

# Нейронная сеть Кохонена

## Материал из MachineLearning.

**Нейронные сети Кохонена** (Kohonen neural network) — класс нейронных сетей, основным элементом которых является *слой Кохонена*. Слой Кохонена состоит из адаптивных линейных сумматоров («линейных формальных нейронов»). Как правило, выходные сигналы слоя Кохонена обрабатываются по правилу «победитель забирает всё»: наибольший сигнал превращается в единичный, остальные обращаются в ноль.

По способам настройки входных весов сумматоров и по решаемым задачам различают много разновидностей сетей Кохонена.<sup>[1]</sup> Наиболее известные из них:

- Сети векторного квантования сигналов<sup>[1]</sup>, тесно связанные с простейшим базовым алгоритмом кластеризации (метод динамических ядер или К-средних, то есть K-means)
- Самоорганизующиеся карты Кохонена (Self-Organising Maps, SOM)<sup>[1]</sup>
- Сети векторного квантования, обучаемые с учителем (Learning Vector Quantization)<sup>[1]</sup>

## Содержание

- 1 Слой Кохонена
  - 1.1 Базовая версия
  - 1.2 Геометрическая интерпретация
- 2 Сети векторного квантования
- 3 Самоорганизующиеся карты Кохонена
  - 3.1 Идея и алгоритм обучения
  - 3.2 Самоорганизующиеся карты и главные многообразия
- 4 Упругие карты
- 5 Сети векторного квантования, обучаемые с учителем
- 6 Учебное программное обеспечение
- 7 Ссылки

## Слой Кохонена

### Базовая версия

Слой Кохонена состоит из некоторого количества  $n$  параллельно действующих линейных элементов. Все они имеют одинаковое число входов  $m$  и получают на свои входы один и тот же вектор входных сигналов  $\mathbf{x} = (x_1, \dots, x_m)$ . На выходе  $j$ -го линейного элемента получаем сигнал

$$y_j = w_{j0} + \sum_{i=1}^m w_{ji} x_i$$

где  $w_{ji}$  — весовой коэффициент  $i$ го входа  $j$ го нейрона,  $w_{j0}$  — пороговой коэффициент.

После прохождения слоя линейных элементов сигналы посылаются на обработку по правилу «победитель забирает всё»: среди выходных сигналов  $y_j$  ищется максимальный; его номер  $j_{\max} = \arg\max_j \{y_j\}$ . Окончательно, на выходе сигнал с номером  $j_{\max}$  равен единице, остальные — нулю. Если максимум одновременно достигается для нескольких  $j_{\max}$ , то либо принимают все соответствующие сигналы равными единице, либо только первый в списке (по соглашению). «Нейроны Кохонена можно воспринимать как набор электрических лампочек, так что для любого входного вектора загорается одна из них.»<sup>[1]</sup>

## Геометрическая интерпретация

Большое распространение получили слои Кохонена, построенные следующим образом: каждому ( $j$ му) нейрону сопоставляется точка  $W_j = (w_{j1}, \dots, w_{jm})$  в  $m$ -мерном пространстве (пространстве сигналов). Для входного вектора  $x = (x_1, \dots, x_m)$  вычисляется его расстояния  $\rho_j(x)$  до точек  $W_j$  и «ближайший получает всё» — тот нейрон, для которого это расстояние минимально, выдаёт единицу, остальные — нули. Следует заметить, что для сравнения расстояний достаточно вычислять линейную функцию сигнала:

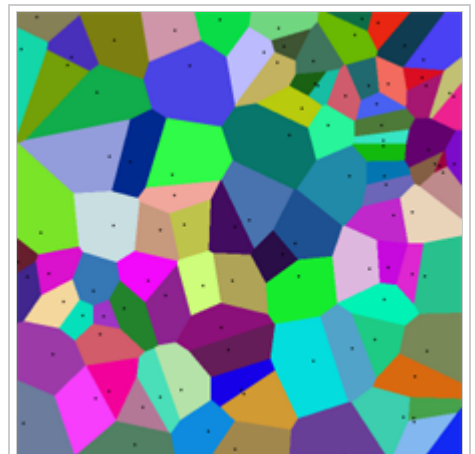
$$\rho_j(x)^2 = |x - W_j|^2 = |W_j|^2 - 2 \sum_{i=1}^m w_{ji} x_i + |x|^2.$$

Последнее слагаемое  $|x|^2$  одинаково для всех нейронов, поэтому для нахождения ближайшей точки оно не нужно. Задача сводится к поиску номера наибольшего из значений линейных функций:

$$j_{\max} = \arg\max_j \left\{ \sum_{i=1}^m w_{ji} x_i - \frac{1}{2} \|W_j\|^2 \right\}.$$

Таким образом, координаты точки  $W_j = (w_{j1}, \dots, w_{jm})$  совпадают с весами линейного нейрона слоя Кохонена (при этом значение порогового коэффициента  $w_{j0} = -|W_j|^2/2$ ).

Если заданы точки  $W_j = (w_{j1}, \dots, w_{jm})$ , то  $m$ -мерное пространство разбивается на соответствующие многогранники Вороного-Дирихле  $V_j$ : многогранник  $V_j$



Разбиение плоскости на многоугольники Вороного-Дирихле для случайно выбранных точек (каждая точка указана в своём многоугольнике).

состоит из точек, которые ближе к  $W_j$ , чем к другим  $W_k$  ( $k \neq j$ ).<sup>[1]</sup>

## Сети векторного квантования

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

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

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

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

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

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

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  $\theta(T) \rightarrow 0$  монотонно при  $T \rightarrow \infty$  и чтобы ряд  $\sum_{T=1}^{\infty} \theta(T)$  расходиллся, например,  $\theta(T) = \theta_0/T$ .

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.

## Самоорганизующиеся карты Кохонена

### Идея и алгоритм обучения

Задача векторного квантования состоит, по своему существу, в наилучшей аппроксимации всей совокупности векторов данных  $k$  кодовыми векторами  $W_j$ . Самоорганизующиеся карты Кохонена (англ. Self-Organising Maps, SOM) также аппроксимируют данные, однако при наличии дополнительной структуры в совокупности кодовых векторов (англ. codebook). Предполагается, что априори задана некоторая симметричная таблица «мер соседства» (или «мер близости») узлов: для каждой пары  $j, l$  ( $j, l = 1, \dots, k$ ) определено число  $\eta_{jl}$  ( $0 \leq \eta_{jl} \leq 1$ ) при этом диагональные элементы таблицы близости равны единице ( $\eta_{jj} = 1$ ).

Векторы входных сигналов  $x$  обрабатываются по одному, для каждого из них находится ближайший кодовый вектор («победитель», который «забирает всё»)  $W_{j(x)}$ . После этого все кодовые векторы  $W_l$ , для которых  $\eta_{j(x)l} \neq 0$ , пересчитываются по формуле

$$W_l^{\text{new}} = W_l^{\text{old}}(1 - \eta_{j(x)l}\theta) + x\eta_{j(x)l}\theta,$$

где  $\theta \in (0, 1)$  — шаг обучения. Соседи кодового вектора — победителя (по априорно заданной таблице близости) сдвигаются в ту же сторону, что и этот вектор, пропорционально мере близости.

Чаще всего, таблица кодовых векторов представляется в виде фрагмента квадратной решётки на плоскости, а мера близости определяется, исходя из евклидова расстояния на плоскости.

