The background of the slide is a light gray gradient, decorated with numerous realistic water droplets of various sizes. Some droplets are large and prominent, while others are small and subtle, scattered across the top and right sides of the frame.

СРАВНЕНИЕ ИСПОЛЬЗОВАНИЯ ПОЛНОСВЯЗНЫХ И СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ ИЗ НАБОРА CIFAR-10

ГАПОНОВ ДМИТРИЙ

207 ГРУППА

CIFAR-10

НАБОР CIFAR-10 СОСТОИТ ИЗ 6000
ИЗОБРАЖЕНИЙ 10 КАТЕГОРИЙ.

РАЗМЕРНОСТЬ ИЗОБРАЖЕНИЯ ИМЕЮТ 32X32
ПИКСЕЛЯ.

ИЗОБРАЖЕНИЯ ЦВЕТНЫЕ, ПОЭТОМУ КАЖДЫЙ
ПИКСЕЛЬ СОСТОИТ ИЗ 3 ЗНАЧЕНИЙ RGB.

ИТОГОВАЯ РАЗМЕРНОСТЬ: 32X32X3 (3072).

ИЗНАЧАЛЬНО НАБОР РАЗДЕЛЕН НА 5000
ИЗОБРАЖЕНИЙ С МЕТКАМИ ДЛЯ
ТРЕНИРОВКИ, И 1000 ДЛЯ ТЕСТИРОВАНИЯ.

В СЛУЧАЕ ВЕКТОРНОГО (А НЕ МАТРИЧНОГО)
ВХОДА БУДЕМ ИСПОЛЬЗОВАТЬ ВЕКТОР

$R\ G\ B\ R\ G\ B\ \dots\ R\ G\ B\} \times 1024$

airplane

automobile

bird

cat

deer

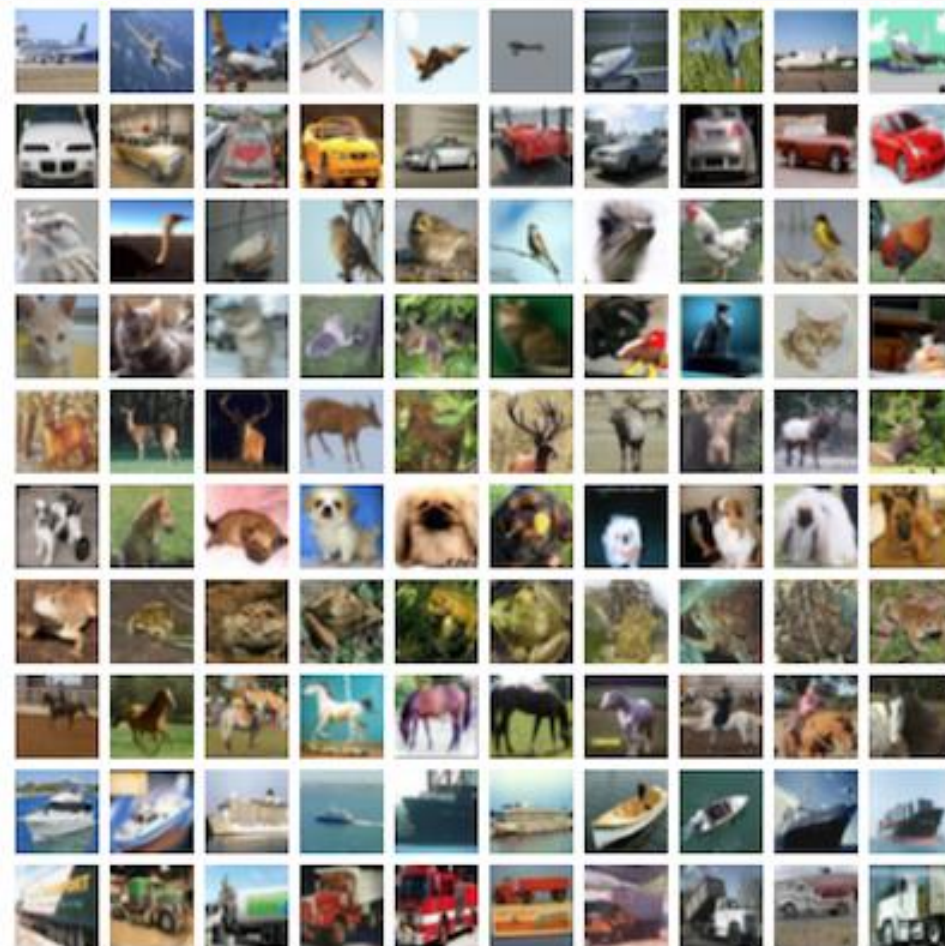
dog

frog

horse

ship

truck



НЕМНОГО О ФИЗИЧЕСКИХ ВОЗМОЖНОСТЯХ

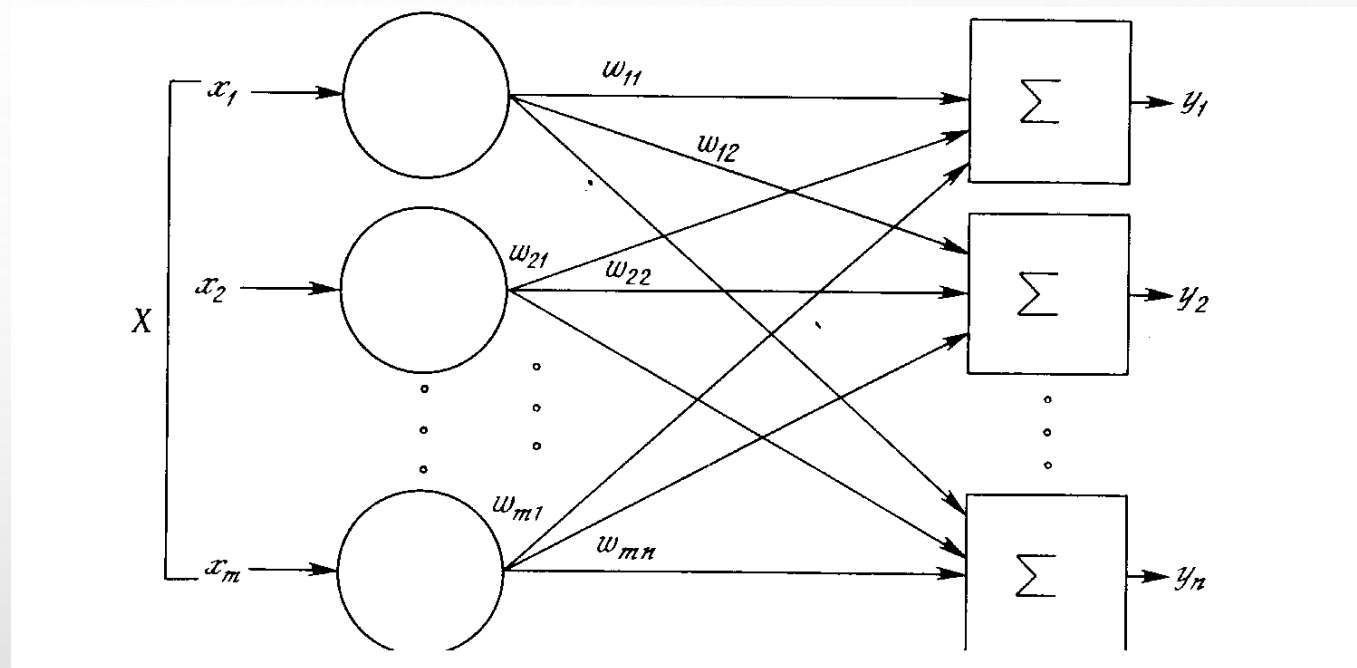
ДЛЯ ОБУЧЕНИЙ ИСПОЛЬЗУЕТСЯ ЯЗЫК PYTHON С
ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ KERAS С БЕКЭНДОМ
НА THEANO.

ВЫЧИСЛЕНИЯ ПРОВОДЯТСЯ НА ГРАФИЧЕСКОМ
УСКОРИТЕЛЕ NVIDIA GEFORCE 940MX.

ПЕРСЕПТРОН

ДЛЯ НАЧАЛА ПОПРОБУЕМ
ИСПОЛЬЗОВАТЬ
ПОЛНОСВЯЗНУЮ СЕТЬ БЕЗ
СКРЫТЫХ СЛОЕВ –
ПЕРСЕПТРОН*.

ПЕРЕЧИСЛИМ НЕКОТОРЫЕ
ОСОБЕННОСТИ СЕТИ И
ОБУЧЕНИЯ:



*Исторически считается что персептрон (Розенблаттома) имеет один скрытый слой, но если принять, что входной слой не обучается, то получится современная трактовка. Будем использовать двухслойный персептрон чтобы посмотреть результат работы «наипростейшей» сети.

