Приложение № 2
к распоряжению РТУ МИРЭА
от__ _____№_____

Титульный лист материалов по дисциплине
(заполняется по каждому виду учебного материала)

ДИСЦИЛИНА Проектирование и обучение нейронных сетей
(полное наименование дисциплины без сокращений)

ИНСТИТУТ Информационые технологии

КАФЕДРА Вычислительная техника
полное наименование кафедры

ВИД УЧЕБНОГО МАТЕРИАЛА Лекция
(в соответствии с пп.1-11)

ПРЕПОДАВАТЕЛЬ Сорокин Алексей Борисович
(фамилия, имя, отчество)

СЕМЕСТР 7 семестр, 2023/2024
(указать семестр обучения, учебный год)

9. ЛЕКЦИЯ. Нейронные сети Кохонена

Задачи кластеризации

Сети (слои) Кохонена относятся к самоорганизующимся нейронным сетям. Самоорганизующаяся сеть позволяет выявлять кластеры (группы) входных векторов, обладающих некоторыми общими свойствами.

Кластеризация — это разделение исследуемого множества объектов на группы "похожих" объектов, называемых кластерами. Задача кластеризации принципиально отличается от задачи классификации. Решением задачи классификации является отнесение каждого из объектов к одному из заранее определенных классов. В задаче кластеризации происходит отнесение объекта к одному из заранее неопределенных классов. Разбиение объектов по кластерам осуществляется при одновременном формировании кластеров.

Формально задача кластеризации описывается следующим образом. Дано множество объектов $I = \{i_1, i_2, \dots, i_n\}$, каждый из которых харектиризуется вектором x_j, j =1,2,…,n атрибутов (параметров): $x_j = \{x_{j1}, x_{j2}, \dots, x_{jm}\}$. Требуется построить множество кластеров C и отображение F множества I на множество C, то есть $F : I \to C$. Задача кластеризации состоит в построении множества

$$C = \{c_1, c_2, \dots, c_k, \dots, c_g\}$$

где k_c – кластер, содержащий «похожие» объекты из множества I

$$c_k = \left\{ i_j, i_p \middle| i_j \in I, i_p \in I \text{ и } d(i_j, i_p) < \sigma \right\}, \quad (1)$$

σ – величина, определяющая меру близости для включения объектов в один кластер, $d(i_j, i_p)$ – мера близости между объектами, называемая расстоянием.

Если расстояние $d(i_j, i_p)$ меньше некоторого значения σ, то объекты считаются близкими и помещаются в один кластер. В противном случае считается, что объекты отличны друг от друга и их помещают в разные кластеры. Условие (1) известно, как *гипотеза компактности*.

Кластеризация основана на использовании расстояния между векторами. Неотрицательное число $d(x, y)$ называется расстоянием (метрикой) между векторами x и y, если выполняются следующие условия:

1. $d(x, y) \geq 0$ для всех x и y.
2. $d(x, y) = 0$, тогда и только тогда, когда x = y.
3. $d(x, y) = d(y, x)$.
4. $d(x, y) \leq d(x, k) + d(k, y)$ — неравенство треугольника.

Кластеризация позволяет сгруппировать сходные данные, что облегчает решение ряда задач Data Mining (интеллектуальный анализ данных):

- Изучение данных, облегчение анализа. Содержательный анализ полученных кластеров позволяет обнаружить закономерности. Например, можно выявить группы клиентов сети сотовой связи, для которых можно предложить новый тарифный план. Другие примеры — выявление групп покупателей торговой сети, сегментация рынка. Анализ содержания кластера позволяет применить к объектам различных кластеров разные методы анализа.
- Прогнозирование. Относя новый объект к одному из кластеров, можно прогнозировать поведение объекта, поскольку его поведение будет схожим с поведением объектов кластера.
- Обнаружение аномалий. Содержательный анализ кластеров помогает выявить аномалии. Обычно, это кластеры, в которые попадает мало объектов.

Структура сети Кохонена

Сеть Кохонена – это искусственная нейронная сеть, содержащая входной слой и слой активных нейронов. Поскольку входной слой просто распределяет входные сигналы по нейронам активного слоя, такую сеть можно считать однослойной.

Обычно активный слой одномерный или двумерный. Это отражает представления нейробиологии, в соответствии с которыми сенсорная информация, представленная множеством сигналов, отображается в линейные или планарные структуры коры головного мозга.

