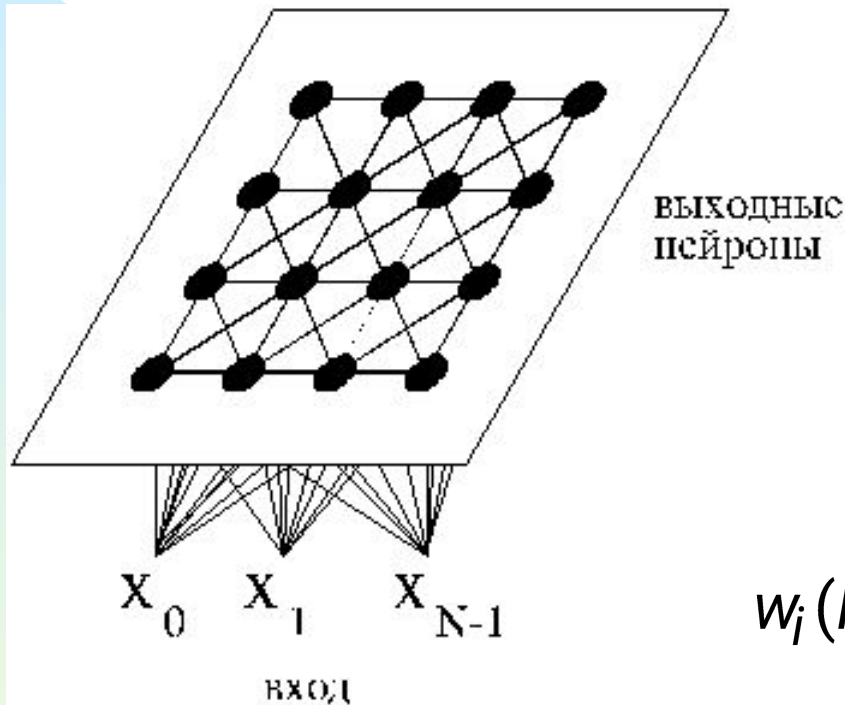




---

# Сеть Кохонена

# Сеть с самоорганизацией на основе конкуренции



$$w_i = (w_{i0}, w_{i1}, \dots, w_{i(N-1)})^T$$

$$d(x, w_j) = \min_{0 \leq i < N} d(x, w_i)$$

$$w_i(k+1) = w_i(k) + \eta_i(k)[x - w_i(k)]$$

Модификация весов по правилу Кохонена.

Адаптируются только веса, лежащие в некоторой окрестности нейрона-победителя.

# Меры расстояния между векторами

евклидова мера: 
$$d(x, w_i) = \|x - w_i\| = \sqrt{\sum_j (x_j - w_{ij})^2}$$

скалярное произведение: 
$$d(x, w_i) = 1 - \|x\| \|w_i\| \cos(\angle x, w_i)$$

Манхэттен: 
$$d(x, w_i) = \sqrt{\sum_j |x_j - w_{ij}|}$$

мера относительно нормы  $L_\infty$ : 
$$d(x, w_i) = \max_j |x_j - w_{ij}|$$

# Нормализация векторов

Необходимость в нормализации при небольшой размерности входных векторов.

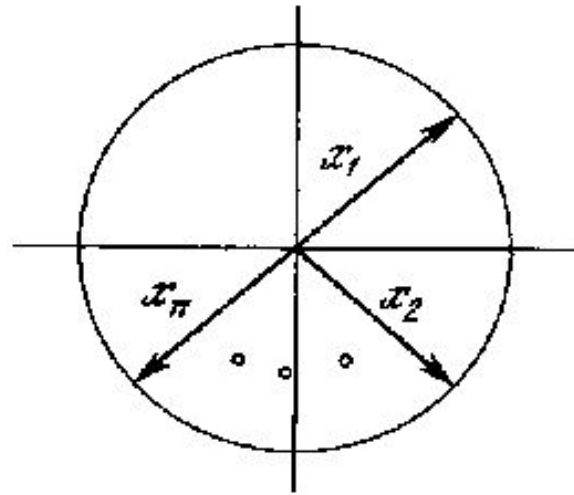
При  $N > 200$  эффект от нормализации уменьшается.

