Lab V - Artificial Neural Networks

module V

DT4031, *Applied Machine Learning*
Jens Lundström

Introduction & Assignment

In this lab you will be using the knowledge acquired from module V, Artificial Neural Networks (ANN).

The tasks in this lab are recognized from lab 1 and are from a general perspective similar, to train and build a model able to recognize hand-written digits. However, this time we will ANN's which are able to learn features rather than apply manual feature engineering (using the HOG transform) and *k*-NN as we did in lab 1. When designing, training, and evaluating ANN's we have to be aware of *overfitting* which is a concept we have discussed throughout the course. ANN's are generally very flexible and if not carefully designed and used may overfit easily, this is the focus of the lab – to design and apply methods to avoid overfitting (which occurs when learning the digits present in the training set rather than learning the general features of digits and therefore resulting in a higher generalization error).

Lab report instructions

First task - get a general feel for how the number of neurons in a layer and the number of layers affect the ability to learn and generalize. Explore this (approximately 15 minutes) by browsing to:

https://playground.tensorflow.org/

1. Select the *circle* data
2. Click play button to start learning and study how decision boundaries are formed and their shape
3. Remove/add neurons, retrain and study the results
4. Remove/add layers (and their individual neurons) and study the result
5. Select the *spiral* data and repeat steps 2-4
6. Formulate the lessons learned after you have been playing around with the tool

As you may have noticed from the *playground*, neural networks tend to be very good at fitting data, however the real challenge is not to fit data, instead we would like to build a model able to generalize well. Therefore, it is all about finding a compromise between a model having *not enough capacity* and a model having *too much capacity*. Unfortunately, there is no equation nor rules guiding us how to select this, we need to explore this by experimentation.

Second task, open the Notebook which is based on Tensorflow (the neural network API that we have been discussing in class), go through it by studying and running each cell. As you can see, the ANN model is quite complex. Make a graphical sketch of the network and report on the number of weights trained, also answer the question: what is happening with the validation loss just after a few epochs? Make sure to save your plots (and/or save the loss and accuracy curves in unique variable names).

Third task, replace the model by a much simpler just having one hidden layer (with 4 neurons). Remove some layer(s) and reduce the number of neurons. Run the *whole* Notebook again and study the results. Describe your conclusions after studying generalization properties (validation accuracy and validation loss in relation to training loss). Would you say the model is having *too much capacity* or *not enough capacity?*

Fourth task, go back to the first (more complex) model but add random dropout of neurons in some layers by the model code:

```
model_deeper_with_dropout = tf.keras.models.Sequential([
  tf.keras.layers.Dense(512, activation='relu', input_shape=(28*28,)),
  tf.keras.layers.Dropout(0.5),
```

```
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(64, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

Dropout is a form of *regularization* aiming to reduce the complexity of the model (and hopefully therefore also avoiding overfitting). Again, run the *whole* Notebook and study the results, make a plot with all the previous models training/validation losses *in the same plot*. Describe your conclusions.

The lab solution shall be documented in the form of a written report. As a data scientist it is your task to explain what you have done in a detail such that the solution is reproducible, the reasoning behind the decisions, the outcome (results) and the conclusions you have drawn from such results. It is suggested to have the following lab report structure: Introduction & Motivation; Methods & Solutions; Results; Conclusions & Discussion; Appendix. Keep the number of pages in the lab report limited (excluding *Appendix* which should contain Notebook code and figures not directly linked to the result).

Lab assistance

Feel free to reach out to Guojun or Zeinab if you have questions regarding the lab.

If you're experiencing the training to take too long time you could test the code on one of the GPU-equipped servers at Halmstad University for students (currently not tested), please use:

london.hh.se, bermingham.hh.se, liverpool.hh.se, bristol.hh.se
SSH on port 20022 from anywhere.
Also accessible with Microsoft Remote Desktop (RDP) on port 3389 from student wired network.