



HybridTech 31 января 2018 в 16:47

# Сверточная нейронная сеть, часть 1: структура, топология, функции активации и обучающее множество

Машинное обучение \*

Из песочницы

## Предисловие

Данные статьи ([часть 2](#)) являются частью моей научной работы в ВУЗе, которая звучала так: «Программный комплекс детектирования лиц в видеопотоке с использованием сверточной нейронной сети». Цель работы была — улучшение скоростных характеристик в процессе детектирования лиц в видеопотоке. В качестве видеопотока использовалась камера смартфона, писалось десктопное ПС (язык Kotlin) для создания и обучения сверточной нейросети, а также мобильное приложение под Android (язык Kotlin), которая использовала обученную сеть и «пыталась» распознать лица из видеопотока камеры. Результаты скажу получились так себе, использовать точную копию предложенной мной топологии на свой страх и риск (я бы не рекомендовал).

### ► Теоретические задачи

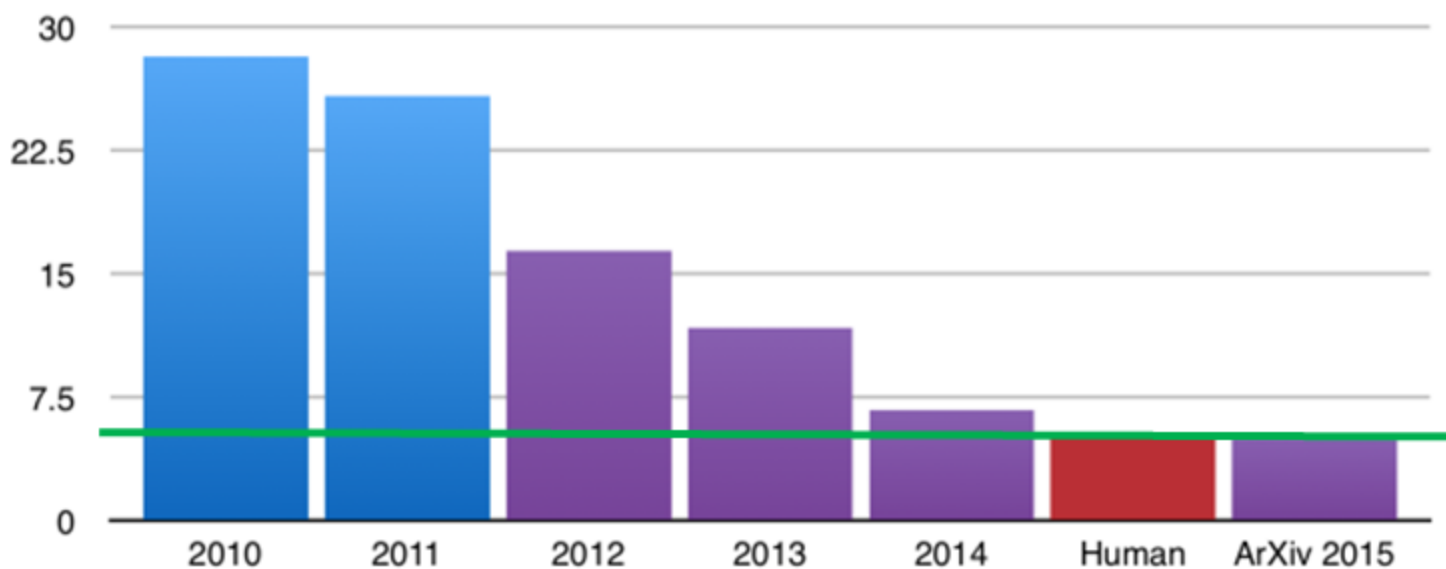
## Введение

Наилучшие результаты в области распознавания лиц показала Convolutional Neural Network или сверточная нейронная сеть (далее – СНС), которая является логическим развитием идей таких архитектур НС как когнитрона и неокогнитрона. Успех обусловлен возможностью учета двумерной топологии изображения, в отличие от многослойного персептрона.

Сверточные нейронные сети обеспечивают частичную устойчивость к изменениям масштаба, смещениям, поворотам, смене ракурса и прочим искажениям. Сверточные нейронные сети объединяют три архитектурных идеи, для обеспечения инвариантности к изменению масштаба, повороту сдвигу и пространственным искажениям:

- локальные рецепторные поля (обеспечивают локальную двумерную связность нейронов);
- общие синаптические коэффициенты (обеспечивают детектирование некоторых черт в любом

На данный момент сверточная нейронная сеть и ее модификации считаются лучшими по точности и скорости алгоритмами нахождения объектов на сцене. Начиная с 2012 года, нейросети занимают первые места на известном международном конкурсе по распознаванию образов ImageNet.



Именно поэтому в своей работе я использовал сверточную нейронную сеть, основанную на принципах неокогнитрона и дополненную обучением по алгоритму обратного распространения ошибки.

## Структура сверточной нейронной сети

СНС состоит из разных видов слоев: сверточные (convolutional) слои, субдискретизирующие (subsampling, подвыборка) слои и слои «обычной» нейронной сети – персептрона, в соответствии с рисунком 1.

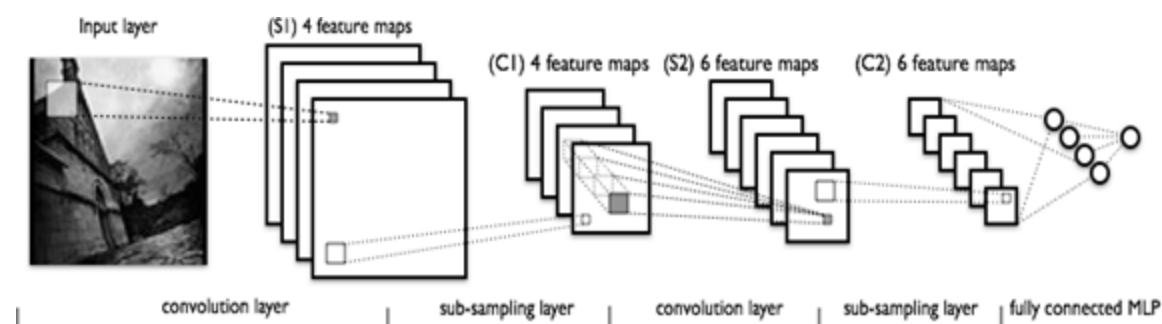


Рисунок 1 – топология сверточной нейронной сети

Первые два типа слоев (convolutional, subsampling), чередуясь между собой, формируют входной вектор признаков для многослойного персептрона.

