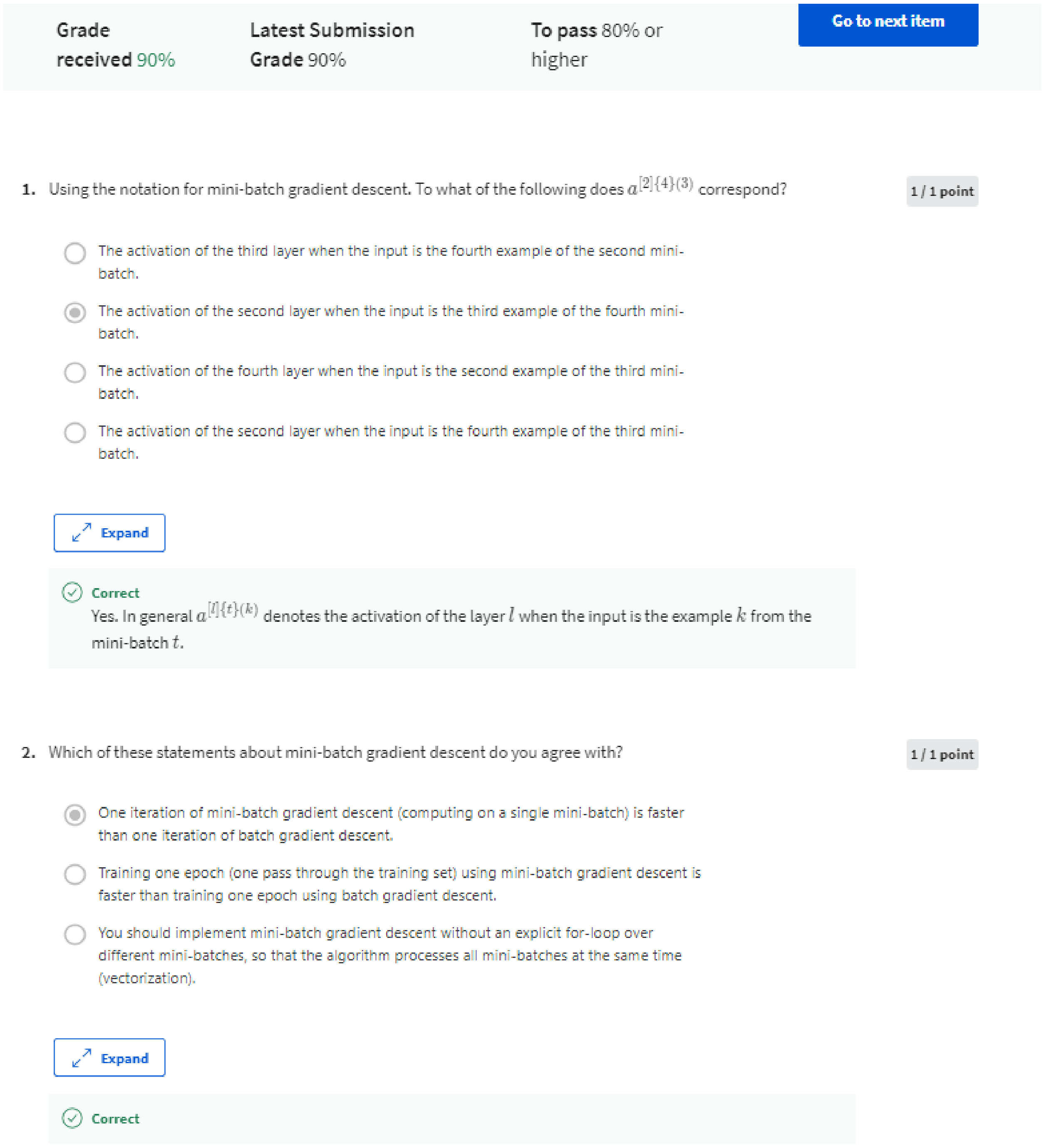
Quiz



Grade received 90%

Latest Submission Grade 90%

To pass 80% or higher

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 third layer when the input is the fourth example of the second mini-batch.

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

○ 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.

✓ 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

● One iteration of mini-batch gradient descent (computing on a single mini-batch) is faster than one iteration of batch gradient descent.

