# Hyperparameter tuning, Batch Normalization, Programming Frameworks

Quiz, 10 questions

1 point

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?

- True
- False

---

1 point

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?

- True
- False

---

1 point

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:

- 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

---

1 point

4.

If you think $\beta$ (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?

```
r = np.random.rand()
beta = r*0.09 + 0.9
```

```
r = np.random.rand()
beta = 1-10**(- r - 1)
```

```
r = np.random.rand()
beta = 1-10**(- r + 1)
```

```
r = np.random.rand()
beta = r*0.9 + 0.09
```

---

1 point

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?

- True
- False

---

1 point

6.
In batch normalization as presented in the videos, if you apply it on the $l$th layer of your neural network, what are you normalizing?

- $W^{[l]}$
- $z^{[l]}$
- $a^{[l]}$
- $b^{[l]}$

---

1 point

7.

# Hyperparameter tuning, Batch Normalization, Programming Frameworks

Quiz, 10 questions

In the normalization formula $z^{(i)}_{norm} = \frac{z^{(i)}-\mu}{\sqrt{\sigma^2+\varepsilon}}$, why do we use epsilon?

- In case $\mu$ is too small
- To speed up convergence
- To have a more accurate normalization
- To avoid division by zero

1 point

8.

Which of the following statements about $\gamma$ and $\beta$ in Batch Norm are true?

- [x] They can be learned using Adam, Gradient descent with momentum, or RMSprop, not just with gradient descent.
- [ ] The optimal values are $\gamma = \sqrt{\sigma^2 + \varepsilon}$, and $\beta = \mu$.
- [x] They set the mean and variance of the linear variable $z^{[l]}$ of a given layer.
- [ ] There is one global value of $\gamma \in \mathfrak{R}$ and one global value of $\beta \in \mathfrak{R}$ for each layer, and applies to all the hidden units in that layer.
- [ ] $\beta$ and $\gamma$ are hyperparameters of the algorithm, which we tune via random sampling.

1 point

9.

After training a neural network with Batch Norm, at test time, to evaluate the neural network on a new example you should:

- Perform the needed normalizations, use $\mu$ and $\sigma^2$ estimated using an exponentially weighted average across mini-batches seen during training.
- Use the most recent mini-batch's value of $\mu$ and $\sigma^2$ to perform the needed normalizations.
- Skip the step where you normalize using $\mu$ and $\sigma^2$ since a single test example cannot be normalized.
- 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.