



**ІТМО**

**НЕЙРОННЫЕ СЕТИ**

- Обучение с учителем (supervised learning) – есть размеченные данные (для каждого примера есть «решение или метка класса»)
  - С частичным привлечением учителя (semi-supervised learning) – для части прецедентов задается пара «ситуация, решение», а для части - только «ситуация»
- Обучение без учителя (unsupervised learning) – есть неразмеченные данные («ситуация»), требуется сгруппировать объекты



- Обучение с подкреплением (reinforcement learning) – есть размечаемые данные («ситуация, предполагаемое решение»). Алгоритм обучения работает через вознаграждение за правильное решение или наказание за неправильное.



# Выборки

ІТМО





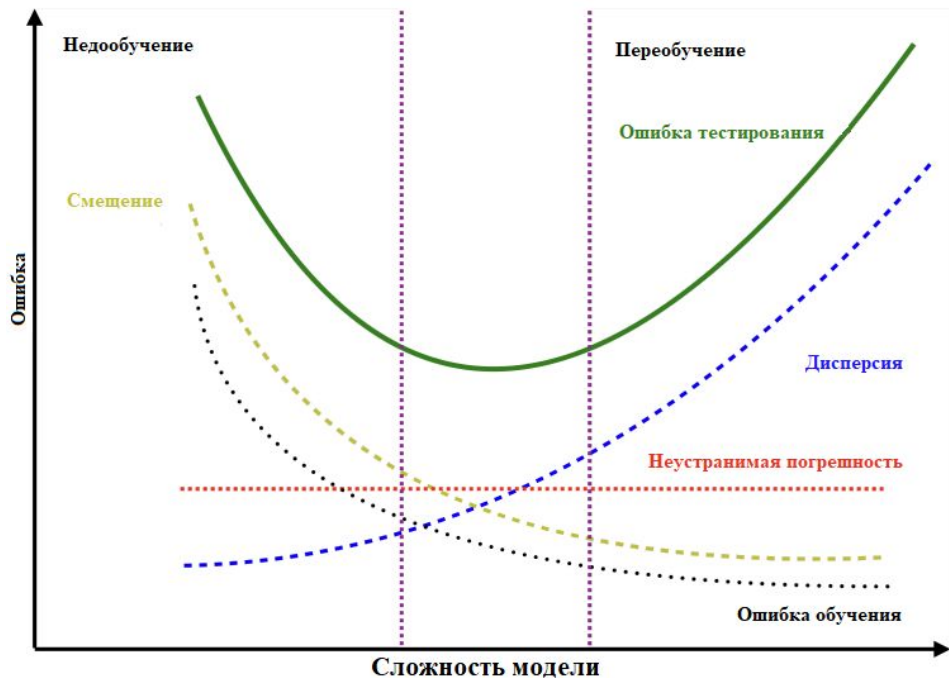
- Исходные
- Производные
  - Агрегированные – показатели, определенные по группе (сумма, среднее, минимум, максимум)
  - Индикаторы – наличие или отсутствие характеристики
  - Отношения – взаимосвязь между двумя или более значениями данных
  - Отображения – преобразование непрерывных в категориальные

# Извлечение признаков

- Тексты – это токенизация
- Изображения – извлечение краев и цветовые пятна
- Дата и время – полезно вычленить выходные и праздники, дни недели
- Местоположение (адрес или координаты) - извлечь плотность, средний доход по району
- Номер телефона – регион и оператор связи
- Лаги по времени. Окно 3-7 последовательностей



# Ошибка обобщения



Ошибка обобщения — сумма смещения, дисперсии и величины, называемой неустраняемой погрешностью, которая является результатом шума в самой задаче.



# Сложность

количество настраиваемых параметров архитектуры модели, другими словами, сложность модели определяет ее информационную емкость. При увеличении сложности модели происходит уменьшение смещения и увеличение разброса.







- Недообучение (underfitting) – когда модель, построенная с помощью алгоритма, является слишком упрощенной, чтобы представлять базовую взаимосвязь между признаками и классом в обучающей выборке.
- Это явление можно заметить по большой ошибке на обучающей выборке (еще говорят, что «не удаётся настроиться на выборку»). Помимо простоты модели, недообучение может возникать еще и из-за малого количества эпох обучения.



- Переобучение (overfitting) – когда модель, построенная с помощью алгоритма, настолько сложна, что модель слишком точно приближает обучающую выборку и становится чувствительной к шуму.
- Это явление можно заметить по увеличивающейся разнице между ошибкой на обучающей выборке и тестовой выборке с каждой эпохой обучения. Поэтому при обучении строится график изменения ошибки на обучающей и тестовой выборках. Переобученная модель обладает низкой обобщающей способностью, в эксплуатации она будет часто ошибаться.

# Нейросети по характеру связей

ІТМО

- Прямого распространения
- Обратного распространения, или рекуррентные
- Радиально-базисные функции
- Самоорганизующиеся

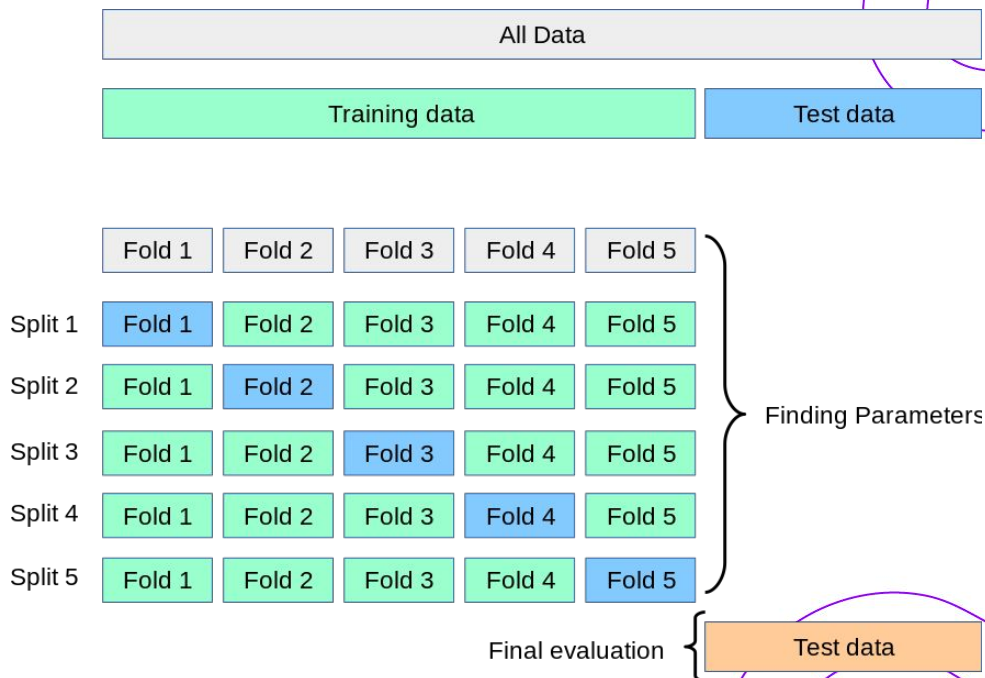


# Кросс-валидация (скользящий контроль)

- Метод отложенных данных (holdout method) – разделение 70-30 или 60-40 или 80-20.
- Оценка ошибки близка к ошибке модели на новых данных, но сильно зашумлена.
- Для борьбы с шумом многократно случайно разделяют обучающую и тестовую выборку, параметр ошибки при этом усредняют.
- Но в процессе итераций каждая точка данных будет попадать в тестовое подмножество различное число раз, что может привести к смещению оценки.



