Deep Learning

# Mini Project 1

Mammad Hajili **294808**, Reza Hosseini **281368**, Alexandru Mocanu **295172**

May 2019

## 1. Introduction

Image classification is a very common task in the field of Computer Vision. In this project the objective is to see how different architectures behave in comparing two digits represented as 14x14 images. Several techniques can be used to carry out this task and among these, we will study the impact of using weight sharing, through convolutional layers, and auxiliary losses.

## 2. Solutions

We have tried three different approaches to solve this problem, which we will next describe in detail. In each of them, we used fully-connected as well as convolutional architectures for the feature extractor, to see the difference that weight sharing makes.

### 2.1 Direct digit comparison

Our first solution, used as a baseline, is to use architectures that output directly only the target value, 0 or 1, making a direct comparison of the two digits. The network is composed in this case of a feature extracting module and a classification module. The loss we used is a binary cross-entropy loss.
The advantage of this approach is that it is the fastest of the three, as it requires the least amount of computation, given a fixed structure for the feature extractor and the classification module that does the digit comparison. The disadvantage is the lack of interpretability, as from a direct output of the comparison label we can not tell if the network actually understands what each digit in the pair is.

### 2.2 Digit comparison with auxiliary loss

In the second approach, as we also had the labels of the digits in a pair, we decided to use these to enhance the results. We used the same feature extractor as before, kept the classification module doing the digit comparison and we additionally included two classification modules that also take the features from the feature extractor as input to classify each digit in the pair. The digit classifications are done outputting probabilities for each digit class with softmax and selecting the class with the highest probability. The total cost of this task consists of the original cost of digit comparison plus cross-entropy losses for the classification of each digit.
The advantage of this structure is that we now know what the network understands from the input. The disadvantage is the additional amount of computation involved into doing the separate digit classifications.

### 2.3 Digit classification

In our last solution, we decided to see how a simple digit classifier would cope with the task. We therefore split the pairs in separate images and trained several architectures to do digit classification on them. The loss of a pair is equal to the sum of the cross-entropy losses for each image in the pair.
The advantage is that we again have good interpretability and this task is a very common one. The disadvantage is that, as we have to pass each image separately through the network, the time spent on training may be longer than in the previous cases.

## 3. Results

For each of the solutions described above, we have run experiments with sevearal architectures and with SGD and Adam optimizers. For the architecure description, fully-connected layers are represented as the number of units, convolutional layers as NUMCHANNELSxKERNSIZE_STRIDE_PADDING pooling layers as (POOLTYPE)(POOLSIZE). We have run each setup 10 times with 250 epochs per run. For SGD we used a learning rate of 0.01 and for Adam a learning rate of 0.001. For evaluation, we considered the test accuracies, with mean value and standard deviation, expressed in percentages. When possible, we also included auxiliary accuracies for digit classification.
The results are the following:

### 3.1 Direct digit comparison

The architecture is defined for the whole network. We use the following the architectures: **mlp1** - 100; **mlp2** - 100,50; **mlp3** - 100,50,25; **conv1** - 32x5_1_0,100; **conv2** - 32x5_1_0,32x3_1_0,100; **conv3** - 32x5_1_0,64x3_1_0,32x3_1_0,100.

| Architecture | Test accuracy (SGD) | Test accuracy (Adam) |
|---|---|---|
| mlp1 | $77.16 \pm 0.42$ | $79.54 \pm 0.56$ |
| mlp2 | $79.48 \pm 0.53$ | $\mathbf{83.73} \pm 0.47$ |
| mlp3 | $79.41 \pm 0.88$ | $81.04 \pm 0.51$ |
| conv1 | $79.18 \pm 0.62$ | $80.36 \pm 0.56$ |
| conv2 | $\mathbf{83.82} \pm 0.79$ | $80.93 \pm 0.38$ |
| conv3 | $80.53 \pm 0.86$ | $83.04 \pm 1.04$ |

Table 1: The average accuracy and standard deviation of models for direct digit comparison

### 3.2 Digit comparison with auxiliary loss

The architecture is defined for the feature extractor, then the target classifier and finally the digit classifiers (which have the same structure, so only one architecture is specified for those). We use the following architectures: **mlp1** - 100; 100; 100; **mlp2** - 100; 100; 120,84; **mlp3** - 100,50; 100; 120,84; **conv1** - 32x5_1_0; 100; 120,84; **conv2** - 32x5_1_0,32x3_1_0; 100; 120,84; **conv3** - 32x5_1_0,64x3_1_0,32x3_1_0; 100; 120,84; **conv4** - 6x3_1_0,maxpool2,16x3_1_0; 100; 120,84.

