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

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

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. Suppose you don't face any memory-related problems. Which of the following make more use of vectorization. 1 / 1 point

- Stochastic Gradient Descent, Batch Gradient Descent, and Mini-Batch Gradient Descent all make equal use of vectorization.
- Stochastic Gradient Descent
- Batch Gradient Descent ✓
- Mini-Batch Gradient Descent with mini-batch size $m/2$.

Correct
Yes. If no memory problem is faced, batch gradient descent processes all of the training set in one pass, maximizing the use of vectorization.

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

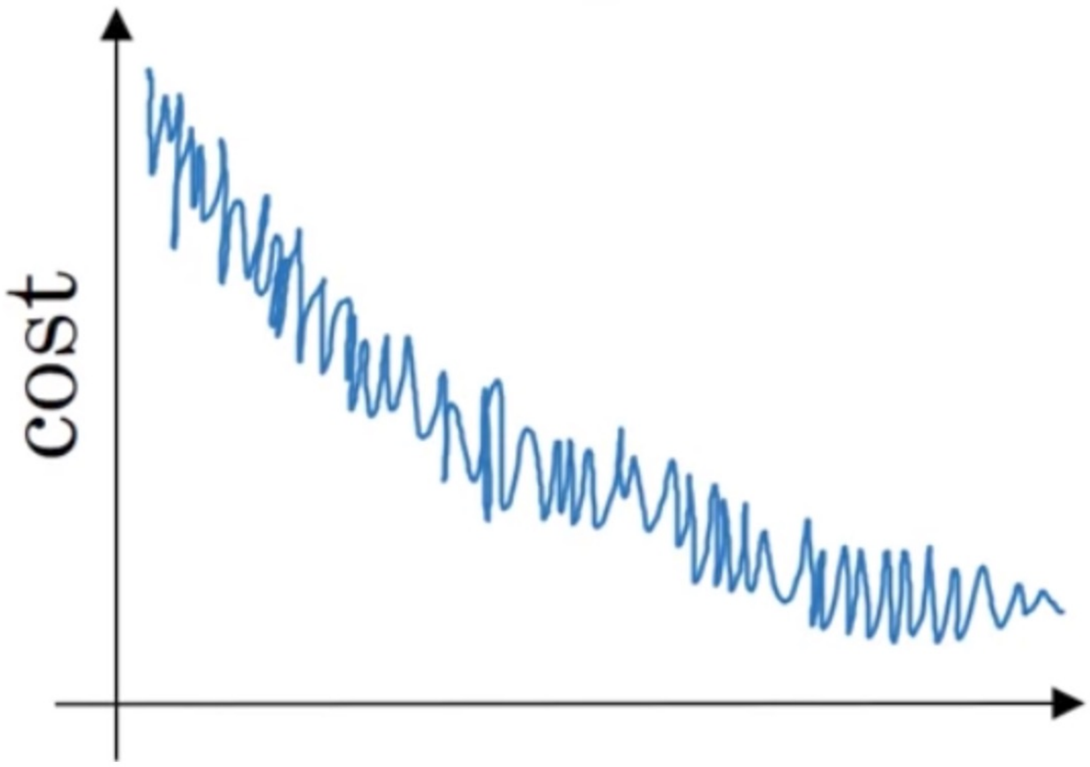
- [ ] If the mini-batch size is m, you end up with stochastic gradient descent, which is usually slower than mini-batch gradient descent.
- [x] If the mini-batch size is m, you end up with batch gradient descent, which has to process the whole training set before making progress.

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

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, 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.

- ◯ 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.

Correct

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

March 1st: $\theta_1 = 30^\circ \text{ C}$

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

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?

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

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 $\frac{15}{1-(0.5)^2} = 20.$

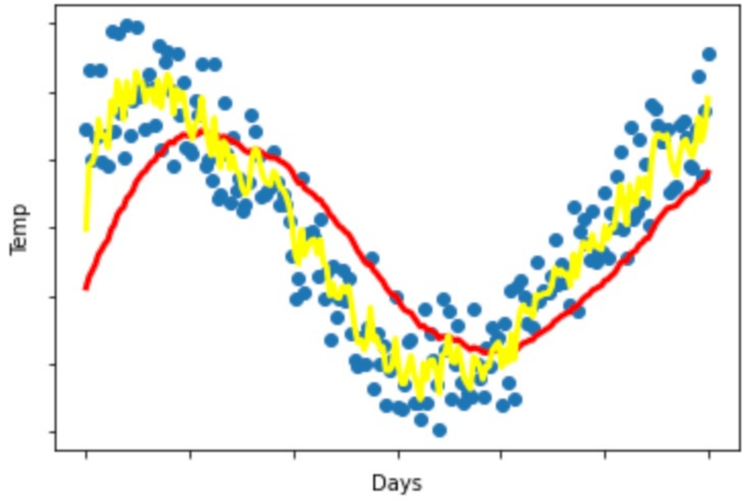
1 / 1 point

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

- (●) $\alpha = 1.01^t \, \alpha_0$
- ( ) $\alpha = \frac{\alpha_0}{1+3\,t}$
- ( ) $\alpha = e^{-0.01\,t} \alpha_0$.
- ( ) $\alpha = \frac{\alpha_0}{\sqrt{1+t}}$.

