

2022 Корлякова М.О.

# Самоорганизация





# Самоорганизация. Свойства

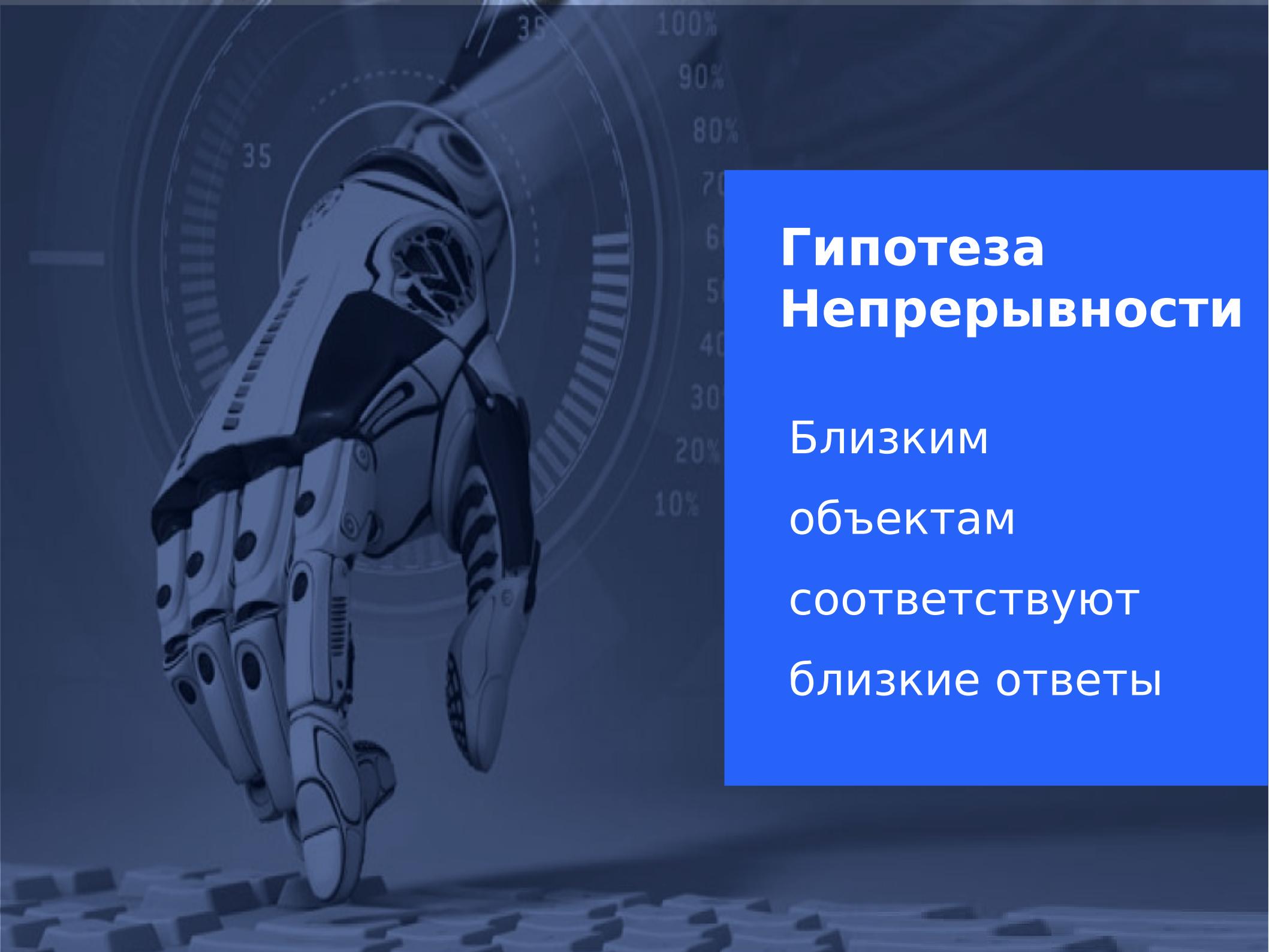
---

- Глобальный порядок определяется локальными взаимодействиями
- Изменение силы связей ведет к самоусилению системы
- Ограниченнное число ресурсов ведет к конкуренции.
- Изменение силы связей ведет к кооперации
- Описание образов избыточно

The background of the slide features a robotic hand reaching towards a circular dial. The dial has various scales and percentages (100%, 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20%, 10%) and numerical values (35, 30, 25, 20, 15, 10).

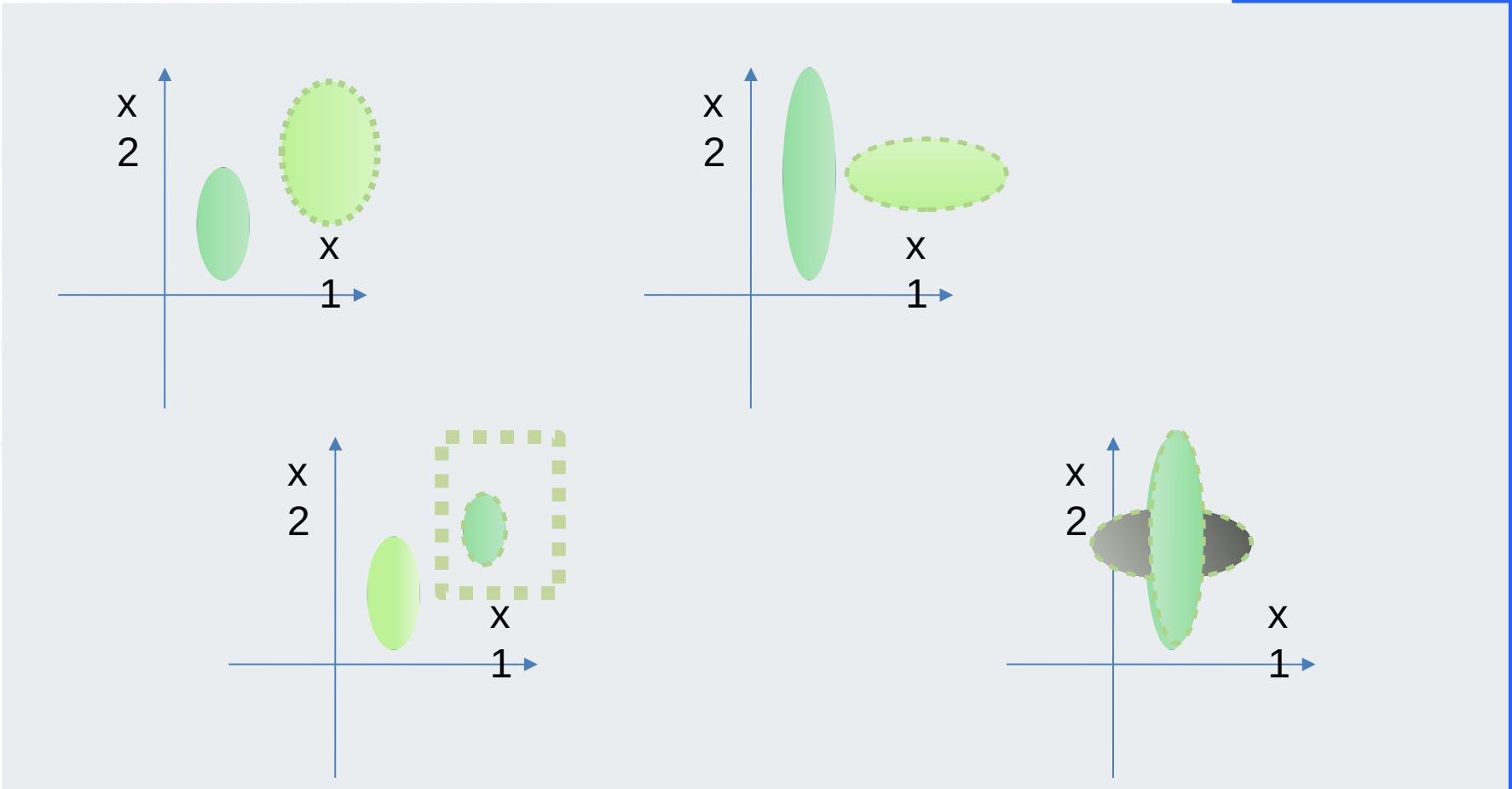
## Гипотеза Компактности

Похожие  
объекты близки  
друг к другу в  
пространстве  
признаков



# Гипотеза Непрерывности

Близким  
объектам  
соответствуют  
близкие ответы



**Как выглядит  
близость:**

# Расстояние между объектами

$$\rho(x, x_i) = \left( \sum_{j=1, n} w_j |x^j - x_i^j|^p \right)^{1/p}$$





# Меры расстояния

---

## евклидова мера

$$d(X, Wi) = |X - Wi|^2 = \sqrt{\sum[j=1:N]((x_j - w_{ij})^2)};$$

## скалярное произведение

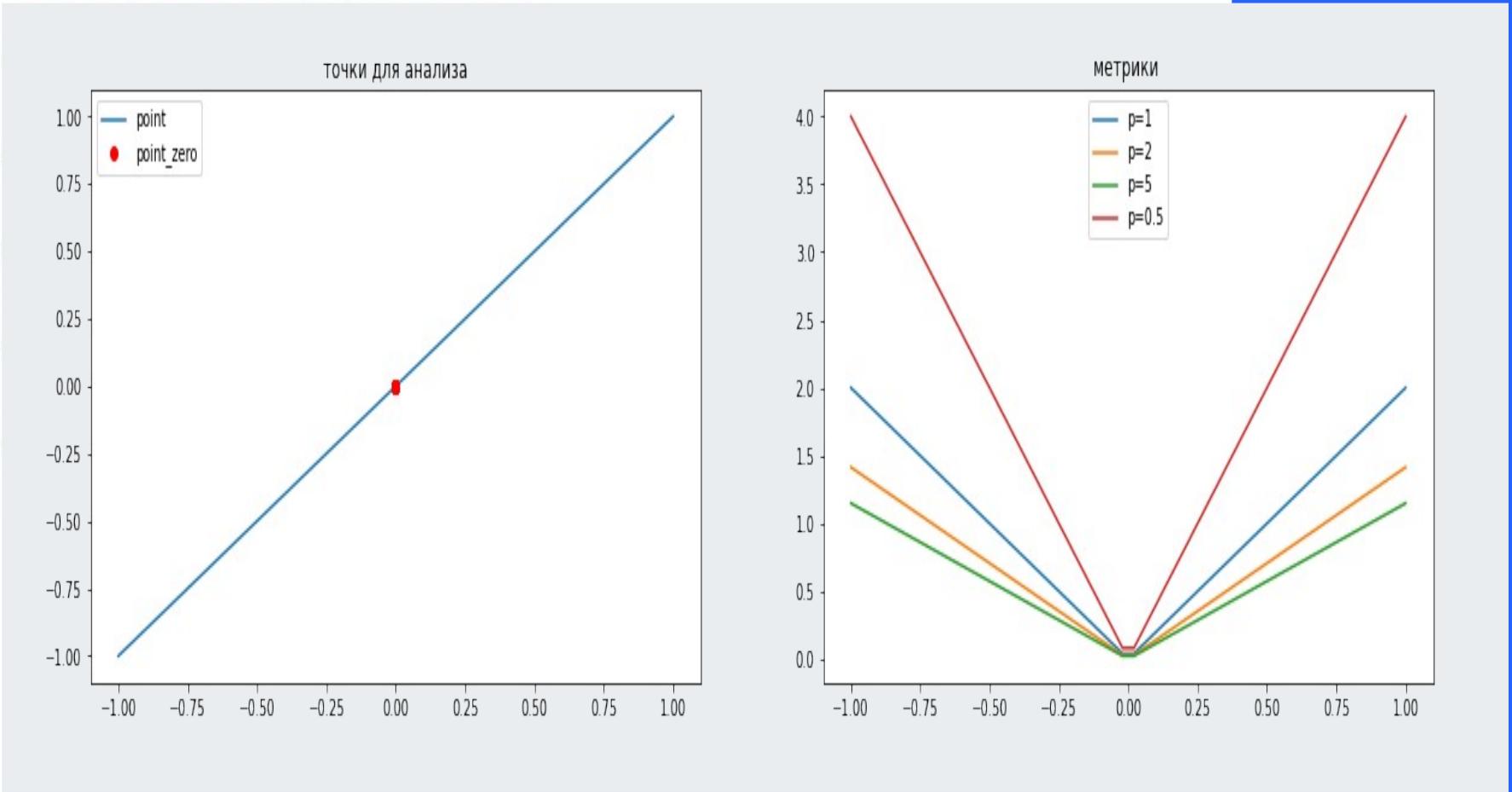
$$d(X, Wi) = 1 - \langle X | Wi \rangle = 1 - |X|^2 * |Wi|^2 * \cos(X, Wi);$$

## манхэттенское расстояние

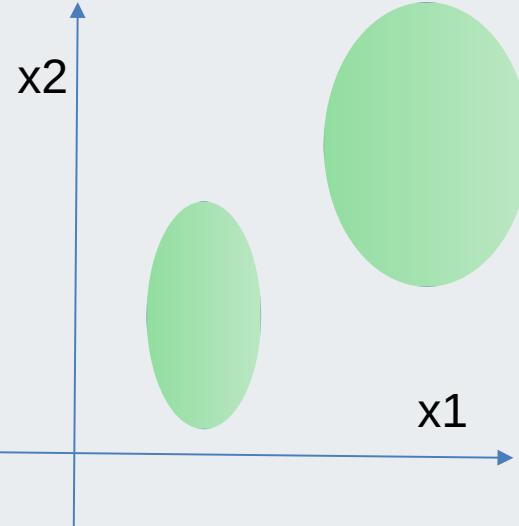
$$d(X, Wi) = \sum[j=1:N](|x_j - w_{ij}|);$$

## supremum-норма

$$d(X, Wi) = \max_j(|x_j - w_{ij}|).$$



# Влияние расстояния:



$$T = \{(X_i)\}, i = 1, N$$
$$\rho : X \times X \rightarrow [0, \infty)$$

$Y$

$a : X \rightarrow Y$

# Unsupervised Learning:

$$E = \min \sum_{X \in W_k} \frac{1}{|X_{wk}|} \rho(X, W_k)$$

$$E = \max \sum_{W_h W_k} \rho(W_h, W_k)$$

**Основная идея -  
Критерий:**



# Области применения:

---

- Упрощение сложных задач классификации;
- Сокращение объема хранения;
- Выделить нетипичные объекты.

# Проблемы:

- Нет точной постановки задачи
- Много критериев
- Много эвристик
- Результат зависит от метрики

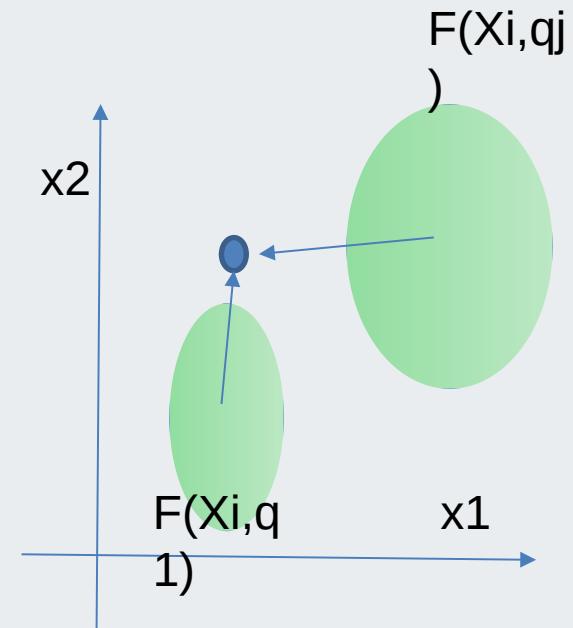


---

# Обучение как восстановление распределения

## Максимизация логарифма правдоподобия

$$L(w, q) = \ln \prod_{i=1, N} p(X_i) = \sum_{i=1, N} \ln \sum_{j=1, K} w_j F(X_i, q_j)$$



# ЕМ-алгоритм

Начальное приближение

Цикл:

1. Expectation

Оценка скрытых переменных

$$G = (g_{ij}), g_{ij} = P(j|x_i)$$

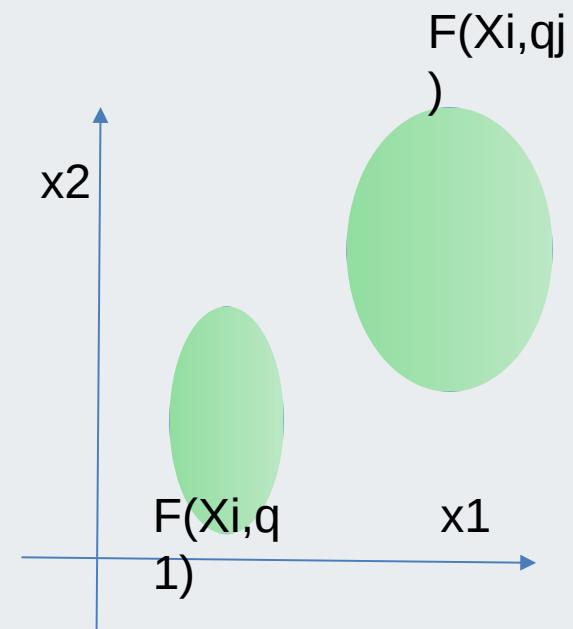
2. Maximization

максимизация правдоподобия

по отдельным компонентам

$$(w, q)$$

3. Stop-condition – пока нет  
стабилизации



# ЕМ-алгоритм

# ЕМ-алгоритм

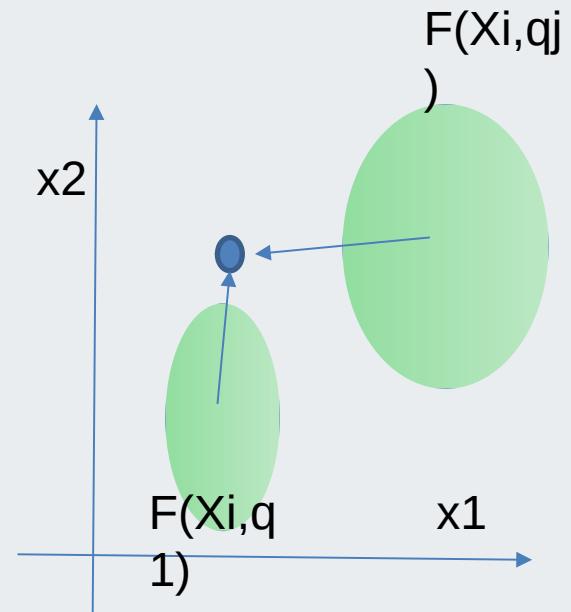
Максимизация логарифма  
правдоподобия

$$\ln P(X) = \sum_{i=1, N} \ln \left( \sum_{j=1, K} w_j F(X_i, q_j) \right)$$

$$g_i(j) = P(j | X_i)$$

$$\ln P(X) = \sum_{i=1, N} \ln \left( \sum_{j=1, K} \frac{g_i(j)}{g_i(j)} w_j F(X_i, q_j) \right)$$

$$\sum_{i=1, N} \ln \left( \sum_{j=1, K} \frac{g_i(j)}{g_i(j)} w_j F(X_i, q_j) \right) \geq \sum_{i=1, N} \sum_{j=1, K} g_i(j) \ln \left( \frac{1}{g_i(j)} w_j F(X_i, q_j) \right)$$



