A1

(a)

False. Deep neural networks have non-convex loss surfaces, so gradient descent does not guarantee the global optimum.

(b)

False. Initializing all weights to zero prevents breaking symmetry, causing identical updates and hindering training.

(c)

True. Non-linear activation functions enable the network to learn non-linear decision boundaries, which would be impossible with purely linear transformations.

(d)

False. Although the backward pass is more expensive than the forward pass, it is typically of the same order of magnitude and not prohibitively larger (big O time is the same).

(e)

False. Neural networks are powerful and extensible, but they are not always the best choice for every circumstance due to factors like data requirements, computational cost, and interpretability.

A2

We are given the infinite-dimensional feature map

$$\phi(x) = \left[\frac{1}{\sqrt{0!}}e^{-x^2/2}x^0,\ \frac{1}{\sqrt{1!}}e^{-x^2/2}x^1,\ \frac{1}{\sqrt{2!}}e^{-x^2/2}x^2,\ \dots\right].$$

For inputs x and x', we have

$$\phi(x) = \left[\frac{e^{-x^2/2}x^0}{\sqrt{0!}},\ \frac{e^{-x^2/2}x^1}{\sqrt{1!}},\ \frac{e^{-x^2/2}x^2}{\sqrt{2!}},\ \dots\right],$$

$$\phi(x') = \left[\frac{e^{-x'^2/2}x'^0}{\sqrt{0!}},\ \frac{e^{-x'^2/2}x'^1}{\sqrt{1!}},\ \frac{e^{-x'^2/2}x'^2}{\sqrt{2!}},\ \dots\right].$$

The inner product is computed as

$$\langle \phi(x), \phi(x')\rangle = \sum_{n=0}^{\infty}\left(\frac{e^{-x^2/2}x^n}{\sqrt{n!}}\right)\left(\frac{e^{-x'^2/2}x'^n}{\sqrt{n!}}\right).$$

Simplifying:

$$\langle \phi(x), \phi(x')\rangle = e^{-x^2/2}e^{-x'^2/2}\sum_{n=0}^{\infty}\frac{(xx')^n}{n!}$$

After the expression

$$\sum_{n=0}^{\infty}\frac{(xx')^n}{n!} = e^{xx'},$$

(Taylor series expansion of e^x), we obtain

$$\langle \phi(x), \phi(x')\rangle = e^{-x^2/2}e^{-x'^2/2}e^{xx'} = e^{-\frac{x^2+x'^2}{2}+xx'}.$$

Writing xx' as a fraction with denominator 2, we have

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

Noting that

$$x^2 + x'^2 - 2xx' = (x-x')^2,$$

we obtain

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

Hence,

$$\boxed{\langle \phi(x), \phi(x')\rangle = e^{-\frac{(x-x')^2}{2}}.}$$

■

A3

(a)

RBF Kernel best $\lambda = 1.6681\text{e-}3$, best $\gamma = 10.541635373659384$.
Polynomial Kernel best $\lambda = 1\text{e-}5$, best degree $d = 15$.

(b)

