Structuring Machine Learning Projects

dzlabs

September 2018

1 Week 1 - ML Strategy (1)

1.1 Introduction to ML Strategy

1.1.1 Why ML Strategy

To improve your deep learning algorithm, you may end up with lot of ideas for improvement like:

- Collect more data
- Collect more diverse training set
- Train algorithm longer with gradient descent
- Try Adam instead of gradient descent
- Try bigger network
- Try smaller network
- Try dropout
- Add L_2 regularization
- Network architecture: Activation functions, # hidden units

1.1.2 Orthogonalization

Orthogonalization or orthogonality is a system design property that assures that modifying an instruction or a component of an algorithm will not create or propagate side effects to other components of the system. It becomes easier to verify the algorithms independently from one another, it reduces testing and development time.

When a supervised learning system is design, these are the 4 assumptions that needs to be true and orthogonal.

1. Fit training set well in cost function
   - If it doesn't fit well, the use of a bigger neural network or switching to a better optimization algorithm might help.
2. Fit development set well on cost function
   - If it doesn't fit well, regularization or using bigger training set might help.
3. Fit test set well on cost function
   - If it doesn't fit well, the use of a bigger development set might help
4. Performs well in real world
   - If it doesn't perform well, the development test set is not set correctly or the cost function is not evaluating the right thing.

1.2 Setting up your goal

1.2.1 Single number evaluation metric

1.2.2 Satisficing and Optimizing metric

If there are multiple things you care about, e.g.:

- the optimizing metric that you want to do as well as possible on
- one or more as satisfying metrics were you'll be satisfied

Almost it does better than some threshold you can now have an almost automatic way of quickly looking at multiple core size and picking the best one. These evaluation matrix must be evaluated or calculated on a *training* set or a *development* set or maybe on the *test* set.

1.2.3 Train/dev/test distributions

Setting up the training, development and test sets have a huge impact on productivity. It is important to choose the development and test sets from the **same distribution** and it must be **taken randomly** from all the data. **Guideline:** Choose a development set and test set to reflect data you expect to get in the future and consider important to do well.

1.2.4 Size of the dev and test sets

1.2.5 When to change dev/test sets and metrics

1.3 Comparing to human-level performance

1.3.1 Why human-level performance?

1.3.2 Avoidable bias

By knowing what the human-level performance is, it is possible to tell when a training set is performing well or not.

Example: Cat vs Non-Cat

	Classification error (%)	
	Scenario A	Scenario B
Humans	1	7.5
Training error	8	8
Development error	10	10

In this case, the human level error as a proxy for Bayes error since humans are good to identify images. If you want to improve the performance of the training set but you can't do better than the Bayes error otherwise the training set is overfitting. By knowing the Bayes error, it is easier to focus on whether bias or variance avoidance tactics will improve the performance of the model.

Scenario A There is a 7% gap between the performance of the training set and the human level error. It means that the algorithm isn't fitting well with the training set since the target is around 1%. To resolve the issue, we use bias reduction technique such as training a bigger neural network or running the training set longer.

Scenario B The training set is doing good since there is only a 0.5% difference with the human level error. The difference between the training set and the human level error is called avoidable bias. The focus here is to reduce the variance since the difference between the training error and the development error is 2%. To resolve the issue, we use variance reduction technique such as regularization or have a bigger training set.

1.3.3 Surpassing human-level performance

Example1: Classification task

	Classification error (%)	
	Scenario A	Scenario B
Team of Humans	0.5	0.5
One Human	1.0	1
Training error	0.6	0.3
Development error	0.8	0.4

Scenario A In this case, the Bayes error is 0.5%, therefore the available bias is 0.1% et the variance is 0.2%.

Scenario B In this case, there is not enough information to know if bias reduction or variance reduction has to be done on the algorithm. It doesn't mean that the model cannot be improve, it means that the conventional ways to know if bias reduction or variance reduction are not working in this case.

There are many problems where machine learning significantly surpasses human-level performance, especially with structured data:

- Online advertising

- Product recommendations
- Logistics (predicting transit time)
- Loan approvals

1.3.4 Improving your model performance

1.4 Machine Learning flight simulator

QUIZ - Bird recognition in the city of Peacetopia (case study)

1. Problem Statement: This example is adapted from a real production application, but with details disguised to protect confidentiality.

