Одно- и многослойные нейронные сети,

Обучение методом обратного распространения ошибки

- Осовский С. **Нейронные сети для обработки информации** / Пер. с польского И.Д. Рудинского - М.: Финансы и статистика, 2002

Однослойная сеть

Однослойную сеть образуют нейроны, расположенные в одной плоскости. Каждый *i*-й нейрон имеет поляризацию (связь с весом w_{i0}, по которой поступает единичный сигнал), а также множество связей с весами w_{ij}, по которым поступают входные сигналы x_j.

Значения весов подбираются в процессе обучения сети, заключающемся в приближении выходных сигналов y_i к ожидаемым значениям d_i. Мерой близости считается значение целевой функции $E(w_{ij})$. При использовании *p* обучающих векторов $\langle \vec{x}, \vec{d} \rangle$ для обучения сети, включающей *M* выходных нейронов, целевая функция определяется евклидовой метрикой в виде:

$$E = \frac{1}{2}\sum_{k=1}^{p} \| \vec{y^{(k)}} - \vec{d^{(k)}} \|^2 = \frac{1}{2}\sum_{k=1}^{p}\sum_{i=1}^{M} (y_i^{(k)} - d_i^{(k)})^2$$

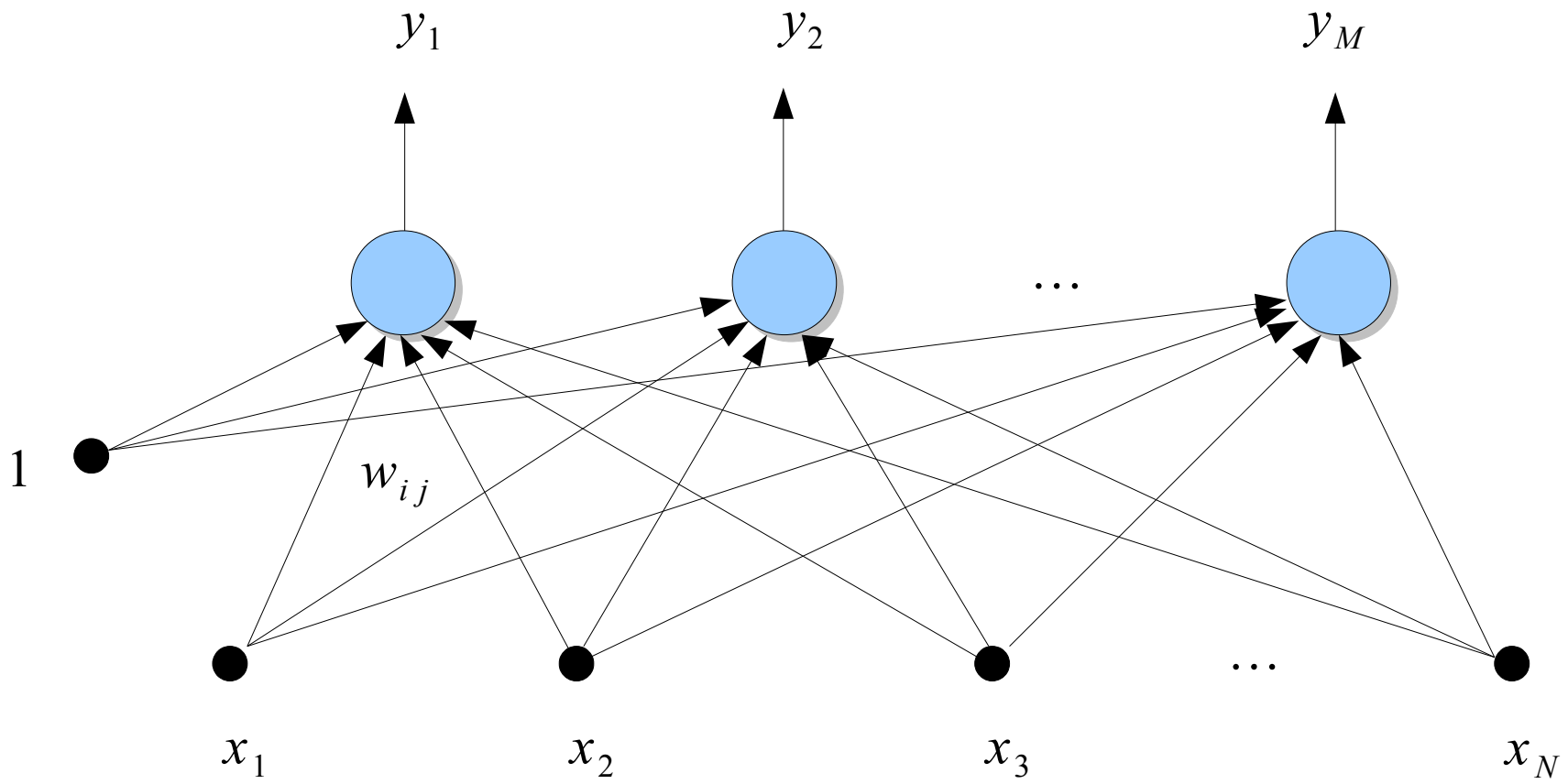


Рис. 1. Схема однослойной ИНС

Расположенные на одном уровне нейроны функционируют независимо друг от друга, поэтому возможности такой сети ограничиваются свойствами отдельных нейронов.

