Уральский федеральный университет

Методы машинного обучения

И все-таки про нейронные сети.
Часть 1. Имплементация на Numpy.

План

- Краткая история (очень кратко)
- Однослойный перцептрон
- Многослойный перцептрон
- Двухслойная полносвязная нейронная сеть

Нейронные сети (краткая история)

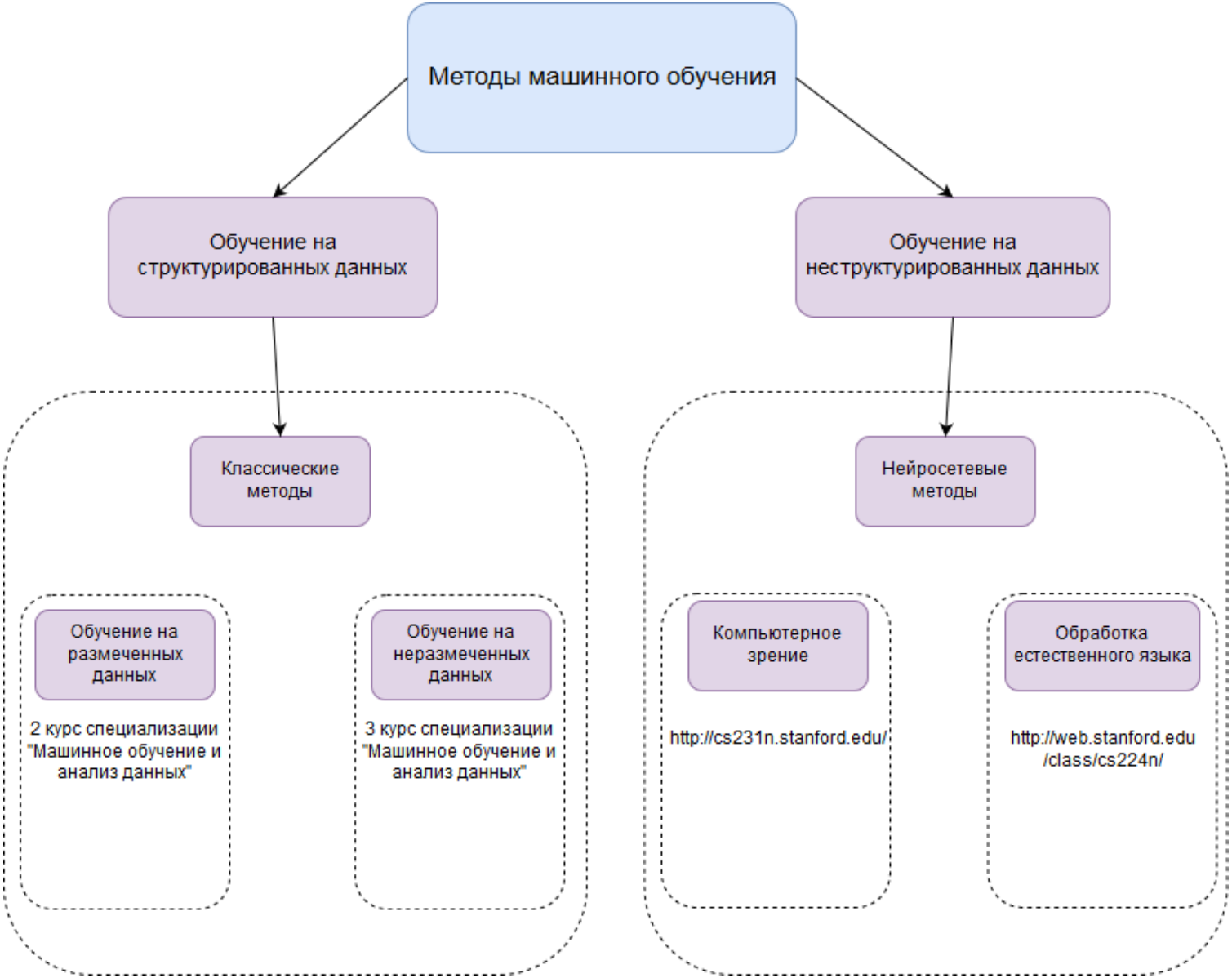
- 60-е годы – высокий «оптимизм»
- 70-е годы – 90-е годы – период спада в нейронных сетях. (классические методы работают куда лучше)
- 1986-й год – Джеффри Хинтон изобретает «обратное распространение ошибки»
- 1990-й год – Ян Лекун изобретает сверточные нейронные сети, но из-за «обманутых ожиданий» сообщество не уделяют этому открытию внимания.
- До 2012 года нейросетевые методы «тихо» изучались Джеффри Хинтоном, Яном Лекуном и прочими

