2024 AI604 course
Deep learning for computer vision
Midterm Answers

1. Linear classifier

This question is about the properties of multi-class SVM loss vs. softmax loss. Fill in the appropriate answers in the table below.

	SVM loss	Softmax loss
What is the min/max possible loss?		
What happens to loss if the score of the correct class decreases slightly when it is much larger than other scores?		
What happens to loss if the score of the correct class decreases slightly when it is equal to other scores?		
At initialization W is small so all s ≈ 0. What is the loss Li, assuming C classes?		

Answer:

	SVM loss	Softmax loss
What is the min/max possible loss?	Min 0/max inifity	Min 0/max inifity
What happens to loss if the score of the correct class decreases slightly when it is much larger than other scores?	No change	Loss changes
What happens to loss if the score of the correct class decreases slightly when it is equal to other scores?	Loss changes	Loss changes
At initialization W is small so all s ≈ 0. What is the loss Li, assuming C classes?	(C-1)	Log(c)

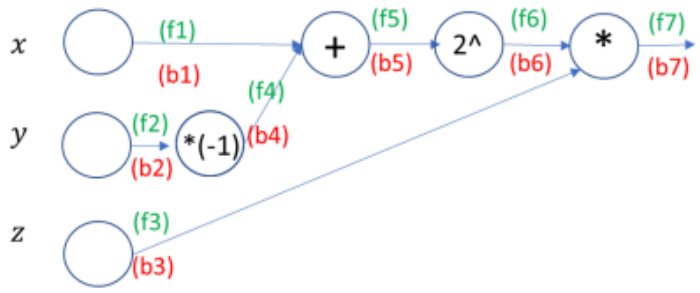
Svm 답안: loss changes 혹은 loss increases
Softmax 답안: loss changes or loss increases

Grading criteria:
Each 1 point, total 8 points

Backprop (10 pts, 4/6)

Consider the following computational graph:

- f=z * 2^(x-y)



2. Consider the loss function as the mean square error. Derive the gradients df/dx, df/dy, and df/dz required for learning each model constant in analytic form.

Answer: df/dx= ln2 *z * 2^(x-y), df/dy = -ln2*z * 2^(x-y), df/dz= 2^(x-y),

Grading criteria:

Your answer needs to be expressed with x, y and z. No partial points will not given in other cases.

1 point deduction for each wrong answer; if all answers are wrong, you will get 0 point.

3. When x=2, y=1,z=-3, use backpropagation to compute the gradients. Compute f1,f2, … f7 and then b1, b2, … b6, (b7=1) sequentially. Your answers should include b1~b7.

Answer: b1-7):

-6 ln2,

6 ln2,

2,

-6 ln2,

-6 ln2,

-3,

1

df/dx= -6ln(2), df/dy=6ln(2), df/dz=2

Grading criteria:

Each answer is 1 point, from b1 to b6.

4. CNN architecture

(1) The following is the first convolutional layer of AlexNet. Calculate the output size, memory usage, number of parameters, and FLOPs. If the calculations are complex, expressing them as formulas will also be accepted as correct answers.

	Input size			Layer				Output size					
Layer	C	H	W	filters	kernel	stride	pad	C	H	W	memory (KB)	params (k)	flop (M)
conv1	3	227		64	11	4	2						

Answer:

a. 64
b. 56
c. 64*56*56*4/1024
d. (3*11*11+1)*64/10^3
e. (64*56*56)*(3*11*11)/10^6

Grading criteria:

Each answer is 2 points, total 10 points.

If unit is wrong, no partial point is given.

5. Which of the following statements best explains why the Adam optimizer is considered superior to basic Stochastic Gradient Descent (SGD) in many deep learning applications?

A) Adam uses momentum to build velocity and maintain direction but has a simpler implementation than SGD.

B) Adam combines both momentum and adaptive learning rates, allowing it to perform well with
minimal hyperparameter tuning.

C) Adam avoids local minima better than SGD by performing full batch updates instead of minibatch updates.

D) Adam computes an exact Hessian matrix to provide second-order optimization, improving
convergence speed over SGD.

Answer:

B) Adam combines both momentum and adaptive learning rates, allowing it to perform well with minimal hyperparameter tuning.