# Кросс-валидация (скользящий контроль)



Контроль по k-блокам (k-fold cross-validation) - данные случайным образом делятся на k непересекающихся подмножеств (5, 10 или 20). После циклического перебора всех k подмножеств полученная оценка усредняется.

# Оценка классификации

	Y=1	Y=-1
A(x)=1	True positive	False positive
A(x)=-1	False negative	True negative



$$N = TP + FP + FN + TN$$

Полнота (способность обнаруживать данный класс, sensitivity)

$$recall = \frac{TP}{TP + FN}$$

Точность (способность отличать этот класс от других)

$$precision = \frac{TP}{TP + FP}$$

Достоверность (правильность)

$$accuracy = \frac{TP + TN}{N}$$

# Оценка классификации

**Специфичность** (specificity), чаще всего применяется в медицинской статистике. Высокая специфичность позволяет отсеять людей, у которых действительно нет этого заболевания.



$$TNR = \frac{TN}{TN + FP}$$

**Коэффициент корреляции Мэтьюса** (в статистике известен как фи-коэффициент) может применяться как мера качества для бинарной классификации при высоком дисбалансе классов. Принимает значение в диапазоне  $[-1,1]$ , где 1 – идеальное предсказание, 0 – случайное предсказание, -1 – полное расхождение между предсказанием и наблюдением.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

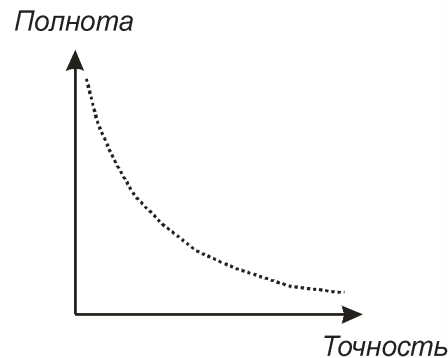
# F-мера

Полнота и точность противоречат друг другу. Всегда можно повысить полноту до 1 при очень низкой точности.



$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}, \quad \alpha \in [0, 1]$$

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, \quad \beta^2 = \frac{(1 - \alpha)}{\alpha}, \beta^2 \in [0, \infty)$$



В некоторых задачах необходимо отдавать предпочтение полноте ( $\beta > 1$ ) или точности ( $\beta < 1$ ).

В противном случае можно использовать **сбалансированную F-меру** ( $\beta = 1$ ):

$$F_1 = \frac{2PR}{P + R}$$



# ROC-кривая

Вектора вероятностей классов, порог позволяет разделить классы, кривая строится для разных значений порога. Выбор порога обусловлен задачей, можно сдвинуть в сторону того или иного класса.

Для каждого класса своя ROC-кривая при многоклассовой классификации

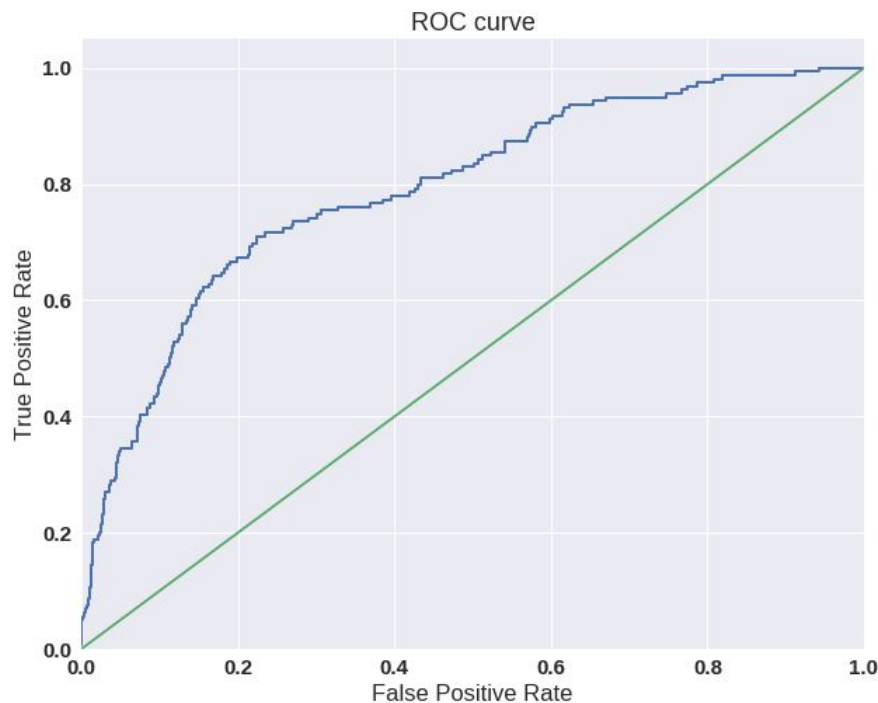
Клиент	Вероятность невозврата
Mike	0.78
Jack	0.45
Larry	0.13
Kate	0.06
William	0.03
Jessica	0.02

Отказ

$p^*=0.15$

Одобрение

# ROC-кривая



$TPR = recall$

$FPR = \frac{FP}{FP+TN}$  - доля объектов negative класса, предсказанных неверно

Каждая точка на кривой соответствует значению порога

**AUC** эквивалентна **вероятности**, что классификатор присвоит больший вес случайно выбранной положительной сущности, чем случайно выбранной отрицательной



$$A = \int_{-\infty}^{\infty} y(T)x'(T)dT = \int_{-\infty}^{\infty} TPR(T)FPR'(T)dT = \int_{-\infty}^{\infty} TPR(T)P_0(T)dT = \langle TPR \rangle$$

В идеальном случае  $AUC=1$

Чем она больше тем алгоритм лучше

$AUC = 0.5$  – случайное гадание

$AUC < 0.5$  - классификатор действует с точностью до наоборот

## Средняя абсолютная ошибка (MAE)



$$MAE = \frac{1}{n} \sum_{i=1}^n |a(x_i) - y_i|$$

**Среднеквадратическая ошибка (MSE)** применяется когда надо подчеркнуть большие ошибки, но поэтому она более чувствительна к выбросам, чем MAE.