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

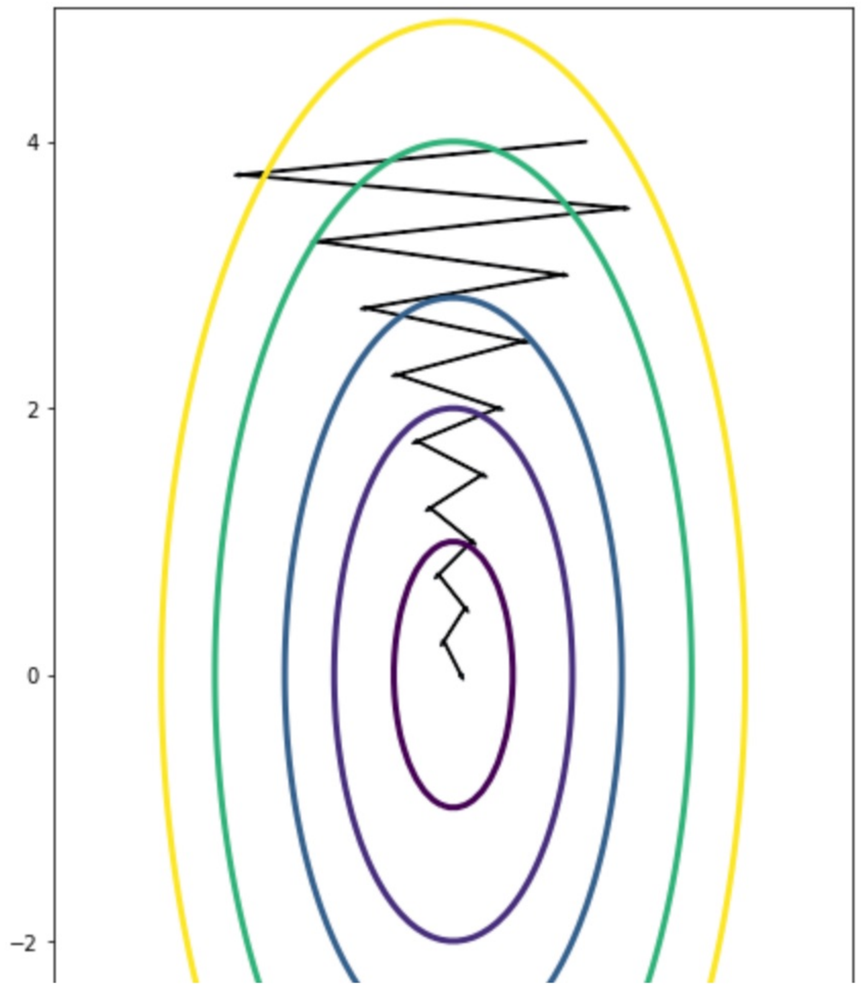


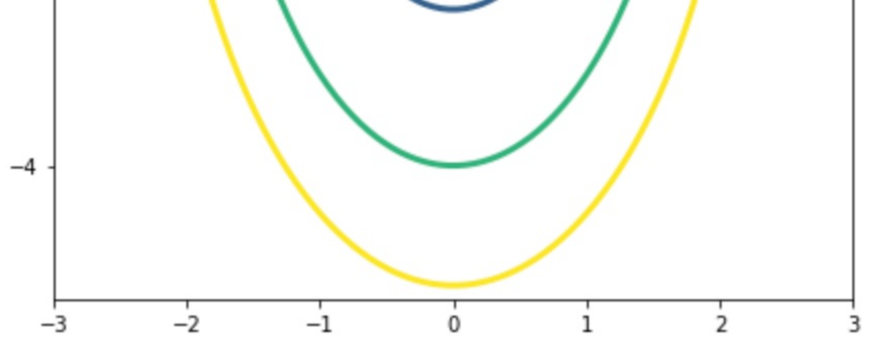
- ( ) $\beta_1 > \beta_2$.
- ( ) $\beta_1 = \beta_2$.

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

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

8. Consider the figure: 1 / 1 point





Suppose this plot was generated with gradient descent with momentum $\beta = 0.01$. What happens if we increase the value of β to 0.1?

- ○ The gradient descent process starts moving more in the horizontal direction and less in the vertical.
- ○ The gradient descent process moves more in the horizontal and the vertical axis.
- ○ The gradient descent process starts oscillating in the vertical direction.
- ◉ The gradient descent process moves less in the horizontal direction and more in the vertical direction.

Correct
Yes. The use of a greater value of β causes a more efficient process thus reducing the oscillation in the horizontal direction and moving the steps more in the vertical direction.

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**

- ☑ 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.

- [ ] Try initializing the weight at zero.
- [x] Try mini-batch gradient descent.

Correct
Yes. Mini-batch gradient descent is faster than batch gradient descent.

- [x] Try using Adam.

Correct
Yes. Adam combines the advantages of other methods to accelerate the convergence of the gradient descent.

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

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