**Е-шаг:**  $L = \sum_{i=1, N} \sum_{j=1, K} g_i(j) \ln \left( \frac{1}{g_i(j)} w_j F(X_i, q_j) \right)$

# ЕМ-алгорит

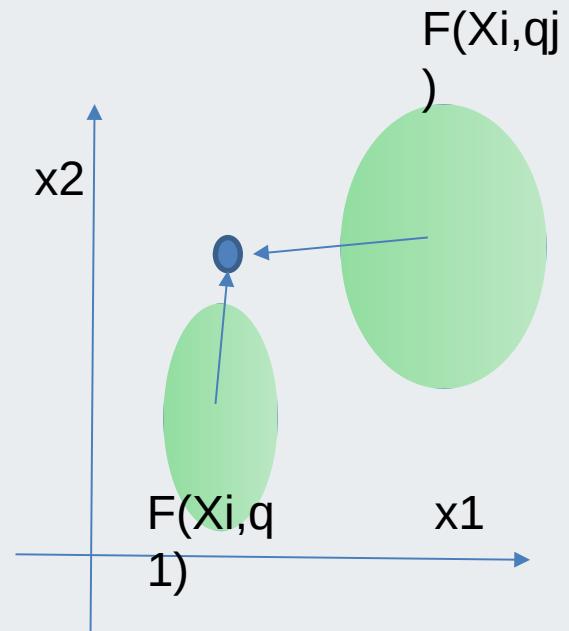
$$L = \sum_{i=1, N} \sum_{j=1, K} g_i(j) \ln \left( \frac{1}{g_i(j)} w_j F(X_i, q_j) \right)$$

$$\ln P(X) - L(w, q) = \sum_{i=1, N} \sum_{j=1, K} g_i(j) \ln \left( \frac{P(X_i) g_i(j)}{w_j F(X_i, q_j)} \right)$$

$$P(j | X_i) = \left( \frac{P(X_i)}{w_j F(X_i, q_j)} \right)^{-1}$$

$$\ln P(X) - L(w, q) = \sum_{i=1, N} \sum_{j=1, K} g_i(j) \ln \left( \frac{g_i(j)}{P(j | X_i)} \right)$$

$$\ln P(X) - L(w, q) = \sum_{i=1, N} E_{g_i(j)} \left[ \left( \frac{g_i(j)}{P(j | X_i)} \right) \right] = \sum_{i=1, N} KL(g_i(j) \| P(j | X_i))$$



**Е-шаг:**  $g_i(j) = P(j | X_i) = \left( \frac{P(X_i)}{w_j F(X_i, q_j)} \right)^{-1}$

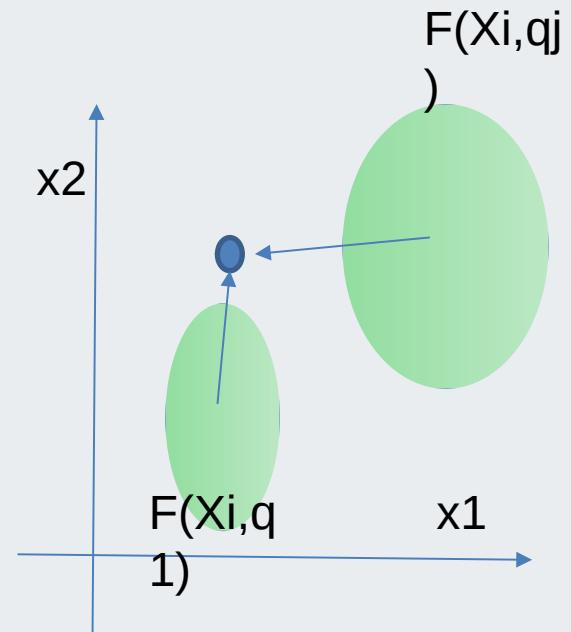
# ЕМ-алгорит

Оптимизация w,q

$$L = \sum_{i=1, N} \sum_{j=1, K} g_i(j) \ln \left( \frac{1}{g_i(j)} w_j F(X_i, q_j) \right)$$

$$L = \sum_{i=1, N} \sum_{j=1, K} g_i(j) \ln(w_j F(X_i, q_j)) - \sum_{i=1, N} \sum_{j=1, K} g_i(j) \ln(g_i(j))$$

$$\max_{w, q} \left( \sum_{i=1, N} \sum_{j=1, K} g_i(j) \ln(w_j F(X_i, q_j)) \right)$$



**М-шаг:**

$$w_j = \frac{1}{N} \sum_{i=1, N} g_i(j)$$

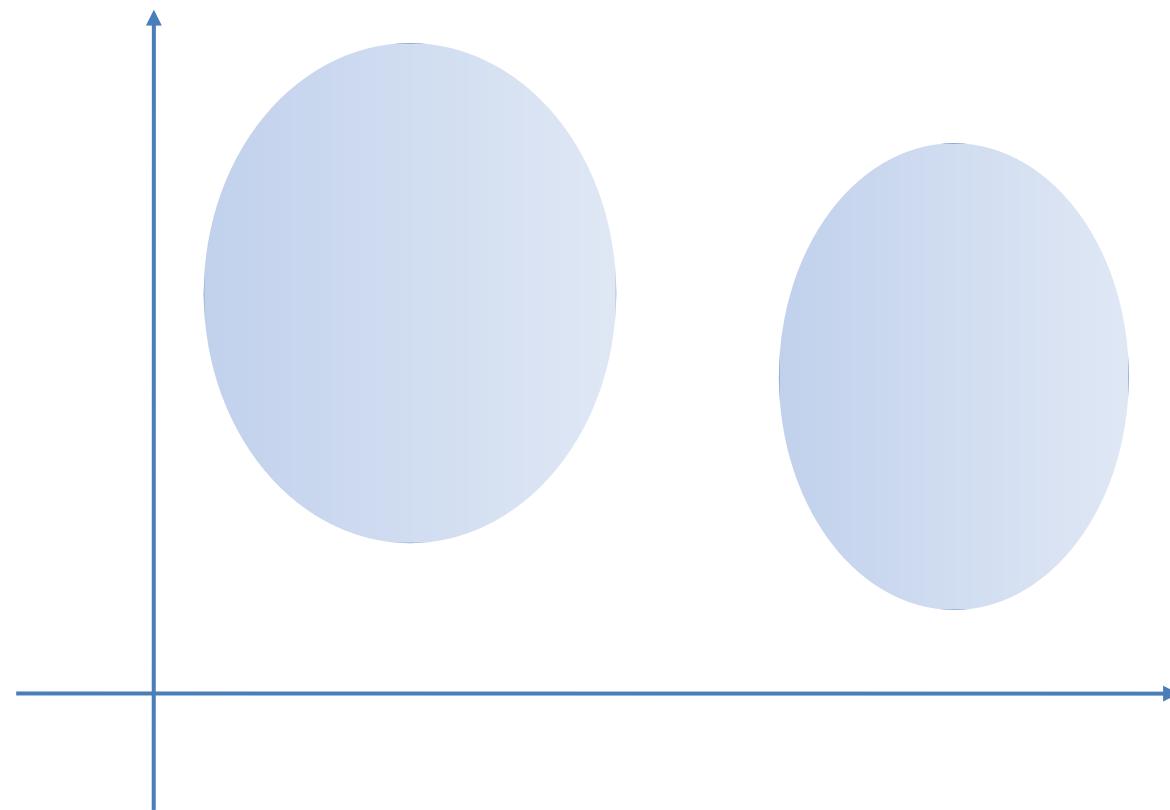
$$q = \arg \max \sum_{i=1, N} g_i(j) \ln(F(X_i, q_j))$$



---

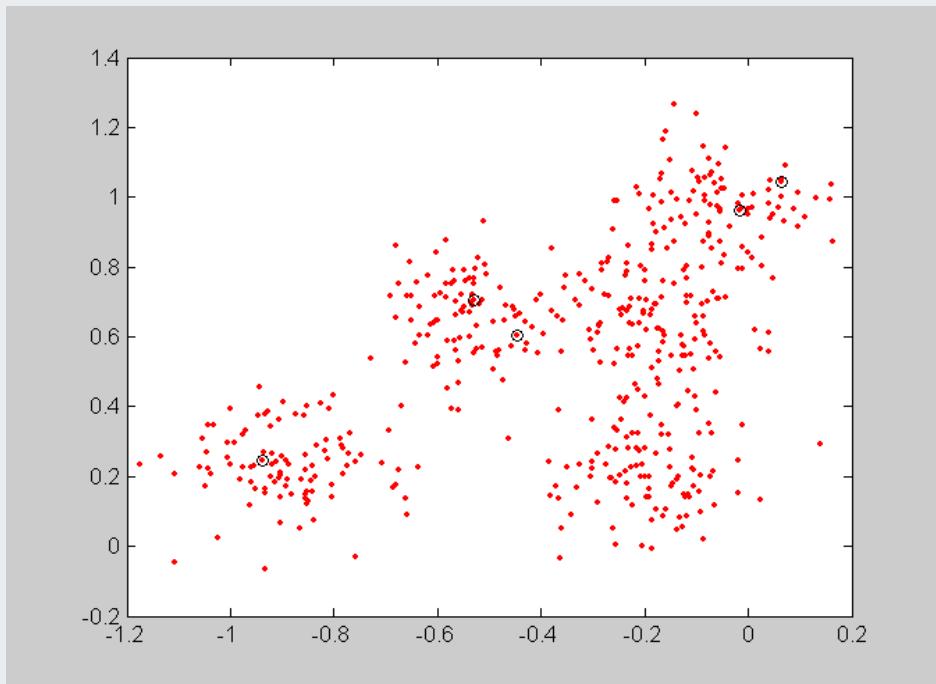
# Реализация обучения без учителя

# K-means

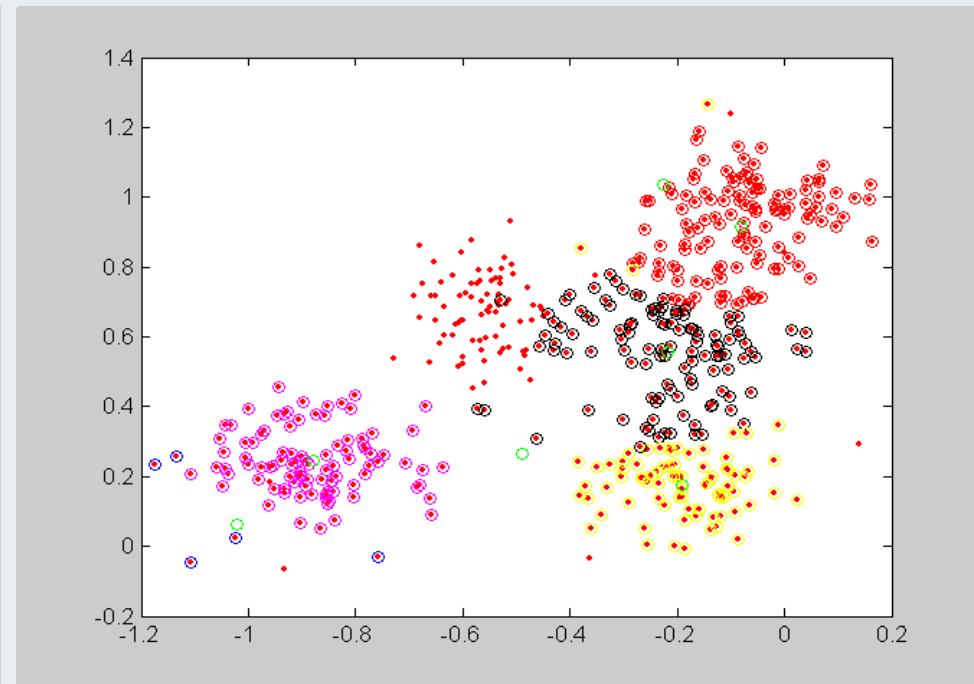


# Пример

Исходное множество



Результат



# Алгоритм к-средних

1. Номер итерации  $s=0$
2. Связать с каждым кластером  $K_j$  объект  $X_i$  из обучающей выборки (случайно).
3. если число кластеров меньше  $N$ , то перейти к процедуре формирования кластеров (п.4.).
4. Вычислить расстояния от всех объектов до всех центров кластеров.
5. присоединить объект  $X_i$  к кластеру  $C_k$ , если
6. повторить для всех объектов выборки.
7. вычислить новое положение центров кластеров
8. где  $N_b$  – число примеров множества  $C_b$ .
9. Повторять от п.4. пока кластер смещается более чем на  $\delta$  (задано пользователем), иначе остановить процесс.



## **Достоинства алгоритма k- средних:**

---

- простота использования;
- быстрота использования;
- понятность и прозрачность алгоритма.

# Недостатки алгоритма k- средних:

- Чувствителен к выбросам, которые могут искажать среднее
- Может медленно работать на больших базах данных



# Принцип

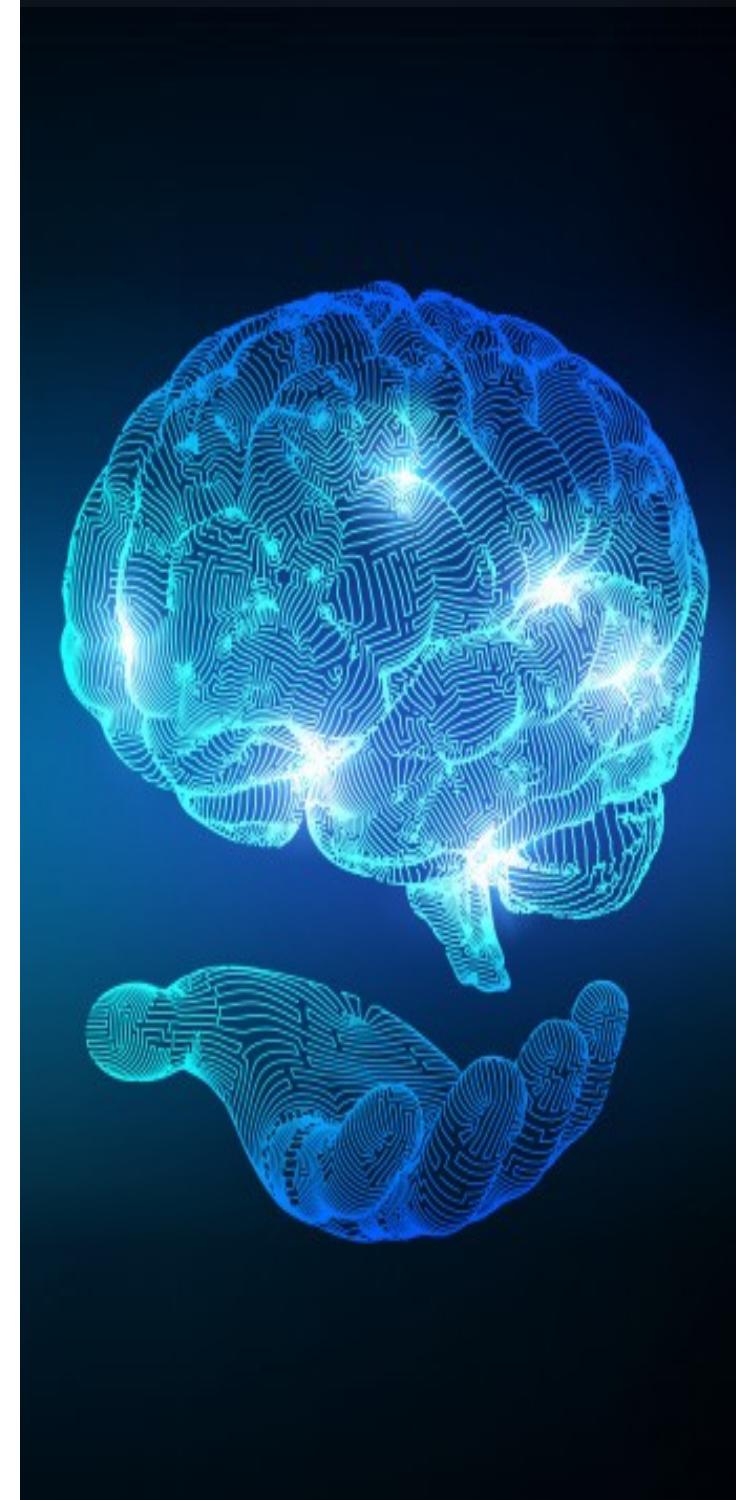
---

**Обучение.** Самоорганизация на основе конкуренции.

**Обобщение.** Ближайший нейрон

Использует специальные функции преобразования для разложения входного образа.

