Bulletin of the *Transilvania* University of Braşov • Vol. 13 (62) No. 1 - 2020
Series I: Engineering Sciences
https://doi.org/10.31926/but.ens.2020.13.62.1.4

THE IMPACT OF RESIDUAL LAYERS AND INCEPTION MODULES ON META-LEARNING

Luciana BULARCA[1]

Abstract: *This paper presents an algorithm with which artificial agents should be able to learn and adapt quickly from only a few examples. This algorithm respect the initial goal of neural networks to be able to learn and adapt in time. However, the work is just on the beginning and the best results have not been reached yet. This paper aims to present the impact of residual layers and inception modules on meta-learning in order to obtain an improvement of results.*

Key words: *meta-learning, inception modules, residual layers, adapt, few examples.*

1. Introduction

The initial goal of neural networks was to solve the problems in the same way that human brains works. In the meantime the attention was directed to performance due the need for it and leaving aside the initial purpose because of the difficulty. To achieve the highest performance in solving the tasks, the learning task used large datasets and high capacity of neural networks architectures. Therefore, the larger is the database, the better is the accuracy and performance. It took a long time for researchers to discover a solution for an algorithm that works almost like human brain. This solution is meta-learning. The idea behind meta-learning, or learning to learn, is the ability to adapt to solve new problems using previous experience. Like humans, this method can handle multiple tasks with a minimum of information. The meta-learning system has two main parts. In the first step the model is exposed to a variety of learning tasks in the training part and in the second part then the learning ability for new tasks is tested using a small number of training samples. This approach is compatible with any model trained with gradient descent procedure and applicable to a variety of different learning problems, including classification, regression, and reinforcement learning.

In this work the focus is on two architectures that I will use for classification. To achieve these architectures I used two most recent ideas: Residual connections introduced by He et al. in [2] and the Inception architecture [9]. These two variants are high-capacity neural networks architectures and their use in original variants will produce overfitting. For this reason these models must be created with low capacity. To

[1] Dept. of Electronics and Computers, *Transilvania* University of Braşov, Romania.

analyse the performance of fast learning I used the models for two experiments. The first experiment is for 1-shot and the second experiment is for 5-shot learning.

This paper is structured as follows. Section II provides an overview of related work. Section III present the implementation details. Section IV provides the experiments and results. Finally, Section V describes the conclusions and future work.

2. Related Work

There has been a lot of work regarding the meta-learning. Here is provided a brief overview of some existing meta-learning methods.

There are three approaches to meta-learning depending on which part it is focused on: recurrent model meta-learning, metric meta-learning and optimizer meta-learning. The meta-learning approach based on recurrent model is a method of learning sequence with sequence which assumes that each task is organized into in-out pairs. The meta-learning approach based on classification tasks is a method in which the samples are learned episodically, from episode to episode classes, labels and samples are shuffled. Using a recurrent neural network with augmented memory, according to [7], the classification for new samples is based on the pairs of samples labelled previously in the same episode. This approach is used in [5], [7] and [11]. Metric meta-learning approach is based on learning a metric space where similar samples are closer. In the case of new sample they are assigned with a label based on the distance in the learned metric space. This approach is used in [4], [8] and [12]. Optimizer meta-learning approach is based on the training of initialization and updating mechanism to adapt to new tasks. Using gradient descent, it provides useful information to adapt to new tasks. This approach is used in [1], [6].

In this work I started from the algorithm presented in [1]. The main idea of the algorithm is to instruct the model's initial parameters with a small amount of data using one or mode gradient steps that can produce an internal representation to produce good results for multiple tasks. This procedure, which is presented in the article [1], is optimized so that its adaptation happens in a space suitable for quick learning.

3. Implementation Details

This work has three experiments. The first experiments was to reproduce the results from article [1] using the linear architecture, the second experiment was to change the model by adding residual layers and the last one was to use inception modules.

3.1. Linear Architecture

The first step was to reproduce the training bases on the architecture suggested in article [1] to observe and get the result presented. The architecture used to perform the image classification contain 4 modules with a 3 x 3 convolutions and 64 filters, followed by batch normalization [3], and 2 x 2 max-pooling. The model architecture is presented in Figure 1.

