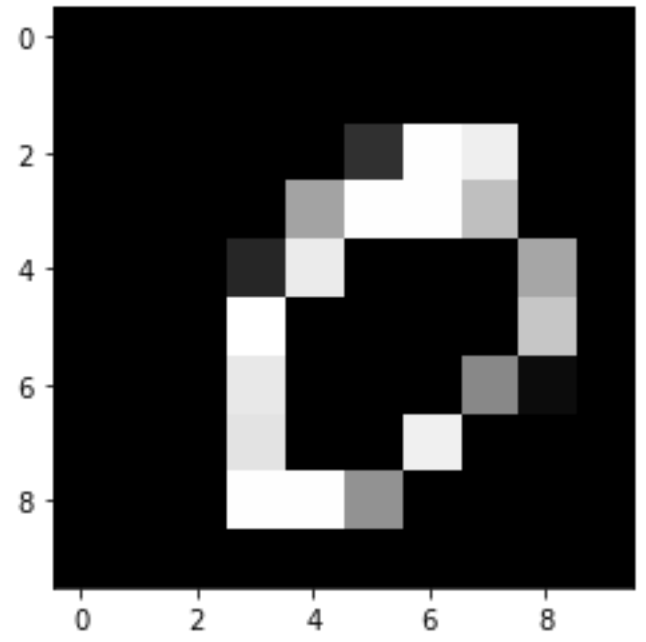
Функции активации. Инициализация весов

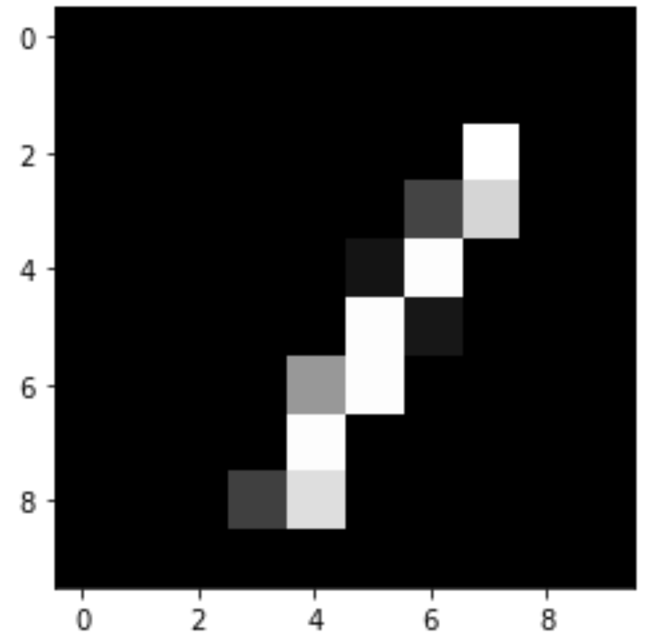
Recap

Бинарная классификация

Объекты $x_1, \ldots, x_n$

Ответы $y_1, \ldots, y_n \in [0,1]$



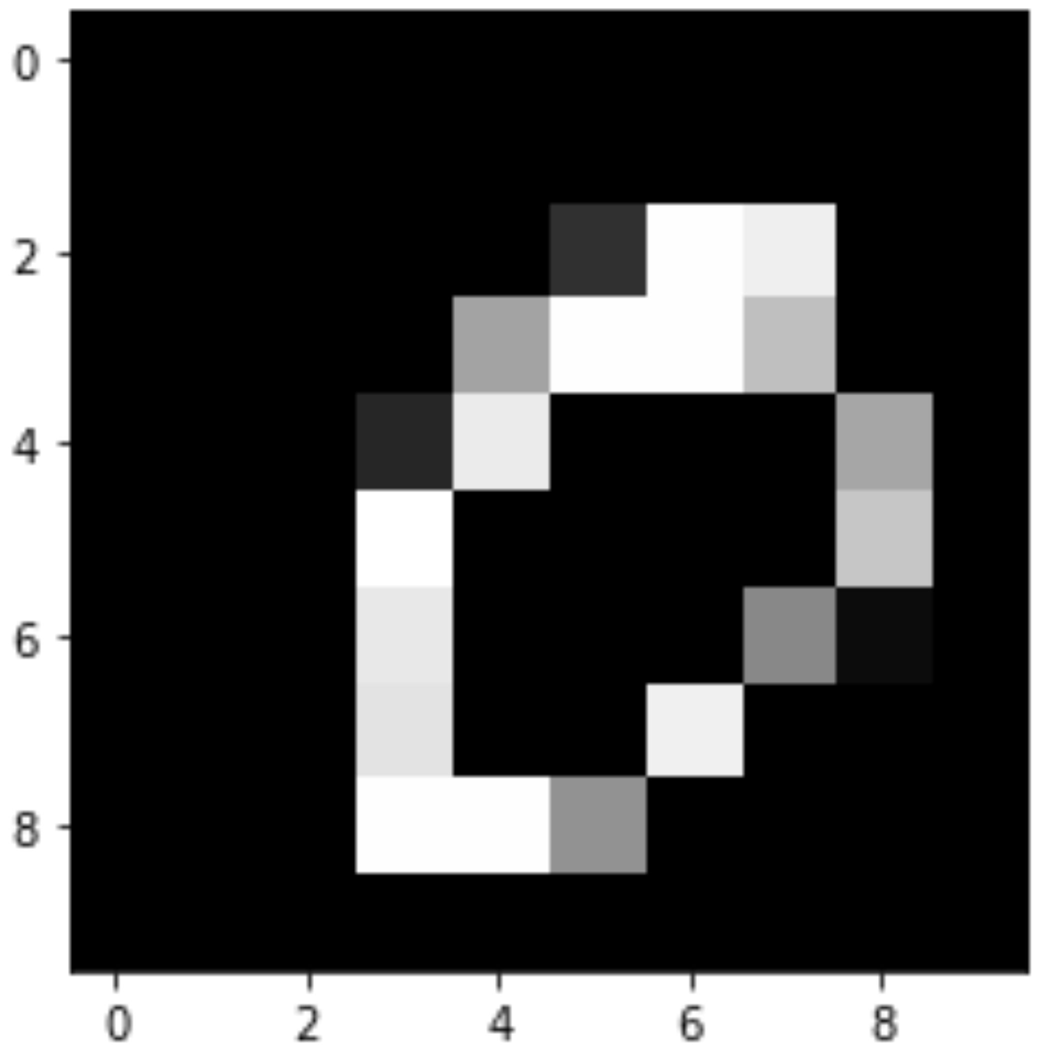


Алгоритм предсказания $f(x, \theta) = P(y = 1 \,|\, x, \theta) > \dfrac{1}{2}$

нейросеть из двух блоков вида Linear -> Sigmoid

Бинарная классификация

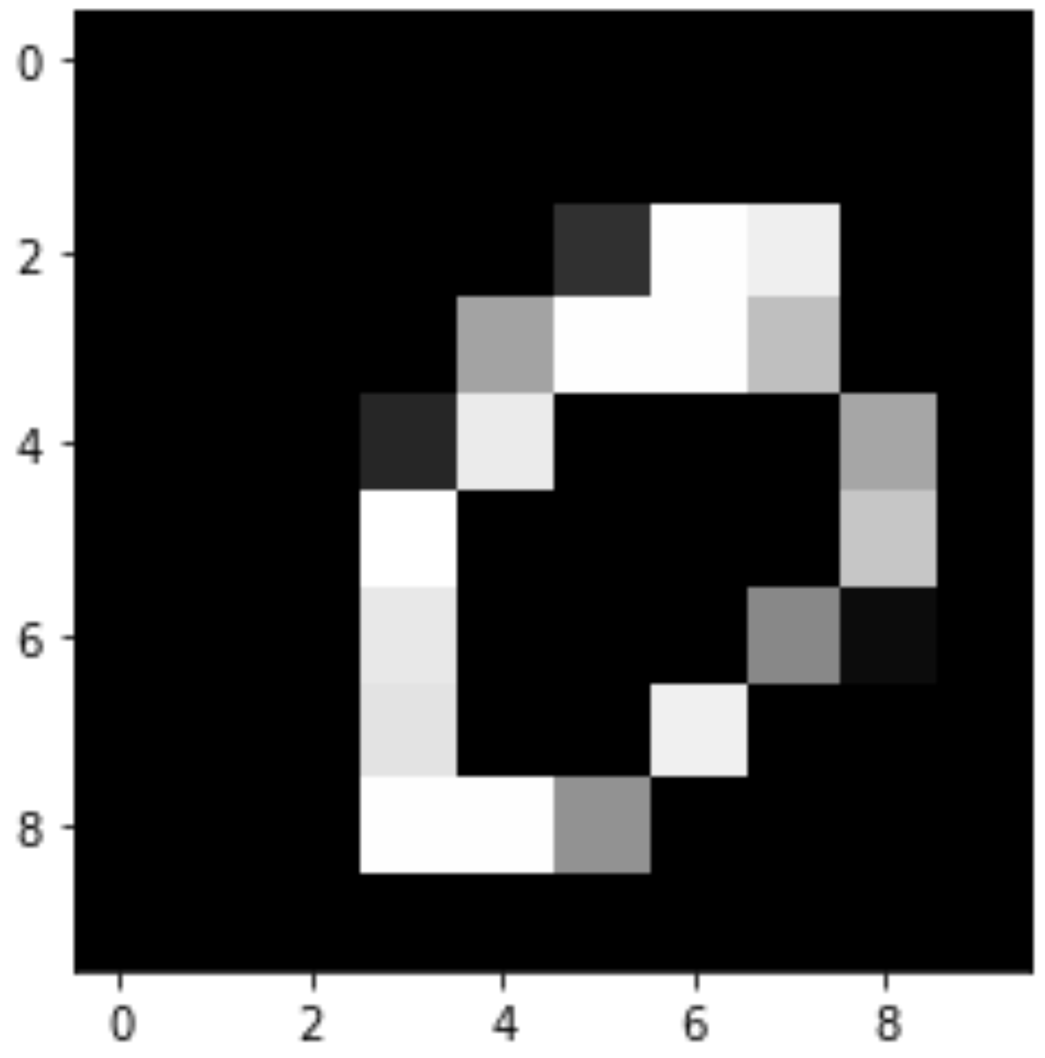
Input



x - ч/б картинка

Бинарная классификация

Input



x - ч/б картинка

```
[[0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.1875    , 0.984375  , 0.92578125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.63671875,
  0.984375  , 0.984375  , 0.73828125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.1484375 , 0.91015625,
  0.        , 0.        , 0.        , 0.64453125, 0.        ],
 [0.        , 0.        , 0.        , 0.98828125, 0.        ,
  0.        , 0.        , 0.        , 0.765625  , 0.        ],
 [0.        , 0.        , 0.        , 0.8984375 , 0.        ,
  0.        , 0.        , 0.52734375, 0.046875  , 0.        ],
 [0.        , 0.        , 0.        , 0.87890625, 0.        ,
  0.        , 0.9296875 , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.984375  , 0.984375  ,
  0.56640625, 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ]],
```

