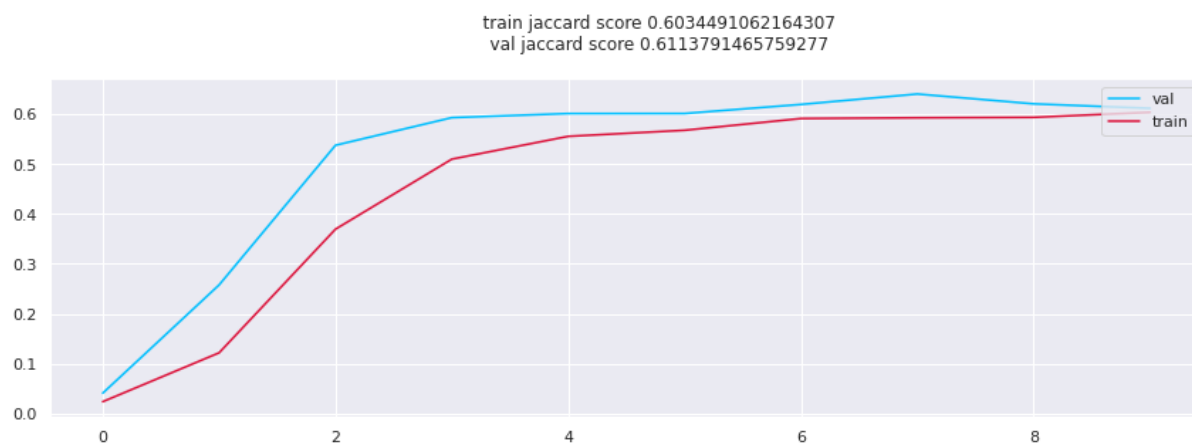
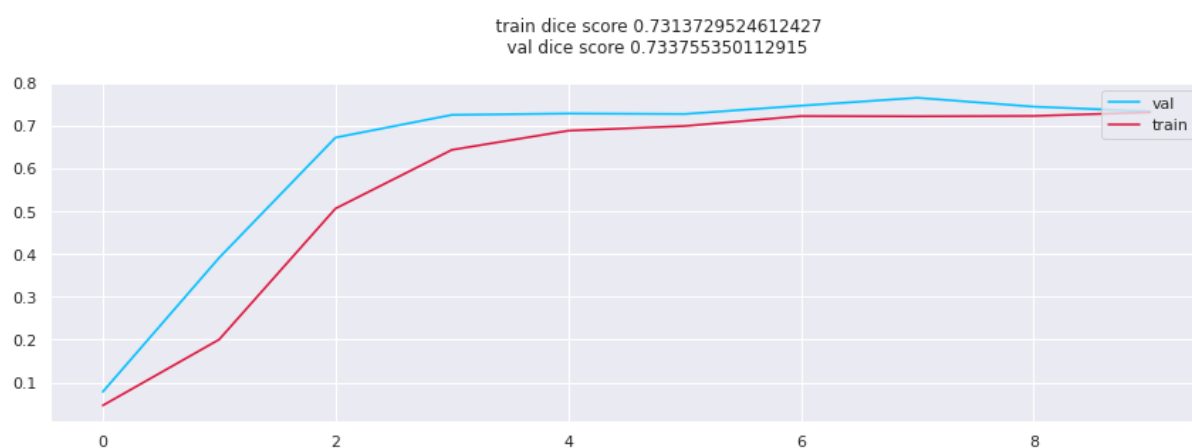
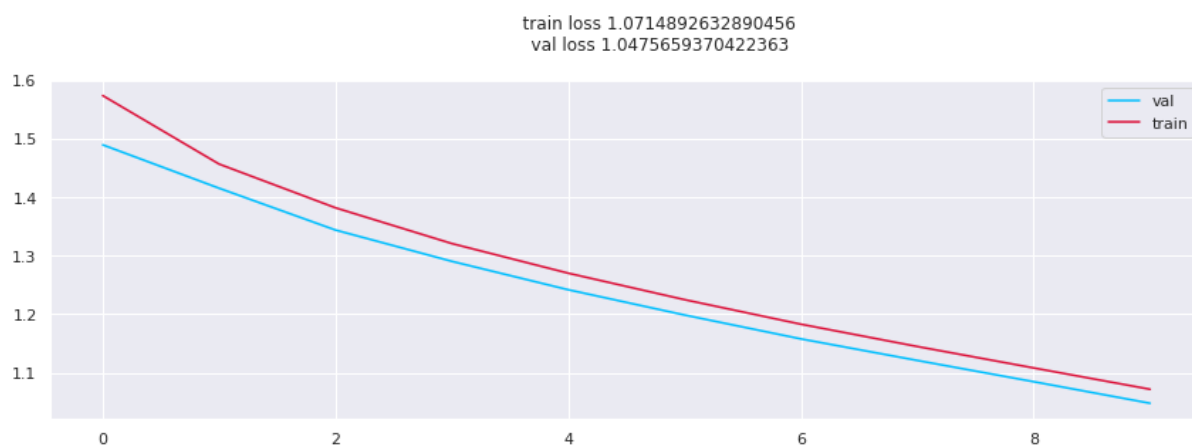


Рисунок 3.12 – Графік зміна функції втрат и метрик під час навчання моделі з активаційною функцією AdSoftPlus