Чем ошибка меньше, тем лучше, но важно учитывать масштаб данных. Чтобы MSE имел размерность исходных данных, из него извлекают корень и получают RMSE, что затрудняет сравнение на разных наборах.

$$MSE = \frac{1}{n} \sum_{i=1}^n (a(x_i) - y_i)^2$$

$$RMSE = \sqrt{MSE}$$

# Оценка регрессии

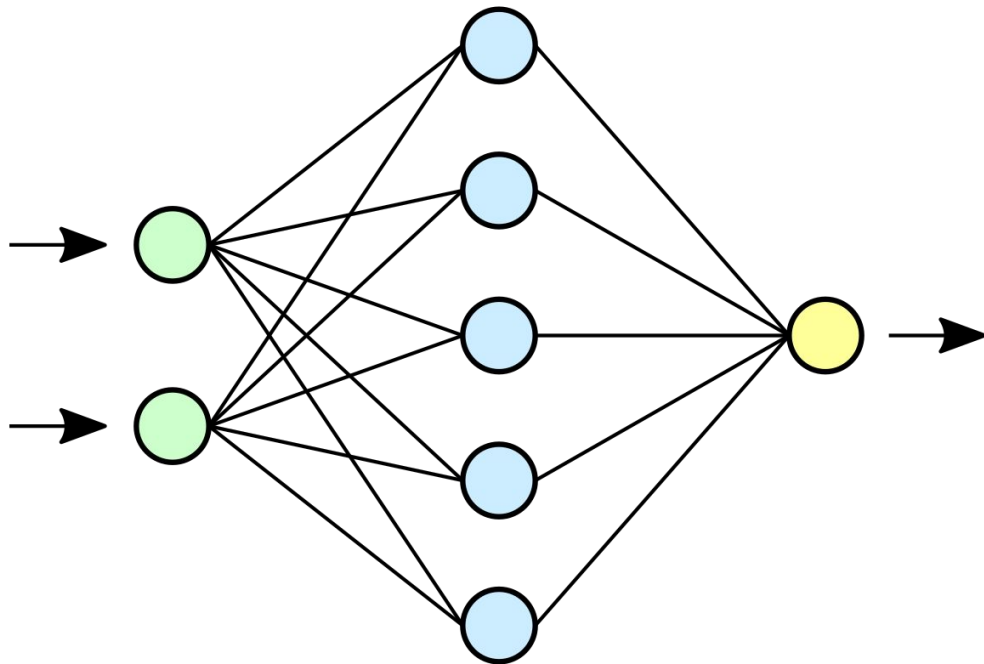
**Коэффициент детерминации** ( $R^2$ ) – доля дисперсии, объясненная моделью, в общей дисперсии целевой переменной. Чем ближе к 1, тем модель лучше объясняет данные. Модели с коэффициентов детерминации больше 0,8 можно считать хорошими.  $\bar{y}$  – среднее арифметическое  $y_i$ .

$$R^2 = 1 - \frac{\sum_{i=1}^n (a(x_i) - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

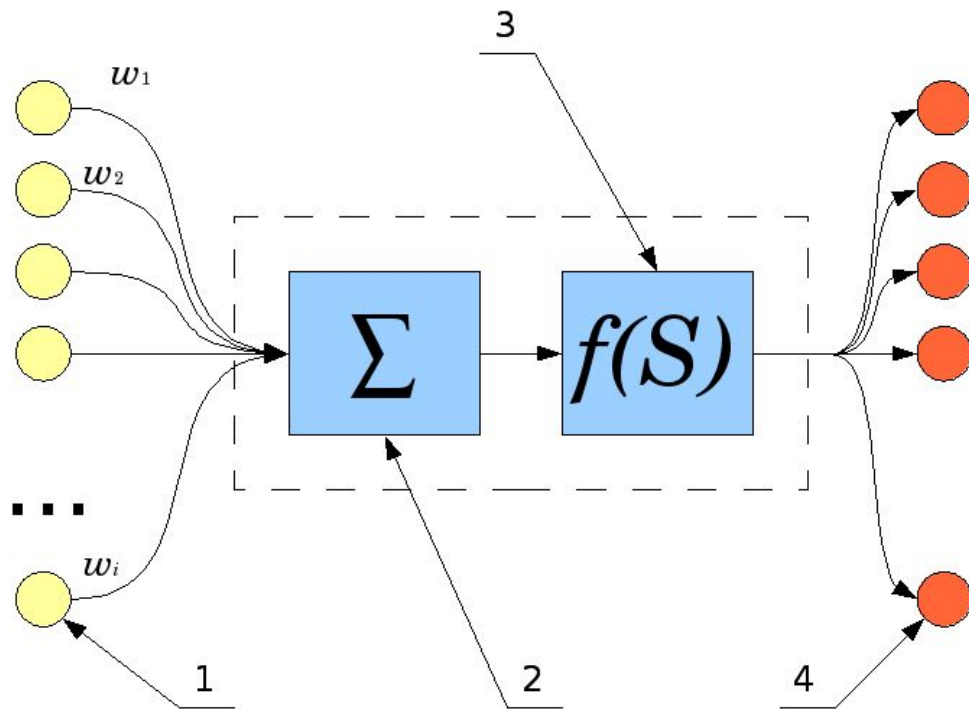
**SMAPE** (Symmetric Mean Absolute Percentage Error) рассматривает не абсолютные, а относительные ошибки на примерах

