**1/ Error Analysis:**

* You have dog and cat classifier. Your team want to perform better on dog.
* Should you start project focusing on the dog problem?
* Get 100 mislabel dev set examples and exam them manually how many are dog?
* If 5% is images are dog. This means that if you solve the dog problem you only solve 5/100 dog problem. => your error now will go from 10% => 9.5%
* If 50% is images og dogs => Start the project above will solve 5% error rate

Timeline

Description automatically generated

Calendar

Description automatically generated with medium confidence

a/ Cleaning Up Incorrectly Labeled Data – Maybe its not the neural network’s fault:

Timeline

Description automatically generated

Graphical user interface, text, application

Description automatically generated

b/ Build your first systems quickly, then iterate:

Graphical user interface, text

Description automatically generated

* Bias and variance can give you a good sense that your speaker is far from the source which allow you to build techniques to solve this.

**2/ Mismatched Training and Dev/Test Set:**

a/ Training and testing on different distribution:

Diagram

Description automatically generated with medium confidence

* Issue with this is that a lot of image in dev is from your webpages instead of mobile app that you want

Diagram

Description automatically generated=> For deployment, you want dev and test set from the application you want.

Text

Description automatically generated

* How to detect data mismatch error

Text, letter

Description automatically generated

b/ Addressing Data Mismatch:

* Carry out manual error analysis to try to understand difference between training and dev/test sets.
* Make training data more similar, or collect more data similar to the dev/test set
* Manually have dev/test to be synthesis to the training data

**2/ Learning from Multiple Tasks**

a/ Transfer Learning:

* You trained nn on image recognition. You want to use the same nn for xray scan, you need to delete the weight to the last layer and the last layer and create new set of random initialize weight and have that output radiology diagnosis.
* In training phase: swap new dataset x,y of xray. Then retrain dataset on that entire network. You can either freeze other weight and train last weight or retrain whole thing or all layers

A picture containing diagram

Description automatically generated

* Transfer Learning is used when the you have a lot of data from the problem you are transferring from and usually relatively less data for the problem you transferring to.

Graphical user interface, text, application

Description automatically generated

b/ Multi-task learning:

* When to use multi-task learning?

+ Training on a set of tasks that coule benefit from having shared lower-level features

+ Usually: Amount of data you have for each task is quite similar.

+ Can train a big enough neural network to do well on all the tasks

**3/ End to End Deep Learning:**

* There have been some data processing systems, or learning systems that require multiple stages of processing. 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.
* Simplify the system.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Application

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Table

Description automatically generated with low confidence