EE559 Mini deep-learning Classification with weight sharing and auxiliary losses [1]

Mariam Hakobyan - Mazen Fouad A-wali Mahdi - Nguyen Minh Nguyet

mariam.hakobyan@epfl.ch, fouad.mazen@epfl.ch, minh.nguyen@epfl.ch

I. Introduction

In this project, we built deep neural networks to compare two digits of a pair of two-channel images. *MNIST* handwritten digits dataset has been processed for the pairs of grayscale images of size 14×14, to predict if the first digit is less or equal to the second one. For training purposes, the binary class was created for each pair of images by comparing the available digits manually. We have described and implemented two different architectures for the binary classification task, where the first one uses only images and binary classes, while the second one uses the digit information of each image in the intermediary training part of the network. The experiments have been done for 10 rounds with randomized weight initialization with 1000 pairs of training and test data, and the results show that the second architecture achieved considerably lower error rate (about 12%) in 2.4s for 10362 parameters.

II. Architectures of ConvNets

A. First Architecture - Simple binary classification network

The first network is a straightforward binary classification one, which only uses the binary classes available for each pair. Figure 1 demonstrates the architecture of the first network.

Convolution of the neurons weights with the input volume is the base thing of the network architecture. Taking the fact into consideration that the inputs are images, the Convolutional Neural Networks with the advantage of representing the input into a layer with **width, height, depth** have been applied to the classification problem. Two 2D Convolutional layers along with ReLU activation function and max-pooling operator have been applied on the $2d$ input channel, with convolving kernels of size 3 (because our images are of size 14×14 small kernel window was preferred). For preventing overfitting, dropout has been applied after the second convolution to ignore the 0.2 fraction of units and corresponding activation. Afterward, three linear fully connected layers with their ReLU functions have been applied on the output of the ConvNet and the output have been rescaled by Softmax function so that the elements of final output Tensor lie in the range $[0, 1]$ and sum to 1.

The network with 6592 parameters has been trained for minimizing the negative log likelihood loss with SGD optimizer. The training and testing accuracy results are shown in the following table. Experiments carried out with this architecture show that we could attain the best test accuracy of 76.42%, with 5.19% standard deviation from the mean. The test accuracy in this case is not sufficiently high (we expect it to be higher than 85 $\%$), but this network has an advantage of its simple architecture. To further boost this level of accuracy, we could further add alternative layers and re-order (or change) the available layers. Also, in order to achieve better classification results by this architecture, we need

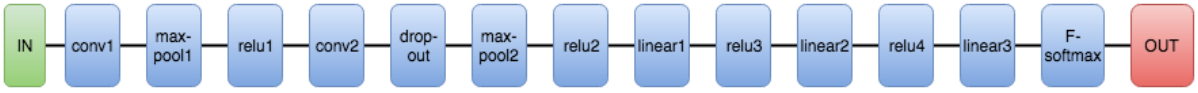


Fig. 1: *The first architecture for binary classification.*

Net1	Mean accuracy	Max accuracy	Median accuracy	Std of Accuracy
Training	80.52%	90.12%	82.34%	8.66
Testing	71.75%	76.42%	73.43%	5.19

TABLE I: Train and Test accuracy of Network 1

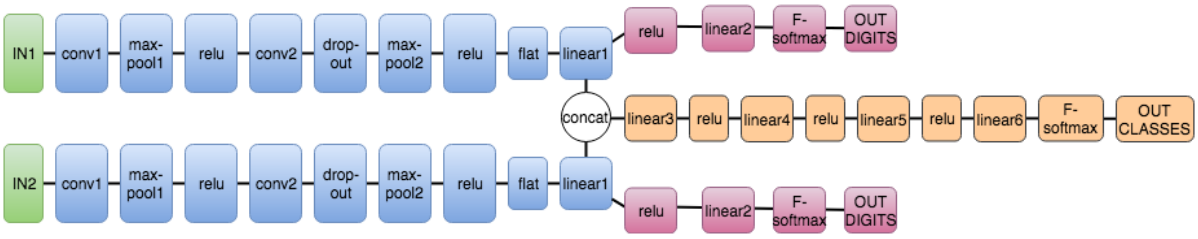


Fig. 2: *The second architecture for binary classification.*

to tune the parameters for the layers, which is tedious and the accuracy is still not high enough. Therefore, the project continues with the second architecture, which takes advantage of not only the Boolean values but also the classes of the two digits in each pair.