Различают два варианта сетей Кохонена:

1. Слой Кохонена. В нем нейроны активного слоя не упорядочены. В процессе обучения подстраиваются веса только одного нейрона-победителя.

2. Карта Кохонена (Self-Organized Map – SOM). В данном случае нейроны активного слоя образуют регулярную структуру. Такая карта применяется обычно для кластеризации графических изображений и звуковых сигналов

В процессе обучения SOM после предъявления достаточного числа образов все выходные нейроны сети разбиваются на подмножества, каждое из которых «откликается» на образы соответствующего класса.

Важное свойство SOM после обучения заключается в том, что чем более похожи объекты из обучающей выборки, тем ближе должны находиться нейроны активного слоя, ассоциированные с кластерами данных объектов. Расстояние между нейронами определяется топологией сети. На рис. 1 приведены варианты нейрона с восемью или шестью соседями.

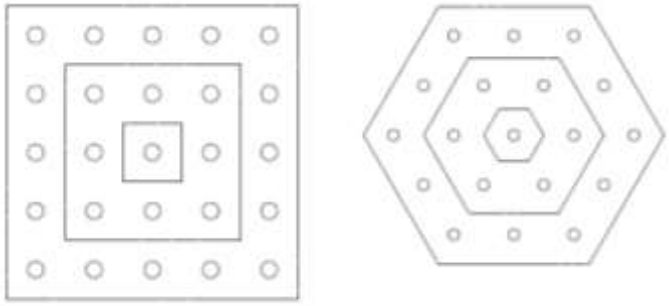

Рис. 1. Варианты топологии двумерной SOM

Таким образом, нейроны с номерами i и j характеризуются своими позициями I и J, с помощью которых можно описать функцию близости (neighbourhood function), которая может иметь вид гауссовой функции

$$g(i,j) = \exp\left(-\frac{\|I-J\|^2}{2\sigma^2}\right),$$

где σ – уменьшающаяся во времени константа

С помощью функции близости описывается область взаимодействия нейронов слоя Кохонена при обучении. На рис. 2 представлена одномерная сеть Кохонена из N нейронов.

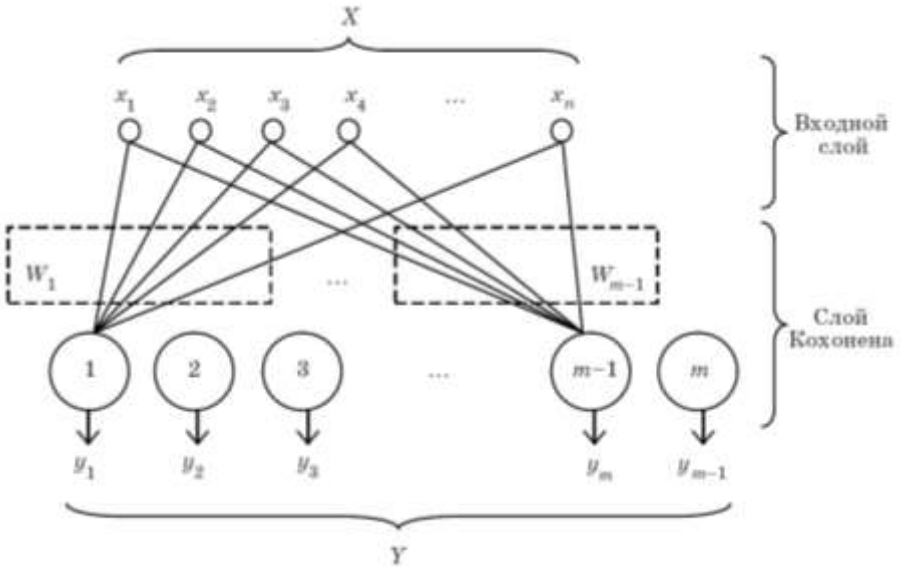

Рис. 2. Одномерная сеть Кохонена

С каждым нейроном Кохонена ассоциируется весовой вектор W, которому соответствует точка входного пространства.

Входной слой просто передает компоненты входного вектора X на каждый нейрон слоя Кохонена. Предполагается, что $N >> m$.

