

## НЕЙРОННЫЕ СЕТИ. ОБУЧЕНИЕ БЕЗ УЧИТЕЛЯ.

На рисунке 1 показана структура нейронной сети Кохонена. Входные нейроны  $I$  (показанные кружками) служат лишь точками разветвления и не выполняют вычислений. Каждый нейрон этого слоя соединен с каждым нейроном слоя  $K$  (называемого слоем Кохонена, сами нейроны  $K$  называются нейронами Кохонена) весом  $w_{ij}$ . Эти веса в целом рассматриваются как матрица весов  $W$ . Сеть Кохонена функционирует в двух режимах: в нормальном, при котором принимается входной вектор  $X$  и выдается выходной вектор  $Y$ , и в режиме обучения, при котором подается входной вектор и веса корректируются, чтобы дать требуемый выходной вектор. Это напоминает другие разновидности нейронных сетей, различие состоит в операциях, выполняемых нейронами Кохонена.

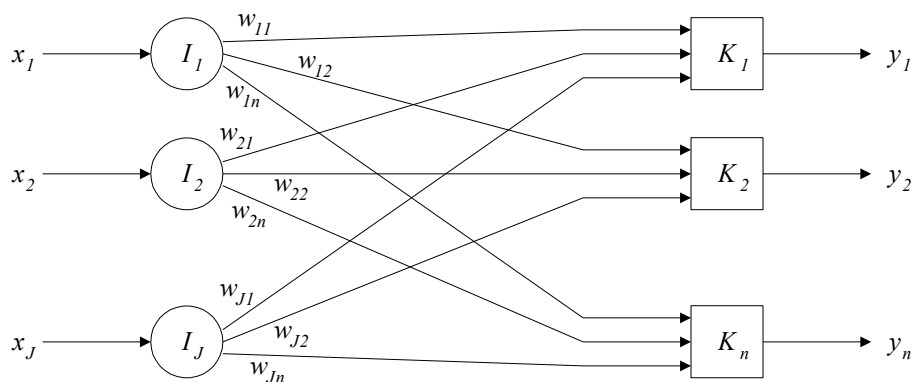


Рисунок 1

В своей простейшей форме слой Кохонена функционирует в духе «победитель забирает все», то есть для данного входного вектора один и только один нейрон Кохонена выдает на выходе логическую единицу, а все остальные выдают ноль. Нейроны Кохонена можно воспринимать как набор электрических лампочек, и для любого входного вектора «загорается» одна из них.

Подобно нейронам большинства сетей, выход  $NET$  каждого нейрона Кохонена является просто суммой взвешенных входов. Это может быть выражено следующим образом:

$$NET_j = \sum_i x_i w_{ij}$$

где  $NET_j$  - это выход  $NET$  нейрона Кохонена с номером  $j$ . Нейрон Кохонена с максимальным значением  $NET_j$  является «победителем». Его выход равен единице, у остальных он равен 0.

Слой Кохонена классифицирует входные векторы в группы схожих. Это достигается с помощью такой подстройки весов слоя Кохонена, что близкие входные векторы активируют один и тот же нейрон данного слоя. Обучение Кохонена является самообучением, протекающим без учителя. Поэтому не нужно предсказывать, какой именно нейрон Кохонена будет активироваться для заданного входного вектора. Необходимо лишь гарантированно добиться, чтобы в результате обучения разделялись несхожие входные векторы.

Перед тем, как предъявить входные векторы сети, их желательно предварительно обработать, нормализовав. Операция выполняется с помощью деления каждой компоненты входного вектора на длину вектора. Таким образом, входной вектор превращается в единичный вектор с тем же самым направлением, т.е. вектор единичной длины в  $J$  мерном пространстве.

При обучении слоя Кохонена на вход подается входной вектор и вычисляются его скалярные произведения с векторами весов, связанными со всеми нейронами Кохонена. Нейрон с максимальным значением скалярного произведения объявляется «победителем» и его веса подстраиваются. Так как скалярное произведение, используемое для вычисления величин NET, является мерой сходства между входным вектором и вектором весов, то процесс обучения состоит в выборе нейрона Кохонена с весовым вектором, наиболее близким к входному вектору, и дальнейшем приближении весового вектора к входному. Снова отметим, что процесс является самообучением, выполняемым без учителя. Сеть самоорганизуется таким образом, что данный нейрон Кохонена имеет максимальный выход для данного входного вектора. Уравнение, описывающее процесс обучения имеет следующий вид:

$$w_n = w_c + \alpha(x - w_c),$$

где  $w_n$  – новое значение веса, соединяющего входную компоненту  $x$  с выигравшим нейроном;  $w_c$  – предыдущее значение этого веса;  $\alpha$  – коэффициент скорости обучения, который может варьироваться в процессе обучения.

Каждый вес, связанный с выигравшим нейроном Кохонена, изменяется пропорционально разности между его величиной и величиной входа, к которому он присоединен. Направление изменения минимизирует разность между весом и его входом.

На рисунке 2 этот процесс показан геометрически в двумерном виде. Сначала находится вектор  $\mathbf{X} - \mathbf{W}_c$ , для этого проводится отрезок из конца  $\mathbf{W}$  в конец  $\mathbf{X}$ . Затем этот вектор укорачивается умножением его на скалярную величину  $\alpha$ , меньшую единицы, в результате чего получается вектор изменения  $\delta$ . Окончательно новый весовой вектор  $\mathbf{W}_n$

является отрезком, направленным из начала координат в конец вектора  $\delta$ . Отсюда можно видеть, что эффект обучения состоит во вращении весового вектора в направлении входного вектора без существенного изменения его длины.

