1. MSE and RMSE are relatively the same when determining performance of models. However, it is more appropriate to use RMSE due to being having the same units as the response variable. RMSE and MSE are great for when attempting to penalize large errors. This is due to the nature of squaring the difference of the observed and predicted. Both MSE and RMSE are great in penalizing large differences. A metric such as MAE would not be as penalizing from outliers compared to RMSE/MSE.
2. A : As one adds more features to the dataset, this can result in overfitting which is not ideal for any model. As such, it is important to increase the dataset size so that the model has more information to determine the importance of each feature.
3. C: Insufficient training data size. Like problem 2, if the training loss is decreasing but the validation loss is constant or even increasing, that is a sign of overfitting. In cases where overfitting exists, it is best to increase the training data size for more information.
4. a) True, Perceptron is a linear classifier. Separates data with a hyperplane.

b) True, Perceptron will never converge on non-linearly separable data.

1. a) True, Logistic Regression is a linear classifier.

b) False, Logistic Regression does not always have a unique solution.

1. E: Both stochastic and batch gradient descent will eventually converge to the global optimum.
2. y = (7/6)\*x + e. To minimize e, use least squares function applying sum of residual squared: . To find minima, take the derivative with respect to Beta and set it equal to 0. Differentiating gives . Isolating B gives Diagram, schematic

   Description automatically generated.
3. Posted to GitHub. Coding Perceptron and Logistic Regression assignment is also found in same Github Link in the “code\_masters” folder: <https://github.com/DanielKim512/Intro2DL.git>
4. No, it is not a good idea to initialize parameters with zeros. It is important to produce different weights for the model to learn differently. Initializing as zero would just produce same results repeatedly with no updates.
5. When y = 1, it will give us the cost function of the 1 class which is -log(f(x)). If y=1 and f(x) = 1, the loss is 0, meaning that the prediction is 100% accurate. This cost function in general would not necessarily work in actual implementations because f(x) cannot be 0 as when y = 1 because log(0) is undefined (or y= 0 and f(x) = 1). Therefore in practice, it is ideal to offset to prevent f(x) from hitting 0.
6. First example: (0,-1,1) ; Second example: (-1.4,-0.7,-1) ; Third example: None, because the boundary is not linear. Perceptron is a linear classifier.
7. dE/da = (ca\*exp(-cx))/(1+exp(-ecax))^2 ; dE/db = (cb\*exp(-cy))/(1+exp(-cby))^2 ; dE/dc = ((ax+by)exp(-cax-cby))/(1+exp(-cax-cby))^2 ; dE/dd = (zexp(-dz))/(1+exp(-dz))^2