5 Feature Extraction

1. Briefly define The following terms:

a. Feature Engineering

modifying measured values to make them suitable for classification.

b. Feature Selection

choosing a subset of measured/possible features to use for classification.

c. Feature Extraction

projecting the chosen feature vectors into a new feature space.

d. Dimensionality Reduction

selecting/extracting feature vectors of lower dimensionality (length) than the original feature vectors.

e. Deep Learning

the process of training a neural network that has many layers (> 3, typically $\gg 3$), or performing multiple stages of (nonlinear) feature extraction prior to classification.

2. List 5 methods that can be used to perform feature extraction.

Any five from:

- *Principal Component Analysis (PCA)*
- *Whitening*
- *Linear Discriminant Analysis (LDA)*
- *Independent Component Analysis (ICA)*
- *Random Projections*
- *Sparse Coding*

3. Write pseudo-code for the Karhunen-Loève Transform method for performing Principal Component Analysis (PCA).

1. *Subtract the mean from all data vectors.*
2. *Calculate the covariance matrix of the zero-mean data.*
3. *Find the eigenvalues and eigenvectors of the covariance matrix.*
4. *Order eigenvalues from large to small, and discard small eigenvalues and their respective vectors. Form a matrix ($\hat{V}$) of the remaining eigenvectors.*
5. *Project the zero-mean data onto the PCA subspace by multiplying by the transpose of $\hat{V}$.*

4. Use the Karhunen-Loève Transform to project the following 3-dimensional data onto the first two principal components (the MATLAB command eig can be used to find eigenvectors and eignevalues).

$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 3 \\ 5 \\ 1 \end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}.$

The mean of the data is $\mu = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$*, hence, the zero-mean data is:*

$\mathbf{x}'_1 = \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix}, \mathbf{x}'_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{x}'_3 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}, \mathbf{x}'_4 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}.$

The covariance matrix is

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T$$

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}'_i)(\mathbf{x}'_i)^T$$

$$\mathbf{C} = \frac{1}{4}\left[\begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix}(-1 \quad -1 \quad 0) + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}(0 \quad 0 \quad 0) + \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}(1 \quad 2 \quad 0) + \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}(0 \quad -1 \quad 0)\right]$$

$$\mathbf{C} = \frac{1}{4}\left[\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}\right]$$

$$\mathbf{C} = \frac{1}{4}\begin{pmatrix} 2 & 3 & 0 \\ 3 & 6 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Calculating eigenvectors and eignevalues of $\mathbf{C}$ *(using MATLAB command eig):*

$$\mathbf{V} = \begin{pmatrix} 0 & -0.8817 & 0.4719 \\ 0 & 0.4719 & 0.8817 \\ 1 & 0 & 0 \end{pmatrix} \qquad \mathbf{E} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.0986 & 0 \\ 0 & 0 & 1.9014 \end{pmatrix}$$

We need to choose the two eigenvectors corresponding to the two largest eigenvectors:

$$\hat{\mathbf{V}} = \begin{pmatrix} 0.4719 & -0.8817 \\ 0.8817 & 0.4719 \\ 0 & 0 \end{pmatrix}$$

Projection of the data onto the subspace spanned by the 1st two principal components is given by: $\mathbf{y}_i = \hat{\mathbf{V}}^T(\mathbf{x}_i - \mu)$

Hence, $\mathbf{y}_i = \begin{pmatrix} 0.4719 & 0.8817 & 0 \\ -0.8817 & 0.4719 & 0 \end{pmatrix}\mathbf{x}'_i$

Therefore,

$$\mathbf{y}_1 = \begin{pmatrix} 0.4719 & 0.8817 & 0 \\ -0.8817 & 0.4719 & 0 \end{pmatrix}\begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1.3536 \\ 0.4098 \end{pmatrix}$$

$$\mathbf{y}_2 = \begin{pmatrix} 0.4719 & 0.8817 & 0 \\ -0.8817 & 0.4719 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\mathbf{y}_3 = \begin{pmatrix} 0.4719 & 0.8817 & 0 \\ -0.8817 & 0.4719 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 2.2353 \\ 0.0621 \end{pmatrix}$$

$$\mathbf{y}_4 = \begin{pmatrix} 0.4719 & 0.8817 & 0 \\ -0.8817 & 0.4719 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -0.8817 \\ -0.4719 \end{pmatrix}$$

Note: the new data, $\mathbf{y}$, has zero mean and the covariance matrix is: $\begin{pmatrix} 1.9015 & 0 \\ 0 & 0.0986 \end{pmatrix}$ (the eigenvalues of the original covariance matrix measure variance along each principal component).

