

Congratulations! You passed!

Grade
received **80%**

Latest Submission
Grade 80%

To pass 80% or
higher

[Go to next item](#)

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 / 1 point

- ☐ $a^{[8]}(7)(3)$
☐ $a^{[3]}(7)(8)$
☒ $a^{[3]}(8)(7)$
☐ $a^{[8]}(3)(7)$

[Expand](#)

 **Correct**

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

1 / 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).
☐ Training one epoch (one pass through the training set) using mini-batch gradient descent is faster than training one epoch using batch gradient descent.
☒ When the mini-batch size is the same as the training size, mini-batch gradient descent is equivalent to batch gradient descent.

[Expand](#)

 **Correct**

Correct. Batch gradient descent uses all the examples at each iteration, this is equivalent to having only one mini-batch of the size of the complete training set in mini-batch gradient descent.


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

0 / 1 point

- ☒ 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.

 **Correct**

- ☒ 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 1, you lose the benefits of vectorization across examples in the mini-batch.
☐ If the mini-batch size is m , you end up with stochastic gradient descent, which is usually slower than mini-batch gradient descent.

[Expand](#)

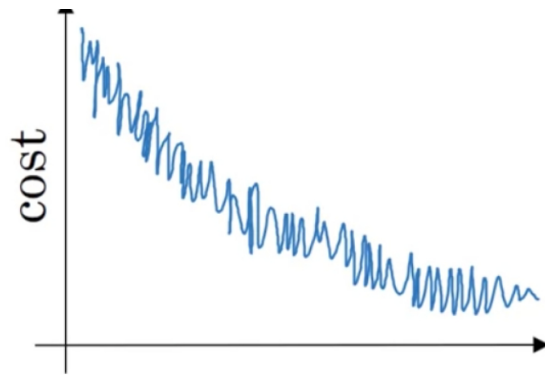
 **Incorrect**

You didn't select all the correct answers

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 looks like this:

1 / 1 point





You notice that the value of J is not always decreasing. Which of the following is the most likely reason for that?

- ☐ A bad implementation of the backpropagation process, we should use gradient check to debug our implementation.
- ☒ In mini-batch gradient descent we calculate $J(\hat{y}^{(t)}, y^{(t)})$ thus with each batch we compute over a new set of data.
- ☐ You are not implementing the moving averages correctly. Using moving averages will smooth the graph.
- ☐ The algorithm is on a local minimum thus the noisy behavior.

↗ Expand

✓ **Correct**

Yes. Since at each iteration we work with a different set of data or batch the loss function doesn't have to be decreasing at each iteration.

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

1 / 1 point

March 1st: $\theta_1 = 10^\circ \text{ C}$

March 2nd: $\theta_2 = 25^\circ \text{ C}$

Say you use an exponentially weighted average with $\beta = 0.5$ to track the temperature: $v_0 = 0$, $v_t = \beta v_{t-1} + (1 - \beta) \theta_t$. If v_2 is the value computed after day 2 without bias correction, and $v_2^{\text{corrected}}$ is the value you compute with bias correction. What are these values?

- ☒ $v_2 = 15, v_2^{\text{corrected}} = 20$.
- ☐ $v_2 = 20, v_2^{\text{corrected}} = 20$.
- ☐ $v_2 = 20, v_2^{\text{corrected}} = 15$.
- ☐ $v_2 = 15, v_2^{\text{corrected}} = 15$.

↗ Expand

✓ **Correct**

Correct. $v_2 = \beta v_{t-1} + (1 - \beta) \theta_t$ thus $v_1 = 5, v_2 = 15$. Using the bias correction $\frac{v_t}{1 - \beta^t}$ we get $\frac{15}{1 - (0.5)^2} = 20$.

