УДК 517.9

Нейронные сети и карты Кохонена

© Р. Ф. Миникаев[1], З. Я. Якупов[2]

Аннотация. В работе проведено исследование карт Кохонена и выявлены процессы самоорганизации. Получены оптимальные параметры алгоритма самоорганизующихся карт для кластеризации сетевых соединений и построено приложение, реализующее данный алгоритм.

Ключевые слова: нейронные сети, карты Кохонена, векторное квантование.

1. Введение

Самоорганизующаяся карта (СОК) является новым, эффективным программным инструментом для визуализации многомерных данных. В своём основном варианте СОК создаёт граф подобия входных данных. Она преобразует нелинейные статистические соотношения между многомерными данными в простые геометрические связи между изображающими их точками на устройстве отображения низкой размерности, обычно в виде регулярной двумерной сетки узлов. Поскольку СОК осуществляет сжатие информации с сохранением в получаемом изображении наиболее важных топологических и/или метрических связей между первичными элементами данных, можно также считать, что с её помощью порождаются абстракции (обобщения) некоторого вида. Эти два характерных свойства СОК, визуализацию и обобщение, можно использовать различными способами в решении сложных задач таких, как анализ процессов, машинное восприятие, управление, передача информации. В той форме, в которой она существует в настоящее время, СОК была задумана Т. Кохоненом в 1982 г.

2. Инициализация

Изначально, самоорганизующаяся карта представляет собой сетку (прямоугольную или гексагональную) из узлов, соединённых между собой связями. Также определяется количество нейронов в сети.

Каждый из узлов описывается двумя векторами. Первый - вектор веса m_i, имеющий такую же размерность, что и входные данные. Второй - координаты узла на карте, далее вектор r_i. Перед началом обучения карты необходимо проинициализировать весовые коэффициенты нейронов.

Случайная инициализация. Причина использования случайных начальных значений в приводимых демонстрационных примерах состояла в том, что алгоритмы СОК можно инициализировать с помощью произвольных значений кодирующих векторов $m_i(0)$. Другими словами, было показано, что начальные неупорядоченные векторы будут упорядочены, если процесс обучения достаточно длителен; в обычных приложениях

[1]Студент, Казанский государственный технический университет имени А. Н. Туполева, г. Казань; minikaev@gmail.com.

[2]Доцент, Казанский государственный технический университет имени А. Н. Туполева, г. Казань; zymat@bk.ru.

это происходит за несколько сотен шагов. Это не означает, однако, что случайная инициализация обеспечивает наилучшее или наискорейшее обучение и что ее следует использовать на практике.

Линейная инициализация. Поскольку векторы $m_i(0)$ могут быть произвольными, можно предположить, что полезным будет любое упорядоченное начальное состояние, даже если значения, составляющие его, не обеспечивают адекватного представления для $p(x)$ - плотности распределения вероятностей. В методе, который с успехом использовался, для автокорреляционной матрицы, соответствующей вектору x, вначале определялись два собственных вектора с наибольшими собственными значениями. Затем в двумерном линейном подпространстве определялся прямоугольный массив (с прямоугольной или гексагональной регулярной решеткой), центр тяжести которого совпадал со средним значением входных параметров $x(t)$, два его измерения - с найденными двумя наибольшими собственными значениями автокорреляционной матрицы. Точки данного массива служили начальными значениями $m_i(0)$. Если требуется получить в СОК приблизительно равномерную пространственную решетку, то количества элементов в горизонтальном и вертикальном направлениях решетки должны быть соответственно пропорциональны двум наибольшим собственным значениям, о которых говорилось выше.

Так как теперь $m_i(0)$ уже упорядочены и задаваемая ими точечная плотность примерно приближает функцию $p(x)$, можно начинать обучение непосредственно с фазы сходимости. При этом для достижения состояния равновесия можно с самого начала задавать значения для $\alpha(t)$ значительно меньшими единицы и использовать функцию соседства, ширина которой близка к соответствующему значению этого параметра на завершающих итерациях процесса.

