ECE 239AS, Winter 2019 **Midterm**

Department of Electrical Engineering Prof. J.C. Kao
UCLA ECE TAs W. Chuang & M. Kleinman & K. Liang & A. Wickstrom

UCLA True Bruin academic integrity principles apply.
4 cheat sheets allowed
6:00 - 7:50pm.
Wednesday, 20 Feb 2019.

State your assumptions and reasoning.
No credit without reasoning.
Show all work on these pages.

Name: _____

Signature: _____

ID#: _____

Problem 1	_____	/ 20
Problem 2	_____	/ 20
Problem 3	_____	/ 20
Problem 4	_____	/ 20
Problem 5	_____	/ 20
Total	_____	/ 100

1. **Training, validation, and testing.**

   (a) (5 points) Briefly explain the purpose of the training, validation, and testing set. Comment on how many times each set should be used when doing k-fold cross validation.

   (b) (10 points) Consider a dataset where a signal evolves through time. The data is sampled at an extremely high rate, meaning that adjacent time points are highly correlated. Consider also that this dataset has few trials. Your colleague comes up with an idea to increase the number of effective trials: he decides to subsample the data with no overlapping time points (so, for instance time points $1, 3, 5, 7, 9...$ could represent one effective trial and time points $2, 4, 6, 8, ...$ another). He tells you he now has twice the number of effective trials.

      i. (3 points) Why might it be beneficial to subsample the data to increase the number of effective trials?

ii. (7 points) After subsampling the data, your colleague randomly split his trials (of which he now has double) into a training and validation set. He tells you that his validation accuracy is much higher than anything you have achieved. Suggest why this might be the case.

(c) (5 points) Another colleague splits the data into 80% training, 10% validation, and 10% test. This colleague did not subsample the data. He has experimented with various architectures, each time training on the training set, validating on the validation set, and then testing on the testing set. Based on his testing accuracy, he would see which hyperparameters worked well, and make modifications to his model, reiterating the above process. Will the final test accuracy reported be an accurate proxy for how well his model will generalize to new data? Briefly explain.

2. **Backpropagation.** Consider a 3 layer neural network (NN), with $\mathbf{x} \in \mathbb{R}^n$ as input and $\mathbf{y} \in \mathbb{R}^m$ as the target value. The NN is constructed as the following:

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}||h_3(h_2(h_1(\mathbf{x}))) - \mathbf{y}||^2 \\ h_1(\mathbf{x}) &= \text{PReLU}(\mathbf{W}_1\mathbf{x}) \\ h_2(\mathbf{x}) &= \text{ELU}(\mathbf{W}_2\mathbf{x}) \\ h_3(\mathbf{x}) &= \mathbf{W}_3\mathbf{x} \end{aligned}$$

where

- $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{n \times n}$ and $\mathbf{W}_3 \in \mathbb{R}^{m \times n}$
- For the PReLU unit, $f(x) = \max(\alpha x, x)$.
- For the ELU unit, $f(x) = \max(\alpha e^x - 1, x)$.

(a) (5 points) Draw the computational graph for this neural network.

(b) (15 points) Calculate $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_1}, \frac{\partial \mathcal{L}}{\partial \mathbf{W}_2}, \frac{\partial \mathcal{L}}{\partial \mathbf{W}_3}$ using backpropagation. You may define intermediate variables and write the gradients in terms of these intermediate variables.

Hint: For a ReLU function $f(x) = \max(0, x)$, we compute the gradient for $x < 0$ and $x > 0$ separately. For instance, $\frac{\partial f(x)}{\partial x} = \mathbb{1}\{x > 0\} \cdot 1 + \mathbb{1}\{x < 0\} \cdot 0$. Compute the gradient for PReLU and ELU with the same method. (Additional space provided on next page.)

(Additional space for question 2b.)

3. **Regularization.**

   (a) (5 points) You are tasked with training a feedforward neural network. Your boss asks you to shrink the effective size of the model by using regularization. How can you complete the task?

   (b) (5 points) Why can $L2$ regularization be referred to as "weight decay"?

(c) (5 points) How does batch normalization help a neural network to be more robust to initialization?

(d) (5 points) How does dropout act as a regularizer?

4. **Optimization.**

   (a) (12 points) Suppose, oddly, that an optimizer takes several gradient steps and arrives back at a same parameter setting, $\mathbf{W}$, that is *exactly* the same as in a prior step. This setting of $\mathbf{W}$ does not correspond to a local optima of the loss function. Imagine you are optimizing this neural network using a batch algorithm, i.e., using the entire training set to calculate a gradient. You are also using a fixed learning rate.

      i. (4 points) Consider that the optimizer was a naive gradient descent optimizer (with no momentum or adaptive gradients). Is the gradient step that you will take the second time you are at this parameter setting, $\mathbf{W}$, (circle one)

         - larger (in magnitude)
         - smaller (in magnitude)
         - the exact same, or
         - can't be determined (i.e. not enough information)

         than the gradient step the first time the optimizer was at the parameter setting, $\mathbf{W}$? Justify your answer.

ii. (4 points) Consider that the optimizer was Adagrad. Is the gradient step that you will take the second time you are at this parameter setting, $\mathbf{W}$, (circle one)

- larger (in magnitude)
- smaller (in magnitude)
- the exact same, or
- can't be determined (i.e. not enough information)

