Homework #3

CSE 446/546: Machine Learning
Zuo Wang

Question 1

(a.)

We should decrease σ because decreasing the bandwith allows us to have a more localized and complex model. Mathmatically speaking, decreasing σ makes the kernel value smaller (exponential of a larger negative value) therefore the penalty term is smaller, resulting in a more complex model.

(b.)

False. Most neural networks actually have non-convex loss functions.

(c.)

False. If we initialize all weights to zero when training a deep neural network, the gradient of all the weights will be the same during backpropagation, resulting in all neurons learning the same features and updating their weights identically.

(d.)

True. Linear activation functions would result in a network that can only learn linear decision boundaries.

(e.)

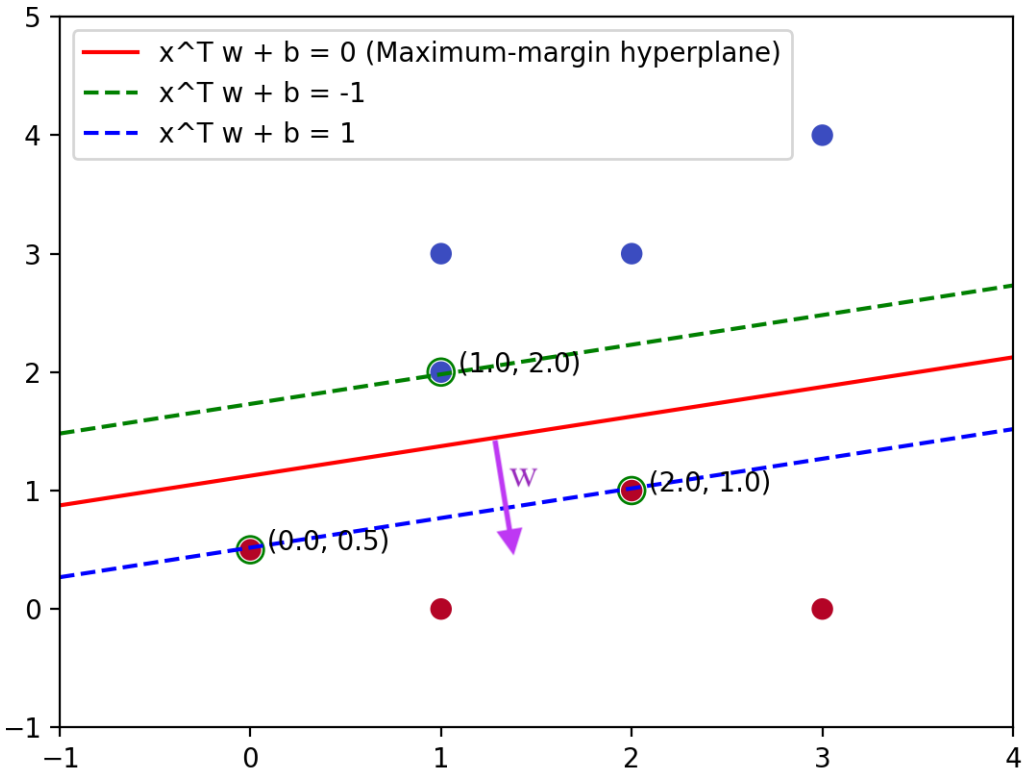
True. The backward pass step requires computing the gradients of the loss with respect to the parameters, which involves matrix multiplications, and derivating activation function. And that is more time-consuming compared to the forward pass step.

(f.)

False. While neural network is consider one of the best models in most cases, it is not always the best choice for every circumstance. Generally speaking, neural networks are more complex, require a longer runtime, are better for larger dataset, and tend to overfit. In some cases, simple models with regularization might just perform better.

Question 2

(a.)



As shown, the support vectors are (0, 0.5), (2, 1), and (1, 2)

(b.)

From the support vectors and their corresponding labels, we know that:

$$\begin{bmatrix}1\\2\end{bmatrix} w + b = -1,\ \begin{bmatrix}0\\0.5\end{bmatrix} w + b = 1,\ \begin{bmatrix}2\\1\end{bmatrix} w + b = 1$$

Looking at the last two equations, we can construct the matrix:

$$\begin{aligned} w &= \left[\begin{array}{cc|c} 0 & 0.5 & 1-b \\ 2 & 1 & 1-b \end{array}\right] \\ &= \left[\begin{array}{cc|c} 2 & 1 & (1-b) \\ 0 & 1 & 2(1-b) \end{array}\right] \\ &= \left[\begin{array}{cc|c} 1 & 0 & -\frac{1}{2}(1-b) \\ 0 & 1 & 2(1-b) \end{array}\right] \\ &= \begin{bmatrix} -\frac{1}{2}(1-b) & 2(1-b) \end{bmatrix} \end{aligned}$$