Самоорганизующиеся карты Кохонена служат, в первую очередь, для визуализации и первоначального («разведывательного») анализа данных.<sup>[1]</sup> Каждая точка данных отображается соответствующим кодовым вектором из решётки. Так получают представление данных на плоскости («карту данных»). На этой карте возможно отображение многих слоёв: количество данных, попадающих в узлы (то есть «плотность данных»), различные функции данных и так далее. При отображении этих слоёв полезен аппарат географических информационных систем (ГИС). В ГИС подложкой для изображения информационных слоёв служит географическая карта. Карта данных является

подложкой для произвольного по своей природе набора данных. Она служит заменой географической карте там, где ее просто не существует. Принципиальное отличие в следующем: на географической карте соседние объекты обладают близкими географическими координатами, на карте данных близкие объекты обладают близкими свойствами. С помощью карты данных можно визуализировать данные, одновременно нанося на подложку сопровождающую информацию (подписи, аннотации, атрибуты, информационные раскраски).<sup>[1]</sup> Карта служит также информационной моделью данных. С её помощью можно заполнять пробелы в данных. Эта способность используется, например, для решения задач прогнозирования.

## Самоорганизующиеся карты и главные многообразия

Идея самоорганизующихся карт очень привлекательна и породила массу обобщений, однако, строго говоря, мы не знаем, что мы строим: карта — это результат работы алгоритма и не имеет отдельного («объектного») определения. Есть, однако, близкая теоретическая идея — главные многообразия (principal manifolds).<sup>[1]</sup> Эти многообразия обобщают линейные главные компоненты. Они были введены как линии или поверхности, проходящие через «сердину» распределения данных, с помощью **условия самосогласованности**: каждая точка  $x$  на главном многообразии  $M$  является условным математическим ожиданием тех векторов  $z$ , которые проектируются на  $x$  (при условии  $x = P(z)$ , где  $P$  — оператор проектирования окрестности  $M$  на  $M$ ),

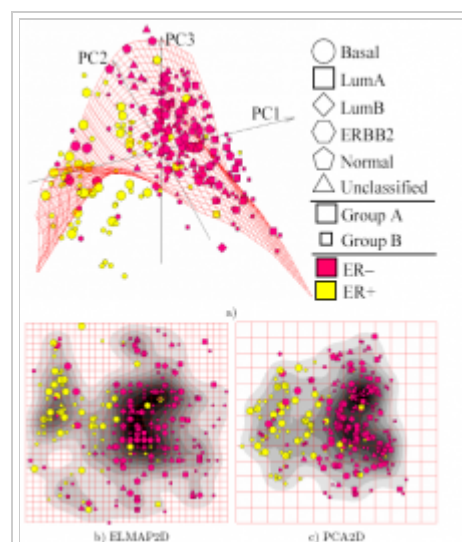
$$x = E(z|P(z) = x).$$

Самоорганизующиеся карты могут рассматриваться как аппроксимации главных многообразий и популярны в этом качестве<sup>[1]</sup>.

## Упругие карты

Метод аппроксимации многомерных данных, основанный на минимизации «энергии упругой деформации» карты, погружённой в пространство данных, был предложен А. Н. Горбанём в 1996 году, и впоследствии развит им совместно с А. Ю. Зиновьевым, А. А. Россиевым и А. А. Питенко.<sup>[1]</sup> Метод основан на аналогии между главным многообразием и эластичной мембраной и упругой пластиной. В этом смысле он является развитием классической идеи сплайна (хотя упругие карты и не являются многомерными сплайнами).

Пусть задана совокупность входных векторов  $S$ . Так же, как и сети векторного квантования и самоорганизующиеся карты, упругая карта представлена как совокупность кодовых векторов (узлов)  $W_j$  в пространстве сигналов. Множество данных  $S$  разделено на классы  $K_j$ , состоящие из тех точек  $x \in S$ , которые ближе к  $W_j$ , чем к другим



Визуализация набора данных по экспрессии генов в раке молочной железы с использованием упругих карт (b) и метода главных компонент (c). Классы точек показаны с использованием

$W_l$  ( $l \neq j$ ). Искажение кодирования  $D$

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

может трактоваться как суммарная энергия пружин единичной жёсткости, связывающих векторы данных с соответствующими кодовыми векторами.

