# DO DEEP CONVOLUTIONAL NETWORKS REALLY NEED TO BE DEEP AND CONVOLUTIONAL

**Samuel LEONARDO GRACIO** **Victor STIMPFLING** **Martin VERSTRAETE**

## Abstract

In this paper we try to reproduce some of the experiments conducted by G.Urban et al. (1) on the CIFAR-10 dataset and see if we come to the same conclusion when working with CIFAR-100 dataset. The results are less promising on the second dataset. Globally our results support the hypothesis carried by the paper of G.Urban et al. (1). Indeed we were able to show empirically and given two specific data sets that convolution is the best way to classify images. As a matter of fact our student networks needed at least some convolutional layers in order to perform well. Moreover we could show that depth is a really important parameter, since deeper networks usually performed better, even if the shallow networks were trained with leading method such as distillation.

## 1 Introduction

Despite all the evidences presented by G.Urban et al. (1) (2017), we decided to repeat empirical experiments in order to check if their conclusions are relevant for a more complex data set. To do so we choose to firstly reproduce the main experiments performed by G.Urban et al. (1) and secondly to adjust our network in order to match the minimal requirement to be performing on CIFAR-100. Thus, in the second part of this study our networks will be structurally speaking the same as in the first part but with a different number of parameters in order to avoid underfitting. Lots of works support the idea that deeper is better, but only few have tried to establish whether the good performances of the classical deep convolutional networks are linked to the fact that they are convolutional, deep or have in general more parameters. Our goal is to check whether shallow and/or non-convolutional networks trained with leading methods can be as accurate or at least show close performance to the one of the state-of-the-art networks.

Our study is of great interest since it should establish empirically that in the domain of image classification multi-layer perceptrons are incapable to reach the level of accuracy obtained by a convolutional neural network. Moreover this project should show that for convolutional networks, depth is a critical parameter in order to increase the accuracy of the network.

Our first task will be to try to reproduce a high accuracy network to build a strong teacher model for distillation technique. Our teacher model consists in 14 layers including convolutional, max pooling and fully connected layers as explained in the method section. Secondly we will use all known leading methods in order to compute a set of shallow and/or non-convolutional networks. To do so we will use methods that has proven their worth namely knowledge distillation and model compression. Those methods were introduced in (3) by C.Bucilu, R.Caruana, and A.Niculescu-Mizil and settled by G.Urban et al. (1), Ba and Caruana (2) and Hinton et al. (4) in their respective work. As explained in the first part of this introduction the differences in accuracy between our networks of interest can be explained by a multitude of factors and combinations of them but only depth and the presence or absence of convolution is of great interest for us. In order to have as readable results as possible, the

global architecture of our networks, the number of parameters, the training sets and learning process will have to stay, to the greatest extend, unchanged.

In this paper we corroborate the idea carried by G.Urban et al. this is to say that convolution is needed in order to obtain satisfactory results in the domain of image classification. This could be distinctly seen on CIFAR-10 but also noticed on CIFAR-100. As in (1) our findings support that despite the fact that the models are trained with distillation multiple convolutional layers are needed in order to obtain high accuracy. This is to confirm that depth is an important parameter.

## 2 Related Work

The main reference paper for our work is the one written by G.Urban et al. (1). It presents a state-of-the art network and some student networks all tested and trained on the data-set CIFAR-10. The student networks are trained to mimic the state-of-the-art network this ensure to have as good results as possible for shallow networks as introduced by C.Bucilu et al. (3). This paper shows that on this data-set, student neural networks with a reasonable number of hidden layers only achieved poor results compared to student convolutional networks. Indeed the best neural network could only achieve an accuracy of 71.5% whereas the worst convolutional network with only one convolutional layer achieved an accuracy of 84.5%. When both CNN and Neural networks had the same number of parameters the difference was even more noticeable. Furthermore this paper underlines that for both CNN and Neural Network, depth is, under reasonable conditions, a critical parameter for the performance of the student networks. This is especially true if we look at the CNNs where the accuracy increases by 6.8% when comparing a 4 layer network to a 1 layer network with the same number of parameters. This paper suggest that more research must be conducted in order to verify that the results they obtained empirically can be applied to more complex data-set. However they underlined that it is highly unlikely that a shallow non-convolutional network could ever achieve accuracy comparable to the one of a convolutional model.

