Задача 1

- Пусть нейронная сеть состоит из следующий слоев:

  1. Входной слой: 4 нейрона, без активационной функции,

  2. Средний слой: 2 нейрона, ReLU ($\max(0, x)$),

  3. Выходной слой: 2 нейрона, softmax,

$$g_i(x) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Задача 1

- Первый и второй нейроны входного слоя соединены с первым нейроном среднего слоя; третий и четвертый нейроны входного слоя соединены со вторым нейроном среднего слоя.

- Как реализовать стохастический градиентный спуск для функции потерь, основанной на перекрестной энтропии?

Решение

- Формула стохастического градиентного спуска:

$$w := w - \eta \nabla L_w(x_{(i)}),$$

где функция потерь

$$L_w(x_{(1)}, \dots, x_{(n)}) = \sum_{i=1}^{n} L_w(x_{(i)}).$$

- Веса́ нейронной сети $w = (w_1, w_2, w_3, w_4)$, расположены на ребрах, соединяющих нейроны входного слоя со средним.

Решение

- Функция потерь, основанная на перекрестной энтропии, для бинарного классификатора:

$$L_w(x_1, x_2, x_3, x_4) = -y \log F_{0;w}(x_1, x_2, x_3, x_4)$$

$$-(1 - y) \log F_{1;w}(x_1, x_2, x_3, x_4),$$

где $y \in \{0, 1\}$, F_0, F_1 – вероятности первого и второго класса, возвращаемые классификатором.

Решение

- Градиент от функции потерь – вектор производных по весам w_1, w_2, w_3, w_4.

- На основании схемы нейронной сети получим выражение для F_1, F_2:

$$F_{0;w}(x_1, x_2, x_3, x_4) = (1 + \exp(\max(0, w_1 x_1 + w_2 x_2) - \max(0, w_3 x_3 + w_4 x_4)))^{-1}$$

$$F_{1;w}(x_1, x_2, x_3, x_4) = (1 + \exp(\max(0, w_3 x_3 + w_4 x_4) - \max(0, w_1 x_1 + w_2 x_2)))^{-1}$$

Решение

- Таким образом, функция потерь

$$L_w(x_1, x_2, x_3, x_4) =$$

$$y \log(1 + \exp(\max(0, w_1 x_1 + w_2 x_2)$$

$$- \max(0, w_3 x_3 + w_4 x_4)))$$

$$+(1 - y) \log(1 + \exp(\max(0, w_3 x_3 + w_4 x_4)$$

$$- \max(0, w_1 x_1 + w_2 x_2)))$$

- Осталось взять производные $\frac{\partial L_w}{\partial w_i}$, $i = 1, 2, 3, 4$.

Задача 2 (Тема «Фильтр Блума»)

Пусть доступно n бит памяти, а множество ключей S содержит m элементов. Вместо использования k хеш-функций можно разделить n бит на k массивов и хешировать по одному разу в каждый массив. Какова вероятность ложноположительного значения в зависимости от n, m и k?

Exercise 4.3.2: Suppose we have n bits of memory available, and our set S has m members. Instead of using k hash functions, we could divide the n bits into k arrays, and hash once to each array. As a function of n, m, and k, what is the probability of a false positive? How does it compare with using k hash functions into a single array?