

Лекция по дисциплине «Нейронные сети»

Лекция 1.12 Модели нейронных сетей- 1 час

Цель: - Рассмотреть основные понятия моделей нейронных сетей;

План: - Нейронные сети Кохонена и самоорганизующиеся карты.

- Сети векторного квантования
- Самоорганизующиеся карты Кохонена

Нейронные сети Кохонена представляют собой отдельный класс нейронных сетей и относятся к самоорганизующимся нейронным сетям. Нейронная сеть Кохонена является однослойной сетью, каждый нейрон которой соединен со всеми компонентами n -мерного входного вектора. Входной вектор – это описание одного из объектов, подлежащих кластеризации. Количество нейронов m совпадает с количеством кластеров, которое должна выделить сеть. В качестве нейронов сети Кохонена применяются линейные взвешенные сумматоры. Входной вектор сети имеет вид $X=(x_1, x_2, \dots, x_n)$. Каждый j -ый нейрон описывается вектором весов синаптических связей в n -мерном пространстве $W_j=(w_{1j}, w_{2j}, \dots, w_{nj})$.

Структура сети Кохонена представлена на рисунке П.4.5.

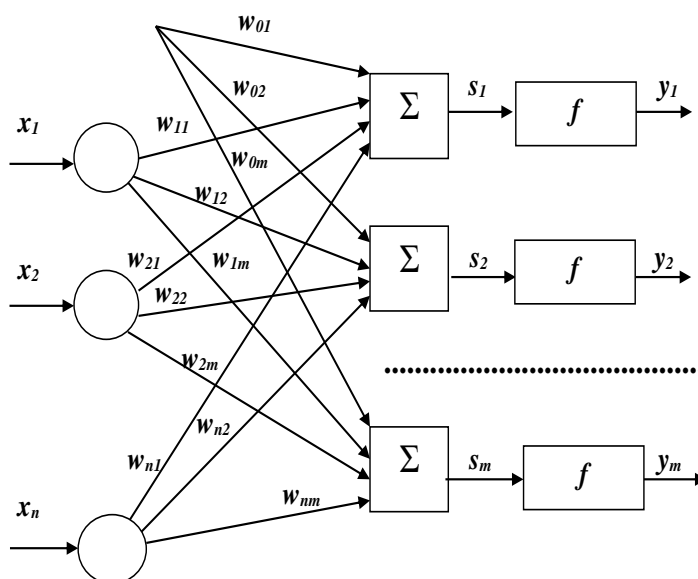


Рисунок П.4.5. Структура нейронной сети Кохонена.

Сеть Кохонена состоит из m параллельно действующих линейных элементов. Все они имеют одинаковое число входов n и получают на свои входы один и тот же вектор входных сигналов $X=(x_1, x_2, \dots, x_n)$. Выходной сигнал j -го линейного элемента будет:

$$y_j = w_{0j} + \sum_{i=1}^n w_{ij} x_i,$$

где w_{0j} – пороговый коэффициент; w_{ij} – весовой коэффициент i -го входа j -го нейрона; $i=1, \dots, n$ – номер входного сигнала; $j=1, m$ – номер линейного элемента.

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

$$d(x, W_j) = \min_{1 \leq j \leq m} d(x, W_j),$$

m – Количество нейронов, j – номер нейрона-победителя,

$d(x, W_j)$ – расстояние (в смысле выбранной метрики) между векторами x и W_j .

В качестве меры расстояния используется *евклидова мера* $\rho_j(x)$ до точек W_j .

$$\rho_j(x) = d(x, W_j) = \|x - W_j\|^2 = \|x\|^2 - 2 \sum_{j=1}^m x_j \cdot w_{ij} + \|W_j\|^2$$

Здесь $\|y\|$ - Евклидова длина вектора: $\|y\|^2 = \sum_{j=1}^m y_j^2$. Первое слагаемое

одинаково для всех нейронов, поэтому для нахождения ближайшей точки оно не нужно.

После прохождения слоя линейных элементов сигналы посылаются на обработку по правилу «победитель забирает всё». Среди выходных сигналов выполняется поиск максимального, определяется его номер. Окончательно, на выходе сигнал с найденным номером равен единице, остальные – нулю. Если максимум одновременно достигается для нескольких, то:

- либо принимают все соответствующие сигналы равными единице;
- либо равным единице принимают только первый сигнал в списке (по соглашению).