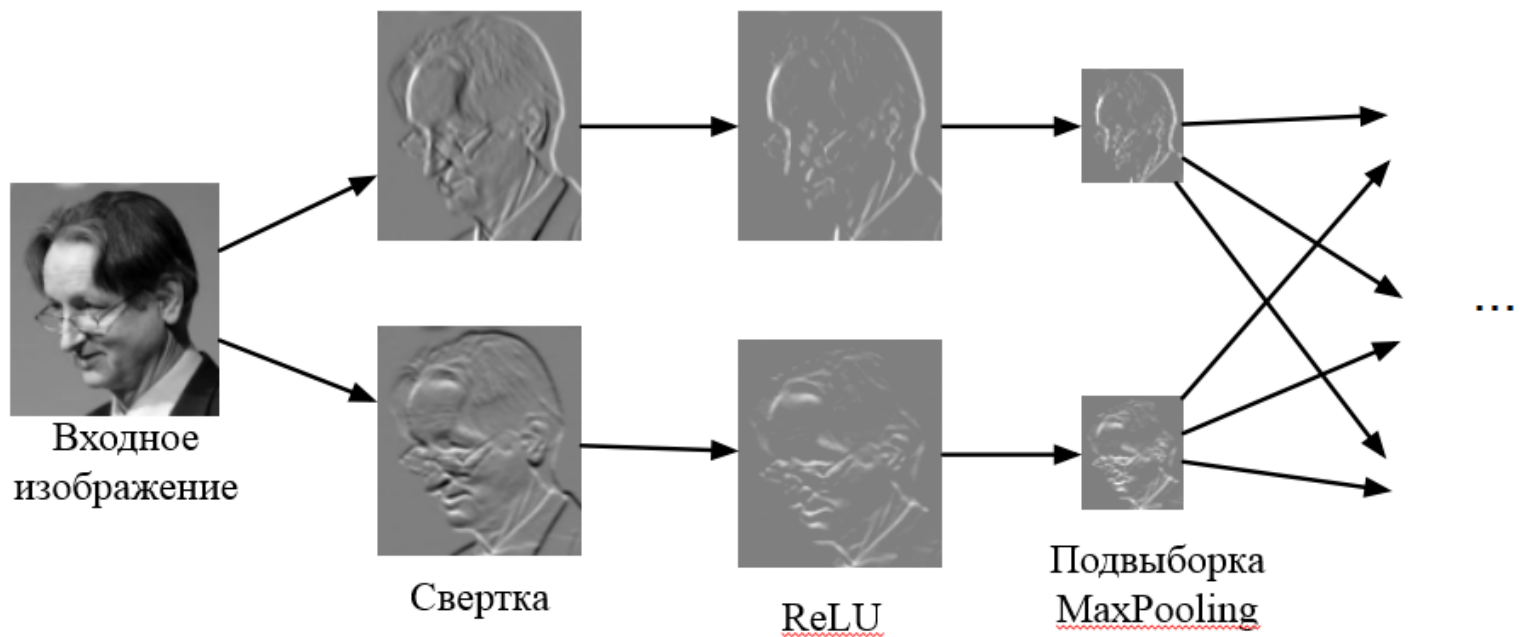
Свое название сверточная сеть получила по названию операции – свертка, суть которой будет описана дальше.

Сверточные сети являются удачной серединой между биологически правдоподобными сетями и обычным многослойным персептроном. На сегодняшний день лучшие результаты в распознавании изображений получают с их помощью. В среднем точность распознавания таких сетей превосходит обычные ИНС на 10-15%. СНС – это ключевая технология Deep Learning.

Основной причиной успеха СНС стало концепция общих весов. Несмотря на большой размер, эти сети имеют небольшое количество настраиваемых параметров по сравнению с их предком – неоконитроном. Имеются варианты СНС (Tiled Convolutional Neural Network), похожие на неоконитрон, в таких сетях происходит, частичный отказ от связанных весов, но алгоритм обучения остается тем же и основывается на обратном распространении ошибки. СНС могут быстро работать на последовательной машине и быстро обучаться за счет чистого распараллеливания процесса свертки по каждой карте, а также обратной свертки при распространении ошибки по сети.

На рисунке ниже продемонстрирована визуализация свертки и подвыборки:

### Операция свертки + ReLU + подвыборка



#### ► Модель нейрона

### Топология сверточной нейросети

Определение топологии сети ориентируется на решаемую задачу, данные из научных статей и собственный экспериментальный опыт.

Можно выделить следующие этапы влияющие на выбор топологии:

- определить решаемую задачу нейросетью (классификация, прогнозирование, модификация);
- определить ограничения в решаемой задаче (скорость, точность ответа);
- определить входные (тип: изображение, звук, размер: 100x100, 30x30, формат: RGB, в градациях серого) и выходные данные (количество классов).

Решаемая моей нейросетью задача – классификация изображений, конкретно лиц. Накладываемые ограничения на сеть — это скорость ответа – не более 1 секунды и точность распознавания не менее 70%. Общая топология сети в соответствии с рисунком 2.

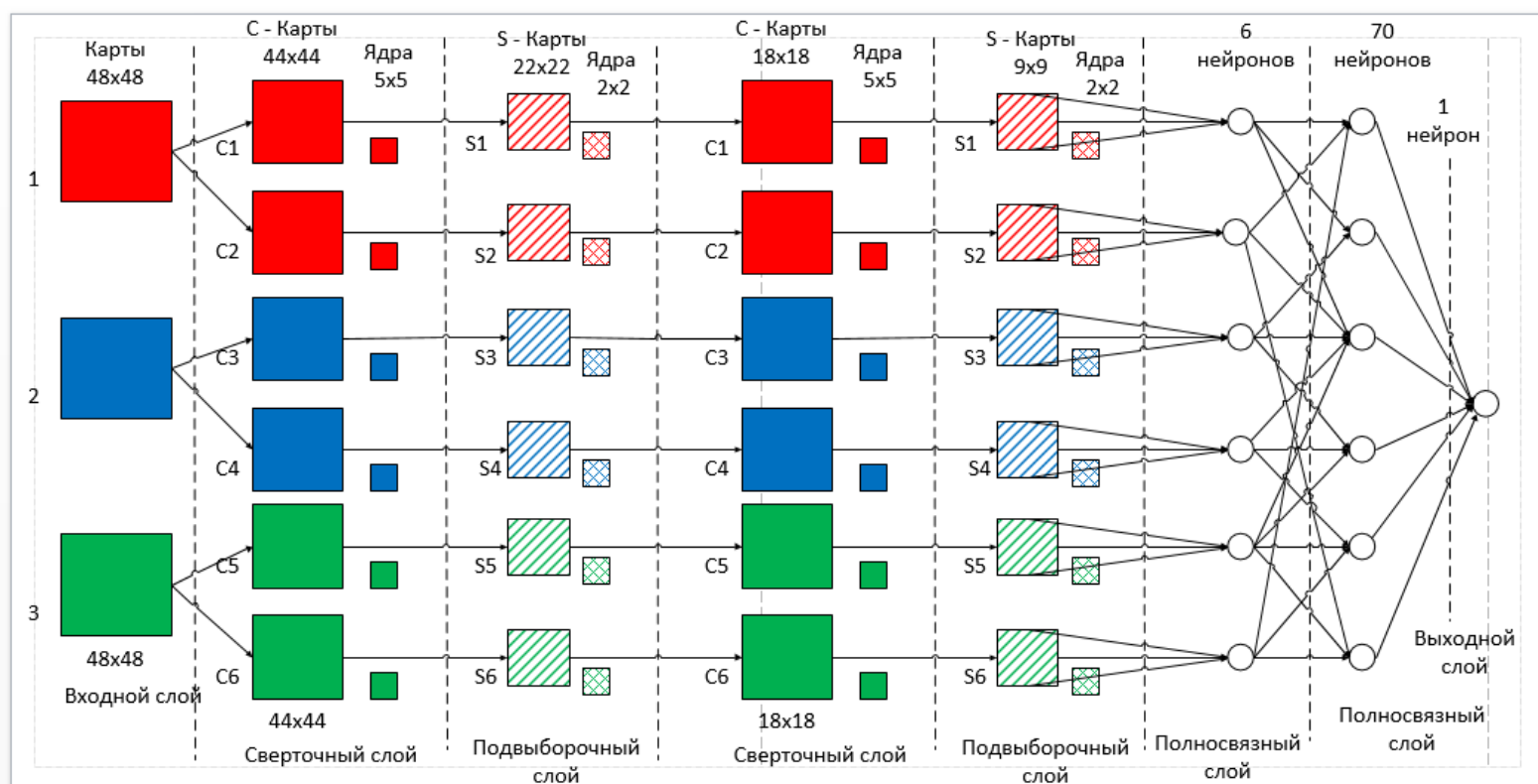


Рисунок 2 — Топология сверточной нейросети

## Входной слой

Входные данные представляют из себя цветные изображения типа JPEG, размера 48x48 пикселей. Если размер будет слишком велик, то вычислительная сложность повысится, соответственно ограничения на скорость ответа будут нарушены, определение размера в данной задаче решается методом подбора. Если выбрать размер слишком маленький, то сеть не сможет выявить ключевые признаки лиц. Каждое изображение разбивается на 3 канала: красный, синий, зеленый. Таким образом получается 3 изображения размера 48x48 пикселей.

Входной слой учитывает двумерную топологию изображений и состоит из нескольких карт (матриц), карта может быть одна, в том случае, если изображение представлено в оттенках серого, иначе их 3, где каждая карта соответствует изображению с конкретным каналом (красным, синим и зеленым).

