**✔ Congratulations! You passed!**

✔ 1 / 1
points

**1.**
If you have 10,000,000 examples, how would you split the train/dev/test set?

○ 60% train . 20% dev . 20% test

○ 98% train . 1% dev . 1% test

**Correct**

○ 33% train . 33% dev . 33% test
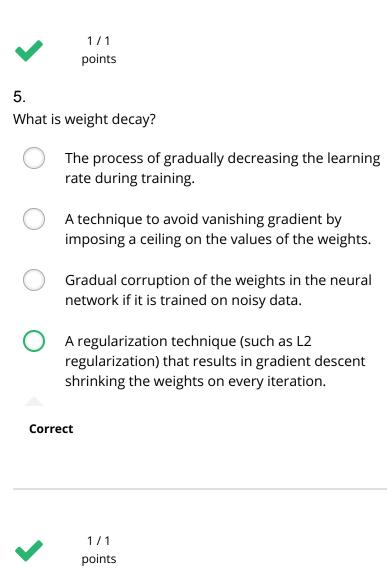
---

✔ 1 / 1
points

**2.**
The dev and test set should:

○ Come from the same distribution

**Correct**

○ Come from different distributions

○ Be identical to each other (same (x,y) pairs)

○ **Have the same number of examples**

---

✔ 1 / 1
points

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.)

☑ Increase the number of units in each hidden layer

**Correct**

☑ Make the Neural Network deeper

**Correct**

☑ Get more test data

**Un-selected is correct**

☑ Get more training data

**Un-selected is correct**

☐ Add regularization

**Un-selected is correct**

---

✔ 1 / 1
points

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.)

☐ Increase the regularization parameter lambda

**Correct**

☐ Decrease the regularization parameter lambda

**Un-selected is correct**

☐ Get more training data

**Correct**

☐ Use a bigger neural network

**Un-selected is correct**

✓ 1 / 1 points

## 5.
What is weight decay?

○ 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.

○ 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.

**Correct**

---

✓ 1 / 1 points

## 6.
What happens when you increase the regularization hyperparameter lambda?

○ Weights are pushed toward becoming smaller (closer to 0)

**Correct**

○

○ 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)

---

✔ 1 / 1 points

7.
With the inverted dropout technique, at test time:

○ You apply dropout (randomly eliminating units) and do not 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) but keep the 1/keep_prob factor in the calculations used in training.

○ 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

**Correct**

---

✔ 1 / 1 points

8.

Increasing the parameter keep_prob from (say) 0.5 to 0.6 will likely cause the following: (Check the two that apply)

☐ Increasing the regularization effect

**Un-selected is correct**

☐ Reducing the regularization effect

**Correct**

☐ Causing the neural network to end up with a higher training set error

**Un-selected is correct**

☐ Causing the neural network to end up with a lower training set error

**Correct**

---

✔ 1 / 1 points

9.

Which of these techniques are useful for reducing variance (reducing overfitting)? (Check all that apply.)

☐ Xavier initialization

**Un-selected is correct**

☐ Data augmentation

**Correct**

☐ Vanishing gradient

**Un-selected is correct**

☐ Exploding gradient

**Un-selected is correct**

☐ L2 regularization

**Correct**

☐ Dropout

**Correct**

☐ Gradient Checking

**Un-selected is correct**

---

✓ 1 / 1
points

10.
Why do we normalize the inputs $x$?

○

Normalization is another word for regularization--It helps to reduce variance

○ It makes it easier to visualize the data

○ It makes the parameter initialization faster

◉ It makes the cost function faster to optimize

**Correct**