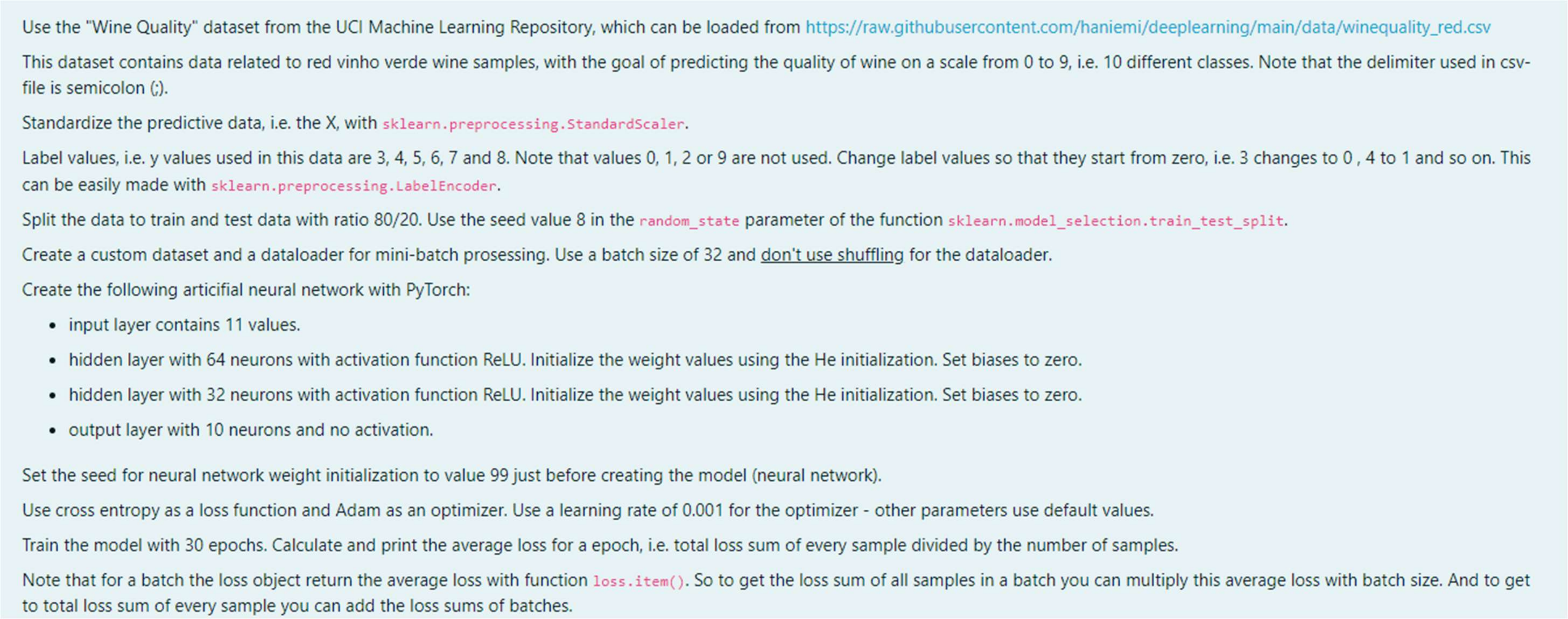
Exercise 9.1



Use the "Wine Quality" dataset from the UCI Machine Learning Repository, which can be loaded from https://raw.githubusercontent.com/haniemi/deeplearning/main/data/winequality_red.csv

This dataset contains data related to red vinho verde wine samples, with the goal of predicting the quality of wine on a scale from 0 to 9, i.e. 10 different classes. Note that the delimiter used in csv-file is semicolon (;).

Standardize the predictive data, i.e. the X, with `sklearn.preprocessing.StandardScaler`.

Label values, i.e. y values used in this data are 3, 4, 5, 6, 7 and 8. Note that values 0, 1, 2 or 9 are not used. Change label values so that they start from zero, i.e. 3 changes to 0 , 4 to 1 and so on. This can be easily made with `sklearn.preprocessing.LabelEncoder`.

Split the data to train and test data with ratio 80/20. Use the seed value 8 in the `random_state` parameter of the function `sklearn.model_selection.train_test_split`.

Create a custom dataset and a dataloader for mini-batch prosessing. Use a batch size of 32 and don't use shuffling for the dataloader.

Create the following artificial neural network with PyTorch:

- input layer contains 11 values.
- hidden layer with 64 neurons with activation function ReLU. Initialize the weight values using the He initialization. Set biases to zero.
- hidden layer with 32 neurons with activation function ReLU. Initialize the weight values using the He initialization. Set biases to zero.
- output layer with 10 neurons and no activation.

Set the seed for neural network weight initialization to value 99 just before creating the model (neural network).

Use cross entropy as a loss function and Adam as an optimizer. Use a learning rate of 0.001 for the optimizer - other parameters use default values.

Train the model with 30 epochs. Calculate and print the average loss for a epoch, i.e. total loss sum of every sample divided by the number of samples.

Note that for a batch the loss object return the average loss with function `loss.item()`. So to get the loss sum of all samples in a batch you can multiply this average loss with batch size. And to get to total loss sum of every sample you can add the loss sums of batches.

What is the Loss value after the 30th epoch? Give the answer rounded to three decimals.

Evaluate the model on the test dataset. What is the accuracy? Give the answer rounded to three decimals. Note that the answer is given as a decimal value, e.g. 0.452 means 45.2%.

Perform hyperparameter tuning with a set of learning rates and betas to find the best hyperparameters for the Adam optimizer. Use the following values for the `lr` and `betas` parameters. Make all value combinations, i.e. $4 \times 6 = 24$ combinations:

- learning_rates = $[0.0005, 0.001, 0.01, 0.1]$
- betas = $[(0.9, 0.999), (0.95, 0.999), (0.99, 0.999), (0.9, 0.99), (0.9, 0.985), (0.9, 0.98)]$

Set the seed for neural network weight initialization to value 99 just before creating a model (neural network) and use 30 epochs.

What is the value of `lr` for the best accuracy achieved?

What is the value of `betas` for the best accuracy achieved?