Machine Learning Individual Assignment #2

Maksim Gusev (S00241141)

Atlantic Technologic Institute

Abstract

This article considered the transfer learning methodology of VGG16 model to the Kaggle's GTSRB dataset. Revealed the convolutional neural network term and main aspects, as well as VGG architecture and transfer learning principles to feature extraction and fine-tuning. Showed results of transfer learning with under-fit, fit and over-fit graphics.

Keywords: Transfer learning, VGG16, GTSRB dataset, Convolutional Neural Network, Machine Learning.

1 Introduction

This article is a part of the Machine Learning (COMP09012) module assignment taken at the Atlantic Technological University. The main purpose is to perform the transfer learning via pre-trained convolutional neural network model VGG16 and retrain it to recognize traffic signs from Kaggle dataset (GTSRB – German Traffic Sign Recognition Benchmark).
In second chapter of the article, considered the convolutional neural network (CNN or ConvNet), the VGG16 pretrained model and methodology of transfer learning. The third chapter is dataset analysis and model training. The last fourth chapter is results and conclusions.

Figure 1 GTSRB dataset (Meta).

2 Methodology

Started methodology from the explanation of convolutional neural network (CNN or ConvNet). CNN – class of neural networks, which specialized on the data processing, which have mesh topology structure (e.g. photos, images). A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid, where each cell contains visual data: brightness and colour. As well as neurons in the brain only responds to stimulus in a limited region of the visual field, called the receptive field in the biological vision system, each neuron in a CNN also processes data only in its receptive field. The layers are arranged in such a way that they detect simpler patterns first (lines, curves, etc.) and then more complex patterns (faces, objects, etc.).

2.1 VGG16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the article “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves an accuracy of 92.7% - top 5 when tested on ImageNet in the task of recognizing objects in an image. Model was trained on dataset consists of over 14 million images belonging to 1000 classes.

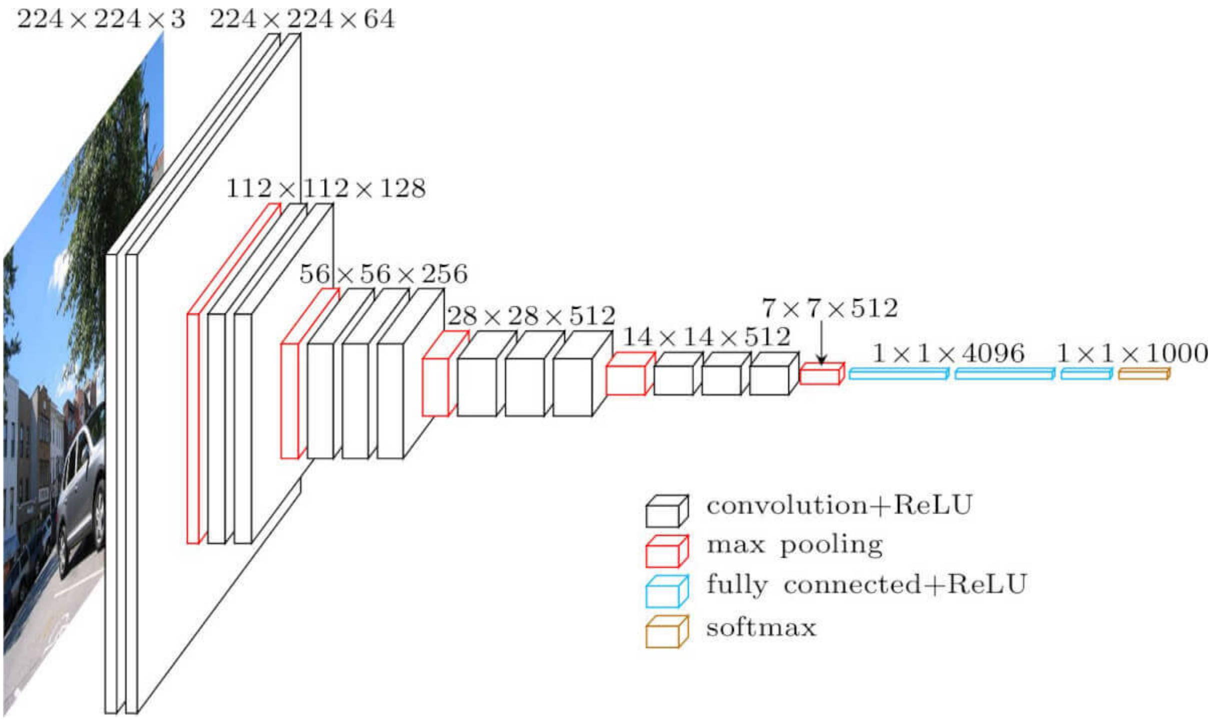

Figure 2: Architecture of VGG16 model

2.2 Transfer learning

Transfer Learning allows using the experience gained in solving one problem to solve another, similar problem. The neural network is first trained on a large amount of data, then on the target set. To conduct the Transfer Learning process needs two parts: availability of suitable pre-trained model and model reassignment via feature extraction or fine-tuning.

