

Алгоритм обратного распространения ошибки (backpropagation) используется для настройки весов нейронной сети, минимизируя ошибку между предсказанием сети и реальными значениями.

1. Инициализация. Веса инициализируются случайными числами.
2. Прямой проход (Forward Pass). Для каждого входного примера вычисляется предсказание сети. Рассчитывается ошибка.
3. Обратное распространение ошибки (Backward Pass).

Обратный проход сети используется для вычисления градиентов ошибки относительно весов.

Вычисление градиентов ошибки

На выходном слое вычисляется производная функции потерь по выходу сети

$$\delta_L = \frac{\partial L}{\partial a_L} \cdot \sigma'(z_L),$$

где $\sigma'(z_L)$ - производная функции активации последнего слоя.

Ошибка распространяется на скрытые слои:

$$\delta_l = (\delta_{l+1} \cdot w_{l+1}) \cdot \sigma'(z_l),$$

где δ_{l+1} - ошибка на предыдущем слое,

w_{l+1} - веса, соединяющие текущий слой с предыдущим,

$\sigma'(z_l)$ - производная функции активации текущего слоя.

Вычисляется градиент весов:

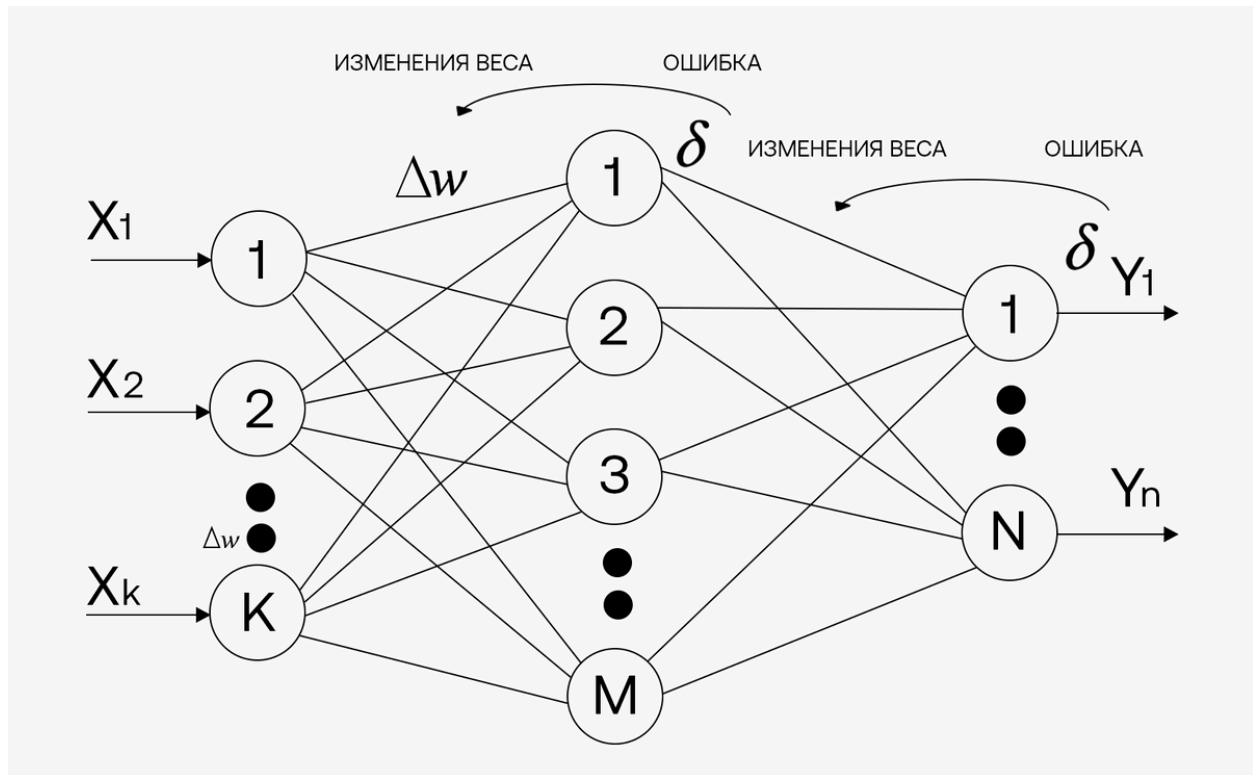
$$\frac{\partial L}{\partial w_L} = \delta_l \cdot a_{l-1}$$

4. Выполняется обновление весов (с помощью градиентного спуска):

$$w_{k+1} = w_k + \eta \cdot \frac{\partial L}{\partial w_k},$$

где η - скорость обучения (learning rate)

Процесс (прямой проход → обратное распространение → обновление весов) повторяется для всех батчей, пока ошибка не станет достаточно малой или не будет достигнут заданный критерий остановки (например, количество эпох).



Для обучения многослойного персептрона задаче классификации используются функции активации sigmoid и softmax, функция потерь - бинарная кросс-энтропия.

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$$

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

$$\text{BCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)],$$

где N - количество примеров в выборке

y_i - истинная метка

p_i - предсказанная вероятность

Набор данных был разделен на тренировочный (80%) и тестовый (20%).

Для выбора признаков для предсказания использован алгоритм жадного прямого выбора.

Сначала для предсказания выбираются все признаки, модель обучается, считается качество. Далее из набора удаляется один признак, обучается другой персептрон, считается качество. Если качество улучшилось - изменение в наборе признаков для обучения сохраняется. Если качество ухудшилось - признак возвращается в набор.

Процедура повторяется для всех признаков. Максимальное полученное качество:

Accuracy: 0.92982

Cross-entropy: -0.87003