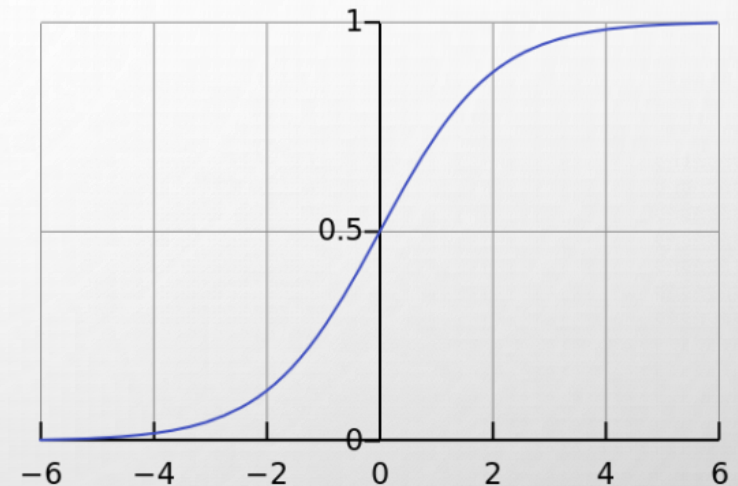
ПЕРСЕПТРОН И ОБУЧЕНИЕ

- ФУНКЦИЯ АКТИВАЦИИ – СИГМОИД

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

- ФУНКЦИЯ ПОТЕРИ –
СРЕДНЕКВАДРАТИЧНОЕ ОТКЛОНЕНИЕ

$$E_{total} = \sum \frac{1}{2} (target - output)^2$$



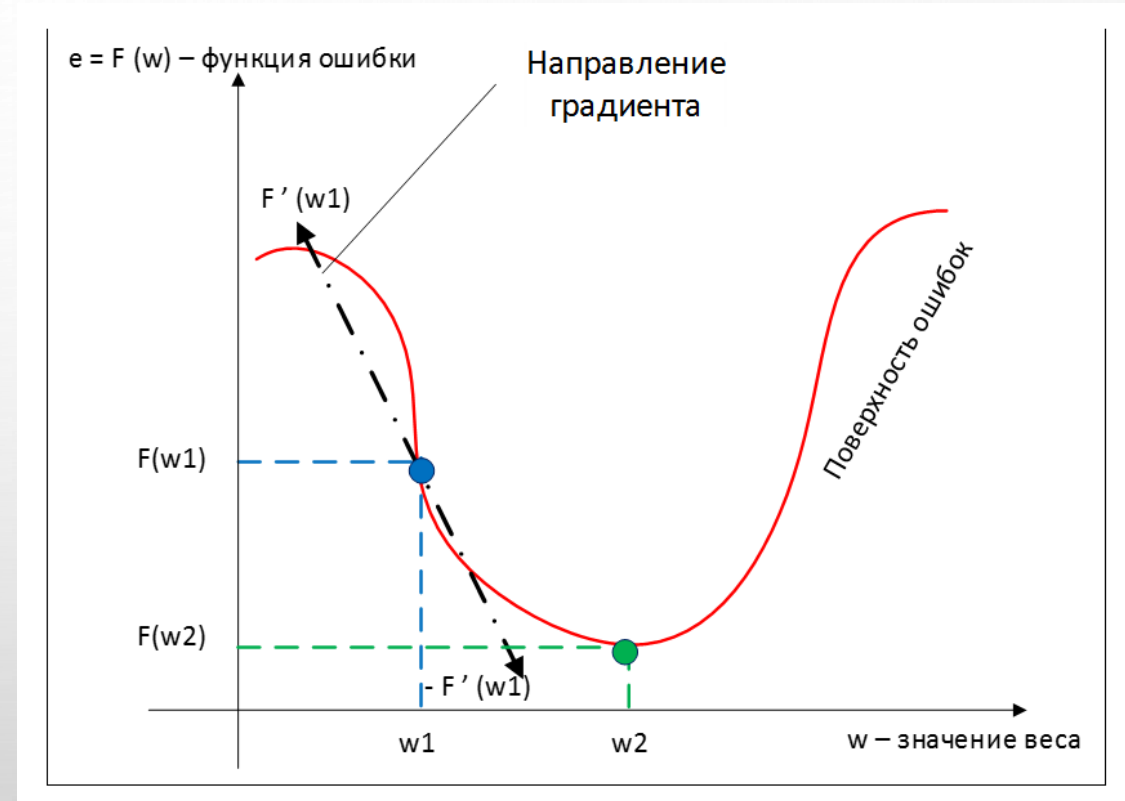
ПАРАМЕТРЫ ОБУЧЕНИЯ

- ЕРОСН – ОДИН ПРОГОН ТРЕНИРОВОЧНОГО МНОЖЕСТВА. ДЛЯ ОБУЧЕНИЯ УКАЗЫВАЕТСЯ КОЛ-ВО ЭПОХ.
- BATCH SIZE – ПАРАМЕТР ОБУЧЕНИЯ, УКАЗЫВАЮЩИЙ РАЗМЕР СЕРИИ ТРЕНИРОВОЧНОГО МНОЖЕСТВА, ПОСЛЕ КОТОРОЙ ПРОИЗВОДИТСЯ КОРРЕКЦИЯ ВЕСОВ.
- VALIDATION SET – ЧАСТЬ ТРЕНИРОВОЧНОГО МНОЖЕСТВА, С ПОМОЩЬЮ КОТОРОЙ ОТСЛЕЖИВАЕТСЯ РЕЗУЛЬТАТ ОБУЧЕНИЯ В САМОМ ПРОЦЕССЕ ОБУЧЕНИЯ. РЕЗУЛЬТАТ РАССЧИТЫВАЕТСЯ ПОСЛЕ КАЖДОЙ ЭПОХИ ОБУЧЕНИЯ.
- НОРМИРОВАНИЕ – ПРИВЕДЕНИЕ ВХОДНЫХ ЗНАЧЕНИЙ К ДИАПАЗОНУ (0; 1)

ПАРАМЕТРЫ ОБУЧЕНИЯ

- ОПТИМИЗАТОР – АЛГОРИТМ ОБНОВЛЕНИЯ ВЕСОВ.
- В ОСНОВНОМ ИСПОЛЬЗУЕТСЯ МЕТОД ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ.

ОН ОСНОВАН НА ТОМ, ЧТО МОЖНО ПРИЙТИ К ЛОКАЛЬНОМУ МИНИМУМУ, ЕСЛИ ДВИГАТЬСЯ В НАПРАВЛЕНИИ АНТИГРАДИЕНТА. ДЛЯ ЭТОГО ДЛЯ КАЖДОГО НЕЙРОНА В ОБРАТНОМ ПОРЯДКЕ СЛЕДОВАНИЯ СЛОЕВ РАССЧИТЫВАЕТСЯ ОТКЛОНЕНИЕ И ВЫЧИСЛЯЕТСЯ НАПРАВЛЕНИЕ ГРАДИЕНТА.