Our paper of interest was itself an answer to an earlier paper by Ba and Caruana (2). This work suggested that in the domain of speech recognition and especially on the TIMIT data-set, shallow neural networks were able using model compression to learn as complex functions as deeper convolutional neural networks. In this situation the difference in percent error was only of 1.5%. However when it comes to image classification the results they obtained on CIFAR-10 and it's extension was not as easily readable as the one obtained on the speech recognition problem. In fact the student neural networks obtained only poor results. Thus one convolutional layer had to be added in order to obtain results comparable to the one of the state-of-the-art model.The difference in percent error between the teacher CNN and the student with only one convolutional layer was only of 3.2%. Again the conclusion of this part was that shallower networks are able to learn complex functions. The first paper by G.Urban et al. (1) criticizes to some extent this conclusion since the CNN used as a teacher and thus as reference was not really state-of-the-art. As a matter of fact their best network showed an accuracy of 89% whereas the one trained by G.Urban et al. could reach an accuracy of 93.8 %, showing a maximal difference between the teacher and the best shallow student network of 6.1%.

## 3 Data

### 3.1 Data-set

In the end of the day two similar data sets were used, CIFAR-10 and CIFAR-100, both datasets contain 50000 train images and 10000 test images but the first one has only ten labels while the second one contains 100 labels.

### 3.2 Data-regularization

One of the main regularization we applied to the models is the data augmentation which will be described in the next section.

We used Batch Normalization before every activation layer and some drop-out as well, preventing the models from overfitting too easily.

### 3.3 Data Augmentation

First, we tried to implement the data augmentation technique that was used in (1) but that gave us inconsistent results as the RGB values could go out of the 0-1 interval. So we decided to do our own data augmentation on the fly for the teacher model. We simply created a function able to crop each image to a fixed size of 28 by 28, at random positions for the training pictures and in the center for the testing pictures. After that, we flipped horizontally each image with a probability of 0.5, using a Keras already implemented function. The transformed pictures were then generated by the ImageDataGenerator.flow() method from Keras at each epoch.

Then, the same was done for the students models as it was impossible to do offline augmentation just like in the paper for memory usage reasons.

## 4 Methods

In this paper we revisit the experiment done by G.Urban et al. (1) on CIFAR-10 in their work on the same subject, in order to get good basis on the subject and imagine structures of teacher models.

All the implementations were realised with Keras on Google Colab, which gives us unlimited access to a K80 GPU.

### 4.1 Teacher model

The structure of our teacher network was directly extracted from the work of G.Urban et al. (1). Their teacher model had the following architecture, $76c^2-mp-126c^2-mp-148c^4-mp-1200fc^2-10fc$ where $fc$ stands for fully connected layer, $c$ for convolutional layer and $mp$ for max pooling. The number before the convolutional layer shows the number of filters and nodes applied at each layer. All convolutional filters are of size 3 by 3 applied with a padding so that the output has the same size as the input and a stride of 1. In the reference work, all parameters were optimized by Bayesian optimization. In our project we used ADAM and RMSPROP gradient descent with different parameters as summarized in Tab.1 for the networks on CIFAR-10 and in Tab.4 for the network on CIFAR-100. The other parameters such as dropout rate , number of epochs, batch size and number of pixels cropped varied according to the previously obtained results. We could have done a more complex method in order to optimize our model, like a true Bayesian optimization, but the final result given by this approach was still a really good result according to the different benchmarks on the CIFAR dataset available on the Internet.

### 4.2 Model compression and Distillation

Our goal is to establish whether shallow networks are able to learn as complex functions as deep convolutional networks. We do not want our study to be limited by the learning algorithm that may still evolve through time. To do so distillation and model compression can be used. Through this method "dark knowledge" is injected into our shallow network as explained by C.Bucilu et al. (3).