Краткий перечень названий

- Neural Networks
- Deep Learning
- Deep Machine Learning
- Multilayer Perceptron
- Глубокое (глубинное) обучение
- Нейронные сети
- Искусственный интеллект (это не верно)

- Все это – лишь маркетинговые названия

Местоположение нейронных сетей



Самое существенное отличие

- Самое существенное отличие не столько в том, что нейросетевые методы работают над неструктурированными данными
- Самое существенное отличие – ручной подбор признаков исследователей как бы «перенесен» на ручной подбор архитектуры нейросети (оптимизация, функция потерь, количество слоев, количество нейронов в каждом слое)

Почему?

- Функция потерь – ключевая характеристика системы. Именно ее значение влияет на ошибку, которая распространится по всей сети во время обратного прохода

Функция потерь (Loss function)

- Cost function
- Objective function
- Error function
- Функция потерь
- Функция стоимости
- Целевая функция
- Функция(функционал ошибки)

- В разных парадигмах эти понятия совпадают

Не дайте себя запутать

- Функция потерь (loss function)

$$l(f(x_i|\theta), y_i) = (f(x_i|\theta) - y_i)^2$$

- Функция стоимости (cost function)

$$MSE(\theta) = \frac{1}{N}\sum_{i=1}^{N}(f(x_i|\theta) - y_i)^2$$

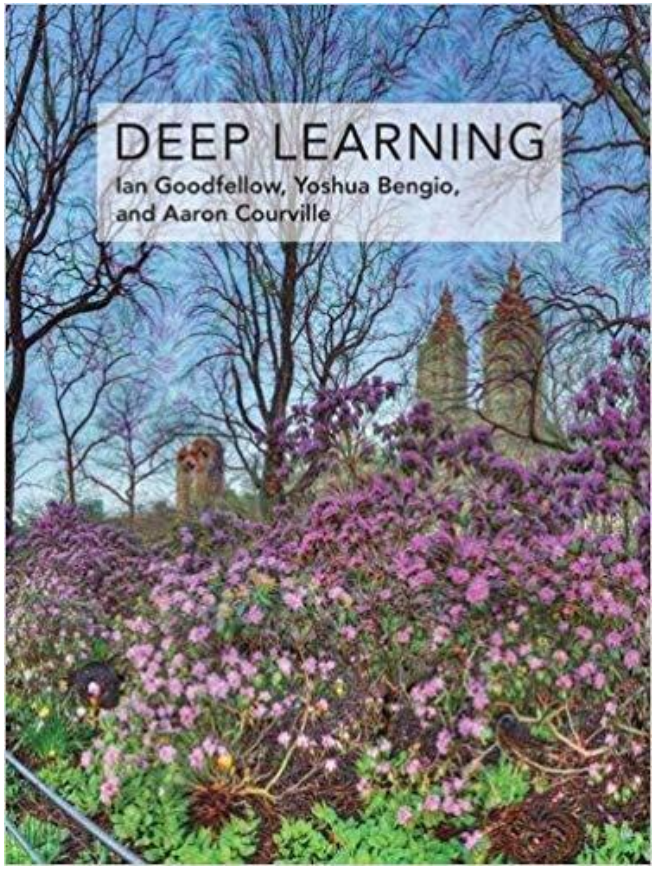
- Целевая функция (objective function)

