Project Title**: Transfer Learning for Stanford cars dataset**

Group Members: Grebenkin Ivan, Ramil Gizatullin
Repository: https://github.com/Lolik111/transfer-learning-tf
Course: Adv. Machine Learning
Innopolis University

Project Idea

The concept of our work is that we are going to apply one of two popular neural network concepts within the Transfer Learning topic. In practice, the training of whole CNN is usually not done from scratch with arbitrary initialization. The reason is that it is usually not possible to find a set of data of sufficient size required for the network of the required depth. Instead, CNN is preliminarily trained on a very large data set, and then the weights of the trained CNN are used either as an initialization or as a highlight of the distinguishing features for a particular task.

So the first concept can be considered is Densely Connected Convolutional Networks[1]. The whole point of its architecture is in Dense Block. It consists of the following basic elements: BatchNorm -- ReLU -- 1x1 Convolution -- BatchNorm -- ReLU -- 3x3 Convolution. And each base element pushes its features to all the next elements in the block, thus, we achieve a dense-connectivity between all basic elements. The features are concatenated, rather than summed, so the number of channels grows linearly to the number of filters. But this is compensated by the fact that the basic elements themselves are more narrow. Also, in order to reduce the growth of the number of channels after each Dense Block, we use a Transition Block consisting of 1x1 Convolution (number of filters can vary) and 2x2 average Pooling.

The second one is Xception network[2]. Its architecture is based on the assumption that these two types of information can be processed in series without losing the quality of the network, and decomposes the conventional convolution into pointwise convolution (which handles only inter-channel correlation) and the spatial convolution (which only processes spatial correlation within a single channel).

The second is more interesting for us in terms of immersion in the Transfer Learning topic. But at the same time it has more computational complexity. We will try to build two experiments. And if the second one does not converge sufficiently quickly during training, we will focus only on the first one.

Dataset Information

We are going to use Stanford cars dataset [3] containing 16,185 images of 196 classes of cars. Each class will be splitted roughly in a 70-30 train/test split.

Timeline

Team will work according following timeline:

- Keras model to tensorflow convertation (27.03 - 28.03)
- Developing input pipeline (28.03 - 01.04)
- First iteration of training and evaluation, parameters tuning (01.04 - 12.04)
- Second iteration of training and evaluation (12.04 - 23.04)
- Report writing, final submission and preparation of presentation (23.04 - 25.04)

References

1. https://arxiv.org/abs/1608.06993
2. https://arxiv.org/abs/1610.02357
3. http://ai.stanford.edu/~jkrause/cars/car_dataset.html