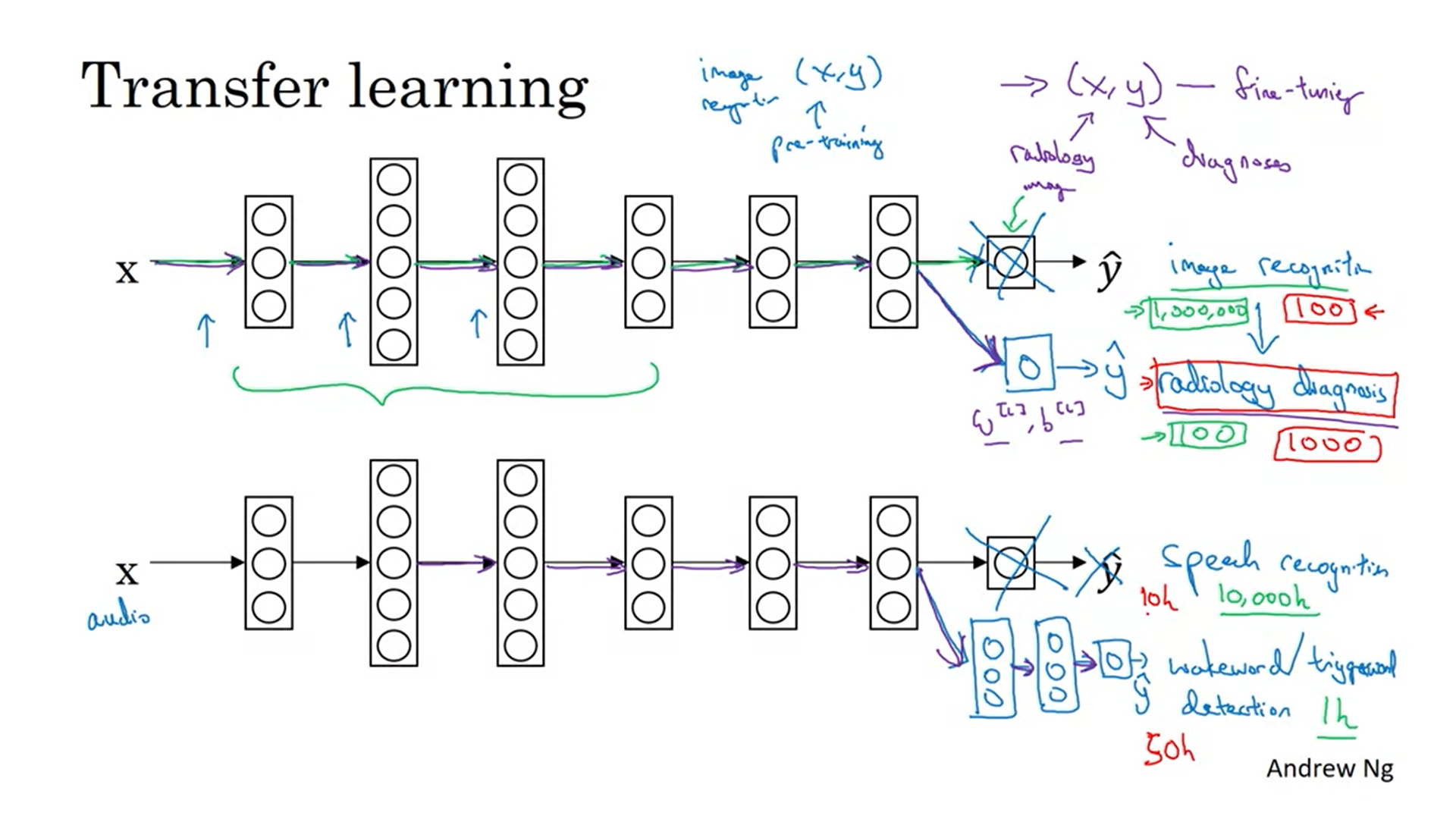


One of the most powerful ideas in deep learning is to take knowledge learned from one task for another task (e.g. cat recognition -> x-ray images)

Delete the last layer including weights. Add a new last layer. Initialize the last layer’s weights randomly. Train the last layer on the new data (radiology data).