На каждом шаге обучения из исходного набора данных случайно выбирается один из векторов

$$x = \Big\{\xi_1, \xi_2, \ldots, \xi_n\Big\}^T \in \mathbb{R}^n,$$

где $n \in \mathcal{N}$, а затем производится поиск наиболее похожего на него вектора коэффициентов нейронов. При этом выбирается нейрон-победитель, который наиболее похож на вектор входов, иными словами, определяется расстояние между векторами, которое обычно вычисляется в евклидовом пространстве. Если обозначим нейрон-победитель символом c, то получим:

$$||x - m_c|| = \min\Big\{||x - m_i||\Big\}$$

или

$$c = \arg\min\Big\{d(x, m_i)\Big\},$$

т.е. c - номер элемента, для которого расстояние до x минимально. В случае, если таких победителей больше одного, то случайным образом выбирается единственный нейрон-победитель.

После того, как найден нейрон-победитель, производится корректировка весов нейросети. При этом вектор, описывающий нейрон-победитель, и векторы, описывающие его соседей в сетке, перемещаются в направлении входного вектора по формуле:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)],$$

где $h_{ci}(t)$ - функция соседства.

3. Процесс самоорганизации

Начать процесс самоорганизации, используя «широкую» функцию соседства, можно двумя способами. Во-первых, взяв большой радиус множества соседства $N_c(0)$. Во-вторых, среднеквадратичное отклонение функции h_{ci} может быть того же порядка, что и половина наибольшей протяженности массива. В обоих случаях обычно нет риска завершить процесс обучения в одной из «метастабильных» конфигураций карты (в которой средняя ожидаемая мера искажения или средняя ожидаемая ошибка квантования соответствует локальному минимуму вместо глобального). Однако, с инвариантной по времени функции соседства ситуация может оказаться совершенно другой, особенно если функция соседства будет «узкой».

В работе [1] проанализированы «метастабильные состояния» для случая одномерного массива. Здесь вначале функция соседства определяется как выпуклая на некотором множестве $I \equiv \Big\{0, 1, 2, \ldots, N\Big\}$, если из условий

$$|s - q| > |s - r|$$

и

$$|s - q| > |r - q|$$

следует, что

$$[h(s, s) + h(s, q)] < [h(s, r) + h(r, q)]$$

для всех $s, r, q \in \mathcal{I}$. В противном случае функция соседства определялась как вогнутая.

Основной полученный результат состоит в том, что если функция соседства является выпуклой, то устойчивых состояний, отличных от упорядоченных, не существует. Если функция соседства вогнутая, существуют метастабильные состояния, которые могут резко замедлить процесс упорядочения. Если, однако, в начале процесса обучения функция соседства выпукла, примерно как функция Гаусса $h_{ci}(0)$ в своей средней части при большом среднеквадратическом отклонении, упорядочение может быть достигнуто почти наверное; после завершения упорядочения функция соседства может быть сужена для более точной аппроксимации функции $p(x)$.

Условия упорядочения в общем случае будут наиболее строгими, если пространство входных сигналов имеет ту же размерность, что и решетка нейронов; на практике, однако, размерность пространства входных сигналов обычно много больше размерности решетки, что облегчает выполнение упорядочения.

Циклический процесс обучения, перебирающий входные данные, заканчивается по достижении картой допустимой (заранее заданной аналитиком) погрешности, или по совершении заданного количества итераций. Например, вычисление ошибки карты можно рассчитать как среднее арифметическое расстояний между наблюдениями и векторами весов соответствующего им нейрона - победителя:

$$\frac{1}{N}\sum_{i=1}^{N} ||x_i - m_c||,$$

где N - количество элементов набора входных данных.

Для визуализации структуры кластеров, полученных в результате обучения карты, применяется унифицированная матрица расстояний. Элементы матрицы определяют расстояние между весовыми коэффициентами каждого нейрона и его ближайшими соседями. Затем эти значения используются для определения цвета, которым узел будет

отрисован. При таком использовании узлам с большим расстоянием между ними и соседями соответствует чёрный цвет, а близлежащим узлам - белый.

Следует подчеркнуть, что нет никаких теоретических оснований, согласно которым рассматриваемый рекурсивный алгоритм основного варианта СОК должен определяться какой-то целевой функцией E, описывающей, например, среднее ожидаемое значение меры искажения. Есть только следующие факты:

1. Обычная оптимизация функции E по методу стохастической аппроксимации приводит к основному алгоритму СОК.

