Лекция 10
Нейронные сети

Макаренко В.А., Габдуллин Р.А.

МГУ им. М.В. Ломоносова

17 марта 2023

История

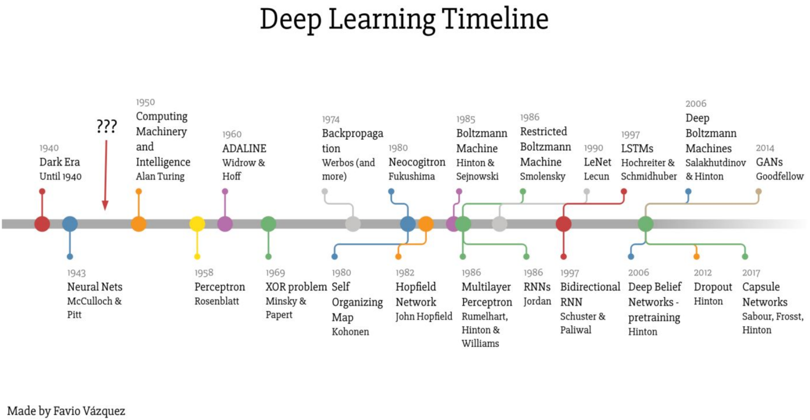


Рис.: Источник: maelfabien.github.io

Модель нейрона МакКаллока-Питтса

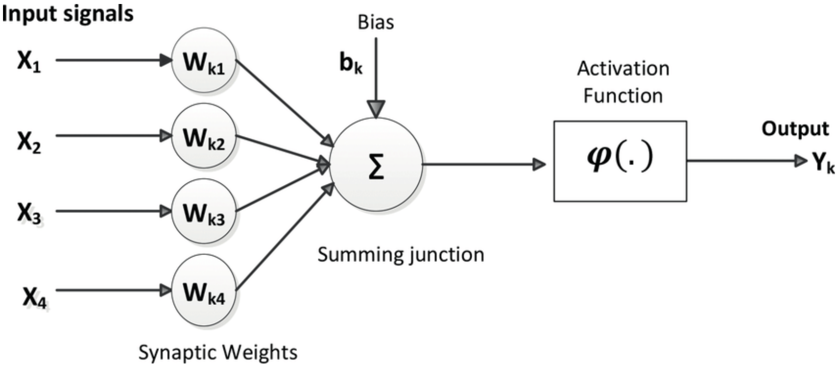


Рис.: Источник: researchgate.net

Многослойная нейронная сеть

Пусть $x \in \mathbb{R}^p$ – вектор признаков.

Граф вычислений:

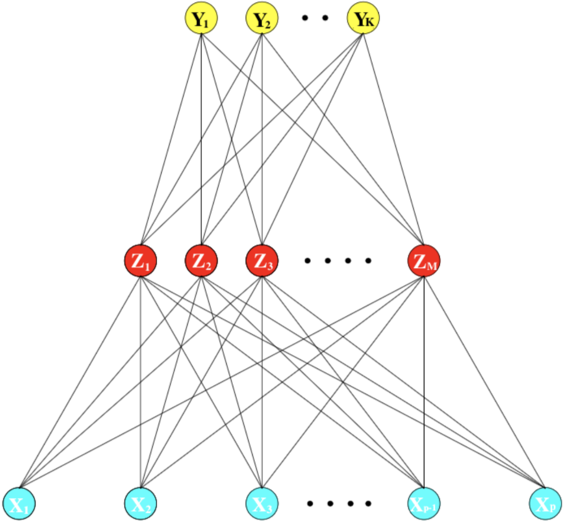
$$
\begin{array}{rcc}
z_1(x) = f_1(x, w_1), & z_1(x) \in \mathbb{R}^{p_1}, & w_1 \in \mathbb{R}^{m_1}, \\
z_2(x) = f_2(z_1(x), w_2), & z_2(x) \in \mathbb{R}^{p_2}, & w_2 \in \mathbb{R}^{m_2}, \\
& \dots & \\
z_{n-1}(x) = f_{n-1}(z_{n-2}(x), w_{n-1}), & z_{n-1}(x) \in \mathbb{R}^{p_{n-1}}, & w_{n-1} \in \mathbb{R}^{m_{n-1}}, \\
y(x) = f_n(z_{n-1}(x), w_n), & y(x) \in \mathbb{R}^K, & w_n \in \mathbb{R}^{m_n}.
\end{array}
$$

Полносвязная нейронная сеть:

$$
f_{ij}(z, W, w_0) = \sigma_i \left(w_{0j} + \sum_{k=1}^{p_{i-1}} W_{jk} z_k \right),
$$

$$
1 \leqslant j \leqslant p_i, \quad W \in \mathbb{R}^{p_i \times p_{i-1}}, \quad w_0 \in \mathbb{R}^{p_i}.
$$

Многослойная нейронная сеть



Вычислительные возможности нейронных сетей

$\sigma(z)$ – сигмоида, если $\lim\limits_{z\to-\infty} \sigma(z) = 0$ и $\lim\limits_{z\to+\infty} \sigma(z) = 1$.

