29th International Conference
ENGINEERING MECHANICS 2023
Milovy, Czech Republic, May 9 – 11, 2023

THE INFLUENCE OF FIRST CNN LAYER INITIALIZATION ON TRAINING CONVERGENCE

Krejsa J. [*], Vechet S. [*], Chen K.S. [**]

Abstract: *During evaluation of convolution neural networks on the task of sign language single hand alphabet classification we have discovered that in small but not negligible number of cases the training of the network does not converge at all. This paper investigates the problem that we believe is independent of the application. While the true cause of training divergence was not discovered, we can offer the reader an easy solution from practical point of view – initialization of the first CNN layer using pretrained networks parameters.*

Keywords: Convolution neural networks, Training, Initialization, Convergence.

1. Introduction

Artificial neural networks became a useful tool in a wide range of engineering tasks. With the recent developments the convolution neural networks (CNN) proved to be able to handle both regression and classification tasks for 2D (and partially 3D) inputs. Those networks in general consist of two parts, the first one handles feature extraction, usually in cascade arrangement, where the first layer extracts the simplest features, such as edge detection, with subsequent layers extracting more and more complex features, while the second part is usually rearranged into folly connected 1D form and handles classification / regression part of the task.

The common approach in engineering related tasks is to use network pretrained on huge datasets, that are available online, thus use the pretrained feature extractors, then retrain the classification / regression layer (depending on the task in question) and possibly fine tune the complete network when large enough training data set is available. However, there are tasks where it is advantageous to train the network from scratch, as the features are task specific. This was the case in our task, where Czech sign language single hand alphabet letters classification problem was addressed using convolution neural networks. The task was resolved (see Krejsa 2020) with over 87% accuracy on verification dataset for the gesturers not previously presented to the network.

There are several ways how to further fine tune the CNN, one of the essential parameters is its topology (number of neurons in each layer, number of layers, etc.). In order to improve the accuracy of the network, a number of experiments with different topology was performed with the goal to find the optimum one. The best topologies exceeded previously obtained results, reaching accuracy over 91%. To further improve the results, the additional extension of training set was provided by newly recorded data and additional fine-tuning of network topology was performed. During those experiments we have made an interesting observation, as some of the networks with otherwise best topology failed to train completely. This paper is focused on our investigation of such problem and the ways how it can be avoided.

[*] Assoc. Prof. Ing. Jiří Krejsa PhD, Assoc. Prof. Ing. Stanislav Věchet, PhD, Institute of Thermomechanics of the Czech Academy of Sciences, Brno department, Czech Republic, krejsa@fme.vutbr.cz

[**] Prof. Kuo-Shen Chen, PhD.: National Cheng Kung University, Department of Mechanical Engineering, No.1, Ta-Hsueh Road, Tainan 701; Taiwan, kschen@ncku.edu.tw

Related literature is mainly focused on the influence of various types of random initialization, especially in common fully connected networks, see e.g. work of Du (2018) and Wanchen (2020), an interesting Laplacian distribution based initialization methods is presented in Catalbas (2018), when deep convolution network architecture similar in principle to the one of ours is considered.

The paper is organized as follows: The structure of the network is given in detail, problem observation is specified, all proposed hypotheses are given and corresponding experimental results are presented.

2. Materials and methods

2.1. Network description

The topology of the network in question is shown in Table 1., where Conv stands for convolution layer, Pool stands for pooling layer, ReLU stands for Rectified Linear Unit function, Max stands for Maximum pooling function, SoftMax is normalized exponential function and Params indicate number of trainable parameters. The classification part consists of two fully connected layers. The total number of parameters to train was 72145.

All computations were performed using TensorFlow library, Adam – stochastic gradient descent method (see Kingma 2015) was used for training. To prevent overtraining the common techniques of L2 regularization and dropout of 50% were used in all experiments.

Tab. 1: Topology of the network.

