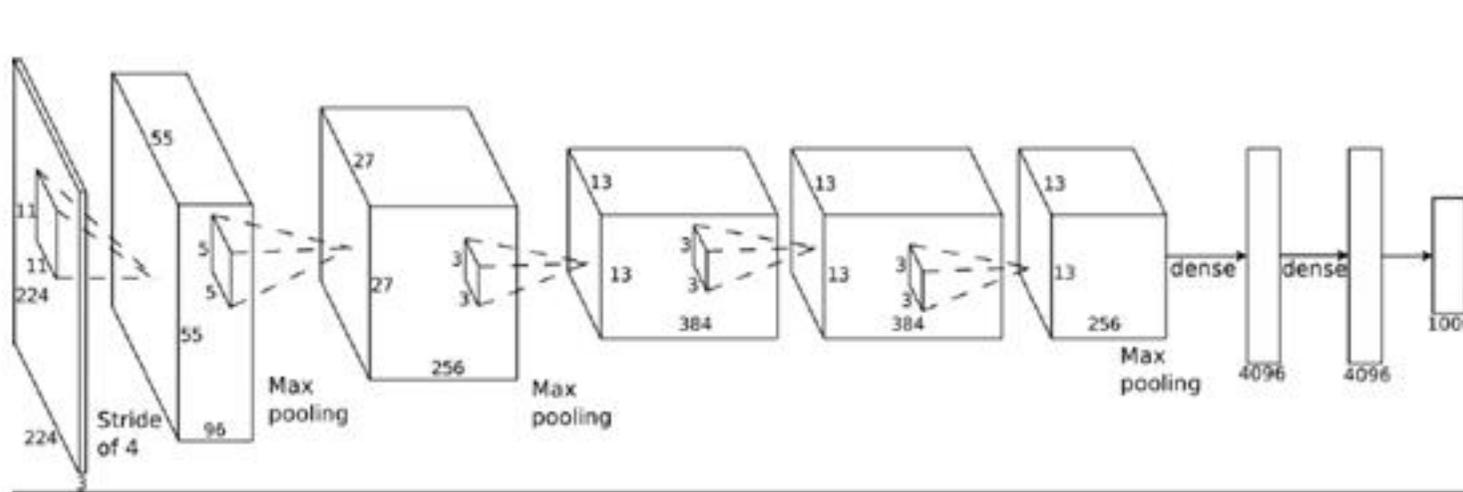
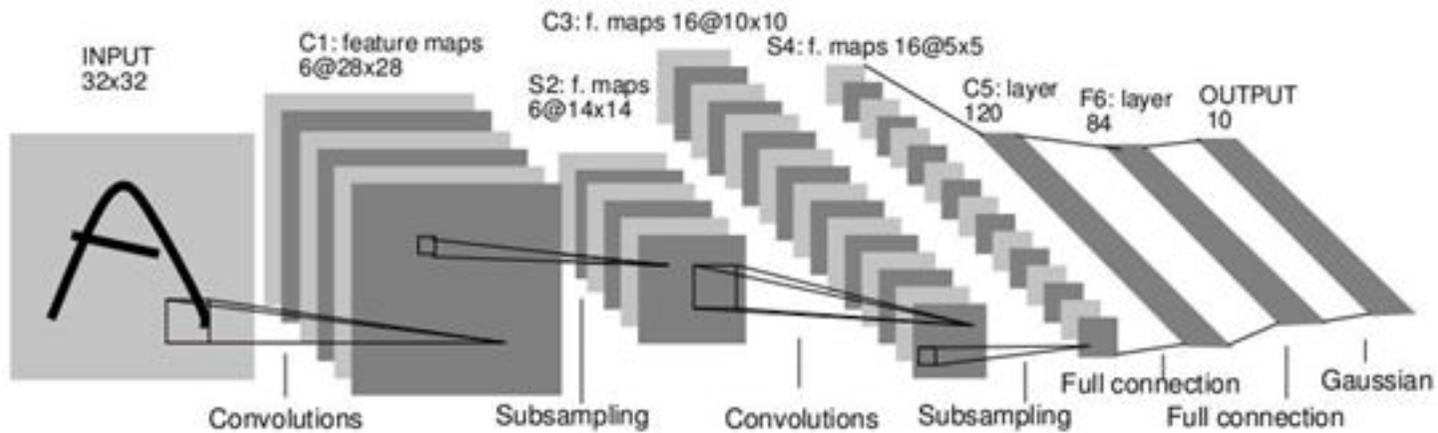


Нейросетевые алгоритмы



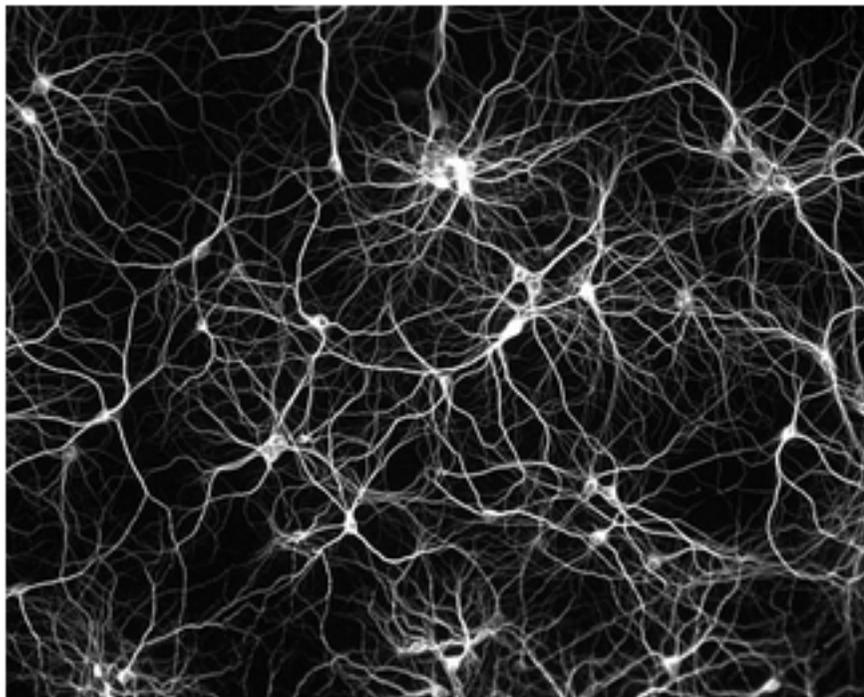
Антон Конушин

Благодарность



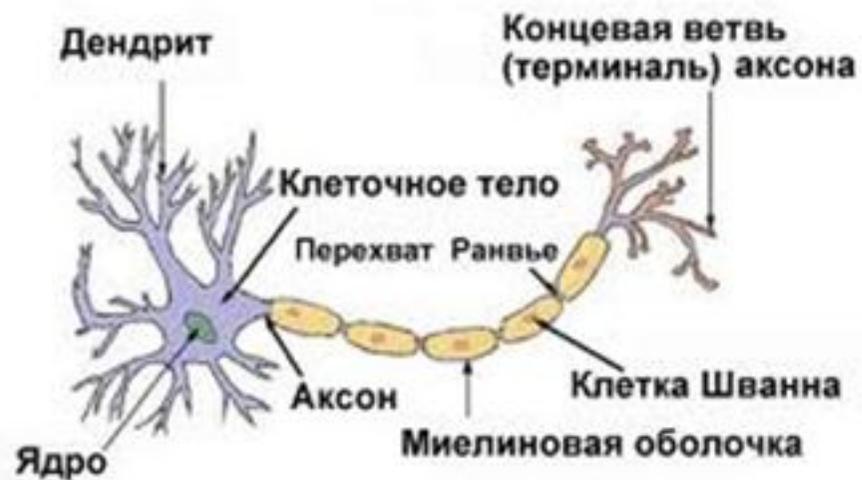
В лекции использованы материалы курса
«Машинное обучение» К.В. Воронцова

Структура мозга



Нейросеть

Типичная структура нейрона



Отдельный
нейрон

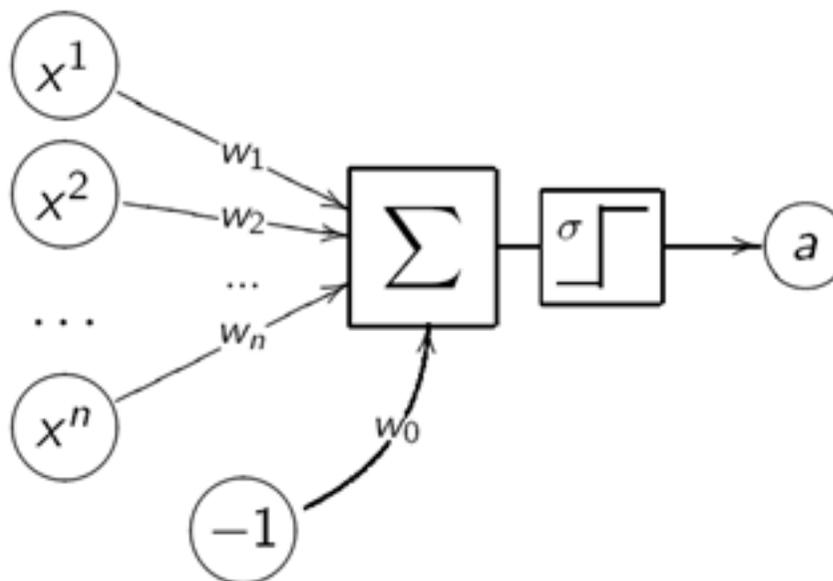


Математическая модель нейрона

Линейная модель нейрона МакКаллока-Питтса [1943]:

$$a(x, w) = \sigma(\langle w, x \rangle) = \sigma\left(\sum_{j=1}^n w_j f_j(x) - w_0\right),$$

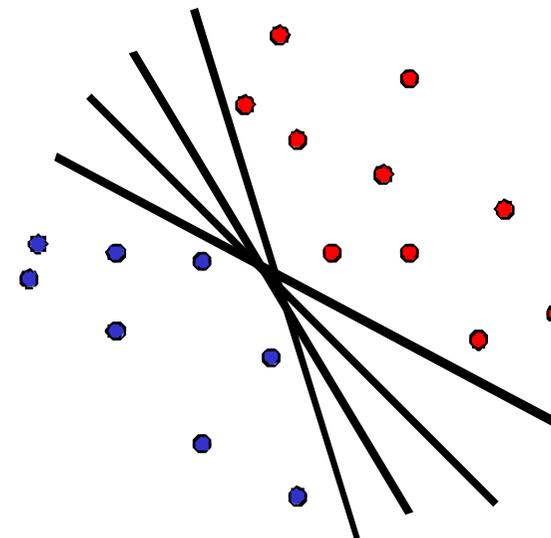
где $\sigma(s)$ — функция активации (в частности, sign).





Нейрон = линейный классификатор

$$a(x, w) = \text{sign} \left(\sum_{j=1}^n w_j f_j(x) - w_0 \right)$$



- Нейрон задаёт отдельный линейный классификатор
- Обучать его (настраивать веса) мы можем как угодно



Градиентный метод обучения

Минимизация аппроксимированного эмпирического риска:

$$Q(w; X^\ell) = \sum_{i=1}^{\ell} \mathcal{L}(\langle w, x_i \rangle y_i) \rightarrow \min_w.$$

Численная минимизация методом *градиентного спуска*:

$w^{(0)}$:= начальное приближение;

$$w^{(t+1)} := w^{(t)} - \eta \cdot \nabla Q(w^{(t)}), \quad \nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j} \right)_{j=0}^n,$$

где η — *градиентный шаг*, называемый также *темпом обучения*.

$$w^{(t+1)} := w^{(t)} - \eta \sum_{i=1}^{\ell} \mathcal{L}'(\langle w^{(t)}, x_i \rangle y_i) x_i y_i.$$

Идея ускорения сходимости:

брать (x_i, y_i) по одному и сразу обновлять вектор весов.

SG: Стохастический градиентный спуск



Вход:

выборка X^ℓ ; темп обучения η ; параметр λ ;

Выход:

веса w_0, w_1, \dots, w_n ;

- 1: инициализировать веса w_j , $j = 0, \dots, n$;
- 2: инициализировать текущую оценку функционала:
 $Q := \sum_{i=1}^{\ell} \mathcal{L}(\langle w, x_i \rangle y_i)$;
- 3: **повторять**
- 4: выбрать объект x_i из X^ℓ (например, случайно);
- 5: вычислить потерю: $\varepsilon_i := \mathcal{L}(\langle w, x_i \rangle y_i)$;
- 6: градиентный шаг: $w := w - \eta \mathcal{L}'(\langle w, x_i \rangle y_i) x_i y_i$;
- 7: оценить значение функционала: $Q := (1 - \lambda)Q + \lambda \varepsilon_i$;
- 8: **пока** значение Q и/или веса w не стабилизируются;



Спуск с правилом Хабба

Задача классификации: $X = \mathbb{R}^{n+1}$, $Y = \{-1, +1\}$,

$$\mathcal{L}(a, y) = (-\langle w, x \rangle y)_+.$$

Линейный классификатор:

$$a(x, w) = \text{sign}\langle w, x \rangle.$$

Градиентный шаг — **правило Хэбба** [1949]:

$$\text{если } \langle w, x_i \rangle y_i < 0 \text{ то } w := w + \eta x_i y_i,$$

Если $X = \{0, 1\}^n$, $Y = \{0, +1\}$, то правило Хэбба переходит в правило **перцептрона Розенблатта** [1957]:

$$w := w - \eta (a(x_i, w) - y_i) x_i.$$



Обоснование

Задача классификации: $X = \mathbb{R}^{n+1}$, $Y = \{-1, 1\}$.

Теорема (Новиков, 1962)

Пусть выборка X^ℓ линейно разделима:

$\exists \tilde{w}, \exists \delta > 0: \langle \tilde{w}, x_i \rangle y_i > \delta$ для всех $i = 1, \dots, \ell$.

Тогда Алгоритм SG с правилом Хэбба находит вектор весов w ,

- разделяющий обучающую выборку без ошибок;
- при любом начальном положении $w^{(0)}$;
- при любом темпе обучения $\eta > 0$;
- независимо от порядка предъявления объектов x_i ;
- за конечное число исправлений вектора w ;
- если $w^{(0)} = 0$, то число исправлений $t_{\max} \leq \frac{1}{\delta^2} \max \|x_i\|$.



Плюсы и минусы

Достоинства:

- 1 легко реализуется;
- 2 легко обобщается на любые f , \mathcal{L} ;
- 3 возможно динамическое (потокковое) обучение;
- 4 на сверхбольших выборках не обязательно брать все x_i ;

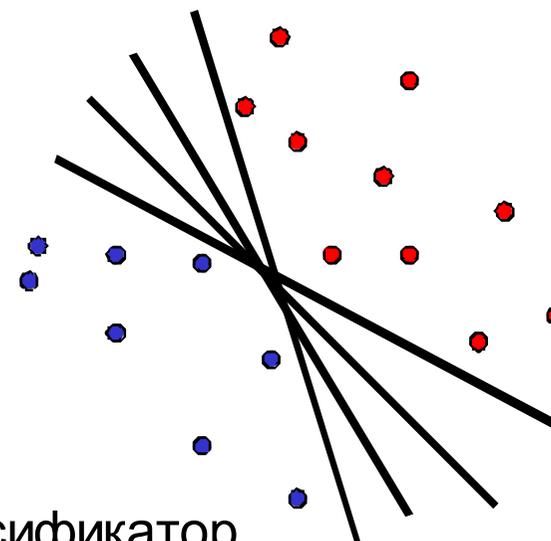
Недостатки:

- 1 возможна расходимость или медленная сходимость;
- 2 застревание в локальных минимумах;
- 3 подбор комплекса эвристик является искусством;
- 4 проблема переобучения;



Нейрон = линейный классификатор

$$a(x, w) = \text{sign} \left(\sum_{j=1}^n w_j f_j(x) - w_0 \right)$$



- Нейрон задаёт отдельный линейный классификатор
- Обучать его (настраивать веса) мы можем как угодно
- Градиентный спуск (SG) – один из методов настройки с кучей недостатков
- SVM – другой метод настройки линейных классификаторов, максимизирующий отступ



Масштаб задач

Задача классификации: $Y = \{\pm 1\}$, $a(x, w) = \text{sign}\langle w, x_i \rangle$;

$$Q(w; X^\ell) = \sum_{i=1}^{\ell} \mathcal{L}(\underbrace{\langle w, x_i \rangle y_i}_{M_i(w)}) \rightarrow \min_w;$$

Задача регрессии: $Y = \mathbb{R}$, $a(x, w) = \sigma(\langle w, x_i \rangle)$;

$$Q(w; X^\ell) = \sum_{i=1}^{\ell} (\sigma(\langle w, x_i \rangle) - y_i)^2 \rightarrow \min_w;$$

**Насколько богатый класс функций реализуется нейроном?
А сетью (суперпозицией) нейронов?**



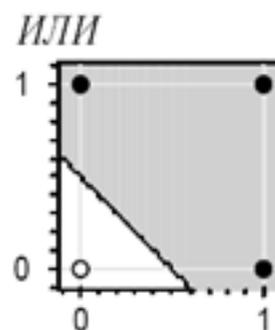
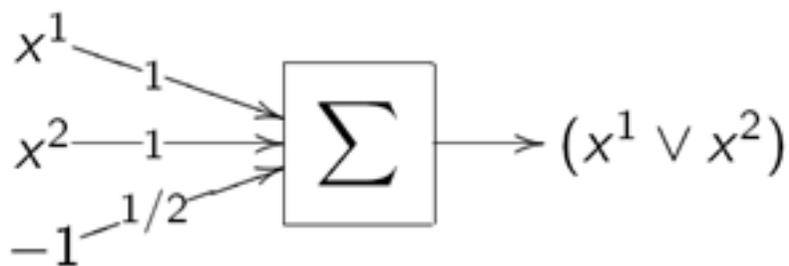
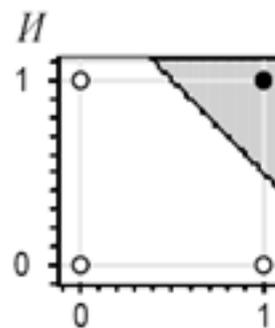
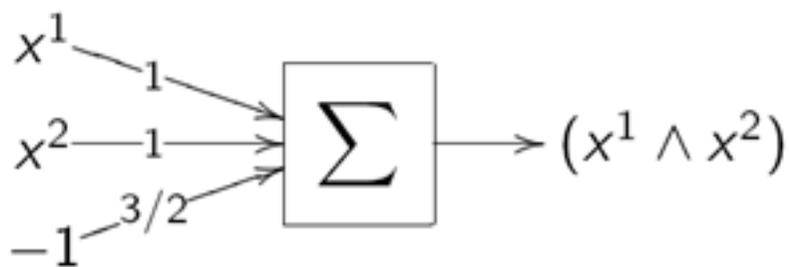
Логические элементы

Функции И, ИЛИ, НЕ от бинарных переменных x^1 и x^2 :

$$x^1 \wedge x^2 = [x^1 + x^2 - \frac{3}{2} > 0];$$

$$x^1 \vee x^2 = [x^1 + x^2 - \frac{1}{2} > 0];$$

$$\neg x^1 = [-x^1 + \frac{1}{2} > 0];$$



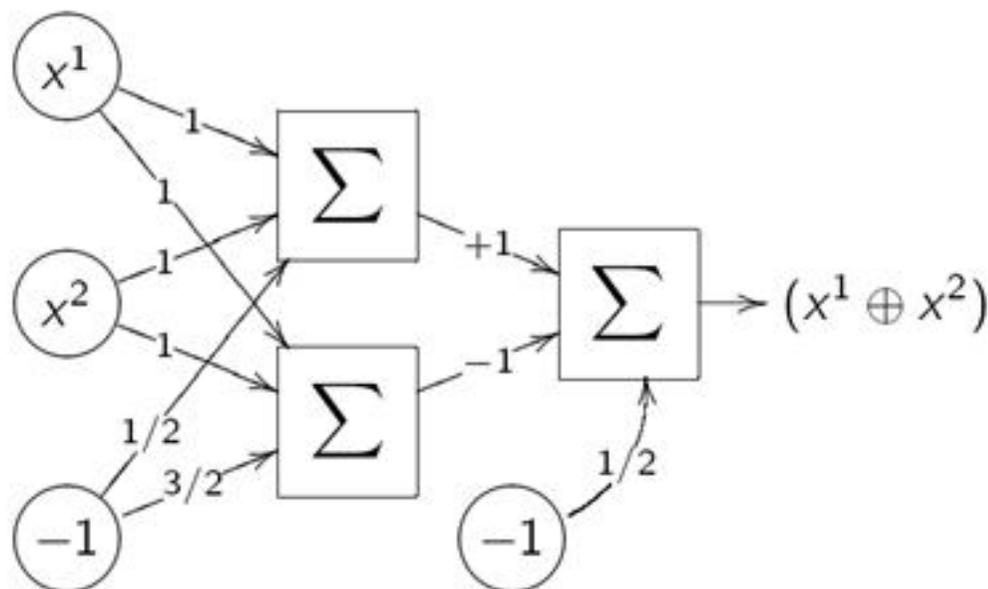


Исключающее ИЛИ (XOR)

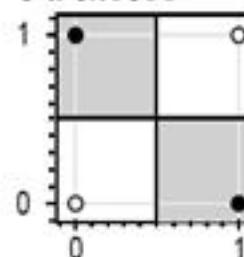
Функция $x^1 \oplus x^2 = [x^1 \neq x^2]$ не реализуема одним нейроном.

