Assignment 1: Artificial Neural Networks

February 10, 2026

0 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 ...

ASSIGNMENT

You and your team will have to code a fully functioning artificial neural network. You will also have to create a report explaining what you did. Please include answers to all the questions from Section 1 in a **structured way** by following the Jupyter notebook template Where asked, 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 need to train and test your network can be found on Brightspace. There are three files in a comma-separated value format:

- `features.txt`: 7854 samples of 10 features
- `targets.txt`: 7854 target classes of the samples mentioned above
- `unknown.txt`: 784 samples of 10 features, with no labels available

It is your job to load the data, prepare it, and use it to train a neural network. Finally, you should use your network to recognize the classes of the unknown samples. In Vocareum, we will publish an assignment where you will auto-grade your predictions. Your grade will be based partly on the accuracy of your network on this set.

You are expected to develop your feed-forward network (based on our template), and make sure that it is able to learn using the back-propagation algorithm. Write clear code, indent where needed and add comments to clarify what is happening. We have also 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.

Your work will be graded in two ways. **You can receive up to 70 points for the 22 questions in Section 1**. We provide you with the evaluation criteria for each of them, make sure that your answers are complete. Furthermore, **you will receive up to 30 points for the general criteria**: quality of the report, quality of the code, scientific method, visual aids, and reproducibility, as explained in Section 2. This means that you can score up to 100 points total (grade 10).

OUTPUT

The last code block of your notebook should generate a list of predictions for the classes of unknown samples. The predictions should be outputted as comma-separated values to a file called `XX_classes.txt` (where XX is your group number). It must be a **text file (.txt)** with all 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.

DELIVERABLES

You will submit a report in the Jupyter Notebook format, consisting of your implementation and your answers the questions from Section 1. **You should hand in the final working version of your implementation on Brightspace (an .ipynb file)**. Your notebook must (1) train your perceptron on the XOR problem and (2) execute the entire training and prediction sequence for the artificial neural network. Check if the notebook works as expected before pressing the `Submit` button!

Your Jupyter Notebook should follow the template that we make available for you. This means that it will include your implementation and the answers to all questions below in a structured manner: `Question 1 – Code 1 – Answer 1 – Question 2 – Code 2 – Answer 2` ...

Furthermore, we ask you to deliver a file with predictions for the unknown samples in format as described in section 0 via on Brightspace (same file, two platforms).

Note: Your implementation will only be graded from 0 to 6 points (see rubric in Section 2). **Do not rely on the code to support your answers for the Questions in Section 1.**

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

You will use Python for this assignment. You are also allowed to use NumPy and pandas as linear algebra/data analysis libraries - **these are the only libraries to develop the neural network that are allowed for this assignment**. You may however use other libraries for tasks unrelated to the functionality of your neural network (for instance for plotting). We provide you with a Python environment with these libraries included (setup instructions on Brightspace). We require you to use that environment specifically to avoid compatibility issues.

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 encouraged to make use of the provided Gitlab repositories. You may ask questions on Answers-EWI (while keeping in mind that it's a public forum, so please don't share partial solutions), or to a TA during the labs.

USE OF AI TOOLS

For transparency, each submitted assignment must include a short statement describing whether and how AI-based tools were used during the preparation of the work. Where applicable, brief examples of how AI tools were used are welcome. This statement must be included as part of the submitted report.

If we believe that AI was used beyond what is stated, or that the submitted work is not the student's own, we reserve the right to request an in-person explanation of the submission to verify the student's understanding.

The deadline for submission is Monday **3rd of March 2026 at 23:59**.

1 QUESTIONS

The following sections will help you develop your neural network. Each question needs 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, **which should work for an arbitrary number of hidden layers**.

Hint: A good rule of thumb when reporting on your work is to:

1. Observe the patterns – e.g. describe what you noticed with regards to the behavior of your algorithm or results of your experiment, possibly referring to data in visual aids.
2. Explain your observations – e.g. state what you think impacts the particular behavior of your algorithm (data, hyper-parameters, ...) or influences your results.
3. Elaborate on your explanation – e.g. put it into a broader context, link to the knowledge from lectures, conduct further experiments to give evidence for your statements.

If you provide clear explanations with supporting evidence (also in form of meaningful visual aids discussed in the text), we will be able to see that you really mastered the material!

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

1.1 SET-UP

Working code is a **knock-out criterion**. If your code does not run, your submission will not be graded. Make sure to extend the main Jupyter Notebook so that when it is ran, it trains a neural network (with hyper-parameters of your choice), predicts the labels of unknown instances, and saves the predictions into a file. You will submit this notebook – along with the rest of your code and any data files – on Brightspace.

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

1.2 ARCHITECTURE

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

Question 1: (4 pt.)
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.

Rubric 1:

- (2 pt.) Functional perceptron code.
- (1 pt.) Correct and informative plots.
- (1 pt.) Sensible discussion of all plots.

An ANN is much more powerful than a single perceptron. You will now design the topology of your network. For every question, briefly explain how you arrived at the answer.

Question 2: (1 pt.)
How many input neurons are needed in our classification task as described above?

Rubric 2:

- (1 pt.) Fully correct answer with brief explanation.

Question 3: (1 pt.)
How many output neurons do you require?

Rubric 3:

- (1 pt.) Fully correct answer with brief explanation.

Question 4: (1 pt.)
How many hidden neurons and layers will your network have? Give an initial guess, later you will tune these values and analyze the network's performance.

Rubric 4:

- (1 pt.) Sensible answer with brief explanation.

Question 5: (1 pt.)
Which activation function(s) will you use?

Rubric 5:
- (1 pt.) Sensible answer with brief explanation.

Question 6: (2 pt.)
Give a schematic diagram of your complete network based on your previous answers.

Rubric 6:
- (2 pt.) Fully correct and readable diagram.

1.3 TRAINING

Now develop your entire Artificial Neural Network. Remember that the weight update rule for hidden neurons in an ANN 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 be trained to get accustomed to the type of data it will be processing.
Note: Your ANN should support an arbitrary number of hidden layers!

Question 7: (2 pt.)
How do you divide your data to ensure unbiased estimate of performance?

Rubric 7:
- (1 pt.) Sensible approach to divide the data.
- (1 pt.) Sensible justification, including ways to ensure reproducibility.

Question 8: (2 pt.)
How do you evaluate the performance of your network? Justify your answer.

Rubric 8:
- (1 pt.) Sensible approach to measure performance.
- (1 pt.) Sensible justification for the performance metric.

Question 9: (2 pt.)
When and why do you decide to end the training?

Rubric 9:

- (1 pt.) Sensible approach to decide on ending the training.
- (1 pt.) Sensible justification for this approach.

Question 10: (4 pt.)
Train your network 10 times, with a few different initialization approaches (e.g., random, uniform). How does the initialization impact the performance? Plot your results.

Rubric 10:

- (2 pt.) Correct and readable plot.
- (1 pt.) Sensible **discussion** of the observed results based on the plot.
- (1 pt.) Sensible **explanation** of the observed results.

1.4 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.

Question 11: (4 pt.)
Train your network with different amounts of hidden neurons. At least 4 times choose within the range of 7-30 hidden neurons for each layer that you are using. 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.

Rubric 11:

- (2 pt.) Correct and readable plot.
- (1 pt.) Sensible **discussion** of the observed results based on the plot.
- (1 pt.) Sensible **explanation** of the observed results.

Question 12: (4 pt.)
Pick the architecture with the best results and plot its performance on the training set and the validation set during training, across epochs. Justify your choice.

Rubric 12:
- (2 pt.) Correct and readable plot.
- (1 pt.) Correct choice of architecture based on previous results.
- (1 pt.) Sensible justification of this choice.

1.5 Evaluation

It is now time to evaluate your network, and use visual aids to present its performance.

Question 13: (3 pt.)
What is the success rate of your network on the test set? How does it compare to the results of the validation set?

Rubric 13:
- (1 pt.) Correct and clear answer.
- (1 pt.) Sensible comparison of the success rates.
- (1 pt.) Sensible explanation of the observed results.

Question 14: (4 pt.)
Show and discuss a confusion matrix constructed on your test set. How should it be read? Where does your network make the most mistakes?

Rubric 14:
- (1 pt.) Correct and readable confusion matrix.
- (1 pt.) Correct explanation how to understand the confusion matrix.
- (2 pt.) Sensible discussion of the performance based on the confusion matrix.

Question 15: (20 pt.)
Feed the unknown set (provided on Brightspace) to the network. Export the resulting classes as a comma-separated file exactly as explained in section 0. Use the Vocareum autograder to decide whether you are satisfied with the performance of your network. Report the autograder score.

Rubric 15:

- (20 pt.) Up to 20 points awarded based on the performance of the network.

1.6 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.

A useful 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.

Question 16: (2 pt.)
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 two networks?

Rubric 16:

- (1 pt.) Successfully used the Jupyter notebook.
- (1 pt.) Sensible discussion of the differences between hyper-parameter values.

Question 17: (2 pt.)
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?

Rubric 17:

- (1 pt.) Correctly reported optimal hyper-parameters found by `scikit-learn`.
- (1 pt.) Sensible discussion and explanation of the differences in performance.

1.7 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 in general. 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. Bias may carry some consequences for a certain group of individuals or introduce problems into society that are unwanted.

Question 18: (3 pt.)
Can you think of an example of misclassification and some potentially harmful consequences resulting from bias? Write a short paragraph (at most 5 sentences) explaining these. Example from the lecture slides is not accepted.

Rubric 18:

- (1 pt.) Proper example relating to the problem at hand.
- (2 pt.) Insightful discussion of the example.

Question 19: (3 pt.)
Think about ways to mitigate the harm **produced** by unjust classification? Give two approaches (think broader than technical solutions). Can you be sure that the problem is solved now?

Rubric 19:

- (1 pt.) Sensible first approach with brief explanation.
- (1 pt.) Sensible second approach with brief explanation.
- (1 pt.) Insightful discussion of the problem at hand.

1.8 PEN AND PAPER

We also have two questions for you to be solved on paper which serve as the preparation for the final exam. They are independent from the rest of the questions in this document. You are asked to submit only one set of answers but we strongly suggest that each team member solves them separately so that you all can practice with the calculations.

Question 20: (3 pt.)
When gathering data about the products, our robot may want to take a photo of the object and extract the features from the intensity of pixels in that photo. Imagine that the process works as follows. First, the robot uses `2 x 2` max pooling filter, reducing the image dimensionality to `4 x 4`. Then, on the transformed photo it uses the *sharpen* kernel (Figure 1.1) **without** padding. Finally, it adds the values per row (such that x_1 is the sum of the top row, x_2 is the sum of the bottom row), resulting in two numbers that will be used as input into a small neural network. Find the inputs for the photo shown in Figure 1.2 below. Present all steps of your work.

Rubric 20:

- (1 pt.) Correct application of the max pooling filter.
- (1 pt.) Correct application of the sharpen kernel.
- (1 pt.) Correct final answer.

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Figure 1.1: *Sharpen kernel*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 6 & 2 & 1 \\ 1 & 6 & 8 & 4 & 1 \\ 1 & 2 & 4 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Figure 1.2: *Photo taken by the robot*

Question 21: (2 pt.)
Use the inputs x_1 and x_2 found in the previous exercise and the network architecture presented in Figure 1.3 to find the output on neuron y. Assume that neurons z_1, z_2, and y use ReLU as the activation function. You can also assume that both bias terms – x_0 and z_0 – have value 0.

Rubric 21:

- (1 pt.) Correct calculations for the hidden layer.
- (1 pt.) Correct final answer.

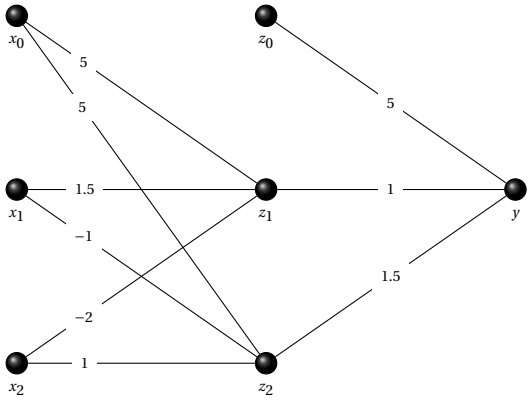

Figure 1.3: *A neural network used by the robot*

1.9 Division of work

Question 22: (0 pt.)
Finally, we ask you to indicate for each group member the approximate contribution **in terms of effort** for each of the criteria in Table 1.1 below. For example, if all four group members contributed equally to the writing of the final report, you should indicate 25% for each of them in the *Report (original draft)* component.

This information will not be directly considered during the grading, but we will take it into account in case any conflicts arise. **Be honest about your own contributions and make sure as a group that everyone agrees about the table included in your final report**. Also, remember that you can always – individually or as a group – reach us at *ci-cs-ewi@tudelft.nl*.

Component	Name A	Name B	Name C	Name D
Code (design)	A	B	C	D
Code (implementation)	A	B	C	D
Code (validation)	A	B	C	D
Experiments (execution)	A	B	C	D
Experiments (analysis)	A	B	C	D
Experiments (visualizations)	A	B	C	D
Report (original draft)	A	B	C	D
Report (reviewing and editing)	A	B	C	D

Table 1.1: *Distribution of work*

2 General rubric

Example calculation for criterion *1. Quality of the report*: if your report is (1) *mostly structured*, it has (2) *clear, well-reasoned answers* and (3) *clear, in-depth discussions*, but it contains (4) *many spelling and grammar mistakes* you would receive for this criterion:

$$3 \cdot 0.25 + 6 \cdot 0.25 + 6 \cdot 0.25 + 0 \cdot 0.25 = 3.75 \text{ out of } 6 \text{ points.}$$

Criterion	0 pt.	3 pt.	6 pt.
1. Quality of the report (each sub-criterion 25%)	Report not structured Many unclear answers Little to no discussion Many spelling/grammar mistakes	Report mostly structured Mostly clear answers Mostly correct discussions Some mistakes	Report properly structured Clear, well-reasoned answers Clear, in-depth discussions Almost no mistakes
2. Quality of the code (each sub-criterion 25%)	Code not structured Improper formatting Little to no comments Many implementation mistakes One hidden layer supported	Code mostly structured Mostly correct formatting Most objects have comments Some mistakes –	Code fully structured Clear, proper formatting Clear, thorough comments Almost no mistakes Any number of hidden layers
3. Scientific method (each sub-criterion 20%)	Approach not adequate Work not sound or reliable Code not tested Results not verified Missing/incoherent arguments	Approach mostly adequate Work mostly sound and reliable – – Mostly coherent arguments	Approach fully adequate Work fully sound and reliable Code tested by unit tests Results mostly verified Coherent, concise arguments
4. Visual aids (each sub-criterion 33%)	Plot types not suitable Many plots are not readable Many missing elements	At most 1 unsuitable plot Clear, readable plots Plots have some missing elements	Plot types always suitable Clear, readable plots No missing elements; all plots have identifiers, clear captions, and references in text
5. Reproducibility (each sub-criterion 33%)	Results not reproducible No code for plots/tables Results have unclear provenance	Results mostly reproducible Code mostly available –	Results fully reproducible Code for plots/tables available Report follows experiments

To be obtained from Questions 1 - 22: 70 **points**
To be obtained from the general rubric: $5 \cdot 6 = 30$ **points**
Total: 100 **points**