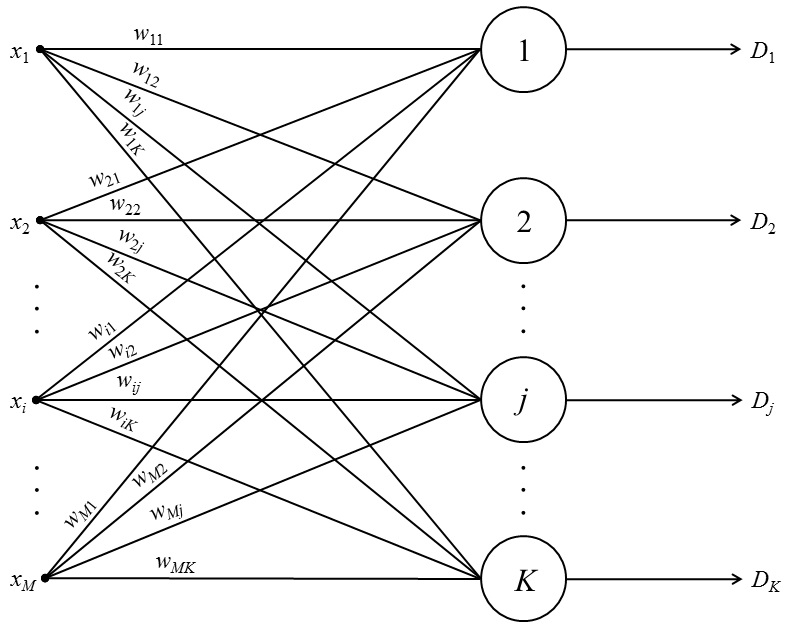
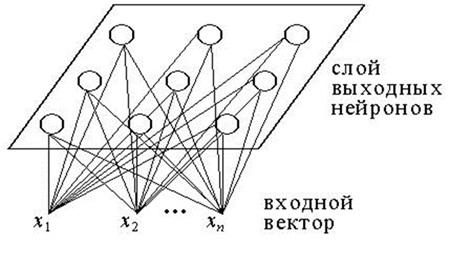
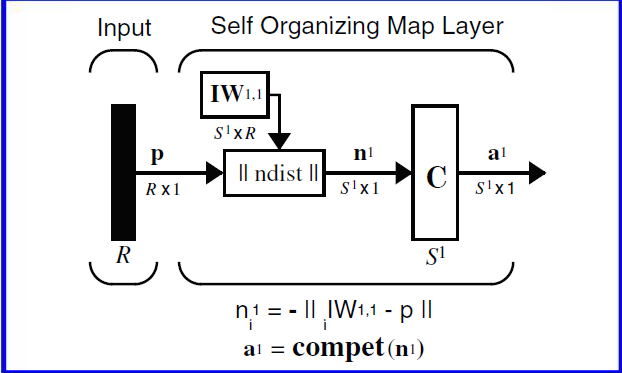
**Сеть Кохонена**







НС Кохонена обучается без учителя и решает задачу кластеризации. Структура нейронной сети содержит единственный слой нейронов (слой Кохонена) без коэффициентов смещения. Количество нейронов равно количеству кластеров, количество входных переменных нейронной сети равно числу признаков.

Суть обучения [**нейронной сети Кохонена**](http://www.aiportal.ru/articles/neural-networks/network-kohonen.html) заключается в такой подстройке весов, при которой близкие входные векторы будут активировать один и тот же нейрон Кохонена. Обучение слоя Кохонена является самообучением, протекающим без учителя. В связи с этим трудно заранее сказать, какой именно нейрон Кохонена будет активироваться заданным входным вектором. От процесса обучения требуется лишь, чтобы в результате обучения разделялись несхожие входные векторы. 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.

Алгоритм Кохонена:

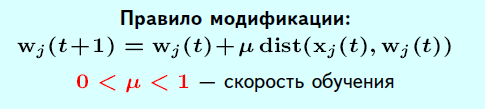
1) Инициализация. Устанавливаем начальные веса, скорость обучения и радиус обучения.