$$SMAPE = \frac{1}{n} \sum_{i=1}^n \frac{2|a(x_i) - y_i|}{y_i + a(x_i)}$$

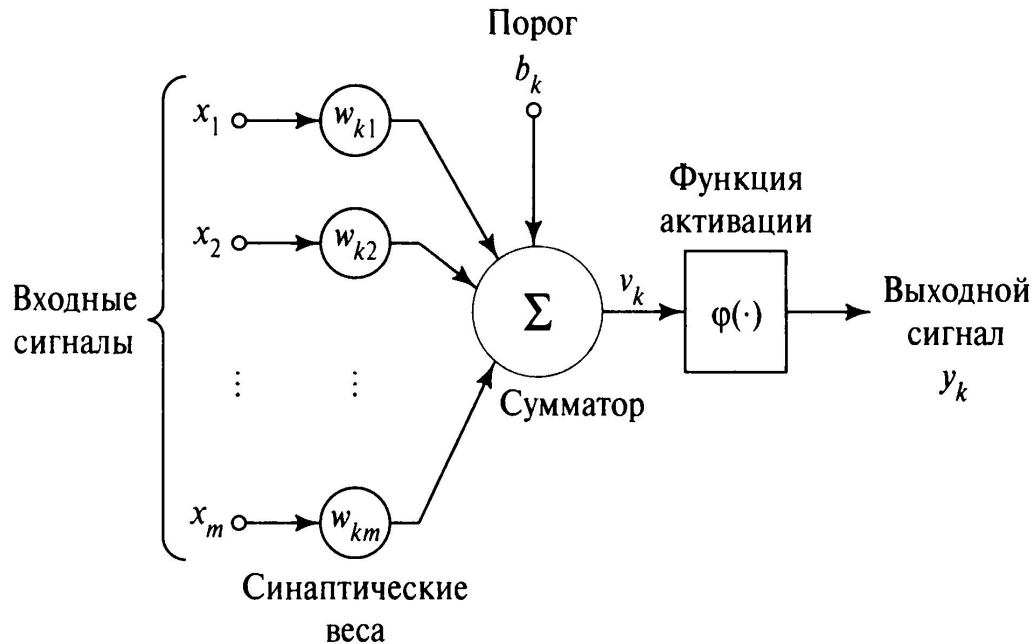




# Формальный нейрон



# Формальный нейрон



$$u_k = \sum_{j=1}^m w_{kj} x_j$$

$$y_k = \varphi(u_k + b_k)$$

Сигмоида  $f(S) = (1 + e^{-aS})^{-1}$

Гиперболический тангенс

$$f(S) = \tanh(S) = \frac{e^S - e^{-S}}{e^S + e^{-S}}$$

ReLU  $f(S) = \max(0, S)$

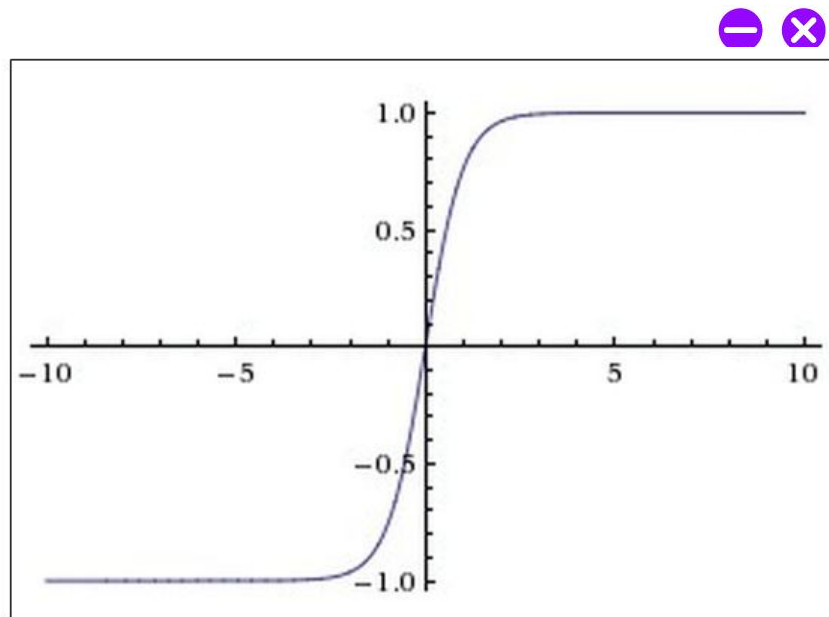
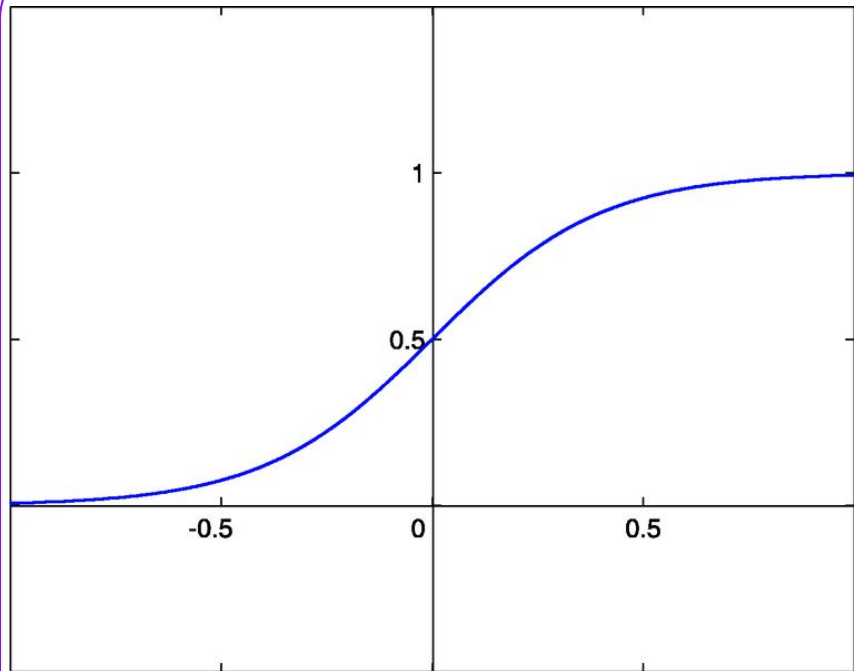
Пороговая (Хевисайда)

