Лабораторная работа 4: Создание и обучение простой нейронной сети

Цель

Цель данной лабораторной работы — освоить создание и обучение простой нейронной сети с использованием библиотек TensorFlow и Keras. Студенты научатся строить нейронные сети для задач классификации, проводить оценку их производительности и визуализировать результаты.

Описание

В рамках лабораторной работы студенты будут создавать и обучать простую нейронную сеть для задачи классификации, используя TensorFlow и Keras. Лабораторная работа включает в себя проектирование архитектуры сети, обучение модели на данных, оценку производительности и визуализацию результатов обучения и тестирования.

Общие задачи

1. **Создание модели**: Постройте архитектуру нейронной сети с использованием TensorFlow/Keras для решения задачи классификации.
2. **Обучение модели**: Обучите созданную модель на предоставленных данных.
3. **Оценка модели**: Оцените производительность модели с использованием метрик точности, потерь и других метрик.
4. **Визуализация результатов**: Постройте графики, показывающие процесс обучения, такие как графики потерь и точности, и визуализируйте результаты тестирования модели.

Индивидуальные задания

Каждому студенту предоставляется уникальная в соответствии с его номером в списке группы (смотреть в SSO).

1. **Установка и проверка TensorFlow**
    - **1.1:** Установите TensorFlow и Keras в вашем рабочем окружении. Напишите код для проверки успешной установки, который создаст и выведет базовую модель нейронной сети с одним скрытым слоем.
    - **1.2:** Реализуйте аналогичный код для задачи XOR, создавая базовую модель нейронной сети с одним скрытым слоем и проверяя её работоспособность.
2. **Загрузка и предобработка данных (MNIST)**
    - **2.1:** Загрузите набор данных MNIST и выполните его предобработку: нормализуйте изображения и разделите данные на обучающий, валидационный и тестовый наборы.
    - **2.2:** Реализуйте генерацию синтетического набора данных для задачи XOR (входные данные: (0,0), (0,1), (1,0), (1,1); целевые значения: 0, 1, 1, 0) и подготовьте данные для подачи в нейронную сеть.
3. **Проектирование модели с одним скрытым слоем (MNIST)**
    - **3.1:** Спроектируйте и реализуйте нейронную сеть с одним скрытым слоем для задачи классификации MNIST. Входной слой должен

соответствовать размерности входных данных, скрытый слой должен иметь 128 нейронов с функцией активации ReLU, и выходной слой должен иметь 10 нейронов с функцией активации softmax.
    - **3.2:** Спроектируйте и реализуйте нейронную сеть с одним скрытым слоем для задачи XOR. Скрытый слой должен содержать 4 нейрона с функцией активации ReLU, а выходной слой должен иметь 1 нейрон с функцией активации Sigmoid.
4. **Проектирование модели с двумя скрытыми слоями (MNIST)**
    - **4.1:** Реализуйте нейронную сеть с двумя скрытыми слоями для задачи MNIST. Первый скрытый слой должен содержать 64 нейрона с функцией активации ReLU, а второй скрытый слой — 32 нейрона с функцией активации ReLU. Выходной слой должен иметь 10 нейронов с функцией активации softmax.
    - **4.2:** Реализуйте нейронную сеть с двумя скрытыми слоями для задачи XOR. Первый скрытый слой должен содержать 4 нейрона с функцией активации ReLU, а второй скрытый слой — 4 нейрона с функцией активации ReLU. Выходной слой должен иметь 1 нейрон с функцией активации Sigmoid.
5. **Добавление слоя Dropout (MNIST)**
    - **5.1:** Внесите слой Dropout после первого скрытого слоя в модели с двумя скрытыми слоями для MNIST. Установите коэффициент Dropout равным 0.5 и проанализируйте его влияние на переобучение.
    - **5.2:** Добавьте слой Dropout с коэффициентом 0.5 после первого скрытого слоя в модели с двумя скрытыми слоями для XOR. Оцените влияние Dropout на переобучение и производительность модели.
6. **Добавление L2-регуляризации (MNIST)**
    - **6.1:** Добавьте L2-регуляризацию к весам первого скрытого слоя модели для MNIST. Установите коэффициент регуляризации равным 0.01 и оцените влияние на результаты обучения.
    - **6.2:** Внедрите L2-регуляризацию в первый скрытый слой модели для XOR. Установите коэффициент регуляризации равным 0.01 и оцените его влияние на результаты модели.
7. **Настройка скорости обучения (MNIST)**
    - **7.1:** Обучите модель с различными значениями скорости обучения (0.001, 0.01, 0.1) для MNIST. Проанализируйте, как изменение скорости обучения влияет на скорость сходимости и качество модели.
    - **7.2:** Обучите модель с различными значениями скорости обучения (0.001, 0.01, 0.1) для XOR. Проанализируйте, как скорость обучения влияет на сходимость и качество модели.
8. **Обучение модели и визуализация потерь (MNIST)**
    - **8.1:** Обучите модель с одним скрытым слоем для MNIST и постройте графики изменения потерь и точности на обучающем и валидационном наборах данных.
    - **8.2:** Обучите модель для XOR и постройте графики потерь и точности на обучающем и валидационном наборах данных.
9. **Построение матрицы ошибок (MNIST)**
    - **9.1:** Постройте матрицу ошибок для тестового набора данных после обучения модели MNIST. Проанализируйте, какие классы чаще всего ошибаются и почему.

