

# Лекция 5. Нейронные сети: обучение без учителя

---

Главная черта, делающая обучение без учителя привлекательным, – это его "самостоятельность".

Процесс обучения, как и в случае обучения с учителем, заключается в подстраивании весов синапсов. Некоторые алгоритмы, правда, изменяют и структуру сети, то есть количество нейронов и их взаимосвязи, но такие преобразования правильнее назвать более широким термином – самоорганизацией, и в рамках данной главы они рассматриваться не будут.

Очевидно, что подстройка синапсов может проводиться только на основании информации, доступной в нейроне, то есть его состояния и уже имеющихся весовых коэффициентов. Исходя из этого соображения и, что более важно, по аналогии с известными принципами самоорганизации нервных клеток, построены алгоритмы обучения Хебба.

Сигнальный метод обучения Хебба заключается в изменении весов по следующему правилу:

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot y_i^{(n-1)} \cdot y_j^{(n)} \quad (1)$$

Где  $y_i^{(n-1)}$  – выходное значение нейрона  $i$  слоя  $(n-1)$ ,  $y_j^{(n)}$  – выходное значение нейрона  $j$  слоя  $n$ ;  $w_{ij}(t), w_{ij}(t-1)$  – весовой коэффициент синапса, соединяющего эти нейроны, на итерациях  $t$  и  $t-1$  соответственно;  $\alpha$  – коэффициент скорости обучения.

Здесь и далее, для общности, под  $n$  подразумевается произвольный слой сети. При обучении по данному методу усиливаются связи между возбужденными нейронами.

Существует также и дифференциальный метод обучения Хебба.

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot [y_i^{(n-1)}(t) - y_i^{(n-1)}(t-1)] \cdot [y_j^{(n)}(t) - y_j^{(n)}(t-1)] \quad (2)$$

Здесь  $y_i^{(n-1)}(t)$  и  $y_i^{(n-1)}(t-1)$  - выходное значение нейрона  $i$  слоя  $n-1$  соответственно на итерациях  $t$  и  $t-1$ ;  $y_i^{(n)}(t)$  и  $y_i^{(n)}(t-1)$  - то же самое для нейрона  $j$  слоя  $n$ .

Как видно из формулы (2), сильнее всего обучаются синапсы, соединяющие те нейроны, выходы которых наиболее динамично изменились в сторону увеличения.

Полный алгоритм обучения с применением вышеприведенных формул будет выглядеть так:

1. На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения.
2. На входы сети подается входной образ, и сигналы возбуждения распространяются по всем слоям согласно принципам классических прямоточных (feed forward) сетей, то есть для каждого нейрона рассчитывается взвешенная сумма его входов, к которой затем применяется активационная функция нейрона, в результате чего получается его выходное значение  $y_i^{(n)}, i = 0 \dots M_i - 1$ , где  $M_i$  – число нейронов в слое  $i$ ;  $n = 0 \dots N - 1$ , а  $N$  – число слоёв в сети.
3. На основании полученных выходных значений нейронов по формуле (1) или (2) производится изменение весовых коэффициентов.
4. Цикл с шага 2, пока выходные значения сети не застabilизируются с заданной точностью.

Применение этого нового способа определения завершения обучения, отличного от использовавшегося для сети обратного распространения, обусловлено тем, что подстраиваемые значения синапсов фактически не ограничены.

На втором шаге цикла попеременно предъявляются все образы из входного набора.

Следует отметить, что вид откликов на каждый класс входных образов не известен заранее и будет представлять собой произвольное сочетание состояний нейронов выходного слоя, обусловленное случайным распределением весов на стадии инициализации.

Вместе с тем, сеть способна обобщать схожие образы, относя их к одному классу. Тестирование обученной сети позволяет определить топологию классов в выходном слое.

Для приведения откликов обученной сети к удобному представлению можно дополнить сеть одним слоем, который, например, по алгоритму обучения однослойного перцептрона необходимо заставить отображать выходные реакции сети в требуемые образы.

Другой алгоритм обучения без учителя – алгоритм Кохонена – предусматривает подстройку синапсов на основании их значений от предыдущей итерации.