Каждый i-й нейрон 2-го слоя имеет собственный вектор весов Wi, который сравнивается с входным вектором X. Сравнение предполагает вычисление

расстояния между X и Wi, так что в слое Кохонена появляется нейрон-победитель с номером j, веса которого имеют наименьшее расстояние до входного вектора:

$$j = \arg\min_i \|X - W_i\|$$

В качестве метрики здесь может выступать евклидово расстояние

$$\|X - W\| = \sqrt{\sum_{i=1}^{n} (x_i - w_i)^2} \quad (1)$$

Если векторы X и W нормализованные, то в качестве меры близости можно использовать скалярное произведение, и тогда выход нейрона можно описать формулой

$$y_j = W_j X = \sum_{i=1}^{n} w_{ij} x_i, \quad (2)$$

и выход нейрона j оказывается максимальным при одинаковых X и W:

$$j = \arg\max_i \|W_i X\|.$$

Нормализация векторов выполняется по формулам

$$w_i' = \frac{w_i}{\sqrt{\sum_{j=1}^{n} w_j^2}}, \quad x_i' = \frac{x_i}{\sqrt{\sum_{j=1}^{n} x_j^2}}, \quad i = \overline{1,n}.$$

Таким образом, результатом работы слоя конкурирующих нейронов при подаче на входной слой некоторого вектора X является определение нейрона, который имеет наибольший выходной сигнал yj (нейрон-победитель). Этот нейрон обладает весовым вектором Wj, который наиболее близок к входному вектору.

Нейроны слоя Кохонена работают не изолированно, между ними существуют соревновательные связи, с помощью которых близкие нейроны усиливают сигналы друг друга. Формула (2) дополняется вторым слагаемым:

$$y_j = \sum_{i=1}^{n} w_{ij} x_i + \sum_{l=j} g(j,l) y_l, \quad (3)$$

где $g(j,l)$ – сила связи между нейронами, которую часто называют функцией соседства.

Возможны разные варианты $g(j,l)$.

Простейший вариант функции соседства одномерного слоя

$$g(i,j) - \begin{cases} 1, & |i-j| \le r, \\ 0, & |i-j| > r. \end{cases}$$

Гауссова функция соседства может быть описана в виде

$$g(i,j) - \exp\left(\frac{R^2}{2\sigma^2}\right)$$

где σ – выбранная константа; R – расстояние между нейронами Кохонена, вычисляемое по формуле

$$R - \left\| r_i - r_j \right\|^2$$

Здесь r_i и r_j – координаты нейронов.

Помимо гауссовой функции можно использовать функцию, которая имеет вид «мексиканской шляпы» (вэйвлет — математическая функция, позволяющая анализировать различные частотные компоненты данных) (рис. 3).

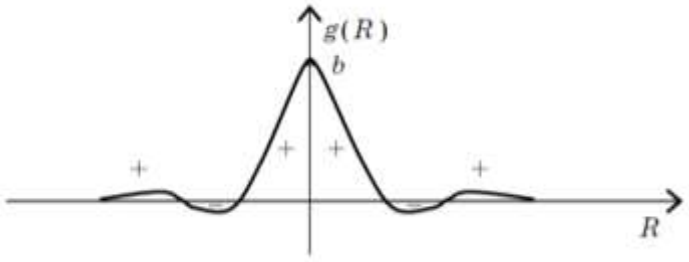

Рис. 3. Вариант описания взаимодействия нейронов

Как следует из рис. 3, вокруг точки возбуждения имеется небольшая окрестность, в которой нейроны переходят в активное состояние. За пределами данной окрестности происходит торможение активности нейронов. Наблюдается также слабая связь активности между нейронами с места возбуждения и нейронами с удаленных участков. На практике можно использовать упрощенный подход к описанию функции взаимодействия (рис. 4).

