Your grade: 100%

Your latest: **100%** • Your highest: **100%** • To pass you need at least 80%. We keep your highest score.

Next item

1. To help you practice strategies for machine learning, this week we'll present another scenario and ask how you would act. We think this "simulator" of working in a machine learning project will give you an idea of what leading a machine learning project could be like!

1 / 1 point

You are employed by a startup building self-driving cars. You are in charge of detecting road signs (stop sign, pedestrian crossing sign, construction ahead sign) and traffic signals (red and green lights) in images. The goal is to recognize which of these objects appear in each image. As an example, this image contains a pedestrian crossing sign and red traffic lights.

$$y^{(i)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{array}{l} \text{"stop sign"} \\ \text{"pedestrian crossing sign"} \\ \text{"construction ahead sign"} \\ \text{"red traffic light"} \\ \text{"green traffic light"} \end{array}$$

Your 100,000 labeled images are taken using the front-facing camera of your car. This is also the distribution of data you care most about doing well on. You think you might be able to get a much larger dataset off the internet, which could be helpful for training even if the distribution of internet data is not the same.

Suppose that you came from working with a project for human detection in city parks, so you know that detecting humans in diverse environments can be a difficult problem. What is the first thing you do? Assume each of the steps below would take about an equal amount of time (a few days).

- (●) Train a basic model and proceed with error analysis.
- ( ) Leave aside the pedestrian detection, to move faster and then later solve the pedestrian problem alone.
- ( ) Start by solving pedestrian detection, since you already have the experience to do this.
- ( ) Spend a few days collecting more data to determine how hard it will be to include more pedestrians in your dataset.

Expand

Correct
Correct. As discussed in the lecture, it is better to create your first system quickly and then iterate.

2. Your goal is to detect road signs (stop sign, pedestrian crossing sign, construction ahead sign) and traffic signals (red and green lights) in images. The goal is to recognize which of these objects appear in each image. You plan to use a deep neural network with ReLU units in the hidden layers. For the output layer, which of the following gives you the most appropriate activation function?

1 / 1 point

- ( ) Linear
- ( ) Softmax
- (●) Sigmoid
- ( ) ReLU

Expand

Correct
Correct. This works well since the output would be valued between 0 and 1 which represents the probability that one of the possibilities is present in an image.

3. When trying to determine what strategy to implement to improve the performance of a model, we manually check all images of the training set where the algorithm was successful. True/False?

1 / 1 point

- (●) False
- ( ) True

Expand

Correct
Correct. This set should be too large to manually check all the images. It is better to focus on the images that the algorithm got wrong from the dev set. Also, choose a large enough subset that we can manually check.

4. After working on the data for several weeks, your team ends up with the following data:

1 / 1 point

- 100,000 labeled images taken using the front-facing camera of your car.
- 900,000 labeled images of roads downloaded from the internet.
- Each image's labels precisely indicate the presence of any specific road signs and traffic signals or combinations of them. For example, $y^{(i)} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ means the image contains a stop sign and a red traffic light.

When using a non fully labeled image such as $y^{(i)} = \begin{bmatrix} 0 \\ ? \\ 1 \\ ? \\ 1 \end{bmatrix}$, which of the following strategies is most appropriate to calculate the loss function to train as a multi-task learning problem?

- ○ Make the missing entries equal to 1.
- ○ It is not possible to use non fully labeled images if we train as a multi-task learning problem.
- ○ Make the missing entries equal to 0.
- ◉ Calculate the loss as $\sum \mathcal{L}(\hat{y}_j^{(i)}, y_j^{(i)})$ where the sum goes over all the know components of $y^{(i)}$.

Expand

Correct
Correct. We can't use the components of the labels that are missing but we can use the ones we have to train the model.

5. The distribution of data you care about contains images from your car's front-facing camera; which comes from a different distribution than the images you were able to find and download off the internet. How should you split the dataset into train/dev/test sets?

1 / 1 point

- ○ Choose the training set to be the 900,000 images from the internet along with 20,000 images from your car's front-facing camera. The 80,000 remaining images will be split equally in dev and test sets.
- ○ Mix all the 100,000 images with the 900,000 images you found online. Shuffle everything. Split the 1,000,000 images dataset into 600,000 for the training set, 200,000 for the dev set and 200,000 for the test set.
- ◉ Choose the training set to be the 900,000 images from the internet along with 80,000 images from your car's front-facing camera. The 20,000 remaining images will be split equally in dev and test sets.
- ○ Mix all the 100,000 images with the 900,000 images you found online. Shuffle everything. Split the 1,000,000 images dataset into 980,000 for the training set, 10,000 for the dev set and 10,000 for the test set.

Expand

Correct
Yes. As seen in the lecture, it is important that your dev and test set have the closest possible distribution to "real" data. It is also important for the training set to contain enough "real" data to avoid having a data-mismatch problem.

6. Assume you've finally chosen the following split between the data:

1 / 1 point

Dataset:	Contains:	Error of the algorithm:
Training	940,000 images randomly picked from (900,000 internet images + 60,000 car's front-facing camera images)	1%
Training-Dev	20,000 images randomly picked from (900,000 internet images + 60,000 car's front-facing camera images)	5.1%
Dev	20,000 images from your car's front-facing camera	5.6%
Test	20,000 images from the car's front-facing camera	6.8%

You also know that human-level error on the road sign and traffic signals classification task is around 0.5%. Which of the following is true?

- ○ You have a high bias.
- ◉ You have a high variance problem.
- ○ You have a large data-mismatch problem.
- ○ The size of the train-dev set is too high.

Expand

Correct
Correct. Since the difference between the training-dev error and the training error is high.

7. Assume you've finally chosen the following split between the data:

1 / 1 point

Dataset:	Contains:	Error of the algorithm:
Training	940,000 images randomly picked from (900,000 internet images + 60,000 car's front-facing camera images)	2%
Training-Dev	20,000 images randomly picked from (900,000 internet images + 60,000 car's front-facing camera images)	2.3%
Dev	20,000 images from your car's front-facing camera	1.3%
Test	20,000 images from the car's front-facing camera	1.1%

You also know that human-level error on the road sign and traffic signals classification task is around 0.5%. Based on the information given you conclude that the Bayes error for the dev/test distribution is probably higher than for the train distribution. True/False?

- ○ True
- ◉ False

Expand

Correct

8. You decide to focus on the dev set and check by hand what the errors are due to. Here is a table summarizing your discoveries:

1 / 1 point

Overall dev set error	15.3%
Errors due to incorrectly labeled data	4.1%
Errors due to foggy pictures	3.0%
Errors due to partially occluded elements.	7.2%
Errors due to other causes	1.0%

In this table, 4.1%, 7.2%, etc. are a fraction of the total dev set (not just examples of your algorithm mislabeled). For example, about 7.2/15.3 = 47% of your errors are due to partially occluded elements.

You shouldn't invest all your efforts to get more images with partially occluded elements since 4.1 + 3.0 + 1.0 = 8.1 > 7.2. True/False?

- ◉ False
- ○ True

Expand

Correct

Correct. These kinds of arguments don't help us to decide on the strategy to follow. Other factors should be used, such as the tradeoff between the cost of getting new images and the improvement of the system performance.

9. You decide to focus on the dev set and check by hand what the errors are due to. Here is a table summarizing your discoveries:

1 / 1 point

Overall dev set error	15.3%
Errors due to incorrectly labeled data	4.1%
Errors due to foggy pictures	3.0%
Errors due to partially occluded elements.	7.2%
Errors due to other causes	1.0%

In this table, 4.1%, 7.2%, etc. are a fraction of the total dev set (not just examples of your algorithm mislabeled). For example, about 7.2/15.3 = 47% of your errors are due to partially occluded elements.

You find out that there is an anti-reflective film guarantee to eliminate the sun reflection, but it is quite costly. Which of the following gives the best description of what the investment in the film can do to the model?

