**Optimization Algorithms**

**Total points**10

**1.**

**Question 1**

Which notation would you use to denote the 3rd layer’s activations when the input is the 7th example from the 8th minibatch?

**1 point**

a^{[3]\{7\}(8)}*a*[3]{7}(8)

a^{[3]\{8\}(7)}*a*[3]{8}(7)

a^{[8]\{7\}(3)}*a*[8]{7}(3)

a^{[8]\{3\}(7)}*a*[8]{3}(7)

**2.**

**Question 2**

Suppose you don't face any memory-related problems. Which of the following make more use of vectorization.

**1 point**

Mini-Batch Gradient Descent with mini-batch size m/2*m*/2.

Stochastic Gradient Descent

Stochastic Gradient Descent, Batch Gradient Descent, and Mini-Batch Gradient Descent all make equal use of vectorization.

Batch Gradient Descent

**3.**

**Question 3**

Which of the following is true about batch gradient descent?

**1 point**

It is the same as the mini-batch gradient descent when the mini-batch size is the same as the size of the training set.

It is the same as stochastic gradient descent, but we don't use random elements.

It has as many mini-batches as examples in the training set.

**4.**

**Question 4**

While using mini-batch gradient descent with a batch size larger than 1 but less than m the plot of the cost function J*J* looks like this:

Chart, line chart, scatter chart

Description automatically generated

Which of the following do you agree with?

**1 point**

No matter if using mini-batch gradient descent or batch gradient descent something is wrong.

If you are using mini-batch gradient descent, this looks acceptable. But if you're using batch gradient descent, something is wrong.

If you are using batch gradient descent, this looks acceptable. But if you're using mini-batch gradient descent, something is wrong.

If you are using mini-batch gradient descent or batch gradient descent this looks acceptable.

Which of the following do you agree with?

**0 / 1 point**

No matter if using mini-batch gradient descent or batch gradient descent something is wrong.

If you are using mini-batch gradient descent, this looks acceptable. But if you're using batch gradient descent, something is wrong.

If you are using batch gradient descent, this looks acceptable. But if you're using mini-batch gradient descent, something is wrong.

If you are using mini-batch gradient descent or batch gradient descent this looks acceptable.

**Incorrect**

No. The cost is larger than when the process started, this is not right at all.

**----**

**5.**

**Question 5**

Suppose the temperature in Casablanca over the first two days of March are the following:

March 1st: \theta\_1 = 10^{\circ} \text{ C }*θ*1​=10∘ C

March 2nd: \theta\_2 = 25^{\circ} \text{ C }*θ*2​=25∘ C

Say you use an exponentially weighted average with \beta = 0.5*β*=0.5 to track the temperature: v\_0 = 0*v*0​=0, v\_t = \beta v\_{t-1} + (1- \beta) \, \theta\_t*vt*​=*βvt*−1​+(1−*β*)*θt*​. If v\_2*v*2​ is the value computed after day 2 without bias correction, and v\_2^{\text{corrected}}*v*2corrected​ is the value you compute with bias correction. What are these values?

**1 point**

v\_2 = 20*v*2​=20, v\_2^{\text{corrected}} = 15*v*2corrected​=15.

v\_2 = 20*v*2​=20, v\_2^{\text{corrected}} = 20*v*2corrected​=20.

v\_2 = 15*v*2​=15, v\_2^{\text{corrected}} = 20*v*2corrected​=20.

v\_2 = 15*v*2​=15, v\_2^{\text{corrected}} = 15*v*2corrected​=15.

**6.**

**Question 6**

Which of these is NOT a good learning rate decay scheme? Here, t*t* is the epoch number.

**1 point**

\alpha = 1.01^{t} \, \alpha\_0*α*=1.01*tα*0​

\alpha = e^{-0.01\, t}\alpha\_0*α*=*e*−0.01*tα*0​.

\alpha = \frac{\alpha\_0}{\sqrt{1 + t}}*α*=1+*t*​*α*0​​.

\alpha = \frac{\alpha\_0}{1 + 3\, t}*α*=1+3*tα*0​​

**---------  
Question 6**

Which of these is NOT a good learning rate decay scheme? Here, t*t* is the epoch number.

**0 / 1 point**

\alpha = 1.01^{t} \, \alpha\_0*α*=1.01*tα*0​

\alpha = e^{-0.01\, t}\alpha\_0*α*=*e*−0.01*tα*0​.

\alpha = \frac{\alpha\_0}{\sqrt{1 + t}}*α*=1+*t*​*α*0​​.

\alpha = \frac{\alpha\_0}{1 + 3\, t}*α*=1+3*tα*0​​

**Incorrect**

Incorrect. This is a good learning rate decay since it is a decreasing function of t*t*.

**7.**

**Question 7**

You use an exponentially weighted average on the London temperature dataset. You use the following to track the temperature: v\_{t} = \beta v\_{t-1} + (1-\beta)\theta\_t*vt*​=*βvt*−1​+(1−*β*)*θt*​. The red line below was computed using \beta = 0.9*β*=0.9. What would happen to your red curve as you vary \beta*β*? (Check the two that apply)

Chart, scatter chart

Description automatically generated

**1 point**

Decreasing \beta*β* will shift the red line slightly to the right.