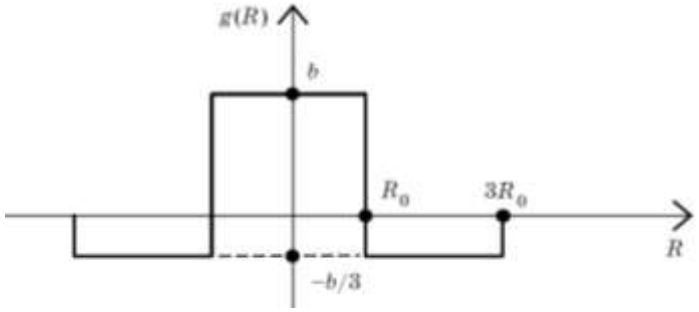

Рис. 4. Упрощенное представление силы связей слоя Кохонена

В соответствии с рис. 4 силу связи можно описать следующим образом:

$$g(R)=\begin{cases} b, & R\in[-R_0,R_0], \\ -\dfrac{b}{3}, & R\in[-R_0,\ -3R_0], \\ \dfrac{b}{3}, & R\in[R_0,\ 3R_0], \\ 0 & \text{в противном случае.} \end{cases}$$

При рассмотрении сети с двумерным слоем Кохонена функции взаимного влияния нейронов должны быть описаны для плоскости (рис. 5). Можно рассматривать также трехмерный слой Кохонена.

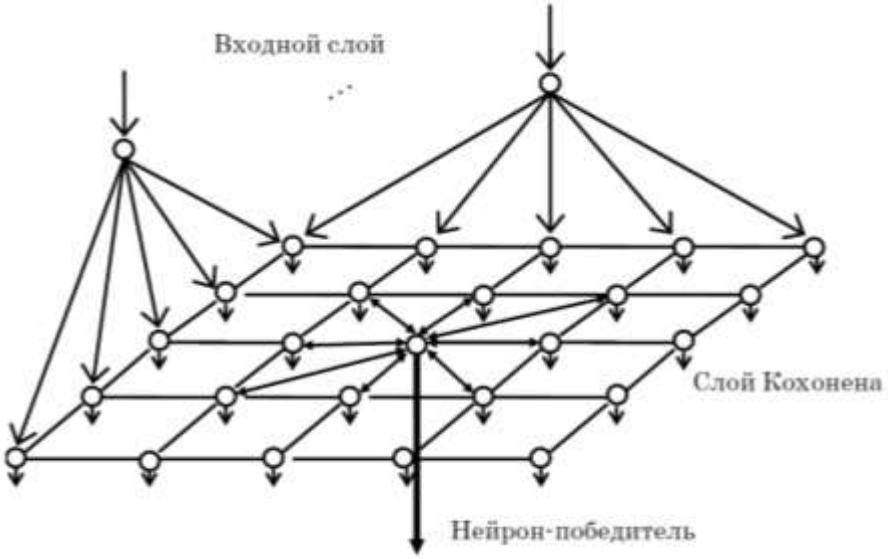

Рис. 5. ИНС с двумерным слоем Кохонена

Таким образом, сеть (слой) Кохонена – это однослойная сеть, построенная из нейронов типа WTA (Winner Takes All — победитель получает все).

Обучение сети Кохонена

Алгоритмы обучения сети Кохонена реализуют процесс обучения без учителя. При обучении n входных весов нейрона Кохонена рассматриваются как вектор в n-мерном пространстве. До обучения веса получают малые случайные значения. Затем каждый n-мерный вектор нормализуется в вектор единичной длины (с тем же направлением, что и исходный вектор).

При $n = 2$ все входные векторы можно изобразить на единичной окружности, при $n = 3$ – на единичной сфере, а для $n > 3$ следует рассматривать единичную гиперсферу.

Входные векторы обучающего набора также нормализуются, после чего сеть обучается по следующему алгоритму:

1. Подается входной вектор X.

2. Определяется нейрон-победитель с номером j по формуле (2).

3. Происходит подстройка весов нейронов.

а) если обучается слой конкурирующих нейронов, то обучается один нейрон:

$$W_i^{\mathrm{T}}(t+1) = W_i^{\mathrm{T}}(t) + \eta(X(t) - W_i^{\mathrm{T}}(t));$$

б) если обучается SOM, то происходит подстройка весов нейронов Кохонена по формуле

$$W_i^{\mathrm{T}}(t+1) = W_i^{\mathrm{T}}(t) + \eta g(i,j)(X(t) - W_i^{\mathrm{T}}(t)).$$

Здесь η – коэффициент скорости обучения.

4. Предъявляется новый входной вектор X, и шаги 1–4 повторяются.

Значение η должно постепенно уменьшаться.

В конце обучения влияние нейронов друг на друга может стать таким малым, чтобы обучался всего один нейрон, т. е. при подаче на вход сети некоторого входного вектора будет возбуждаться только один нейрон.

