

- Цифровые технологии
 - Перевод 3х статей с английского языка
 - Подготовить краткое содержание каждой статьи
 - Подготовить презентацию по краткому содержанию
 - Подготовить доклад (необходимо выбрать дату для доклада)

- Интеллектуальный анализ данных
 - написать 3 программы
 - оформить и подать документы для регистрации ПО

Интеллектуальный анализ данных

Разрабатываем систему интеллектуального анализа данных с применением нейронных сетей.

К 13.02 необходимо:

- Определиться с данными, которые вы планируете анализировать
- Добавить в таблицу примерное название вашей программы:

https://docs.google.com/spreadsheets/d/1_dR1lorp_-MFdCkw9KsZjC-5OpdnzhJVildRCxJHqP4/edit?usp=sharing

- Создать блокнот с кодом из модуля 2 и разобрать основные этапы

Как выбрать данные?

Посмотреть на тему вашей магистерской работы и подумать какие данные вы планируете в ней обрабатывать.

Даже если работа планируется расчетная, в любом случае рассчитывается она на каких-то данных или диапазонах.

Необходимо подумать какие параметры вы используете, и какие планируете получить.

В случае, если реальных данных пока нет, их можно заменить модельными - сгенерированными рандомно в заданных диапазонах.

Можно взять данные из литературы (возможно есть работы, в которых аналогичные данные уже были получены).

Нейронные сети

Искусственные нейросеть (ИНС) — это программная реализация нейронных структур нашего мозга.

Нейроны могут изменять тип передаваемых сигналов в зависимости от электрических или химических сигналов, которые в них передаются.

Нейросеть в человеческом мозге — огромная взаимосвязанная система нейронов, где сигнал, передаваемый одним нейроном, может передаваться в тысячи других нейронов. Обучение происходит через повторную активацию некоторых нейронных соединений. Из-за этого увеличивается вероятность вывода нужного результата при соответствующей входной информации (сигналах). Такой вид обучения использует обратную связь — при правильном результате нейронные связи, которые выводят его, становятся более плотными.

Искусственные нейронные сети имитируют поведение мозга в простом виде. Они могут быть обучены контролируемым и неконтролируемым путями. В контролируемой ИНС, сеть обучается путем передачи соответствующей входной информации и примеров исходной информации.

Например, спам-фильтр в электронном почтовом ящике: входной информацией может быть список слов, которые обычно содержатся в спам-сообщениях, а исходной информацией — классификация для уведомления (спам, не спам). Такой вид обучения добавляет веса связям ИНС. Неконтролируемое обучение в ИНС пытается «заставить» ИНС «понять» структуру передаваемой входной информации «самостоятельно».

