Congratulations! You passed!

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

Go to next item

1. Which notation would you use to denote the 4th layer's activations when the input is the 7th example from the 3rd mini-batch?

1 / 1 point

- $a^{[7]\{3\}(4)}$
- $a^{[3]\{7\}(4)}$
- $a^{[4]\{3\}(7)}$ (selected)

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

- 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. (selected)

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.

1 / 1 point

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

  Correct

- [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

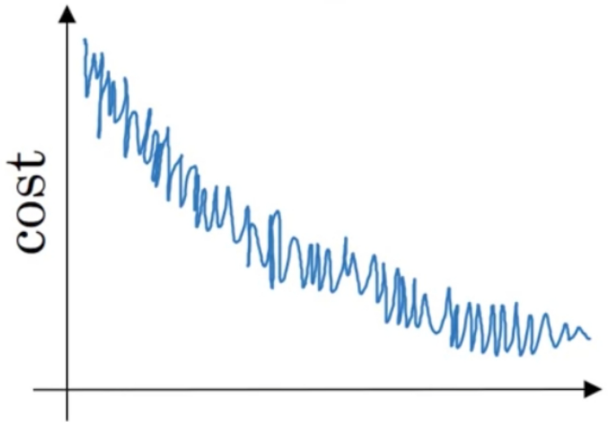
- [ ] If the mini-batch size is m, you end up with stochastic gradient descent, which is usually slower than mini-batch gradient descent.
- [ ] If the mini-batch size is 1, you end up having to process the entire training set before making any progress.

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?

- The algorithm is on a local minimum thus the noisy behavior.
- 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.

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 January are the same:

1 / 1 point

Jan 1st: $\theta_1 = 10^oC$

Jan 2nd: $\theta_2 = 10^oC$

(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 $\beta = 0.5$ 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 $v_2^{corrected}$ 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.)

- $v_2 = 7.5, v_2^{corrected} = 10$
- $v_2 = 10, v_2^{corrected} = 7.5$
- $v_2 = 7.5, v_2^{corrected} = 7.5$
- $v_2 = 10, v_2^{corrected} = 10$

Expand

Correct

6. Which of the following is true about learning rate decay?

1 / 1 point

- The intuition behind it is that for later epochs our parameters are closer to a minimum thus it is more convenient to take larger steps to accelerate the convergence.
- The intuition behind it is that for later epochs our parameters are closer to a minimum thus it is more convenient to take smaller

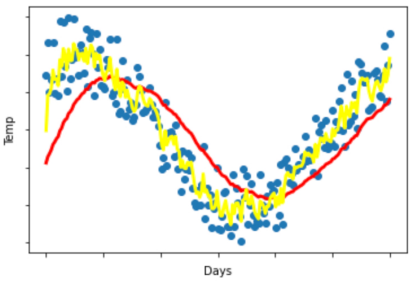
The intuition behind it is that for later epochs our parameters are closer to a minimum thus it is more convenient to take smaller steps to prevent large oscillations.

- We use it to increase the size of the steps taken in each mini-batch iteration.
- It helps to reduce the variance of a model.

Expand

Correct
Correct. Reducing the learning rate with time reduces the oscillation around a minimum.

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. Which of the following are true about gradient descent with momentum? **1 / 1 point**

- [x] Increasing the hyperparameter β smooths out the process of gradient descent.

Correct
Correct. Gradient descent with momentum makes use of moving averages, which smooths out the gradient descent process.

- [ ] It decreases the learning rate as the number of epochs increases.
- [x] It generates faster learning by reducing the oscillation of the gradient descent process.

Correct
Correct. The use of momentum makes each step of the gradient descent more efficient by reducing oscillations.

- [x] Gradient descent with momentum makes use of moving averages.

Correct
Correct. Gradient descent with momentum makes use of moving averages, which smooths out the gradient descent process.

Expand

Correct
Great, you got all the right answers.

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 all the weights to zero
- [x] Try using Adam

  Correct
- [x] Try tuning the learning rate α

  Correct
- [x] Try better random initialization for the weights

  Correct
- [x] Try mini-batch gradient descent

  Correct

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

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.