НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ МИФИ

ЛЕКЦИЯ 09. НЕЙРОСЕТЕВЫЕ ПОДХОДЫ К ОБРАБОТКЕ ИЗОБРАЖЕНИЙ

Демидов Д.В.

Обработка аудиовизуальной информации.
Бакалавры, 6 семестр. Магистры, 9 семестр

План лекции

- Искусственные нейронные сети
- Перцептрон Розенблатта
  - Однослойный перцептрон
  - Многослойный перцептрон (multi-layer perceptron)
  - Метод обратного распространения ошибки (backpropagation)
- Глубинные сети (Deep Learning)
  - Свёрточные нейронные сети (Convolutional neural networks)
  - Развёртывающиеся сети (Deconvolutional networks)

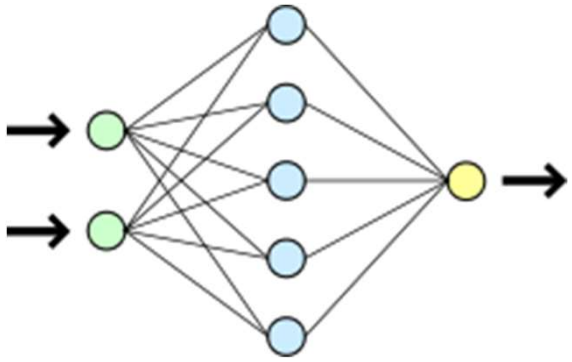
3

Искусственные нейронные сети

Модель нейрона
Перцептрон
Функции активации
Обучение как оптимизация

Искусственная нейронная сеть

- **ИНС** = Математическая модель + программно-аппаратное воплощение
- **Нейрон** – вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает дальше. Нейроны делятся на три основных типа: входной, промежуточный (скрытый) и выходной.
  - **Дендрит** – вход нейрона
  - **Аксон** – единственный выход нейрона
- **Синапс** – место контакта нейронов, характеризующееся весом связи.
- **Слой** – совокупность нейронов одного типа или выполняющих одну общую задачу

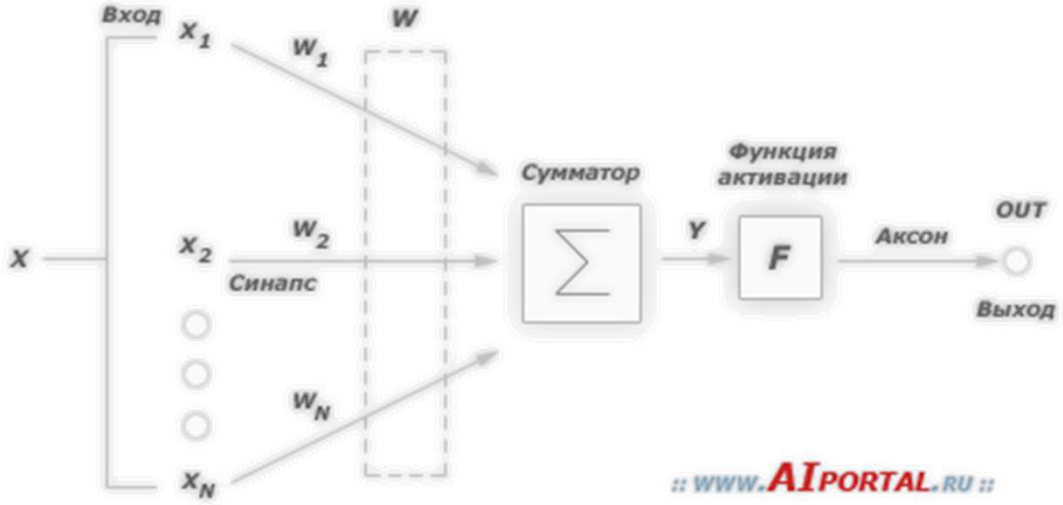


Классификация по представлению графа

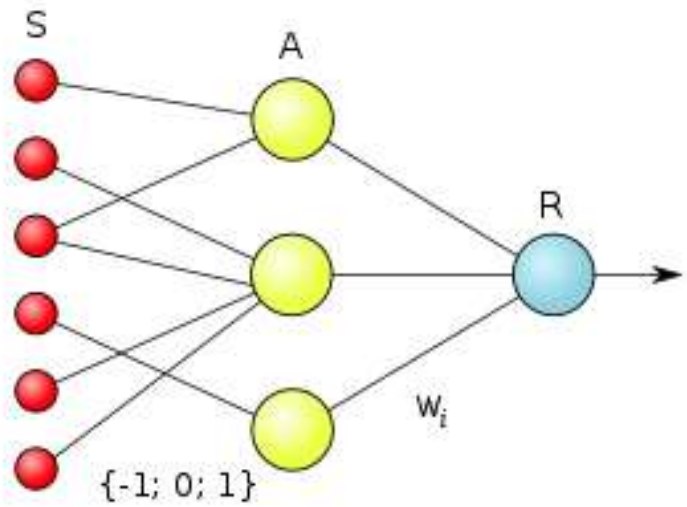
- Сети прямого распространения – feed-forward – сети в виде ациклических графов, в которых сигнал проходит прямо от входов к выходу.
- Рекуррентные сети – recurrent neural networks, RNN – сети, содержащие циклы

Модель нейрона

- **Нейрон** – узел сети, реализующий нелинейную (как правило) функцию от линейной комбинации входов.
- **Взвешенный сумматор** – вычислитель линейной комбинации всех входных сигналов $u(x) = \sum_{i=1}^{n} w_i x_i + w_0 x_0$. Иногда используется смещение (сдвиг) функции активации $w_0 x_0$.
- **Функция активации** (*активационная функция, функция возбуждения, передаточная функция, функция срабатывания*) – функция, вычисляющая выходной сигнал искусственного нейрона. В качестве аргумента принимает сигнал, получаемый на выходе входного сумматора $y = f(u)$.



