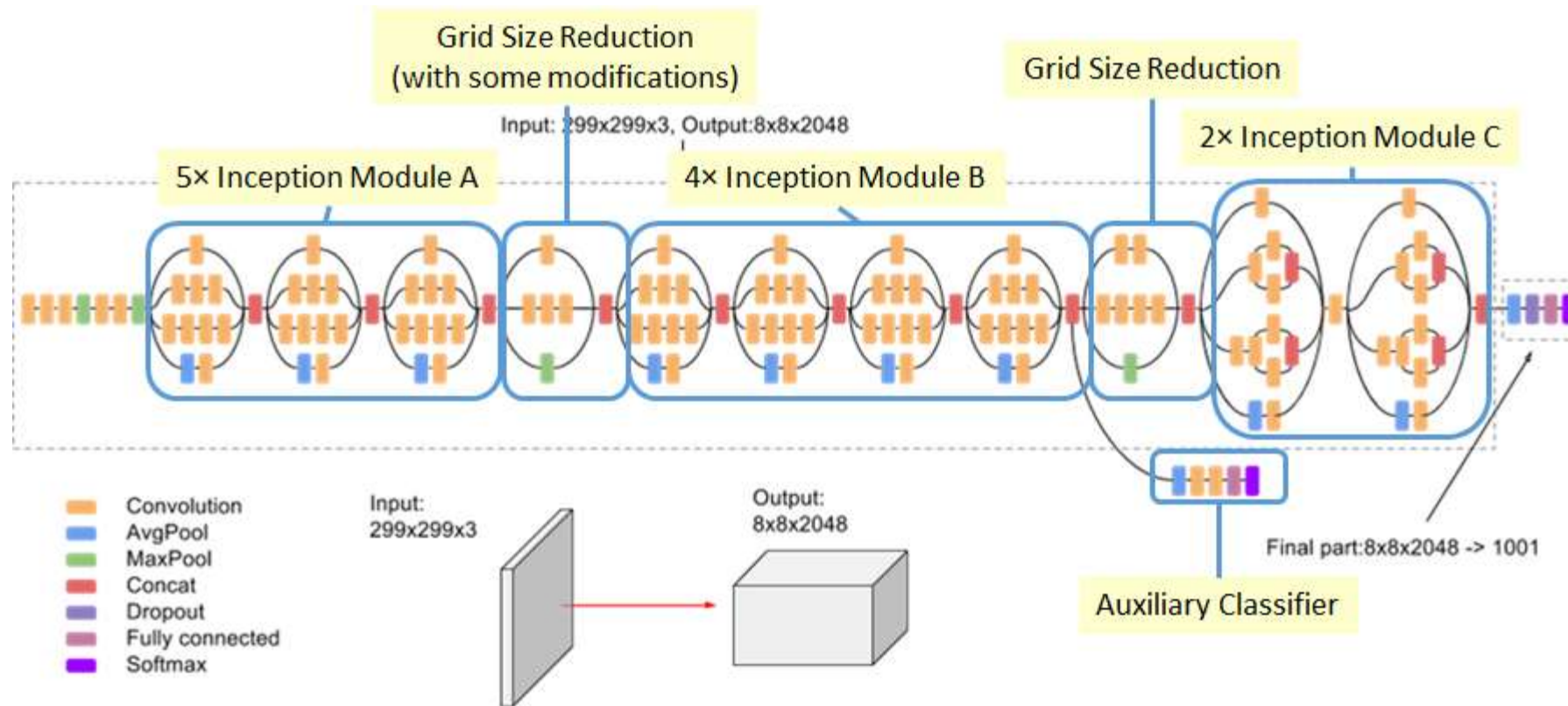


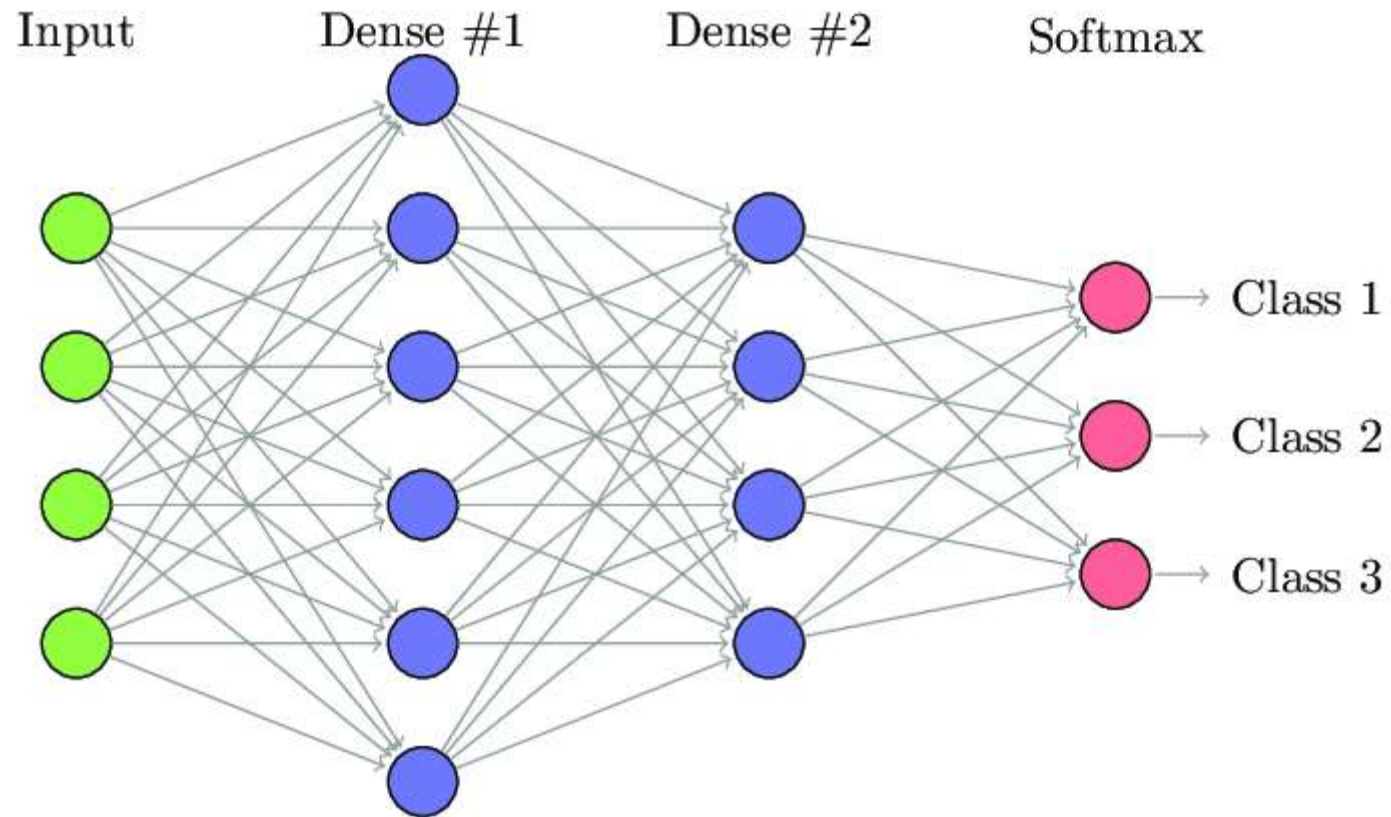
Введение в Нейронные сети

Рахматуллин Рамазан, 182