○ 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).

✓ Correct

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 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.

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

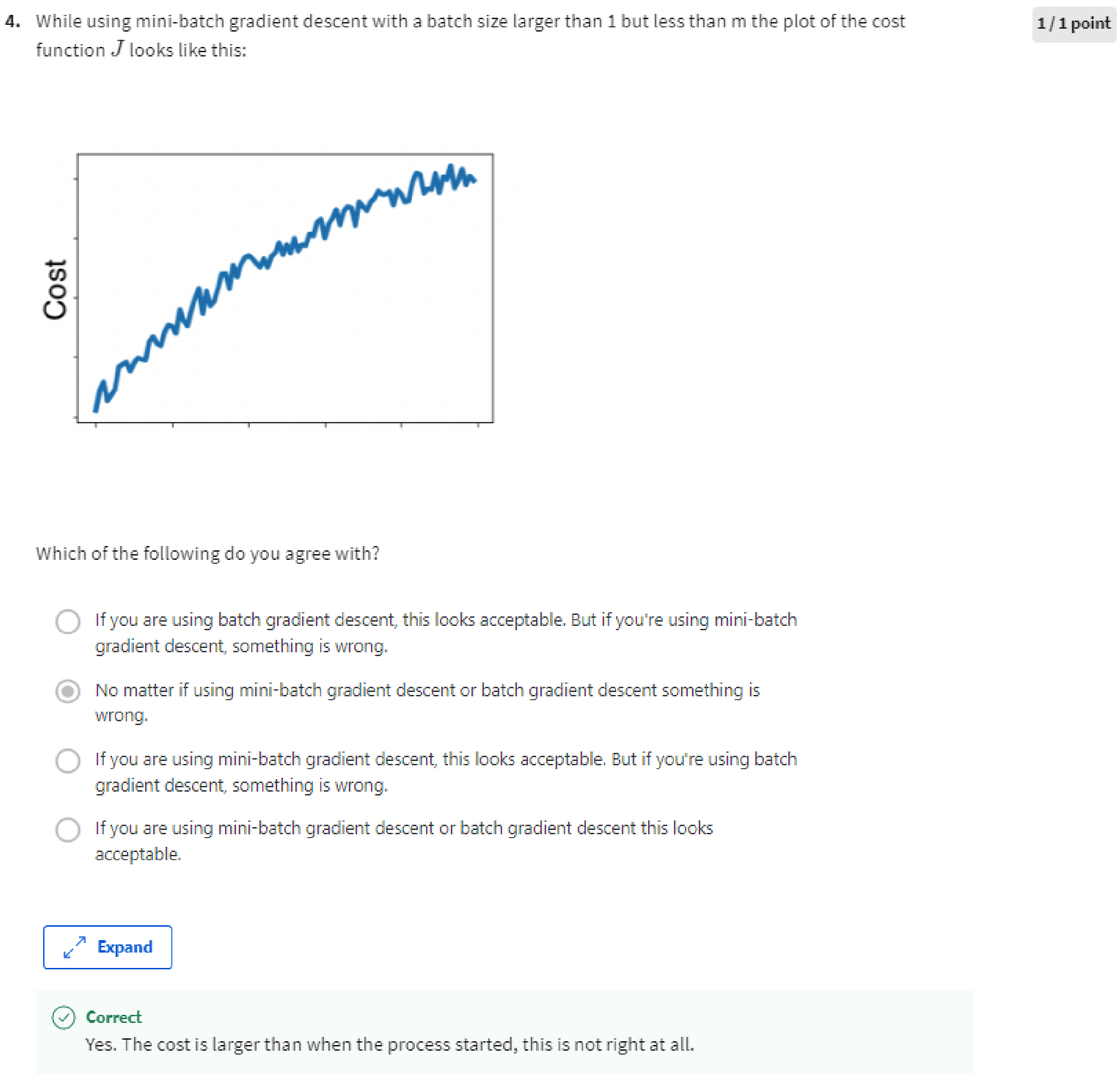
✓ Correct

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

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

Which of the following do you agree with?