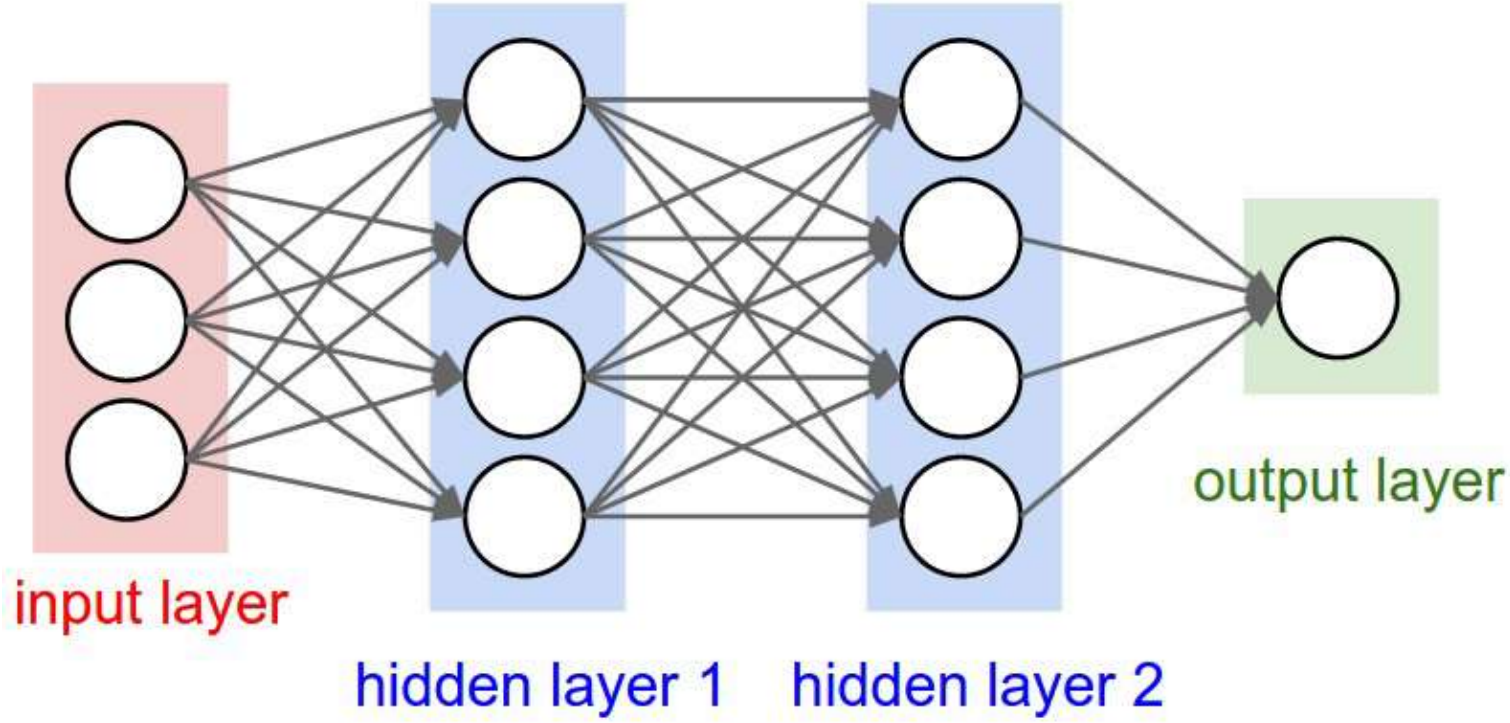
Перцептрон Розенблатта



- Логическая схема элементарного перцептрона. Веса S-A связей могут иметь значения −1, +1 или 0 (то есть отсутствие связи). Веса A-R связей W могут быть любыми.

Многослойный персептрон

- Полносвязная сеть прямого распространения (feed-forward). Входной сигнал распространяется в прямом направлении, от слоя к слою.
- Обобщение однослойного персептрона.



Свойства многослойного персептрона

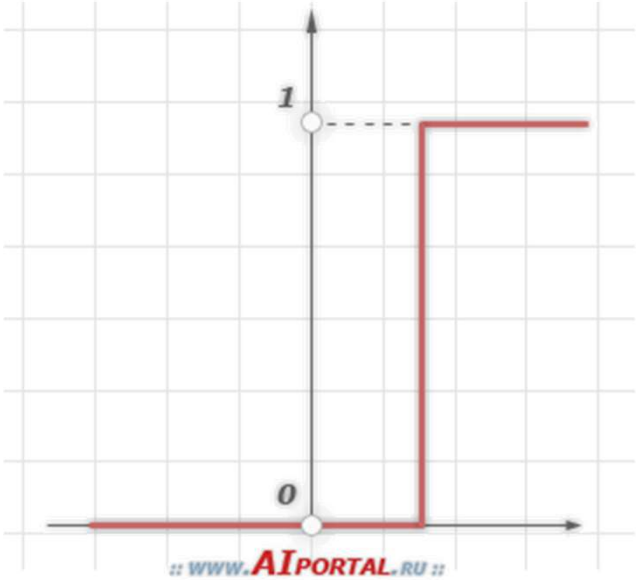
- Каждый нейрон сети имеет нелинейную функцию активации. Например, логистическая функция.
- Несколько скрытых слоев
- Высокая связность

Функции активации

Пороговая функция
Линейная функция
Сигмоидальная функция
Радиально-базисная функция

Пороговая функция

- **Единичный скачок (жесткая пороговая функция,** threshold activation function, Heaviside step function) – если входное значение меньше порогового, то значение функции активации равно минимальному допустимому, иначе – максимально допустимому
- Не дифференцируема. Не пригодна в алгоритмах обучения, требующих дифференцируемости.



$$f(x) = \begin{cases} 1 & if\ x \geq T = -w_0 x_0 \\ 0 & otherwise \end{cases}$$