То, что мы оптимизируем

- Таким образом, это в принципе разные понятия

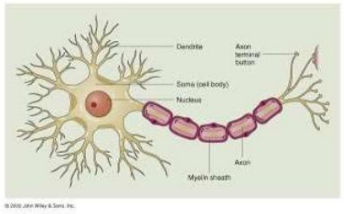
Определение из Deep Learning Book

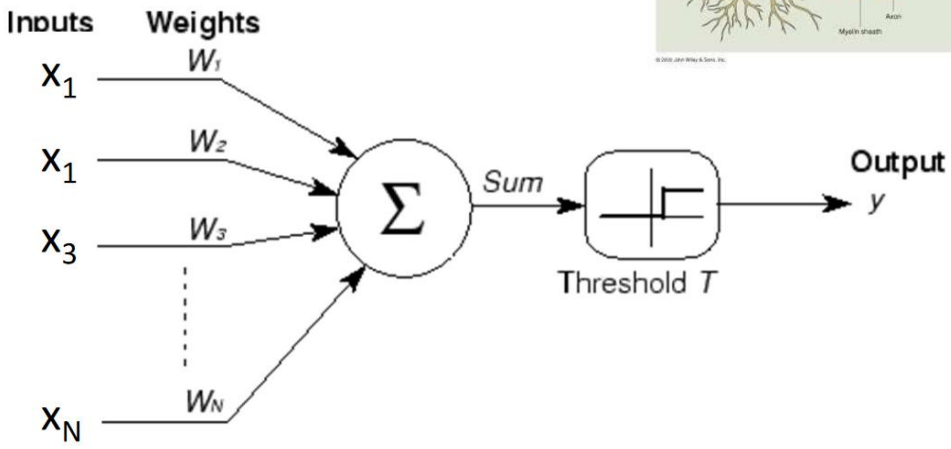
- The function we want to minimize or maximize is called the objective function, or criterion. When we are minimizing it, we may also call it the cost function, loss function, or error function. In this book, **we use these terms interchangeably**, though some machine learning publications assign special meaning to some of these terms. © deeplearningbook.org



Однойслойный перцептрон

- Розенблатт, 1957 год



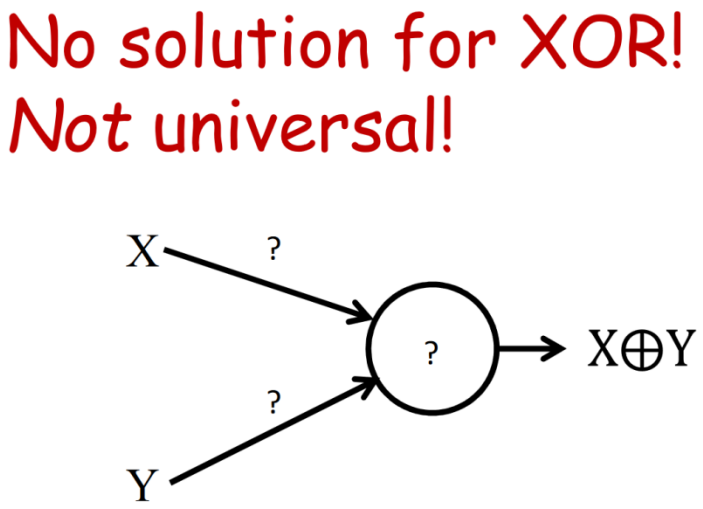


$$y = \begin{cases} 1 \; if \sum_i w_i x_i \geq T \\ 0 \; else \end{cases}$$

http://deeplearning.cs.cmu.edu/slides.spring19/lec2.universal.pdf

Проблема

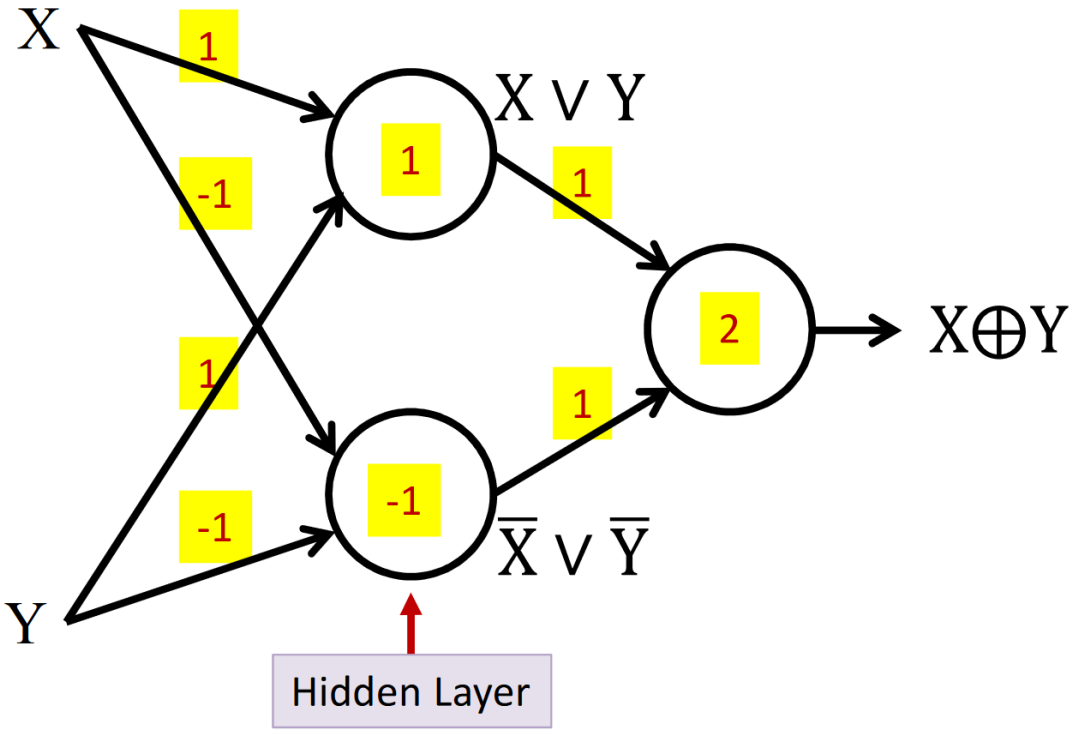
- Мински, 1968 год
- Невозможно смоделировать «исключающее или»