Layer	Depth	Fcn	Output	Params
Conv 3x3	18	ReLU	222x222	504
Conv 3x3	18	ReLU	220x220	2934
Pool 2x2		Max	110x110	
Conv 3x3	22	ReLU	106x106	9922
Conv 3x3	22	ReLU	104x104	4378
Pool 2x2		Max	52x52	
Conv 3x3	26	ReLU	50x50	5174
Conv 3x3	26	ReLU	48x48	6110
Pool 2x2		Max	24x24	
Conv 3x3	32	ReLU	22x22	7520
Conv 3x3	32	ReLU	20x20	9248
Pool 2x2		Max	10x10	
Conv 3x3	36	ReLU	8x8	10404
Conv 3x3	36	ReLU	6x6	11700
Pool 2x2		Max	3x3	
Full	12	ReLU		3900
Full	27	SoftMax		351

2.2. Discovering the problem

Original training dataset consisted of recordings of 31 gesturers, with data augmentation of translation, rotation and both uniform and non-uniform scaling, leading to total of 595.256 images. The dataset was extended using the recordings of additional 3 gesturers (one hearing impaired female, one sign language interpreter female and on hearing impaired male), resulting in additional 131.477 images (22% of original dataset).

Retraining the network already trained on original dataset improved the test results for additional 2% (93% in total), however, when further experiments were performed with slightly different topology that was trained from the beginning with the extended dataset, the training in some cases failed completely.

2.3. Hypothesis 1 – data itself

Our first hypothesis regarding the cause of the divergency was the data itself. As visual inspection did not reveal any significant differences compared to the original data, numerical experiment was performed, using the batch of networks with the topology described above, trained with original and extended dataset. The batch consisted of 100 networks for each group and convergence/divergence was evaluated after 30 epochs. The results are summarized in Table 2, and one can see that the hypothesis had to be rejected, as networks trained with original data exhibited similar rate of divergence. However, it is remarkable that the rate of divergence was not discovered earlier, as the percentage is surprisingly high.

Tab. 2: Divergence percentage of the network on original / extended dataset.

Dataset	Converged	Diverged
Original	72	28
Extended	69	31

2.4. Hypothesis 2 – sensitive topology

The next idea was that the topology that exhibited the best results so far is very sensitive to weight initialization. Therefore we arranged second experiment with different topologies and number of trainable parameters, including both previously used topologies and new ones, differing mainly in the number of nodes in particular layers (the overall structure having convolution layers followed by max pooling was kept). Extended data was used in all experiments. The results are summarized in Table 3.

Tab. 3: Results for different topologies.

Parameters	Converged	Diverged	Test success rate [%]
19245	100	0	78.9
23505	98	2	80.7
62179	97	3	88.1
72145	73	27	93.2
78512	85	15	91.2
81477	94	6	91.1

It is clear that the topology has an influence on convergence/divergence and the topology that brought the best classification results on test data is the most sensitive, however, it is also apparent that the particular topology is not an exception regarding the divergence of the training.

2.5. Hypothesis 3 – initial weight settings

By rejecting previous hypotheses, we have turned our attention into the network initialization. Therefore, we needed to confirm, that the initial weight settings is determinative factor. Other set of experiments was prepared, with single network initialized and trained repeatedly. To further confirm the independence of the process on the training dataset, both original and extended datasets were used. The results were convincing, as none of the networks converged, the training failed in all of them.

3. Replacing the first layer

We have done a number of other experiments regarding the way the initial weights are generated, together with the different settings of learning algorithm parameters. All those factors do influence the convergence/divergence rate, but only to certain extent. Details are out of the scope of this paper and in principle do not resolve the problem completely.

To ensure the convergence while keeping the proposed most promising topology of the network, we decided to replace the weights of the very first layer with pretrained weights of the converged network. The visualization of the kernels of both the untrained and trained network are shown on Figure 1. The first layer kernels perform the very basic processing of the input image and produce the simplest features, therefore can be considered independent on the task itself.

