Quiz-03 Results for Ziyu Han

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

Score for this attempt: **7.5** out of 10

Submitted Feb 5 at 3:55pm

This attempt took 2,386 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 version of gradient descent changes each weight by an amount proportional to the accumulated $\delta E/\delta w$.

$$\Delta w = -\epsilon \frac{\delta E}{\delta w}$$

Select all that are true about this method:

- [ ] This method converges as rapidly as methods that make use of second derivatives.
- [ ] It cannot be implemented by local computations in parallel hardware.
- [x] It’s simpler than methods that use second derivatives.

"This method does not converge as rapidly as methods which make use of the second derivatives, but it is much simpler [...]" p535

- [x] It can be improved without sacrificing simplicity and locality.

"It can be significantly improved, without sacrificing the simplicity and locality, [...]" p535

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

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 (and beyond)

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

Question 4

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.

Partial

Question 5

0.5 / 1 pts

(Select all that apply) At any point, the gradient of a scalar function with multivariate inputs…

Hint: Lecture 4, "Gradient of a scalar function of a vector" and "properties of a gradient".

- [ ] Is parallel to equal-value contours of the function
- [ ] Is in the direction of steepest ascent
- [x] Is the vector of local partial derivatives w.r.t. all the inputs
- [ ] Is in the direction of steepest descent

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.

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

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

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
- [x] 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

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
- (selected) 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

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

- Always converges to a local minimum
- Does not always converge
- Always converges to a global minimum
- 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: **7.5** out of 10