AI HARDWARE

In-focus: DNN Learning Curves

STUDENT: Stanciu Iulia-Cristina

Strategies for the handwritten digit recognition:

For this exercise there are used two machine learning strategies for digit recognition.
The MNIST database is a 70000 samples set made out of images of 784 pixels. The pixels are being analysed to determine the handwritten digit that can be seen in the picture.

Both algorithms require a well-thought division of the provided date into three categories:

- A training set

The training set is usually bigger than the validation set or the test set and it is used to estimate the weights of the neural network

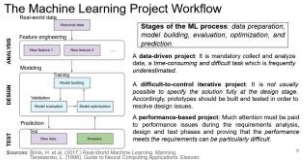
- A validation set

The validation set is used is used in the model evaluation part of the training and it is used to optimize the learning rate of the algorithm.

- A test set

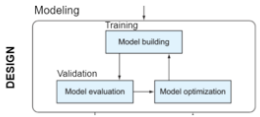
The test set is used for the final evaluation of the algorithm. The performance of the algorithm is given by the processing of the test set and the comparison between the true answer and the predicted one.

Machine learning workflow and learning curves:

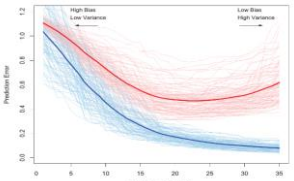


Modelling is one of the most important stages of the ML workflow.

The modelling part of the process consists of a loop involving the computation of the models, estimating the performance for the training set and a model evaluation which is done by processing the validation set.

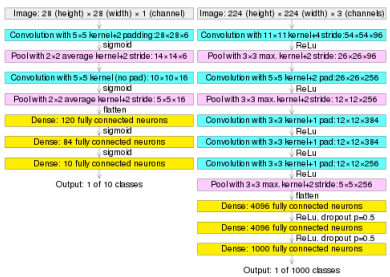


Learning curves are the visual representation of how an algorithm learns during the evaluation and validation part of the process. The "learning" refers to the improvement of the accuracy of the algorithm in a given period of time or of repetition of the loop.

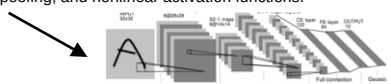


LeNet:

LeNet is a convolutional neural network structure proposed by LeCun. It is a simple convolutional neural network, a neural network whose artificial neurons can respond to a part of the surrounding cells in the coverage range and perform well in large-scale image processing.
LeNet-5 was one of the earliest convolutional neural networks.
(In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of artificial neural network (ANN), most commonly applied to analyze visual imagery. CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer.)
LeCun combined a convolutional neural network trained by backpropagation algorithms to read handwritten numbers and successfully applied it in identifying handwritten zip code numbers provided by the US Postal Service.



LeNet-5 consists of seven layers. In addition to input, every other layer can train parameters. Every convolutional layer includes three parts: convolution, pooling, and nonlinear activation functions.



Analyzing the MNIST dataset with SciKit-Learn and SKORCH and LeNet Convolutional Network respectively:

For the purpose of this exercise the accuracy of the algorithm and the learning curves are analysed for different values of parameters: division of the dataset into training, validation and test sets (the partition can be seen below), values for the number of neurons on the hidden layer of the first algorithm (values of 98, 130,196), values for number of epochs and number of neurons that interconnect the two last layers of LeNet (tested for values of 50, 150 and 200).
As it can be seen it the table below, having the same number of epochs, the accuracy of the algorithms rises while increasing the number of neurons on the hidden layer. At the same time, a large enough training set further improves the accuracy of the algorithm.
Considering the LeNet algorithm, the bigger the number of neurons that interconnects the last two layers, the better the testing accuracy, giving the best results for 200 neurons: accuracy > 98%.