Model compression consists in training the student network with the output of a teacher network. This techniques enables us to learn directly the function learned by the big network. In the model compression process, the label used to train the student networks are not the softmax output of the teacher networks. Indeed the softmax operation leads to a drastic decrease in the weight of the non-predicted class, limiting the influence of the "dark knowledge" we aim at exploiting. To avoid those drawbacks we used what is often called the logits. Those are the values used as inputs for the softmax layer.

Now that our labels are different, the cross entropy loss is not suitable anymore. We thus have to implement a new loss function. We chose to use the mean squared error loss (Eq.1).

$$\mathcal{L}(W) = \frac{1}{N}\sum_{n} ||g(x^{(n)};W) - z^{(n)}||_2^2 \tag{1}$$

where $g(x^{(n)};W)$ is the prediction of the student network on the $n^{th}$ training data example and $z^{(n)}$ is the soft target obtained from the teacher model.

The concepts of distillation and model compression seem to be, according to the work of Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil (3) a leading method in order to train shallow networks. However this method is limited by the accuracy of the teacher network. In fact if the teacher is not really accurate, the student network will accumulate errors and learn inexact functions. One solution could have been to modify the soft labels with the hard labels or to change the loss function to have a weighted average of a function of the hard and soft labels as suggested by Hinton et al. in (5).

### 4.3 Student Networks

Since our project is bidirectional, the following experiment plan appeared as adapted. We will first compute both CNN and MLP with the same depth in order to be able to compare the relevance of convolution on the accuracy of the network. Furthermore we have to compute MLPs and CNNs of different depths in order to check for the influence of the depth on the accuracy of networks.

#### 4.3.1 Fully connected neural networks

For both CIFAR 10 and 100 datasets, we trained a series of student models using the distillation technique described before. All the networks had roughly the same number of parameters, 1 million for CIFAR-10 and 10 million for CIFAR-100. We trained networks whose depth went from 1 to 4 layers. The exact structure of each network is described in the tables 2 and 5 using the previously described nomenclature. Their accuracy were computed by applying the softmax function on their predictions. This can be considered as the best approach since we used a coherent number of parameters to avoid either overfitting and underfitting. Moreover, the fact that we chose the distillation technique has improved our results. If we had more time, we could have made a better optimization of the different parameters of our student networks.

#### 4.3.2 Convolutional neural networks

We repeated the same process with the CNNs to see if the convolutional layers really gave an added value. Each convolutional layer was followed by a max pooling layer. Again the exact structure of each convolutional network is described in the tables 3 and 6 using the previously described nomenclature. The results of those experiments for the CNN and MLP networks, are given in the tables 2,3,5 and 6 in the next section.

## 5 Experiments

Our first experiments summarized in Table 1 aims at finding the most accurate network on CIFAR-10. This is done by varying different network-related parameters. Unlike in our reference paper (1) we had to vary those parameters manually. Nevertheless, our best network achieved an accuracy of 91.34 %. That is a satisfactory accuracy considering that we did not do any Bayesian optimization and that we did not use offline data augmentation. We considered those results to be good enough in order to use distillation.

Considering the experiments on the student networks, all our results are approximately 15% lower in accuracy than the one of the student networks in the paper. However considering the fact that we did no offline data augmentation and no hyper parameter optimization on our student networks, adding the fact that our teacher student does not have the same accuracy as the one in the paper, those results can be considered as satisfactory. In fact the performed data augmentation that consisted in generating 160 images from an unique image labelled with the soft labels of the original image was too expensive in memory. Nevertheless even if our networks are not as powerful as the one of the paper the dependency between accuracy and depth or convolution seems to follow the same pattern. Given our resources, the accuracy we obtained for our teacher model was unexpected. Despite our unexceptional results on the student networks, we were able to capture the main dependencies related to our parameters of interest. CNN are always better than MLP and the deeper they are, the better they are.