Два способа реализации:

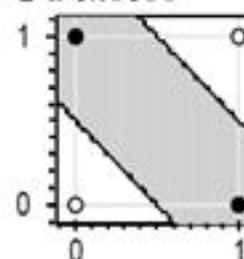
- Добавлением нелинейного признака:
 $x^1 \oplus x^2 = [x^1 + x^2 - 2x^1x^2 - \frac{1}{2} > 0]$;
- **Сетью** (двухслойной суперпозицией) функций И, ИЛИ, НЕ:
 $x^1 \oplus x^2 = [(x^1 \vee x^2) - (x^1 \wedge x^2) - \frac{1}{2} > 0]$.



1-й способ



2-й способ



Какие функции можно приближать нейросетью?



Утверждение

Любая булева функция представима в виде ДНФ, следовательно, и в виде двухслойной сети.

Решение тринадцатой проблемы Гильберта:

Теорема (Колмогоров, 1957)

Любая непрерывная функция n аргументов на единичном кубе $[0, 1]^n$ представима в виде суперпозиции непрерывных функций одного аргумента и операции сложения:

$$f(x^1, x^2, \dots, x^n) = \sum_{k=1}^{2n+1} h_k \left(\sum_{i=1}^n \varphi_{ik}(x^i) \right),$$

где h_k, φ_{ik} — непрерывные функции, и φ_{ik} не зависят от f .



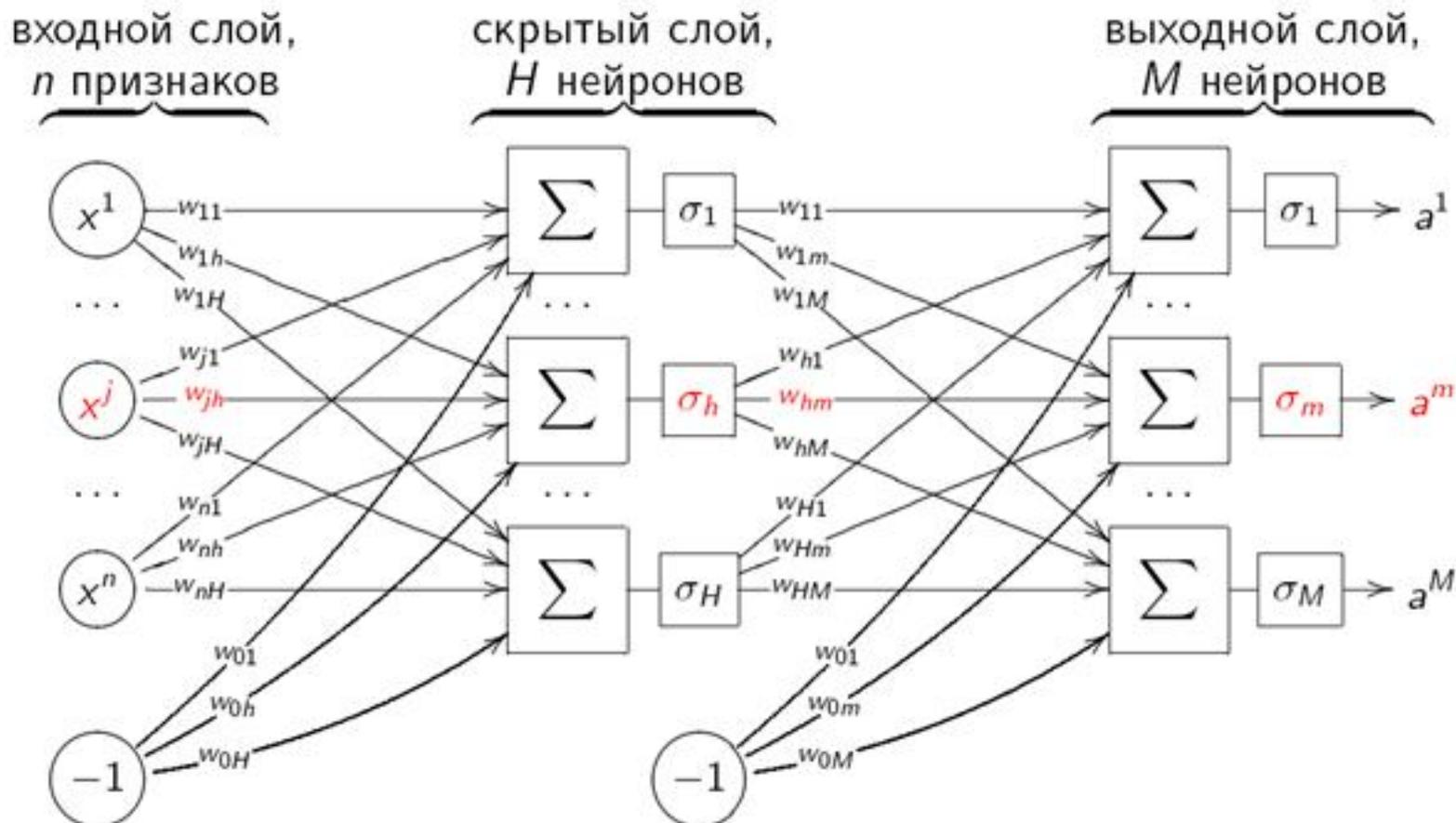
Представимость функций

- С помощью линейных операций и одной нелинейной функции активации можно вычислить любую непрерывную функцию с любой желаемой точностью
- Однако эти теоремы ничего не говорят о числе слоёв нейронной сети и о числе нейронов в каждом слое
- Архитектуру сети приходится подбирать экспериментально



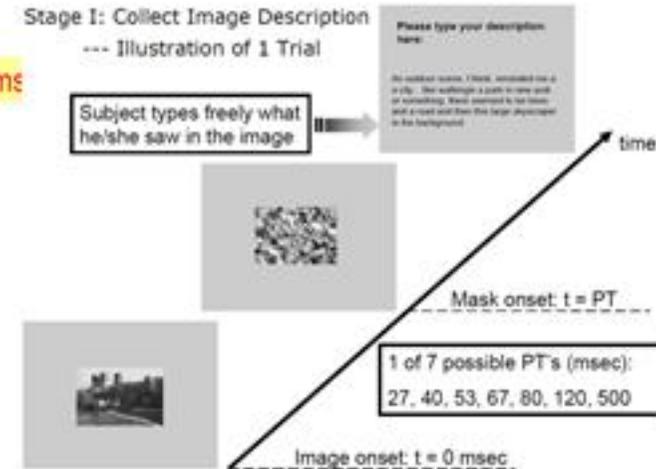
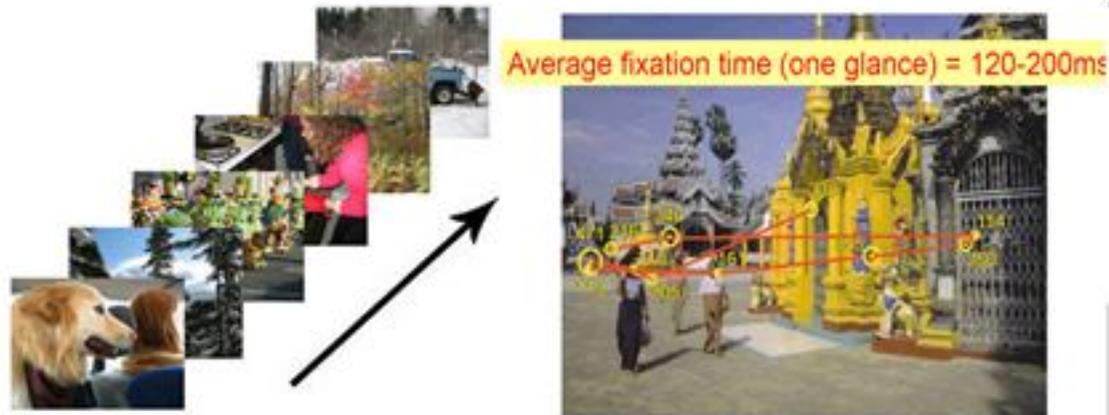
Многослойная нейросеть

Пусть для общности $Y = \mathbb{R}^M$, для простоты слоёв только два.

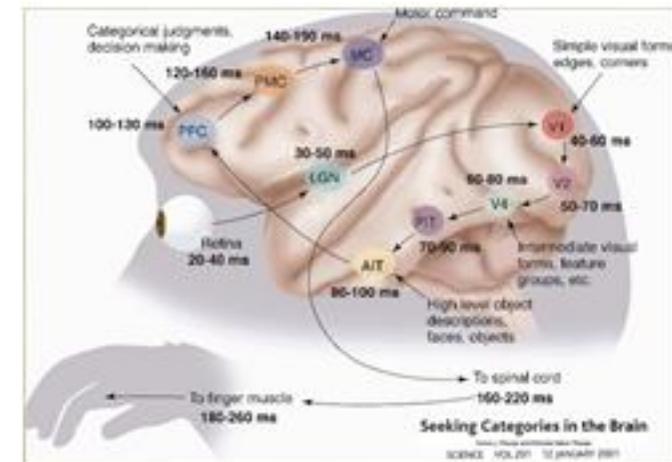


- Передача сигналов идёт в одном направлении (feed-forward)
- Сеть можно разделить на «слои», по числу предшествующих нейронов на пути сигнала

Скорость обработки

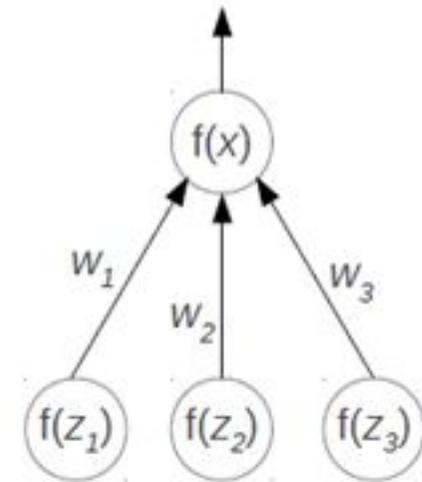
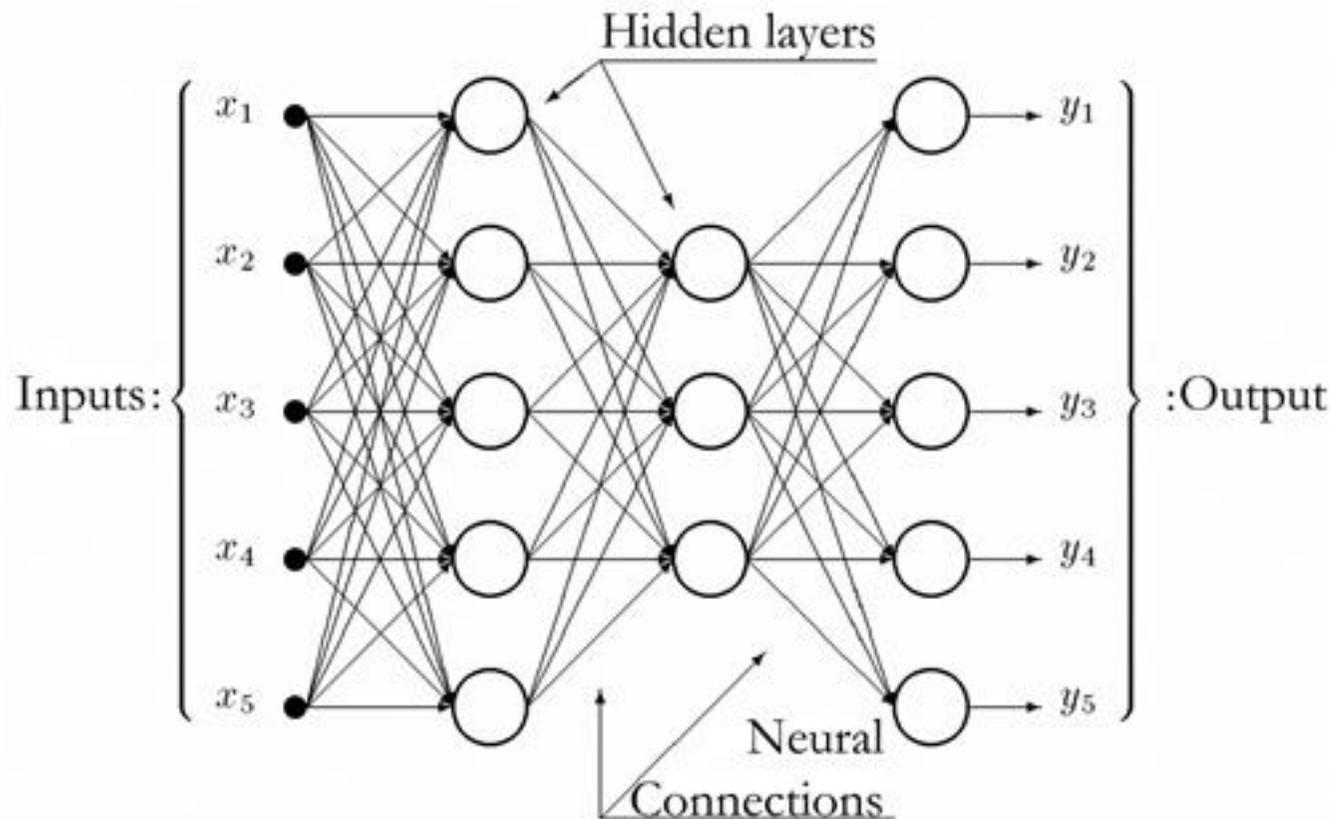


- Эксперименты по оценке скорости распознавания отдельных и серий изображений в отсутствие априорной информации о содержании изображений (заранее неизвестно, что будет в изображении)
 - Мэри Поттер, Фей-Фей Ли
- Очень многое человек успевает извлечь за 1 взгляд (~100-200 миллисек)
- Скорость распознавания заставляет предполагать, что в таком режиме распознавание не требует обратных связей
- Это некоторое обоснование «feed-forward architectures», т.е. без обратных связей





Масштабность задачи



$$x = w_1 f(z_1) + w_2 f(z_2) + w_3 f(z_3)$$

x is called the total input to the neuron, and $f(x)$ is its output

- У каждой связи есть «вес» – параметр
- Количество нейронов и количество связей определяет количество параметров
- Огромное количество параметров



Как обучать?

- **Эволюционная стратегия**
 - Случайно меняем (jitter) веса и оставляем, если сеть улучшилась
 - Рассматривать не будем
- **Отрицание проблемы (denial)**
 - Однослойные сети
 - Эвристическое задание параметров
- **Прокрастинация**
 - Откладываем главную задачу «на потом», обучаем веса на каждом слое так, чтобы они извлекали какую-то полезную информацию
- **Оптимизация всех параметров в совокупности за счёт матана (calculus)**
 - Градиентный спуск – обратное распространение ошибки

Эвристическое задание параметров

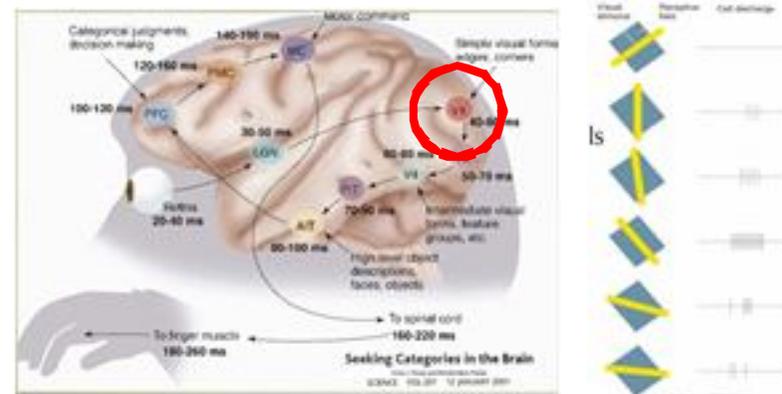


Знания о зрении человека

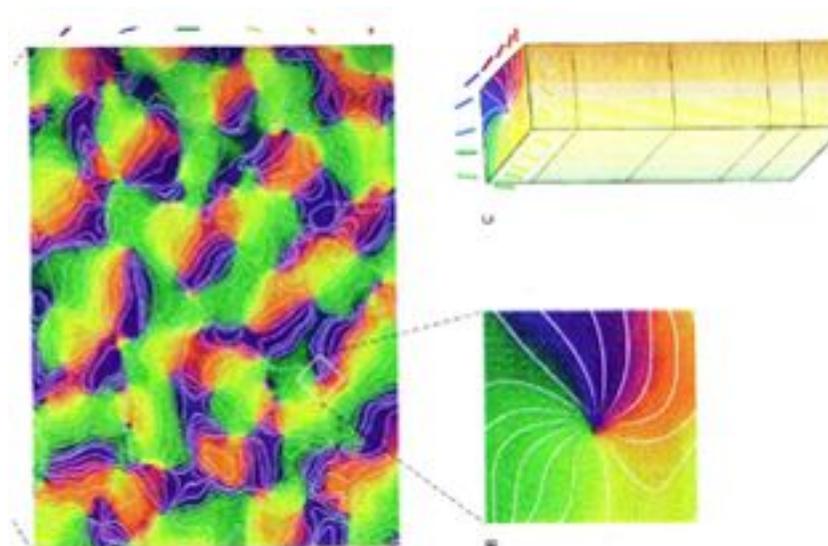
Воспользуемся знаниями об обработке визуальной информации в мозге человека.