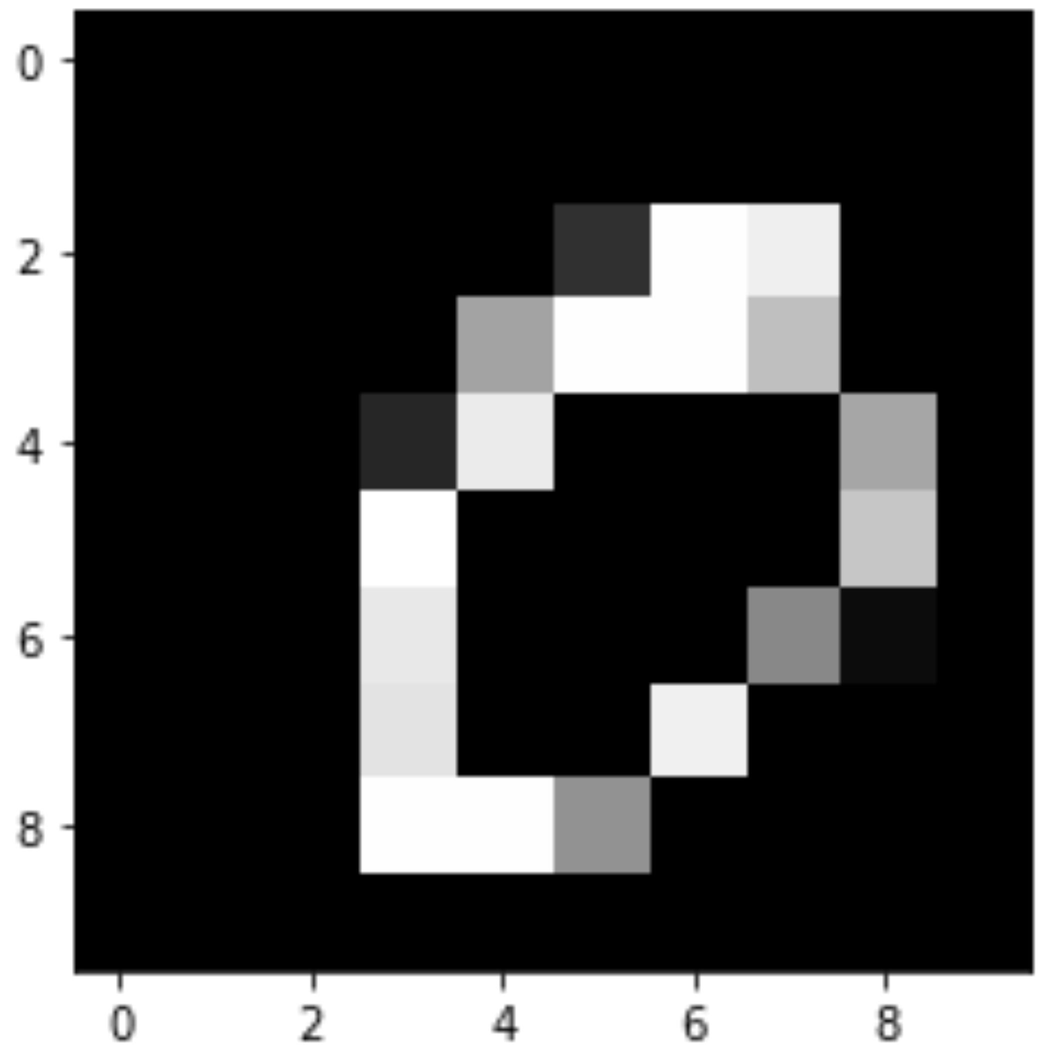
=

массив с элементами от 0 до 1

0 - черный, 1 - белый

Бинарная классификация

Input




```
[[0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.1875    , 0.984375  , 0.92578125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.63671875,
  0.984375  , 0.984375  , 0.73828125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.1484375 , 0.91015625,
  0.        , 0.        , 0.        , 0.64453125, 0.        ],
 [0.        , 0.        , 0.        , 0.98828125, 0.        ,
  0.        , 0.        , 0.        , 0.765625  , 0.        ],
 [0.        , 0.        , 0.        , 0.8984375 , 0.        ,
  0.        , 0.        , 0.52734375, 0.046875  , 0.        ],
 [0.        , 0.        , 0.        , 0.87890625, 0.        ,
  0.        , 0.9296875 , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.984375  , 0.984375  ,
  0.56640625, 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ]],
```

x - ч/б картинка

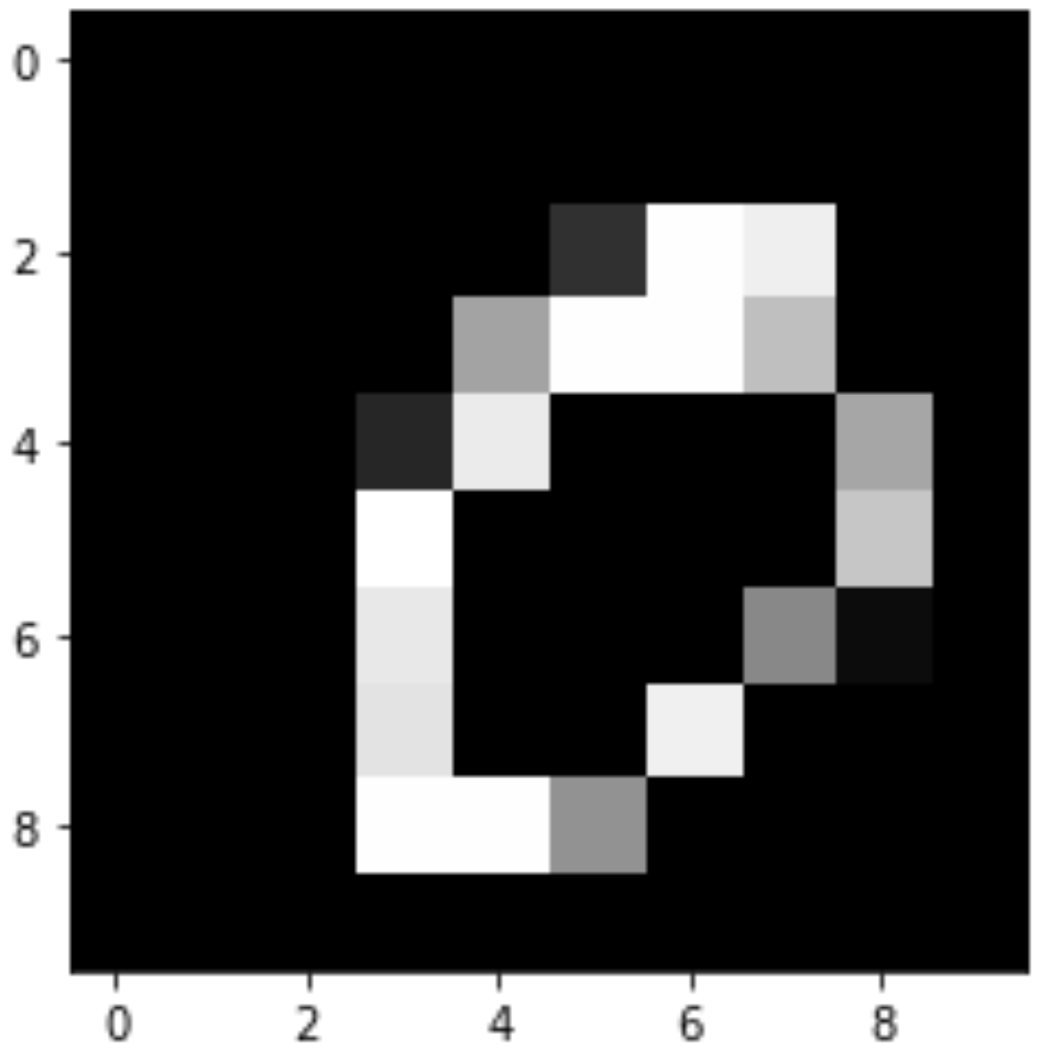
=

массив с элементами от 0 до 1
0 - черный, 1 - белый

Какая размерность?

Бинарная классификация

Input



```
[[0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.1875    , 0.984375  , 0.92578125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.63671875,
  0.984375  , 0.984375  , 0.73828125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.1484375 , 0.91015625,
  0.        , 0.        , 0.        , 0.64453125, 0.        ],
 [0.        , 0.        , 0.        , 0.98828125, 0.        ,
  0.        , 0.        , 0.        , 0.765625  , 0.        ],
 [0.        , 0.        , 0.        , 0.8984375 , 0.        ,
  0.        , 0.        , 0.52734375, 0.046875  , 0.        ],
 [0.        , 0.        , 0.        , 0.87890625, 0.        ,
  0.        , 0.9296875 , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.984375  , 0.984375  ,
  0.56640625, 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ]],
```