Входные данные каждого конкретного значения пикселя нормализуются в диапазон от 0 до 1, по формуле:

$$f(p, min, max) = \frac{p - min}{max - min},$$

где  $f$  – функция нормализации;

$p$  – значение конкретного цвета пикселя от 0 до 255;

$min$  – минимальное значение пикселя – 0;

$max$  – максимальное значение пикселя – 255.

## Сверточный слой

Сверточный слой представляет из себя набор карт (другое название – карты признаков, в обиходе это обычные матрицы), у каждой карты есть синаптическое ядро (в разных источниках его называют по-разному: сканирующее ядро или фильтр).

Количество карт определяется требованиями к задаче, если взять большое количество карт, то повысится качество распознавания, но увеличится вычислительная сложность. Исходя из анализа научных статей, в большинстве случаев предлагается брать соотношение один к двум, то есть каждая карта предыдущего слоя (например, у первого сверточного слоя, предыдущим является входной) связана с двумя картами сверточного слоя, в соответствии с рисунком 3. Количество карт – 6.

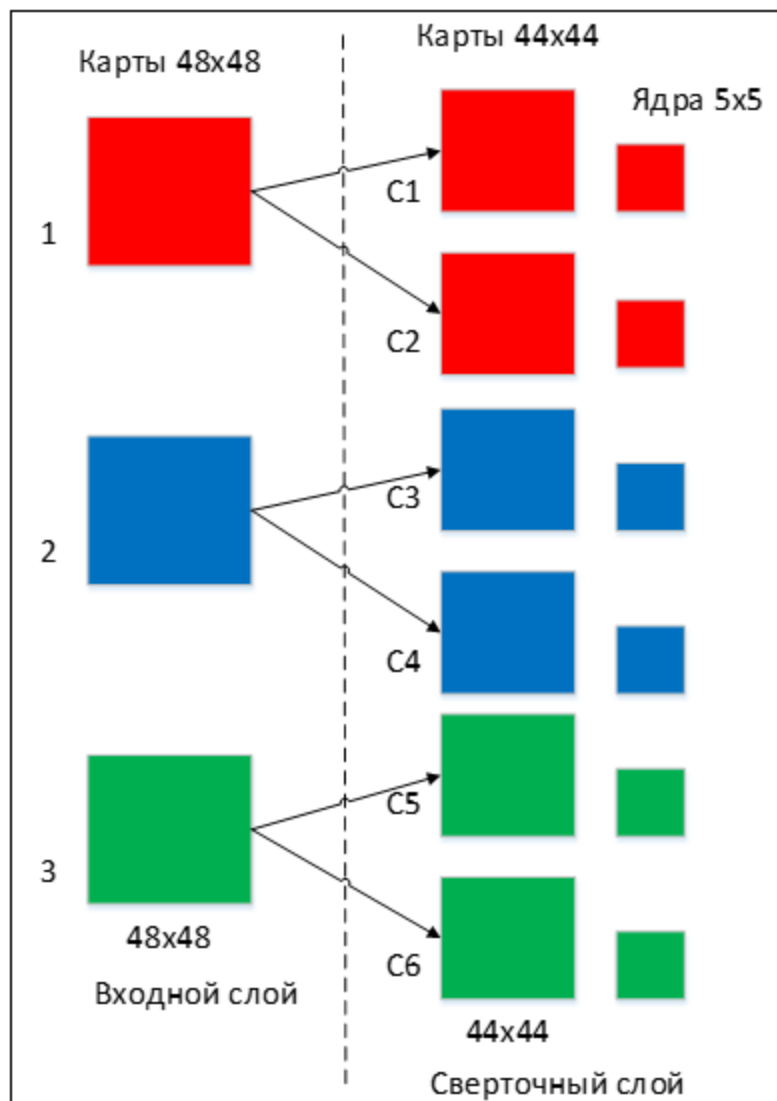


Рисунок 3 — Организация связей между картами сверточного слоя и предыдущего

Размер у всех карт сверточного слоя – одинаковы и вычисляются по формуле 2:

$$(w, h) = (mW - kW + 1, mH - kH + 1),$$

где  $(w, h)$  – вычисляемый размер сверточной карты;

$mW$  – ширина предыдущей карты;

$mH$  – высота предыдущей карты;

$kW$  – ширина ядра;

$kH$  – высота ядра.

Ядро представляет из себя фильтр или окно, которое скользит по всей области предыдущей карты и находит определенные признаки объектов. Например, если сеть обучали на множестве лиц, то одно из ядер могло бы в процессе обучения выдавать наибольший сигнал в области глаза, рта, брови или носа.

другое ядро могло бы выявлять другие признаки. Размер ядра обычно берут в пределах от 3x3 до 7x7.

Если размер ядра маленький, то оно не сможет выделить какие-либо признаки, если слишком большое, то увеличивается количество связей между нейронами. Также размер ядра выбирается таким, чтобы размер карт сверточного слоя был четным, это позволяет не терять информацию при уменьшении размерности в подвыборочном слое, описанном ниже.

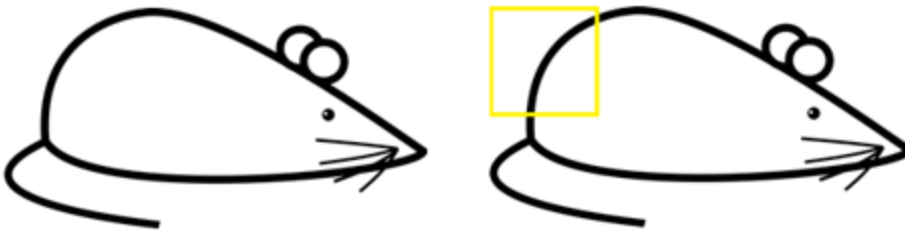
Ядро представляет собой систему разделяемых весов или синапсов, это одна из главных особенностей сверточной нейросети. В обычной многослойной сети очень много связей между нейронами, то есть синапсов, что весьма замедляет процесс детектирования. В сверточной сети – наоборот, общие веса позволяет сократить число связей и позволить находить один и тот же признак по всей области изображения.

## Пример ядра с обученным признаком

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



## Входное изображение



В этом месте, где желтое окно, будет большой отклик (сигнал), что говорит о наличии этого признака на изображении.

Изначально значения каждой карты сверточного слоя равны 0. Значения весов ядер задаются случайным образом в области от -0.5 до 0.5. Ядро скользит по предыдущей карте и производит операцию свертки, которая часто используется для обработки изображений, формула:

$$(f * g)[m, n] = \sum_{k, l} f[m - k, n - l] * g[k, l],$$

где  $f$  – исходная матрица изображения;

$g$  – ядро свертки.

Неформально эту операцию можно описать следующим образом — окном размера ядра  $g$  проходим с заданным шагом (обычно 1) все изображение  $f$ , на каждом шаге поэлементно умножаем содержимое окна на ядро  $g$ , результат суммируется и записывается в матрицу результата, как на рисунке 4.