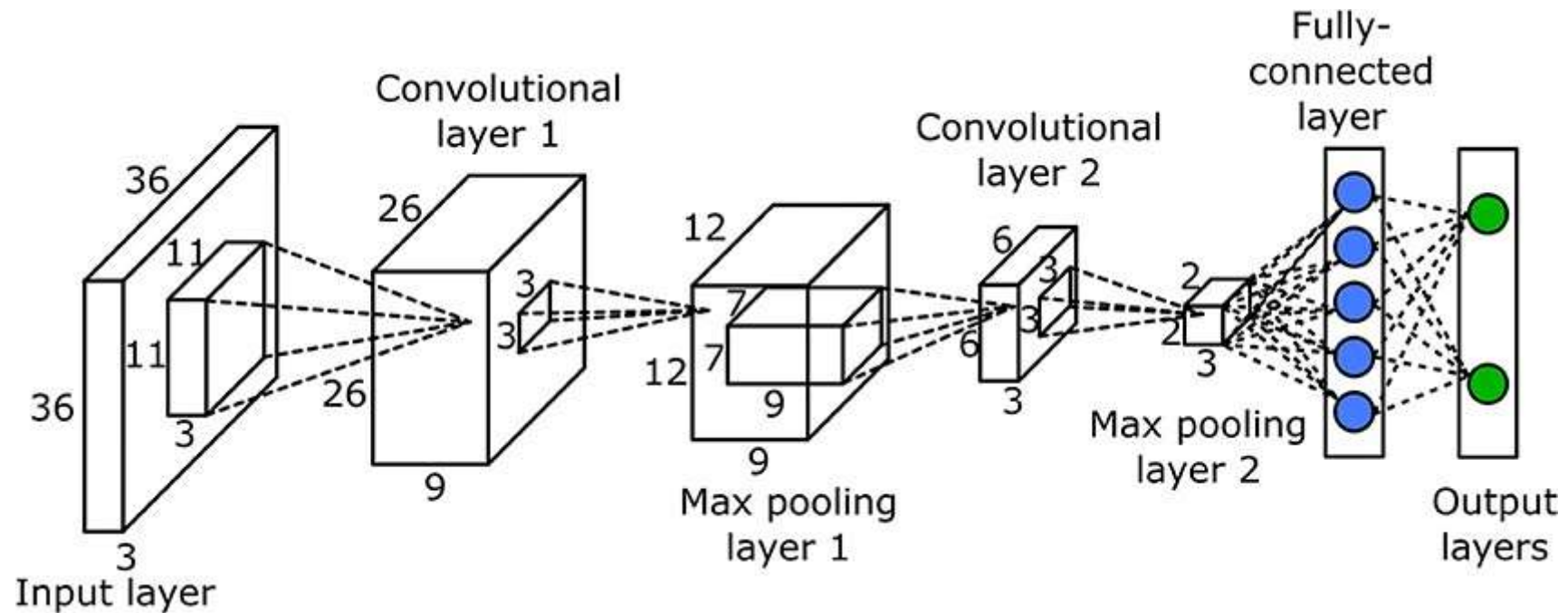
Виды и применения нейронных сетей



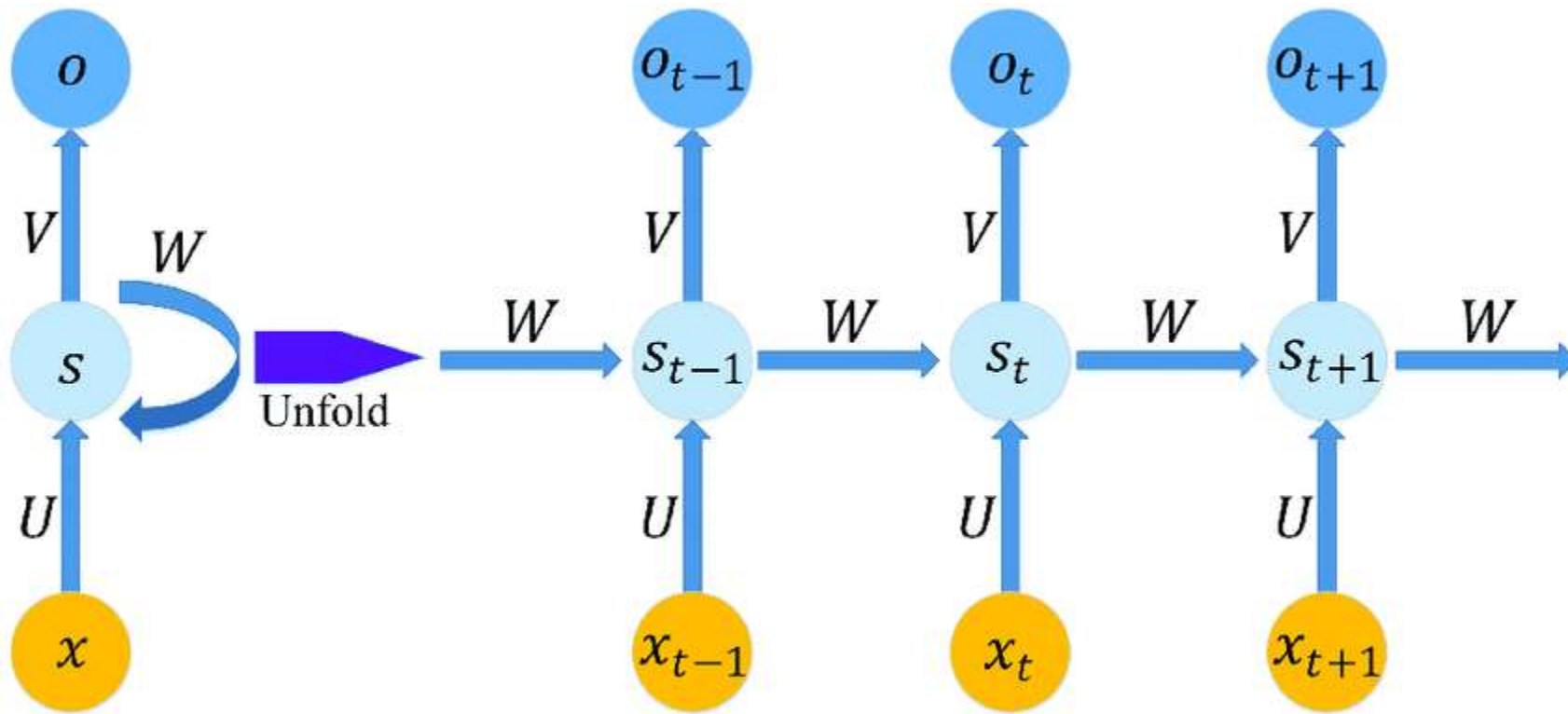
Dense layer



Convolutional layer



