



Машинное обучение

Лекция 13. Введение в глубокое обучение, полносвязные нейронные сети

Автор: Рустам Азимов

Санкт-Петербургский государственный университет

Санкт-Петербург

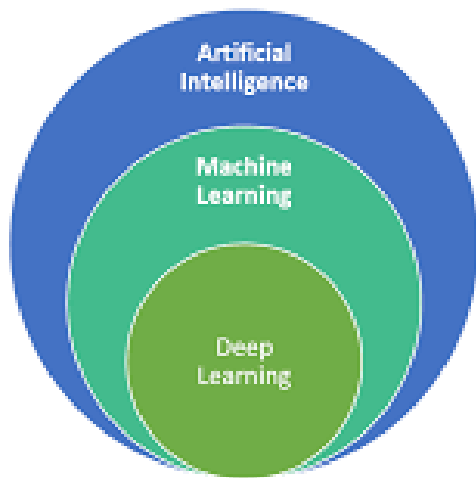


Figure 1: artificial intelligence, machine learning and deep learning Source: Nadia BERRHANE (M2 IESCI, 2018)

- Ключевым понятием являются нейронные сети
- Впервые упоминались в прошлом веке
- Необходимы большие вычислительные мощности и объём данных
- В 2005-2006 годах происходит прорыв в области НС и в решении некоторых задач (например, распознавание речи)

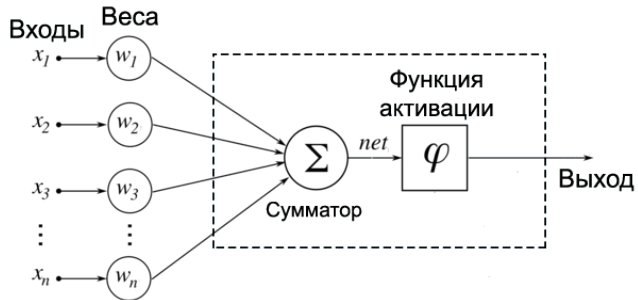
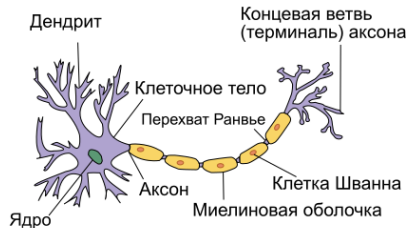
- Нейронные сети довольно универсальны, и небольшие доработки в архитектуре позволяют решать широкий круг задач
- Часто применяется технология предобучения на большом объеме неразмеченных данных с последующим дообучением на небольшом объеме размеченных данных

- В классическом ML очень важный этап - инженерия признаков
- Итоговое качество сильно зависит от этого этапа
- В DL упрощается и даже исключается инженерия признаков
- DL модели способны самостоятельно извлекать признаки из данных, иногда даже за гранью понимания человека
- Это делает DL модели крайне эффективными и популярными для решения сложных задач с большим объёмом данных
- Однако требуется больше вычислительных мощностей

- Распознавание речи (Microsoft Cortana, Xbox, Skype Translator, Amazon Alexa, Google Now, Apple Siri, Baidu и iFlyTek)
- CV - Компьютерное зрение (распознавание образов с результатами превосходящими человеческий глаз)
- NLP - Обработка естественного языка (например, LSTM для машинного перевода и языкового моделирования)
- Рекомендательные системы
- ...

