Quiz 3

- Due No due date
- Points 10
- Questions 10
- Available Jan 30 at 6pm - Feb 1 at 11:59pm
- Time Limit None
- Allowed Attempts 3

Instructions

This quiz primarily covers lectures 5-6, but you are expected to be familiar with concepts from previous lectures as well.

Several of the questions refer to hidden slides that were not presented in class.

Some of the questions also require you to read additional material, links to which are posted in the quiz questions.

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	22 minutes	6.75 out of 10
LATEST	Attempt 3	7 minutes	6.25 out of 10
	Attempt 2	22 minutes	6.75 out of 10
	Attempt 1	89 minutes	6.75 out of 10

Correct answers are hidden.

Score for this attempt: 6.25 out of 10
Submitted Feb 1 at 11:20pm
This attempt took 7 minutes.
Incorrect

Question 1
0 / 1 pts

For this question, please read the paper: [Rumelhart, Hinton and Williams (1986 (http://www.cs.toronto.edu/~hinton/absps/naturebp.pdf)) (http://www.cs.toronto.edu/~hinton/absps/naturebp.pdf)](http://www.cs.toronto.edu/~hinton/absps/naturebp.pdf) .

[Can be found at: http://www.cs.toronto.edu/~hinton/absps/naturebp.pdf]

One drawback of the learning procedure in the paper is that the error-surface may contain local minima so that gradient descent is not guaranteed to find a global minimum.

This happens if the network has **more** than enough connections.

[✓ True]
False

"Adding a few more connections creates extra dimensions in weight-space and these dimensions provide paths around the barriers that create poor local minima in the lower dimensional subspaces" - p535

Answer key: Happens if the network has *just* enough connections. The question here asks if the network has "more than enough" connections, which in that case, it will be able to create a path to go around this barrier.

Partial

Question 2

0.75 / 1 pts

(Select all that apply) As discussed in lecture, which of the following is true for the backpropagation algorithm?

Hint: Lecture 5, starting at "training by backprop". Lec 5 (Pages 58 - 92)

- [✓] It cannot be performed without first doing a feed-forward pass of the input(s) through the network
- [ ] It can be used to compute the derivative of the divergence with respect to the input of the network
- [✓] It is used to compute derivatives that are required for the gradient descent algorithm that trains the network
- [ ] It computes the derivative of the average divergence for a batch of inputs
- [✓] It computes the derivative of the divergence between the true and desired outputs of the network for a training input

Question 3

1 / 1 pts

Consider a perceptron in a network that has the following vector activation:

$$y_j = \prod_{j \neq i} z_i$$

Where y_j is the j-th component of column vector y, and z_i is the i-th component of column vector z. Using the notation from lecture, which of the following is true of the derivative of y w.r.t. z? (select all that are true)

Hint: Vector Calculus Notes 1 (lecture 5, slide 135 and beyond)

- [✓] It is a matrix whose (i, j)th component where $i \neq j$ is given by $\prod_{k \neq i, k \neq j} z_k$
- [ ] It is a row vector whose i-th component is given by $\prod_{j \neq i} z_j$
- [ ] It is a column vector whose i-th component is given by $\prod_{j \neq i} z_j$
- [✓] It will be a matrix whose diagonal entries are all 0.
- [ ] It is a matrix whose (i,j)th component is given by $z_i z_j$

Question 4

1 / 1 pts

Backpropagation can be applied to any differentiable activation function.

[✓ True]

False

Incorrect

Question 5

0 / 1 pts

Let *d* be a scalar-valued function with multivariate input, *f* be a vector-valued function with multivariate input, and *X* be a vector such that y = d(f(X)). Using the lecture's notation, assuming the output of *f* to be a column vector, the derivative $\nabla_f y$ of y with respect to f(X) is…

Hint: (Lecture 4 and) Lecture 5, Vector calculus, Notes 1.

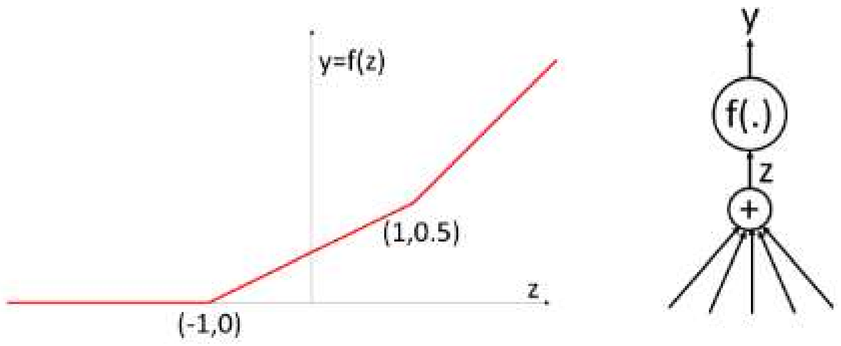
- [✓] A row vector
- [✓] A column vector
- [ ] Composed of the partial derivatives of y w.r.t the components of X
- [ ] A matrix