- **9.2:** Постройте матрицу ошибок для тестового набора данных после обучения модели XOR. Проанализируйте, какие ошибки наиболее часты и как их можно исправить.

10. **Применение модели к новым данным (MNIST)**
    - **10.1:** Примените обученную модель MNIST к новым изображениям (из другого набора данных или случайно выбранным). Визуализируйте и проанализируйте предсказания модели.
    - **10.2:** Используйте обученную модель XOR для предсказания на новых данных. Визуализируйте и проанализируйте результаты предсказаний.
11. **Оптимизация параметров модели (MNIST)**
    - **11.1:** Используйте GridSearchCV для поиска оптимальных гиперпараметров (количество нейронов в скрытых слоях, скорость обучения) для модели MNIST и оцените, как оптимизация влияет на производительность модели.
    - **11.2:** Используйте GridSearchCV для нахождения оптимальных гиперпараметров (количество нейронов в скрытых слоях, скорость обучения) для модели XOR и оцените влияние на производительность модели.
12. **Кросс-валидация модели (MNIST)**
    - **12.1:** Реализуйте кросс-валидацию для оценки производительности модели MNIST. Используйте 5 фолдов и сравните средние значения точности на каждом фолде.
    - **12.2:** Реализуйте кросс-валидацию для оценки производительности модели XOR. Используйте несколько фолдов и сравните результаты.
13. **Обучение модели на подмножествах данных (MNIST)**
    - **13.1:** Обучите модель MNIST на различных подмножествах данных (например, только на 50% данных) и проанализируйте, как это влияет на производительность и точность модели.
    - **13.2:** Обучите модель XOR на различных подмножествах данных и проанализируйте, как это влияет на точность модели.
14. **Сохранение и загрузка модели (MNIST)**
    - **14.1:** Сохраните обученную модель MNIST в файл и загрузите её обратно. Проверьте, что результаты предсказаний после загрузки совпадают с результатами до сохранения.
    - **14.2:** Сохраните обученную модель XOR в файл и загрузите её обратно. Проверьте, что результаты предсказаний после загрузки совпадают с результатами до сохранения.
15. **Построение ROC-кривой (MNIST)**
    - **15.1:** Постройте ROC-кривую для вашей модели MNIST и вычислите AUC (площадь под кривой). Проанализируйте, насколько хорошо модель разделяет различные классы.
    - **15.2:** Постройте ROC-кривую для модели на задаче XOR и вычислите AUC (площадь под кривой). Проанализируйте, насколько хорошо модель разделяет классы.

Вопросы для подготовки

1. Какие основные компоненты входят в архитектуру нейронной сети?

2. Что такое функция активации, и какие функции активации используются в нейронных сетях?
3. Каковы основные метрики для оценки производительности нейронной сети?
4. Что такое регуляризация, и какие методы регуляризации применяются в нейронных сетях?
5. Как работает метод ранней остановки (Early Stopping) и зачем он используется?
6. Что такое Dropout и как он помогает предотвратить переобучение?
7. Какие методы используются для визуализации процесса обучения нейронной сети?
8. Как использовать GridSearch или RandomSearch для оптимизации гиперпараметров модели?
9. Какие функции потерь можно использовать в задачах классификации и регрессии?
10. Как визуализировать веса и активации нейронной сети для анализа её работы?

Критерии Оценки:
- Написание кода для решения индивидуальной задачи: 2 балл
- Объяснение и понимание написанного кода во время защиты: 1 балла
- Ответ на один из теоретических вопросов, выбранный преподавателем: 1 балл

***** Обязательно предоставьте отчет в формате Word, соответствующий всем стандартам.***