Deep Learning

Урок 1

Егор Конягин

МФТИ & АО ”ЦОСиВТ”

Содержание

1. Введение

2. История Deep learning

3. Логистическая регрессия

4. Технические вопросы

Введение

Машинное обучение

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.

Глубокое обучение

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

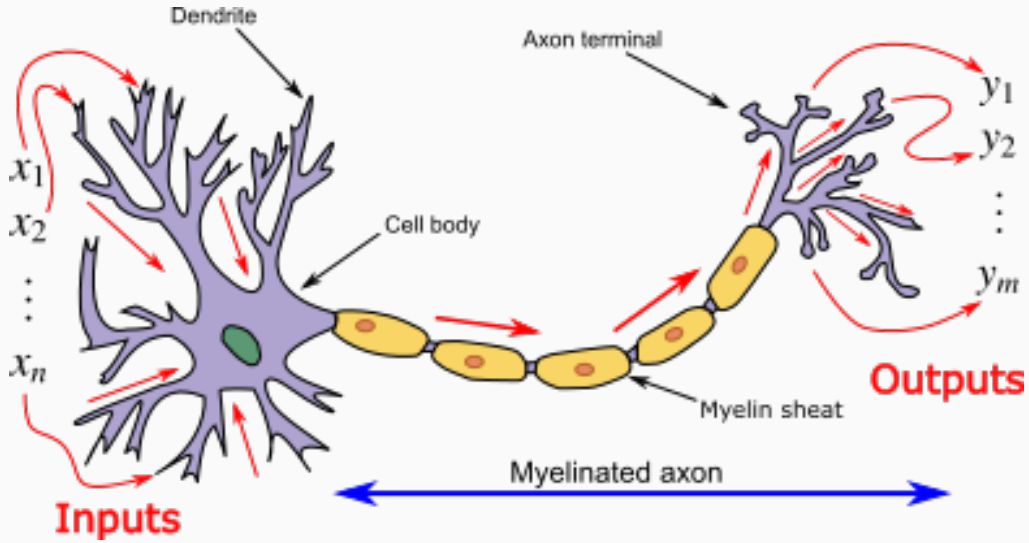

Рис. 1: Математическая модель биологического нейрона - перцептрон

История Deep learning

Исторический обзор

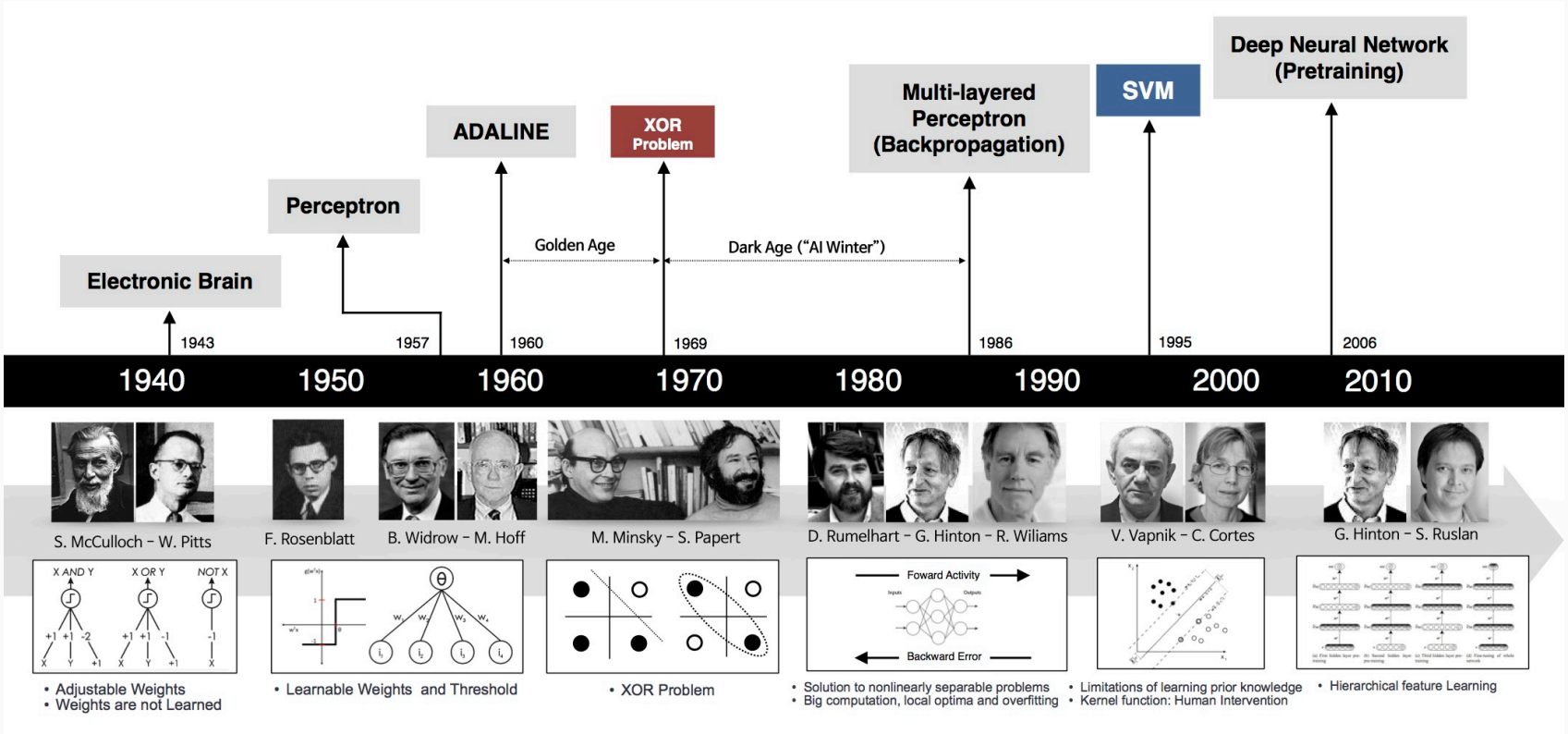

Рис. 2: Хронолония развития DL. Источник: nvidia AI blog

Метод группового учёта аргументов

Метод группового учёта аргументов - алгоритм аппроксимации, предложенный советским ученым Алексеем Ивахненко в 1968 году. Считается одним из ранних методов deep learning.

Общий вид аппроксимирующей модели:

$$\hat{Y}(x_1, \ldots, x_n) := a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} \sum_{j=i}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n} \sum_{j=i}^{n} \sum_{k=j}^{n} a_{ijk} x_i x_j x_k + \cdots \tag{1}$$

Коэффициенты $a_{i \ldots j \ldots k}$ находятся методом наименьших коэффициентов.

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:

$$\nabla \mathcal{L} := (\frac{\partial \mathcal{L}}{\partial w_1} \cdots \frac{\partial \mathcal{L}}{\partial w_N})^T \tag{2}$$

4. Обновление параметров модели:

$$w_i := w_i - \alpha \cdot \nabla \mathcal{L}_i \tag{3}$$

5. Переход к п. 2.

Универсальная теорема об аппроксимации

Универсальная теорема об аппроксимации - теорема, звучащая следующим образом:

Пусть $\varphi : \mathcal{R} \to \mathcal{R}$ - 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 действительнозначных векторов w_i таких что:

$$F(x) := \sum_{i=1}^{N} v_i \varphi \left(w_i^T x + b_i \right), \tag{4}$$

$$|F(x) - f(x)| < \varepsilon. \tag{5}$$

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 интерпретровать как нейронную сеть, состоящую из одного единственного нейрона.

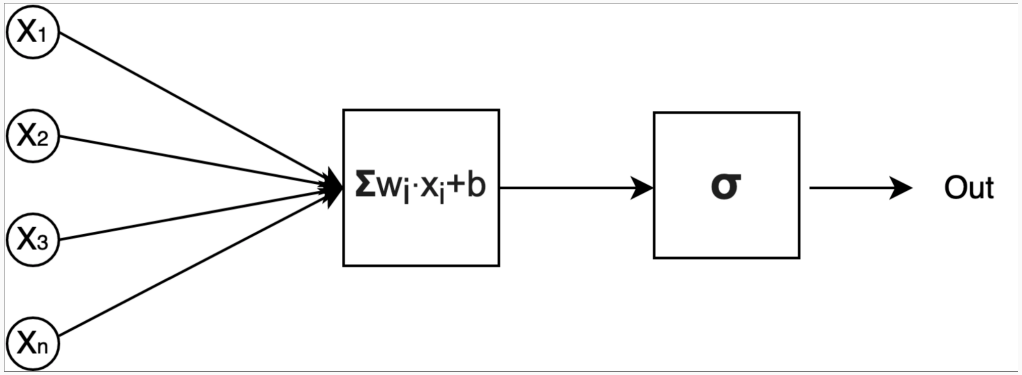

