Back

Optimization algorithms

Quiz, 10 questions

**Congratulations! You passed!**

Next Item

Question 1

Correct

1 / 1 points

**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?

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

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

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

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

**Correct**

Question 2

Incorrect

0 / 1 points

**2. Question 2**

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

One iteration of mini-batch gradient descent (computing on a single mini-batch) is faster than one iteration of batch gradient descent.

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).

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

**This should not be selected**

Question 3

Incorrect

0 / 1 points

**3. Question 3**

Why is the best mini-batch size usually not 1 and not m, but instead something in-between?

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

**This should be selected**

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

**This should not be selected**

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

**This should not be selected**

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.

**This should be selected**

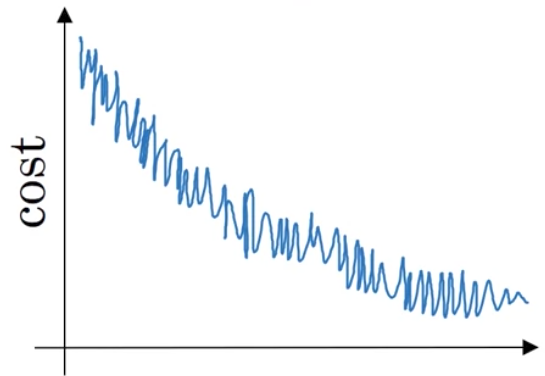
Question 4

Correct

1 / 1 points

**4. Question 4**

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



Which of the following do you agree with?

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

**Correct**

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

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.

Question 5

Correct

1 / 1 points

**5. Question 5**

Suppose the temperature in Casablanca over the first three 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 lecture, so 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 is bias correction doing.)

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

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

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

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

**Correct**

Question 6

Correct

1 / 1 points

**6. Question 6**

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

\alpha = \frac{1}{1+2\*t} \alpha\_0*α*=1+2∗*t*1​*α*0​

\alpha = e^t \alpha\_0*α*=*etα*0​

**Correct**

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

\alpha = 0.95^t \alpha\_0*α*=0.95*tα*0​

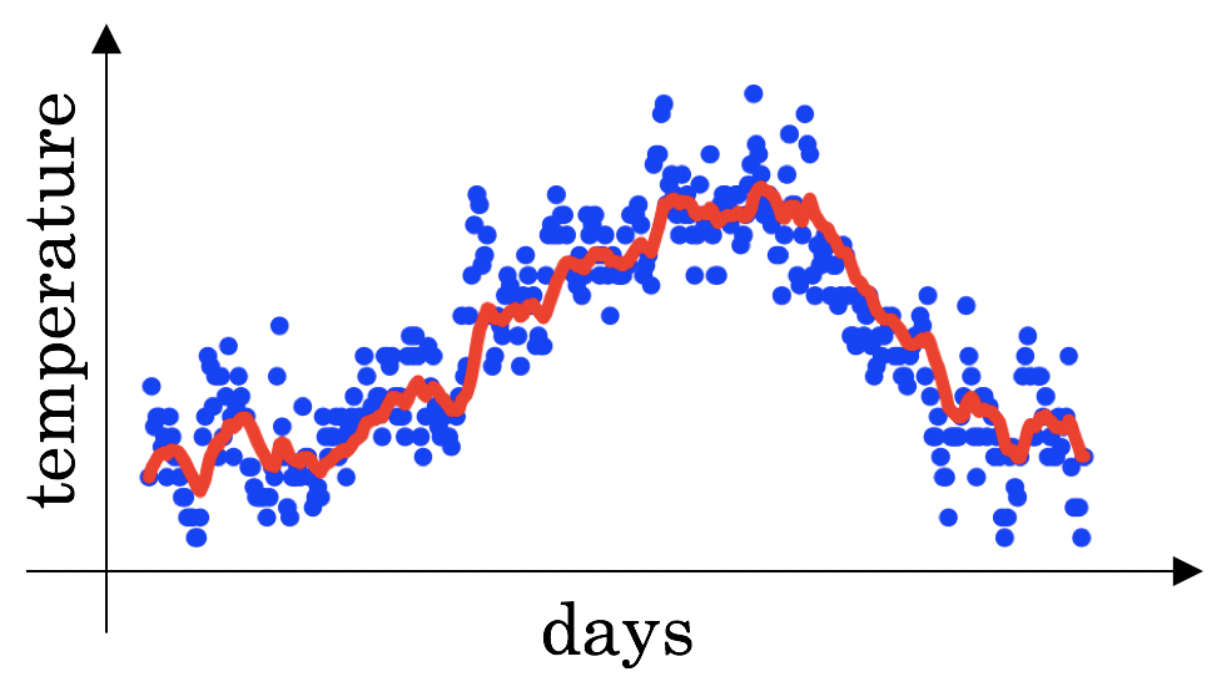
Question 7

Correct

1 / 1 points

**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)



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

**Un-selected is correct**

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

**Correct**

True, remember that the red line corresponds to \beta = 0.9*β*=0.9. In lecture we had a green line $$\beta = 0.98) that is slightly shifted to the right.