Линейная функция

- **Линейная функция** – линейно связывает взвешенную сумму входных сигналов с выходом.
  - Простая линейная функция

$$f(x) = tx$$

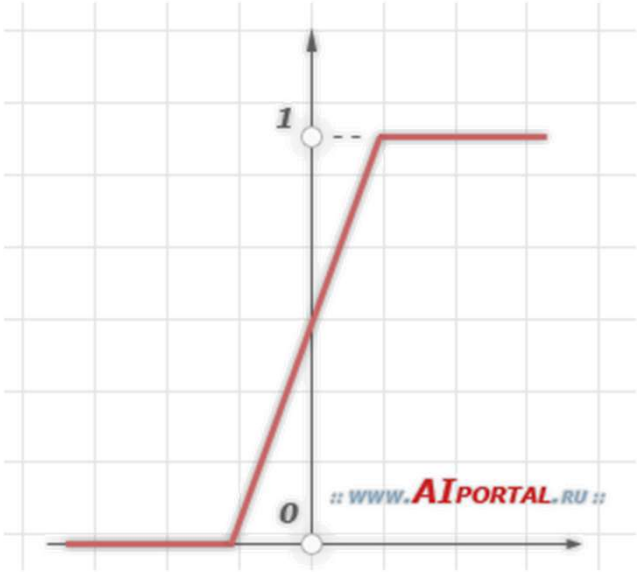
  - Полулинейная функция

$$f(x) = \begin{cases} 0 & if x \leq 0 \\ tx & otherwise \end{cases}$$

  - Линейная функция с насыщением (линейный порог, гистерезис)

$$f(x) = \begin{cases} 0 & if\ x \leq a \\ 1 & if\ x \geq b \\ x & otherwise \end{cases}$$

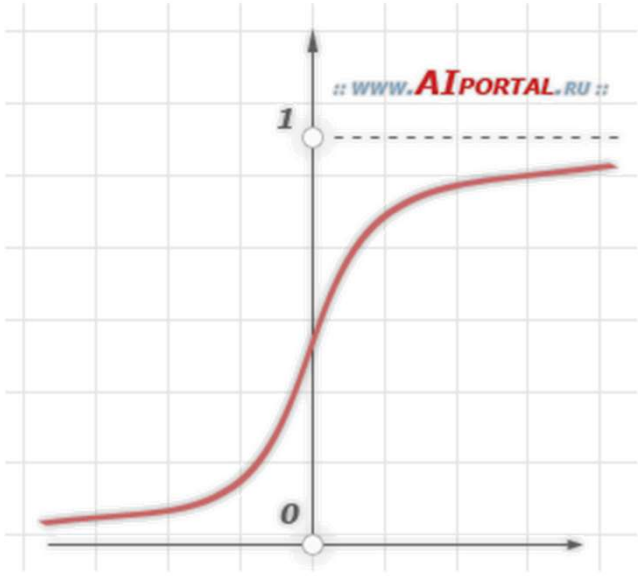
Rectified linear unit: max(x,0)



Сигмоидальная функция

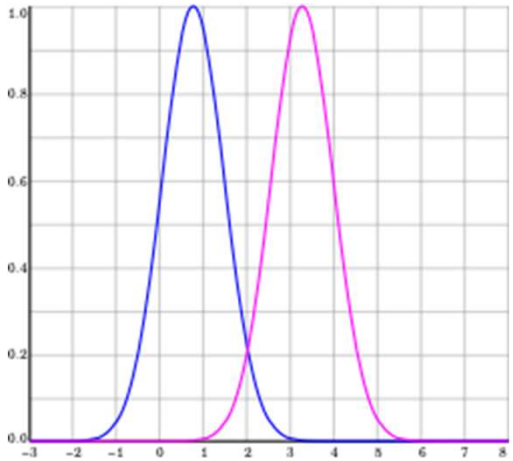
- **Сигмоид** – монотонно возрастающая всюду дифференцируемая S-образная нелинейная функция с насыщением. Сигмоид позволяет усиливать слабые сигналы и не насыщаться от сильных сигналов.
  - **Логистическая функция** – сигмоид вида $f(y) = \frac{1}{1+e^{-\alpha y}}$ с производной вида $\frac{\partial f(y)}{\partial y} = \alpha f(y)\big(1 - f(y)\big)$. Интервал значений [0;1].
  - **Гиперболический тангенс** – сигмоид вида $f(y) = th\left(\frac{y}{\alpha}\right) = \frac{e^{\frac{y}{\alpha}} - e^{-\frac{y}{\alpha}}}{e^{\frac{y}{\alpha}} + e^{-\frac{y}{\alpha}}}$.

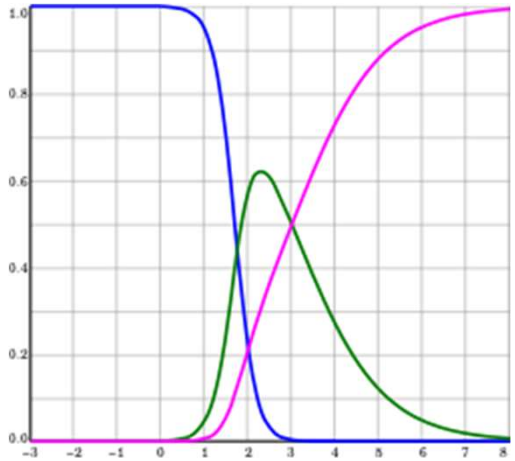
    Интервал значений [-1;1]



Радиально-базисная функция

- Радиально-базисная функция передачи принимает в качестве аргумента расстояние между входным вектором и некоторым наперед заданным центром активационной функции.
- Значение этой функции тем выше, чем ближе входной вектор к центру.
  - функция Гаусса
  - https://en.wikipedia.org/wiki/Radial_basis_function_network





Обучение

Функция потерь (loss function)

Оптимизация

Функция оценки работы нейросетевого классификатора

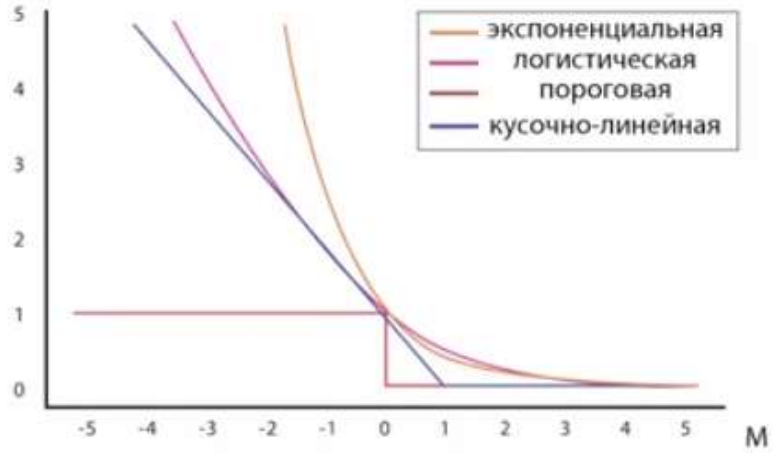
- **Функция оценки работы сети (функция потерь, функция ошибки)** – количество примеров из D, на которых решающая функция F ошибается.