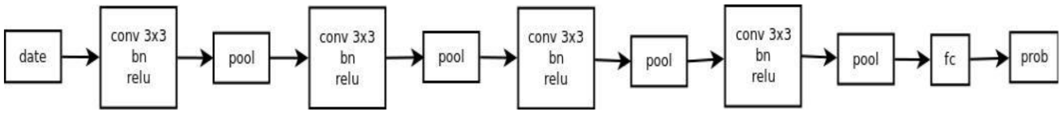

Fig. 1. *The linear architecture*

3.2. Residual Layers

The second choice of architecture is an architecture composed of three residual layers [10]. The choice to use these layers is because the use of residual layers gives good results in training, which is why they are often used in large architectures, their purpose is to provide a fast connection between layers without having to learn. These residual layers consist in two branches, where the first branch uses a 3 x 3 convolution and the second branch uses a 1 x 1 convolution and then a 3 x 3 convolution. It is important that the exit values in each branch to have the same dimension in order to make a sum of the branches. The architecture is presented in Figure 2. This model have 43872 parameters.

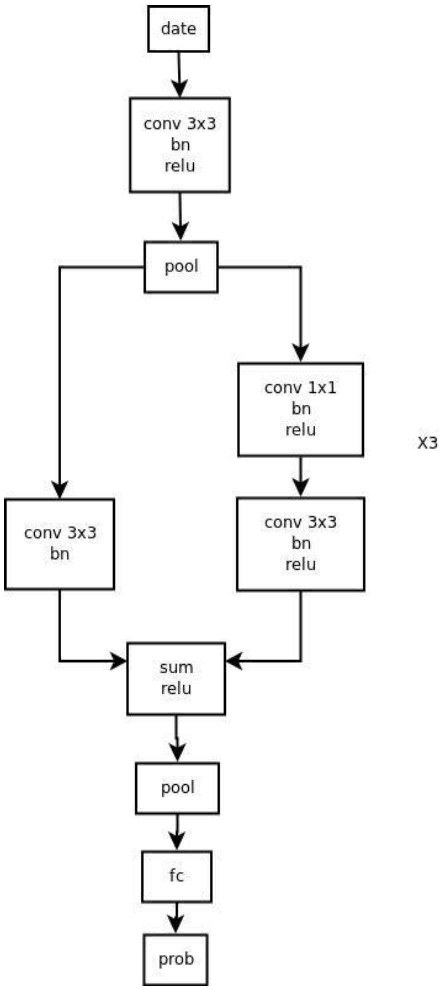

Fig. 2. *The architecture using residual layers*

3.3. Inception Modules

For the inception version of the architecture I used two modules that aim to analyse the features in images on different scalability: 1 x 1, 3 x 3 and 5 x 5. The model contain 2 inception layers, each of inception layers consist in four branches. On two branches is only one convolutional layer but with the follow difference, on one of branches is 1x1 convolutional layer and the other is 3x3 convolutional layer. Another branch consist in 1x1 and 3x3 convolutional layers. The last branch has 1x1 convolutional layer and 2 3x3 convolutional layers. All this branches are concatenated at the end of the module and aims to analyse the features on different scalability. The architecture is presented in Figure 3.

