1)

A) False. Neural networks are non-convex, meaning gradient descent won't guarantee to provide best results.
B) False as doing so would result in the same values being passed onto the next layer, which would make the final output predictable which removes the purpose of using a neural network.
C) True as we use non-linear activation functions to introduce non-linearity into the network to allow us to model responses that are non-linear with respect to inputs; This results in non-linear decision boundaries that we can use to predict.
D) True. Forward passes involve multiplication or using activation functions. However, backpropagation involves updating weight and bias and calculating gradients, which is more likely to be complex and lead to larger time complexity.
E) False. There are a lot of considerations done when picking a model to best fit it's problem(such as designing it's structure and picking it's hyperparameters), which may not reflect all possible problems. Furthermore, there could be simpler models, possibly linear regression, that can occasionally accomplish what a neural network can do.

2)

By the definition of kernel:

$$K(x, x') = \phi(x) * \phi(x')$$

For a general i-th component:

$$= \left(\frac{1}{\sqrt{i!}} e^{\frac{-x^2}{2}} x^i\right) * \left(\frac{1}{\sqrt{i!}} e^{\frac{-x'^2}{2}} x'^i\right)$$

$$= \frac{1}{\sqrt{i!}} \frac{1}{\sqrt{i!}} e^{\frac{-x^2}{2}} e^{\frac{-x'^2}{2}} x^i x'^i$$

$$= \frac{1}{i!} e^{\frac{-(x^2+x'^2)}{2}} x^i x'^i$$

Combining all components components together(as only the value i changes for each component):

$$= e^{\frac{-(x^2+x'^2)}{2}} \sum_{i=0}^{\infty} \frac{(xx')^i}{i!}$$

Using the definition of Taylor expansion, where $e^y = \sum_{n=0}^{\infty} \frac{y^n}{n!}$:

$$= e^{\frac{-(x^2+x'^2)}{2}} * e^{xx'} = e^{\frac{-(x^2+x'^2)}{2}} * e^{\frac{2xx'}{2}}$$

$$= e^{\frac{-x^2-x'^2+2xx'}{2}}$$

$$= e^{\frac{-(x-x')^2}{2}}$$

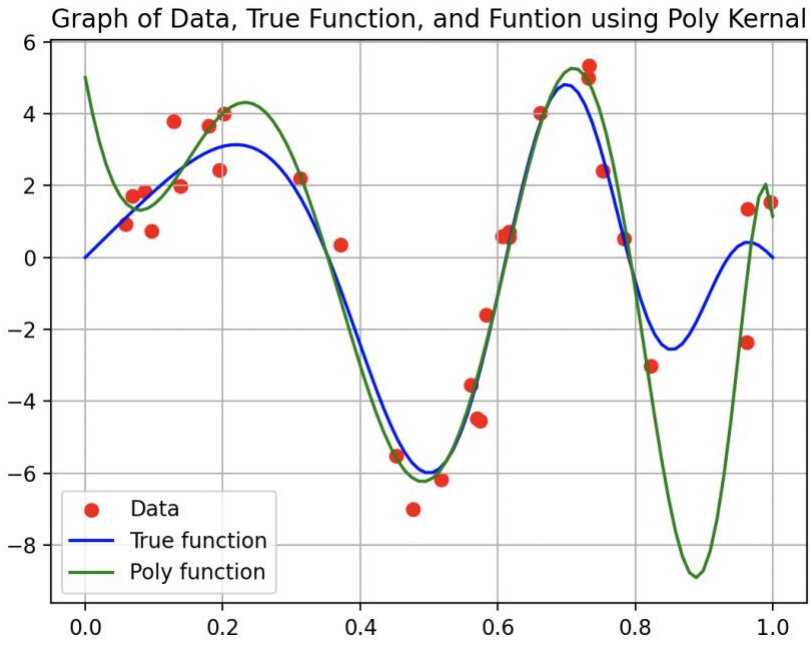
3)