And we plug w back into $\begin{bmatrix}1\\2\end{bmatrix} w + b = -1$:

$$\begin{aligned} \begin{bmatrix}1\\2\end{bmatrix} \cdot \begin{bmatrix} -\frac{1}{2}(1-b) & 2(1-b) \end{bmatrix} + b &= -1 \\ -\frac{1}{2}(1-b) + 4(1-b) + b &= -1 \\ \frac{7}{2}(1-b) + b &= -1 \\ -\frac{5}{2}b &= -\frac{9}{2} \\ b &= \frac{9}{5} \end{aligned}$$

$$\boxed{b = \frac{9}{5} \quad w = \begin{bmatrix} \frac{2}{5} & -\frac{8}{5} \end{bmatrix}}$$

(c.)

Denote the two hyperplanes $x^T w + b = 1$ and $x^T w + b = -1$ as H_1 and H_2, respectively.
Let's pick a random point x_0 on H_1, we know that Now, the projection of x_0 onto H_1 is given by $P_H(x_0) = x_0 - \frac{w^T x_0 - c}{||w||_2^2} w$ where the offset $c = w^T x_0 = -1 - b$.
And since the distance between H_1 and H_2 is euqal to the distance x_0 on H_1 and it's projection on H_2, we can just calculate $||x_0 - P_H(x_0)||_2$ to determine the distance between H_1 and H_2:

$$\begin{aligned}
\|x_0 - P_H(x_0)\|_2^2 &= \left\| x_0 - \left(x_0 - \frac{w^T x_0 - c}{||w||_2^2} w \right) \right\|_2 \\
&= \left\| \frac{w^T x_0 - (-1 - b)}{||w||_2^2} w \right\|_2 \\
&= \left\| \frac{(1 - b) + 1 + b}{||w||_2^2} w \right\|_2 \\
&= \left\| \frac{2}{w} \right\|_2 \\
&= \frac{2}{\|w\|_2}
\end{aligned}$$

Therefore, the distance between the two separating hyperplanes is $\frac{2}{\|w\|_2}$

Question 3

$$\begin{aligned}
\phi(x) \cdot \phi(x') &= \sum_{i=0}^{\infty} \left(\frac{1}{\sqrt{i!}} e^{\frac{-x^2}{2}} x^i \right) \cdot \left(\frac{1}{\sqrt{i!}} e^{\frac{-x'^2}{2}} (x')^i \right) \\
&= \sum_{i=0}^{\infty} \frac{1}{i!} e^{\frac{-(x^2+x'^2)}{2}} x^i (x')^i \\
&= e^{\frac{-(x^2+x'^2)}{2}} \sum_{i=0}^{\infty} \frac{1}{i!} x^i (x')^i
\end{aligned}$$

Because the Taylor expansion of $e^z = \sum_{n=0}^{\infty} \frac{1}{n!} z^n$, the summation above just becomes $e^{xx'}$, therefore:

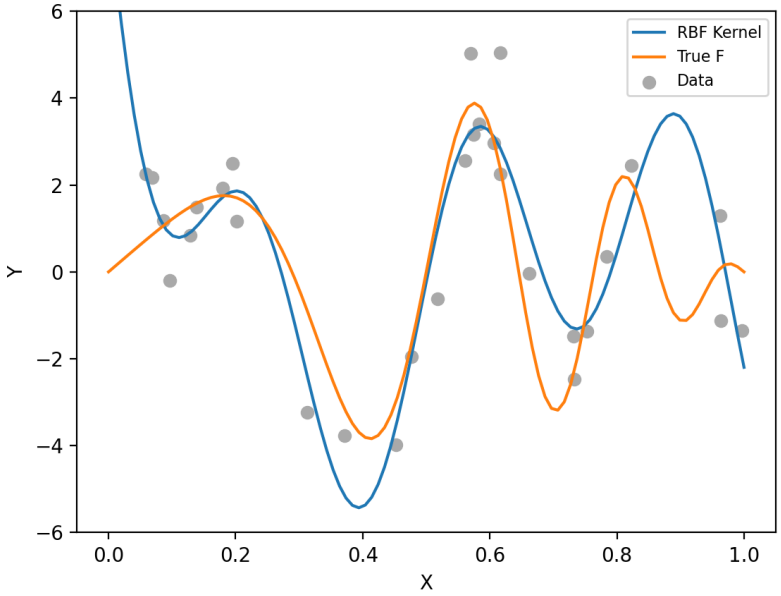
$$\begin{aligned}
\phi(x) \cdot \phi(x') &= e^{\frac{-(x^2+x'^2)}{2}} \cdot e^{xx'} \\
&= e^{\frac{-x^2-x'^2+2xx'}{2}} \\
&= e^{\frac{-(x-x')^2}{2}}
\end{aligned}$$