Recurrent neural network



Применения нейронных сетей

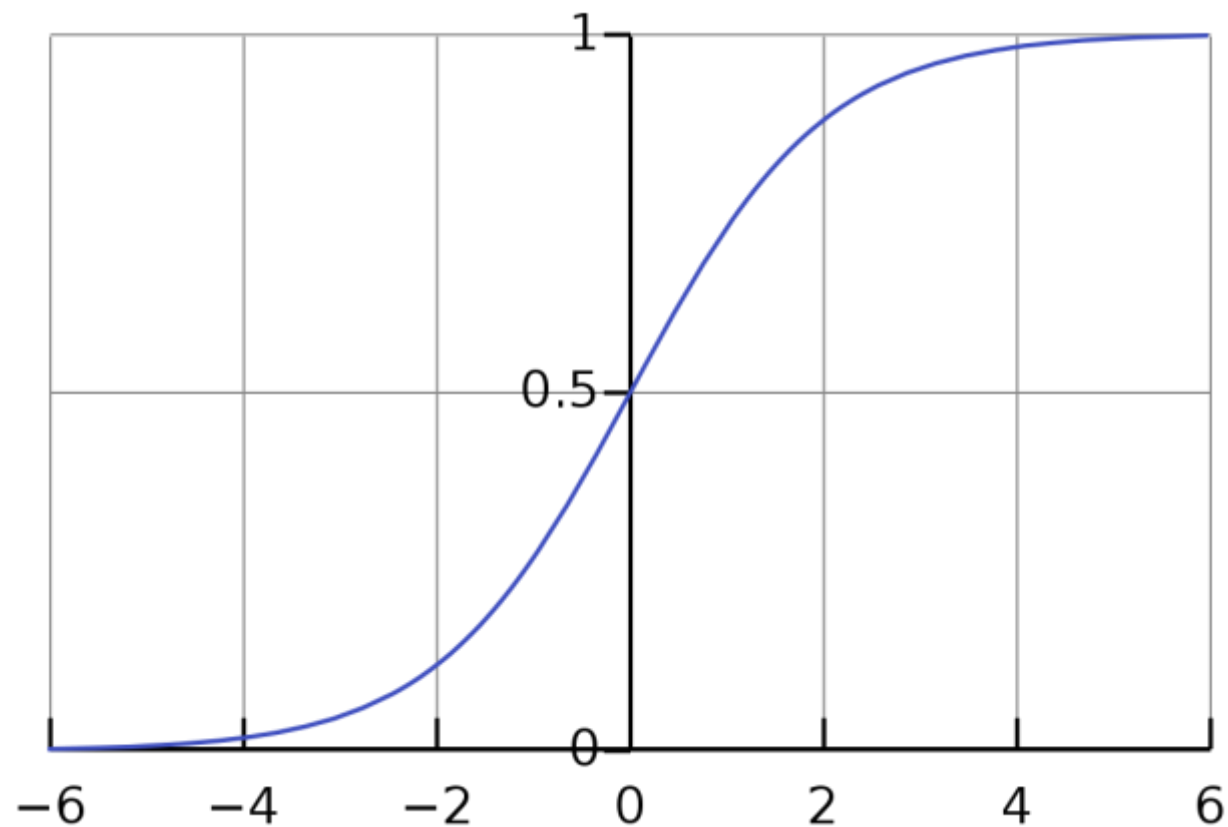
- Обработка естественного языка
- Классификация аудио, изображений
- Обучение с подкреплением
- Распознавание образов
- Кластеризация
- Задача коммивояжера
- Все остальные задачи машинного обучения