SOM – Self-Organizing Maps

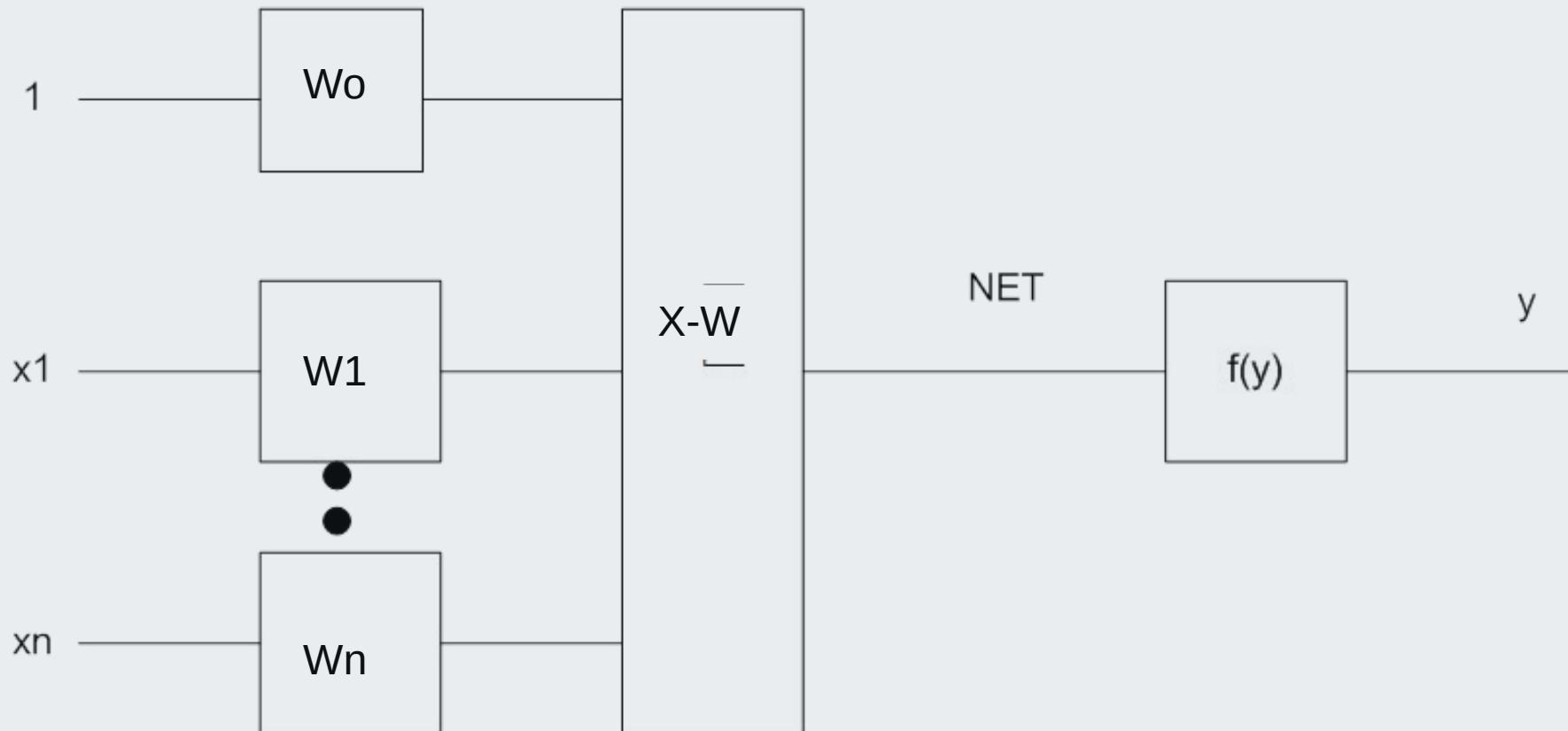


# Нейрон WTA

Победитель получает все.

Winner Take All

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot (x_j^k - w_{ij}(t))$$

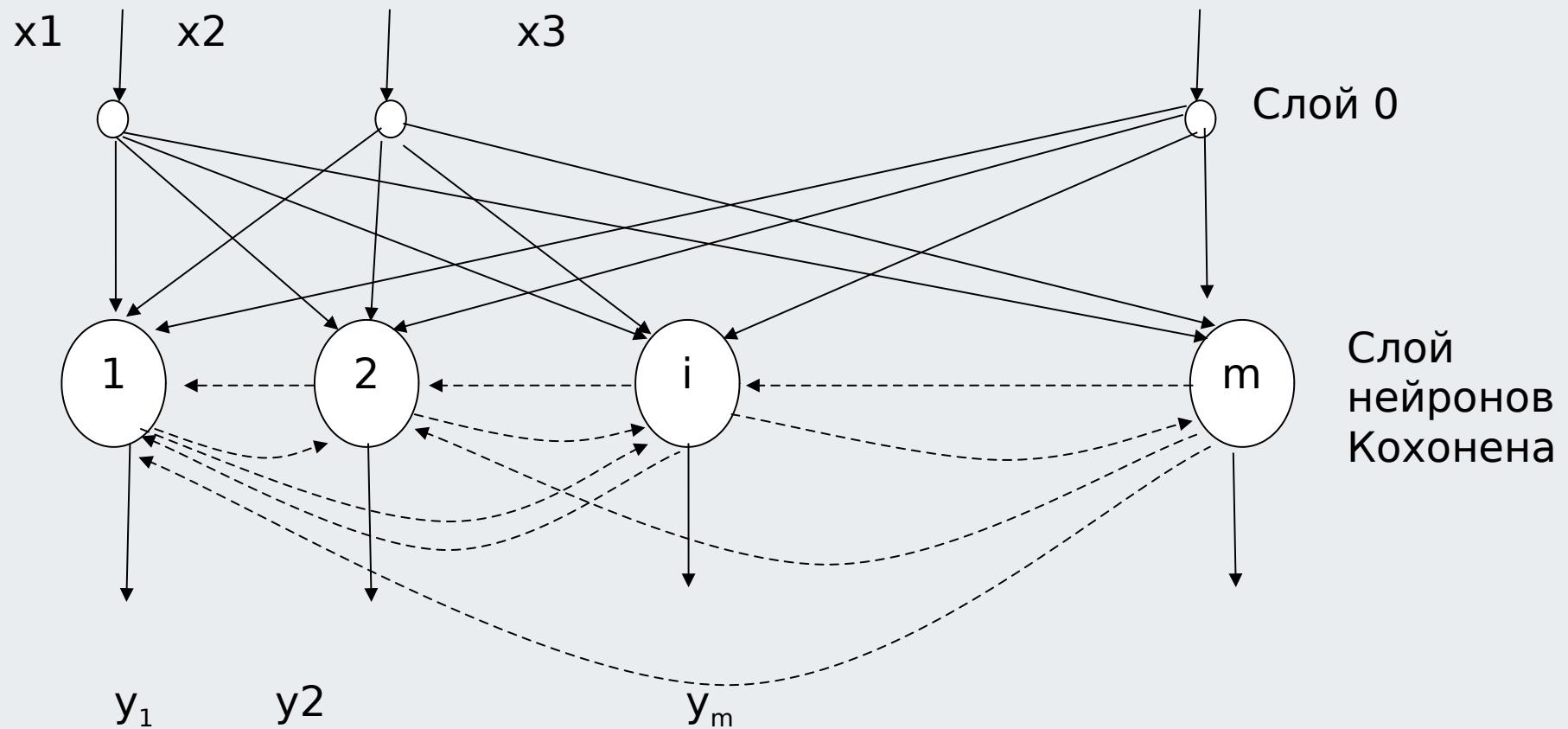


## Модель сети Кохонена

Сеть Кохонена - Алгоритм векторного кодирования.

Реализует топологическое отображение, которое оптимально размещает фиксированное количество векторов в входном пространстве высокой размерности.

# Архитектура сети SOM Кохонена





# Модель сети Кохонена

---

Один слой.

Нейроны Кохонена (РБФ).

Горизонтальные связи.

Обратных связей нет.

Обучение без учителя.

Обучение по алгоритмам соревнования.

Задача – кластеризация.

Число входов = размеру образца.

Число выходов = числу нейронов.

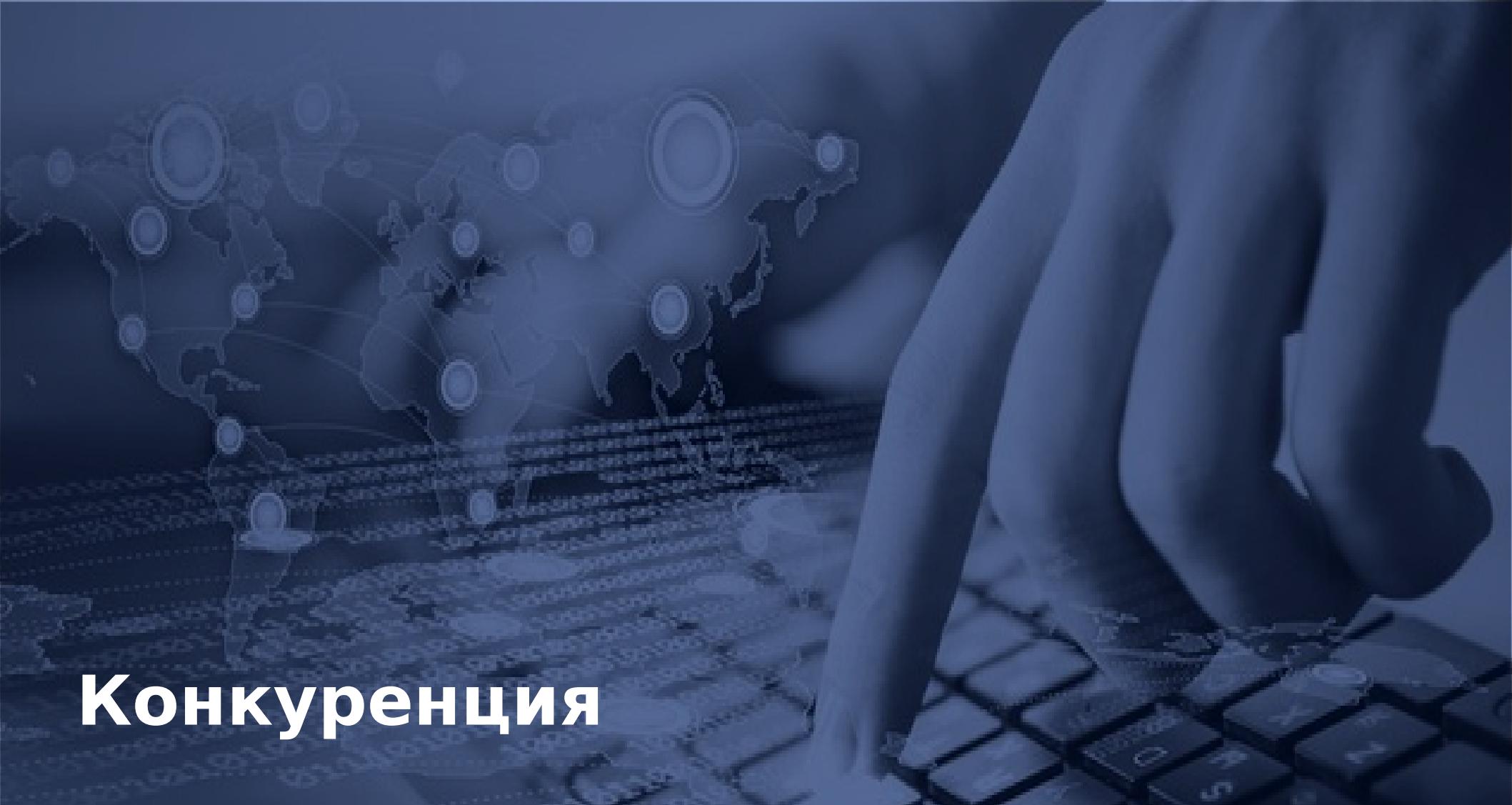
## Минимизация погрешности квантования

$$E_q = \frac{1}{p} \cdot \sum_{k=1}^p d(\mathbf{X}^k, \mathbf{W}_{w(k)}) ,$$

# Цель обучения

## **Формирование сети Кохонена**

- Инициализация
- Конкуренция (выбор нейрона победителя)
- Кооперация (для победителя найти его окрестность)
- Синаптическая адаптация (изменяем победителя и его окрестность)



## Конкуренция

$$l(x) = \arg \min \| X - W_j \|, \quad j=1, \dots, p$$

p - число нейронов

# Кооперация

Латеральное взаимодействие

$h_{ji}$  – топологическая окрестность с центром в  $i$ .

$d_{ji}$  – латеральное расстояние от нейрона  $I$  к нейрону  $j$

$(d_{ji}^2 = || r_j - r_i ||^2, r_i$  – позиция нейрона  $i$  )

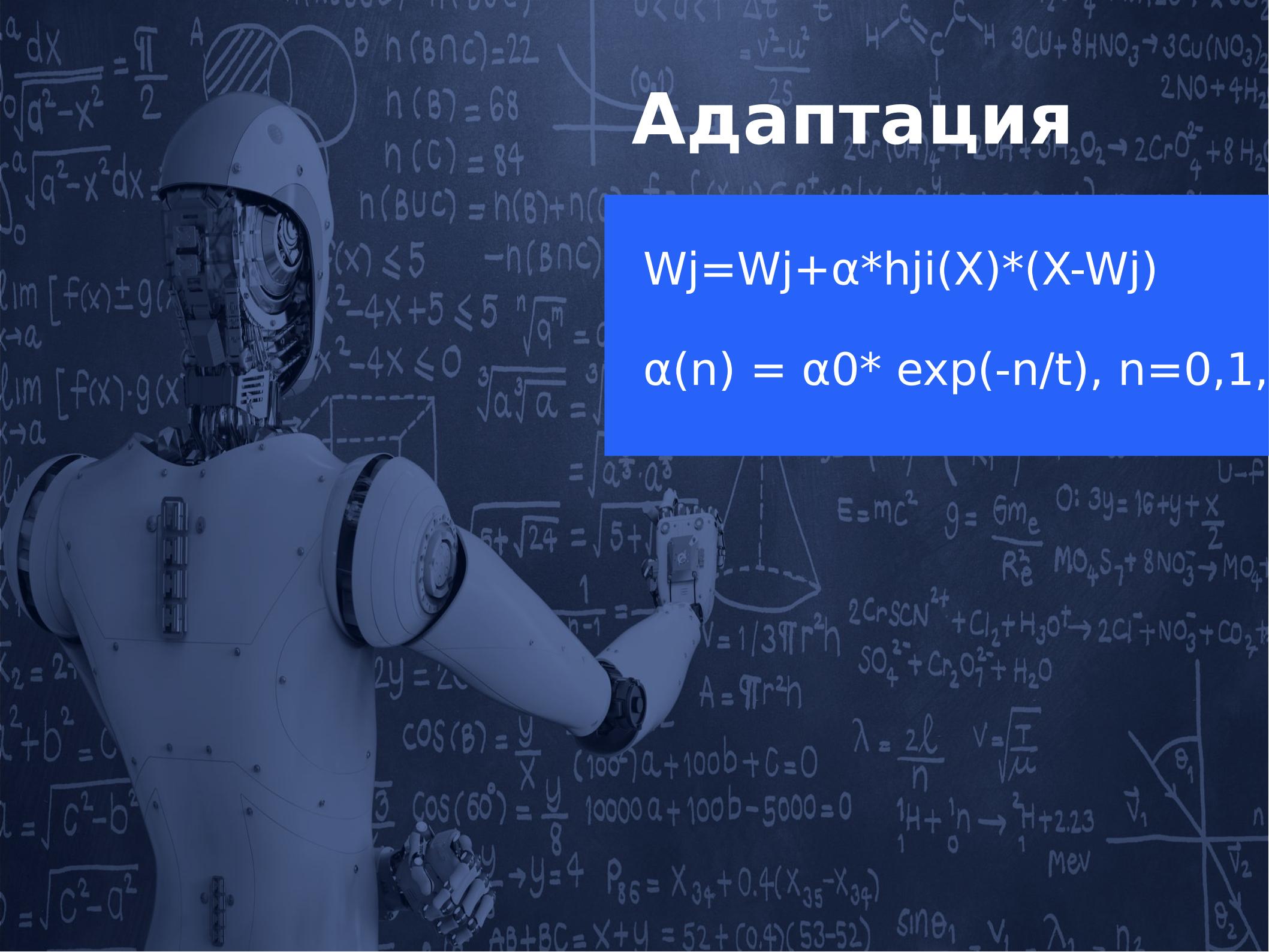
$h_{ji}(X) = \exp(-d_{ji}^2 / (2 * g^2))$

