|  |
| --- |
| TOI Applied Mathematics |
| Deep Learning Minor |
| Assignment 1 |
| Inholland, 1 februari 2024 |

in tijden van social media.

Lectoraat Jeugd en Samenleving, 2020

in tijden van social media.

Lectoraat Jeugd en Samenleving, 2020

in tijden van social media.

Lectoraat Jeugd en Samenleving, 2020



# Introduction

In this assignment you will first preprocess, then analyze structured data (about penguins) using two neural networks. The first network you will build form scratch and will have only one node with (obviously) one output and having an arbitrary number of inputs. The second network will be a fully connected dense network built using Keras and TensorFlow. You will compare the performance of both approached and draw conclusions about the effects of using a very simple architecture versus a more complicated one.

# Context

Given is a dataset with data collected on penguins of three different species on three different island near Antarctica. The question is whether attributes like the length and depth of the bill – the beak of a penguin – and / or the length of its flippers are a predictor for the weight of one.

# Handing in your work

For this assignment you create multiple Python files in one Jupyter Notebook project. You push all code to our GitLab, for which you will get an account. The URL for our GitLab is <https://appliedmath.toi.inholland.nl>. Please take note that this is not the same as GitHub, which is a Microsoft tool!

Your work needs to be there as a correct project. This means that we can connect Jupyter to GitLab and directly import the project in the IDE. If we first need to download the files manually, then import them into a project, you used GitLab as a file server instead of a version control system, and your work will not be graded.

Configuring Jupyter for use with GitLab is explained in <https://docs-gcp.qubole.com/en/latest/user-guide/notebooks-and-dashboards/notebooks/jupyter-notebooks/managing-jupy-notebook-versions/link-jupy-notebook-gitlab.html>

# Assignment

## Part 1: Cleaning, preprocessing, and feature analysis

The available data is in a coma separated file called palmerpenguins\_original.csv. It contains the following attributes:

* species
* island
* bill\_length\_mm
* bill\_depth\_mm
* flipper\_length\_mm
* body\_mass\_g
* sex
* year

Examine the data and if necessary, write some code to clean and / or encode. Determine what attributes to use as predictor but use at least three.

## Part 2 – A custom neural network

The network to build will look like the following:

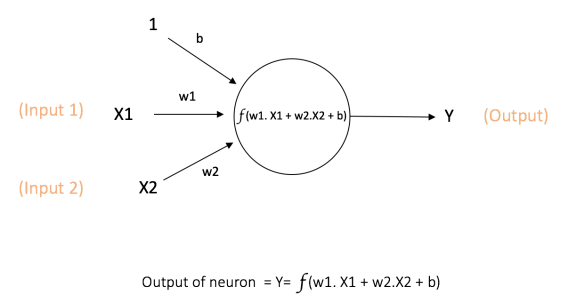


Image 1 [https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/]

While extremely limited, this network has all the elements a multi-layered neural network would have. You need to implement a loss function, activation function, weight initialization, the feed forward mechanism and the backpropagation. After that you use this network to train a regression for the penguins, predicting the weight from other attributes. The steps to take are:

1. Write an activation function. Choose between Sigmoid, ReLu or Swish. Explain in the final report why this one.
2. Create a function which sets the initial weights of the network. This means the input of the function is the network itself, the output the same network but now with the weights being set to an initial value. Initialize all weights with a random value.
3. Create the (basic) node code
4. Implement the feed forward of a neural network for this one node, using the loss function you created.
5. Write a loss function. Choose either Square Loss, Absolute Loss or Huber Loss. Explain in the final report why you chose this one.
6. Create a function which implements the back propagation with partial derivatives. Do not use a library, calculate the (value of) the partial derivatives with your own code.
7. Create the rest of the code implementing the node with the chosen functions and an arbitrary number of inputs.

Write Python code – in your Jupyter notebook project - to split the dataset – penguins – in training and testing. Use your initialization function to initialize the model, then run the training set for a set number of epochs. The goal is to predict the weight of a penguin given the other attributes. Make notes on the outcome.