Задача сводится к поиску номера наибольшего из значений линейных функций, т.е.:

$$j_{\max} = \arg \max_j \{y_j\} = \arg \max_j \left\{ \sum_{j=1}^m x_j \cdot w_{ij} - \frac{1}{2} \|W_j\|^2 \right\}.$$

Таким образом, координаты точки $W_j = (w_{1j}, w_{2j}, \dots, w_{nj})$ совпадают с весами линейного слоя Кохонена, при этом значение порогового коэффициента

$$w_{0j} = \frac{1}{2} \|W_j\|^2.$$

Вокруг нейрона-победителя образуется *окружение (neighborhood)*, или радиус обучения (*radius of learning*). Радиус обучения определяет сколько нейронов кроме нейрона-победителя участвуют в обучении (корректируют свои веса) на данной итерации. Под радиусом в данном случае подразумевается расстояние в пространстве векторов весов нейронов. То есть, любой нейрон, расстояние от вектора весов которого до вектора весов нейрона-победителя меньше, чем радиус обучения, участвует в коррекции весов на данной итерации. Радиус обучения максимален на данной итерации

и уменьшается с увеличением числа итерации 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 по правилу Кохонена. При этом выделяют сети с неупорядоченными нейронами, называемые слоями Кохонена, и сети с упорядоченными нейронами, называемые самоорганизующимися картами (Self-Organizing Maps – SOM) Кохонена.

Слой Кохонена состоит из адаптивных линейных сумматоров («линейных формальных нейронов»). Как правило, выходные сигналы слоя Кохонена обрабатываются по правилу «Победитель получает всё», наибольший сигнал превращается в единичный, остальные обращаются в ноль.

«Нейроны Кохонена можно воспринимать как набор электрических лампочек, так что для любого входного вектора загорается одна из них».

Сети векторного квантования. Задача векторного квантования с k кодовыми векторами W_j для заданной совокупности входных векторов X ставится как задача минимизации искажения при кодировании, то есть при замещении каждого вектора из X соответствующим кодовым вектором. В базовом варианте сетей Кохонена используется метод наименьших квадратов и искажение D вычисляется по формуле

$$D = \sum_{i=1}^n \sum_{x \in K_j} \|x - W_j\|^2,$$

где K_j состоит из тех точек $x \in X$, которые ближе к W_j , чем к другим W_i ($i \neq j$). Другими словами, K_j состоит из тех точек $x \in X$, которые кодируются кодовым вектором W_j .

Если совокупность X задана и хранится в памяти, то стандартным выбором в обучении соответствующей сети Кохонена является метод K -средних. Это метод расщепления:

- при данном выборе кодовых векторов (они же весовые векторы сети) W_j минимизацией D находим множества K_j , которые состоят из тех точек $x \in X$, которые ближе к W_j , чем к другим W_i ;

- при данном разбиении X на множества K_j минимизацией D находим оптимальные позиции кодовых векторов W_j , для оценки по методу наименьших квадратов это просто средние арифметические:

$$W_j = \frac{1}{|K_j|} \sum_{x \in K_j} x,$$

где $|K_j|$ – число элементов в K_j .

Далее итерируем. Этот метод расщепления сходится за конечное число шагов и даёт локальный минимум искажения.

Если же, например, совокупность X заранее не задана, или по каким-либо причинам не хранится в памяти, то широко используется онлайн метод. Векторы входных сигналов x обрабатываются по одному, для каждого из них находится ближайший кодовый вектор («победитель», который «забирает всё») $W_{j(x)}$. После этого данный кодовый вектор пересчитывается по формуле:

$$W_{j(x)}^{new} = W_{j(x)}^{old} (1 - \theta) + \theta x,$$

где $\theta \in (0,1)$ – шаг обучения. Остальные кодовые векторы на этом шаге не изменяются.

Для обеспечения стабильности используется онлайн метод с затухающей скоростью обучения. Если T – количество шагов обучения, то полагают

$\theta = \theta(T)$. Функцию $\theta(T) > 0$ выбирают таким образом, чтобы монотонно при $T \rightarrow \infty$ ряд $D = \sum_{T=1}^n \theta(T)$ расходился, например, $\theta(T) = \frac{\theta_0}{T}$.