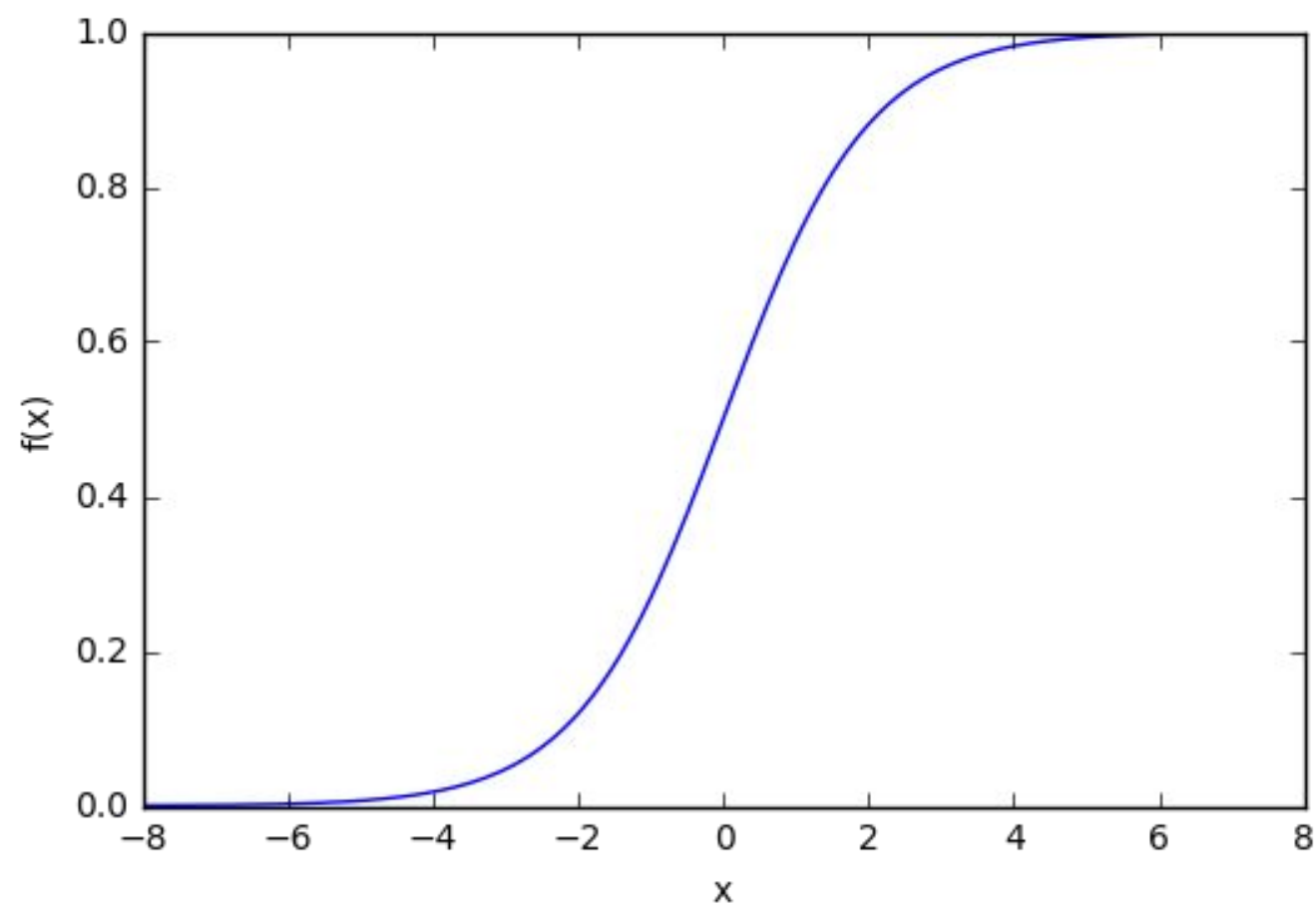
Структура ИНС

Биологический нейрон имитируется в ИНС через активационную функцию. В задачах классификации (например определение спам-сообщений) активационная функция должна иметь характеристику «включателя».

Иными словами, если вход больше, чем некоторое значение, то выход должен изменять состояние, например с 0 на 1 или -1 на 1 Это имитирует «включение» биологического нейрона. В качестве активационной функции обычно используют сигмоидальную функцию:

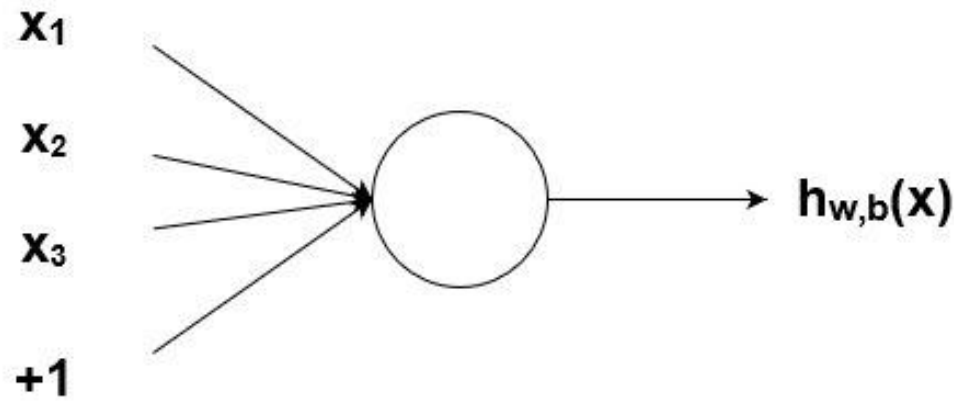
$$f(z) = \frac{1}{1 + \exp(-z)}$$

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



Узлы

Каждый узел принимает взвешенный вход, активирует активационную функцию для суммы входов и генерирует ВЫХОД.