Increasing \beta*β* will shift the red line slightly to the right.

Decreasing \beta*β* will create more oscillation within the red line.

Increasing \beta*β* will create more oscillations within the red line.

**8.**

**Question 8**

Consider this figure:

Diagram

Description automatically generated

These plots were generated with gradient descent; with gradient descent with momentum (\beta*β* = 0.5); and gradient descent with momentum (\beta*β* = 0.9). Which curve corresponds to which algorithm?

**1 point**

(1) is gradient descent with momentum (small \beta*β*), (2) is gradient descent with momentum (small \beta*β*), (3) is gradient descent

(1) is gradient descent. (2) is gradient descent with momentum (large \beta*β*) . (3) is gradient descent with momentum (small \beta*β*)

(1) is gradient descent with momentum (small \beta*β*). (2) is gradient descent. (3) is gradient descent with momentum (large \beta*β*)

(1) is gradient descent. (2) is gradient descent with momentum (small \beta*β*). (3) is gradient descent with momentum (large \beta*β*)

**9.**

**Question 9**

Suppose batch gradient descent in a deep network is taking excessively long to find a value of the parameters that achieves a small value for the cost function \mathcal{J}(W^{[1]},b^{[1]},...,W^{[L]},b^{[L]})J(*W*[1],*b*[1],...,*W*[*L*],*b*[*L*]). Which of the following techniques could help find parameter values that attain a small value for \mathcal{J}J? (Check all that apply)

**1 point**

Try initializing the weight at zero.

Try mini-batch gradient descent.

Normalize the input data.

Try using Adam.

**10.**

**Question 10**

In very high dimensional spaces it is most likely that the gradient descent process gives us a local minimum than a saddle point of the cost function. True/False?

**1 point**

False

True

**Question 10**

In very high dimensional spaces it is most likely that the gradient descent process gives us a local minimum than a saddle point of the cost function. True/False?

**0 / 1 point**

False

True

**Incorrect**

Incorrect. Due to the high number of dimensions it is much more likely to reach a saddle point, than a local minimum.

PASS 2 – 90% correct:

[**Skip to Main Content**](https://www.coursera.org/learn/deep-neural-network/exam/rqT6n/optimization-algorithms/attempt#main)

[[](https://www.coursera.org/)](https://www.coursera.org/)

Top of Form

SEARCH IN COURSE

Search

Bottom of Form

* **1**
* Viroopax Mirji

1. [Improving Deep Neural Networks: Hyperparameter Tuning, Regularization and Optimization](https://www.coursera.org/learn/deep-neural-network/home/welcome)
2. [Week 2](https://www.coursera.org/learn/deep-neural-network/home/week/2)
3. Optimization Algorithms

[**Previous**](https://www.coursera.org/learn/deep-neural-network/ungradedLti/REwvX/lecture-notes-w2)[**Next**](https://www.coursera.org/learn/deep-neural-network/programming/390Oe/optimization-methods)

## Optimization Algorithms

## Lecture Notes (Optional)

## Quiz

**[Quiz:](https://www.coursera.org/learn/deep-neural-network/exam/rqT6n/optimization-algorithms)**[Optimization Algorithms](https://www.coursera.org/learn/deep-neural-network/exam/rqT6n/optimization-algorithms)

[10 questions](https://www.coursera.org/learn/deep-neural-network/exam/rqT6n/optimization-algorithms)

## Programming Assignment

## Heroes of Deep Learning (Optional)

# **Optimization Algorithms**

Quiz20 minutes • 20 min

### Submit your assignment

**Due** March 20, 11:59 PM PDTMar 20, 11:59 PM PDT

**Attempts** 3 every 8 hours

Try again

### Receive grade

**To Pass** 80% or higher

### Your grade

70%

View Feedback

We keep your highest score

Like

Dislike

Report an issue

**Back**

# Optimization Algorithms

Graded Quiz • 20 min

**Due**Mar 20, 11:59 PM PDT

## Optimization Algorithms

**Total points**10

### 1.

**Question 1**

Which notation would you use to denote the 3rd layer’s activations when the input is the 7th example from the 8th minibatch?

**1 point**

a^{[8]\{7\}(3)}*a*[8]{7}(3)

a^{[8]\{3\}(7)}*a*[8]{3}(7)

a^{[3]\{8\}(7)}*a*[3]{8}(7)

a^{[3]\{7\}(8)}*a*[3]{7}(8)

### 2.

**Question 2**

Which of these statements about mini-batch gradient descent do you agree with?

**1 point**

You should implement mini-batch gradient descent without an explicit for-loop over different mini-batches so that the algorithm processes all mini-batches at the same time (vectorization).

When the mini-batch size is the same as the training size, mini-batch gradient descent is equivalent to batch gradient descent.

Training one epoch (one pass through the training set) using mini-batch gradient descent is faster than training one epoch using batch gradient descent.