x - ч/б картинка

$=$

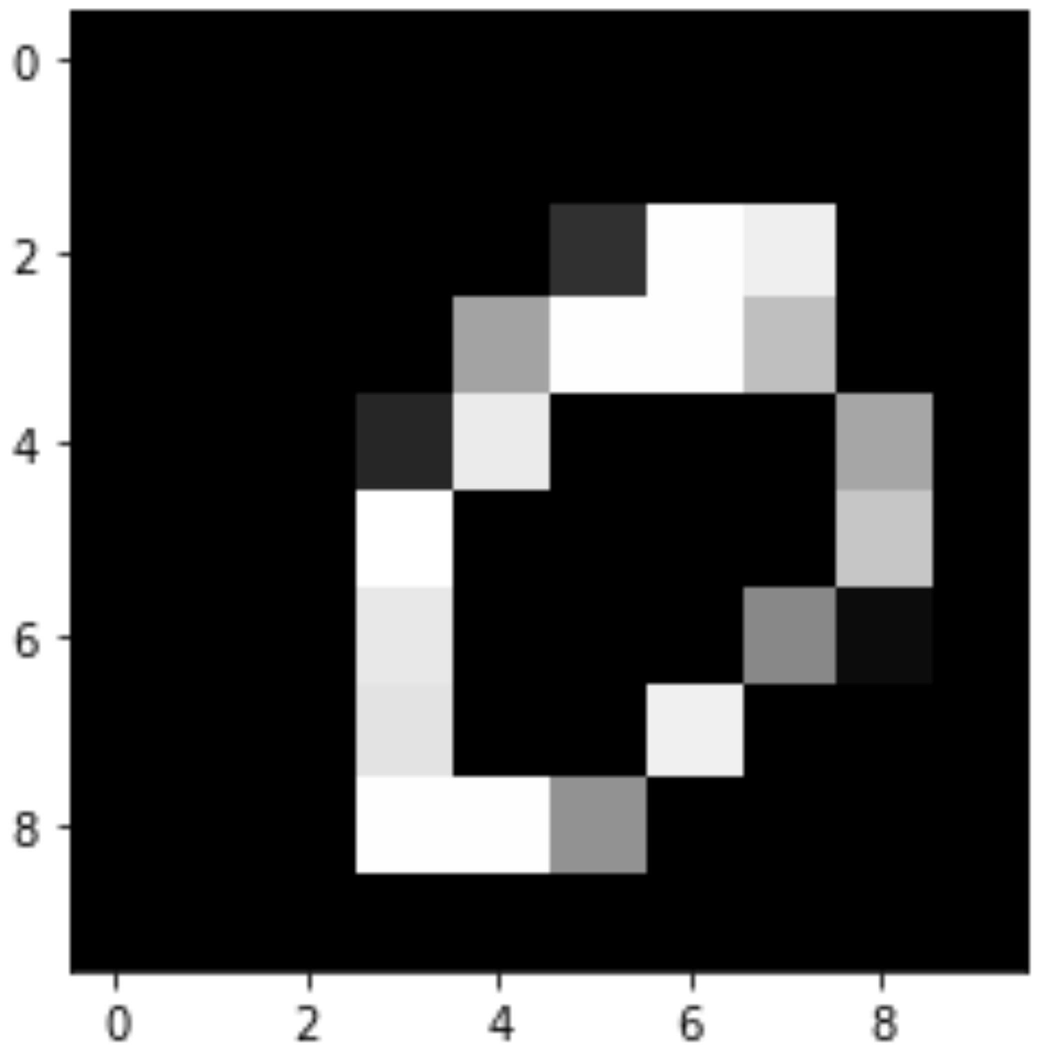
массив с элементами от 0 до 1

0 - черный, 1 - белый

$$H \times W \times C$$

Бинарная классификация

Input




```
[[0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.1875    , 0.984375  , 0.92578125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.63671875,
  0.984375  , 0.984375  , 0.73828125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.1484375 , 0.91015625,
  0.        , 0.        , 0.        , 0.64453125, 0.        ],
 [0.        , 0.        , 0.        , 0.98828125, 0.        ,
  0.        , 0.        , 0.        , 0.765625  , 0.        ],
 [0.        , 0.        , 0.        , 0.8984375 , 0.        ,
  0.        , 0.        , 0.52734375, 0.046875  , 0.        ],
 [0.        , 0.        , 0.        , 0.87890625, 0.        ,
  0.        , 0.9296875 , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.984375  , 0.984375  ,
  0.56640625, 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ]],
```

x - ч/б картинка

$=$

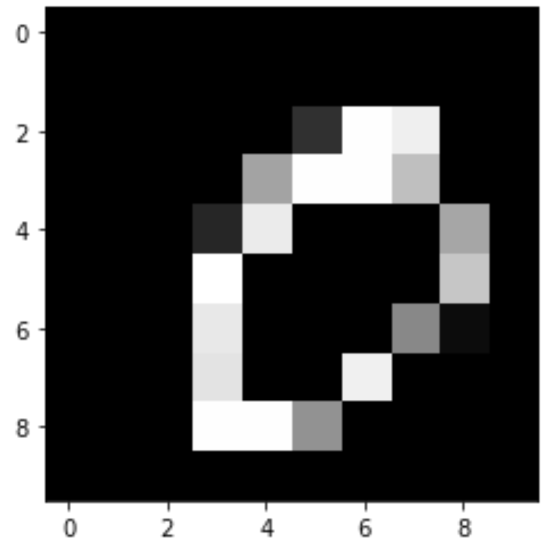
массив с элементами от 0 до 1

0 - черный, 1 - белый

$$H \times W \times C = 10 \times 10 \times 1$$

Бинарная классификация: forward pass

Input



$$x =$$

```
[[0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.1875    , 0.984375  , 0.92578125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.63671875,
  0.984375  , 0.984375  , 0.73828125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.1484375 , 0.91015625,
  0.        , 0.        , 0.        , 0.64453125, 0.        ],
 [0.        , 0.        , 0.        , 0.98828125, 0.        ,
  0.        , 0.        , 0.        , 0.765625  , 0.        ],
 [0.        , 0.        , 0.        , 0.8984375 , 0.        ,
  0.        , 0.        , 0.52734375, 0.046875  , 0.        ],
 [0.        , 0.        , 0.        , 0.87890625, 0.        ,
  0.        , 0.9296875 , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.984375  , 0.984375  ,
  0.56640625, 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ]],
```

$$H \times W \times C = 10 \times 10 \times 1$$

$\longrightarrow$ Linear

H, W, и C — это обозначения, которые часто используются для описания размеров изображений в машинном обучении и компьютерном зрении:

- H (Height) — высота изображения (в пикселях).
- W (Width) — ширина изображения (в пикселях).
- C (Channels) — количество каналов изображения.

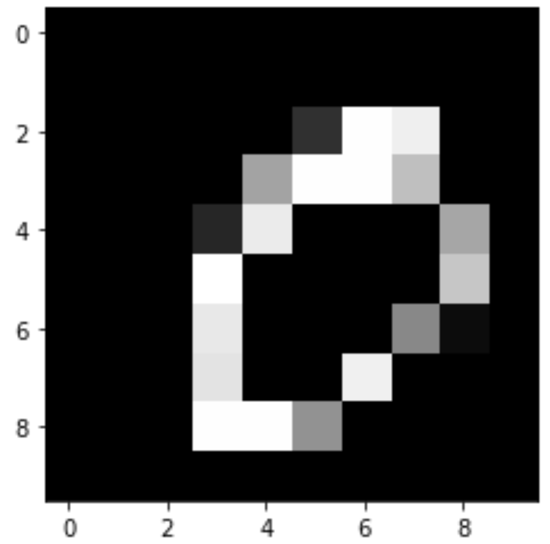
Например:
- Для черно-белого (градаций серого) изображения, C = 1, так как есть только один канал.
- Для цветного изображения в формате RGB, C = 3, так как у нас есть три канала: красный (Red), зеленый (Green) и синий (Blue).

Таким образом, например, изображение размером 128 x 128 x 3 будет иметь:
- Высоту 128 пикселей,
- Ширину 128 пикселей,
- Три цветовых канала (RGB).

В случае черно-белого изображения, размер может быть записан как 128 x 128 x 1.

Бинарная классификация: forward pass

Input



$$x =$$

```
[[0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.1875    , 0.984375  , 0.92578125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.63671875,
  0.984375  , 0.984375  , 0.73828125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.1484375 , 0.91015625,
  0.        , 0.        , 0.        , 0.64453125, 0.        ],
 [0.        , 0.        , 0.        , 0.98828125, 0.        ,
  0.        , 0.        , 0.        , 0.765625  , 0.        ],
 [0.        , 0.        , 0.        , 0.8984375 , 0.        ,
  0.        , 0.        , 0.52734375, 0.046875  , 0.        ],
 [0.        , 0.        , 0.        , 0.87890625, 0.        ,
  0.        , 0.9296875 , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.984375  , 0.984375  ,
  0.56640625, 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ]],
```

$$H \times W \times C = 10 \times 10 \times 1$$

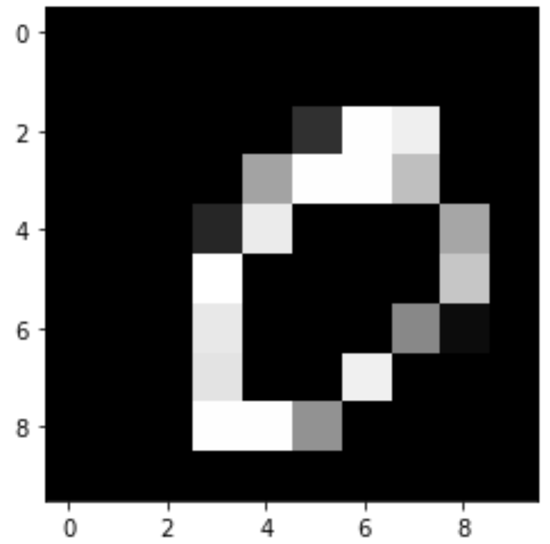
$$\longrightarrow \quad W_1 \cdot x_{input}$$

параметры: W_1 - матрица размера

$d1 \times d2$ - веса линейного слоя

Бинарная классификация: forward pass

Input



$=$ x

```
[[0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.1875    , 0.984375  , 0.92578125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.63671875,
  0.984375  , 0.984375  , 0.73828125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.1484375 , 0.91015625,
  0.        , 0.        , 0.        , 0.64453125, 0.        ],
 [0.        , 0.        , 0.        , 0.98828125, 0.        ,
  0.        , 0.        , 0.        , 0.765625  , 0.        ],
 [0.        , 0.        , 0.        , 0.8984375 , 0.        ,
  0.        , 0.        , 0.52734375, 0.046875  , 0.        ],
 [0.        , 0.        , 0.        , 0.87890625, 0.        ,
  0.        , 0.9296875 , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.984375  , 0.984375  ,
  0.56640625, 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ]],
```

$$H \times W \times C = 10 \times 10 \times 1$$

$\longrightarrow$

$$W_1 \cdot x_{input}$$

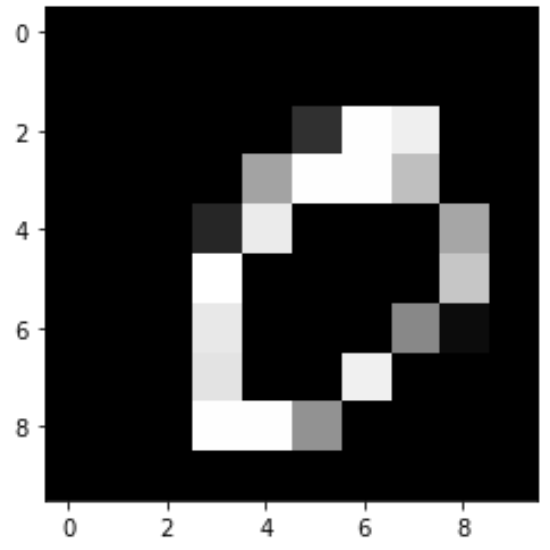
параметры: W_1 - матрица размера

$d1 \times d2$ - веса линейного слоя

Какие параметры выбираем?