$$L(D, F) = \sum_i [F(x_i) \neq y_i]$$

- Как правило, явно зависит от выходных сигналов сети и неявно от всех параметров сети.
- Выбирается так, что:
  - Равна нулю при всех правильных ответах
  - Больше нуля при ошибках
  - Является верхней границей доли неправильных ответов
  - Является гладкой (дифференцируемой)
- Позволяет:
  - количественно указать насколько хорошо или плохо сеть решает поставленную задачу.
  - свести обучение к задаче оптимизации: надеемся, что минимизация функции потерь приведёт к минимизации доли неправильных ответов.

Различные функции потерь

- Примеры функций потерь
  - Простая
  - Квадратичная
  - Прямоугольная
  - Логистическая
  - Экспоненциальная (с насыщением)
  - Перекрёстная энтропия



- https://ru.coursera.org/learn/supervised-learning/lecture/2AzdX/funktsii-potier-v-zadachakh-klassifikatsii

Виды обучения

- По размеченности данных:
  - **Обучение с учителем** (supervised) - обучение с привлечением лишь размеченных данных
  - **Обучение без учителя** (unsupervised) - без привлечения каких-либо размеченных данных для тренировки
  - **Обучение с частичным привлечением учителя** (semi-supervised) - использует обычно небольшое количество размеченных данных и большое количество неразмеченных данных.
- По фазам:
  - Предобучение отдельных фрагментов глубинной сети – например, с использованием ограниченной машины Больцмана
  - Дообучение – основное обучение глубинной сети

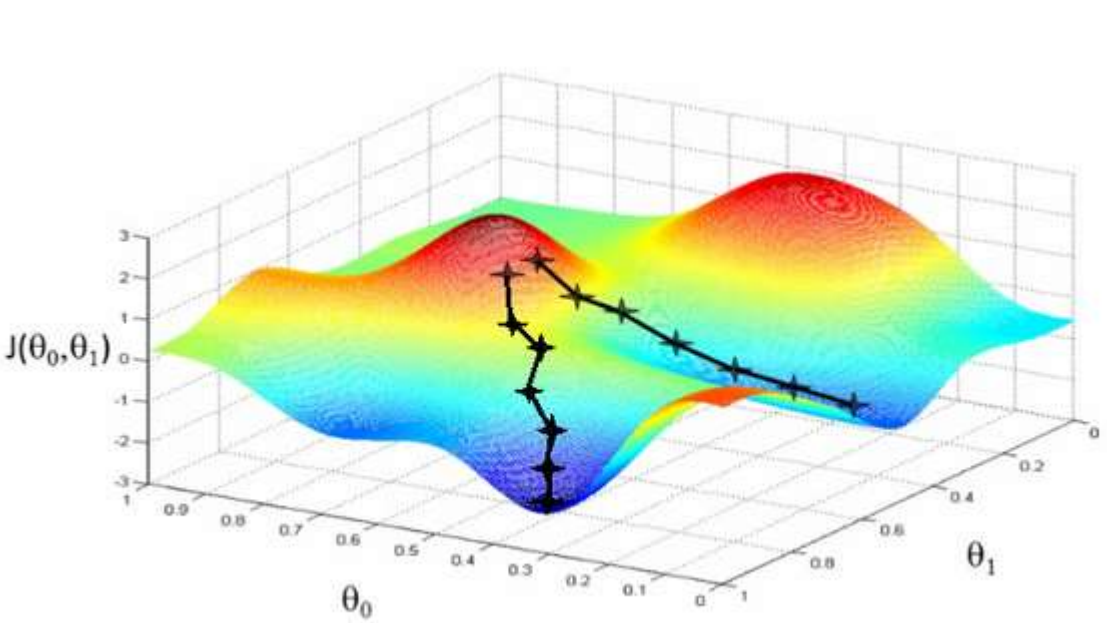
Термины

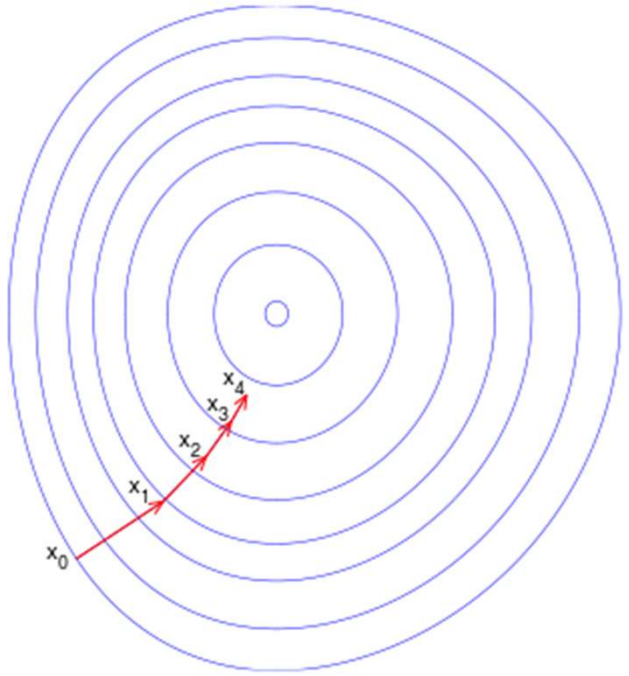
- **Итерация** – количество тренировочных наборов, пройденных сетью.
- **Эпоха** (epoche) – количество полных прохождений тренировочных наборов.
- **Ошибка** (error) – относительная величина, отражающая расхождение между ожидаемым и полученным ответами.
  - MAE – mean absolute error – средняя абсолютная ошибка
  - MSE – mean squared error – среднеквадратическая ошибка
  - Root MSE – корень СКО
  - Arctan

