Train:dev:test distributions

(DESCRIPTION)
Text, Setting up your goal. Train, dev, test, distributions.

(SPEECH)
The way you set up your training dev, or development sets and test sets, can have a huge impact on how rapidly you or your team can make progress on building machine learning application.

The same teams, even teams in very large companies, set up these data sets in ways that really slows down, rather than speeds up, the progress of the team.

Let's take a look at how you can set up these data sets to maximize your team's efficiency.

(DESCRIPTION)
New slide, Cat classification dev, test, sets.

(SPEECH)
In this video, I want to focus on how you set up your dev and test sets.

So, that dev set is also called the development set, or sometimes called the hold out cross validation set.

And, workflow in machine learning is that you try a lot of ideas, train up different models on the training set, and then use the dev set to evaluate the different ideas and pick one.

And, keep innovating to improve dev set performance until, finally, you have one clause that you're happy with that you then evaluate on your test set.

Now, let's say, by way of example, that you're building a cat crossfire, and you are operating in these regions: in the U.S, U.K, other European countries, South America, India, China, other Asian countries, and Australia.

So, how do you set up your dev set and your test set?

Well, one way you could do so is to pick four of these regions.

I'm going to use these four but it could be four randomly chosen regions.

And say, that data from these four regions will go into the dev set.

And, the other four regions, I'm going to use these four, could be randomly chosen four as well, that those will go into the test set.

It turns out, this is a very bad idea because in this example, your dev and test sets come from different distributions.

I would, instead, recommend that you find a way to make your dev and test sets come from the same distribution. So, here's what I

(DESCRIPTION)
An image of an archery target appears.

(SPEECH)
mean.

One picture to keep in mind is that, I think, setting up your dev set, plus, your single role number evaluation metric, that's like placing a target and telling your team where you think is the bull's eye you want to aim at.

Because, what happen once you've established that dev set and the metric is that, the team can innovate very quickly, try different ideas, run experiments and very quickly use the dev set and the metric to evaluate crossfires and try to pick the best

(DESCRIPTION)
An image of a circular idea appears. It includes three arrows within the circle, idea, code, experiment.

(SPEECH)
one.

So, machine learning teams are often very good at shooting different arrows into targets and innovating to get closer and closer to hitting the bullseye.

So, doing well on your metric on your dev sets.

And, the problem with how we've set up the dev and test sets in the example on the left is that, your team might spend months innovating to do well on the dev set only to realize that, when you finally go to test them on the test set, that data from these four countries or these four regions at the bottom, might be very different than the regions in your dev set.

So, you might have a nasty surprise and realize that, all the months of work you spent optimizing to the dev set, is not giving you good performance on the test set.

So, having dev and test sets from different distributions is like setting a target, having your team spend months trying to aim closer and closer to bull's eye, only to realize after months of work that, you'll say, "Oh wait, to test it, I'm going to move target over here." And,

(DESCRIPTION)
Another archery target is drawn near the original photo.

(SPEECH)
the team might say, "Well, why did you make us spend months optimizing for a different bull's eye when suddenly, you can move the bull's eye to a different location somewhere else?" So, to avoid this, what I recommend instead is that, you take all this randomly shuffled data into the dev and test set.

So that, both the dev and test sets have data from all eight regions and that the dev and test sets really come from the same distribution, which is the distribution of all of your data mixed together.

Here's

(DESCRIPTION)
New slide, True story, details changed. Optimizing on dev set on loan approvals for medium income zip codes.

(SPEECH)
another example. This is a, actually, true story but with some details changed.

So, I know a machine learning team that actually spent several months optimizing on a dev set which was comprised of loan approvals for medium income zip codes.

So, the specific machine learning problem was, "Given an input X about a loan application, can you predict why and which is, whether or not, they'll repay the loan?" So, this helps you decide whether or not to approve a loan.

And so, the dev set came from loan applications.

They came from medium income zip codes.

Zip codes is what we call postal codes in the United States.

But, after working on this for a few months, the team then, suddenly decided to test this on data from low income zip codes or low income postal codes.

And, of course, the distributional data for medium income and low income zip codes is very different.

And, the crossfire, that they spend so much time optimizing in the former case, just didn't work well at all on the latter case.

And so, this particular team actually wasted about three months of time and had to go back and really re-do a lot of work.

(DESCRIPTION)
Two archery targets are drawn to illustrate the differing locations of the goals.

(SPEECH)
And, what happened here was, the team spent three months aiming for one target, and then, after three months, the manager asked, "Oh, how are you doing on hitting this other target?" This is a totally different location.

And, it just was a very frustrating experience for the team.

(DESCRIPTION)
New slide, Guideline.

(SPEECH)
So, what I recommand for setting up a dev set and test set is, choose a dev set and test set to reflect data you expect to get in future and consider important to do well on.

And, in particular, the dev set and the test set here, should come from the same distribution.

So, whatever type of data you expect to get in the future, and once you do well on, try to get data that looks like that.

And, whatever that data is, put it into both your dev set and your test set.

Because that way, you're putting the target where you actually want to hit and you're having the team innovate very efficiently to hitting that same target, hopefully, the same targets well.

Since we haven't talked yet about how to set up a training set, we'll talk about the training set in a later video.

But, the important take away from this video is that, setting up the dev set, as well as the validation metric, is really defining what target you want to aim at.

And hopefully, by setting the dev set and the test set to the same distribution, you're really aiming at whatever target you hope your machine learning team will hit.

The way you choose your training set will affect how well you can actually hit that target.

But, we can talk about that separately in a later video.

So, I know some machine learning teams that could literally have saved themselves months of work could they follow the guidelines in this video.

So, I hope these guidelines will help you, too.

Next, it turns out, that the size of your dev and test sets, how to choose the size of them, is also changing the area of deep learning.

Let's talk about that in the next video.