Функции активации

- Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$
- Tanh: $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- ReLU: $f(x) = \max(0, x)$
- Leaky ReLU: $f(x) = \begin{cases} 0.01x, & x < 0 \\ x, & x \geq 0 \end{cases}$
- Softmax: $f_i(\vec{x}) = \frac{e^{x_i}}{\sum_j e^{x_j}}$
- Функции свертки

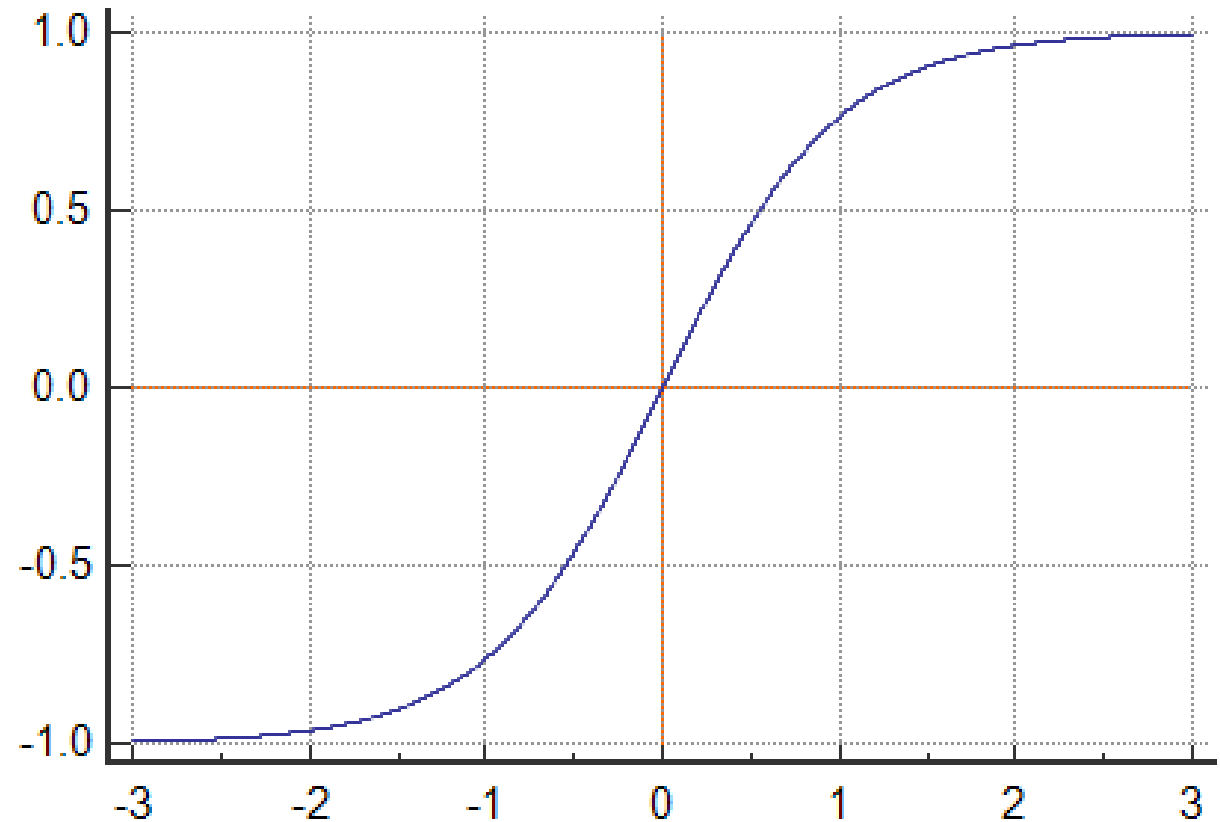
Sigmoid

- $\sigma(x) = \frac{1}{1+e^{-x}}$
- Диапазон (0, 1)
- Долго считать
- Несимметрична относительно 0
- Имеет маленький градиент для больших по модулю значений