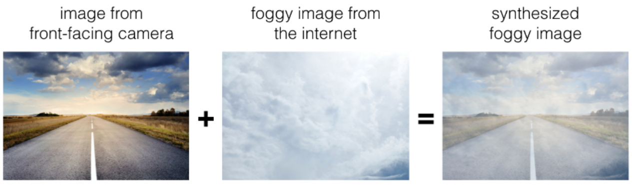
- ◉ The film will reduce the dev set error with 7.2% at the most.
- ○ The film will reduce at least 7.2% of the dev set error.
- ○ The overall test set error will be reduced by at most 7.2%.

Expand

Correct

Yes. Remember that this 7.2% gives us an estimate for the ceiling of how much the error can be reduced when the cause is fixed.

10. You decide to use data augmentation to address foggy images. You find 1,000 pictures of fog off the internet, and “add” them to clean images to synthesize foggy days, like this: **1 / 1 point**



We can't use this data since they have a different distribution from the ones we used (internet and front-facing camera). True/False?

○ True

◉ False

Expand

Correct
Correct. The new synthesized images are added to the training set and as long as they look realistic to the human eye this will be useful data to train the model.

11. After working further on the problem, you've decided to correct the incorrectly labeled data. Your team corrects the labels of the wrongly predicted images on the dev set. Which of the following is a necessary step to take? **1 / 1 point**

○ Create a train-dev set to estimate how many incorrectly labeled examples are in the train set.

◉ Correct the labels of the test set.

○ Correct the labels of the train set.

○ Use a correctly labeled version and an incorrectly labeled version to make the model more robust.

Expand

Correct
Correct. Recall that the dev set and the test set must come from the same distribution.

12. So far your algorithm only recognizes red and green traffic lights. One of your colleagues in the startup is starting to work on recognizing a yellow traffic light. (Some countries call it an orange light rather than a yellow light; we'll use the US convention of calling it yellow.) Images containing yellow lights are quite rare, and she doesn't have enough data to build a good model. She hopes you can help her out using transfer learning. **1 / 1 point**

What do you tell your colleague?

◉ She should try using weights pre-trained on your dataset, and fine-tuning further with the yellow-light dataset.

○ You cannot help her because the distribution of data you have is different from hers, and is also lacking the yellow label.

○ If she has (say) 10,000 images of yellow lights, randomly sample 10,000 images from your dataset and put your and her data together. This prevents your dataset from “swamping” the yellow lights dataset.

○ Recommend that she try multi-task learning instead of transfer learning using all the data.

Expand

Correct
Yes. You have trained your model on a huge dataset, and she has a small dataset. Although your labels are different, the parameters of your model have been trained to recognize many characteristics of road and traffic images which will be useful for her problem. This is a perfect case for transfer learning, she can start with a model with the same architecture as yours, change what is after the last hidden layer and initialize it with your trained parameters.

13. One of your colleagues at the startup is starting a project to classify stop signs in the road as speed limit signs or not. He has approximately 30,000 examples of each image and 30,000 images without a sign. He thought of using your model and applying transfer learning but then he noticed that you use multi-task learning, hence he can't use your model. True/False? **1 / 1 point**

○ True

◉ False

Expand

Correct

Correct. When using transfer learning we can remove the last layer. That is one of the aspects that is different from a binary classification problem.

14. **To recognize a stop sign you use the following approach:** First, you use a neural network to predict bounding box co-ordinates around all traffic signs (if any) within an input image. You then pass the results to a different neural network to determine if the predicted traffic signs (if any) are a stop sign or not. We are using multi-task learning. True/False?

1 / 1 point

- True
- False

Expand

Correct

Correct. Multi-task learning is about joining several tasks that can benefit from each other. Since there are 2 different neural networks being used here that do not share weights (i.e. structure), this problem has 2 single task learning neural networks and not a multi-task learning setup.

15. An end-to-end approach doesn't require that we hand-design useful features, it only requires a large enough model. True/False?

1 / 1 point

- True
- False

Expand

Correct

Correct. This is one of the major characteristics of deep learning models, that we don't need to hand-design the features.