

Multi-Layer Knowledge Transfer for Neural Networks

Project by Lennard Kiehl and Roman Remme

September 2017

Contents

1	Introduction	1
2	Intermediate Layer Matching	1
3	The Models	1
4	The Dataset	1
5	Training Methodology	3
6	Results	3

1 Introduction

In this project we extend the Ideas of deep knowledge transfer between neural networks introduced in [1] by not only linking the last, but also intermediate layers of the networks. For a number of pairs of intermediate layers in the big and small network mean squared error (MSE) terms are added to the loss function.

2 Intermediate Layer Matching

Our loss function consists of four terms:

- The "hard" loss: The cross-entropy of the output distribution of the network with the correct labels
- The "soft" last layer loss: The cross-entropy of the output distribution with the "soft-targets", the output of the cumbersome model to extract knowledge from. Here, softmax layers with temperature T are used.
- The intermediate layer loss: This is the mean squared error between the activations of pairs of layers in a certain set.

The third term is the new part of our approach. A theoretical advantage is that training times should be reduced, as gradients do not have to be propagated through the whole network to reach the first layers. Also this is a

3 The Models

As a big model to distil knowledge from we used VGG-16 (see [3]). For the smaller model that was trained with the help of the big one a similar architecture was used, where the number of convolutions between pooling layers was reduced from 2-3 to just one. We also cut one of the fully connected layers (see Table 1). The similarity made it possible to compare intermediate activations at many points in the model.

4 The Dataset

We used CIFAR-10 (see [2]) as the dataset to train our models on. It consists of 50000 training and 10000 test RGB images of size 32 by 32 pixels. Each image belongs to one of ten classes. Some of the classes are a lot harder to distinguish than others, for example "Automobile" and "Truck". This makes the knowledge distillation procedure used in [1] promising.

To use the VGG architecture with these low-resolution images, they are scaled up to 224 by 224 pixels.

Table 1: **Network configurations.** The convolutional layers are denoted as "conv(kernel size)-(number of channels)" and the fully connected layers as "FC-(number of output channels)". ReLu units are omitted for brevity. The left column numbers the layers of both networks whose activations were linked in the loss functions of section X.

Network Configurations with linkable layers		
	VGG-16	VGG-7
	input (224 by 224 RGB image)	
link 1	conv3-64 conv3-64	conv3-64
	maxpool 2x2	
link 2	conv3-128 conv3-128	conv3-128
	maxpool 2x2	
link 3	conv3-256 conv3-256 conv3-256	conv3-256
	maxpool 2x2	
link 4	conv3-512 conv3-512 conv3-512	conv3-512
	maxpool 2x2	
link 5	conv3-512 conv3-512 conv3-512	conv3-512
	maxpool 2x2	
link 6	FC-4096 FC-4096 FC-10	FC-4096 FC-10
	softmax	

Table 2: **Last layer transfer results.** bla bla

Temperature	Test set accuracy
0.6	76.3 %
1	76.5 %
1.5	77.0 %
2	77.4 %
2.5	76.7 %
3	77.2 %
5	73.1 %
10	64.4 %
40	67.3 %

Table 3: **Intermediate Layers transfer results.** used last layer with temperature 2 and $\alpha = 10$

Linked layers	β	Test set accuracy
3	10	85.9%
2, 3, 4, 5	10	87.0%
2, 3, 4, 5	40	87.9%
5	10	87.7%
3, 4	10	84.8%
1, 2, 3, 4, 5, 6	10	81.2%
2, 3, 4, 5, 6	10	78.5%

5 Training Methodology

We used stochastic gradient descent with momentum 0.9 (CITATION NEEDED?) as an optimizer. We started with a learning rate of 0.004 and let it decay by a factor of 10 every 10 epochs for 25 epochs.

6 Results

References

- [1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the knowledge in a neural network”. In: *arXiv preprint arXiv:1503.02531* (2015).
- [2] Alex Krizhevsky and Geoffrey Hinton. “Learning multiple layers of features from tiny images”. In: *Technical report, University of Toronto* (2009).

- [3] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* abs/1409.1556 (2014). URL: <http://arxiv.org/abs/1409.1556>.