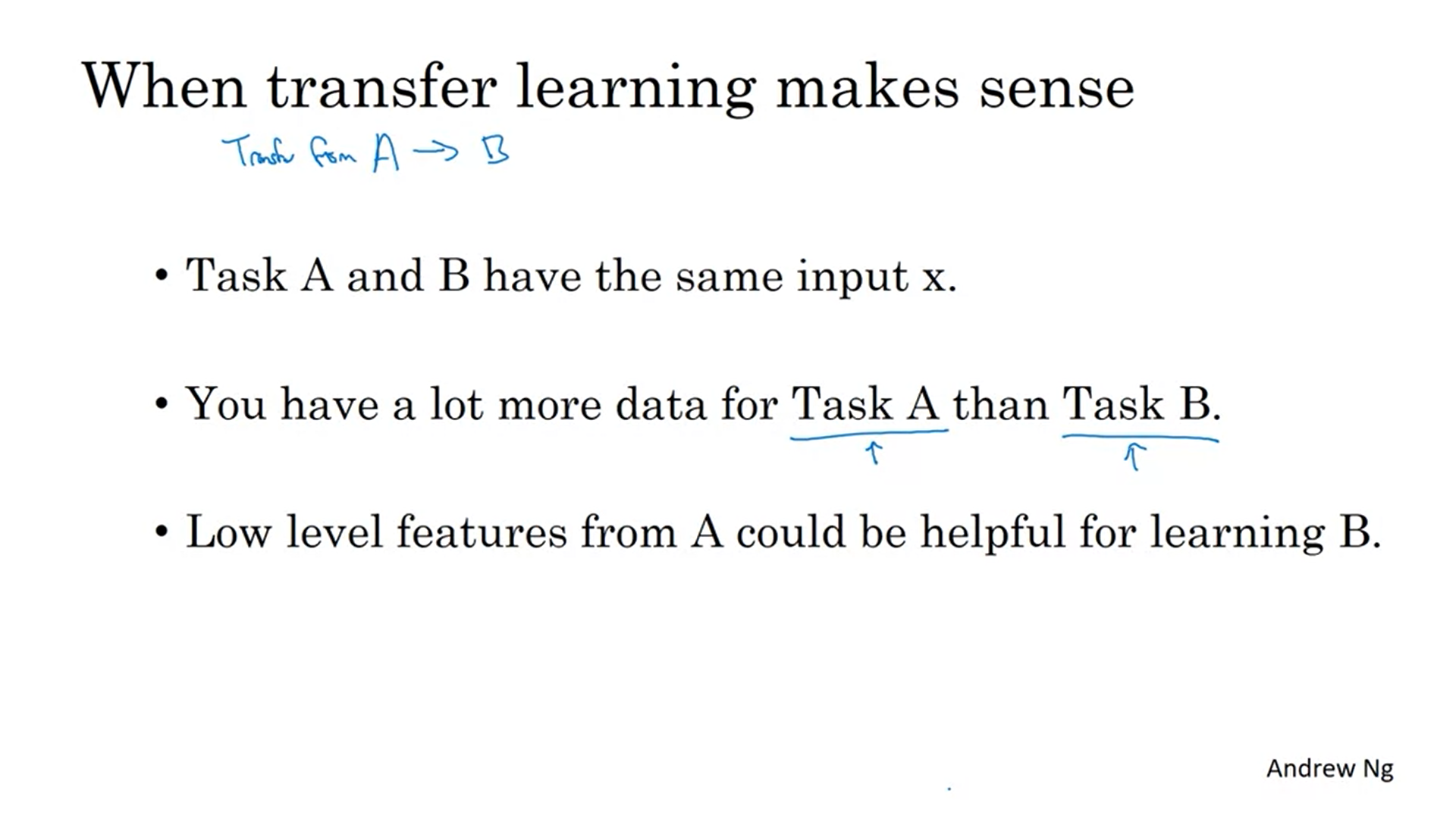
If you have a big dataset you can retrain all layers. With a small dataset just retrain the last layer(s).

Pretraining and fine-tuning.

This can be helpful because a lot of the lower level features (like edges, lines, curves) will transfer and be useful for other tasks.

Another example is a speech recognition system transferred to a wakeword detection system. Take out the last layer and create several new layers. Depending on the amount of data you have you might just retrain the new layers or the entire network.

Transfer learning helps you get going when you do not have a lot of data.