Интуитивно кажется, что основная информация в картинке содержится в краях (границах)

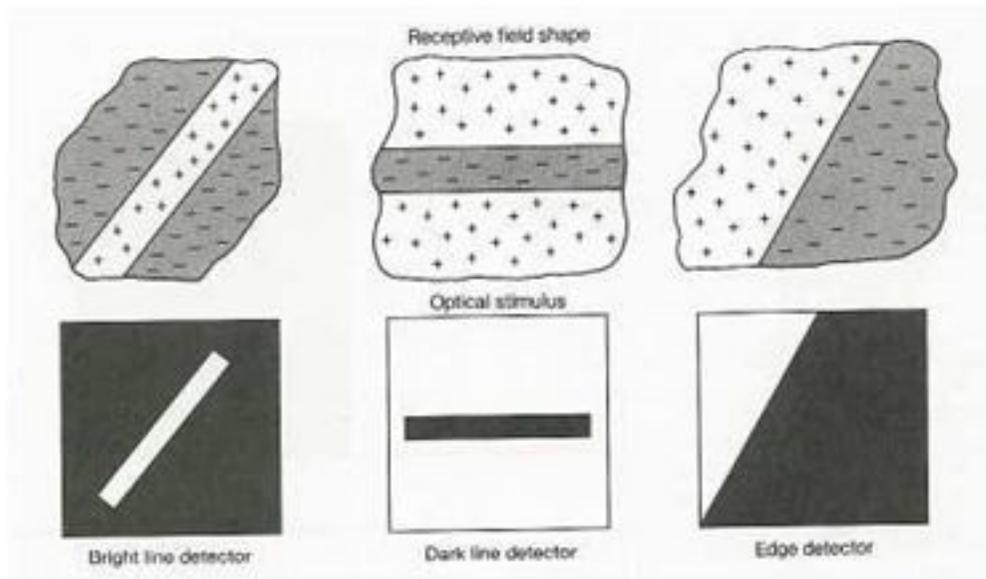


- В первичной визуальной коре головного мозга есть клетки, чувствительные к краям определенной ориентации
- Для каждой области есть набор таких клеток, чувствительные к краям разной ориентации



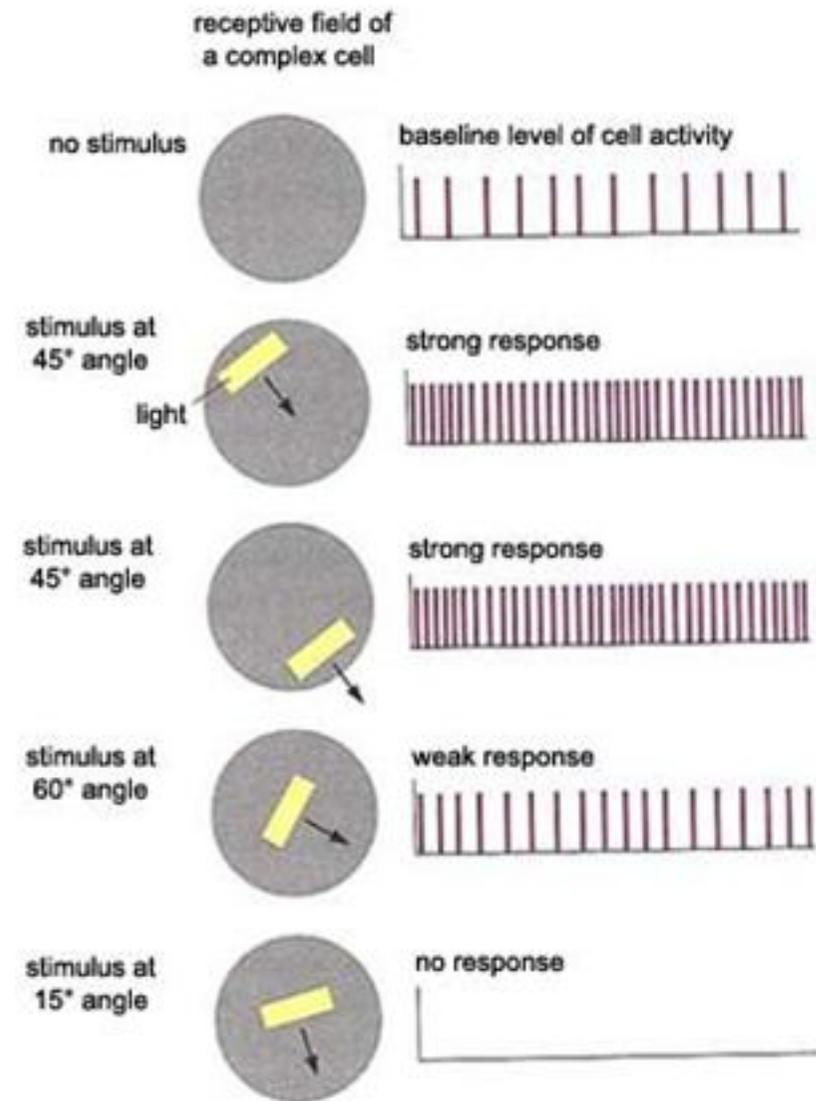


Простые (S) и сложные (C) клетки



Простые клетки

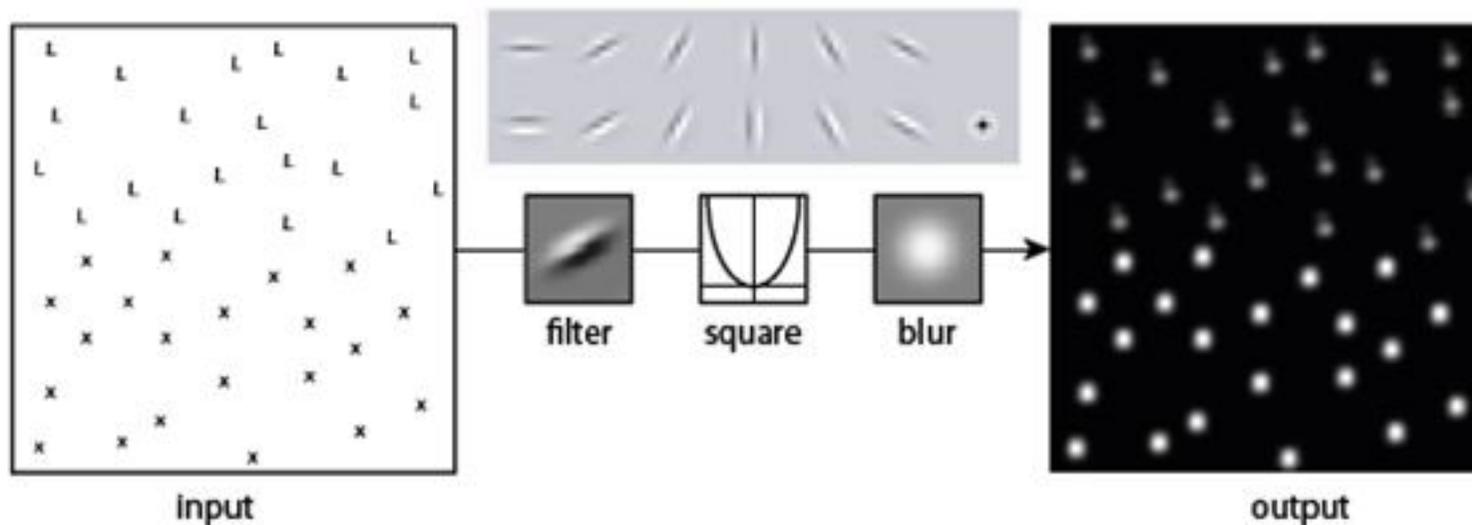
- Простые клетки чувствительны к контрастным объектам определённого размера, ориентации и положения
- Сложные клетки *инвариантны* к сдвигам в небольшой окрестности
- Как их смоделировать?



Сложные клетки



Банки текстурных фильтров



- Выберем набор (банк) фильтров, каждый из которых чувствителен к краю определенной ориентации и размера
- Каждый пиксель изображения после обработки банком фильтров даёт вектор признаков
- Этот вектор признаков эффективно описывает локальную текстуру окрестности пикселя

Pietro Perona and Jitendra Malik «Detecting and Localizing edges composed of steps, peaks and roofs», ICCV 1990



Фильтры Габора

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

$$x' = x \cos(\theta) + y \sin(\theta)$$

$$y' = -x \sin(\theta) + y \cos(\theta)$$

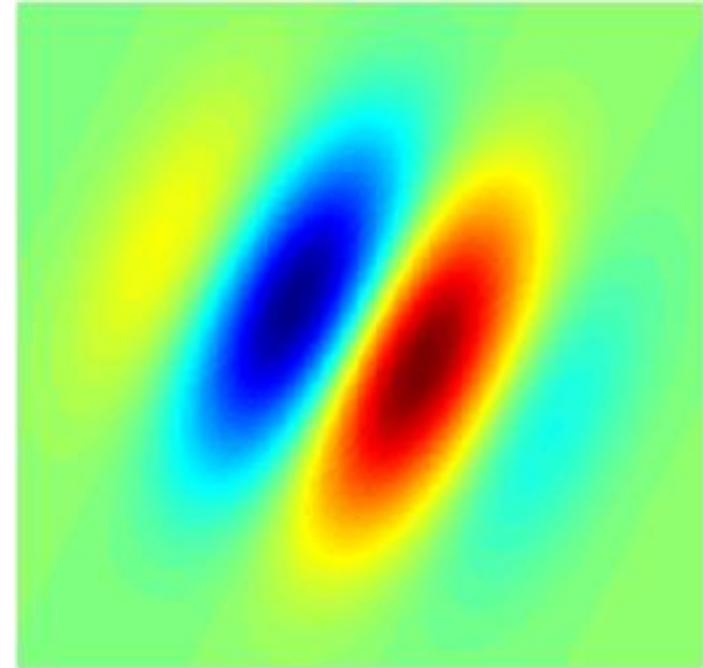
θ - ориентация

λ - длина волны

σ - сигма гауссиана

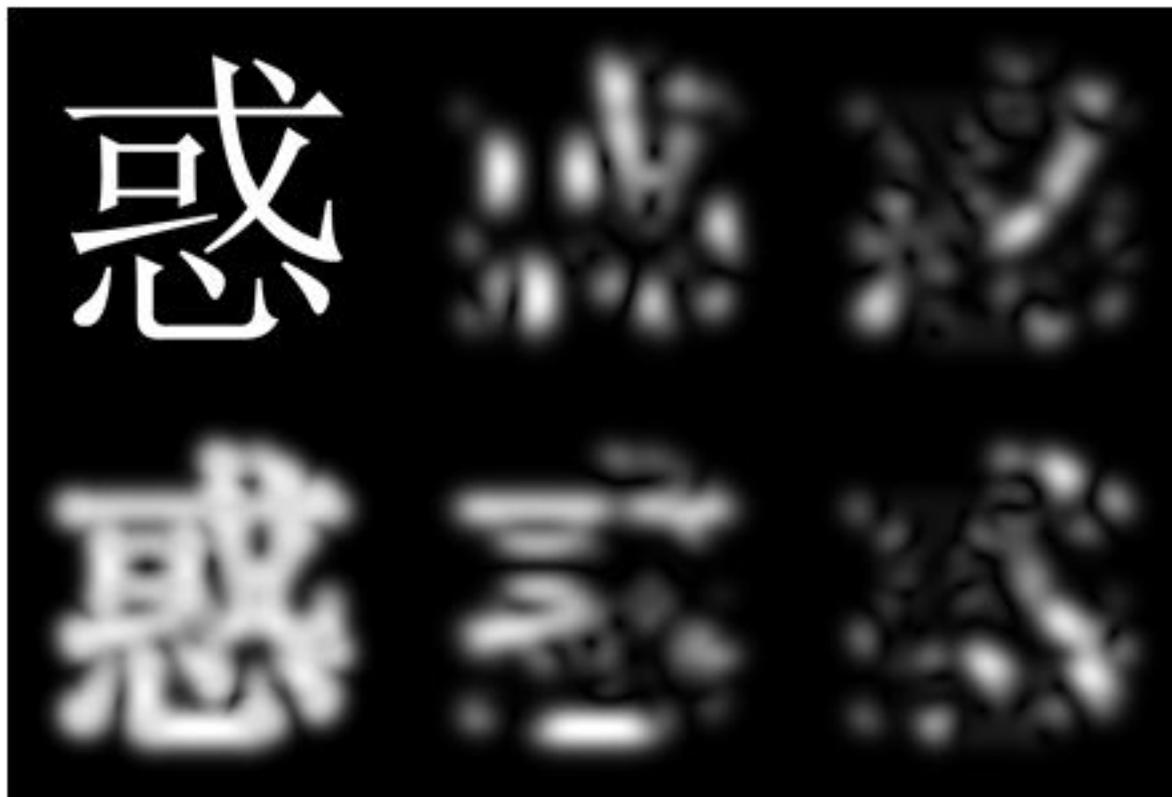
γ - соотношение размеров (aspect ratio), «эллиптичность фильтра»

ψ - сдвиг фазы



- 2D фильтр Габора – ядро гауссиана, домноженное на синусоиду
- Предложены в 1947 Денисом Габором (нобелевским лауреатом), независимо переоткрыты в 1980 году
- Позволяет сделать банк фильтров, для выделения краёв разной ориентации, масштаба и положения в окрестности

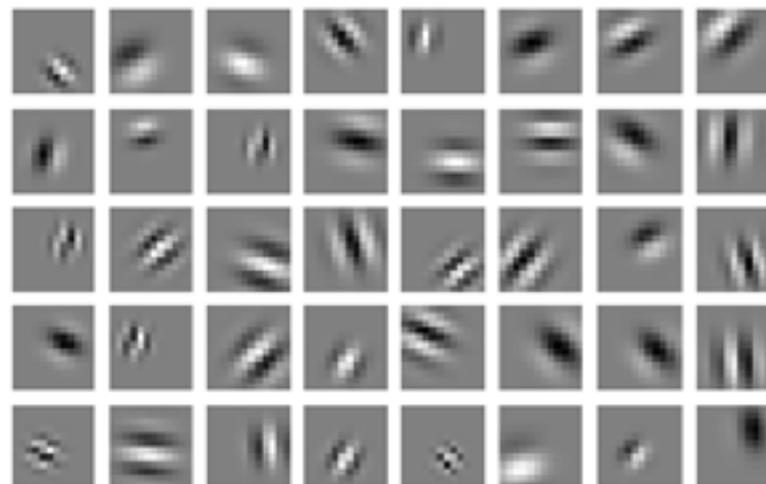
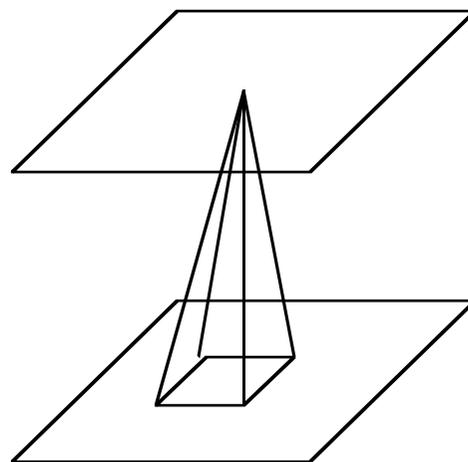
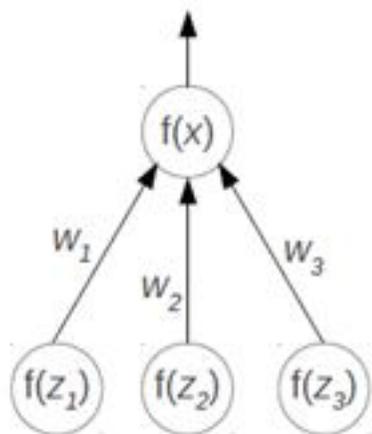
Поиск краёв с помощью Габора



- Слева вверху – иероглиф
- 4 справа – применение фильтров Габора с ориентациями 0, 45, 90 и 135 градусов
- Слева внизу – совмещение результатов фильтрации



Нейрон как линейный фильтр

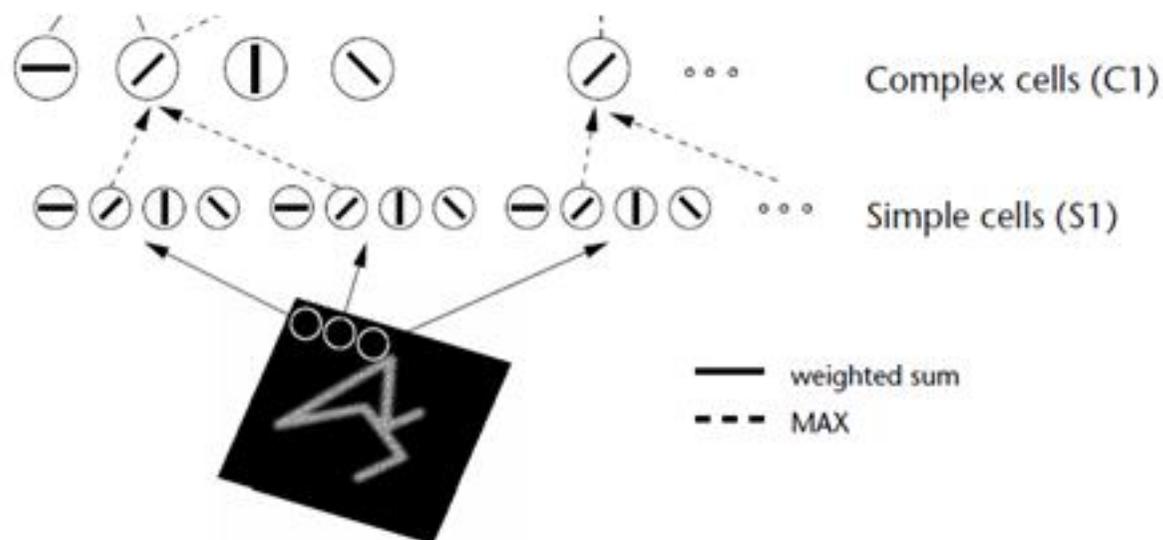


$$x = w_1 f(z_1) + w_2 f(z_2) + w_3 f(z_3)$$

- Операцию линейной фильтрации (свёртки) для одного пикселя можно реализовать одним нейроном
- Свёртку изображения целиком можно реализовать как «слой» нейронов, веса которых одинаковы
 - *Свёрточный слой*
- Набор свёрточных слоёв реализует свёртку с банком фильтров



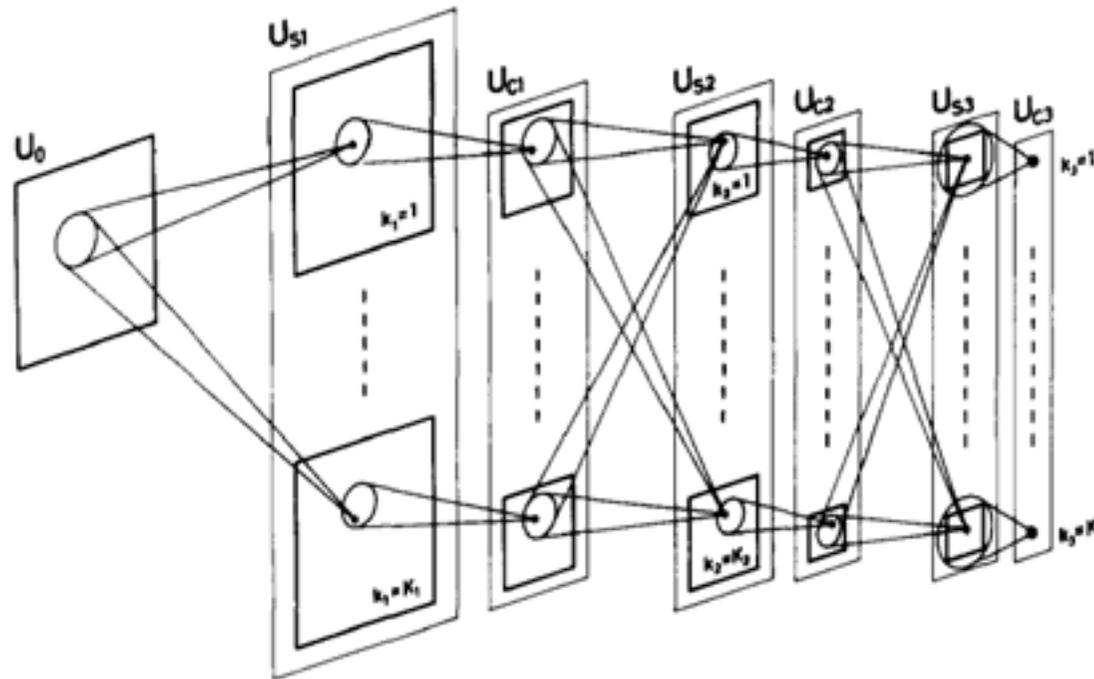
Инвариантность через MAX-pooling



Инвариантность можно обеспечить за счёт применения оператора MAX на выходах набора «простых» клеток, чувствительных к краю одной ориентации, но в разных точках одной области

Riesenhuber, M. & Poggio, T. (1999). [Hierarchical Models of Object Recognition in Cortex](#). *Nature Neuroscience* 2: 1019-1025.