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

**Correct**

True, remember that the red line corresponds to \beta = 0.9*β*=0.9. In lecture we had a yellow line $$\beta = 0.98 that had a lot of oscillations.

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

**Un-selected is correct**

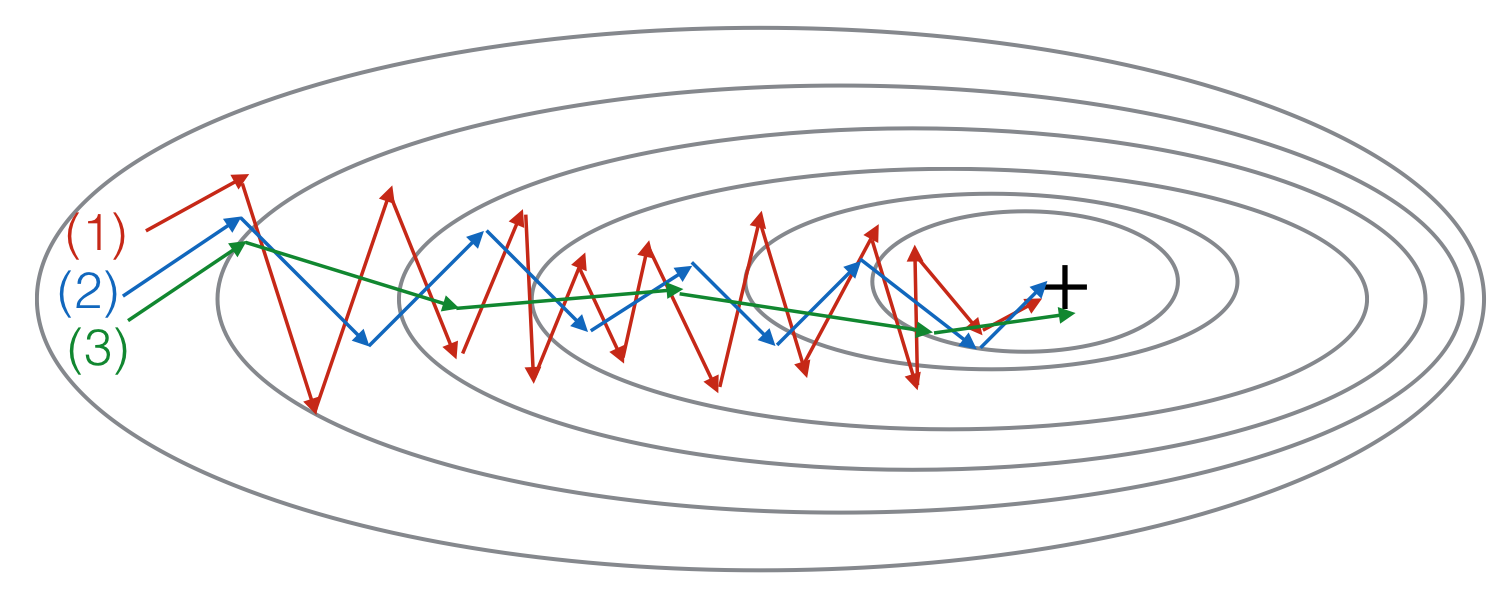
Question 8

Correct

1 / 1 points

**8. Question 8**

Consider this figure:



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

**Correct**

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

Question 9

Correct

1 / 1 points

**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)

Try mini-batch gradient descent

**Correct**

Try initializing all the weights to zero

**Un-selected is correct**

Try using Adam

**Correct**

Try better random initialization for the weights

**Correct**

Try tuning the learning rate \alpha*α*

**Correct**

Question 10

Correct

1 / 1 points

**10. Question 10**

Which of the following statements about Adam is False?

Adam combines the advantages of RMSProp and momentum

Adam should be used with batch gradient computations, not with mini-batches.

**Correct**

The learning rate hyperparameter \alpha*α* in Adam usually needs to be tuned.

We usually use “default” values for the hyperparameters \beta\_1, \beta\_2*β*1​,*β*2​ and \varepsilon*ε* in Adam (\beta\_1 = 0.9*β*1​=0.9, \beta\_2 = 0.999*β*2​=0.999, \varepsilon = 10^{-8}*ε*=10−8)