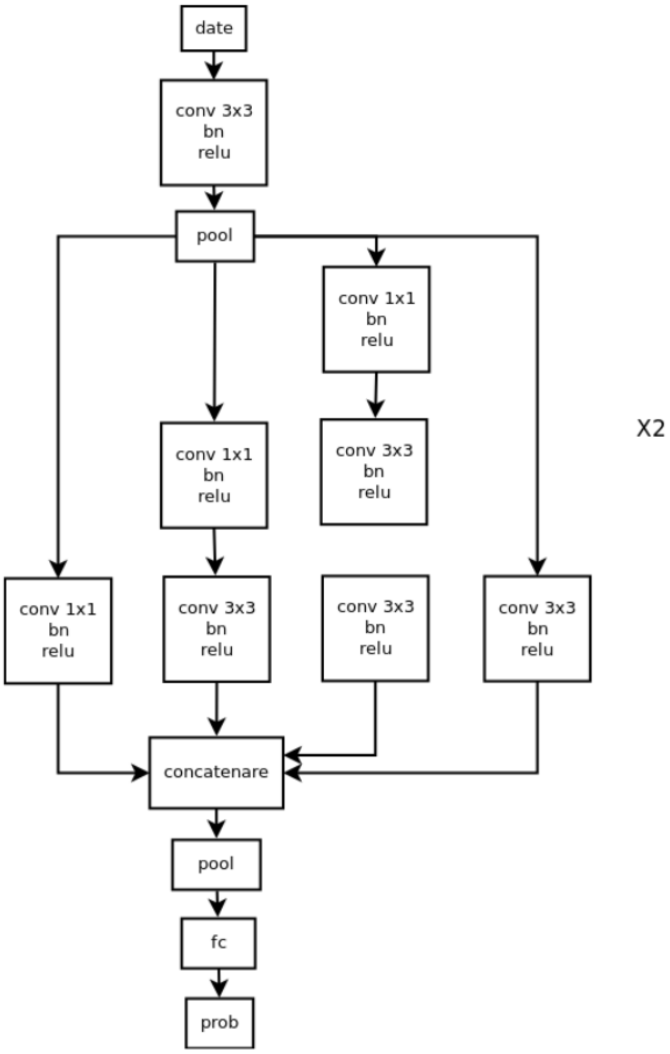

Fig. 3. *The architecture using inception modules*

4. Experiments and Results

Fig. 4. *Examples from Mini-ImageNet dataset*

Figure 4 presents some examples of images from the dataset used. The Mini-ImageNet dataset was proposed by Ravi & Larochelle [6], and involves 64 training classes, 12 validation classes, and 25 test classes. This dataset is the most common used few-shot learning benchmarks. I follow the experimental protocol proposed in [1], which involves fast learning of N-way classification with 1 or 5 shots. Set up problem of N-way: N represent unseen classes and K represent different instances of each of the N classes. All the experiments were based on 1 shot-learning and 5-shot learning.

To reduce overfitting I used 32 filters per layer. For all models, the loss function is the cross-entropy error between the predicted and true class.

The training for 1-shot classification was performed on three types of models: linear, residual layers and inception modules. For each experiment were used 5 classes with 16 samples per class from the Mini-ImageNet database. For the 5-shot classification I followed the experimental protocol and used 5 classes with 20 samples per class from the same database. The image used are in RGB format with size of 84 x 84.

This experimental protocol presents two steps, first step is for meta-learning which aims to determinate a set of adaptable parameters with 5 or 25 samples. The second step is the learning phase, in this step the remaining 75 samples are used and the gradient descend updates for 1 or 5 times depending on the classification method. The adaptation process is similar to the previous one. The samples used is divided into two parts, using K samples for the meta-learning stage and K*N-K samples for the adaptation stage. Similar to learning the previous case, we use the samples to achieve meta-learning with the difference that the parameters used in the first pass through the network are the parameters obtained in the previous meta-learning stage. To learn the classes of interest, after using previous experience, five updates are made with a small step of gradient. At the end of the five passes through the model, performance is obtained to determine the new classes. The learning rate used is 0.001.

The results for 5-way are present in Table 1. I compared performance with other existing methods. Although is not state-of-the-art, the performance achieved in near, and performs better than the other methods used in fine-tuning, nearest neighbour baseline, matching-learner net [12].

The performance for different algorithms Table 1

Mini-ImageNet	5-way Accuracy	
	1-shot	5-shot
Fine-tuning	28.86%	49.79%
Nearest neighbour baseline	40.38%	50.39%
Matching-learner nets [12]	42.72%	54.58%
Meta-learner LSTM [6]	42.67%	59.89%
MAML [1]	**45.72%**	**61.51%**
Residual layers	44.10%	**62.57%**
Inception modules	**46.36%**	58.87%

In Figure 5 are presented the performances for the experiments on 5-way 1-shot. The Figure 5 shows that using inception modules improves the performance on meta-learning algorithm.

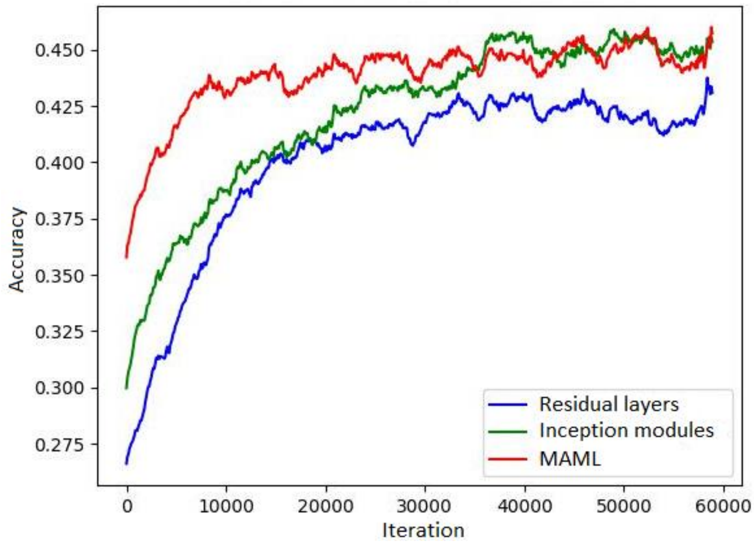

Fig. 5. *5-way 1-shot performances*

Another experiment was 3-way based. The performance obtained using the residual layers is presented in Figure 6. In this experiment, each class was trained using only 20 samples with 0.01 learning rate, in the validation part the model aim to adapt and learn new classes with a small number of samples.