5. What is the proportion of the variance explained by the 1st two principal components in the preceding question?

Proportion of the variance is given by sum of eigenvalues for selected components divided by the sum of all eigenvalues.

For the preceding question this is

$$\frac{1.9015 + 0.0986}{1.9015 + 0.0986 + 0} = 1$$

Note: the 1st principal component alone would explain $\frac{1.9015}{1.9015+0.0986+0} = 0.95$ of the variance, so we could project data onto only the 1st PC without losing too much information.

6. Use the Karhunen-Loève Transform to project the following 2-dimensional dataset onto the first principal component (the MATLAB command eig can be used to find eigenvectors and eignevalues).
$\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 6 \end{pmatrix}, \begin{pmatrix} 8 \\ 7 \end{pmatrix}, \begin{pmatrix} 9 \\ 7 \end{pmatrix}$.

The mean of the data is $\mu = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$, hence, the zero-mean data is:
$\begin{pmatrix} -5 \\ -4 \end{pmatrix}, \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 4 \\ 2 \end{pmatrix}$.

The covariance matrix is

$$\mathbf{C} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T$$

$$\mathbf{C} = \frac{1}{6} \left[\begin{pmatrix} -5 \\ -4 \end{pmatrix} (-5 \quad -4) + \begin{pmatrix} -2 \\ 0 \end{pmatrix} (-2 \quad 0) + \begin{pmatrix} 0 \\ -1 \end{pmatrix} (0 \quad -1) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} (0 \quad 1) + \begin{pmatrix} 3 \\ 2 \end{pmatrix} (3 \quad 2) + \begin{pmatrix} 4 \\ 2 \end{pmatrix} (4 \quad 2) \right]$$

$$\mathbf{C} = \frac{1}{6}\left[\begin{pmatrix} 25 & 20 \\ 20 & 16 \end{pmatrix} + \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 9 & 6 \\ 6 & 4 \end{pmatrix} + \begin{pmatrix} 16 & 8 \\ 8 & 4 \end{pmatrix}\right]$$

$$\mathbf{C} = \frac{1}{6}\begin{pmatrix} 54 & 34 \\ 34 & 26 \end{pmatrix} = \begin{pmatrix} 9 & 5.67 \\ 5.67 & 4.33 \end{pmatrix}$$

Calculating eigenvectors and eignevalues of $\mathbf{C}$ *(using MATLAB command eig):*

$$\mathbf{V} = \begin{pmatrix} 0.5564 & -0.8309 \\ -0.8309 & -0.5564 \end{pmatrix} \qquad \mathbf{E} = \begin{pmatrix} 0.5384 & 0 \\ 0 & 12.7949 \end{pmatrix}$$

We need to choose the eigenvector corresponding to the largest eigenvector:

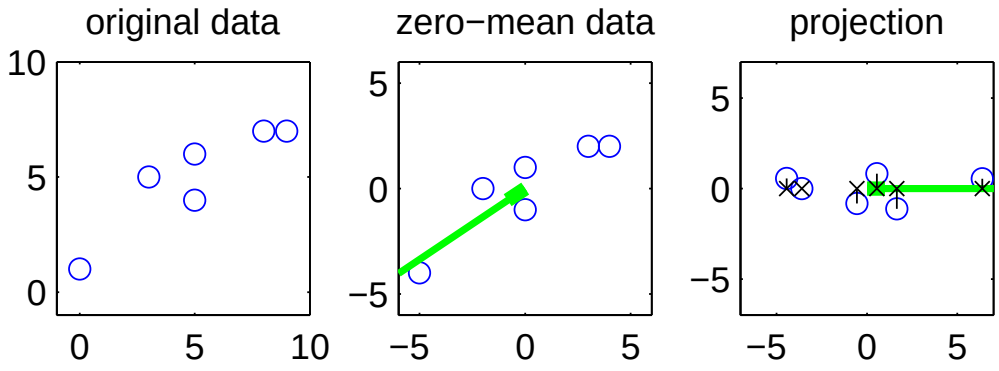
$$\hat{\mathbf{V}} = \begin{pmatrix} -0.8309 \\ -0.5564 \end{pmatrix}$$