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot \left[ y_i^{(n-1)} - w_{ij}(t-1) \right] \quad (3)$$

Из вышеприведенной формулы видно, что обучение сводится к минимизации разницы между входными сигналами нейрона, поступающими с выходов нейронов предыдущего слоя  $y_i^{(n-1)}$ , и весовыми коэффициентами его синапсов.

Полный алгоритм обучения имеет примерно такую же структуру, как в методах Хебба, но на шаге 3 из всего слоя выбирается нейрон, значения синапсов которого максимально походят на входной образ, и подстройка весов по формуле (3) проводится только для него.

Эта, так называемая, аккредитация может сопровождаться затормаживанием всех остальных нейронов слоя и введением выбранного нейрона в насыщение.

Выбор такого нейрона может осуществляться, например, расчетом скалярного произведения вектора весовых коэффициентов с вектором входных значений.

Максимальное произведение дает “выигравший” нейрон.



Другой вариант – расчет расстояния между этими векторами в  $p$ -мерном пространстве, где  $p$  – размер векторов.

$$D_j = \sqrt{\sum_{i=0}^{p-1} (y_i^{(n-1)} - w_{ij})^2} \quad (4)$$

где  $j$  – индекс нейрона в слое  $n$ ,  $i$  – индекс суммирования по нейронам слоя  $(n-1)$ ,  $w_{ij}$  – вес синапса, соединяющего нейроны; выходы нейронов слоя  $(n-1)$  являются входными значениями для слоя  $n$ .

Корень в формуле (4) брать не обязательно, так как важна лишь относительная оценка различных  $D_j$

В данном случае, "побеждает" нейрон с наименьшим расстоянием. Иногда слишком часто получающие аккредитацию нейроны принудительно исключаются из рассмотрения, чтобы "уравнять права" всех нейронов слоя. Простейший вариант такого алгоритма заключается в торможении только что выигравшего нейрона.

При использовании обучения по алгоритму Кохонена существует практика нормализации входных образов, а так же – на стадии инициализации – и нормализации начальных значений весовых коэффициентов.

$$x_i = \frac{x_i}{\sqrt{\sum_{j=0}^{n-1} x_j^2}} \quad (5)$$

где  $x_i$  –  $i$ -ая компонента вектора входного образа или вектора весовых коэффициентов, а  $n$  – его размерность. Это позволяет сократить длительность процесса обучения.

Инициализация весовых коэффициентов случайными значениями может привести к тому, что различные классы, которым соответствуют плотно распределенные входные образы, сольются или, наоборот, раздробятся на дополнительные подклассы в случае близких образов одного и того же класса.

Для избегания такой ситуации используется метод выпуклой комбинации [Лигун А. А., Малышева А. Д. Математическая обработка результатов эксперимента. // Днепропетровск: ДИИ, 1992—47с ].

Суть его сводится к тому, что входные нормализованные образы подвергаются преобразованию:

$$x_i = \alpha(t) \cdot x_i + (1 - \alpha(t)) \cdot \frac{1}{\sqrt{n}} \quad (6)$$

где  $x_i$  –  $i$ -ая компонента входного образа,  $n$  – общее число его компонент,  $\alpha(t)$  – коэффициент, изменяющийся в процессе обучения от 0 до 1, в результате чего вначале на входы сети подаются практически одинаковые образы, а с течением времени они все больше сходятся к исходным. Весовые коэффициенты устанавливаются на шаге инициализации равными величине:

$$w_0 = \frac{1}{\sqrt{n}} \quad (7)$$

где  $n$  – размерность вектора весов для нейронов инициализируемого слоя

Были рассмотрены:

- Сигнальный метод Хебба
- Дифференциальный метод Хебба
- Алгоритм Кохонена подстройки весов синапсов от предыдущей итерации
- Нормализация входных образов, дальнейшая подстройка весов.

На основе рассмотренного выше метода строятся нейронные сети особого типа – так называемые самоорганизующиеся структуры – **self-organizing feature maps**.

Для них после выбора из слоя  $n$  нейрона  $j$  с минимальным расстоянием  $D_j$  (4) обучается по формуле (3) не только этот нейрон, но и его соседи, расположенные в окрестности  $R$ . Величина  $R$  на первых итерациях очень большая, так что обучаются все нейроны, но с течением времени она уменьшается до нуля. Таким образом, чем ближе конец обучения, тем точнее определяется группа нейронов, отвечающих каждому классу образов.