① 11, 25 Дубинин А. О.
М80-407Б-17.

№25 Обучение многослойных сетей прямого распространения. Алгоритм обратного распространения ошибки – общая схема. Цепное правило дифференцирования и его роль в алго. back propagation. Сопоставление алгоритма наискорейшего спуска и алгоритма LMS для минимизации ошибки.

• Общая схема обучения многослойных нейронных сетей

1. После того, как мы выбрали структуру НС, необходимо её обучение, т.е осуществить подбор значений синаптических весов и смещений сети

(2) Эту задачу мы будем решать, реализуя процесс обучения с учителем. Суть состоит в следующем:

Для сети зададим нач. значения весов W (обычно небольшие, равномерно распределенные на $[0, 1]$). Подадим на вход НС некоторый набор сигналов

$$X^* = (X_1^*, \ldots, X_N^*).$$

Распространяясь по сети, этот набор порождает выходной набор

$$y = (y_1, \ldots, y_n).$$

Значения вектора y зависят от X^* и от W.

2. Пусть для данного X^* известно, каким должно быть „правильное" (желаемое) значени

y^* выходного вектора y.

3 (именно это значение предъ-
являются учителем на
соответствующем шаге)

Найдем уклонение значения
выхода y^* от выхода y сети,
который получился для значения
x^* при фиксированном W;
оно для данного x^* предста-
вляет собой ошибку, явля-
ющуюся функцией от
параметров W

$$\varepsilon(W) = \| y^* - y(x^*, W) \|$$

пара $p = \langle x^*, y^* \rangle$
– обучающий пример, а
набор из N_p пар – обучающий
набор.

Функция ошибки м.б. опреде-
лена, как на $p \in P$, так и
на $P = \{P_i\}$

$\mathcal{E} = \mathcal{E}(p, W)$; $\mathcal{E} = \mathcal{E}(P, W)$.

4) Задача обучения сети — это задача минимизации функции ошибки по настраиваемым параметрам (весам и смещениям) W:

$$\mathcal{E}(W^*) = \min_W \mathcal{E}(p, W)$$

для отдельных примеров.

$$\mathcal{E}(W^*) = \min_W \mathcal{E}(P, W)$$

для случая использования обучающего набора в целом.

- Алгоритм обратного распространения ошибки

Алгоритм backpropagation и его разновидности — основа одного из наиболее часто применяемых подходов к решению задачи обучения многослойных сетей.

(5) Идея алгоритма совокупность шагов:

1. Весам сети W присваиваются небольшие начальные значения.

2. Выбирается очередной обучающий пример $p_k = \langle x_k^*, y_k^* \rangle$ из обучающего набора

$$P = \{p_k\} = \{\langle x_k^*, y_k^* \rangle, k = 1, \dots, N_p\},$$

3. Вычисляется выход сети $y(x_k^*, W)$, отвечающий входному вектору x_k^* и текущим W

4. На основе желаемого y_k^* для p_k и вычисленного на предыдущем шаге выхода $y(x_k^*, W)$ вычисляется ошибка сети $e_k(\cdot)$ для

данного примера:

6

$\varepsilon_k(X_k^*, W) = \| y_k^* - y(x_k^*, W) \|$

5. Шаги 2, 3 и 4 повторяются для всех обучающих примеров p_k, $k = 1, \dots, N_p$, при одном и том же значении W; эти N_p шагов составляют одну эпоху в процессе обучения сети.

6. Вычисляется суммарная ошибка сети для r-й эпохи

$$\varepsilon^{(r)} = \sum_{k=1}^{N_p} \varepsilon_k.$$

7. Веса сети W корректируются так, чтобы уменьшить ошибку $\varepsilon^{(r)}(\cdot)$

8. Шаги со 2 по 7 повторяются до тех пор, пока не будет выполнено условие

7 $\varepsilon^{(4)}(W) \le \varepsilon$, либо по пределу макс. числа эпох

$$W^{r+1} = W^{r} - \eta^{r} \frac{\partial \varepsilon}{\partial W}$$

корректировка весов,

η^{r} - темп обучения

В однослойной сети обучение осуществляется с использованием дельта правила.
В случае многослойной сети задача усложняется. Для выходного слоя проблем нет. Для скрытых слоёв все по-другому. Проблема состоит в том, что данные за непосредственного вычисления значений ошибки на выходах нейронов недоступны. Надо его определять

ошибку на выходах скрытых слоев по ошибки на пред. слое. Идет распространение назад. Надо уметь эффективно вычислять значения частных производных ф-ций ошибки сети. Цель BP состоит в том, чтобы при вычислении градиента $\partial E / \partial W$ уйти от проклятия размерности

$$w^m_{i,j}(k+1) = w^m_{i,j}(k) - \alpha \frac{\partial E}{\partial w^m_{i,j}}$$

$$b^m_i(k+1) = b^m_i(k) - \alpha \frac{\partial E}{\partial b^m_i}$$

- Цепное правило

Сложная сеть представляет композицию отображений

Производная сложной ф-ции

(9) Равна произведению её производной по промежуточному аргументу на производную этого аргумента по независимой переменной — правило цепочки дифференцирования.

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$$

С помощью правила можно вычислить $\partial E / \partial W$

$$W^m(k+1) = W^m(k) - \alpha \delta^m (y^{m-1})^T$$

$$b^m(k+1) = b^m(k) - \alpha \delta^m$$

$$\delta^m = \frac{\partial E}{\partial v^m} = \left[\frac{\partial E}{\partial v_1^m} \; \frac{\partial E}{\partial v_2^m} \; \dots \; \frac{\partial E}{\partial v_{S^m}^m}\right]^T$$

- Сопоставление LMS и алго наискорейшего спуска.
- Алгоритм BP в исходном варианте основан на оптимизационной схеме наискорейшего спуска = SD BP (Steppest Descent BP).

Алгоритм SDBP, ~~как и~~ можно трактовать как обобщение алгоритма LMS. Он хорошо справляется с задачей обучения однослойной сети, квадр. ф. ошибки которой имеет единственный минимум. Однако функция ошибки многослойных сетей является многоэкстремальной.

$$W^1(k+1) = W^1(k) - 2\alpha\delta^1(y^0)^T =$$

$$= W^1(k) + 2\alpha e x^T$$

(11) $b^1(k+1) = b^1(k) - 2\alpha s = b^1(k) + 2\alpha e$ – это в точности алгоритм LMS. Таким образом ВР можно считать обобщением алгоритма LMS.