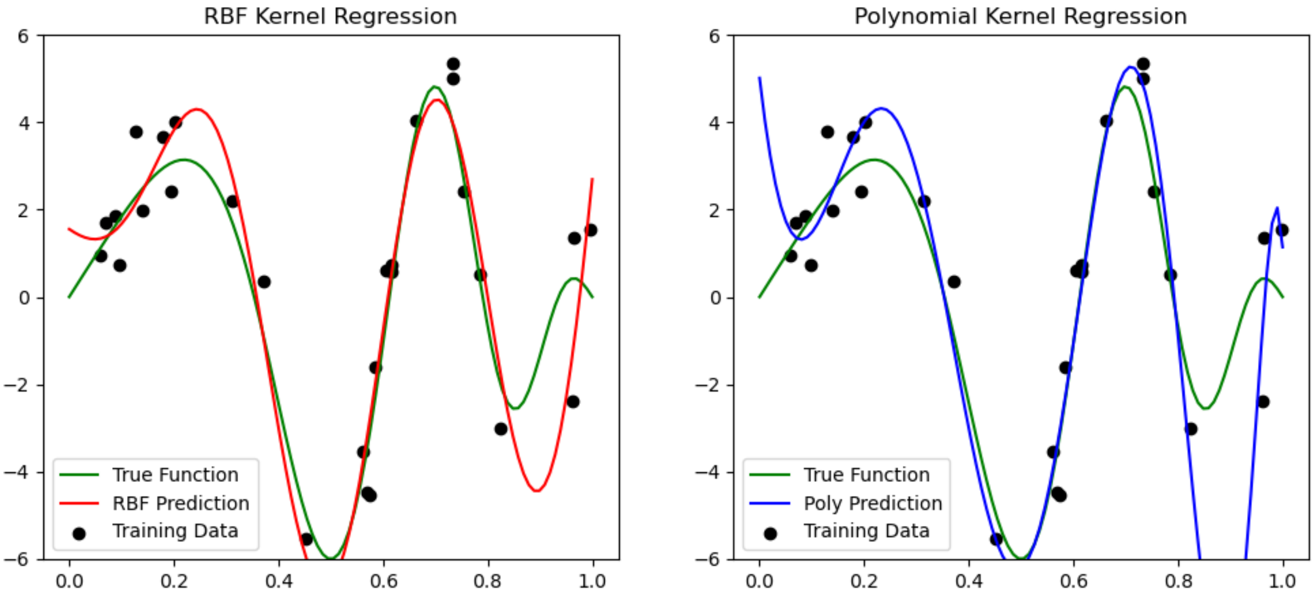

Figure 1: RBF vs. Polynomial Kernel Regression Plots

A4

(b)

Cross-Entropy Loss: Figure **??** shows the training and validation loss curves for all models trained with the cross-entropy loss function.

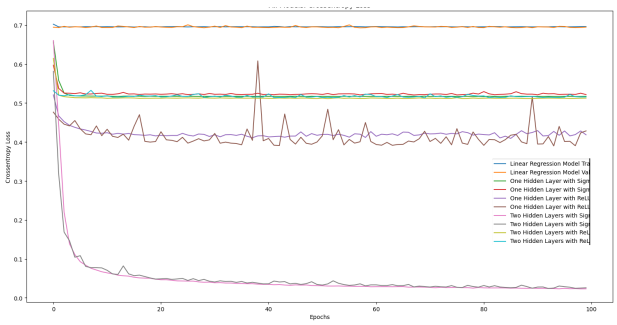

Figure 1: Training and Validation Loss Curves for All Models (Cross-Entropy Loss).

MSE Loss: Figure **??** shows the training and validation loss curves for all models trained with the MSE loss function.

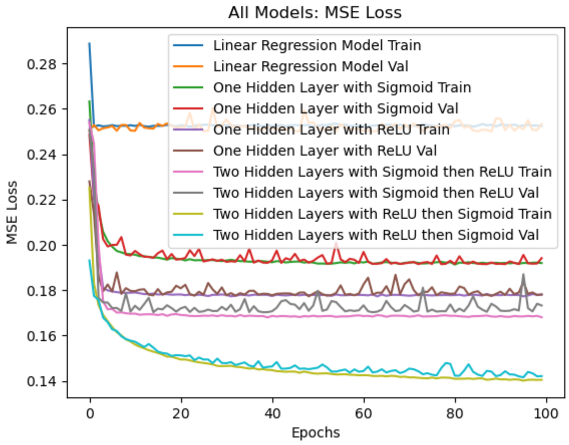

Figure 2: Training and Validation Loss Curves for All Models (MSE Loss).

(c)

Cross-Entropy Loss: The best model trained with cross-entropy loss was a network with two hidden layers using Sigmoid then ReLU activation functions.

It achieved a test set accuracy of 0.9928. The scatter plot of its predictions is shown in Figure **??**.

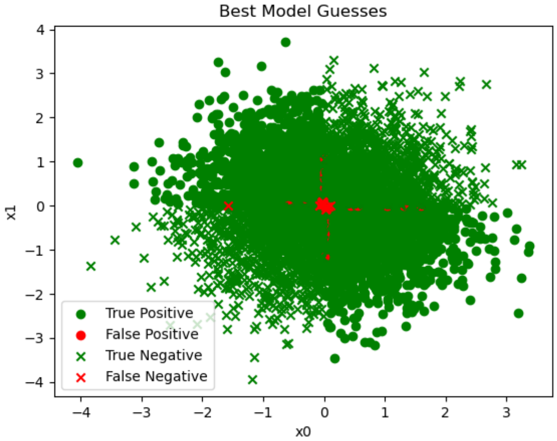

Figure 3: Scatter Plot of Best Cross-Entropy Model (Two Hidden Layers with Sigmoid then ReLU).