Каждый вес, связанный с выигравшим нейроном Кохонена, изменяется пропорционально разности между его величиной и величиной входа, к которому он присоединен. Для случая двух входов это можно представить геометрически (рис. 6).

Если бы с каждым нейроном Кохонена ассоциировался только один вектор, то слой Кохонена мог бы быть обучен путем простого присвоения весам значений этого входного вектора. Но поскольку обучение происходит по всем векторам, веса нейрона получают в конце концов усреднением по всем векторам, которые должны его активизировать.

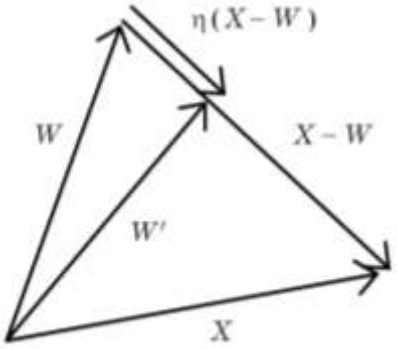

Рис. 6. Сближение вектора весов и входного вектора

Нормализованные входные и весовые векторы можно рассматривать как точки на поверхности единичной гиперсферы. Входные векторы группируются в классы (образы). Если переписать весовой вектор в область, соответствующую центру некоторого класса, то нейрон будет возбуждаться при появлении на входе любого вектора этого класса.

Входные векторы, которым обучают слой Кохонена, обычно распределены по поверхности гиперсферы не равномерно, а занимают локальные ее участки. Если до обучения выбирать веса случайно, можно получить ситуацию, при которой большинство весов будет настолько удалено от входных векторов, что многие нейроны Кохонена всегда будут иметь нулевой выход. Поэтому необходимо, чтобы в окрестности каждого входного вектора находились весовые векторы. Для того чтобы добиться этого, применяют *метод выпуклой комбинации*.

Все веса приравниваются к одной и той же величине $1/\sqrt{n}$ где n – размерность входного и весового векторов, т. е. все компоненты весовых векторов совпадают и сами они имеют единичную длину. Каждой компоненте вектора входа X придается значение

$$X_i = \propto X_i + \left(\frac{1}{\sqrt{n}}\right)(1-\propto)$$

Сначала α мало, и все входные компоненты близки к весовым компонентам. Потом α увеличивают, и входной вектор становится все более похож на самого себя. Весовые векторы отслеживают один вектор или небольшую их группу.

Кроме метода выпуклой комбинации существуют и другие методы. Например, к входным векторам можно добавлять шумы, чтобы каждый нейрон в конце концов захватил свой весовой вектор.

Еще один метод заставляет каждый нейрон Кохонена действовать «по справедливости», так, чтобы нейроны, которые возбуждаются чаще других, были подавлены, и могли обучиться все нейроны.

Пример

Рассмотрим простой пример обучения слоя конкурирующих нейронов. Пусть даны четыре входных вектора:

$$X_1 = \begin{bmatrix}1\\1\\0\\0\end{bmatrix}, \quad X_2 = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}, \quad X_3 = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \quad X_4 = \begin{bmatrix}0\\0\\1\\1\end{bmatrix}.$$

Они должны быть кластеризованы сетью Кохонена, имеющей два нейрона. При обучении изменяются веса только нейрона-победителя. Пусть веса нейронов получили некоторые случайные начальные значения (ненормализованные):

$$W_1^{\mathrm{T}} = \begin{bmatrix} 0{,}2 \\ 0{,}6 \\ 0{,}5 \\ 0{,}9 \end{bmatrix}, \qquad W_2^{\mathrm{T}} = \begin{bmatrix} 0{,}8 \\ 0{,}4 \\ 0{,}7 \\ 0{,}3 \end{bmatrix}.$$

На первом шаге обучения подается входной вектор X_1 и вычисляется расстояние от него до весовых векторов (по формуле (1)):

$$d_1 = \left\| X_1 - W_1^{\mathrm{T}} \right\| = \sqrt{\sum_{i=1}^{4} (x_i - w_i)^2} \approx 1{,}36,$$

$$d_1 = \sqrt{(1-0{,}2)^2 + (1-0{,}6)^2 + (0-0{,}5)^2 + (0-0{,}9)^2} \approx 1{,}36$$

$$d_2 = \left\| X_1 - W_2^{\mathrm{T}} \right\| = \sqrt{\sum_{i=1}^{4} (x_i - w_i)^2} \approx 0{,}99.$$

$$d_2 = \sqrt{(1-0{,}8)^2 + (1-0{,}4)^2 + (0-0{,}7)^2 + (0-0{,}3)^2} \approx 0{,}99$$