Бинарная классификация: forward pass

Input



=

x

```
[[0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.1875    , 0.984375  , 0.92578125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.63671875,
  0.984375  , 0.984375  , 0.73828125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.1484375 , 0.91015625,
  0.        , 0.        , 0.        , 0.64453125, 0.        ],
 [0.        , 0.        , 0.        , 0.98828125, 0.        ,
  0.        , 0.        , 0.        , 0.765625  , 0.        ],
 [0.        , 0.        , 0.        , 0.8984375 , 0.        ,
  0.        , 0.        , 0.52734375, 0.046875  , 0.        ],
 [0.        , 0.        , 0.        , 0.87890625, 0.        ,
  0.        , 0.9296875 , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.984375  , 0.984375  ,
  0.56640625, 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ]],
```

$H \times W \times C = 10 \times 10 \times 1$

$\rightarrow$

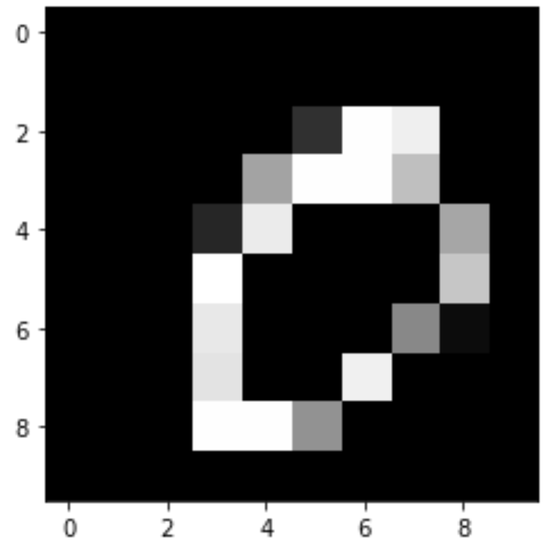
$W_1 \cdot x_{input}$

параметры: W_1 - матрица размера $d1 \times d2$ - веса линейного слоя

Нужно сначала представить картинку в виде вектора размера $H \cdot W \cdot C = 10 \cdot 10 \cdot 1 = 100$

Бинарная классификация: forward pass

Input



$$x =$$

```
[[0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.1875    , 0.984375  , 0.92578125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.63671875,
  0.984375  , 0.984375  , 0.73828125, 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.1484375 , 0.91015625,
  0.        , 0.        , 0.        , 0.64453125, 0.        ],
 [0.        , 0.        , 0.        , 0.98828125, 0.        ,
  0.        , 0.        , 0.        , 0.765625  , 0.        ],
 [0.        , 0.        , 0.        , 0.8984375 , 0.        ,
  0.        , 0.        , 0.52734375, 0.046875  , 0.        ],
 [0.        , 0.        , 0.        , 0.87890625, 0.        ,
  0.        , 0.9296875 , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.984375  , 0.984375  ,
  0.56640625, 0.        , 0.        , 0.        , 0.        ],
 [0.        , 0.        , 0.        , 0.        , 0.        ,
  0.        , 0.        , 0.        , 0.        , 0.        ]],
```

$$H \times W \times C = 10 \times 10 \times 1$$

$$\rightarrow$$

```
[0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.1875,
0.984375,
0.92578125,
0.0,
```

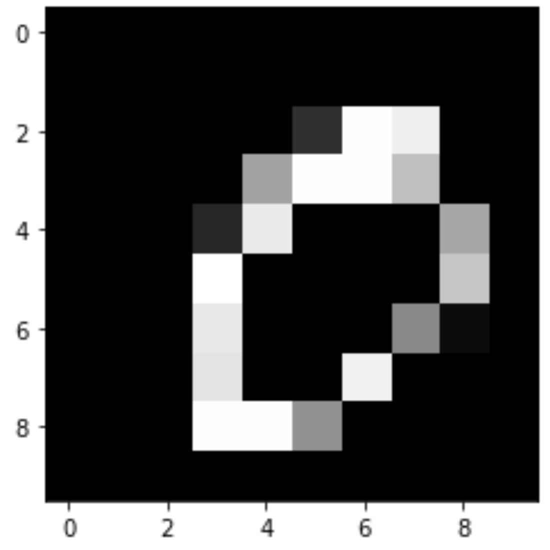
$$H \cdot W \cdot C = 10 \cdot 10 \cdot 1 = 100$$

Операция flatten

Операция flatten (сглаживание) — это преобразование многомерного массива (тензора) в одномерный вектор. В контексте нейронных сетей и обработки изображений, это полезная операция, когда нужно перейти от двумерного или трехмерного изображения к одномерному вектору, чтобы передать его на вход полносвязного слоя нейронной сети.

Бинарная классификация: forward pass

Input



x

```
[0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.1875,
0.984375,
0.92578125,
0.0,
```

$$H \cdot W \cdot C = 10 \cdot 10 \cdot 1 = 100$$

$$W_1 \cdot x_{input}$$

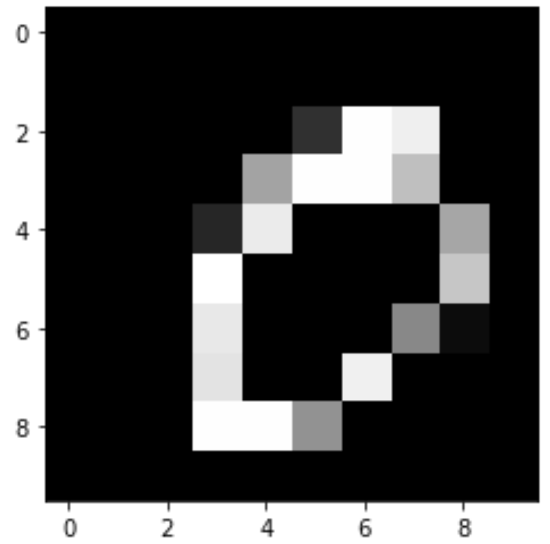
параметры: W_1 - матрица размера

$d1 \times d2$ - веса линейного слоя

Какие параметры выбираем?

Бинарная классификация: forward pass

Input



$\rightarrow$

x

```
[0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.1875,
0.984375,
0.92578125,
0.0,
```

$$H \cdot W \cdot C = 10 \cdot 10 \cdot 1 = 100$$

$\rightarrow$

$$W_1 \cdot x_{input}$$

параметры: W_1 - матрица размера

128×100 - веса линейного слоя

Бинарная классификация: forward pass

Input

x

```
[0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.1875,
0.984375,
0.92578125,
0.0,
```

- вектор размера

$H \cdot W \cdot C = 10 \cdot 10 \cdot 1 = 100$

$\rightarrow$

$W_1 \cdot x_{input}$

параметры: W_1 - матрица размера

128×100 - веса линейного слоя

$\rightarrow$

Output текущего слоя

x_{output_1}

- вектор размера 128

Бинарная классификация: forward pass

Input

x

```
[0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.0,
0.1875,
0.984375,
0.92578125,
0.0,
```

- вектор размера

$H \cdot W \cdot C = 10 \cdot 10 \cdot 1 = 100$

$W_1 \cdot x_{input}$

параметры: W_1 - матрица размера

128×100 - веса линейного слоя

Output текущего слоя

x_{output_1}

- вектор размера 128

Передаем дальше

Бинарная классификация: forward pass

Input текущего слоя

x_{output_1}

- вектор размера 128

$\rightarrow$

Бинарная классификация: forward pass

Input текущего слоя

Output текущего слоя

x_{output_1} - вектор размера 128 $\rightarrow$ Sigmoid $\rightarrow$ x_{output_2}

Бинарная классификация: forward pass

Input текущего слоя

Output текущего слоя

x_{output_1} - вектор размера 128

$$\rightarrow \quad \frac{1}{1 + e^{-x_{output_1}}} \quad \rightarrow$$

x_{output_2}

Бинарная классификация: forward pass

Input текущего слоя

Output текущего слоя

x_{output_1} - вектор размера 128

$\rightarrow$

$$\frac{1}{1+e^{-x_{output_1}}}$$

$\rightarrow$

x_{output_2} - вектор размера 128

Бинарная классификация: forward pass

Input текущего слоя

Output текущего слоя

x_{output_2} - вектор размера 128 $\rightarrow$ Linear $\rightarrow$

Бинарная классификация: forward pass

Input текущего слоя

Output текущего слоя

x_{output_2} - вектор размера 128 $\rightarrow$ $W_2 \cdot x_{output_2}$ $\rightarrow$

Какой размер W_2 выбираем?

Бинарная классификация: forward pass

Input текущего слоя

Output текущего слоя

x_{output_2} - вектор размера 128

$\rightarrow$

$W_2 \cdot x_{output_2}$

$\rightarrow$

W_2 размера 1×128

Бинарная классификация: forward pass

Input текущего слоя

Output текущего слоя

x_{output_2} - вектор размера 128 $\rightarrow$ $W_2 \cdot x_{output_2}$ $\rightarrow$ x_{output_3} - число, размер 1

W_2 размера 1×128

Хотим оценить $P(y = 1 \mid x, \theta)$

Бинарная классификация: forward pass

Input текущего слоя

Output текущего слоя

x_{output_3} - число, размер 1 $\longrightarrow$ Sigmoid $\longrightarrow$ $P(y = 1 \mid x, \theta)$

Бинарная классификация: forward pass

Input текущего слоя

Output текущего слоя

x_{output_3} - число, размер 1

$$\rightarrow \quad \frac{1}{1+e^{-x_{output_3}}} \quad \rightarrow \quad P(y=1\,|\,x,\theta)$$

Бинарная классификация: forward pass

Input текущего слоя

Output текущего слоя

x_{output_3} - число, размер 1

$$\rightarrow \frac{1}{1+e^{-x_{output_3}}} \rightarrow P(y=1\,|\,x,\theta)$$

Функция потерь

$$L(\mathbf{y},\hat{\mathbf{y}}) = -\left[y \log P(y=1\,|\,x,\theta) + (1-y)\log(1-P(y=1\,|\,x,\theta))\right]$$

Бинарная классификация: backward pass

Input текущего слоя

Output текущего слоя

x_{output_3} - число, размер 1

$$\rightarrow \quad \frac{1}{1+e^{-x_{output_3}}} \quad \rightarrow \quad P(y=1\,|\,x,\theta)$$

Хотим посчитать $\frac{\mathrm{d}L}{\mathrm{d}W_1}, \frac{\mathrm{d}L}{\mathrm{d}W_2}$

и сделать шаг градиентного спуска

$$W_{new_i} = W_i - \alpha\frac{\mathrm{d}L}{\mathrm{d}W_i}$$

Функция потерь

$$L(\mathbf{y},\hat{\mathbf{y}}) = -\left[y\log P(y=1\,|\,x,\theta) + (1-y)\log(1-P(y=1\,|\,x,\theta))\right]$$

Бинарная классификация: backward pass

Input текущего слоя

Output текущего слоя

x_{output_3} - число, размер 1

$$\rightarrow \quad \frac{1}{1+e^{-x_{output_3}}} \quad \rightarrow \quad P(y=1\,|\,x,\theta)$$

$$\frac{\mathrm{d}L}{\mathrm{d}L} = 1$$

Функция потерь

$$L(\mathbf{y},\hat{\mathbf{y}}) = -\left[y\log P(y=1\,|\,x,\theta) + (1-y)\log(1-P(y=1\,|\,x,\theta))\right]$$

