COMP-598 Mini-Project #3 Handwritten Digits Classification

David Rapoport, Zafarali Ahmed, Pascale Gourdeau
Team Name: TBD

I. Introduction

The MNIST database**??** of handwritten digits is often used as a baseline to compare performance of different classifiers. In this paper, we use a modified version of the MNIST database where the digits have been modified by (1) embossing, (2) rotation, (3) rescaling, and (4) random texturing. This makes the problem much more difficult and we demonstrate poor performance with linear classifiers.

In this work we explore the performance of three classifiers: (1) Logistic Regression, (2) Support Vector Machine and (3) The Feed-forward Neural Network.

¡Something about results¿

We then explore two algorithms, the tried and test Convolution Neural Network **????** and the brand new Spatial Transformer Network **??** to find that ......

II. Related Work

III. Data

The dataset was obtained from the Kaggle Competition Website. It is a modification of the MNIST **??** database. A sampling of some digits can be seen in Fig 1 and it is aparent that this proves to be a difficult task for even humans to distinguish. In particular, we expect the digits 6 and 9 to be almost indistinguishable.

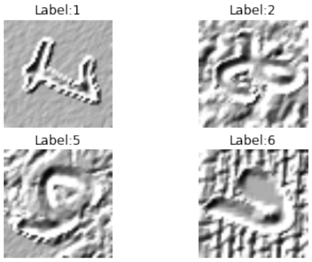

Fig. 1. A sampling of the modified MNIST database

IV. Methodology

We used the NumPy package and the scikit-learn library to perform feature extraction and selection, implement our classifiers, and analyse our results.

A. Feature Extraction and Preprocessing

B. Feature Selection

C. Classification Algorithms

1) Baseline: Logistic Regression: In Logistic Regression, we want to estimate the probability that some random vector $X = (x_1, \ldots, x_n)$ has a class $Y = y_k$, $P(Y = y_k | X = (x_1, \ldots, x_n))$. In the binary case we can derive the following using Bayes rule and conditioning:

$$\begin{aligned} P(Y=1 \mid X) &= \frac{P(X, Y=1)}{P(X)} \\ &= \frac{P(X \mid Y=1)\cdot P(Y=1)}{P(X \mid Y=1)\cdot P(Y=1) + P(X \mid Y=0)\cdot P(Y=0)} \\ &= \frac{1}{1+\exp(-a)} = \sigma(a) \text{ (Sigmoid Function)} \end{aligned}$$

where $a = \ln\left(\frac{P(Y=1 \mid X)}{P(Y=0 \mid X)}\right)$ is the log-odds ratio. By approximating the log-odds ratio as a linear decision boundary of the features and weights, $w^T x$ we can use this as an estimate of the class being $Y = 1$. We can optimize the Log-likelihood or Cross-Entropy function:

$$L(w) = -\sum_{i=1}^{n} y_i \log\left(\sigma(w^T x_i)\right) + (1-y_i)\log\left(1-\sigma(w^T x_i)\right) \tag{1}$$

and search for the optimal set of weights using the *gradient descent algorithm* with N steps and update rule 2:

$$w_{k+1} = w_k + \alpha_k \sum_{i=1}^{n} \left(x_i\left(y_i - \sigma(w_k^T x_i)\right)\right) \tag{2}$$

2) Linear SVM:

3) Fully Connected Feedforward Neural Network:

4) Convolution Neural Network: The convolution neural network **??** is a neural network with specialized layers in which not all neurons are connected to each other. Infact the main component is the subblayer (Fig 2) which contains the Convolution2D and MaxPool2d pair of layers. A convolution of an image is the result where each pixel is the weighted sum of its neighbouring pixels (moving window weighted sum). This means that our 2D convolution takes a weighted sum of each neighbouring pixel of the handwritten digit. *num_filters* is the number of moving windows of size *filter_size* × *filter_size* we do convolutions with. In MaxPooling we select the maximum in a region of size *pool_size* × *pool_size* pixels obtained from the 2D convolution.

As shown in the (simplified) figure, the convolution conducted on the input is transformed into the 3D spatial arrangement of the filters. This is then subsampled (by selecting the MAX element from each subsection of the filter) to produce the set of outputs that can be reused in future layers. In our architecture we stacked K of the Convolution2D-Maxpool sublayers ($K = \{1, 2, 3, 4\}$) and finally ran the outputs through a fully connected dense layer which had the same number of units as *num_filters* in the last Convolution2D-Maxpool subplayer. To obtain the final output, we ran this through a $p = 0.5$ dropout and then into a fully connected dense layer of size 10 to make the predictions. This is summarized in Fig 3.