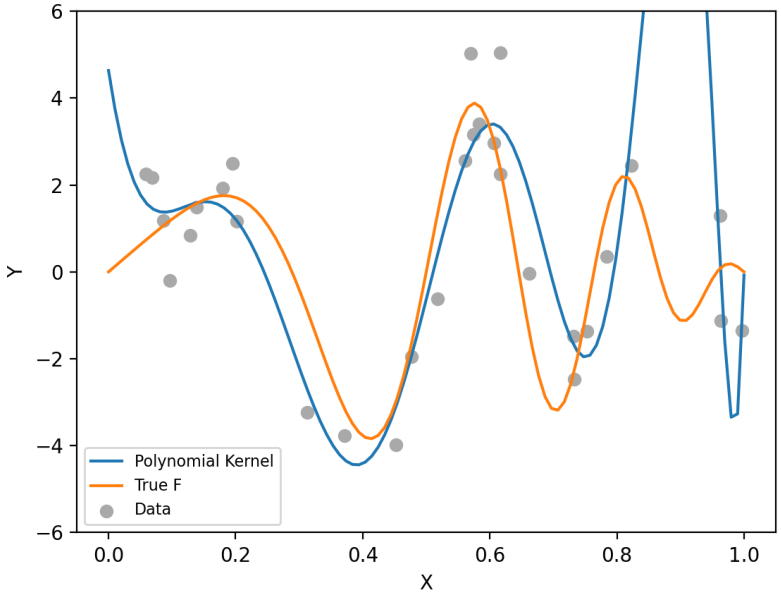
Question 4

(a.)

For the RBF kernel, $\lambda_{optimal} = 10^{-3},\ \gamma_{optimal} = 10.5416$
For the polynomial kernel, $\lambda_{optimal} = 10^{-3},\ d_{optimal} = 20.5102$

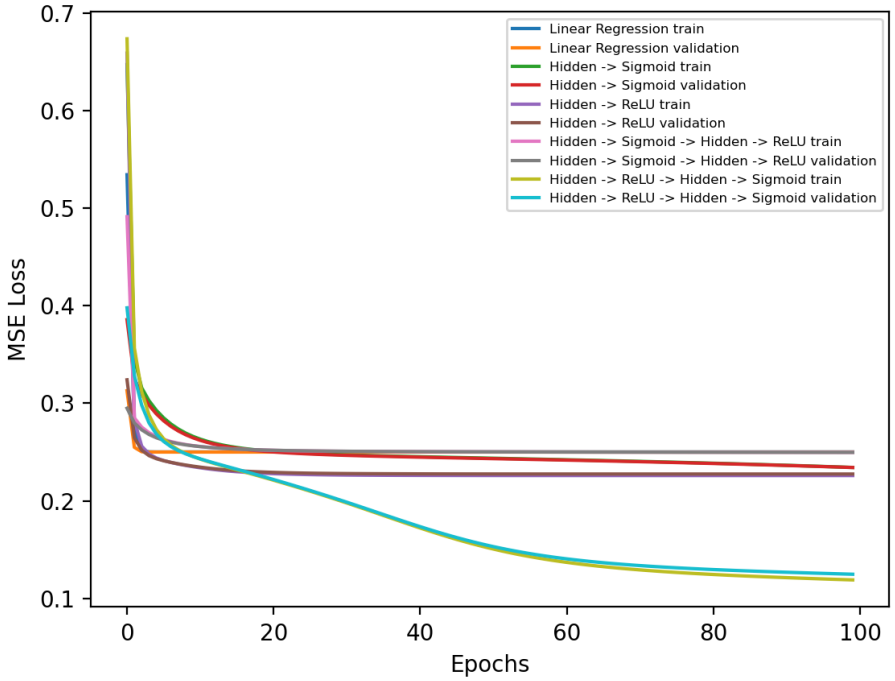
(b.)