Projection of the data onto the subspace spanned by the 1st principal component is given by: $\mathbf{y}_i = \hat{\mathbf{V}}^T(\mathbf{x}_i - \mu)$

Hence, $\mathbf{y}_i = \begin{pmatrix} -0.8309 & -0.5564 \end{pmatrix}\left[\mathbf{x}_i - \begin{pmatrix} 5 \\ 5 \end{pmatrix}\right]$

Therefore, the new dataset is: $6.3801, 1.6618, 0.5564, -0.5564, -3.6055, -4.4364.$

The projection looks like this:



7. Apply two epochs of a batch version of Oja's learning rule to the same data used in the previous question. Use a learning rate of 0.01 and an initial weight vector of [-1,0].

For the Batch Oja's learning rule, weights are updated such that:
$\mathbf{w} \leftarrow \mathbf{w} + \eta \sum_i y(\mathbf{x}_i^t - y\mathbf{w})$. *Here,* $\eta = 0.01$.

Epoch 1, initial $\mathbf{w} = [-1, 0]$

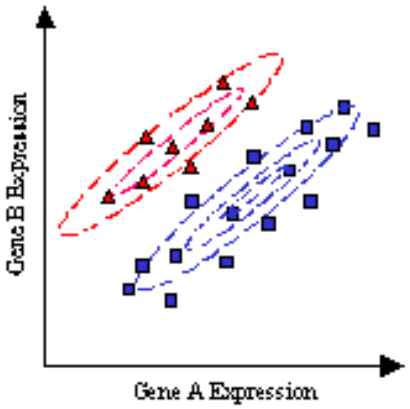
$\mathbf{x}^t$	$y = \mathbf{wx}$	$\mathbf{x}^t - y\mathbf{w}$	$\eta y(\mathbf{x}^t - y\mathbf{w})$	$\mathbf{w}$
$(-5, -4)$	5	$(0, -4)$	$(0, -0.2)$	
$(-2, 0)$	2	$(0, 0)$	$(0, 0)$	
$(0, -1)$	0	$(0, -1)$	$(0, 0)$	
$(0, 1)$	0	$(0, 1)$	$(0, 0)$	
$(3, 2)$	-3	$(0, 2)$	$(0, -0.06)$	
$(4, 2)$	-4	$(0, 2)$	$(0, -0.08)$	
		total weight change	$(0, -0.34)$	$(-1, -0.34)$

Epoch 2, initial $\mathbf{w} = [-1, -0.34]$

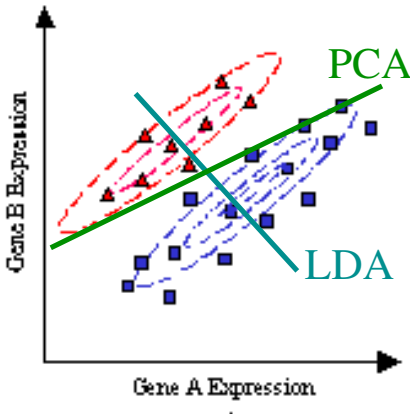
$\mathbf{x}^t$	$y = \mathbf{wx}$	$\mathbf{x}^t - y\mathbf{w}$	$\eta y(\mathbf{x}^t - y\mathbf{w})$	$\mathbf{w}$
$(-5, -4)$	6.36	$(1.36, -1.84)$	$(0.087, -0.117)$	
$(-2, 0)$	2	$(0, 0.68)$	$(0, 0.0136)$	
$(0, -1)$	0.34	$(0.34, -0.88)$	$(0.001, -0.003)$	
$(0, 1)$	-0.34	$(-0.34, 0.88)$	$(0.001, -0.003)$	
$(3, 2)$	-3.68	$(-0.68, 0.75)$	$(0.025, -0.028)$	
$(4, 2)$	-4.68	$(-0.68, 0.41)$	$(0.318, -0.019)$	
		total weight change	$(0.146, -0.156)$	$(-0.854, -0.496)$

Note:
after 3 epochs: $\mathbf{w} = (-0.8468, -0.5453)$
after 4 epochs: $\mathbf{w} = (-0.8302, -0.5505)$
after 5 epochs: $\mathbf{w} = (-0.8333, -0.5565)$
after 6 epochs: $\mathbf{w} = (-0.8302, -0.5556)$
cf., result for previous question when calculating first PC via the Karhunen-Loève Transform.