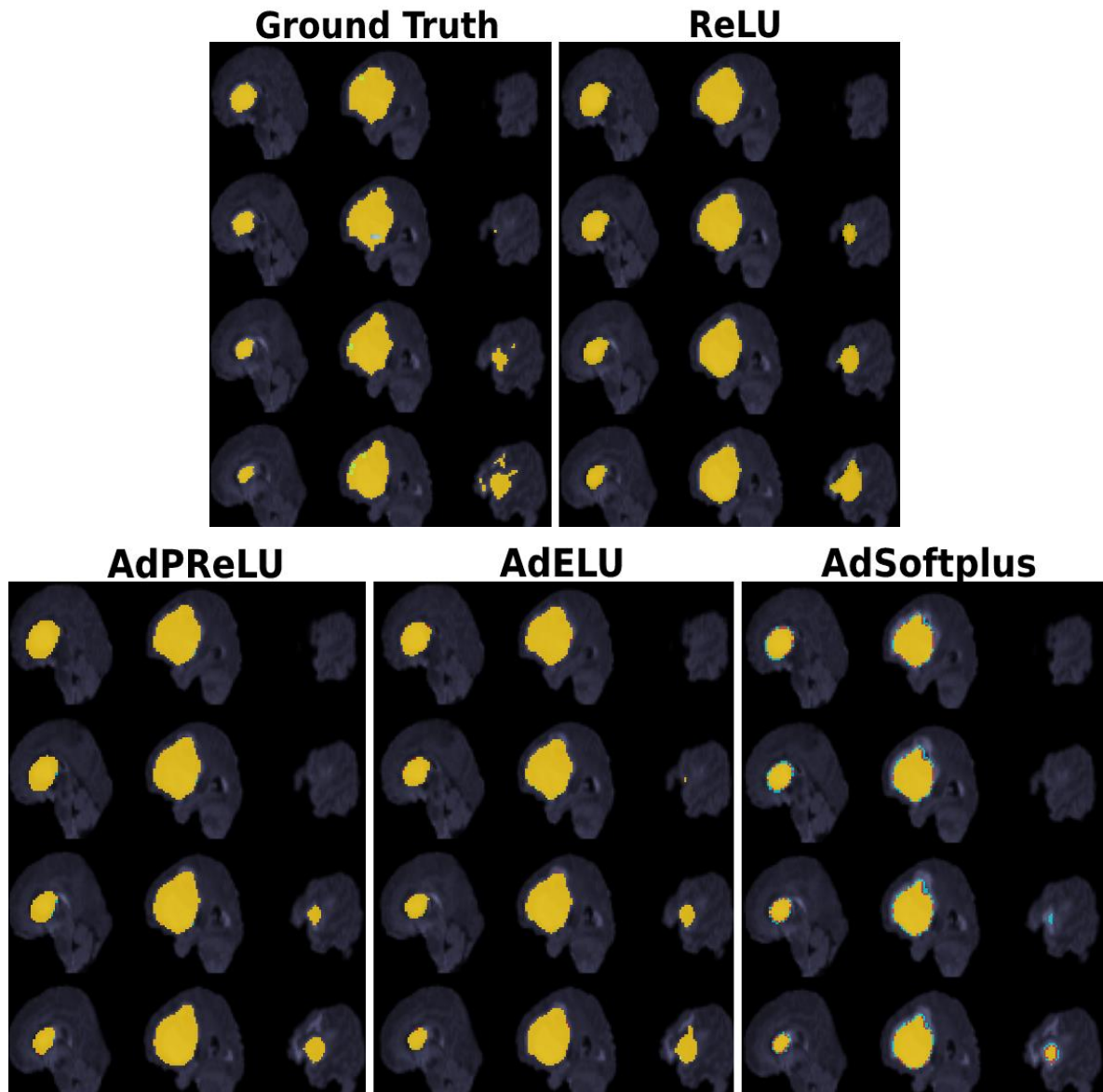


Рисунок 3.13 – Приклад прогнозів моделей

Показники метрик Jaccard і Dice залежать від заданого порога впевненості моделі, що на конкретній ділянці той чи інший клас. У даному випадку був використаний поріг 50% впевненості. Звичайно добре, коли розподіл прогнозів отриманих з виходу моделі мають такий вигляд, що отримуємо добре збалансовані показники TPR і TNR відносно порога 50%, але отримання такої моделі не входить в дане дослідження. У такому випадку також варто виміряти перфоменс моделі відносно метрик, які інваріантні до порога впевненості. Зважаючи на те, що дані BRaTS20 вкрай

не збалансовані через переважання фону на сканах, а не сегментаційної маски, замість часто застосовуваної ROC-кривої, побудовані Precision-Recall криві.

Precision-Recall крива менше схильна до некоректного вимірювання якості класифікації через не збалансованість даних. Через те, що Precision і Recall зосереджені на позитивному класі (клас, який у меншості) та не дбають про TFR (клас, який у більшості).

Найбільший показник Average Precision (AP=0.91) в AdPReLU, потім йдуть AdELU з AP=0.9 та ReLU з AP=0.89, і найгірший AP=0.87 в AdSoftPlus. Отримані показники AP (рис. 3.14 – 3.15) не сходяться з ранжуванням якості моделей відносно Jaccard і Dice.