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

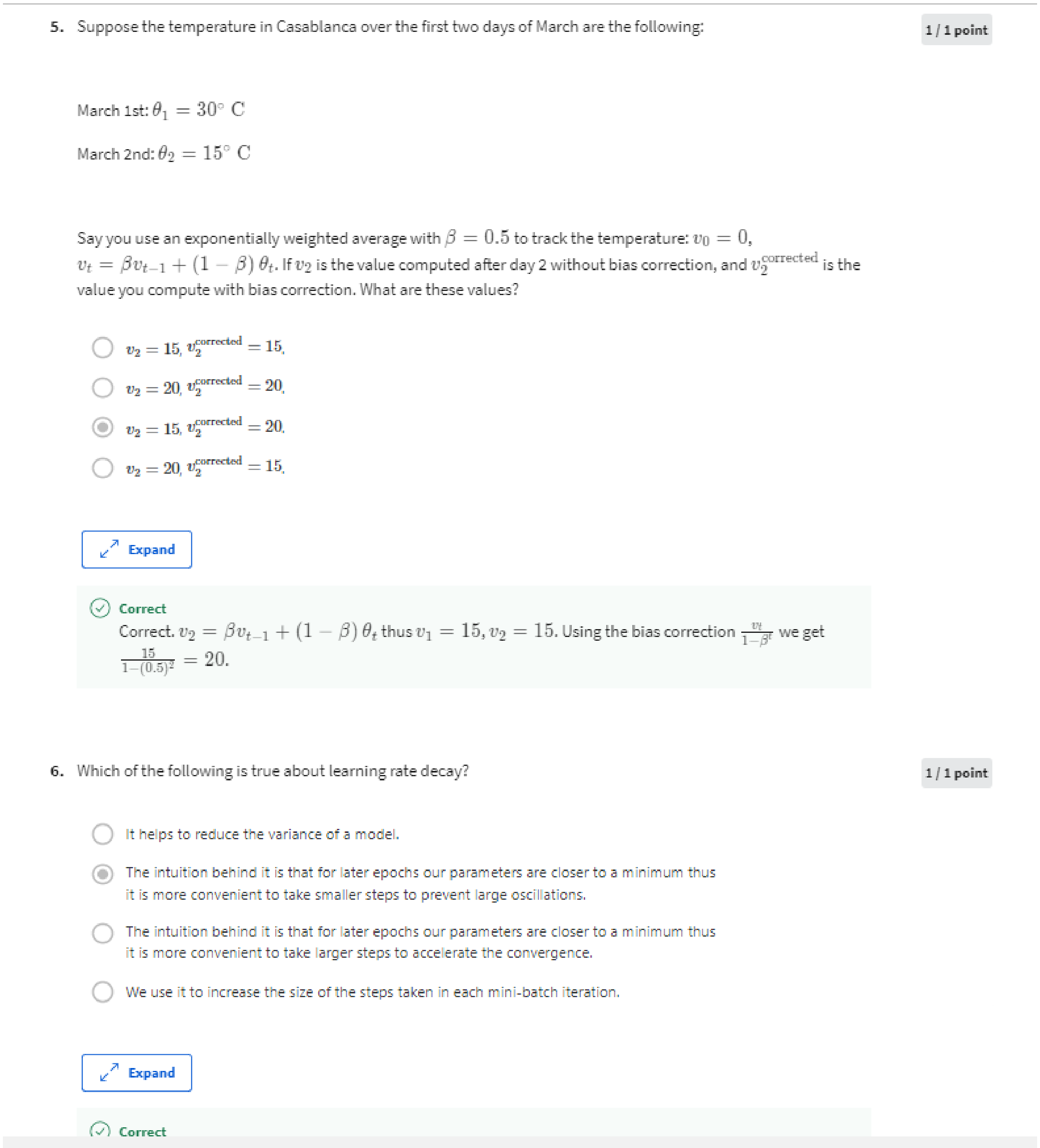
● No matter if using mini-batch gradient descent or batch gradient descent something is wrong.

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

○ If you are using mini-batch gradient descent or batch gradient descent this looks acceptable.

✓ **Correct**

Yes. The cost is larger than when the process started, this is not right at all.



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?

○ $v_2 = 15$, $v_2^{\text{corrected}} = 15$.

○ $v_2 = 20$, $v_2^{\text{corrected}} = 20$.