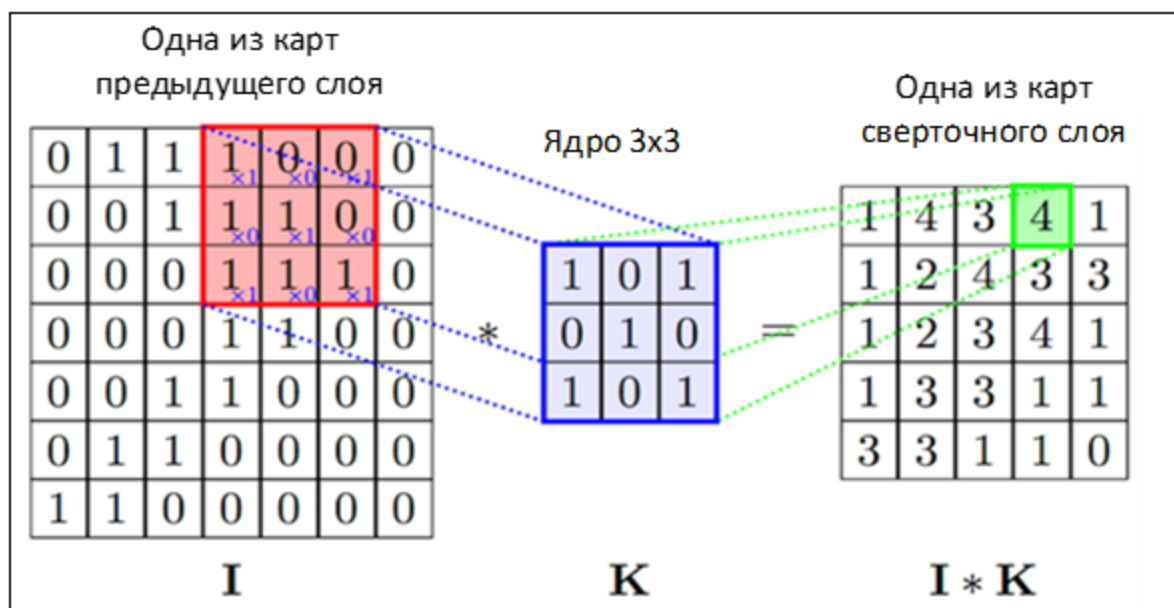
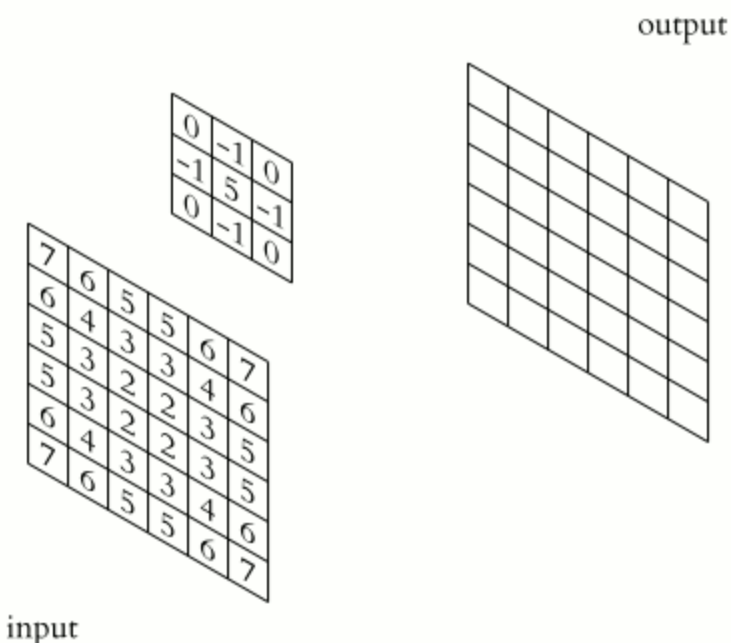


Рисунок 4 — Операция свертки и получение значений сверточной карты (valid)



Операция свертки и получение значений сверточной карты. Ядро смещено, новая карта получается



того же размера, что и предыдущая (same)

При этом в зависимости от метода обработки краев исходной матрицы результат может быть меньше исходного изображения (valid), такого же размера (same) или большего размера (full), в соответствии с рисунком 5.

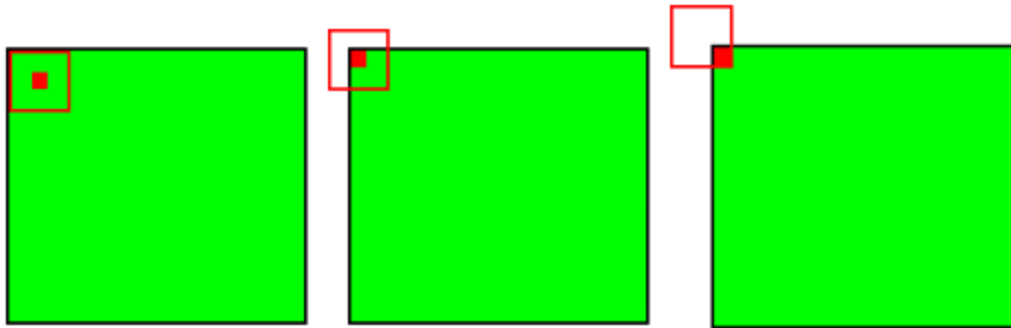


Рисунок 5 — Три вида свертки исходной матрицы

В упрощенном виде этот слой можно описать формулой:

$$x^l = f(x^{l-1} * k^l + b^l),$$

где  $x^l$  – выход слоя  $l$ ;

$f()$  – функция активации;

$b^l$  – коэффициент сдвига слоя  $l$ ;

$*$  – операция свертки входа  $x$  с ядром  $k$ .

При этом за счет краевых эффектов размер исходных матриц уменьшается, формула:

$$x_j^l = f\left(\sum_i x_i^{l-1} * k_j^l + b_j^l\right),$$

где  $x_j^l$  – карта признаков  $j$  (выход слоя  $l$ );

$f()$  – функция активации;

$b^l$  – коэффициент сдвига слоя  $l$  для карты признаков  $j$ ;

$k_j^l$  – ядро свертки  $j$  карты, слоя  $l$ ;

$*$  – операция свертки входа  $x$  с ядром  $k$ .

Подвыборочный слой

## Подвыборочный слой

Подвыборочный слой также, как и сверточный имеет карты, но их количество совпадает с предыдущим (сверточным) слоем, их 6. Цель слоя – уменьшение размерности карт предыдущего слоя. Если на предыдущей операции свертки уже были выявлены некоторые признаки, то для дальнейшей обработки настолько подробное изображение уже не нужно, и оно уплотняется до менее подробного. К тому же фильтрация уже ненужных деталей помогает не переобучаться.

В процессе сканирования ядром подвыборочного слоя (фильтром) карты предыдущего слоя, сканирующее ядро не пересекается в отличие от сверточного слоя. Обычно, каждая карта имеет ядро размером  $2 \times 2$ , что позволяет уменьшить предыдущие карты сверточного слоя в 2 раза. Вся карта признаков разделяется на ячейки  $2 \times 2$  элемента, из которых выбираются максимальные по значению.

Обычно в подвыборочном слое применяется функция активации RelU. Операция подвыборки (или MaxPooling – выбор максимального) в соответствии с рисунком 6.

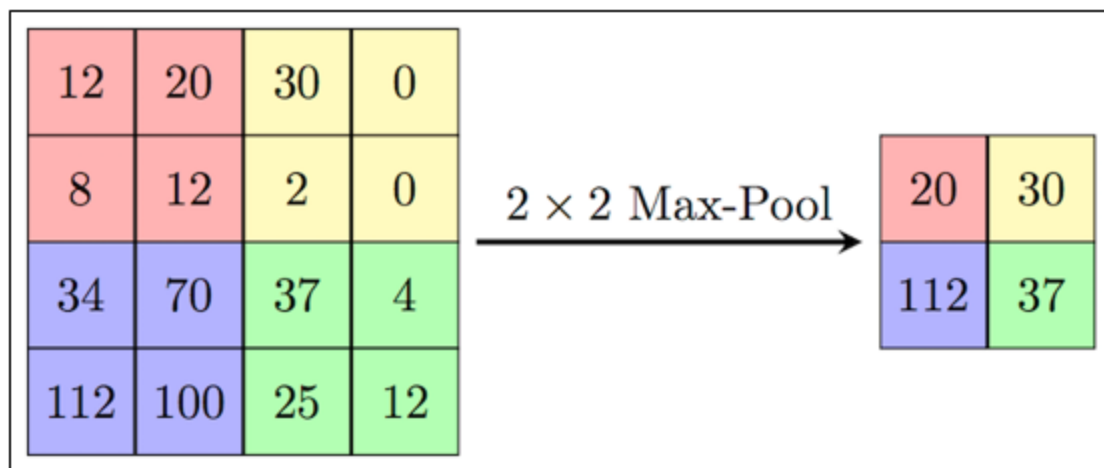


Рисунок 6 — Формирование новой карты подвыборочного слоя на основе предыдущей карты сверточного слоя. Операция подвыборки (Max Pooling)

