

# СЕТИ И КАРТЫ КОХОНЕНА

## 1. Сети Кохонена

Сети Кохонена (Kohonen T.) [1 – 3] относятся к *самоорганизующимся* нейронным сетям. Самоорганизующаяся сеть позволяет выявлять *кластеры* (группы) входных векторов, обладающих некоторыми общими свойствами. При этом выделяют *сети с неупорядоченными нейронами* (часто называемые *слоями Кохонена*) и *сети с упорядочением нейронов* (часто называемые *самоорганизующимися картами*, или *SOM – self-organizing map*). Карты Кохонена наглядно отражают на двумерной карте объекты с близкими свойствами.

Сеть Кохонена (рис. 1) – это однослойная сеть, каждый нейрон которой соединен со всеми компонентами  $n$ -мерного входного вектора. Входной вектор – это описание одного из объектов, подлежащих кластеризации. Количество нейронов совпадает с количеством кластеров, которое должна выделить сеть. В качестве нейронов сети Кохонена применяются линейные взвешенные сумматоры. Каждый  $j$ -ый нейрон описывается вектором весов  $\mathbf{w}_j = (w_{1j}, w_{2j}, \dots, w_{mj})$ , где  $m$  – число компонентов входных векторов. Входной вектор имеет вид  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{im})$ .

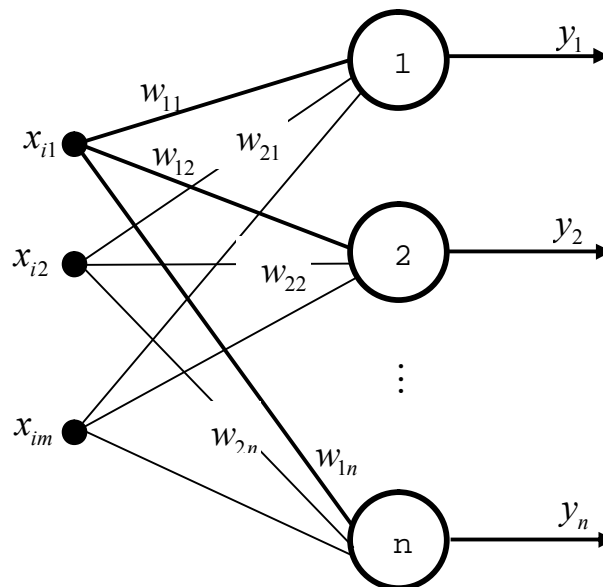


Рис. 1. Структура сети Кохонена

В сетях Кохонена используется обучение без учителя. Для обучения сети применяются *механизмы конкуренции*. При подаче на вход сети вектора  $\mathbf{x}$  побеждает тот нейрон, вектор весов которого в наименьшей степени отличаются от входного вектора. Для нейрона-победителя выполняется соотношение

$$d(\mathbf{x}, \mathbf{w}_j) = \min_{1 \leq i \leq n} d(\mathbf{x}, \mathbf{w}_i),$$

где  $n$  – количество нейронов,  $j$  – номер нейрона-победителя,  $d(\mathbf{x}, \mathbf{w})$  – расстояние (в смысле выбранной метрики) между векторами  $\mathbf{x}$  и  $\mathbf{w}$ .

Чаще всего в качестве меры расстояния используется *евклидова мера*

$$d(\mathbf{x}, \mathbf{w}_i) = \|\mathbf{x} - \mathbf{w}_i\| = \sqrt{\sum_{j=1}^n (x_j - w_{ij})^2}.$$

Используются и другие меры расстояния (метрики).

Вокруг нейрона-победителя образуется *окружение (neighborhood)*, или *радиус обучения (radius of learning)*. Радиус обучения определяет сколько нейронов кроме нейрона-победителя участвуют в обучении (т.е. корректируют свои веса) на данной итерации. Под радиусом в данном случае подразумевается расстояние в пространстве векторов весов нейронов. Т.е. любой нейрон, расстояние от вектора весов которого до вектора весов нейрона-победителя менее радиуса обучения, участвует в коррекции весов на данной итерации. Радиус обучения максимален на первой итерации и уменьшается с увеличением числа итераций таким образом, что в конце обучения корректирует свои веса только нейрон победитель.

Веса нейрона-победителя и всех нейронов, лежащих в пределах его радиуса обучения, подвергаются обучению (адаптации) по *правилу Кохонена*

$$\mathbf{w}_i^{(k+1)} = \mathbf{w}_i^{(k)} + \eta_i^{(k)} [\mathbf{x} - \mathbf{w}_i^{(k)}],$$

где  $\mathbf{x}$  – входной вектор,  $k$  – номер цикла обучения,  $\eta_i^{(k)}$  – *коэффициент скорости обучения*  $i$ -го нейрона из радиуса обучения в  $k$ -ом цикле обучения.

Веса нейронов, находящихся за пределами радиуса обучения не изменяются.