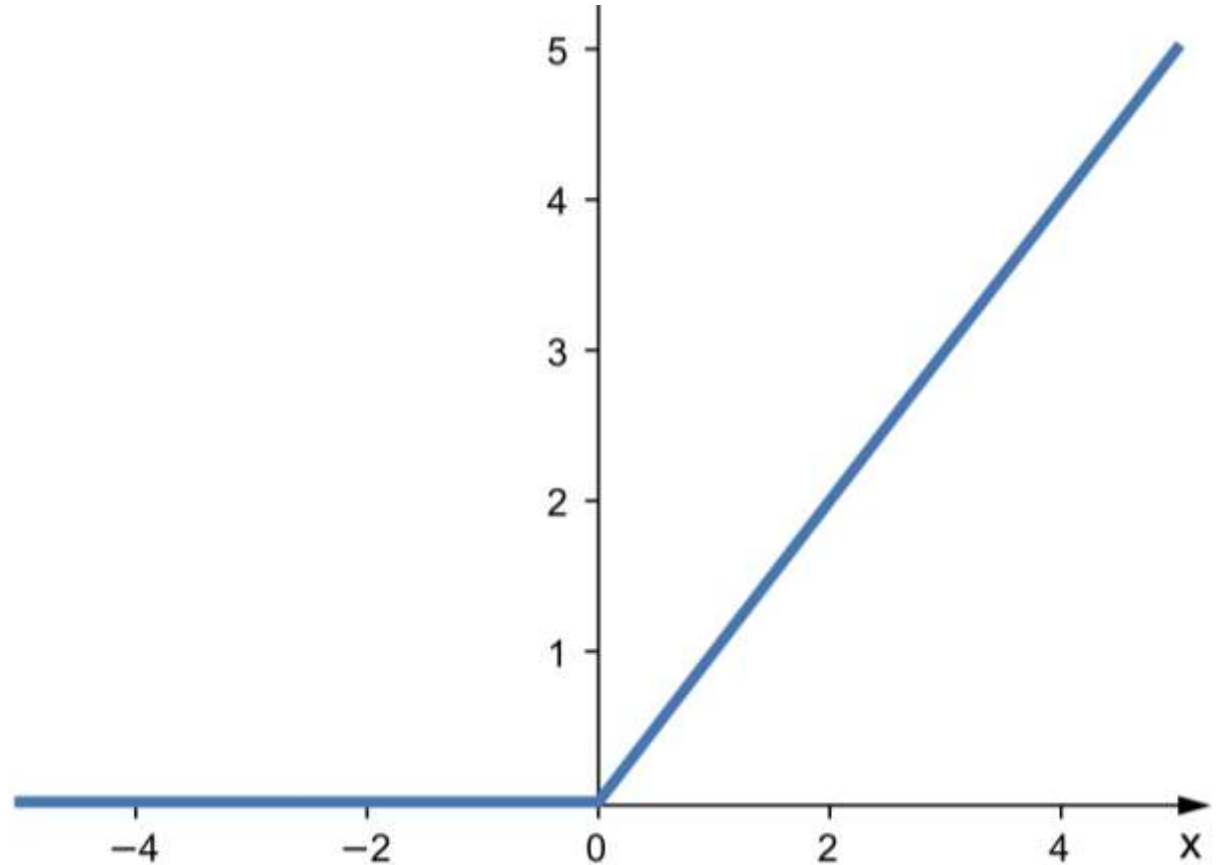
Tanh

- $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- Диапазон $(-1, 1)$
- Ведет себя как sigmoid с некоторыми различиями
- Симметрична относительно 0
- Большой градиент