Neocognitron (1980)



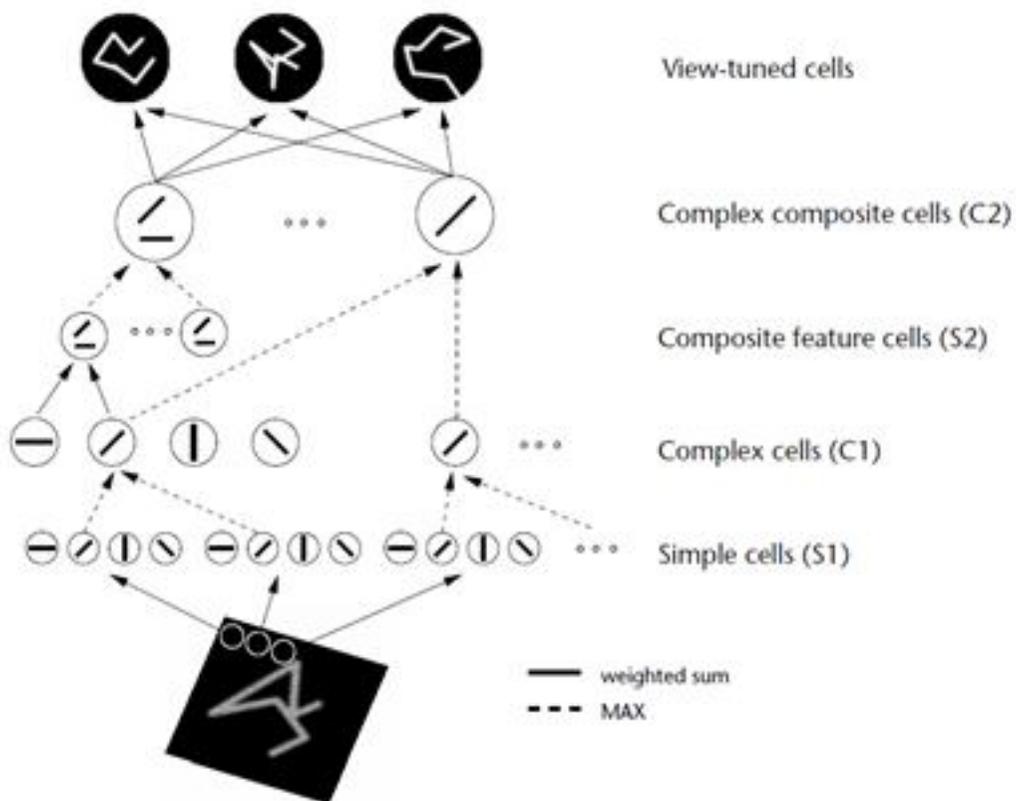
- Многослойная нейросеть с чередующимися S и C слоями
 - S-слои – линейные фильтры изображения («свёрточный слой»)
 - C-слои – MAX операторы, дающие инвариантность
- На верхнем уровне обеспечивается инвариантность по положению по всему изображению

K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4): 93-202, 1980.



Модель HMAX

Неосогнитрон с 4мя слоями следующей архитектуры:

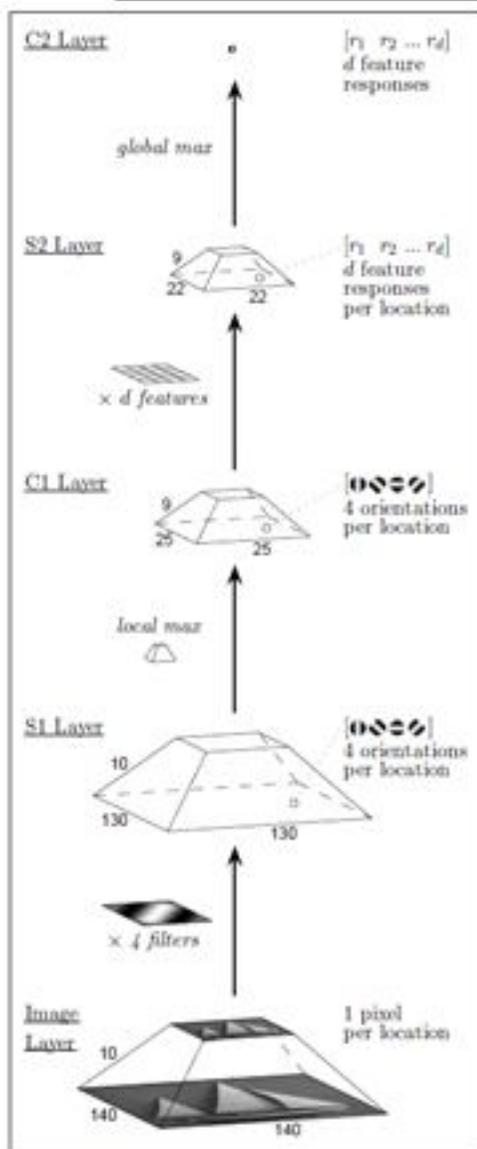


- S1 – фильтры Габора
- C1 – максимум по локальной окрестности
- S2 – фрагменты изображений из обучающей выборки как сложные признаки
- C2 – максимум по всему изображению

Riesenhuber, M. and T. Poggio. [Hierarchical Models of Object Recognition in Cortex](#), *Nature Neuroscience*, 2, 1019-1025, 1999.



Признаки в модели HMAX



Обучение:

- Выбор N случайных фрагментов
- Возможно, проверка и выбор наиболее информативных фрагментов
- Каждый фрагмент – «признак» высокого уровня

Выход алгоритма:

- N переменных, показывающих максимальный отклик каждого из N высокоуровневых признаков
- Классификация через SVM и т.д.



Распознавание событий

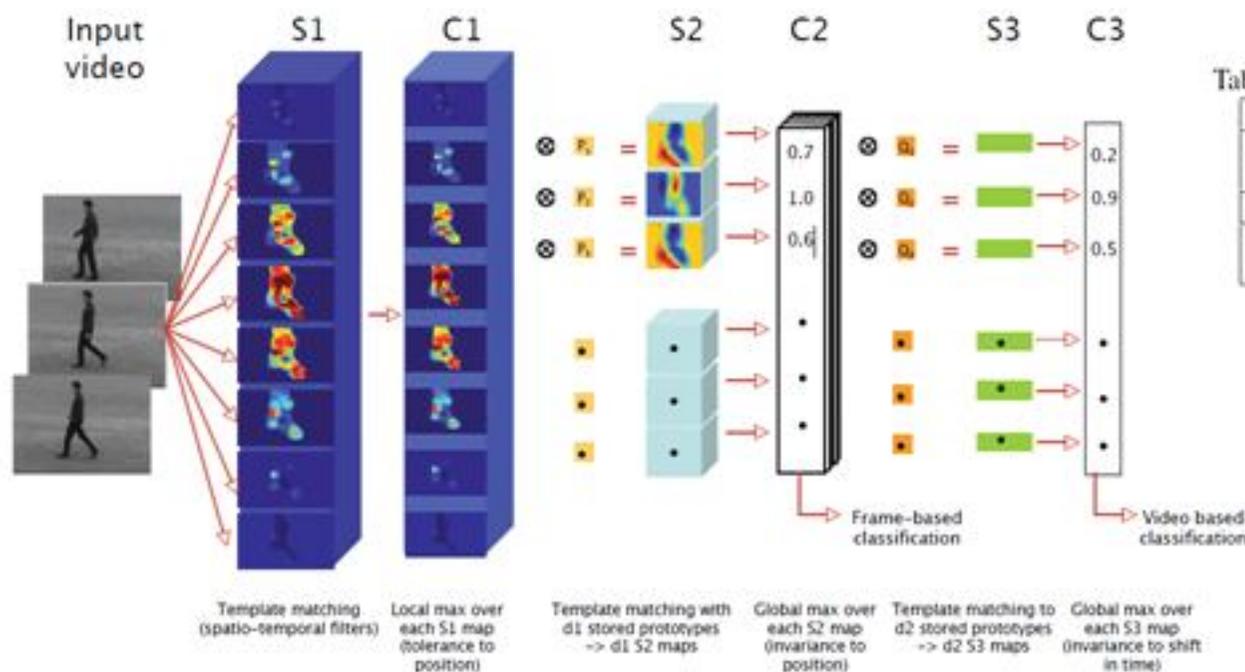


Table 6. Average performance for shape vs. motion cues.

HOG/HOF	HOGHOF	HOG	HOF
Original	20.44%	15.01%	17.95%
Stabilized	21.96%	15.47%	22.48%
C2	Motion+Shape	Shape	Motion
Original	22.83%	13.40%	21.96%
Stabilized	23.18%	13.44%	22.73%

Точность на HMDB (2011) в сравнении HOG/HOF признаками

- Замена S1 признаков изображений на признаки по пространственно-временному объёму
- Например I_t/I_x и I_t/I_y

H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. ICCV, 2007



Резюме эвристических методов

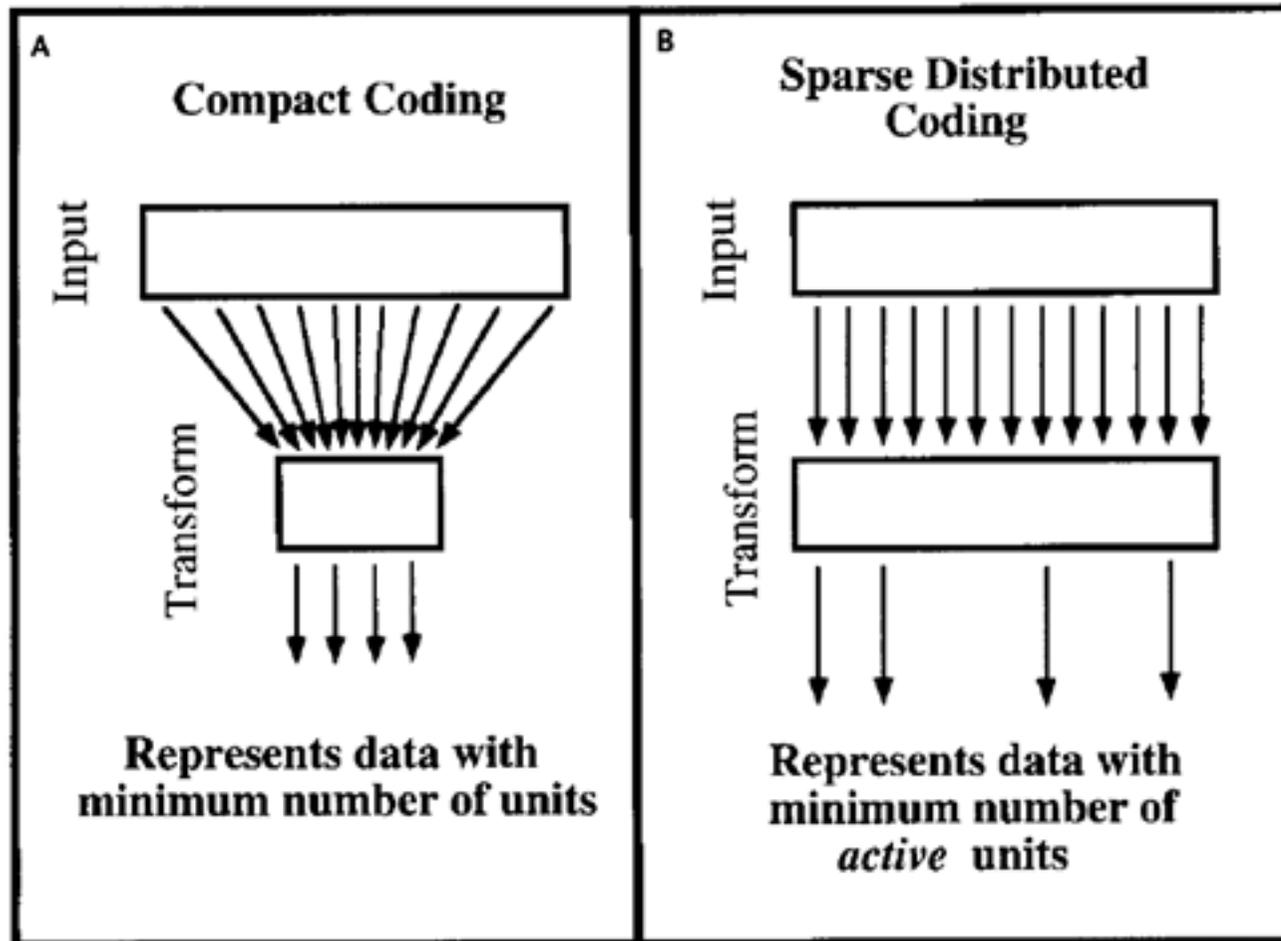
- Эвристические или биологически-обусловленные методы демонстрируют неплохие результаты
- Но про любые уровни, старше первого, мы очень мало знаем, поэтому не понимаем, как задавать веса на них

Прокрастинация или послойное
обучение



“Compact” vs “Sparse”

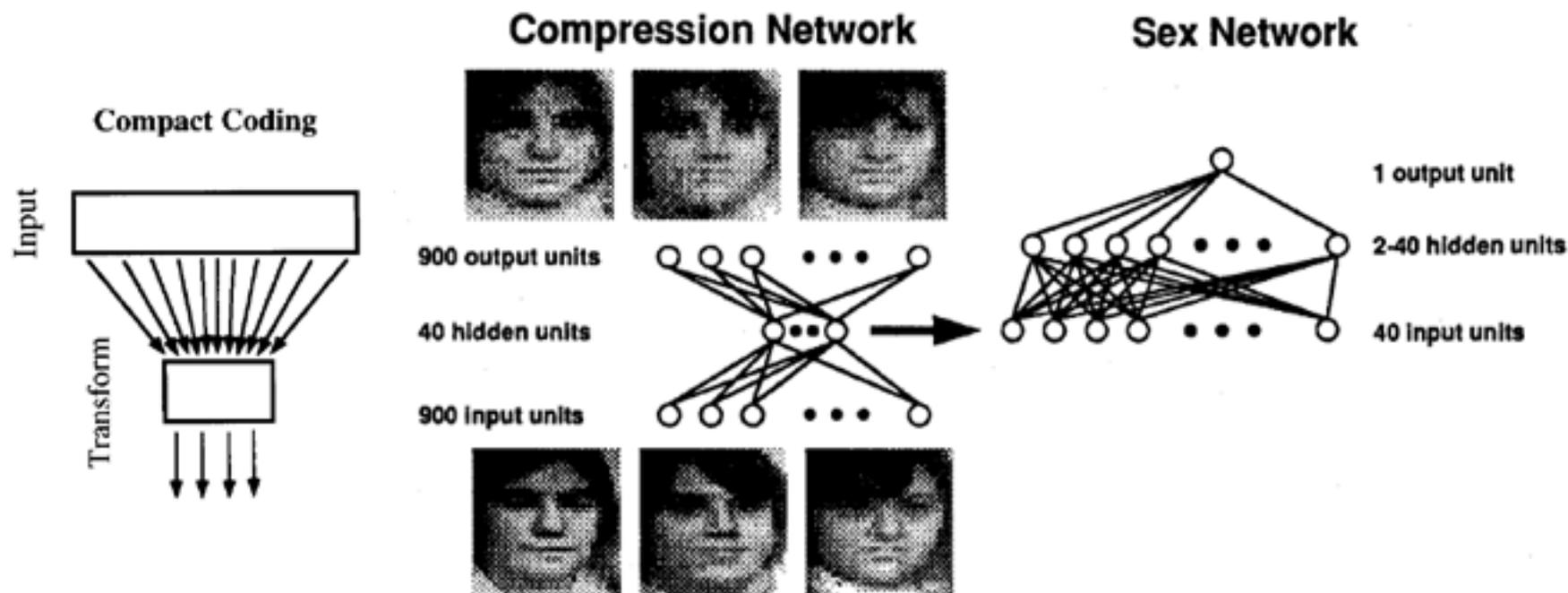
Какое поведение мы хотим получить на следующем уровне?