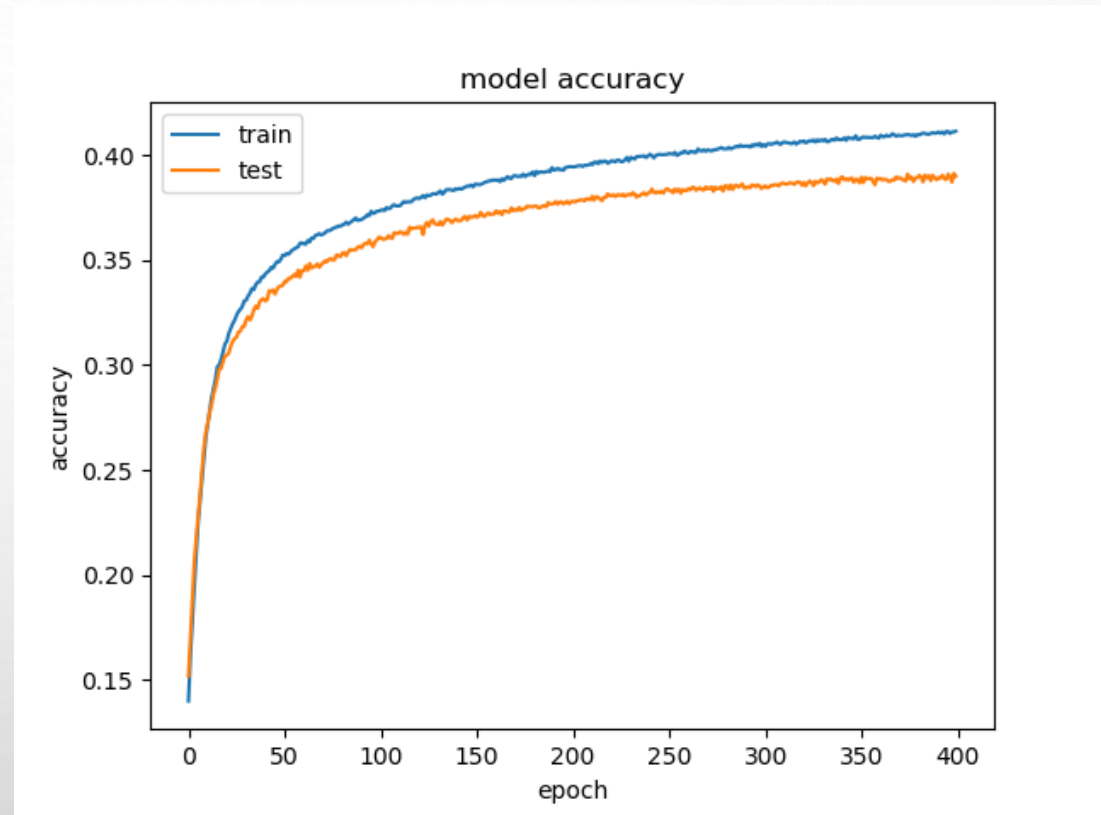
СТРУКТУРА СЕТИ PN

- NB_EPOCH = 400
- BATCH_SIZE = 128
- OPTIMIZER = SGD
(СТАХОСТИЧЕСКИЙ ГРАДИЕНТНЫЙ МЕТОД)
- VALIDATION_SPLIT = 0.2
- НОРМАЛИЗАЦИЯ /= 250
- ВХОДНОЙ СЛОЙ – 3072 НЕЙРОНА
- ВЫХОДНОЙ СЛОЙ – 10 НЕЙРОНОВ
(ПО КОЛ-ВУ КАТЕГОРИЙ)
- КОЛ-ВО ПАРАМЕТРОВ ОБУЧЕНИЯ - 30730

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 10)	30730
activation_1 (Activation)	(None, 10)	0
Total params: 30,730		
Trainable params: 30,730		
Non-trainable params: 0		

ОБУЧЕНИЕ РН

- 2S ЭПОХА (13.5 МИНУТ)
- 48 US/STEP
- TRAIN_ACC: 0,4115
- VAL_ACC: 0,3900
- TEST_ACC: 3,981



ВЫВОДЫ ОБУЧЕНИЯ РН

КАК ВИДНО, ГРАФИКИ ТОЧНОСТИ ПРАКТИЧЕСКИ ВЫПРЯМИЛИСЬ, Т.Е. ПРИ ДАЛЬНЕЙШЕМ ОБУЧЕНИИ МЫ ПРАКТИЧЕСКИ НЕ СМОЖЕМ УВЕЛИЧИТЬ ТОЧНОСТЬ (УПИРАЕМСЯ В АСИМПТОТУ).

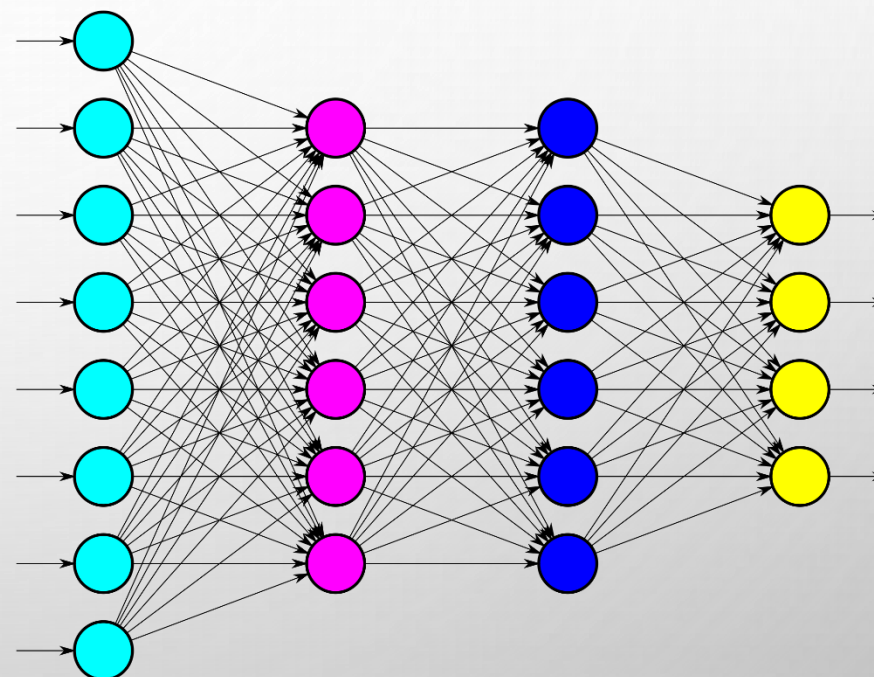
ТОЧНОСТЬ В 40% НАМ КОНЕЧНО ЖЕ АБСОЛЮТНО НЕ ГОДИТСЯ.

СДЕЛАЕМ ВЫВОД, ЧТО АРХИТЕКТУРА СЕТИ СЛИШКОМ ПРОСТА ДЛЯ НАШЕЙ ЗАДАЧИ. ПОПРОБУЕМ УСЛОЖНИТЬ СЕТЬ, ЧТОБЫ ОНА СМОГЛА НАХОДИТЬ БОЛЕЕ СЛОЖНЫЕ ЗАВИСИМОСТИ.

РАССМОТРИМ МНОГОСЛОЙНЫЙ ПЕРСЕПТРОН:

МНОГОСЛОЙНЫЙ ПЕРСЕПТРОН

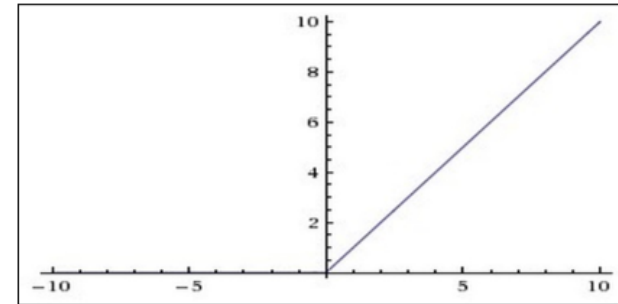
- ПЕРСЕПТРОН С ОДНИМ И БОЛЕЕ СКРЫТЫМ СЛОЕМ.
- СПОСОБЕН НАХОДИТЬ БОЛЕЕ СЛОЖНЫЕ СВЯЗИ МЕЖДУ ПАРАМЕТРАМИ (НЕЛИНЕЙНОСТЬ)
- ОБУЧАЕТСЯ ТАК ЖЕ МЕТОДОМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ



RELU

- ОТНОСИТЕЛЬНО НОВАЯ ФУНКЦИЯ АКТИВАЦИИ
- СИГМОИД ИЗ-ЗА ОГРАНИЧЕННОГО ДИАПАЗОНА ЗНАЧЕНИЙ ВЫЗЫВАЕТ ЗАТУХАНИЯ АЛГОРИТМА РАСПРОСТРАНЕНИЯ ОШИБКИ (ВЕСА ИЗМЕНЯЮТСЯ ВСЕ С МЕНЬШЕЙ СКОРОСТЬЮ ОТ СЛОЯ К СЛОЮ)
- В ВЫХОДНОМ СЛОЕ ТАК ЖЕ ИСПОЛЬЗУЕТСЯ СИГМОИД

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

DROPOUT

- DROPOUT(S) – ФУНКЦИЯ, ОТБРАСЫВАЮЩАЯ ЗНАЧЕНИЯ СЛОЯ С ВЕРОЯТНОСТЬЮ S.
- ТАКОЕ ПРОРЕЖИВАНИЕ ПОБУЖДАЕТ СЕТЬ НЕ «ЗАПОМИНАТЬ» НЕКОТОРЫЕ ПРИЗНАКИ И ОПИРАТЬСЯ ПРИ РАСПОЗНАВАНИИ ТОЛЬКО НА НИХ, А ИСКАТЬ БОЛЕЕ ОБОБЩЕННЫЕ ПРИЗНАКИ; Т.Е. ПОМОГАЕТ ИЗБЕЖАТЬ ПЕРЕОБУЧЕНИЯ.

