Your grade: 100%

Your latest: **100%** • Your highest: **100%** • To pass you need at least 80%. We keep your highest score.

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

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.
- (x) When the mini-batch size is the same as the training size, mini-batch gradient descent is equivalent to batch gradient descent.

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 m, you end up with batch gradient descent, which has to process the whole training set before making progress.

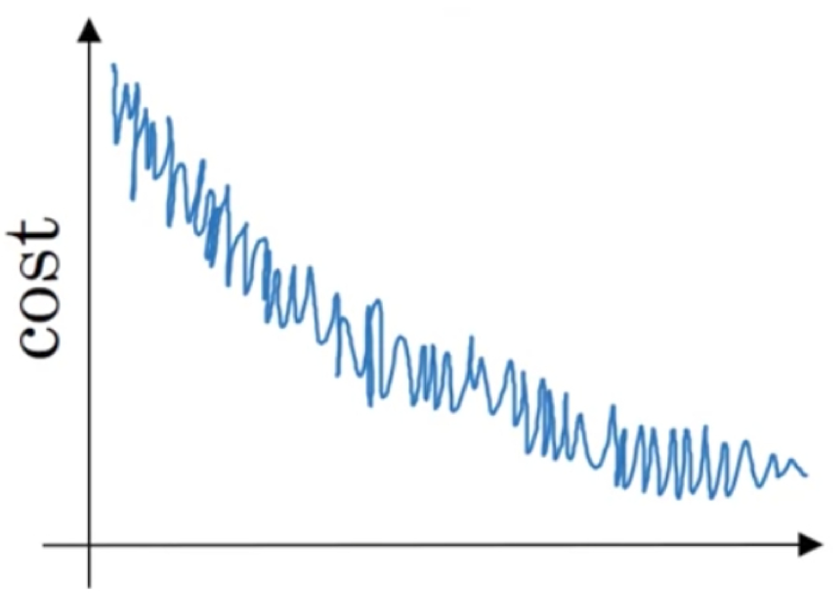
Correct

- [ ] If the mini-batch size is 1, you end up having to process the entire training set before making any progress.
- [x] If the mini-batch size is 1, you lose the benefits of vectorization across examples in the mini-batch.

Correct

- [ ] If the mini-batch size is m, you end up with stochastic gradient descent, which is usually slower than mini-batch gradient descent.

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



Which of the following do you agree with?

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

Correct

5. Suppose the temperature in Casablanca over the first two days of March are the following: **1 / 1 point**

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

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

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

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

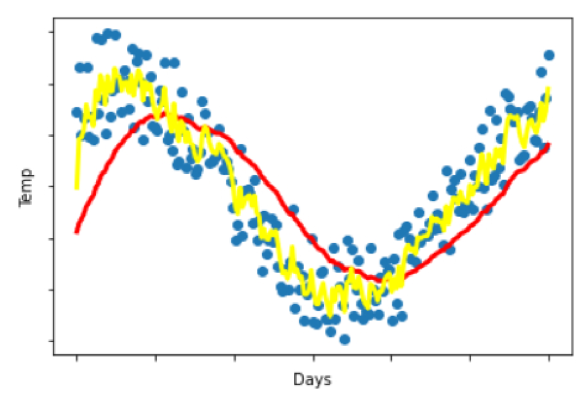
Correct
Correct. $v_2 = \beta v_{t-1} + (1-\beta)\,\theta_t$ thus $v_1 = 5, v_2 = 15$. Using the bias correction $\frac{v_t}{1-\beta^t}$ we get $\frac{15}{1-(0.5)^2} = 20.$

6. Which of these is NOT a good learning rate decay scheme? Here, t is the epoch number. **1 / 1 point**

- (x) $\alpha = e^t \alpha_0$
- ( ) $\alpha = 0.95^t \alpha_0$
- ( ) $\alpha = \frac{1}{1+2*t}\alpha_0$
- ( ) $\alpha = \frac{1}{\sqrt{t}}\alpha_0$

Correct

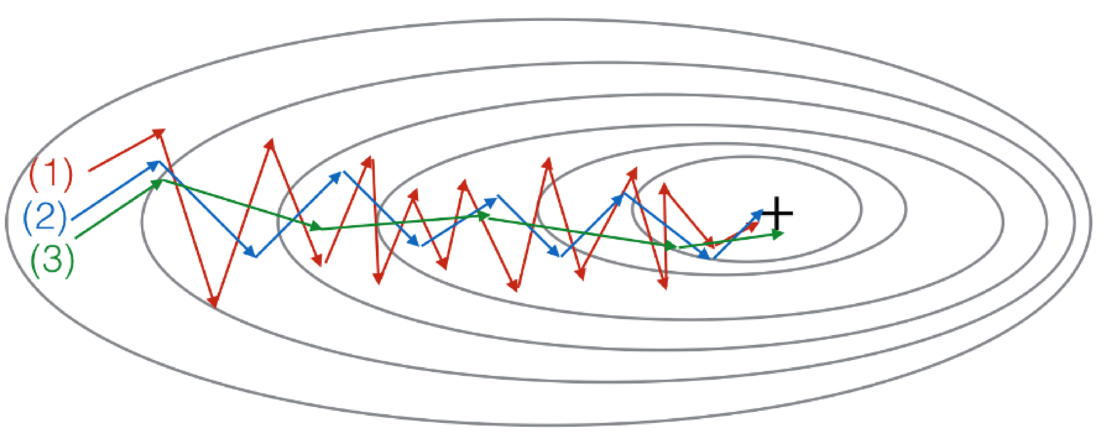
7. You use an exponentially weighted average on the London temperature dataset. You use the following to track the temperature: $v_t = \beta v_{t-1} + (1-\beta)\theta_t$. The yellow and red lines were computed using values $beta_1$ and $beta_2$ respectively. Which of the following are true? **1 / 1 point**



- ( ) $\beta_1 = \beta_2.$
- (x) $\beta_1 < \beta_2.$
- ( ) $\beta_1 > \beta_2.$
- ( ) $\beta_1 = 0, \beta_2 > 0.$

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

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**

- [x] Try mini-batch gradient descent

Correct

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

Correct

- [x] Try tuning the learning rate α

Correct

- [x] Try using Adam

Correct

10. Which of the following statements about Adam is ***False***? **1 / 1 point**

- ( ) Adam combines the advantages of RMSProp and momentum
- ( ) We usually use "default" values for the hyperparameters β_1, β_2 and ε in Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999, \varepsilon = 10^{-8}$)
- (x) Adam should be used with batch gradient computations, not with mini-batches.
- ( ) The learning rate hyperparameter α in Adam usually needs to be tuned.

Correct