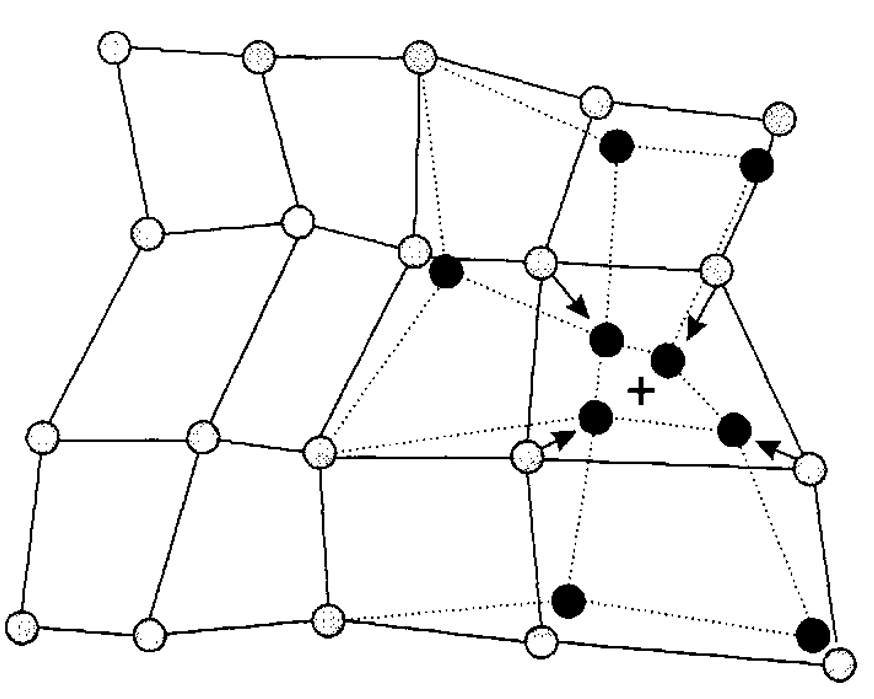
2) Возбуждение. Подается входной вектор.

3) Конкуренция. Рассчитывается расстояние между векторами весов всех нейронов выходного слоя и вектором входного воздействия. Тот нейрон, для которого расстояние окажется наименьшим, и будет нейроном-победителем.

4) Объединение. Определяются все нейроны, расположенные в пределах радиуса обучения относительно нейрона-победителя.

5) Подстройка. Производится модификация весов нейронов в пределах радиуса обучения.





**Карта Кохонена**

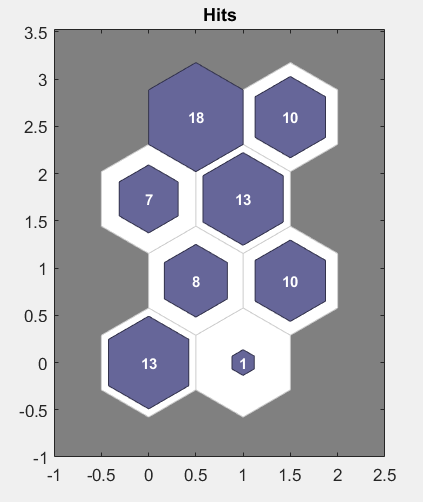
Чем отличаются понятия «сеть Кохонена» и «карта Кохонена»?

1) сеть Кохонена используется только для кластеризации объектов, а визуализация результатов будет производиться с помощью таблиц или диаграмм. Карта Кохонена позволяет визуализировать результаты кластеризации, в том числе многомерные.

2) в сети число выходных нейронов соответствует количеству кластеров, которое должно быть получено, а в карте — количеству сегментов, из которого будет состоять карта, или, иными словами, размеру карты.

Из вышесказанного можно сделать вывод, что карта Кохонена — это разновидность сети Кохонена.

Карта Кохонена состоит из сегментов прямоугольной или шестиугольной формы, называемых ячейками. Каждая ячейка связана с определенным выходным нейроном и представляет собой «сферу влияния» данного нейрона. Объекты, векторы признаков которых оказываются ближе к вектору весов данного нейрона, попадают в ячейку, связанную с ним. Следовательно, если объекты на карте расположены близко друг к другу, то и векторы признаков этих объектов близки, и наоборот.



