Machine Learning for Agricultural Applications

Assignment 9

Prof. Dr. Niels Landwehr
Dr. Julian Adolphs

Summer Term 2020

Release: 02.07.2020
Discussion: 09.07.2020

Task 1 – Transfer Learning [25 points]

Transfer learning is a machine learning technique in which a network, that has been trained to perform a specific task, is re-used as a starting point for another similar task. Transfer learning is usually chosen for tasks where your dataset has too little data to train a full-scale model from scratch. Another advantage is that starting from a pre-trained model can dramatically reduce the computation time required compared to training a model from scratch. The common transfer learning workflow is:

- Take layers from a previously trained model.
- Freeze them, to avoid destroying any of the information they contain during future training rounds.
- Add some new, trainable layers on top of the frozen layers. They will learn to turn the old features into predictions on a new dataset.
- Train the new layers on your dataset.
- Fine-tuning (optional step): which consists of unfreezing the entire model you obtained above, and re-train it on the new data with a very low learning rate. This can potentially achieve meaningful improvements, by incrementally adapting the pre-trained features to the new data.

Read the tutorial `https://keras.io/guides/transfer_learning/` and run the program on your own machine (if you have a GPU) or on Colab.

If you do not train on the cats_and_dogs data set, how good does your pre-trained model work?

What is data augmentation and what is the benefit?

How is the test accuracy affected, if you leave out data augmentation?

What is the difference between the accuracy of the base-model after training and the fine-tuned model?

Plot the accuracy during training and fine-tuning.

Solution: ex9_1.py

```
Without training of the base model, accuracy:                45.79 %

Without data augmentation, trained base model (20 epochs):   96.82 %
Without data augmentation, fine-tuned model (10 epochs):     97.59 %

With data augmentation, trained base-model (20 epochs), acc: 96.78 %
With data augmentation, fine-tuned model (10 epochs), acc:   98.07 %
```

Task 2 – Plant Disease Detection [25 points]

In this task, you will download the "New Plant Disease Dataset" from Kaggle (`https://www.kaggle.com/vipoooool/new-plant-diseases-dataset/kernels`). You need a Kaggle account, but it's free of charge (and useful anyway).

Your task is to perform transfer learning on this dataset. Although the dataset includes augmented data pictures (only flipped and rotated), you should try, if you can improve the model by data augmentation. The dataset is already split into train, validation and test data.

Repeat the transfer learning steps listed in Task 1. As VGG16 is a huge model, including fine tuning might not be possible depending on your available memory. As pre-trained model use VGG16 with `imagenet`-weights.

Plot the progress of the validation accuracy during training.

Now change to the smaller model `MobileNet`, again with `imagenet`-weights. Compare the validation accuracy of the model (without fine-tuning) with the outcome of the VGG16-model.

As MobileNet is much smaller than VGG16, include fine-tuning and try how far you can increase the validation accuracy.

Figure 1: Leaf of a healthy tomato plant (left) and two examples of tomato leafs with different diseases (center and right).

```
Solution:   ex9_2_VGG16.py  and  ex9_2_MobileNet.py

Without data augmentation after 5 epochs:   95,93%,   800 sec = 13 min
with data augmentation after 5 epochs:      93,65%,  3600 sec = 60 min

MobileNet:
20 epoch training, val_acc:                 71.90 %,  3000 sec = 50 min
20 epochs training + 10 epochs fine-tuning: 90.31 %,  5500 sec = 90 min

VGG16:
20 epochs training, val_acc:                95.79 %,  3500 sec = 58 min

Better model gives higher accuracy in shorter time without fine-tuning!
```