A pretrained model is one that has been created and trained by someone else to solve a problem similar to ours. In practice, someone with large computing resources, they construct a large neural network to solve a specific problem, train it on a large data set (Big Data), such as ImageNet or Wikipedia Corpus. So, for example, VGG19 has 143.667.240 parameters and is used for image classification. By “open” we mean that the model is made public and can be freely used.

In Deep Learning architectures, the initial layers learn general information, and the layers at the last level learn more specific features. Many models are trained on all kinds of situations, for example ImageNet contains 1 million images with 1000 classes, so there is no need to change the overall picture that the current model sees. Instead, it is more useful to supplement it with new specific features, retraining only the last layers to repurpose it for your own needs. By affecting more overfitting layers, the risk of overfitting increases.

Feature extraction uses the representations obtained by the previous model to extract features from new samples, which are then run through the new classifier. This method simply adds a classifier that will be trained from scratch on top of a pre-trained model to solve the objective function. Convolutional neural network (CNN) architectures usually consist of two parts: convolutional and fully connected. For feature extraction, the convolutional part remains unchanged. Whereas Fine Tuning captures the last few convolutional layers.

Both methods of model reassignment can improve the accuracy of the model, but provided that there is enough data. Otherwise, the network will not "feel" the changes from the new data set and will not be able to reprofil.

Feature extraction is used when the problem being solved of the past network is similar to the target one. But if there are significant differences, then additional training is used, which is more expensive from a computational point of view.

3 Dataset analysis and model training

Data obtained from the real world can be vague and imbalanced. Some road signs, a driver could see every crossroad. Other sings could see only in a few places. Kaggle's GTSRB dataset imitated this imbalance.
If the total number of all images of train dataset part is 39209 images, separated to 43 classes. Major part of this parts sliced to 17 classes and has 28710 images or 73.2% of all data. It means, 26 classes have only 10499 images or 26.7%. One class could take from 5.7% to 0.5% from the training dataset.

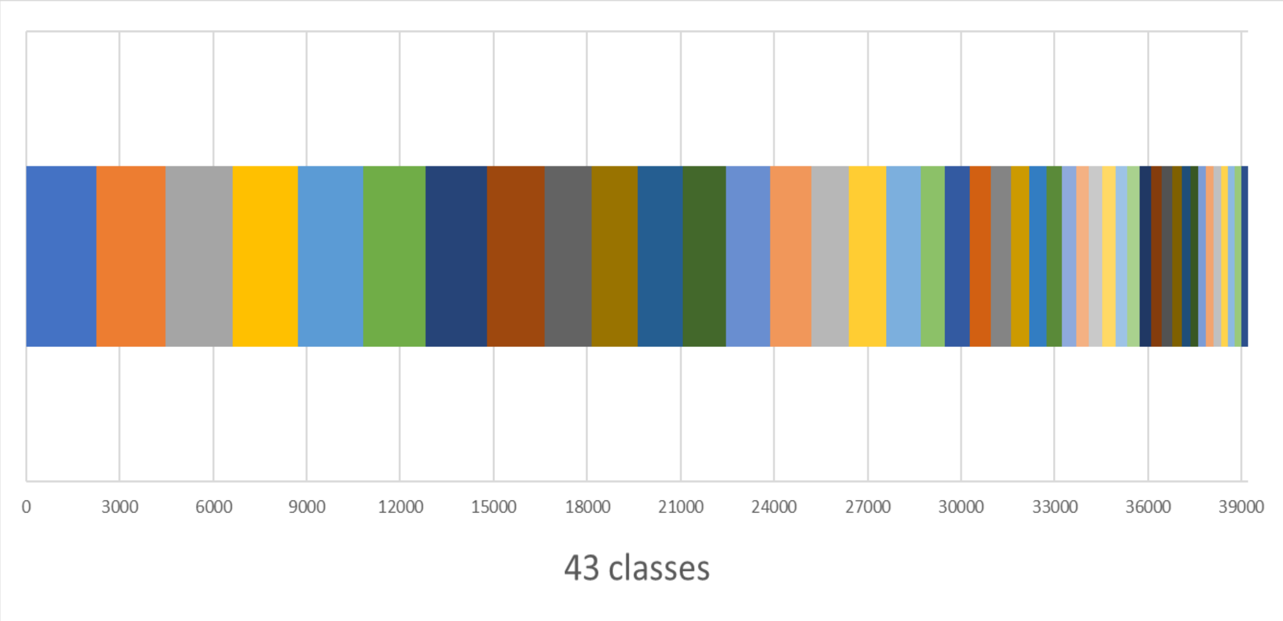

Figure 3: Dataset analysis data.

Kaggle said that, the default input for the VGG16 model is 224x224. However, the author reduced input of images to 40x40. The processing and calculation power of student equipment is limited. The rescaling, cropping and centering of all images was conducted through the "ImageDataGenerator" function.
The author decided to choose the Feature extraction instead of Fine-Tuning. Because of the VGG16 was trained on the similar task and the application of Fine-Tuning required more time and calculation and pre-processing power. All the solutions and functions could be observed in GitHub (https://github.com/GusevPortfolio/Machine-Learning-1-assignment-).