● $v_2 = 15$, $v_2^{\text{corrected}} = 20$.

○ $v_2 = 20$, $v_2^{\text{corrected}} = 15$.

✓ **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 $\frac{15}{1-(0.5)^2} = 20.$

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

1 / 1 point

○ It helps to reduce the variance of a model.

● 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.

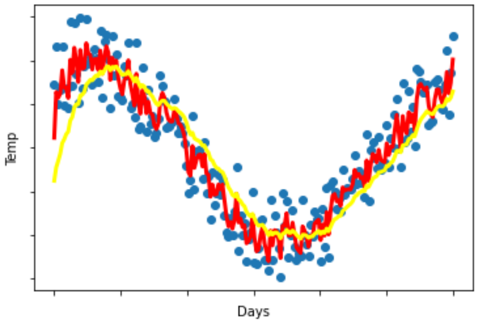
○ The intuition behind it is that for later epochs our parameters are closer to a minimum thus it is more convenient to take larger steps to accelerate the convergence.

○ We use it to increase the size of the steps taken in each mini-batch iteration.

✓ **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 β_1 and β_2 respectively. Which of the following are true?

1 / 1 point



○ $\beta_1 = \beta_2$.

● $\beta_1 > \beta_2$.

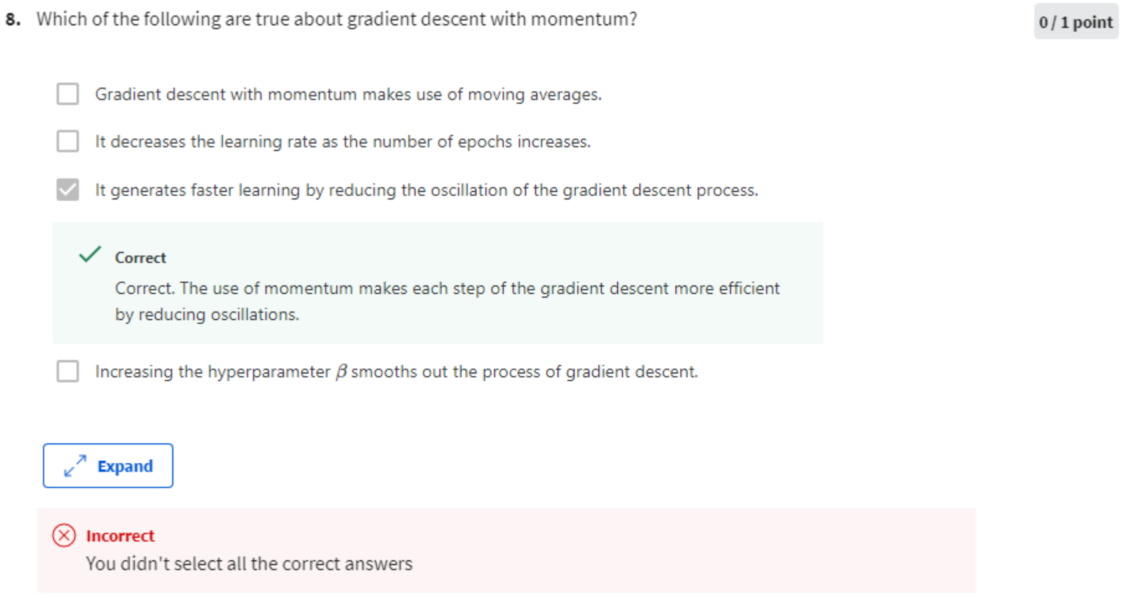
○ $\beta_1 = 0, \beta_2 > 0$.

○ $\beta_1 < \beta_2$.

Expand

✓ **Correct**

Correct. $\beta_1 > \beta_2$ since the red curve is noisier.



8. Which of the following are true about gradient descent with momentum?

0 / 1 point

□ Gradient descent with momentum makes use of moving averages.

□ It decreases the learning rate as the number of epochs increases.

☑ It generates faster learning by reducing the oscillation of the gradient descent process.