Методы оптимизации

- GD – Gradient Descent – Градиентный спуск
- SGD – Stochastic Gradient Descent – Стохастический градиентный спуск
- SGDm – Stochastic Gradient Descent with Momentum – Стохастический градиентный спуск с инерцией
- NGD – Normalized Gradient Descent – Нормализованный градиентный спуск
- AdaM – метод адаптивной инерции
- AdaGrad – метод адаптивного градиента
- AdaDelta – метод адаптивного шага обучения
- Rprop – resilient backpropagation – ускоренный МОРО
- RMSprop - Root Mean Square Propagation – Метод адаптивного скользящего среднего градиентов
- BGD – Batch Gradient Descent – пакетный градиентный спуск

Градиентный спуск





- Основная идея: двигаться в направлении наибольшего уменьшения ошибки

Градиентный спуск

- Требуется изменять весовые коэффициенты сети так, чтобы минимизировать среднюю ошибку на выходе при подаче на вход последовательности обучающих входных данных.
- Формально, чтобы сделать всего один шаг (одно изменение параметров сети), необходимо:
  - подать на вход сети последовательно абсолютно весь набор обучающих данных,
  - для каждого объекта обучающих данных вычислить ошибку и
  - рассчитать необходимую коррекцию коэффициентов сети (но не делать эту коррекцию),
  - и уже после подачи всех данных рассчитать сумму в корректировке каждого коэффициента сети (сумма градиентов) и
  - произвести коррекцию коэффициентов «на один шаг».

Градиентный спуск с инерцией

- Двигаясь в одном направлении, увеличивать шаг, имитируя движение по инерции с высоты вниз.

Стохастический градиентный спуск

- При большом наборе обучающих данных GD будет работать крайне медленно.
- На практике часто производят корректировку коэффициентов сети после каждого элемента обучения, где значение градиента аппроксимируются градиентом функции стоимости, вычисленном только на одном элементе обучения.
- Такой метод называют ***стохастическим градиентным спуском***.

Метод обратного распространения ошибки (МОРО)

Алгоритм: ***BackPropagation*** $(\eta, \alpha, \{x_i^d, t^d\}_{i=1,d=1}^{n,m}, \text{steps})$

1. Инициализировать $\{w_{ij}\}_{i,j}$ маленькими случайными значениями, $\{\Delta w_{ij}\}_{i,j} = 0$
2. Повторить NUMBER_OF_STEPS раз:

   .Для всех d от 1 до m:

   1. Подать $\{x_i^d\}$ на вход сети и подсчитать выходы o_i каждого узла.
   2. Для всех $k \in Outputs$

      $$\delta_k = o_k(1 - o_k)(t_k - o_k).$$

   3. Для каждого уровня l, начиная с предпоследнего:

      Для каждого узла j уровня l вычислить

      $$\delta_j = o_j(1 - o_j) \sum_{k \in Children(j)} \delta_k w_{j,k}.$$

   4. Для каждого ребра сети {i, j}

      $$\Delta w_{i,j}(n) = \alpha \Delta w_{i,j}(n-1) + (1 - \alpha)\eta \delta_j o_i.$$
      $$w_{i,j}(n) = w_{i,j}(n-1) + \Delta w_{i,j}(n).$$

3. Выдать значения w_{ij}.

где α — коэффициент инерциальности для сглаживания резких скачков при перемещении по поверхности целевой функции

Переобучение

- **Переобучение** – overtraining – это результат чрезмерной подгонки сети к обучающим примерам.
- Если в результате обучения нейронная сеть хорошо распознает примеры из обучающего множества, но не приобретает свойство обобщения, т.е. не распознает или плохо распознает любые другие примеры, кроме обучающих, то говорят, что сеть переобучена.

Паралич сети

- В процессе обучения сети значения весов могут в результате коррекции стать очень большими величинами.
- Это может привести к тому, что все или большинство нейронов будут функционировать при очень больших значениях в области, где производная сжимающей функции очень мала.
- Так как посылаемая обратно в процессе обучения ошибка пропорциональна этой производной, то процесс обучения может практически замереть.
- В теоретическом отношении эта проблема плохо изучена. Обычно этого избегают уменьшением размера шага η, но это увеличивает время обучения.
- Различные эвристики использовались для предохранения от паралича (Paralysis of backpropagation network) или для восстановления после него, но пока что они могут рассматриваться лишь как экспериментальные.

Теорема сходимости перцептрона

- Элементарный перцептрон, обучаемый по методу коррекции ошибки (с квантованием или без него), независимо от начального состояния весовых коэффициентов и последовательности появления стимулов всегда приведёт к достижению решения за конечный промежуток времени.

