Department of Computer Science, Electrical and Space Engineering
Luleå University of Technology

D7041E "Applied artificial intelligence"

(Please report any found inconsistencies to Evgeny as soon as you find them)

LAB 1: From NN to CNN

1. Introduction

In this lab we will work with data-driven classification problem. The task is to assign an input data one label from a finite set of categories. The challenge here is that there is no model describing each category available in advance. That is we cannot hard-code it in an algorithm, instead we need to learn this model. In this lab you will work with one instance of this problem – the image classification problem. In particular we will classify hand-written digits from the publicly available dataset MNIST. In this lab you will use a Python project ImageClassification, which is available in Canvas.

2. Nearest Neighbor classifier: skills of working with datasets and hyperparameters

The first approach you will test has little to do with learning, however we will use it for mastering the skills of working with datasets as well as a baseline for benchmarking performances of more advanced approaches.

Task 1. MNIST data set:

One commonly used toy image classification dataset is the MNISTdataset. This dataset consists of 60,000 tiny images that are 28 pixels high and wide. Each image is labeled with one of 10 classes ("0", "1", "2", "3", "4", "5", "6", "7", "8", "9"). These 60,000 images are partitioned into a training set of 50,000 images and a test set of 10,000 images.

1. Download ImageClassification Python project from Canvas, unzip it and import into the Eclipse´s workspace.
2. In your working directory (where you unzipped the Python project) create a new folder "Datasets". In "Datasets" folder create MNIST_pkl subfolder.
3. Download to "MNIST_pkl" folder the dataset (in Python-friendly format) from http://deeplearning.net/data/mnist/mnist.pkl.gz
4. (Basic Python skills). Complete the code of the PlotSample function, which takes the number of a sample as an argument and plots it using the functionality of the **matplotlib** library.
5. For a sample of your choice determine (and plot) the distribution of values across all features.
    - What are the minimum, maximum, mean and standard deviation values?
6. (EXTRA – if you have time) Following the explanations in http://cs231n.github.io/neural-networks-2/ perform Principle Component Analysis on MNIST data se and plot several images reduced to 100 highest principle components.

Task 2. 1-NN classifier.

Class NearestNeighborClass implements a special case of the K-Nearest Neighbors classifier (i.e. 1-NN) using L1 norm. The 1-NN classifier takes a test image, compares it to every single one of the training images, and predicts the label of the closest training image.

1. Run the provided code for 1-NN classifier. What is the accuracy of the method?
2. Modify the code of the NearestNeighborClass so that it uses L2 norm for classification. How the classification accuracy has changed now?

3. Extend the code of the NearestNeighborClass with necessary methods, which implement k-NN klassifier. That is instead of finding the single closest image in the training set, you will find the top **k** closest images, and have them vote on the label of the test image.

Task 3. Hyperparameters, cross-validation

The k-NN classifier requires a setting for k as well as there is an option of choosing one or another distance function (e.g. L1 or L2 norm). These choices are examples of so called **hyperparameters**, which needs to be optimized for best classification accuracy. **In machine learning one should avoid adjusting these parameters based on the test data!** This is rather obvious since in this case your algorithm will be optimized for the particular data set. A typical approach for choosing test parameters is to have yet other data set for validation purposes. In some cases, when the dataset is small, a technique called cross-validation is used. It is illustrated in the figure below.

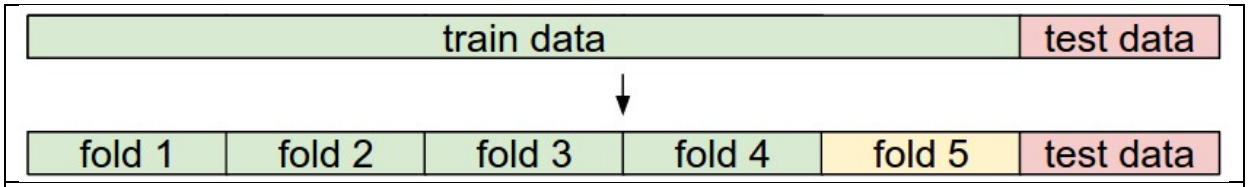

Figure. Data split for cross-validation

The training set is split into folds. In the figure above 5 folds are displayed. One fold for example fifth (shown by yellow color) is chosen for validation while other four for training. In cross-validation one iterates the validation fold when choosing the optimal values for hyper parameters. Typical number of folds you can see in practice would be 3-fold, 5-fold or 10-fold cross-validation.

In the very end once the model is trained and the best values of hyperparameters were determined, the model is evaluated a single time on the test data (red).

1. Extend the code of the NearestNeighborClass with necessary methods to find the best value of k using 3-fold cross-validation method. Use L2 norm as the distance function.

2. What is the classification accuracy on the test data for the best value of **k**?

Task 4. Multi-layer perceptron and backpropagation

Multi-layer perceptron was one of the first successful neuro-inspired computational models. In its original version, however, it was very difficult to parametrize and optimize, as such its performance was in many practical cases sub-optimal. Backpropagation algorithm was a breakthrough solution, which paved the way to the modern development of the artificial neural network technology and deep learning. The goal of this task is to understand the fundamentals of the backpropagation algorithm and study classification performance of the multi-layer perceptron.

1. Understand the implementation structure of the multilayer perceptron in Backpropagation source folder of the ImageClassification Python project.
    a. Be able to explain the principle of backpropagation algorithm;
    b. Be able to explain the meaning and the role of the Softmax function;
    c. Be able to name typically used non-linear output functions and implications of choosing one or another for implementation.
2. Write an equation for calculating Deltas on layer **i:**
    a. Hint. In essence this equation is a dot product of weights matrix on layer **i** and Deltas on layer **i+1** multiplied by the derivative of the output on layer **i**.
3. Run the code with the suggested configuration of the hyperparameters: number of epochs = 70 and learning rate =0.05. What is the classification accuracy?
4. Run the code with Learning rate =0.005 and Learning rate =0.5. Explain the observed differences in the functionality of the multi-layer perceptron.
5. Extend the code implementing the ReLU output function. Run the perceptron with the suggested by default configuration of hyperparameters: number of epochs = 70 and learning rate =0.05. What is the classification accuracy?

Task 5. Deep learning with Convolutional neural networks

Convolutional neural network is an example of neuro-inspired learning architecture tailored to the processing of images. In this task you will try using an existing framework for designing convolutional neural networks. We will experiment with the mainstream optimal implementation of CNN on GPU machines from NVIDIA called DIGITS.

1. The DIGITS framework from NVIDIA is installed on a GPU equipped machine in AWS cloud.
2. Open the following page in your web-browser to access the GUI of the running DIGITS framework: **52.8.116.64**
    a. **NOTE! Due to cost reasons the access to this machine is limited to scheduled hours. The schedule will be announced separately.**
3. Follow the "Getting Started" guide from https://github.com/NVIDIA/DIGITS/blob/master/docs/GettingStarted.md and familiarize yourself with the environment.
4. Train a CNN on the MNIST dataset use as suggested by the guide LeNet CNN's architecture
5. Be able to explain the functionality of different structural elements of the LeNET CNN
6. Monitor the dynamics of learning process (the development of accuracy and Loss functions with the number of epochs) for different values of learning rate and solver types.
7. Prepare a test dataset of 10 - 20 images with samples of written *you* digits (you choose the digits to test). Use "Classify One" functionality of DIGITS.
    a. What is the classification accuracy of the trained CNN on your dataset?

Congrats, you have just become familiar with fundamentals of the cutting edge learning techniques for data-driven classification! Well done!