D. J. Field, “What is the goal of sensory coding?” *Neural Computation*, 1994.



Идея сжатия – «Autoencoder»

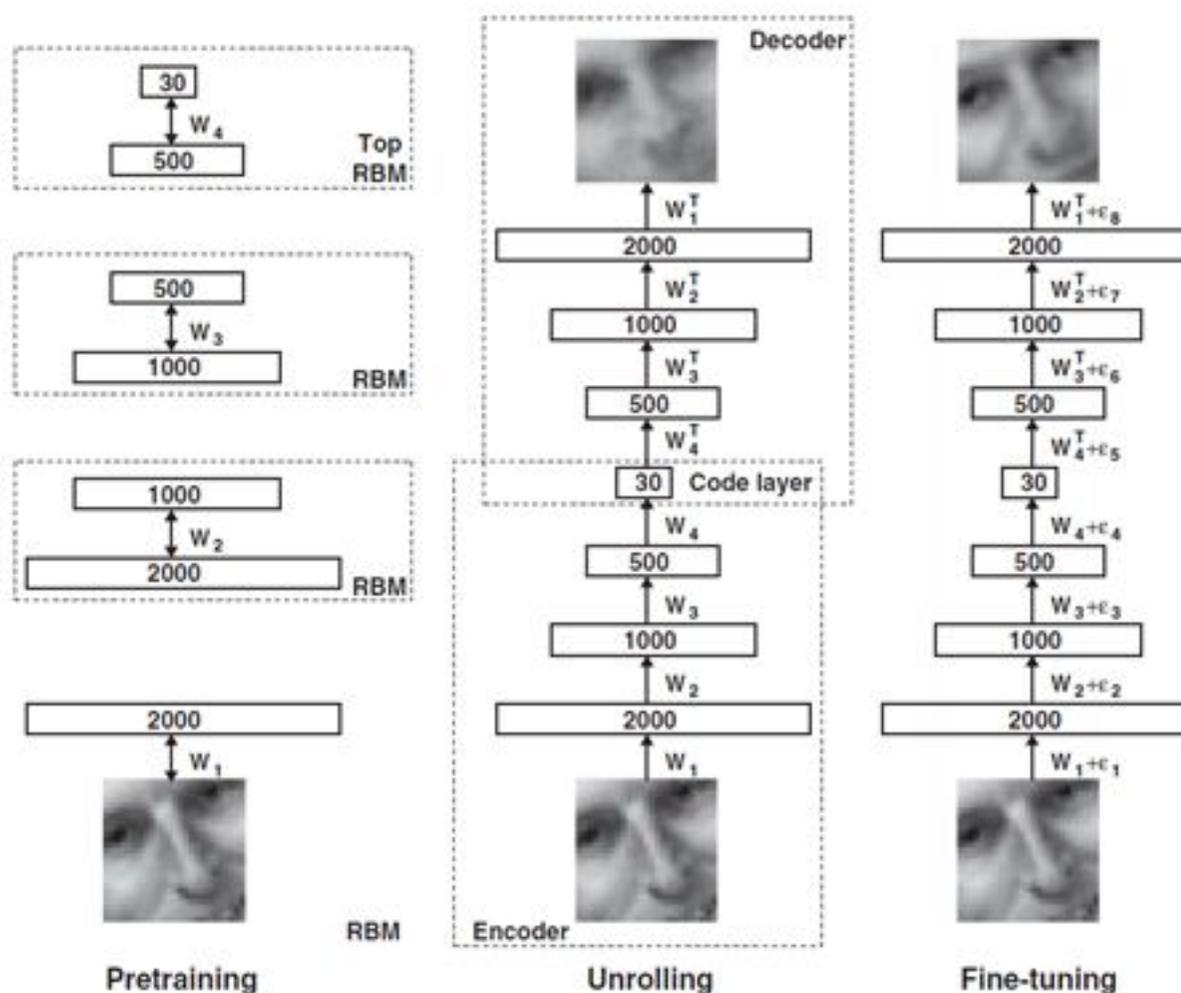


- Autoencoder – сеть, у которой равно число входов и выходов (N), и меньшее число нейронов в скрытом слое
- Смысл – описать входной сигнал коротким вектором скрытого слоя так, чтобы по нему можно было восстановить исходный сигнал
- Промежуточное представление – вектор-признак для классификации

Touretzky, Lippman. Sexnet: a neural network identifies sex from human faces. 1991



Многослойные сети



Последовательно обучаем всё более сжатые представления

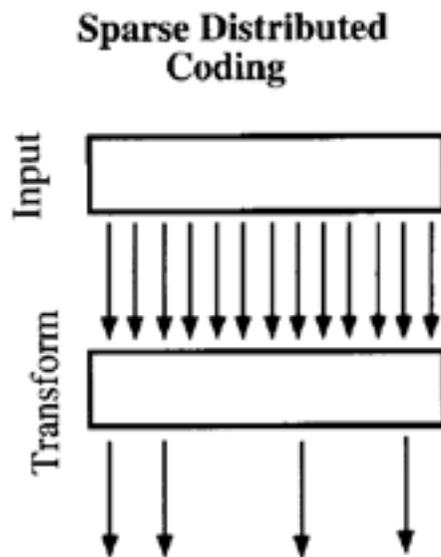
- Autoencoders
- RBM (Restricted Boltzmann Machine)

Hinton, G.E., Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006), 504–507.



Разреженный (sparse) подход

Формализуем разреженное представление:



$y \in \mathcal{R}^m$ - сигнал

$x \in \mathcal{R}^n$ - представление ($n \gg m$)

$D \in \mathcal{R}^{n \times m}$ - словарь

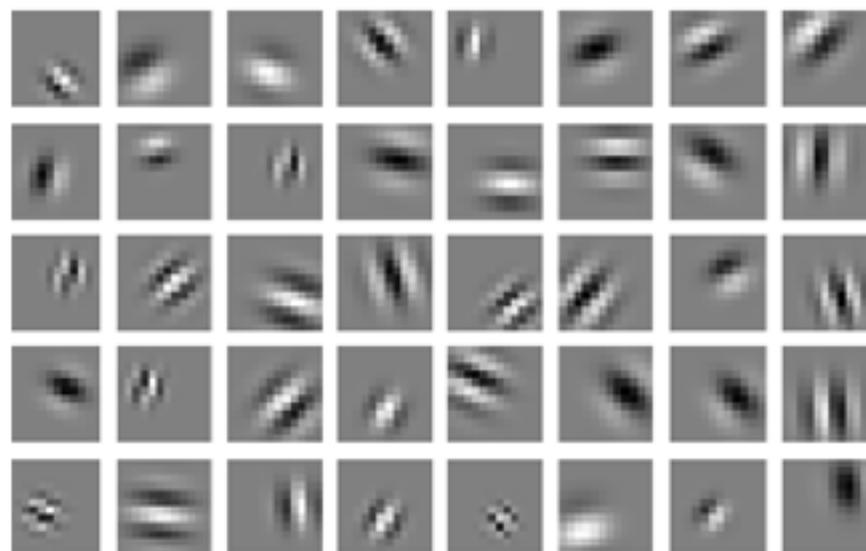
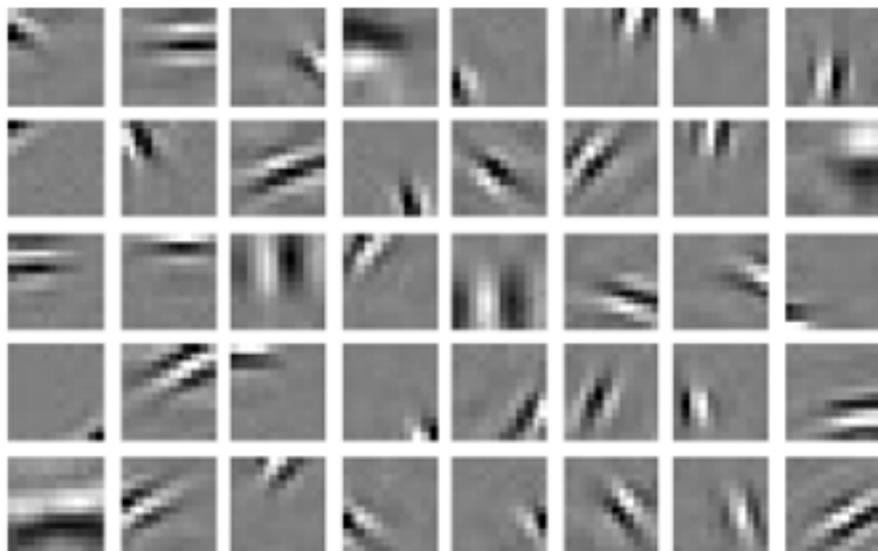
$$y = Dx \quad \|x\|_0 < L$$

- 0-норма – «псевдонорма», равна количеству ненулевых элементов вектора x
- Поскольку точного равенства достичь часто нельзя, ищем минимум нормы $\|Dx - y\|$ при условии разреженного представления x
- 0-норма очень сложна в использовании, поэтому её иногда заменяют L1-нормой



Разреженное представление изображения

Построим такое представление D по коллекции изображений:



Визуализация элементов разреженного представления D (словаря)

Фильтры Габора

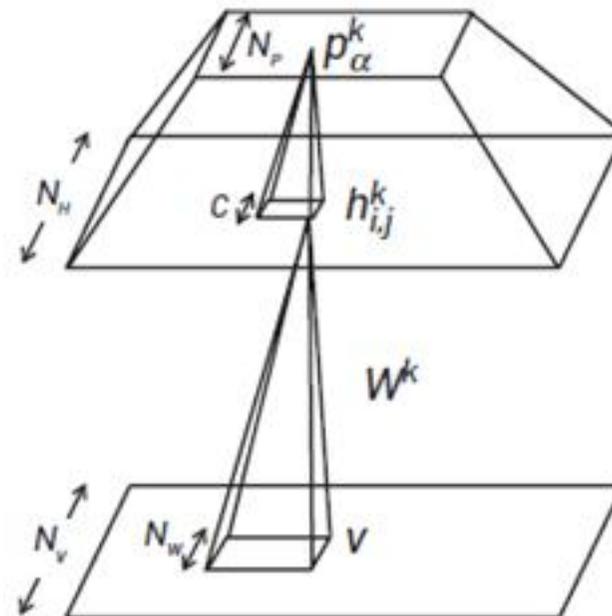
- Словарь разреженного представления оказался очень похож на банк фильтров Габора!
- Свидетельство в пользу гипотезы об использовании разреженных представлений в биологических системах распознавания

B.A. Olshausen and B.J. Field, *Emergence of simple-cell receptive field properties by learning a sparse code for natural images*, *Nature*, 381 (1996), pp. 607–609.



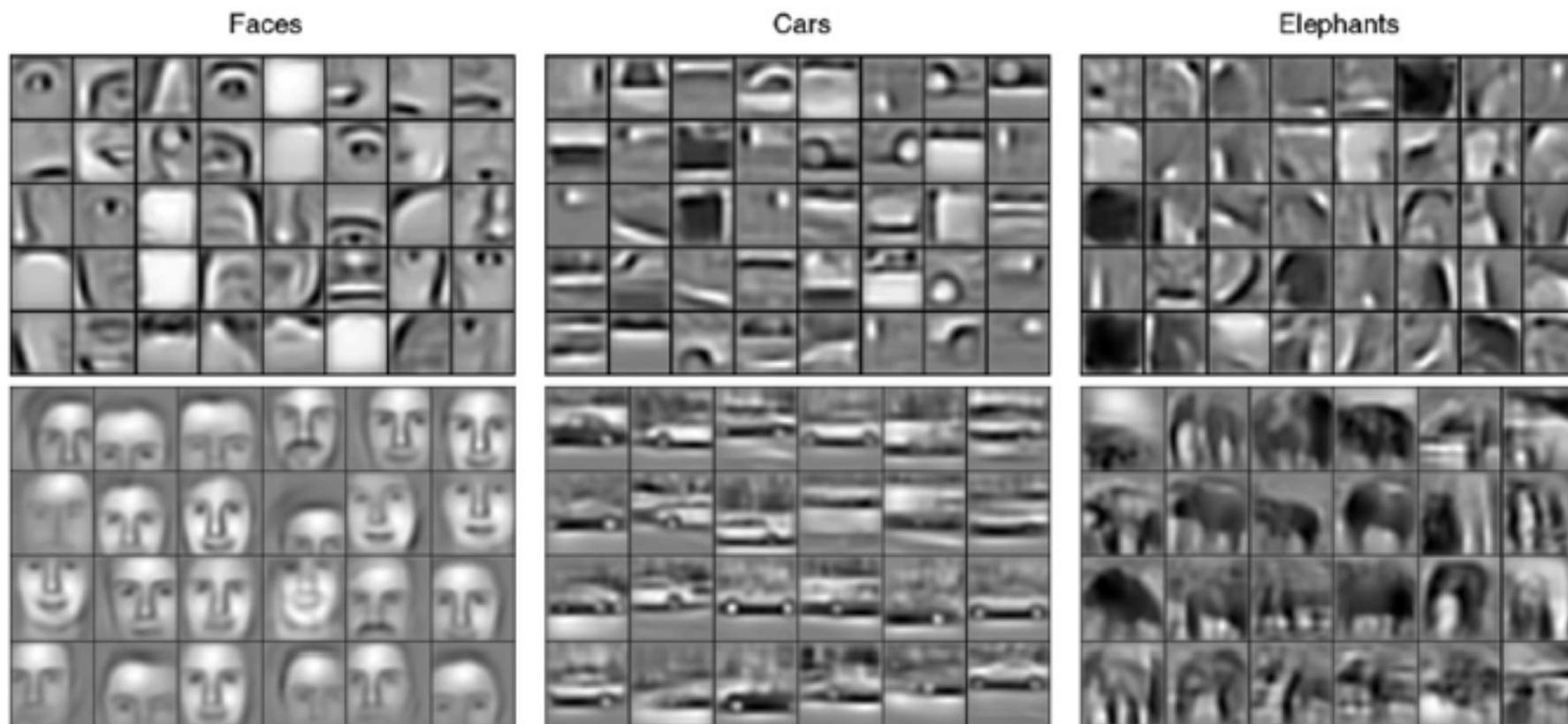
Свёрточный разреженный базис

- Есть методы, которые позволяют получить банк линейных фильтров, удовлетворяющий условиям разреженности
- Тогда можно послойно построить свёрточную сеть
- Гипотеза – нам может быть достаточно меньше изображений, чем без такого послойного обучения





Интересные примеры



Визуализация фрагментов изображений, который вызывают наибольший отклик признаков высокого уровня



Резюме прокрастинации

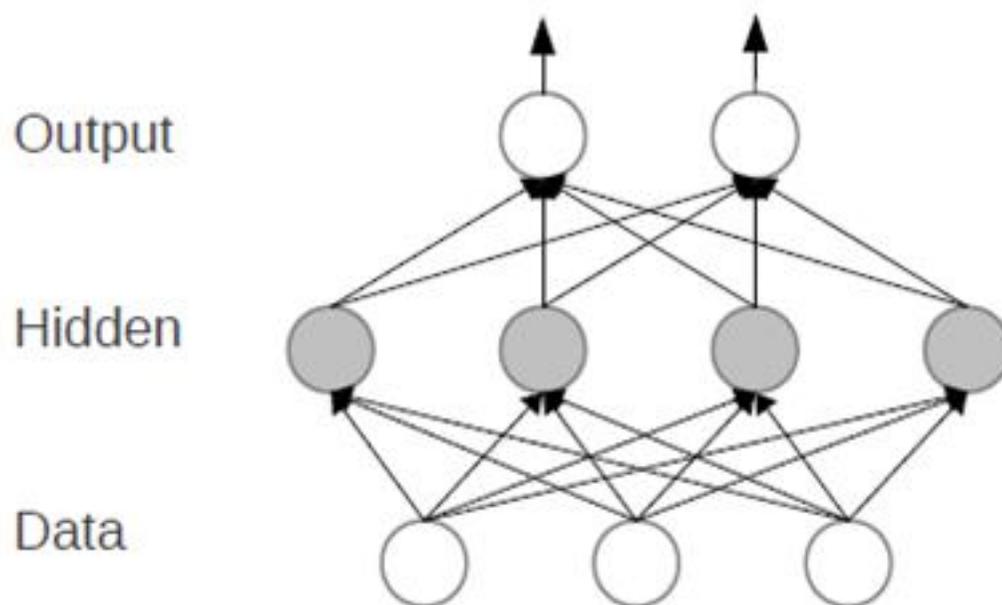
- Послойное обучение также показывает интересные результаты
- Можем использовать для получения хорошего начального приближения (претренировка)
 - В этом ключе активно исследуется

Градиентный спуск или «обратное распространение ошибки»

«Обратное распространение ошибки»



Длинная история идеи:

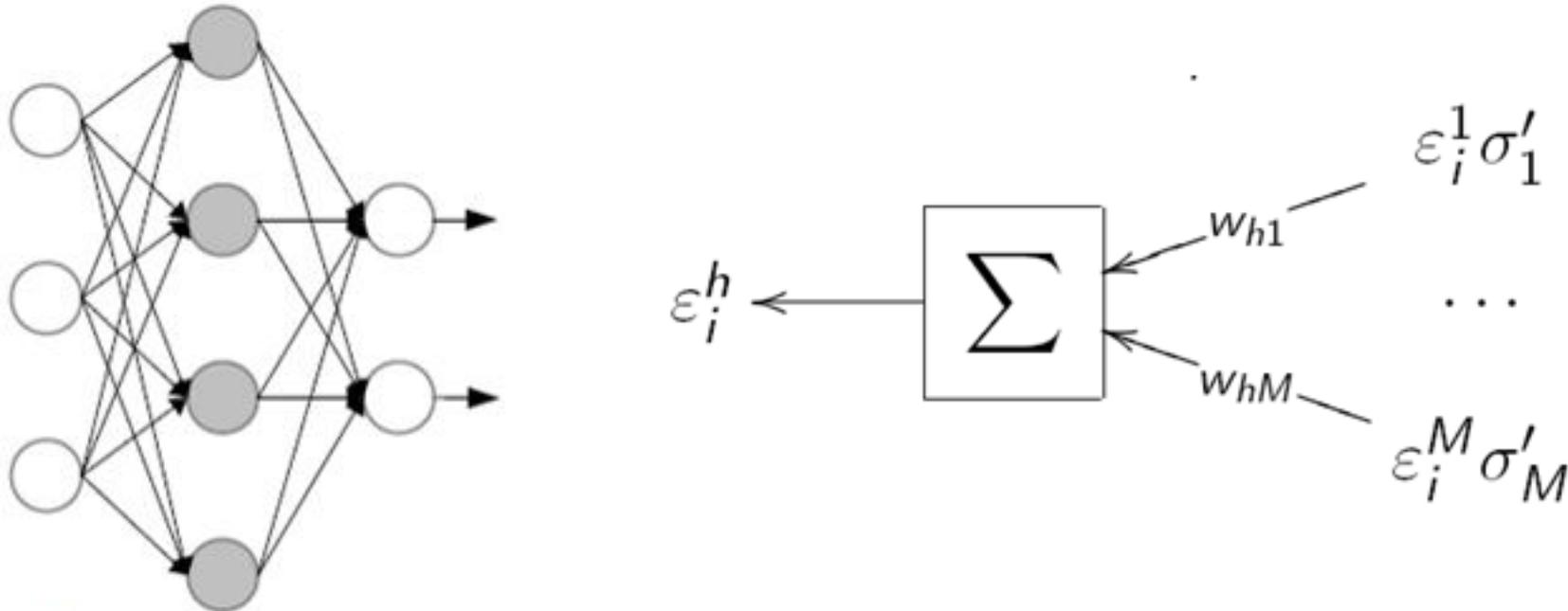


- Нейросеть вычисляет дифференцируемую функцию от своих входов
- Можем аналитически вывести, как нужно менять каждый параметр при градиентном спуске

- [↑](#) *Галушкин А. И.* Синтез многослойных систем распознавания образов. — М.: «Энергия», 1974.
- [↑](#) *Werbos P. J.*, Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
- [↑](#) [1](#) [2](#) *Rumelhart D.E., Hinton G.E., Williams R.J.*, Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing, vol. 1, pp. 318—362. Cambridge, MA, MIT Press. 1986.