Another batch of experiments was performed, with the first layer kernel values loaded from trained network (and being the same for all the networks in the batch) and all other trainable parameters being initialized randomly. The results were surprising, all 100 networks converged successfully, proving that the change in the first layer only is sufficient to prevent the divergence of training.

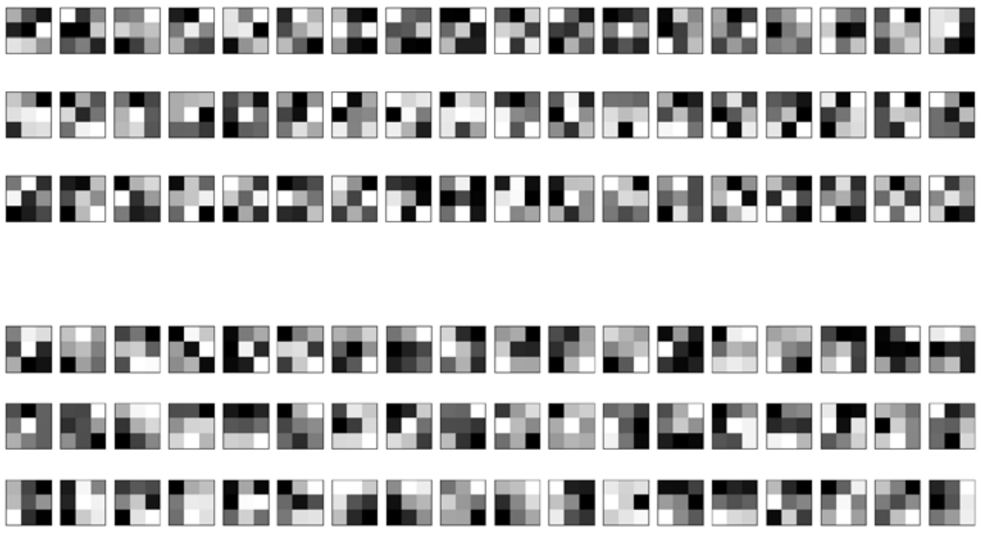

Fig. 1: Visualization of first layer filters (convolution kernels). Randomly initialized (top), trained (bottom). Filters are shown for all three RGB channels (three rows) and all 18 nodes (columns).

4. Conclusions and future work

The most valuable conclusion that we derived from our experiments is the fact, that the probability of the training to diverge increases with the overall performance of trained network. We have not encounter such behavior in other tasks and therefore we can not be sure whether this is a general property of CNNs, as no similar observation was found in the literature.

The way how to tackle the problem finally appeared to replace the very first layer of the network with pretrained values. While this approach does not explain the reason of the network to be stack in local minima right from the start of the training, we believe that for practical purposes it might help others when searching for optimal topology.

In future work we want to test the partial only replacement of first layer kernels.

Acknowledgement

The results were obtained with institutional support RVO 61388998 of the Institute of Thermomechanics of Czech Academy of Sciences and with support of Mobility Plus Projects with Ministry of Science and Technology of CAS, project no. MOST-20-06.

References

Çatalbaş B, Çatalbaş B, Morgül Ö., (2018), A new initialization method for artificial neural networks: Laplacian, 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, pp. 1-4.

Du S., Lee J., Tian Y., Singh A., Poczos B., (2018) Gradient Descent Learns One-hidden-layer CNN: Don’t be Afraid of Spurious Local Minima, Proceedings of the 35th International Conference on Machine Learning, PMLR 80:1339-1348.

Kingma D., Ba J., (2015), Adam: A Method for Stochastic Optimization, 3rd International Conference for Learning Representations, San Diego.

Krejsa J., Vechet S., (2020) Czech Sign Language Single Hand Alphabet Letters Classification, Proceedings of the 2020 19th International Conference on Mechatronics - Mechatronika, ME 2020. New York: IEEE.

Wanchen L, (2020), Analysis on the Weight initialization Problem in Fully-connected Multi-layer Perceptron Neural Network, 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, China, 2020, pp. 150-153.