$$f(S) = \begin{cases} 1, & S > T \\ 0, & \text{else} \end{cases}$$

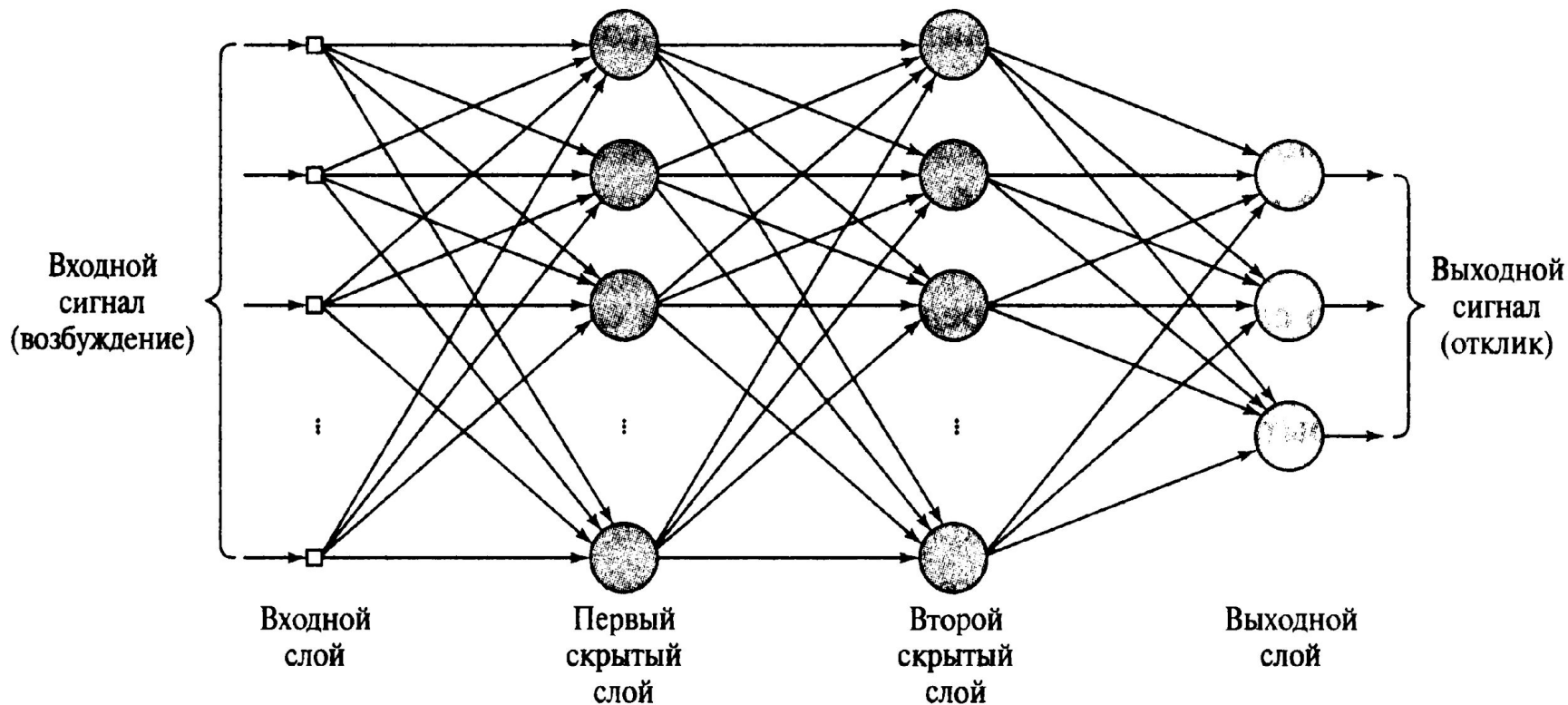


# Сигмоида

ІТМО



# Многослойный персептрон

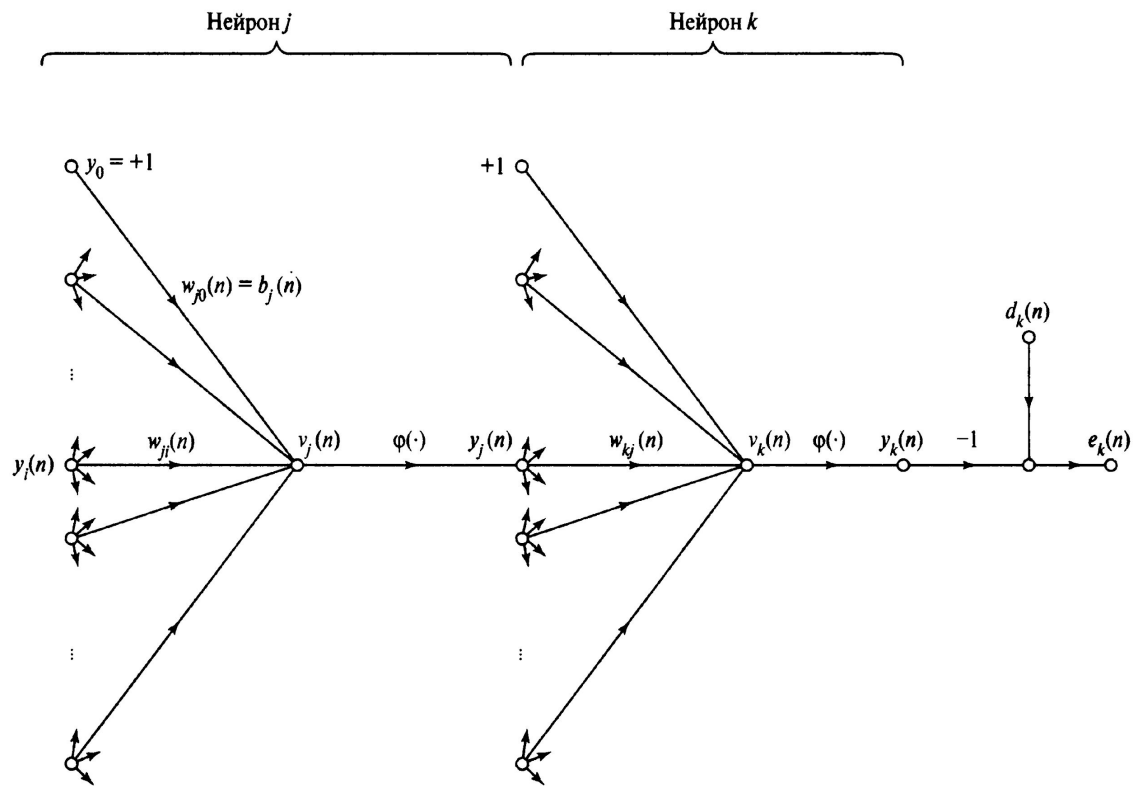


**Локальные признаки** (local feature) извлекаются в первом скрытом слое.

**Глобальные признаки** (global feature) извлекаются во втором скрытом слое. В частности, нейрон второго скрытого слоя "обобщает" выходные сигналы нейронов первого скрытого слоя, относящихся к конкретной области входного пространства



# Распространение сигнала



# Ошибка обучения

$e_j(n) = d_j(n) - y_j(n)$  - сигнал ошибки

$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$  - общая энергия ошибки сети (C – количество классов)

$E_{av}(n) = \frac{1}{N} \sum_{n=1}^N E(n)$  - функция стоимости (cost function) – мера эффективности обучения (N – количество примеров)

Другая функция стоимости – перекрестная энтропия (cross-entropy):

$$H(P, Q) = - \sum_x P(x) \log_2 Q(x)$$

$P$  – распределение истинных ответов

$Q$  – распределение вероятностей прогнозов модели

$$CE = -(y \log_2(p) + (1 - y) \log_2(1 - p))$$

$y$  – индикатор правильности присвоения метки класса для экземпляра



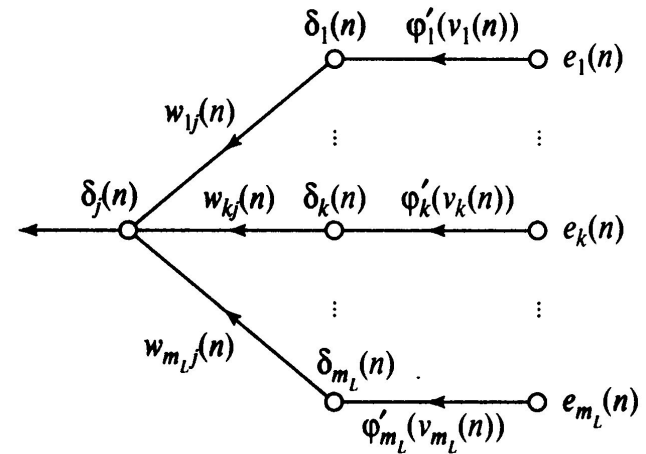
# Обратное распространение ошибки ИТМО

$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$  - дельта-правило

$\delta_j(n) = e_j(n) \varphi'_j(v_j(n))$  - локальный градиент

$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$

$\varphi'_j(v_j(n)) = \alpha y_j(n) [1 - y_j(n)]$  для СИГМОИДЫ



# Скорость обучения

Если параметр  $\eta$  мал, алгоритм замедляется (overdamped), и траектория изменения  $w_{ji}(n)$  соответствует гладкой кривой.

