Neural Networks

1) Here are a few types of gradient descent updates:

Stochastic Gradient Descent: Uses only a single training example to calculate the gradient and update parameters.

Batch Gradient Descent: *Calculate the gradients for the whole dataset and perform just one update at each iteration.*

Mini-batch Gradient Descent: *Mini-batch gradient is a variation of stochastic gradient descent where instead of a single training example, a mini-batch of samples is used. It's one of the most popular optimization algorithms.*

What are the pros and cons of each one and when should you use them? Walk us through your intuition and give us examples of when you are going to use what update.

2) Our internal AI team has been training a model for a very long time and our biggest problem is that our model does not learn anything. Our cost function is not going down, and our gradients in a layer seem to be the same. We have been debugging our gradient update function but it seems like it is correct. Do you have any tips to help us solve this problem? Walk us through your thought process and show us on the board what is going on.

3) You are asked to build a classification model about meteorites impact with Earth (important project for human civilization). After preliminary analysis, you get 99% accuracy. Should you be happy? Why not? What can you do about it?

4) You are working on a classification problem. For validation purposes, you've randomly sampled the training data set into train and validation. You are confident that your model will work incredibly well on unseen data since your validation accuracy is high. However, you get shocked after getting poor test accuracy. What went wrong?

5) Why should features be normalized when applying regularization to logistic regression and neural nets? Any further reasons to do it for neural nets even if we are not using regularization?

6) Let's explore saliency, a measure of how important a feature is for classification. We define the saliency of the ith input feature for a given example (x, y) to be the absolute value of the partial derivative of the log likelihood of the sample prediction, with respect to that input feature $\frac{\partial LL}{\partial x_i}$. In the images below, we show both input images and the corresponding saliency of the input features (in this case, input features are pixels):

First consider a trained logistic regression classifier with weights w. Like the logistic regression classifier that you wrote in your homework it predicts binary class labels. In this question we allow the values of **x** to be real numbers, which doesn't change the algorithm (neither training nor testing).
(You can do the same problem with a neural net and back prop but it gets a bit more complex)

a) What is the Log Likelihood of a single training example (x, y) for a logistic regression classifier?
b) Calculate is the saliency of a single feature (x_i) in a training example (x, y).
c) Show that the ratio of saliency for features i and j is the ratio of the absolute value of their weights $\frac{|w_i|}{|w_j|}$.