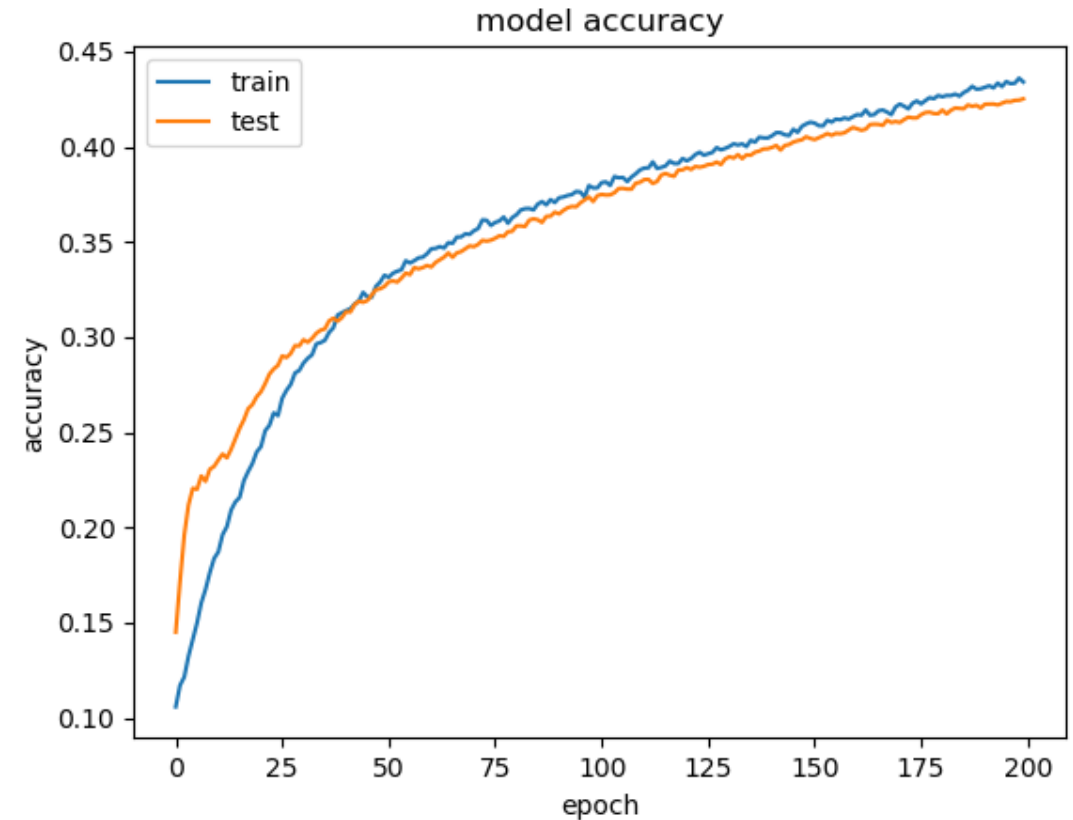
СТРУКТУРА СЕТИ MLPN

- NB_EPOCH = 200
- ОСТАЛЬНОЕ – АНАЛОГИЧНО ПЕРСЕПТРОНУ.
- ВХОД: 3072
- СЛОИ: (5000, 7000, 5000, 3000)
- ВЫХОД: 10
- ДЛЯ КАЖДОГО СКРЫТОГО СЛОЯ:
DROPOUT(0,25), ACTIVATION = 'RELU'
- КОЛ-ВО ПАРАМЕТРОВ: 100'410'010
- 100 МИЛЛИОНОВ!!!

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 5000)	15365000
dropout_1 (Dropout)	(None, 5000)	0
dense_2 (Dense)	(None, 7000)	35007000
dropout_2 (Dropout)	(None, 7000)	0
dense_3 (Dense)	(None, 5000)	35005000
dropout_3 (Dropout)	(None, 5000)	0
dense_4 (Dense)	(None, 3000)	15003000
dropout_4 (Dropout)	(None, 3000)	0
dense_5 (Dense)	(None, 10)	30010
Total params: 100,410,010		
Trainable params: 100,410,010		
Non-trainable params: 0		

ОБУЧЕНИЕ MLPN

- 82S ЭПОХА (275 МИНУТ)
- 2 MS/STEP
- TRAIN_ACC: 0,4340
- VAL_ACC: 0,4251
- TEST_0,4342



ВЫВОДЫ ОБУЧЕНИЯ MLRN

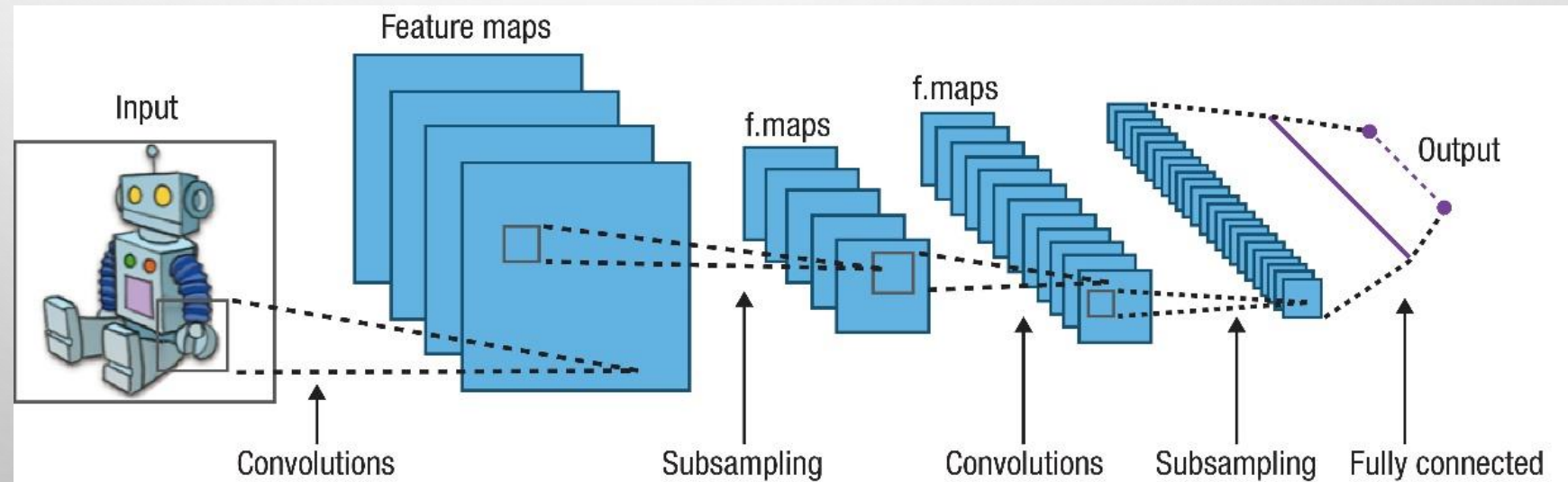
КАК ВИДИМ ТОЧНОСТЬ НА ТЕСТОВОМ МНОЖЕСТВЕ ДАЖЕ ВЫШЕ ТОЧНОСТИ НА ТРЕНИРОВОЧНОМ, ЧТО ОЗНАЧАЕТ ЧТО МЫ ОСТАНОВИЛИ ОБУЧЕНИЕ СЛИШКОМ РАНО. НО СЕТЬ ОБУЧАЕТСЯ ОЧЕНЬ ДОЛГО (4.5 ЧАСА), И, КАК ВИДНО ИЗ ГРАФИКА, ХОТЬ ТОЧНОСТЬ ЕЩЕ УВЕЛИЧИВАЕТСЯ, МОЖНО СЧИТАТЬ, ЧТО ЧЕРЕЗ ЭПОХ 100-200 ГРАФИК СТАНЕТ СТОЛЬКО ЖЕ ПОЛОГИМ КАК И В СЛУЧАЕ С ПЕРСЕПТРОНОМ, А ТОЧНОСТЬ НЕ ПРЕВЫСИТ 50-55%. (ЭТО КОНЕЧНО ЖЕ СТОИТ ПРОВЕРИТЬ, НО В СИЛУ БОЛЬШОГО ВРЕМЕНИ НА ОБУЧЕНИЯ МЫ НЕ БУДЕМ ПРОДЕЛЫВАТЬ, А ПОПЫТАЕМСЯ НАЙТИ БОЛЕЕ БЫСТРЫЙ СПОСОБ ПОЛУЧЕНИЯ РЕЗУЛЬТАТА).

ЕСТЕСТВЕННО ПОЛУЧЕННАЯ ТОЧНОСТЬ В 44% НАС НЕ УСТРАИВАЕТ.

ПОПРОБУЕМ ДОБИТЬСЯ ЛУЧШЕГО РЕЗУЛЬТАТА ИСПОЛЬЗУЯ СВЕРТОЧНЫЕ СЕТИ:

- СВЕРТОЧНЫЕ СЕТИ ИМЕЮТ СЛЕДУЮЩУЮ СТРУКТУРУ ЧЕРЕДОВАНИЯ СЛОЕВ: СВЕРТКИ (С АКТИВАЦИЕЙ), ОПЕРАЦИИ ПУЛИНГА. ВОЗМОЖНО «УГЛУБЛЕНИЕ» СВЕРТКИ ДОБАВЛЕНИЕМ НОВЫХ СЛОЕВ. В КОНЦЕ ВСЕ СВЯЗЫВАЕТСЯ ПОЛНОСВЯЗНЫМ СЛОЕМ (СЛОЯМИ) И ВЫНОСИТСЯ ОКОНЧАТЕЛЬНЫЙ РЕЗУЛЬТАТ.

