

## **Лабораторная работа № 6. Сегментация клеток на изображениях с использованием свёрточной нейронной сети (CNN)**

Целью лабораторной работы является изучение и практическое применение свёрточных нейронных сетей (CNN) для задачи сегментации клеток на изображениях клеток. В ходе работы студенты научатся разрабатывать простую архитектуру CNN, проводить её обучение на биомедицинских данных и оценивать качество сегментации с использованием метрик Dice и IoU. Работа также направлена на понимание влияния различных гиперпараметров сети и функций активации на результативность сегментации.

Задачи:

1. Изучить основы свёрточных нейронных сетей (CNN) и их применение для сегментации изображений.
2. Разработать и обучить простую CNN для сегментации клеток на изображениях клеток.
3. Оценить качество сегментации с использованием соответствующих метрик.

### **Теоретические основы:**

Сегментация изображений – это задача разделения изображения на области, каждая из которых соответствует какому-либо объекту или его части. В биомедицинских приложениях, таких как диагностика клеток, сегментация помогает выявить границы клеток, что важно для дальнейшего анализа. Свёрточные нейронные сети широко применяются для таких задач, поскольку они эффективно выявляют пространственные структуры и паттерны в изображениях.

### ***Функции активации в свёрточных нейронных сетях***

Функции активации играют ключевую роль в работе нейронных сетей, включая свёрточные нейронные сети (CNN). Они определяют, будет ли нейрон активирован, и преобразуют входные сигналы в выходные. Без функций активации сеть была бы просто линейной комбинацией входов, что ограничило бы её способности к обучению и распознаванию сложных паттернов. Основные функции активации, используемые в CNN:

#### **1.1 ReLU (Rectified Linear Unit)**

Функция ReLU является одной из самых популярных функций активации в современных CNN благодаря своей простоте и эффективности. Она задаётся следующим образом:

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

ReLU делает все отрицательные значения на выходе нейрона равными нулю, а положительные значения остаются неизменными. Это помогает сети быстро сходиться во время обучения и избегать проблемы затухающих градиентов.

Преимущества:

- Простота вычислений.
- Ускоряет обучение за счёт устранения отрицательных значений.
- Справляется с проблемой исчезающих градиентов, что важно для глубоких нейронных сетей.

Недостатки:

- "Проблема мёртвых нейронов": если входное значение меньше или равно нулю, нейрон не обновляется и может "замереть" во время обучения.

### 1.2 Сигмоидная функция (Sigmoid)

Сигмоидная функция преобразует любое входное значение в диапазон от 0 до 1:

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

Сигмоидная функция популярна в ранних моделях нейронных сетей, особенно на финальных слоях для двоичной классификации, так как её выход можно интерпретировать как вероятность класса.

Преимущества:

- Подходит для задач классификации, где требуется результат в диапазоне от 0 до 1.
- Интуитивно понятна в смысле вероятностного вывода.

Недостатки:

- Проблема затухающих градиентов: для очень больших или маленьких значений градиенты становятся почти равными нулю, что замедляет обучение сети.
- Нецентрированная: выходные значения всегда положительные, что может привести к тому, что градиенты будут однонаправленными.

### 1.3 Гиперболический тангенс

Tanh-функция — это масштабированная версия сигмоидной функции. Она преобразует входные значения в диапазон от -1 до 1:

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

Преимущества:

- Тангенс-функция имеет выходные значения, центрированные относительно нуля, что делает её более подходящей для некоторых нейронных сетей по сравнению с сигмOIDом.

– В сравнении с сигмоидной функцией, она лучше справляется с корректировкой весов в процессе обучения.

Недостатки:

– Как и сигмоид, Tanh может страдать от проблемы затухающих градиентов.

#### 1.4 Leaky ReLU

Leaky ReLU – это модифицированная версия ReLU, которая решает проблему "мёртвых нейронов". Она задаётся следующим образом:

$$f(x) = x, \text{ если } x > 0, f(x) = \alpha x, \text{ если } 0 \geq x$$

где  $\alpha$  – это небольшой положительный коэффициент (например, 0.01), который позволяет небольшие отрицательные значения.

Преимущества:

- Избегает проблемы "мёртвых нейронов".
- Сохраняет вычислительную простоту ReLU.

Недостатки:

- Не всегда улучшает производительность модели.

## 2. Метрики для оценки сегментации: Dice и IoU

Для оценки качества сегментации изображений широко используются метрики Dice и Intersection over Union (IoU). Эти метрики позволяют количественно оценить, насколько хорошо модель сегментировала объекты, сопоставляя прогнозы модели с истинными метками (Ground Truth).

### 2.1 Коэффициент совпадения Дайса (Dice)

Метрика Dice используется для измерения схожести двух наборов данных – в данном случае предсказанных и истинных сегментаций. Она задаётся как:

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|}$$

где:

- A – предсказанная маска (сегментированная область),
- B – истинная маска (реальная разметка объекта),
- $|A \cap B|$  – пересечение областей A и B.