You are a famous researcher in the City of Peacetopia. The people of Peacetopia have a common characteristic: they are afraid of birds. To save them, you have **to build an algorithm that will detect any bird flying over Peacetopia** and alert the population.

The City Council gives you a dataset of 10,000,000 images of the sky above Peacetopia, taken from the city's security cameras. They are labelled:

- y = 0: There is no bird on the image
- y = 1: There is a bird on the image

Your goal is to build an algorithm able to classify new images taken by security cameras from Peacetopia. There are a lot of decisions to make:

- What is the evaluation metric?
- How do you structure your data into train/dev/test sets?

`Metric of success` The City Council tells you that they want an algorithm that

1. Has high accuracy
2. Runs quickly and takes only a short time to classify a new image.
3. Can fit in a small amount of memory, so that it can run in a small processor that the city will attach to many different security cameras.

`Note:` Having three evaluation metrics makes it harder for you to quickly choose between two different algorithms, and will slow down the speed with which your team can iterate. True/False?

- True (X)
- False

2. After further discussions, the city narrows down its criteria to:

- "We need an algorithm that can let us know a bird is flying over Peacetopia as accurately as possible."
- "We want the trained model to take no more than 10sec to classify a new image."
- "We want the model to fit in 10MB of memory."

If you had the three following models, which one would you choose?

- Test Accuracy: 97%, Runtime: 1 sec, Memory size: 3MB
- Test Accuracy: 99%, Runtime: 13 sec, Memory size: 9MB
- Test Accuracy: 97%, Runtime: 3 sec, Memory size: 2MB
- Test Accuracy: 98%, Runtime: 9 sec, Memory size: 9MB (X)

3. Based on the city's requests, which of the following would you say is true?

- Accuracy is an optimizing metric; running time and memory size are a satisficing metrics. (X)
- Accuracy is a satisficing metric; running time and memory size are an optimizing metric.

- Accuracy, running time and memory size are all optimizing metrics because you want to do well on all three.
- Accuracy, running time and memory size are all satisficing metrics because you have to do sufficiently well on all three for your system to be acceptable.

4. `Structuring your data:` Before implementing your algorithm, you need to split your data into train/dev/test sets. Which of these do you think is the best choice?

- Train: 6,000,000; Dev: 3,000,000; Test: 1,000,000
- Train: 6,000,000; Dev: 1,000,000; Test: 3,000,000
- Train: 3,333,334; Dev: 3,333,333; Test: 3,333,333
- Train: 9,500,000; Dev: 250,000; Test: 250,000 (X)

5. After setting up your train/dev/test sets, the City Council comes across another 1,000,000 images, called the "citizens' data". Apparently the citizens of Peacetopia are so scared of birds that they volunteered to take pictures of the sky and label them, thus contributing these additional 1,000,000 images. These images are different from the distribution of images the City Council had originally given you, but you think it could help your algorithm.

You should not add the citizens' data to the training set, because this will cause the training and dev/test set distributions to become different, thus hurting dev and test set performance. True/False?

- True
- False (X)

6. One member of the City Council knows a little about machine learning, and thinks you should add the 1,000,000 citizens' data images to the test set. You object because:

- This would cause the dev and test set distributions to become different. This is a bad idea because you're not aiming where you want to hit. (X)
- The 1,000,000 citizens' data images do not have a consistent x–¿y mapping as the rest of the data (similar to the New York City/Detroit housing prices example from lecture).
- The test set no longer reflects the distribution of data (security cameras) you most care about. (X)
- A bigger test set will slow down the speed of iterating because of the computational expense of evaluating models on the test set.

7. You train a system, and its errors are as follows (error = 100%-Accuracy):

Training set error	4.0%
Dev set error	4.5%

This suggests that one good avenue for improving performance is to train a bigger network so as to drive down the 4.0% training error. Do you agree?