### 3.

**Question 3**

Why is the best mini-batch size usually not 1 and not m, but instead something in-between? Check all that are true.

**1 point**

If the mini-batch size is m, you end up with stochastic gradient descent, which is usually slower than mini-batch gradient descent.

If the mini-batch size is m, you end up with batch gradient descent, which has to process the whole training set before making progress.

If the mini-batch size is 1, you lose the benefits of vectorization across examples in the mini-batch.

If the mini-batch size is 1, you end up having to process the entire training set before making any progress.

### 4.

**Question 4**

Suppose your learning algorithm’s cost J*J*, plotted as a function of the number of iterations, looks like this:

Chart, line chart

Description automatically generated

Which of the following do you agree with?

**1 point**

Whether you’re using batch gradient descent or mini-batch gradient descent, this looks acceptable.

Whether you’re using batch gradient descent or mini-batch gradient descent, something is wrong.

If you’re using mini-batch gradient descent, something is wrong. But if you’re using batch gradient descent, this looks acceptable.

If you’re using mini-batch gradient descent, this looks acceptable. But if you’re using batch gradient descent, something is wrong.

### 5.

**Question 5**

Suppose the temperature in Casablanca over the first two days of January are the same:

Jan 1st: \theta\_1 = 10^o C*θ*1​=10*oC*

Jan 2nd: \theta\_2 = 10^o C*θ*2​=10*oC*

(We used Fahrenheit in the lecture, so we will use Celsius here in honor of the metric world.)

Say you use an exponentially weighted average with \beta = 0.5*β*=0.5 to track the temperature: v\_0 = 0*v*0​=0, v\_t = \beta v\_{t-1} +(1-\beta)\theta\_t*vt*​=*βvt*−1​+(1−*β*)*θt*​. If v\_2*v*2​ is the value computed after day 2 without bias correction, and v\_2^{corrected}*v*2*corrected*​ is the value you compute with bias correction. What are these values? (You might be able to do this without a calculator, but you don't actually need one. Remember what bias correction is doing.)

**1 point**

v\_2 = 7.5*v*2​=7.5, v\_2^{corrected} = 10*v*2*corrected*​=10

v\_2 = 10*v*2​=10, v\_2^{corrected} = 10*v*2*corrected*​=10

v\_2 = 7.5*v*2​=7.5, v\_2^{corrected} = 7.5*v*2*corrected*​=7.5

v\_2 = 10*v*2​=10, v\_2^{corrected} = 7.5*v*2*corrected*​=7.5

### 6.

**Question 6**

Which of these is NOT a good learning rate decay scheme? Here, t*t* is the epoch number.

**1 point**

\alpha = \frac{\alpha\_0}{\sqrt{1 + t}}*α*=1+*t*​*α*0​​.

\alpha = 1.01^{t} \, \alpha\_0*α*=1.01*tα*0​

\alpha = e^{-0.01\, t}\alpha\_0*α*=*e*−0.01*tα*0​.

\alpha = \frac{\alpha\_0}{1 + 3\, t}*α*=1+3*tα*0​​

### 7.

**Question 7**

You use an exponentially weighted average on the London temperature dataset. You use the following to track the temperature: v\_{t} = \beta v\_{t-1} + (1-\beta)\theta\_t*vt*​=*βvt*−1​+(1−*β*)*θt*​. The yellow and red lines were computed using values \beta\_1*β*1​ and \beta\_2*β*2​ respectively. Which of the following are true?

Chart, shape, arrow

Description automatically generated

**1 point**

\beta\_1 =0*β*1​=0, \beta\_2 >0*β*2​>0.

\beta\_1 = \beta\_2*β*1​=*β*2​.

\beta\_1 < \beta\_2*β*1​<*β*2​.

\beta\_1 > \beta\_2*β*1​>*β*2​.

### 8.

**Question 8**

Consider the figure:

Shape, circle

Description automatically generated

Suppose this plot was generated with gradient descent with momentum \beta = 0.01*β*=0.01. What happens if we increase the value of \beta*β* to 0.10.1?

**1 point**

The gradient descent process starts oscillating in the vertical direction.

The gradient descent process moves less in the horizontal direction and more in the vertical direction.

The gradient descent process starts moving more in the horizontal direction and less in the vertical.

The gradient descent process moves more in the horizontal and the vertical axis.

### 9.

**Question 9**

Suppose batch gradient descent in a deep network is taking excessively long to find a value of the parameters that achieves a small value for the cost function \mathcal{J}(W^{[1]},b^{[1]},...,W^{[L]},b^{[L]})J(*W*[1],*b*[1],...,*W*[*L*],*b*[*L*]). Which of the following techniques could help find parameter values that attain a small value for \mathcal{J}J? (Check all that apply)

**1 point**

Try using Adam.

Normalize the input data.

Try mini-batch gradient descent.

Try initializing the weight at zero.

### 10.

**Question 10**

In very high dimensional spaces it is most likely that the gradient descent process gives us a local minimum than a saddle point of the cost function. True/False?

**1 point**

True

False