

СОЗДАНИЕ ПРОСТОЙ НЕЙРОННОЙ СЕТИ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Икромов Хусан Холмахаматович,
Андижанский машиностроительный институт

Аннотация. В данной статье рассматриваются основы нейронных сетей и процесс их создания с помощью языка программирования Python. Описаны ключевые компоненты нейронной сети: нейроны, слои, веса и функции активации. Подробно объясняется, как работает нейронная сеть, включая этапы прямого и обратного распространения. Статья предоставляет пошаговое руководство по созданию простой нейронной сети с использованием библиотеки Keras (часть TensorFlow), от установки библиотек и подготовки данных до компиляции модели, её обучения и оценки. Пример описывает задачу бинарной классификации с использованием полносвязной нейронной сети, что позволяет читателям понять основные принципы создания и применения нейросетей в реальных задачах.

Ключевые слова. Нейронная сеть, Python, Keras, TensorFlow, машинное обучение, слои, нейроны, функция активации, бинарная классификация, полносвязная сеть, обратное распространение, оптимизация, Adam, ReLU, сигмоида.

Нейронная сеть — это вычислительная модель, вдохновленная биологической нервной системой, в частности, мозгом. Она состоит из искусственных нейронов, которые соединены друг с другом слоями. Цель нейронной сети — научиться находить скрытые зависимости в данных и применять эти знания для решения задач, таких как классификация, регрессия или распознавание образов.

Основные компоненты нейронной сети:

1. Нейроны. Единичные вычислительные элементы, которые получают несколько входных сигналов, обрабатывают их и передают выходной сигнал.
 2. Слои. Нейроны группируются в слои:
 - Входной слой. Получает данные, которые поступают в сеть.
 - Скрытые слои. Обрабатывают данные на основе весов и функции активации.
 - Выходной слой. Возвращает итоговый результат.
- Вес. Коэффициенты, которые определяют важность каждого входного сигнала.
 - Функция активации. Определяет, будет ли нейрон активен (выдаст ли он сигнал). Примеры функций активации — ReLU, сигмоида и tanh.

Процесс работы нейронной сети:

- Прямое распространение. Входные данные проходят через сеть от входного слоя к выходному, и на каждом шаге они обрабатываются нейронами с применением весов и функций активации.
- Обратное распространение. После получения результата сеть сравнивает его с правильным ответом и с помощью оптимизатора (например, Adam) корректирует веса так, чтобы уменьшить ошибку в следующий раз.

Создание простой нейронной сети на языке программирования Python

Автор: Икромов Х.Х.
20.12.2024 21:51 -

Пример создания простой нейронной сети на Python

Для создания простой нейронной сети на Python можно использовать библиотеку Keras, которая является частью TensorFlow. Вот пример простой нейронной сети для задачи классификации:

Шаг 1: Установка TensorFlow

Для начала установим библиотеку TensorFlow (если она еще не установлена):

```
pip install tensorflow
```

Шаг 2: Импорт библиотек

```
import tensorflow as tf
```

```
from tensorflow.keras.models import Sequential
```

Создание простой нейронной сети на языке программирования Python

Автор: Икромов Х.Х.
20.12.2024 21:51 -

```
from tensorflow.keras.layers import Dense
```

```
import numpy as np
```

Шаг 3: Подготовка данных

Создадим фиктивные данные для примера. Допустим, мы хотим решить задачу бинарной классификации:

```
# Генерируем данные
```

```
X = np.random.rand(1000, 10) # 1000 примеров, каждый с 10 признаками
```

```
y = np.random.randint(2, size=(1000, 1)) # 1000 меток, по одной на каждый пример (0 или 1)
```

Шаг 4: Создание нейронной сети

Создадим простую сеть с двумя скрытыми слоями и одним выходным слоем:

Создание простой нейронной сети на языке программирования Python

Автор: Икромов Х.Х.
20.12.2024 21:51 -

```
# Инициализируем последовательную модель
```

```
model = Sequential()
```

```
# Добавляем первый скрытый слой: 32 нейрона, функция активации ReLU
```

```
model.add(Dense(32, activation='relu', input_shape=(10,))) # Входной слой ожидает 10 признаков
```

```
# Добавляем второй скрытый слой: 16 нейронов, функция активации ReLU
```

```
model.add(Dense(16, activation='relu'))
```

```
# Добавляем выходной слой: 1 нейрон, функция активации сигмоида для бинарной классификации
```

```
model.add(Dense(1, activation='sigmoid'))
```

Шаг 5: Компиляция модели

Скомпилируем модель, указав функцию потерь и оптимизатор:

```
model.compile(optimizer='adam',  
  
              loss='binary_crossentropy',  
  
              metrics=['accuracy'])
```

Шаг 6: Обучение модели

Обучим модель на наших данных:

```
model.fit(X, y, epochs=10, batch_size=32)
```

Шаг 7: Оценка модели

После обучения можно оценить качество модели:

```
python
```

Копировать код

```
test_loss, test_acc = model.evaluate(X, y)
```

```
print(f'Точность на тестовых данных: {test_acc}')
```

Как это работает?

1. `Sequential` модель — это линейная модель, где слои идут друг за другом.
2. `Dense` слой — это полносвязный слой, в котором каждый нейрон предыдущего слоя соединён с каждым нейроном следующего.
3. Функция активации `ReLU` — позволяет модели справляться с нелинейными зависимостями.
4. Функция активации сигмоида — используется в выходном слое для бинарной классификации.
5. `Adam` оптимизатор — один из самых популярных методов для настройки весов сети.

Таким образом, с использованием таких простых технологий, как Keras и TensorFlow, можно быстро создать и обучить нейронную сеть, даже не обладая глубокими знаниями в машинном обучении.

Литература

1. Ikromov Khusan Kholmakhamatovich, . (2023). HISTORICAL CONTEXT OF DEVELOPMENT OF INFORMATION SYSTEMS AND DATABASE MANAGEMENT. International Journal of Pedagogics, 3(11), 119–123.
<https://doi.org/10.37547/ijp/Volume03Issue11-23>
2. Ikromov, X. TA'LIM JARAYONIDAGI SUN'IY INTELLEKTGA ASOSLANGAN AXBOROTNI QAYTA ISHLASH VOSITALARINING TAHLILI. O 'ZBEKISTON RESPUBLIKASI OLIY TA'LIM, FAN VA INNOVATSIYALAR VAZIRLIGI ANDIJON DAVLAT UNIVERSITETI UMUMIY PEDAGOGIKA KAFEDRASI, 315.
3. Mamasidiqov Baxodir Qobuljon o'g'li. (2024). O'ZBEKISTONDA BOZOR IQTISODIYOTIGA O'TISHNING YANGI BOSQICHI. IQTISODIYOT VA MATEMATIKA: Andijon mashinasozlik instituti "Axborot texnologiya" kafedrası, katta o'qituvchisi Zulfixarov Iloxom Maxmudovich taqrizi ostida . IQRO INDEXING, 9(2), 52-55.
<http://worldlyjournals.com/index.php/IFX/article/view/2446>

Создание простой нейронной сети на языке программирования Python

Автор: Икромов Х.Х.
20.12.2024 21:51 -