Формально слой может быть описан формулой:

$$x^l = f(a^l * subsample(x^{l-1}) + b^l),$$

где  $x^l$  – выход слоя  $l$ ;

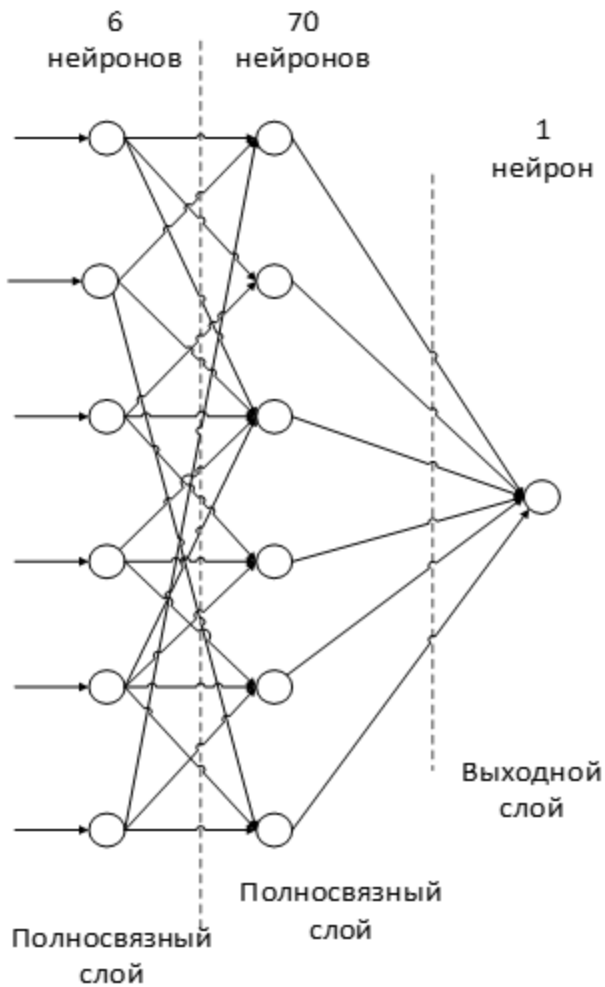
$f()$  – функция активации;

$a^l, b^l$  – коэффициенты сдвига слоя  $l$ ;

$subsample()$  – операция выборки локальных максимальных значений.

## Полносвязный слой

Последний из типов слоев это слой обычного многослойного персептрона. Цель слоя – классификация, моделирует сложную нелинейную функцию, оптимизируя которую, улучшается качество распознавания.



Нейроны каждой карты предыдущего подвыборочного слоя связаны с одним нейроном скрытого слоя. Таким образом число нейронов скрытого слоя равно числу карт подвыборочного слоя, но связи могут быть не обязательно такими, например, только часть нейронов какой-либо из карт подвыборочного слоя быть связана с первым нейроном скрытого слоя, а оставшаяся часть со вторым, либо все нейроны первой карты связаны с нейронами 1 и 2 скрытого слоя. Вычисление значений нейрона можно описать формулой:

$$x_j^l = f \left( \sum_i x_i^{l-1} * w_{i,j}^{l-1} + b_j^{l-1} \right),$$

где  $x_j^l$  – карта признаков  $j$  (выход слоя  $l$ );

$f()$  – функция активации;

$b^l$  – коэффициент сдвига слоя  $l$ ;

$w_{i,j}^l$  – матрица весовых коэффициентов слоя  $l$ .

## Выходной слой

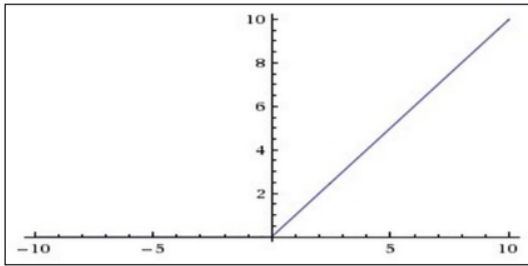
Выходной слой связан со всеми нейронами предыдущего слоя. Количество нейронов соответствует количеству распознаваемых классов, то есть 2 – лицо и не лицо. Но для уменьшения количества связей и вычислений для бинарного случая можно использовать один нейрон и при использовании в качестве функции активации гиперболический тангенс, выход нейрона со значением -1 означает принадлежность к классу “не лица”, напротив выход нейрона со значением 1 – означает принадлежность к классу лиц.

## Выбор функции активации

Одним из этапов разработки нейронной сети является выбор функции активации нейронов. Вид функции активации во многом определяет функциональные возможности нейронной сети и метод обучения этой сети. Классический алгоритм обратного распространения ошибки хорошо работает на двухслойных и трехслойных нейронных сетях, но при дальнейшем увеличении глубины начинает испытывать проблемы. Одна из причин — так называемое затухание градиентов. По мере распространения ошибки от выходного слоя к входному на каждом слое происходит домножение текущего результата на производную функции активации. Производная у традиционной сигмоидной функции активации меньше единицы на всей области определения, поэтому после нескольких слоев ошибка станет близкой к нулю. Если же, наоборот, функция активации имеет неограниченную производную (как, например, гиперболический тангенс), то может произойти взрывное увеличение ошибки по мере распространения, что приведет к неустойчивости процедуры обучения.

В данной работе в качестве функции активации в скрытых и выходном слоях применяется гиперболический тангенс, в сверточных слоях применяется ReLU. Рассмотрим наиболее распространенные функции активации, применяемые в нейронных сетях.

## ReLU (rectified linear unit)



$$f(s) = \max(0, s)$$

$$f'(s) = \begin{cases} 1, & s > 0 \\ \text{rand}(0.01, 0.05), & s \leq 0 \end{cases}$$

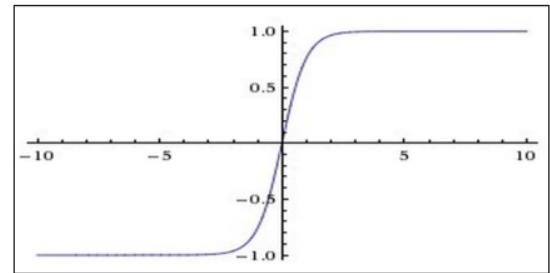
### Достоинства

- лишена ресурсоемких операций
- отсекает ненужные детали
- отсутствует разрастание/затухание градиента
- быстрое обучение

### Недостатки

- не всегда надежна, в процессе обучения может "умирать"
- сильно зависима от инициализации весов

## Гиперболический тангенс



$$f(s) = \frac{e^{2s} - 1}{e^{2s} + 1}$$

$$f'(s) = 1 - f(s)^2$$

### Достоинства

- простое вычисление производной через значение своей функции
- область значений от -1 до 1

### Недостатки

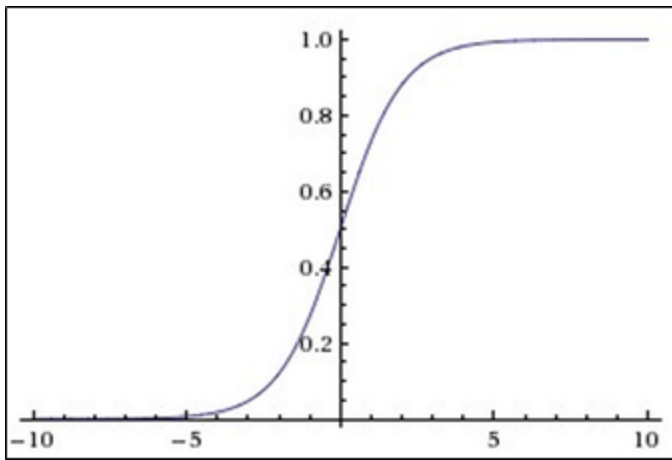
- затухание или увеличение градиента
- ресурсоемкая по сравнению с ReLU

## Функция активации сигмоиды

Эта функция относится к классу непрерывных функций и принимает на входе произвольное вещественное число, а на выходе дает вещественное число в интервале от 0 до 1. В частности, большие (по модулю) отрицательные числа превращаются в ноль, а большие положительные – в единицу. Исторически сигмоида находила широкое применение, поскольку ее выход хорошо интерпретируется, как уровень активации нейрона: от отсутствия активации (0) до полностью насыщенной активации (1). Сигмоида (sigmoid) выражается формулой:

$$f(s) = \frac{1}{1 + e^{-s}}.$$

График сигмоидальной функции в соответствии с рисунком ниже:



Крайне нежелательное свойство сигмоиды заключается в том, что при насыщении функции с той или иной стороны (0 или 1), градиент на этих участках становится близок к нулю.

