[redacted]

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

Score for this attempt: **8** out of 10

Submitted Feb 5 at 1:28pm

This attempt took 891 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:

- [ ] 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

[redacted]

This method converges as rapidly as methods that make use of second derivatives.

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

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

Incorrect

Question 4

0 / 1 pts

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.

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

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)). 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

- ( ) 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)
- ( ) A column vector given by $J_f(X)\nabla_f \mathrm{y}$
- (•) A row vector given by $\nabla_f y J_f(X)$
- ( ) A matrix given by $\nabla_f \mathrm{y} J_f(X)$

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 $\nabla_{subgrad} RELU(x)$

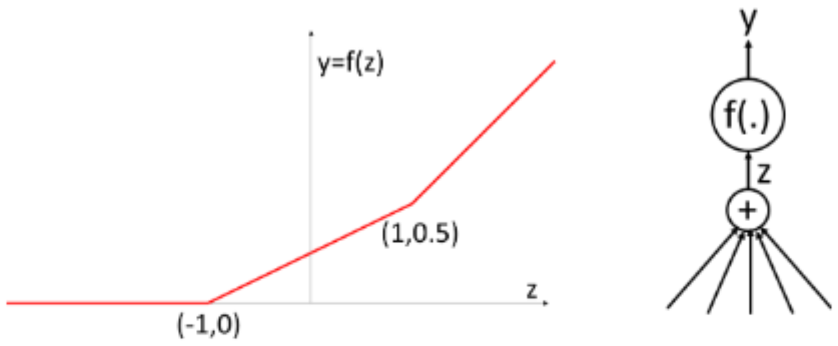
Hint: Lecture 5, slides 112-114.

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

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

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

Incorrect

Question 8

0 / 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 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

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 a global minimum

Does not always converge

Always converges to some point

Always converges to a local minimum

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

Quiz Score: **8** out of 10