Assignment 2
This assignment is based on an assignment given in the Machine
Learning and Pattern Recognition course in our university. The first
exercise is supposed to test you on your ability to load and
manipulate data in python. This is an absolutely crucial skill you
need to have to work on any kind of machine learning project. The
second exercise might be a bit hard depending on your background.
Linear regression is a rather intuitive method that can be
effectively used on its own for some problems or a as a baseline for
more complex problems. You can use any kind of tutorials and
materials online but please do not copy code from anywhere. The
crucial part here is for you to actually understand how linear
regression works so do not use ready made models but design the
model yourself. You can however use an already made function for
fitting the weights (np.linalg.lstsq())
In terms of libraries use only numpy, any image processing library
(eg mathplotlib) and any kind of library you want for loading the
data.

The data will be available on our GitHub.

1.
a) Plot a line graph showing the sequence in amp_data, and a
histogram of the amplitudes in this sequence. Include these plots in
your report, with one to three sentences about anything you notice
that might be important for modelling these data.

b) Randomly shuffle the rows of the matrix. Then split the data into
training (70%), validation (15%), and testing (15%). Each dataset
should take the first D = 20 columns as a matrix of inputs X, and
take the final column to create a vector of targets y. Name the
resulting six arrays: X_shuf_train, y_shuf_train, X_shuf_val,
y_shuf_val, X_shuf_test and y_shuf_test. The shuffling means that
our training, validation and testing datasets all come from the same
distribution. Include your code for creating the six arrays above
from the original amp_data array.

2.
Given just one row of inputs, we could fit a curve of amplitude
against time through
the 20 points, and extrapolate it one step into the future.
Plot the points in one row of your X_shuf_train data against the
numbers t = 0/20, 1/20, 2/20, ..., 19/20 representing times. We can
fit this sequence with various linear regression models, and
extrapolate them to predict the 21st time step at time 20/20 = 1.
Indicate the point you're predicting from y_shuf_train on the plot
at t=1.

Fit and plot a straight line to the 20 training points using linear
regression.