Напомним, что в процессе обратного распространения ошибки данный (локальный) градиент умножается на общий градиент. Следовательно, если локальный градиент очень мал, он фактически обнуляет общий градиент. В результате, сигнал почти не будет проходить через нейрон к его весам и рекурсивно к его данным. Кроме того, следует быть очень осторожным при инициализации весов сигмоидных нейронов, чтобы предотвратить насыщение. Например, если исходные веса имеют слишком большие значения, большинство нейронов перейдет в состояние насыщения, в результате чего сеть будет плохо обучаться.

Сигмоидальная функция является:

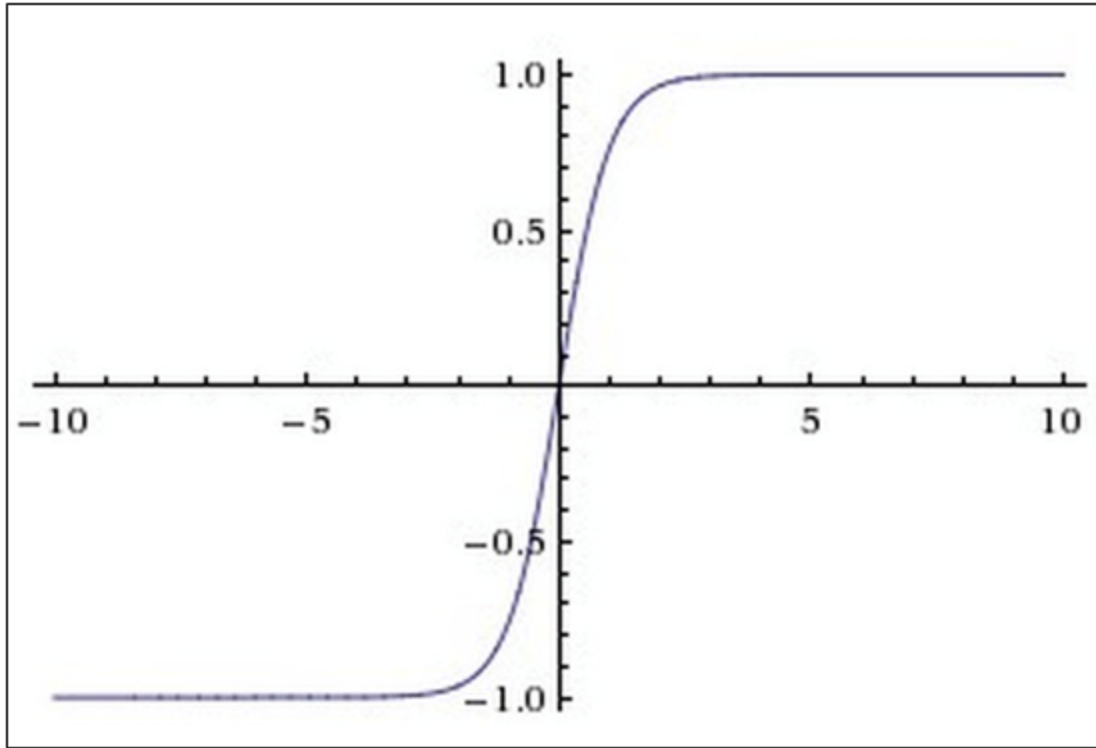
- непрерывной;
- монотонно возрастающей;
- дифференцируемой.

### Функция активации гиперболический тангенс

В данной работе в качестве активационной функции для скрытых и выходного слоев используется гиперболический тангенс. Это обусловлено следующими причинами:

- симметричные активационные функции, типа гиперболического тангенса обеспечивают более быструю сходимость, чем стандартная логистическая функция;
- функция имеет непрерывную первую производную;
- функция имеет простую производную, которая может быть вычислена через ее значение, что дает экономию вычислений.

График функции гиперболического тангенса показан на рисунке:

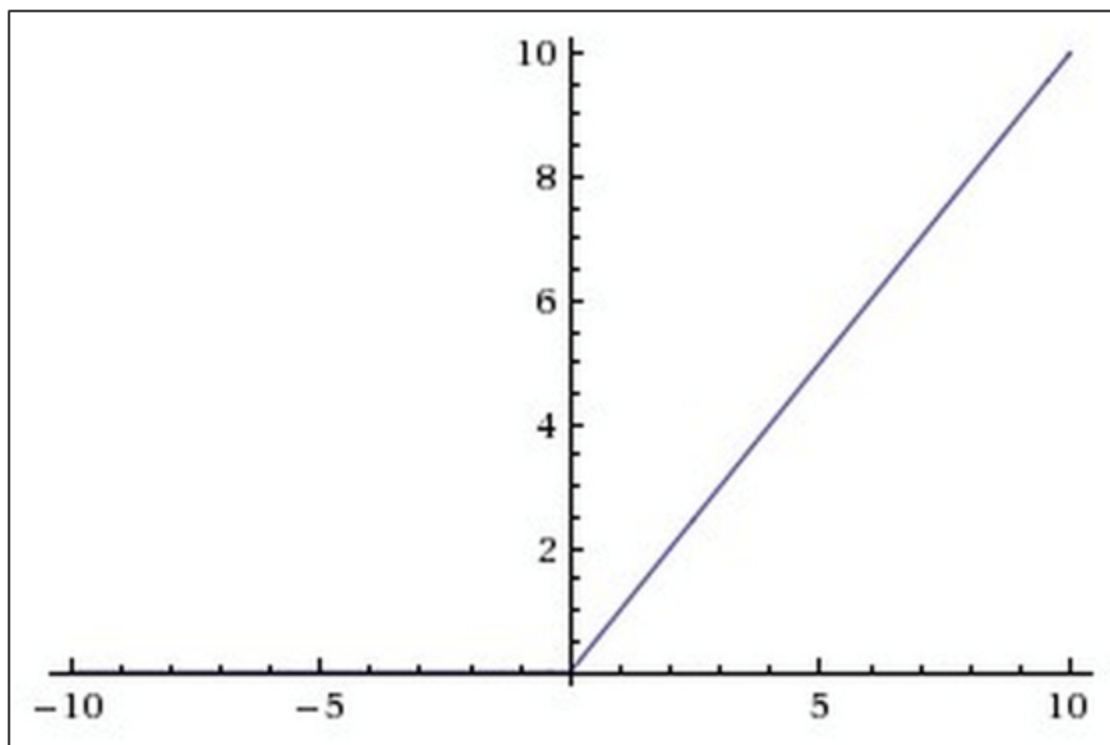


### Функция активации ReLU

Известно, что нейронные сети способны приблизить сколь угодно сложную функцию, если в них достаточно слоев и функция активации является нелинейной. Функции активации вроде сигмоидной или тангенциальной являются нелинейными, но приводят к проблемам с затуханием или увеличением градиентов. Однако можно использовать и гораздо более простой вариант — выпрямленную линейную функцию активации (rectified linear unit, ReLU), которая выражается формулой:

$$f(s) = \max(0, s)$$

График функции ReLU в соответствии с рисунком ниже:



Преимущества использования ReLU:

- ее производная равна либо единице, либо нулю, и поэтому не может произойти разрастания или затухания градиентов, т.к. умножив единицу на дельту ошибки мы получим дельту ошибки, если же мы бы использовали другую функцию, например, гиперболический тангенс, то дельта ошибки могла, либо уменьшиться, либо возрасти, либо остаться такой же, то есть, производная гиперболического тангенса возвращает число с разным знаком и величиной, что можно сильно повлиять на затухание или разрастание градиента. Более того, использование данной функции приводит к прореживанию весов;
- вычисление сигмоиды и гиперболического тангенса требует выполнения ресурсоемких операций, таких как возведение в степень, в то время как ReLU может быть реализован с помощью простого порогового преобразования матрицы активаций в нулю;
- отсекает ненужные детали в канале при отрицательном выходе.