Таким образом, победившим оказывается второй нейрон, и его веса подстраиваются (η = 0,6):

$$W_2^{\mathrm{T}} = \begin{bmatrix} 0{,}8 \\ 0{,}4 \\ 0{,}7 \\ 0{,}3 \end{bmatrix} + 0{,}6 \cdot \left(\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0{,}8 \\ 0{,}4 \\ 0{,}7 \\ 0{,}3 \end{bmatrix} \right) = \begin{bmatrix} 0{,}92 \\ 0{,}76 \\ 0{,}28 \\ 0{,}12 \end{bmatrix}.$$

На втором шаге обучения подается входной вектор X_2:

$$d_1 = \left\| X_2 - W_1^{\mathrm{T}} \right\| = \sqrt{\sum_{i=1}^{4} (x_i - w_i)^2} \approx 0{,}81,$$

$$d_2 = \left\| X_2 - W_2^{\mathrm{T}} \right\| = \sqrt{\sum_{i=1}^{4} (x_i - w_i)^2} \approx 1{,}5.$$

Победившим оказывается первый нейрон, и его веса подстраиваются:

$$W_1^{\mathrm{T}} = \begin{bmatrix} 0{,}2 \\ 0{,}6 \\ 0{,}5 \\ 0{,}9 \end{bmatrix} + 0{,}6 \cdot \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0{,}2 \\ 0{,}6 \\ 0{,}5 \\ 0{,}9 \end{bmatrix} \right) = \begin{bmatrix} 0{,}08 \\ 0{,}24 \\ 0{,}20 \\ 0{,}96 \end{bmatrix}.$$

Аналогично подаются векторы X_3 и X_4, корректируются веса, и первая эпоха обучения заканчивается.

На второй и последующих эпохах значение η постепенно уменьшается, происходит приближение весов к предельным значениям:

$$W_1^T = \begin{bmatrix} 0 \\ 0 \\ 0{,}5 \\ 1 \end{bmatrix}, \quad W_2^T = \begin{bmatrix} 1 \\ 0{,}5 \\ 0 \\ 0 \end{bmatrix}.$$

Таким образом, центр первого кластера (W_1) оказывается усреднением компонент векторов X_2 и X_4, а центр второго кластера (W_2) – усреднением компонент векторов X_1 и X_3.

Контекстные карты

Существуют два фундаментально отличных метода визуализации самоорганизующихся карт признаков. Первый из этих методов состоит в построении гибкой сети, в которой векторы синоптических весов являются указателями, направленными от соответствующих нейронов во входное пространство. Этот метод визуализации особенно полезен для правильного отображения свойства топологического упорядочивания алгоритма SOM.

Во втором методе визуализации нейронам в двумерной решетке (представляющим выходной слой сети) назначаются метки классов, в зависимости от того, как каждый из примеров (не встречавшихся ранее) возбудил конкретный нейрон в самоорганизующейся сети. В результате этой второй ступени моделирования нейроны в двумерной решетке разбиваются на некоторое количество когерентных областей (coherent region). Здесь под когерентностью понимается то, что каждая из групп нейронов представляет обособленное множество непрерывных символов или меток. Это подразумевает, что при восстановлении хорошо упорядоченной карты использовались правильные условия.

Для примера рассмотрим множество данных, представленных в табл. 1. В этой таблице приведены характеристики отдельных животных. На основании данных каждого из столбцов таблицы можно составить описание некоторого животного. При этом значение 1 подразумевает наличие, а 0 - отсутствие одного из 13 свойств, перечисленных в таблице слева. Отдельные атрибуты (например, "2 ноги" и "4 ноги") Коррелируют, в то время как остальные нет. Для каждого из представленных животных можно составить характеристический код (attribute code) Xa, состоящий из 13 элементов. Само животное определяется символьным кодом (symbolic code) Xs, по которому никак нельзя собрать информацию о сходствах и различиях между отдельными животными. Например, в данном примере Xs представляет собой вектор, k-й элемент (соответствующий номеру

животного в списке) которого имеет некоторое значение а; при этом все остальные элементы равны нулю. Параметр а определяет относительное влияние характеристического и символьного кодов друг на друга (рис.7).