- Yes, because having 4.0% training error shows you have high bias. (X 1)
- Yes, because this shows your bias is higher than your variance.
- No, because this shows your variance is higher than your bias.
- No, because there is insufficient information to tell. (X 2)

8. You ask a few people to label the dataset so as to find out what is human-level performance. You find the following levels of accuracy:

Bird watching expert #1	0.3% error
Bird watching expert #2	0.5% error
Normal person #1 (not a bird watching expert)	1.0% error
Normal person #2 (not a bird watching expert)	1.2% error

If your goal is to have "human-level performance" be a proxy (or estimate) for Bayes error, how would you define "human-level performance"?

- 0.0% (because it is impossible to do better than this)
- 0.3% (accuracy of expert #1) (X)
- 0.4% (average of 0.3 and 0.5)
- 0.75% (average of all four numbers above)

9. Which of the following statements do you agree with?

- A learning algorithm's performance can be better than human-level performance but it can never be better than Bayes error. (X 2)
- A learning algorithm's performance can never be better than human-level performance but it can be better than Bayes error.
- A learning algorithm's performance can never be better than human-level performance nor better than Bayes error.
- A learning algorithm's performance can be better than human-level performance and better than Bayes error. (X 1)

10. You find that a team of ornithologists debating and discussing an image gets an even better 0.1% performance, so you define that as "human-level performance." After working further on your algorithm, you end up with the following:

Human-level performance	0.1%
Training set error	2.0%
Dev set error	2.1%

Based on the evidence you have, which two of the following four options seem the most promising to try? (Check two options.)

- Train a bigger model to try to do better on the training set. (X)
- Get a bigger training set to reduce variance.
- Try decreasing regularization. (X)
- Try increasing regularization.

11. You also evaluate your model on the test set, and find the following:

Human-level performance	0.1%
Training set error	2.0%
Dev set error	2.1%
Test set error	7.0%

What does this mean? (Check the two best options.)

- You should get a bigger test set.
- You should try to get a bigger dev set. (X)
- You have overfit to the dev set. (X)
- You have underfit to the dev set.

12. After working on this project for a year, you finally achieve:

Human-level performance	0.10%
Training set error	0.05%
Dev set error	0.05%

What can you conclude? (Check all that apply.)

- If the test set is big enough for the 0.05% error estimate to be accurate, this implies Bayes error is ≤ 0.05 (X)
- It is now harder to measure avoidable bias, thus progress will be slower going forward. (X)
- With only 0.09% further progress to make, you should quickly be able to close the remaining gap to 0%
- This is a statistical anomaly (or must be the result of statistical noise) since it should not be possible to surpass human-level performance.

13. It turns out Peacetopia has hired one of your competitors to build a system as well. Your system and your competitor both deliver systems with about the same running time and memory size. However, your system has higher accuracy! However, when Peacetopia tries out your and your competitor's systems, they conclude they actually like your competitor's system better, because even though you have higher overall accuracy, you have more false negatives (failing to raise an alarm when a bird is in the air). What should you do?

- Look at all the models you've developed during the development process and find the one with the lowest false negative error rate.
- Ask your team to take into account both accuracy and false negative rate during development. (X 1)
- Rethink the appropriate metric for this task, and ask your team to tune to the new metric.
- Pick false negative rate as the new metric, and use this new metric to drive all further development. (X 2)

14. You've handily beaten your competitor, and your system is now deployed in Peacetopia and is protecting the citizens from birds! But over the last few months, a new species of bird has been slowly migrating into the area, so the performance of your system slowly degrades because your data is being tested on a new type of data.

You have only 1,000 images of the new species of bird. The city expects a better system from you within the next 3 months. Which of these should you do first?

- Use the data you have to define a new evaluation metric (using a new dev/test set) taking into account the new species, and use that to drive further progress for your team.
- Put the 1,000 images into the training set so as to try to do better on these birds.
- Try data augmentation/data synthesis to get more images of the new type of bird. (X 1)
- Add the 1,000 images into your dataset and reshuffle into a new train/dev/test split. (X 2)

15. The City Council thinks that having more Cats in the city would help scare off birds. They are so happy with your work on the Bird detector that they also hire you to build a Cat detector. (Wow Cat detectors are just incredibly useful aren't they.) Because of years of working on Cat detectors, you have such a huge dataset of 100,000,000 cat images that training on this data takes about two weeks. Which of the statements do you agree with? (Check all that agree.)