Значение Dice коэффициента варьируется от 0 до 1, где 1 означает идеальное совпадение предсказания с реальной разметкой.

Преимущества:

- Удобна для задач биомедицинской сегментации, где важна точная оценка пересечения сегментаций.
- Чувствительна к небольшим отклонениям в сегментации.

Недостатки:

– Может быть чувствительна к размеру объектов (маленькие объекты могут давать очень низкий Dice).

## 2.2 Intersection over Union (IoU)

Метрика IoU, также известная как Jaccard Similarity, измеряет отношение площади пересечения предсказанной маски и истинной маски к площади их объединения:

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

где:

- $|A \cap B|$  – площадь пересечения предсказанной и истинной масок,
- $|A \cup B|$  – площадь объединения предсказанной и истинной масок.

Значение IoU также варьируется от 0 до 1, где 1 означает полное совпадение сегментаций.

Преимущества:

- Устойчива к небольшим отклонениям, так как включает объединение площадей.
- Широко используется как стандартная метрика для оценки сегментации в различных областях.

Недостатки:

- Может давать менее высокие значения на малых объектах, что может усложнять оценку их качества.

Сравнение Dice и IoU

Метрики Dice и IoU тесно связаны. На практике метрика Dice часто оказывается более "оптимистичной" по сравнению с IoU, так как удваивает вес пересечения  $A \cap B$  в числителе. В случае идеальной сегментации обе метрики дадут одинаковый результат – 1, однако в реальных условиях часто Dice даёт несколько более высокие оценки, чем IoU.

## План работы

1. Разработать архитектуру простой CNN для сегментации клеток.
2. Обучить модель на изображениях клеток.
3. Оценить производительность модели с помощью метрик Dice и Intersection over Union (IoU).
4. Визуализировать результаты сегментации и сравнить с оригинальными изображениями.

Программное обеспечение:

- Язык программирования: Python
- Библиотеки: TensorFlow/Keras, OpenCV, Matplotlib, NumPy

## **Задания по вариантам**

### ***Вариант 1***

1. Разработайте простую CNN с функцией активации ReLU с одним свёрточным слоем для сегментации клеток.
2. Реализуйте аугментацию данных (повороты) для улучшения сегментации.
3. Используйте небольшой датасет, содержащий изображения клеток и соответствующие маски для сегментации.
4. Обучите модель на этом датасете.
5. Оцените качество сегментации с помощью метрик Dice и IoU.
6. Визуализируйте результаты на нескольких тестовых изображениях.

### ***Вариант 2***

1. Разработайте простую CNN с функцией активации Sigmoid с одним свёрточным слоем для сегментации клеток.
2. Реализуйте аугментацию данных (сдвиг) для улучшения сегментации.
3. Используйте небольшой датасет, содержащий изображения клеток и соответствующие маски для сегментации.
4. Обучите модель на этом датасете.
5. Оцените качество сегментации с помощью метрик Dice и IoU.
6. Визуализируйте результаты на нескольких тестовых изображениях.

### ***Вариант 3***

1. Разработайте простую CNN с функцией активации Tanh с одним свёрточным слоем для сегментации клеток.
2. Реализуйте аугментацию данных (масштабирование) для улучшения сегментации.
3. Используйте небольшой датасет, содержащий изображения клеток и соответствующие маски для сегментации.
4. Обучите модель на этом датасете.
5. Оцените качество сегментации с помощью метрик Dice и IoU.
6. Визуализируйте результаты на нескольких тестовых изображениях.

### ***Вариант 4***

1. Разработайте простую CNN с функцией активации ReLU с одним свёрточным слоем для сегментации клеток.
2. Реализуйте аугментацию данных (сдвиг) для улучшения сегментации.
3. Используйте небольшой датасет, содержащий изображения клеток и соответствующие маски для сегментации.
4. Обучите модель на этом датасете.
5. Оцените качество сегментации с помощью метрик Dice и IoU.

6. Визуализируйте результаты на нескольких тестовых изображениях.

#### ***Вариант 5***

1. Разработайте простую CNN с функцией активации Sigmoid с одним свёрточным слоем для сегментации клеток.

2. Реализуйте аугментацию данных (масштабирование) для улучшения сегментации.

3. Используйте небольшой датасет, содержащий изображения клеток и соответствующие маски для сегментации.

4. Обучите модель на этом датасете.

5. Оцените качество сегментации с помощью метрик Dice и IoU.

6. Визуализируйте результаты на нескольких тестовых изображениях.

#### ***Вариант 6***

1. Разработайте простую CNN с функцией активации Tanh с одним свёрточным слоем для сегментации клеток.

2. Реализуйте аугментацию данных (поворот) для улучшения сегментации.

3. Используйте небольшой датасет, содержащий изображения клеток и соответствующие маски для сегментации.

4. Обучите модель на этом датасете.

5. Оцените качество сегментации с помощью метрик Dice и IoU.

6. Визуализируйте результаты на нескольких тестовых изображениях.