http://deeplearning.cs.cmu.edu/slides.spring19/lec2.universal.pdf

Многослойный перцептрон

- Может моделировать «исключающее или»



http://deeplearning.cs.cmu.edu/slides.spring19/lec2.universal.pdf

Теорема Цыбенко

- Универсальная теорема аппроксимации утверждает, что нейронная сеть с одним скрытым слоем может приближать любую функцию из класса непрерывных функций.
- Однако отсюда не следует, что нейронная сеть (многослойный перцептрон) – является самым оптимальным методом в смысле качества обобщения, количества нейронов и т.д.

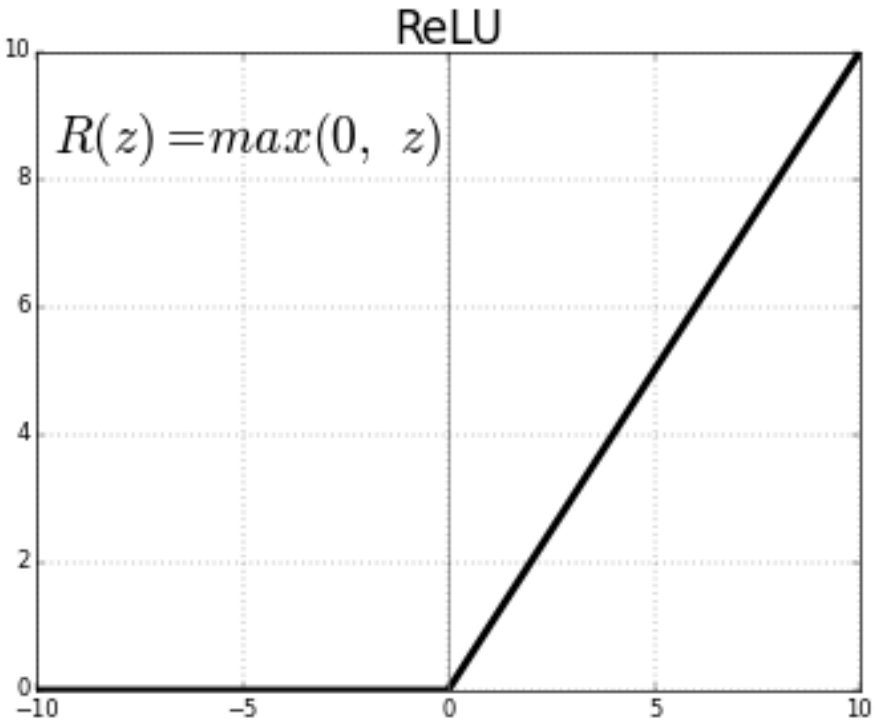
Добавление слоев

- Добавление «скрытых слоев» в структуру нейронной сети позволяет экспоненциально уменьшить число нейронов
- Именно поэтому «второе возрождение» нейронных сетей началось с того, что люди научились обучать слои с количеством слоев больше 5
- Вообще термин «deep neural network» достаточно противоречивый, например, в Carnegie Mellon говорят что сеть глубокая, когда у нее больше чем 2 скрытых слоя, а в некоторых источниках, когда больше чем 5 скрытых слоев