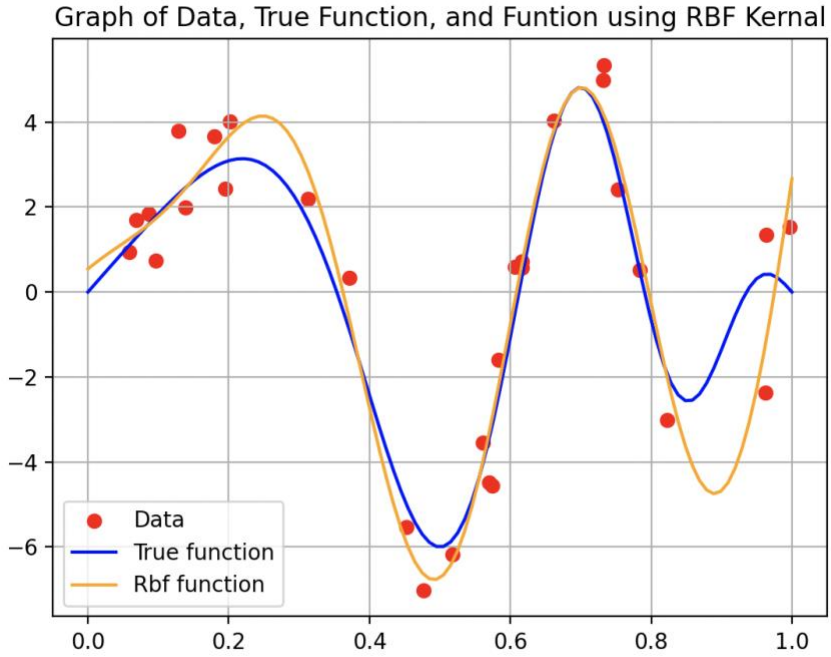
A)

For poly kernel, the best lambda and d hyperparameters are lamba = 1e-05 and d = 15.0.

For RBF kernel, the best lambda is 0.001 and gamma is 11.201924992299844.

B)