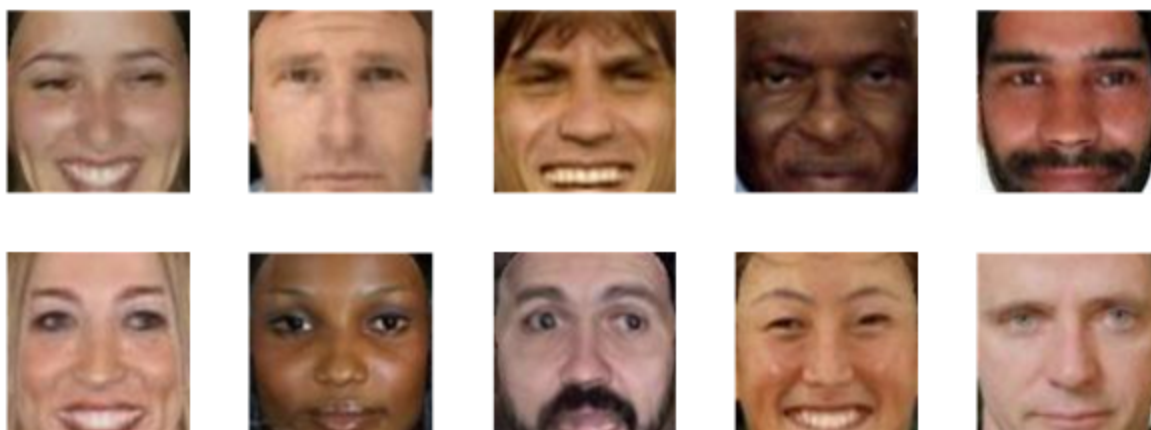
Из недостатков можно отметить, что ReLU не всегда достаточно надежна и в процессе обучения может выходить из строя («умирать»). Например, большой градиент, проходящий через ReLU, может привести к такому обновлению весов, что данный нейрон никогда больше не активируется. Если это произойдет, то, начиная с данного момента, градиент, проходящий через этот нейрон, всегда будет равен нулю. Соответственно, данный нейрон будет необратимо выведен из строя. Например, при слишком большой скорости обучения (learning rate), может оказаться, что до 40% ReLU «мертвы» (то есть, никогда не активируются). Эта проблема решается посредством выбора надлежащей скорости обучения.



## Обучающие выборки использующиеся в экспериментах

Обучающая выборка состоит из положительных и отрицательных примеров. В данном случае из лиц и “не лиц”. Соотношение положительных к отрицательным примерам 4 к 1, 8000 положительных и 2000 отрицательных.

В качестве положительной обучающей выборки использовалась база данных LFW3D [7]. Она содержит цветные изображения фронтальных лиц типа JPEG, размером 90x90 пикселей, в количестве 13000. База данных предоставляется по FTP, доступ осуществляется по паролю. Для получения пароля необходимо заполнить на главной странице сайта простую форму, где указать свое имя и электронную почту. Пример лиц из базы данных показан в соответствии с рисунком ниже:



В качестве отрицательных обучающих примеров использовалась база данных SUN397 [8], она содержит огромное количество всевозможных сцен, которые разбиты по категориям. Всего 130000 изображений, 908 сцен, 313000 объектов сцены. Общий вес этой базы составляет 37 GB. Категории изображений весьма различны и позволяют выбирать более конкретную среду, где будет использоваться конечное ПС. Например, если априори известно, что детектор лиц предназначен только для распознавания внутри помещения, то нет смысла использовать обучающую выборку природы, неба, гор и т.д. По этой причине автором работы были выбраны следующие категории изображений: жилая комната, кабинет, классная комната, компьютерная комната. Примеры изображений из обучающей выборки SUN397 показаны в соответствии с рисунком ниже:



## Результаты

Прямое распространение сигнала от входного изображения размером 90x90 пикселей занимает 20 мс (на ПК), 3000 мс в мобильном приложении. При детектировании лица в видеопотоке в разрешении 640x480 пикселей, возможно детектировать 50 не перекрытых областей размером 90x90 пикселей. Полученные результаты с выбранной топологией сети хуже по сравнению с алгоритмом Виолы-Джонса.

## Выводы

Сверточные нейронные сети обеспечивают частичную устойчивость к изменениям масштаба, смещениям, поворотам, смене ракурса и прочим искажениям.

**Ядро** — представляет из себя фильтр, который скользит по всему изображению и находит признаки лица в любом его месте (инвариантность к смещениям).

**Подвыборочный слой** дает:

- увеличение скорости вычислений (минимум в 2 раза), за счет уменьшения размерности карт предыдущего слоя;
- фильтрация уже ненужных деталей;
- поиск признаков более высокого уровня (для следующего сверточного слоя).

**Последние слои** – слои обычного многослойного персептрона. Два полносвязных и один выходной. Этот слой отвечает за классификацию, с математической точки зрения моделирует сложную нелинейную функцию, оптимизируя которую улучшается качество распознавания. Число нейронов в слое 6 по числу карт признаков подвыборочного слоя.

## Возможные улучшения

- рассмотреть нейросети Fast-RCNN, YOLO;
- распараллеливание процесса обучения на графические процессоры;
- использование Android NDK (C++) для улучшения производительности

Обучение сверточной нейронной сети описано во [второй части](#).

## Ссылки

— [Что такое свёрточная нейронная сеть](#)

— Обучающие множества:

[Effective Face Frontalization in Unconstrained Images](#) // Effective Face.

[SUN Database](#) // MIT Computer Science and Artificial Intelligence Laboratory

— [Информация по сверточным нейронным сетям](#)

— [О функциях обучения нейросети](#)

— [Виды нейронных сетей](#) (подобная схема классификации нейронных сетей)

— Нейронные сети для начинающих: [раз](#) и [два](#).

**Теги:** [сверточная нейронная сеть](#), [CHC](#), [convolution neuron network](#), [cnn](#), [ядро](#), [subsampling](#), [relu](#), [функции активации](#), [обучающее множество](#)

**Хабы:** [Машинное обучение](#)

### Редакторский дайджест

Присылаем лучшие статьи раз в месяц



**12**

Карма

**0**

Рейтинг

**Иван Голиков** @HybridTech

Пользователь

## Комментарии 7

**DrAndyHunter**

31.01.2018 в 17:43

Вопрос. Почему сразу решили использовать нейронную сеть, а не скажем инструменты машинного обучения, тот же opencv?



0

[Ответить](#)**HybridTech**

31.01.2018 в 17:48

На февраль 2017 в OpenCV отсутствовала сверточная сеть, там реализован обычный многослойный персептрон. Плюс хотелось самому разобраться во всех деталях, особенно разобрать процесс обучения таких сетей.



0

[Ответить](#)**roryorangepants**

31.01.2018 в 20:05

Мне кажется, вы путаете тёплое и мягкое.

Сверточная сеть — это алгоритм машинного обучения (абстрактная методология).

OpenCV — это фреймворк (причем не машинного обучения, а компьютерного зрения), т.е. конкретная реализация.

Мне кажется, если уже говорить о выборе инструментария, то интереснее вопрос «Почему автор выбрал Kotlin?»



+1

[Ответить](#)**masai**

01.02.2018 в 00:56

Я думаю, @DrAndyHunter просто оговорился. Справедливости ради, OpenCV не фреймворк, а библиотека, содержащая среди прочего и алгоритмы машинного обучения.



0

[Ответить](#)**ChePeter**

31.01.2018 в 20:44

«Соотношение положительных к отрицательным примерам 1 к 4, 8000 положительных и 2000

отрицательных.» — чего то так.