*Коэффициент скорости обучения*  $\eta_i^{(k)}$  разбивается на две части: функцию соседства  $\eta_i(d, k)$  и функции скорости обучения  $a(k)$

$$\eta_i^{(k)} = \eta_i(d, k) a(k). \quad (1)$$

В качестве *функции соседства* применяется или константа

$$\eta_i(d, k) = \begin{cases} \text{const}, & d_i \leq \sigma(k), \\ 0, & d_i > \sigma(k) \end{cases} \quad (2)$$

или Гауссова функция

$$\eta_i(d, k) = e^{-\frac{d_i}{2\sigma(k)}}.$$

При этом лучший результат получается при использовании Гауссовой функции расстояния. В (1) и (2)  $d_i$  – расстояние между  $i$ -м нейроном и нейроном-победителем. При этом  $\sigma(k)$  является убывающей функцией от номера цикла обучения. Наиболее часто используется функция, линейно убывающая от номера цикла обучения.

Рассмотрим теперь *функцию скорости обучения*  $a(k)$ . Эта функция также представляет собой функцию, убывающую от номера цикла обучения. Наиболее часто используются два варианта этой функции: линейная и обратно пропорциональная от номера цикла обучения вида

$$a(k) = \frac{A}{k + B},$$

где  $A$  и  $B$  это константы. Применение этой функции приводит к тому, что все векторы из обучающей выборки вносят примерно равный вклад в результат обучения.

*Обучение состоит из двух основных фаз:* на первоначальной фазе выбирается достаточно большое значение скорости обучения и радиуса обучения, что позволяет расположить векторы нейронов в соответствии с распределением примеров в выборке. На заключительной фазе производится точная подстройка весов, когда значения параметров скорости обучения много меньше начальных.

Обучение продолжается до тех пор, пока погрешность сети (*погрешность квантования*) при  $p$  входных векторах

$$E = \frac{1}{p} \sum_{i=1}^p \|\mathbf{x}_i - \mathbf{w}_j\|^2 \quad (3)$$

не станет малой величиной ( $\mathbf{w}_j$  – вектор весов нейрона-победителя).

При обучении сети Кохонена возникает проблема так называемых "*мертвых*" нейронов. Одно из ограничений всякого конкурирующего слоя состоит в том, что некоторые нейроны оказываются незадействованными. Это проявляется в том, что нейроны, имеющие начальные весовые векторы, значительно удаленные от векторов входа, никогда не выигрывают конкуренции, независимо от того, как долго продолжается обучение. В результате оказывается, что такие векторы не используются при обучении и соответствующие нейроны никогда не оказываются победителями. Такие "нейроны-неудачники" называют "*мертвыми*" нейронами, поскольку они не выполняют никакой полезной функции. Таким образом, входные данные будут интерпретироваться меньшим числом нейронов, а погрешность квантования (3) увеличивается. Поэтому надо дать шанс победить всем нейронам. Для этого алгоритм обучения модифицируют таким образом, чтобы нейрон-победитель терял активность. Одним из приемов учета активности нейронов является подсчет *потенциала*  $p_i$  каждого нейрона в процессе обучения [1]. Первоначально нейронам присваивается потенциал  $p_i(0) = \frac{1}{n}$ , где  $n$  – число нейронов (кластеров). В  $k$ -ом цикле обучения потенциал определяется по правилам:

$$p_i(k) = \begin{cases} p_i(k-1) + \frac{1}{n}, & i \neq j, \\ p_i(k-1) - p_{\min}, & i = j, \end{cases}$$

где  $j$  – номер нейрона-победителя.

Если значение потенциала  $p_i(k)$  падает ниже уровня  $p_{\min}$ , то нейрон исключается из рассмотрения – "отдыхает". При  $p_{\min} = 0$  нейроны не исключаются из борьбы. При  $p_{\min} = 1$  нейроны побеждают по очереди, так как в каждый цикл обучения только один из них готов к борьбе. На практике хороший результат получается при  $p_{\min} \approx 0.75$ .

Другой подход состоит в искусственном изменении расстояния между вектором весов и обучающим вектором. Например, можно к большому расстоянию добавлять положительное смещение, что позволяет нейрону стать конкурентным с нейроном-победителем.

В сети Кохонена входные значения желательно (хотя и не обязательно) *нормировать*. Для этого следует воспользоваться одной из следующих формул:

$$x_{ni} = \frac{x_i}{\sqrt{\sum_{i=1}^n x_i^2}},$$
$$x_{ni} = \frac{x_i}{|x_i|},$$

где  $x_{ni}$  – нормированный компонент входного вектора.

## 2. Карты Кохонена

*Карты Кохонена (самоорганизующиеся карты, или SOM – self-organizing map)* [2 – 3] предназначены для визуального представления *многомерных* свойств объектов на двумерной карте.

Карта Кохонена состоит из ячеек прямоугольной или шестиугольной (рис. 2) формы.