Способы нормализации:

1. 
$$x_i \leftarrow \frac{x_i}{\sqrt{\sum_i x_i^2}}$$

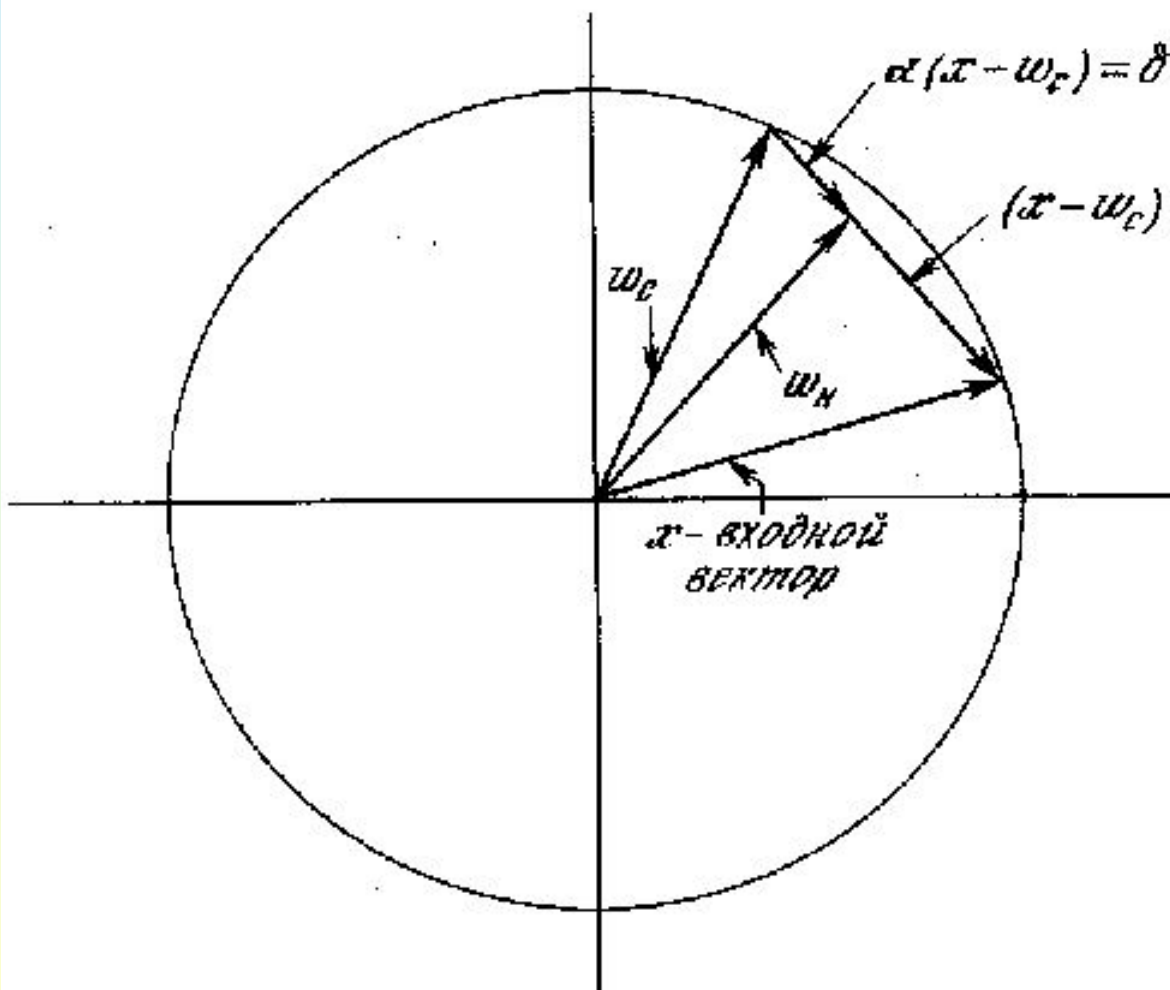
2. Увеличение размерности пространства на одну координату, чтобы:

$$\sum_{i=0}^N x_i^2 = 1$$

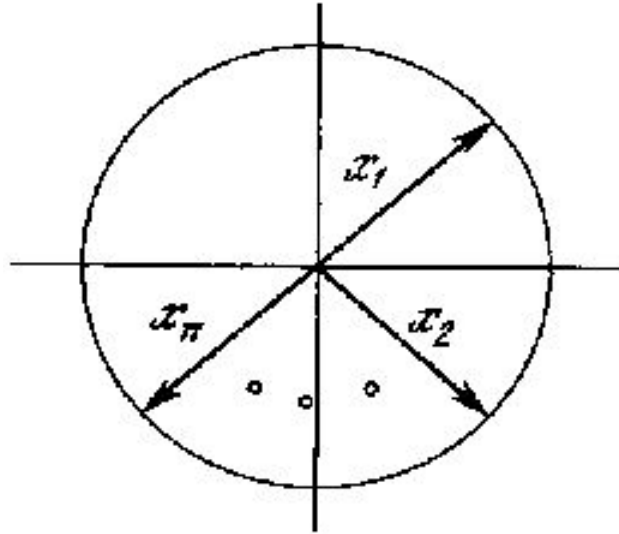


# Общий смысл обучения

$$w_H = w_C + \alpha[x - w_C]$$



# Инициализация весов



1. Инициализация случайным образом (с нормализацией).
2. Метод выпуклой комбинации:

$$w_{ij} = \frac{1}{\sqrt{N}}$$

$$x_j = \alpha x_j + \frac{1-\alpha}{\sqrt{N}}$$

3. Добавление шума к входным векторам.

# Проблема мертвых нейронов

1. Начальная адаптация всех весов.
2. Учет активности нейрона («чувство справедливости»)
  - контролирование частоты выигрыша каждого нейрона;
  - вычисление потенциала нейронов:

$$p_i(k+1) = \begin{cases} p_i(k) + \frac{1}{N} (i \neq h); \\ p_i(k) - p_{\min} (i = h) \end{cases}$$

$$p_{\min} \approx 0.75$$

- модификация расстояния между входным вектором и вектором весов:

$$d(x, w_i) \leftarrow N_i d(x, w_i)$$

# Алгоритмы обучения

$$E = \frac{1}{N} \sum_{i=1}^N \|x_i - w_{w(i)}\|^2$$

1. WTA с учетом активности нейронов (CWTA):

$$w_h = w_h + \eta [x - w_h]$$

2. WTM (Winner takes most):

$$w_i = w_i + \eta_i G(i, x) [x - w_i]$$

а) классический алгоритм Кохонена:

- функция  $G$  определяет прямоугольную окрестность;
- соседство гауссовского типа:

$$G(i, x) = e^{-\frac{d^2(i, w)}{2\lambda^2}}$$

б) алгоритм нейронного газа:

Сортировка векторов весов по расстояниям до входного вектора;

$G(i, x) = e^{-\frac{m(i)}{\lambda}}$ , где  $m(i)$  - номер нейрона  $i$  в полученной перестановке по расстояниям.

# Сравнение алгоритмов обучения

1. CWTА.
2. WTM - нейронный газ.
3. Классический алгоритм Кохонена.

# Применение сети Кохонена

## 1. Компрессия данных.

Кадр разбивается на части (входные вектора).

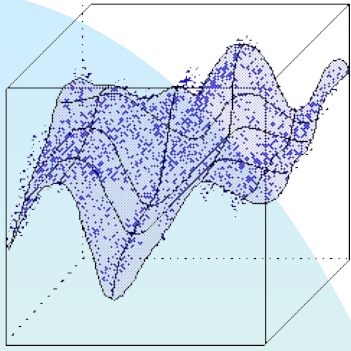
Каждой части сопоставляется нейрон-победитель.

Степень компресии изображений: 16.

## 2. Диагностирование неисправностей.

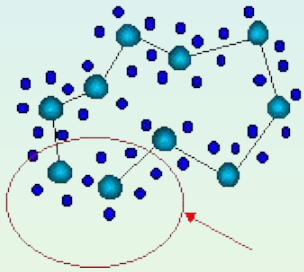
## 3. Визуализация многомерной информации.

# Визуализация многомерной информации

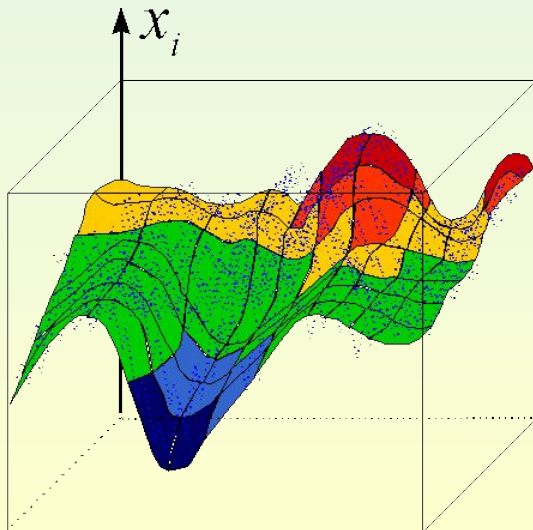


Двумерная топографическая карта набора трехмерных данных.