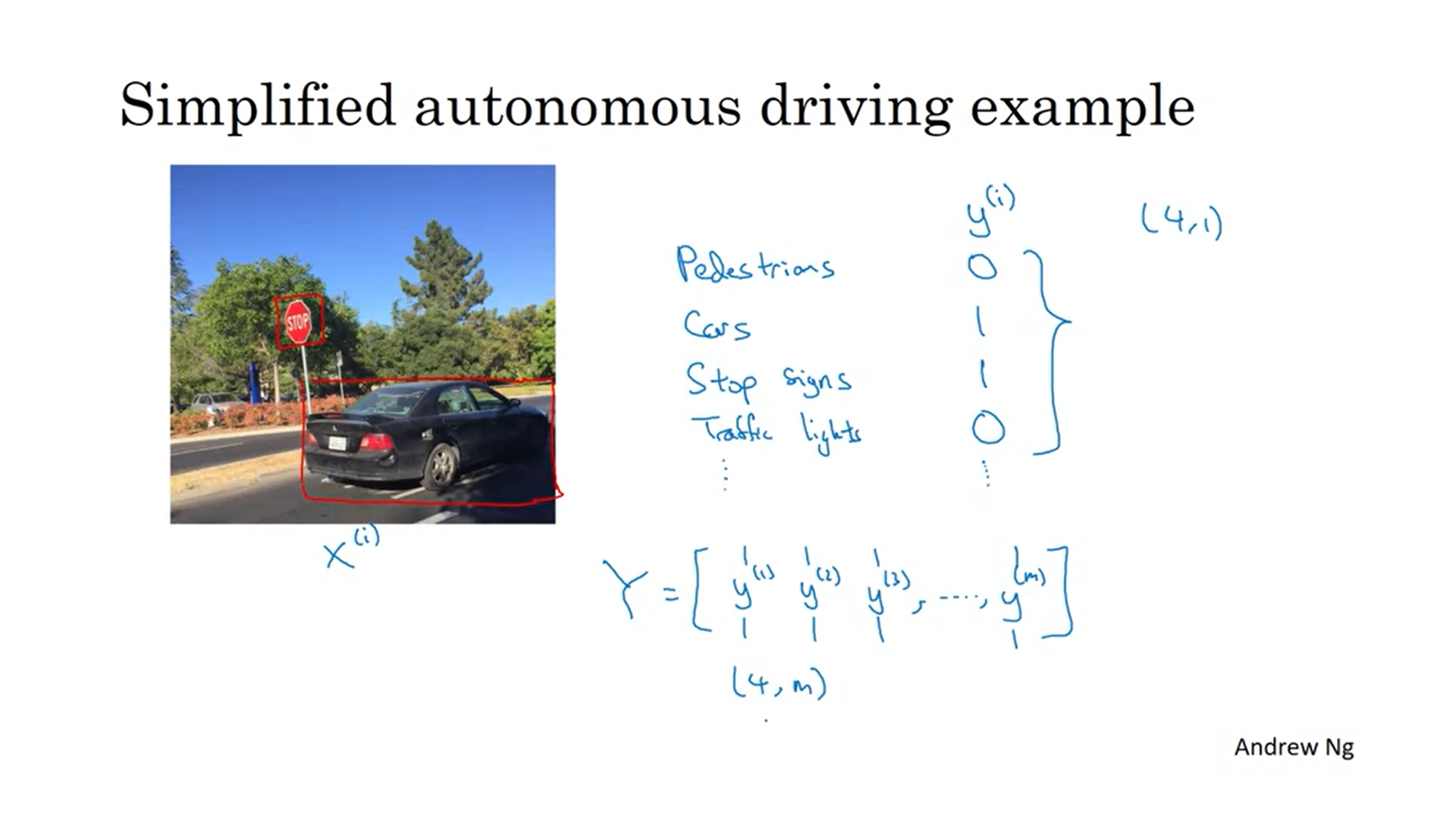


It does not make sense when you have a lot more data for Task B than Task A. The value of one example for your specific task is still much more valuable. Thus, in order to get gains you need a lot more data from the task that you are not interested in to benefit from it.

