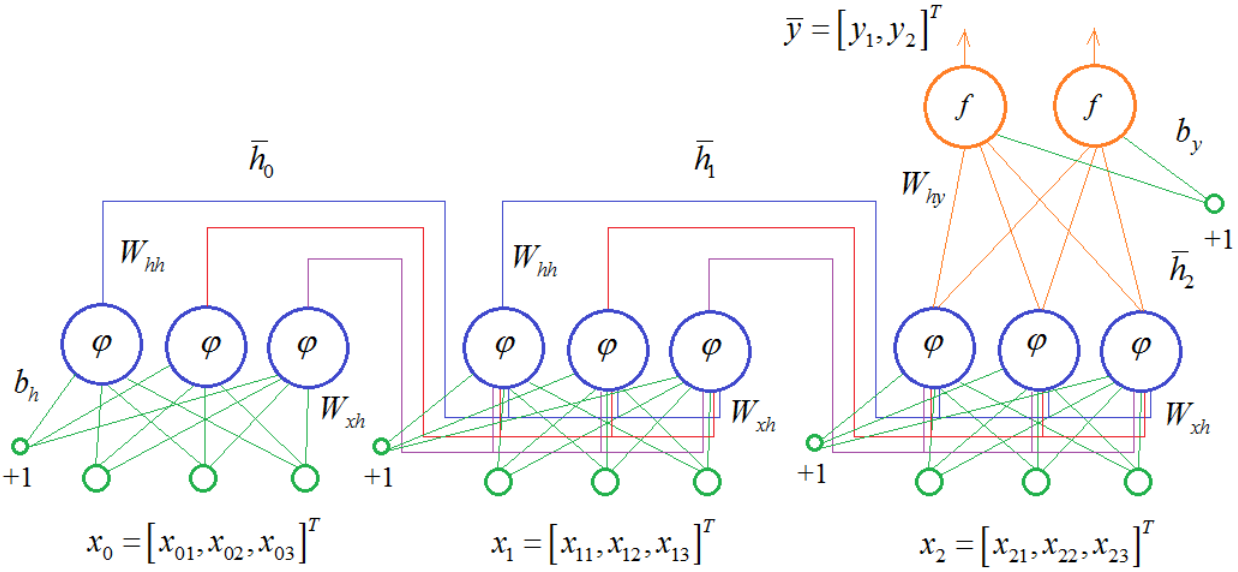
## Пример алгоритма обратного распространения ошибки по времени (Back Propagation Through Time)




Развернем рекуррентную НС на три шага во времени. В каждый момент времени на ее вход подается вектор $\bar{x}$, и на третьем шаге смотрим выходное значение $\bar{y}$. Для обучения такой сети можно использовать алгоритм

back propagation

с учетом временного характера поведения сети. В нашем случае рекуррентная сеть построена по принципу

many to one (многие к одному)

то есть, множество входных сигналов и один выходной. Так как мы рассматриваем задачу классификации, то в качестве функции активации выходных нейронов выберем softmax, а потери будем считать через кросс-энтропию:

$$E = -\sum_{i=1}^{M} t_i \log(y_i) = -\sum_{i=1}^{M} t_i \log(softmax(v_i))$$

где M = 2 – число выходных нейронов; $v_i$ - индуцированный (суммарный) сигнал на входах нейронов последнего слоя; $t_i = \{0;1\}$ - требуемые выходные значения (в нашем случае 1 или 0); log – логарифм, например, натуральный.

Первым делом нам нужно вычислить градиент функции потерь E от входного значения $v_i$. Для строго определенного класса

$$i = C$$

функция

$$E = -\log\left(softmax\left(v_C\right)\right)$$

Следовательно:

$$\frac{\partial E}{\partial v_i} = \begin{cases} y_i - 1, & i = C \\ y_i, & i \neq C \end{cases}$$

Например, если на выходе наблюдаем значения:

$$\overline{y} = \left[0,8;0,2\right]^T$$

то $\frac{\partial E}{\partial \overline{v}}$ при $C = 1$ образует вектор:

$$\frac{\partial E}{\partial \overline{v}} = \left[0,8-1;0,2\right]^T = \left[-0,2;0,2\right]^T$$

Как видите, все предельно просто. Далее, определим градиент для весов матрицы $W_{hy}$:

$$\frac{\partial E}{\partial W_{hy}} = \frac{\partial E}{\partial \overline{v}} \cdot \frac{\partial \overline{v}}{\partial W_{hy}}$$

Так как

$$\overline{v} = W_{hy} \cdot \overline{h}_n + b_y$$

(здесь $\overline{h}_n = \overline{h}_2$), то

$$\frac{\partial \overline{v}}{\partial W_{hy}} = \overline{h}_n^T$$

и градиент изменения весов:

$$\frac{\partial E}{\partial W_{hy}} = \frac{\partial E}{\partial \overline{v}} \cdot \overline{h}_n^T$$

Аналогично вычисляется градиент изменения биаса:

$$\frac{\partial E}{\partial b_y} = \frac{\partial E}{\partial \overline{v}} \cdot \frac{\partial \overline{v}}{\partial b_y} = \frac{\partial E}{\partial \overline{v}}$$