На множестве узлов задана дополнительная структура: некоторые пары связаны «упругими связями», а некоторые тройки объединены в «рёбра жёсткости». Обозначим множество пар, связанных упругими связями, через  $E$ , а множество троек, составляющих рёбра жёсткости, через  $G$ . Например, в квадратной решётке ближайшие узлы (как по вертикали, так и по горизонтали) связываются упругими связями, а ребра жёсткости образуются вертикальными и горизонтальными тройками ближайших узлов. Энергия деформации карты состоит из двух слагаемых:

$$\begin{aligned} \text{энергия растяжения} \quad U_E &= \lambda \sum_{(W_i, W_j) \in E} |W_i - W_j|^2, \\ \text{энергия изгиба} \quad U_G &= \mu \sum_{(W_i, W_j, W_l) \in G} |W_i - 2W_j + W_l|^2, \end{aligned}$$

где  $\lambda, \mu$  — соответствующие модули упругости.

Задача построения упругой карты состоит в минимизации функционала

$$U = D + U_E + U_G;$$

Если разбиение совокупности входных векторов  $S$  на классы  $K_j$  фиксировано, то минимизация  $U$  — линейная задача с разреженной симметричной матрицей коэффициентов. Поэтому, как и для сетей векторного квантования, применяется метод расщепления: фиксируем  $\{W_j\}$  — ищем  $\{K_j\}$  — для данных  $\{K_j\}$  ищем  $\{W_j\}$  — для данных  $\{W_j\}$  ищем  $\{K_j\}$  — ... Алгоритм сходится к (локальному) минимуму  $U$ .

Метод упругих карт позволяет решать все задачи, которые решают самоорганизующиеся карты Кохонена, однако обладает большей регулярностью и предсказуемостью. При увеличении модуля изгиба  $\mu$  упругие карты приближаются к линейным главным компонентам. При уменьшении обоих модулей упругости они превращаются в Кохоненовские сети векторного квантования. В настоящее время упругие карты интенсивно используются для анализа многомерных данных в биоинформатике.<sup>[1]</sup> Соответствующее программное обеспечение опубликовано и свободно доступно на сайте Института Кюри (Париж).<sup>[1][1]</sup>

На рисунке представлены результаты визуализации данных по раку молочной железы. Эти данные содержат 286 примеров с указанием уровня экспрессии 17816 генов<sup>[1]</sup>. Они доступны онлайн как ставший классическим тестовый

размера (ER - статус эстроген-рецептора), формы (GROUP - риск развития метастаз) и цвета (TYPE - молекулярный тип опухоли).

На панели (а) показана конфигурация узлов двумерной упругой карты в проекции на первые три главные компоненты. Сравнивая (b) и (c), можно заметить, что базальный тип опухоли как кластер лучше отделен на нелинейной проекции (b).

пример для визуализации и картографии данных<sup>[1]</sup>.

## Сети векторного квантования, обучаемые с учителем

Решается задача классификации. Число классов может быть любым. Изложим алгоритм для двух классов,  $A$  и  $B$ . Исходно для обучения системы поступают данные, класс которых известен. Задача: найти для класса  $A$  некоторое количество  $k_A$

кодовых векторов  $W_j^A$ , а для класса  $B$  некоторое (возможно другое) количество  $k_B$  кодовых векторов  $W_l^B$  таким образом, чтобы итоговая сеть Кохонена с  $k_A + k_B$  кодовыми векторами  $W_j^A, W_l^B$  (объединяем оба семейства) осуществляла классификацию по следующему решающему правилу:

если для вектора входных сигналов  $x$  ближайший кодовый вектор («победитель», который в слое Кохонена «забирает всё»)

принадлежит семейству  $\{W_j^A\}$ , то  $x$

принадлежит классу  $A$ ; если же ближайший к  $x$  кодовый вектор

принадлежит семейству  $\{W_l^B\}$ , то  $x$  принадлежит классу  $B$ .

