

# Assignment 1: Artificial Neural Network

---

February 15, 2022

## 1 INTRODUCTION

A grocery-robot has no knowledge of what kind of product an apple is. However, it can determine the features of a product using its sensors. Based on features such as the shape, color, weight, and wrapping of a product, it can recognize specific classes.

The problem can be simplified into **classification**, which can be performed by an Artificial Neural Network (ANN). The ANN could be able to recognize products by detecting a number of separate features:

- Roundness
- Color
- Weight
- etc ...

Based on the value of these features, the ANN could then categorize the product into various classes such as:

- Fruit
- Meat
- Candy
- etc ...

## 1.1 ASSIGNMENT

You and your team will have to code a fully functioning neural network. You will also have to write a report explaining what you did. Please include answers to all the questions from Section 2 in a **structured way** by numbering your answers. Where possible, include plots and graphics to support your answers and explain these properly.

You will create a network that can classify an input of **10 features** into one of **7 classes**. The data you will need to train and test your network can be found on Brightspace. There are three files containing comma-separated values:

- `features.txt`: 7854 samples of 10 features
- `targets.txt`: 7854 target classes of these samples
- `unknown.txt`: 784 samples of 10 features, with no classes known

It's your job to load the data, divide it into training sets and use it to train a neural network. Finally, you should use your network to recognize the classes of the unknown samples. You cannot know how well your network performed on these samples until after the deadline. Your grade will be based partly on the performance of your network on this set.

You are expected to write the entire network from scratch: **a feed-forward neural network** with a **back-propagation training algorithm**. Write clear code, indent where needed and add comments to clarify what is happening. We have provided you with a Jupyter Notebook that can efficiently create and train neural networks. You will compare the results of this notebook with your own neural network. The following paragraphs contain instructions for the assignment, read them very carefully.

## 1.2 OUTPUT

The main method of your neural network should generate the file **Group\_XX\_classes.txt** (XX is your group number) containing the resulting classes of the unknown samples. It should be outputted as comma-separated values at top level folder (called ann) as **ann/Group\_XX\_classes.txt**. It must be a **text file (.txt)** with values in a single line. Example: 7,2,4,5,6,1,3,1, . . . .

Note that grading of this part of the assignment will be automated. This means that deviation from the required format will result in 0 points for this component.

## 1.3 DELIVERABLES

Create a report to answer the questions from Section 2 . Besides this, you have to clean up your code and deliver it as a compressed archive. Have a main file which starts (and contains

comments to explain) the entire training sequence. The following files should be delivered by replacing XX with your group number (e.g. 12, 31, 03):

- **Group\_XX.zip**: Compressed file for submission (zip). Should contain the following:
  - **Group\_XX\_report.pdf**: The report as **PDF document (.pdf)**. Aim for 5 pages, your report must be no longer than 6 pages (excluding a reference list). This is a hard limit, extra pages will not be graded.  
*Note*: your code will not be directly graded - do not rely on it to support your answers, but it will be checked for irregularities.
  - **Group\_XX\_code**: All your commented scripts and functions you created for neural network should be inside this **file folder**. Refer section 2.1 for more details. If you used the GitLab repository provided to you, please provide us the address and branch name (which should not be updated after the submission of this assignment).
  - **Group\_XX\_classes.txt**: The resulting classes of the unknown samples should be outputted as described in section 1.2.

To speed up the grading process, we want you to deliver your work exactly as outlined above. **If (a) your files are not of this format or (b) the main function of your code does not compile/run, your work will not be graded.**

*Note*: You are free to use Python or Java for this assignment. You are also allowed to use [NumPy](#) or [ND4J](#) as linear algebra libraries - these are the only libraries to develop the neural network that are allowed for this assignment. You may however use plotting functions with any tool of your choice (including plotting libraries).

The use of Python 3 and Jupyter Notebook is required only for section 2.6, to compare the output of your work to that of *Scikit*.

Fraud will not be tolerated. You are allowed to discuss concepts and ideas with colleagues from other groups, but you are not allowed to share code outside your own group. The same applies to submissions from previous years. You are highly encouraged to make use of the provided Gitlab repositories.

You may ask questions on Stackoverflow (while keeping in mind that it's a public forum, so please don't share partial solutions), or to a TA during the labs.

**The deadline for the assignment is Tuesday, March 1<sup>st</sup> at 17:00.**

Late submissions will be rejected, no exceptions. Good luck!

## 2 QUESTIONS

The following sections can help you through the development of the neural network. Each of the questions need to be answered explicitly in the report. It is very much recommended to take some time to think about the network architecture before you start development.

The central part of this assignment is to code a fully working neural network with the back-propagation training algorithm (your back-propagation should work for at least one hidden layer).

For every question, it is important that you provide **clear explanations** supported by **good plots** (where required), so we can see that you mastered the material.

*Note:* Please make sure all the plots, diagrams and other images have a good quality. Furthermore, all plots should only present relevant information, have clear labels of the axes, be well scaled, and with titles. Please avoid unnecessary graphs and make sure your report is well-formatted and without grammatical mistakes.

### 2.1 SET-UP

Working code is a **knock-out criterion**. If your code does not compile and/or run, your submission will not be graded. Make sure to have a `Main.py` or `Main.java` file containing a main method that the TA can run. At the beginning of your report, please explain (in at most 2 sentences) how to run your code, the path to your main method and any additional packages/libraries a TA may need to install. In your main method, add a **small** block of comments at the top explaining the layout of your code (including where your jupyter notebooks can be found).

In case there are any problems running your code, your group will be contacted by a TA by email.

## 2.2 ARCHITECTURE

The basic element of a neural network is a perceptron. A perceptron is able to learn from its errors and improve after training.

1. (1 point) Develop (program) a single perceptron first. Show with three graphs (one per logic function) presenting the error over epochs that your perceptron is able to learn the OR and the AND functions but not the XOR function.

An ANN is much more powerful than a single perceptron. In the book and during the lectures the multi-layer network for an XOR-operator was introduced, which might help you to create your - more sophisticated - network. You will now design the topology of your network. For every question, briefly discuss how you arrived at your answers.

2. (0.5) How many input neurons are needed for this assignment?
3. (0.5) How many output neurons do you require?
4. (0.5) How many hidden neurons and layers will your network have? (Give an initial guess, later you will try different sizes of hidden layer and analyze the network's performance).
5. (0.5) Which activation function(s) will you use?
6. (1) Give a schematic diagram of your complete network.

## 2.3 TRAINING

Now develop (program) your entire MLP network. Remember that the weight update rule for hidden neurons in an MLP is slightly different than it is for a perceptron. Also remember to take into account the derivative of your activation function. The network you've developed should now be trained to get accustomed to the type of data it will be processing. It's customary to divide data into a training, validation and test set.

7. (2) How and why did you divide your data into a training, validation and test set?
8. (1) How do you evaluate the performance of your network? Briefly justify your answer.
9. (2) When and why do you decide to end the training?
10. (2) Train your network 10 times, each with different initial weights. How does the initialization impact the performance? Plot your results.

## 2.4 OPTIMIZATION

You will now use cross-validation to choose the 'optimal' number of hidden neurons. It is important to keep all other hyper-parameters fixed (learning rate, momentum etc.) to ensure that the results only portray the influence of the number of hidden neurons. Because the

performance of a trained network is somewhat random due to initializations, perform each training 10 times and use the average as an indicator for performance.

11. (3) Train your network with different amounts of hidden neurons. At least 3 times chosen within the range of 7-30 hidden neurons. Generate a plot of the final performance versus the number of hidden neurons in the network. Explain what you observe and what might be the cause of these observations.
12. (2) Pick the architecture with the best result and show a plot of the performance of the training set and the validation set during training, across epochs. Justify your choice.

## 2.5 EVALUATION

It is now time to evaluate your network. Show graphically how your network performs.

13. (2) What is the success rate of your network on the test set? How does it compare to the results of the validation set?
14. (2) Show and discuss a confusion matrix of your test set. How should it be read? Where did your network make the most mistakes? (Search online for the meaning and purpose of a confusion matrix).
15. (10) Feed the unknown set (provided on Brightspace) to the network. Export the resulting classes as a comma-separated file exactly as in section 1.2.

## 2.6 SCIKIT-LEARN

The Python library *scikit-learn* provides algorithms for training neural networks, among other things. We made a Jupyter notebook which uses these features to create and train an ANN, similar to the one you just built. You can find the notebook on Brightspace.

An interesting option in scikit-learn is grid search, which lets you exhaustively search a few points in the hyper-parameter space for optimized values. The notebook uses grid search to train and estimate the performance of a neural network for you.

16. (1) Download and open the Jupyter notebook. Tweak the settings of the grid search, and run all cells to find optimized parameters, according to scikit-learn. Are the values of these parameters different from the ones you chose? What differences can you see in the training behaviour and performance of the scikit network, compared to yours?
17. (2) Finally, take the optimal hyper-parameters found by scikit-learn and plug them into your own network. Does this give you better performance? Why do you think these parameters are better / worse?

## 2.7 REFLECTION

For this question you do not need to keep your specific assignment in mind, rather we ask you to reflect on Artificial Neural Networks and classification. A grocery-robot may have no knowledge of what kind of product an apple is and if it grabs a pear and classifies it as an apple, it would be unpractical, but no real harm is done. However, we use classification in a lot of other places too. This may carry some consequences for a certain group of individuals or introduce problems into society that are unwanted.

18. (2) Can you think of an example of mis-classification and some potentially harmful consequences resulting from it? (Example from the lecture slides is not accepted). Write a short paragraph (at most 5 sentences) explaining these.
19. (2) Do you know of any ways to mitigate the harm done by unjust classification? Give two. Can you be sure that the problem is solved now?