B. Second Architecture - Siamese network like architecture

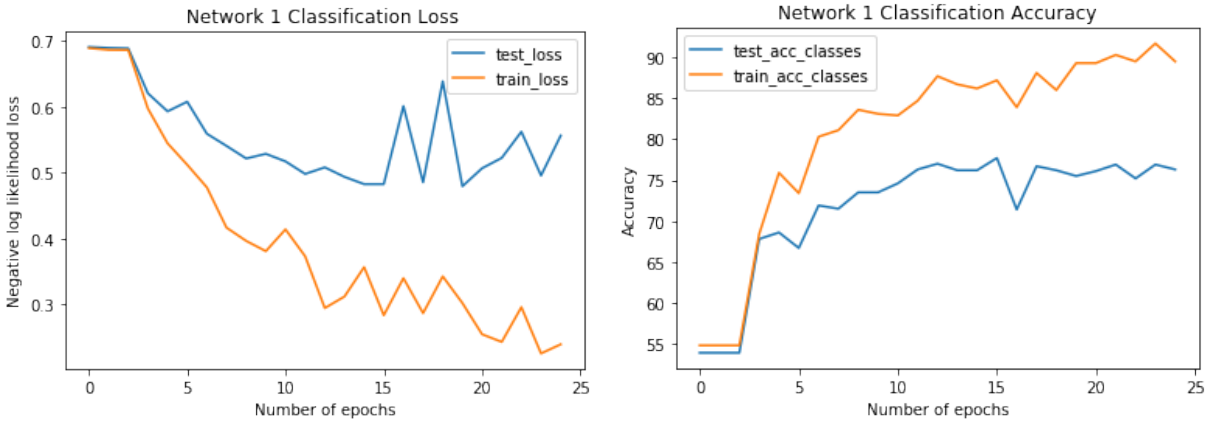
The second network architecture was inspired by the concept of Siamese neural network where two or more similar sub-networks learn to differentiate between two inputs [1], [2]. We used the advantage of the digit classes availability, thus we trained 2 network which has one sub-network of classifying the digit classes and another sub-network that learns the binary classifier based on feature representation vectors of each image. The network worked well because the input for the 2nd sub-network contains information about the digits of input images. The architecture is introduced in Fig 2. The first part is quite similar to the architecture we introduced above for network 1, two conv2d layers with ReLU activation function, along with dropout and max-pooling the later operator for reducing the dimension of the data in the specific layer and the number of parameters. The first sub-network learns digit classification, producing the feature representation vectors for each image to be used in the second network. After getting the linear feature representation of both images, feature vectors have been concatenated to give as an input for the second sub-network for target binary classification. Because the input of the second sub-network is a vector encapsulating the information of the digits of the images, simple linear layers have been applied for this network to learn the parameters of target classification.

Although the network is divided into two sub-networks, the training has been done on one auxiliary loss only which is the sum of two losses from sub-network1 and sub-network2. As the first network, we used Negative log-likelihood loss function to get the losses of the trained network and applied SGD for optimizing the overall loss with learning rate 0.01. The average time needed for training this network is $3.9s$ for 10362 parameters. The training and testing results are shown in the following table.

Net2	Mean accuracy	Max accuracy	Median accuracy	Std of Accuracy
Training	81.63%	90.34%	84.81%	13.73
Testing	77.67%	85.68%	80.29%	13.59

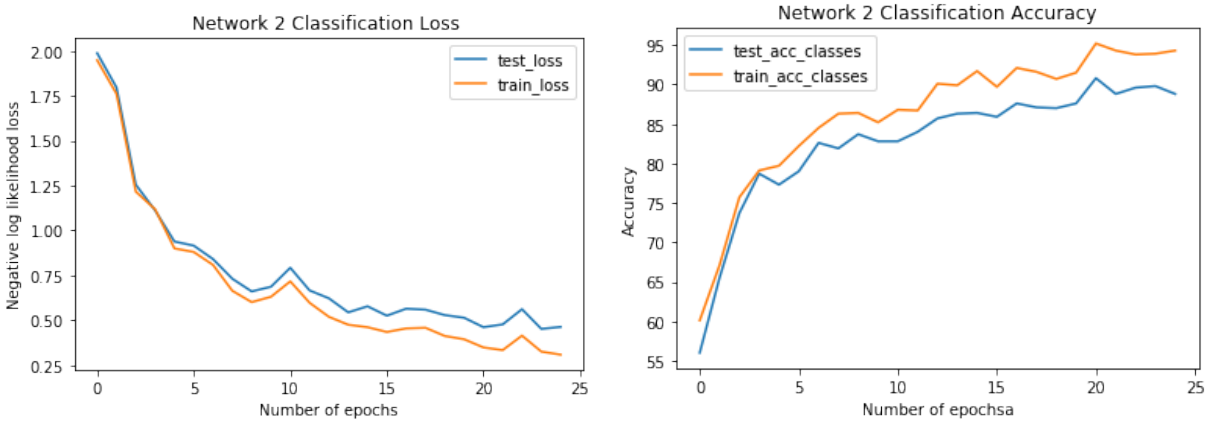
TABLE II: Statistics for Network 2

1000 pairs of train and test data used to train and test network for 10 rounds, each training phase run on the $64 \times 2 \times 14 \times 14$ block from the training data initialized randomly for 25 epochs. We have obtained 85.68% and on median 80.29% accuracy on testing data with standard deviation of 13.59. The loss function of train and test converges more uniformly than for the first network. The results are shown in the Fig 4.



(a) (b)

Fig. 3: Network 1 performance



(a) (b)

Fig. 4: Network 2 performance

III. Conclusions

According to the results we obtained, using the second network, which makes use of both the binary and the digit labels of the images, provides a better outcome than using only the binary labels of the images. This demonstrates the usefulness of auxiliary loss for the training of the network.

References

[1] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[2] Wenpeng Yin, Hinrich Schtze, Bing Xiang, and Bowen Zhou. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272, dec 2016.