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

1 / 1 point

- ☐ $\alpha = \frac{\alpha_0}{1 + 3t}$
- ☐ $\alpha = \frac{\alpha_0}{\sqrt{1 + t}}$.
- ☒ $\alpha = 1.01^t \alpha_0$
- ☐ $\alpha = e^{-0.01t} \alpha_0$

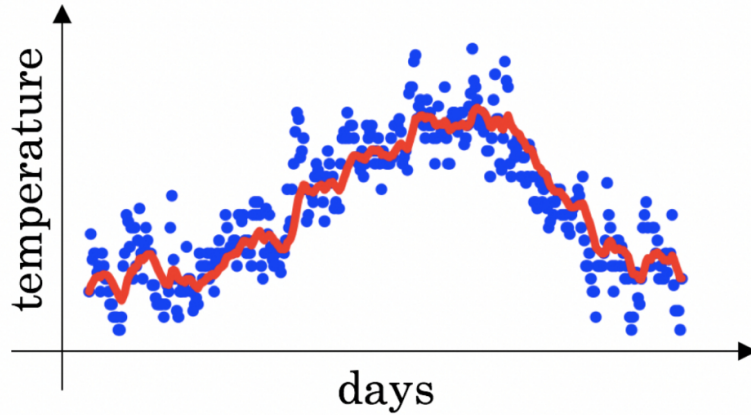
↗ Expand

✓ **Correct**

Correct. This is not a good learning rate decay since it is an increasing function of t .

7. You use an exponentially weighted average on the London temperature dataset. You use the following to track the temperature: $v_t = \beta v_{t-1} + (1 - \beta)\theta_t$. The red line below was computed using $\beta = 0.9$. What would happen to your red curve as you vary β ? (Check the two that apply)

0 / 1 point



☒ Decreasing β will shift the red line slightly to the right.

! This should not be selected
False.

☐ Increasing β will shift the red line slightly to the right.

☐ Decreasing β will create more oscillation within the red line.

☒ Increasing

β

will create more oscillations within the red line.

! This should not be selected

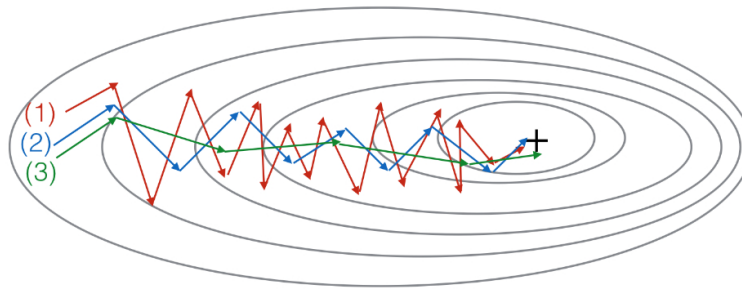
Expand

✗ Incorrect

You didn't select all the correct answers

8. Consider this figure:

1 / 1 point



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

☐ (1) is gradient descent with momentum (small β), (2) is gradient descent with momentum (small β), (3) is gradient descent

☐ (1) is gradient descent with momentum (small β), (2) is gradient descent, (3) is gradient descent with momentum (large β)

☒ (1) is gradient descent, (2) is gradient descent with momentum (small β), (3) is gradient descent with momentum (large β)

☐ (1) is gradient descent, (2) is gradient descent with momentum (large

β

(3) is gradient descent with momentum (small

Expand

✓ Correct

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 $\mathcal{J}(W^{[1]}, b^{[1]}, \dots, W^{[L]}, b^{[L]})$. Which of the following techniques could help find parameter values that attain a small value for \mathcal{J} ? (Check all that apply)

1 / 1 point

☒ Normalize the input data.

✓ Correct

Yes. In some cases, if the scale of the features is very different, normalizing the input data will speed up the training process.

☐ Try initializing the weight at zero.

☒ Try mini-batch gradient descent.

✓ Correct

Yes. Mini-batch gradient descent is faster than batch gradient descent.

☒ Try using Adam.

✓ Correct

Yes. Adam combines the advantages of other methods to accelerate the convergence of the gradient descent.

↗ Expand

✓ Correct

Great, you got all the right answers.

10. Which of the following are true about Adam?

1 / 1 point

- ☐ Adam automatically tunes the hyperparameter α .
- ☐ The most important hyperparameter on Adam is ϵ and should be carefully tuned.
- ☒ Adam combines the advantages of RMSProp and momentum.
- ☐ Adam can only be used with batch gradient descent and not with mini-batch gradient descent.

↗ Expand

✓ Correct

True. Precisely Adam combines the features of RMSProp and momentum that is why we use two-parameter β_1 and β_2 , besides ϵ .