Несмотря на то, что однослойная сеть имеет небольшое практическое применение, её продолжают использовать там, где для решения поставленной задачи задачи достаточно и одного слоя нейронов.

Выбор архитектуры такой сети весьма прост. Количество входных нейронов (сенсоров) определяется размерностью входного вектора $\vec{x}$, а количество выходных нейронов определяется размерностью вектора $\vec{d}$.

Обучение сети производится, как правило, с учителем и является точной копией обучения одиночного нейрона.

Многослойный перцептрон (многослойная ИНС)

Многослойная сеть состоит из нейронов, расположенных на разных уровнях, причём, помимо входного и выходного слоёв, имеется ещё, как минимум, один внутренний, т. е. **скрытый**, слой.

Такая нейронная сеть также называется **многослойным перцептроном**.

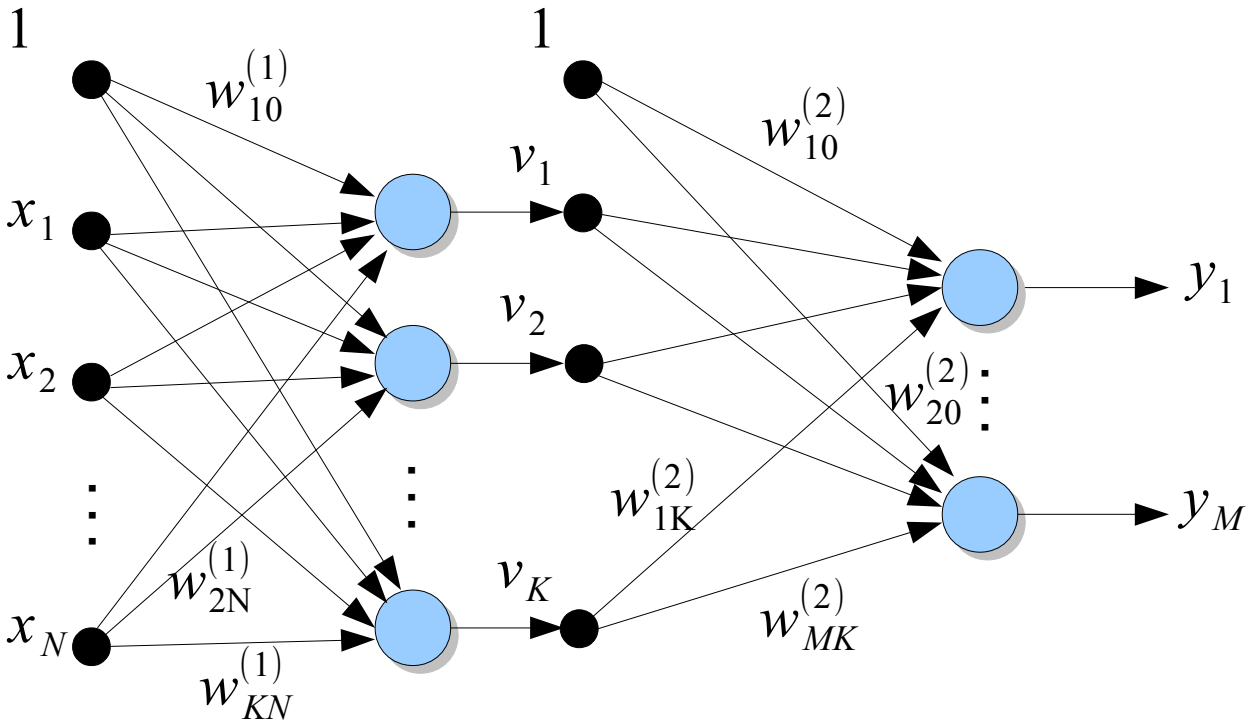


Рис. 2. Двухслойная полносязная ИНС с одним скрытым слоем

Выходные сигналы нейронов скрытого слоя обозначаются $v_j, j=1,2,\ldots,K$, а выходного слоя - $y_j, j=1,2,\ldots,M$.