Рис. 3: Модель логистической регрессии

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

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{6}$$

Логистическая регрессия. Описание модели

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.

Логистическая регрессия. Инициализация весов

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

Логистическая регрессия. Forward propagation

Forward propagation в данной задаче описывается 3-мя уравнениями. Для i-ого образца $x^{(i)}$:

$$z^{(i)} := w^T x^{(i)} + b \tag{7}$$

$$\hat{y}^{(i)} := sigmoid(z^{(i)}) \tag{8}$$

$$\mathcal{L}(\hat{y}^{(i)}, y^{(i)}) := -y^{(i)} \log(\hat{y}^{(i)}) - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \tag{9}$$

Логистическая регрессия. Backward propagation

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

$$J := \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(a^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\log(a^{(i)}) + (1-y^{(i)})\log(1-a^{(i)}) \tag{10}$$

подсчитываем градиент функции ошибки (backward propagation):

$$\frac{\partial J}{\partial w} := \nabla J_w = \frac{1}{m}X(A-Y)^T \tag{11}$$

$$\frac{\partial J}{\partial b} := \frac{dJ}{db} = \frac{1}{m}\sum_{i=1}^{m}(a^{(i)} - y^{(i)}) \tag{12}$$

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

Логистическая регрессия. Обновление параметров

Обновление параметров в данной модели описывается двумя простыми уравнениями:

$$w := w - \alpha \cdot \nabla J_w \tag{13}$$

$$b := b - \alpha \cdot \frac{dJ}{db} \tag{14}$$

Технические вопросы

Среда разработки. Библиотеки

Задание можно выполнять либо в среде **Google Colab** (облачная среда), либо в среде **Jupyter Notebook** (локальная среда). На первом этапе мы будем использовать библиотеку **NumPy** для проведения всех вычислений.

Описание домашнего задания

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.

Работа с файлами в среде Google Colab

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__"
```