Short Assignment 5

This is an individual assignment.

For the analytical problems, you can write your answers in markdown cells with $\LaTeX$ or `push` a single pdf to your repository with all handwritten answers.

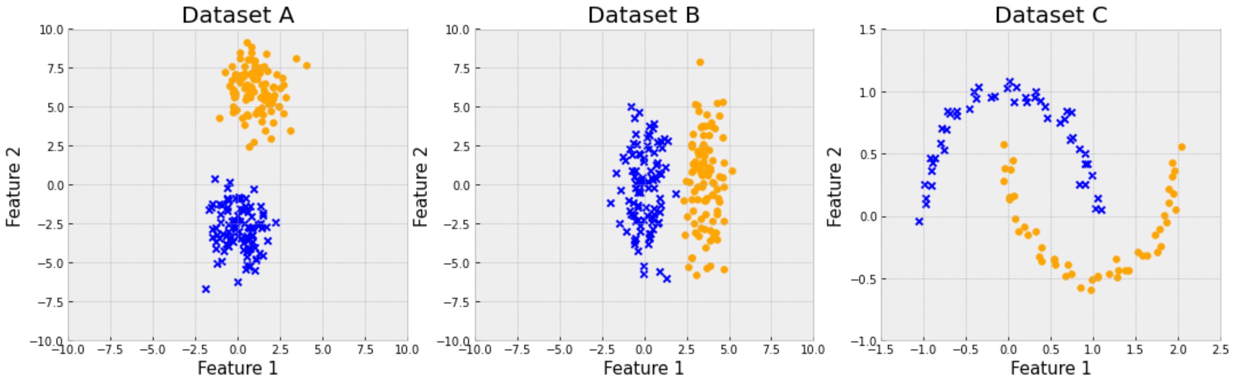
- *Always* show complete work and justify your answers.

Problem 1 (2 points)

In [1]:

```
from IPython.display import Image
Image('2-D-datasets.png',width=800)
```

Out[1]:



Suppose you would like to apply Principal Component Analysis (PCA) to reduce the dimensionality of each of these datasets from 2-D to 1-D where the two clusters remain separated in the projection space. For each dataset (A, B and C), address each of the following two questions:

1. (1 point) **Will PCA be effective at keeping the two clusters separated in the 1-D projection? Why or why not? If yes, state what characteristics of the dataset allow PCA to be effective. If no, state what characteristics of the dataset cause PCA to fail.**
2. (1 point) **Can you think of another dimensionality reduction technique that would be successful at reducing the dimensionality for this dataset while maintaining (or increase) class separability? State the other method and describe why it would be successful.**

1. Dataset A: PCA would likely be effective because the two clusters are linearly separable and have a clear direction of maximum variance, which PCA can capture as the first principal component.

Dataset B: PCA would not be effective because the clusters are not linearly separable in the original feature space. The overlapping of clusters along the line of maximum variance means that PCA will not maintain class separability after dimensionality reduction.

Dataset C: PCA would not be effective. The clusters are arranged in a nonlinear fashion, and PCA, which is a linear method, cannot capture the nonlinear relationship between the two features.

2. T-Distributed Stochastic Neighbor Embedding. It can find patterns in the data thatpreserve the local structure of the data, potentially allowing for better class separability in a lower dimensional space even when linear separability is not present.

```
In [ ]:
```

Problem 2 (3 points)

Consider the data matrix $\mathbf{X}$ of size $3 \times N$, where N is the number of samples. The covariance matrix $\mathbf{K}$, of size 3×3 has 3 eigenvectors $v_1 = [-0.99, 0.09, 0]^T$, $v_2 = [0, 0, 1]^T$ and $v_3 = [-0.09, -0.99, 0]^T$ with eigenvalues $\lambda_1 = 0.98$, $\lambda_2 = 0.5$ and $\lambda_3 = 1.98$, respectively. Answer the following questions:

1. (1 point) **What linear transformation would you use to uncorrelate the data $\mathbf{X}$? Provide a numerical solution and justify your answer.**
2. (1 point) **Use Principal Component Analysis (PCA) to project the 3-dimensional space to a 2-dimensional space. Define the linear transformation (using a numerical answer).**
3. (0.5 points) **What is the amount of explained variance of this 2-D projection? Show your work.**
4. (0.5 points) **Let $\mathbf{Y}$ be the data (linear) transformation obtained by principal component transform of $\mathbf{X}$ onto a 2-dimensional space. What is resulting covariance matrix of transformed data $\mathbf{Y}$? Use a numerical answer and justify your answer.**

1. We can use the eigenvectors of the covariance matrix $\mathbf{K}$ as the linear transformation matrix. The eigenvectors of $\mathbf{K}$ are already orthogonal to each other, which means that they are uncorrelated. Therefore, we can use the matrix $\mathbf{V}$, where $\mathbf{V} = [v_1, v_2, v_3]$, as the linear transformation matrix to uncorrelate the data $\mathbf{X}$.

$$\mathbf{X} = \mathbf{V}^T\mathbf{X}$$

1. Take the first two eigenvectors of the covariance matrix $\mathbf{K}$ as the linear transformation matrix. The first two eigenvectors correspond to the two largest eigenvalues, which represent the directions of maximum variance in the data. Therefore, we can use the matrix $\mathbf{W}$, where $\mathbf{W} = [v_1, v_3]$, as the linear transformation matrix to project the 3-dimensional space to a 2-dimensional space.

$$\mathbf{Y} = \mathbf{W}^T\mathbf{X}$$

1. The amount of explained variance of this 2-D projection is the sum of the two largest eigenvalues divided by the sum of all the eigenvalues. sum of all the eigenvalues:

$$\sum_{i=1}^{3} \lambda_i = 0.98 + 0.5 + 1.98 = 3.46$$

sum of the two largest eigenvalues:

$$\lambda_1 + \lambda_3 = 0.98 + 1.98 = 2.96$$

the amount of explained variance of this 2-D projection:

$$\frac{\lambda_1 + \lambda_3}{\sum_{i=1}^{3} \lambda_i} = 0.85$$

1. The resulting covariance matrix of the transformed data $\mathbf{Y}$:

$$\mathbf{K_Y} = \mathbf{W}^T\mathbf{K}\mathbf{W}$$

$$\mathbf{K_Y} = \begin{bmatrix} -0.99 & -0.09 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} 0.98 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1.98 \end{bmatrix} \begin{bmatrix} -0.99 & -0.09 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.95 & 0 \\ 0 & 1.98 \end{bmatrix}$$

The resulting covariance matrix is diagonal, meaning that the two dimensions are uncorrelated. This is expected since PCA is designed to find the directions of maximum variance in the data, which are uncorrelated.

In [ ]:

In [ ]:

In [ ]:

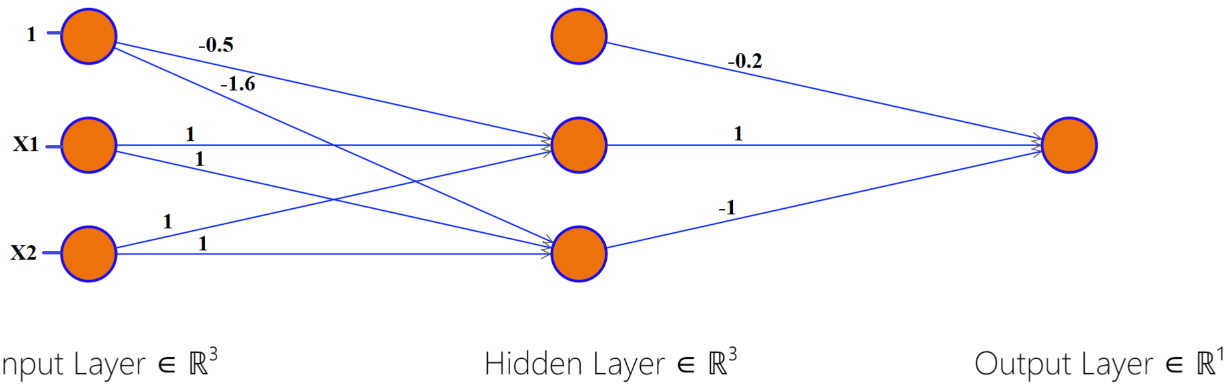
Problem 3 (4 points)

