# Lab Assignment 2 - Principal Component Analysis (PCA) and Neural Networks

**4/12/2025**

Attempt 1 | In Progress | **NEXT UP: Submit assignment** | Add comment

**Unlimited Attempts Allowed**

20/11/2025 to 4/12/2025

**Details**

Go through each link provided and implement the tasks below.

Make a short report (pdf format) that includes: (1) plots/diagrams generated during training and evaluation, (2) key console output such as model summaries and training logs (loss/accuracy per epoch). Since the labs are only graded Pass/Fail, the report can be very simple, just copy and paste your results and figures given when running your program.

Upload your code as .py file/s.

During the scheduled lab sessions, present your code and report to a teacher. The focus should be on your observations, interpreting the results (what changed, why it matters).

## Tasks

**Principal component analysis (PCA)**

1- Load or create a dataset with more than 2 dimensions.
2- Find the first 2 principal components with and without using sklearn.
3- Practice using PCA to preserve a certain percentage of variance.
4- Train a classification or regression neural network by using:
   a) The Original dataset
   b) Principal components
5- Repeat part 4 using Kernel PCA (linear, sigmoid, RBF).
6- Build a pipeline to tune the hyperparameters of Kernel PCA and also the neural network. Which hyperparameters can be tuned?

**Sample codes for PCA:**

https://github.com/ageron/handson-ml2/blob/master/08_dimensionality_reduction.ipynb
(https://github.com/ageron/handson-ml2/blob/master/08_dimensionality_reduction.ipynb)

**Classification problem**

1- Load IRIS and MNIST fashion datasets from Keras.
2- Build and train a neural network for classifying the loaded labeled datasets.
3- Tune the hyperparameters (including hidden layer size and activation functions). What else can be tuned?
4- Plot the loss and accuracy for training and testing datasets.
5- Save the weights of the layers and use callbacks during the training process.
6- Practice saving and loading the trained model.

**Sample codes for classification and regression models:**

https://github.com/ageron/handson-ml2/blob/master/10_neural_nets_with_keras.ipynb
(https://github.com/ageron/handson-ml2/blob/master/10_neural_nets_with_keras.ipynb)

**Regression problem**

1- Load California housing dataset and split it to training and testing datasets.

2- Build and train a neural network for price prediction. Tune the hyperparameters and discuss the results.
3- Plot the network history.
4- Change the learning rate. Train the network again and plot the network history. Discuss the results.
5- Save the weights of the layers and use callbacks during the training process.
6- Practice saving and loading the trained model.

**Sample codes for classification and regression models:**

https://github.com/ageron/handson-ml2/blob/master/10_neural_nets_with_keras.ipynb
(https://github.com/ageron/handson-ml2/blob/master/10_neural_nets_with_keras.ipynb)

**Choose a submission type**

Web URL | Upload | Canvas Studio | Office 365

(https://hv.instructure.com/courses/10339/modules/items/460071)
assignment
(https://hv.instructure.com/courses/10339/modules/items/460