### 3.3 Digit classification

The architecture is defined as the feature extractor and then the digit classifiers. We use the following architectures: **mlp1** - 100; 100; **mlp2** - 100; 120,84; **mlp3** - 100,50; 120,84; **conv1** - 32x5_1_0; 120,84; **conv2** - 32x5_1_0,32x3_1_0; 120,84; **conv3** - 32x_1_0,64x3_1_0,32x3_1_0; 120,84; **con4** - 6x3_1_0,maxpool2,16x3_1_0; 120,84.

| | SGD | | Adam | |
|---|---|---|---|---|
| Architecture | Test accuracy | Test aux accuracy | Test accuracy | Test aux accuracy |
| mlp1 | $79.52 \pm 0.42$ | $31.46 \pm 1.25$ | $79.44 \pm 0.65$ | $83.58 \pm 0.48$ |
| mlp2 | $80.99 \pm 0.76$ | $18.51 \pm 4.86$ | $80.77 \pm 0.61$ | $81.92 \pm 1.00$ |
| mlp3 | $79.61 \pm 0.57$ | $13.96 \pm 1.86$ | $\mathbf{82.57} \pm 0.69$ | $76.74 \pm 1.88$ |
| conv1 | $79.4 \pm 0.66$ | $\mathbf{73.42} \pm 4.95$ | $79.91 \pm 0.44$ | $87.02 \pm 0.66$ |
| conv2 | $\mathbf{82.36} \pm 0.65$ | $68.14 \pm 3.89$ | $81.73 \pm 1.02$ | $\mathbf{87.7} \pm 0.89$ |
| conv3 | $80.08 \pm 1.29$ | $64.03 \pm 4.37$ | $82.48 \pm 0.71$ | $85.31 \pm 0.86$ |
| conv4 | $78.09 \pm 1.41$ | $20.46 \pm 9.58$ | $81.71 \pm 0.98$ | $83.6 \pm 1.34$ |

Table 2: The accuracy and standard deviation of models for digit comparison with auxiliary loss

| | SGD | | Adam | |
|---|---|---|---|---|
| Architecture | Test accuracy | Test aux accuracy | Test accuracy | Test aux accuracy |
| mlp1 | $69.14 \pm 7.07$ | $38.8 \pm 8.62$ | $94.08 \pm 0.35$ | $90.2 \pm 0.29$ |
| mlp2 | $65.49 \pm 7.0$ | $33.98 \pm 11.03$ | $94.98 \pm 0.86$ | $92.3 \pm 0.93$ |
| mlp3 | $54.77 \pm 2.81$ | $12.84 \pm 2.78$ | $92.77 \pm 0.61$ | $90.43 \pm 0.94$ |
| conv1 | $\mathbf{87.33} \pm 2.19$ | $\mathbf{79.72} \pm 3.56$ | $96.52 \pm 0.44$ | $95.41 \pm 0.44$ |
| conv2 | $84.3 \pm 3.77$ | $70.48 \pm 7.81$ | $96.36 \pm 0.92$ | $94.98 \pm 0.87$ |
| conv3 | $81.59 \pm 3.37$ | $66.38 \pm 7.53$ | $\mathbf{96.65} \pm 0.66$ | $\mathbf{95.5} \pm 0.75$ |
| conv4 | $78.45 \pm 12.46$ | $58.91 \pm 24.88$ | $96.48 \pm 0.26$ | $94.8 \pm 0.21$ |

Table 3: The accuracy and standard deviation of models for digit classification

## 4. Conclusions

In general both direct digit comparison and comparison with auxiliary loss approaches are performing similarly for SGD and Adam optimizers. For all 4 combinations(direct with SGD and Adam, auxiliary loss with SGD and Adam) average test accuracy is around 82-83%. In case of digit classification approach, for SGD optimizer, accuracies of simple architectures are under 70%. However with convolutional layer(s) test accuracy jumps up to 87.33%. The same approach with Adam optimizer shows significant improvement, where test accuracy of simple fully connected network reaches 94.98%, and with using convolution layers the model gives the best average test accuracy among all the models, with 96.65%.