- Buying faster computers could speed up your teams' iteration speed and thus your team's productivity. (X)
- Needing two weeks to train will limit the speed at which you can iterate. (X)
- Having built a good Bird detector, you should be able to take the same model and hyperparameters and just apply it to the Cat dataset, so there is no need to iterate.
- If 100,000,000 examples is enough to build a good enough Cat detector, you might be better of training with just 10,000,000 examples to gain a $\approx$ 10x improvement in how quickly you can run experiments, even if each model performs a bit worse because it's trained on less data. (X)

2 Week 2 - ML Strategy (2)

2.1 Error Analysis

2.1.1 Carrying out error analysis

Manually examining mistakes that your algorithm is making, can give you insights into what to do next.
Suppose you have a Cat classifier with acurracy of 90%, and error of 10%. Looking at the wronly classifed pictures, you realize they're Dog's pictures.
Question: Should you try to make your cat classifier do better on dogs?
This will take a long time (may be months), what you should do as part of the error analysis:

- Get 100 mislabeled dev set examples
- Count up how many are dogs.

You found that 5% percent of the errors are dogs, so the best you can do is you improve the algo error from 10% to 9.5%, Ceiling or upper bound when working on the dog problem helps you.

2.1.2 Cleaning up incorrectly labeled data

When trying to understand what mistakes it is making, you should look at the data and try to counter the fraction of errors. This small time of counting data can really help you prioritize where to go next and trying to decide what ideas or what directions to prioritize things.

2.1.3 Build your first system quickly, then iterate

`Build system quickly, then iterate`: Depending on the area of application, the guideline below will help you prioritize when you build your system.
Guideline:

1. Set up development/ test set and metrics
    - Set up a target
2. Build an initial system quickly
    - Train training set quickly: Fit the parameters
    - Development set: Tune the parameters
    - Test set: Assess the performance
3. Use Bias/Variance analysis & Error analysis to prioritize next steps

2.2 Mismatched training and dev/test set

2.2.1 Training and testing on different distributions

Training a cat classifier for mobile where the uploaded pictures have poor quality. Best option is the training to include 200,000 images from the web and 5,000 from the mobile app. The dev set will be 2,500 images from the mobile app, and the test set will be 2,500 images also from the mobile app.

- The advantage of this way of splitting up your data into train, dev, and test, is that you're now aiming the target where you want it to be. Your dev set has data uploaded from the mobile app and that's the distribution of images you really care about, in order to build a machine learning system that does really well on the mobile app distribution of images.
- The disadvantage is that now your training distribution is different from your dev and test set distributions.

2.2.2 Bias and Variance with mismatched data distributions

2.2.3 Addressing data mismatch

This is a general guideline to address data mismatch:

- Perform manual error analysis to understand the error differences between training, development/test sets. Development should never be done on test set to avoid overfitting.
- Make training data or collect data similar to development and test sets. To make the training data more similar to your development set, you can use is artificial data synthesis. However, it is possible that if you might be accidentally simulating data only from a tiny subset of the space of all possible examples.

2.3 Learning from multiple tasks

2.3.1 Transfer learning

Example in radiology, it's difficult to get that many x-ray scans to build a good radiology diagnosis system. So in that case, you might find a related but different task, such as image recognition, where you can get maybe a million images and learn a lot of load-over features from that, so that you can then try to do well on Task B on your radiology task despite not having that much data for it.

2.3.2 Multi-task learning

> Multi-task learning and transfer learning are both important tools to have in your tool bag.

`Transfer learning` and `Multi-task learning` often you're presented in a similar way, in practice there are a lot more applications of transfer learning than of multi-task learning. In fact, it's just difficult to set up or to find so many different tasks that you would actually want to train a single neural network for. Again, with some sort of computer vision, object detection examples being the most notable exception.

2.4 End-to-end deep learning

2.4.1 What is end-to-end deep learning?

There have been some data processing systems, or learning systems that require multiple stages of processing. And what end-to-end deep learning does, is it can take all those multiple stages, and replace it usually with just a single neural network.