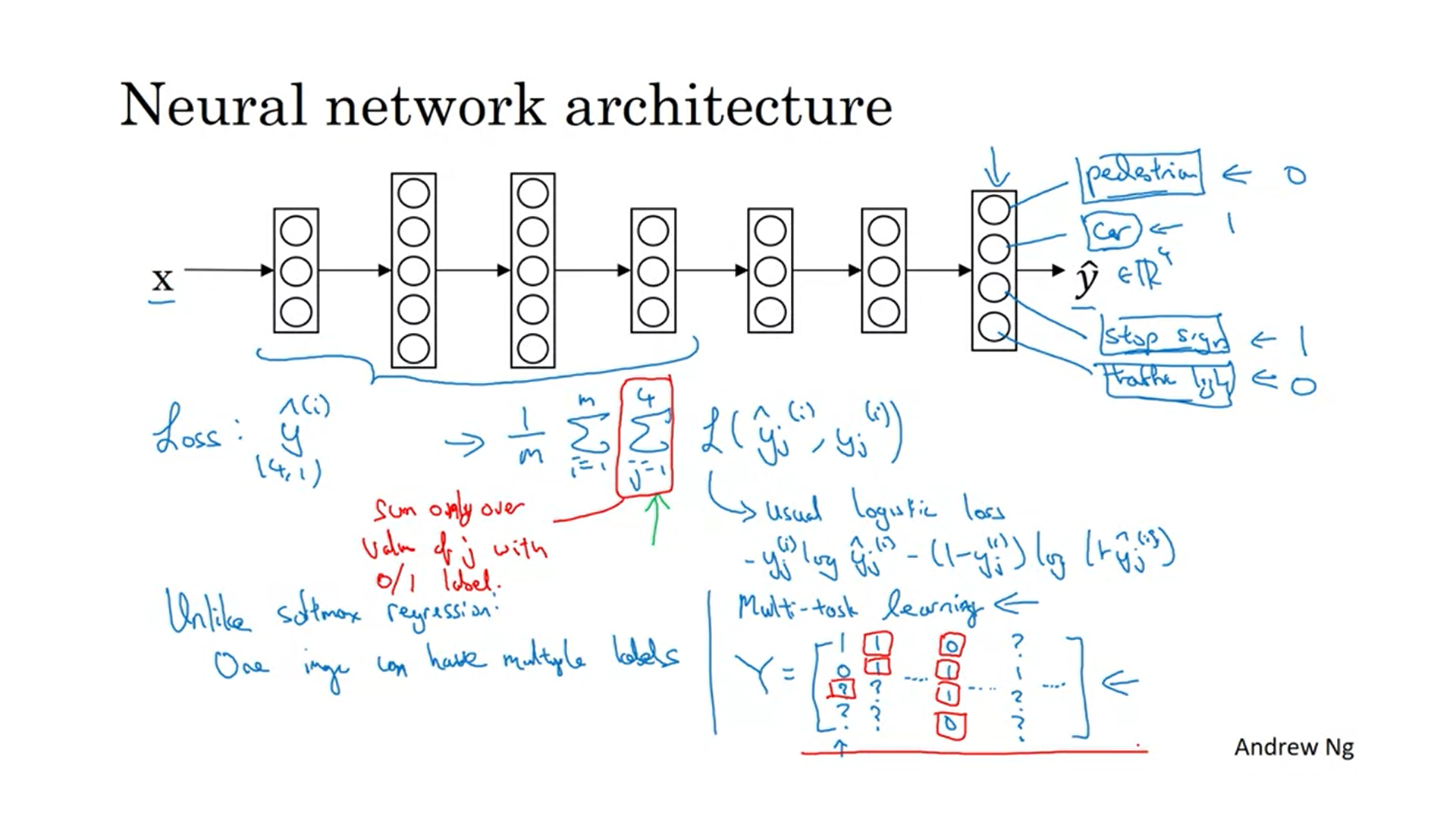
Transfer learning is really useful if it is hard to collect data for your specific task.



In transfer learning you have a sequential process. In multi-task learning you have a simultaneous process. Each of these tasks hopefully helps all of the other tasks.