Нейронная сеть

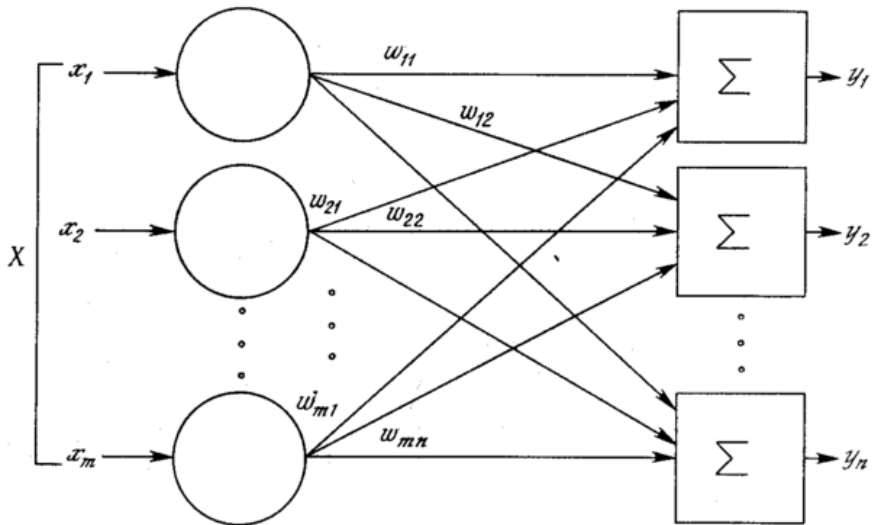
Типичная структура нейрона



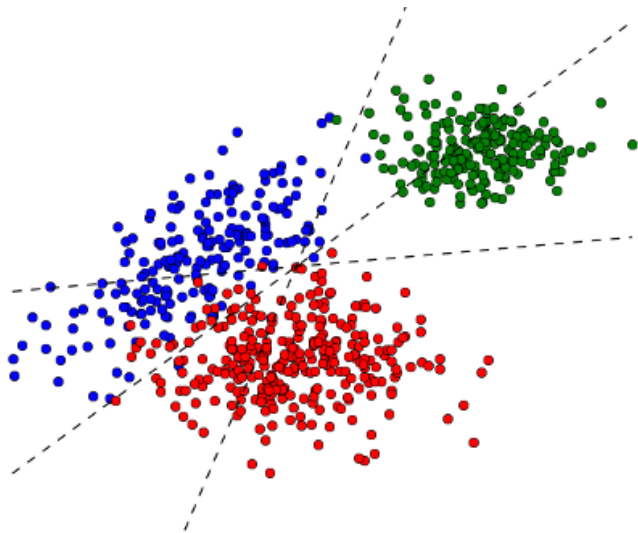
Линейный классификатор

- Рассмотрим пример классификации рукописных цифр по ч/б изображениям 8×8
- Распрямляем матрицу 8×8 в вектор 1×64
- Классов у цифр 10, поэтому чтобы получить предсказания умножаем такие вектора на матрицу W весов размера 64×10
- Добавляем свободный член для каждого из 10 классов с помощью вектора b размера 1×10
- В итоге получаем вектор размера 1×10 с числами (по одному числу для каждой из 10 цифр)
- Эти числа должны описывать соответствие переданной на вход картинке по отношению к каждому из 10 классов (наибольшее число сигнализирует наше предсказание и соответствующий нейрон - активировался)
- Получили многоклассовую линейную классификацию