Бинарная классификация: backward pass

Input текущего слоя

Output текущего слоя

x_{output_3} - число, размер 1

$$\rightarrow \quad \frac{1}{1+e^{-x_{output_3}}} \quad \rightarrow \quad P(y=1\,|\,x,\theta)$$

$$\frac{\mathrm{d}L}{\mathrm{d}L} = 1$$

Функция потерь

$$L(\mathbf{y},\hat{\mathbf{y}}) = -\left[y\log P(y=1\,|\,x,\theta) + (1-y)\log(1-P(y=1\,|\,x,\theta))\right]$$

$$\frac{\mathrm{d}L}{\mathrm{d}P} = -\frac{y}{P} + \frac{1-y}{1-P}$$

Бинарная классификация: backward pass

Input текущего слоя

Output текущего слоя

x_{output_3} - число, размер 1

$$\rightarrow \frac{1}{1+e^{-x_{output_3}}} \rightarrow P(y=1\,|\,x,\theta)$$

$$\leftarrow$$

$$\frac{\mathrm{d}L}{\mathrm{d}P} = -\frac{y}{P} + \frac{1-y}{1-P}$$

Бинарная классификация: backward pass

Input текущего слоя

Output текущего слоя

x_{output_3}

- число, размер 1

$\rightarrow$

$$\frac{1}{1 + e^{-x_{output_3}}}$$

$\rightarrow$ $P(y = 1 \mid x, \theta)$

$\leftarrow$

$$\frac{\mathrm{d}P}{\mathrm{d}x_{output_3}} = P(1 - P)$$

$$\frac{\mathrm{d}L}{\mathrm{d}P} = -\frac{y}{P} + \frac{1 - y}{1 - P}$$

Бинарная классификация: backward pass

Input текущего слоя

Output текущего слоя

x_{output_3}

- число, размер 1

$\longrightarrow$ $\dfrac{1}{1+e^{-x_{output_3}}}$ $\longrightarrow$ $P(y=1\,|\,x,\theta)$

$$\frac{\mathrm{d}L}{\mathrm{d}x_{output_3}} = \frac{\mathrm{d}L}{\mathrm{d}P}\frac{\mathrm{d}P}{\mathrm{d}x_{output_3}}$$

$$\frac{\mathrm{d}P}{\mathrm{d}x_{output_3}} = P(1-P)$$

$$\frac{\mathrm{d}L}{\mathrm{d}P} = -\frac{y}{P} + \frac{1-y}{1-P}$$

Бинарная классификация: backward pass

Input текущего слоя

x_{output_2} - вектор размера 128

$\rightarrow$

$W_2 \cdot x_{output_2}$

$\rightarrow$

$\leftarrow$

Output текущего слоя

x_{output_3} - число, размер 1

$$\frac{\mathrm{d}L}{\mathrm{d}x_{output_3}} = \frac{\mathrm{d}L}{\mathrm{d}P}\frac{\mathrm{d}P}{\mathrm{d}x_{output_3}}$$

Бинарная классификация: backward pass

Input текущего слоя

Output текущего слоя

x_{output_2} - вектор размера 128

$\rightarrow$

$W_2 \cdot x_{output_2}$

$\rightarrow$

$\leftarrow$

x_{output_3} - число, размер 1

$$\frac{\mathrm{d}x_{output_3}}{\mathrm{d}W_2} = x^T_{output_2}$$

$$\frac{\mathrm{d}L}{\mathrm{d}x_{output_3}}$$

Бинарная классификация: backward pass

Input текущего слоя

x_{output_2} - вектор размера 128

$\rightarrow$

$W_2 \cdot x_{output_2}$

$\rightarrow$

$\leftarrow$

Output текущего слоя

x_{output_3} - число, размер 1

$$\frac{\mathrm{d}x_{output_3}}{\mathrm{d}W_2} = x^T_{output_2}$$

$$\frac{\mathrm{d}L}{\mathrm{d}x_{output_3}}$$

$$\frac{\mathrm{d}L}{\mathrm{d}W_2} = \frac{\mathrm{d}L}{\mathrm{d}x_{output_3}} \frac{\mathrm{d}x_{output_3}}{\mathrm{d}W_2} = \frac{\mathrm{d}L}{\mathrm{d}x_{output_3}} x^T_{output_2}$$

Бинарная классификация: backward pass

Input текущего слоя

Output текущего слоя

x_{output_2} - вектор размера 128 $\rightarrow$ $W_2 \cdot x_{output_2}$ $\rightarrow$ x_{output_3} - число, размер 1

$\leftarrow$

$$\frac{dx_{output_3}}{dW_2} = x^T_{output_2} \qquad \frac{dL}{dx_{output_3}}$$

$$\frac{dL}{dW_2} = \frac{dL}{dx_{output_3}} \frac{dx_{output_3}}{dW_2} = \frac{dL}{dx_{output_3}} x^T_{output_2} \Rightarrow W_{new_2} = W_2 - \alpha \frac{dL}{dW_2}$$

Бинарная классификация: backward pass

Input текущего слоя

Output текущего слоя

x_{output_2} – вектор размера 128

$\rightarrow$

$\leftarrow$

$W_2 \cdot x_{output_2}$

$\rightarrow$

$\leftarrow$

x_{output_3} – число, размер 1

$$\frac{\mathrm{d}x_{output_3}}{\mathrm{d}x_{output_2}} = W_2^T$$

$$\frac{\mathrm{d}L}{\mathrm{d}x_{output_3}}$$

Бинарная классификация: backward pass

Input текущего слоя

Output текущего слоя

x_{output_2} - вектор размера 128

$\rightarrow$

$\leftarrow$

$W_2 \cdot x_{output_2}$

$\rightarrow$

$\leftarrow$

x_{output_3} - число, размер 1

$$\frac{\mathrm{d}L}{\mathrm{d}x_{output_2}} = \frac{\mathrm{d}L}{\mathrm{d}x_{output_3}} \frac{\mathrm{d}x_{output_3}}{\mathrm{d}x_{output_2}}$$

$$\frac{\mathrm{d}x_{output_3}}{\mathrm{d}x_{output_2}} = W_2^T$$

$$\frac{\mathrm{d}L}{\mathrm{d}x_{output_3}}$$

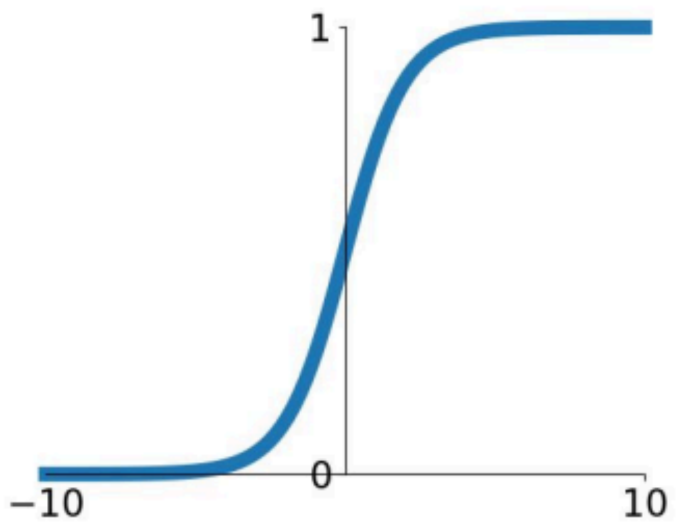
Train loop

```
for epoch in range(epochs):                # эпоха - проход по датасету
    model.train()                  # переключаем все в режим тренировки (DO/BN/...)
    for x, gt in tqdm(train_loader): # датасет разбит на (мини)батчи
        logits = network.forward(x)  # предсказания сети
        loss = loss_fn(logits, gt)   # подсчет ошибки
        accuracy = accuracy(logits, gt) # подсчет метрик
        loss.backward()              # подсчет градиентов
        network.apply_updates()      # обновление весов
    model.eval()                   # переключаем все в режим валидации (DO/BN/...)
    for x, gt in tqdm(val_loader):   # валидация
        logits = network.forward(x)
        loss = loss_fn(logits, gt)      # подсчет ошибок
        accuracy = accuracy(logits, gt) # подсчет метрик
```

Функции активации

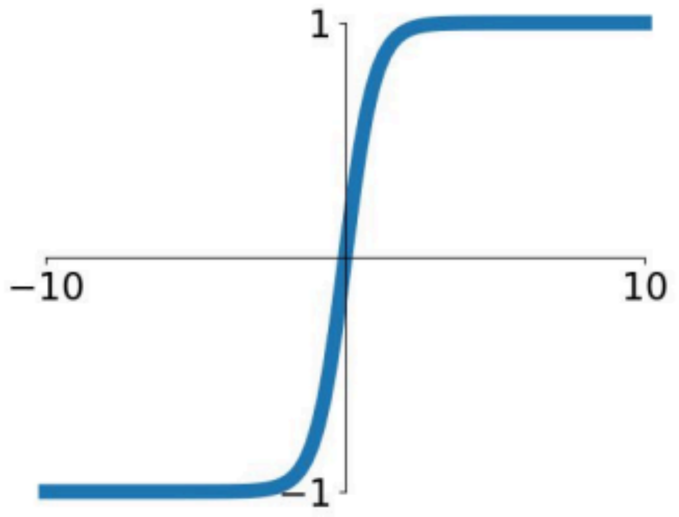
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



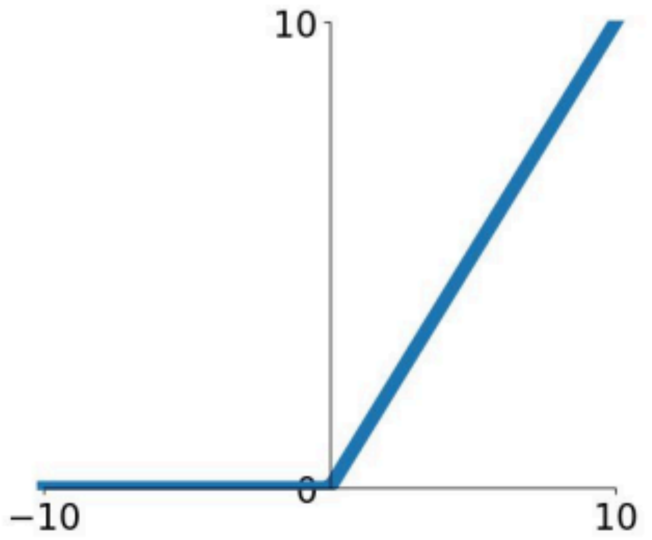
tanh

$\tanh(x)$



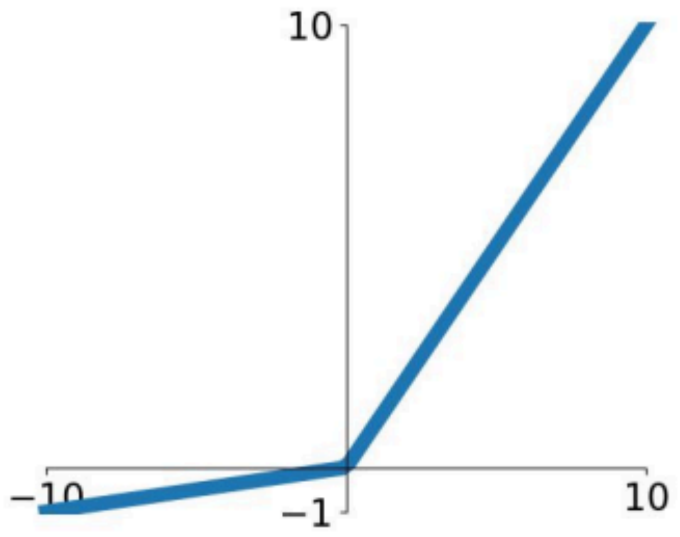
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$



Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

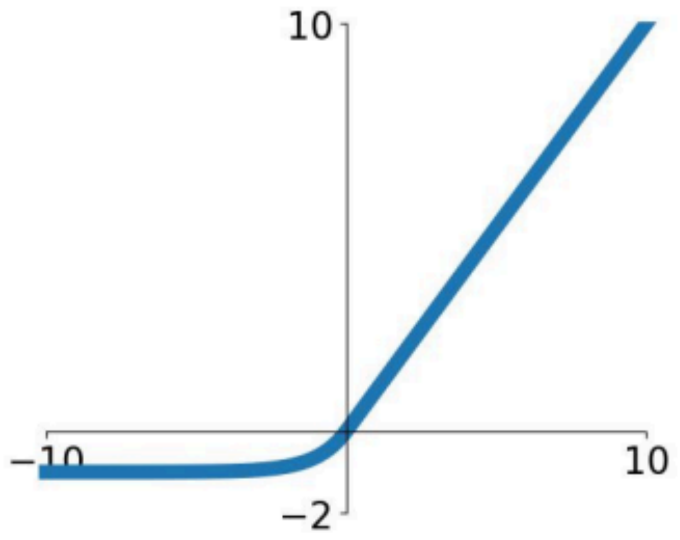
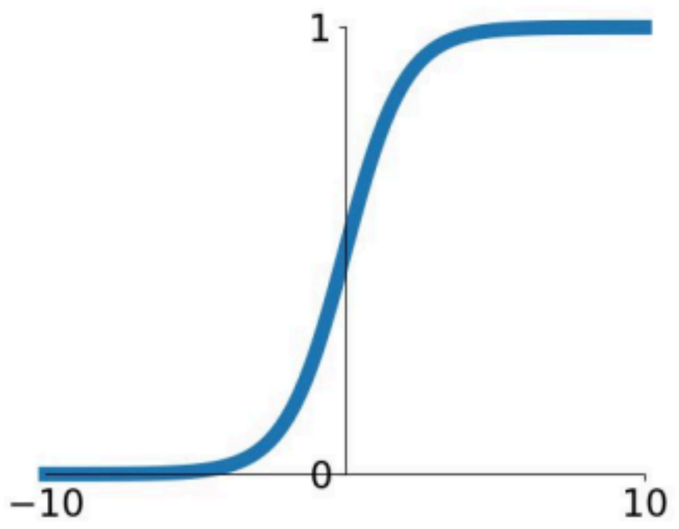


Image credit

Функции активации: Sigmoid

Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



Выход в диапазоне от 0 до 1

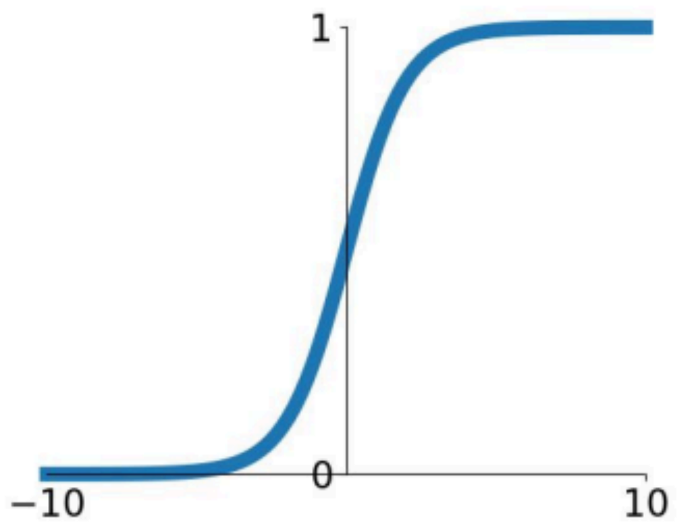
Проблемы?

Image credit

Функции активации: Sigmoid

Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



Выходы в диапазоне от 0 до 1

производная становится очень малой для больших и малых значений входа. Это вызывает затухание градиентов, что замедляет или останавливает обучение в этих областях

есть нейрон с сигмоидной активацией, и его входное значение xx очень велико (например, 10). В этом случае: Сигмоидная функция даст значение, близкое к 1. Локальный градиент будет очень маленьким (близким к нулю). Это значит, что даже если ошибка большая, вес, связанный с этим нейроном, обновится на незначительное количество, потому что градиент ошибки очень мал. Аналогичная ситуация возникает, когда xx очень мало (например, x=–10), и выход сигмоиды приближается к 0. Это называется затуханием градиентов (vanishing gradients), потому что градиенты становятся настолько малы, что теряют своё влияние на обновление весов при обратном распространении

Выходы не центрированы, т е сигмоида всегда дает значения в диапазоне от 0 до 1, а не от -1 до 1 или вокруг 0. Поскольку выходы сигмоиды всегда положительные, это влияет на вычисление градиентов. Когда значения активаций на каждом шаге обратного распространения ошибки всегда положительные, это может вызывать смещение в направлении градиента.
В частности, это может замедлить обучение, потому что градиенты на каждом слое будут постоянно смещены в одну сторону (например, к увеличению весов), что затрудняет эффективное обновление весов

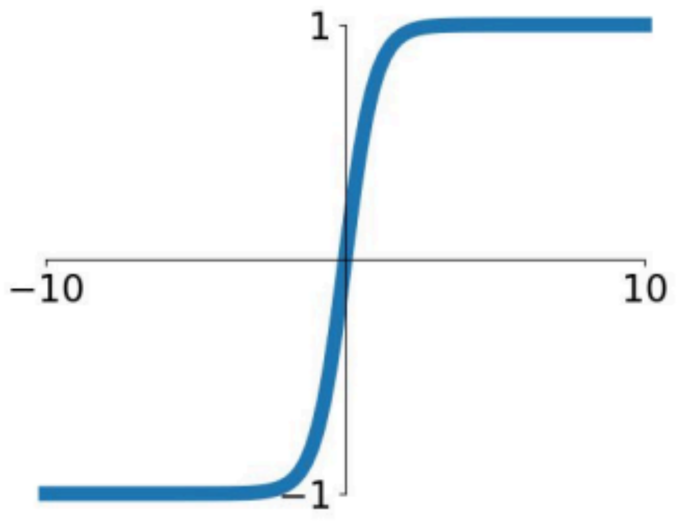
- На краях одинаково работает
- Local grad на краях маленький
- Выходы не центрированы

Image credit

Функции активации: Tanh

tanh

$\tanh(x)$



Выходы в диапазоне от -1 до 1

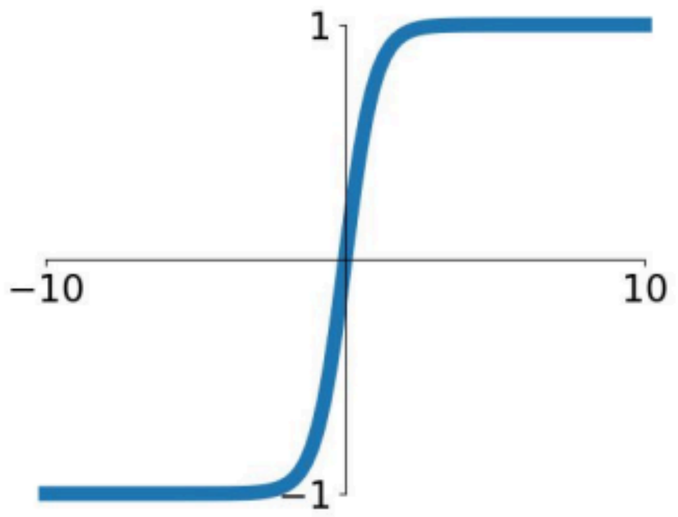
- Выходы центрированы

Image credit

Функции активации: Tanh

tanh

$\tanh(x)$



Выходы в диапазоне от -1 до 1

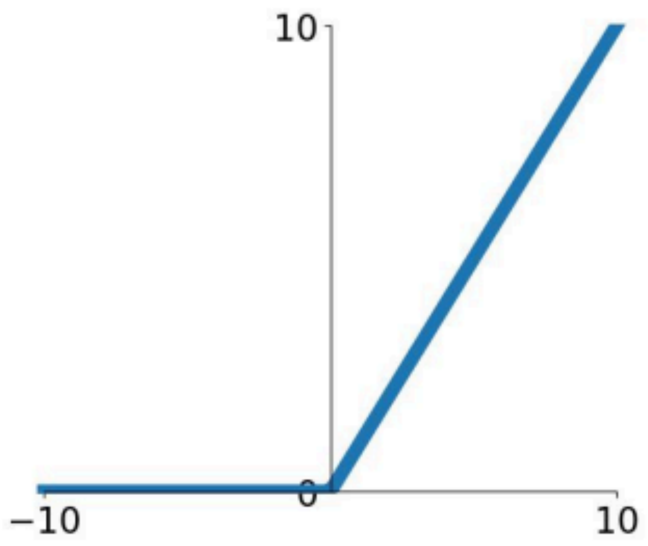
- На краях одинаково работает
- Local grad на краях маленький
- Выходы центрированы

Image credit

Функции активации: ReLU

ReLU

$$\max(0, x)$$



Для её вычисления не требуется экспоненциальная функция. Это делает ReLU значительно проще и быстрее в вычислении.

Выходы в диапазоне от 0 до +inf

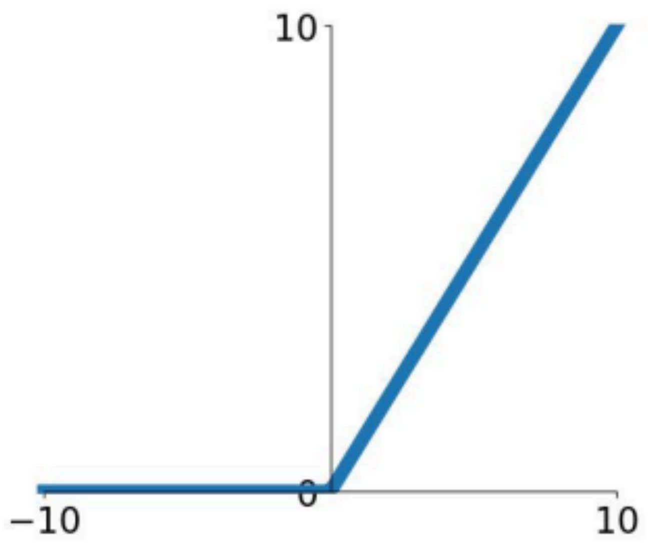
- На краях работает по-разному
- Нет вычислений exp
- Быстрая сходимость (x6)

Image credit

Функции активации: ReLU

ReLU

$$\max(0, x)$$



Выходы в диапазоне от 0 до +inf

- На краях работает по-разному
- Нет вычислений exp
- Быстрая сходимость (x6)

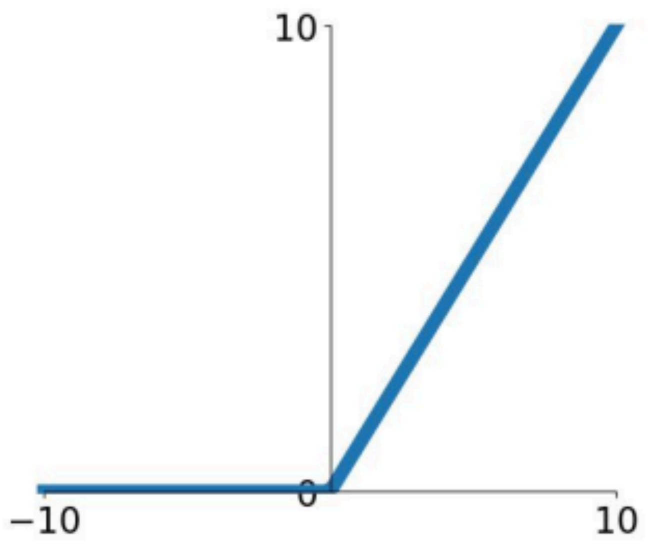
- Выходы не центрированы
- Local grad для x < 0

Image credit

Функции активации: ReLU

ReLU

$\max(0, x)$



Выходы в диапазоне от 0 до +inf

- На краях работает по-разному
- Нет вычислений exp
- Быстрая сходимость (x6)
- Выходы не центрированы
- Local grad для x < 0

Проблема "мертвых" нейронов: Когда входы в ReLU всегда отрицательны, градиенты для этих нейронов равны нулю. Это приводит к тому, что веса, связанные с этими нейронами, не обновляются во время обучения. Этот эффект называют "мертвыми" нейронами (dead neurons). Нейроны с нулевыми градиентами не участвуют в обновлении весов, что может затруднить обучение и привести к менее эффективной модели.