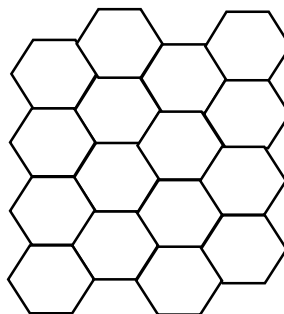


Рис. 2. Шестиугольные ячейки сети Кохонена

Каждой ячейке соответствует нейрон сети Кохонена. Обучение нейронов производится точно так же, как и нейронов сети Кохонена. Объекты, векторы признаков которых близки, попадают в одну ячейку или в ячейки, расположенные вблизи. Следовательно, *двумерная* карта Кохонена отражает на плоскости близость *многомерных* векторов признаков.

Ячейки, как мы уже отметили, могут быть прямоугольными или шестиугольными. Шестиугольные ячейки более корректно отображают расстояние между объектами на карте, т. к. для этих ячеек расстояние между центрами смежных ячеек одинаковы (рис. 3).

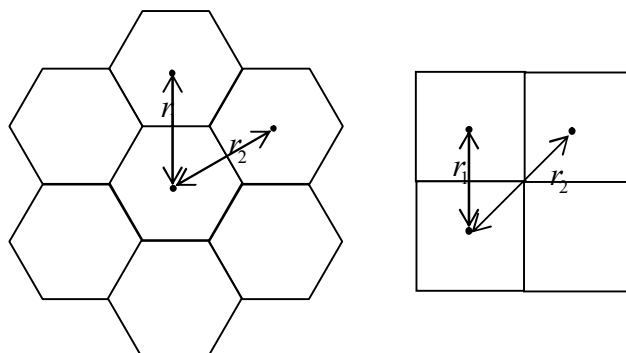


Рис. 3. Шестиугольные и прямоугольные ячейки

Поэтому чаще применяют шестиугольные ячейки.

Карта Кохонена отражает близость многомерных векторов признаков, то есть сходство объектов. Но обычно требуется анализировать, по каким конкретно параметрам проявляется сходство объектов. Для этого используется *раскраска карт Кохонена*. Для этого *строится столько карт, сколько параметров анализируется*. Каждая карта соответствует одному параметру объекта. Ячейки карты раскрашиваются в разные цвета (или оттенки серого цвета) в зависимости от значения параметров, соответствующих каждой ячейке. В каждую ячейку в общем случае попадает несколько объектов. Поэтому вычисляется или среднее значение параметра объектов каждой ячейки или минимальное или максимальное значение [4]. Если в ячейку не попал ни один объект (ячейке соответствует мертвый нейрон), то в качестве значения ячейки берется вес нейрона, соответствующий рассматриваемому параметру. Выделяются диапазоны значений параметра. Каждому диапазону ставится в соответствие цвет (или оттенок серого), и ячейки карты "раскрашиваются" соответствующими цветами.

Рассмотрим *пример* [5] с кредитованием физических лиц. С помощью самоорганизующихся карт Кохонена можно посмотреть зависимости между различными характеристиками заемщиков и выделить сегменты заемщиков, объединив их по схожим признакам. Результаты отображаются на картах. Каждому параметру заемщика соответствует своя карта (рис. 4). Карты построены в аналитической платформе Deductor.

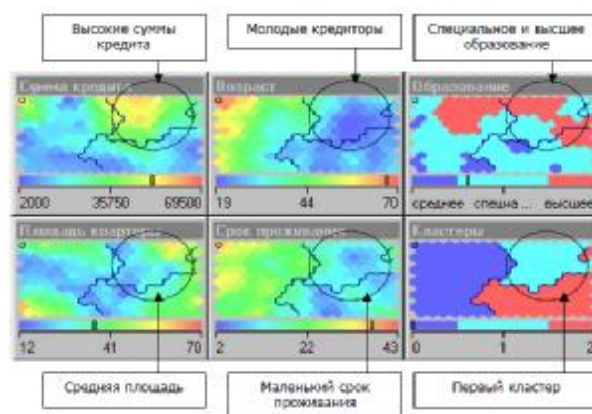


Рис. 4. Примеры карт Кохонена

Deductor, кроме того, строит кластеры. К примеру, в один и тот же кластер были сгруппированы молодые заемщики с маленьким сроком проживания в данной местности, специальным или средним образованием, средней жилплощадью, берущие высокие суммы кредита. В результате кластеризации заемщики со схожими характеристиками попадут в один кластер, и поэтому для них можно применять одинаковые правила выдачи кредита, т.е. для каждого кластера определить, стоит ли выдавать кредит его представителям.

## Литература

1. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М.: Финансы и статистика, 2002. – 244 с.
2. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1104 с.
3. Кохонен, Т. Самоорганизующиеся карты / Т. Кохонен. – М.: БИНОМ. Лаборатория знаний, 2008. – 655 с.
4. Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям. – СПб.: Питер, 2009. – 624 с.
5. Аналитическая платформа Deductor. Руководство аналитика. – BaseGroup Labs. – 2006. – 118 с.