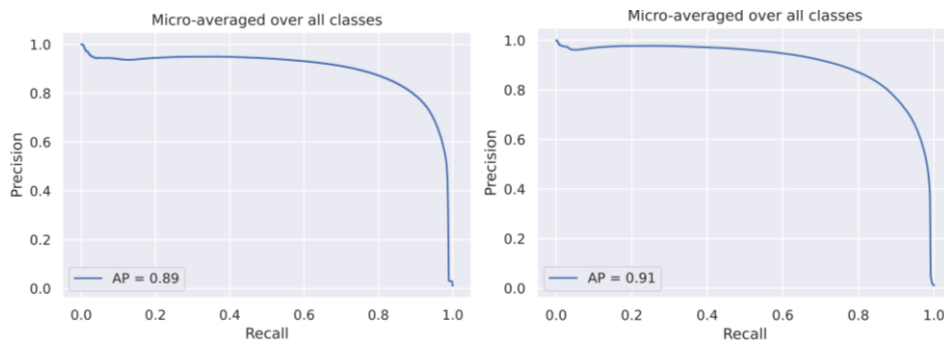


Рисунок 3.14 – Precision-Recall криві для ReLU (ліворуч) і AdPReLU (праворуч)

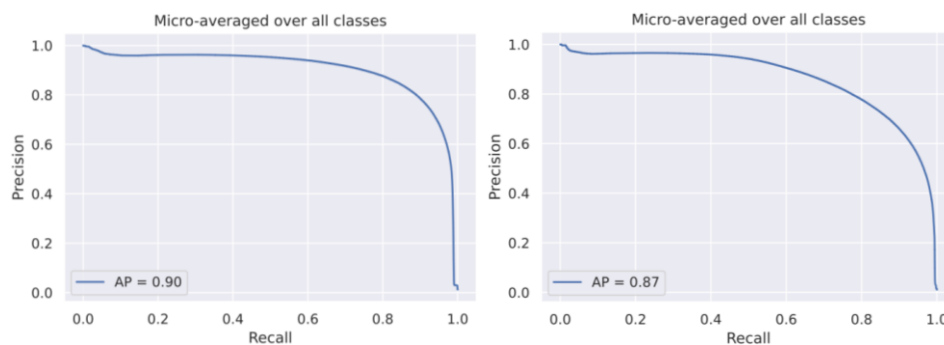


Рисунок 3.15 – Precision-Recall криві для AdELU (ліворуч) і AdSoftPlus (праворуч)

Чи можна сказати, що результат вимірювання AP AdPReLU, який обігнав AdELU на 0.01 пункту, а ReLU на 0.02 пункта, дійсно робить найбільш точні прогнози? Можна зробити таке припущення, але для перевірки чи є ця відмінність статистично значущою, добре провести статистичний тест гіпотези на моделях, у яких ідентичні всі параметри, крім випадкових початкових значень (random seed) або розбити дані на кілька частин (фолдів) та навчити по моделі кожну із них. Статистичний тест цієї гіпотези є добрим стартом для подальших досліджень.

3.4.2 Поведінка параметрів активаційних функцій під час навчання

Цікавим є те, що найбільш лабільними виявилися параметри активаційних функцій, які знаходяться ближче до виходу енкодера та декодера, через таку поведінку градієнта в енкодері, вихід якого знаходиться у середині U-Net, не можна сказати, що це ознака градієнта, що згасає. У цілому значення градієнтів досить малі (рис. 3.16, 3.17, 3.20, 3.22, 3.24, 3.26), щодо цього можна висунути дві взаємозалежні гіпотези: маленький градієнт наслідок відносно невеликої помилки за одну ітерацію (не епоху) так як батч має тільки 2 семпли й застосування методики накопичення градієнта з невеликим кроком для даного розміру батча (ваги оновлюються кожну 8-му ітерацію відносно суми градієнтів за ці 8-м ітерацій). Маленький градієнт є причиною невеликої зміни параметра (рис. 3.18, 3.19, 3.21, 3.23, 3.25, 3.27).

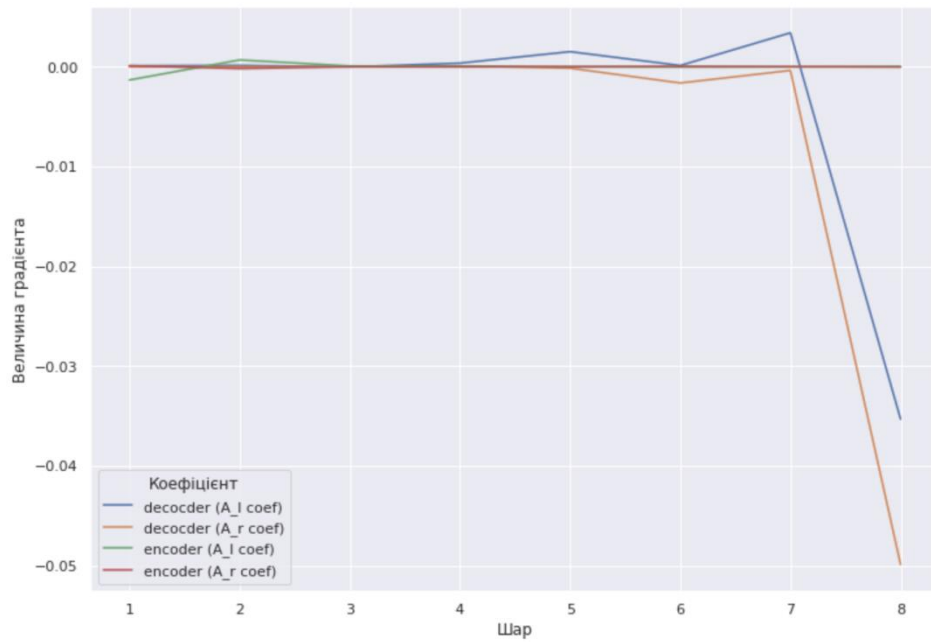


Рисунок 3.16 – Величина градієнта крізь шари для моделі з активаційною функцією AdPReLU під час 5-ї епохи