Узел является «местоположением» активационной функции, он принимает взвешенные входы, складывает их, а затем вводит их в активационную функцию. Вывод активационной функции представлен через h .

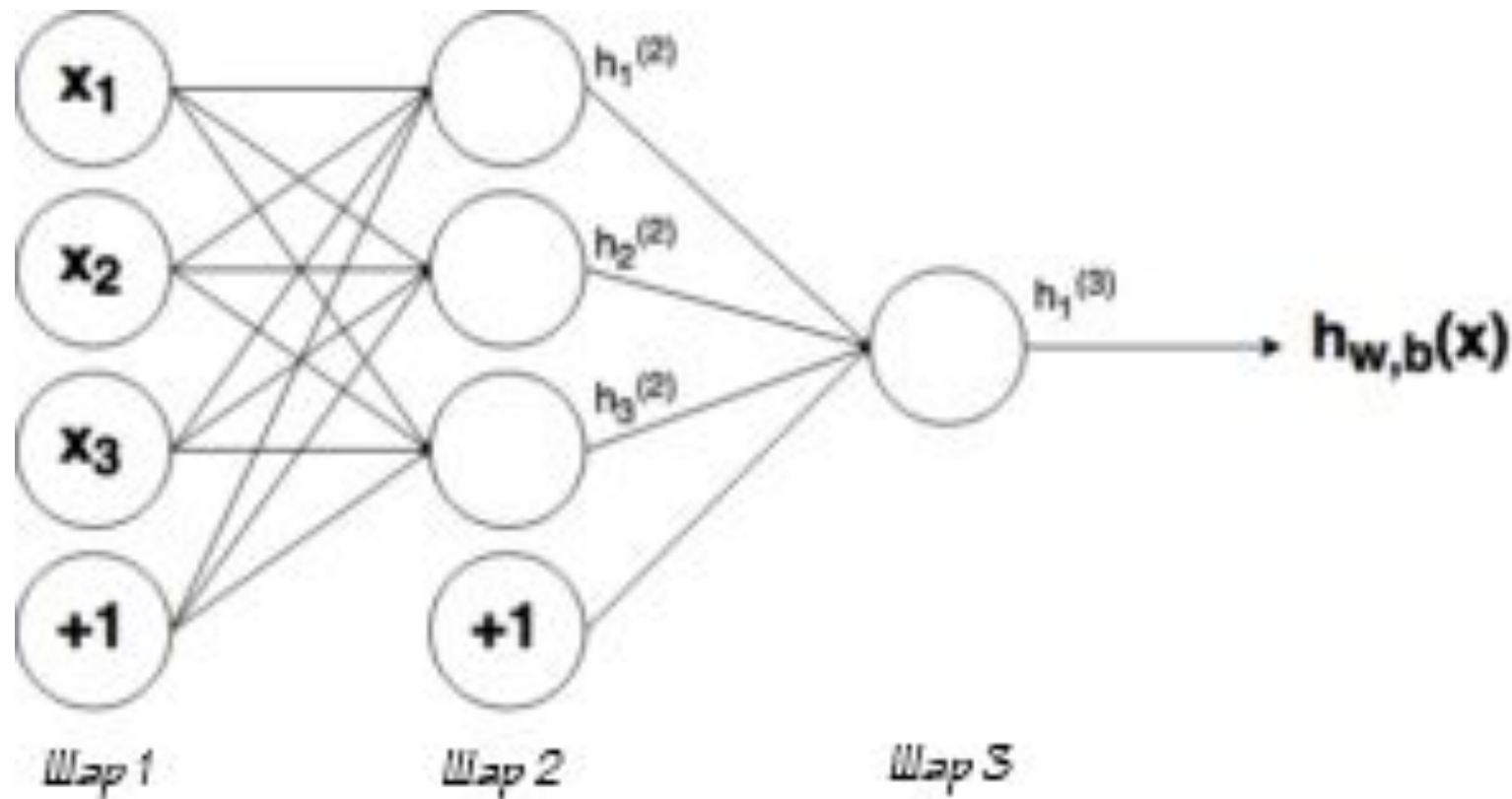
По весу берутся числа (не бинарные), которые затем умножаются на входе и суммируются в узле. Иными словами, взвешенный вход в узел имеет вид:

