CS/ECE/ME 532

Unit 6 Practice Problems

1. Neural Network Basic. Consider the following neural network.

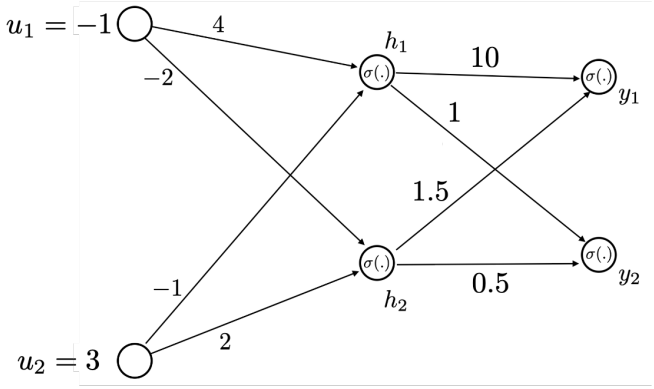

Figure 1: A Neural Network Architecture

a) How many hidden layers are in this network? How many neurons are in this network? How many outputs does this network have? What is $\sigma(\cdot)$ called?

SOLUTION: One hidden layer/6 neurons including the input neurons/2 outputs/activation function

b) Assume that the neural network shown above uses an activation called *Leaky ReLU*, defined as follows:

$$\sigma(z) = \begin{cases} z & \text{if } z > 0 \\ 0.01z & \text{otherwise.} \end{cases}$$

Note that Leaky ReLU behaves differently from ReLU when the input value is less than 0. The values labeled h_1, h_2, y_1, y_2 are the outputs of the corresponding nodes. Find the numerical values of h_1, h_2, y_1, y_2 for an input feature $u_1 = -1, u_2 = 3$.

SOLUTION:

$$h_1 = \sigma(-4 - 3) = \sigma(-7) = -0.07$$
$$h_2 = \sigma(2 + 6) = \sigma(8) = 8$$
$$y_1 = \sigma(-0.7 + 12) = \sigma(11.3) = 11.3$$
$$y_2 = \sigma(-0.07 + 4) = \sigma(3.93) = 3.93.$$

2. **SGD to learn the weights for a single neuron** We use the single neuron shown in the figure for classification. Here x_j^i is the j-th feature in the i-th training sample and the output is $\widehat{y}^i = \sigma\left(\sum_{j=0}^P w_j x_j^i\right)$, where the activation function is *ReLU6*, defined as follows:

$$\sigma(z) = \min(\max(0, z), 6).$$

Note that ReLU6 behaves differently from ReLU when the input value is larger than 6.

Now suppose we use ridge regression for the loss function

$$f^i(\boldsymbol{w}) = \frac{1}{2}(\widehat{y}^i - y^i)^2 + \lambda \sum_{j=0}^P w_j^2.$$

Derive the gradient for the update step, $\nabla f^{i_t}(\boldsymbol{w}^{(t)})$ and write the update equation for $\boldsymbol{w}^{(t+1)}$.

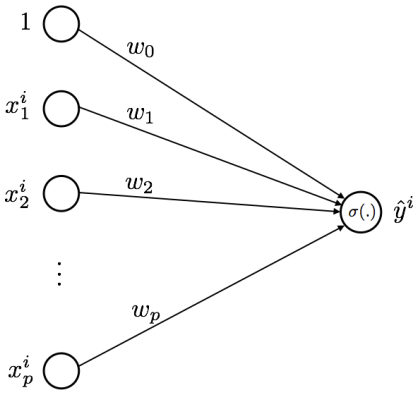

Figure 2: A Neuron

SOLUTION: Let $x_0^i = 1$ for all i. Note that

$$\frac{df^{i_t}(\boldsymbol{w}^{(t)})}{dw_k} = (\widehat{y}^{i_t} - y^{i_t})\frac{d\widehat{y}^{i_t}}{dw_k} + 2\lambda w_k.$$