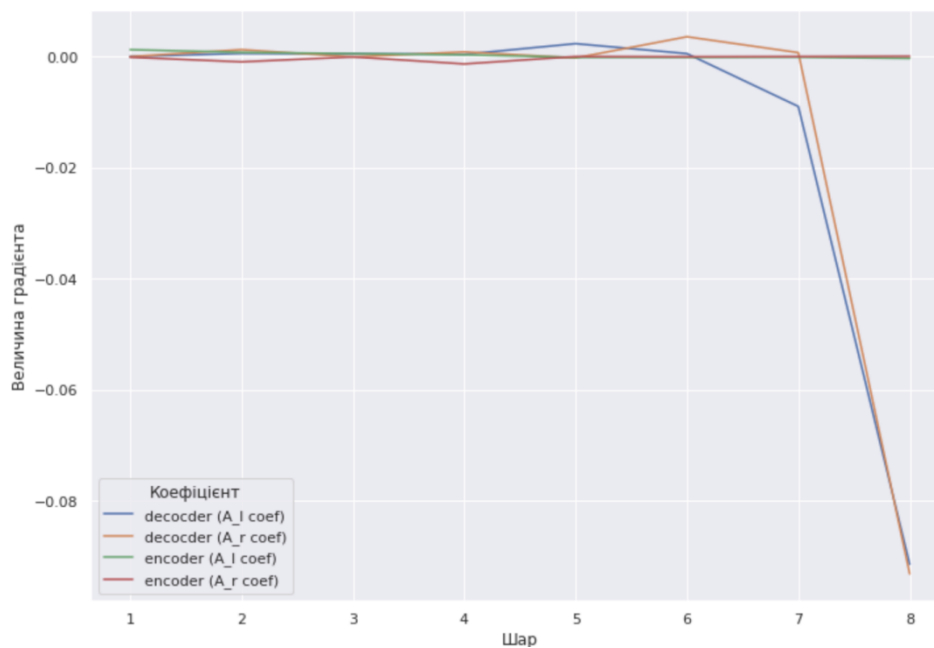


Рисунок 3.17 – Величина градієнта крізь шари для моделі з активаційною функцією AdPReLU під час 10-ї епохи

