Fall 2021: ELG7186
Assignment 3

Due: Monday, November 8th, 2021, 11:00pm in Virtual Campus
University of Ottawa - Université d'Ottawa

Jochen Lang

1 CNNs, Keras API and Regression

This assignment will give you a chance to familiarize yourself with CNNs and with the different techniques for monitoring and controlling the training process in tensorflow. In this assignment, we will look at weed leaves, in particular, we want to predict the number of leaves of the plant. The more leaves, the more the weed has grown.

You must use Keras with the tensorflow backend, i.e., the package `tensorflow.keras`. For this assignment, you may use other tensorflow packages and scikit-learn, scikit-image or pandas but *not* other deep learning frameworks, e.g., pytorch, mxnet etc.

2 Data Preparation

The data for this assignment are plant images at different resolution captured with a variety of cameras. There are images showing plants with approximatelty 1,2,3,4 and 6 leafs. The images are part of a Leaf counting dataset by Teimouri et al. [1] which can be downloaded from the Aarhus University, Denmark

`https://vision.eng.au.dk/leaf-counting-dataset/`

. However, you must work with the subset of images posted on BrightSpace as `training.zip` and `testing.zip`. There are 200 images for each of the 5 classes. As Figure 1 shows, there is a great variety of plants and image conditios. The dataset is split into a training and a testing set where there are 180 images per class for training and validation; and 20 images for testing.

Figure 1: Examples of plant images with 1, 2, 3, 4 and 6 leaves, respectively.

3 Basic Transfer Learning

For this assignment, you are asked to use the Keras implementation of VGG-16 as a starting point. Have a look at the transfer learning example jupyter notebook `mnistVGG.jypnb` to get started.

3.1 Classification Network [3]

Using the first 2 blocks of VGG-16 add extra Keras layers to create your own version of a CNN network for the classification of the images according to the number of leaves in the plant images. Note that there will be 5 classes. The last layer from VGG-16 will be `block2_pool` and you are allowed to add no more than *five* fully connected or convolutional layers to the network including the final output layer. You can use as many pooling, flattening, 1×1 convolution layers, etc. as you wish but do *not* use any regularization. Train this simple network on the training set while monitoring convergence on the validation set. As input to the model use images of size no larger than 128×128. Note, it is highly recommended to use even smaller input images to try things out. You are not expected to fine-tune the initial VGG layers. Print your learning curves for training and validation. Give the confusion matrix of your network on the training including validation and testing data sets.

3.2 Basic Transfer Learning: Regression Network [2]

Repeat the steps of Question 3.1 but turn it into a regression problem, i.e., your network needs to output a single float value ranging between 0 to 6 corresponding to the number of leaves. Again, please print your learning curves for training and validation. You are not expected to fine-tune the initial VGG layers. Give your mean squared error on training including validation and testing data sets.

3.3 Discussion [1]

The size of the training data is quite small. Discuss based on your learning curves if overfitting is occurring with your networks in Question 3.1 and Question 3.2.

4 Improving the Model

Regularization and data augmentation are common strategies to deal with small datasets.

4.1 Regularization [1.5]

Incorporate two regularization methods (e.g., Batch Normalization, Dropout, Weight Normalization etc.) into your layers of the network. Please pick the model from Question 3.1 and Question 3.2 that performs better. You are not expected to fine-tune the initial VGG layers. Again, please print your learning curves for training and validation and print the corresponding metrics for your model.

5 Data Augmentation [1.5]

Perform data augmentation for training the same model as in Question 4.1. You are not expected to fine-tune the initial VGG layers. Again, please print your learning curves for training and validation and print the corresponding metrics for your model.

5.1 Discussion [1]

Discuss based on your learning curves and final metrics in Question 4.1 and in Question 5, how large a improvement can be observed from regularization and data augmentation.

6 Submission

You will need to submit your solution in a Jupyter file, do *not* submit the image data. Make sure you have run all the cells. All text must be embedded in the Jupyter file, I will not look at separately submitted text files. If your Jupyter file needs a local python file to run, please submit it as well. Assignment submission is only though Virtual Campus by the deadline. No late submissions are allowed, you can submit multiple times but only your last submission is kept and marked.

References

[1] N. Teimouri, M. Dyrmann, P. R. Nielsen, S. K. Mathiassen, G. J. Somerville, and R. N. Jørgensen, "Weed growth stage estimator using deep convolutional neural networks," *Sensors*, vol. 18, no. 5, 2018.