1. Which of the following are true about multi-layer neural networks? **Circle all true statements.**

    A. In general, they consist of only compositions of linear functions.

    B. Multi-layer neural networks with non-linear activation functions can correctly represent the XOR function.

    C. The decision boundary of a multi-layer neural network in a binary classification problem is always a straight line.

    D. In *binary* classification problems, we must use the softmax activation function to turn network outputs into probabilities.

    E. None of these are true about multi-layer neural networks.

    Solution: B is the only correct response. A is incorrect because neural networks most often also use non-linear activation functions. C is incorrect because the decision boundary may be more complex. D is incorrect because we can use the sigmoid activation for binary classification.

2. You have trained a neural network and found that is has high test error when evaluated on data points that it did not see during training. However the training error was very low. You plan to re-train the network. What changes should you make to lower test error? **Circle all possible changes.**

    A. Add L2 regularization or weight decay to training to force small weights.

    B. Decrease the number of hidden layers.

    C. Increase the number of hidden layers.

    D. If using sigmoid activation, change to relu.

    E. None of the above will help.

    Solution: A and B. The problem is overfitting (low train error; high test error) and can be mitigated by lowering the complexity of the neural network. Both A and B would accomplish that. C is incorrect because adding hidden layers increases model complexity which makes overfitting more likely. D is incorrect because changing the activation will not necessarily affect generalization.

3. What is the number of trainable parameters in a multi-layer neural network for 10-class classification with 2 hidden layers of sizes 10 hidden units and 20 hidden units respectively if the number of inputs is 3?

    A. 430

    B. 470

C. 512

D. 600

E. None of these.

Solution: B. First count the weights: 3 x 10 + 10 x 20 + 20 x10 = 430. Then count the biases: 10 + 20 + 10 = 40. The total is 470.

4. How many trainable parameters are there in a convolutional layer with 2 input channels of size 100 x 100, 20 3x3 kernels, stride 10, and no padding?

   A. 560

   B. 380

   C. 10

   D. 360

   E. None of these.

Solution: 380 = 360 weights + 20 biases. One bias for each kernel. Each kernel has 18 weights and there are 20 kernels.

5. Why are convolutional neural networks useful for image inputs? **Circle all correct statements.**

   A. They build in translation invariance.

   B. They build in locality for feature detection.

   C. They can learn any possible function of inputs to outputs.

   D. They have fewer trainable parameters than feed forward networks.

   E. Convolutional neural networks are not useful for image inputs.

Solution: A, B, and D. C is incorrect because this reason would not make CNNs more well-suited for image data.

6. What statements about linear models and neural networks are TRUE? **Circle all TRUE statements.**

   A. Linear models are for regression and neural networks are for classification.

   B. Neural networks involve non-linear computation and linear models do not.

   C. A logistic regression model is identical to a neural network with no hidden layers and sigmoid activation on the output.

D. Linear models can represent linear functions and multi-layer neural networks with non-linear activations cannot.

E. None of these are TRUE.

Solution: B and C. A is not true because you can use either for classification and regression. D is not true because multi-layer neural networks can still represent linear functions.

7. Consider a simple neural network (two inputs and one hidden layer of size 2) for regression. Let the network's output be $\hat{y} = \mathbf{w}^\top \mathbf{h} + \mathbf{b_2}$ where $\mathbf{w}$ is a weight vector, $\mathbf{h}$ is the hidden activations, and $\mathbf{b_2}$ is the bias. The hidden activations are computed as $\mathbf{h} = \texttt{relu}(\mathbf{W}_1^\top \mathbf{x} + \mathbf{b}_1)$ where $\mathbf{W}_1$ is the weight matrix of the first layer and $\mathbf{b}_1$ is the first layer bias.

   (a) Suppose our training data has only one data point, $(\mathbf{x}, y)$. We will train to minimize the mean squared error loss function: $L = (\hat{y} - y)^2$. What is the partial derivative of L with respect to $\hat{y}$?

   (b) What is the partial derivative of L with respect to b_2?

   (c) The activation $\mathbf{h}$ is the vector $[h_1, h_2]$. Use $w_{2,1}$ and $w_{2,2}$ to represent the components of the second layer weight vector, i.e., $\mathbf{w} = [w_{2,1}, w_{2,2},]$. What is the partial derivative of L w.r.t. h_1?

   (d) Let the weights in the first layer be given as $\mathbf{W_1} = [[w_{1,1}, w_{1,2}], [w_{2,1}, w_{2,2}]]$. Use x_1 and x_2 to represent the components of $\mathbf{x}$. What is the partial derivative of L w.r.t. $w_{1,1}$?

Solution: (a). $\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y) \cdot 1$

(b) $\hat{y} = w_1 h_1 + w_2 h_2 + b_2$. $\frac{\partial \hat{y}}{\partial b_2} = 1$. $\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial b_2} = 2(\hat{y} - y)$

(c) $\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h_1}$. $\frac{\partial \hat{y}}{\partial h_1} = w_{2,1}$. So $\frac{\partial L}{\partial h_1} = 2(\hat{y} - y) \cdot w_{2,1}$

(d) $\frac{\partial L}{\partial w_{1,1}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h_1} \frac{\partial h_1}{\partial w_{1,1}}$

$\frac{\partial h_1}{\partial w_{1,1}} = x_1$.

$\frac{\partial L}{\partial w_{1,1}} = 2(\hat{y} - y) \cdot w_{2,1} \cdot x_1$