Смысл градиентного спуска

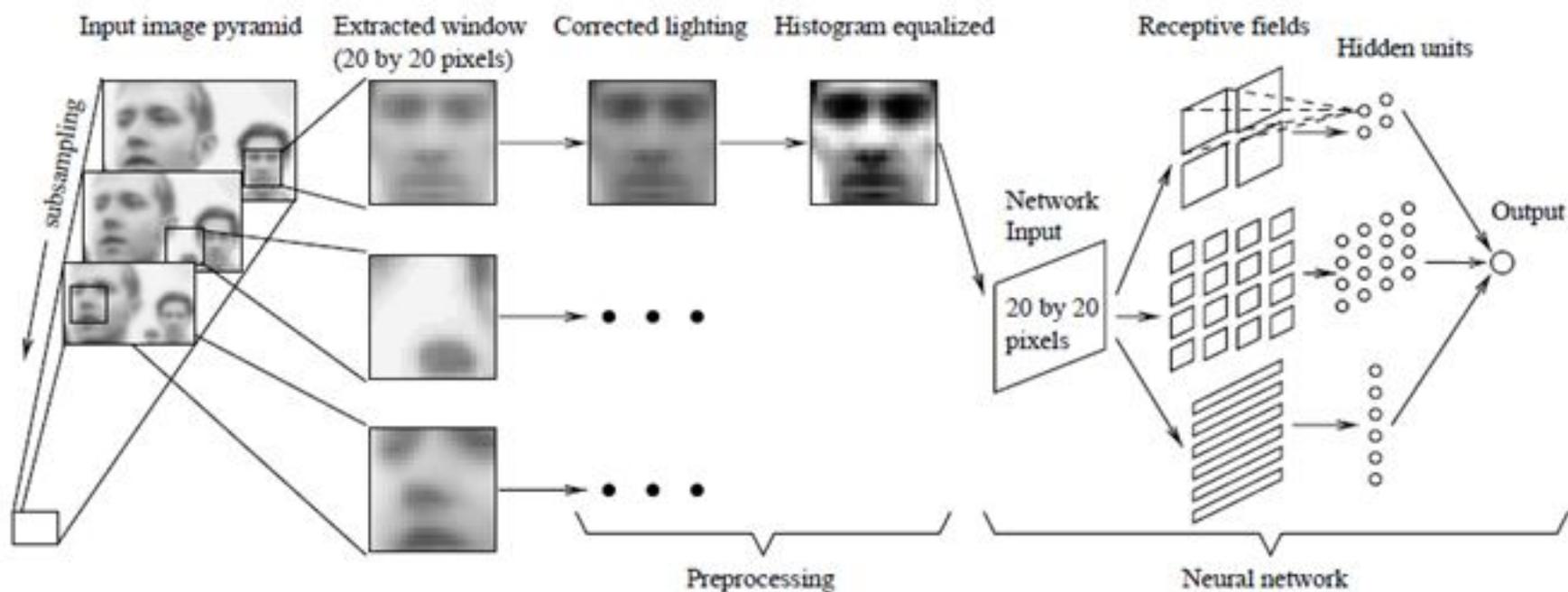


- На каждом шаге градиентного спуска нам нужно для каждого параметра вычислить градиент уменьшения ошибки
- Оказалось, что градиент для каждого параметра нейрона можем рассчитать как взвешенное произведение ошибок всех связанных с ним последующий нейронов на производную функции активации нейрона
- Для всей сети мы можем это сделать за один обратный проход, как-бы запустив сеть «задом наперёд»



Rowley face detector (1998)

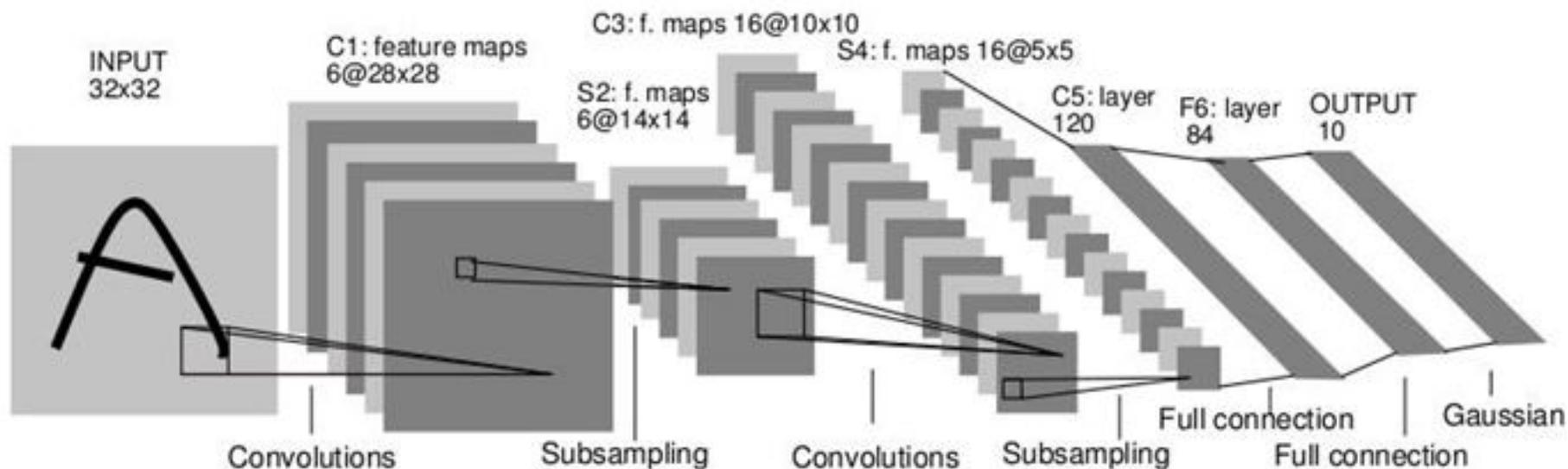
- Метод обратного распространения ошибки оказался очень эффективным
- Пример – детектор лица, лучший до Viola-Jones



B. Rowley, T. Kanade. Neural Network-Based Face Detection. PAMI, 1998.



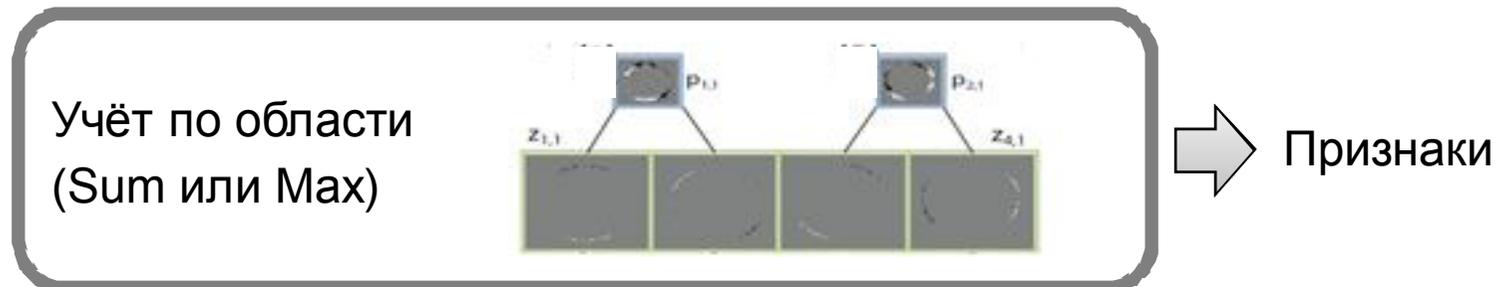
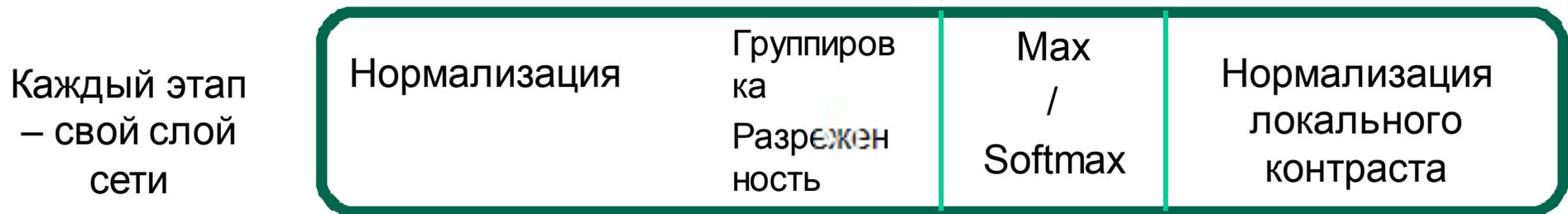
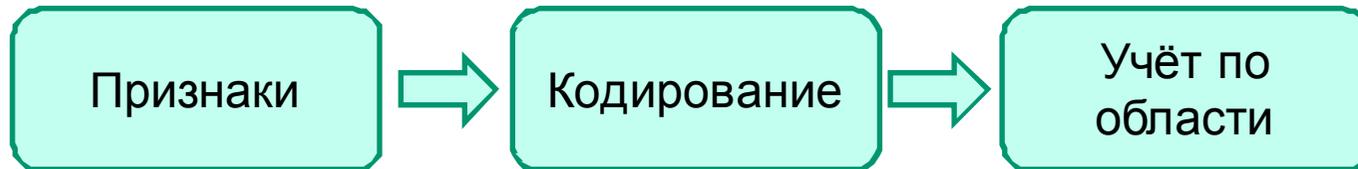
Свёрточные сети



- Неокогнитрон + обратное распространение ошибки = свёрточная сеть (Convolutional Neural Network, CNN)
- Поскольку в свёрточном слое веса всех нейронов одинаковы, то по сравнению с сетями другой архитектуры, число параметров свёрточной сети меньше
- Очень эффективная архитектура для распознавания изображений

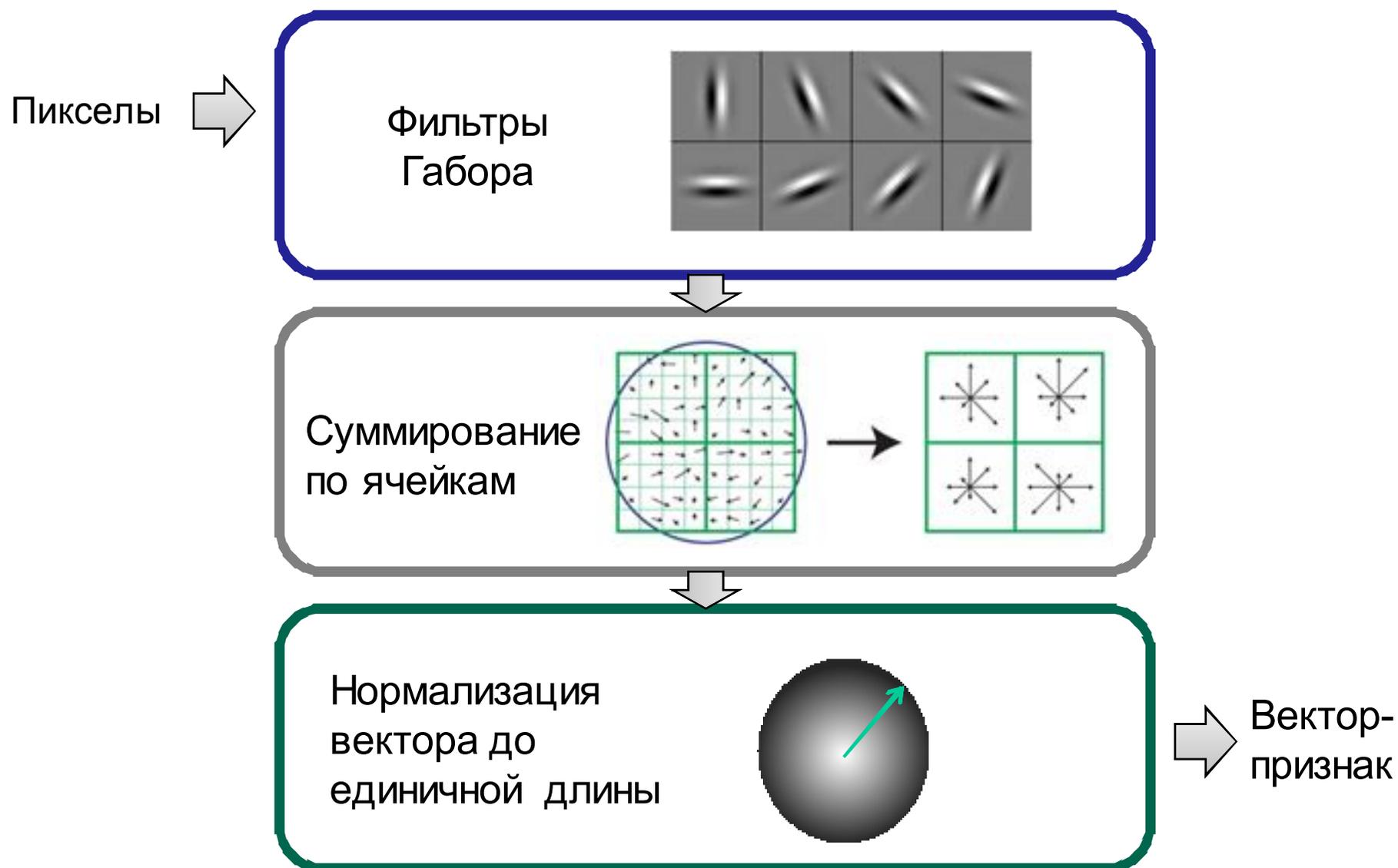
LeCun, Y., Bottou, L., Bengio, Y., and Hader, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE

Реализуем общую схему вычисления признаков



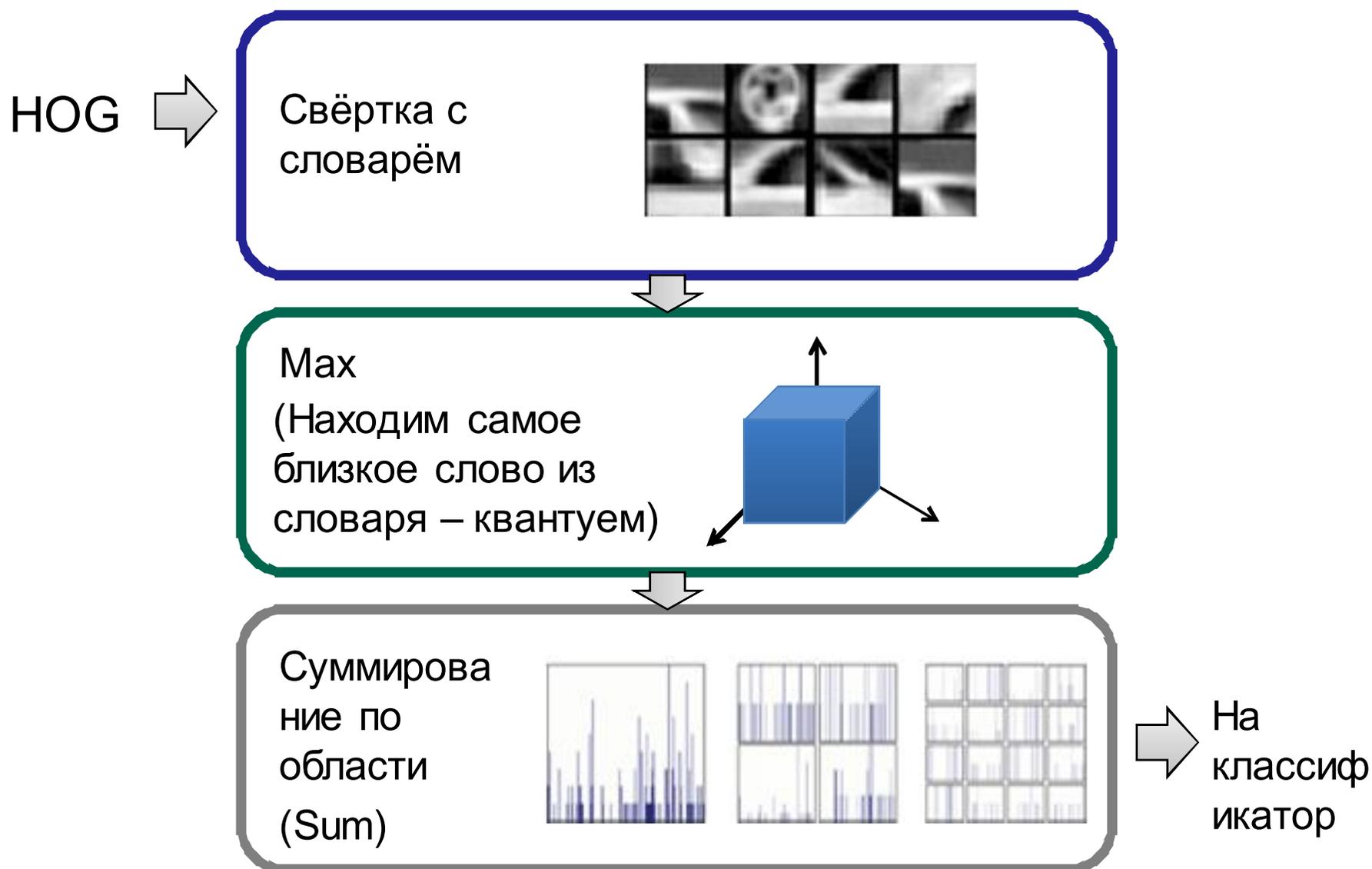


Дескриптор HOG





Мешок визуальных слов





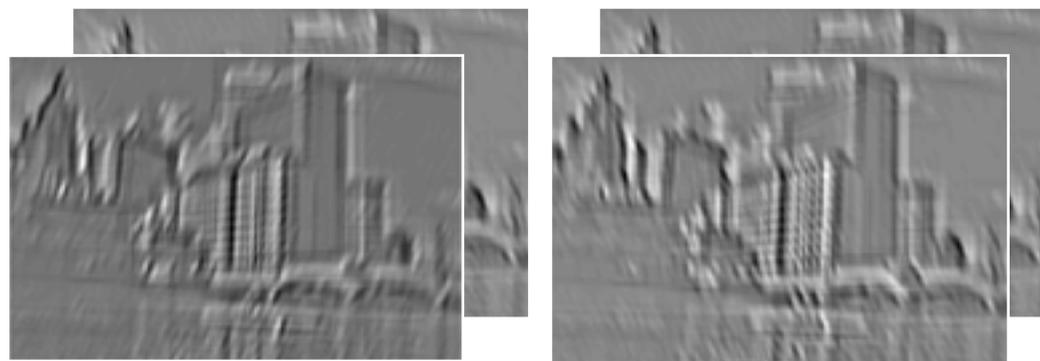
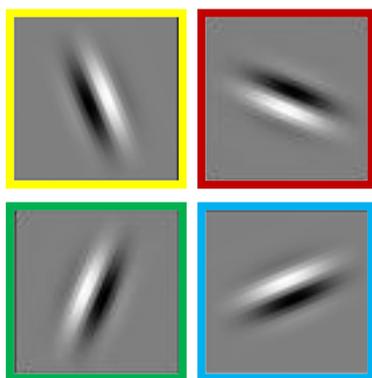
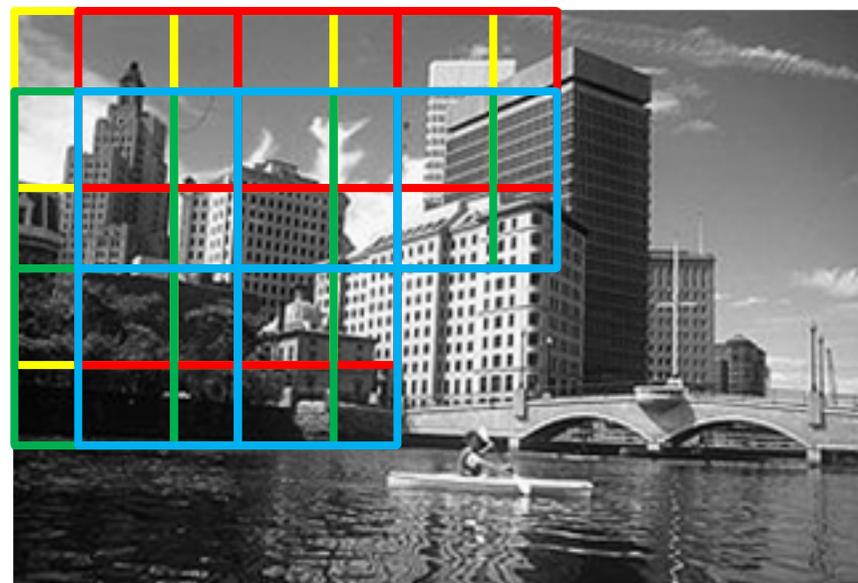
Вывод

