Teachers: J. Hennebert, M. Melchior
Assistants: C. Gisler, M. Willi, Y.-I. Beffa

Practical Work 04 – 18/03/2021
Model Selection

Objectives

The main objectives of this Practical Work for Week 4 are the following :

a) Play through an example of overfitting and determine the optimal model complexity. Use 5-fold cross validation.

b) Optional : Compose the confusion matrix from the scores obtained from a classification model, compute the different performance measures and learn the characteristics these are capable of highlighting.

c) Use sklearn to do hyper-parameter tuning in stages for MNIST. Determine one best model (without using the test set) and report the final test score (test error computed by using the test set).

Submission

- **Deadline** : Wednesday 31 March, 10am
- **Format** : For exercises 1 and 3 a Jupyter notebook including comments and results. For the optional exercise, you can also submit your notebook including the results.
- Please **only one submission per group** - with a clear indication of the group id.

Exercise 1 Bias Variance Tradeoff

The objective of this exercise is to build a classification system to predict whether a student gets admitted into a University or not based on their results on two exams [1].

You have historical data from previous applicants that you can use as a training set. For each training example i, you have the applicant's scores on two exams $(x_1^{(i)}, x_2^{(i)})$ and the admissions decision $y^{(i)}$. Your task is to build a classification model that estimates an applicant's probability of admission based on the scores from those two exams.

In the notebook `overfitting_stud.ipynb`, you'll find the code to load the data and further instructions.

a) Construct a dummy predictor that does random predictions. Show numerically that it produces a performance equal to 1/#classes - i.e. here equal to 50%.

b) Construct different (polynomial) models of different complexities (different degree number of parameters). Train these models with the training set and use the trained parameters for doing predictions. Measure the error rate on the training and the test set.

   Remark : Do the programming so that you can easily change the input dataset.

c) Determine the model best suited for the problem at hand and justify why it is the best model.

d) For all this use two different versions of the data :

   i) First version : `scores_train_1.csv` and `scores_test_1.csv` for training and testing, respectively.

   ii) Second version : `scores_train_2.csv` and `scores_train_2.csv` for training and testing, respectively.

Exercise 2 Optional : Performance Measures

Let's assume we have trained a digit classification system able to categorise images of digits from 0 to 9.

After training, the system has been run against a test set (independent of the training set) including $N_t = 10'000$ samples. The output of the system is provided by a softmax layer and are given as estimates of the a posteriori probabilities $P(C_k|\mathbf{x})$ for $k = 0, 1, 2 \ldots, 9$.

In the file `confusion_a.csv`, you find the output of a first system A with the a posteriori probabilities $P(C_k|\mathbf{x})$ in the first 10 columns and with the ground truth y in the last column.

a) Write a function to take classification decisions on such outputs.

b) What is the overall error rate of the system ?

c) Compute and report the confusion matrix of the system A.

d) What are the worst and best classes in terms of precision and sensitivity (recall) ?

[1] Data source : Andrew Ng - Machine Learning class Stanford

e) In file `confusion_b.csv` you find the output of a second system B. Which of the systems, A or B, performs better in terms of error rate and F1 ?

You can use the jupyter notebook `confusion_matrix_stud.ipynb`.

Exercise 3 Model Selection

Here you carry through a model selection procedure by using sklearn's NN-modeling and training capabilities.

As dataset, you will use MNIST - a dataset that you already know well from the previous exercises so you can concentrate on the modeling part.

In the file `model_selection_stud.ipynb`, you find the functionality to load the data and the skeleton for how to proceed. Stage-wise proceed as follows :

a) Learn how to use the sklearn functionality : How to configure the MLP classifier class and how to compute the error rate (accuracy) and the confusion matrix.

b) Prepare your sets by splitting into 3 parts : training-validation-testing sets. Use the training and validation sets for tuning and selecting your models. Use the testing set for the last step.

c) Run a first training with a MLP without hidden layer, with hyper parameters as given in the notebook.

d) Tune learning rate, batchsize and number of epochs to find a best MLP without hidden layer for the given classification task.

e) Add a single hidden layer and find now the best MLP for the given classification task. Also explore how the model performance (test error) depends on the model complexity.

f) Finally add at max three hidden layers and find the best MLP for the given classification task.

g) Now use the test set (and only now) to report the performance of your best model.

Exercise 4 Optional : Review Questions

a) Explain the terms bias and variance. What are the factors that make the bias larger or how can it be made smaller ? What factors lead to a large variance or how can it be reduced ?

b) Why is the training error an increasing function of the split ratio (fraction of samples used for training) ? Why is the validation error a decreasing function of the split ratio ?

c) Describe in words how you construct a confusion matrix. How can you compute the accuracy from it ? How does this relate to the error rate ?

d) Describe the terms class accuracy, recall and precision. Describe typical situations where you would like to obtain a high recall or a high precision, respectively.