$$x_1 w_1 + x_2 w_2 + x_3 w_3 + b$$

где w_i — числовые значения веса (b мы будем обсудим позже). Весы нам нужны, они являются значениями, которые будут меняться в течение процесса обучения. b является весом элемента смещения на 1, включение веса b делает узел гибким. Проще это понять на примере.

Составленная структура

В полной нейронной сети находится много таких взаимосвязанных между собой узлов. Структуры таких сетей могут принимать мириады различных форм, но самая распространенная состоит из входного слоя, скрытого слоя и выходного слоя.



Нейросеть в 11 строчек на Python

Нейросеть, тренируемая через обратное распространение (backpropagation), пытается использовать входные данные для предсказания выходных.

Вход			Выход
0	0	1	0
1	1	1	1
1	0	1	1
0	1	1	0

```
import numpy as np
```

```
# Сигмоида
```

```
def nonlin(x,deriv=False):  
    if(deriv==True):  
        return x*(1-x)  
    return 1/(1+np.exp(-x))
```

```
# набор входных данных
```

```
X = np.array([ [0,0,1],  
               [0,1,1],  
               [1,0,1],  
               [1,1,1] ])
```

```
# выходные данные
```

```
y = np.array([[0,0,1,1]]).T
```

```
# сделаем случайные числа более определёнными
```

```
np.random.seed(1)
```

```
# инициализируем веса случайным образом со средним 0
```

```
syn0 = 2*np.random.random((3,1)) - 1
```

```
for iter in range(10000):
```

```
    # прямое распространение
```

```
    l0 = X
```

```
    l1 = nonlin(np.dot(l0,syn0))
```

```
    # насколько мы ошиблись?
```

```
    l1_error = y - l1
```

```
    # перемножим это с наклоном сигмоиды
```

```
    # на основе значений в l1
```

```
    l1_delta = l1_error * nonlin(l1,True) # !!!
```

```
    # обновим веса
```

```
    syn0 += np.dot(l0.T,l1_delta) # !!!
```

```
print("Выходные данные после тренировки:")
```

```
print(l1)
```

Выходные данные после тренировки:

[[0.00966449]

[0.00786506]

[0.99358898]

[0.99211957]]

Переменные и их описание

X — матрица входного набора данных; строки — тренировочные примеры

y — матрица выходного набора данных; строки — тренировочные примеры

I_0 — первый слой сети, определённый входными данными

I_1 — второй слой сети, или скрытый слой

syn_0 — первый слой весов, Synapse 0, объединяет I_0 с I_1 .

