CS4364/6364 Intro to Machine Learning
Dr. Dobolyi

Exam 2

Please **DO NOT START** the exam until instructed, out of fairness to all students. 60 minutes.

Score: _____ / 31 pts

Name: ____________________________

1. **Very short** answers (2 pts each == total):

   a. Explain how you would calculate the Mean Squared Error (MSE) loss for a linear regression? [You can give a formula, or explain the logic in English]

   b. Explain why we generally avoid calculating the zero-one (0-1) loss.

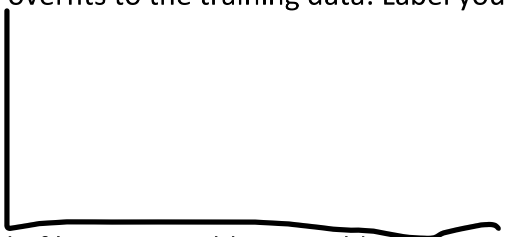
   c. What is an activation function in a perceptron/NN (not just an example, but explain how it works please)?

   d. Why do gradients vanish during back-propagation for the lowest layers of the model (near the input layer)?

   e. What is a feature map in a CNN (explain what/why it exists, not just giving an example please)?

   f. What is the biggest benefit from transfer learning (from a model trained on ImageNet, for example)?

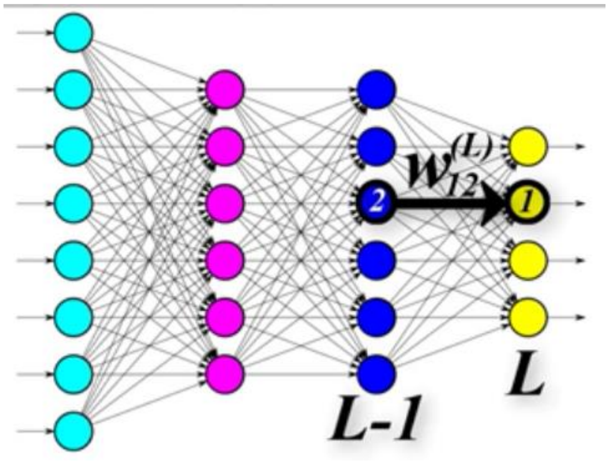
   g. Complete the graph below of training and validation loss where the model is learning, but overfits to the training data. Label your two axes and two lines.



   h. For what kind of learning problem would you use a Markov Model (as opposed to the classification/regression we've seen this semester)?

2. **Short** answers (4 pts each == total):

   a. When updating weight in a NN during back-propagation, if it is **weight$_{12}$** we want to update in the graph below, what other inputs/outputs/weights/cost/loss do we need to calculate as partial derivatives to arrive at the {derivative of the overall loss/cost with respect to **weight$_{12}$**}, to apply the chain rule? You don't have to provide the formulas, you just have to explain what elements go into it (so your answer will be a list of items, no math necessary). As a reminder, **weight$_{12}$** is at the second-to-last layer of this NN: the last layer **L** makes predictions for one of each of the four possible classes.



   b. Explain what each of the four image transformations are doing during training a model, and why it's useful to have that transformation for training:

```
train_transforms = [
  transforms.RandomHorizontalFlip(),
  transforms.Resize(size=[224,224]),
  transforms.ToTensor(),
  transforms.Normalize([0.1306],[0.3081])
]
```

Multiple choice answers (1 pts each == 28 total):

3. Imagine we have a sigmoid (logistic regression) function that predicts values between 0 and 1. What is the range of the Negative Log Loss Function, in terms of values this loss function can take on for the actual prediction from that sigmoid output?
   a. 0 to 1 inclusive
   b. 0 to infinity
   c. 0 to negative infinity
   d. Negative infinity to positive infinity
4. In the question above, imagine the ground truth target was 1, and the model also predicted 1. What would the loss be for this sample?
   a. 1
   b. 0
   c. Positive infinity
   d. Negative infinity
5. Why would we choose to use regularization when calculating our weights?
   a. To pressure some weights to be zero
   b. To keep weights small
   c. To improve generalization
   d. A and B
   e. A, B, and C
   f. None of the above
6. A linear regression that has a redundant feature like height_in_cm and height_in_inches will be unable to learn proper weights for these models on the training dataset, so you must remove one of the two in order to train the model successfully.
   a. True
   b. False
7. When doing binary classification on ground truths of -1 and 1, we can calculate the margin $yx^T\theta$. What is true about this margin (assume it is never 0)?
   a. It will be positive for a correct prediction, and negative for an incorrect prediction.
   b. It is desirable, in terms of training a good model, to have larger margin values.
   c. We want to minimize the margin values.
   d. A and B
   e. A and C
   f. A, B and C
   g. None of the above.
8. Gradient descent will always result in learning the weights to yield the best/minimal/optimal loss for a specific model type and loss function.
   a. True
   b. False
9. It is often desirable to have “dead nodes” in a NN (a sparse representation).
   a. True
   b. False