

**УДК 681.322**

## **МОДЕЛИ ПОИСКА СТРУКТУР ДАННЫХ НА ОСНОВЕ КОНКУРЕНЦИИ И КООПЕРАЦИИ**

**Виноградов Г.П., Мальков А.А.**

*(Тверской государственной технической университет,*

*Тверь)*

*kja227@list.ru*

*Рассматриваются и анализируются методы поиска кластеров, основанные на конкуренции и кооперации нейронов.*

Ключевые слова: кластер, нейроны, самоорганизация, нейронные сети.

### **Введение**

В процессе самоорганизации нейронных сетей в соответствии с поданными входными сигналами осуществляется активизация нейронов так, что в результате изменения их синаптических весов происходит адаптация к поступающим обучающим выборкам. Возникает своего рода положительная обратная связь – для части нейронов, синаптические веса которых в большей степени соответствуют входным сигналам, их синаптические веса еще более возрастают. В результате происходит естественное расслоение нейронов на отдельные группы [1]. Отдельные нейроны или их группы сотрудничают между собой и активизируются в ответ на возбуждение, подаваемое конкретной обучающей выборкой. Одновременно происходит подавление других нейронов. В результате наблюдается конкуренция между отдельными нейронами или группами и сотрудничество их внутри группы. Конкуренция наблюдается также и между нейронами внутри группы [2].

Среди механизмов самоорганизации можно выделить два основных класса: самоорганизация, основанная на ассоциативном правиле Хебба, и механизм конкуренции между нейронами на базе обобщенного правила Кохонена.

Однако для всех способов обучения сети необходима избыточность обучающих данных, в противном случае обучение просто невозможно.

## **1. Теоретический анализ**

Пусть имеется непрерывное пространство входных сигналов (воздействий), которое обозначим через  $X$ . Это пространство характеризуется определенными метрическими связями векторов  $x \in X$ . Обозначим через  $A$  дискретное выходное пространство. Его топология определяется упорядоченным множеством нейронов вычислительных узлов решетки. Обозначим через  $\Phi$  нелинейное преобразование (которое называется картой признаков), выполняющее отображение входного пространства  $X$  в выходное пространство  $A$ :

$$(1) \quad \Phi: X \rightarrow A.$$

Алгоритм формирования самоорганизующейся сети начинается с инициализации синаптических весов сети. Обычно это происходит с помощью назначения синаптическим весам малых значений, сформированных генератором случайных чисел. Это позволяет обеспечить отсутствие какого-либо порядка признаков. После корректной инициализации сети для формирования карты самоорганизации запускаются три основных процесса: конкуренция, кооперация и синаптическая адаптация.

## **2. Процесс конкуренции**

Пусть  $m$  – размерность входного пространства, входной вектор выбирается из входного пространства случайно и обозначается так:

$$(2) \quad x = [x_1, x_2, \dots, x_m]^T.$$

Вектор синаптических весов каждого из нейронов сети имеет ту же размерность, что и входное пространство. Обозначим синаптический вес нейрона  $j$  следующим образом:

$$(3) \quad w_j = [w_{j1}, w_{j2}, \dots, w_{jm}]^T, \quad j = \overline{1, L},$$

где  $L$  – общее количество нейронов в сети.

Для того, чтобы подобрать наилучший вектор  $w_j$ , соответствующий входному вектору  $x$ , сравнить скалярные произведения  $w_j^T \cdot x$ ,  $j=1, \dots, L$  и выбрать наибольшее значение. При этом предполагается, что ко всем нейронам применяется некоторое значение насыщения. Таким образом, выбирая нейрон с наибольшим значением произведения  $w_j^T \cdot x$ , определяется нейрон, местоположение которого должно стать центром топологической окрестности возбужденного нейрона.

Отметим, что наилучший критерий соответствия, основанный на максимизации скалярного произведения  $w_j^T \cdot x$ , математически эквивалентен минимизации Евклидова расстояния между векторами  $x$  и  $w_j$ . Если использовать индекс  $i(x)$  для идентификации этого нейрона, который лучше всего соответствует входному сигналу  $x$ , то эту величину можно определить с помощью следующего соотношения

$$(4) \quad i(x) = \arg \min_j \|x - w_j\|, \quad j = \overline{1, L}.$$

Нейрон с номером  $i(x)$ , удовлетворяющий условию (4) называется нейроном-победителем для данного входного вектора  $x$ .

За счет процесса конкуренции между нейронами непрерывное входное пространство входных сигналов активизации нейронов отображается в дискретное выходное пространство.