4 Results and conclusions

After the data pre-processing and model training, the model showed the next results:
Education in 10 Epochs: Train Accuracy = 90.8% / Test Accuracy = 76.6%
Education in 30 Epochs: Train Accuracy = 94.4% / Test Accuracy = 78.5%
Education in 50 Epochs: Train Accuracy = 98.0% / Test Accuracy = 78.6%

The model after the training in 10 epochs showed the best results in 76.6% with 90.8%. That results were the initial point. The increasing of accuracy and training loss reduction is a next step.
The model after the training in 30 epochs showed the best results in 78.5% with 94.4%. That +3.6% for the train accuracy and +1.9% for the test accuracy.
The model after the training in 50 epochs showed the best results in 78.6% with 98.0%. That +3.6% for the train accuracy and only +0.1% for the test accuracy.
That showed us that the most suitable quantity of epochs is 30 (fit). The model with 50 epochs increased the train accuracy, however only 0.1% of the test. That are the direct signs of an overfit model.

To increase the accuracy, reduce the training loss and avoid the overfitting, the model is required the more training data or more balanced dataset.

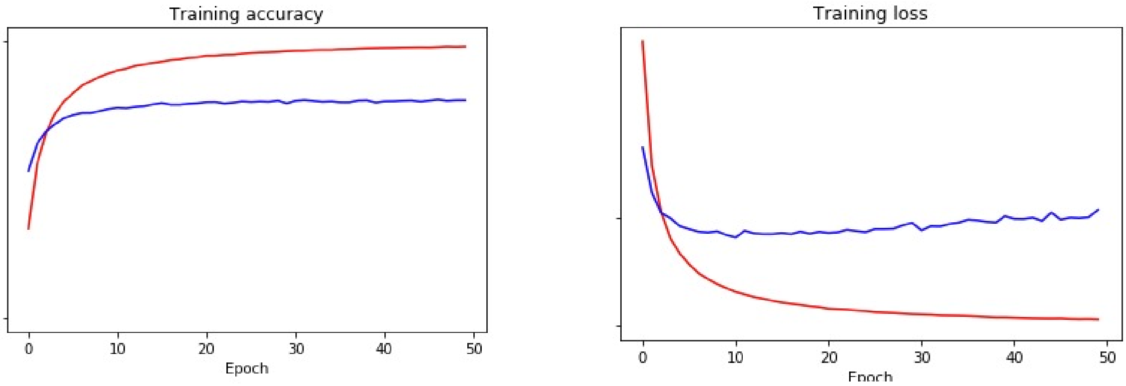

Figure 4: Accuracy and loss graphics (red – training, blue – validation).

References

[Goodfellow et al., 2017] Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press, ISBN: 9780262337434. http://www.deeplearningbook.org.
[Metz, 1978], CE (October 1978). "Basic principles of ROC analysis" (PDF). Semin Nucl Med. 8 (4): 283–98. doi:10.1016/s0001-2998(78)80014-2. PMID 112681.
[Haixiang et al., 2017] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications, 73:220–239, 2017. https://doi.org/10.1016/j.eswa.2016.12.035
[Brodeson et al., 2010] Brodersen, K.H.; Ong, C.S.; Stephan, K.E.; Buhmann, J.M. (2010). The balanced accuracy and its posterior distribution. Proceedings of the 20th International Conference on Pattern Recognition, 3121-24.
[Sasaki, 2007] Yutaka Sasaki. The truth of the F-measure. Teach Tutor Mater, 01 2007, https://www.cs.odu.edu/~mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf
[Shmilovici, 2005] Shmilovici, A. (2005). Support Vector Machines. In: Maimon, O., Rokach, L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA. https://doi.org/10.1007/0-387-25465-X_12
[Wu et al., 2004] Wu, Lin and Weng, "Probability estimates for multi-class classification by pairwise coupling", JMLR 5:975-1005, 2004.
[Nagpal, 2022] Anuja Nagpal; L1 and L2 Regularization Methods, Explained, Jan. 05, 2022
[Sci-Kit Learn Package, 2022] Sci-Kit Learn Package, scikit. Available at: https://scikit-learn.org/stable/ (Accessed: November 8, 2022).
[Simple Imputer] Sklearn.impute.SimpleImputer, scikit. Available at: sklearn.impute.SimpleImputer (Accessed: November 10, 2022).
[Ordinal Encoder] Sklearn.preprocessing.OrdinalEncoder,scikit. Available at: sklearn.preprocessing.OrdinalEncoder (Accessed: November 10, 2022).
[ML Group Code, 2022] Stanley, S., McGovern, D. ML Group Code, GitHub. Available at: https://github.com/DavinMcGowan/Machine-Learning-Group-Assignment-/blob/main/ML_Assignment_Group_A.ipynb (Accessed: November 8, 2022).