|  |
| --- |
| 1.  Question 1  If you have 10,000,000 examples, how would you split the train/dev/test set?  1 point    60% train . 20% dev . 20% test    98% train . 1% dev . 1% test    33% train . 33% dev . 33% test  2.  Question 2  The dev and test set should:  1 point    Come from the same distribution    Come from different distributions    Be identical to each other (same (x,y) pairs)    **Have the same number of examples**  3.  Question 3  If your Neural Network model seems to have high bias, what of the following would be promising things to try? (Check all that apply.)  1 point    Get more training data    Add regularization    Make the Neural Network deeper    Increase the number of units in each hidden layer    Get more test data  4.  Question 4  You are working on an automated check-out kiosk for a supermarket, and are building a classifier for apples, bananas and oranges. Suppose your classifier obtains a training set error of 0.5%, and a dev set error of 7%. Which of the following are promising things to try to improve your classifier? (Check all that apply.)  1 point    Increase the regularization parameter lambda    Decrease the regularization parameter lambda    Get more training data    Use a bigger neural network  5.  Question 5  What is weight decay?  1 point    Gradual corruption of the weights in the neural network if it is trained on noisy data.    A regularization technique (such as L2 regularization) that results in gradient descent shrinking the weights on every iteration.    The process of gradually decreasing the learning rate during training.    A technique to avoid vanishing gradient by imposing a ceiling on the values of the weights.  6.  Question 6  What happens when you increase the regularization hyperparameter lambda?  1 point    Weights are pushed toward becoming smaller (closer to 0)    Weights are pushed toward becoming bigger (further from 0)    Doubling lambda should roughly result in doubling the weights    Gradient descent taking bigger steps with each iteration (proportional to lambda)  7.  Question 7  With the inverted dropout technique, at test time:  1 point    You do not apply dropout (do not randomly eliminate units) and do not keep the 1/keep\_prob factor in the calculations used in training    You apply dropout (randomly eliminating units) but keep the 1/keep\_prob factor in the calculations used in training.    You do not apply dropout (do not randomly eliminate units), but keep the 1/keep\_prob factor in the calculations used in training.    You apply dropout (randomly eliminating units) and do not keep the 1/keep\_prob factor in the calculations used in training  8.  Question 8  Increasing the parameter keep\_prob from (say) 0.5 to 0.6 will likely cause the following: (Check the two that apply)  1 point    Increasing the regularization effect    Reducing the regularization effect    Causing the neural network to end up with a higher training set error    Causing the neural network to end up with a lower training set error  9.  Question 9  Which of these techniques are useful for reducing variance (reducing overfitting)? (Check all that apply.)  1 point    Exploding gradient    Dropout    Data augmentation    Vanishing gradient    Xavier initialization    L2 regularization    Gradient Checking  10.  Question 10  Why do we normalize the inputs x*x*?  1 point    It makes the parameter initialization faster    It makes the cost function faster to optimize    It makes it easier to visualize the data    Normalization is another word for regularization--It helps to reduce variance |
| **Hyperparameter tuning, Batch Normalization, Programming Frameworks**  **TOTAL POINTS 10**  1.  Question 1  If searching among a large number of hyperparameters, you should try values in a grid rather than random values, so that you can carry out the search more systematically and not rely on chance. True or False?  1 point    True    False  2.  Question 2  Every hyperparameter, if set poorly, can have a huge negative impact on training, and so all hyperparameters are about equally important to tune well. True or False?  1 point    True    False  3.  Question 3  During hyperparameter search, whether you try to babysit one model (“Panda” strategy) or train a lot of models in parallel (“Caviar”) is largely determined by:  1 point    Whether you use batch or mini-batch optimization    The presence of local minima (and saddle points) in your neural network    The amount of computational power you can access    The number of hyperparameters you have to tune  4.  Question 4  If you think \beta*β* (hyperparameter for momentum) is between on 0.9 and 0.99, which of the following is the recommended way to sample a value for beta?  1 point    2  beta = r\*0.09 + 0.9        2  1  beta = 1-10\*\*(- r - 1)  r = np.random.rand()        2  1  beta = 1-10\*\*(- r + 1)  r = np.random.rand()        1  2  r = np.random.rand()  beta = r\*0.9 + 0.09      5.  Question 5  Finding good hyperparameter values is very time-consuming. So typically you should do it once at the start of the project, and try to find very good hyperparameters so that you don’t ever have to revisit tuning them again. True or false?  1 point    True    False  6.  Question 6  In batch normalization as presented in the videos, if you apply it on the l*l*th layer of your neural network, what are you normalizing?  1 point    b^{[l]}*b*[*l*]    a^{[l]}*a*[*l*]    W^{[l]}*W*[*l*]    z^{[l]}*z*[*l*]  7.  Question 7  In the normalization formula z\_{norm}^{(i)} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \varepsilon}}*znorm*(*i*)​=*σ*2+*ε*​*z*(*i*)−*μ*​, why do we use epsilon?  1 point    To have a more accurate normalization    In case \mu*μ* is too small    To speed up convergence    To avoid division by zero  8.  Question 8  Which of the following statements about \gamma*γ* and \beta*β* in Batch Norm are true?  1 point    There is one global value of \gamma \in \Re*γ*∈ℜ and one global value of \beta \in \Re*β*∈ℜ for each layer, and applies to all the hidden units in that layer.    They set the mean and variance of the linear variable z^[l]*z*[*l*] of a given layer.    The optimal values are \gamma = \sqrt{\sigma^2 + \varepsilon}*γ*=*σ*2+*ε*​, and \beta = \mu*β*=*μ*.    They can be learned using Adam, Gradient descent with momentum, or RMSprop, not just with gradient descent.    \beta*β* and \gamma*γ* are hyperparameters of the algorithm, which we tune via random sampling.  9.  Question 9  After training a neural network with Batch Norm, at test time, to evaluate the neural network on a new example you should:  1 point    If you implemented Batch Norm on mini-batches of (say) 256 examples, then to evaluate on one test example, duplicate that example 256 times so that you’re working with a mini-batch the same size as during training.    Perform the needed normalizations, use \mu*μ* and \sigma^2*σ*2 estimated using an exponentially weighted average across mini-batches seen during training.    Use the most recent mini-batch’s value of \mu*μ* and \sigma^2*σ*2 to perform the needed normalizations.    Skip the step where you normalize using \mu*μ* and \sigma^2*σ*2 since a single test example cannot be normalized.  10.  Question 10  Which of these statements about deep learning programming frameworks are true? (Check all that apply)  1 point    Deep learning programming frameworks require cloud-based machines to run.    A programming framework allows you to code up deep learning algorithms with typically fewer lines of code than a lower-level language such as Python.    Even if a project is currently open source, good governance of the project helps ensure that the it remains open even in the long term, rather than become closed or modified to benefit only one company. |