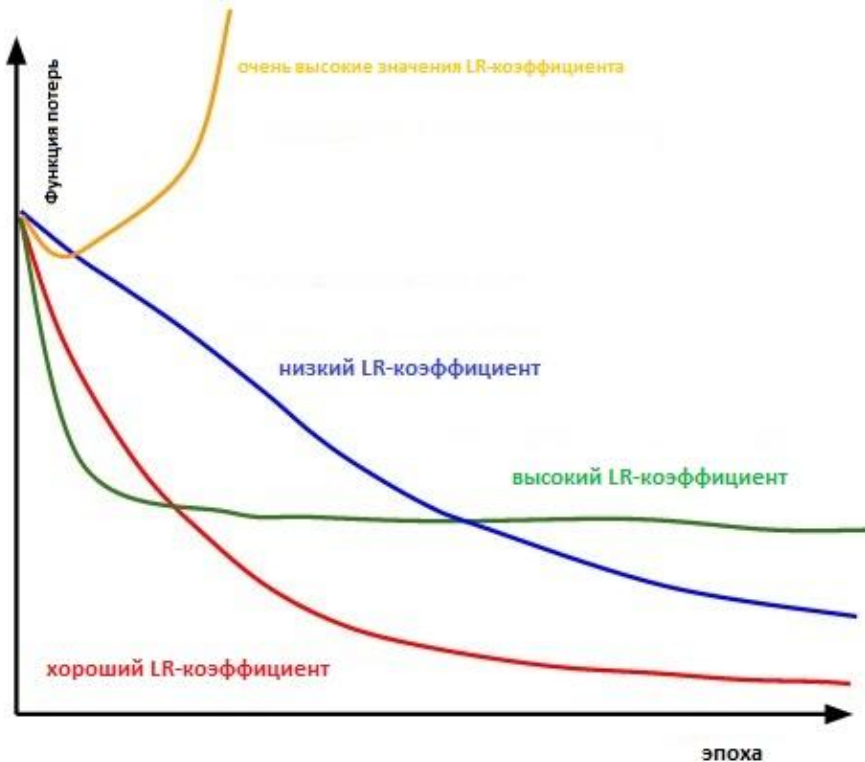
Если параметр  $\eta$  велик, алгоритм ускоряется (underdamped), и траектория  $w_{ji}(n)$  принимает зигзагообразный вид.

Если параметр  $\eta$  превосходит некоторое критичное значение, алгоритм становится неустойчивым (т.е. расходящимся).



# Влияние скорости на процесс обучения

ІТМО





# Свертка



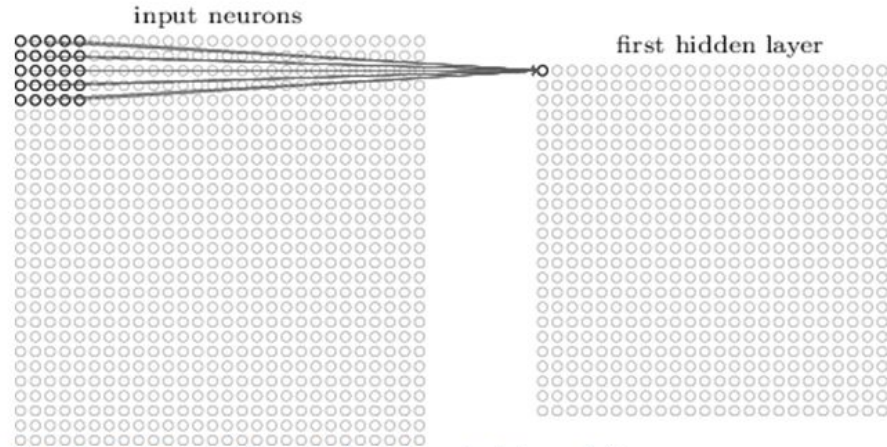
1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved  
Feature

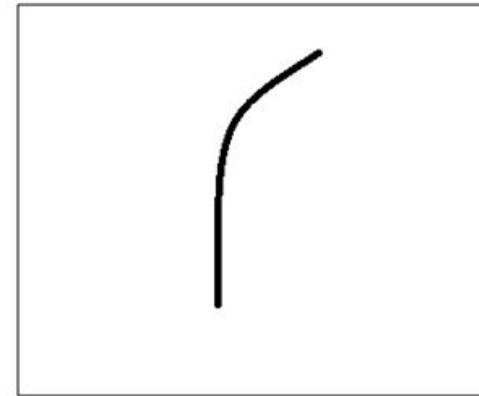
# Свертка



Visualization of 5 x 5 filter convolving around an input volume and producing an activation map

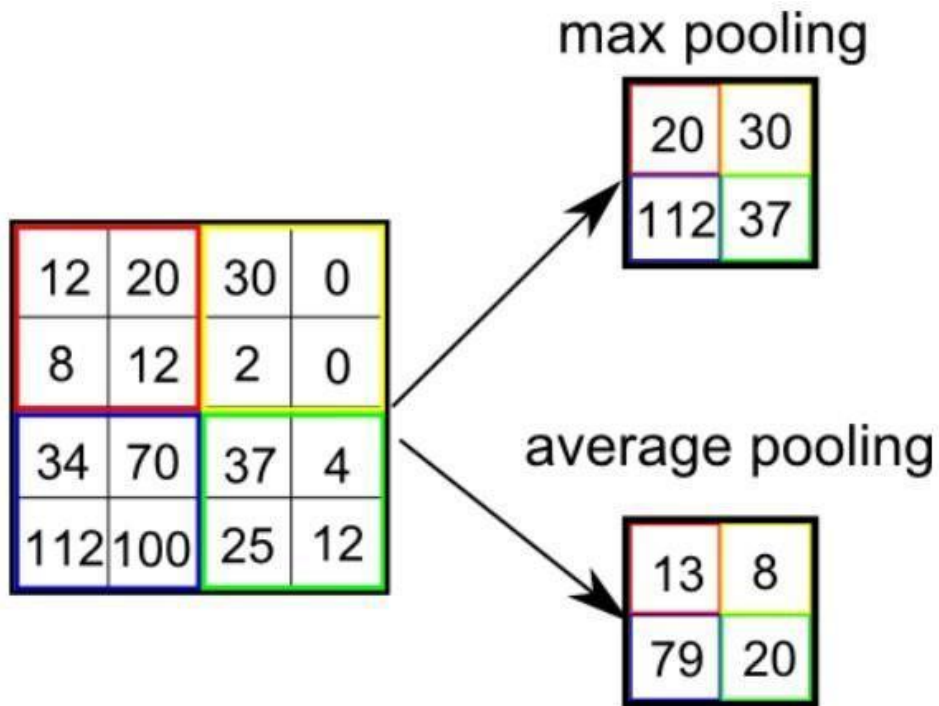
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

