24-789 Deep learning for Engineers
Spring 2022
Midterm Exam
3/31 5:00PM - 4/02 5:00PM (EST)
Time Limit: 48 Hours

Name (Print): _____

Andrew ID: _____

This exam contains 6 pages (including this cover page) and 3 problems. Check to see if any pages are missing. Enter all requested information on the top of this page, and put your name on the top of every page, in case the pages become separated.

You are required to show your work on each problem on this exam. The following rules apply:

- **Organize your work**, in a reasonably neat and coherent way, in the space provided. Work scattered all over the page without a clear ordering will receive very little credit.
- **Mysterious or unsupported answers will not receive full credit**. A correct answer, unsupported by calculations, explanation, or algebraic work will receive no credit; an incorrect answer supported by substantially correct calculations and explanations might still receive partial credit.
- If you need more space, use the back of the pages; clearly indicate when you have done this.

Do not write in the table to the right.

Problem	Points	Score
1	50	
2	20	
3	30	
Total:	100	

1. (50 points) Short Answer Questions

   (a) (5 points) Why do modern deep learning models no longer use threshold function (like shown in Eq. 1) as activation function?

   $$f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \tag{1}$$

   (b) (5 points) Name 2 reasons why ReLU (Rectified Linear Unit) is better than Sigmoid and Tanh (hyperbolic tangent) activation functions in deep neural networks?

   (c) (5 points) Name 2 advantages of CNN (convolutional neural network) comparing to fully connected neural networks.

   (d) (5 points) Assume you are training a fully connected neural network for classification and there is no bug in your code. After the loss function converges, although the prediction accuracy is high on training set, the prediction accuracy on test set is low. List 2 things you would like to try to improve the performance. And explain why they should work?

   (e) (5 points) Assume you are training a fully connected neural network for classification and there is no bug in your code. This time, the loss function converges after only few epochs but the prediction accuracy on both training set and test set are low. List 2 things you would like to try to improve the performance. And explain why they should work?

   (f) (5 points) What is the most critical problem that LSTM (long-short term memory) solves in comparison to vanilla RNN (recurrent neural network) models? And how does it solve that?

   (g) (5 points) Assume you want to solve a 1D signal classification problem using deep learning (assume all the signals have the same length). What is the difference between using 1D CNN and LSTM?

   (h) (5 points) What is the difference between PCA (principle component analysis) and autoencoder? How will you design an autoencoder architecture that performs the same as PCA (specify number of layers, activation functions, etc.).

   (i) (5 points) Why KL divergence term is included in VAE (variational autoencoder)? What if we remove the KL divergence term in training a VAE model?

   (j) (5 points) Suppose you are training a GAN (generative adversarial network) and use 1 to represent real input (x from training set) and 0 for fake one (from generator). After training for thousands of epochs, GAN loss (given in Eq. 2, where z represent random noise from Gaussian, $\mathcal{D}$ represents discriminator and $\mathcal{G}$ represents generator) converges to 0. Does it guarantee your GAN model is well-trained and able to generate realistic samples? Explain your answers.

   $$\min_G \max_D \mathcal{L}_{GAN} = \log \mathcal{D}(x) + \log(1 - \mathcal{D}(\mathcal{G}(z))) \tag{2}$$

2. (20 points) Calculation

Please give your calculation steps and necessary descriptions. You will lose points if only results are given.

(a) (5 points) In one layer of a CNN model, you get an input 7×7 feature map. Then you are using a 3×3 filter, with padding size 2 and stride size 2. What is the size of the new feature map after this convolutional layer?

With the same input feature map, this time you are using a 5×5 filter with stride 1. What is the padding size to make size of output feature map the same as the input?

(b) (5 points) Fig. 1 shows the architecture of LeNet, where in each convolutional layer, $C@W \times W$ represents the size of the output feature maps (C for number of channels and W for height and width). Calculate the number of parameters in this CNN model. (hint: don't forget biases in both convolutional filters and fully connect layers, also you should only consider trainable parameters)

Size of filter and output feature map in each layer is given:
input: 1x32x32
conv1: f(6x5x5)@stride1 $\rightarrow$ 6x28x28
s2: pooling(2x2) $\rightarrow$ 6x14x14
conv3: f(16x5x5)@stride1 $\rightarrow$ 16x10x10
s4: pooling(2x2) $\rightarrow$ 16x5x5
conv5: f(120x5x5)@stride1 $\rightarrow$ 120x1x1
fc6: 120 $\rightarrow$ 84
fc7: 84 $\rightarrow$ 10

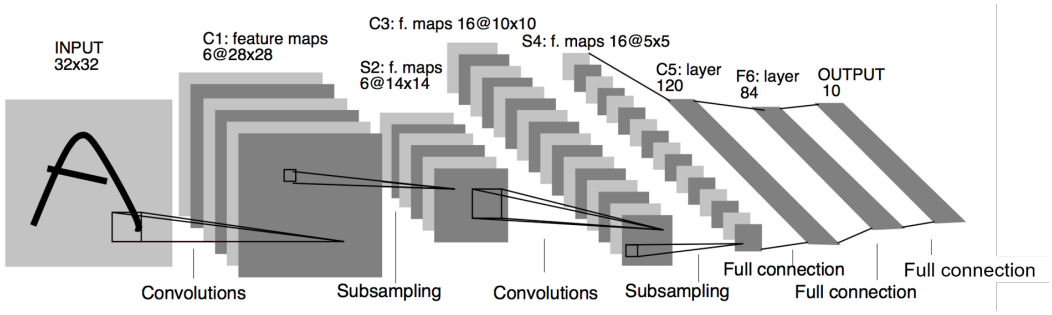

Figure 1: LeNet

(c) (5 points) Let $f(\cdot)$ be the objective function you would like to minimize. At time step t, you get $x_t = 3$, $f(x_t) = 6$ and $f'(x_t) = 1$. After one iteration of gradient descent with learning rate $\eta = 0.1$, give your solution to x_{t+1}.