To compute $\frac{d\widehat{y}^{i_t}}{dw_k}$, we use the chain rule:

$$\frac{d\widehat{y}^{i_t}}{dw_k} = \frac{d\widehat{y}^{i_t}}{dz^{i_t}}\frac{dz^{i_t}}{dw_k},$$

where $z^{i_t} = \sum_{j=0}^P w_j x_j^{i_t}$. By the definition of ReLU6, it is clear $\frac{d\widehat{y}^{i_t}}{dz^{i_t}} = \mathbb{1}\{0 \le z^{i_t} \le 6\} = \mathbb{1}\{0 \le \sum_{j=0}^P w_j x_j^{i_t} \le 6\}$, where $\mathbb{1}(\text{condition}) := \begin{cases} 1 & \text{if condition is true} \\ 0 & \text{otherwise} \end{cases}$. Also, it is clear $\frac{dz^{i_t}}{dw_k} = x_k^{i_t}$. Thus, $\frac{d\widehat{y}^{i_t}}{dw_k} = x_k^{i_t} \cdot \mathbb{1}\{0 \le \sum_{j=0}^P w_j x_j^{i_t} \le 6\}$.

Define $\delta^{i_t} = (\widehat{y}^{i_t} - y^{i_t}) \cdot \mathbb{1}\{0 \le \sum_{j=0}^P w_j x_j^{i_t} \le 6\}$ so we can write

$$\nabla f^{i_t}(\boldsymbol{w}^{(t)}) = \delta^{i_t}\boldsymbol{x}^{i_t} + 2\lambda \boldsymbol{w}^{(t)}$$

and thus the update is

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \alpha_t \delta^{i_t}\boldsymbol{x}^{i_t} - 2\alpha_t\lambda \boldsymbol{w}^{(t)}$$

3. Kernel Regression. Consider the Gaussian kernel regression with ℓ_2 regulariztaion. The first hyperparameter is σ for the Gaussian kernel $K(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(-||\boldsymbol{u} - \boldsymbol{v}||_2^2/(2\sigma^2)\right)$. The second hyperparameter is λ which is the weight for the ℓ_2 regularization term.

Now, consider the following five choices:

a) $\lambda = 0.01, \sigma = 0.04$

b) $\lambda = 0.01, \sigma = 0.2$

c) $\lambda = 0.01, \sigma = 1$

d) $\lambda = 1, \sigma = 0.04$

e) $\lambda = 1, \sigma = 0.2$

Find the correct mapping between each of these choices and each of the following options (each row corresponds one choice).