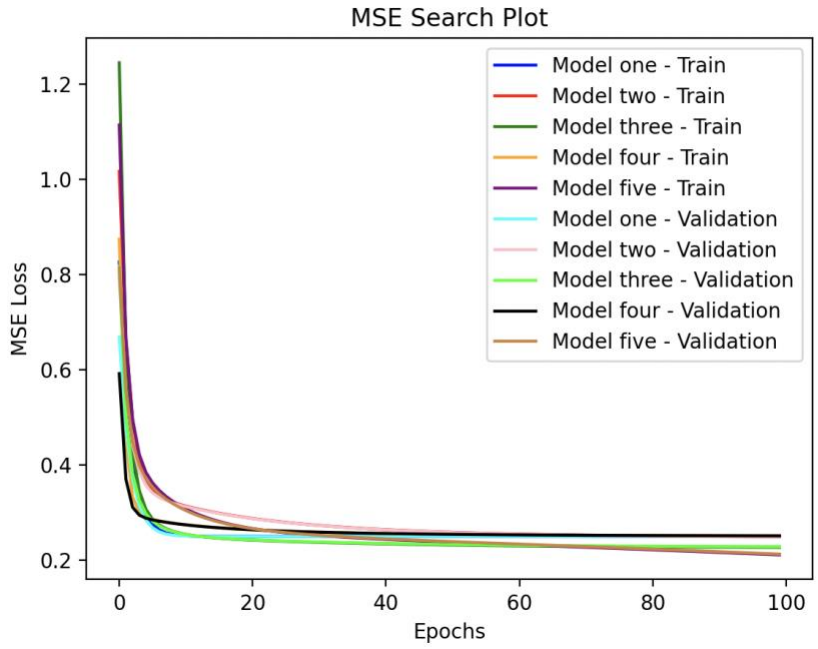


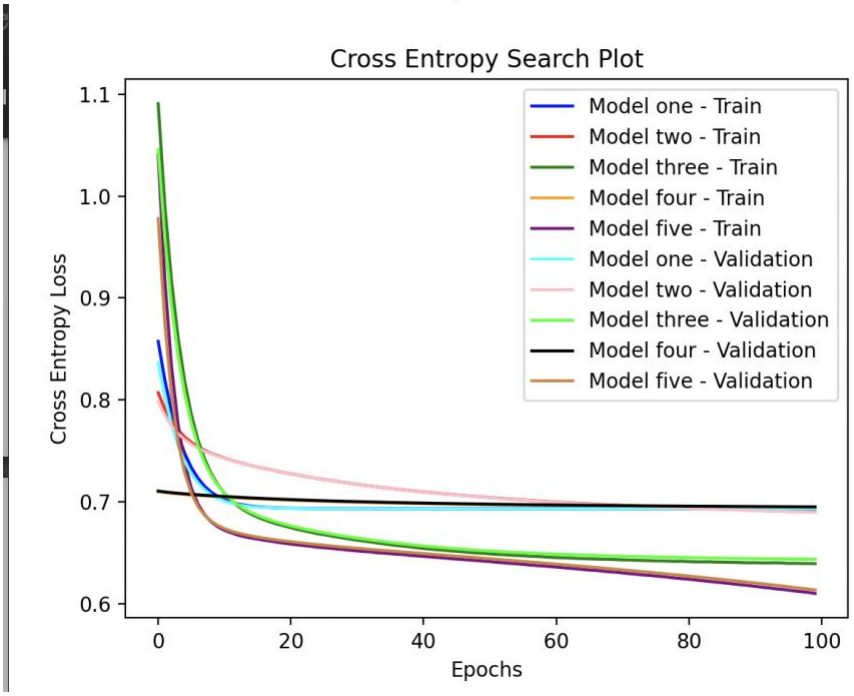


4)

A)

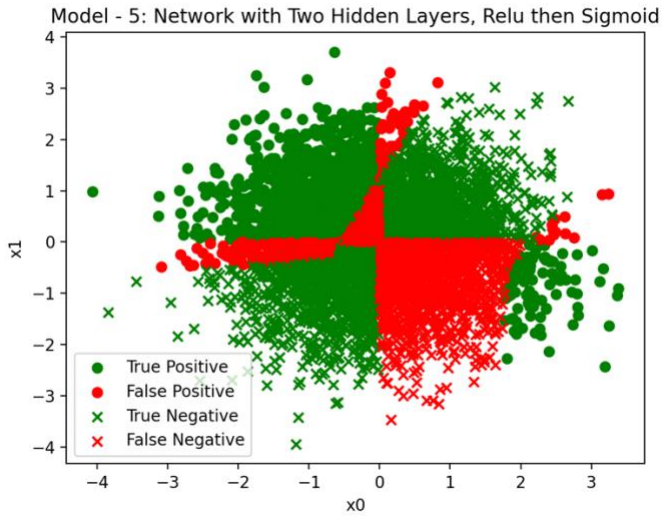
B)


MSE Search Plot
MSE Loss
Epochs
Model one - Train
Model two - Train
Model three - Train
Model four - Train
Model five - Train
Model one - Validation
Model two - Validation
Model three - Validation
Model four - Validation
Model five - Validation

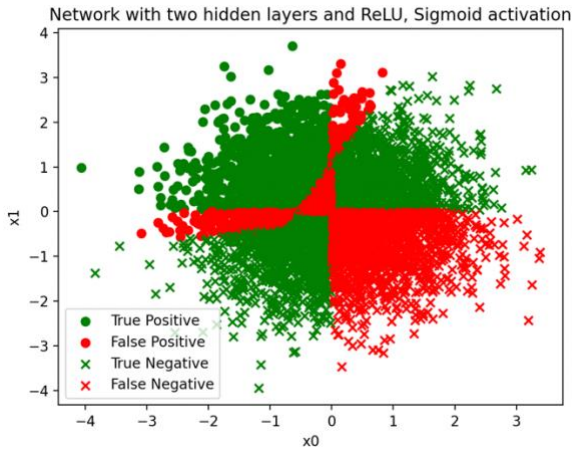

Cross Entropy Search Plot
Cross Entropy Loss
Epochs
Model one - Train
Model two - Train
Model three - Train
Model four - Train
Model five - Train
Model one - Validation
Model two - Validation
Model three - Validation
Model four - Validation
Model five - Validation

C)

For the MSE search, the best performing architecture was model 5, Network with two hideen layers and ReLu, Sigmoid activation after corresponding hidden layers. It's accuracy was 0.6562.