СВЕРТОЧНЫЕ СЕТИ

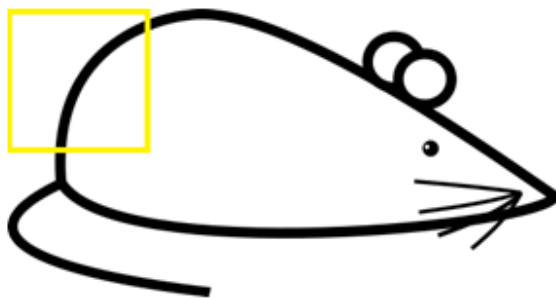


ОПЕРАЦИЯ СВЕРТКИ

- СМЫСЛ ОПЕРАЦИИ ЗАКЛЮЧАЕТСЯ В ТОМ, ЧТО ЯДРО – МАТРИЦА, ОБОЗНАЧАЮЩАЯ КАКОЙ-ЛИБО ПРИЗНАК, НАКЛАДЫВАЕТСЯ НА ИЗОБРАЖЕНИЕ, ПРОИЗВОДИТСЯ СКАЛЯРНОЕ ПЕРЕМНОЖЕНИЕ ЗНАЧЕНИЙ, И НА ВЫХОДЕ МЫ ПОЛУЧАЕМ ЧИСЛО – ХАРАКТЕРИСТИКУ НАХОЖДЕНИЯ ДАННОГО ПРИЗНАКА НА УЧАСТКЕ ИЗОБРАЖЕНИЯ.



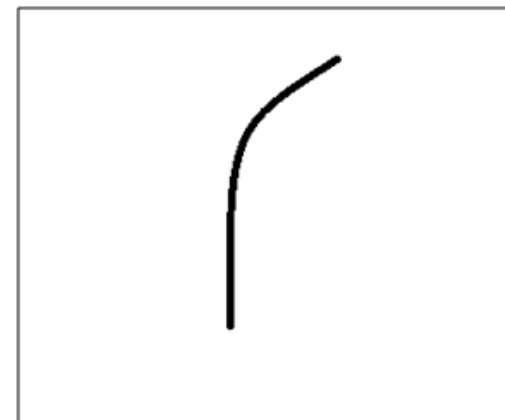
Original image



Visualization of the filter on the image

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

Pixel representation of filter



Visualization of a curve detector filter



Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

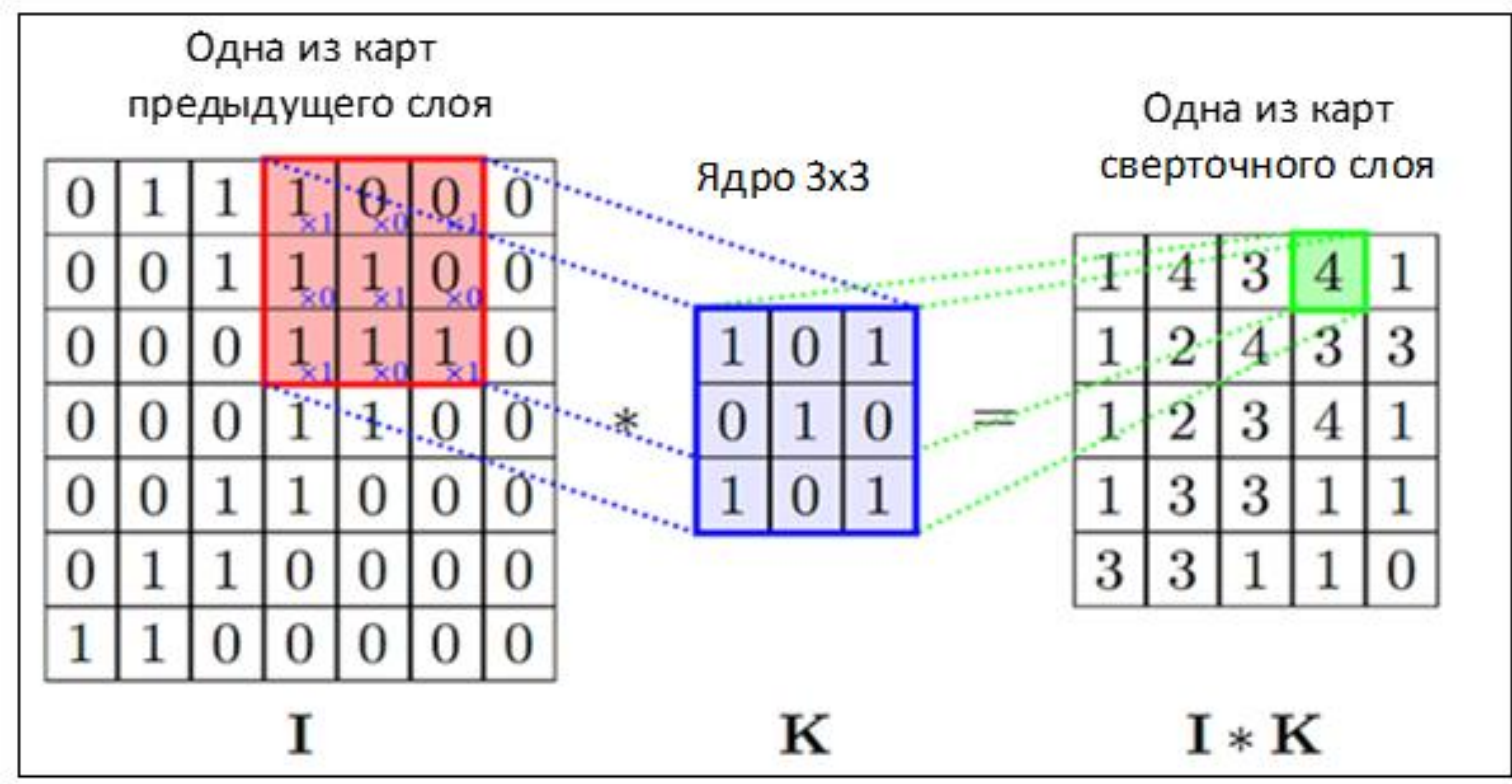
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = $(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$ (A large number!)

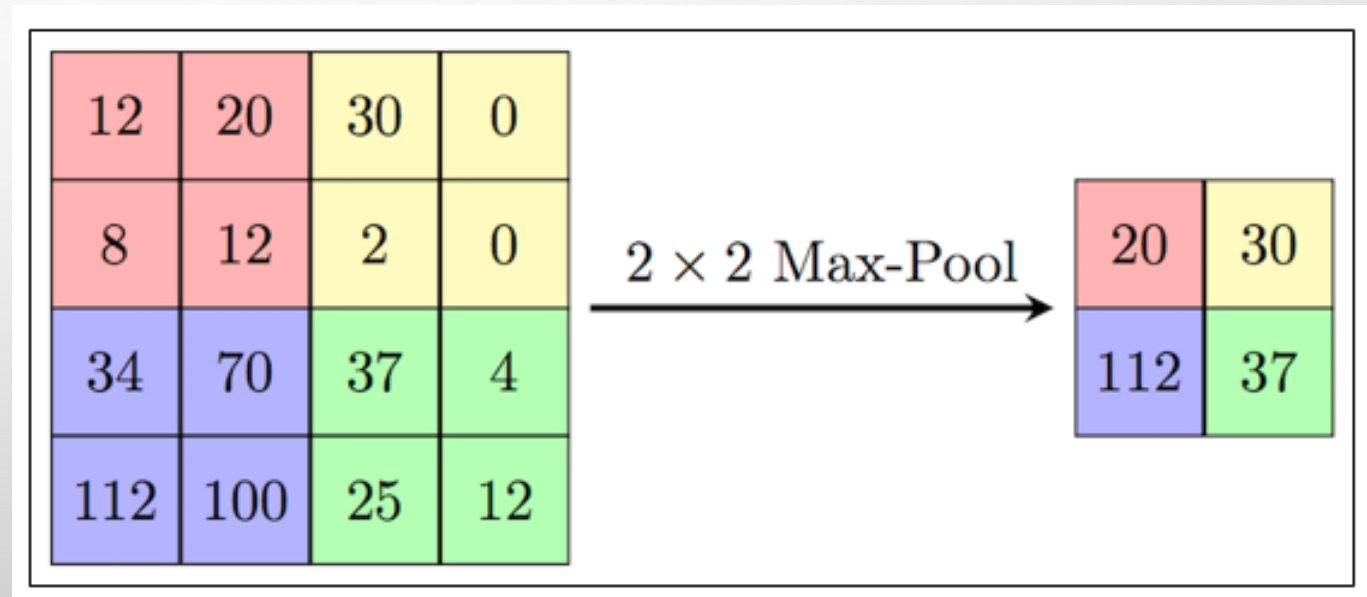
ОПЕРАЦИЯ СВЕРТКИ

- ПРИ ПРОХОЖДЕНИИ ЯДРОМ ПО ВСЕВОЗМОЖНЫМ УЧАСТКАМ ИЗОБРАЖЕНИЯ ПОЛУЧАЕТСЯ КАРТА ДАННОГО ПРИЗНАКА
- ЗА ОДИН СВЕРТОЧНЫЙ СЛОЙ СОЗДАЕТСЯ НЕСКОЛЬКО КАРТ ПРИЗНАКОВ