- С помощью свёрточной нейросети можно реализовать большинство эвристических методов вычисления признаков изображения (гистограммы цветов, HOG, мешки визуальных слов)
- Последующие слои реализуют какие-то признаки «более высокого уровня»
 - Какие именно – хороший вопрос, активно исследуется
- При обучении свёрточной сети эти признаки *обучаются*, а не задаются пользователем
- Свёрточная нейросеть – схема для обучения иерархических признаков



Некоторые детали - tiling

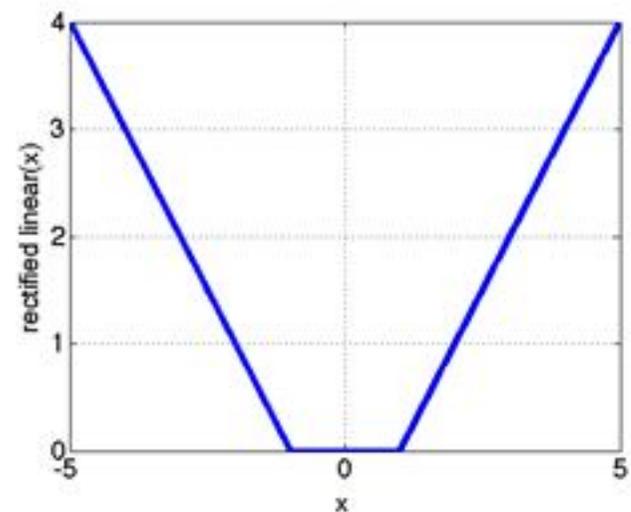
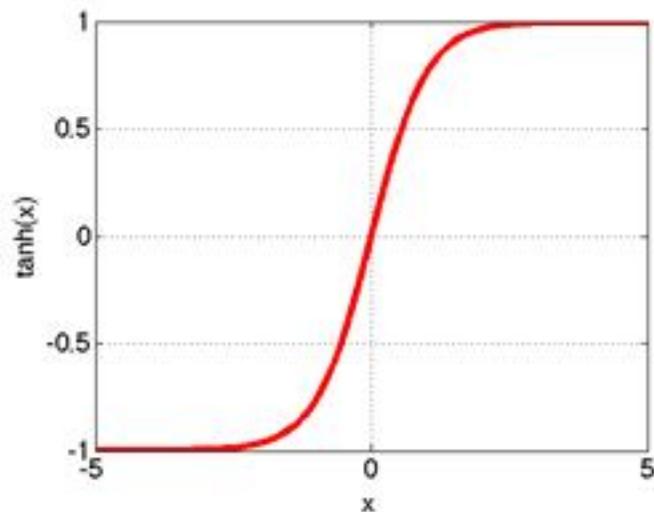
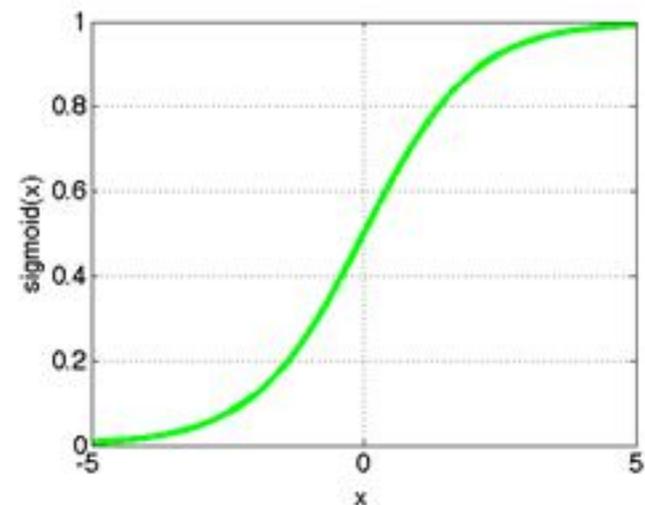
- Свёртка для каждого пикселя может быть слишком долго
- Можем повторять фильтры каждый n пикселей
- Это позволяет вычислить большее число фильтров за то же время





Функции активации (нелинейность)

- Rectified linear (ReLU)
 - $\max(0, X)$
- Tanh
- Sigmoid: $1/(1+\exp(-x))$

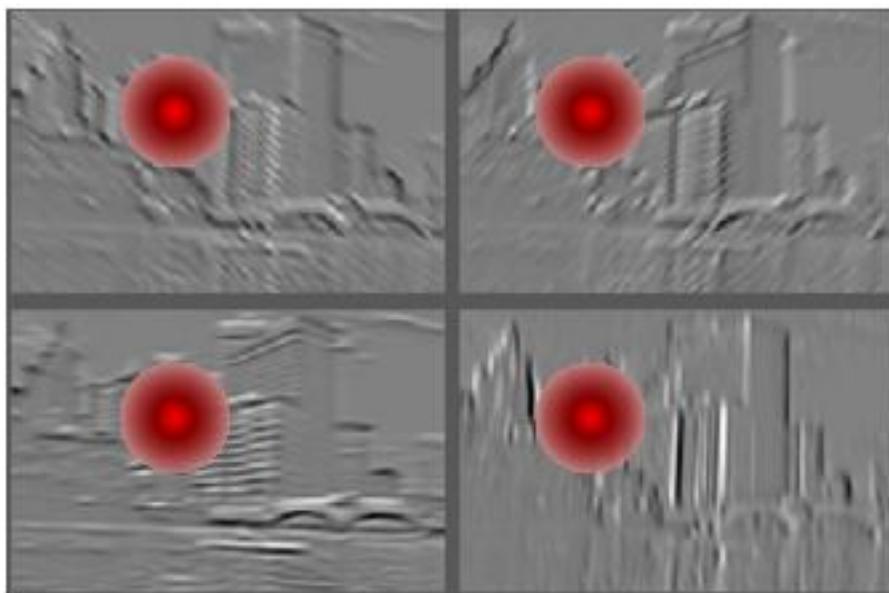




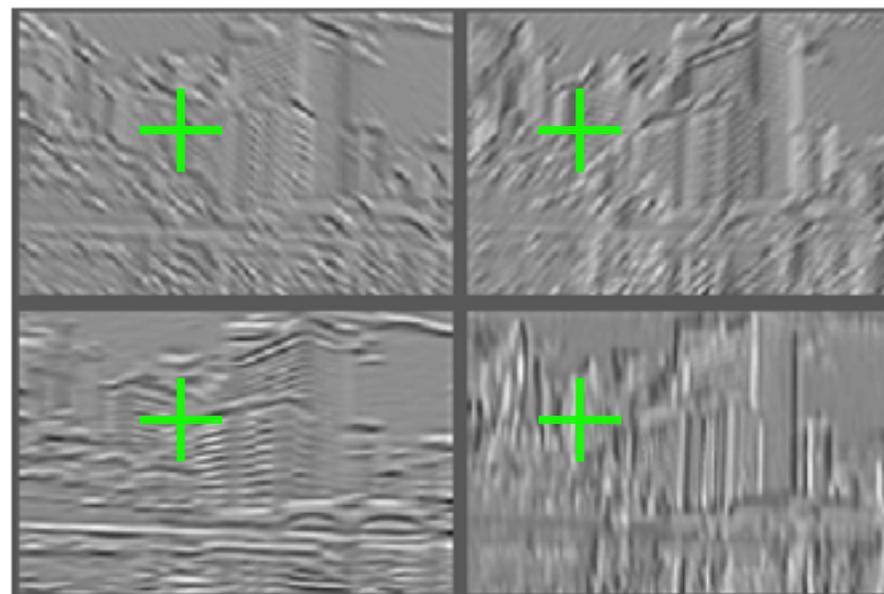
Нормализация

Локальная нормализация:

- Local mean = 0, local std. = 1, “Local” \rightarrow 7x7 Gaussian



Карты признаков

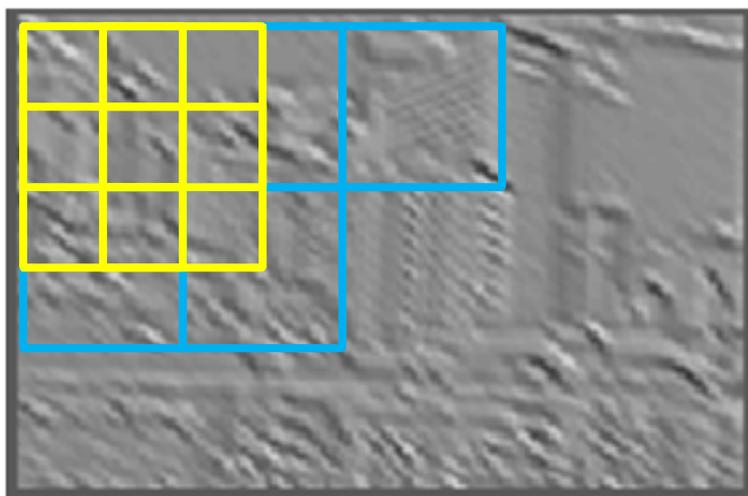


После нормализации
контраста



Учёт по областям

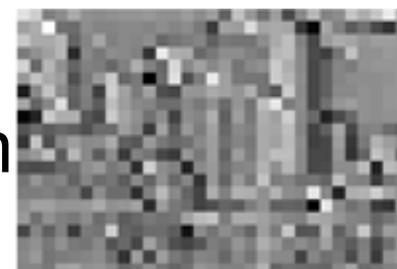
- **Sum** или **Max**
- Регионы с перекрытием или без перекрытия



Max

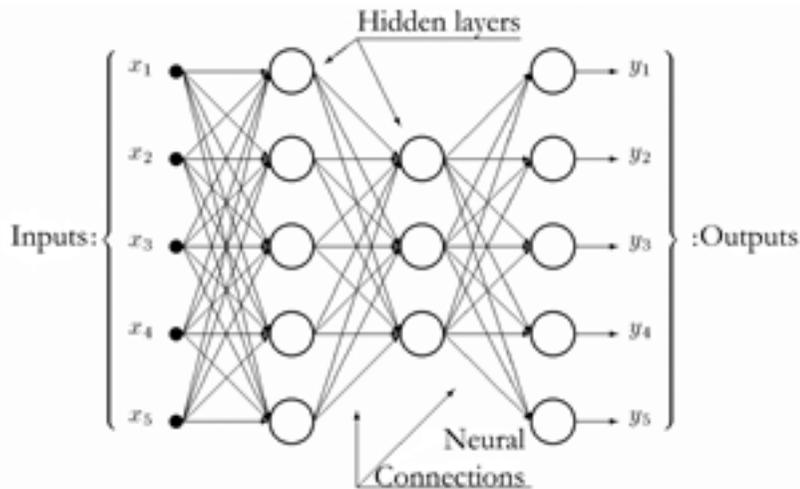


Sum





Классификатор



При обучении сети на задаче многоклассовой классификации на последнем слое столько нейронов, сколько классов

- Чаще всего последним этапом делают операцию Softmax:

$$P(y = j | \mathbf{x}) = \frac{e^{\mathbf{x}^T \mathbf{w}_j}}{\sum_{k=1}^K e^{\mathbf{x}^T \mathbf{w}_k}}$$

- «Сжимаем» произвольный вектор длины N в вектор со значениями $(0, 1)$ и суммой 1
- Это позволяет трактовать выход как вероятности для каждого класса

Этапный результат



14М изображений
Будет 1000 на
каждую категорию

Vegetable, veggie, veg

Eatible seeds or roots or stems or leaves or bulbs or tubers or nonseed fruits of any of numerous herbaceous plant

1369 pictures

73.50% Popularity Percentage



- Numbers in brackets: the number of synonyms in the subtitle 1.
- ImageNet 2011 Winter Release (17)
- animal, animate being, beast, br...
- aport, athletics (165)
- fabric, cloth, material, textile (2)
- instrumentality, instrumentation
- appliance (50)
- structure, construction (1238)
- fruit (308)
- flower (461)
- fungus (302)
- tree (992)
- vegetable, veggie, veg (175)**
- fennel, Florence fennel, finoc...
- cucumber, cuke (1)
- squash (16)
- cruciferous vegetable (18)
- pieplant, rhubarb (0)
- root vegetable (25)
- solanaceous vegetable (25)
- greens, green, leafy vegetabl...
- pot herb (0)
- legume (37)
- raw vegetable, rabbit food (0)
- artichoke, globe artichoke (0)
- artichoke heart (0)
- asparagus (0)
- plantain (0)
- truffle, earthnut (0)
- pumpkin (0)
- mushroom (0)



Large-scale visual recognition (LSVR)



Task 1: Classification



Car

- Predict a class label
- 5 predictions / image
- 1000 classes
- 1,200 images per class for training
- Bounding boxes for 50% of training.

**Task 2: Detection
(Classification + Localization)**



classification Car

- Predict a class label and a bounding box
- 5 predictions / image
- 1000 classes
- 1,200 images per class for training
- Bounding boxes for 40% of training.

Task 3: Fine-grained classification



classification Walker hound

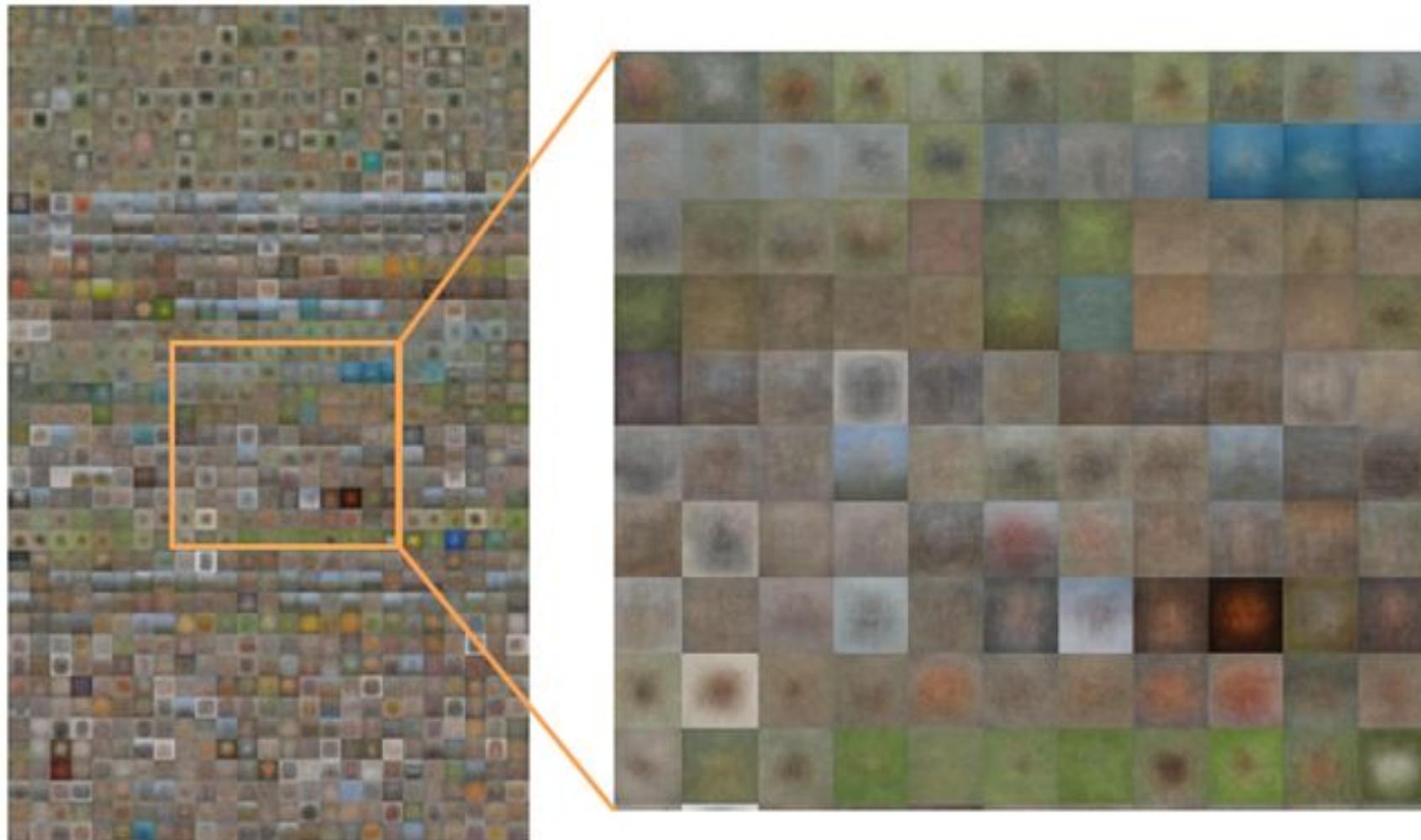
- Predict a class label given a bounding box in test
- 1 prediction / image
- 120 dog classes (subset)
- ~200 images per class for training (subset)
- Bounding boxes for 100% of training

1.2M изображений для обучения

Изображения в среднем



Average Test images



Нейросети - победители



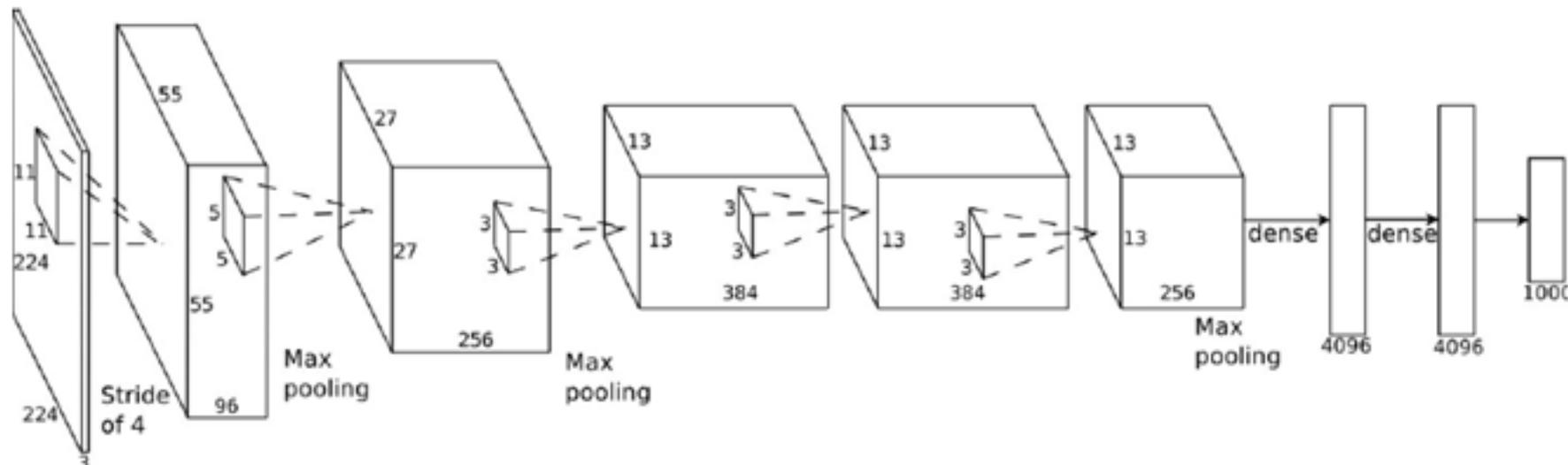
Winner

SuperVision

Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton
University of Toronto



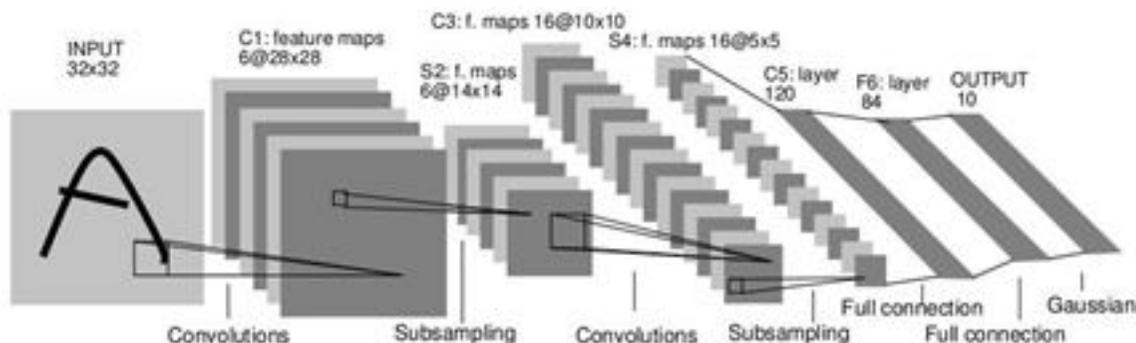
SuperVision



- 650,000 neurons
- 60,000,000 parameters
- 630,000,000 connections
- 1 машина, 2 GPU по 2Gb, 5GB Ram, 27Gb HDD, 1 неделя на обучение