Consider the following neural network architecture: an input layer with 2 units, 1 hidden layer with 2 units and the output layer with 1 unit.

In [2]:

```
Image('neural_network.png', width=900)
```

Out[2]:



The weight matrix/vector for the hidden and output layers are $W_H = \begin{bmatrix} -0.5 & -1.6 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$ **and** $W_O = \begin{bmatrix} w_C \\ w_A \\ w_B \end{bmatrix} = \begin{bmatrix} -0.2 \\ 1 \\ -1 \end{bmatrix}$**, respectively.**

Consider the threshold activation function for the hidden layer, $\phi_T(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$**, and the sigmoid activation function for the output layer,** $\phi_s(x) = \frac{1}{1+e^{-x}}$**. Recall that** $\phi_s'(x) = \phi_s(x)(1 - \phi_s(x))$**.**

1. (1 point) **Draw the decision function this architecture is currently learning.**

1. (2 points) **Consider the binary cross-entropy as the objective function**

$$H(y) = \sum_{i=1}^{N} -t_i \ln(y_i) - (1 - t_i) \ln(1 - y_i)$$

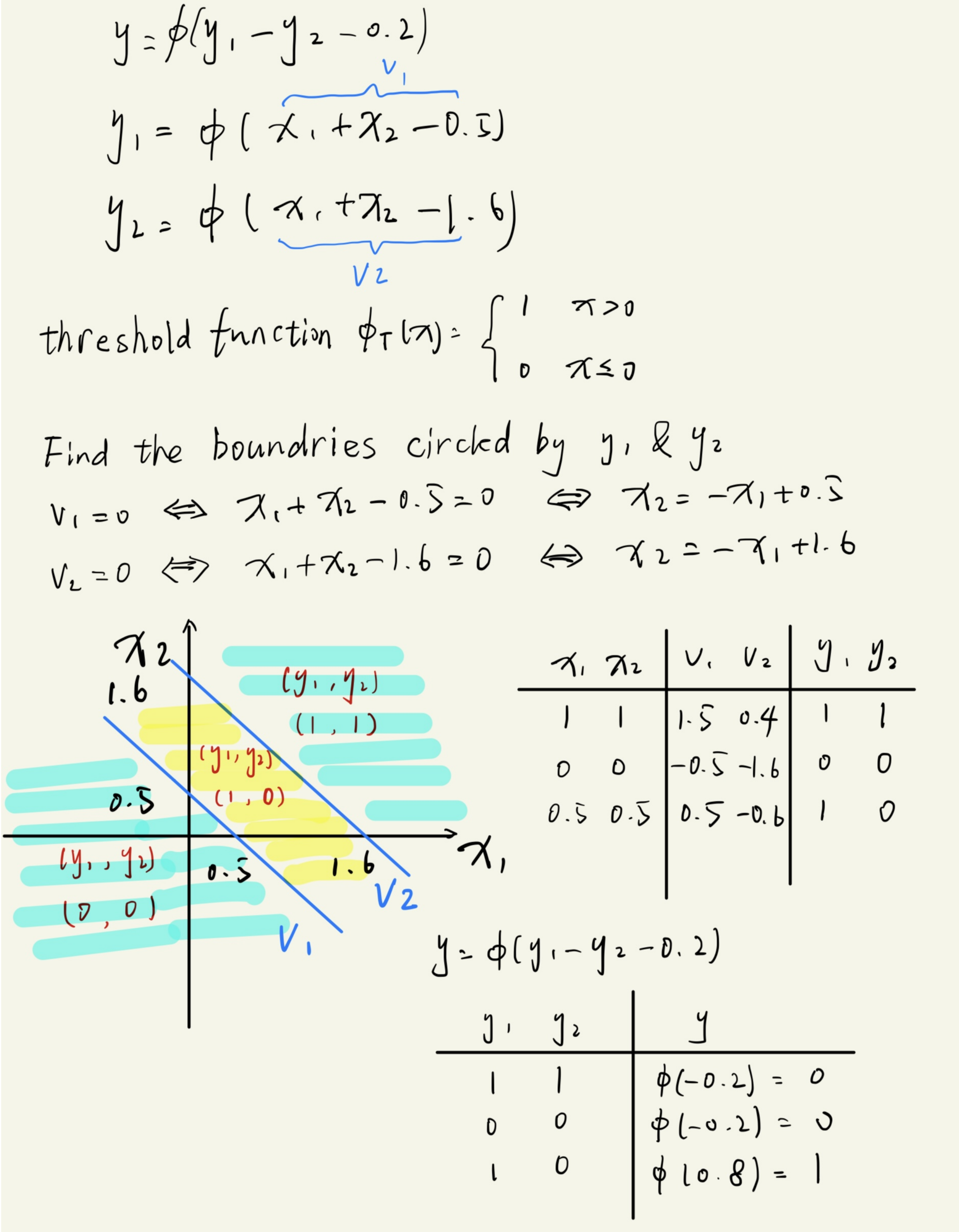
where t_i **is the target value of sample** $\mathbf{x}_i$ **and** y_i **is the output of the mapper function.**