2. Эвристически полученный основной алгоритм СОК характеризует непараметрическую регрессию, что часто позволяет отразить важные и интересные топологические связи между кластерами первичных данных.

Чтобы показать, какую в действительности роль играет в рассматриваемом контексте формализм «функции энергии», требуется проделать более глубокий анализ среднего ожидаемого значения меры искажения [2].

Индекс c «победителя» представляет собой разрывную функцию от x и всех m_i. Значимость этого индекса можно увидеть более ясно, если интеграл E в формуле

$$E = \int ep(x)dx = \int \sum_{i \in L} h_{ci} d(x, m_i) p(x) dx,$$

при

$$d(x, m_i) = ||x - m_i||^2$$

представить как сумму интегралов, взятых по тем областям X_i, для векторов x, для которых ближайшим эталонным вектором будет m_i (т. е. по фрагментам мозаики Вороного, см. рис. (3.1)):

$$E = \sum_i \int\limits_{x \in X_j} \sum_k h_{ik} ||x - m_k||^2 p(x) dx, \tag{3.1}$$

При вычислении истинного (глобального) градиента функции E по произвольному m_j необходимо принимать во внимание члены двух различных видов: для первого из них подынтегральное выражение будет варьируемым, а пределы интегрирования постоянны, для второго - подынтегральное выражение остается неизменным, но пределы интегрирования варьируются (вследствие изменения m_j). Обозначим эти слагаемые, соответственно, как G и H:

$$\nabla_{m_j} E = G + H. \tag{3.2}$$

Не трудно показать, что

$$G = -2 \sum_i \int\limits_{x \in X_j} h_{ij}(x - m_j) p(x) dx = -2 \int\limits_{\cup_j X_j} h_{cj}(x - m_j) p(x) dx. \tag{3.3}$$

В классическом методе векторного квантования $H = 0$ вследствие того, что $h_{ck} = \delta_{ck}$. Тогда, при пересечении границы между X_i и X_j в мозаике Вороного, члены $||x - m_i||^2$ и $||x - m_j||^2$ остаются равными. Для общего случая функции h_{ck} вычисление H представляет собой весьма трудоемкую задачу, так как подынтегральные выражения сильно меняются при пересечении границ мозаики Вороного.

Для начала воспользуемся тем фактом, что при варьировании m_j сдвигаются только границы мозаики, выделяющие X_j.

Рассмотрим рис. (3.1), где дается схема выделения области X_j, а также сдвиг ее границ, вызванный отклонением dm_j. Первое изменение $\nabla_{m_j} E$ для расстояния $||x - m_k||^2$, вычисленного по топологической окрестности элемента m_j, получается интегрированием по заштрихованному участку (дифференциальному элементу гиперобъема). Поскольку все границы здесь представляют собой сегменты срединных плоскостей (гиперплоскостей) между соседними элементами m_i, представляется, что можно хотя бы приблизительно оценить своего рода «средний сдвиг».

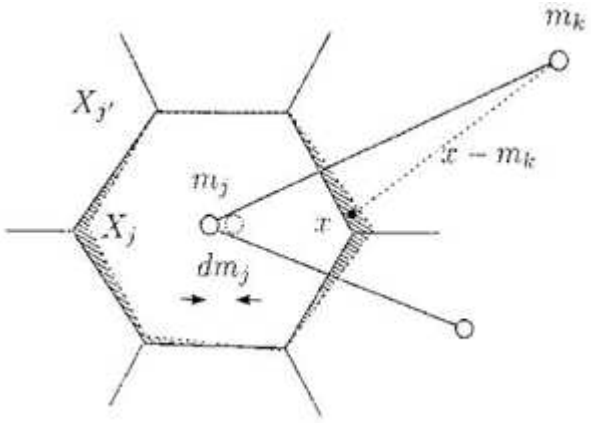

Рисунок **3.1**

Пример граничного эффекта при варьировании m_j.

