Congratulations! You passed!

TO PASS 80% or higher

Keep Learning

GRADE
100%

Practical aspects of deep learning

LATEST SUBMISSION GRADE
100%

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

- ○ 33% train . 33% dev . 33% test
- ○ 60% train . 20% dev . 20% test
- ◉ 98% train . 1% dev . 1% test

✓ **Correct**

2. The dev and test set should: **1 / 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**

✓ **Correct**

3. If your Neural Network model seems to have high variance, what of the following would be promising things to try? **1 / 1 point**

- [ ] Get more test data
- [ ] Increase the number of units in each hidden layer
- [ ] Make the Neural Network deeper
- [x] Add regularization

✓ **Correct**

- [x] Get more training data

✓ **Correct**

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 / 1 point**

- [x] Increase the regularization parameter lambda

✓ **Correct**

- [ ] Decrease the regularization parameter lambda
- [x] Get more training data

✓ **Correct**

- [ ] Decrease the regularization parameter lambda
- [x] Get more training data

✓ **Correct**

- [ ] Use a bigger neural network

5. What is weight decay? **1 / 1 point**

- ○ 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.

- [ ] Use a bigger neural network

5. What is weight decay? **1 / 1 point**

- ○ 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**

6. What happens when you increase the regularization hyperparameter lambda? **1 / 1 point**

- ◉ Weights are pushed toward becoming smaller (closer to 0)
- ○ Weights are pushed toward becoming bigger (further from 0)
- ○ 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
- ○ You apply dropout (randomly eliminating units) and do not keep the 1/keep_prob factor in the calculations used in training

✓ Correct

8. Increasing the parameter keep_prob from (say) 0.5 to 0.6 will likely cause the following: (Check the two that apply) **1 / 1 point**

- [ ] Increasing the regularization effect
- [x] Reducing the regularization effect

✓ Correct

- [ ] Causing the neural network to end up with a higher training set error

9. Which of these techniques are useful for reducing variance (reducing overfitting)? (Check all that apply.) **1 / 1 point**

- [x] L2 regularization

✓ Correct

- [ ] Xavier initialization
- [ ] Exploding gradient
- [ ] Gradient Checking
- [x] Data augmentation
- [ ] 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)

✓ Correct

7. With the inverted dropout technique, at test time: **1 / 1 point**

- [ ] You do not apply dropout (do not randomly eliminate units), but keep the 1/keep_prob

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

Correct