# Deep Learning. Однослойные нейросети. Метрики качества

Урок 2

Егор Конягин

МФТИ & АО ”ЦОСиВТ”

# Содержание

# Логистическая регрессия. Повторение

# Логистическая регрессия. Повторение

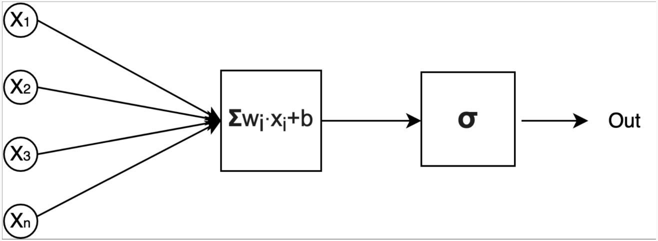


**Рис. 1:** Модель логистической регрессии

# Логистическая регрессия. Повторение - II

Реализация логистической регрессии состоит из трёх шагов: инициализации весов, шага forward prop и шага backprop. Это можно описать следующими уравнениями:

$$z^{(i)} = w^T x^{(i)} + b \tag{1}$$

$$\hat{y}^{(i)} = a^{(i)} = \sigma(z^{(i)}). \tag{2}$$

$$J(\hat{y}^{(i)}, y^{(i)}) = -\sum_i y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \tag{3}$$

$$\frac{\partial J}{\partial w} = \nabla J_w = \frac{1}{m} X (A - Y)^T \tag{4}$$

$$\frac{\partial J}{\partial b} = \frac{dJ}{db} = \frac{1}{m} \sum_{i=1}^{m} (a^{(i)} - y^{(i)}) \tag{5}$$

# Логистическая регрессия. Повторение - III

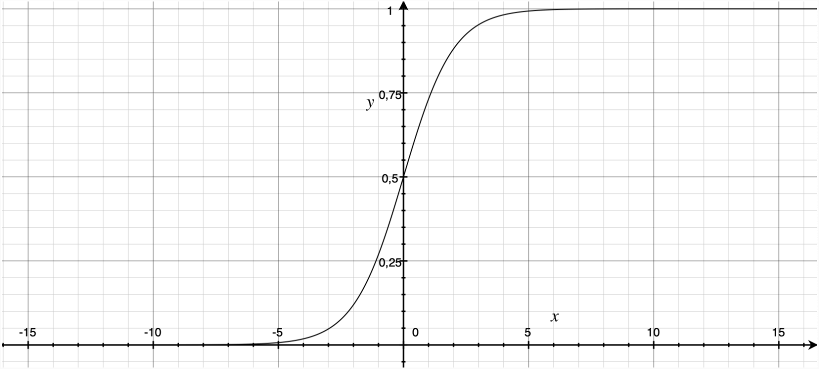


Рис. 2: График сигмоидной функции

# Метрики качества

# Метрики

## Зачем нужны метрики?

Метрики качества, в отличие от функции потерь, никак не используются при обучении. Они рассчитываются исключительно для наглядности и удобства. Однако именно они по большому счету отображают, насколько хорошо алгоритм справляется со сформулированной задачей.

Три классические метрики качества задачи классификации:

- accuracy;
- precision;
- recall.

# Accuracy, precision, recall

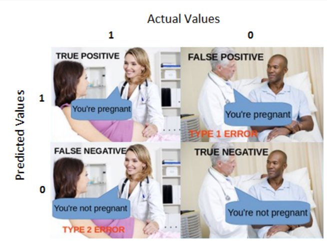


**Рис. 3:** Классификация ошибок

$$acc = \frac{TP + TN}{TP + TN + FP + FN}; \tag{6}$$

$$\mathcal{P} = \frac{TP}{TP + FP}; \tag{7}$$

$$\mathcal{R} = \frac{TP}{TP + FN}; \tag{8}$$

# Другие метрики

Выбор метрики обусловлен конечной решаемой задачей и целью. Часто неверные ответы предсказательной системы приносят компании убытки, а убытки могут зависеть от типа ошибки (False negative VS False positive).