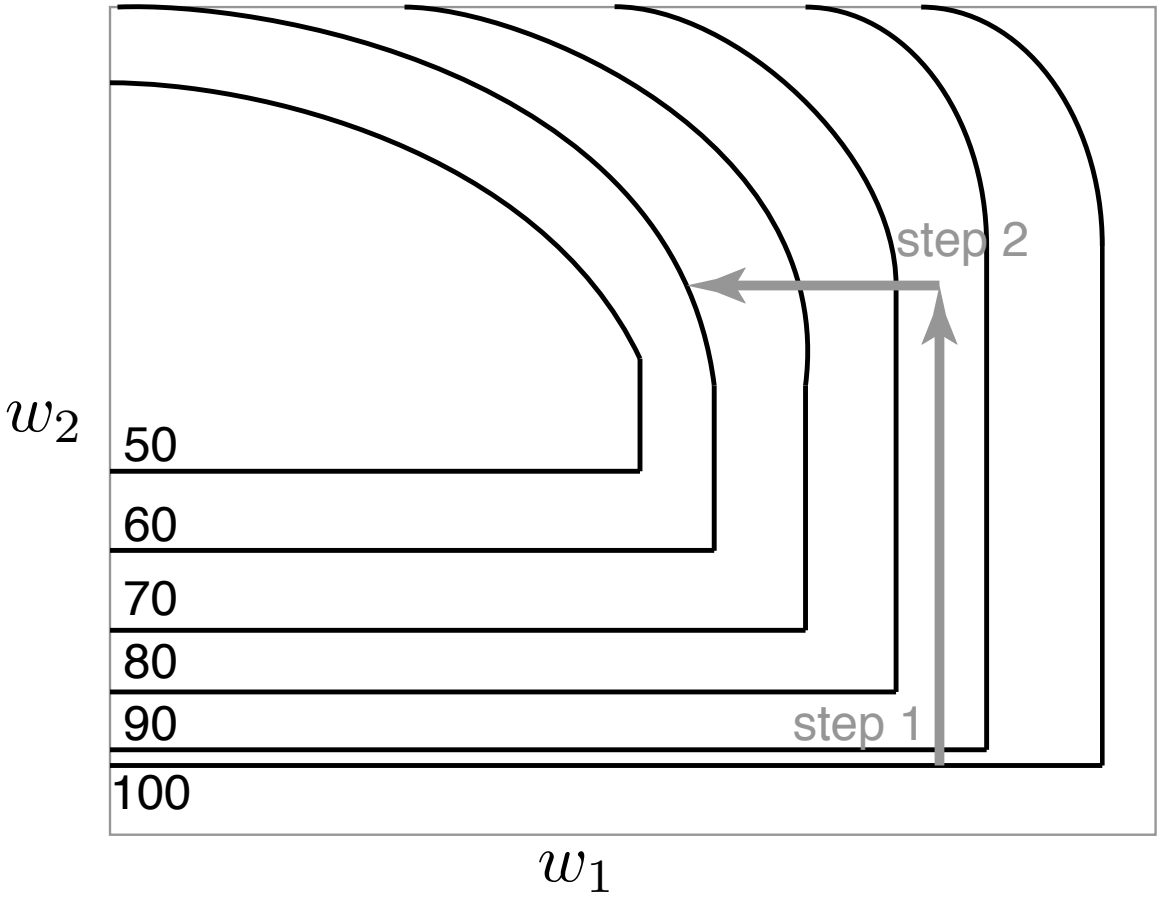
than the gradient step the first time the optimizer was at the parameter setting, $\mathbf{W}$? Justify your answer.

iii. (4 points) Consider that the optimizer was RMSprop. Is the gradient step that you will take the second time you are at this parameter setting, $\mathbf{W}$, (circle one)

- larger (in magnitude)
- smaller (in magnitude)
- the exact same, or
- can't be determined (i.e. not enough information)

than the gradient step the first time the optimizer was at the parameter setting, $\mathbf{W}$? Justify your answer.

(b) (8 points) The following figure is a contour plot where the contour lines denote the value of the loss as a function of two weight variables (corresponding to the x and y-axis). Imagine we have taken two gradient steps given by the gray arrows. Sketch on the same plot the weight update steps taken by SGD, SGD+momentum, and Adagrad after step 2. Do not perform any calculations; the sketch should be arrived at through intuition. Give at most a two sentence explanation for each arrow you've drawn.



5. **Convolutional neural networks.** Consider a convolutional layer C followed by a max pooling layer P. The input to layer C is $120 \times 120 \times 50$. Layer C has 20 filters, each of which is of size 4×4. The convolution padding is 1 and the stride is 2. Layer P performs max pooling over each of the layer C's output feature maps, over a 3×3 receptive field, and stride 1. Given $x_1, x_2, ..., x_n$ all scalars, we assume:

   - A scalar multiplication $x_i \cdot x_j$ accounts for one FLOP;
   - A scalar addition $x_i + x_j$ accounts for one FLOP;
   - A max operation max$(x_1, x_2, ..., x_n)$ accounts for $n - 1$ FLOPs.

   You do not need to calculate the products you write out (e.g., answers maybe left in terms like "$(x \cdot y \cdot w) \cdot z$").

   (a) (5 points) What is the total number of trainable parameters? Please account for the bias term.

   (b) (4 points) What is the size of layer C's output feature map?

(c) (4 points) What is the size of layer P's output feature map?

(d) (7 points) How many FLOPs are there in layers C and P during one forward pass? Please include the bias term when calculating the FLOPs.