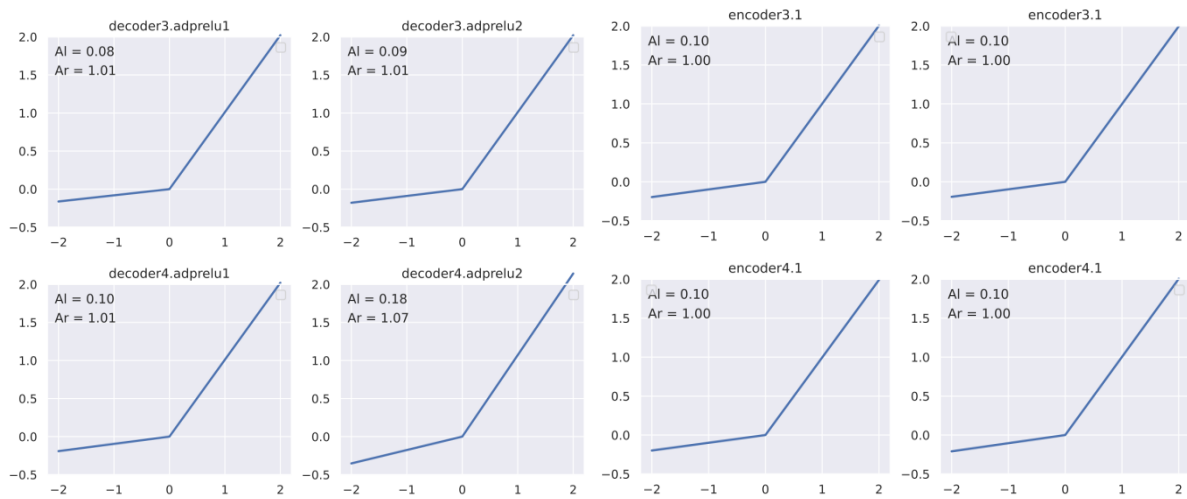


Рисунок 3.18 – Зміна параметрів функцій активації AdSoftPlus під час 5-ї епохи

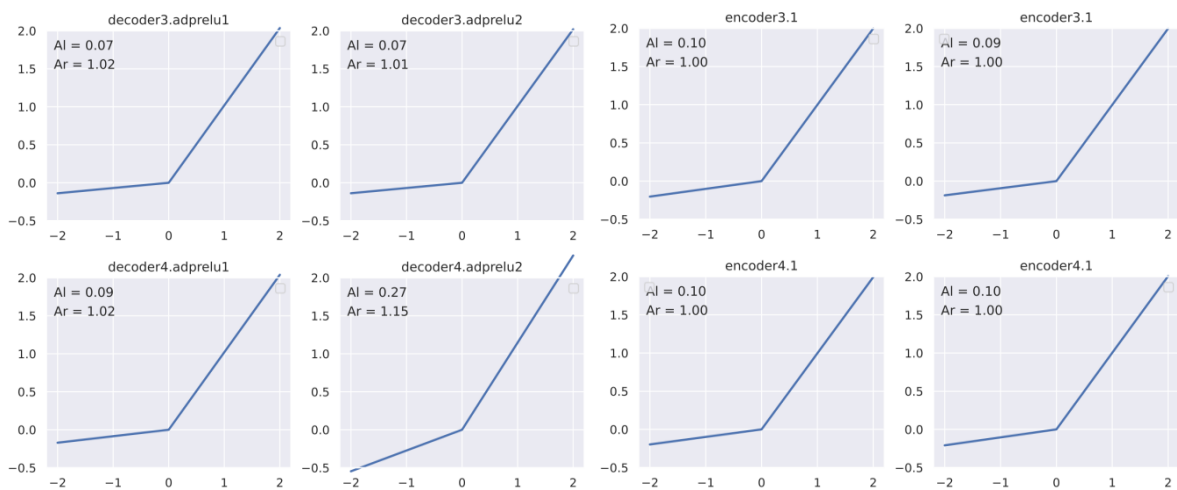


Рисунок 3.19 – Зміна параметрів функцій активації AdSoftPlus під час 10-ї епохи

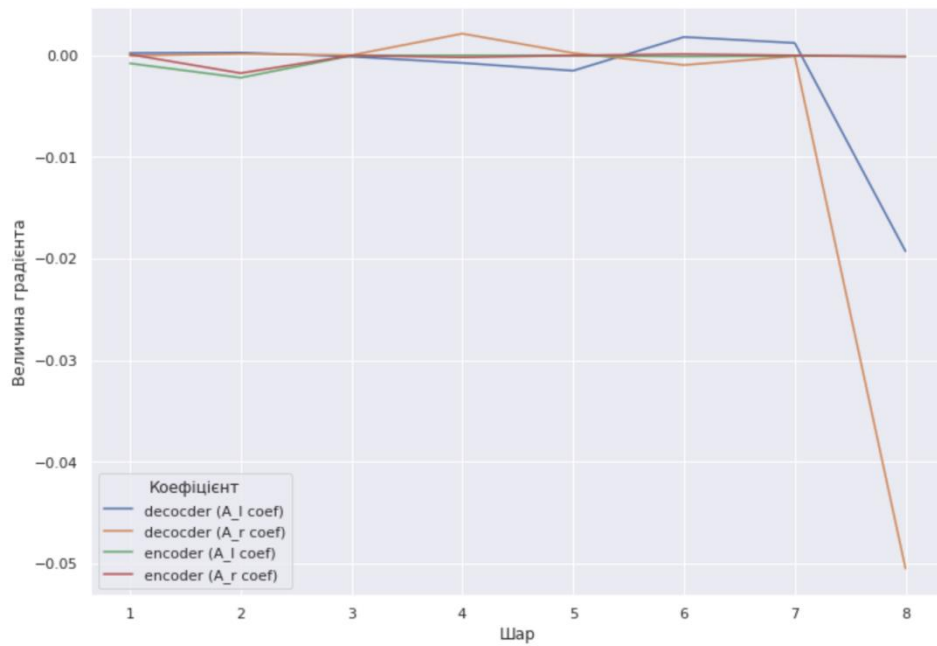


Рисунок 3.20 – Величина градієнта крізь шари для моделі з активаційною функцією AdELU під час 5-ї епохи

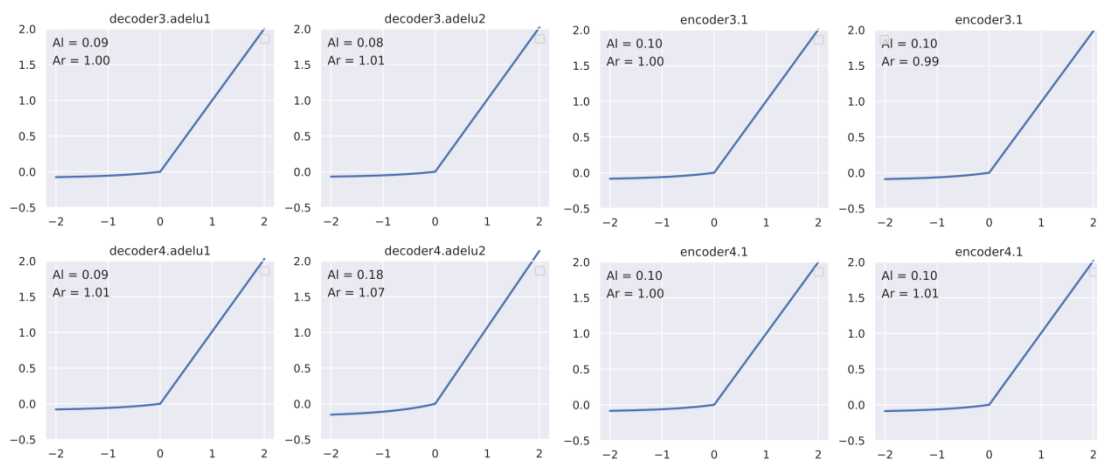


Рисунок 3.21 – Зміна параметрів функцій активації AdELU під час 5-ї епохи

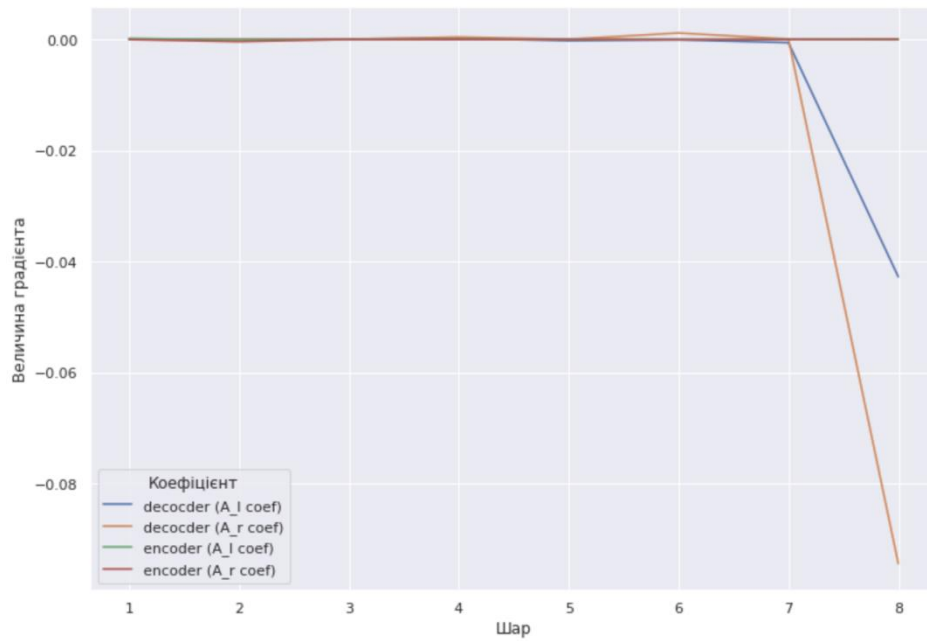


Рисунок 3.22 – Величина градієнта крізь шари для моделі з активаційною функцією AdELU під час 10-ї епохи