- $\mathcal{F}_{Score} = \dfrac{2\mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}}$;
- $R_{Square} = 1 - \dfrac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y}_i)^2}$;
- $IoU\ (A, B) = \dfrac{\mu(A \cap B)}{\mu(A \cup B)}$;

# Однослойные нейронные сети. Постановка задачи

# Постановка задачи

Данный датасет имеет следующий вид: два признака ($x, y$—координаты) и цвет (то есть предсказываемая величина) (синий или красный). Наша нейросеть будет предсказывать цвет (1 - синий, 0 - красный).

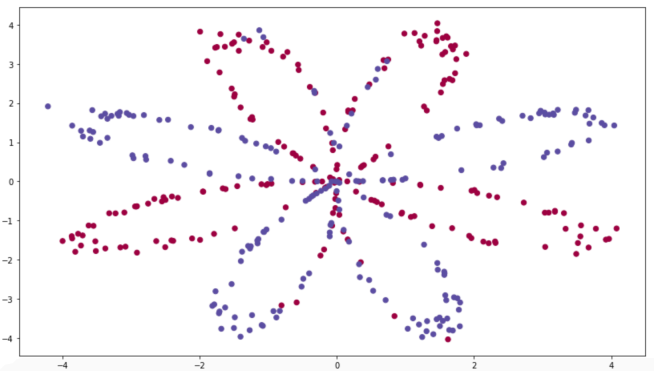


**Рис. 4:** Датасет точек

# Постановка задачи - II

Мы уже обсуждали, что логистическая регрессия справляется только с линейно разделимыми данными. В данном случае ее лучший результат имеет следующий вид:

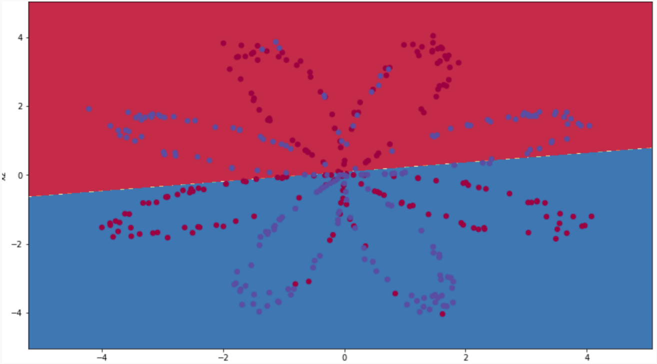


**Рис. 5:** Логистическая регрессия. Качество (accuracy): 47%

# Однослойные нейронные сети. Описание архитектуры

# Описание архитектуры

Однослойные нейронные сети иногда называют Shallow neural networks (в противоположность deep neural nets). Некоторые гиперпараметры (те величины, которые описывают архитектуру нейронной сети) нашей нейронной сети уже заранее известны: это кол-во входных признаков (их два) и кол-во выходных значений (оно одно). Нам остается спроектировать только кол-во нейронов в промежуточном слое.

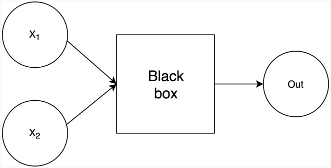


**Рис. 6:** Модель нейронной сети

# Описание архитектуры - II

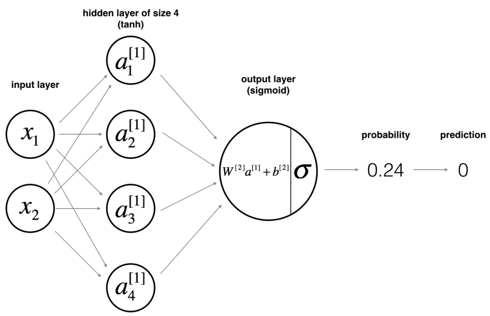


**Рис. 7:** Вариант архитектуры нейронной сети. Источник: Stanford University (Andrew Ng)

Здесь представлена нейронная сеть с одним скрытым слоем (а всего слоёв два). Именно об этих нейронных сетях пойдет сегодня разговор.