The experiments on CIFAR-100 are more tedious. The latest work show an accuracy around 90% on CIFAR-100. However those are achieved by modern networks (7) it was hard for us to reach those kind of accuracy. Our best network on CIFAR-100 reached an accuracy of 63.82% but we

performed almost no hyper-parameter optimization and only with CNN, like for CIFAR-10. Those outcomes can explain the poor results obtained for the student models. We did try to change the number of parameters in our student models in CIFAR-100 because we noticed that the models were underfitting with only 1M parameters. This is due to the bigger number of labels. Despite our networks were not really powerful a tendency was still noticeable. In fact accuracy increased when we had convolutions and depth had generally a positive influence on our CNNs. The hypotheses that were true for CIFAR-10 seem always true for a more complex dataset, CIFAR-100. All the different results are available on the different tables below :

### 5.1 Table of results for CIFAR-10

Table 1: Teacher model trial using keras tensorflow with architecture: $76c^2 - mp - 126c^2 - mp - 148c^4 - mp - 1200fc^2 - 10fc$ on CIFAR-10 (the other parameters for ADAM optimization are beta-1=.9, beta-2=.999, epsilon=None, ams-grad=False ; decay is 1e-6 for RMSPROP and 0.0 for ADAM)

| Model | Data Augmentation | Dropout weight | Epochs | Optimizer | Lr | Batch-size | Test accuracy |
|---|---|---|---|---|---|---|---|
| 1 | 28 pixel crop | 0.25-0.25-0.25-0.25-0.25 | 50 | RMSPROP | 0.001 | 32 | 0.8704 |
| 2 | 28 pixel crop | 0.2-0.3-0.37-0.42-0.42 | 50 | RMSPROP | 0.001 | 32 | 0.8720 |
| 3 | 30 pixel crop | 0.2-0.3-0.37-0.42-0.42 | 50 | RMSPROP | 0.001 | 32 | 0.8742 |
| 4 | 28 pixel crop | 0.2-0.3-0.37-0.42-0.42 | 50 | RMSPROP | 0.001 | 32 | 0.8818 |
| 5 | 30 pixel crop | 0.25-0.25-0.25-0.25-0.25 | 50 | RMSPROP | 0.001 | 32 | 0.8604 |
| 6 | 30 pixel crop | 0.2-0.3-0.35-0.40-0.40 | 60 | RMSPROP | 0.001 | 32 | 0.8789 |
| 7 | 28 pixel crop | 0.2-0.3-0.35-0.40-0.40 | 60 | RMSPROP | 0.001 | 64 | 0.8931 |
| 8 | 28 pixel crop | 0.2-0.3-0.37-0.42-0.42 | 100 | ADAM | 0.001 | 32 | 0.9013 |
| 9 | 28 pixel crop | 0.2-0.3-0.37-0.42-0.42 | 200 | ADAM | 0.001 and 0.0003 | 32 | 0.9134 |
| 10 | 28 pixel crop | 0.2-0.3-0.37-0.42-0.42 | 100 | ADAM | 0.0003 | 32 | 0.8973 |
| 11 | 28 pixel crop | 0.2-0.3-0.37-0.42-0.42 | 100 | RMSPROP | 0.0001 | 16 | 0.8894 |

Table 2: Results for the neural networks student models on CIFAR-10

| Model | Architecture | Parameters | Accuracy |
|---|---|---|---|
| 1 layer | $500fc$ | 1M | 43.02% |
| 2 layer | $500fc - 200fc$ | 1M | 47.15% |
| 3 layer | $400fc - 300fc - 200fc$ | 1M | 49.92% |
| 4 layer | $400fc - 350fc - 200fc - 100fc$ | 1M | 47.25% |

Table 3: Results for the convolutional neural networks student models on CIFAR-10

| Model | Architecture | Parameters | Accuracy |
|---|---|---|---|
| 1 layer | $75c - mp - 75fc$ | 1M | 67.03% |
| 2 layer | $100(c - mp)^2 - 250fc$ | 1M | 74.81% |
| 3 layer | $100c - mp - 110(c - mp)^2 - 1000fc$ | 1M | 74.32% |
| 4 layer | $75c - mp - 100(c - mp)^3 - 1100fc$ | 1M | 75.96% |