Каждая точка в трехмерном пространстве попадает в свою ячейку сетки имеющую координату ближайшего к ней нейрона из двумерной карты.



Свойство локальной близости: близкие на карте области близки в исходном пространстве. Наоборот в общем случае не верно.



Раскраска по  $i$ -му признаку.

Карты всех признаков образуют топографический атлас.

# Визуализация деятельности российских банков

## *Исходные данные.*

1800 банков.

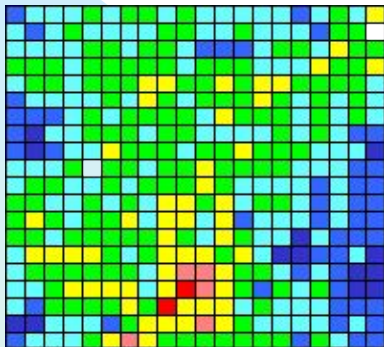
Каждый банк описывается 30 финансовыми показателями.

## *Задача*

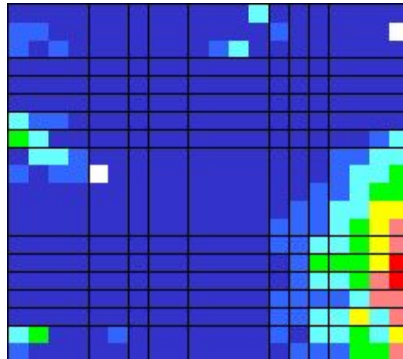
Построить 2 обобщенных параметра, представляющие данные о банках с минимальной потерей информации.



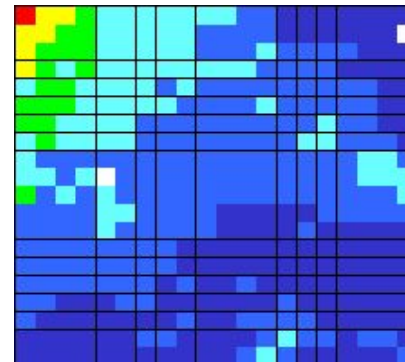
# Анализ карт Кохонена



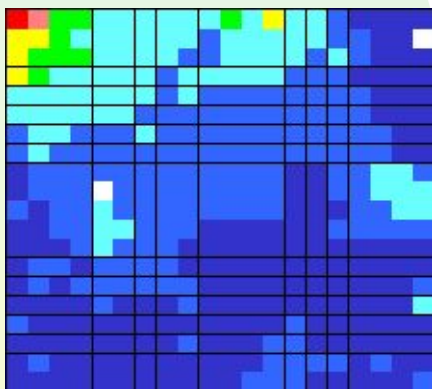
Размер активов



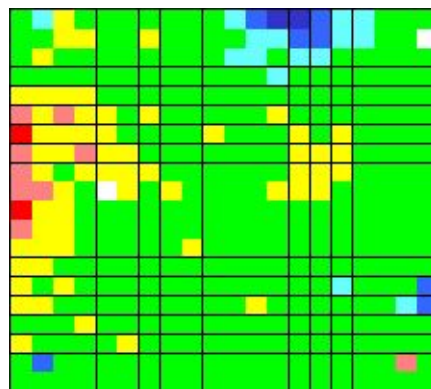
Уставной фонд  
(относительный  
размер)



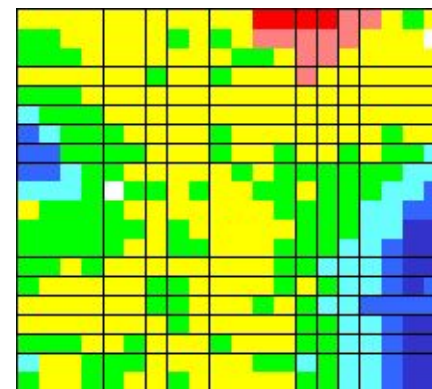
Всего доходов



Всего расходов



Прибыль/убыток



Всего обязательств

# Достоинства и недостатки

Достоинства:

1. Быстрое обучение;
2. Устойчивость к помехам.

Недостатки:

1. Не обладает свойствами аппроксиматора;
2. Решение задач кластеризации только при известном числе кластеров.