0 Ответить




 **kalisto21**  
02.02.2018 в 11:52

@HybridTech рекомендую ознакомиться с [диссертацией](#) Калиновского И.А. «Метод нейросетевого детектирования лиц в видеопотоке сверхвысокого разрешения», в которой рассматривалась аналогичная задача с акцентом на вычислительную эффективность топологии СНС. Полный текст дисера доступен на [сайте ТГУ](#), а проект лежит на [GitHub](#).

0 Ответить



 **HybridTech**  
02.02.2018 в 12:00

Да, я смотрел эту работу и Макаренко: «Алгоритмы и программная система классификации». Хорошие работы.

0 Ответить



Только полноправные пользователи могут оставлять комментарии. [Войдите](#), пожалуйста.

Реклама



**HUAWEI WiFi AX2**  
Поддержка 5 ГГц и Wi-Fi 6 | HarmonyOS Mesh+ | Родительский контроль  
На базе **HarmonyOS**  
consumer.huawei.com/ru. ВайФай АИкс2. Меш+. Фактические характеристики устройств могут отличаться. ООО «Техкомпания Хуавэй» ОГРН 1027739023212, г. Москва, Крылатская ул., 17, корп. 2

## ПОХОЖИЕ ПУБЛИКАЦИИ

29 июня в 18:18

**Наглядно о том, как работает свёрточная нейронная сеть**

+11

13K

114

8 +8

18 сентября 2020 в 11:01

**Машинное обучение. Нейронные сети (часть 3) — Convolutional Network под микроскопом. Изучение АПИ Tensorflow.js**

14 мая 2020 в 23:12

## Сверточная нейронная сеть и ее интеграция в iOS (часть 2)

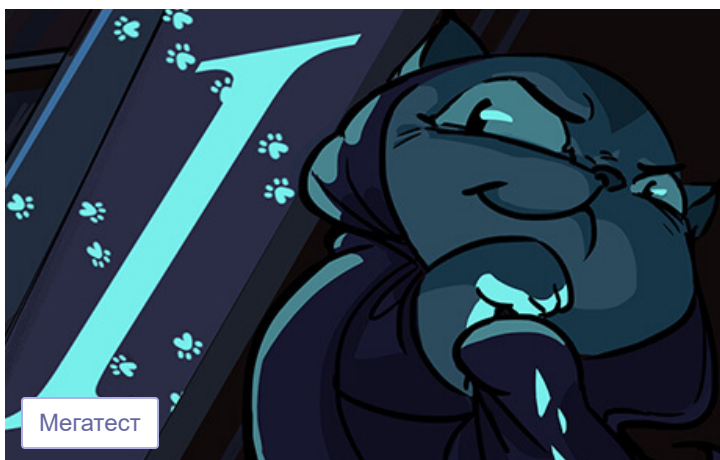
+3

2.1K

10

0

МИНУТОЧКУ ВНИМАНИЯ

[Разместить](#)

Испытание для тех, кто не боится шифров

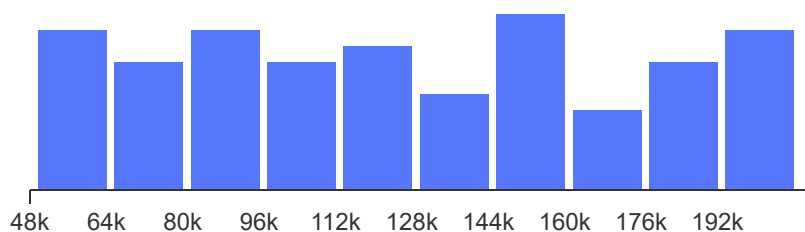


Как разрабы в Т-позе распутывали IT-системы

### СРЕДНЯЯ ЗАРПЛАТА В IT

**119 688** ₽/мес.

— средняя зарплата во всех IT-специализациях по данным из 12 493 анкет, за 2-ое пол. 2021 года. Проверьте «в рынке» ли ваша зарплата или нет!

[Проверить свою зарплату](#)

### ЛУЧШИЕ ПУБЛИКАЦИИ ЗА СУТКИ

вчера в 15:34

## О форме Земли: тыква или дыня? Геодезия и отвага

 +55

 4.5K

 18

 8 +8

вчера в 14:02

### Доказательная медицина в оториноларингологии за последние 10 лет (мифы и что вы не знали про это)

 +46

 8.7K

 71

 40 +40

вчера в 10:05

### Вторая лунная гонка. Что получают завоеватели?

 +44

 9.6K

 33

 33 +33

вчера в 11:58

### Смерть легионера. Серия «Жертвы Везувия», часть 1

 +43

 5.2K

 9

 9 +9

вчера в 20:57

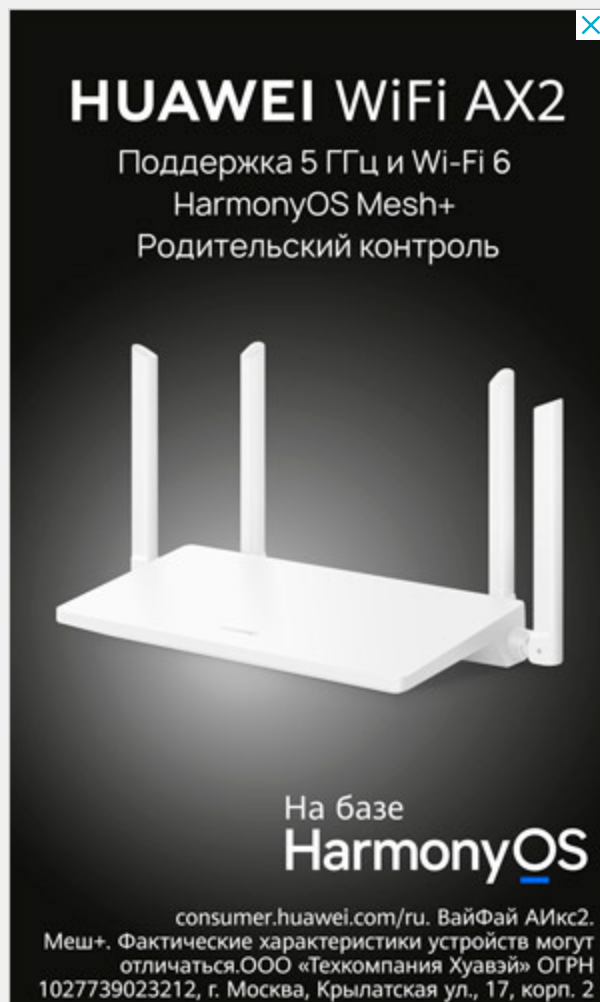
### Металлогалогенные лампы: маленькое домашнее солнце

 +39

 6.4K

 20

 44 +44



ЧИТАЮТ СЕЙЧАС

«Сбер» оказался недоволен производительностью российских серверов на базе процессоров «Эльбрус-8С»

31K 118 +118

Непереводимые слова: 7 русских лексем, которых не хватает в английском

15K 54 +54

Авторы коллективного иска против Apple утверждают, что Apple Watch опасно носить

5.6K 5 +5

Кризис видеокарт — что делать и кто виноват

5.7K 32 +32

Госкорпорация Ростех показала беспилотник С-70 «Охотник» с плоским реактивным соплом

5.8K 14 +14



Турбо

РАБОТА

Data Scientist  
144 вакансии

Все вакансии

Ваш аккаунт

Войти  
Регистрация

Разделы

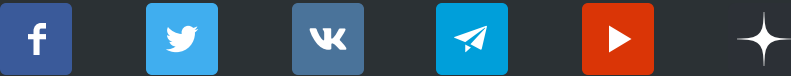
Публикации  
Новости  
Хабы  
Компании  
Авторы  
Песочница

Информация

Устройство сайта  
Для авторов  
Для компаний  
Документы  
Соглашение  
Конфиденциальность

Услуги

Реклама  
Тарифы  
Контент  
Семинары  
Мегапроекты



Настройка языка

О сайте

Техническая поддержка

Вернуться на старую версию