Warsztaty Badawcze. Unsupervised Pretraining

Bartłomiej Eljasiak, Tomasz Krupiński, Dominik Pawlak

April 22, 2021

1 Introduction

Unsupervised pre-training has proven empirically, that it helps CNN to achieve better results. Basically, unsupervised pre-training is learning weights without supervision. This idea can be achieved with lots of different methods.

One of the those is autoencoding. From the articles and papers we have found, the main conclusion is that the most popular unsupervised pre-training is CAE (convolutional autoencoder), therefore our main idea is to apply it.

In one of the usages of CAE the whole idea depends on the assumption, that it is easier for model to learn layer by layer rather than all layers at once.

Convolutional autoencoder, using layer-wise technique can works like that:

1. create one layer, that is supposed to extract features
2. create a copy of layers from input to features, transposed/reversed, therefore making the model to recreate input
3. train the model, evaluating it by similarity to the original input
4. freeze the layers, and either add another one and repeat the whole process starting at 2. or extract weights learned by this CAE model and use them in proper CNN.

The analogy is: Make a person dismantle a furniture, then make them assemble it. Then add next person, and do it again and again... Then put those people to some factory that dismantles furniture to achieve some goal, and that factory should get better results after some time, than a factory with random people after the same time.

2 Pre-Training CNNs Using Convolutional Autoencoders [1]

The authors of this paper created Convolutional Auto-Encoder (CAE) to to train the convolutional layers independently from a classification task in order to learn a new data representation. They used popular MNIST and CIFAR-10

datasets.
The network consists of several convolution and pooling layers transforming the input to a high dimensional feature map representation and then reconstructing the input using strided transposed convolutions. Authors did not use the layer-wise technique, but trained the whole encoder at once, but mentioned that it can be done both ways.
More detailed information below:

Layer Type	input	convolution	max-pool	convolution	max-pool	convolution
Filter Size	none	5x5	2x2	5x5	2x2	3x3
Channels	1(gray)|3(rgb)	100	100	150	150	200
Activation	none	scaled-tanh	none	scaled-tanh	none	scaled-tanh
Size	(H, W, C)	$(H, W, 100)$	$(\frac{H}{2}, \frac{W}{2}, 100)$	$(\frac{H}{2}, \frac{W}{2}, 150)$	$(\frac{H}{4}, \frac{W}{4}, 150)$	$(\frac{H}{4}, \frac{W}{4}, 200)$

Table 1: CAE/CNN: shared encoding layers. Channels refers to the amount of feature maps in the given layer

3 Tutorial To Pre-Training CNNs Using Convolutional Autoencoders [2]

This article teaches how to use supervised and unsupervised pre-training. The method is once again, CAE.

"Specifically, we will develop an autoencoder model that will be trained to reconstruct input data. In order to use this unsupervised model for classification, we will remove the output layer, add and fit a new output layer for classification."

The greatest part about it, are the code fragments with clear plots and explanations. We believe that those steps might create the unsupervised pre-training we need.

4 Our plans

We plan to apply this method to increase our model's accuracy. The main issue is, that our DNN model is much more complex than the architectures in papers and tutorials (more than 10 convolution layers/blocks compared to about 3), therefore we still need time to write the actual code. Anyway, the idea is to reproduce the process in the introduction part and then use those learned weights as the starting weights in our model.

References

[1] Maximilian Kohlbrenner, Russell Hofmann, Sabbir Ahmmed, Youssef Kashef, *Pre-Training CNNs Using Convolutional Autoencoders*, 2017

[2] Jason Brownlee, *How to Use Greedy Layer-Wise Pretraining in Deep Learning Neural Networks*