8. The graph below shows a two-dimensional dataset in which examplars come from two classes. Exemplars from one class are plotted using triangular markers, and exemplars from the other class are plotted using square markers.



- **Draw the approximate direction of the first principal component of this data.**
- **Draw the approximate direction of the axis onto which the data would be projected using LDA.**



9. Briefly describe the optimisation performed by Fisher's Linear Discriminant Analysis to find a projection of the original data onto a subspace.

LDA searches for a discriminative subspace in which exemplars belonging to the same class are as close together as possible while patterns belonging to different classes are as far apart as possible.

10. For the data in the Table below use Fisher's method to determine which of the following projection weights is more effective at performing Linear Discriminant Analysis (LDA).

- $\mathbf{w}^T = [-1, 5]$
- $\mathbf{w}^T = [2, -3]$

Class	Feature vector $\mathbf{x}^T$
1	$[1, 2]$
1	$[2, 1]$
1	$[3, 3]$
2	$[6, 5]$
2	$[7, 8]$

Fisher's Linear Discriminant Analysis maximise the cost function $J(\mathbf{w}) = \frac{sb}{sw}$*, where:*

$$sb = |\mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)|^2$$

$$sw = \sum_{\mathbf{x}\in\omega_1} (\mathbf{w}^T(\mathbf{x} - \mathbf{m}_1))^2 + \sum_{\mathbf{x}\in\omega_2} (\mathbf{w}^T(\mathbf{x} - \mathbf{m}_2))^2$$

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{x\in\omega_i} x$$

Sample mean for class 1 is $\mathbf{m}_1^T = \frac{1}{3}([1, 2] + [2, 1] + [3, 3]) = [2, 2]$.

Sample mean for class 2 is $\mathbf{m}_2^T = \frac{1}{2}([6, 5] + [7, 8]) = [6.5, 6.5]$.

$\mathbf{w}^T = [-1, 5]$*:*
Between class scatter (sb) = $|\mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)|^2$
$= |[-1, 5] \times ([2, 2] - [6.5, 6.5])^T|^2 \ = \ |[-1, 5] \times [-4.5, -4.5]^T|^2 \ = \ | - 18|^2 \ = \ 324$

Within class scatter (sw) = $\sum_{\mathbf{x}\in\omega_1} (\mathbf{w}^T(\mathbf{x} - \mathbf{m}_1))^2 + \sum_{\mathbf{x}\in\omega_2} (\mathbf{w}^T(\mathbf{x} - \mathbf{m}_2))^2$
$= ([-1, 5] \times ([1, 2] - [2, 2])^T)^2 + ([-1, 5] \times ([2, 1] - [2, 2])^T)^2 + ([-1, 5] \times ([3, 3] - [2, 2])^T)^2 + ([-1, 5] \times ([6, 5] - [6.5, 6.5])^T)^2 + ([-1, 5] \times ([7, 8] - [6.5, 6.5])^T)^2 \ = \ 140$

Cost $J(\mathbf{w}) = \frac{sb}{sw} \ = \ \frac{324}{140} \ = \ 2.3143$

For $\mathbf{w}^T = [2, -3]$*:*
Between class scatter (sb) = $|\mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)|^2$
$= |[2, -3] \times ([2, 2] - [6.5, 6.5])^T|^2 \ = \ |[2, -3] \times [-4.5, -4.5]^T|^2 \ = \ |4.5|^2 \ = \ 20.25$

