

# 1 Детерминированные нейронные сети.

Сначала напомним, что такое обычные(детерминированные) нейронные сети и как они обучаются.

Основная задача обычных искусственных нейронных сетей(ANN) в том, чтобы аппроксимировать некоторую зависимость выхода  $y$  от входа  $x$ :  $y = \Phi(x)$ .

Для простоты будем рассматривать обычные *полносвязные* сети со входом  $x$ , скрытыми(промежуточными) состояниями слоёв  $\mathbf{h}_i$ , функциями активации  $a_i(\cdot)$  и выходом  $y$ :

$$\mathbf{h}_0 = x$$

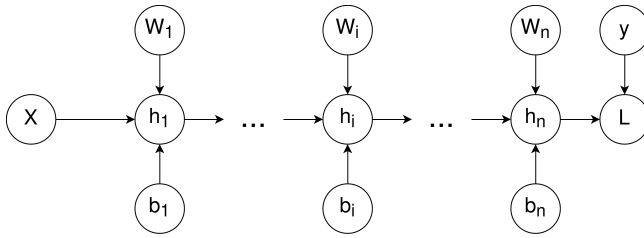
$$\mathbf{h}_i = a_i(\mathbf{W}_i \cdot \mathbf{h}_{i-1} + \mathbf{b}_i), i = \overline{1...n}$$

$$\mathbf{h}_n = \hat{y}$$

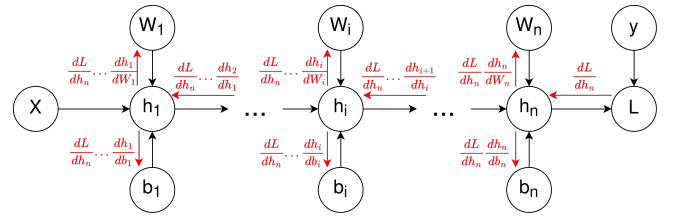
$$L = \mathcal{L}(\hat{y}, y),$$

где  $\mathcal{L}(\cdot, \cdot)$  - функция ошибки.

Обозначим параметры модели на  $i$ -ом слое  $\theta_i = (\mathbf{W}_i, \mathbf{b}_i)$ , а параметры всей модели через  $\Theta = \{\theta_i : i = \overline{1...n}\}$ . Чаще всего нейронные сети принято рассматривать, как вычислительный граф/граф вычислений. Такой подход удобен с инженерной точки зрения, поскольку позволяет воспользоваться инструментом автоматического дифференцирования, и используется во всех современных фреймворках: PyTorch, TensorFlow и прочие. Граф вычислений является ациклическим ориентированным графом, составленным из вершин-переменных и вершин-операций(Рисунок 1).



(a) Полносвязная сеть в виде графа вычислений



(b) Обратное распространение ошибки по графу вычислений детерминированной полносвязной сети