HW #1

You should submit detailed step-by-step answers for the first question and your own Jupyter Notebook for the second question. TAs will run some of your submissions in Google Colab to verify. If no detailed answers/no Jupyter Notebook submission, 0 pts.

1. Manually do the principal component analysis of the following 3x3 data matrix. Each row means a data vector. Use the **eigenvector with the largest eigenvalue** to transform the data matrix. In other words, you PCA result should return 3x1 matrix. (30pts)

[[1,2,3]

[4,4,5]

[1,4,2]]

Note: Equations and the calculations done by hand are written in the next page. Values were written in fraction during calculation for convenience, and later rewritten in decimals using a calculator.

1) Normalize

```
[[-1.        -1.33333333 -0.33333333]
 [ 2.          0.66666667  1.66666667]
 [-1.          0.66666667 -1.33333333]]
```

2) Covariance Matrix

```
[[3.         1.          2.5        ]
 [1.         1.33333333 0.33333333]
 [2.5        0.33333333 2.33333333]]
```

3) Eigendecomposition

Eigen Value:

λ_1≈5.42166

λ_2≈1.24501

λ_3 = 0

Eigen Vector:
v_1≈(1.18579, 0.371575, 1)
v_2≈(-0.133156, -2.26631, 1)
v_3 = (-1, 0.5, 1)

4) Pincipal Component

λ_1≈5.42166

v_1≈(1.18579, 0.371575, 1)

5) Transform

[[-2.01455447]
[4.28595861]
[-2.27140414]]

$$X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 4 & 5 \\ 1 & 4 & 2 \end{bmatrix} \qquad \bar{X} = [2 \;\; 10/3 \;\; 10/3]$$

$$X - \bar{X} = \begin{bmatrix} -1 & -4/3 & -1/3 \\ 2 & 2/3 & 5/3 \\ -1 & 2/3 & -4/3 \end{bmatrix}$$

$$Q = \frac{1}{2}\begin{bmatrix} -1 & 2 & -1 \\ -4/3 & 2/3 & 2/3 \\ -1/3 & 5/3 & -4/3 \end{bmatrix}\begin{bmatrix} -1 & -4/3 & -1/3 \\ 2 & 2/3 & 5/3 \\ -1 & 2/3 & -4/3 \end{bmatrix} \qquad Q = X^T X \;\; (\because \text{row} = \text{data})$$

$$= \frac{1}{2}\begin{bmatrix} 6 & 2 & 5 \\ 2 & 24/9 & 2/3 \\ 5 & 2/3 & 42/9 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 1 & 2.5 \\ 1 & 4/3 & 1/3 \\ 2.5 & 1/3 & 7/3 \end{bmatrix}$$

$$Q\vec{e} = \lambda\vec{e}$$

$$(Q - \lambda I)\vec{e} = 0 \rightarrow \det(Q - \lambda I) = 0$$

$$\det(Q-\lambda I) = \begin{vmatrix} 3-\lambda & 1 & 2.5 \\ 1 & 4/3-\lambda & 1/3 \\ 2.5 & 1/3 & 7/3-\lambda \end{vmatrix}$$

$$= (3-\lambda)\begin{vmatrix} 4/3-\lambda & 1/3 \\ 1/3 & 7/3-\lambda \end{vmatrix} - \begin{vmatrix} 1 & 1/3 \\ 2.5 & 7/3-\lambda \end{vmatrix} + 2.5\begin{vmatrix} 1 & 4/3-\lambda \\ 2.5 & 1/3 \end{vmatrix}$$

$$= (3-\lambda)\{(4/3-\lambda)(7/3-\lambda) - \tfrac{1}{9}\} - \{(7/3 - \lambda) - \tfrac{5}{6}\} + 2.5\{1/3 - \tfrac{20}{6} + \tfrac{5}{2}\lambda\}$$

$$= (3-\lambda)\{\lambda^2 - 11/3\,\lambda + 3\} + \tfrac{29}{4}\lambda - 9 = 0$$

Solve for λ (eigen value) $\longrightarrow$ Eigen vectors

$\lambda_1 = \frac{1}{6}(20 + \sqrt{157}) \approx 5.42166\ldots$

$\lambda_2 = \frac{1}{6}(20 - \sqrt{157}) \approx 1.24501\ldots$

$\lambda_3 = 0$

$\vec{e}_1 = \left[\frac{3(25+2\sqrt{157})}{64+5\sqrt{157}}, \frac{2(11+\sqrt{157})}{64+5\sqrt{157}}, 1\right] = [1.18578758, 0.37157517, 1]$