Within class scatter (sw) = $\sum_{\mathbf{x}\in\omega_1} (\mathbf{w}^T(\mathbf{x} - \mathbf{m}_1))^2 + \sum_{\mathbf{x}\in\omega_2} (\mathbf{w}^T(\mathbf{x} - \mathbf{m}_2))^2$
$= ([2, -3] \times ([1, 2] - [2, 2])^T)^2 + ([2, -3] \times ([2, 1] - [2, 2])^T)^2 + ([2, -3] \times ([3, 3] - [2, 2])^T)^2 + ([2, -3] \times ([6, 5] - [6.5, 6.5])^T)^2 + ([2, -3] \times ([7, 8] - [6.5, 6.5])^T)^2 \ = \ 38.5$

Cost $J(\mathbf{w}) \ = \ \frac{sb}{sw} \ = \ \frac{20.5}{38.5} \ = \ 0.526$

As $J(\mathbf{w})$ *given by* $\mathbf{w}^T = [-1, 5]$ *is higher, it is a more effective projection weight.*

Note, projection of the data into the new feature space defined by the two projection weights is:

Class	*Feature vector* $\mathbf{x}^T$	$y = \mathbf{w}^T\mathbf{x}$	
		$\mathbf{w}^T = [-1, 5]$	$\mathbf{w}^T = [2, -3]$
1	$[1, 2]$	$[-1, 5] \times [1, 2]^T = 9$	$[2, -3] \times [1, 2]^T = -4$
1	$[2, 1]$	$[-1, 5] \times [2, 1]^T = 3$	$[2, -3] \times [2, 1]^T = 1$
1	$[3, 3]$	$[-1, 5] \times [3, 3]^T = 12$	$[2, -3] \times [3, 3]^T = -3$
2	$[6, 5]$	$[-1, 5] \times [6, 5]^T = 19$	$[2, -3] \times [6, 5]^T = -3$
2	$[7, 8]$	$[-1, 5] \times [7, 8]^T = 33$	$[2, -3] \times [7, 8]^T = -10$

It can be seen that after projection by $\mathbf{w}^T = [-1, 5]$ *the data is linearly separable, while after projection by* $\mathbf{w}^T = [2, -3]$ *it is not.*

11. An Extreme Learning Machine consists of a hidden layer with six neurons, and an output layer with one neuron. The weights to the hidden neurons have been assigned the following random values:

$$\mathbf{V} = \begin{pmatrix} -0.62 & 0.44 & -0.91 \\ -0.81 & -0.09 & 0.02 \\ 0.74 & -0.91 & -0.60 \\ -0.82 & -0.92 & 0.71 \\ -0.26 & 0.68 & 0.15 \\ 0.80 & -0.94 & -0.83 \end{pmatrix}$$

The weights to the output neuron are: $\mathbf{w} = (0, 0, 0, -1, 0, 0, 2)$**. All weights are defined using augmented vector notation. Hidden neurons are Linear Threshold units, while the output neuron is linear. Calculate the response of the output neuron to each of the following input vectors:** $\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}$**.**

If we place all the augmented input patterns into a matrix we have the following dataset:

$$\mathbf{X} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

The response of each hidden neuron to a single exemplar is defined as $y = H(\mathbf{vx})$*, where H is the heaviside function. The response of all six hidden neurons to all four input patterns, is given by:*

$$\mathbf{Y} = H[\mathbf{VX}] = H\left[\begin{pmatrix} -0.62 & 0.44 & -0.91 \\ -0.81 & -0.09 & 0.02 \\ 0.74 & -0.91 & -0.60 \\ -0.82 & -0.92 & 0.71 \\ -0.26 & 0.68 & 0.15 \\ 0.80 & -0.94 & -0.83 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}\right]$$

$$\mathbf{Y} = H\begin{pmatrix} -0.62 & -1.53 & -0.18 & -1.09 \\ -0.81 & -0.79 & -0.90 & -0.88 \\ 0.74 & 0.14 & -0.17 & -0.77 \\ -0.82 & -0.11 & -1.74 & -1.03 \\ -0.26 & -0.11 & 0.42 & 0.57 \\ 0.80 & -0.03 & -0.14 & -0.97 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

The response of the output neuron to a single exemplar is defined as $z = \mathbf{wy}$*. The response of the output neuron to all four input patterns, is given by:*

$$\mathbf{Z} = \mathbf{wY} = \begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} = (1 \quad -1 \quad 0 \quad 0)$$

