**Лабораторная работа по теме**   
"Рекуррентные нейронные сети"

**1. Название работы**

Рекуррентные нейронные сети: моделирование временных зависимостей.

**2. Цель работы**

Изучение и практическое применение рекуррентных нейронных сетей для анализа и моделирования временных рядов.

**3. Основные задачи работы**

1. Изучение теоретических основ рекуррентных нейронных сетей.

2. Разработка и обучение рекуррентной нейронной сети для анализа временных рядов.

3. Применение обученной модели для прогнозирования временных рядов.

4. Оценка и анализ полученных результатов.

**4. Теоретическая часть работы**

Рекуррентные нейронные сети (RNN) представляют собой класс архитектур глубокого обучения, предназначенных для работы с последовательными данными. Они отличаются от других типов нейронных сетей тем, что имеют циклическую связь, позволяющую передавать информацию от предыдущих шагов в текущий шаг. Это позволяет RNN учитывать контекст и зависимости внутри последовательности.

Основной компонент RNN - это рекуррентный слой, состоящий из набора нейронов, которые обмениваются информацией между собой на разных временных шагах. На каждом временном шаге RNN получает входные данные и скрытое состояние (hidden state) из предыдущего шага. Затем они комбинируются и передаются через активационную функцию, чтобы получить выходной результат и новое скрытое состояние. Этот процесс повторяется для каждого временного шага, что позволяет RNN моделировать последовательности переменной длины.

Математически RNN можно описать следующим образом:

- Входные данные на временном шаге t: X(t)

- Скрытое состояние на временном шаге t: H(t)

- Веса и смещения RNN: W

- Активационная функция: f()

На каждом временном шаге RNN выполняет следующие шаги:

1. Вычисление комбинированного входа: Z(t) = X(t) \* W(x) + H(t-1) \* W(h) + b

Здесь W(x) и W(h) - веса для входного и скрытого состояний соответственно, b - смещение.

2. Применение активационной функции: H(t) = f(Z(t))

Обычно в качестве активационной функции используется гиперболический тангенс или сигмоида.

3. Вычисление выходного результата: Y(t) = H(t) \* W(y)

Здесь W(y) - веса для выходного слоя.

Таким образом, RNN способна учитывать информацию из предыдущих шагов и передавать ее в следующие шаги, что позволяет моделировать зависимости и контекст в последовательных данных.

Пример использования RNN может быть задача анализа текста, где последовательность состоит из слов. RNN может обрабатывать каждое слово по очереди, учитывая его контекст в предложении для более точного предсказания, например, определения тональности текста или генерации текста.

Также RNN может применяться в области обработки речи, машинного перевода, генерации последовательностей и других задачах, где последовательные данные играют важную роль.

**5. Порядок выполнения работы**

1. Изучение теоретических основ рекуррентных нейронных сетей и их применения для анализа временных рядов.

2. Подготовка набора данных временного ряда для обучения и тестирования модели.

3. Разработка архитектуры рекуррентной нейронной сети с использованием фреймворка глубокого обучения, такого как TensorFlow.

4. Обучение разработанной модели на обучающем наборе данных с использованием алгоритма обратного распространения ошибки.

5. Применение обученной модели для прогнозирования значений временного ряда на тестовом наборе данных.

7. Анализ полученных результатов, выявление достоинств и недостатков модели.

**6. Контрольные вопросы**

1. Чем отличается рекуррентная нейронная сеть от обычной прямой нейронной сети?

2. Какие проблемы могут возникнуть при обучении рекуррентных нейронных сетей и как они могут быть решены?

3. Какая роль у рекуррентного слоя в рекуррентной нейронной сети?

4. Какие метрики можно использовать для оценки результатов прогнозирования временных рядов?

**Задание 1. Задача генерации текста посимвольно с использованием RNN**

В данной задаче входной последовательностью для RNN являются первые шесть символов текста, а целью является предсказание следующего (седьмого) символа. Затем предсказанный символ добавляется к входной последовательности, и процесс повторяется для генерации последующих символов.

Процесс рекуррентной генерации текста состоит из следующих шагов:

1. Подготовка обучающего набора: Обучающий набор представляет собой текстовые данные, которые могут быть скопированы или взяты из какого-либо источника. Весь текст разбивается на отдельные символы, и каждый символ становится элементом последовательности.

2. Предобработка данных: Символы текста преобразуются в числовой формат, например, с помощью one-hot encoding, где каждый символ представлен вектором размерности, равной общему количеству уникальных символов в обучающем наборе.

3. Создание RNN модели: Модель RNN создается, состоящая из рекуррентного слоя (RNN layer) и выходного слоя (output layer). Рекуррентный слой обрабатывает входную последовательность символов и передает информацию о предыдущем состоянии в следующий шаг. Выходной слой генерирует вероятности для следующего символа.

4. Обучение модели: RNN модель обучается на обучающем наборе с использованием метода обратного распространения ошибки. Происходит подбор оптимальных весов и параметров модели для минимизации ошибки предсказания следующего символа.

5. Генерация текста: После завершения обучения модели можно использовать для генерации нового текста. Процесс начинается с задания начальной последовательности из шести символов. Затем RNN модель принимает эту последовательность в качестве входа и предсказывает вероятности для следующего символа. Символ с наибольшей вероятностью выбирается и добавляется к текущей последовательности. После этого процесс повторяется для генерации следующего символа. Таким образом, можно генерировать текст, символ за символом, продолжая последовательность до достижения заданной длины или условия остановки.

6. Оценка результатов: Сгенерированный текст можно оценить с помощью различных метрик, таких как **перплексия**, сходство с исходным текстом или субъективное оценивание качества текста.