## 2. Процесс кооперации

Нейрон-победитель находится в центре топологической окрестности сотрудничающих нейронов. Будем считать, что нейрон-победитель всегда пытается возбудить пространство близ-

ких к нему нейронов. Причем топологическая окрестность победившего нейрона  $j$  плавно сходит на нет с увеличением расстояния. Обозначим через  $h_{i,j}$  топологическую окрестность с центром в победившем нейроне  $I$ , состоящую из множества возбуждаемых (кооперирующихся) нейронов, типичный представитель которого имеет индекс  $j$ . Пусть  $d_{i,j}$  – латеральное расстояние между победившим ( $i$ ) и ( $j$ ) вторично возбужденным нейронами. Тогда можно предположить, что топологическая окрестность  $h_{i,j}$  является унимодальной функцией латерального расстояния  $d_{i,j}$  и представляется в виде функции Гаусса:

$$(5) \quad h_{j,i} = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2}\right).$$

Параметр  $\sigma$  называется эффективной шириной топологической окрестности. Этот параметр определяет уровень, до которого нейрон из топологической окрестности нейрона победителя участвует в процессе обучения. Хотя можно рассматривать и другие формы топологической окрестности, но гауссова топологическая окрестность является наиболее подходящей с биологической точки зрения, чем, например, прямоугольная.

Для обеспечения кооперации между соседними нейронами необходимо, чтобы топологическая окрестность  $h_{i,j}$  была зависимой от латерального расстояния между нейроном-победителем ( $i$ ) и возбужденным нейроном ( $j$ ) в выходном пространстве, а не от какой-либо меры длины в исходном входном пространстве, что и отражено в (5). В случае одномерной решетки расстояние  $d_{i,j}$  является целым числом, равным модулю  $|i-j|$ . В случае двумерной решетки это расстояние определяется выражением

$$(6) \quad d_{j,i}^2 = \|r_j - r_i\|^2,$$

где дискретный вектор  $r_j$  определяет позицию возбуждаемого нейрона  $j$ , а  $r_i$  – победившего нейрона  $i$ . Оба этих измерения проводятся в дискретном выходном пространстве.

Для карт самоорганизации одним из условий является уменьшение с течением времени размера топологической окрестности. Это требование удовлетворяется за счет постепенного уменьшения ширины  $\sigma$  функции топологической окрестности  $h_{i,j}$ . Популярным вариантом зависимости величины  $\sigma$  от дискретного времени  $n$  является экспоненциальное убывание:

$$(7) \quad \sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right), \dots n = 0, 1, 2, \dots,$$

где  $\sigma_0$  – начальное значение величины  $\sigma$ ;  $\tau_1$  – заданная временная константа. Отсюда, топологическая окрестность будет иметь форму, зависящую от времени, то есть:

$$(8) \quad h_{j,i}(n) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2(n)}\right), \quad n = 0, 1, 2, \dots,$$

где  $\sigma(n)$  определяется по формуле (7). Таким образом, при увеличении количества итераций  $n$  ширина  $\sigma(n)$  экспоненциально убывает, а вместе с ней соответственно сжимается и топологическая окрестность. Будем называть  $h_{i,j}(n)$  функцией окрестности.

### 3. Процесс адаптации

Для того, чтобы сеть могла саморганизовываться, вектор синаптических весов  $w_j$  нейрона  $j$  должен изменяться в соответствии с входным вектором  $x$ . В постулате Хебба синаптический вес усиливается при одновременном возникновении пресинаптической и постсинаптической активности. Но в случае обучения без учителя изменение в связях происходят только в одном направлении, что, в конечном счете, приводит все веса к точке насыщения. Для того чтобы обойти эту проблему, немного изменим гипотезу Хебба и введем в нее слагаемое забывания  $g(y_j)w_j$ , где  $w_j$  – вектор синаптических весов нейрона  $j$ ;  $g(y_j)$  – некоторая положительная скалярная функция отклика  $y_j$ . Единственным требованием, налагаемым на функцию  $g(y_j)$  является равенство нулю постоянного слагаемого разложения в ряд Тей-

лора функции  $g(y_j)$ , что в свою очередь влечет выполнение соотношения:

$$(9) \quad g(y_j) = 0 \text{ при } y_j = 0.$$

Имея такую функцию, изменение вектора весов нейрона  $j$  в решетке можно выразить следующим образом:

$$(10) \quad \Delta w_j = \eta y_j \cdot x - g(y_j) w_j,$$

где  $\eta$  – параметр скорости обучения алгоритма. Первое слагаемое в правой части (10) является слагаемым Хебба, а второе – слагаемым забывания. Для того, чтобы выполнялось условие (9), выберем следующую линейную функцию  $g(y_j)$ :

$$(11) \quad g(y_j) = \eta y_j.$$

Выражение (10) можно упростить, если принять, что

$$(12) \quad y_j = h_{j,i}(x)$$

Тогда подставляя (11-12) в (10) получим:

