Congratulations! You passed!

Grade received 100% **Latest Submission Grade** 100% **To pass** 80% or higher

Go to next item

1. Using the notation for mini-batch gradient descent. To what of the following does $a^{[2]\{4\}(3)}$ correspond? 1 / 1 point

- The activation of the fourth layer when the input is the second example of the third mini-batch.
- The activation of the second layer when the input is the fourth example of the third mini-batch.
- The activation of the third layer when the input is the fourth example of the second mini-batch.
- (selected) The activation of the second layer when the input is the third example of the fourth mini-batch.

Expand

Correct
Yes. In general $a^{[l]\{t\}(k)}$ denotes the activation of the layer l when the input is the example k from the mini-batch t.

2. Which of these statements about mini-batch gradient descent do you agree with? 1 / 1 point

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

Expand

Correct

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 / 1 point

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

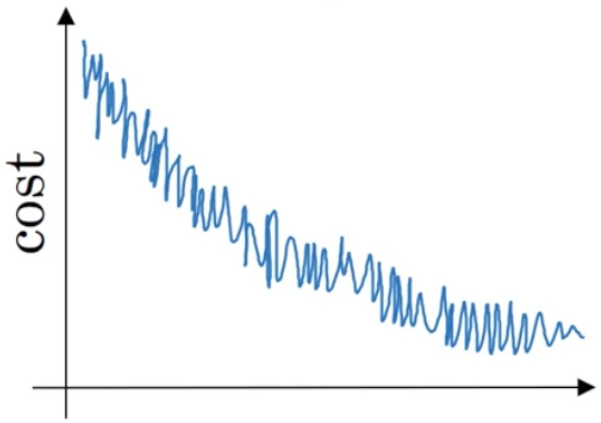
- [ ] If the mini-batch size is m, you end up with stochastic gradient descent, which is usually slower than mini-batch gradient descent.
- [x] If the mini-batch size is 1, you lose the benefits of vectorization across examples in the mini-batch.

✓ Correct

Expand

Correct
Great, you got all the right 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.
- The algorithm is on a local minimum thus the noisy behavior.
- 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.

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?

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

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**

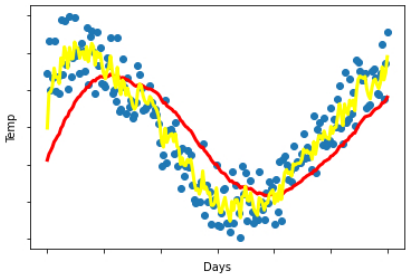
- $\alpha = e^{-0.01\,t}\alpha_0.$
- $\alpha = \frac{\alpha_0}{\sqrt{1+t}}.$

- $\alpha = 1.01^t \alpha_0$
- $\alpha = \frac{\alpha_0}{1+3t}$

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 yellow and red lines were computed using values $beta_1$ and $beta_2$ respectively. Which of the following are true? **1 / 1 point**

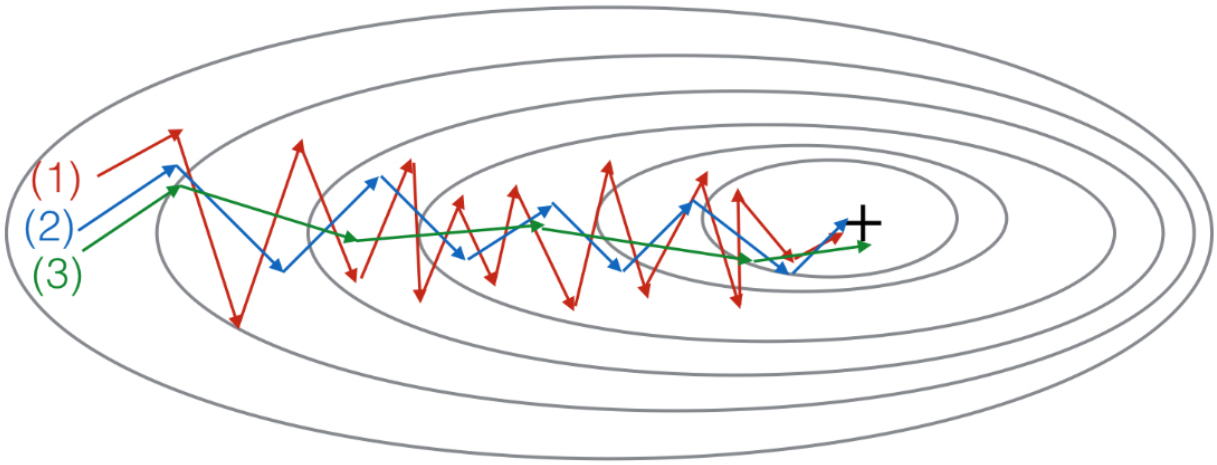


- $\beta_1 = \beta_2$.
- $\beta_1 < \beta_2$.
- $\beta_1 > \beta_2$.
- $\beta_1 = 0$, $\beta_2 > 0$.

Expand

Correct
Correct. $\beta_1 < \beta_2$ since the yellow curve is noisier.

8. Consider this figure: **1 / 1 point**



These plots were generated with gradient descent; with gradient descent with momentum (β = 0.5); and gradient descent with momentum (β = 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. (2) is gradient descent with momentum (large β). (3) is gradient descent with momentum (small β)

- (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)

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 $\mathcal{J}(W^{[1]}, b^{[1]}, ..., W^{[L]}, b^{[L]})$. 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**

- [ ] Try initializing the weight at zero.
- [x] Try using Adam.

  Correct
  Yes. Adam combines the advantages of other methods to accelerate the convergence of the gradient descent.
- [x] Try mini-batch gradient descent.

  Correct
  Yes. Mini-batch gradient descent is faster than batch gradient descent.
- [x] 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.

Expand

Correct
Great, you got all the right answers.

10. Which of the following are true about Adam? **1 / 1 point**

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

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.