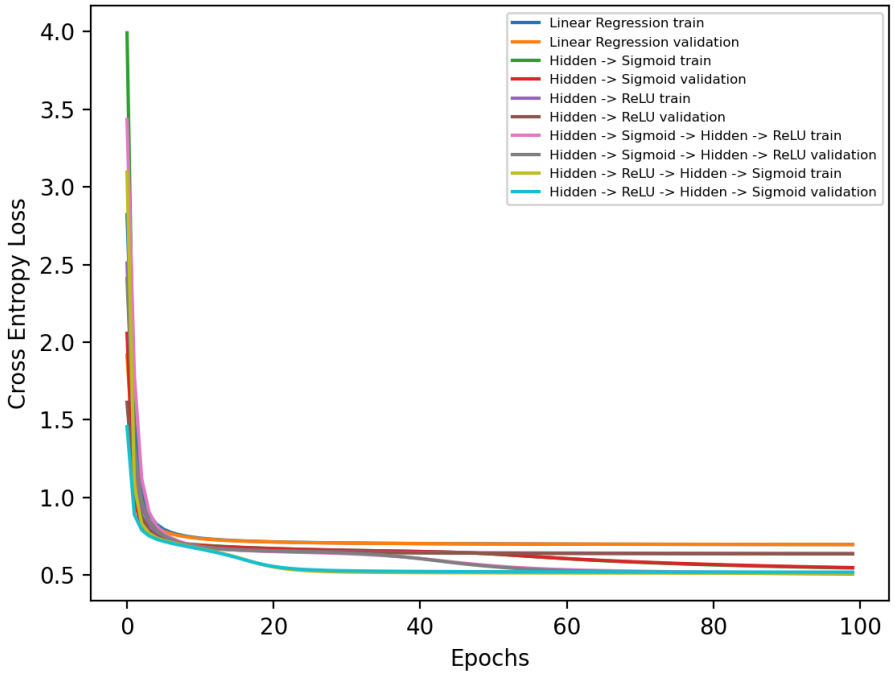




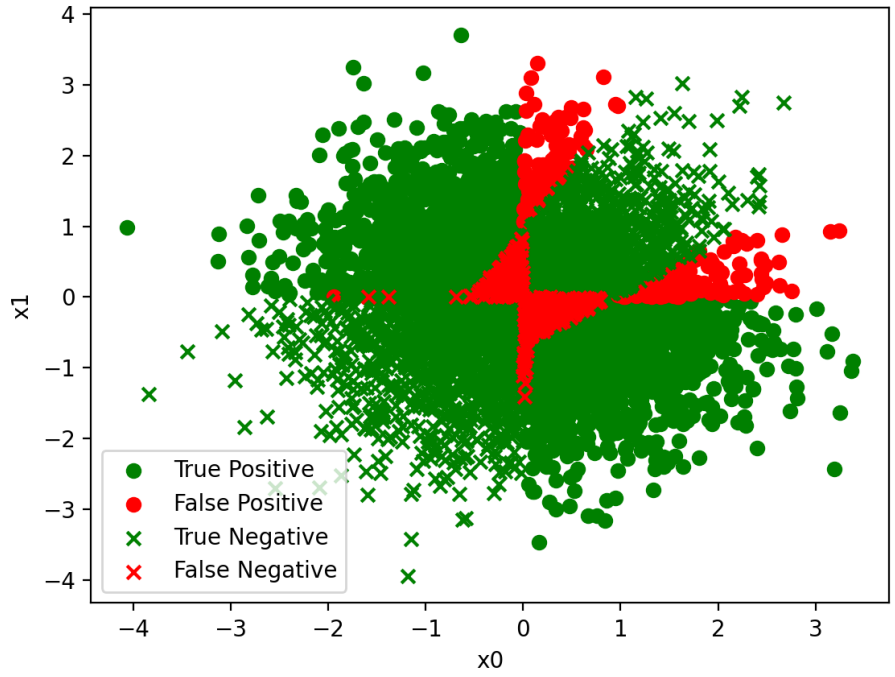
Question 5

(b.)


Linear Regression train
Linear Regression validation
Hidden -> Sigmoid train
Hidden -> Sigmoid validation
Hidden -> ReLU train
Hidden -> ReLU validation
Hidden -> Sigmoid -> Hidden -> ReLU train
Hidden -> Sigmoid -> Hidden -> ReLU validation
Hidden -> ReLU -> Hidden -> Sigmoid train
Hidden -> ReLU -> Hidden -> Sigmoid validation
MSE Loss
Epochs
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100


Linear Regression train
Linear Regression validation
Hidden -> Sigmoid train
Hidden -> Sigmoid validation
Hidden -> ReLU train
Hidden -> ReLU validation
Hidden -> Sigmoid -> Hidden -> ReLU train
Hidden -> Sigmoid -> Hidden -> ReLU validation
Hidden -> ReLU -> Hidden -> Sigmoid train
Hidden -> ReLU -> Hidden -> Sigmoid validation
Cross Entropy Loss
Epochs
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
20
40
60
80
100

(c.)

