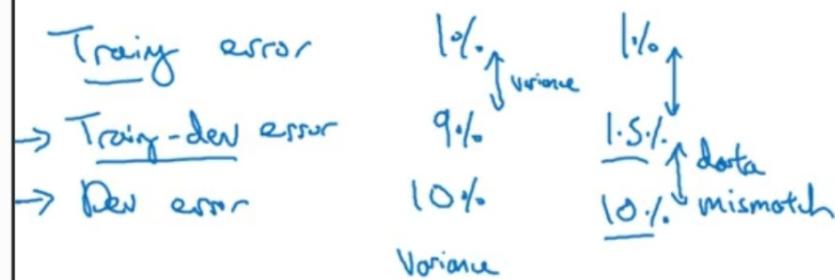
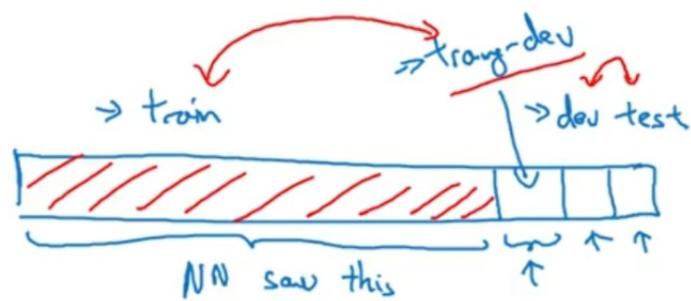


Build your first system quickly, then iterate

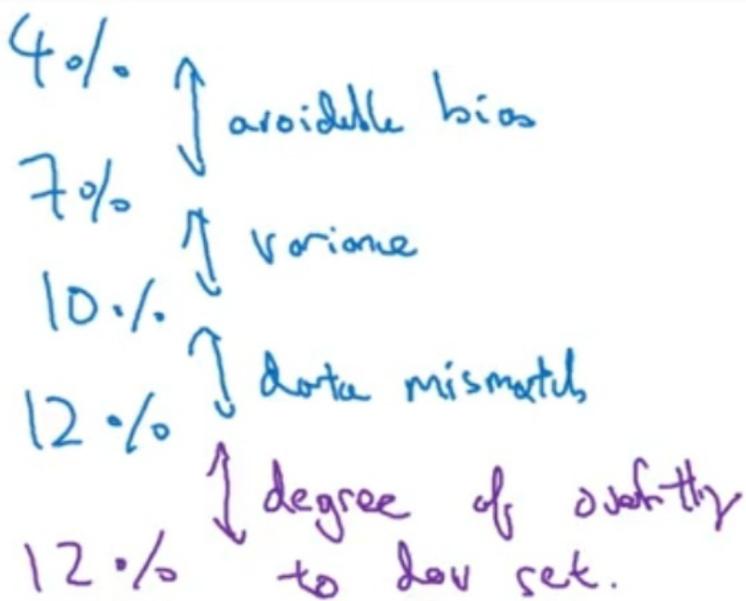
- Set up dev/test set and metric
- Build initial system quickly
- Use Bias/Variance analysis & Error analysis to prioritize next steps.

Training-dev set: Same distribution as training set, but not used for training



Human level

- Training set error
Training-dev set error
→ Dev error
→ Test error



When transfer learning makes sense

Transfer from A \rightarrow B

- Task A and B have the same input x.
- You have a lot more data for Task A than Task B.
- Low level features from A could be helpful for learning B.

When multi-task learning makes sense

- Training on a set of tasks that could benefit from having shared lower-level features.
- Usually: Amount of data you have for each task is quite similar.

So to summarize, multi-task learning enables you to train one neural network to do many tasks and this can give you better performance than if you were to do the tasks in isolation. Now one note of caution, in practice I see that transfer learning is used much more often than multi-task learning.

So I do see a lot of tasks where if you want to solve a machine learning problem but you have a relatively small data set, then transfer learning can really help.

Where if you find a related problem but you have a much bigger data set, you can train in your neural network from there and [then transfer it to the problem where we have very low data](#). So transfer learning is used a lot today.

There are some applications of transfer multi-task learning as well, but multi-task learning I think is used much less often than transfer learning.

And maybe the one exception is computer vision object detection, where I do see a lot of applications of training a neural network to detect lots of different objects. And that works better than training separate neural networks and detecting the visual objects.

But on average I think that even though transfer learning and multi-task learning often you're presented in a similar way, in practice I've seen a lot more applications of transfer learning than of multi-task learning.

I think because often 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. So that's it for multi-task learning.