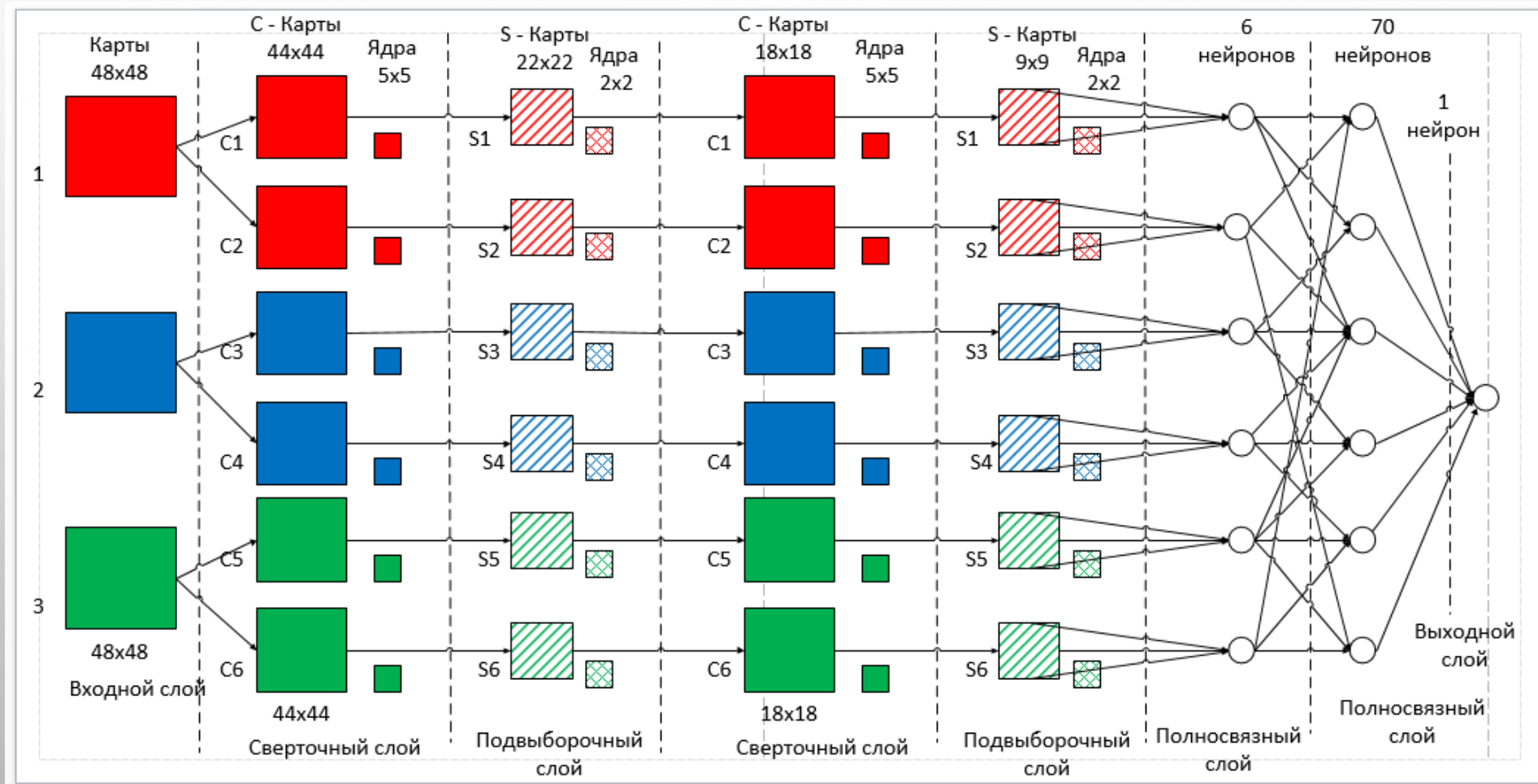


ОПЕРАЦИЯ ПУЛИНГА

- 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.

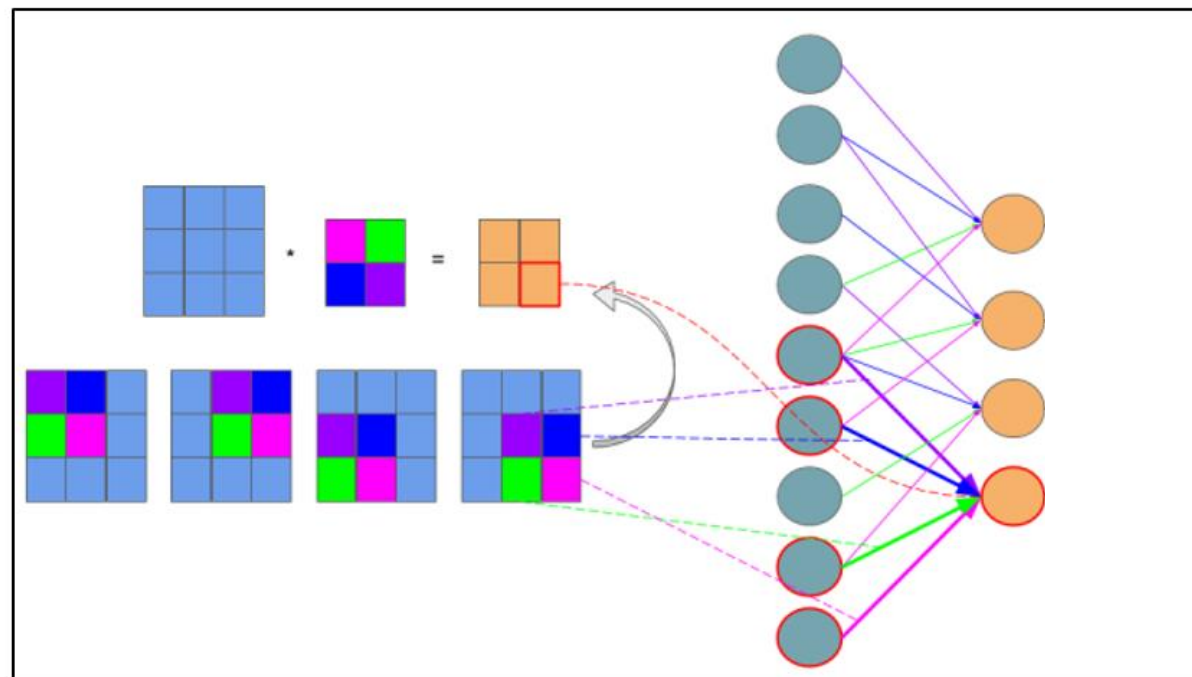


ОБЩАЯ СТРУКТУРА ДЛЯ ЦВЕТНОГО ИЗОБРАЖЕНИЯ



ОБУЧЕНИЕ СВЕРТОЧНОЙ СЕТИ

- МОЖНО УВИДЕТЬ, ЧТО ОПЕРАЦИЯ СВЕРТКИ ПРЕДСТАВИМА В ВИДЕ СКРЫТОГО СЛОЯ КАК В МНОГОСЛОЙНОМ ПЕРСЕПТРОНЕ С ОТЛИЧИЕМ, ЧТО СВЯЗИ ЯВЛЯЮТСЯ РАЗДЕЛЯЕМЫМИ.



ОБУЧЕНИЕ СВЕРТОЧНОЙ СЕТИ

- ТАК ЖЕ ЗАМЕТИМ, ЧТО ПУЛИНГОВОМУ СЛОЮ НЕ ТРЕБУЕТСЯ ВЫЧИСЛЕНИЕ ОШИБКИ. ДЕЛЬТА ОШИБКИ ДЛЯ «ОТСЕЯННЫХ» ЗНАЧЕНИЙ В КАРТЕ СВЕРТКИ БЕРЕТСЯ ЗА 0.
- СЛЕДОВАТЕЛЬНО МЫ МОЖЕМ ОБУЧАТЬ СВЕРТОЧНУЮ НЕЙРОННУЮ СЕТЬ МЕТОДОМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ.
- В НАШЕЙ РАБОТЕ БУДЕМ ИСПОЛЬЗОВАТЬ ОПТИМИЗАТОР RMS_PROP ИЗ «СЕМЕЙСТВА» ADAGRAD, СУТЬ КОТОРОГО В «ХРАНЕНИИ» НЕКОТОРЫХ ПРИЗНАКОВ.