6. Which of the following best describes the role of the chain rule in backpropagation through a
neural network?
A) The chain rule helps compute the weights for each layer during the forward pass.
B) The chain rule ensures that gradients from each layer are propagated forward to minimize
loss.
C) The chain rule allows for calculating the gradient of the loss with respect to each parameter
by combining local gradients along the computational graph.
D) The chain rule provides a method for updating the biases independently of the weights.

Answer:
C) The chain rule allows for calculating the gradient of the loss with respect to each parameter by combining local gradients along the computational graph.

7. You are training a convolutional neural network (CNN) with Batch Normalization layers using the ReLU activation function. However, during training, you notice that some neurons in the network are not updating their weights and have consistently
zero activations.
Which of the following could explain this behavior? (Select all that apply)
(a) The input data contains negative values.
(b) The learning rate is too high, causing gradient explosion.
(c) The neurons encountered the "dying ReLU" problem.
(d) The initial weights of the network were too small.

Solution to Q1 : (b),(c)
Grading criteria:
If you chose (b) or (c), all points will be given.

8. Consider two different optimizers: Stochastic Gradient Descent (SGD) and Adam, used to train a deep neural network on the same dataset.
Which of the following statements are true regarding their behavior during
training? (Select all that apply)
a) Adam generally converges faster than SGD due to adaptive learning rates.
b) Adam is more prone to overfitting since it performs better on training data.
c) SGD with momentum can potentially match or exceed Adam's performance

with well-tuned hyperparameters.
d) Adam performs better on very large batch sizes compared to SGD.

Answer: (a), (c)
Grading criteria:
If you chose (a) or (c), all points will be given.

9. Charlie trained two models on a given image dataset. The two models follow a Convolutional Neural Network (CNN) and Fully-Connected Neural Network (FCNN) architecture. Charlie had good classification performance with both models and saved the hyperparameters for each model.
Suppose that each image in the given dataset (both train and test) has all its pixels randomly shuffled, like a slide puzzle. The order of shuffling is the same for each image. If we train each model with the hyperparameters Charlie found and measure its performance, how would it perform compared to the original performance? Select the correct answers.

(a) Since the same hyperparameters were used and shuffled in the same order, both models performed similarly to the original.
(b) The performance of the FCNN model is about the same as before, but the performance of the CNN is degraded.
(c) The performance of the CNN model is about the same as before, but the performance of the FCNN is degraded.
(d) Shuffling each pixel makes it harder for the two models to learn, as they are not recognizable even to humans, thus reducing the performance of both models.

Answer: (b)

10. Which of the following statements is not true about the Rectified Linear Unit (ReLU) activation function?

a. ReLU does not suffer from saturation in its positive region.

b. ReLU is computationally efficient and converges faster than Sigmoid and Tanh.

c. ReLU outputs are zero-centered, which helps in gradient updates.

d. ReLU can suffer from "dead neurons" when the input is negative.

Answer: c

11. Choose the incorrect statement regarding the interpretation of Dropout.

(a) It acts as data augmentation to prevent the model from overfitting.

(b) It encourages the network to learn redundant features.

(c) It trains an ensemble of many subnetworks.

(d) It can be interpreted as a regularization technique that prevents the model from memorizing the training data.

Solution: (a)

12. What are the main differences between Batch Normalization, Layer Normalization, and Instance Normalization, and which one would be better to use in CNNs with larger batch sizes?

Answer. Each normalization technique differs in the dimensions over which the normalization is applied. Batch Normalization normalizes over the mini-batch dimension and computes the mean and variance across the entire mini-batch for each feature. Layer Normalization normalizes across all the features of a single data point over the entire layer. Instance Normalization normalizes each feature map independently for each sample in the mini-batch. Among them, batch normalization is the ideal one in CNN with a larger batch size since it helps address internal covariate shifts and enables faster training.

Grading criteria:

(2.5) BN, (2.5) LN, (2.5) IN, (2.5) BN is more effective than others given a large-sized batch.

You will be deducted in these cases:

-Lack of details (-1): for example, "LN normalizes each layer" and "IN normalizes each instance."

