

Deep Learning

Урок 1

Егор Конягин

6 августа 2024 г.

1. Введение
2. История Deep learning
3. Логистическая регрессия

Введение

Machine learning (ML) - совокупность методов, использующих статистические закономерности и другие характеристики данных для решения определенного круга задач. Традиционно методы ML делятся на две группы: supervised learning и unsupervised learning.

К задачам **supervised learning** относятся:

- классификация (бинарная и многоклассовая);
- задача регрессии.

К задачам **unsupervised learning** относятся:

- задача кластеризации;
- задача визуализации;
- задача понижения размерности.

Deep learning (DL) - раздел машинного обучения, относящийся к обучению с учителем (supervised learning), в основе которого лежат искусственные нейронные сети (ANN, artificial neural networks).

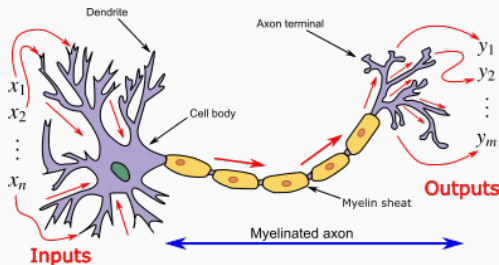


Рис. 1: Математическая модель биологического нейрона - перцептрон

История Deep learning

Исторический обзор

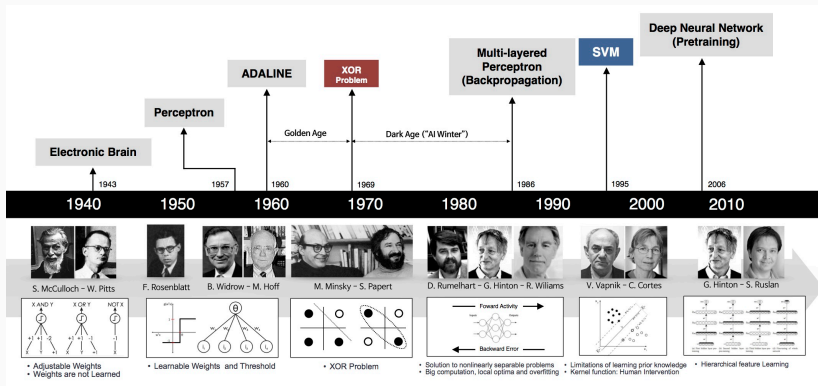


Рис. 2: Хронология развития DL. Источник: nvidia AI blog

Back propagation

Back propagation (метод обратного распространения ошибки) - алгоритм подбора параметров аппроксимирующей модели, предложенный в 1986 г. группой ученых во главе с Джоффри Хинтоном (Google Brain). Алгоритм является итеративным и выглядит следующим образом:

1. Выбор функции ошибки $\mathcal{L}(y, \hat{y})$;
2. Подсчет текущего значения функции ошибки $\mathcal{L}(y, \hat{y})$;
3. Подсчет градиента функции ошибки по параметрам модели:

$$\nabla \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial w_1} \cdots \frac{\partial \mathcal{L}}{\partial w_N} \right)^T \quad (1)$$

4. Обновление параметров модели:

$$w_i = w_i - \alpha \cdot \nabla \mathcal{L}_i \quad (2)$$

5. Переход к п. 2.

Универсальная теорема об аппроксимации

Универсальная теорема об аппроксимации - теорема, звучащая следующим образом:

Пусть $\varphi : \mathcal{R} \rightarrow \mathcal{R}$ - это некая ограниченная, непрерывная и неконстантная функция. Пусть I_m - это m -мерный замкнутый гиперкуб вида $[0, 1]^m$.

Тогда для любого $\varepsilon > 0$ и для любой функции $f \in C(I_m)$ найдется целое число N и набор чисел $v_i, b_i \in \mathcal{R}$ и набор действительных векторов w_i таких что:

$$F(x) = \sum_{i=1}^N v_i \varphi(w_i^T x + b_i), \quad (3)$$

$$|F(x) - f(x)| < \varepsilon. \quad (4)$$

Зима AI. Прорыв deep learning

Зима AI (AI winter) - период начала 2000-ых годов, характеризующийся отсутствием значимых результатов в области DL.

2012 год Нейронная сеть **AlexNet** с существенным отрывом одержала победу в соревновании по классификации изображений ImageNet Large Scale Visual Recognition Challenge.

2016 год Поражение Ли Седоля в игру Go.

Факторы, способствующие развитию и распространению DL:

- необходимые алгоритмы;
- большие объемы данных;
- обучение моделей с использованием GPU.

Логистическая регрессия

Логистическая регрессия. Введение

Логистическая регрессия - алгоритм бинарной классификации, который подходит для работы с линейно разделимыми данными. Логистическую регрессию можно интерпретировать как нейронную сеть, состоящую из одного единственного нейрона.

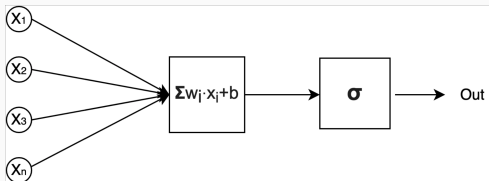


Рис. 3: Модель логистической регрессии

Нам понадобится определение сигмоидной функции $\sigma(x)$:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \quad (5)$$

Обучение логистической регрессии можно разбить на несколько шагов:

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

В данной модели инициализация весов не имеет большого значения, поэтому все веса можно проинициализировать нулями.

Forward propagation в данной задаче описывается 2-мя уравнениями. Для i -ого объекта $x^{(i)}$:

$$z = w \cdot x^{(i)} + b \quad (6)$$

$$\hat{y} = \text{sigmoid}(z) \quad (7)$$

Логистическая регрессия. Backward propagation

Back propagation реализуется с помощью уравнений, написанных ниже. Для подсчитанной функции потерь:

$$J = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(a^{(i)}, y^{(i)}) = -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log(a^{(i)}) + (1 - y^{(i)}) \log(1 - a^{(i)}) \quad (8)$$

подсчитываем градиент функции ошибки (backward propagation):

$$\frac{\partial J}{\partial w} = ? \quad (9)$$

$$\frac{\partial J}{\partial b} = ? \quad (10)$$

ВАЖНО Подсчет функции потерь - это суммирование функций ошибки на каждом образце.

Обновление параметров в данной модели описывается двумя простыми уравнениями:

$$w = w - \alpha \cdot \nabla J_w \quad (11)$$

$$b = b - \alpha \cdot \frac{dJ}{db} \quad (12)$$

В качестве ДЗ предлагается реализовать описанную модель логистической регрессии и обучить ее на датасете изображений. Изображения двух типов: на одних в кадре присутствует коробка, на других - нет. Это задача бинарной классификации.

Для работы с файлами, которые лежат в определенной директории в google drive, необходимо выполнить следующие команды:

```
from google.colab import drive  
drive.mount('/content/gdrive')  
path = "/content/gdrive/My Drive/__path_to_folder__"
```