X_{11}	X_{12}	X_{13}
X_{21}	X_{22}	X_{23}
X_{31}	X_{32}	X_{33}

W_{11}	W_{12}
W_{21}	W_{22}

h_{11}	h_{12}
h_{21}	h_{22}

$$h_{11} = W_{11}X_{11} + W_{12}X_{12} + W_{21}X_{21} + W_{22}X_{22}$$

$$h_{12} = W_{11}X_{12} + W_{12}X_{13} + W_{21}X_{22} + W_{22}X_{23}$$

$$h_{21} = W_{11}X_{21} + W_{12}X_{22} + W_{21}X_{31} + W_{22}X_{32}$$

$$h_{22} = W_{11}X_{22} + W_{12}X_{23} + W_{21}X_{32} + W_{22}X_{33}$$

$$\partial h_{ij} \text{ represents } \frac{\partial L}{\partial h_{ij}}$$

$$\partial w_{ij} \text{ represents } \frac{\partial L}{\partial w_{ij}}$$

$$\partial W_{11} = X_{11} \partial h_{11} + X_{12} \partial h_{12} + X_{21} \partial h_{21} + X_{22} \partial h_{22}$$

$$\partial W_{12} = X_{12} \partial h_{11} + X_{13} \partial h_{12} + X_{22} \partial h_{21} + X_{23} \partial h_{22}$$

$$\partial W_{21} = X_{21} \partial h_{11} + X_{22} \partial h_{12} + X_{31} \partial h_{21} + X_{32} \partial h_{22}$$

$$\partial W_{22} = X_{22} \partial h_{11} + X_{23} \partial h_{12} + X_{32} \partial h_{21} + X_{33} \partial h_{22}$$

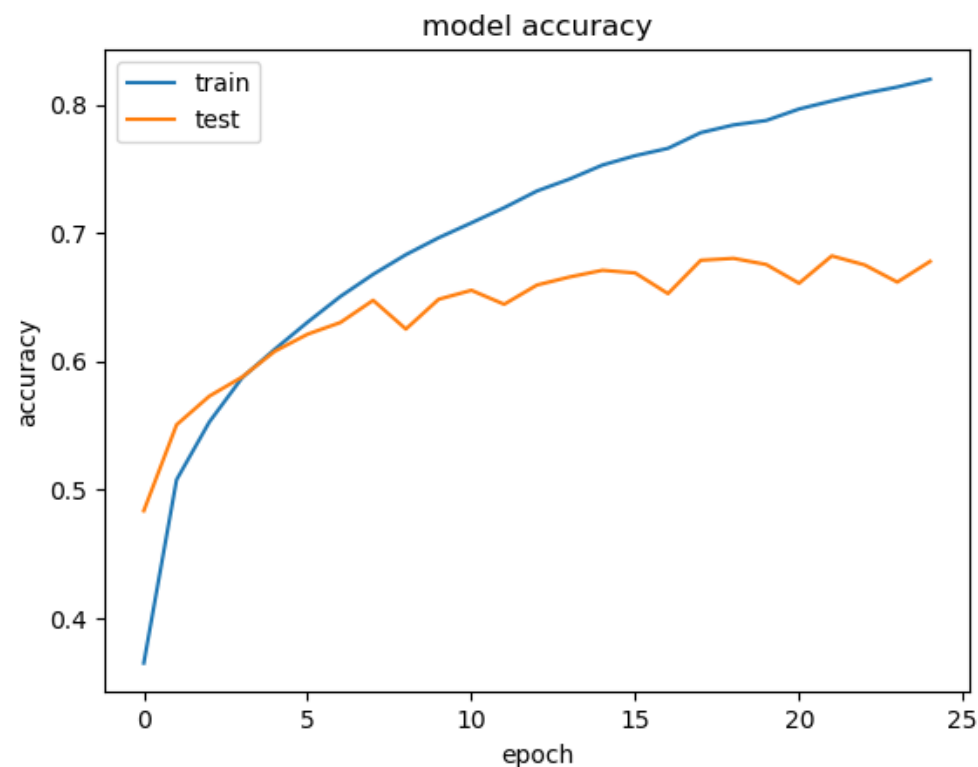
СТРУКТУРА СЕТИ CNN

- NB_EPOCH = 25
- OPTIMIZER = RMS_PROP
- ОСТАЛЬНЫЕ ПАРАМЕТРЫ АНАЛОГИЧНЫ.
- ВХОД: МАТРИЦА ИЗОБРАЖЕНИЯ.
- СЕТЬ: СВЕРТКА + АКТИВАЦИЯ (RELU);
MAX_POOLING; DROPOUT(0,25)
- ПОЛНОСВЯЗНЫЙ СЛОЙ + DROPOUT(0,5)
- ВЫХОД: 10
- КОЛ-ВО ПАРАМЕТРОВ: 3'692'938

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 30, 30, 32)	896
activation_1 (Activation)	(None, 30, 30, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 15, 15, 32)	0
dropout_1 (Dropout)	(None, 15, 15, 32)	0
flatten_1 (Flatten)	(None, 7200)	0
dense_1 (Dense)	(None, 512)	3686912
dropout_2 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130
Total params: 3,692,938		
Trainable params: 3,692,938		
Non-trainable params: 0		

ОБУЧЕНИЕ CNN

- 22S ЭПОХА (10 МИНУТ)
- 554 US/STEP
- TRAIN_ACC: 0,8201
- VAL_ACC: 0,6781
- TEST_ACC: 0,6791



ВЫВОДЫ ОБУЧЕНИЯ CNN

КАК МЫ ВИДИМ, ЗА РЕКОРДНО МАЛОЕ КОЛ-ВО ВРЕМЕНИ ПО СРАВНЕНИЮ С ПЕРСЕПТРОНОМ НА САМОЙ ПРОСТОЙ СВЕРТОЧНОЙ СЕТИ ДОСТИГЛИ РЕЗУЛЬТАТА НА 10% ЛУЧШЕ, ЧЕМ НА ТЯЖЕЛОЙ МНОГОСЛОЙНОЙ СЕТИ – 68%.

ОДНАКО МОЖНО ВИДЕТЬ ОЧЕНЬ БОЛЬШУЮ РАЗНИЦУ МЕЖУ ГРАФИКАМИ ОБУЧАЮЩЕГО И ВАЛИДАЦИОННОГО МНОЖЕСТВА. ЭТО МОЖНО СВЯЗАТЬ С ТЕМ, ЧТО В ПРОЦЕССЕ СВЕРТКИ МЫ ВЫДЕЛЯЕМ ЛИШЬ САМЫЕ ПРИМИТИВНЫЕ ПРИЗНАКИ И ОДИН РАЗ СЖИМАЕМ ПУЛИНГОМ. ТАКИМ ОБРАЗОМ У НАС ПОЛУЧАЕТСЯ СЛИШКОМ МНОГО ПАРАМЕТРОВ В ПОЛНОСВЯЗНОМ СЛОЕ НА СЛИШКОМ МАЛЫЕ И ПРИМИТИВНЫЕ ПРИЗНАКИ, ЧТО ПРИВОДИТ К ПЕРЕОБУЧЕНИЮ.

ПОПРОБУЕМ ОБОБЩИТЬ ПРИМИТИВНЫЕ ПРИЗНАКИ ПО СРЕДСТВОМ УГЛУБЛЕНИЯ СЕТИ, ТЕМ САМЫМ И УМЕНЬШАЯ РАЗМЕРНОСТЬ ПУЛИНГОМ И СООТВЕТСТВЕННО УМЕНЬШАЯ КОЛ-ВО ПАРАМЕТРОВ:

СТРУКТУРА СЕТИ DCNN

- NB_EPOCH = 40
- КОЛ-ВО ПАРАМЕТРОВ: 890'410 – ЭТО БОЛЬШЕ ЧЕМ В ТРИ РАЗА БОЛЬШЕ ПО СРАВНЕНИЮ С НЕГЛУБОКОЙ СЕТЬЮ!!
- МЫ ПЕРИОДИЧЕСКИ ПРОПУСКАЕМ СЛОИ ПУЛИНГА И НЕ НАРАЩИВАЕМ ЧИСЛО КАРТ ПРИЗНАКОВ ДЛЯ ОБОБЩЕНИЯ СУЩЕСТВУЮЩИХ ПРИЗНАКОВ И НЕ СВЕДЕНИЯ РАЗМЕРНОСТИ КАРТ К 0.

```
Conv2D(32, (3, 3))
Activation('relu')
Conv2D(32, (3, 3))
Activation('relu')
MaxPooling2D(pool_size=(2, 2))
Dropout(0.25)

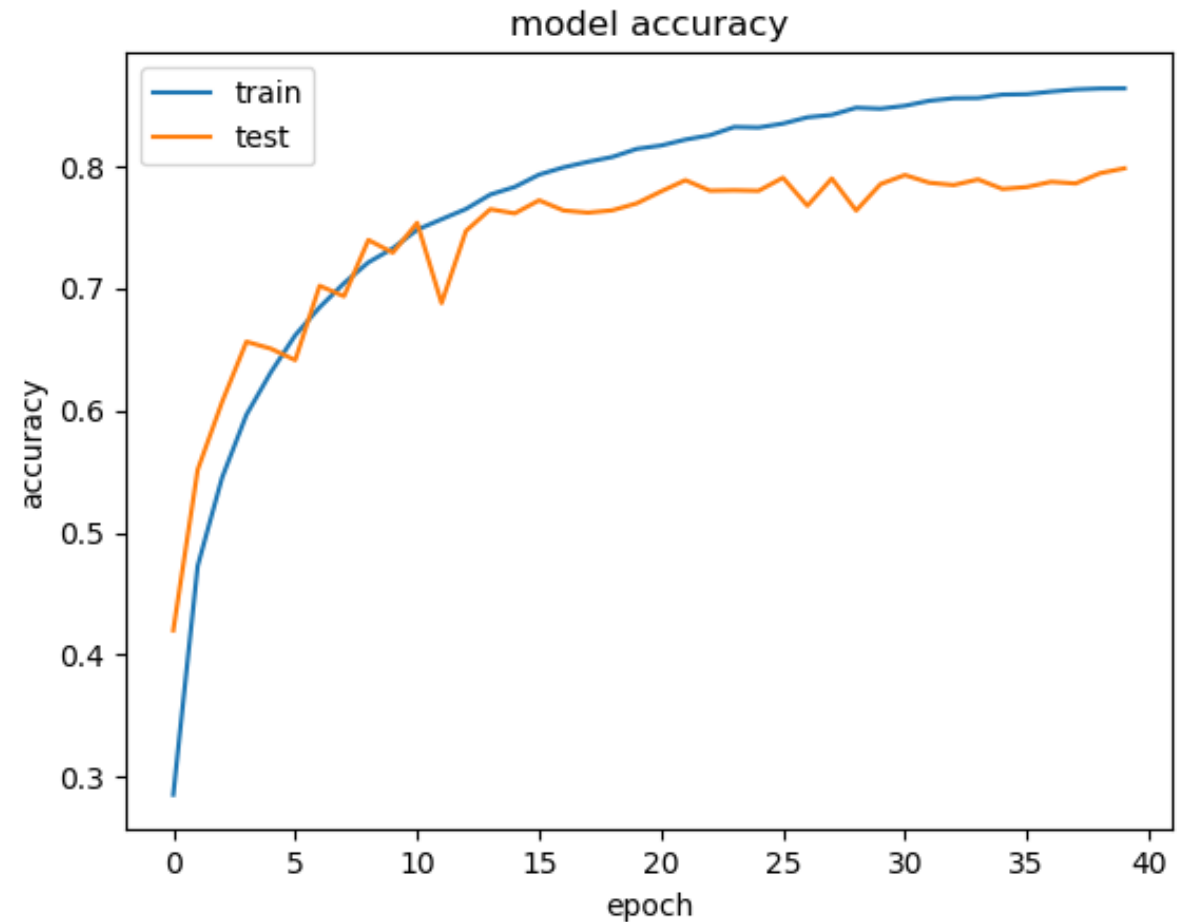
Conv2D(64, (3, 3))
Activation('relu')
Conv2D(64, (3, 3))
Activation('relu')
MaxPooling2D(pool_size=(2, 2))
Dropout(0.25)

Dense(512, activation='relu')
Dropout(0.5)
Dense(10, activation='sigmoid')
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 30, 30, 32)	896
activation_1 (Activation)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 32)	9248
activation_2 (Activation)	(None, 28, 28, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	0
dropout_1 (Dropout)	(None, 14, 14, 32)	0
conv2d_3 (Conv2D)	(None, 12, 12, 64)	18496
activation_3 (Activation)	(None, 12, 12, 64)	0
conv2d_4 (Conv2D)	(None, 10, 10, 64)	36928
activation_4 (Activation)	(None, 10, 10, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 64)	0
dropout_2 (Dropout)	(None, 5, 5, 64)	0
flatten_1 (Flatten)	(None, 1600)	0
dense_1 (Dense)	(None, 512)	819712
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130
Total params: 890,410		
Trainable params: 890,410		
Non-trainable params: 0		

ОБУЧЕНИЕ DCNN

- 47S ЭПОХА (30 МИНУТ)
- 1 MS/STEP
- TRAIN_ACC: 0,8640
- VAL_ACC: 0,7985
- TEST_ACC: 0,7940



ВЫВОДЫ ОБУЧЕНИЯ DCNN

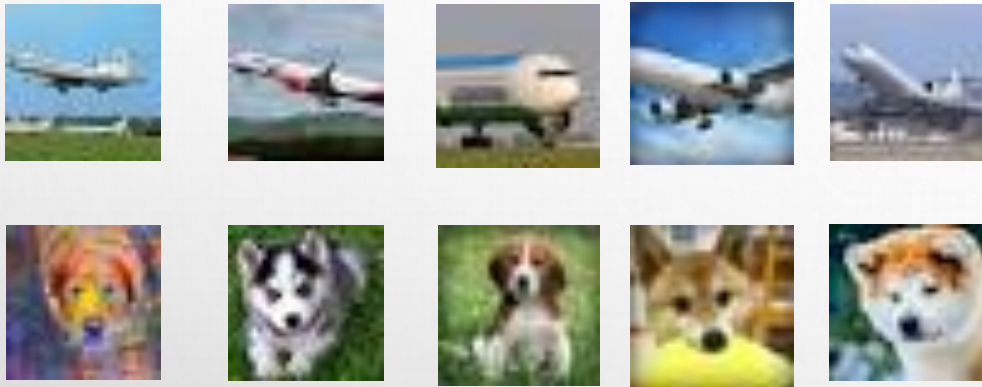
- КАК МЫ ВИДИМ, В УГЛУБЛЕННОМ ВАРИАНТЕ СЕТИ МЫ ПОЛУЧИЛИ НЕ ТАКОЕ БОЛЬШОЕ РАСХОЖДЕНИЕ В ГРАФИКАХ, ЗНАЧИТ НАШЕ ПРЕДПОЛОЖЕНИЕ БЫЛО ВЕРНЫМ.
- МЫ ПОДНЯЛИ ТОЧНОСТЬ НА ОБУЧАЮЩЕМ МНОЖЕСТВЕ НА 4%, А НА ТЕСТОВОМ НА 11%!

ВЫВОДЫ

- ПО ИТОГУ НАШЕГО СРАВНЕНИЯ МЫ МОЖЕМ НАЗВАТЬ ОДНОЗНАЧНОГО ПОБЕДИТЕЛЯ – СВЕРТОЧНЫЕ СЕТИ.
- PN ПРОСТО НЕ СПРАВИЛАСЬ СО СЛОЖНОСТЬЮ ЦВЕТНОГО ИЗОБРАЖЕНИЯ, ТОЧНОСТЬ <40% НАС НЕ УСТРАИВАЕТ.
- MLPN МОЖЕТ БЫТЬ И ДАЛА БЫ РЕЗУЛЬТАТЫ, НО ПРИ БОЛЕЕ ТОНКОЙ СЛОЖНОЙ НАСТРОЙКЕ И ПРИ ОЧЕНЬ ДЛИТЕЛЬНОМ ВРЕМЕНИ ОБУЧЕНИЯ. ДА И СТАВИТЬ НА ТО, ЧТО ТОЧНОСТЬ ВЫЙДЕТ ЗА ПРЕДЕЛЫ 60% ДОСТАТОЧНО СЛОЖНО.
- CNN ПОКАЗАЛА СЕБЯ КАК САМЫЙ ОПТИМАЛЬНЫЙ ПОКАЗАТЕЛЬ СКОРОСТЬ-КАЧЕСТВО. НО МЫ ПРИШЛИ К РАСХОЖДЕНИЮ ЗНАЧЕНИЙ. ОДНАКО ТУ ЖЕ ТОЧНОСТЬ 68% МЫ БЫ МОГЛИ ПОЛУЧИТЬ И ДАЖЕ ЗА ГОРАЗДО МЕНЬШЕЕ ВРЕМЯ! К ЭТОМУ РЕЗУЛЬТАТУ ПОЛНОСВЯЗНЫЕ СЕТИ ДАЖЕ НЕ ПРИБЛИЗИЛИСЬ.
- И САМЫМ ТОЧНЫМ И СТАБИЛЬНЫМ ПОЛУЧИЛАСЬ СЕТЬ DCNN – ВЫДАЛА ЛУЧШИЙ РЕЗУЛЬТАТ ЗА ОПТИМАЛЬНО ПРИЕМЛЕМОЕ ВРЕМЯ – ПОЧТИ 80% ЗА 30 МИНУТ. ТАК ЧТО ОНА СТАНОВИТСЯ СЕГОДНЯ ПОБЕДИТЕЛЕМ.

	Time	Train_acc	Test_acc
PN	13,5	0,4115	0,3981
MLPN	275	0,4340	0,4342
CNN	10	0,8201	0,6791
DCNN	30	0,8640	0,7940

ПРОТЕСТИРУЕМ DCNN



```
[0 4 5 0 0 | 1 5 6 5 5]
```

0 – самолеты, 5 - собаки

- DCNN СПРАВИЛОСЬ С 6 ИЗ 10 ИЗОБРАЖЕНИЙ, ЧТО СОСТАВЛЯЕТ 60%.
- ТАК ЧТО УРА, ТЕПЕРЬ МЫ МОЖЕМ РАСПОЗНАВАТЬ СОБАЧЕК И КОШЕЧЕК!!!! 😊