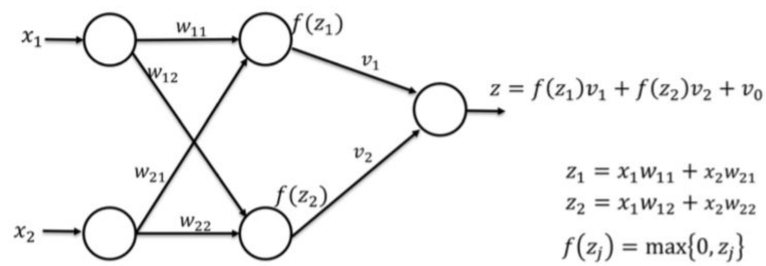
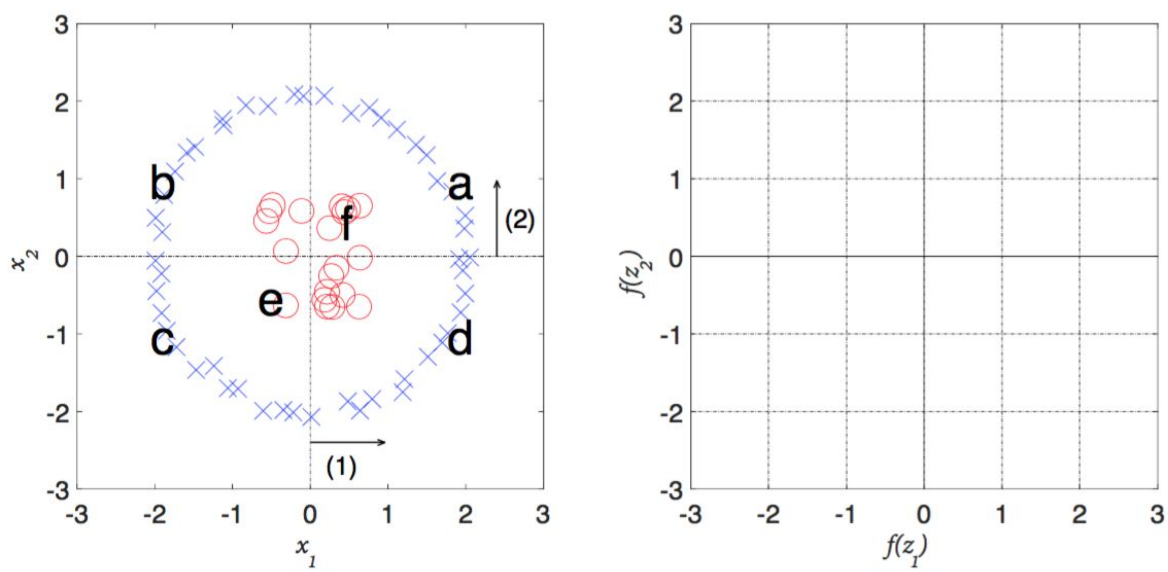
PROBLEM 13

Problem 4 Consider a 2-layer feed-forward neural network that takes in $x \in \mathbb{R}^2$ and has two ReLU hidden units. *Note that the hidden units have no offset parameters.*



(4.1) **(6 points)** The values of the weights in the hidden layer are set such that they result in the z_1 and z_2 "classifiers" as shown in the figure by the decision boundaries and the corresponding normal vectors marked as (1) and (2). Approximately sketch on the right how the input data is mapped to the 2-dimensional space of hidden unit activations $f(z_1)$ and $f(z_2)$. Only map points marked 'a' through 'f'. Keep the letter indicators.



(4.2) **(2 points)** If we keep these hidden layer parameters fixed but add and train additional hidden layers (applied after this layer) to further transform the data, could the resulting neural network solve this classification problem? (Y/N) **( )**

(4.3) **(3 points)** Suppose we stick to the 2-layer architecture but add lots more ReLU hidden units, all of them without offset parameters. Would it be possible to train such a model to perfectly separate these points? (Y/N) **( )**

(4.4) **(3 points)** Initialization of the parameters is often important when training large feed-forward neural networks. Which of the following statements is correct? Check T or F for each statement.

- **( )** If we use tanh or linear units and initialize all the weights to very small values, then the network initially behaves as if it were just a linear classifier
- **( )** If we randomly set all the weights to very large values, or don't scale them properly with the number of units in the layer below, then the tanh units would behave like sign units.
- **( )** A network with sign units cannot be effectively trained with stochastic gradient descent

(4.5) **(3 points)** There are many good reasons to use convolutional layers in CNNs as opposed to replacing them with fully connected layers. Please check T or F for each statement.

- **( )** Since we apply the same convolutional filter throughout the image, we can learn to recognize the same feature wherever it appears.
- **( )** A fully connected layer for a reasonably sized image would simply have too many parameters
- **( )** A fully connected layer can learn to recognize features anywhere in the image even if the features appeared preferentially in one location during training