Improving Deep Neural Networks: Hyperparameter Tuning, Regularization and Optimization

Week 2: Optimization Algorithms

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?

Answer: $a^{[4]3(7)}$.

Comment: 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?

Answer: When the mini batch size is the same as the training size, mini batch gradient descent is equivalent to batch gradient descent.

Comment: Batch gradient descent uses all the example sat 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 Version 1: We usually choose a mini batch size greater than 1 and less than m, because that way we make use vectorization but not fall into the slower case of batch gradient descent.

Answer: True.

Comment: 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.

3 Version 2: Why is the best mini batch size usually not 1 and not m, but instead something in between?

Answer: 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; If the mini batch size is 1, you lose the benefits of vectorization across examples in the mini batch.

4 Version 1: 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 (increase): Which of the following do you agree with?

Answer: No matter if using mini batch gradient descent or batch gradient descent something is wrong.

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

4 Version 2: Suppose your learning algorithm's cost J, plotted as a function of the number of iterations, looks like this (decrease): Which of the following do you agree with?

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

5 Suppose the temperature in Casablanca over the first two days of March are the following:

- March 1st: $\theta_1 = 30°C$
- March 2nd: $\theta_2 = 15°C$

Say you use an exponentially weighted average with $\beta = 0.5$ to track the temperature: $\nu_0 = 0$, $\nu_t = \beta\nu_{t-1} + (1-\beta)\theta_t$. If ν_2 is the value computed after day 2 without bias correction, and $\nu_2^{corrected}$ is the value you compute with bias correction. What are these values?

Answer: $\nu_2 = 15, \nu_2^{corrected} = 20$.

Comment: $\nu_2 = \beta_{\nu_{t-1}} + (1-\beta)\theta_t$ thus $\nu_1 = 15, \nu_2 = 15$. Using the bias correction $\nu_t/(1-\beta^t)$ we get $15/(1-(0.5)^2) = 20$.

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

Answer: $\alpha = e^t\alpha_0$.

7 Version 1: You use an exponentially weighted average on the London temperature dataset. You use the following to track the temperature: $\nu_t = \beta\nu_{t-1} + (1-\beta)\theta_t$. The yellow and red lines were computed using values β_1 and β_2, respectively. Which of the following are true?

Answer: $\beta_1 > \beta_2$.

Comment: The red curve is noisier.

7 Version 2: You use an exponentially weighted average on the London temperature dataset. You use the following to track the temperature: $\nu_t = \beta\nu_{t-1} + (1-\beta)\theta_t$. 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)

Answer: Increasing β will shift the red line slightly to the right; Decreasing β will create more oscillation within the red line.

Comment: Remember that the red line corresponds to $= 0.9$. In the lecture we had a green line $\beta = 0.98$ that is slightly shifted to the right; A yellow line that had a lot of oscillations.

8 Version 1: Consider the figure: (vertical). 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?

Answer: The gradient descent process moves less in the horizontal direction and more in the vertical direction.

Comment: 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.

8 Version 2: Consider the figure: (horizontal). These plots were generated with gradient descent; with gradient descent with momentum ($\beta = 0.5$) and gradient descent with momentum ($\beta = 0.9$). Which curve corresponds to which algorithm?

Answer: (1) is gradient descent. (2) is gradient descent with momentum (small β). (3) is gradient descent with momentum (large β).

Comment: See lecture notes.

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 $J(W^{[1]}, b^{[1]}, \ldots, b^{[L]})$. Which of the following techniques could help find parameter values that attain a small value of J? (Check all that apply)

Answer: Try better random initialization for the weights; Try using gradient descent with momentum; Normalize the input data.

Comment: As seen in previous lectures this can help the gradient descent process to prevent vanishing gradients; The use of momentum can improve the speed of the training. Although other methods might give better results, such as Adam; In some cases, if the scale of the features is very different, normalizing the input data will speed up the training process.

10 Version 1: Which of the following are true about Adam?

Answer: Adam combines the advantages of RMSProp and momentum.

Comment: . See lecture notes. Precisely Adam combines the features of RMSProp and momentum that is why we use two parameter β_1 and β_2, besides ϵ.

10 Version 2: Which of the following statements about Adam is False?

Answer: Adam should be used with batch gradient computations, not with mini batches.