Project 1 of Deep Learning:
Classification, weight sharing, auxiliary losses

Yann Bouquet, 273827, yann.bouquet@epfl.ch, Master in Data Science,
Thomas Fry, 280922, thomas.fry@epfl.ch, Master in Computational Science and Engineering,
EPFL

***Abstract*—This project in Deep Learning is about comparing two digits visible in a two-channel image from the MNIST database and formatted with the prologue code from the class. In this report are described the building of different learning architectures and their numerous enhancements to reach a better prediction. Amongst them were implemented weight sharing, auxiliary losses, data augmentation and hyperparameters optimization. Finally an accuracy of 0.9.. was reached.**

I. Introduction

A figure is, by definition **??**, the symbol of a number. Symbol implies language, reading and thus recognition. By the time of five years old, children can recognize digits and letters, small and large, handwritten and machine printed... Humans take this ability for granted, but the learning process of recognition is questioned when they ask the machine to carry out the task for them. Despite a few decades of research, humans remain the best readers of pattern recognizers, and still we are far from describing the successful phenomena behind recognition. Figure recognition made a leap in the 90's with the LeNet architecture, a straightforward neural network for handwritten character recognition.

II. Exploratory data analysis

The MNIST database has handwritten digits images, with a total training set of 60,000 examples and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image. Those images have been formatted with an average pooling and then gathered into pairs, with the prologue code. Thus each image is a 14 $\times$ 14 pixels tensor.

III. State of the art

The LeNet5 from Yann LeCun is a neural network with two sets of convolutional and average pooling layers, followed by a flattening convolutional layer, then two fully-connected layers and finally a softmax classifier, with the mean squared error (MSE) as criterion. All experiments and performances described in this paper are estimated on a test set of 1000 pairs of image, for architecture trained on also 1000 image pairs, with stochastic gradient descent as optimizer, through ten simulations where both data and weight initialization are randomized. LeNet5 was built for 28 $\times$ 28 images, so it was modified. The adaptated architecture includes two sets of convolution layers, one max pooling layer, one flattening layer and four fully-connected layers, with the cross entropy (CE) as criterion. Also, because LeNet5 takes one image at a time, the 1000 pairs of image were split and merge into one set of 2000 images. LeNet5 implemented in the previous format reached an accuracy of 97.27% with a standard deviation of $\sigma = 0.36$. This result stands as the benchmark for future comparisons with the following architectures. Except for the optimization of the final architecture, all performances described in this paper are estimated with the parameters in I.

Parameter	Symbol	Selected value
Number of epochs	n_{epochs}	100
Batch size	$\lvert B \rvert$	5
Learning rate	γ	5.0×10^{-3}

Table I
Fixed values of the parameters used for all performances.

IV. Selection and enhancement of the architecture

A. Weight sharing

B. Auxiliary loss

C. Pretraining

D. Data augmentation

E. Dropout

V. Optimization of the parameters

VI. Results: estimation of the performance

Finally an accuracy of 0.9.. was reached with the ... architecture. Below are the tuned values of the hyperparameters that were used, and the fixed values of the other parameters.

Parameter	Symbol	Selected value
Number of epochs	n_{epochs}	...
Batch size	$\lvert B \rvert$	...
Data addition	D	...

Table II
Fixed values of the parameters used for the final architecture.

Hyperparameter	Symbol	Tuned value
Learning rates	$\gamma_{pretraining}$	1.668×10^{-3}
	$\gamma_{training}$	0.77×10^{-5}
Regularization parameter	$\lambda_{pretraining}$	1.0×10^{-6}
	$\lambda_{training}$	1.0×10^{-1}
Auxiliary loss weight	w_{aux}	2.0×10^{-1}

Table III
TUNED VALUES OF THE HYPERPARAMETERS USED IN THE FINAL ARCHITECTURE.

VII. APPENDIX

Architectures	Type	Criterion	Auxiliary criterion	Weight sharing	Decreasing learning rate	Regularization parameter	Dropout	Image switching	Accuracy	Standard deviation
net	FR	MSE	none	none	✓	✗	✗	✓	75.81	21.85
2channels	BC	CE	none	none	✗	✗	✗	✗	80.16	1.17
2onechannel	BC	CE	none	✓	✗	✗	✗	✗	83.26	1.37
oneimage	BC	CE	none	none	✗	✗	✗	✗	83.67	1.14
2lenet5	BC	CE	none	✗	✗	✗	✗	✗	83.77	0.78
2lenet5	BC	CE	none	✗	✗	✗	✓	✗	84.85	0.72
2onechannel	BC	CE	none	✓	✗	✗	✓	✗	85.25	1.05
2onechannel	BC	CE	none	✓	✗	✗	✓	✓	85.75	1.26
2lenet5	BC	CE	none	✗	✗	✗	✓	✓	86.86	0.75
oneimage	BC	CE	none	none	✗	✗	✓	✗	87.50	0.69
oneimage	BC	CE	none	none	✗	✗	✓	✓	87.80	0.64
2nets	BC	BCE	CE	✗	✗	✗	✗	✗	89.49	7.80
2nets	BC	CE	CE	✗	✗	✗	✗	✗	90.47	4.16
2nets_ws	BC	BCE	CE	✓	✗	✗	✗	✗	94.26	3.69
2nets_ws	BC	CE	CE	✓	✗	✗	✗	✗	95.25	1.69
net2	FR	MSE	none	none	✗	✗	✗	✓	96.99	0.28
LeNet5	FR	MSE	none	none	✓	✗	✗	✓	97.08	0.51
LeNet5	FR	MSE	none	none	✗	✗	✗	✓	97.14	0.47
LeNet5	FR	CE	none	none	✗	✗	✗	✓	97.27	0.36
LeNet5	FR	CE	none	none	✓	✗	✗	✓	97.29	0.52

Table IV
SUMMARY OF THE ACCURACIES IN INCREASING ORDER

Architectures	Type	Number of parameters	Best accuracy (w/o DA)	Standard deviation
net	FR	2.18×10^5	75.81	21.85
2channels	BC	1.25×10^5	80.16	1.17
2onechannel	BC	3.61×10^5	85.75	1.26
2lenet5	BC	7.47×10^5	86.86	0.75
oneimage	BC	3.39×10^5	87.80	0.64
2nets	BC	1.35×10^5	90.47	4.16
2nets_ws	BC	1.35×10^5	95.25	1.69
net2	FR	3.86×10^5	96.99	0.28
LeNet5	FR	3.51×10^5	97.29	0.52

Table V
NUMBER OF PARAMETERS FOR EACH ARCHITECTURE IN INCREASING ACCURACY ORDER