ECE 239AS, Winter 2018 **Midterm**

Department of Electrical Engineering Prof. J.C. Kao

University of California, Los Angeles TAs T. Xing and C. Zheng

UCLA True Bruin academic integrity principles apply.
Open: book, notes, handout, computer.
Closed: internet, except for downloading / viewing from ECE239AS website / CCLE.
6:00 - 7:50pm.
Wednesday, 21 Feb 2017.

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

Name: ____________________

Signature: ____________________

ID#: ________________

Problem 1	______	/	20
Problem 2	______	/	15
Problem 3	______	/	15
Problem 4	______	/	15
Problem 5	______	/	15
Problem 6	______	/	20
Total	______	/	100

1. (X points) **Short answer on machine learning basics.**

   (a) (X points) **SVM basics.** In writing the SVM cost function, the hinge loss has a margin of 1. Notice that we set the margin always to 1, as opposed to having it be a hyperparameter Δ. Why can we do this? Justify your answer (*at most* three sentences).

   (b) (X points) **Comparing a kNN and linear classifier.** Compare bias and variance of a linear classifier to kNN. Keep your answer to *at most* six sentences.

   (c) (X points) **Bias and variance.** Let X be a random variable representing the current time (not distinguishing between a.m and p.m). Suppose X is sampled from a uniform distribution $X \sim U[0, 12]$. Let Z be a random variable representing the difference between the estimated time and the current time, wrapped to $+/-6$ hours, so that $Z \sim U[-6, 6]$. If $z = 0$, that means the estimated time is the current time. Consider two estimators of the time differences.

      i. $\hat{z}_1$ is the estimator for a stopped clock (which always shows the same time). Note that it gives the correct estimate twice a day.

      ii. $\hat{z}_2$ is the estimator for a clock which works with perfect precision but is in the wrong timezone, so it is always one hour late and never gives the correct estimate.

      Calculate the bias and the variance of both estimators $\hat{z}_1$ and $\hat{z}_2$.

   (d) (X points) **Cheating a competition.** You are running a machine learning competition where competitors submit a classifier that makes a prediction. You have one private test dataset that you use to evaluate all submitted classifiers. You found out that one of your colleagues allowed competitors to evaluate their classifier on this test dataset an unlimited amount of times (i.e., whenever they wanted to check). How could a competitor take advantage of this to do better in the competition? Keep your answer to *at most* three sentences.

2. (X points) **Maximum likelihood estimation.** We observe n i.i.d random samples $x_1, x_2, ..., x_n$ that we model as arising from a Poisson distribution. We want to use maximum likelihood estimation method to estimate the parameter, λ, of the Poisson distribution. The probability mass function of a Poisson distribution is given by:

$$\Pr(x = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

   (a) (X points) Write the log-likelihood of observing the data under the model assumption.

   (b) (X points) Calculate the derivative of the log-likelihood w.r.t. the parameter, λ.

   (c) (X points) Calculate the maximum likelihood estimator of λ.

3. (X points). **Backpropagation**

   Let $\mathbf{h} = \text{ReLU}(\mathbf{W}\mathbf{x} + \mathbf{b})$ be a typical layer of a FC network, with $\mathbf{x} \in \mathbb{R}^n$ as input and $\mathbf{h} \in \mathbb{R}^m$ as output. The mean square logarithmic error (MSLE) loss function is a variant of mean square error. The MSLE, defined for *one example (the ith example)* and denoted by $\mathcal{L}_i$ is given by:

$$\mathcal{L}_i = (\log |\mathbf{h}\mathbf{h}^T + \mathbf{I}| - \log |\mathbf{y}\mathbf{y}^T + \mathbf{I}|)^2$$

   where $\mathbf{y}$ is the target value. Note that $|\cdot|$ here denotes the determinant of a matrix.

   (a) (5 points) Write the dimension of following variables $\mathcal{L}_i, \mathbf{W}, \mathbf{b}, \mathbf{I}, \mathbf{y}$. For example, $\mathbf{x} \in \mathbb{R}^n$.