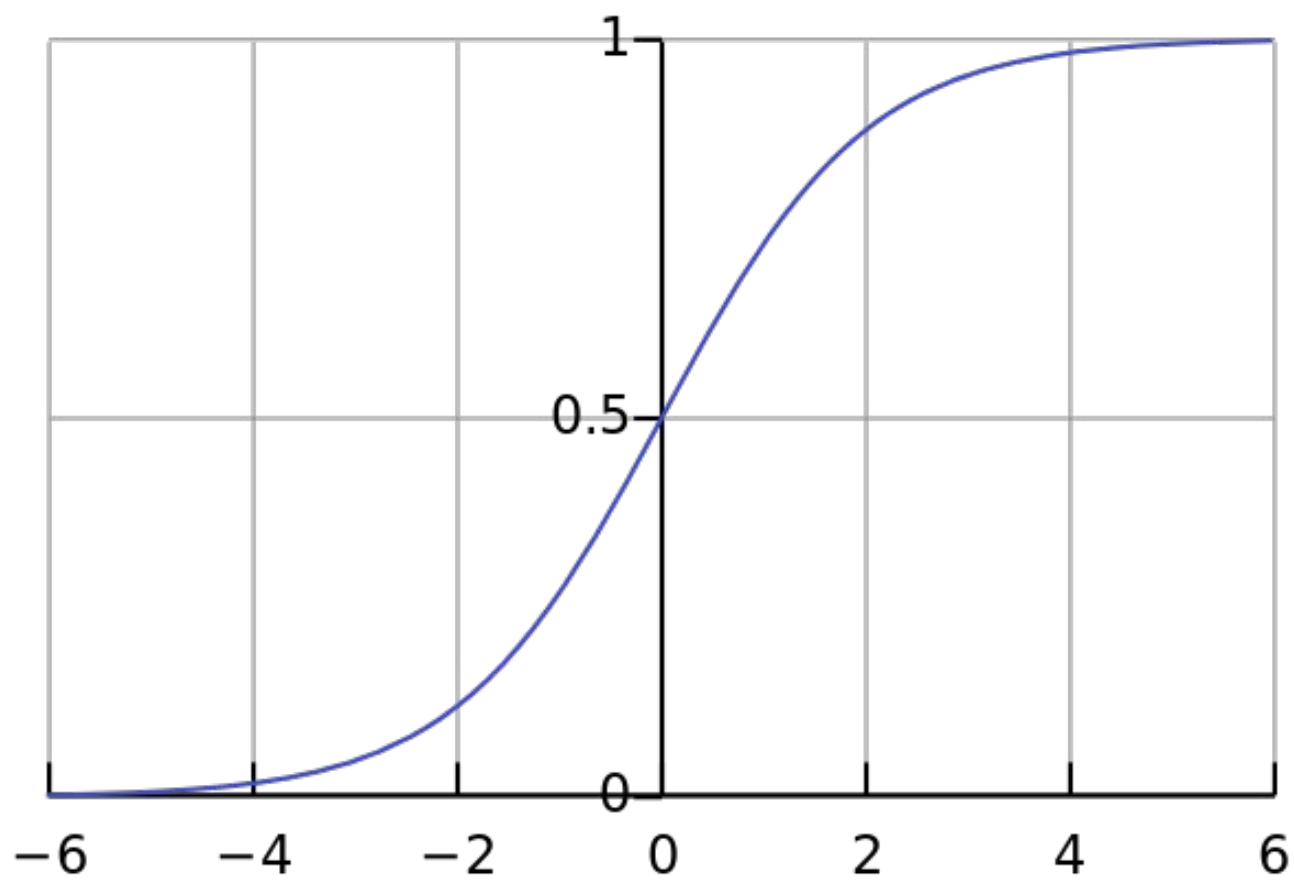
"*" — поэлементное умножение — два вектора одного размера умножают соответствующие значения, и на выходе получается вектор такого же размера

"-" — поэлементное вычитание векторов

$x.\text{dot}(y)$ — если x и y — это вектора, то на выходе получится скалярное произведение. Если это матрицы, то получится перемножение матриц. Если матрица только одна из них — это перемножение вектора и

матрицы

```
def nonlin(x,deriv=False):
```



- сравните l_1 после первой итерации и после последней
- посмотрите на функцию `nonlin`.
- посмотрите, как меняется `l1_error`
- разберите строку 36 – (отмечена !!!)
- разберите строку 39 – (отмечена !!!)
- предсказать выходные данные на основе трёх входных столбцов данных:

Вход		Выход	
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	0