Few-shot learning using pre-training and shots, enriched by known samples

Detlef Schmicker

We use the EMNIST dataset of handwritten digits to test a simple approach for few shot learning. Choosing a fully connected neural network with inputs and layer outputs between 0 and 1 and no bias parameters we first trained the network with a subset of the digits. The pre-trained net is used for few shot learning with the untrained digits. Two basic idea were necessary: first the training of the first layer was disabled (or very slow) during few shot learning, and second using a shot consists of one untrained digit together with four previously trained digits and perform a training up to a predefined threshold. This way we reach a 90% accuracy for all handwritten digits after 10 shots.

I. INTRODUCTION

Neural networks have shown stunning success[4], but they usually needed a very big training database or were able to generate a hight amount of training examples from rules [5]. Obviously it is not always possible to fulfill this restrictions, e. g. if you want to teach a computer by hand. Therefore we are interested to learn from few examples. A number of quite different approaches are used and it is even difficult to classify our approach with discussed [6]. Our approach could be classified by refinement of existing parameters, as well as embedding learning, no meta learning and one could argue, that we use external memory, as we use samples from pre-training during few shot learning. Therefore we combine a number of well known concepts.

A very simple neural network is chosen to keep it similar to what is known from the human brain [1]. To avoid optimizing the neural network for a specific task we use only fully connected layers with weights and without. As activation function a function between 0 and 1 is chosen to keep it similar to what is know about human neurons. The function value is close to 0 for input 0, similar to human neurons, where a 0 (not firing) has no effect on the connected neuron, as it is the zero output is only multiplied by a weight. A few tests indicated, that this restriction was not very important for the success of our few shot learning approach, but as it seems to be close to human neurons, we did not see a reason to change it.

The few shot test cases were taken from the EMNIST dataset of handwritten digits[2]. The EMNIST dataset contains 280000 digits from more than 500 different writers, classified using 10 labels, representing the digits. We used 8 digits for pre-training. The idea is, even humans have seen a lot of lines and shapes during there live before they try to read digits. Therefore the have a pre-trained brain. Than we use the two digits, which the neural network has never seen before. The neural networks learns them from few examples.

The pre-training is done with a standard gradient descend. The few shot learning is also done this way, but only uses one new sample at a time, and stops learning depending on a stop-criteria. Without any additional measures, this does not succeed. Two ideas were necessary to succeed with the approach. First the learning was disabled for the first layer, or at least slowed down for the first layer. Second, with every shot some previously known samples are added, but the stop criteria still depends only on the new sample. This helps the network remembering the old labels. This is consistent with human experience: if you do not use old knowledge, you forget it. This way we reach about 90% accuracy with 10 shot learning[3].

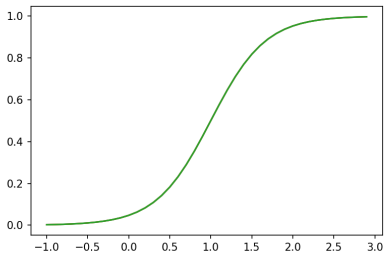

Figure 1: The activation function used.

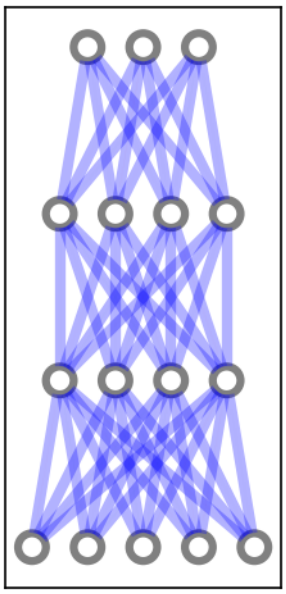

Figure 2: The net used, in this example with 5 inputs, size of the hidden layers 4 and 3 outputs.

II. THE NEURAL NETWORK

Each fully connected layer has i inputs x_i and j outputs y_j, which are related by $y_j = \sum_i w_{ij} x_i$. Each activation layer takes k inputs x_k and has k outputs y_k with the relation $y_k = \frac{1}{1-e^{-(3(x_k-1))^2}}$, which is a sigmoid function, scaled and shifted in x-direction (figure 1). The neural network has one input layer, two hidden layers and an output layer as in figure 2, but the size of the layers are larger in out tests. The loss function $l(y_i) = \frac{1}{2}\sum_i (y_i - \hat{y}_i)^2$ is used, with $\hat{y}$ being the correct value from the training data. Every output of the net corresponds to one of the labels (digits), with this output is 1 and all others 0. The gray scale inputs are scaled to the range from 0 to 1. We clip all weights to a value between -1 and 1, as human neurons do not fire with infinite intensity. The minimum of the loss function with respect to the weights is calculated using a gradient descend method implemented in python [3].

III. THE PRE-TRAINING PROCEDURE

From the EMNIST dataset of handwritten digits we chose 8 digits for the pre-training. With them a neural network with 28x28 input pixels, two hidden layers of size 64 and 10 outputs is trained. From the 10 outputs only 8 are used in the training set, the other two are reserved for few shot learning of the remaining two digits (the digits 2 and 8 in out tests). We trained with a batch size of 1000 for 100000 batches. The EMNIST dataset of handwritten digits contains 280000 digits, which are split into 240000 digits for training and 40000 digits for testing. From the 240000 digits only 192000 digits belonged to the 8 digits we used for pre-training. The pre-training procedure resulted in an accuracy of about 98%, measured with 1000 images from the testing dataset. A prediction is considered correct, if the label belonging to the correct digit had the highest output value of all outputs.