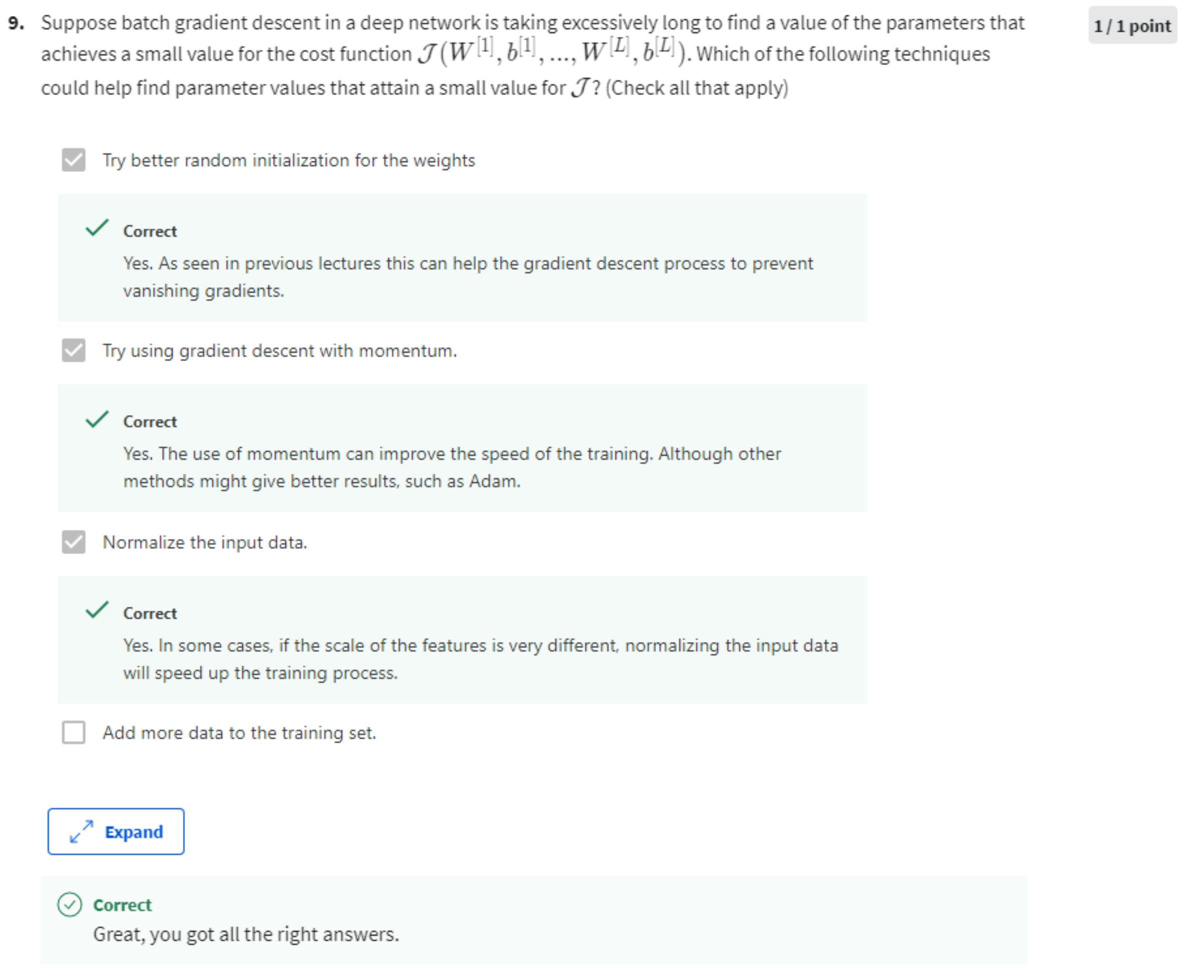
✓ **Correct**

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

□ Increasing the hyperparameter β smooths out the process of gradient descent.

⊗ **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 $\mathcal{J}(W^{[1]}, b^{[1]}, ..., W^{[L]}, b^{[L]})$. 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 better random initialization for the weights

✓ **Correct**

Yes. As seen in previous lectures this can help the gradient descent process to prevent vanishing gradients.

☑ Try using gradient descent with momentum.

✓ **Correct**

Yes. The use of momentum can improve the speed of the training. Although other methods might give better results, such as Adam.

☑ 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.

□ Add more data to the training set.

✓ **Correct**

Great, you got all the right answers.