End to end is only interesting in the case there is a lot of data. E.g. in image recognition, the best approach to date, seems to be a multi-step approach, where first, you run one piece of software to detect the person's face. So this first detector to figure out where's the person's face. Having detected the person's face, you then zoom in to that part of the image and crop that image so that the person's face is centered. Then, it is this picture that I guess I drew here in red, this is then fed to the neural network, to then try to learn, or estimate the person's identity.

2.4.2 Whether to use end-to-end deep learning

It's called end to end because it's trying to map an input to the output directly without mid-steps.

2.5 Machine Learning flight simulator

2.5.1 QUIZ - Autonomous driving (case study)

1. To help you practice strategies for machine learning, in 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 a task of what leading a machine learning project could be like!

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, the above image contains a pedestrian crossing sign and red traffic lights

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, that could be helpful for training even if the distribution of internet data is not the same.

You are just getting started on this project. What is the first thing you do? Assume each of the steps below would take about an equal amount of time (a few days).

- Spend a few days collecting more data using the front-facing camera of your car, to better understand how much data per unit time you can collect.
- Spend a few days getting the internet data, so that you understand better what data is available.
- Spend a few days checking what is human-level performance for these tasks so that you can get an accurate estimate of Bayes error.
- Spend a few days training a basic model and see what mistakes it makes. (X) As discussed in lecture, applied ML is a highly iterative process. If you train a basic model and carry out error analysis (see what mistakes it makes) it will help point you in more promising directions.

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, a softmax activation would be a good choice for the output layer because this is a multi-task learning problem. True/False?

- True (X 1)
- False (X 2)

3. You are carrying out error analysis and counting up what errors the algorithm makes. Which of these datasets do you think you should manually go through and carefully examine, one image at a time?

- 500 images on which the algorithm made a mistake (X) Focus on images that the algorithm got wrong. Also, 500 is enough to give you a good initial sense of the error statistics. There's probably no need to look at 10,000, which will take a long time.
- 500 randomly chosen images
- 10,000 images on which the algorithm made a mistake
- 10,000 randomly chosen images

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

- 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.

Because this is a multi-task learning problem, you need to have all your $y^{(i)}$ vectors fully labeled. If one example is equal to $\begin{bmatrix} 1 \\ ? \\ 1 \\ 1 \\ ? \end{bmatrix}$ then the learning algorithm will not be able to use that example. True/False?

- True

- False (X) As seen in the lecture on multi-task learning, you can compute the cost such that it is not influenced by the fact that some entries haven't been labeled.

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?

- 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. (X) Yes. As seen in 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.

- 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.

- 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.

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

You also know that human-level error on the road sign and traffic signals classification task is around 0.5%. Which of the following are True? (Check all that apply).

- You have a large data-mismatch problem because your model does a lot better on the training-dev set than on the dev set (X)

- Your algorithm overfits the dev set because the error of the dev and test sets are very close.

- You have a large avoidable-bias problem because your training error is quite a bit higher than the human-level error. (X)

- You have a large variance problem because your training error is quite higher than the human-level error.

- You have a large variance problem because your model is not generalizing well to data from the same training distribution but that it has never seen before.

7. Based on table from the previous question, a friend thinks that the training data distribution is much easier than the dev/test distribution. What do you think?

- Your friend is right. (I.e., Bayes error for the training data distribution is probably lower than for the dev/test distribution.) (X 2)

- Your friend is wrong. (I.e., Bayes error for the training data distribution is probably higher than for the dev/test distribution.) (X 1)

- There's insufficient information to tell if your friend is right or wrong. (X 3)

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

In this table, 4.1%, 8.0%, etc.are a fraction of the total dev set (not just examples your algorithm mislabeled). I.e. about 8.0/14.3 = 56

The results from this analysis implies that the team's highest priority should be to bring more foggy pictures into the training set so as to address the 8.0% of errors in that category. True/False?

- True because it is the largest category of errors. As discussed in lecture, we should prioritize the largest category of error to avoid wasting the team's time. (X 2) This is the right way to go unless it is hard to get access to foggy data. Another answer safer and more appropriate in this situation.

- True because it is greater than the other error categories added together (8.0 ¿ 4.1+2.2+1.0).