Векторное квантование является намного более общей операцией, чем кластеризация, поскольку кластеры должны быть разделены между собой, тогда как совокупности K_j для разных кодовых векторов W_j не обязательно представляют собой отдельные кластеры. С другой стороны, при наличии разделяющихся кластеров векторное квантование может находить их и по-разному кодировать.

По способам настройки входных весов сумматоров и по решаемым задачам различают много разновидностей сетей Кохонена. Наиболее известные из них:

- сети векторного квантования сигналов, тесно связанные с простейшим базовым алгоритмом кластерного анализа (метод динамических ядер или К-средних);
- сети векторного квантования, обучаемые с учителем (Learning Vector Quantization).

Самоорганизующиеся карты Кохонена (Self Organizing Maps – SOM) представляют собой соревновательную нейронную сеть с обучением без учителя, выполняющую задачу визуализации и кластеризации многомерных векторов.

Примером таких алгоритмов может служить алгоритм k -ближайших средних (k -means). Важным отличием алгоритма SOM является то, что в нем все нейроны (узлы, центры классов...) упорядочены в некоторую структуру (обычно двумерную сетку). При этом в ходе обучения модифицируется не только нейрон-победитель, но и его соседи, но в меньшей степени. За счет этого SOM можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью (чаще всего,

двумерное), применяется также для решения задач моделирования, прогнозирования и др. Является одной из версий нейронных сетей Кохонена.

При использовании этого алгоритма вектора, схожие в исходном пространстве, оказываются рядом и на полученной карте.

Сигнал в карте Кохонена поступает сразу на все нейроны, веса соответствующих синапсов интерпретируются как координаты положения узла, и выходной сигнал формируется по принципу «победитель забирает всё» – то есть ненулевой выходной сигнал имеет нейрон, ближайший (в смысле весов синапсов) к подаваемому на вход объекту. В процессе обучения веса синаптических связей настраиваются таким образом, чтобы узлы решетки «располагались» в местах локальных сгущений данных, то есть описывали кластерную структуру облака данных, с другой стороны, связи между нейронами соответствуют отношениям соседства между соответствующими кластерами в пространстве признаков.

Удобно рассматривать такие карты как двумерные сетки узлов, размещенных в многомерном пространстве. Изначально самоорганизующаяся карта представляет из себя сетку из узлов, соединенный между собой связями. Кохонен рассматривал два варианта соединения узлов – в прямоугольную и гексагональную сетку – отличие состоит в том, что в прямоугольной сетке каждый узел соединен с 4-мя соседними, а в гексагональной – с 6-ю ближайшими узлами. Для двух таких сеток процесс построения сети Кохонена отличается лишь в том месте, где перебираются ближайшие к данному узлу соседи. Начальное вложение сетки в пространство данных выбирается произвольным образом. В авторском пакете SOM_PAK предлагаются варианты случайного начального расположения узлов в пространстве и вариант расположения узлов в плоскости. После этого узлы начинают перемещаться в пространстве согласно следующему алгоритму:

1. Случайным образом выбирается точка данных x .
2. Определяется ближайший к x узел карты (BMU – Best Matching Unit).

х. Однако, он перемещается не один, а увлекает за собой определенное количество ближайших узлов из некоторой окрестности на карте. Из всех двигающихся узлов наиболее сильно смещается центральный, ближайший к точке данных, узел, а остальные испытывают тем меньшие смещения, чем дальше они от ВМУ. В настройке карты различают два этапа – этап грубой (ordering) и этап тонкой (fine-tuning) настройки. На первом этапе выбираются большие значения окрестностей и движение узлов носит коллективный характер, в результате карта «расправляется» и грубым образом отражает структуру данных. На этапе тонкой настройки радиус окрестности равен 1-2 узла и настраиваются уже индивидуальные положения узлов. Кроме этого, величина смещения равномерно затухает со временем, то есть она велика в начале каждого из этапов обучения и близка к нулю в конце.

4. Алгоритм повторяется определенное число эпох (понятно, что число шагов может сильно изменяться в зависимости от задачи).

Алгоритм повторяется определенное число эпох (понятно, что число шагов может сильно изменяться в зависимости от задачи). Самоорганизующиеся карты Кохонена служат для визуализации и первоначального («разведывательного») анализа данных.

Вопросы

1. К какому типу относятся нейронные сети Кохонена?
2. Чему равно количество нейронов в нейронной сети Кохонена?
3. По какому правилу наибольший сигнал превращается в единичный, остальные обращаются в ноль?