# Обозначения индексов

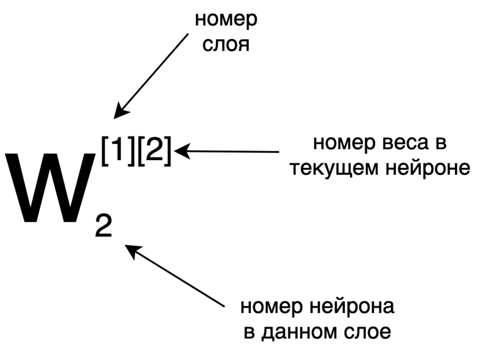


Рис. 8: Обозначения индексов

# Математическое описание однослойной нейросети

# Математическое описание однослойной нейросети

Обучение нейронной сети состоит из тех же пунктов, что и у логистической регрессии:

1. Инициализация весов;
2. Начало цикла
    2.1 Шаг forward propagation;
    2.2 Шаг backward propagation;
    2.3 Обновление весов;
3. Завершение обучения.

Разница состоит в том, что forward propagation и back propagation описываются другими уравнениями (их вид определяется исключительно архитектурой модели).

# Шаг forward propagation

Уравнения имеют следующий вид:

$$z^{[1]} = W^{[1]} \cdot x + b^{[1]} \tag{9}$$

$$a^{[1]} = \sigma^{[1]}(z^{[1]}) \tag{10}$$

$$z^{[2]} = W^{[2]} \cdot a^{[1]} + b^{[2]} \tag{11}$$

$$a^{[2]} = \sigma^{[2]}(z^{[2]}) \tag{12}$$

$$J = -\frac{1}{m} \sum_{i=1}^{m} \left( y^{(i)} \log\left( a^{[2](i)} \right) + (1 - y^{(i)}) \log\left( 1 - a^{[2](i)} \right) \right) \tag{13}$$

$\sigma^{[1]}, \sigma^{[2]}$ - это функции активации соответствующих слоев. В логистической регрессии в качестве таких функций использовалась сигмоида. В данной модели в качестве $\sigma^{[1]}$ воспользуемся гиперболическим тангенсом, в качестве $\sigma^{[2]}$ - сигмоиду.

# Шаг back prop

Back prop описывается следующими уравнениями:

$$dz^{[2]} = a^{[2]} - y; \quad dw^{[2]} = \frac{\partial J}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial w^{[2]}} = \frac{1}{m} \cdot dz^{[2]} \cdot a^{[1]T} \tag{14}$$

$$db_j^{[2]} = \frac{1}{m} \sum_{i=1}^{m} dz_j^{[2](i)}, j = \{1\} \tag{15}$$

$$dz^{[1]} = \frac{\partial J}{\partial a^{[1]}} * \frac{\partial a^{[1]}}{\partial z^{[1]}} = w^{[2]T} \cdot dz^{[2]} * \frac{\partial a^{[1]}}{\partial z^{[1]}} \tag{16}$$

$$dw^{[1]} = \frac{\partial J}{\partial a^{[1]}} \cdot \frac{\partial a^{[1]}}{\partial w^{[1]}} = \frac{1}{m} \cdot dz^{[1]} \cdot x^T \tag{17}$$

$$db_j^{[1]} = \frac{1}{m} \sum_{i=1}^{m} dz_j^{[1](i)}, j = \{1, 2, 3, 4\} \tag{18}$$

# Обновление параметров. Окончание обучения

Обновления параметров происходят тем же образом, что и в случае с логистической регрессии. Важно помнить, что после того как рассчитаны градиенты на текущем шаге, происходит обновление параметров нейросети во всех слоях! Только после этого происходит очередной шаг forward prop. Обучение данной модели прекратится спустя заданное количество выполненных итераций.

# Заключение: функции активации

# Функции активации

Мы уже рассмотрели такие функции активации, как сигмоида и гиперболический тангенс. Ниже представлены другие популярные функции активации:

- $\tanh(x) = \frac{e^x - 1}{e^x + 1}$;
- $ReLU(x) = \max(0, x)$;
- $Softplus(x) = \log(1 + \exp(x))$;

Важно, что все эти функции являются нелинейными! Функция $ReLU(x)$ не является непрерывно дифференцируемой функцией, однако для нее также выполняется универсальная теорема об аппроксимации. Более того, на сегодняшний день это самая распространённая функция активации нейронных сетей.

# Summary

Сегодня мы рассмотрели следующие вопросы и задачи:

- важнейшие метрики: accuracy, precision, recall и др.
- нейронные сети с одним скрытым слоем: forward и back prop;
- поговорили о функциях активации.