Quiz-03

Due Feb 5 at 11:59pm **Points** 10 **Questions** 10
Available Feb 3 at 11:59pm - Feb 5 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.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	198 minutes	9 out of 10
LATEST	Attempt 2	198 minutes	9 out of 10
	Attempt 1	2,386 minutes	7.5 out of 10

ⓘ Correct answers will be available on Feb 6 at 12am.

Score for this attempt: **9** out of 10
Submitted Feb 5 at 7:14pm
This attempt took 198 minutes.

Question 1 1 / 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)** .

[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.

Question 2

1 / 1 pts

(**Select all that apply**) Which of the following is true of the vector and scalar versions of backpropagation?

Hint: Lecture 5

- [ ] Both scalar backpropagation and vector backpropagation are optimization algorithms that are used to find parameters that minimize a loss function

- [ ] Scalar backpropagation is required for scalar activation functions, while vector backpropagation is essential for vector activation functions

- [x] Scalar backpropagation and vector backpropagation only differ in their arithmetic notation and the implementation of their underlying arithmetic

- [x] Scalar backpropagation rules explicitly loop over the neurons in a layer to compute derivatives, while vector backpropagation computes derivative terms for all of them in a single matrix operation

Question 3

1 / 1 pts

Backpropagation can be applied to any differentiable activation function.

- (•) True
- ( ) False

You can propagate derivatives backward through any differentiable activation function, or even activations that only have a finite subgradient.

Question 4

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 (and beyond)

- [x] 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 column vector whose i-th component is given by $\prod_{j \neq i} z_j$
- [ ] It is a row vector whose i-th component is given by $\prod_{j \neq i} z_j$
- [x] 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 5

1 / 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.

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

Question 6

1 / 1 pts

Which of the following are valid subgradients of a RELU, given by Relu(x), at x = 0? We will represent the subgradient as $\nabla_{subgrad} RELU\,(x)$

Hint: Lecture 5, slides 112-114.

- [x] $\nabla_{subgrad} RELU\,(0) = 1$
- [x] $\nabla_{subgrad} RELU\,(0) = 0.5$
- [ ] $\nabla_{subgrad} RELU\,(0) = 1.5$
- [x] $\nabla_{subgrad} RELU\,(0) = 0$
- [ ] $\nabla_{subgrad} RELU\,(0) = -0.5$

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).

- [x] It is always non-negative
- [ ] It goes to 0 when y equals d
- [x] It only depends on the output value of the network for the correct class
- [ ] It's derivative with respect to y goes to zero at the minimum (when y is exactly equal to d)

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, 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

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 keep the step size low throughout to prevent divergence into a local minima

To maintain a step size consistently close to the optimal step size (e.g. close to the inverse second derivative at the current estimate)

See lecture for explanation.

Question 9

1 / 1 pts

Gradient descent with a fixed step size _____ for all convex functions (Fill in the blank)

Hint: Lecture 6

Always converges to some point

Always converges to a local minimum

- Always converges to a global minimum
- Does not always converge

Question 10

1 / 1 pts

Let f be a quadratic function such that at $x = 1$, $f(x) = 10$, $f'(x) = -4$, and $f''(x) = 1$. The minimum has a value of x = 5 and a value of $f(x)$ = 2 . (Truncate your answer to 1 digit after the decimal point i.e. enter your answer in the format x.x, e.g. 4.5)

Hint: Lecture 6 "Convergence for quadratic surfaces"

Answer 1:

5

Answer 2:

2

Quiz Score: **9** out of 10