С каждым кодовым вектором объединённого семейства  $\{W_j^A\} \cup \{W_l^B\}$  связан многогранник Вороного-Дирихле. Обозначим эти многогранники  $V_j^A, V_l^B$  соответственно. Класс  $A$  в пространстве сигналов, согласно решающему

правилу, соответствует объединению  $\bigcup V_j^A$ , а класс  $B$  соответствует

объединению  $\bigcup V_l^B$ . Геометрия таких объединений многогранников может быть весьма сложной (см. Рис с примером возможного разбиения на классы).

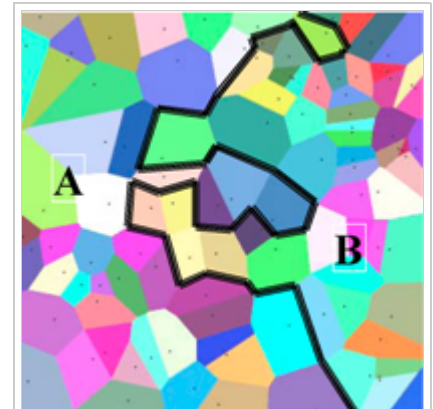
Правила обучения сети онлайн строятся на основе базового правила обучения сети векторного квантования. Пусть на вход системы подаётся вектор сигналов  $x$ , класс которого известен. Если он классифицируется системой правильно, то соответствующий  $x$  кодовый вектор  $W$  слегка сдвигается в сторону вектора сигнала («поощрение»)

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

Если же  $x$  классифицируется неправильно, то соответствующий  $x$  кодовый вектор  $W$  слегка сдвигается в противоположную сторону от сигнала («наказание»)

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

где  $\theta \in (0,1)$  — шаг обучения. Для обеспечения стабильности используется онлайн



Пример возможного разбиения классов, составленного с помощью разбиения Вороного-Дирихле.

метод с затухающей скоростью обучения. Возможно также использование разных шагов для «поощрения» правильного решения и для «наказания» неправильного

Это — простейшая (базовая) версия метода<sup>[1]</sup>. Существует множество других модификаций.

## Учебное программное обеспечение

Java-апплет «Метод главных компонент и самоорганизующиеся карты» (Е.М. Mirkes, Principal Component Analysis and Self-Organizing Maps: applet ([http://www.math.le.ac.uk/people/ag153/homepage/PCA\\_SOM/PCA\\_SOM.html](http://www.math.le.ac.uk/people/ag153/homepage/PCA_SOM/PCA_SOM.html)) . University of Leicester, 2011). Свободно распространяемая программа с моделями метода главных компонент, самоорганизующихся карт (SOM) и растущих самоорганизующихся карт (Growing Self-Organized Maps, GSOM). Используется для выполнения небольших студенческих исследовательских работ по сравнению различных алгоритмов аппроксимации данных.

## Ссылки

*Незарегистрированные пользователи не видят примечаний и основных литературных ссылок (дефект системы). Зарегистрироваться безопасно и просто.*

Источник — «[http://www.machinelearning.ru/wiki/index.php?title=%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F\\_%D1%81%D0%B5%D1%82%D1%8C\\_%D0%9A%D0%BE%D1%85%D0%BE%D0%BD%D0%B5%D0%BD%D0%B0](http://www.machinelearning.ru/wiki/index.php?title=%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C_%D0%9A%D0%BE%D1%85%D0%BE%D0%BD%D0%B5%D0%BD%D0%B0)»

Категории: Энциклопедия анализа данных | Нейронные сети | Кластеризация

---

- Последнее изменение этой страницы: 23:38, 15 мая 2012.
- Содержимое доступно в соответствии с Creative Commons Attribution/Share-Alike.