$g$ -эффективная ширина топологической окрестности

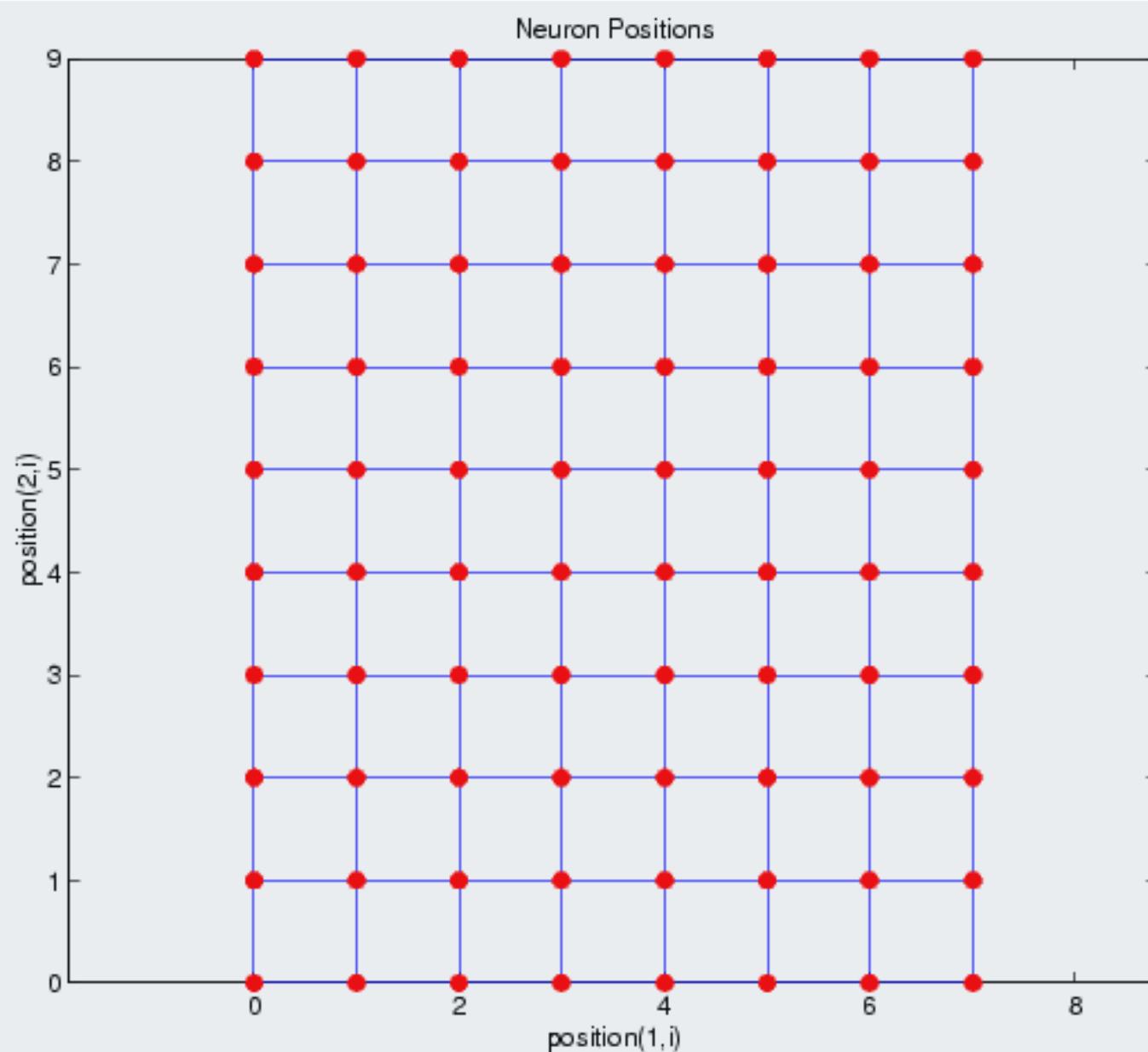
# Адаптация

$$W_j = W_j + \alpha * h_{ji}(X) * (X - W_j)$$

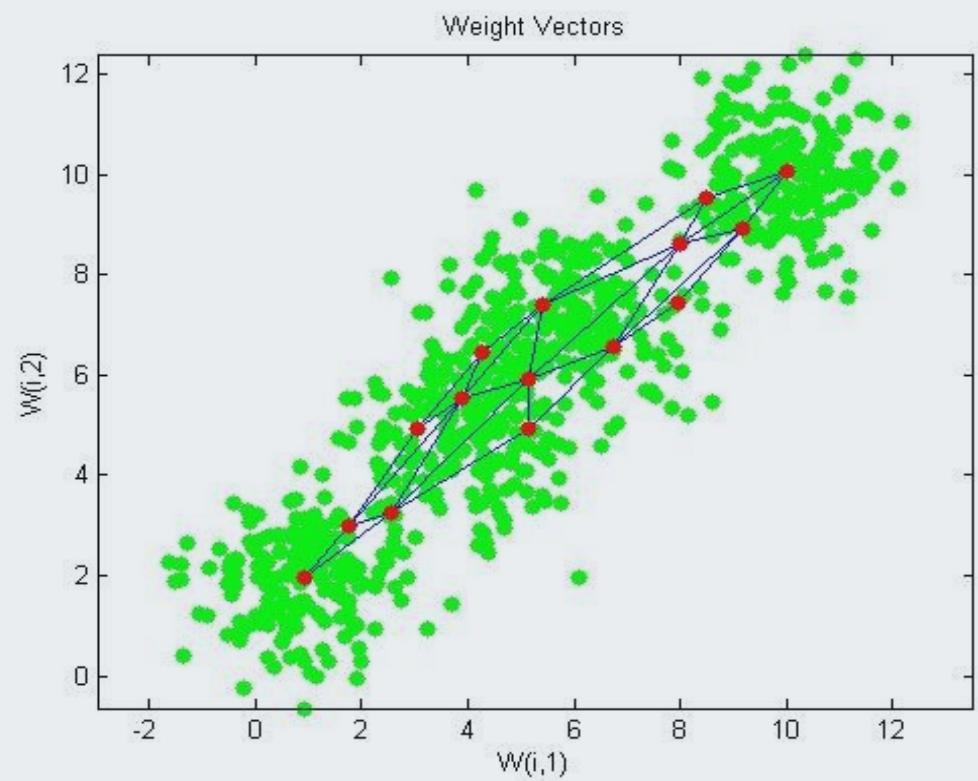
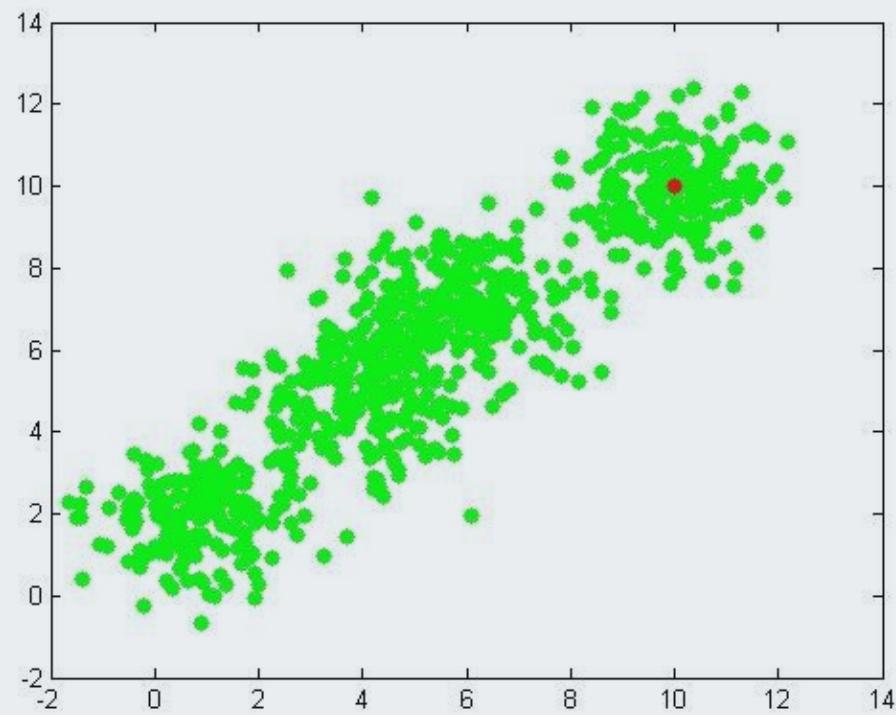
$$\alpha(n) = \alpha_0 * \exp(-n/t), n=0,1,$$



# Двухмерная сеть

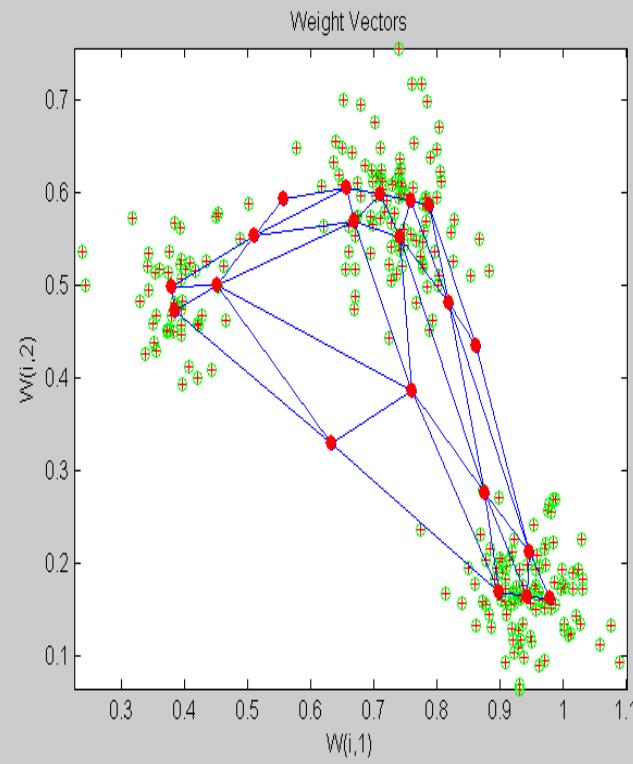


# Развитие сети SOM

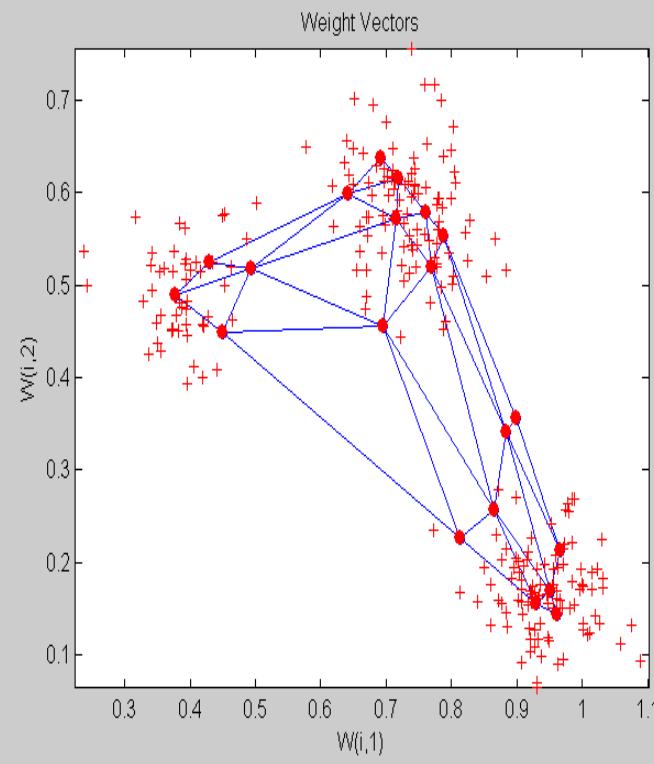


# Число эпох самоорганизации

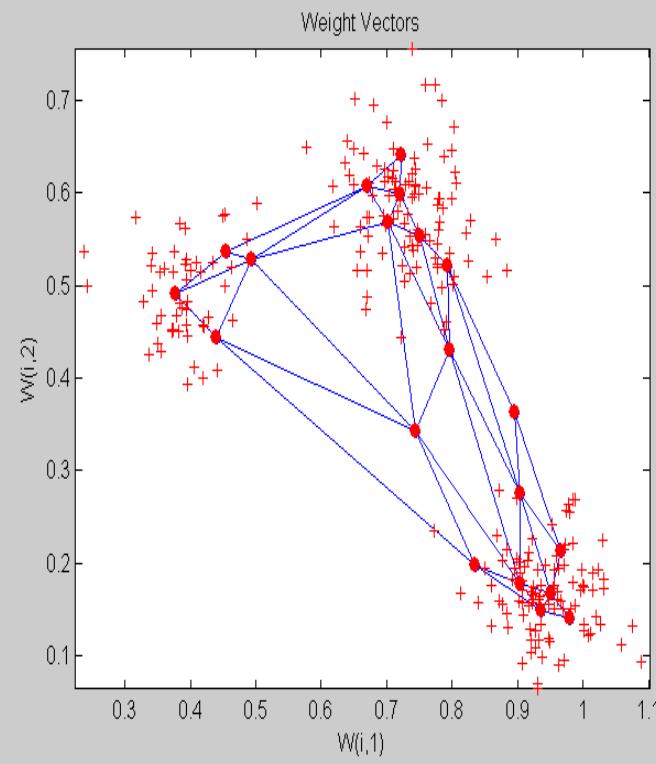
10



20



30





---

# RBM Restricted Boltzmann machine

# Идея РВМ

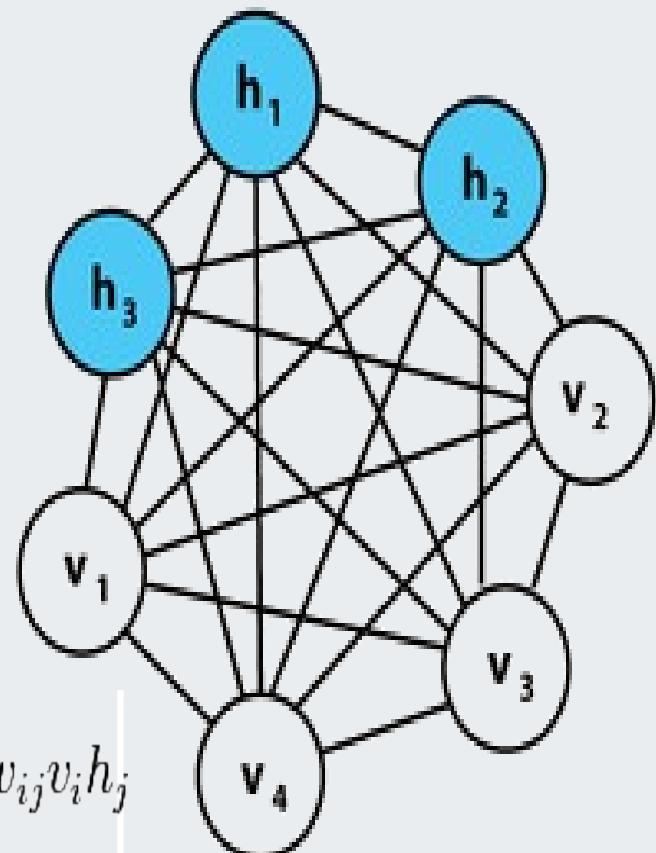
1. Хопфилд  $E = -\frac{1}{2} \sum_{ij} w_{ij} s_i s_j + \sum_i b_i s_i$

2. Обыкновенная Машина Больцмана

$$E = - \sum_{i < j} w_{ij} s_i s_j - \sum_i b_i s_i$$

3. Ограниченнная Машина Больцмана

$$E(v, h) = - \sum_i^n a_i v_i - \sum_j^m b_j h_j - \sum_i^n \sum_j^m w_{ij} v_i h_j$$



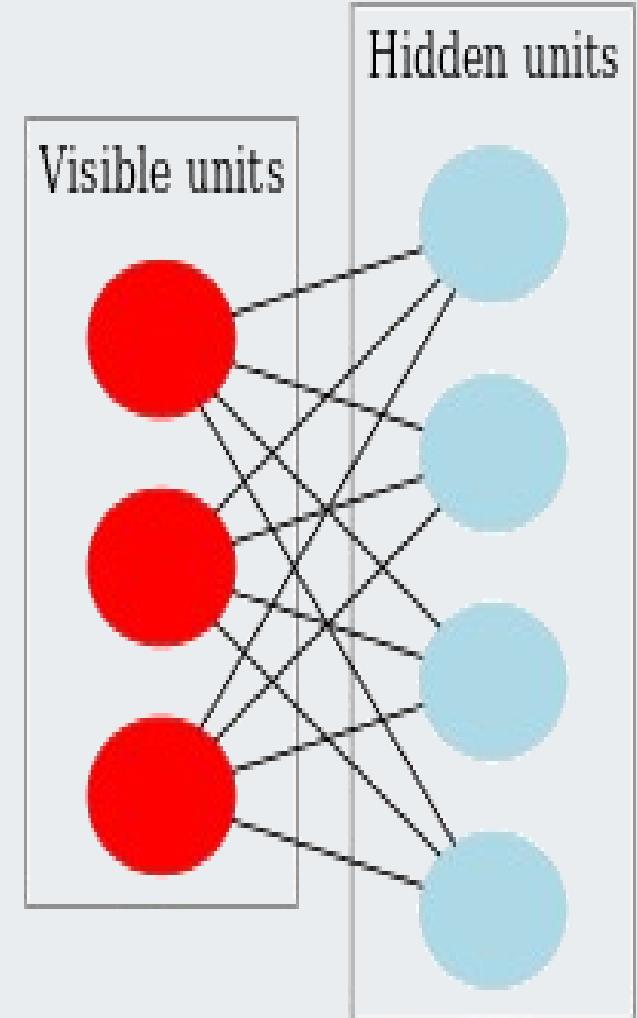
# Идея РВМ

$$E(v, h) = - \sum_i^n a_i v_i - \sum_j^m b_j h_j - \sum_i^n \sum_j^m w_{ij} v_i h_j$$

$$p(v, h) = \frac{1}{Z} e^{-E(v, h)}$$

$$Z = \sum_r^N \sum_t^M e^{-E(v^{(r)}, h^{(t)})}$$

$$p(v) = \sum_t^M P(v, h^{(t)}) = \frac{1}{Z} \sum_t^M e^{-E(v, h^{(t)})}$$

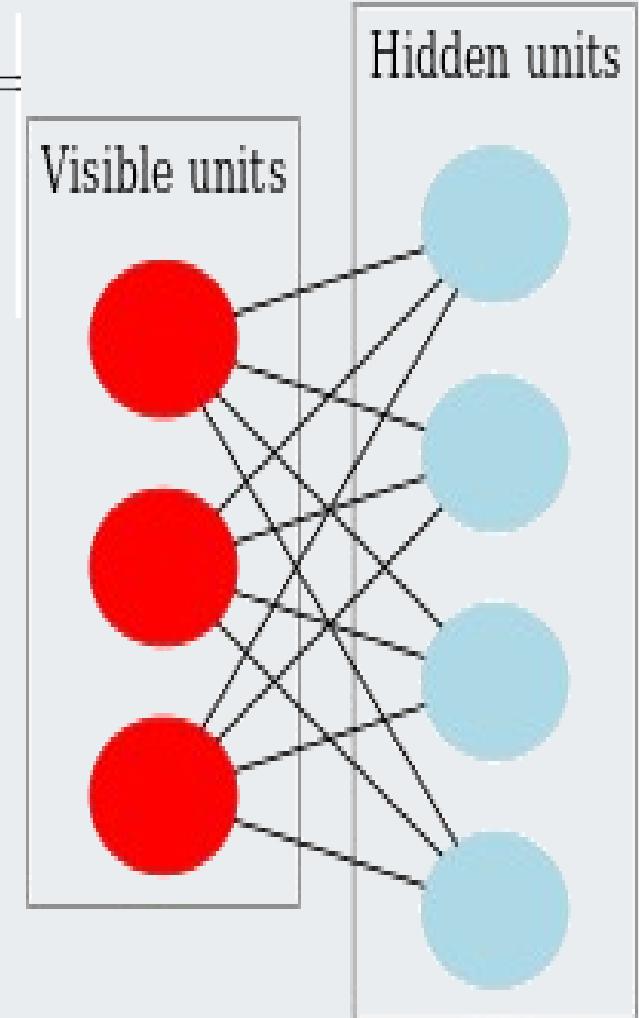


<https://habr.com/ru/post/159909/>

# Идея РВМ

$$P(h_k = 1|v) = \frac{e^{-E_1}}{e^{-E_1} + e^{-E_0}} = \frac{1}{1 + e^{E_1 - E_0}} = \frac{1}{1 + e^{-b - \sum_i^n v_i w_{ik}}} = \\ \sigma \left( -b - \sum_i^n v_i w_{ik} \right)$$

$$P(h|v) = \prod_j^m P(h_j|v)$$



<https://habr.com/ru/post/159909/>

# Идея РВМ

$$\frac{\partial \ln P(v^{(k)})}{\partial w_{ij}} = \frac{1}{P(v^{(k)})} \frac{\partial P(v^{(k)})}{\partial w_{ij}}$$

$$f(E(v, h^{(t)})) = e^{-E(v, h^{(t)})}$$

$$\hat{E} = E(v, h^{(t)})$$

$$P(v) = \frac{f(\hat{E})}{Z}$$

$$\frac{\partial P(v^{(k)})}{\partial w_{ij}} = \frac{Z \cdot \frac{\partial f(\hat{E})}{\partial w_{ij}} - \frac{\partial Z}{\partial w_{ij}} \cdot f(\hat{E})}{Z^2}$$

$$\frac{\partial f(\hat{E})}{\partial w_{ij}} = \frac{\partial f(\hat{E})}{\partial \hat{E}} \frac{\partial \hat{E}}{\partial w_{ij}}$$