Используя эти величины, мы теперь можем применить алгоритм градиентного спуска для корректировки весов $W_{hy}$ и смещения $b_y$:

$$W_{hy} = W_{hy} - \lambda \cdot \frac{\partial E}{\partial \overline{v}} \cdot \overline{h}_n^T$$

$$b_y = b_y - \lambda \cdot \frac{\partial E}{\partial \overline{v}}$$

Осталось выполнить похожие вычисления для весов матриц $W_{hh}, W_{xh}$ и биаса $b_h$. Начнем с матрицы $W_{hh}$:

$$\frac{\partial E}{\partial W_{hh}} = \frac{\partial E}{\partial \overline{v}} \cdot \frac{\partial \overline{v}}{\partial \overline{h}_n} \cdot \frac{\partial \overline{h}_n}{\partial W_{hh}}$$

Это выражение было бы верно для обычных сетей прямого распространения. Но у нас здесь рекуррентная сеть и значение вектора:

$$\overline{h}_n = \varphi\left(W_{hh} \cdot \overline{h}_{n-1} + W_{xh} \cdot \overline{x}_n + b_h\right)$$

зависит от значения на предыдущем шаге. Поэтому для корректного вычисления градиента нам нужно пройти по всей рекурсии до самого начала:

$$\frac{\partial E}{\partial W_{hh}} = \frac{\partial E}{\partial \overline{v}} \cdot \left[ \frac{\partial \overline{v}}{\partial \overline{h}_n} \cdot \frac{\partial \overline{h}_n}{\partial W_{hh}} + \frac{\partial \overline{v}}{\partial \overline{h}_{n-1}} \cdot \frac{\partial \overline{h}_{n-1}}{\partial W_{hh}} + \ldots + \frac{\partial \overline{v}}{\partial \overline{h}_0} \cdot \frac{\partial \overline{h}_0}{\partial W_{hh}} \right]$$

В результате, получаем такую формулу:

$$\frac{\partial E}{\partial W_{hh}} = \frac{\partial E}{\partial \overline{v}} \cdot \sum_{t=0}^{n} \frac{\partial \overline{v}}{\partial \overline{h}_t} \cdot \frac{\partial \overline{h}_t}{\partial W_{hh}}$$

Так как

$$\overline{h}_t = \varphi\left(W_{hh} \cdot \overline{h}_{t-1} + W_{xh} \cdot \overline{x}_t + b_t\right)$$

то

$$\frac{\partial \overline{h}_t}{\partial W_{hh}} = \varphi'\left(x\right) \cdot \overline{h}_{t-1}^T$$

Давайте для определенности положим, что функция активации – это гиперболический тангенс:

$$\varphi\left(x\right) = \tanh\left(x\right)$$

производная которой, равна:

$$\varphi'\left(x\right) = 1 - \tanh^2\left(x\right)$$

Тогда

$$\frac{\partial \overline{h}_t}{\partial W_{hh}} = \left(1 - \overline{h}_t^2\right) \cdot \overline{h}_{t-1}^T$$

Далее, нужно вычислить производную $\frac{\partial \overline{v}}{\partial \overline{h}_t}$. Так как она зависит от всех предыдущих выходов $\overline{h}_t$, то проще всего ее вычислять по рекурсии:

$$\frac{\partial \overline{v}}{\partial \overline{h}_t} = \frac{\partial \overline{v}}{\partial \overline{h}_{t+1}} \cdot \frac{\partial \overline{h}_{t+1}}{\partial \overline{h}_t}$$

где

$$\frac{\partial \overline{h}_{t+1}}{\partial \overline{h}_t} = \varphi'\left(x\right) \cdot W_{hh}^T = \left[1 - \overline{h}_t^2\right] \cdot W_{hh}^T$$

а

$$\frac{\partial \overline{v}}{\partial h_n} = W_{hy}^T$$

Имея эти выражения, мы теперь можем вычислить суммарный градиент по всей рекурсии (то есть по времени) для матрицы весов $W_{hh}$:

$$\frac{\partial E}{\partial W_{hh}} = \frac{\partial E}{\partial \overline{v}} \cdot \sum_{t=0}^{n} \frac{\partial \overline{v}}{\partial \overline{h}_t} \cdot \left(1 - \overline{h}_t^2\right) \cdot \overline{h}_{t-1}^T$$

Аналогично вычисляются следующие градиенты:

$$\frac{\partial E}{\partial W_{xh}} = \frac{\partial E}{\partial \overline{v}} \cdot \sum_{t=0}^{n} \frac{\partial \overline{v}}{\partial \overline{h}_t} \cdot \left(1 - \overline{h}_t^2\right) \cdot \overline{x}_t^T$$

$$\frac{\partial E}{\partial b_h} = \frac{\partial E}{\partial \overline{v}} \cdot \sum_{t=0}^{n} \frac{\partial \overline{v}}{\partial \overline{h}_t} \cdot \left(1 - \overline{h}_t^2\right)$$

Используя эти выражения, выполняем корректировку соответствующих весов:

$$W_{hh} = W_{hh} - \lambda \cdot \frac{\partial E}{\partial W_{hh}}$$
$$W_{xh} = W_{xh} - \lambda \cdot \frac{\partial E}{\partial W_{xh}}$$
$$b_h = b_h - \lambda \cdot \frac{\partial E}{\partial b_h}$$

Вот так выглядит алгоритм обратного распространения ошибки по времени для рекуррентных нейронных сетей.