

ИСКУССТВЕННАЯ НЕЙРОННАЯ СЕТЬ

В этой работе вы познакомитесь с основами работы искусственных нейронов. В следующих статьях мы изучим основы нейронных сетей и напишем простую нейронную сеть на Python.

Искусственная нейронная сеть (ИНС) - математическая модель, основанная на принципах организации и функционирования биологических нейронных сетей - сетей нервных клеток живого организма, а также его программного или аппаратного обеспечения.

ИНС состоит из искусственных нейронов, каждый из которых представляет собой упрощенную модель биологического нейрона. Искусственный нейрон принимает сигналы от множества входов, обрабатывает их в едином порядке и передает результат многим другим искусственным нейронам, что аналогично биологическому нейрону. Биологические нейроны связаны аксонами, соединения называются синапсами. В синапсах наблюдается увеличение или уменьшение электрохимического сигнала. Связи между искусственными нейронами называются синаптическими или простыми синапсами. Есть один параметр синапса - весовой коэффициент, изменение информации в зависимости от его значения происходит при передаче от одного нейрона к другому. Благодаря этому входная информация обрабатывается и преобразуется в результаты, а обучение нейронной сети основывается на экспериментальном выборе такого весового коэффициента для каждого синапса, что приводит к желаемому результату. [1-2].

Нейронные сети состоят из взаимосвязанных нейронов, поэтому нейрон является ключевой частью нейронной сети. Нейроны выполняют всего 2 действия: они умножают входные данные на веса, соединяют их и добавляют прикосновение, а второе действие - активация.

Теперь подробнее о входных данных, весе, отклонениях и функции активации.

Входные данные - это информация, которую нейрон получает от предыдущих нейронов или пользователя.

Каждый вход нейрона взвешивается, и ему изначально присваиваются случайные числа. При обучении нейронной сети значение нейронов и смещений меняется. Входные данные, передаваемые нейрону, умножаются на вес.

Смещения назначаются каждому нейрону, как и вес первого веса, это случайные числа. Смещение облегчает и ускоряет обучение нейронных сетей.

Функция активации, которую мы используем в нашей нейронной сети, называется сигмоидной. Сигмовидная формула:

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

Данная функция заворачивает любое число от $-\infty$ до $+\infty$ в диапазон от 0 до 1.

Если описать математику, задачи, выполняемые нейроном, вы получите 2 формулы:

$f(z)$ - функция активации

z - сумма произведения входных данных с весами, и смещения

i - массив входных данных

w - массив весов

b - смещение

j - количество входных данных и весов

$$z = i_1 * w_1 + \dots + i_j * w_j + b$$

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

Алгоритм работы нейрона следующий[3].



Теперь реализуем этот алгоритм на языке программирования Python. Прежде чем вы сможете запустить нейрон, вы должны загрузить и включить библиотеку numpy. Чтобы загрузить библиотеку, введите в терминале Linux или Windows:

```
pip install numpy
```

Чтобы добавить библиотеку в нашу программу, напишите в начале программы:

```
import numpy as np
```

Чтобы упростить использование и чтение кода в больших программах, мы вводим класс Neuron и пишем в нем конструктор:

```
class Neuron:
    def __init__(self, number_of_weights = 1):
        self.w = np.random.normal(size=number_of_weights)
        self.b = np.random.normal()
```

Конструктор класса принимает количество входов для нейрона в качестве параметра и заполняет набор весов случайными числами, после чего мы генерируем сдвиг. Нам также необходимо инициализировать функцию активации, чтобы нейрон работал. Это будет так:

```
def activate(self, inputs):
    x = np.dot(self.w, inputs) + self.b
    return sig(x)
```

Функция получает ввод, включает его и возвращает сигмоидальный результат. Чтобы функция активации работала, сигмовидная функция должна быть написана:

```
def sig(x): return 1 / (1 + np.exp(-x))
```

Мы разместим эту функцию в файле Math.py. Перед функцией мы снова добавляем numpy:

```
import numpy as np
```

Записываем класс нейрона в файл Neuron.py. Добавьте файл Math.py в Neuron.py:

Темир А. М., докторант, Евразийский национальный университет, г. Нур-султан, Республика Казахстан

Оразбаев Батырбай Бидайбекович, доктор технических наук, профессор

```
from Math import *
```

В итоге мы получаем два файла: Neuron.py

```
import numpy as np
from Math import *
class Neuron:
    def __init__(self, number_of_weights = 1):
        self.w = np.random.normal(size=number_of_weights)
        self.b = np.random.normal()
    def activate(self, inputs):
        x = np.dot(self.w, inputs) + self.b
        return sig(x)
```

Math.py

```
import numpy as np
def sig(x):
    return 1 / (1 + np.exp(-x))
```

И так если подводить итог данной работы то во первых мы изучили принципы работы искусственных нейронных сетей, во вторых изучили формулы которые используются при работе нейронных сетей. А так же написали класс нейронных сетей на языке программирования Python[4].

Список литературы

1. <https://www.it.ua/ru/knowledge-base/technology-innovation/iskusstvennye-nejronnye-seti-ins>
2. <https://postnauka.ru/longreads/155977>
3. <https://www.it.ua/ru/knowledge%E2%80%94base/technology%E2%80%94innovation/machine%E2%80%94learning>
4. <https://habr.com/ru/company/microsoft/blog/316456/>