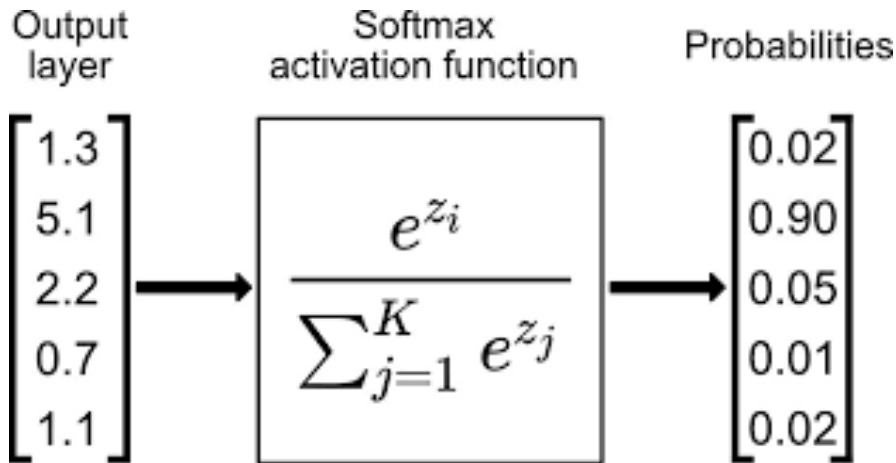
Однослойная нейронная сеть



Разделяющие плоскости

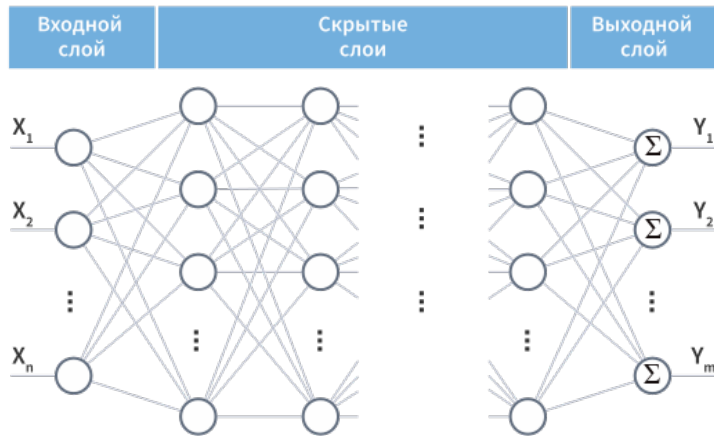


- Как и раньше ищем оптимальные веса W и b
- Градиентный спуск, стохастический градиентный спуск
- Также добавляем регуляризацию (L_1 , L_2)
- Хотели бы получать не просто числа, а оценки вероятностей
- Для бинарной классификации из лекции про логистическую регрессию помним об использовании сигмоиды и применении метода максимального правдоподобия
- В многоклассовом случае сигмоида обобщается функцией **softmax**

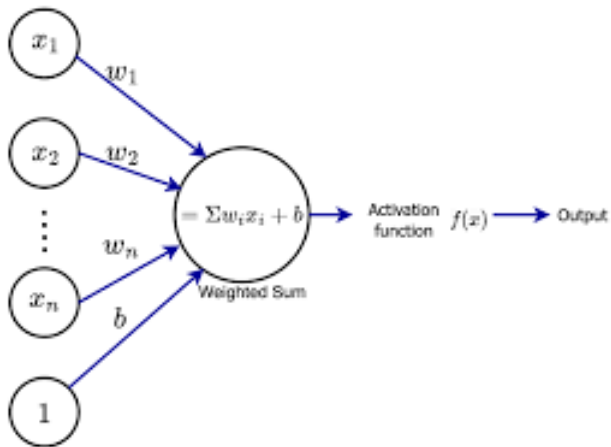


- Наиболее частый выбор на практике из-за хороших результатов
- Дифференцируемость
- Вероятности, соответствующие более большим числам, еще существеннее отрываються от других

Многослойная нейронная сеть

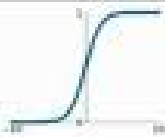


Добавление нелинейности между слоями



Sigmoid

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



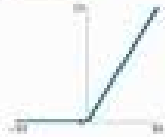
tanh

$$\tanh(x)$$



ReLU

$$\max(0, x)$$



Leaky ReLU

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

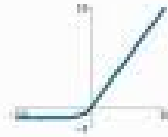


Maxout

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

ELU

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



Шаг градиентного спуска

- Обычно вектор b добавляют в матрицу весов W , а в данные добавляют фиктивный столбец из единиц или минус единиц
- Функция ошибок у нас получилась такая:

$$L(X, W, y) = -\frac{1}{n} \sum_{i=1}^n \ln \frac{\exp(x_i \cdot w^{y_i})}{\sum_{j=1}^m \exp(x_i \cdot w^j)} + \lambda R(w),$$

