Quiz-03 Results for Jiayi Huang

Correct answers are hidden.

Score for this attempt: 9 out of 10
Submitted Sep 14 at 11:08pm
This attempt took 673 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

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

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

Question 4
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)).** Further, $J_f(X)$ is the Jacobian of **f** w.r.t X. Using the lecture's notation, the derivative of y w.r.t. X is…

Hint: Lecture 5, Vector Calculus, Notes 1 and 2, Slides 135-149

- ( ) A matrix given by $\nabla_f \mathbf{y} J_f(X)$
- ( ) A column vector given by $J_f(X) \nabla_f \mathbf{y}$
- (•) A row vector given by $\nabla_f y J_f(X)$
- ( ) Either a column vector given by J_f (X)\nabla_f \text{y} or a row vector given by \nabla_f y J_f (X)

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

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

Question 6
1 / 1 pts

The KL divergence between the output of a multi-class network with softmax output $\boldsymbol{y} = [y_1 \ldots y_K]$ and *desired* output $\boldsymbol{d} = [d_1 \ldots d_K]$ is defined as $\boldsymbol{KL} = \sum_i d_i \log d_i - \sum_i d_i \log y_i$. The first term on the right hand side is the entropy of $\boldsymbol{d}$, and the second term is the *Cross-entropy* between $\boldsymbol{d}$ and $\boldsymbol{y}$, which we will represent as $\boldsymbol{Xent(y,d)}$. 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 $\boldsymbol{Xent(y,d)}$ as the cross-entropy loss.

Defined in this manner, which of the following is true of the cross-entropy loss $\boldsymbol{Xent(y,d)}$? Recall that in this setting both $\boldsymbol{y}$ and $\boldsymbol{d}$ 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's derivative with respect to $\boldsymbol{y}$ goes to zero at the minimum (when $\boldsymbol{y}$ is exactly equal to $\boldsymbol{d}$)
- [ ] It goes to 0 when $\boldsymbol{y}$ equals $\boldsymbol{d}$
- [x] 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 7

1 / 1 pts

Tom decides to construct a new vector activation function based on the Softplus to output probabilities.

$$SR(z_i) = \frac{Softplus(z_i)}{\sum_j Softplus(z_j)}$$

Which of the following statements is true (multiple choice).

Hint: To understand the Softplus check https://en.wikipedia.org/wiki/Rectifier_(neural_networks), The above is also similar to the Softmax activation with Softplus replacing the exponential function (Which are similar: smooth and monotonically increasing). You can check the derivatives in lecture 5 slides 99 - 102

- [x] The derivative of SR(z_i) with respect to z_i will be positive
- [ ] The sign of the derivative of SR(z_i) with respect to z_j (for j != i) depends on the signs of z_i and z_j, and cannot be predicted without knowing them
- [ ] The derivative of SR(z_i) with respect to z_j (for j != i) will be positive
- [ ] The derivative of SR(z_i) with respect to z_i will be negative
- [x] The derivative of SR(z_i) with respect to z_j (for j != i) will be negative
- [ ] The sign of the derivative of SR(z_i) with respect to z_j (for j != i) depends on z_i and z_j, and cannot be predicted without knowing them

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 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, 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

See lecture for explanation.

Incorrect Question 9

0 / 1 pts

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

Hint: Lecture 6

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

It might not necessarily converge, such as the function y = abs(x). Here, the fixed step size might make gradient descent bounce around the minimum.

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