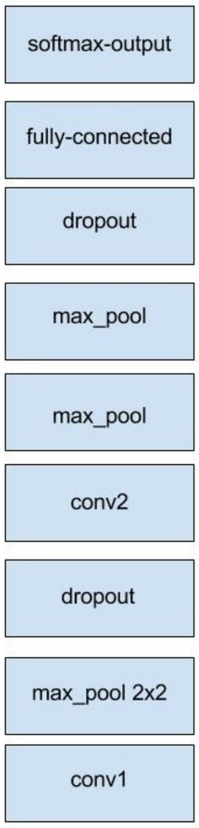
Prediction of Handwritten Characters using Convolutional Neural Network

1. Introduction

The task is to train the Neural Network to correctly classify handwritten characters fed under the form of 16x8 bitmap images. The training set is composed of 41721 examples and the test set on which the predictions are to be made is formed by 10431 inputs.
The images are composed of one feature per pixel (either 0 or 1, with 1→'pixel on') that represent one of the 26 possible lower-case letters of the alphabet.
The idea of using a convolutional neural network is to learn from data recurrent patterns (via training so called “filters”) on the input samples, and once extracted these more refined informations, learn to classify the inputs based on their “projection” after being passed through those filters. The whole process is learnt via data at the same time, and there aren't any pre-set models.

2. Procedure



The architecture of the network is shown on the side.
The convolution had patches 4x4 with padding same, sliding over each possible value. The first convolution extracts 128 features, the second one 256. After each convolutional layer is applied a max-pooling layer with 2x2 patches having a step of two, halving both heigth and witdh of the input. Dropout layers help preventing overfitting, shutting down neurons with a probability of 0.5. The fulli connected layer is connected with the last dropout and adds 1024.

3. Analysis

The kfold validation has been done with k=3, and the accuracies weren't great: 0.7643, 0.7541,0.7696 that represents the same performances obtained when training over the whole input dataset (~0.77).
Many configuration of parameters have been tried, and architectural modifications as

well. It seems hard to tweak parameters in the right way to perform a good predictor.
Changes tried:

- patches: 6x6,5x5,4x4,6x3
- features per convolution: 32,64,128,200,256,400,512,1024
- dropout values: 0.5,0.75,1
- number of fully connected layers: 2,3
- perceptrons on fully connected: 128,256,1024,2048

Basically every configuration has given at best the result of 77% of accuracy.