## Part 3 –Using TensorFlow

In this part you create models for analyzing the same dataset using a complete framework. For the whole course we use TensorFlow with Keras. This allows us to build more complex networks, looking more like:

A diagram of a network

Description automatically generated

Image 2 [https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/]

The dataset used is the same one as for the previous part. You define and train several models, comparing their performance. For this problem you only use the Dense Layer, but you still have different options for the number of nodes in a layer, the number of layers and several functions used. You will experiment with the number of nodes and the number of layers. For the rest of the application choose an appropriate architecture.

Create and train three different models to predict the weight of a penguin from the other attributes. The difference between the models is in the number of layers and in the number of nodes per layer, not in other hyperparameters. Note the testing results you have for your report.

## Handing in the results

Hand in your code from parts 2 and 3 as a Jupyter notebook project on GitLab. Write a report explaining:

* The choices made in part 2
* The results of part 2 (predicting the penguin weights)
* The choices made in part 3
* The results of part 3 (predicting the penguin weights)
* A comparison between the different models trained and tested
* A comparison between the results of part 2 and part 3

Hand in the report via Moodle and the code on GitLab.

## Deadline and resit

The deadline for pushing all your code and uploading the final report to Moodle is Monday March 4, 2024, 09:00. If the assignment as a whole is not at least a 5.5 you get feedback and can push improved code to GitLab and upload an improved report. The deadline for this resit is Monday April 15th, 09:00.

## Grading

Your work is graded according to the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Weight | Topic | Excellent (10) | Sufficient (6) | Poor (2) |
| 2 | Implementing functions | A loss function, activation function and weight initialization scheme have been coded.  Backpropagation has been implemented using partial derivatives  The implementations have no non-trivial errors. | Two of the following have been coded: a loss function, activation function and weight initialization scheme.  Backpropagation has been implemented using partial derivatives  No more than one of these has non-trivial errors. | Either a (loss function and an activation function) *or* (backpropagation) has been coded.  None of these function(s) have non-trivial errors |
| 2 | Combining the functions in a node | All elements have been combined to create a working node.  The node can handle an arbitrary number of inputs.  The node is shown to function as expected by training, validation and testing. | All elements have been combined to create a working node.  The node is shown to function as expected by training, validation and testing. | All elements have been combined to create a working node.  The node is not trained, validated and tested |
| 4 | Building TensorFlow models | Models with different depths and sizes have been created  The models have been trained, validated and tested  Applicable hyperparameters have been chosen | Models with different depths and sizes have been created  The models have been trained, validated and tested  Hyperparameters have been chosen but not all of them are applicable for the problem at hand | Models with different depths and sizes have been created  The models have not been or have been sloppily trained, validated and tested  *or*  Most of the hyperparameter chosen are not applicable for the problem at hand |
| 2 | Drawing conclusions and reporting | For all elements of this assignment (partial) conclusions have been drawn  A comparison between the custom network and the Keras network has been made  A final conclusion about the initial question (about the penguins) has been given  The conclusions have been presented in a structured report | For all elements of this assignment (partial) conclusions have been drawn  A final conclusion about the initial question (about the penguins) has been given  The conclusions have been presented in a structured report | For all elements of this assignment (partial) conclusions have been drawn  or  A final conclusion about the initial question (about the penguins) has been given  The conclusions have been presented in a structured report |

If a criterion does not achieve the standard described under Poor a score of Absent (0) is assigned. For each topic you score between 0 (Absent) and 10 (Excellent) points. The base grade is the weighted average of points scored, with the weights as indicated in the table.

All criteria need to score at least the green level. If one or more criteria score one level below the minimum score the maximum grade is a 4.0. If one or more criteria score two or more levels below the minimum score the maximum grade is a 2.0.

**If the work handed in cannot be imported the grade will be a 0.0, and no feedback is given.**