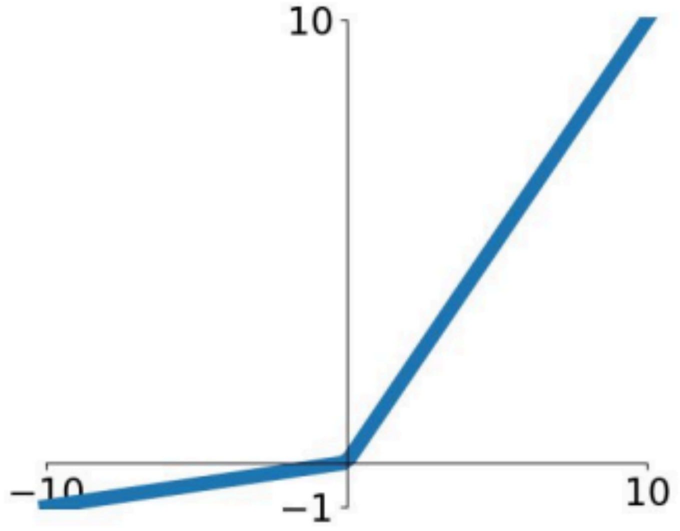
Хороший выбор, но learning rate не должен быть большим

Image credit

Функции активации: Leaky ReLU

Leaky ReLU

$$\max(0.1x, x)$$



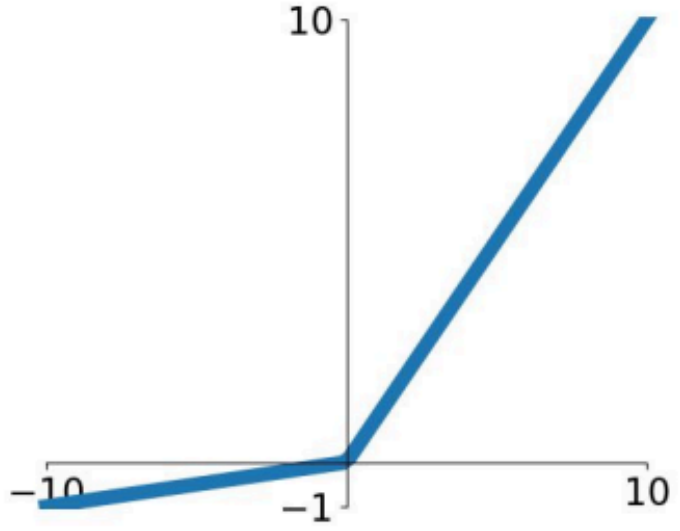
Выходы в диапазоне от -inf до +inf

Image credit

Функции активации: Leaky ReLU

Leaky ReLU

$$\max(0.1x, x)$$



Выходы в диапазоне от -inf до +inf

- На краях работает по-разному
- Нет вычислений exp
- Быстрая сходимость

Image credit

Функции активации: ELU

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

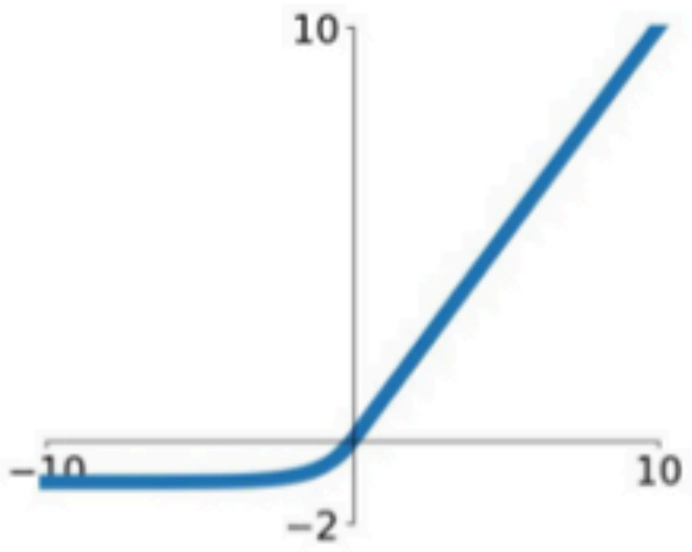
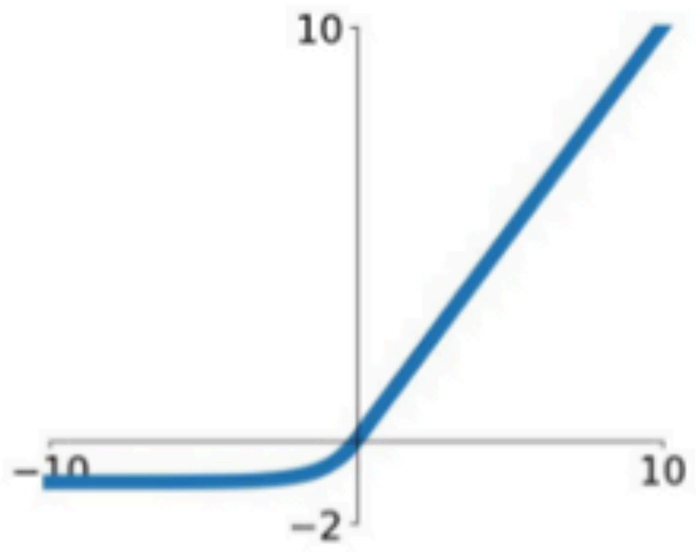


Image credit

Функции активации: ELU

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Выходы в диапазоне от -inf до +inf

- На краях работает по-разному
- Быстрая сходимость
- Вычисляем exp

Image credit

Функции активации: вывод

ReLU - хороший базовый выбор

Можно пробовать **LeakyReLU**, **ELU**, **GELU**, etc.

Избегать **Sigmoid**

Функции активации: вывод

ReLU - хороший базовый выбор

Можно пробовать **LeakyReLU**, **ELU**, **GELU**, etc.

Избегать **Sigmoid**

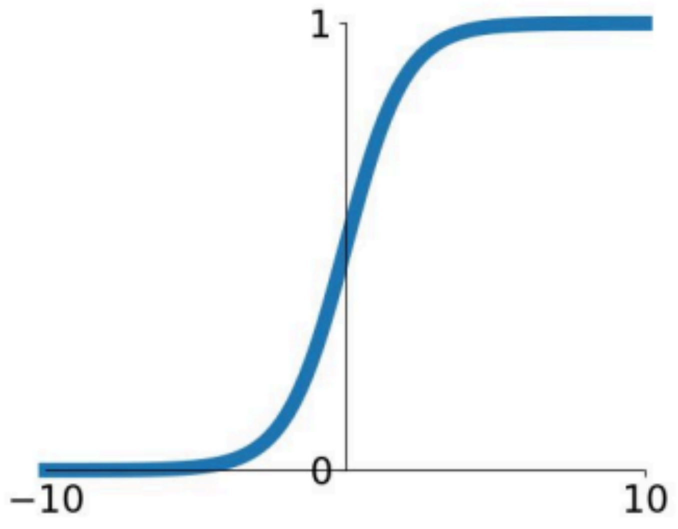
Важно - подбирать lr, инициализации весов…

Виды слоев в нейросетях

Функции активации

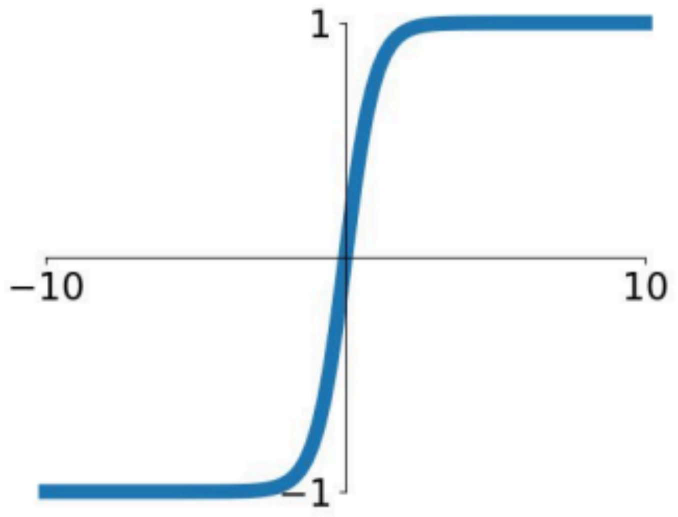
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



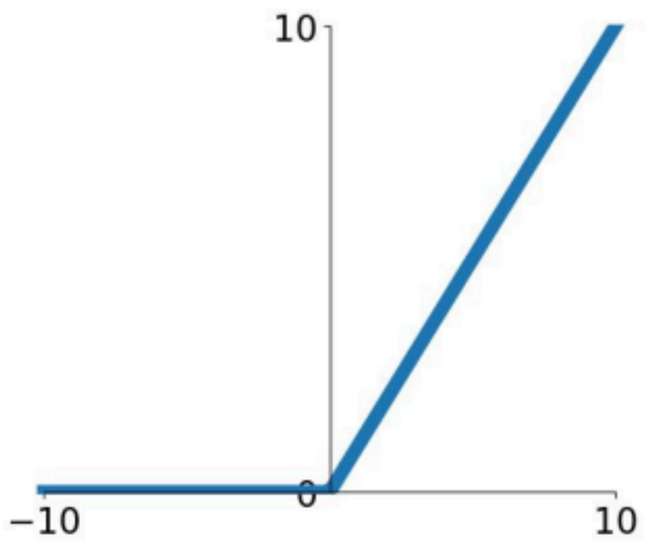
tanh

$\tanh(x)$



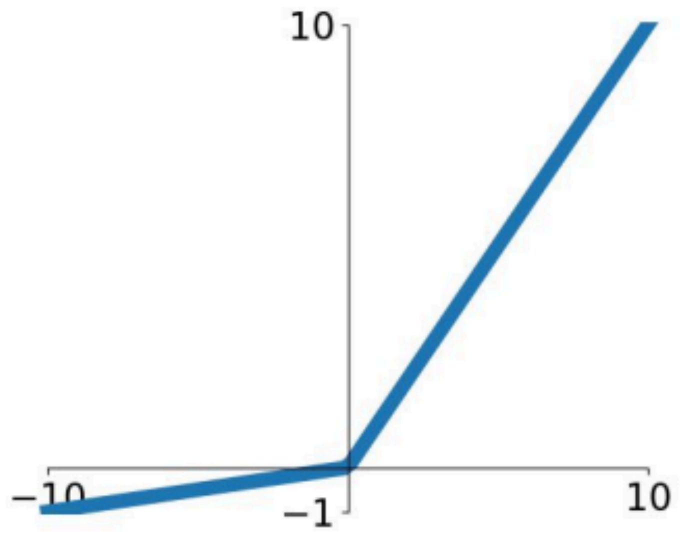
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$



Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

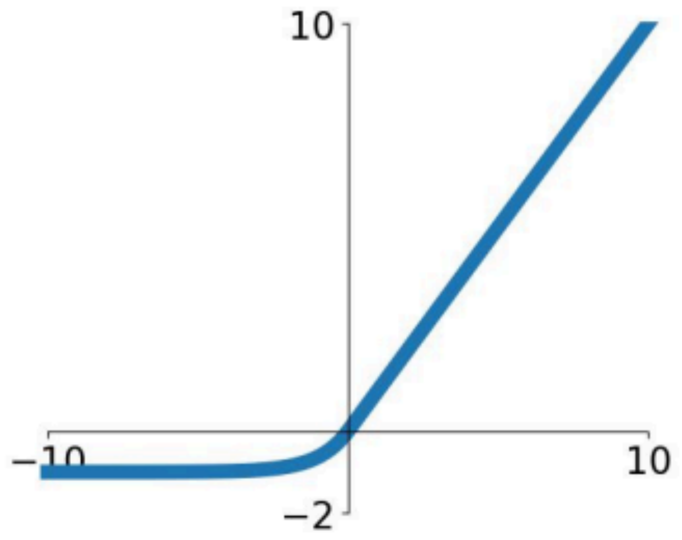


Image credit

Функции активации: вывод

ReLU - хороший базовый выбор

Можно пробовать **LeakyReLU**, **ELU**, **GELU**, etc.

Избегать **Sigmoid**

Важно - подбирать lr, инициализации весов…

Инициализация

Инициализация

Какие значения выбрать при построении сети для весов?

Инициализация

Инициализация нулями?

Инициализация

Инициализация нулями?

Градиентный спуск: $$\theta_{t+1} = \theta_t - \alpha \frac{\mathrm{d}L}{\mathrm{d}\theta}$$

Веса будут меняться одинаково!

Инициализация

Инициализация случайными значениями

А есть значения слишком большие?

Инициализация

Инициализация случайными значениями

А есть значения слишком большие?

Рассмотрим MLP с L слоями Используем Identity активацию*

$$W_1 = W_2 = \ldots = W_L = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix} = 1.5 \cdot I$$

$$y_L = W_L \cdot \ldots \cdot W_1 \cdot x = 1.5^L x$$

*функция активации, которая просто возвращает входное значение без каких-либо изменений. Это одна из самых простых функций активации и используется в некоторых специфических ситуациях в нейронных сетях. Id(x) = x;

Инициализация

Инициализация случайными значениями

А есть значения слишком большие?

Рассмотрим MLP с L слоями Используем Identity активацию

$$W_1 = W_2 = \ldots = W_L = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix} = 1.5 \cdot I$$

$$y_L = W_L \cdot \ldots \cdot W_1 \cdot x = 1.5^L x$$

Backward pass: exploding gradients

Проблема, которая может возникнуть в процессе обучения нейронных сетей, особенно в глубоких сетях или рекуррентных нейронных сетях (RNN). Взрыв градиентов возникает, когда градиенты становятся чрезвычайно большими, что приводит к нестабильности и ухудшению процесса обучения

Инициализация

Инициализация **небольшими** случайными значениями

Инициализация

Инициализация **небольшими** случайными значениями

Рассмотрим MLP с L слоями Используем Identity активацию

$$W_1 = W_2 = \ldots = W_L = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} = 0.5 \cdot I$$

$$y_L = W_L \cdot \ldots \cdot W_1 \cdot x = 0.5^L x$$

Инициализация

Инициализация **небольшими** случайными значениями

Рассмотрим MLP с L слоями Используем Identity активацию

$$W_1 = W_2 = \ldots = W_L = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} = 0.5 \cdot I$$

$$y_L = W_L \cdot \ldots \cdot W_1 \cdot x = 0.5^L x$$

Backward pass: vanishing gradients

Vanishing gradients (затухание градиентов) — это проблема, которая возникает, когда градиенты становятся слишком маленькими при обратном распространении, что затрудняет обучение нейронных сетей.

Инициализация

Инициализация **небольшими** случайными значениями

Поможет калиброванная инициализация: Xavier/Glorot init, He init

Инициализация

Инициализация **небольшими** случайными значениями

Поможет калиброванная инициализация: Xavier/Glorot init, He init

Идея:

- Mean выходов слоев должны быть 0 $\mathrm{E}\ y_{L-1} = \mathrm{E}\ y_L = 0$
- Variance выходов слоев должны быть одинаковыми $\mathrm{Var}\ y_{L-1} = \mathrm{Var}\ y_L$

Стабильные выходы слоев, близкие к нулю и со стабильной дисперсией, способствуют более равномерному и стабильному обновлению весов

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию

$$\mathrm{Var}[y_i] = \mathrm{Var}[w_i x_i] = \mathbb{E}\left[w_i^2 x_i^2\right] - \left(\mathbb{E}[w_i x_i]\right)^2 =$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию

$$\mathrm{Var}[y_i] = \mathrm{Var}[w_i x_i] = \mathbb{E}\left[w_i^2 x_i^2\right] - \left(\mathbb{E}[w_i x_i]\right)^2 =$$
$$= \mathbb{E}[x_i]^2 \,\mathrm{Var}[w_i] + \mathbb{E}[w_i]^2 \,\mathrm{Var}[x_i] + \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$$

Формула для дисперсии произведения независимых с.в.

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию

$$\mathrm{Var}[y_i] = \mathrm{Var}[w_i x_i] = \mathbb{E}\left[w_i^2 x_i^2\right] - \left(\mathbb{E}[w_i x_i]\right)^2 =$$
$$= \cancel{\mathbb{E}[x_i]}^2 \mathrm{Var}[w_i] + \cancel{\mathbb{E}[w_i]}^2 \mathrm{Var}[x_i] + \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$$

Потребуем, чтобы мат.ожидания были 0

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона:

$$\mathrm{Var}[y] = \mathrm{Var}\left[\sum_{i=1}^{n_{\mathrm{out}}} y_i\right] = \sum_{i=1}^{n_{\mathrm{out}}} \mathrm{Var}[w_i x_i] = n_{\mathrm{out}}\ \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона:

$$\mathrm{Var}[y] = \mathrm{Var}\left[\sum_{i=1}^{n_{\text{out}}} y_i\right] = \sum_{i=1}^{n_{\text{out}}} \mathrm{Var}[w_i x_i] = n_{\text{out}}\ \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$$

Дисперсия выхода

Дисперсия входа

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона:

$$\mathrm{Var}[y] = \mathrm{Var}\left[\sum_{i=1}^{n_{\text{out}}} y_i\right] = \sum_{i=1}^{n_{\text{out}}} \mathrm{Var}[w_i x_i] = n_{\text{out}}\ \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$$

Дисперсия выхода = Дисперсия входа

хотим

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона:

должно быть = 1

$$\mathrm{Var}[y] = \mathrm{Var}\left[\sum_{i=1}^{n_{\mathrm{out}}} y_i\right] = \sum_{i=1}^{n_{\mathrm{out}}} \mathrm{Var}[w_i x_i] = n_{\mathrm{out}}\,\mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$$

Дисперсия выхода = Дисперсия входа

хотим

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона: $n_{\mathrm{out}}\,\mathrm{Var}[w_i] = 1$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона: $n_{\mathrm{out}}\,\mathrm{Var}[w_i] = 1$

$$\mathrm{Var}[w_i] = \frac{1}{n_{\mathrm{out}}}$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона: $n_{\mathrm{out}}\,\mathrm{Var}[w_i] = 1$

$$\mathrm{Var}[w_i] = \frac{1}{n_{\mathrm{out}}}$$

А есть тоже самое для backward pass?

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Forward pass:

$$n_{\text{out}} \operatorname{Var}[w_i] = 1$$

$$\operatorname{Var}[w_i] = \frac{1}{n_{\text{out}}}$$

Backward pass:

$$n_{\text{in}} \operatorname{Var}[w_i] = 1$$

$$\operatorname{Var}[w_i] = \frac{1}{n_{\text{in}}}$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Forward pass:

$$n_{\text{out}} \operatorname{Var}[w_i] = 1$$

$$\operatorname{Var}[w_i] = \frac{1}{n_{\text{out}}}$$

Возьмем среднее

Backward pass:

$$n_{\text{in}} \operatorname{Var}[w_i] = 1$$

$$\operatorname{Var}[w_i] = \frac{1}{n_{\text{in}}}$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

$$\mathrm{Var}[w_i] = \frac{2}{n_{\mathrm{in}} + n_{\mathrm{out}}}$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- Mean выходов слоев должны быть 0 $\rightarrow \mathrm{E}\ w_L = 0$
- Variance выходов слоев должны быть одинаковыми $\rightarrow \mathrm{Var}\ w_L = \dfrac{2}{n_{in} + n_{out}}$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- Mean выходов слоев должны быть 0 $\rightarrow \mathrm{E}\, w_L = 0$
- Variance выходов слоев должны быть одинаковыми $\rightarrow \mathrm{Var}\, w_L = \dfrac{2}{n_{in} + n_{out}}$

Какое распределение подходит?

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- Mean выходов слоев должны быть 0 $\rightarrow \mathrm{E}\ w_L = 0$
- Variance выходов слоев должны быть одинаковыми $\rightarrow \mathrm{Var}\ w_L = \dfrac{2}{n_{in} + n_{out}}$

Какое распределение подходит? $w_i \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}, \frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}\right]$

Инициализация: He

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- **Mean выходов слоев может быть не 0 (например, с ReLU активациями)**
- Variance выходов слоев должны быть одинаковыми

Логика вывода похожая

Инициализация: He

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- **Mean выходов слоев может быть не 0 (например, с ReLU активациями)**
- Variance выходов слоев должны быть одинаковыми $\rightarrow \mathrm{Var}\ w_L = \dfrac{2}{n_{in}}$

Какое распределение подходит?

Инициализация: He

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- **Mean выходов слоев может быть не 0 (например, с ReLU активациями)**
- Variance выходов слоев должны быть одинаковыми $\to \text{Var } w_L = \dfrac{2}{n_{in}}$

Какое распределение подходит? $w_i \sim N(0, \sqrt{2/n_{\text{in}}^{(l)}})$

Пара полезных ссылок

https://towardsdatascience.com/

https://www.deeplearning.ai/ai-notes/initialization/index.html