- False because this would depend on how easy it is to add this data and how much you think your team thinks it'll help. (X 3)
- False because data augmentation (synthesizing foggy images by clean/non-foggy images) is more efficient. (X 1)

9. You can buy a specially designed windshield wiper that help wipe off some of the raindrops on the front-facing camera. Based on the table from the previous question, which of the following statements do you agree with?

- 2.2% would be a reasonable estimate of the maximum amount this windshield wiper could improve performance. (X) Yes. You will probably not improve performance by more than 2.2% by solving the raindrops problem. If your dataset was infinitely big, 2.2% would be a perfect estimate of the improvement you can achieve by purchasing a specially designed windshield wiper that removes the raindrops.
- 2.2% would be a reasonable estimate of the minimum amount this windshield wiper could improve performance.
- 2.2% would be a reasonable estimate of how much this windshield wiper will improve performance.
- 2.2% would be a reasonable estimate of how much this windshield wiper could worsen performance in the worst case.

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:

Which of the following statements do you agree with?

- There is little risk of overfitting to the 1,000 pictures of fog so long as you are combing it with a much larger (¿¿1,000) of clean/non-foggy images.
- Adding synthesized images that look like real foggy pictures taken from the front-facing camera of your car to training dataset won't help the model improve because it will introduce avoidable-bias.
- So long as the synthesized fog looks realistic to the human eye, you can be confident that the synthesized data is accurately capturing the distribution of real foggy images (or a subset of it), since human vision is very accurate for the problem you're solving. (X) Yes. If the synthesized images look realistic, then the model will just see them as if you had added useful data to identify road signs and traffic signals in a foggy weather. I will very likely help.

11. After working further on the problem, you've decided to correct the incorrectly labeled data on the dev set. Which of these statements do you agree with? (Check all that apply).

- You should also correct the incorrectly labeled data in the test set, so that the dev and test sets continue to come from the same distribution (X) Yes because you want to make sure that your dev and test data come from the same distribution for your algorithm to make your team's iterative development process is efficient.
- You should correct incorrectly labeled data in the training set as well so as to avoid your training set now being even more different from your dev set. (X 1) No, deep learning algorithms are quite robust to having slightly different train and dev distributions.
- You should not correct the incorrectly labeled data in the test set, so that the dev and test sets continue to come from the same distribution
- You should not correct incorrectly labeled data in the training set as it does not worth the time. (X 3)

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.

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.
- 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. (X 1)
- You cannot help her because the distribution of data you have is different from hers, and is also lacking the yellow label. (X 3)
- Recommend that she try multi-task learning instead of transfer learning using all the data. (X 2)

13. Another colleague wants to use microphones placed outside the car to better hear if there're other vehicles around you. For example, if there is a police vehicle behind you, you would be able to hear their siren. However, they don't have much to train this audio system. How can you help?

- Transfer learning from your vision dataset could help your colleague get going faster. Multi-task learning seems significantly less promising.
- Multi-task learning from your vision dataset could help your colleague get going faster. Transfer learning seems significantly less promising. (X 1)
- Either transfer learning or multi-task learning could help our colleague get going faster.
- Neither transfer learning nor multi-task learning seems promising. (X 2) Yes. The problem he is trying to solve is quite different from yours. The different dataset structures make it probably impossible to use transfer learning or multi-task learning.

14. To recognize red and green lights, you have been using this approach:

- (A) Input an image (x) to a neural network and have it directly learn a mapping to make a prediction as to whether there's a red light and/or green light (y).

A teammate proposes a different, two-step approach:

- (B) In this two-step approach, you would first (i) detect the traffic light in the image (if any), then (ii) determine the color of the illuminated lamp in the traffic light.

Between these two, Approach B is more of an end-to-end approach because it has distinct steps for the input end and the output end. True/False?

- True
- False (X) Yes. (A) is an end-to-end approach as it maps directly the input (x) to the output (y).

15. Approach A (in the question above) tends to be more promising than approach B if you have a ________ (fill in the blank).

- Large training set (X) Yes. In many fields, it has been observed that end-to-end learning works better in practice, but requires a large amount of data.
- Multi-task learning problem.
- Large bias problem.
- Problem with a high Bayes error.