Partial

Question 6

0.5 / 1 pts

The following piecewise linear function with "hinges" at (-1,0) and (1,0.5) is used as an activation for a neuron. The slope of the last segment is 40 degrees with respect to the *z* axis (going anti-clockwise). Our objective is to find a z that minimizes the divergence div(y,d). Which of the following update rules is a valid subgradient descent update rule at z=1? Here η is the step size and is a positive number. The superscript on z represents the step index in an iterative estimate. The derivative $\frac{\partial div(y,\, d)}{\partial\, z}$ is computed at $z^k = 1$. The value of η must not factor into your answer (i.e. remember that η has only been included in the equations for completeness sake and do not argue with us that you can always adjust η to make any answer correct ☺)



Hint: Lecture 5, slides 112-114

- ☐ $z^{k+1} = z^k - \eta 0.1 \frac{\partial div(y,d)}{\partial y}$
- ☐ $z^{k+1} = z^k - \eta \frac{\partial div(y,d)}{\partial y}$
- ☐ $z^{k+1} = z^k + \eta \frac{\partial div(y,d)}{\partial y}$
- ☐ $z^{k+1} = z^k - \eta 0.25 \frac{\partial div(y,d)}{\partial y}$
- ✓ $z^{k+1} = z^k - \eta 0.75 \frac{\partial div(y,d)}{\partial y}$

Incorrect

Question 7

0 / 1 pts

The KL divergence between the output of a multi-class network with softmax output $y = [y_1 \ldots y_K]$ and *desired* output $d = [d_1 \ldots d_K]$ is defined as $KL = \sum_i d_i \log d_i - \sum_i d_i \log y_i$. The first term on the right hand side is the entropy of d , and the second term is the *Cross-entropy* between d and y , which we will represent as $Xent(y,d)$. Minimizing the KL divergence is strictly equivalent to minimizing the cross entropy, since $\sum_i d_i \log d_i$ is not a parameter of network parameters. When we do this, we refer to $Xent(y,d)$ as the cross-entropy loss.

Defined in this manner, which of the following is true of the cross-entropy loss $Xent(y,d)$? Recall that in this setting both y and d may be viewed as probabilities (i.e. they satisfy the properties of a probability distribution).

- ✓ It's derivative with respect to y goes to zero at the minimum (when y is exactly equal to d)

- [ ] It goes to 0 when y equals d
- [ ] It only depends on the output value of the network for the correct class
- [✓] It is always non-negative

If d is not one hot (e.g. when we use label smoothing), the cross entropy may not be 0 when d = y.

For one-hot d, we saw in class that the KL divergence is equal to the cross entropy. Also, in this case, at d=y, the gradient of the DL divergence (and therefore Xent(y,d)) is not 0.

Question 8

1 / 1 pts

In order to maximize the possibility of escaping local minima and finding the global minimum of a generic function, the best strategy to manage step sizes during gradient descent is:

Hint: Lecture 6, "Issues 2"

- ○ To start with a large, non-divergent step size (e.g. less than twice the optimal step size for a quadratic approximation at the initial location) and gradually decrease it over iterations
- ○ To maintain a step size consistently close to the optimal step size (e.g. close to the inverse second derivative at the current estimate)
- ○ To keep the step size low throughout to prevent divergence into a local minima
- [✓] To start with a large, divergent step size (e.g. greater than twice the optimal step size for a quadratic approximation at the initial location) and gradually decrease it over iterations

Question 9

1 / 1 pts

What are the challenges of using Newton’s method with neural networks?

Hint: Lecture 6, "Issues 1"

- [✓] It can produce unstable updates if the optimized function is not strictly convex
- [ ] It has a very large Jacobian
- [ ] It cannot find the minimum in any quadratic function
- [✓] Its memory usage scales with the square of the number of weights
- [✓] It is difficult to compute the inverse of the Hessian

Question 10

1 / 1 pts

Let $f(.)$ be an affine function that you would like to optimize. At your current location, $x = 3$, $f(x) = 7$ and $f'(x) = 2$. After one iteration of gradient descent with a learning rate = 0.1, your new location has a value of x = 2.8 and a value of $f(x)$ = 6.6 . (Truncate your answer to 1 digit after the decimal point, i.e. enter your answer in the format x.x, e.g. 4.5. If you use any other format canvas may mark your answer as being wrong)

Hint: Basic gradient descent from the lectures.

Answer 1:
2.8
Answer 2:
6.6

x_new = x_old - (lr * f ‘(x))

x_new = 3 - 0.2 = 2.8

affine function

f(x) = 7 when x = 3

so function is f(x) = 2*x + 1

new_f(x) = 2*(x_new) + 1

= (2*2.8) + 1 = 6.6

Quiz Score: 6.25 out of 10