Accuracy values of the algorithms for different learning parameters

Partition of MNIST database					Variables		Accuracy Values			LeNet	LeNet Accuracy Values			
Training Set (Number of samples)	Validation Set (Number of samples)	Test Set (Number of samples)	Training Set (percentage)	Validation Set (percentage)	Test Set (percentage)	Number of neurons on the hidden layer (hidden_dim)	max_epochs	Training accuracy (train_acc)	Validation accuracy (val_acc)	Testing accuracy (test_acc)	Number of neurons that interconnect two of the last layers of LeNet	Training accuracy (%)	Validation accuracy (%)	Testing accuracy (%)	Improvement of the previously misclassified images (%)
28000	21000	21000	40	30	30	98	40	97.6500	96.0429	95.9857					
											50	98.9657	98.4286	98.4057	71.40
35000	17500	17500	50	25	25			97.8267	96.4857	96.4914	150	99.2857	98.7029	98.4742	73.60
											200	99.3229	98.7771	98.7543	78.26
42000	14000	14000	60	20	20			97.9167	96.7286	96.6500					
											50	98.4796	98.0000	97.9619	63.53
49000	10500	10500	70	15	15			98.0163	96.8857	96.7714	150	99.3163	98.8571	98.6286	74.88
											200	99.4286	98.9429	98.7809	76.81
56000	7000	7000	80	10	10			98.1143	96.6857	97.0714					
											50	98.9071	98.3905	98.4524	72.54
28000	21000	21000	40	30	30	130	40	97.9464	96.3000	96.4190	150	99.2214	98.6286	98.5666	74.34
											200	99.2750	98.6810	98.6095	76.98
35000	17500	17500	50	25	25			98.0971	96.7371	96.7657	50	98.9229	98.4743	98.3600	68.43
											150	99.1467	98.5714	98.514	71.21
											200	99.3829	98.8114	98.8571	77.61
42000	14000	14000	60	20	20			98.2619	97.0286	96.9286					
49000	10500	10500	70	15	15			98.3673	97.3143	97.2190					
56000	7000	7000	80	10	10			98.4732	97.1000	97.3286					
28000	21000	21000	40	30	30	196	40	98.2679	96.5476	96.5857					
											50	98.9771	98.4686	98.3771	67.69
35000	17500	17500	50	25	25			98.3914	96.7943	96.8971	150	99.1686	98.6057	98.5543	70.19
											200	99.3714	98.8857	98.7257	74.74
42000	14000	14000	60	20	20			98.5476	97.2143	97.0857	50	98.9429	98.6357	98.4500	68.51
											150	99.4000	98.9571	98.8000	75.15
											200	99.4367	99.0095	98.8095	77.78
49000	10500	10500	70	15	15			98.6265	97.3524	97.3143					
56000	7000	7000	80	10	10			98.7214	97.2143	97.4286					

Observation!
For the LeNet algorithm the number of epochs was initially set to 10.

The evolution of the learning curve can be seen by modifying the number of epochs. An example of the leaning process can be seen below. The bigger the number of epochs (until a point), the better the accuracy.

SciKit-Learn
with 10 epochs

=> Final validation accuracy of 94.38%

epoch	train_loss	valid_acc	valid_loss	dur
1	0.9813	0.8791	0.4486	3.5537
2	0.4051	0.9013	0.3499	1.1126
3	0.4075	0.9113	0.3041	1.1590
4	0.3661	0.9203	0.2727	1.1277
5	0.3336	0.9254	0.2511	1.2105
6	0.3096	0.9303	0.2365	1.0181
7	0.2885	0.9334	0.2252	0.9523
8	0.2812	0.9363	0.2098	1.0198
9	0.2664	0.9387	0.2038	1.2156
10	0.2549	0.9424	0.1930	0.9881

Training accuracy: 95.1229
Validation accuracy: 94.3771

SciKit-Learn
with 40 epochs

=> Final validation accuracy of 96.41%

30	0.1505	0.9581	0.1334	1.2223
31	0.1553	0.9589	0.1328	0.9930
32	0.1546	0.9594	0.1318	1.3246
33	0.1498	0.9580	0.1388	0.9783
34	0.1506	0.9607	0.1303	1.4201
35	0.1500	0.9607	0.1275	1.1896
36	0.1438	0.9607	0.1281	0.5621
37	0.1454	0.9617	0.1249	0.5627
38	0.1447	0.9613	0.1267	0.5776
39	0.1438	0.9624	0.1237	0.5622
40	0.1402	0.9614	0.1237	0.5744

Training accuracy: 97.7514
Validation accuracy: 96.4057

LeNet with 5 epochs
=> Final validation accuracy of 98.47%

epoch	train_loss	valid_acc	valid_loss	dur
1	0.4813	0.9659	0.1077	4.8217
2	0.1104	0.9786	0.0664	0.9827
3	0.1352	0.9813	0.0547	1.0000
4	0.1019	0.9851	0.0458	0.9946
5	0.0919	0.9864	0.0453	1.0038

Training accuracy: 98.9571
Validation accuracy: 98.4686

LeNet with 20 epochs
=> Final validation accuracy of 98.95%

10	0.0662	0.9889	0.0375	1.7723
11	0.0612	0.9871	0.0399	1.5279
12	0.0588	0.9886	0.0377	1.1946
13	0.0583	0.9889	0.0382	1.1597
14	0.0528	0.9899	0.0406	1.0105
15	0.0557	0.9893	0.0368	0.9967
16	0.0503	0.9887	0.0394	0.9916
17	0.0498	0.9891	0.0388	0.9936
18	0.0487	0.9901	0.0359	0.9922
19	0.0469	0.9903	0.0325	0.9881
20	0.0450	0.9909	0.0351	1.0127

Training accuracy: 99.6657
Validation accuracy: 98.9543

REFERENCES

[1] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). *"Gradient-based learning applied to document recognition."*
[2] *LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D. (December 1989). "Backpropagation Applied to Handwritten Zip Code Recognition". Neural Computation.*
[3] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. The MIT Press.
[4] Hastie et al. (2008). The Elements of Statistical Learning, Second Edition (corrected 12th printing). Springer.