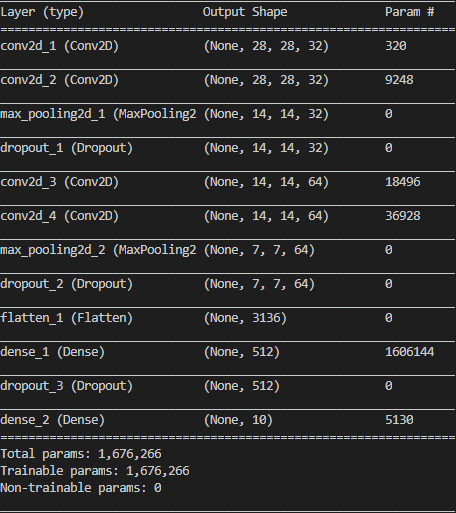
**Задача распознавания цифр 0-9 в различных шрифтах и различных вариациях (применение *italic*, bold и normal).**

Для решения данной задаче я использую сверточную нейронную сеть, имеющую следующую структуру:



Все сверточные слои имеют следующие параметры:

kernel\_size = (3,3)

padding = 'Same'

activation ='relu'

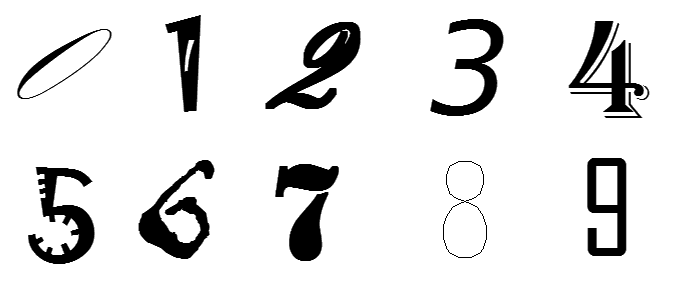
**Набор данных**

Обучение сети происходило на наборе данных The Chars74k,

а именно на подмножестве черно - белых изображений 128x128 EnglishFnt, содержащем 10000 изображений чисел от 0 до 9. Общее число файлов тренировочной части - 7000, контрольной - 1500, проверочной - 1500.

Данные приведены к размеру 28x28, нормализованы и переведены из RGB в grayscale.

Примеры изображений:



**Реализация**

Нейронная сеть реализована на языке Python с использованием библиотек Keras и Matplotlib.

При выборе структуры сети я сравнивал 4 модели при прочих равных значениях (batch\_size = 100, epochs = 15, optimizer = ‘adadelta’):

* Conv2D(32, (3,3)) - Conv2D(64, (3,3)) - MaxPooling(2,2) - Dense(128) - Dense(10)

Число параметров 1,199,882

avg accuracy - 97.80%

* Conv2D(32, (5,5)) - Conv2D(32, (5,5)) - MaxPooling(2,2) - Conv2D(64, (3,3)) - Conv2D(64, (3,3)) - MaxPooling(2,2) - Dense(256) - Dense(10)

Число параметров 887,530

avg accuracy - 98.24%

* Conv2D(32, (3,3)) - Conv2D(32, (3,3)) - MaxPooling(2,2) - Conv2D(64, (3,3)) - Conv2D(64, (3,3)) - MaxPooling(2,2) - Dense(512) - Dense(10)