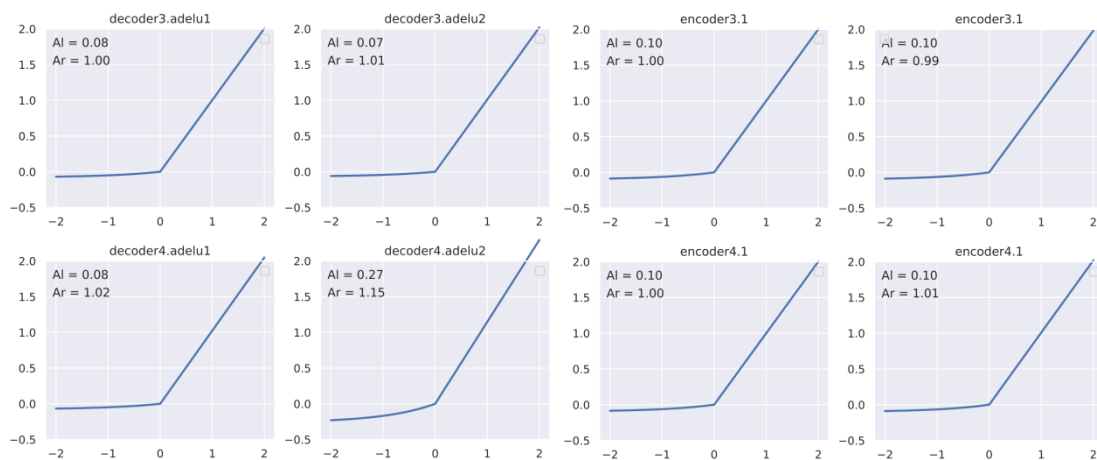


Рисунок 3.23 – Зміна параметрів функцій активації AdELU під час 10-ї епохи

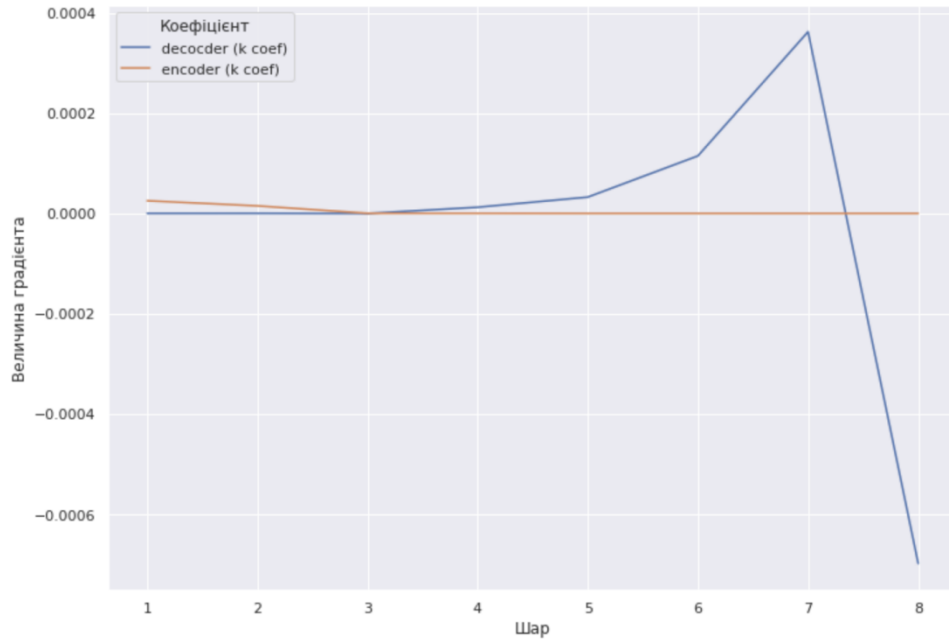


Рисунок 3.24 – Величина градієнта крізь шари для моделі з активаційною функцією AdSoftPlus під час 5-ї епохи