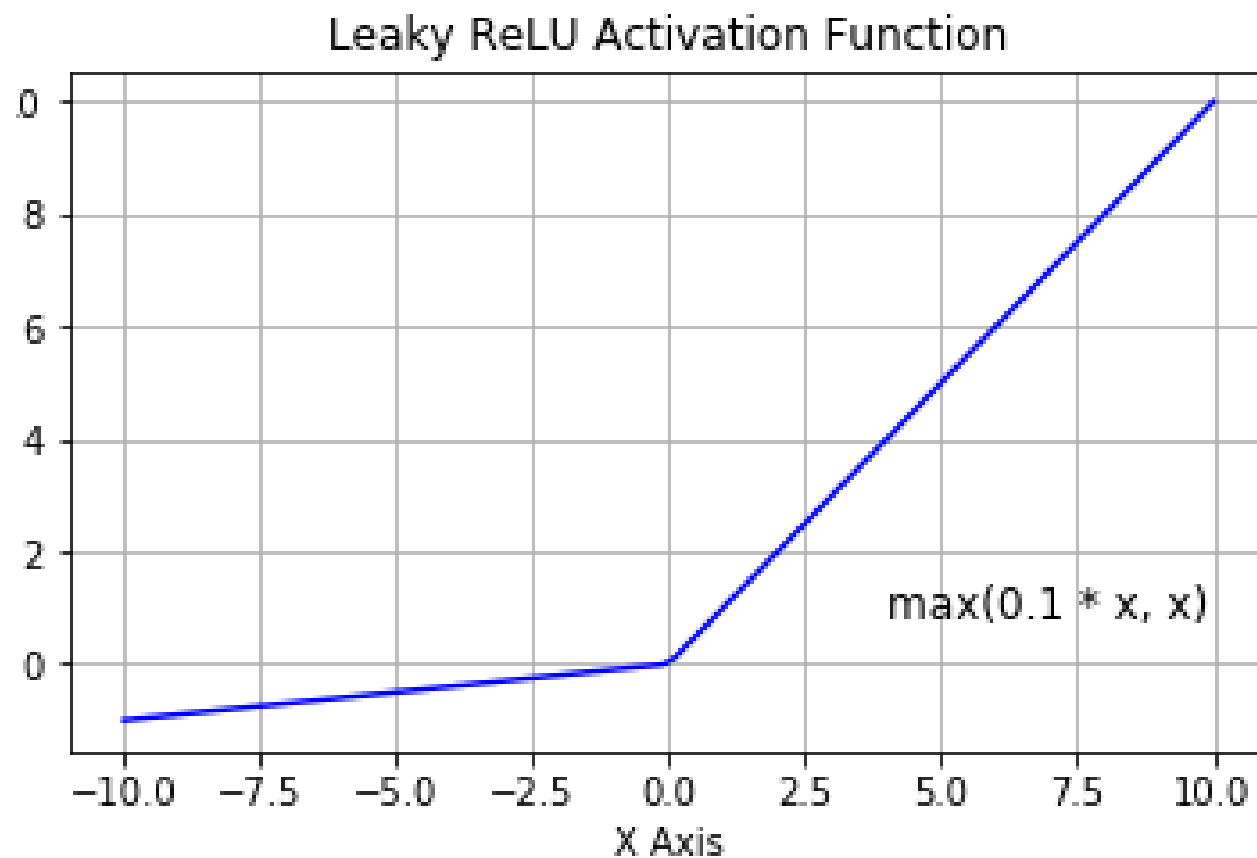
ReLU

- $f(x) = \max(0, x)$
- Диапазон $[0, \infty)$
- Быстро считать
- Зануляется в половине случаев
- Нейрон может “умереть”



Leaky ReLU

- $f(x) = \begin{cases} 0.01x, & x < 0 \\ x, & x \geq 0 \end{cases}$
- Диапазон $(-\infty, \infty)$
- Решает проблему “мертвых” нейронов в ReLU



Softmax

- $f_i(\vec{x}) = \frac{e^{x_i}}{\sum_j e^{x_j}}$
- Обобщение Sigmoid для вектора

Функции свертки

- max
- mean
- sum

Способы обучения

- Backpropagation
- Глобальные методы оптимизации
- Аналитическое решение

Проблема переобучения

- Из-за большого размера, нейронные сети склонны к переобучению

Решения

- Увеличение числа тестовых примеров для изображений, аудио, текста
- Добавление шума к данным
- Использование существующих сетей
- Регуляризация
- Ограничение весов
- Weight decay
- Исключение нейронов

Метод исключения нейронов (Dropout)

- Для dropout слоя задается параметр p
- На этапе тренировки каждый нейрон из слоя не участвует в предсказании с вероятностью p
- На этапе предсказания все нейроны из слоя участвуют в предсказании с коэффициентом $(1 - p)$

Inverse dropout

- На этапе тренировки нейроны также исключаются, но функция активации умножается на $1 / (1 - p)$
- На этапе предсказания используется полная модель

Проблема исчезновения градиентов

- Возникает при backpropagation
- Произведение маленьких производных приводит к занулению производной и остановке тренировки (vanishing gradients)
- Произведение больших производных приводит к слишком сильным изменениям весов (exploding gradients)