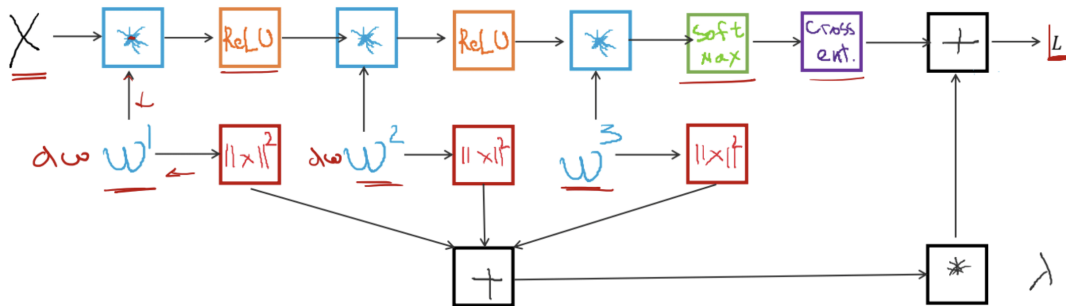
где

$$x_i \cdot w^j = \sum_{k=0}^p x_i^k w_k^j$$

— скалярное произведение соответствующих векторов

- Осталось понять как считать антиградиент функции ошибок
- Нейронные сети могут быть намного сложнее, данных очень много

Computational graph



Backpropagation

Backpropagation: Simple Example

$$f(x, y, z) = (x + y)z$$

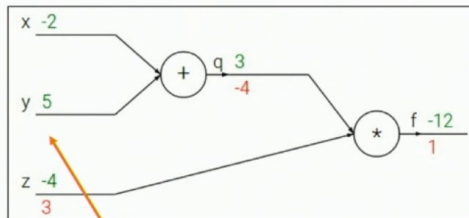
e.g. $x = -2, y = 5, z = -4$

1. Forward pass: Compute outputs

$$q = x + y \quad f = qz$$

2. Backward pass: Compute derivatives

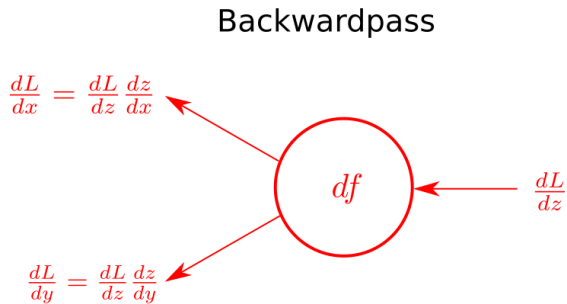
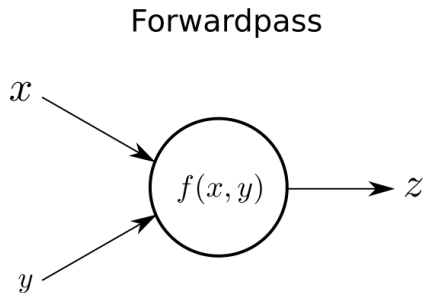
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



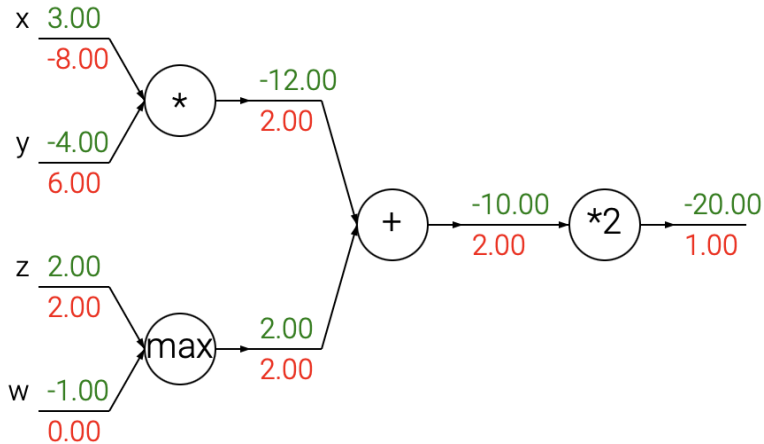
Chain Rule

$$\frac{\partial f}{\partial y} = \frac{\partial q}{\partial y} \frac{\partial f}{\partial q}$$

Общая схема вычисления градиента



Пример



Дополнительные источники

- <https://cs231n.github.io>
- <https://dlcourse.ai>