Двухслойная нейронная сеть

- На вход получает данные размерности (N, D), где N – количество объектов, D – размерность признакового описания
- Умножает вход на матрицу W1 (D, H), применяет функцию активации ReLU, умножает на матрицу W2 (H, C), применяет функцию активации softmax, где H – размерность скрытого слоя, C – количество классов

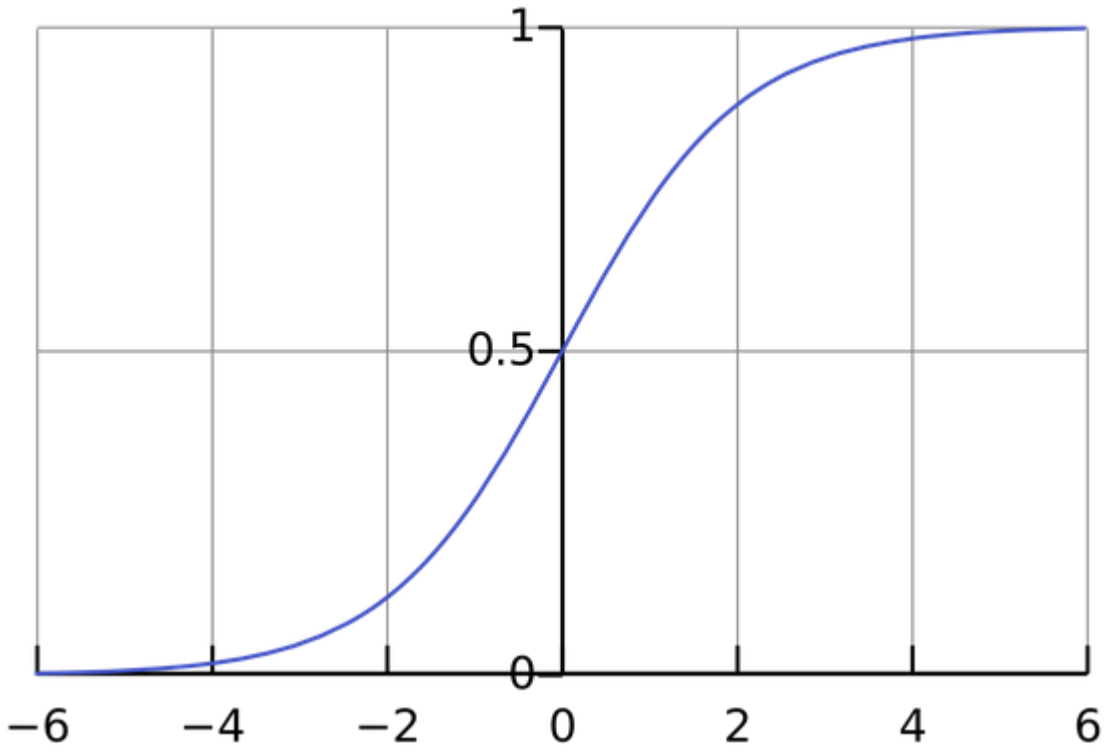
ReLU



https://www.quora.com/Why-do-neural-networks-need-an-activation-function

Sigmoid

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)},$$



https://www.quora.com/Why-do-neural-networks-need-an-activation-function

Реализация нейронной сети

- Итак.
- Z1 = X*W1 + b
- A1 = ReLU(Z1)
- Z2 = A1*W2 +b
- A2 = Softmax(Z2)
- Loss = CrossEntropyLoss(Z2)
- Gradient – обратное распространение ошибки

Обратное распространение ошибки

- Обратное распространение ошибки просто берет значение функции потерь (ошибка между вашим предсказанием и правильным ответом) и проводит его по нейронной сети через производные каждого слоя.
- Для каждой матрицы W, будет свой dW, который мы вычитаем на шаге градиентного спуска
- Самая лучшая лекция об этом: https://www.youtube.com/watch?v=i94OvYb6noo (c) Stanford cs231n, Andrej Karpathy

Реализация нейронной сети

- Напишем двухслойную нейронную сеть.
- Имплементация в тетрадке “two_layer_net.ipynb”

Домашнее задание

- Посмотреть https://www.youtube.com/watch?v=i94OvYb6noo
- Выполнить тетрадку “FullyConnectedNets.ipynb” из домашнего задания №2 cs231n: http://cs231n.github.io/assignments2018/assignment2/