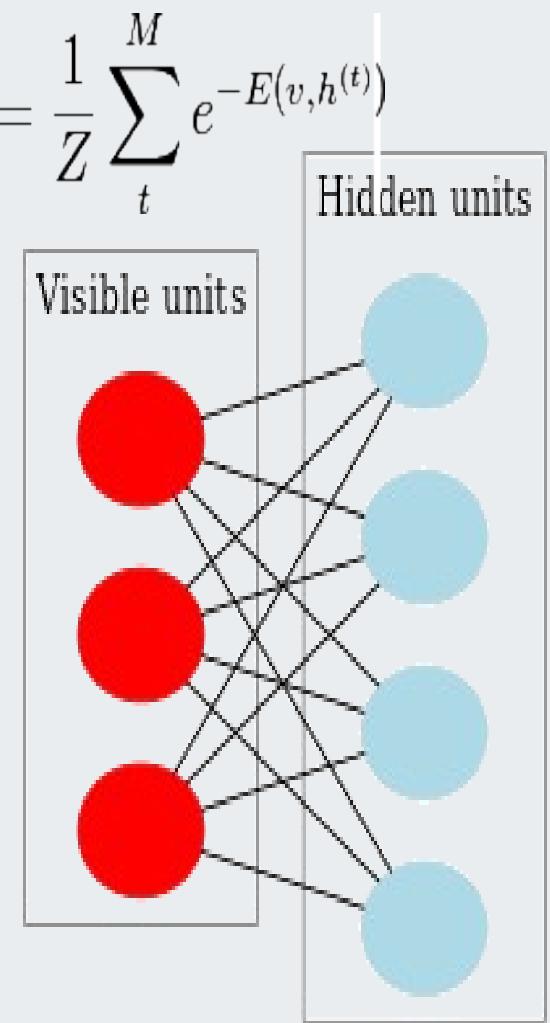
$$\frac{\partial Z}{\partial w_{ij}} = \frac{\partial Z}{\partial \hat{E}} \frac{\partial \hat{E}}{\partial w_{ij}}$$

$$p(v) = \sum_t^M P(v, h^{(t)}) = \frac{1}{Z} \sum_t^M e^{-E(v, h^{(t)})}$$

$$\frac{\partial E(v, h)}{\partial w_{ij}} = -v_i h_j$$

$$\frac{\partial E(v, h)}{\partial a_i} = -v_i$$

$$\frac{\partial E(v, h)}{\partial b_j} = -h_j$$



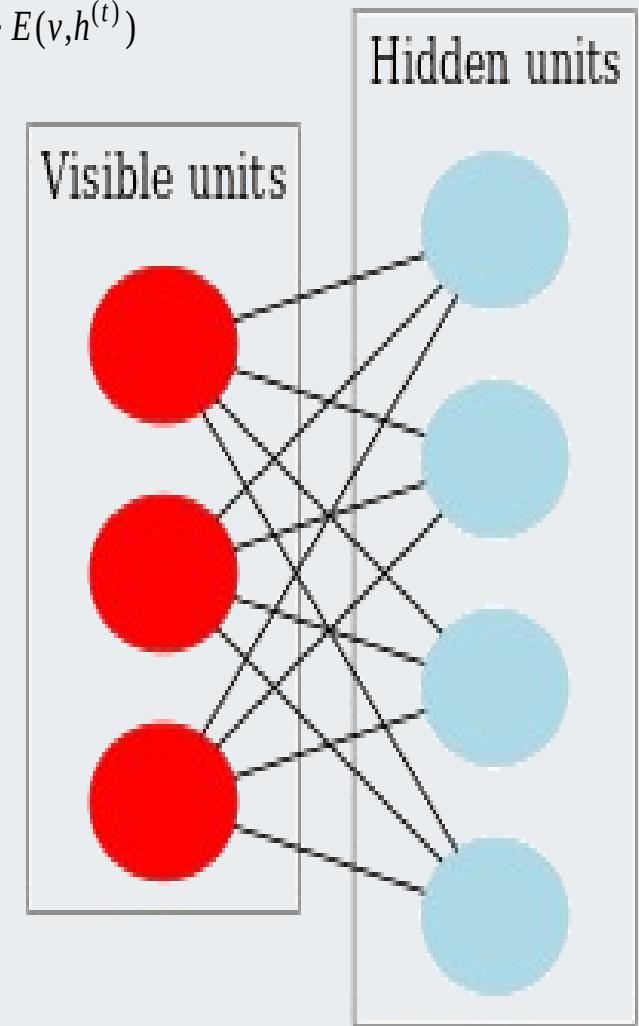
# Идея РВМ

$$f(E(v, h^{(t)})) = e^{-E(v, h^{(t)})}, \quad \hat{E} = E(v, h^{(t)}), \quad \frac{\partial f(\hat{E})}{\partial \hat{E}} = -e^{-E(v, h^{(t)})}$$

$$\frac{\partial e^{-E(v, h)}}{\partial w_{ij}} = e^{-E(v, h)} \frac{\partial (-E(v, h))}{\partial w_{ij}} = v_i h_j e^{-E(v, h)}$$

$$\frac{\partial e^{-E(v, h)}}{\partial a_i} = v_i e^{-E(v, h)}$$

$$\frac{\partial e^{-E(v, h)}}{\partial b_j} = h_j e^{-E(v, h)}$$



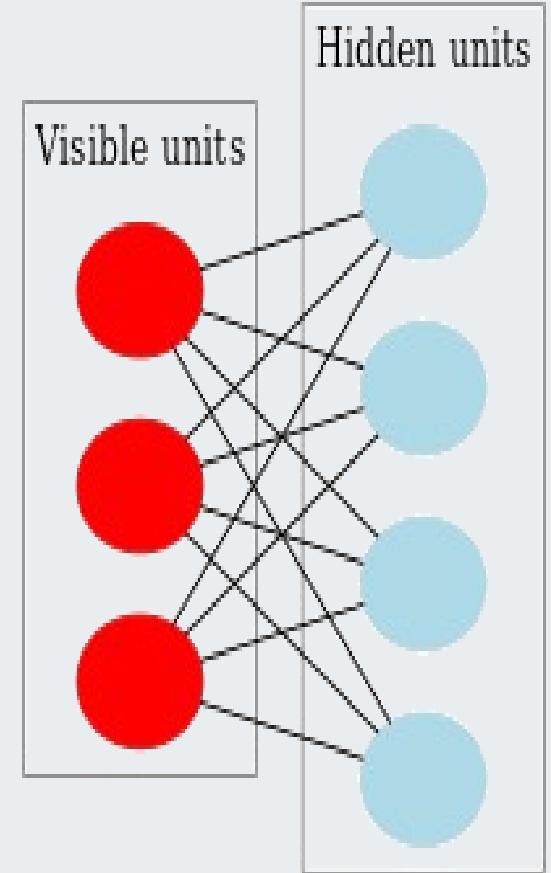
<https://habr.com/ru/post/159909/>

# Идея РВМ

$$\frac{\partial Z}{\partial w_{ij}} = \sum_r^N \sum_t^M \frac{\partial e^{-E(v^{(r)}, h^{(t)})}}{\partial w_{ij}} = \sum_r^N \sum_t^M v_i^{(r)} h_j^{(t)} e^{-E(v^{(r)}, h^{(t)})}$$

$$\frac{\partial Z}{\partial a_i} = \sum_r^N \sum_t^M v_i^{(r)} e^{-E(v^{(r)}, h^{(t)})}$$

$$\frac{\partial Z}{\partial b_j} = \sum_r^N \sum_t^M h_j^{(t)} e^{-E(v^{(r)}, h^{(t)})}$$



$$\frac{\partial P(v^{(k)})}{\partial w_{ij}} = \frac{1}{Z^2} \left( Z \left( \sum_t^M v_i^{(k)} h_j^{(t)} e^{-E(v^{(k)}, h^{(t)})} \right) - \left( \sum_t^M e^{-E(v^{(k)}, h^{(t)})} \right) \left( \sum_r^N \sum_t^M v_i^{(r)} h_j^{(t)} e^{-E(v^{(r)}, h^{(t)})} \right) \right)$$

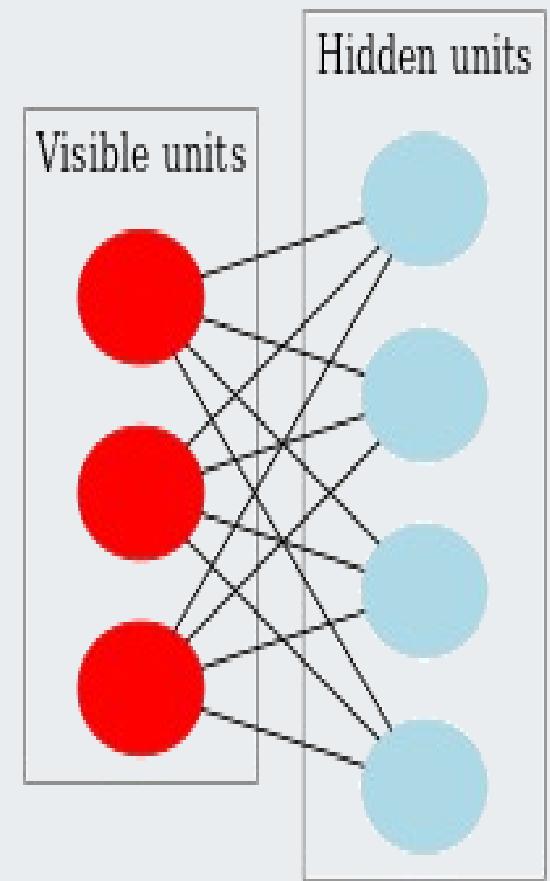
# Идея РВМ

$$\frac{\partial \ln P(v^{(k)})}{\partial w_{ij}} = \frac{1}{P(v^{(k)})} \frac{\partial P(v^{(k)})}{\partial w_{ij}}$$

$$\frac{\partial \ln P(v^{(k)})}{\partial w_{ij}} = \frac{Z}{\sum_t^M e^{-E(v^{(k)}, h^{(t)})}} \frac{1}{Z^2} \left( Z \left( \sum_t^M v_i^{(k)} h_j^{(t)} e^{-E(v^{(k)}, h^{(t)})} \right) - \right.$$

$$- \left( \sum_t^M e^{-E(v^{(k)}, h^{(t)})} \right) \left( \sum_r^N \sum_t^M v_i^{(r)} h_j^{(t)} e^{-E(v^{(r)}, h^{(t)})} \right) =$$

$$= \frac{\sum_t^M v_j^{(k)} h_j^{(t)} e^{-E(v^{(k)}, h^{(t)})}}{\sum_t^M e^{-E(v^{(k)}, h^{(t)})}} - P(v^{(k)}) \frac{\sum_r^N \sum_t^M v_i^{(r)} h_j^{(t)} e^{-E(v^{(r)}, h^{(t)})}}{\sum_t^M e^{-E(v^{(k)}, h^{(t)})}}$$



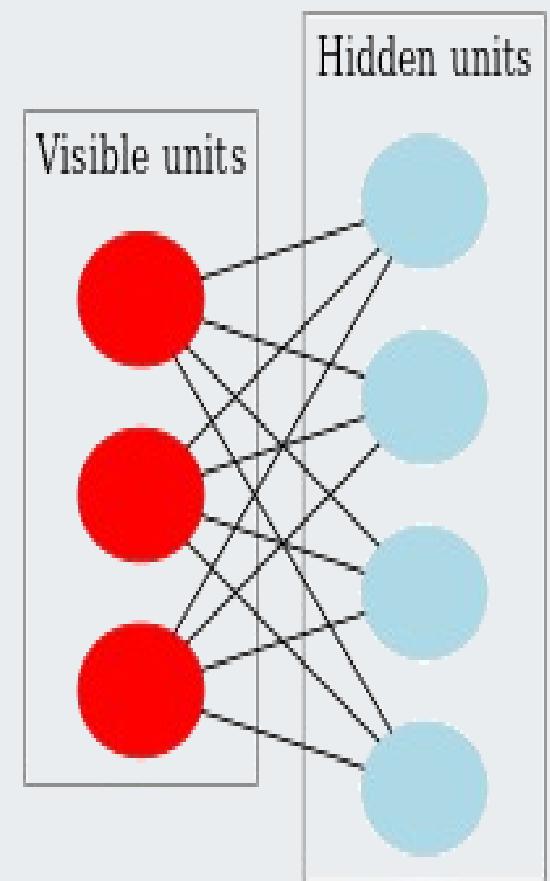
# Идея РВМ

$$\frac{\partial \ln P(v^{(k)})}{\partial w_{ij}} = \sum_t^M v_i^{(k)} h_j^{(t)} P(h^{(t)} | v^{(k)}) - \sum_r^N \sum_t^M v_i^{(r)} h_j^{(t)} P(h^{(t)}, v^{(k)})$$

$$\frac{\partial \ln P(v^{(k)})}{\partial w_{ij}} = M [v_i^{(k)} h_j]_{\text{data}} - M [v_i h_j]_{\text{model}}$$

$$\frac{\partial \ln P(v^{(k)})}{\partial a_i} = v^{(k)} - \sum_r v_i^{(k)} P(v^{(r)} | h^{(t)}) = v_i^k - M [v_i]_{\text{model}}$$

$$\frac{\partial \ln P(v^{(k)})}{\partial b_j} = \sum_t h_j^{(t)} P(h^{(t)} | v^{(k)}) - \sum_r \sum_t h_j^{(t)} P(v^{(r)}, h^{(t)}) = M [h_j]_{\text{data}} - M [h_j]_{\text{model}}$$



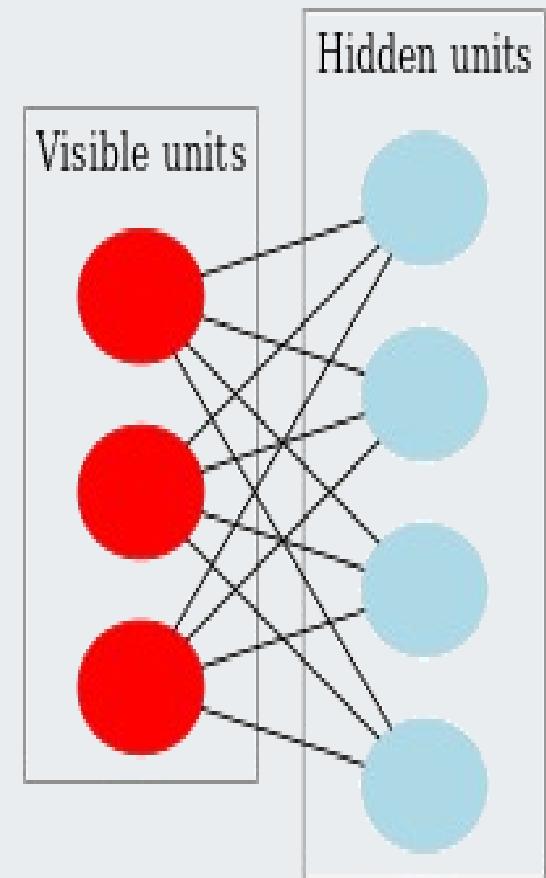
# Идея РВМ

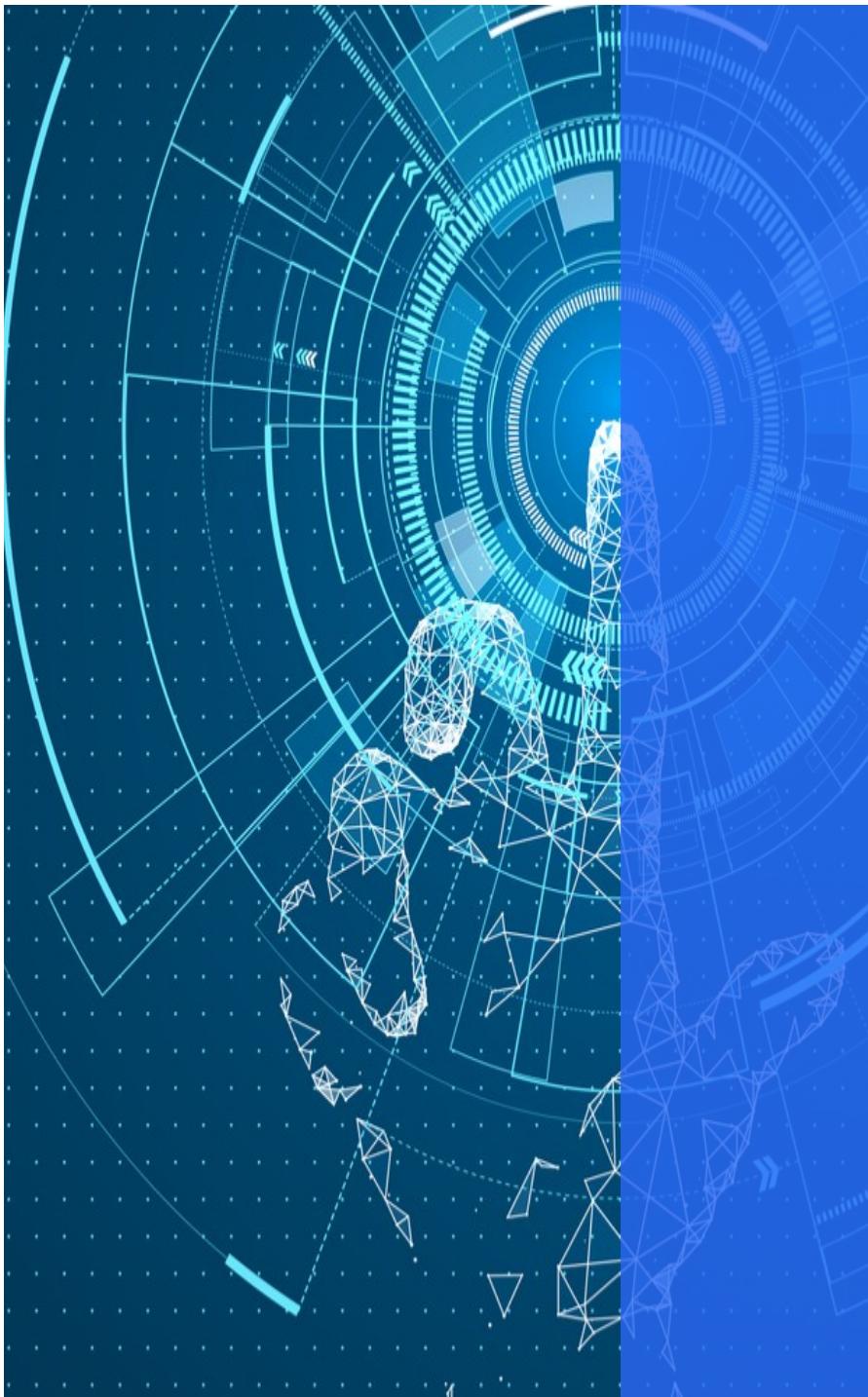
$$\frac{\partial \ln P(v^{(k)})}{\partial b_j} = \sum_t h_j^{(t)} P(h^{(t)} | v^{(k)}) - \sum_r \sum_t h^{(t)} P(v^{(r)}, h^{(t)}) = \\ M[h_j]_{\text{data}} - M[h_j]_{\text{model}}$$