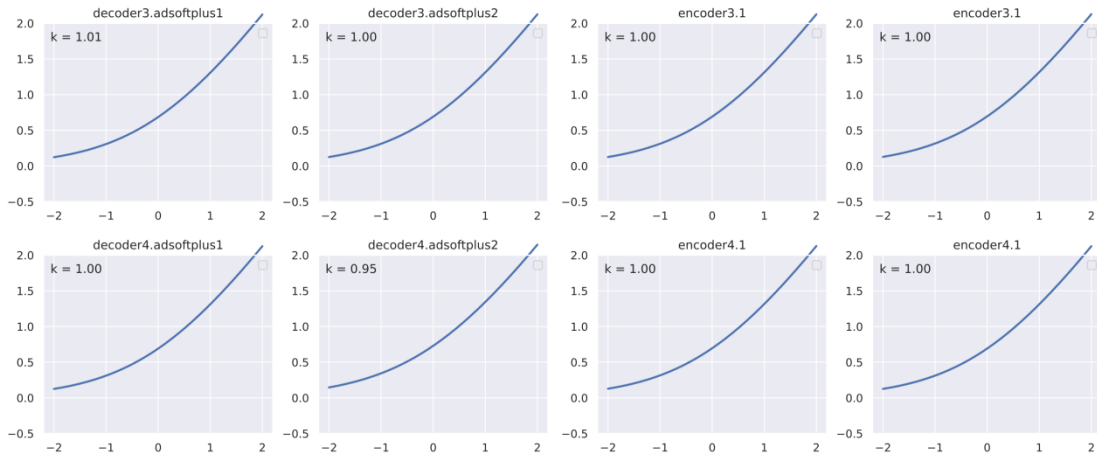


Рисунок 3.25 – Зміна параметрів функцій активації AdSoftPlus під час 5-ї епохи

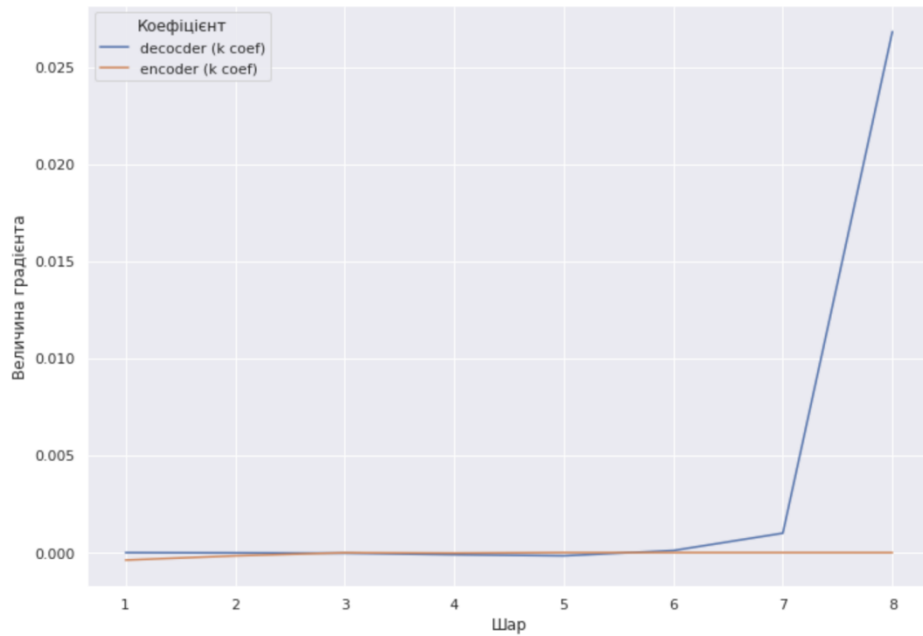


Рисунок 3.26 – Величина градієнта крізь шари для моделі з активаційною функцією AdSoftPlus під час 10-ї епохи

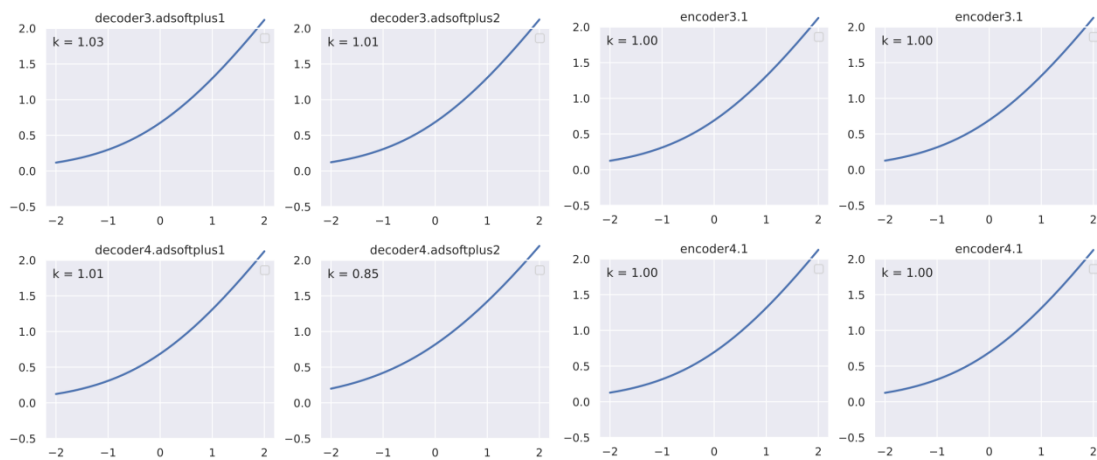


Рисунок 3.27 – Зміна параметрів функцій активації AdSoftPlus під час 10-ї епохи

Також можна відзначити, що адаптивні коефіцієнти функцій коливаються близько до початкових значень.

3.5 Подальший розвиток

За результатами проведеної роботи можна назвати такі потенційні кроки розвитку дослідження:

- провести експеримент із Leaky ReLU. Даний експеримент потрібен для перевірки гіпотези про те, що у моделей з адаптивними версіями функцій активації значення метрик якості вище, ніж у моделі з ReLU, через те, що, як і LReLU, вони не схильні до проблеми нейронів, що вмирають;

- навчати всі моделі до тих пір, поки вони не зіштовхнуться з проблемою перенавчання. Тим самим перевіряючи чи збільшують адаптивні активаційні функції схильність моделей до перенавчання або навпаки мають регуляризаційний ефект;

- у цій кваліфікаційної роботи всі експерименти проводилися с архітектурою U-Net навченою на наборі даних BraTS20, щоб узагальнити отримані результати, необхідно провести таку ж серію експериментів з іншими архітектурами (наприклад, V-Net, UNet ++ та інші) навчених на інших медичних наборах даних.