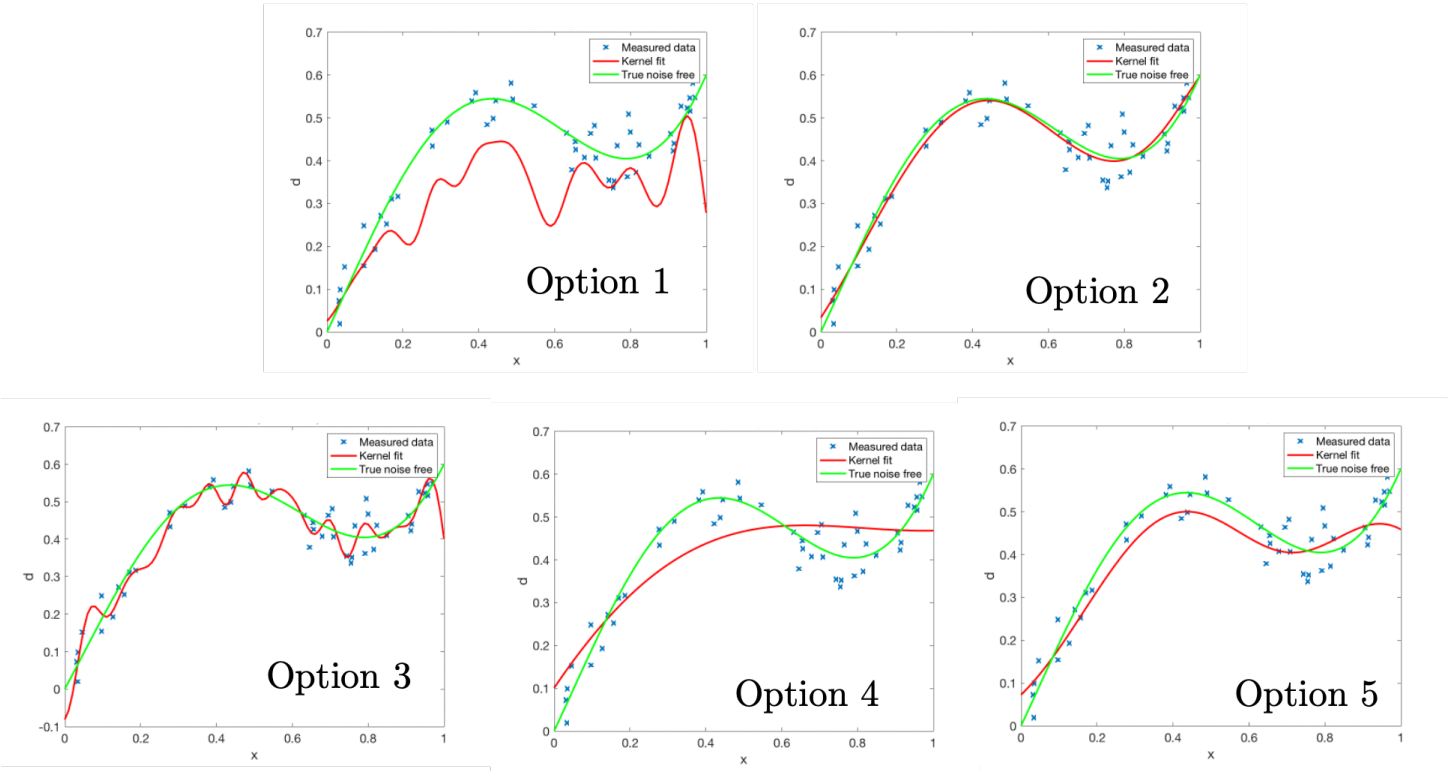

Figure 3: Kernel regression results with five different hyperparameters

SOLUTION: Let us first consider λ. λ is associated with the 2-norm penalty in the ridge regression problem, so as λ increases, the 2-norm penalty receives increasing emphasis. Hence $\boldsymbol{\alpha}$ for the $\lambda = 1$ case will have smaller norm than for the $\lambda = 0.01$ case. This results in underestimation (and a slightly smoother solution) when $\lambda = 1$. From this, one can see Option 1 and Option 5 correspond to $\lambda = 1$, and Option 2, 3, 4 correspond to $\lambda = 0.01$.

Let's now consider σ, which determines the width of the kernels. Larger values of σ result in wider kernels, which will lead to a smoother solution. Smaller values of σ results in narrower kernels, fully overfitting the measured data. Between Option 1 and Option 5, Option 5's fit is smoother, so

Option 5 corresponds to case e) and Option 1 corresponds to case d). Among those with $\lambda = 0.01$, Option 4 has the smoothest fit, and Option 3 fully overfits the data, and Option 2 is something in between them. Thus, Option 4 corresponds to case c), Option 3 corresponds to case a), and Option 2 corresponds to case b).

4. **Kernel Classification.** Consider the kernel classification with the squared error loss using the Gaussian kernel $K(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(-||\boldsymbol{u} - \boldsymbol{v}||_2^2/(2\sigma^2)\right)$ for the following dataset.

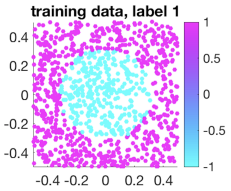

Figure 4: Binary classification dataset

Now, consider the following three choices:

a) $\sigma = 5$

b) $\sigma = 0.05$

c) $\sigma = 0.005$

Find the correct mapping between each of these choices and each of the following options (each row corresponds one choice).