Число параметров 1,676,266

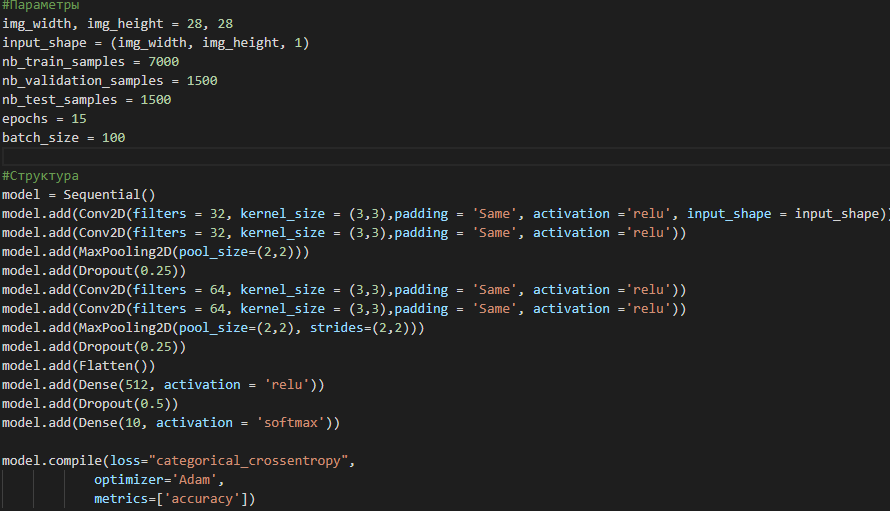
avg accuracy - 98.35%

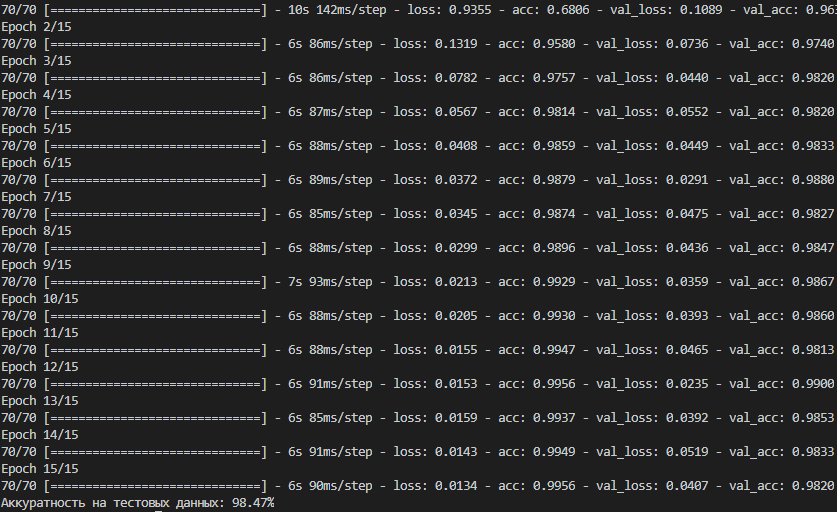
* Conv2D(20, (5,5)) - MaxPooling(2,2) - Conv2D(20, (5,5)) - MaxPooling(2,2) - Dense(128) - Dense(10)

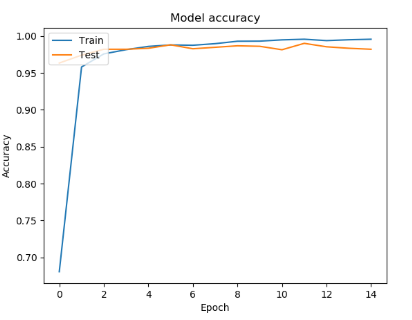
Число параметров 109,650

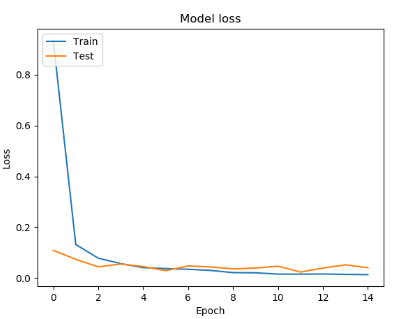
avg accuracy - 98.01%

Тестируя различные параметры, я сравнивал оптимизаторы Adam, Adadelta, RMSprop, Adamax; размеры выборки 50-100; различное количество эпох.

Таким образом финальная версия имела следующий вид:



Графики точности и ошибки:



Источники

* <https://gist.github.com/shravankumar147/9c9cc17e8c7ef7ef46617853377f45f5>
* <https://www.kaggle.com/yassineghouzam/introduction-to-cnn-keras-0-997-top-6>
* <https://github.com/keras-team/keras/blob/master/examples/cifar10_cnn.py>
* <https://github.com/vbystricky/vbystricky_tests/blob/master/mnist_tests/MNIST_test.py>
* <https://vbystricky.github.io/2017/10/mnist_cnn.html>
* <http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/>