12. Given a dictionary, $\mathbf{V}^t$, what is the best sparse code for the signal $\mathbf{x}$ out of the following two alternatives:
i) $\mathbf{y_1}^t = (1,0,0,0,1,0,0,0)$
ii) $\mathbf{y_2}^t = (0,0,1,0,0,0,-1,0)$
Where $\mathbf{V}^t = \begin{pmatrix} 0.4 & 0.55 & 0.5 & -0.1 & -0.5 & 0.9 & 0.5 & 0.45 \\ -0.6 & -0.45 & -0.5 & 0.9 & -0.5 & 0.1 & 0.5 & 0.55 \end{pmatrix}$**, and** $\mathbf{x} = \begin{pmatrix} -0.05 \\ -0.95 \end{pmatrix}$**. Assume that sparsity is measured as the count of elements that are non-zero.**

Both alternatives are equally sparse (2 non-zero elements each), so the best will be the one with the lowest reconstruction error: $\|\mathbf{x} - \mathbf{V}^t\mathbf{y}\|_2$.

For (i) error = $\|\mathbf{x} - \mathbf{V}^t\mathbf{y_1}\|_2$

$$= \left\| \begin{pmatrix} -0.05 \\ -0.95 \end{pmatrix} - \begin{pmatrix} 0.4 & 0.55 & 0.5 & -0.1 & -0.5 & 0.9 & 0.5 & 0.45 \\ -0.6 & -0.45 & -0.5 & 0.9 & -0.5 & 0.1 & 0.5 & 0.55 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right\|_2$$

$$= \left\| \begin{pmatrix} -0.05 \\ -0.95 \end{pmatrix} - \begin{pmatrix} 0.4 - 0.5 \\ -0.6 - 0.5 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} -0.05 \\ -0.95 \end{pmatrix} - \begin{pmatrix} -0.1 \\ -1.1 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} 0.05 \\ 0.15 \end{pmatrix} \right\|_2$$

$$= \sqrt{0.05^2 + 0.15^2} = 0.158$$

For (ii) error = $\|\mathbf{x} - \mathbf{V}^t\mathbf{y_2}\|_2$

$$= \left\| \begin{pmatrix} -0.05 \\ -0.95 \end{pmatrix} - \begin{pmatrix} 0.4 & 0.55 & 0.5 & -0.1 & -0.5 & 0.9 & 0.5 & 0.45 \\ -0.6 & -0.45 & -0.5 & 0.9 & -0.5 & 0.1 & 0.5 & 0.55 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{pmatrix} \right\|_2$$

$$= \left\| \begin{pmatrix} -0.05 \\ -0.95 \end{pmatrix} - \begin{pmatrix} 0.5 - 0.5 \\ -0.5 - 0.5 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} -0.05 \\ -0.95 \end{pmatrix} - \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} -0.05 \\ 0.05 \end{pmatrix} \right\|_2$$

$$= \sqrt{0.05^2 + 0.05^2} = 0.071$$

Therefore, solution (ii) is the better sparse code.

13. Repeat the previous questions when the two alternatives are:
i) $\mathbf{y_1}^t = (1,0,0,0,1,0,0,0)$
ii) $\mathbf{y_2}^t = (0,0,0,-1,0,0,0,0)$

(i) is the same as in the previous question, therefore for (i) the error is 0.158.

For (ii) error = $\|\mathbf{x} - \mathbf{V}^t\mathbf{y_2}\|_2$

$$\left\| \begin{pmatrix} -0.05 \\ -0.95 \end{pmatrix} - \begin{pmatrix} 0.4 & 0.55 & 0.5 & -0.1 & -0.5 & 0.9 & 0.5 & 0.45 \\ -0.6 & -0.45 & -0.5 & 0.9 & -0.5 & 0.1 & 0.5 & 0.55 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right\|_2$$

$$= \left\| \begin{pmatrix} -0.05 \\ -0.95 \end{pmatrix} - \begin{pmatrix} 0.1 \\ -0.9 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} -0.15 \\ -0.05 \end{pmatrix} \right\|_2 = \sqrt{0.15^2 + 0.05^2} = 0.158$$

Hence, error is the same in both cases. We should therefore prefer the sparser solution, which is solution (ii).