По существу, если бы вокруг m_j было так много соседей, что X_j можно было бы аппроксимировать гиперсферой, то при изменении m_j на величину dm_j окрестность X_j сохранила бы свою форму и сдвинулась на расстояние $(1/2)dm_j$. В общем же случае при произвольной размерности пространства значений x и произвольном расположении m_i форма X_j должна изменяться, однако упрощенная и осредненная аппроксимация состоит в том, что изменение формы не учитывается, по крайней мере, в первом приближении, а X_j только сдвигается на $(1/2)dm_j$. Еще одно допущение, позволяющее существенно упростить обсуждаемую задачу, состоит в том, чтобы считать функцию $p(x)$ постоянной на области X_j . Это допущение будет вполне справедливым в случае, когда для аппроксимации $p(x)$ используется значительное число элементов m_j. Если принять оба указанных упрощающих предположения, то вклад в H, получаемый за счет варьирования X_j, обозначаемый H_1, будет приближенно равен разности двух интегралов: интеграла по деформированной области X_j и интеграла по недеформированной области X_j. Это равно интегралу от суммы

$$\sum h_{jk} \left\| x - \left(m_k - \frac{1}{2} dm_j \right) \right\|^2,$$

умноженной на $p(x)$, минус интеграл от суммы

$$\sum h_{jk} ||x - m_k||^2,$$

умноженной на $p(x)$. Представляется, что опять получен случай, где каждый из элементов m_k изменяется на некоторую величину $-(1/2)dm_j$. Следовательно, данное различие может также быть выражено как сумма градиентов интеграла от суммы

$$\sum h_{jk} \left\| x - \left(m_k - \frac{1}{2} dm_j \right) \right\|^2,$$

умноженной на $p(x)$, найденных по каждому из m_k и умноженных на $-(1/2)dm_j$:

$$H_1 = -\frac{1}{2}(-2)\int\limits_{x\in X_j}\sum_{k\neq j} h_{ik}(x-m_k)p(x)dx = \int\limits_{x\in X_c}\sum_{k\neq c} h_{ck}(x-m_k)p(x)dx. \qquad (3.4)$$

Случай $k = c$ в формуле (3.4) можно исключить по той причине, что это слагаемое соответствует основному алгоритму векторного квантования и его вклад, как показано в работе [3], равен здесь нулю.

Не следует забывать, что E представляет собой сумму по всем i, и поэтому второй главный вклад в H, обозначаемый H_2, связан с интегралом по всем частным дифференциальным элементам гиперобъема (заштрихованные сегменты на рис. (3.1)), граничащим с $X_{j'}$, где j' - индекс любой области, граничащей с X_j. Подсчитать H_2 по-видимому еще сложнее, чем получить аппроксимацию для H_1. Есть, однако, веские основания полагать, что в среднем

$$||H_2|| < ||H_1||,$$

поскольку каждая из областей интегрирования в H_2 есть лишь один сегмент дифференциала X_j, а величины $x - m_k$ различны в каждой из этих подобластей. При изучении порядка величины дополнительных коррекций представляется более интересным уделить основное внимание численному анализу слагаемого H_1, которому, кроме всего прочего, можно дать также и наглядную интерпретацию.

Пытаясь вывести результаты, получаемые в рекурсивном пошаговом спуске в «E - ландшафте», можно заметить, что в выражении для G вектор x пробегает все пространство своих значений, а на величину градиента $\nabla_{m_j}E$ влияют все те области X_c, для которых элемент m_j есть топологический сосед в сети. Напротив, в H_1 интеграл берется только по X_j, тогда как подынтегральная функция содержит члены, зависящие от топологических соседей m_k для m_j. По аналогии с соотношениями [4] можно записать (пренебрегая членами, связанными с H_2):

$$m_c(t+1) = m_c(t) + \alpha(t)\Big\{h_{cc}[x(t) - m_c(t)] - \frac{1}{2}\sum_{k\neq c} h_{ck}[x(t) - m_k(t)]\Big\}, \top m_i(t+1) =$$
$$= m_i(t) + \alpha(t)h_{ci}[x(t) - m_i(t)] \qquad (3.5)$$

для $i \neq c$. Напомним, что член

$$-\frac{1}{2}\sum_k h_{ck}[x(t) - m_k(t)]$$

в данном выражении был получен в предположении, что $p(x)$ сохраняет постоянное значение на каждом фрагменте мозаики Вороного. Для узлов, на краях которых мозаики соответствующие области X_i становятся бесконечными, такая аппроксимация уже не будет корректной. Поэтому здесь также появятся граничные эффекты, несколько отличающиеся от тех, что имели место в основном варианте СОК.