Пусть функция активации задана в сигмоидальной форме. С целью упрощения записи будет использоваться расширенное обозначение входного вектора сети в виде $\vec{x}=(x_0, x_1, \ldots, x_N)^T$, где $x_0=1$ соответствует единичному сигналу поляризации. С вектором $\vec{x}$ связаны два выходных вектора сети:

- Вектор фактических выходных сигналов: $\vec{y}=(y_1, y_2, \ldots, y_M)^T$
- Вектор ожидаемых выходных сигналов: $\vec{d}=(d_1, d_2, \ldots, d_M)^T$

Цель обучения состоит в подборе таких значений весов $w_{ij}^{(1)}$ и $w_{ij}^{(2)}$ для всех слоёв сети, чтобы при заданном векторе $\vec{x}$ получить на выходе значения сигналов y_i, которые с требуемой точностью будут совпадать с ожидаемыми значениями $d_i \quad \forall i=1,2,\ldots,M$

С учётом включения веса сигнала поляризации, выходной сигнал *i* -го нейрона скрытого слоя описывается функцией:

$$v_i = f\left(\sum_{j=0}^{N} w_{ij}^{(1)} x_j\right)$$

При этом для входа на следующем слое принимается $v_0 = 1$. В выходном слое *k* -й нейрон вырабатывает выходной сигнал, вычисляемый по формуле:

$$y_k = f\left(\sum_{i=0}^{K} w_{ki}^{(2)} v_i\right) = f\left(\sum_{i=0}^{K} w_{ki}^{(2)}\left(\sum_{j=0}^{N} w_{ij}^{(1)} x_j\right)\right)$$

Из этой формулы видно, что **на значение выходного слоя влияют веса обоих слоёв**.

Необходим способ обучения многослойных сетей с учётом влияния всех слоёв на выход сети.

Алгоритм обратного распространения ошибки

Алгоритм обратного распространения ошибки (error back propagation) определяет стратегию подбора весов многослойной сети с применением градиентных методов оптимизации. Его основу составляет **целевая функция**, формулируемая, как правило,в виде квадратичной суммы разностей между фактическими и ожидаемыми значениями выходных сигналов. В случае единичной обуающей выборки $\langle \vec{x}, \vec{d} \rangle$ целевая функция определяется в виде:

$$E(w) = \frac{1}{2} \sum_{k=1}^{M} (y_k - d_k)^2$$

При большем количестве обучающих выборок $j = 1, 2, \ldots, p$ целевая функция превращается в сумму по всем выборкам:

$$E(w) = \frac{1}{2} \sum_{j=1}^{p} \sum_{k=1}^{M} (y_k^{(j)} - d_k^{(j)})^2$$

Уточнение весов может проводиться после предъявления каждой обучающей выборки (режим «онлайн») либо однократно после предъявления всех выборок, составляющих цикл обучения (режим «оффлайн»)

Обучение производится в несколько этапов:

1. Предъявляется обучающая выборка $\vec{x}$ и рассчитываются значения сигналов соответствующих нейронов сети. Для заданного $\vec{x}$ определяются значения выходных сигналов v_i скрытого слоя, а затем – значения y_i нейронов выходного слоя (в случае двухслойной сети) по формулам:

$$v_i = f\left(\sum_{j=0}^{N} w_{ij}^{(1)} x_j\right) \quad , \quad y_k = f\left(\sum_{i=0}^{K} w_{ki}^{(2)} v_i\right) = f\left(\sum_{i=0}^{K} w_{ki}^{(2)} \left(\sum_{j=0}^{N} w_{ij}^{(1)} x_j\right)\right)$$

2. Минимизируется значение целевой функции.

Если целевая функция непрерывна, то наиболее эффективным способом обучения оказывается применение градиентных методов оптимизации, согласно которым уточнение вектора весов производится по формуле:

$$\vec{w}(k+1) = \vec{w}(k) + \Delta\vec{w}$$

где $\Delta\vec{w} = \eta\, p(\vec{w})$, η - скорость обучения, $p(\vec{w})$ - направление в многомерном пространстве значений весовых коэффициентов $\vec{w}$.

Обучение многослойной сети с применением градиентных методов требует определения вектора градиента **для каждого слоя сети**, что необходимо для правильного выбора направления $p(\vec{w})$. Эта задача имеет очевидное решение только для весов нейронов выходного слоя.

Для обучения внутренних слоёв необходимо учитывать влияние этих слоёв на итоговый выход сети и ту погрешность, которая создаётся под их влиянием.

Для обучения внутренних слоёв с учетом влияния их погрешности была создана специальная стратегия, называемая алгоритмом обратного распространения ошибки.