$$(13) \quad \Delta w_j = \eta h_{j,i}(x) (x - w_j).$$

Учитывая формализацию дискретного времени, для данного вектора синаптических весов  $w_j(n)$  в момент времени  $n$  обновленный вектор  $w_j(n+1)$  в момент времени  $n+1$  можно определить в следующем виде:

$$(14) \quad w_j(n+1) = w_j(n) + \eta(n) h_{j,i}(x) (x - w_j(n)).$$

Это выражение применяется ко всем нейронам решетки, которые лежат в топологической окрестности победившего нейрона  $i$ . Выражение (14) имеет эффект перемещения вектора синаптических весов  $w_j$  победившего нейрона  $I$  в сторону входного вектора  $x$ . При периодическом представлении данных обучения благодаря коррекции в окрестности победившего нейрона векторы синаптических весов будут стремиться следовать распределению входных векторов. Таким образом, представленный алгоритм ведет к топологическому упорядочиванию пространства признаков в выходном пространстве так, что нейроны, кор-

ректируемые в решетке, будут стремиться иметь одинаковые синаптические веса.

Выражение (14) является формулой вычисления синаптических весов карты признаков. В дополнении к этому уравнению для выбора функции окрестности  $h_{j,i(x)}(n)$  требуется учитывать эвристику (8).

Параметр  $\eta(n)$  – скорость обучения также должен зависеть от времени, как это должно быть при стохастической аппроксимации. К примеру, можно начать с некоторого исходного значения  $\eta_0$ , а затем с течением времени постепенно уменьшать его. Это требование можно выполнить, выбрав для параметра интенсивности обучения следующую экспоненциальную функцию:

$$(15) \quad \eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right), \quad n = 0, 1, 2, \dots,$$

где  $\tau_2$  – временная константа алгоритма самоорганизации. Формулы экспоненциального убывания параметров скорости обучения (15) и ширины функции окрестности (8) могут не являться оптимальными. Они просто адекватны процессу самоорганизующегося формирования карты признаков.

#### **4. Описание алгоритма**

Суть алгоритма самоорганизации состоит в простом геометрическом вычислении свойств Хеббоподобного правила обучения и латеральных взаимодействий. Существенными характеристиками этого алгоритма являются следующие.

Непрерывное входное пространство образов активации, которое генерируется в соответствии с некоторым распределением вероятности.

Топология сети в форме решетки, состоящей из нейронов. Она определяет дискретное выходное пространство.

Зависящая от времени функция окрестности  $h_{j,i(x)}(n)$ , которая определена в окрестности нейрона-победителя  $i(x)$ .

Параметр скорости обучения  $\eta(n)$ , для которого задается начальное значение  $\eta_0$  и который постепенно убывает во времени  $n$ , но никогда не достигает нуля.

Для функции окрестности и параметра скорости обучения на этапе упорядочения (то есть приблизительно для первой тысячи итераций) можно использовать формулы (8) и (14) соответственно. Для хорошей статистической точности на этапе сходимости параметр  $\eta(n)$  должен быть установлен в очень малое значение (0,01 или меньше). Функция окрестности в начале этапа сходимости должна содержать только ближайших соседей нейрона-победителя.

Алгоритм проходит четыре основных шага: *инициализация*, *подвыборка*, *поиск максимального соответствия* и *корректировка*. Кратко весь алгоритм можно описать следующим образом:

*Инициализация.* Для исходных векторов семантических весов  $w_j(0)$  выбираются случайные значения. Единственным требованием здесь является различие векторов для разных значений  $j=1, \dots, L$ , где  $L$  – общее количество нейронов в решетке.

*Подвыборка.* Выбираем вектор  $x$  из входного пространства с определенной вероятностью. Размерность вектора равна  $m$ .

*Поиск максимального подобия.* Находим наиболее подходящий (победивший) нейрон  $i(x)$  на шаге  $n$ , используя критерий минимума Евклидова расстояния:

$$(16) \quad i(x) = \underset{j}{\operatorname{arg\,min}} \|x - w_j\|, \quad j = \overline{1, L}.$$

*Коррекция.* Корректируем векторы семантических весов всех активных нейронов, используя формулу:

$$(17) \quad w_{j,i(x)}(n+1) = w_{j,i(x)}(n) + \eta(n) h_{j,i(x)}(n) (x - w_{j,i(x)}(n)),$$

где  $\eta(n)$  – параметр скорости сходимости;  $h_{j,i(x)}(n)$  – функция окрестности в победившем нейроне  $i(x)$ . Оба этих параметра динамически изменяются:

$$(18) \eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right), \quad n = 0, 1, 2, \dots;$$

$$(19) h_{j,i}(n) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2(n)}\right), \quad n = 0, 1, 2, \dots;$$

$$(20) \sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right), \dots n = 0, 1, 2, \dots$$

во время обучения с целью получения лучшего результата.