Решения

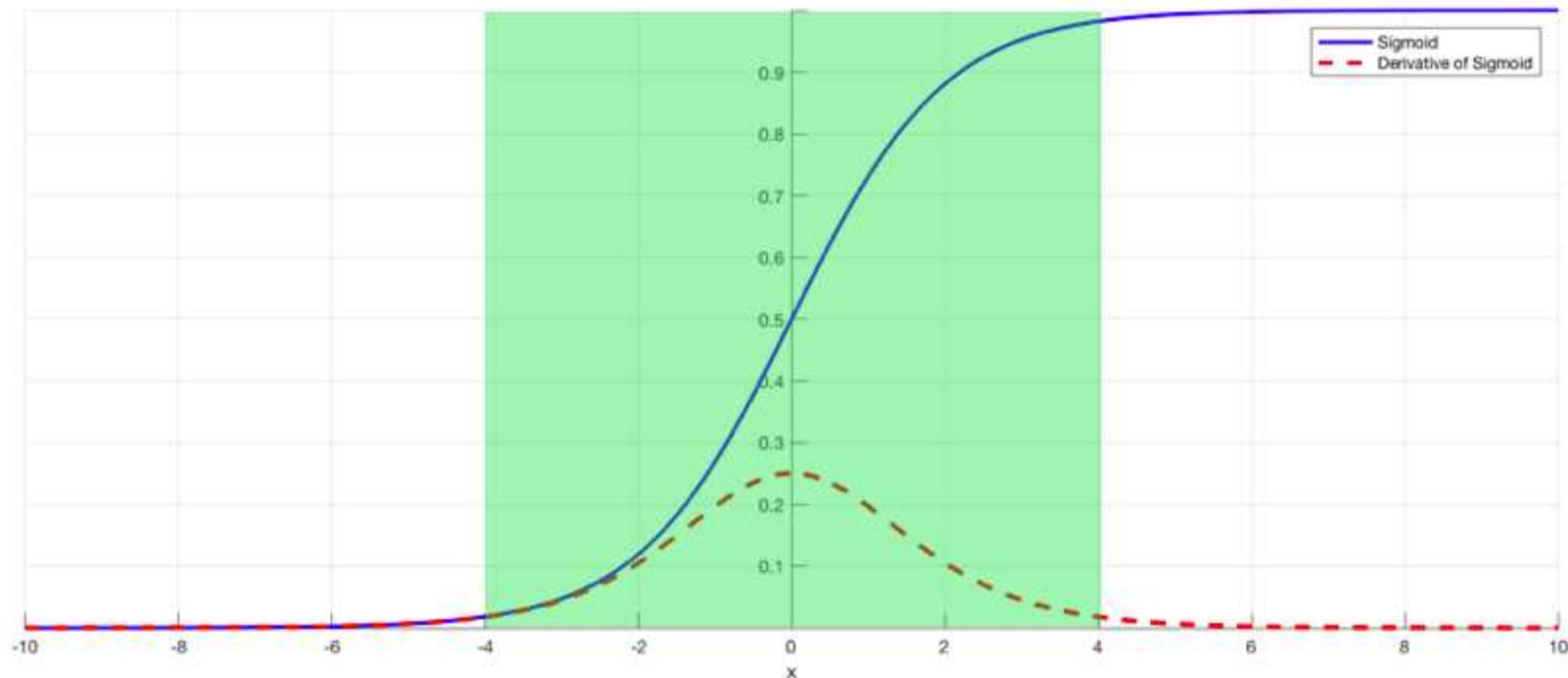
- Добавить ребра через слои как в остаточной сети
- LSTM
- Изменить функцию активации
- Альтернативные методы инициализации
- Тренировка по слоям
- Пакетная нормализация

Пакетная нормализация

- Применяется для mini batch gradient descent
- Добавим слой пакетной нормализации, который принимает на вход результат предыдущего слоя и нормализует значения по каждому столбцу. Таким образом $\hat{x}^{(k)} = \frac{x^{(k)} - E(x^{(k)})}{\sigma^{(k)}}$
- Вектор $\hat{x}^{(k)}$ имеет среднее 0 и дисперсию 1
- Далее денормализуем значения с помощью тренируемых параметров: $y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)}$
- На этапе предсказания используются среднее и стандартное отклонение тестовой выборки

Пакетная нормализация

- Ускоряет сходимость
- Добавляет шум, таким образом регуляризируя
- Решает проблему исчезающих градиентов



ИСТОЧНИКИ

- Inception v3: <https://sh-tsang.medium.com/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c>
- Dense network: https://www.researchgate.net/figure/Example-of-fully-connected-neural-network_fig2_331525817
- Convolutional network: <https://mc.ai/layers-of-a-convolutional-neural-network/>
- Neural network applications: <https://www.xenonstack.com/blog/artificial-neural-network-applications/>
- Activation functions: <https://towardsdatascience.com/activation-functions-b63185778794>
- Dropout: <https://habr.com/ru/company/wunderfund/blog/330814/>
- Vanishing gradient problem:
https://en.wikipedia.org/wiki/Vanishing_gradient_problem
<https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484>
- Batch normalization: https://en.wikipedia.org/wiki/Batch_normalization