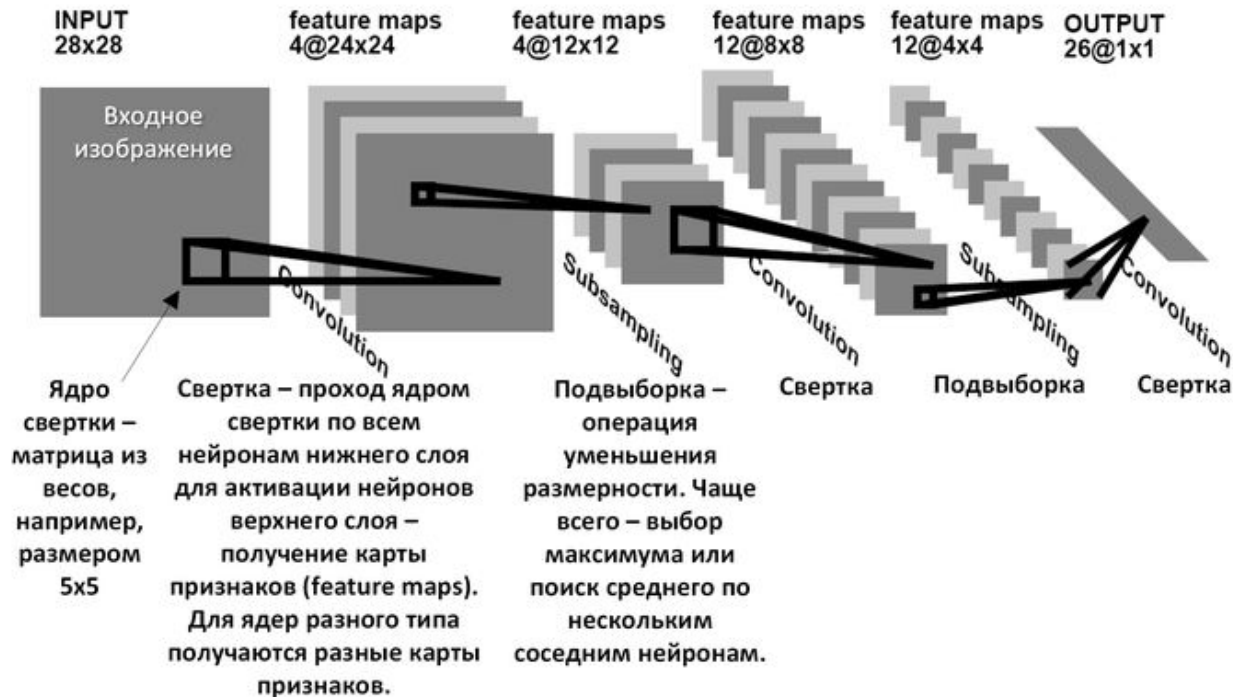


Visualization of a curve detector filter

# Пулинг



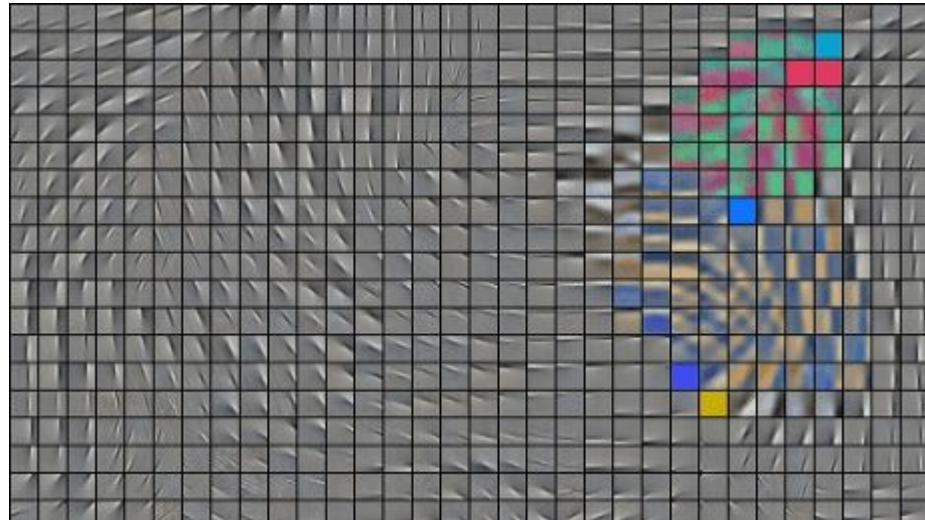
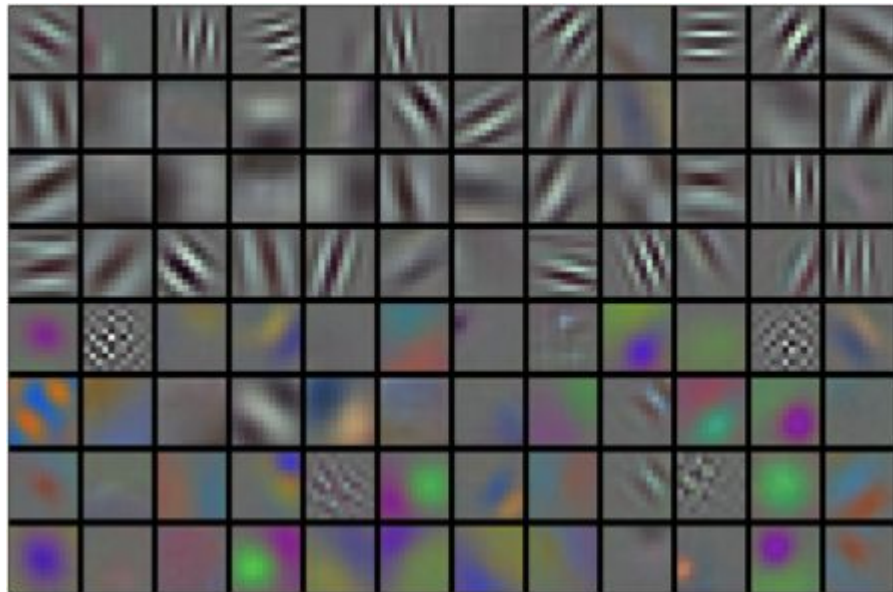
# Le-Net 5