*Продолжение.* Возвращаемся к шагу 2 и продолжаем вычисления до тех пор, пока в карте признаков не перестанут происходить заметные изменения.

## 5. Экспериментальная часть

Работа описанного выше алгоритма проверялась на ряде тестовых примеров. В таблице 1 приведены центры и разбросы для распределения точек для примеров 1, 2. координаты генерировались случайным образом.

Таблица 1

Координаты центров		Разбросы точек			
		Пример 1		Пример 2	
х	у	х	у	х	у
0,3	0,6	±0,15	±0,15	±0,5	±0,2
0,3	0	±0,15	±0,4	±0,15	±0,4
0,3	-0,6	±0,15	±0,15	±0,5	±0,2
-0,3	0	±0,15	±0,15	±0,4	±0,3

На рис. 1 изображены результаты работы алгоритма конкуренции на первом тестовом примере. Квадратами указаны «истинные» центры, а ромбами – центры, найденные алгоритмом

конкуренции. Крестики указывают первоначальное расположение весов нейронов.

Координаты найденных центров приведены в таблице 2.

Таблица 2

Координаты найденных центров			
Пример 1		Пример 2	
0,3100	0,6100	-0,0830	-0,0166
0,3355	0,1080	0,3681	-0,5791
0,3075	-0,5841	-0,1213	-0,6094
-0,2953	0,0180	0,2450	0,5249
		-0,4352	0,0061
		-0,2127	0,6182

### Уточнение координат центров Алгоритм конкуренции

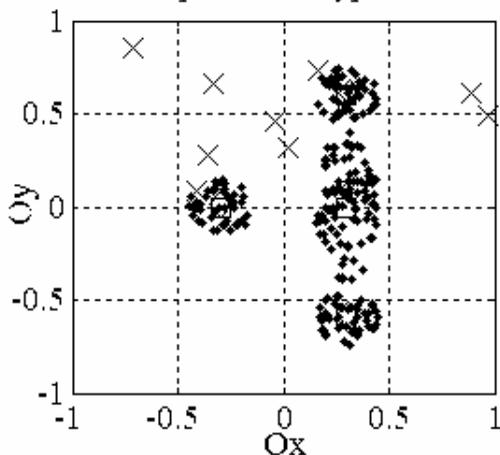
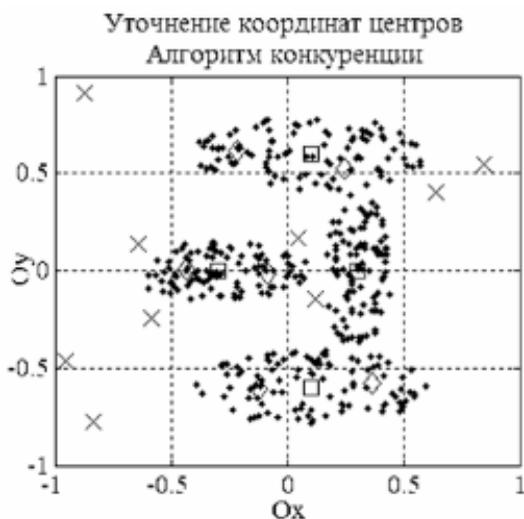


Рис.1. Определение центров для тестового примера 1

Результат сравнивался с результатом работы алгоритма нечеткой самоорганизации. Результаты оказались практически одинаковы – разница получилась около 1%.

На рис. 2 изображены результаты работы алгоритма конкуренции на втором тестовом примере. Здесь «истинные» центры не были смещены по отношению к примеру 1, а разбросы точек были увеличены. Из рис. 2 видно, что алгоритм конкуренции нашел 6 центров кластеризации. Найденные центры расположены таким образом, что они «притягивают» к себе точки разных частей одних и тех же сгущений точек. Кроме того, из рис. 2 видно, что 3 из найденных центров «описывают» «вертикальный» кластер, притягивая к себе определенную часть точек этого кластера. На этом же примере алгоритм нечеткой самоорганизации определил большее количество кластеров, один из которых оказался ложным.



*Рис.2. Определение центров для тестового примера 2*

Таким образом, алгоритм конкуренции показал эффективность, особенно в ситуации, когда кластеры имеют общие точки.

## **Литература**

1. Борисов, В.В. *Нечеткие модели и сети* / В.В. Борисов, В.В. Круглов, А.С. Федулов. М.: Горячая линия – Телеком, 2007. 284 с.
2. Kosko, B. *Differential hebbian learning* / B. Kosko // AIP Conference Proceedings. Vol. 151. 1986. P. 265 – 270.