### 5.2 Table of results for CIFAR-100

Table 4: Teacher model trial using keras tensorflow with architecture: $776c^2 - mp - 126c^2 - mp - 148c^4 - mp - 1200fc^2 - 10fc$ on CIFAR-100 (the other parameters for ADAM optimization are beta-1=.9, beta-2=.999, epsilon=None, ams-grad=False ; decay is 1e-6 for RMSPROP and 0.0 for ADAM)

| Model | Data Augmentation | Dropout weight | Epochs | Optimizer | Lr | Batch-size | Test accuracy |
|---|---|---|---|---|---|---|---|
| 1 | 28 pixel crop | 0.2-0.3-0.37-0.42-0.42 | 50 | ADAM | 0.001 | 32 | 0.6148 |
| 2 | 28 pixel crop | 0.2-0.3-0.37-0.42-0.42 | 50 | RMSPROP | 0.001 | 32 | 0.5834 |
| Note: | Different structure : $76c^2 - mp - 126c^2 - mp - 148c^4 - mp - 1200fc^2 - 100fc$ | | | | | | |
| 3 | 28 pixel crop | 0.2-0.3-0.37-0.42-0.42 | 100 | ADAM | 0.001 | 32 | 0.6382 |

Table 5: Results for the neural networks student models on CIFAR-100

| Model | Architecture | Parameters | Accuracy |
|---|---|---|---|
| 1 layer | $4000fc - 100fc$ | 10M | 5.20% |
| 2 layer | $3000fc - 1000fc - 100fc$ | 10M | 12.34% |
| 3 layer | $2000fc - 1500fc - 2000fc - 100fc$ | 10M | 20.53% |
| 4 layer | $2000fc - 1500fc - 1000fc - 1000fc - 100fc$ | 10M | 18.29% |

Table 6: Results for the convolutional neural networks student models on CIFAR-100

| Model | Architecture | Parameters | Accuracy |
|---|---|---|---|
| 1 layer | $100c - mp - 75fc$ | 10M | 21.85% |
| 2 layer | $300c - mp - 120c - mp - 1600fc$ | 10M | 25.41% |
| 3 layer | $350(c - mp)^3 - 2500fc$ | 10M | 29.37% |
| 4 layer | $375(c - mp)^2 - 110(c - mp)^2 - mp - 2000fc$ | 10M | 27.87% |

## 6 Conclusion

We did reproduce some of the experiments conducted by G.Urban et al. (1) on CIFAR-10 dataset with shallow networks. With this project, we were able to demonstrate empirically some hypotheses made in the original paper : CNN networks are always better than MLP in order to classify images. Moreover, the deeper they are, the better they are. Nevertheless, one of the critics about the original paper was the simplicity of the dataset. Thus, we did experiment these hypotheses on a more complex but still similar dataset : CIFAR-100. The different hypotheses that were true for the CIFAR-10 dataset also appeared to be true for CIFAR-100. Thus, one can consider this as a more general hypothesis : no matter the complexity of the dataset, the CNN will always be better than MLP. Moreover, raising number of layers in a CNN can improve these results.

## References


[1] G.Urban, K. J. Geras, S. E. Kahou, O. Aslan, S. Wang, R. Caruana, A. Mohamed, M. Philipose, and M. Richardson, "Do Deep Convolutional Nets Really Need to be Deep and Convolutional?" , *ICLR*, 2017.

[2] Jimmy Ba and Rich Caruana, "Do deep nets really need to be deep?", *NIPS*, 2014.

[3] Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil, "Model compression", In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 535–541. *ACM*, 2006.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network", *arXiv:1503.02531*, 2015.

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network",*arXiv:1503.02531*,2015.

[6] Train a simple deep CNN on the CIFAR10 small images dataset. *Keras*

[7] Y.Huang, Y.Cheng, D.Chen, H.Lee, J.Ngiam, Q.V.Le, Z.Chen, "GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism", *arXiv:1811.06965*, 2018