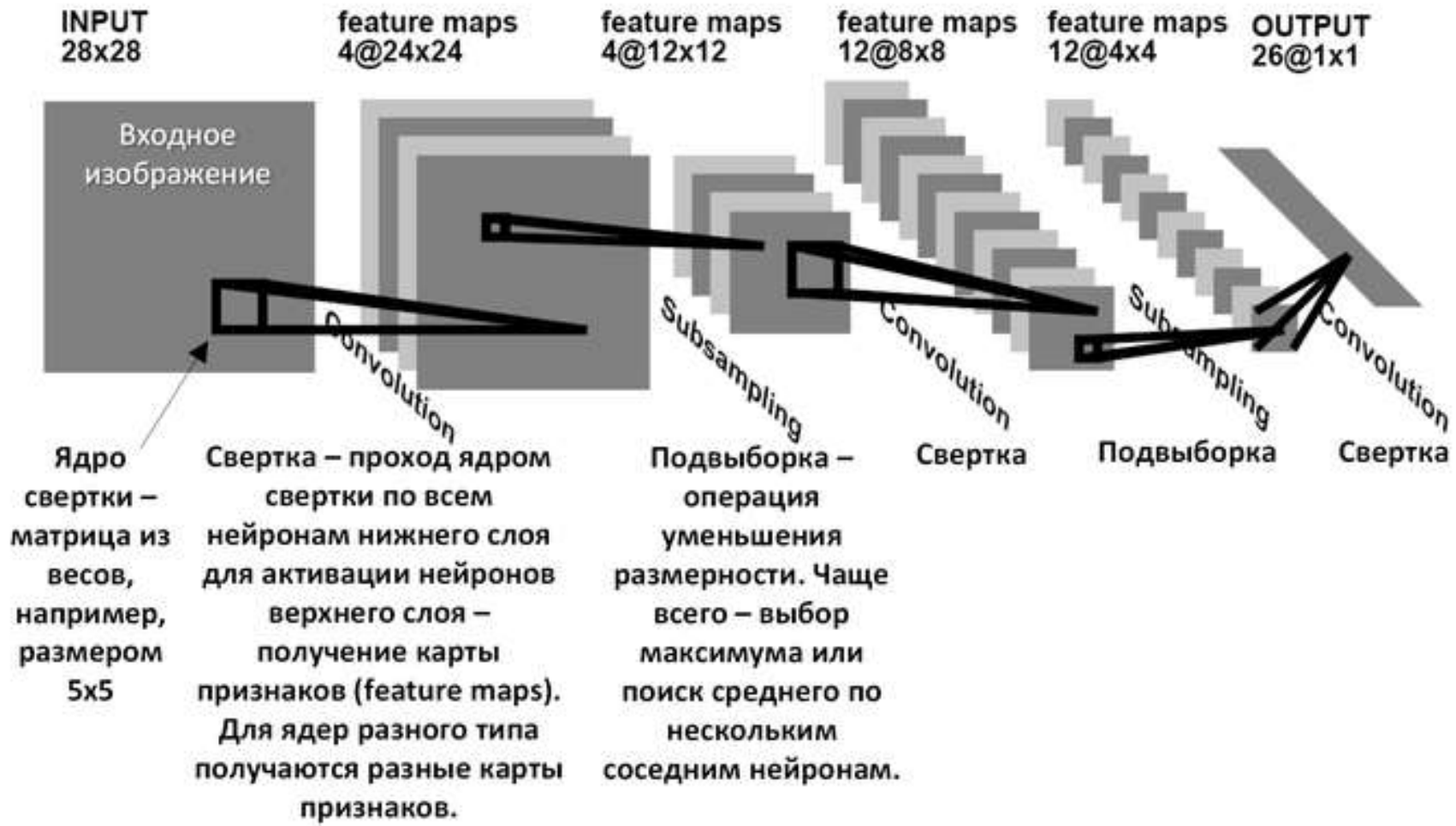
Свёрточные нейронные сети

Понятия
Архитектура
Обучение
Достоинства и недостатки

Основные понятия

- **CNN** – Convolutional neural network – свёрточная нейронная сеть – специальная архитектура искусственных нейронных сетей для обработки изображений.
- Основная идея свёрточных нейронных сетей заключается в чередовании **свёрточных слоев** (*convolution layers*) и **субдискретизирующих слоев** (*subsampling layers)* или **слоёв подвыборки** (*pooling layers*).
- Структура сети однонаправленная без обратных связей, принципиально многослойная.
- **Свёртка** – операция, состоящая в том, что каждый фрагмент изображения умножается на матрицу (**ядро**) свёртки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения.
- Свёртка в CNN аналогична методу скользящего окна взвешенного фильтра.

Архитектура CNN



Слои CNN

- Свёрточные слои – convolution layers
- Слой ректификации – rectification linear unit – ReLU
- Субдискретизирующие слои – subsampling layers
- Полносвязные слои – fully connected layers

Карты признаков – feature maps

- Ядро свёртки можно интерпретировать как графическое кодирование какого-либо признака, например, наличие наклонной линии под определенным углом.
- Тогда следующий слой, получившийся в результате операции свёртки такой матрицей весов, показывает наличие данного признака в обрабатываемом слое с его координатами.
- Формируется так называемая **карта признаков** (feature map).
- Проход каждым набором весов (разными ядрами) формирует свой собственный экземпляр карты признаков, делая нейронную сеть многоканальной (много независимых карт признаков на одном слое).
- Ядра свертки не закладываются исследователем заранее, а формируются самостоятельно путём обучения сети классическим МОРО.

Слой свёртки

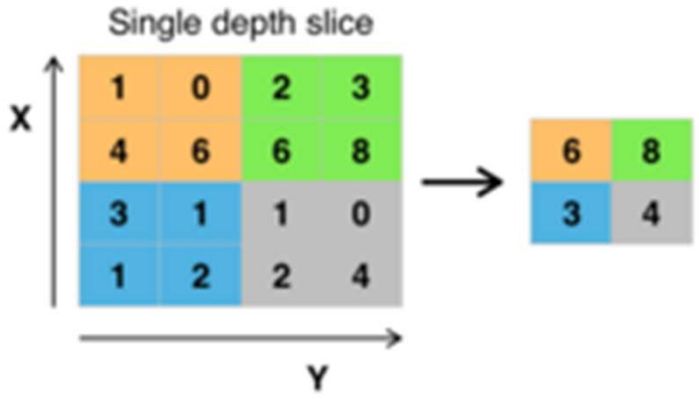
- Основной блок CNN
- Включает в себя для каждого цветового канала свой фильтр, *ядро свёртки* которого обрабатывает предыдущий слой по фрагментам (суммируя результаты матричного произведения для каждого фрагмента).
- Весовые коэффициенты ядра свёртки (небольшой матрицы) неизвестны и устанавливаются в процессе обучения.

Слой ReLU

- ReLU — *rectified linear unit* - блок линейной ректификации.
- Слой ReLU —функция активации после свёрточного слоя
- Для активации обычно выбирается ненасыщаемая полулинейная функция max(x,0).
- Такая функция показывает хорошие результаты при обучении и отвечает за отсечение ненужных деталей в канале (при отрицательном выходе).

Пулинг или слой субдискретизации

- Слой пулинга (подвыборка, субдискретизация) представляет собой нелинейное уплотнение карты признаков, при этом группа пикселей (обычно размера 2×2) уплотняется до одного пикселя, проходя нелинейное преобразование.
- Наиболее употребительна при этом функция максимума.