$\vec{e}_2 = \left[\frac{3(-25+2\sqrt{157})}{-64+5\sqrt{157}}, \frac{2(-11+\sqrt{157})}{-64+5(\sqrt{157})}, 1\right]$

$\vec{e}_3 = \left[-1, \frac{1}{2}, 1\right]$

Take largest eigen value

$\lambda_1 = 5.42166$

$\vec{e}_1 = [1.18578758, 0.37157517, 1]$

$(X - \bar{X})\cdot\vec{e}_1 = [-2.01455447, 4.28595861, -2.27140414]$

Checking in Python:

$\lambda_1 = 5.42166$

$\vec{e}_1 = [0.74342072, 0.23295629, 0.62694258]$ $\longrightarrow$ returns "normalized" eigenvectors

$(X - \bar{X})\cdot\vec{e}_1 = [-1.26300997, 2.68704994, -1.42403996]$

but we can see the results have <u>same ratio</u> with the hand calculated values. Thus correct :)

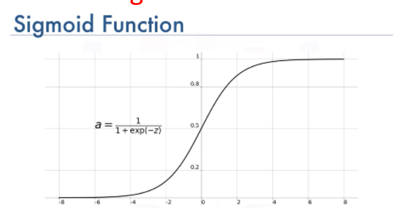
2. In Google Colab, please upload the attached Jupyter notebook and answer the following questions.
   a. Write a complete set of neural network parameters in the attached dense network. How many parameters do you see? (10pts)
      - 1^{st} fully connected layer: weight[32,2] bias[32]
      - 2^{nd} fully connected layer: weight[6,32] bias[6]
      - 3^{rd} fully connected layer: weight[3,6] bias[3]
        ⇨ 3 Weights and 3 biases, total of **315** parameters.

   b. What are hyperparameters in the attached code? (10pts)
      Hyperparameters are the variables set for the network structure and its training. In this code, defined hyperparameters are as follows:
      - Learning rate = 0.001
      - Criterion = Cross Entropy Loss
      - Optimizer = Adam optimizer
      - Activation function = ReLU
      - Batch size = 50
      - Number of epochs = 800

   c. Why does the final linear layer have an output of size 3? (10pts)
      Usually, the final linear layer of the model outputs the score, probability, or the likelihood of the input belonging to each class/label. We interpret model's prediction by choosing the maximum value from this output. Therefore, the network should be constructed in a way that the final output equals the size of the number of classes/labels in the dataset, which in this case is 3.

   d. What is the role of ReLU in the codes? (10pts)
      ReLU is an activation function that transforms the summation of weighted inputs from all the nodes and outputs transformed value. For a value above 0, it outputs as is, but for any value below 0, it outputs 0. Intuitively, this process determines whether a neuron should be activated or not, meaning it has the power to select which signals are to be passed on. Hence the name "activation function". In short, ReLU performs a non-linear transformation to the input.

   e. Try at least three modifications of the model by changing i) the number of layers, ii) the output sizes (dimensionalities) of layers, and/or iii) utilizing other non-linear activations instead of ReLU. For these three modifications, you can decide how and what to change from the model. For each modification, you should train for 800 epochs and take a note of the best test accuracy (not the accuracy from the last epoch but the best accuracy you have observed during the entire training process). **Repeat this 10 times** and report the **average score of those 10 best accuracies**. Training includes some randomness, and you may see different results every training. Therefore, repeating multiple times and reporting their mean best accuracy is required. You should report in the following format. I do not require accuracy improvements for HW1. (30pts)

Original model	No modifications	**80%**
i) the number of layers	3 → 5 dense layers Dimensions: - (2,128) - (128, 64) - (64, 32) - (32, 16) - (16, 3)	**91.2%**
ii) the output sizes (dimensionalities) of layers	3 dense layers (same) Dimensions: - (2, 64) - (64, 32) - (32, 3)	**89.6%**
iii) utilizing other non-linear activations instead of ReLU	ReLU → sigmoid  	**89.8%**

Results (Best Accuracy for each run):

Original = [0.9, 0.9, 0.9, 0.9, 0.9, 0.38, 0.9, 0.9, 0.42, 0.9]

Modification1 = [0.9, 0.9, 0.9, 0.9, 0.9, 0.92, 0.92, 0.92, 0.94, 0.92]

Modification2 = [0.9, 0.88, 0.88, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9]

Modification3 = [0.88, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9]