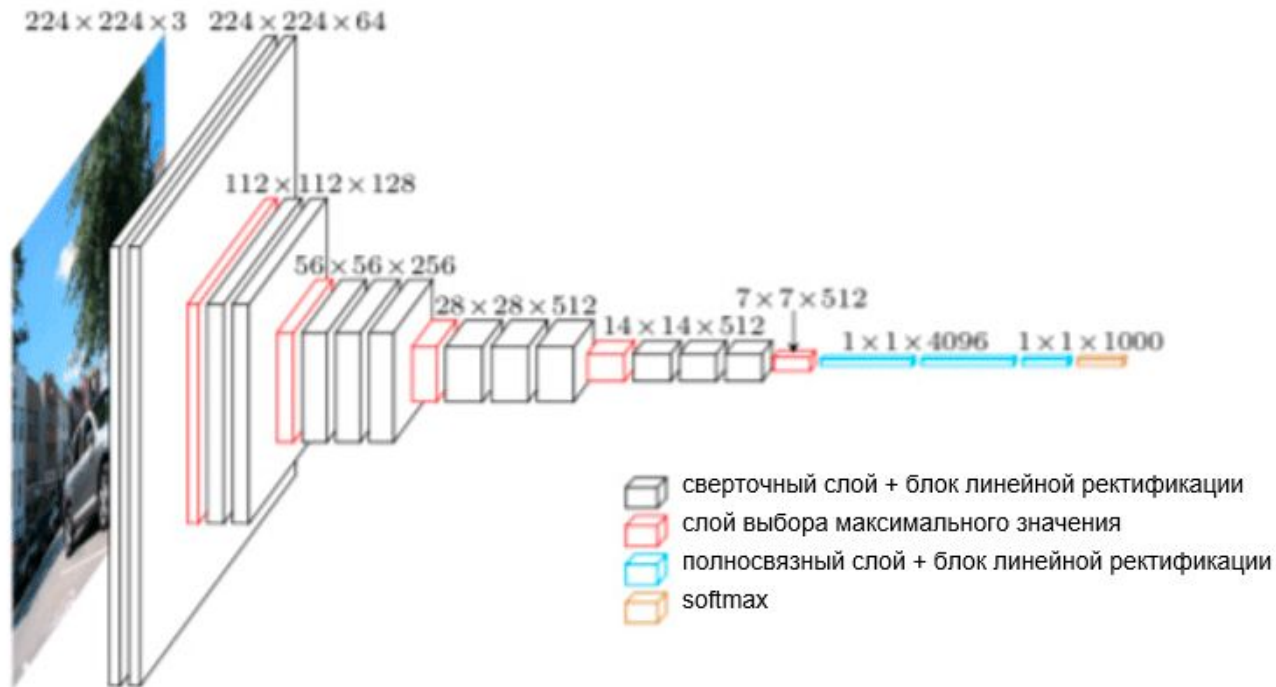


# Ядра сверток

ІТМО



# VGG



# Регуляризация

Стратегии уменьшения ошибки на тестовой выборке, возможно, за счет увеличения на обучающей выборке.

Смысл регуляризационной оценки – увеличение смещения в обмен на уменьшение дисперсии

$$J(w) = MSE + \lambda$$

$\lambda$  - регуляризатор

Чем больше  $\lambda$ , тем сильнее стремление к малым весам. При этом, при слишком больших  $\lambda$  может быть недообучение.





# Штрафы по норме параметров

$$\tilde{J}(w) = J + \alpha\Omega(\theta)$$



$\alpha$  – вес

$\Omega(\theta)$  – член, штрафующий по норме

Предпочитается штрафовать веса, исключая смещения

- Регуляризация Тихонова (по норме L2)
- Регуляризация по норме L1
- Оптимизация с ограничениями

# Регуляризация Тихонова

$$\Omega(\theta) = \frac{1}{2} \|w\|_2^2$$

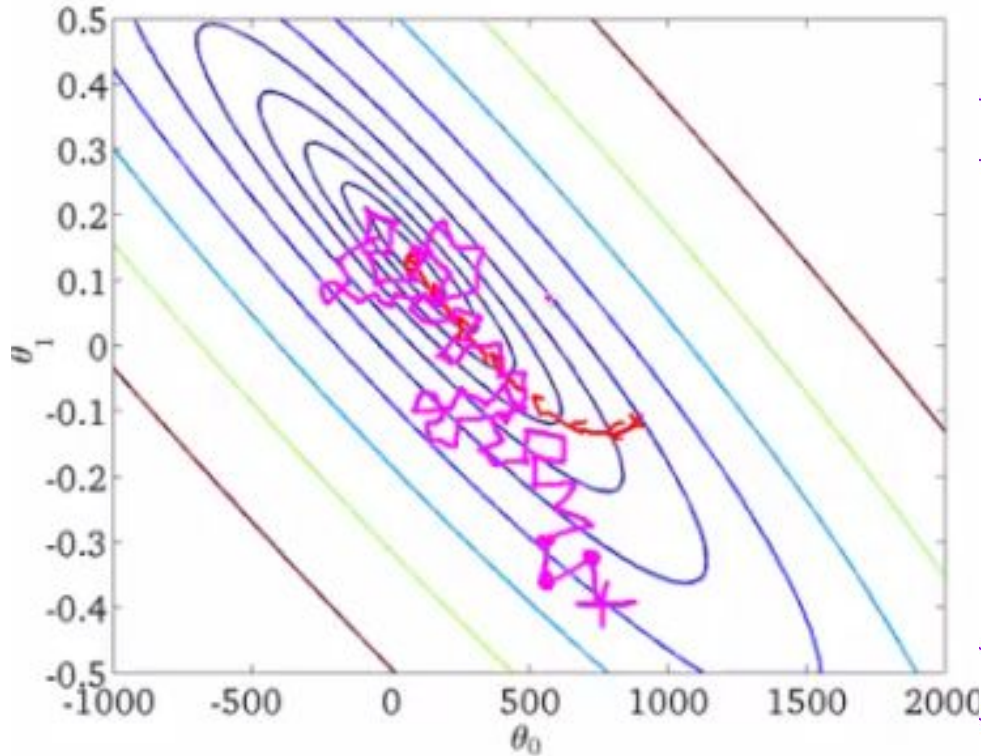
$$\tilde{J}(w, X, y) = J(w, X, y) + \frac{\alpha}{2} w^T w$$

$$\nabla_w \tilde{J}(w, X, y) = \alpha w + \nabla_w J(w, X, y)$$



# Стохастический градиентный спуск

ИТМО



Суть – обновление весов по одному объекту  
1 эпоха = 1 итерация  
На каждой эпохе не гарантируется движение в сторону наискорейшего убывания функции



**Спасибо  
за внимание!**

**ITMO** *re than a*  
**UNIVERSITY**

[a-kugaevskikh@yandex.ru](mailto:a-kugaevskikh@yandex.ru)