ВИСНОВКИ

У цій кваліфікаційній роботі програмно реалізована архітектура U-Net з тривимірними згортковими шарами для вирішення завдання семантичної сегментації багатомодальних МРТ зображень, які мають висоту, ширину, глибину та декілька модальностей. Проведена робота над аналізом та підготовкою багатовимірних даних для навчання. Реалізовані адаптивні ELU, ReLU та SoftPlus, які мають підвищити швидкість та якість навчання. Проведено порівняльний аналіз ефективності застосування адаптивних активаційних функцій для тривимірних згорткових шарів на медичних даних в порівнянні зі статичними активаційними функціями та досліджено їх поведінку під час навчання нейронної мережі. Оцінка якості сегментування проводилася за допомогою метрик Dice Score, Jaccard та Average Precision.

В атестаційній роботі представлені дослідження, які відповідають поставленому завданню та є актуальними для сучасної медичної візуалізації та автоматизації процесу діагностики. Проведені дослідження дали попередні відповіді на питання:

- «Чи доцільно використовувати адаптивні варіанти активаційних функцій для багатовимірних медичних даних?»;
- «Чи дійсно запропоновані модифікації PReLU та ELU підвищують швидкість та якість навчання?».

Отримані результати показують, що моделі з Adaptive Parametric Rectified Linear Unit та Adaptive Exponential Linear Unit мають більш точні результати прогнозування, ніж Exponential Linear Unit та Adaptive Softplus. Варто враховувати те, що результати даного дослідження є попередніми, щоб узагальнити висновки потрібно провести серію експериментів на різних наборах даних та конфігураціях гіперпараметрів, також слід додати статистичні методи перевірки гіпотези про перевагу одних моделей над іншими.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Daptive activation functions accelerate convergence in deep and physics-informed neural networks. *arXiv.org*. URL: <https://arxiv.org/abs/1906.01170> (дата звернення: 15.01.2022).
2. Zhou S. K. Medical image recognition, segmentation and parsing: machine learning and multiple object approaches. Elsevier Science & Technology Books, 2015. 542 p.
3. Компьютерное зрение. *КиберЛенинка*. URL: shorturl.at/gvyAK (дата звернення: 15.01.2022).
4. Fully convolutional networks for semantic segmentation. *arXiv.org*. URL: <https://arxiv.org/abs/1411.4038> (дата звернення: 17.01.2022).
5. Advances in deep learning-based medical image analysis / X. Liu et al. *Health data science*. 2021. Vol. 2021. P. 1–14. URL: <https://inlnk.ru/YAjzvz> (дата звернення: 17.01.2022).
6. AI-driven deep CNN approach for multi-label pathology classification using chest X-Rays / S. Albahli et al. *PeerJ computer science*. 2021. Vol. 7. P. e495. URL: <https://doi.org/10.7717/peerj-cs.495> (дата звернення: 20.01.2022).
7. Cybenko G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals, and systems*. 1989. Vol. 2, no. 4. P. 303–314. URL: <https://doi.org/10.1007/bf02551274> (дата звернення: 25.01.2022).
8. Low J. F. Softplus and softminus. *jiafulow.github.io*. URL: <https://inlnk.ru/AKBzZD> (дата звернення: 1.02.2022).
9. Bai K. A Comprehensive Introduction to Different Types of Convolutions in Deep Learning. towardsdatascience.com. URL: <https://shorturl.at/htBL7> (дата звернення: 01.05.2022)
10. 3D u-net: learning dense volumetric segmentation from sparse annotation. *arXiv.org*. URL: <https://arxiv.org/abs/1606.06650> (дата звернення: 01.02.2022).
11. Hwang H., Rehman H. Z. U., Lee S. 3D u-net for skull stripping in brain

MRI. *Applied sciences*. 2019. Vol. 9, no. 3. P. 569. URL: <https://doi.org/10.3390/app9030569> (дата звернення 05.02.2022).

12. Deep neural network with adaptive parametric rectified unit / Y. Bodyanskiy; A. Deineko; V. Skorik; F. Brodetskyi. *IEEE computingonline*. 2022. URL <https://doi.org/10.47839/ijc.21.1.2512> (дата звернення: 20.04.2022).

13. Python - overview. *Online Tutorials Library*. URL: <https://inlnk.ru/70jLyV> (дата звернення: 09.02.2022).

14. PyTorch - introduction. *Online Tutorials Library*. URL: <https://inlnk.ru/goNVMP> (дата звернення: 25.04.2022).

15. Neuroimaging in Python – NiBabel 3.2.2+146.ga20f13ad documentation. *nipy.org*. URL: <https://inlnk.ru/LANzaO> (дата звернення: 09.01.2022).

16. An introduction to seaborn – seaborn 0.11.2 documentation. *seaborn: statistical data visualization*. URL: <https://seaborn.pydata.org/introduction.html> (date of access: 25.04.2022).

17. Histogram-based normalization technique on human brain magnetic resonance images from different acquisitions / X. Sun et al. *BioMedical Engineering OnLine*. 2015. Vol. 14, no. 1. URL: <https://inlnk.ru/agm8Lk> (дата звернення 10.03.2022).

18. The multimodal brain tumor image segmentation benchmark (BRATS) / B. H. Menze et al. *IEEE transactions on medical imaging*. 2015. Vol. 34, no. 10. P. 1993–2024. URL: <https://doi.org/10.1109/tmi.2014.2377694> (дата звернення 28.03.2022).

19. Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features / S. Bakas et al. *Scientific data*. 2017. Vol. 4, no. 1. URL: <https://doi.org/10.1038/sdata.2017.117> (дата звернення: 12.04.2022).

20. Identifying the best machine learning algorithms for brain tumor segmentation, and overall survival prediction in the BRATS challenge. *arXiv.org*. URL: <https://arxiv.org/abs/1811.02629> (дата звернення: 12.04.2022).