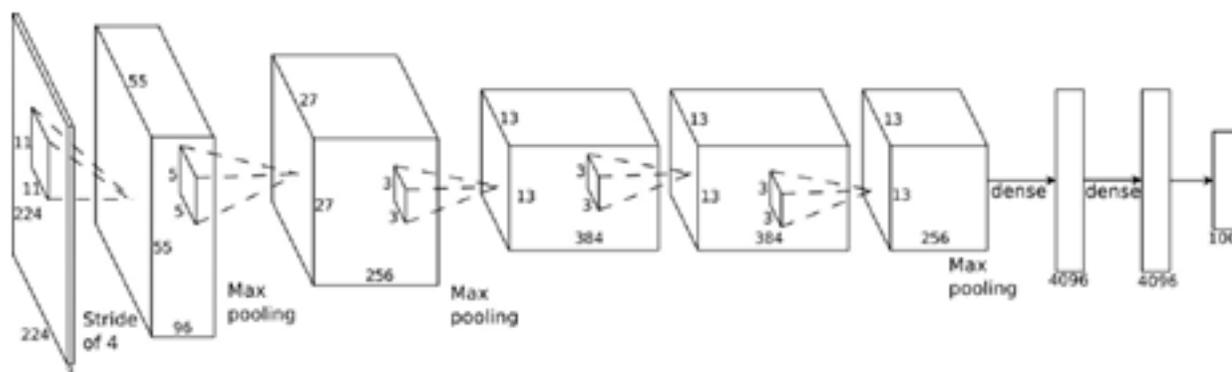


CNN раньше и сейчас



1998 год

- 2 свёрточных слоя (6 и 6 фильтров)
- 2 полносвязанных (120 и 84 нейрона)



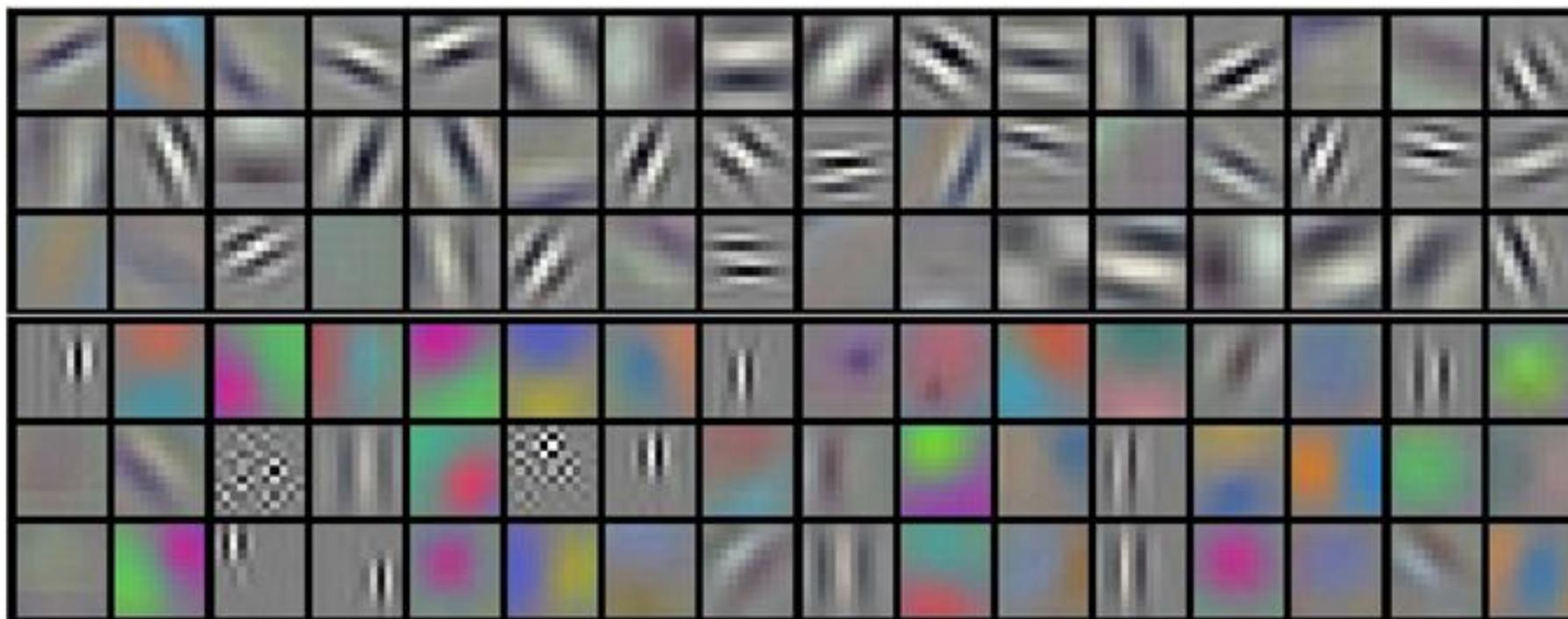
2012 год

- 5 свёрточных слоёв (96, 256, 384, 384, 256 фильтров)
- 2 полносвязанных (4096 и 4096 нейрона)

- Больше слоёв, фильтров, нейронов
- За счёт большого объёма данных и вычислительной мощности смогли обучить такую большую сеть

Krizhevsky A., Sutskever I., Hinton, G. E. (2012) ImageNet Classification with Deep Convolutional Neural Networks // NIPS 2012: Neural Information Processing Systems. Lake Tahoe, Nevada.

Фильтры первого уровня



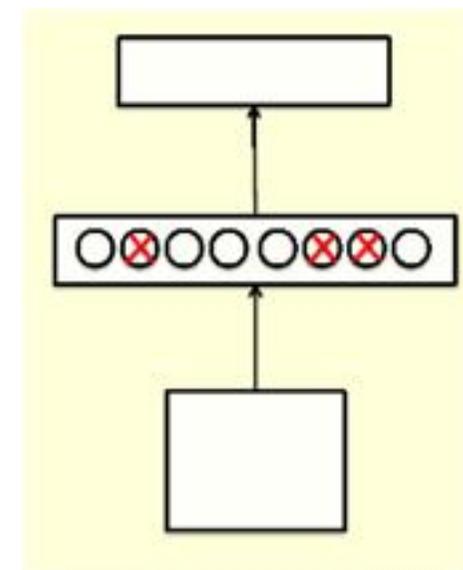
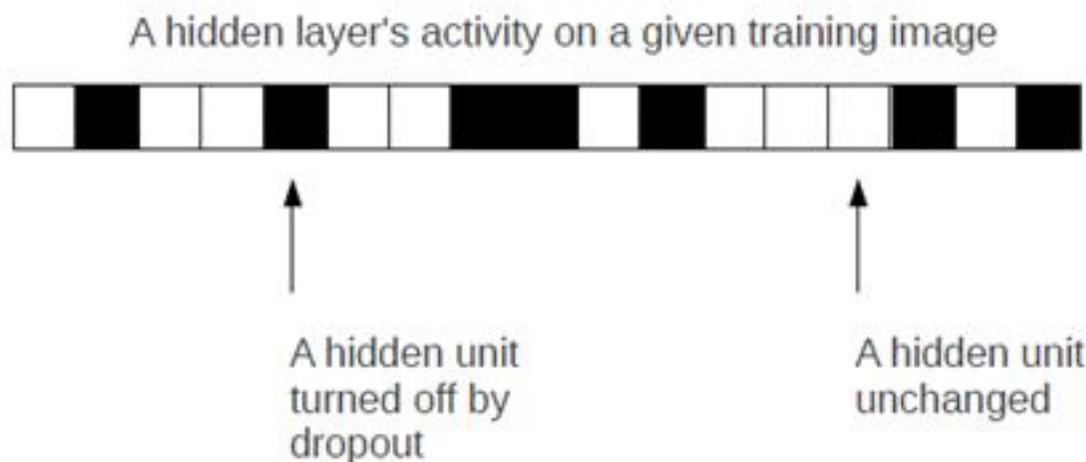
Размножение данных



- Борьба с переобучением
- Из 256×256 случайно выбираем фрагменты 224×224 и их отражения



Dropout



- Отключаем половину нейронов в каждом слое
- Получаем случайную выборку из множества сетей
- Во время тестирования используем «среднюю» сеть с уполовиненными весами

Nitish Srivastava Improving Neural Networks with Dropout.
Master Thesis, 2013

Примеры работы



mite

container ship

motor scooter

leopard

<p>mite</p> <p>black widow</p> <p>cockroach</p> <p>tick</p> <p>starfish</p>	<p>container ship</p> <p>lifeboat</p> <p>amphibian</p> <p>fireboat</p> <p>drilling platform</p>	<p>motor scooter</p> <p>go-kart</p> <p>moped</p> <p>bumper car</p> <p>golfcart</p>	<p>leopard</p> <p>jaguar</p> <p>cheetah</p> <p>snow leopard</p> <p>Egyptian cat</p>
---	---	--	---



grille

mushroom

cherry

Madagascar cat

<p>convertible</p> <p>grille</p> <p>pickup</p> <p>beach wagon</p> <p>fire engine</p>	<p>agaric</p> <p>mushroom</p> <p>jelly fungus</p> <p>gill fungus</p> <p>dead-man's-fingers</p>	<p>dalmatian</p> <p>grape</p> <p>elderberry</p> <p>ffordshire bullterrier</p> <p>currant</p>	<p>squirrel monkey</p> <p>spider monkey</p> <p>titi</p> <p>indri</p> <p>howler monkey</p>
--	--	--	---

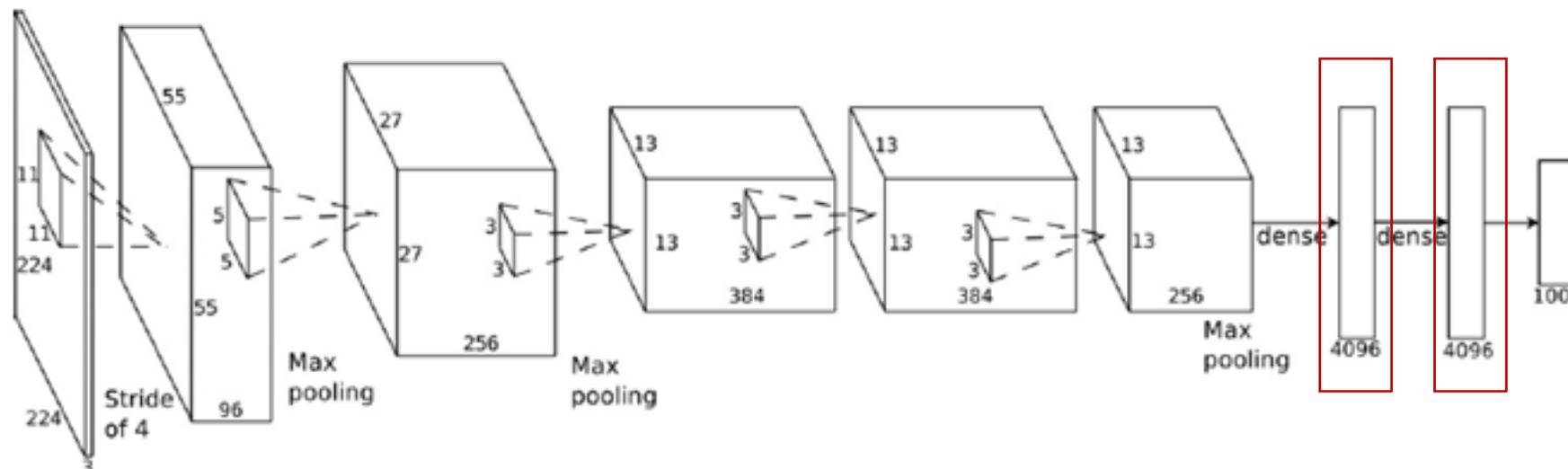
Примеры работы



			
koala	tiger	European fire salamander	loggerhead
<ul style="list-style-type: none"> wombat Norwegian elkhound wild boar wallaby koala 	<ul style="list-style-type: none"> tiger tiger cat jaguar lynx leopard 	<ul style="list-style-type: none"> European fire salamander spotted salamander common newt long-horned beetle box turtle 	<ul style="list-style-type: none"> African crocodile Gila monster loggerhead mud turtle leatherback turtle
			
seat belt	television	sliding door	wallaby
<ul style="list-style-type: none"> seat belt ice lolly hotdog burrito Band Aid 	<ul style="list-style-type: none"> television microwave monitor screen car mirror 	<ul style="list-style-type: none"> sliding door shoji window shade window screen four-poster 	<ul style="list-style-type: none"> hare wallaby wood rabbit Lakeland terrier kit fox



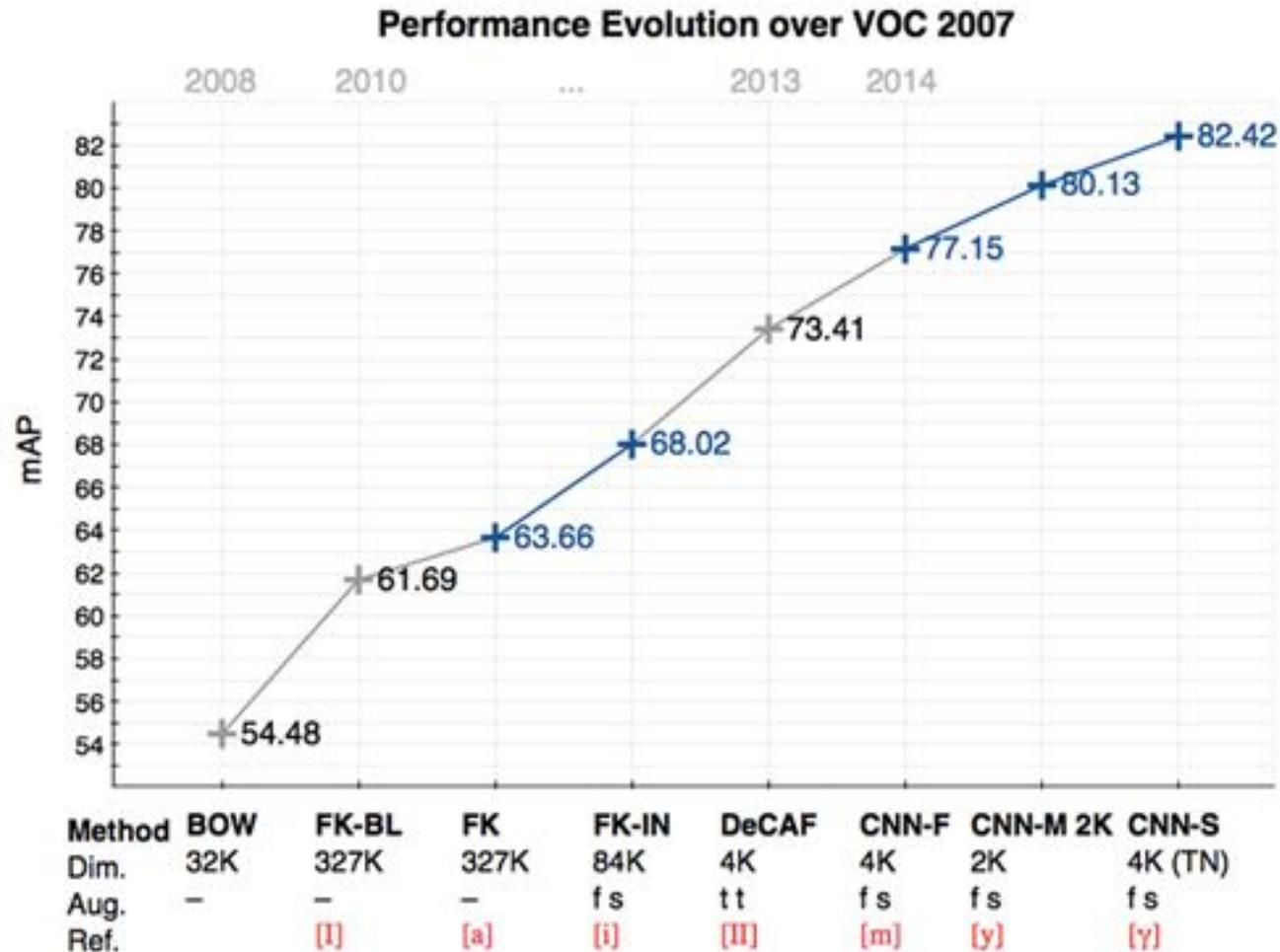
Высокоуровневые признаки



- Эксперименты показали, что выходы полносвязанных слоёв можно использовать как признаки изображения
- Поверх можно обучать разные классификаторы
- Можно обучить сеть на одних данных (ImageNet) и применять на других для вычисления признаков

Donahue et. al. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, 2013

Рост качества по методам



K. Chatfield et. Al. Return of the Devil in the Details: Delving Deep into Convolutional Nets, BMVC 2014



Примеры конфигураций

Arch.	conv1	conv2	conv3	conv4	conv5	full6	full7	full8
CNN-F	64x11x11 st. 4, pad 0 LRN, x2 pool	256x5x5 st. 1, pad 2 LRN, x2 pool	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 x2 pool	4096 drop- out	4096 drop- out	1000 soft- max
CNN-M	96x7x7 st. 2, pad 0 LRN, x2 pool	256x5x5 st. 2, pad 1 LRN, x2 pool	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 x2 pool	4096 drop- out	4096 drop- out	1000 soft- max
CNN-S	96x7x7 st. 2, pad 0 LRN, x3 pool	256x5x5 st. 1, pad 1 x2 pool	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 x3 pool	4096 drop- out	4096 drop- out	1000 soft- max

K. Chatfield et. Al. Return of the Devil in the Details: Delving Deep into Convolutional Nets, BMVC 2014



Резюме

- Наступил очередной ренессанс нейросетей
- Концептуально сети остались такими же, изменения пока в основном небольшие
- Объём данных и рост вычислительных мощностей позволили настраивать сложные сети