(b) (10 points) Calculate $\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}}$ and $\frac{\partial \mathcal{L}_i}{\partial \mathbf{x}}$. You can use the following formula without proof.

$$\begin{aligned}
\frac{\partial \log |\mathbf{X}|}{\partial \mathbf{X}} &= (\mathbf{X}^T)^{-1} \\
\frac{\partial \mathcal{L}_i}{\partial \mathbf{X}} &= \mathbf{W}^T \frac{\partial \mathcal{L}_i}{\partial (\mathbf{W}\mathbf{X})} \\
\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}} &= \frac{\partial \mathcal{L}_i}{\partial (\mathbf{W}\mathbf{x})} \mathbf{x}^T \\
\frac{\partial \mathcal{L}_i}{\partial \mathbf{X}} &= 2 \frac{\partial \mathcal{L}_i}{\partial (\mathbf{X}\mathbf{X}^T)} \mathbf{X}
\end{aligned}$$

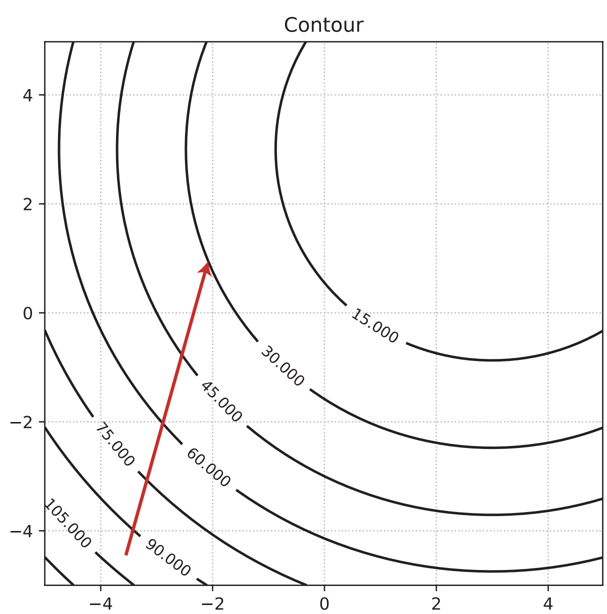
4. (X points) Optimization (momentum, RMSprop, adam).

(a) (X points) **True or false .** What parameters does optimizer SGD+momentum maintain that SGD does not? Answer in one or two sentences.

(b) (X points) **Momentum 2.** Imagine separate optimizations, one with SGD and the other SGD with momentum. In both optimizations, the parameters are approaching the same local optima with the same learning rate. Which optimizer is more likely to stay at this local optima, and why? Justify your answer (*at most* three sentences).

(c) (X points) **Adagrad and RMSProp** *Adagrad* and *RMSprop* modify the learning rate in particular directions based off of historical gradients. What weakness of *Adagrad* does *RMSprop* address? Justify your answer (*at most* three sentences).

(d) (X points) **Gradient steps** The following figure is the 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 one gradient step in the direction of the red



line. Sketch on the same plot the steps taken by SGD, SGD+momentum, and Adagrad. 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. (X points) **Short answer on training neural networks.**

   (a) (X points) **Alien brain.** Imagine you are designing a neural network inspired from an alien brain. In this alien brain, experiments show that their neurons are organized like a feedforward network.

      i. (X points) Curiously, after several experiments, the researchers find that neurons in the first and last layers capture similar features, indicating that they have similar weights. They want you to build a neural network based off of the principle that the first and last layers have similar weights. Assume that the number of parameters in the first and last layer are the same. How would you train a neural network that achieves this? Justify your answer (*at most* three sentences).

      ii. (X points) Your colleague returns and says "these alien brains make no sense! Now it turns out that activations in the 10th layer of the network are almost all entirely zero." You want to incorporate this knowledge into your neural network design. How would you train a neural network that achieves this? Justify your answer (*at most* three sentences).

      iii. (X points) Now your colleague returns and says: "ugh, I'm about to give up. In the 3rd layer, all the activations inexplicably tend to the value 1. I know it makes no sense, but can you incorporate this knowledge into your neural network design?" How would you train a neural network that achieves this? Justify your answer (*at most* three sentences).