$$\Delta w_{ij} = \eta(M[v_i h_j]_{\text{data}} - M[v_i h_j]_{\text{model}})$$

$$\Delta a_i = \eta(v_i - M[v_i]_{\text{model}})$$

$$\Delta b_j = \eta(M[h_j]_{\text{data}} - M[h_j]_{\text{model}})$$



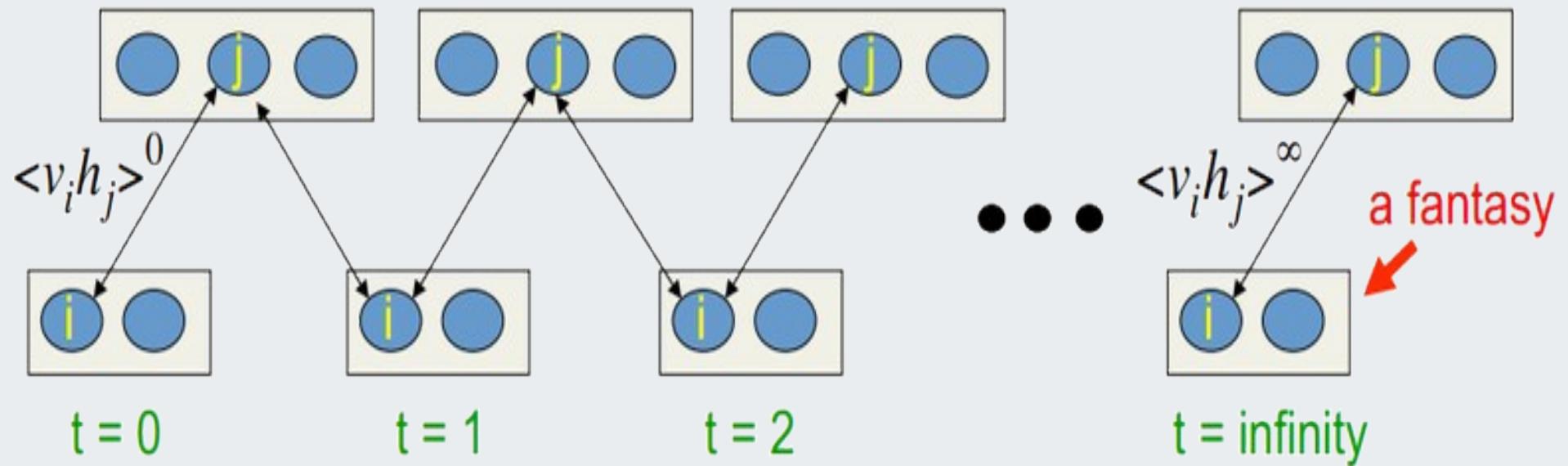


# Обучение Contrastive Divergence

---

1. состояние видимых нейронов приравнивается к входному образу
2. выводятся вероятности состояний скрытого слоя
3. каждому нейрону скрытого слоя ставится в соответствие состояние «1» с вероятностью, равной его текущему состоянию
4. выводятся вероятности видимого слоя на основании скрытого
5. если текущая итерация меньше  $k$ , то возврат к шагу 2
6. выводятся вероятности состояний скрытого слоя

# RBM: Схема



$$\Delta w_{ij} = \eta \left( M [v_i h_j]^{(0)} - M [v_i h_j]^{(\infty)} \right)$$



## RBM: Приложения

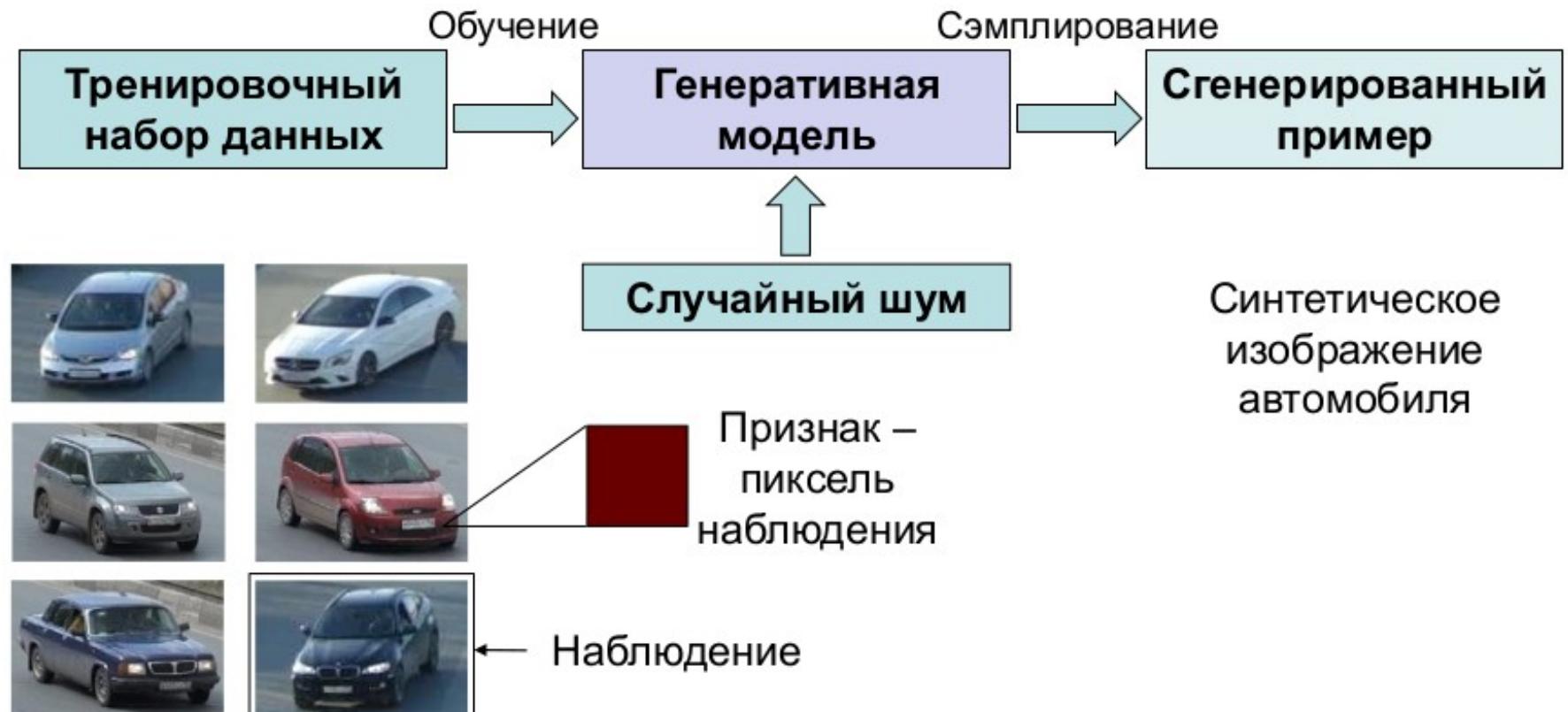
1. Ассоциативная память
2. Самоорганизующиеся модели
3. Формирование скрытых пространств

# ГЕНЕРАТИВНЫЕ МОДЕЛИ

2022 МГТУ

Корлякова М.О

# Общие сведения о генеративных моделях.



# Общие сведения о генеративных моделях.

- Предполагается, что имеется набор данных, содержащий множество примеров сущностей, которые требуется генерировать, – тренировочный набор данных (training dataset)
- Наблюдение (observation) – отдельный элемент тренировочного набора данных
- В задаче генерации изображений каждое наблюдение состоит из множества признаков, как правило, признак – значение интенсивности отдельного пикселя
- Модель обучается генерировать данные (изображения) по тем же правилам, что построен тренировочный набор данных
- В процессе генерации каждому пикселю назначается некоторое значение интенсивности

# Общие сведения о генеративных моделях.

- Генеративная модель должна быть вероятностной
- Если модель в процессе сэмплирования, например, в качестве значения интенсивности пикселя выдает среднее значение по тренировочной выборке, то модель не является генеративной, поскольку обеспечивает всегда одинаковый результат
- Модель должна включать элемент случайности, отражающий индивидуальные особенности сгенерированного примера
- Таким образом, должно существовать вероятностное распределение, которое объясняет, почему некоторые наблюдения (изображения) похожи в тренировочном наборе, а другие – нет
- **Цель – построить модель, наилучшим образом имитирующую это распределение**

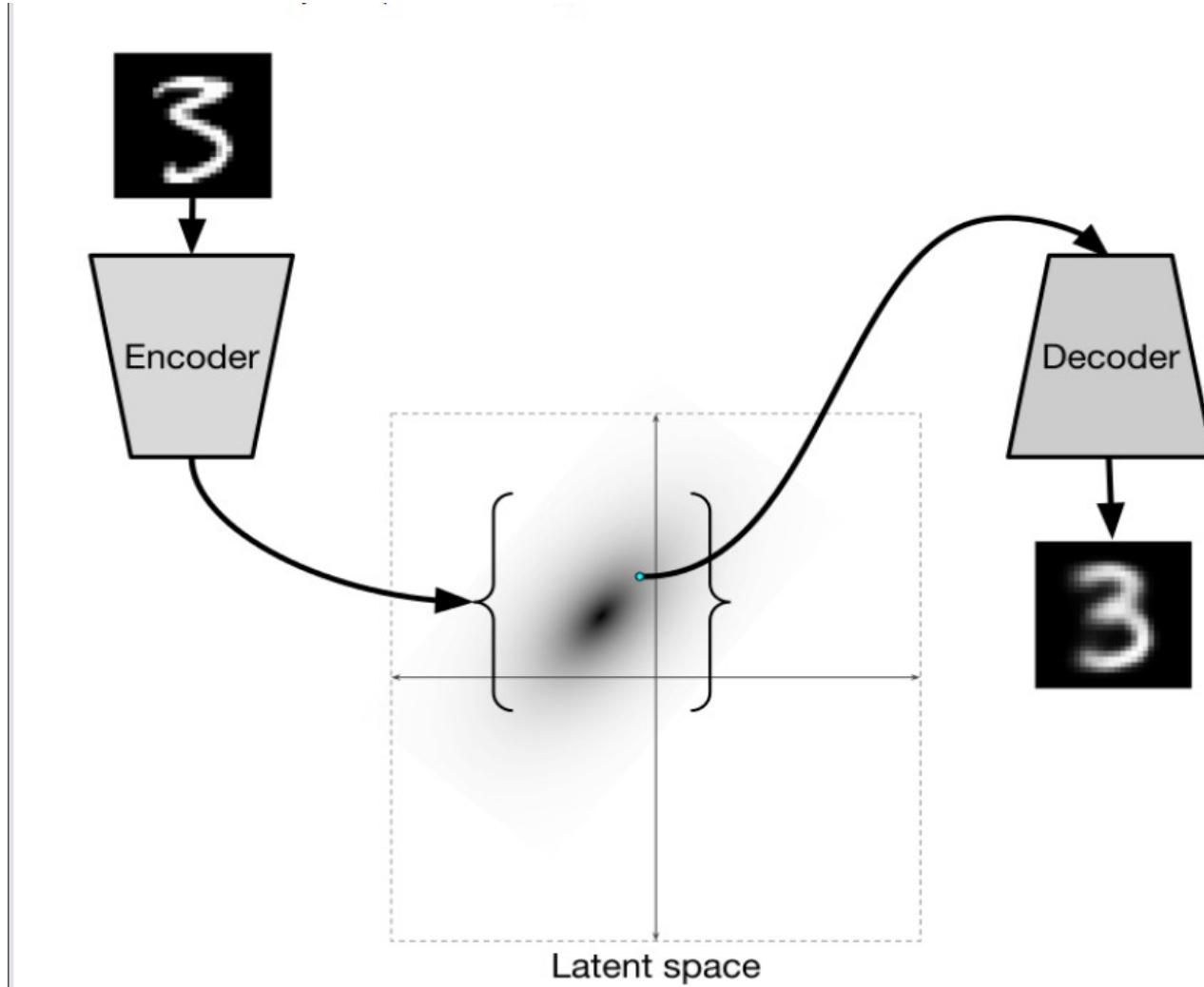
# Генеративные / дискриминативные модели

## Разные цели моделирования:

- Генеративные модели оценивают  $p(x, y)$  – совместное распределение вероятностей (оценивают границы классов)
  - Если набор данных не размечен (решается задача обучения без учителя), то модели оценивают  $p(x)$  – вероятность увидеть наблюдение  $x$
  - Если тренировочный набор размечен, то генеративная модель может оценивать условную вероятность  $p(y|x)$  (вычисляется по правилу Байеса) – вероятность увидеть наблюдение  $x$  при условии, что известна его принадлежность классу  $y$
- Дискриминативные модели оценивают  $p(y|x)$  – вероятность метки  $y$  при условии, что на входе имеется наблюдение  $x$  (оценивают границу между классами)

\* Ng A.Y., Jordan M.I. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naïve Bayes // Advances in Neural Information Processing Systems. – 2002. – [<http://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf>].

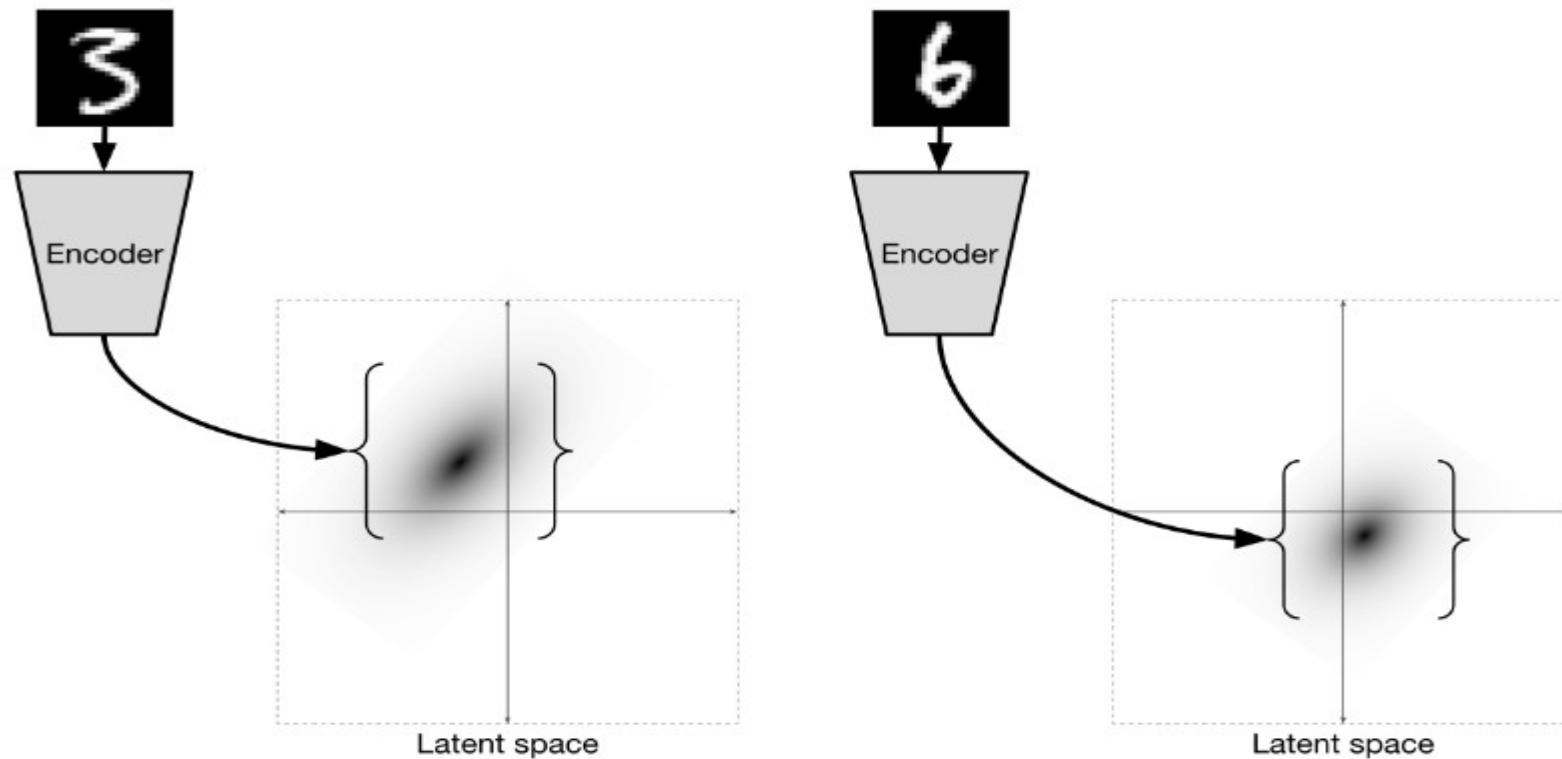
# Общие сведения о VAE.