Прежде, чем перейти к изложению результатов численных экспериментов, дадим интерпретацию дополнительным членам, обусловленным интегралом H_1.

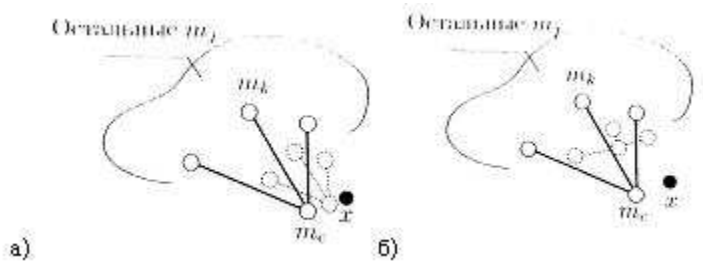

Рисунок **3.2**

Различия в выполняемых корректировках: (a) для базовой СОК, (b) для процесса, задаваемого формулами (3.5)

Рассмотрим рис. (3.2), иллюстрирующий особенности работы старого и нового алгоритма. Предположим, что на некотором шаге процесса обучения значение одного из векторов m_i получилось существенно отличающимся от значений остальных параметрических векторов. Когда вектор X окажется ближайшим именно к нему, в старом алгоритме полагается $m_i = m_c$ и его топологические соседи m_k должны быть сдвинуты по направлению к X. В новом алгоритме по направлению к X сдвигается только $m_k \neq m_c$, тогда как m_c перемещается в противоположном направлении, в сторону центра области соседства элемента m_k.

Численный эксперимент проводился с целью изучения свойств сходимости процесса, определяемого соотношениями (3.5). В этом эксперименте использовался квадратный массив узлов с двумерными векторами x и m_i. Применялись оба алгоритма, как старый, так и определяемый формулой (3.5), с одинаковыми начальными состояниями, значениями $\alpha(t)$ и используемой последовательностью случайных чисел. В качестве начальных векторов брались независимые случайные величины, а значения уменьшались линейно от 0.5 до 0. Ядро функции соседства h_{ck} было принято постоянным для всех узлов в пределах одного шага решетки от узла c в горизонтальном и вертикальном направлениях и $h_{ck} = 0$ в остальных случаях. (Следует напомнить, что в практических приложениях для ускорения сходимости ядра обычно определяются так, чтобы они «сжимались» во времени.) На рис. (3.3) для квадратной решетки показаны значения m_i после 8000 шагов процесса обучения. Функция $p(x)$ имеет постоянное значение внутри квадратной области, ограниченной рамкой, и нулевое значение вне ее. Из многочисленных экспериментов подобного рода (один из которых и показан на (3.3)) следуют два вывода.

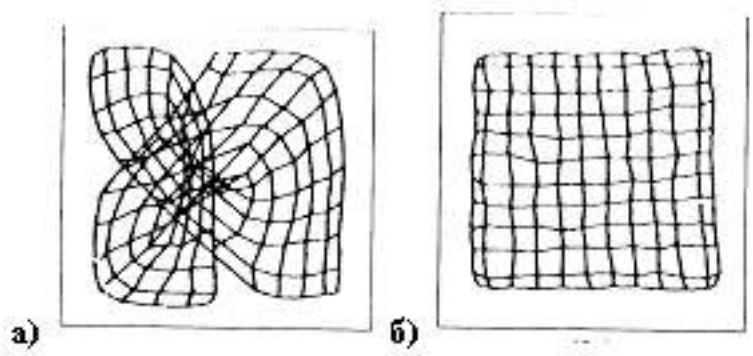

Рисунок **3.3**

Двумерная карта после 8000 шагов: (a) основная СОК с небольшой областью сходства, (b) та же самая область соседства, но в результате использования алгоритма, определяемого соотношениями (3.5).

1. Новый алгоритм упорядочивает несколько быстрее и надежнее; после 8000 шагов (с узкими постоянными ядрами h_{ck}) ни одно из состояний, получаемых с помощью старого алгоритма, еще не было полностью упорядочено, тогда как для нового алгоритма около половины состояний уже стали упорядоченными.