IV. THE FEW SHOT PROCEDURE

Two ideas are used during few shot training: first learning for the first layer was disabled and second for every shot one sample, which was not part of pre-training, was combined with 4 samples, which where part of pre-training. With this batch a gradient descend was performed until the two following stop criteria for the new sample are fulfilled:

1. The value of the corresponding output is bigger than 0.3, *and*
2. The value of the corresponding output is more than 1.5 times the value of the second highest output.

The samples known from pre-training are not taken into account for the stop criteria. We used the two new labels alternately. Two gradient descends with the two different new samples we call one shot.

shot	accuracy old digits	accuracy new digits	accuracy only new digits	overall accuracy
1	0.895	0.492	0.776	0.815
2	0.952	0.574	0.754	0.879
3	0.950	0.521	0.667	0.87
4	0.960	0.535	0.730	0.878
5	0.856	0.621	0.677	0.818
6	0.961	0.708	0.839	0.905
7	0.958	0.666	0.806	0.900
8	0.949	0.747	0.844	0.911
9	0.954	0.811	0.918	0.923
10	0.926	0.836	0.919	0.908

Table I: Experimental results few shot learning

shot	accuracy old digits	accuracy new digits	accuracy only new digits	overall accuracy
1	0.485	0.597	0.706	0.496
2	0.296	0.669	0.716	0.357
3	0.288	0.310	0.67	0.289
4	0.335	0.677	0.727	0.399
5	0.387	0.683	0.811	0.443
6	0.382	0.720	0.806	0.438
7	0.400	0.759	0.893	0.474
8	0.352	0.845	0.900	0.442
9	0.487	0.502	0.820	0.479
10	0.386	0.617	0.947	0.433

Table II: Experimental results without fixed first layer

V. EXPERIMENTAL RESULTS

The result of the few shot learning, measured using the testing data set, is shown in table I. The first accuracy column reports the accuracy, if one just looks how good pre-trained digits, the second how good new digits

shot	accuracy old digits	accuracy new digits	accuracy only new digits	overall accuracy
1	0.569	0.508	0.517	0.558
2	0.672	0.642	0.659	0.661
3	0.565	0.505	0.517	0.557
4	0.649	0.501	0.529	0.626
5	0.564	0.642	0.661	0.584
6	0.684	0.724	0.733	0.696
7	0.364	0.515	0.517	0.398
8	0.458	0.525	0.536	0.474
9	0.454	0.532	0.541	0.477
10	0.451	0.784	0.792	0.508

Table III: Experimental results without enriched samples

shot	accuracy old digits	accuracy new digits	accuracy only new digits	overall accuracy
1	0.896	0.409	0.645	0.809
2	0.754	0.573	0.699	0.722
3	0.926	0.484	0.667	0.841
4	0.855	0.637	0.832	0.816
5	0.836	0.730	0.821	0.813
6	0.890	0.574	0.688	0.830
7	0.935	0.683	0.893	0.880
8	0.879	0.778	0.881	0.860
9	0.878	0.806	0.914	0.865
10	0.874	0.801	0.906	0.862

Table IV: Experimental results with pre-training and few shot training using the first layer with reduced learning rate (by a factor 0.01)

shot	accuracy old digits	accuracy new digits	accuracy only new digits	overall accuracy
1	0.819	0.268	0.661	0.717
2	0.705	0.546	0.667	0.667
3	0.808	0.381	0.617	0.734
4	0.799	0.534	0.834	0.752
5	0.798	0.573	0.852	0.757
6	0.802	0.571	0.850	0.760
7	0.803	0.566	0.849	0.759
8	0.801	0.537	0.744	0.747
9	0.803	0.550	0.760	0.752
10	0.806	0.556	0.769	0.751
100	0.860	0.841	0.964	0.849

Table V: The pre-training is done with the few shot procedure using 5000 shots. This way pre-training and few shot learning is done the same way.

are recognized. The "accuracy only new digits" reports the accuracy, if one tries to distinguish the new digits. Therefore it is just checked, if the correct label of the two new labels has the higher output value. The last column measures, how good the neural net recognizes all 10 digits. A accuracy of around 90% seems quite convincing for few shot learning of handwritten digits from many different writers.

We check both ideas from section IV experimentally. First we did not disable learning for the first layer table II, second we did not enhance the few shot training sample with already learned samples table III. In both case few shot learning failed, as the neural net forgot the pre-trained samples. Therefore it seems, that both ideas are important for the success.

VI. DISCUSSION

Two simple ideas lead to quite successful few-shot learning. First learning of the first layer was disabled during few shot learning and second at every shot four samples from pre-training were added. Both measures could be integrated into a unified learning procedure.

One might use a reduced learning rate of the first layer for pre-training as well as for few shot learning. Reducing the learning rate of the first layer by a factor of 0.01 for pre-training and few shot learning keeps successful, even if it seems a little worse than our original procedure IV.

Enriching the few shot samples with "old samples" during learning can be done straight forward. One might e.g. just keep a sample for each label and use them as "old samples". It is even possible to use the few shot learning procedure for pre-training as welltable V.

[1] Pierre Baldi and Peter Sadowski. A theory of local learning, the learning channel, and the optimality of backpropagation. *Neural Networks*, 83:51 – 74, 2016.

[2] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

[3] Detlef Schmicker. `https://github.com/dsmic/towards_few_shot_learning`.

[4] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[5] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–, October 2017.

[6] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), June 2020.