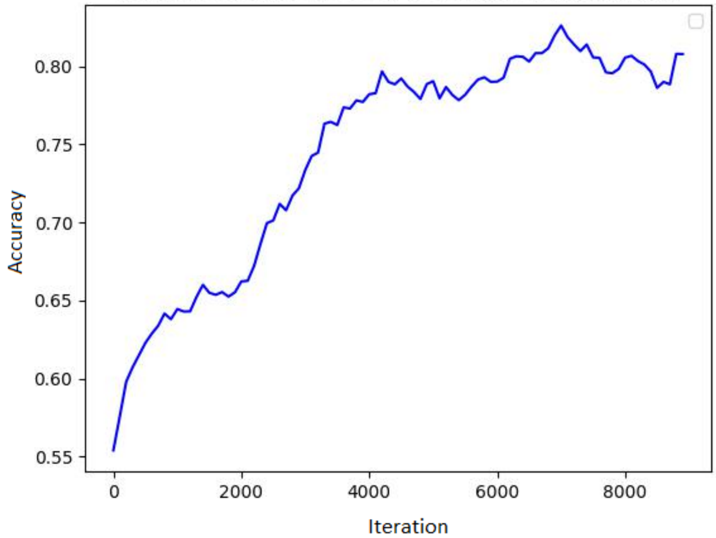

Fig. 6. *The performance for 3-way 5-shot on residual layers model*

For the 3-way training based on residual layers, the performance increase a lot, if for 5-way obtained 62% with 3-way obtained 83%.

5. Conclusion and Future Work

The results obtained using the architectures implemented with residual layers and inception modules shown that residual layers performed on 5-shot learning and the inception modules performed on 1-shot learning. Also, for 3-way 5-shot had 21% improvement in performance. The purpose was that the neural networks to be able to learn and adapt to new task in the same way that human brains. With meta-learning method the model can learn for only few images like humans. Even if the model not achieve the best results on 5 classes, the model achieve a very good performance on 3 classes.

For future work I propose to achieve learning using annotated images for training. The reason why I want to use this approach is because the images from Mini-ImageNet dataset are difficult to learn and the system does not have a large number of images to figure out what it is in the image. The bounding box is the most commonly used technique to make object recognizable for machines by training them to learn from these data and give a relevant output. Although the annotation require much work, the advantage in this case is that the number of examples is small.

References

1. Finn, C., Pieter, A., Sergey, L.: *Model-agnostic meta-learning for fast adaptation of deep networks.* In: arXiv preprint arXiv: 1703.03400, 2017.
2. He, K., Zhang, X., Ren, S., Sun, J.: *Deep residual learning for image recognition.* In: arXivpreprint arXiv: 1512.03385, 2015.

3. Ioffe, S., Szegedy, Ch.: *Batch normalization: Accelerationg deep networks traing by reducing internal covariate shift.* In: International Conference on Machine Learning (ICML), 2015.
4. Koch, G., Zemel, R., Salakhutdinov, R.: *Siamese neural networks for one-shot image rexognition*. In: ICML Deep Lerning Workshop. **2** (2015).
5. Nikhil, M., Mostafa, R., Xi C., Pieter A.: *Meta-Learning with Temporal Convolutions*. In: arXiv preprint arXiv: 1707.03141, 2017.
6. Ravi, S., Larochelle, H.: *Optimization as a model for few-shot learning.* In: International Conference on Learning Representations (ICML), 2017.
7. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: *Meta-learning with memory-augmented neural networks*. 2016.
8. Snell, J., Swersky, K., Zemel, R.S.: *Prototypical Networks for Few-shot Learning*. In: arXiv preprint arXiv: 1703.05175, 2017.
9. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: *Rethinking the inception architecture for computer vision.* In: arXiv preprint arXiv:1512.00567, 2015.
10. Szegedy, Ch., Ioffe, S., Vanhoucke, V., Alemi, Al.: *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.* In: arXiv: 1602.07261, 2016.
11. Tsendsuren, M., Hong, Y.: *Meta Networks*. In: arXiv preprint arXiv: 1703.00837, 2017.
12. Vinyals, O., Blundell, Ch., Lillicrap, T., Wierstra, D., et al.: *Matching networks for one shor learning.* In: Neural Information Processing Systems (NIPS), 2016.