Теорема (Цыбенко, 1989)

Если $\sigma(z)$ – непрерывная сигмоида, то для любой непрерывной на $[0,1]^n$ функции $f(x)$ существуют такие значения параметров $w_h \in \mathbb{R}^n, w_0 \in \mathbb{R}, \alpha_h \in \mathbb{R}$, что двухслойная сеть

$$a(x) = \sum_{h=1}^{H} \alpha_h \sigma(x^T w_h + w_0)$$

равномерно приближает $f(x)$ с любой точностью $\varepsilon > 0$:

$$|a(x) - f(x)| < \varepsilon, \quad \forall x \in [0,1]^n.$$

Функции активации

Распространенные функции активации:

- Логистическая:
$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$
- Гиперболический тангенс:
$$\sigma(x) = \mathrm{th}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$
- Rectified linear unit (ReLU):
$$\sigma(x) = \max\{0, x\}.$$
- Leaked rectified linear unit (Leaked ReLU):
$$\sigma_\alpha(x) = \begin{cases} x, & x \geqslant 0, \\ \alpha x, & x < 0. \end{cases}$$

Обучение нейронной сети

Эмпирический риск:

$$Q(w) = \sum_{k=1}^{\ell} L(y_k, a(x_k, w)).$$

- Минимизация эмпирического риска с помощью градиентных методов оптимизации.
- Используем правило дифференцирования сложных функций для вычисления частных производных:

$$\frac{\partial Q}{\partial w_i} = \frac{\partial Q}{\partial f_{i+1}} \frac{\partial f_{i+1}}{\partial f_i} \frac{\partial f_i}{\partial w_i}.$$

- Каждая итерация обучения состоит из двух проходов по сети: прямого и обратного.
  - Прямой шаг. Вычисляются значения всех слоев f_i и прогнозы $a(x)$.
  - Обратный шаг. Вычисляются все производные $\partial Q / \partial w_i$.

Градиентный спуск и его модификации

Эмпирический риск:

$$Q(w) = \sum_{k=1}^{\ell} L(y_k, a(x_k, w)).$$

- Градиентный спуск:

$$w_{\text{new}} := w_{\text{old}} - \eta \cdot Q'(w_{\text{old}}).$$

- Метод накопления импульса (Momentum):

$$v_{\text{new}} := \gamma v_{\text{old}} + (1-\gamma) Q'(w_{\text{old}}),$$

$$w_{\text{new}} := w_{\text{old}} - \eta v_{\text{new}}.$$

- Ускоренный градиентный метод Нестерова (NAG):

$$v_{\text{new}} := \gamma v_{\text{old}} + (1-\gamma) Q'(w_{\text{old}} - \eta\gamma v_{\text{old}}),$$

$$w_{\text{new}} := w_{\text{old}} - \eta v_{\text{new}}.$$

Dropout

- **Этап обучения**. На каждой итерации градиентного спуска «выключаем» j-й нейрон i-го слоя с вероятностью p_i независимо от других нейронов.
- **Этап применения**. Включаем все нейроны, но с поправкой $(1 - p_i)$.

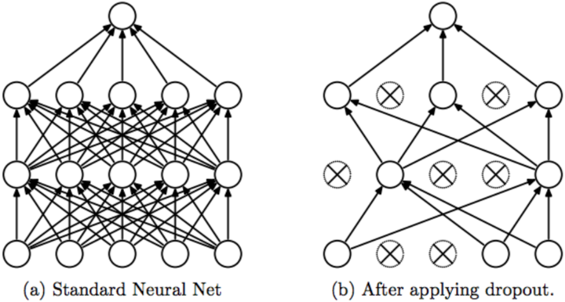


Рис.: Источник: medium.com

Пакетная нормализация данных (Batch normalization)

$B = \{x_i\}$ – mini-batch данных.
$B^\ell = \{u_i^\ell\}$ – векторы объектов на выходе ℓ-го слоя.

Пакетная нормализация:

- Стандартизируем j-ю компоненту вектора u_i^ℓ по пакету:

$$\hat{u}_{ij}^\ell = \frac{u_{ij}^\ell - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}}, \quad \mu_j = \frac{1}{|B|} \sum_{x_i \in B} u_{ij}^\ell, \quad \sigma_j^2 = \frac{1}{|B|} \sum_{x_i \in B} (u_{ij}^\ell - \mu_j)^2.$$

- Добавляем линейный слой с настраиваемыми весами:

$$\tilde{u}_{ij}^\ell = \gamma_j^\ell \hat{u}_{ij}^\ell + \beta_j^\ell.$$

Рекуррентные нейронные сети

- В рекуррентных слоях появляется «время».
- В каждый следующий момент времени t на вход сети подается очередной элемент x_t из входной последовательности.
- Рекуррентный слой хранит скрытое состояние h_t, которое обновляется с приходом нового элемента последовательности.

$$h_t = \sigma_1(W_x x_t + W_h h_{t-1}), \quad y_t = \sigma_2(W_y h_t).$$

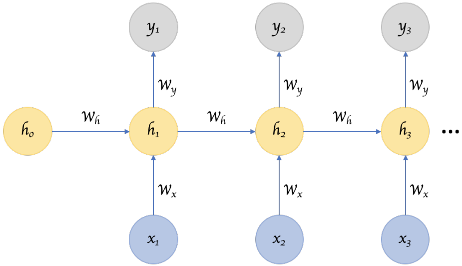


Рис.: Источник: towardsdatascience.com

Transfer learning

- Используем модель, обученную для решения похожей задачи.
- Используем выходы одного из последних слоев в качестве признаков для решения задачи.
- Обучаем новую модель (часто довольно простую) на новых признаках.

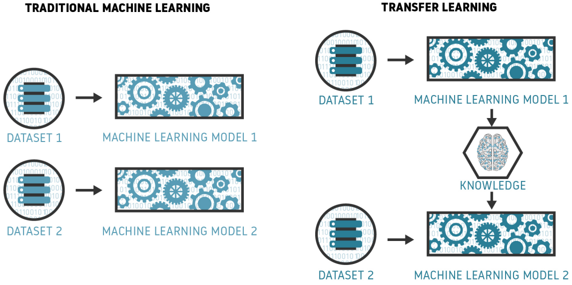


Рис.: Источник: datascience.aero