

# Нейронные сети

Ксемидов Борис Сергеевич

Chillers

30 апреля, 2020

# Математическая модель нейрона

Нейрон можно представить в следующем виде:

$$y = \sigma(wx - b)$$

$$w = (w_1, w_2, \dots, w_n), x = (x_1, x_2, \dots, x_n)$$

# Функции активации

Существуют множество функций активации, вот некоторые из них:

- $\sigma(x) = \frac{1}{1+e^{-x}}$  - сигмоида;
- $\sigma(x) = \frac{e^{z_i}}{\sum_k e^{z_k}}$  - softmax;
- $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$  - гиперболический тангенс;
- $\sigma(x) = x$  - линейная.

# Архитектуры нейронных сетей

Архитектур нейронных сетей существует большое множество, наиболее популярными и известными из них являются:

- прямого распространения (персептрон, многослойный персептрон);
- сверточные;
- рекуррентные.

Для оценки ошибки предсказания нейронной сети на данных существуют функции ошибки:

- $Q(x, w, y) = \frac{1}{n}(a(w, x) - y)^2$  - MSE (mean square error);
- $Q(x, w, y) = \frac{1}{n}\sqrt{(a(w, x) - y)^2}$  - RMSE (root mean square error).

# Нейронная сеть прямого распространения

## Обозначения

Синие нейроны - входные, красные - скрытые, зеленые - выходные.

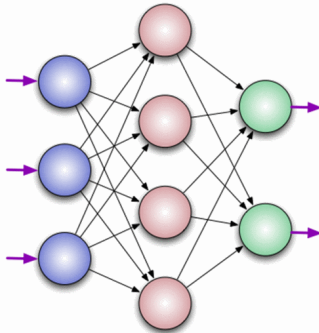


Рисунок: Пример многослойного персептрона

# Обучение многослойного персептрона

Обучение многослойного персептрона производится с помощью алгоритма обратного распространения ошибки.

Пусть выбрана MSE:  $Q(x, w, y) = \frac{1}{n}(a(w, x) - y)^2$ .

Тогда для алгоритма обратного распространения ошибки необходимо продифференцировать её по весам:

$$\nabla w = -\mu \frac{\partial Q}{\partial w}, \mu \in R$$

Для распространения ошибок по скрытым слоям:

$$\nabla \delta = \frac{\partial Q}{\partial x}$$

# Алгоритм обратного распространения ошибки

Алгоритм обратного распространения ошибки:

- 1 правка весов;
- 2 посчитать величину ошибки для следующего слоя;
- 3 распространить ошибку на следующий слой.



# Обучение нейронной сети

## Эпоха

Весь датасет прошел через нейронную сеть в прямом и обратном направлении только один раз.

## Батч

Часть датасета выбранного размера.

## Итерация

Прохождение батча в прямом и обратном направлении только один раз.

# Проблемы обучения сетей

Существуют следующие проблемы обучения нейронных сетей:

- паралич сети;
- локальные минимумы.

# Паралич сети

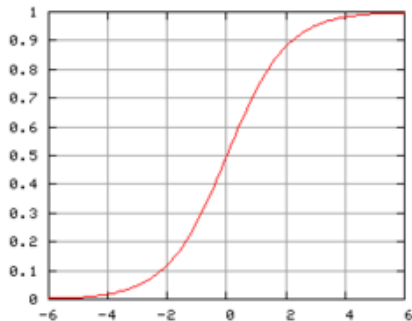


Рисунок: Сигмоида

# Проблема локального минимума

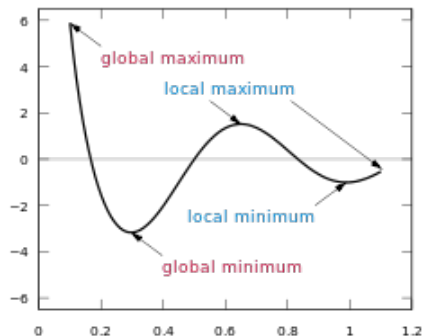


Рисунок: Метод градиентного спуска

# Проблемы многослойного персептрона

Существуют следующие проблемы многослойного персептрона:

- затухающие градиенты;
- взрывные градиенты.

# Свертка

Основа сверточных сетей - фильтры, производящие операции свертки.