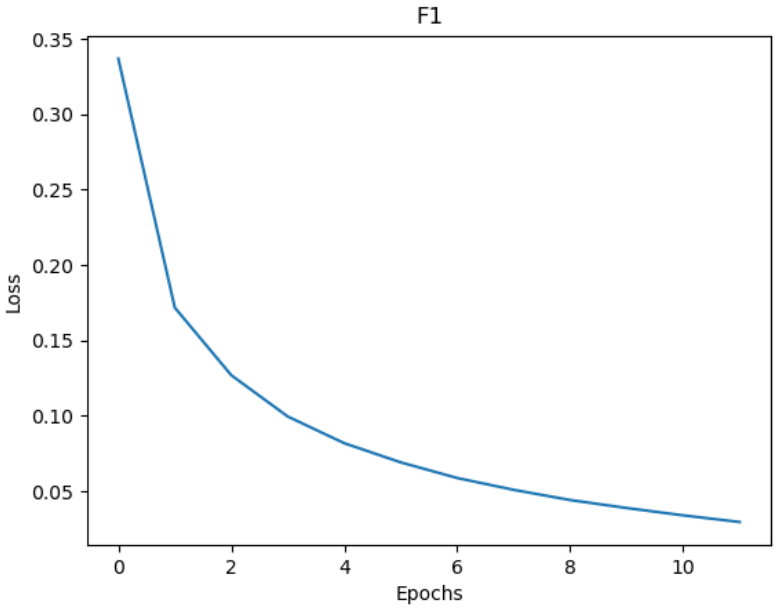


For cross entropy search, the best performing model was the fifth mode as specified in the spec as well. It's accuracy was 0.6384.



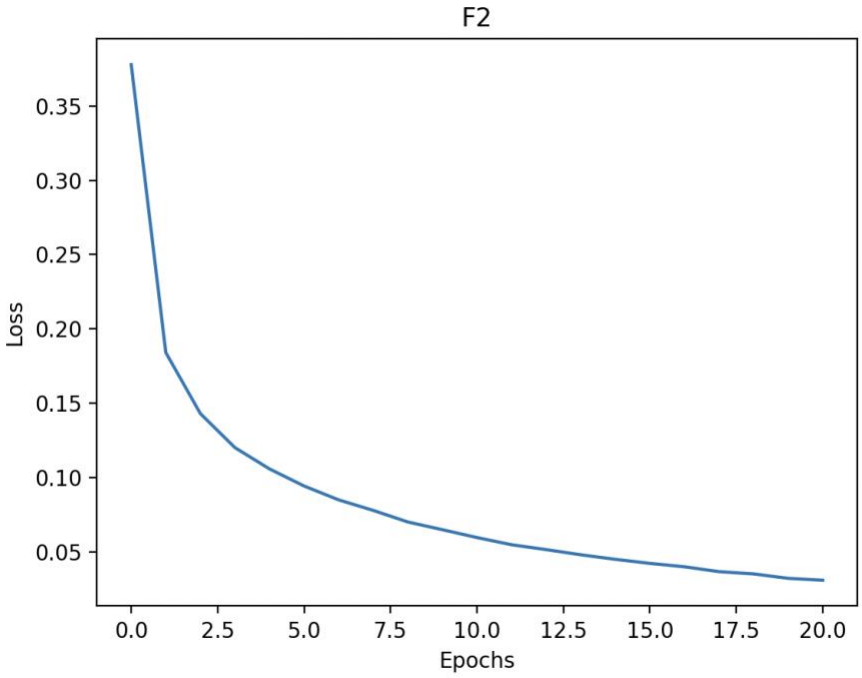
5)

A)



F1 is the wide/shallow network. The loss on the test data was 0.0927. The Accuracy on the test data was 0.9733.

B)



F2 is the deep/narrow network. The loss on the test data was 0.1320. The Accuracy was 0.9684.

C)

The total number of parameters for the F1 model is 50890. The total number of parameters for the F2 model is 26506.

I wouldn't say one approach is better than the other as in exchange for using more space, F1 had a better test accuracy. There are merits to both models, but if you want the best outcome, F1 would be the preferable model.

6)

27 hours