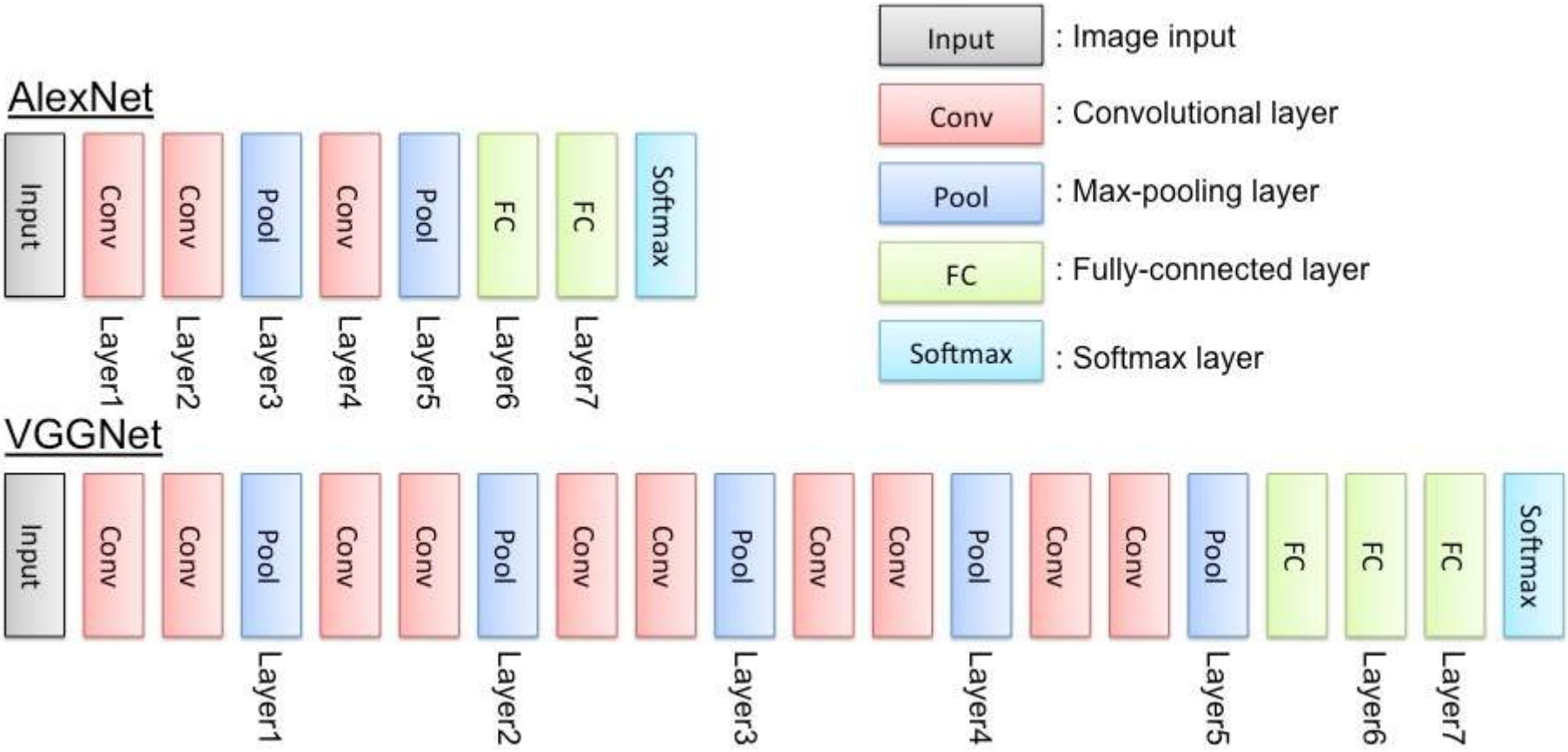
Операция субдискретизации

- Позволяет существенно уменьшить пространственный объём изображения.
- Информация о факте наличия искомого признака важнее точного знания его координат, поэтому из нескольких соседних нейронов карты признаков выбирается максимальный и принимается за один нейрон уплотнённой карты признаков меньшей размерности.
- За счёт данной операции, помимо ускорения дальнейших вычислений, сеть становится более инвариантной к масштабу входного изображения.
- Фильтрация уже ненужных деталей помогает сети не переобучаться.
- Слой пулинга, как правило, вставляется после слоя свёртки перед слоем следующей свёртки.

Полносвязные слои

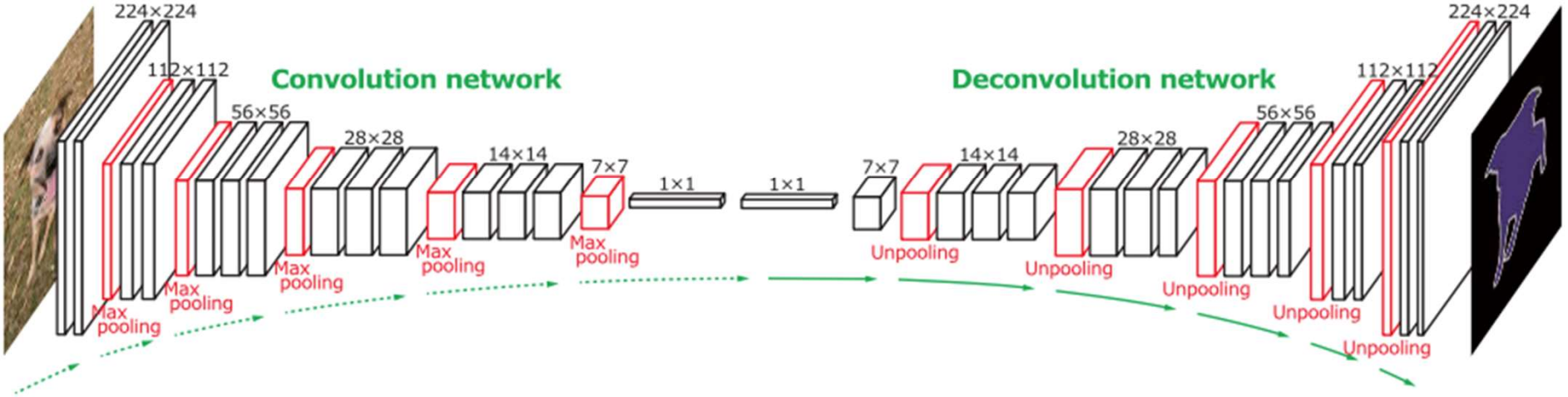
- После нескольких прохождений свёртки изображения и уплотнения с помощью пулинга система перестраивается от конкретной сетки пикселей с высоким разрешением к более абстрактным картам признаков.
- Как правило на каждом следующем слое увеличивается число каналов и уменьшается размерность изображения в каждом канале.
- В конце концов остаётся большой набор каналов, хранящих небольшое число данных (даже один параметр), которые интерпретируются как самые абстрактные понятия, выявленные из исходного изображения.
- Эти данные объединяются и передаются на обычную полносвязную нейронную сеть, которая тоже может состоять из нескольких слоёв. При этом полносвязные слои уже утрачивают пространственную структуру пикселей и обладают сравнительно небольшой размерностью (по отношению к количеству пикселей исходного изображения).