-If the dimensions of BN/LN/IN you specified are wrong, subtract -1 score.

(BN's mean/std dim: 1xCx1x1, LN's mean/std dim: Nx1x1x1, IN's mean/std dim: NxCx1x1)

(BN is applied to Nx1xHxW and produces the mean/variance of 1xCx1x1.

LN is applied to 1xCxHxW and produces the mean/variance of Nx1x1x1.

IN is applied to 1x1xHxW and produces the mean/variance of NxCx1x1.)

-No specific answer or information: wrong answer

-LN/IN: no details. The description for batch/feature/spatial dimensions is missing. (-1. -1)

13. Suppose you designed a neural network. Even though you gave them enough learning rates, you notice that the training loss is almost unchanged.

(a) Explain why this problem occurs and how to mitigate it.

A1: This issue is often caused by the vanishing gradient problem, where the gradients become smaller and smaller during backpropagation, making it difficult for the early layers to learn effectively. One way to mitigate this is through proper weight initialization. By initializing weights using methods based on the number of input and output neurons in each layer, such as uniform or normal distribution, you can ensure that the signal's variance remains stable as it passes through the layers.

(b) Describe how the choice of activation function affects this and what activation function can be used if the chosen one has a negative impact.

A2: Since the output of the activation function directly affects the gradients, functions like Sigmoid and Tanh can cause vanishing gradients because their gradients approach zero as the input value grows larger in magnitude. ReLU, while maintaining consistent gradients in the positive region, has a problem in the negative region where gradients become zero, a condition known as the "dying ReLU" problem. To address this, you can use modified activation functions like Leaky ReLU, which allow a small, non-zero gradient in the negative region.

14. Initialization

(1) Considering Xavier initialization, fill in the blanks.

```
dim=[4096] * 7
hs =[]
x= np.random.randn(16, dims[0])
for Din, Dout in zip(dims[:-1], dim[1:]):
    W = np.random.randn(Din, Dout)_____
    x = np.tanh(x.dot(W))
    hs.append(x)
```

(2) As implemented in (1), Xavier Initialization is designed to ensure that the input and output _____ are equal, allowing the activations in a layer to update correctly.

(3)

(4) In the code in (1), we replaced tanh with ReLU and observed significant portions of activations collapse to zero. How can we mitigate this issue by means of the initialization? What is the name of such initialization?

Answer:

(1) /np.sqrt(Din)

(2) variance

(3) MSRA initialization (or Kaiming init)

Grading criteria:

(a) 3 points

(b) 3 points

(c) 4 points

In (b), answers that mean the same (e.g., std, distribution and scale) are right.

In (c), if your answers include the write name (e.g, Kaiming or MSRA), you will get the full points. If your explanation is wright but you did not mention the name, you will get 1 point.

15. Hyperparameter Search

In the context of hyperparameter search methods, in what case is random search more effective than grid search? Provide a short answer.

Answer: When the importance of parameters greatly differs from each other.

Grading criteria:

If your answers are reasonable, you will get full points.

True/False

Consider a pre-trained convolutional neural network (CNN) that was trained on the ImageNet dataset to classify images into 1000 categories. You are now using this pre-trained model for a new task, which is to classify a smaller dataset of 50 categories in a completely different domain (e.g., medical images). You decide to use transfer learning to fine-tune the model for this new task.

Which of the following statements is true? Select all that apply.

16. True or False: If your new dataset is very small, it is usually better to fine-tune the entire pre-trained model rather than freezing the earlier layers.

Answer – **False:** If your new dataset is very small, it is often better to **freeze the earlier layers** and only fine-tune the later layers (or add new layers). Fine-tuning the entire pre-trained model on a small dataset may lead to overfitting because the model has too many parameters relative to the amount of data available.

17. True or False: Transfer learning is generally more effective when the new task's dataset is large, as a large dataset helps prevent overfitting to the pre-trained model's original task.

Answer – **False:** Transfer learning is often more effective when the new task's dataset is **small**, not large. Transfer learning is particularly useful when you don't have enough data to train a model from scratch, as the pre-trained model provides a strong starting point. If the new dataset is large, training a model from scratch might be a feasible option, and transfer learning might not be necessary.