2. Граничные эффекты в новом алгоритме выражены сильнее.

Частный случай. Если плотность распределения вероятностей $p(x)$ входных данных принимает дискретные значения, как это, например, имеет место в известной задаче коммивояжера, исходный алгоритм СОК может быть получен из средней ожидаемой меры искажения [5]. В свете проведенного выше обсуждения это вполне понятный результат, поскольку значения $p(x)$ на границах мозаики Вороного тогда равны нулю и дополнительный член H, обусловленный варьированием пределов интегрирования, исчезает.

4. Модификация определения понятия «победитель»

Доказано [6], [7], что если критерий отбора в определении «победителя» *с* модифицировать следующим образом

$$\sum_i h_{ci}||x - m_i||^2 = \min_j \Big\{ \sum_j h_{ji}||x - m_i||^2 \Big\}. \tag{4.1}$$

где h_{ci} - та же самая функция соседства, которая использовалась в процессе обучения, то средняя ожидаемая мера искажения становится функцией энергии или потенциала, минимизация которой может выполняться с помощью обычного метода градиентного спуска. Хотя это наблюдение и представляет определенный математический интерес, следующие факты несколько обесценивают его значение в нейросетевом моделировании.

1. В физиологическом толковании СОК функция соседства h_{ck} определяет только управляющее воздействие на синаптическую пластичность, но не прямое управление сетевой активностью в функции определения «победителя», как это подразумевается в выражении (4.1).

2. Величины h_{ji}, используемые выше, должны удовлетворять соотношению $\sum_j h_{ji} = 1$, в силу чего, в общем случае, $h_{ji} \neq h_{ij}$ при $i \neq j$.

В практических нейросетевых алгоритмах вычисление выражения (4.1) для нахождения «победителя» будет также более трудоемким по сравнению с простым подсчетом расстояний.

Несмотря на это, подобного рода теоретические построения полезны, поскольку позволяют пролить дополнительный свет на природу процессов СОК.

Целью работы является кластеризация сетевых пакетов, созданных программными закладками методом самоорганизующихся карт Кохонена. В качестве 9-тимерного входного вектора используется набор из IP получателя (4 числа); обратная зона DNS IP адреса получателя (3 цифры); электронная почта из WHOIS записи (2 цифры). В результате выполнения алгоритма получается кластеризация сетевой активности программных закладок.

Список литературы

1. Erwin E., Obermayer K., Schulten K. Self-organizing maps: Ordering, convergence properties and energy functions // Biological Cybernetics. - 1992. Vol. 67, - p. 47-55.

2. Kohonen T. Artificial Neural Network. - Amsterdam: North Holland, 1991. - Vol. 2, - p.981- 990.

3. Kohonen T. Things you havenot heard about the Self-Organizing Map // In Proc. ICNN‘93, Int.Conf. on Neural Networks. - 1993. - p. 1147-1156.

4. Кохонен Т. Самоорганизующиеся карты - М.: БИНОМ, Лаборатория знаний, 2008. - 655 с.

5. Ritter H., Martinetz T., Schulten K. Neural Computation and Self-Organizing Maps: An Introduction. Addison-Wesley, Reading, MA, 1992. - p. 350.

6. Luttrell S.P. Code vector density in topographic mappings. DRA, Malvern, 1992.

7. Heskes T.M., Kappen B. Error potential for self-organization // Int. Conf. on Neural Networks. - 1993. - Vol. 3, - p.1219-1223.

Neural Networks and Kohonon maps.

© **R. F. Minikaev**[3], **Z. Y. Yakupov**[4]

Abstract. In the work research of Kohonen maps is analyzed and self-organizing processes are revealed. Optimal parameters of the algorithm of self-organizing maps for clustering network connections are received and the application realizing given algorithm is created.

Key Words: Neural Networks, Kohonen maps, vector quantization.

[3]Student, Kazan State Technical University after A. N. Tupolev, Kazan; minikaev@gmail.com.
[4]Associate professor, Kazan State Technical University after A. N. Tupolev, Kazan; zymat@bk.ru.