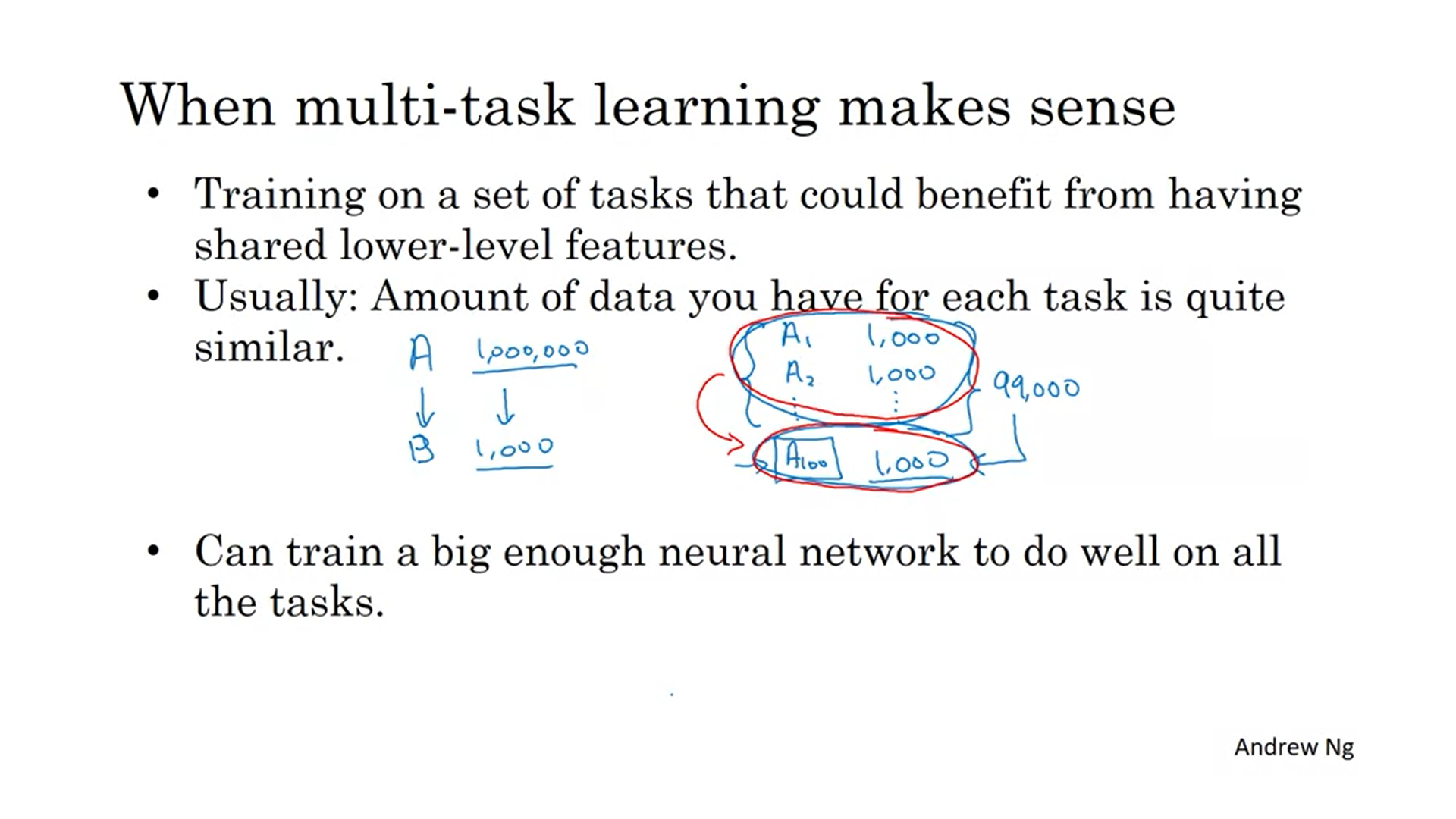
In autonomous driving you need to detect multiple things.



One image can have multiple labels as opposed to softmax.

You could also have trained 4 different neural nets but like this the initial features are shared. This can perform better.

Multitask learning also works if not everything is labelled fully.



In the example these were all features of roads.

For A100 in isolation, you would only have 1000 examples. With multitask learning you have many more. It needs to be the case that the other tasks in aggregation have much more data!

If the network is not big enough, multitask learning can also hurt performance.

In practice transfer learning is used much more than multitask learning. The best scenario for multitask learning is computer vision (this might be the only exception where multitask learning is more popular than transfer learning).