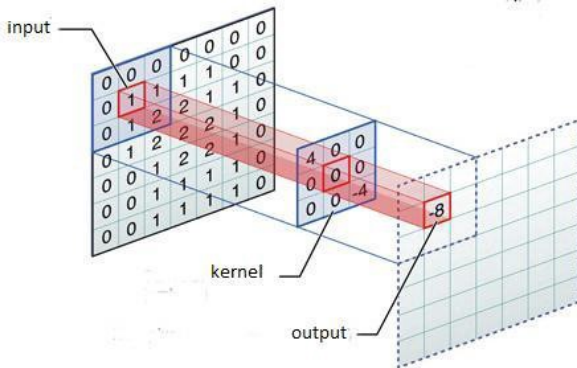


Рисунок: Свертка изображения

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

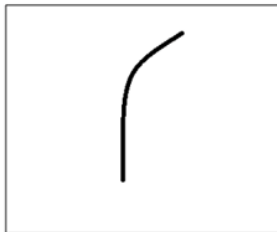


Рисунок: Пример обученного ядра

# MaxPooling

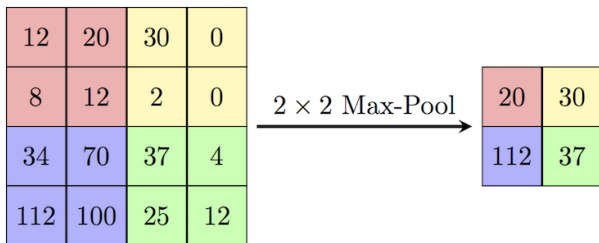


Рисунок: Пример подвыборки



# Сверточные нейронные сети

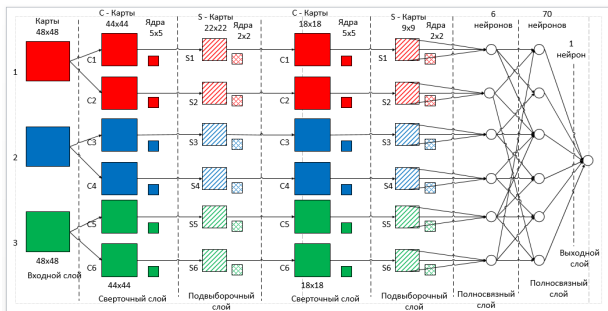


Рисунок: CNN

Генеративно-сопоставительные сети предназначены для создания сущностей, похожих на объекты из реального мира.

GAN состоит из:

- дискриминатора;
- генератора.

# Дискриминатор

Дискриминатор предназначен для классификации поддельных изображений, генерируемых генератором, и настоящих. Представляется в виде сверточной нейронной сети.

# Генератор

Генератор моделирует распределение отдельных классов.  
Представляется в виде "обратной" сверточной нейронной сети.

# Архитектура GAN

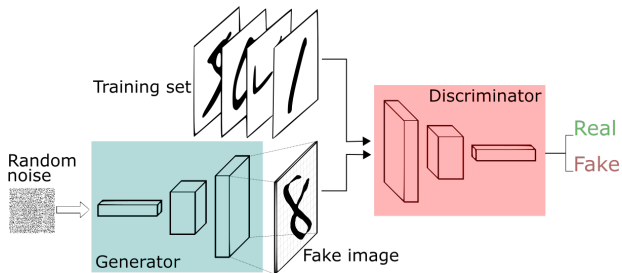


Рисунок: GAN

# Автоэнкодер

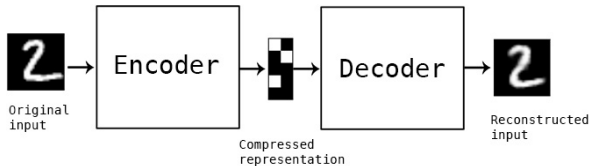


Рисунок: Автоэнкодер

# Виды нейронных сетей

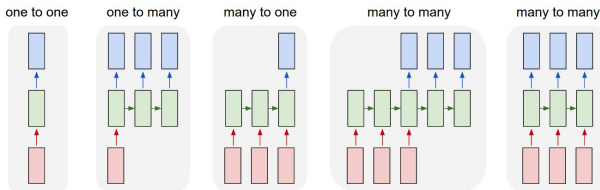


Рисунок: Типы нейронных сетей

# Реккурентные нейронные сети

RNN предназначены для обработки последовательностей.

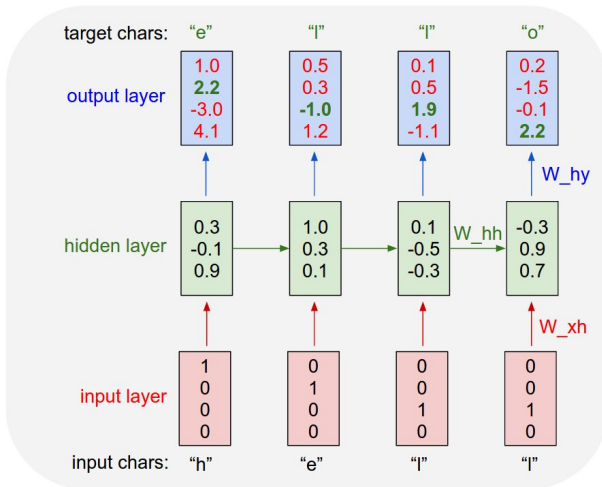


Рисунок: Пример RNN



Отличие RNN от LSTM заключается в фильтрах.

Существуют три фильтра:

- фильтр забывания (что забыть);
- слой входного фильтра (что добавить);
- слой выходного фильтра (выделение требуемой информации).

# LSTM

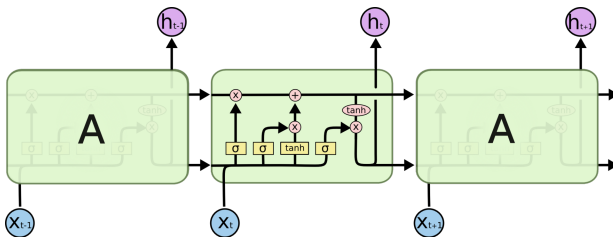


Рисунок: Архитектура LSTM

Спасибо за внимание!