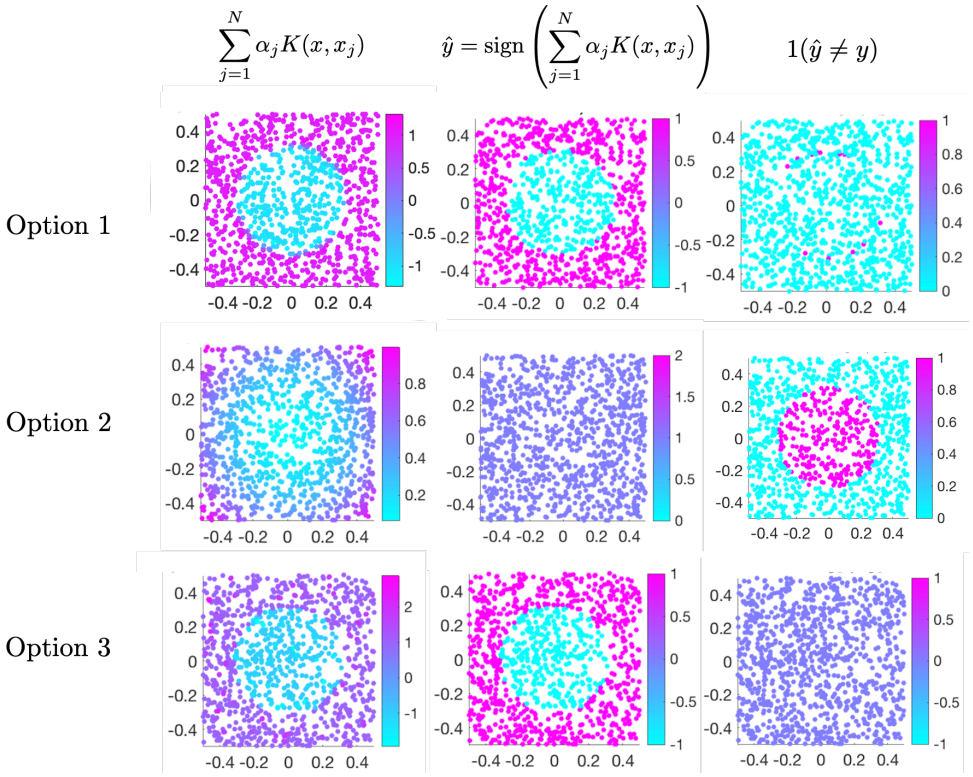

Figure 5: Kernel classification results with three different options

SOLUTION: σ controls the smoothness of the decision boundary by controlling the "width" of the kernels. Wide kernels (large σ) result in smooth decision boundaries. Narrow kernels (small σ) result in more complex decision boundaries, making a fewer mistakes on train set. Thus, a) corresponds to Option 2, b) corresponds to Option 1, and c) corresponds to Option 3.

5. **Kernel Classification.** When using a Gaussian kernel, is there a downside to choosing a very small value for σ?

SOLUTION: Overfitting is very likely. Each of the kernels tends to contribute to a single measurement ($\boldsymbol{K}$ is nearly diagonal), which results in perfect classification, but would also result in very poor performance generalizing to data not used to train the classifier.

6. Kernel SVM. You use a kernel-based support vector machine for binary classification with labels $d^i = \{+1, -1\}$. Given training features and labels $(\boldsymbol{x}^i, d^i), i = 1, 2, \ldots, N$ you use a kernal $K(\boldsymbol{u}, \boldsymbol{v})$ and design the classifier weights $\boldsymbol{\alpha}$ as

$$\widehat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \left(1 - d^i \sum_{j=1}^{N} \alpha_j K(\boldsymbol{x}^i, \boldsymbol{x}^j) \right)_+ + \lambda \sum_{i=1}^{N} \sum_{j=1}^{n} \alpha_i \alpha_j K(\boldsymbol{x}^i, \boldsymbol{x}^j)$$

a) Assume the optimization problem has been solved to obtain the weights $\boldsymbol{\alpha}$. Express the classification procedure for a measured feature $\boldsymbol{x}$.

b) Suppose $N = 1000$ and $\alpha_i = 0, i = 1, 2, \ldots, 99, 102, 103, \ldots, 1000$. Identify the support vectors and write the classification procedure in terms of the support vectors.

SOLUTION:

a) Kernel regression gives $\widehat{d(\boldsymbol{x})} = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}^i)$. Therefore, the classification procedure is

$$\text{sign} \left\{ \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}^i) \right\}$$

b) The only non zero α_i are α_{100} and α_{101}. Therefore, $\boldsymbol{x}^{100}$ and $\boldsymbol{x}^{101}$ are the support vectors. This means that the classification procedure is

$$\text{sign} \left\{ \alpha_{100} K(\boldsymbol{x}, \boldsymbol{x}^{100}) + \alpha_{101} K(\boldsymbol{x}, \boldsymbol{x}^{101}) \right\}$$