Use backpropagation with online learning to update the weight w_A **directly connected to the output layer. Consider the data point** $\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ **with** $t = 1$ **and a learning rate of** $\eta = 0.01$**. Show all your work.**

In [3]:

```
Image('1.jpg', width=900)
```

Out[3]:



$$y = \phi(y_1 - y_2 - 0.2)$$

$$y_1 = \phi(\overbrace{x_1 + x_2 - 0.5}^{v_1})$$

$$y_2 = \phi(\underbrace{x_1 + x_2 - 1.6}_{V2})$$

threshold function $\phi_T(x) = \begin{cases} 1 & x > 0 \\ 0 & x \le 0 \end{cases}$

Find the boundries circled by y_1 & y_2

$v_1 = 0 \Leftrightarrow x_1 + x_2 - 0.5 = 0 \Leftrightarrow x_2 = -x_1 + 0.5$

$V_2 = 0 \Leftrightarrow x_1 + x_2 - 1.6 = 0 \Leftrightarrow x_2 = -x_1 + 1.6$

x2
1.6
0.5
x1
0.5
1.6
V2
V1
(y1, y2)
(1, 1)
(y1, y2)
(1, 0)
(y1, y2)
(0, 0)

x_1	x_2	v_1	v_2	y_1	y_2
1	1	1.5	0.4	1	1
0	0	-0.5	-1.6	0	0
0.5	0.5	0.5	-0.6	1	0

$$y = \phi(y_1 - y_2 - 0.2)$$

y_1	y_2	y
1	1	$\phi(-0.2) = 0$
0	0	$\phi(-0.2) = 0$
1	0	$\phi(0.8) = 1$

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```
```

On-Time (1 point)

Submit your assignment before the deadline.

Submit Your Solution

Confirm that you've successfully completed the assignment.

Along with the Notebook, include a PDF of the notebook with your solutions.

`add` and `commit` the final version of your work, and `push` your code to your GitHub repository.

Submit the URL of your GitHub Repository as your assignment submission on Canvas.
