A picture containing text

Description automatically generated

ITC6230: Deep Learning - Final Project

The American College of Greece

MSc in Data Science

Dimitrios Lezos April 2024

# Documentation

## Running the Neural Network

To run the Neural Network you need to execute:

python neural\_network.py <configuration\_file>

where <configuration\_file> is the file containing the NN configuration, e.g. config.json

## Code description

The code consists of three python files that contain full **doc-strings**:

1. **neural\_network.py**: This file contains class ANN that represents the network as a list of Layer objects.

It is also the entry point for the execution of the application: It reads the configuration, creates the ANN, fits with the train data and evaluates with the test data

1. **Layer.py**: This file contains class Layer which represents a layer of the ANN, the Layer holds the weights for each input and bias value along with the partial derivative θL that needs to be passed back for each input. It supports summing the θLs for all the inputs in a block until the weights are updated where it zero’s the summed values.
2. **functions.py**: This file contains the various functions used in the ANN:
   1. **relu – relu\_derivative**
   2. **sigmoid – sigmoid\_derivative**
   3. **tanh – tanh\_derivative**
   4. **linear – linear\_derivative**
   5. **mse\_loss**
   6. **cce\_loss**

## Configuration File

The configuration file is a JSON file containing the configuration of the Neural Network, it has the structure:

{

"train\_data\_filename": "train.csv",

"train\_data\_skiprows": [],

"test\_size": 0.3,

"random\_state": 42,

"test\_data\_filename": "test.csv",

"test\_data\_skiprows": [],

"header": 0,

"ANN": {…}

}

Where:

1. **train\_data\_filename**: The name of the file containing train data
2. **train\_data\_skiprows**: Array of the 0 index rows of training data to skip, usually due to bad data, as used in pd.read\_csv(skiprows=…)
3. **test\_size**: the proportion of train data to use as validation, as used in train\_test\_split(test\_size=…)
4. **random\_state**: the random state used in the splitting of the train data, as used in train\_test\_split(random\_state=…)
5. **test\_data\_filename**: The name of the file containing test data
6. **test\_data\_skiprows**: Array of the 0 index rows of test data to skip, usually due to bad data, as used in pd.read\_csv(skiprows=…)
7. **header**: the index of the header row in the data, put Null if no header row exists
8. **ANN**: The configuration of the ANN, described below

### ANN Configuration

The configuration of the ANN in the JSON file has the following form:

“ANN”: {

"adam": {…}

"layers": […],

"loss\_function": "mse",

"batch\_size": 10,

"epochs": 40,

"tol": 0.0001

}

Where:

1. **adam**: The configuration parameters of the Adam optimization algorithm, described bellow
2. **layers**: An array containing the configuration of each Layer described bellow
3. **loss\_function**: The loss function to use can be one of: “mse”, “cce”
4. **batch\_size**: The batch size to use
5. **epochs**: The number of epochs to run the training
6. **tol**: The minimum loss function result that when reached will stop the training

### adam Configuration

The configuration of the adam parameters in the JSON file has the following form:

"adam": {

"a": 0.001,

"b1": 0.9,

"b2": 0.999,

"e": 0.00000001

}

Where **a, b1, b2, e** the values of the adam parameters

### layers Configuration

The configuration of the layers in the JSON file is an array of which each entry has the following form:

"layers": [

{

"nodes": 3,

"activation": "relu",

"input\_weights": null,

"biases": null

}

]

Where:

1. **nodes**: The number of nodes in this layer
2. **activation**: The activation function of this layer, can be one of “relu”, “sigmoid”, “tanh”, “linear”
3. **input\_weights**: An array of the initial input weights, if not provided the weights will be initialized to random values produced following the normal distribution with mean=0.0 and s=1
4. **biases**: An array of the initial biases, if not provided the biases will be initialized to random values produced following the normal distribution with mean=0.0 and s=1