Примеры архитектур CNN

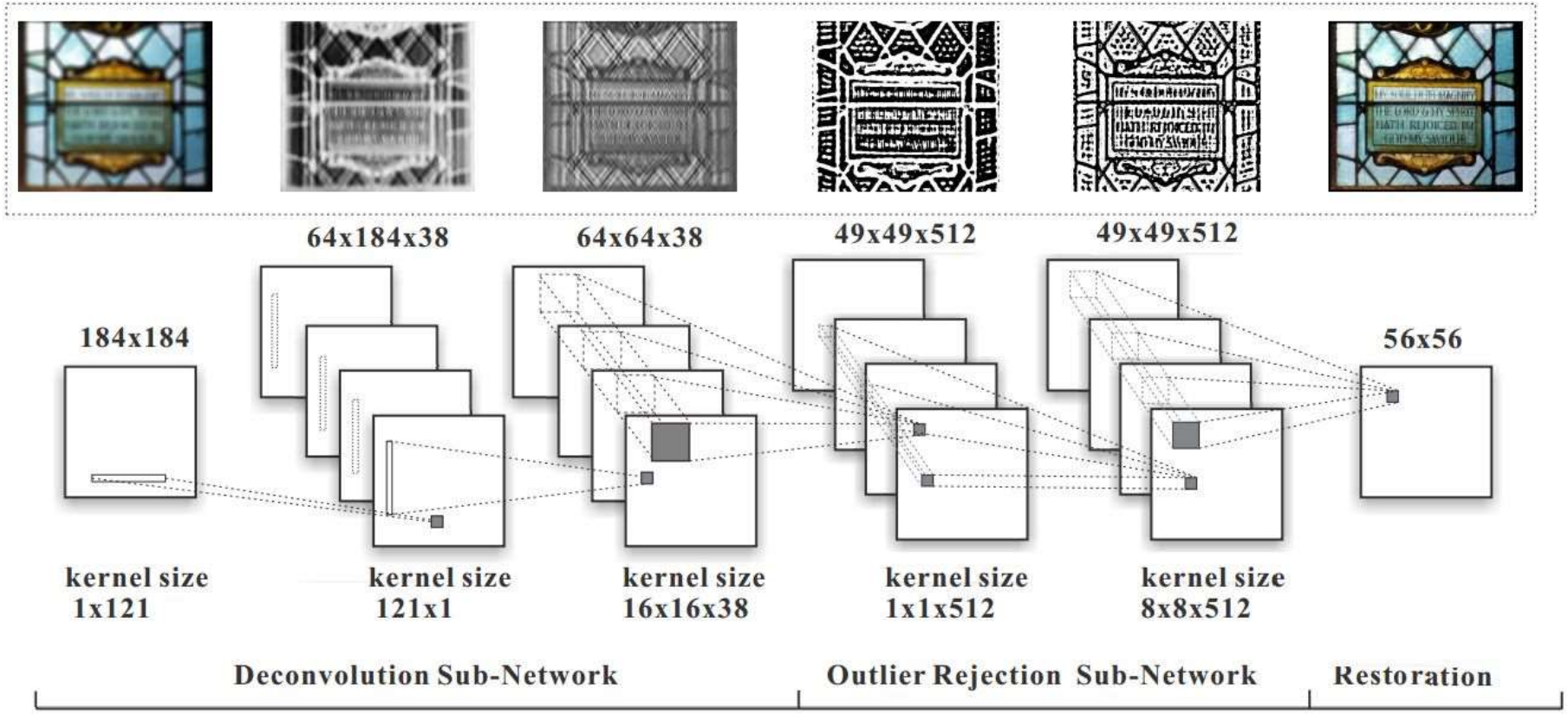


Развёртывающиеся сети

- Deconvolutional neural networks
- DNN – Обратные по отношению к CNN – способны реконструировать изображения на основе карт признаков
- Unpooling-слои увеличивают размерность выходного изображения



Примеры


64x184x38
64x64x38
49x49x512
49x49x512
184x184
56x56
kernel size
1x121
kernel size
121x1
kernel size
16x16x38
kernel size
1x1x512
kernel size
8x8x512
Deconvolution Sub-Network
Outlier Rejection Sub-Network
Restoration

Что почитать

- Теорема сходимости перцептрона https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D1%81%D1%85%D0%BE%D0%B4%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8_%D0%BF%D0%B5%D1%80%D1%86%D0%B5%D0%BF%D1%82%D1%80%D0%BE%D0%BD%D0%B0
- *Фрэнк Розенблатт.* Принципы нейродинамики и теория механизмов мозга — М.: «Мир», 1965.
- Шпаргалка по разновидностям нейронных сетей https://tproger.ru/translations/neural-network-zoo-1/, http://www.asimovinstitute.org/wp-content/uploads/2016/09/networkZooPoster.png
- Самое главное о нейронных сетях. Лекция в Яндексе https://habrahabr.ru/company/yandex/blog/307260/
- Предобучение нейронной сети с использованием ограниченной машины Больцмана https://habrahabr.ru/post/163819/

Что ещё почитать

- https://www.quora.com/What-are-differences-between-update-rules-like-AdaDelta-RMSProp-AdaGrad-and-AdaM
- https://www.coursera.org/specializations/machine-learning-data-analysis
- Другие архитектуры
  - сеть Элмана с обратным распространением [Elman, 1990];
  - нейросетевая модель когнитрона [Fukushima, 1975] и неокогнитрона Фукушимы [Fukushima, 1980];
  - нейросетевая модель распознавания иероглифических текстов [Кугаевских, 2012];
- Deep Learning, NLP, and Representations https://habrahabr.ru/post/253227/