**Задача Коммивояжера**

Алгоритм решения задачи следует из оригинальной схемы Кохонена, в которую вносятся лишь небольшие изменения. Используется сеть, состоящая из двух одномерных слоев нейронов (т.е. содержащая лишь один слой синаптических весов). Входной слой состоит из трех нейронов, а выходной - из *N* (по числу городов). Каждый нейрон входного слоя связан с каждым выходным нейроном. Все связи вначале инициируются случайными значениями. Для каждого города входной 3-мерный вектор формируется из двух его координат на плоскости, а третья компонента вектора представляет из себя нормирующий параметр, вычисляемый так, чтобы все входные вектора имели одинаковую Евклидову длину и никакие два вектора не были бы коллинеарны. Это эквивалентно рассмотрению двумерных координат городов, как проекций трехмерных векторов, лежащих на сфере. Обозначим через 3-мерный вектор синаптических связей, связывающих *j-*й выходной нейрон с входными нейронами. Если - трехмерный входной вектор, определяющий *i*-й город, то активация *j-го*выходного нейрона при подаче на вход определяется скалярным произведением (,). Выходной нейрон, для которого это произведение максимально, называется *образом* города.

Алгоритм формирования маршрута формулируется следующим образом. Выбираются значения для параметра усиления и радиуса взаимодействия *r*. Следующий цикл выполняется вплоть до выполнения условия .

1) Выбирается случайный город *с*.

2) Определяется номер образа города в выходном слое - .

3) Векторы связей , соединяющих нейрон , и всех его 2*r* близлежащих соседей справа и слева: *j = - r, - r +1, ..., , ..., + r - 1, + r*модифицируются следующим образом:

,

где - Евклидова норма вектора . Для устранения концевых эффектов слой выходных нейронов считается кольцевым, так что *N*-й нейрон примыкает к первому.

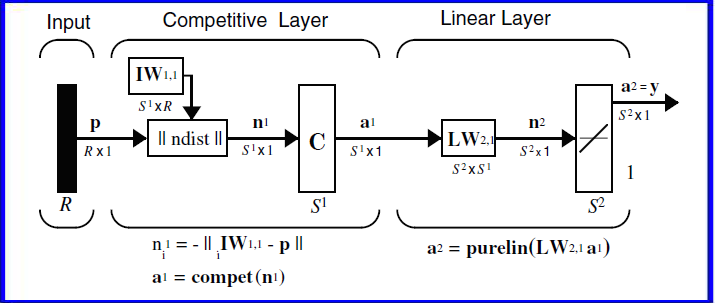
4) Радиус взаимодействия постепенно уменьшается согласно некоторому правилу (например, вначале можно положить , затем за первые 10% циклов снизить его до значения 1, которое далее поддерживается постоянным).

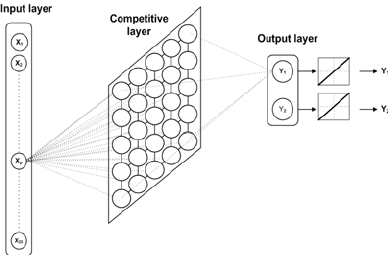
5) Параметр усиления постепенно снижается на небольшую величину ( например, в экспериментах Фавата и Уолкера он линейно уменьшался до нуля).

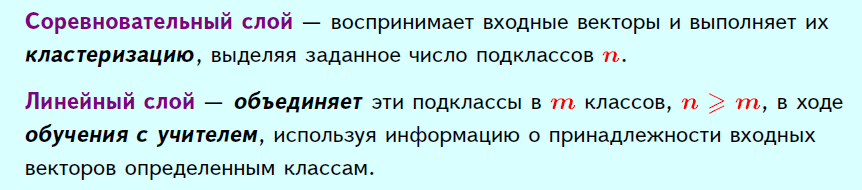
Конкретный вид законов изменения радиуса взаимодействия и параметра усиления, как правило, не имеет большого значения.

После завершения процесса обучения, положение города в маршруте определится положением его образа в кольцевом выходном слое. Иногда случается, что два или большее число городов отображаются на один и тот же выходной нейрон. Подобная ситуация может интерпретироваться так, что локальное упорядочивание этих городов не имеет значения и требует только локальной оптимизации части маршрута. При нескольких десятках городов такая оптимизация может скорректировать его длину на величину до 25%. Для сотен городов она, как правило, не улучшает результат и поэтому не используется.

**Сеть векторного квантования (LVQ)**







Алгоритм для соревновательного слоя:

