⚠ This quiz has been regraded; your score was not affected.

Quiz-03

- Due Sep 15 at 11:59pm
- Points 10
- Questions 10
- Available Sep 13 at 6pm - Sep 15 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	Regraded
LATEST	Attempt 1	1,379 minutes	6 out of 10	6 out of 10

⚠ Correct answers are hidden.

Score for this attempt: 6 out of 10
Submitted Sep 15 at 3:44pm
This attempt took 1,379 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.

IncorrectQuestion 2
0 / 1 pts

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

Hint: Lecture 5

- ☐ Scalar backpropagation and vector backpropagation only differ in their arithmetic notation and the implementation of their underlying arithmetic
- ☑ Both scalar backpropagation and vector backpropagation are optimization algorithms that are used to find parameters that minimize a loss function
- ☑ 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
- ☐ Scalar backpropagation is required for scalar activation functions, while vector backpropagation is essential for vector activation functions

IncorrectQuestion 3
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.

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

IncorrectQuestion 4
Original Score: 0 / 1 pts Regraded Score: 0 / 1 pts

This question has been regraded.

Which of the following is true given the Hessian of a scalar function with multivariate inputs?

Hint: Lec 4 "Unconstrained minimization of a function". Also note that an eigen value of 0 indicates that the function is flat (to within the second derivative) along the direction of the corresponding Hessian Eigenvector.

- [ ] The eigenvalues are all strictly positive at a local minimum.
- [ ] The eigenvalues are all strictly negative at a local maximum.
- [x] The eigenvalues are all strictly positive at global minima, but not at local minima.
- [x] The eigenvalues are all non-negative at local minima.

Question 5
1 / 1 pts

We are given a binary classification problem where the training data from both classes are linearly separable. We compare a perceptron, trained using the perceptron learning rule with a sigmoid-activation perceptron, trained using gradient descent that minimizes the L2 Loss. In both cases, we restrict the weights vector of the perceptron to have finite length. In all cases, we will say the algorithm has found a “correct” solution if the learned model is able to correctly classify the training data. Which of the following statements are true (select all that are true).

Hint: See slides 13-32, lecture 6

- [x] The perceptron algorithm will always find the correct solution.
- [ ] We cannot make any statement about the truth of falsity of the other options provided, based only on the information provided.
- [x] There are situations where the gradient-descent algorithm will not find the correct solution.
- [ ] The gradient-descent algorithm will always find the correct solution.

UnansweredQuestion 6
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).

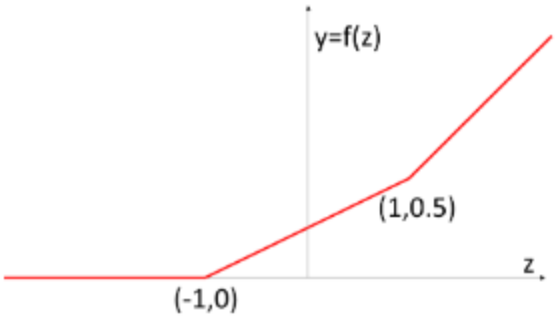
- [ ] It only depends on the output value of the network for the correct class
- [ ] It is always non-negative
- [ ] It goes to 0 when y equals d
- [ ] It's derivative with respect to y goes to zero at the minimum (when y is exactly equal to d)

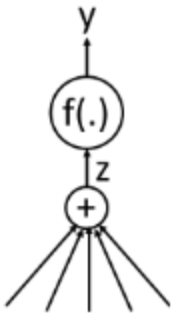
If d is not one hot (e.g. when we use label smoothing), the cross entropy may not be 0 when d = y.

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 7
1 / 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

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

The correct choices for n are those that align with the slope of the piecewise linear function at z = 1

Question 8

1 / 1 pts

Gradient descent yields a solution that is not sensitive to how a network's weights are initialized.

Hint: Basic gradient descent from lecture 5 - slide 5

- ○ True
- ◉ False

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 a local minimum
- ○ Always converges to a global minimum
- ◉ Does not always converge
- ○ Always converges to some point

Question 10

1 / 1 pts

Let $f(.)$ be a scalar-valued function with multivariate input and $x = [x_1, x_2]$ be a two-component vector such that $y = f(x)$. y is being minimized using RProp from lecture. In the k-th iteration, the derivative of y with respect to x_1 is $\frac{dy}{dx_1} = 2$, the derivative of y with respect to x_2 is $\frac{dy}{dx_2} = -1$. As a result, x_1 has a step size of $\Delta x_1^{(k)} = 1$ and x_2 has a step size of $\Delta x_2^{(k)} = 1$. At the (*k*+1)-th iteration, the derivative of y with respect to x_1 is $\frac{dy}{dx_1} = 0.5$ and the derivative of y with respect to x_2 is $\frac{dy}{dx_2} = 1$. Which of the following is true about the step size at the (*k*+1)-th iteration?

Hint: Lecture 6, RProp

- $\Delta x_1^{(k+1)} > 1$ and $\Delta x_2^{(k+1)} < 1$
- $\Delta x_1^{(k+1)} > 1$ and $\Delta x_2^{(k+1)} > 1$
- $\Delta x_1^{(k+1)} < 1$ and $\Delta x_2^{(k+1)} < 1$
- $\Delta x_1^{(k+1)} < 1$ and $\Delta x_2^{(k+1)} > 1$

Quiz Score: 6 out of 10