Now assume you get $f'(x_{t+1}) = 0$, does it guarantee x_{t+1} is the global optimizer. If yes, explain why. If not, give a solution which help find the global optimizer.

(d) (5 points) Suppose in one layer of a CNN, you get a feature map and want to conduct convolution using the filter as given in Fig. 2. With stride 1 and padding 0, calculate the new feature map after convolution operation.

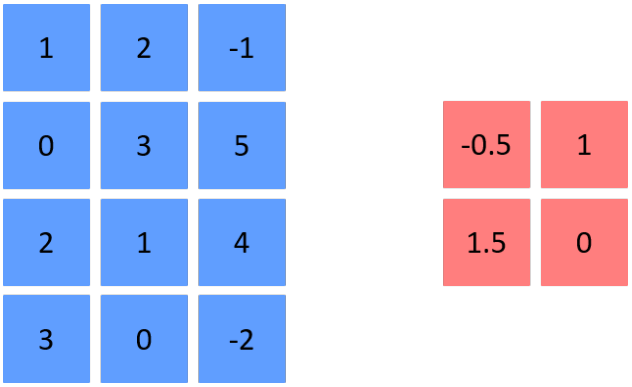

Figure 2: Left: feature map, right: convolutional filter

3. (30 points) Multiple choice questions

Please select ALL that apply! You don't need to give the explanation of your choices. To get full points of each question, you have to select all the correct option(s). You lose 1 point if correct option is missing and lose all 3 points if you select a wrong option.

(a) (3 points) In CNN training, when backpropagating through a max pooling layer, gradient at the input locations of the non maximum values is:
A) Identical to the derivative at the location of the maximum value
B) Identical to the input value at the location of the maximum value
C) Identical to the input value at the location of the non maximum value
D) 0

(b) (3 points) In CNN training, when backpropagating through a mean pooling layer, gradient at the output of a mean pool filter is
A) equally distributed over the input pool
B) assigned to the input location of the maximum value
C) 0
D) distributed over the inputs in proportion to their values

(c) (3 points) Consider applying a 5x5x5 convolution filter on an RGB image input of dimension (3,H,W). Which of the following statements are true?
A) The output is of dimension (5,H,W)
B) The output is of dimension (3,H,W)
C) The output is of dimension (5,5,5)
D) None of above

(d) (3 points) When using gradient descent to find the minimum of a function, which of the followings are correct:
A) Accurately solving for the location that has minimum gradient.
B) Using the Hessian of the function to solve for the location of the minimum
C) Starting at some initial location and iteratively moving in the opposite direction of the gradient, until find the minimum.
D) Starting at some initial location and iteratively moving in the direction of the gradient, until find the minimum.

(e) (3 points) Select all the statements that are correct:

A) The output of a neuron has a probabilistic interpretation when using sigmoid activations.

B) A fully-connected neural network with linear activation functions is equivalent to a single perceptron with linear activation.

C) For a multilayer fully connected neural network with Tanh as activation function at each layer. If you scale up all the weights (including biases) by 2, the output of the model keeps exactly the same.

D) Threshold activations cannot be used for backpropagation.

(f) (3 points) What are the benifits of using L2 regularization of weights in a network?

A) It ensures that neuron activations are sparse (i.e. only a few are non-zero), improving interpretability of their output

B) It prevents activation functions from becoming too steep, and restricting the network output from changing too steeply with input.

C) It prevents overfitting to training data

D) It guarantees to restrict the weights to lie within a unit hypersphere, preventing floating point overflow

(g) (3 points) What are the improvements of momentum learning rule comparing to vanilla gradient descent?

A) The magnitude of the changes to the parameters can increase without bound if the gradients don't change very much across iterations.

B) Momentum learning allows us to forget about the learning rate since the learning rule will automatically adjust the step size.

C) Momentum learning guarantees the network to find global minima as opposed to local minima.

D) Momentum learning smooths noisy gradients, reduces oscillations, and encourages updates in directions with smooth convergence behaviors.

(h) (3 points) In RNN, given that the following operation holds for the hidden unit $h(t) = f(wh(t-1)+cx(t))$, where f is a non-linear activation function and w and c are positive scalars. The output of the RNN is $h(T)$, where $T \to \infty$. Which of the following are true about $h(t)$ when $x(0)$ is negative, and all subsequent $x(t)$ values are 0, namely $x(t) = 0$, $t > 0$? (Assume $h(-1)$ is 0. In these statements the word "activation" refers to the function f)

A) Using Tanh activation, the output eventually saturates to 1.

B) Using ReLU activation, the output is zero.

C) Using Sigmoid, the output eventually saturates to 0.

D) Using ReLU activation, it is sensitive and the output can blow up

(i) (3 points) Assume you are training a VAE model, and after thousands of iterations the loss converges but all the reconstructed samples are almost the same even with different inputs. What can be solutions to the issue:

A) Increasing the weight of KL divergence term in the loss function and retraining the model from scratch

B) Decreasing the weight of KL divergence term in the loss function and retraining the model from scratch

C) Increasing learning rate and keeping training current model

D) None of above

(j) (3 points) Assume you are training a GAN model, and after thousands of iterations the loss converges but all the generated samples from random noise are the same. What can be solutions to the issue:

A) Use different learning rate for the discriminator and the generator

B) Using a uniform distribution instead of a Gaussian distribution to sample input noise

C) Asynchronously update the discriminator and the generator

D) None of above