   (b) (X points) **Activation functions.** For each of these activation functions, state whether it (1) saturates (and if so, for what values does it saturate), (2) has the zig-zagging gradients problem (i.e., in a network using only this activation function, are gradient steps zig-zagging?), (3) is differentiable everywhere. **Briefly** justify each solution you give.

      i. Hyperbolic tangent: $\tanh(x)$.
      ii. Leaky ReLU: $\max(0.01x, x)$.
      iii. ELU: x if $x > 0$ and $\alpha(\exp(x) - 1)$ if $x \leq 0$.
      iv. Exponential: $\exp(x)$.
      v. Shifted logarithm: $f(x) = 0$ if $x \geq 0$, $f(x) = \log(1 - x)$ if $x < 0$.

6. **Short answer on convolutional neural networks.**

   (a) (X points) **AlexNet.** AlexNet is an architecture that we've mentioned several times in class. In this question, you will calculate attributes of some AlexNet layers.

      i. (X points) The 3rd, 4th, and 5th convolutional layers in AlexNet have the following properties:

         - 3rd Conv Layer: 384 filters that are 3x3 applied at stride 1, pad 1.
         - 4th Conv Layer: 384 filters that are 3x3 applied at stride 1, pad 1.
         - 5th Conv Layer: 256 filters that are 3x3 applied at stride 1, pad 1.

         The input to the 3rd conv layer is a tensor that is $27 \times 27 \times 96$. After the 5th conv layer, the output is a tensor that is $27 \times 27 \times 256$. Calculate the number of parameters in the 3rd, 4th, and 5th convolutional layers, not including biases. (To be clear, do not include biases in your calculation, but only the convolutional filter weights.)

ii. (X points) How many input neurons is each output neuron in the 3rd convolutional layer connected to? Do not include biases in this calculation.

iii. (X points) After the 5th Conv Layer, there are 3×3 max pooling filters applied at stride 2. The output of this operation is $6 \times 6 \times 256$. Calculate the number of trainable parameters in the max pooling layer. Do not include biases in this calculation.

iv. (X points) After this pooling layer, the data representation is a $6 \times 6 \times 256$ tensor. This data representation becomes the input to a fully connected layer with 4096 units. Calculate the number of parameters in this fully connected layer. Do not include biases in this calculation.

v. (X points) Consider the memory required to store the output of each layer. Assume all numbers are represented as singles (i.e., with 4 bytes). How much memory is required to store the output of the 5th Conv Layer, the pooling layer, and the fully connected layer?

(b) (X points) **Receptive fields.** We define the receptive field as the spatial extent of the inputs (to the first layer) seen by a given output neuron (in the last layer). For example, if we use one convolutional layer with a 3×3 filter applied at stride 1, then the effective receptive field is 3×3, since each output neuron is connected to a 3×3 patch of the inputs. We would say that the receptive field is 3 (assume that the height and width of the filters are always matched). Assume the input is an $m \times m \times d$ tensor. Consider the following questions:

i. What is the receptive field of a stack of three 3×3 convolutional filters applied at stride 2? Assume the stride is lawful.

ii. What is the receptive field of a stack of three 1×1 convolutional filters applied at stride 1?

iii. What is the receptive field of a fully connected layer?

(c) (X points) **Runtime for a FC net vs a Conv net.** Goodfellow, on p. 326, argues that if there are m inputs and n outputs, a fully connected layer will have runtime $\mathcal{O}(m \times n)$, which is the cost of the matrix-vector multiply. However, if there is a conv net with filters having k parameters, the runtime will be $\mathcal{O}(k \times n)$, suggesting the runtime of a conv net is faster. Let's do an example to see if this is really the case.

i. Imagine that a FC network has an input that is $m \times m \times d$. Imagine that the output has n neurons. How many multiplication operations need to be done to compute the activation of the n output neurons?

ii. Now, with the same input as in part (i), imagine that we design a convolutional neural network with d filters that are applied at stride 1 and pad 0. What is the size of each filter required so that there are n total output neurons?

iii. How many multiplication operations need to be done to compute the activation of these n output neurons in the convolutional neural network? Express the answer in terms of just m, d, and n.

iv. Under what conditions does the number of multiplication operations in a CNN exceed an FC net? Make intuitive sense of this result.