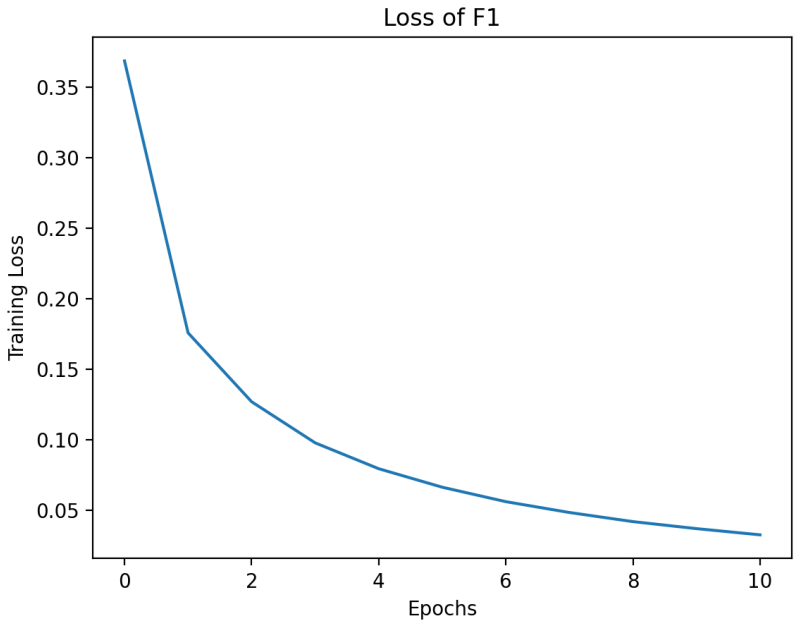


Best model for MSE loss: Hidden → ReLU → Hidden → Sigmoid (accuracy socre of 0.8764)

Best model for Cross Entropy loss: Hidden → Sigmoid → Hidden → ReLU (accuracy socre of 0.7398)

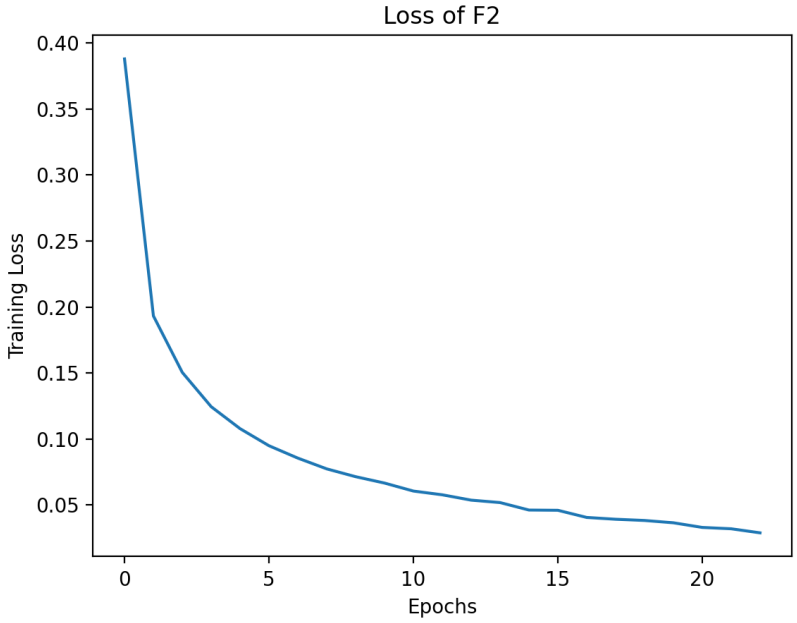
Question 6

(a.)



Loss: 0.09190159004156331
Accuracy: 0.9736

(b.)



Loss: 0.15115708102590775
Accuracy: 0.9674

(c.)

The number of parameters for F1 is 50890, and it's 26506 for F2.
In comparison to F2, F1 model is more complex, achieved higer accuracy and lower loss, and converged in less epoches. Therefore in this case, I would say that F1(wide & shallow) is better. However, we can only conclude that for this specific dataset. Both approaches have their own advantages and considerations, and their performance, again, depend on the specific problem and dataset we're dealing with.