축하합니다! 통과하셨습니다!

받은 학점 80% 최신 제출물 학점 80% 통과 점수: 80% 이상

다음 항목으로 이동

1. Which notation would you use to denote the 3rd layer's activations when the input is the 7th example from the 8th minibatch? 0 / 1점

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

더 보기

틀립니다

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

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

더 보기

맞습니다

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 / 1점

- [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

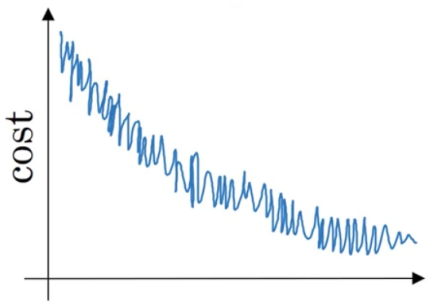
- [ ] If the mini-batch size is m, you end up with stochastic gradient descent, which is usually slower than mini-batch gradient descent.
- [ ] If the mini-batch size is 1, you end up having to process the entire training set before making any progress.
- [x] If the mini-batch size is 1, you lose the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

  Correct

더 보기

맞습니다
Great, you got all the right answers.

4. Suppose your learning algorithm's cost J, plotted as a function of the number of iterations, looks like this: 1 / 1점



Which of the following do you agree with?

- Whether you're using batch gradient descent or mini-batch gradient descent, something is wrong.
- Whether you're using batch gradient descent or mini-batch gradient descent, this looks acceptable.

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

더 보기

맞습니다

5. Suppose the temperature in Casablanca over the first two days of January are the same: 1 / 1점

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.)

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

더 보기

맞습니다

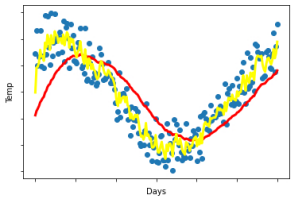
6. Which of the following is true about learning rate decay? 1 / 1점

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

더 보기

맞습니다
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 / 1점

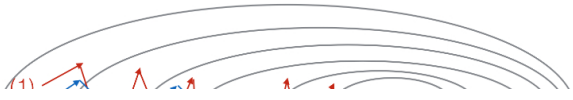


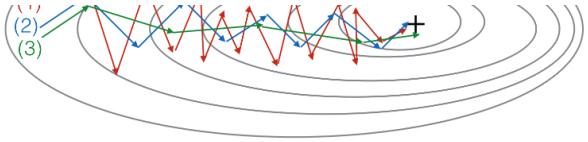
- $\beta_1 > \beta_2$.
- $\beta_1 < \beta_2$.
- $\beta_1 = 0$, $\beta_2 > 0$.
- $\beta_1 = \beta_2$.

더 보기

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

8. Consider this figure: 1 / 1점





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. (2) is gradient descent with momentum (large β). (3) is gradient descent with momentum (small 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 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)

더 보기

맞습니다

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 / 1점

- [x] Try using Adam
  - Correct
- [x] Try mini-batch gradient descent
  - Correct
- [ ] Try initializing all the weights to zero
- [x] Try tuning the learning rate
  - Correct
- [x] Try better random initialization for the weights
  - Correct

더 보기

맞습니다
Great, you got all the right answers.

10. Which of the following are true about Adam? 0 / 1점

- Adam automatically tunes the hyperparameter α.
- 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.

더 보기

틀립니다
False. The mechanics of Adam works the same with the complete batch or with mini-batches.