Как уже говорилось при рассмотрении алгоритма обучения нейрона с помощью градиентного метода, веса нейронов корректируются в сторону убывания целевой функции, т. е. в сторону, противоположную градиенту с координатами:

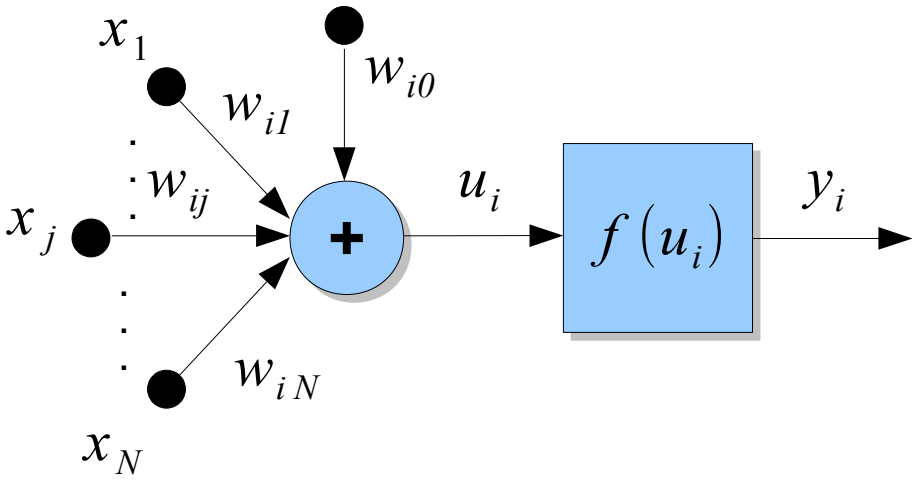
$$\left(\nabla_0 E, \nabla_1 E, \nabla_2 E, \ldots, \nabla_N E\right)^T \quad \text{, где} \quad \nabla_j E = \frac{\partial E}{\partial w_{ij}}$$

Компонент градиента, как приводилось ранее, расписывается следующим образом:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_i} \cdot \frac{dy_i}{du_i} \cdot \frac{\partial u_i}{\partial w_{ij}} = \left(y_i - d_i\right) \cdot f'\left(u_i\right) \cdot x_j$$

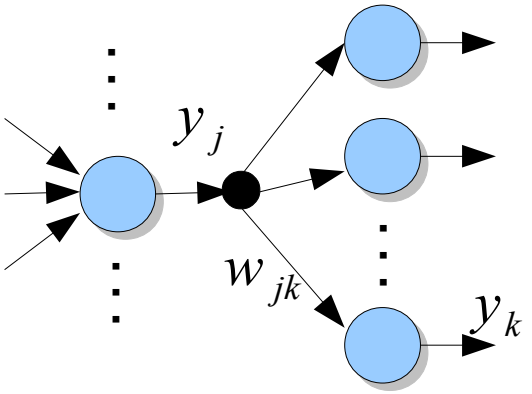
Однако применить её можно только к обучению нейронов выходного слоя.

Для обучения внутренних слоёв необходимо учитывать их влияние на целевую функцию.



Рассмотрим зависимость целевой функции от выхода *j*-го нейрона предпоследнего слоя (*n* < *N*):

$$\frac{\partial E}{\partial y_j} = \sum_{k=1}^{K^{n+1}} \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{du_k} \cdot \frac{\partial u_k}{\partial y_j} = \sum_{k=1}^{K^{n+1}} \frac{\partial E}{\partial y_k} \cdot f'(u_k) \cdot w_{jk}^{(n+1)}$$



где суммирование производится по всем *K* нейронам следующего слоя (*n*+1), сязанным с рассматриваемым j-м нейроном текущего слоя *n*.

Обозначим: $\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot f'(u_j)$, тогда получим рекуррентное соотношение:

$$\frac{\partial E}{\partial y_j^{(n)}} \cdot f'(u_j) = \left[\sum_{k=1}^{K^{n+1}} \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot f'(u_j) = \delta_j^{(n)}$$

При этом для выходного слоя *N* : $\delta_j^N = (y_j^N - d_j) \cdot f'(u_j)$

Таким образом, для произвольного слоя *n* компоненты градиента вычисляются:

$$\Delta w_{ij}^{(n)} = -\eta \delta_j^{(n)} y_i^{(n-1)}$$

очевидно, что для первого слоя (*n*=1): $y_i^{(1)} = x_i$

Иногда для устранения эффекта осцилляции вводят коэффициенты инерционности обучения:

$$\Delta w_{ij}^{(n)}(t+1) = -\eta \delta_j^{(n)} y_i^{(n-1)} + \alpha w_{ij}^{(n)}(t), \qquad |\alpha| < 1$$

или

$$\Delta w_{ij}^{(n)}(t+1) = -\eta \left[\mu \Delta w_{ij}^{(n)}(t) + (1-\mu) \delta_j^{(n)} y_i^{(n-1)} \right], \qquad |\mu| < 1$$

Алгоритм обратного распространения ошибок можно описать в следующем виде: