Cyron library documentation

1)Preprocessing data

To use your input data with our NN lib includes:

- **class Data -** to read, preprocess, scale your data with a convenient format.
- **Read_data -** to read data from the path given and transform into `Data::set` type
- **Scale -** to scale input data before sending to the NN model
- **Score -** calculates the % of successful prediction

2)Creating a model

Then the input data must be split to create a training and testing set.

Data::split() - takes Data::set and size of a part as an input, returns std::tuple of two Data::sets.

After that the training set should be transformed into a vector of probabilities(0,1), to be used by NN. Then with the following parameters, the model can be created:

Feedforward NN

- **X_train** (`Data::set`),
- **Y_train_prepared -** (`Data::set`),
- **layer_dimensions -** (type `std::vector<int>`) - list of dimensions of {input_layer,hidden_layer, ouput_layer}

Convolutional NN

- **X_train** (`Data::set`),
- **Y_train_prepared -** (`Data::set`),
- **out_layer_dimension**(`int`)
- **Input_data_item_size_x**(`int`)
- **input_data_item_size_y**(`int`)

3)Training a model

Depending on the requirements for productivity and result, the following function for training is implemented:

NeuralNetwork::train() - with parameters:

- **iteration number**(int),
- **learing rate**(double),
- **number of threads** (int)

4)Predicting:

To get a prediction from NN lib includes the following functions:

NeuralNetwork::predict() **-** with parameter **X_test** (Data::set&)

returning Y_res (Data::set).