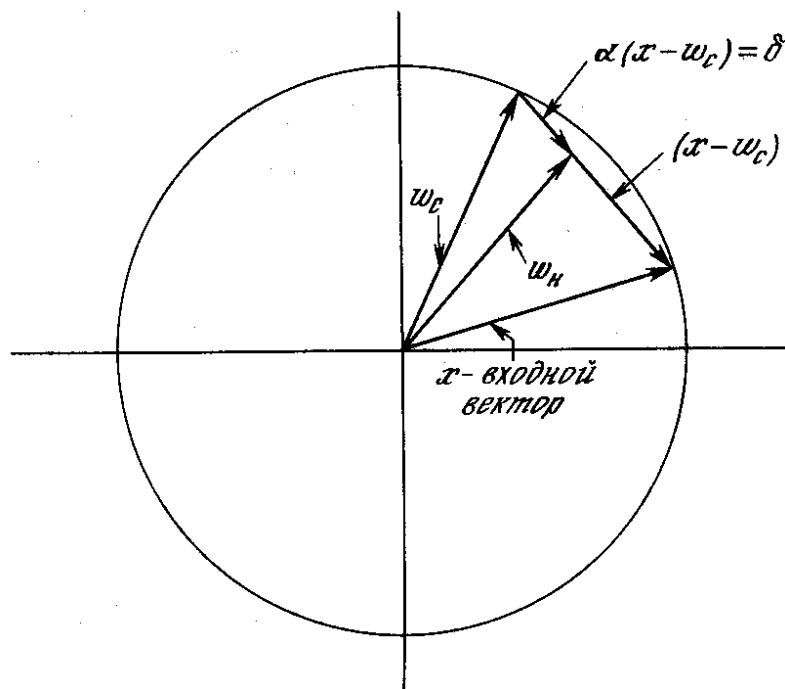


Рисунок 2. Вращение весового вектора в процессе обучения ( $W_n$  – вектор новых весовых коэффициентов,  $W_c$  – вектор старых весовых коэффициентов)

Переменная  $\alpha$  является коэффициентом скорости обучения, который вначале обычно равен  $\sim 0,7$  и может постепенно уменьшаться в процессе обучения. Это позволяет делать большие начальные шаги для быстрого грубого обучения и меньшие шаги при подходе к окончательной величине.

Если бы с каждым нейроном Кохонена ассоциировался один входной вектор, то слой Кохонена мог бы быть обучен с помощью одного вычисления на вес. Веса нейрона-победителя приравнялись бы к компонентам обучающего вектора ( $\alpha = 1$ ). Как правило, обучающее множество включает много сходных между собой входных векторов, и сеть должна быть обучена активировать один и тот же нейрон Кохонена для каждого из них. В этом случае веса этого нейрона должны получаться усреднением входных векторов, которые должны его активировать. Постепенное уменьшение величины  $\alpha$  уменьшает воздействие каждого обучающего шага, так что окончательное значение будет средней величиной от входных векторов, на которых происходит обучение. Таким образом, веса, ассоциированные с нейроном, примут значение вблизи «центра» входных векторов, для которых данный нейрон является «победителем».

Всем весам сети перед началом обучения следует придать начальные значения. Общепринятой практикой при работе с нейронными сетями является присваивание весам небольших случайных значений. При обучении слоя Кохонена случайно выбранные весовые векторы следует нормализовать. Окончательные значения весовых векторов после обучения совпадают с нормализованными входными векторами. Поэтому нормализация перед началом обучения приближает весовые векторы к их окончательным значениям, сокращая, таким образом, обучающий процесс.

Рандомизация весов слоя Кохонена может породить серьезные проблемы при обучении, так как в результате ее весовые векторы распределяются равномерно по поверхности гиперсферы. Из-за того, что входные векторы, как правило, распределены неравномерно и имеют тенденцию группироваться на относительно малой части поверхности гиперсферы, большинство весовых векторов будут так удалены от любого входного вектора, что они никогда не будут давать наилучшего соответствия. Эти нейроны Кохонена будут всегда иметь нулевой выход и окажутся бесполезными. Более того, оставшихся весов, дающих наилучшие соответствия, может оказаться слишком мало, чтобы разделить входные векторы на классы, которые расположены близко друг к другу на поверхности гиперсферы.

Допустим, что имеется несколько множеств входных векторов, все множества сходные, но должны быть разделены на различные классы. Сеть должна быть обучена активировать отдельный нейрон Кохонена для каждого класса. Если начальная плотность весовых векторов в окрестности обучающих векторов слишком мала, то может оказаться невозможным разделить сходные классы из-за того, что не будет достаточного количества весовых векторов в интересующей нас окрестности, чтобы приписать по одному из них каждому классу входных векторов.

Наоборот, если несколько входных векторов получены незначительными изменениями из одного и того же образца и должны быть объединены в один класс, то они должны включать один и тот же нейрон Кохонена. Если же плотность весовых векторов очень высока вблизи группы слегка различных входных векторов, то каждый входной вектор может активировать отдельный нейрон Кохонена.

## **ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ**

Применить нейронную сеть Кохонена с самообучением для задачи кластеризации. На первом этапе сгенерировать случайные точки на плоскости вокруг 2 центров

кластеризации (примерно по 20-30 точек). Далее считать, что сеть имеет два входа (координаты точек) и два выхода – один из них равен 1, другой 0 (по тому, к какому кластеру принадлежит точка). Подавая последовательно на вход (вразнобой) точки, настроить сеть путем применения описанной процедуры обучения так, чтобы она приобрела способность определять, к какому кластеру принадлежит точка. Коэффициент  $\alpha$  выбрать, уменьшая его от шага к шагу по правилу  $\alpha = (50-i)/100$ , причем для каждого нейрона это будет своё значение  $\alpha$ , а подстраиваться на каждом шаге будут веса только одного (выигравшего) нейрона.