Животное		*Г о л у б ь*	*К у р и ц а*	*У т к а*	*Г у с ь*	*С о в а*	*Я с т р е б*	*О р е л*	*Л и с а*	*С о б а к а*	*В о л к*	*К о ш к а*	*Т и г р*	*Л е в*	*Л о ш а д ь*	*З е б р а*	*К о в а*
Размер	*Маленький*	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	Средний	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	Крупный	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Имеет	*2 ноги*	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 ноги	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	Шерсть	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	Копыта	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	Гриву	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	Перья	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Любит	*Охоту*	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	Бегать	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	Летать	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	Плавать	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Рис. 7. Пример контекстной карты

Для того чтобы гарантировать, что данный характеристический код является доминантным, а выбирается равным 0,2. Входной вектор для каждого из животных состоит из 29 элементов и представляет собой объединение характеристического Ха и символьного Xs кодов:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_s \\ \mathbf{x}_a \end{bmatrix} = \begin{bmatrix} \mathbf{x}_s \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{x}_a \end{bmatrix}$$

Наконец, каждый из векторов данных нормализуется к единичной длине. Примеры из таким образом созданного множества данных подаются на вход двумерной решетки нейронов размером 10×10, при этом синаптические веса нейронов настраиваются в соответствии с алгоритмом SOM. Обучение продолжалось в течение 2000 итераций, после чего карта признаков должна была достичь стабильного состояния. После этого самоорганизующейся сети подавались на вход векторы $x = [x_s, 0]^{\mathrm{T}}$, содержащие только символьные коды одного из животных, и определялся нейрон с самым сильным откликом. Эта процедура повторялась для всех 16 животных.

В процессе выполнения описанного алгоритма была получена карта, показанная на рис. 8.

собака	.	.	лиса	.	.	кошка	.	.	орел
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	сова
.	.	.	.	.	.	тигр	.	.	.
волк	.	.	.	.	.	.	.	.	ястреб
.	.	.	лев	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	голубь
лошадь	.	.	.	.	.	.	курица	.	.
.	.	.	.	корова	.	.	.	.	гусь
зебра	.	.	.	.	.	.	утка	.	.

Рис. 8. Контекстная карта

В ней маркированные нейроны дали самый сильный отклик на соответствующие примеры. Точками на карте обозначены нейроны с более слабыми откликами.

На следующем рис. показан результат моделирования для той же самоорганизующейся сети (рис.9).

собака	собака	лиса	лиса	лиса	кошка	кошка	кошка	орел	орел
собака	собака	лиса	лиса	лиса	кошка	кошка	кошка	орел	орел
волк	волк	волк	лиса	кошка	тигр	тигр	тигр	сова	сова
волк	волк	лев	лев	лев	тигр	тигр	тигр	ястреб	ястреб
волк	волк	лев	лев	лев	тигр	тигр	тигр	ястреб	ястреб
волк	волк	лев	лев	лев	сова	голубь	ястреб	голубь	голубь
лошадь	лошадь	лев	лев	лев	голубь	курица	курица	голубь	голубь
лошадь	лошадь	зебра	корова	корова	корова	курица	курица	голубь	голубь
зебра	зебра	зебра	корова	корова	корова	курица	курица	утка	гусь
зебра	зебра	зебра	корова	корова	корова	утка	утка	утка	гусь

Рис. 9. Результат моделирования

Однако на этот раз каждый из нейронов сети быть маркирован названием того животного, для которого его отклик был самым сильным. На этом рисунке видно, что карта признаков четко отслеживает "родственные взаимосвязи" между 16 различными животными. На нем можно выделить три обособленных кластера, один из которых представляет птиц, второй мирных животных, а третий хищников. Карта признаков, показанная на последнем рис., называется контекстной (contextual) или семантической (semantic).

Контекстные карты, получившиеся в результате использования алгоритма SOM, нашли свое применение в решении задач создания фонетических классов для текста, исследования Земли или дистанционного зондирования, а также в исследовании данных и извлечении скрытых закономерностей (data exploration & data mining).