Congratulations! You passed!

Grade received 90% To pass 80% or higher

Go to next item

Optimization Algorithms

Latest Submission Grade 90%

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

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

Expand

Correct
Yes. In general $a^{[l](i)(t)}$ 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

- 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).
- 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.

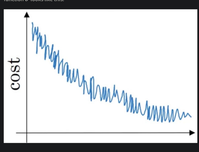
3. We usually choose a mini-batch size greater than 1 and less than m, because that way we make use of vectorization but not fall into the slower case of batch gradient descent. 1 / 1 point

- True
- False

Expand

Correct
Correct. Precisely by choosing a batch size greater than one we can use vectorization; but we choose a value less than m so we won't end up using batch gradient descent.

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?

- In mini-batch gradient descent we calculate $J(\mathcal{U}^{\{t\}}, b^{\{t\}})$ 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.
- 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.

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: $\theta_1 = 30^\circ\ \mathrm{C}$

March 2nd: $\theta_2 = 15^\circ\ \mathrm{C}$

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 $v_2^{corrected}$ is the value you compute with bias correction. What are these values?

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

Expand

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

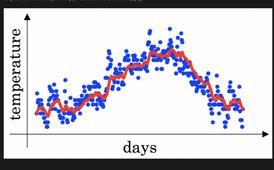
6. Which of the following is true about learning rate decay? 1 / 1 point

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

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 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) 1 / 1 point



- [ ] Decreasing β will shift the red line slightly to the right.
- [x] Increasing β will shift the red line slightly to the right.

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

- [x] Decreasing β will create more oscillation within the red line.

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

- [ ] Increasing β will create more oscillations within the red line.

Expand

Correct
Great, you got all the right answers.

8. Which of the following are true about gradient descent with momentum? 0 / 1 point

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

Incorrect
You didn't select all the correct 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 $\mathcal{J}(W^{[1]}, b^{[1]}, \ldots, W^{[l]}, b^{[l]})$. 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

- [x] Try tuning the learning rate α

  Correct

- [x] Try using Adam

  Correct

- [x] Try mini-batch gradient descent

  Correct

- [ ] Try initializing all the weights to zero
- [x] Try better random initialization for the weights

  Correct

Expand

Correct
Great, you got all the right answers.

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 / 1 point

- True
- False

Expand

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