<https://keras.io/examples/generative/vae/>

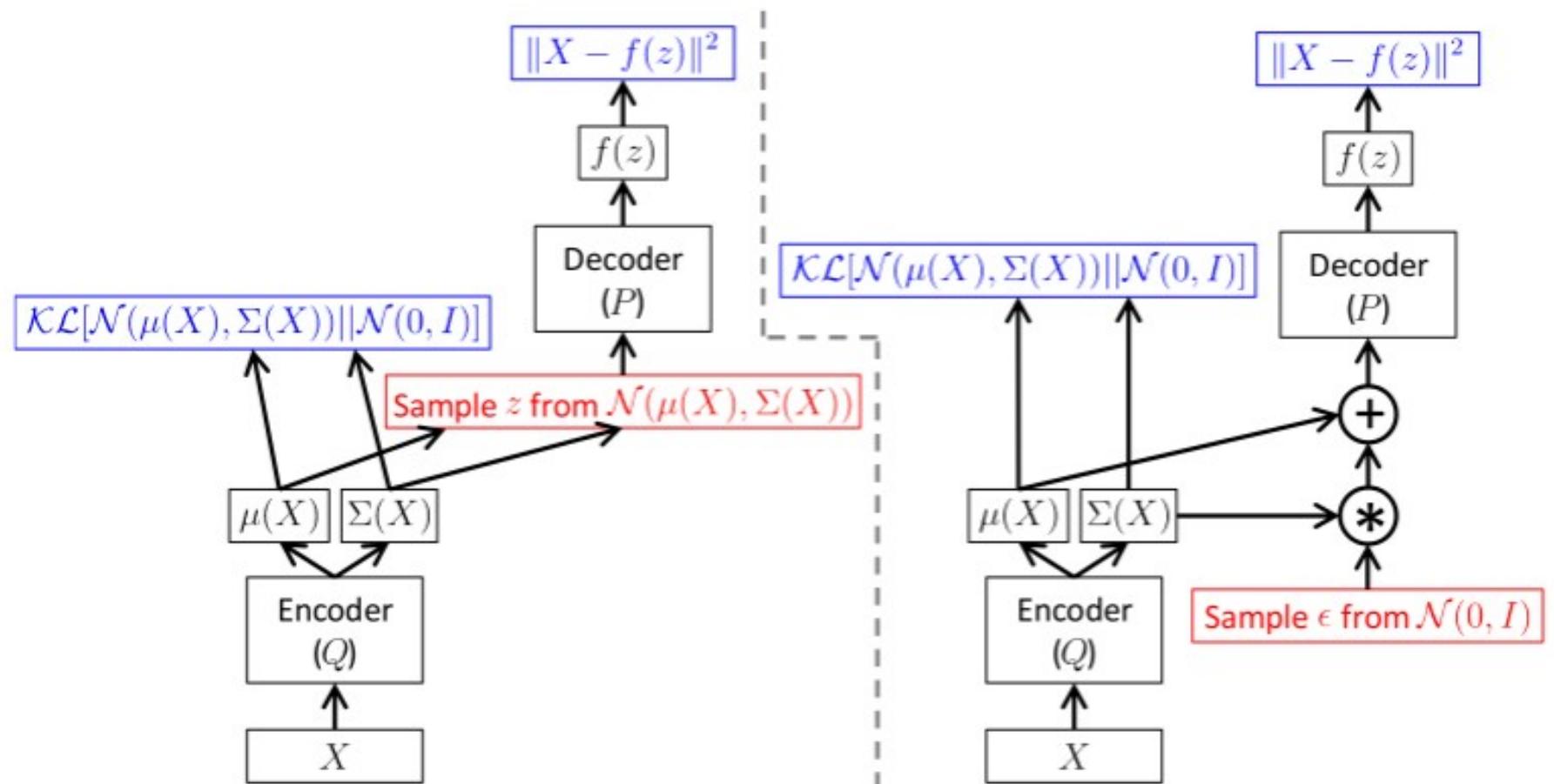
# Общие сведения о VAE.

$$Q(Z|X; \theta_1) = N(\mu(X; \theta_1), \Sigma(X; \theta_1))$$





# Общие сведения о VAE.



# Задание для РК2

1. Собрать пример вариационного энкодера для данных Fashion-MNIST
2. Обучить
3. Показать движение в латентном пространстве (примеры генерации вдоль координат латентного пространства)



# Общие сведения о GAN.

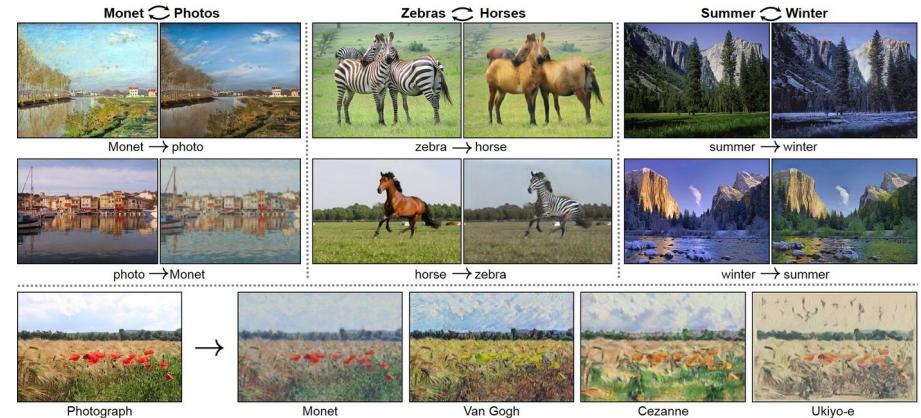


2014

2015

2016

2017



A small bird with a black head and wings and features grey wings

256x256  
Stage-I GAN  
without CA



256x256  
Stage-I GAN  
with CA



256x256  
StackGAN  
with CA,  
Text twice

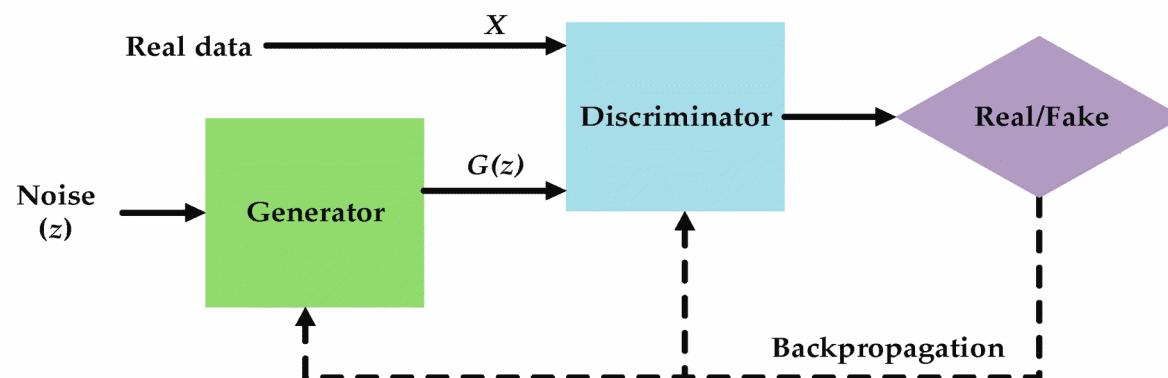
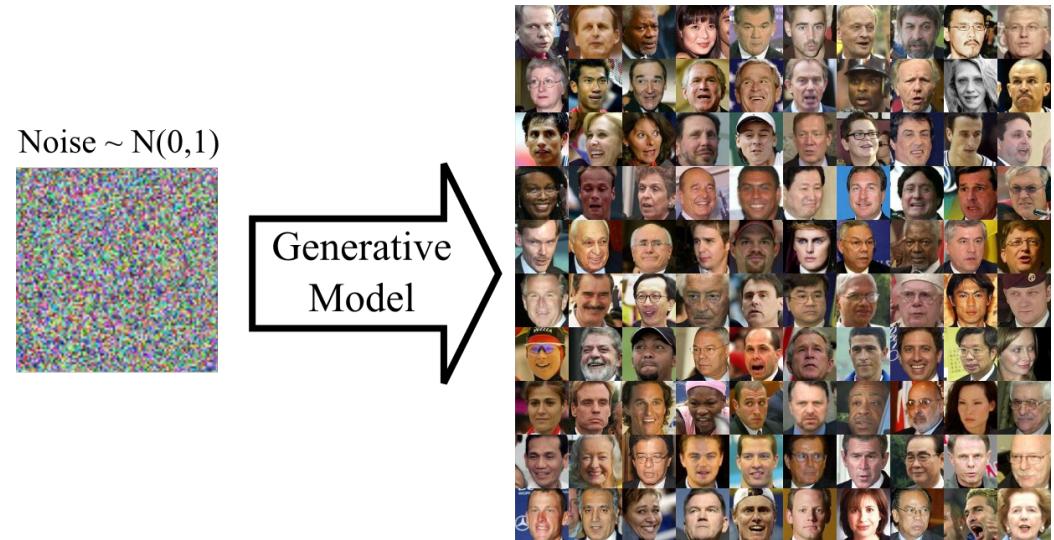
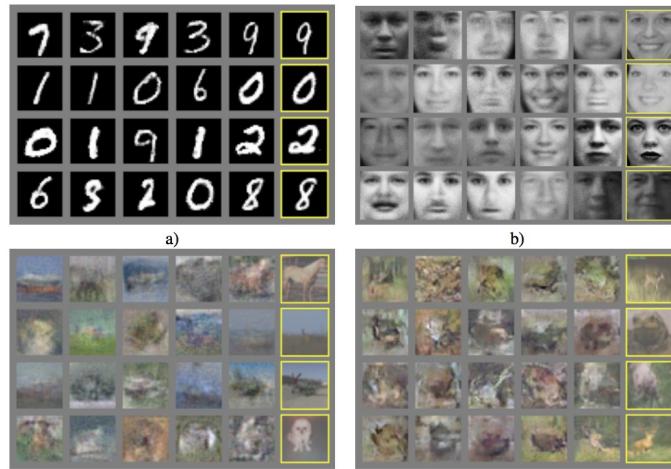


This bird is completely red with black wings and pointy beak





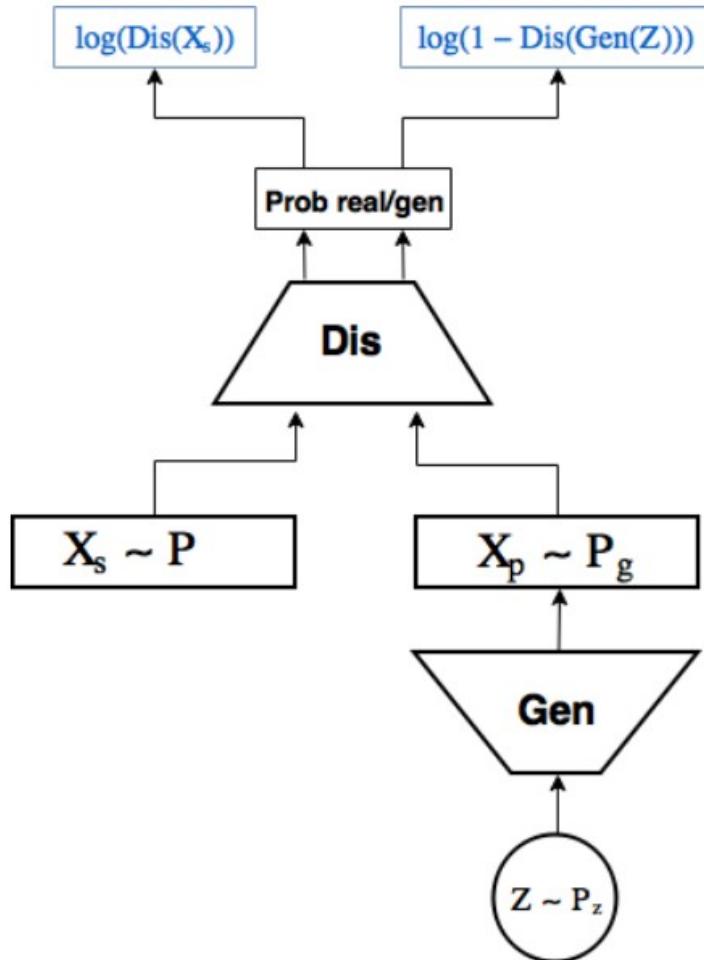
# Архитектура GAN



[https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/generative/ipynb/dcgan\\_overriding\\_train\\_step.ipynb](https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/generative/ipynb/dcgan_overriding_train_step.ipynb)



# Архитектура GAN



Обозначения:

- $X$  – множество наблюдений из распределения  $p_{data}$
- $Z$  – пространство латентных факторов из распределения  $p_Z$  (например, множество случайных векторов из равномерного распределения на отрезке 0; 1 )

## Генератор

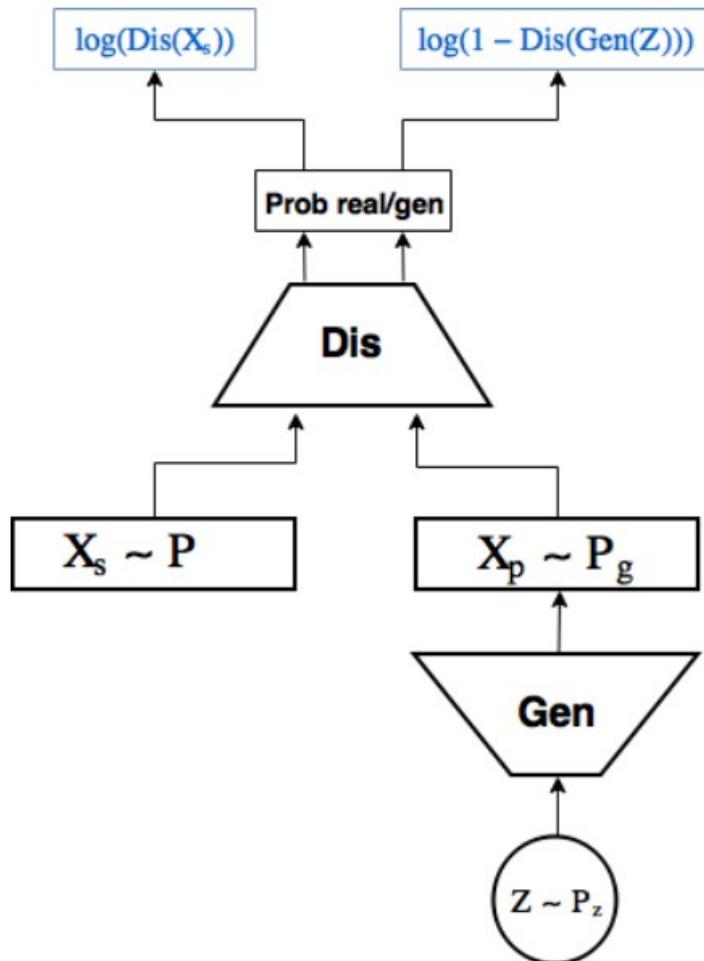
- Отображение  $G: Z \rightarrow \mathbb{R}^n$  с параметрами  $\theta$
- Цель  $G$  – сгенерировать образец, максимальное похожий на наблюдения из распределения  $p_{data}$

## Дискриминатор

- Отображение  $D: \mathbb{R}^n \rightarrow [0; 1]$  с параметрами  $\gamma$
- Цель  $D$  – выдавать максимальную оценку на наблюдениях из  $X$  и минимальную оценку на образцах, сгенерированных  $G$



# Архитектура GAN



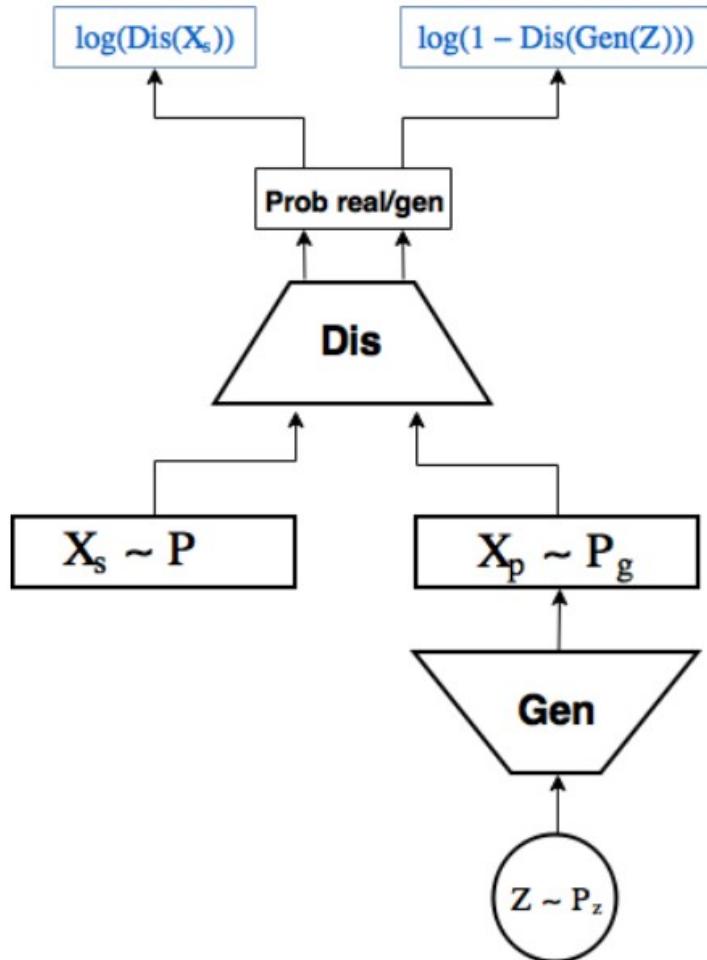
Обозначения:

- $\theta$  и  $\gamma$  – параметры нейронных сетей, соответствующих генератору и дискриминатору
- $p_{gen}$  – распределение образцов, порождаемое генератором

Задача – построить распределение  $p_{gen}$ , которое наилучшим образом описывает распределение  $p_{data}$



# Архитектура GAN



- **Функция ошибки:**

$$E_{x \sim p_{\text{data}}} [\log D(x)] + E_{\tilde{x} \sim p_{\text{gen}}} [\log(1 - D(\tilde{x}))],$$

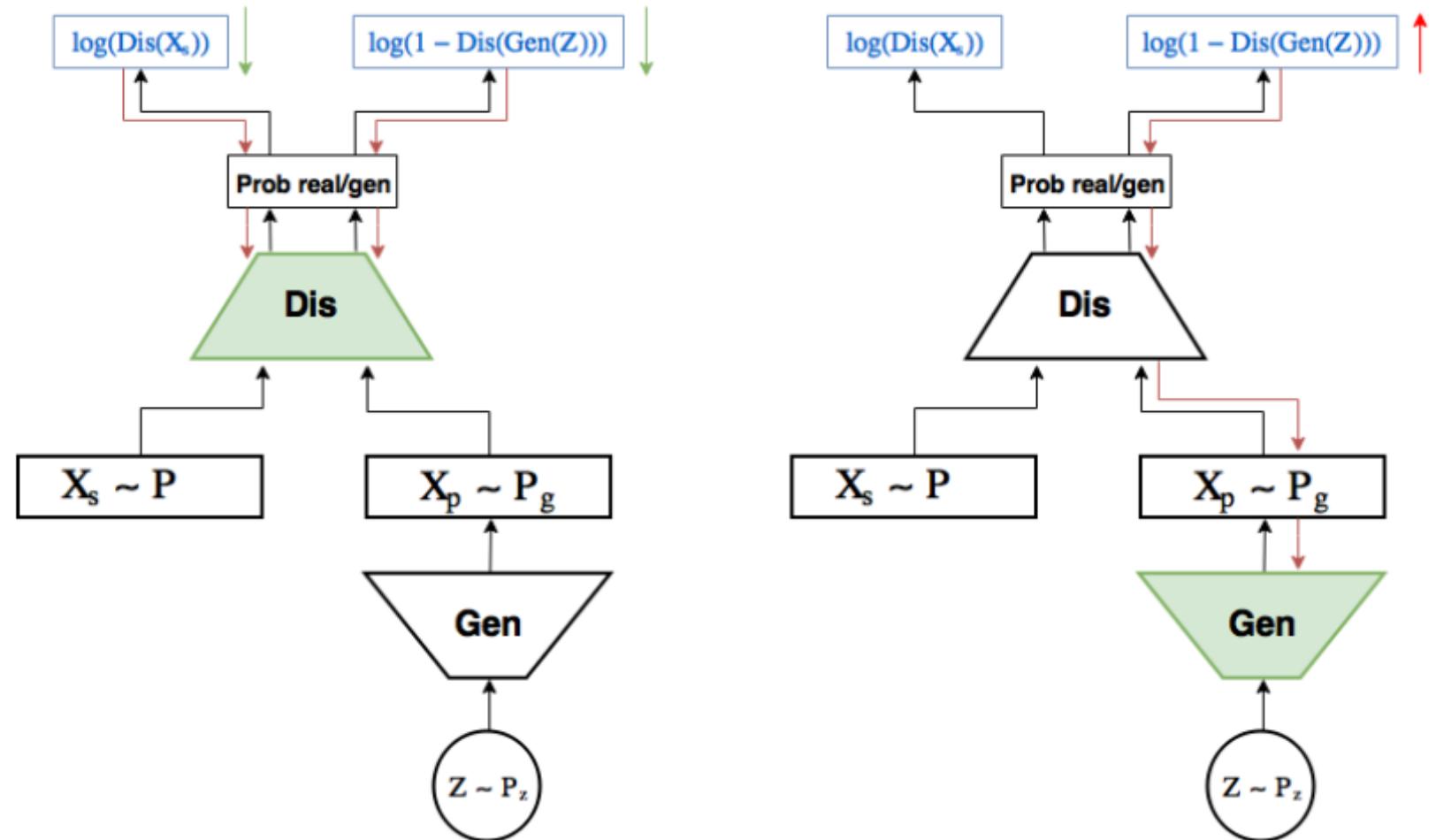
где  $E_{\tilde{x} \sim p_{\text{gen}}} [\log(1 - D(\tilde{x}))] = E_{z \sim p_z} [\log(1 - D(G(z)))]$

- **Задача обучения** генеративной состязательной сети сводится к решению задачи оптимизации (минимаксная игра)

$$\min_G \max_D E_{x \sim p_{\text{data}}} [\log D(x)] + E_{z \sim p_z} [\log(1 - D(G(z)))]$$



# Архитектура GAN





# Классификация GAN

- Полносвязные генеративные состязательные сети (Fully Connected GANs)
- Условные генеративные состязательные сети (Conditional GANs, CGAN)
- Пирамида Лапласа из состязательных сетей (Laplacian Pyramid of Adversarial Networks, LAPGAN)
- Глубокие сверточные генеративные состязательные сети (Deep Convolutional GANs, DCGAN)
- Генеративные рекуррентные состязательные сети (Generative Recurrent Adversarial Networks, GRAN)
- ...

\* Alqahtani H., Kavakli-Thorne M., Kumar G. Applications of Generative Adversarial Networks (GANs): An Updated Review // Archives of Computational Methods in Engineering. – 2019.



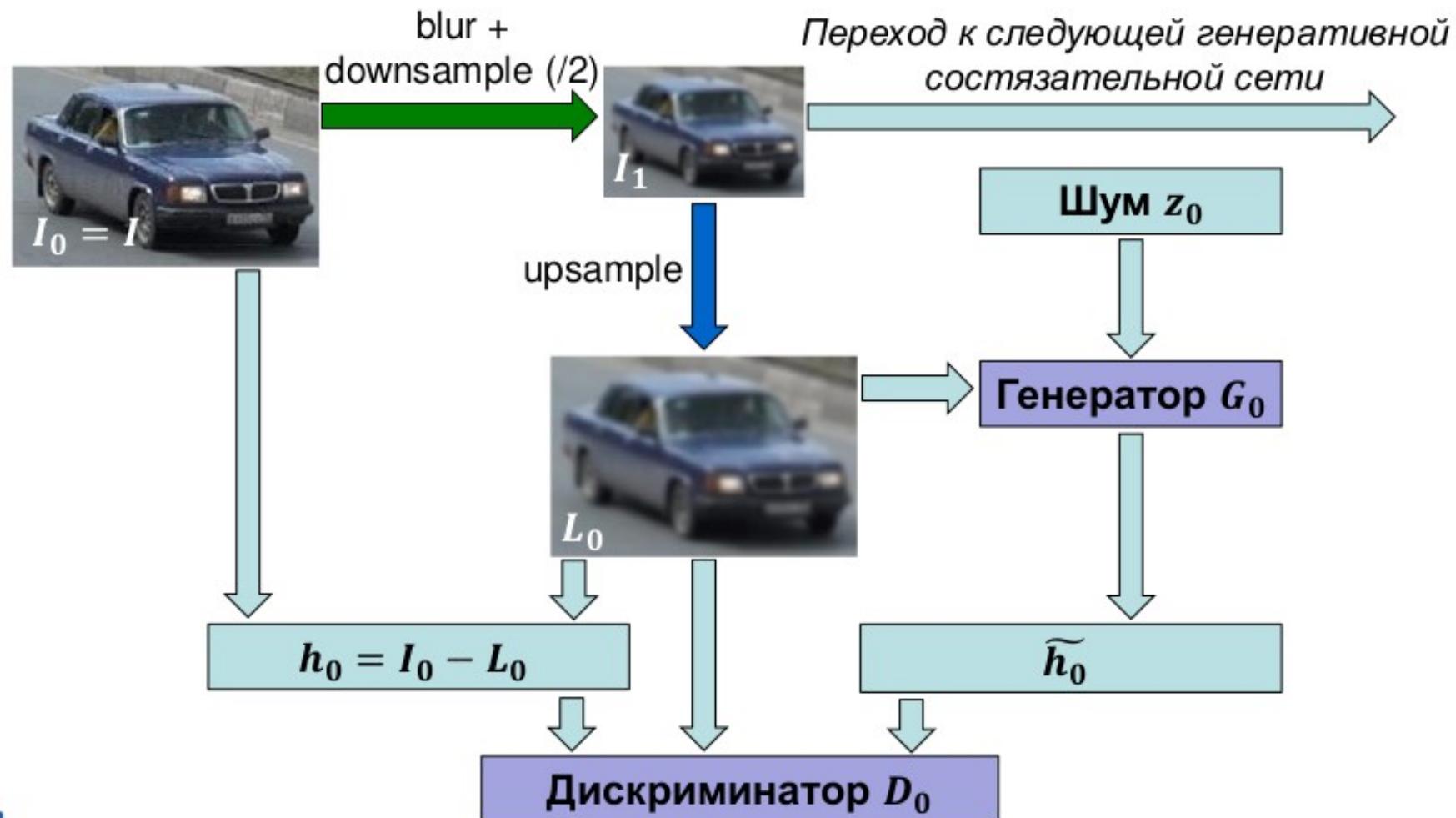
# Классификация GAN

- Генеративные состязательные сети, максимизирующие информацию (Information Maximizing GANs, InfoGAN)
- Двунаправленные генеративные состязательные сети (Bidirectional GANs, BiGAN)
- Генеративные автокодировщики (Adversarial Autoencoders, AAE)
- Гибриды вариационных автокодировщиков и генеративных состязательных сетей (Variational Autoencoder-GAN, VAE-GAN)
- ....

\* Alqahtani H., Kavakli-Thorne M., Kumar G. Applications of Generative Adversarial Networks (GANs): An Updated Review // Archives of Computational Methods in Engineering. – 2019.



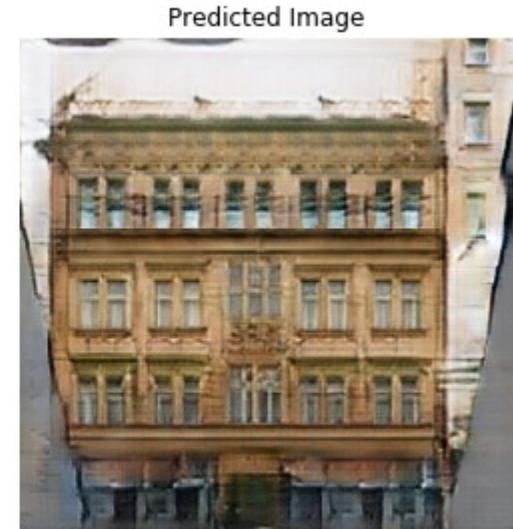
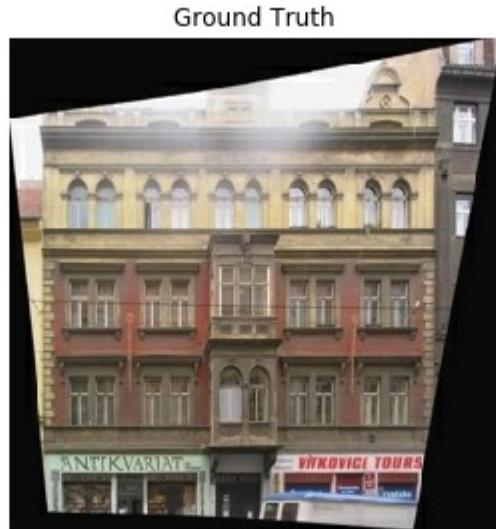
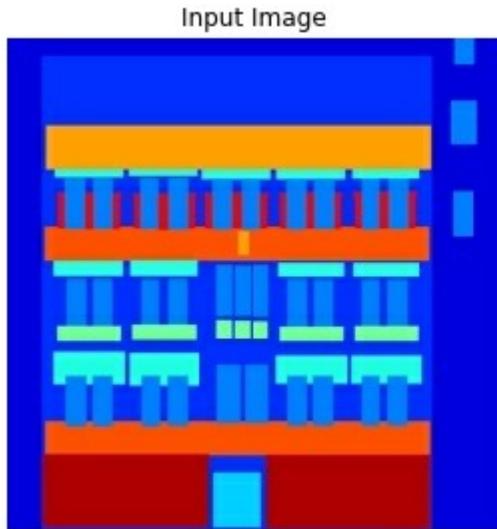
# Пирамиды Лапласа





GeekBrains

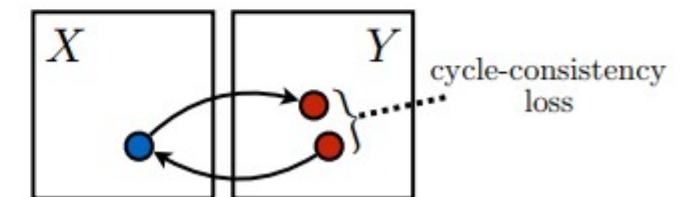
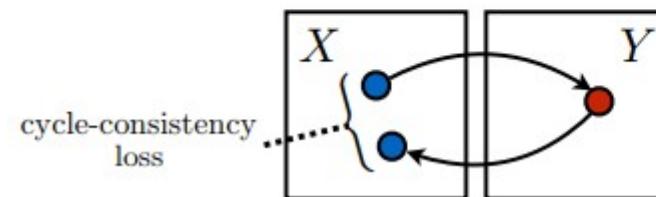
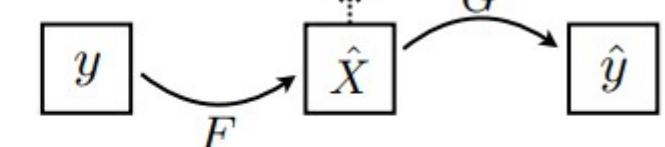
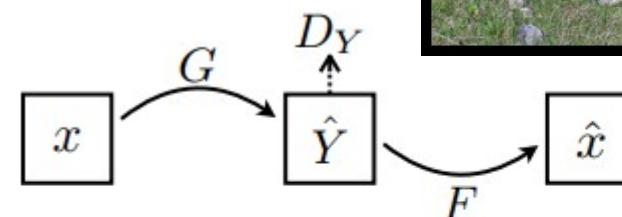
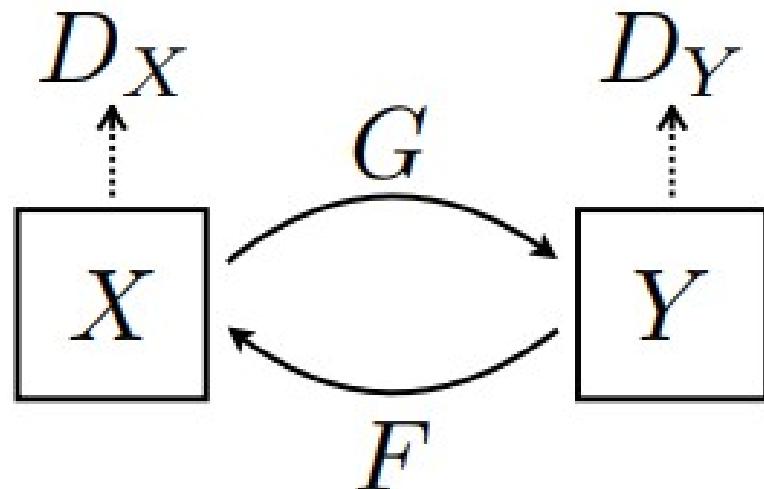
# Pix2Pix



[https://www.tensorflow.org/tutorials/generative/  
pix2pix#define\\_the\\_loss\\_functions\\_and\\_the\\_optimizer](https://www.tensorflow.org/tutorials/generative/pix2pix#define_the_loss_functions_and_the_optimizer)



# Архитектура CycleGAN





# Примеры архитектур

Увеличение количества данных (data augmentation) – генерация синтетических данных, похожих на данные в некоторой существующей выборке, но содержащих различные трансформации, с целью их последующего использования, как правило, для расширения тренировочного набора данных



CycleGAN [<https://www.nature.com/articles/s41598-019-52737-x>] (2019) – задача сегментации данных компьютерного томографии



Data Augmentation GAN (DAGAN) [<https://arxiv.org/pdf/1711.04340.pdf>] (2018) – увеличение количества данных за счет трансформаций



Balancing GAN (BAGAN) [<https://arxiv.org/pdf/1803.09655.pdf>] (2018) – восстановление сбалансированности набора данных



# Примеры архитектур

Laplacian Pyramid of Adversarial Networks (LAPGAN)  
[<https://arxiv.org/pdf/1506.05751.pdf>] (2015) – повышение  
разрешения изображений

Generative Adversarial What-Where Network (GAWWN)  
[<https://arxiv.org/pdf/1610.02454.pdf>] (2016) – генерация  
изображений с использованием дополнительного описания  
и информации о расположении объектов

Generative Adversarial Network for image Super-Resolution  
(SRGAN) [<https://arxiv.org/pdf/1609.04802.pdf>] (2017) –  
повышение разрешения изображения

Self-Attention Generative Adversarial Network (SAGAN)  
[<https://arxiv.org/pdf/1805.08318.pdf>] (2019) – моделирование  
зависимостей между отдельными частями изображения  
для генерации изображений высокого разрешения

# Техники обучения моделей

Metric learning

SimCLR

Генерации эмбедингов:

- используют сходство, а не класс
- используют аугментацию

# Техники обучения моделей

## Перенос обучения

# Техники обучения моделей

## Методы дистиляции



GeekBrains

# Вопросы