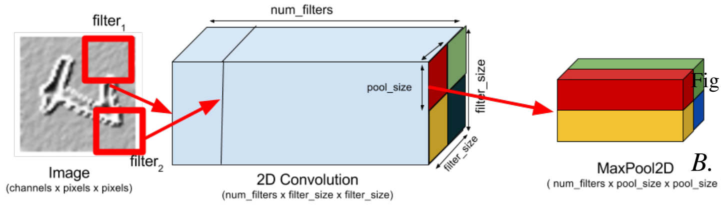
filter1
num_filters
pool_size
filter_size
filter_size
filter2
Image
(channels x pixels x pixels)
2D Convolution
(num_filters x filter_size x filter_size)
MaxPool2D
(num_filters x pool_size x pool_size)

Fig. 2. A simplified example of the sublayer containing Convolution2D layer and a MaxPool2D layer. The variables correspond to the authors implementation of the network. The final network used one,two and three of these sublayers in tandem.

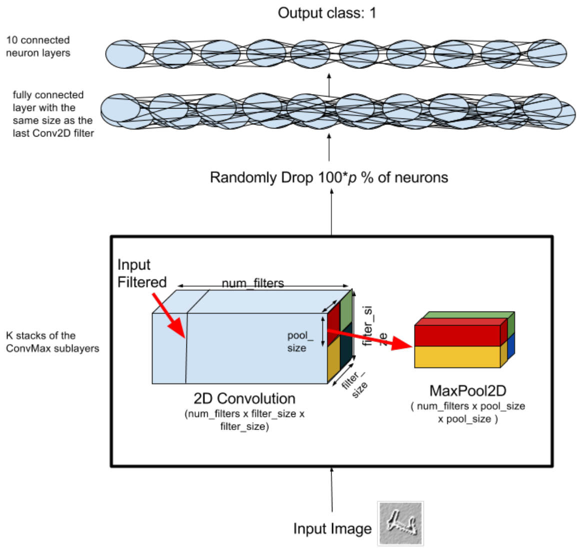
Output class: 1
10 connected neuron layers
fully connected layer with the same size as the last Conv2D filter
Randomly Drop 100*p % of neurons
K stacks of the ConvMax sublayers
Input Filtered
num_filters
pool_size
filter_size
filter_size
2D Convolution
(num_filters x filter_size x filter_size)
MaxPool2D
(num_filters x pool_size x pool_size)
Input Image

Fig. 3. The Convolution Neural Architechture included k stacked sublayers shown in Fig 2 connected to a dropout layer and finally fed into a fully connected layer containing 10 neurons that did the final classification.

5) *Spatial Transformer Network:*

D. *Cross-Validation and Choice of Hyperparameters*

V. Results

A. *Convolution Neural Network*

ROC (?) curves are shown in

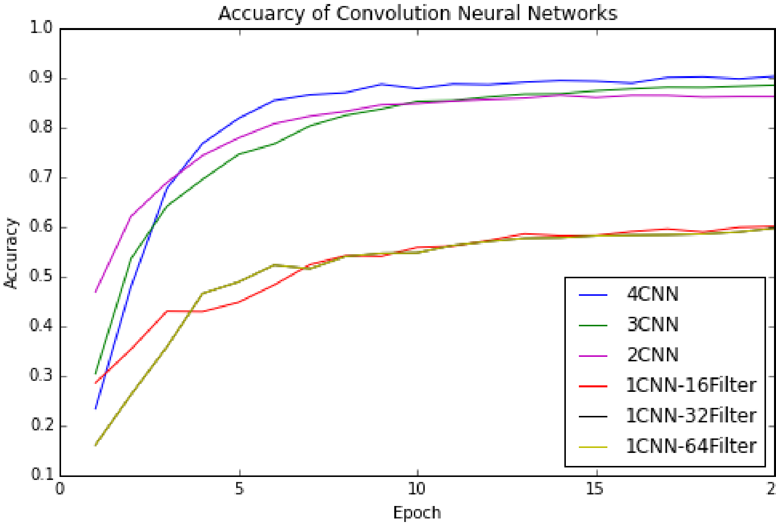

Fig. 4. ROC Curves for our CNN architectures

B. *Spatial Transformer Network*

VI. Discussion

A. *Feature Extraction and Selection*

B. *Classifier Performance*

C. *Future Work*

VII. Statement of Contributions

VIII. Integrity of Work

We hereby state that all the work presented in this report is that of the authors.

References

[1] Lecun Y., Bengio Y., and Haffner P. *Gradient-Based Learning Applied to Document Recognition* Proceedings of the IEEE, 1998.

[2] Sebastini, F. *Machine Learning in Automated Text Categorization* ACM Computing Surveys (CSUR) 34 2002.

[3] Claudiu D. and Meier U. et al. *Convolutional Neural Network Committees For Handwritten Character Classification* ACM Computing Surveys (CSUR) 34 2002.

[4] Jaderberg, M., et al. *Spatial Transformer Networks* ArXiv 2015

[5] LeCun Y., Cortes, C. and Burges C.J.C. *The MNIST Database of handwritten digits* http://yann.lecun.com/exdb/mnist/