MSE Loss: The best model trained with MSE loss was a network with two hidden layers using ReLU then Sigmoid activation functions. It achieved a test set accuracy of 0.7212. The scatter plot of its predictions is shown in Figure **??**.

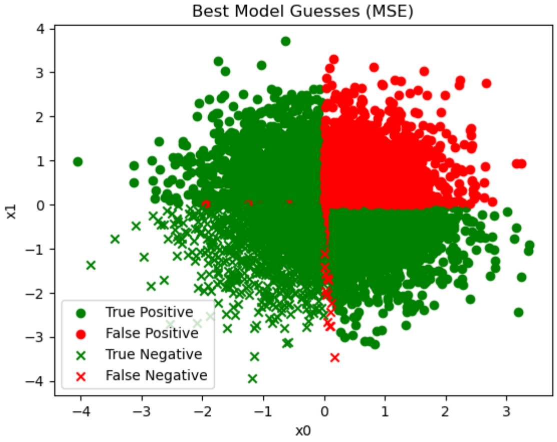

Figure 4: Scatter Plot of Best MSE Model (Two Hidden Layers with ReLU then Sigmoid).

A5

(a)

F1 Results:

- Test Accuracy: 0.9741
- Test Loss: 0.0885
- Total Parameters: $50,890$

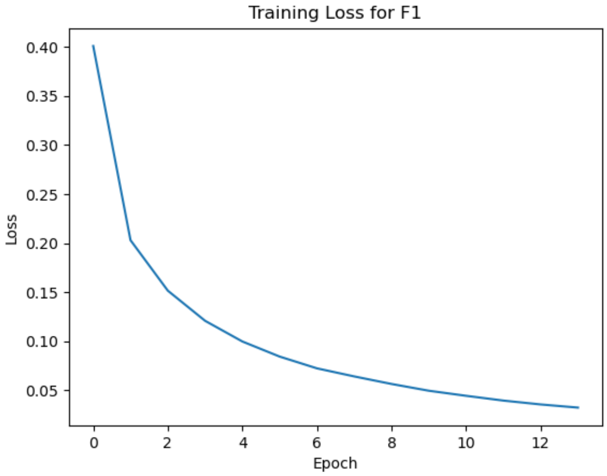

Figure 1: Training loss versus epoch for the F1 model (shallow and wide).

(b)

F2 Results:

- Test Accuracy: 0.9780
- Test Loss: 0.0767
- Total Parameters: $109,386$

(c)

The F1 model, being shallow and wide, has fewer parameters ($50,890$) compared to the F2 model, which is deeper and narrower ($109,386$). Despite having fewer

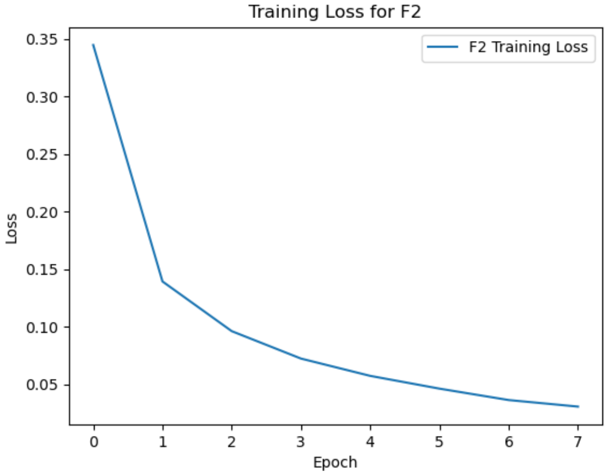

Figure 2: Training loss versus epoch for the F2 model (narrow and deep).

parameters, the F1 model achieves a high test accuracy of 0.9741, which is only slightly lower than the F2 model's test accuracy of 0.9780.

The F2 model's deeper architecture allows it to capture more complex patterns in the data, which likely contributes to its slightly better performance. However, this comes at the cost of a significantly higher number of parameters, which increases the computational and memory requirements.

A6

I spent about 20 hours on this homework.