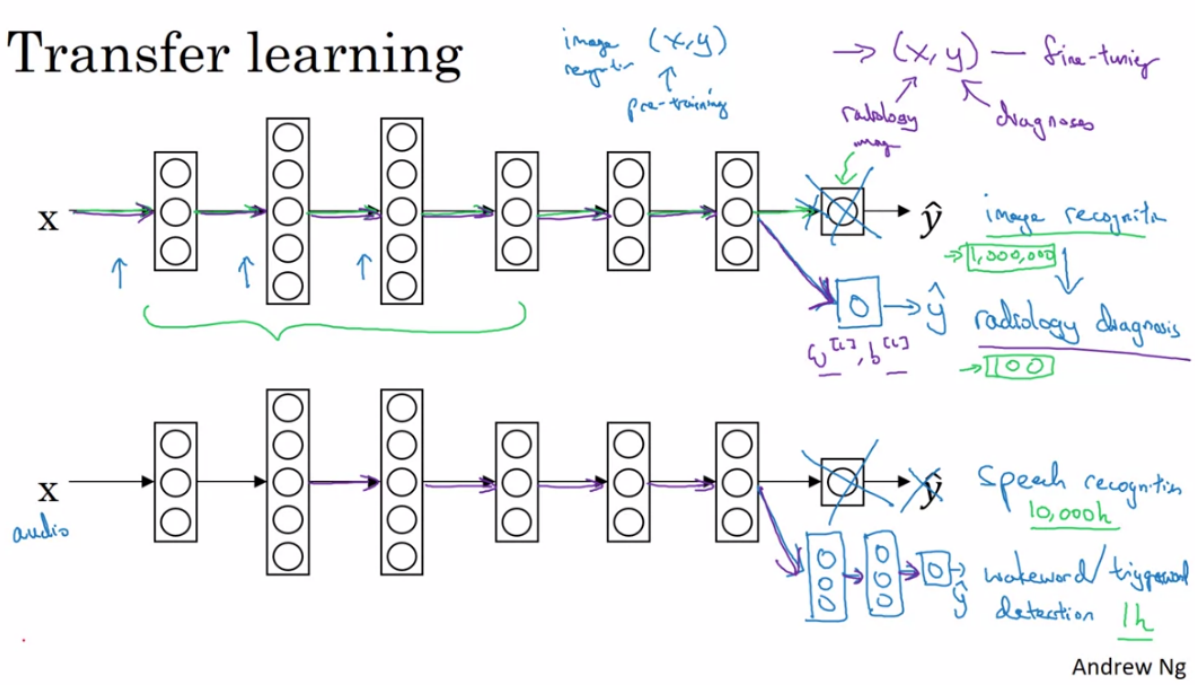
Week 2-2 Transfer Learning & Multi-task Learning

笔记本:	DL 3 - Structuring ML Projects		
创建时间:	2021/1/11 14:17	**更新时间:**	2021/1/11 14:32

Transfer learning makes sense when you have a lot of data for the problem you're transferring from and usually relatively less data for the problem you're transferring to.



Andrew Ng

And the rule of thumb is maybe if you have a small data set, then just retrain the one last layer at the output layer. Or maybe that last one or two layers. But if you have a lot of data, then maybe you can retrain all the parameters in the network.

And if you retrain **all the parameters** in the neural network, then this initial phase of training on image recognition is sometimes called **pre-training**, because you're using image recognitions data to pre-initialize or really pre-train the weights of the neural network. And then if you are updating all the weights afterwards, then training on the radiology data sometimes that's called f**ine tuning**

And the reason this can be helpful is that a lot of the **low level features** such as detecting edges, detecting curves, detecting positive objects

When transfer learning makes sense

Transfer from A → 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.

Multi-task Learning

In multi-task learning, you start off **simultaneously**, trying to have one neural network **do several things at the same time**. And then each of these task **helps** hopefully **all of the other task.**

Simplified autonomous driving example

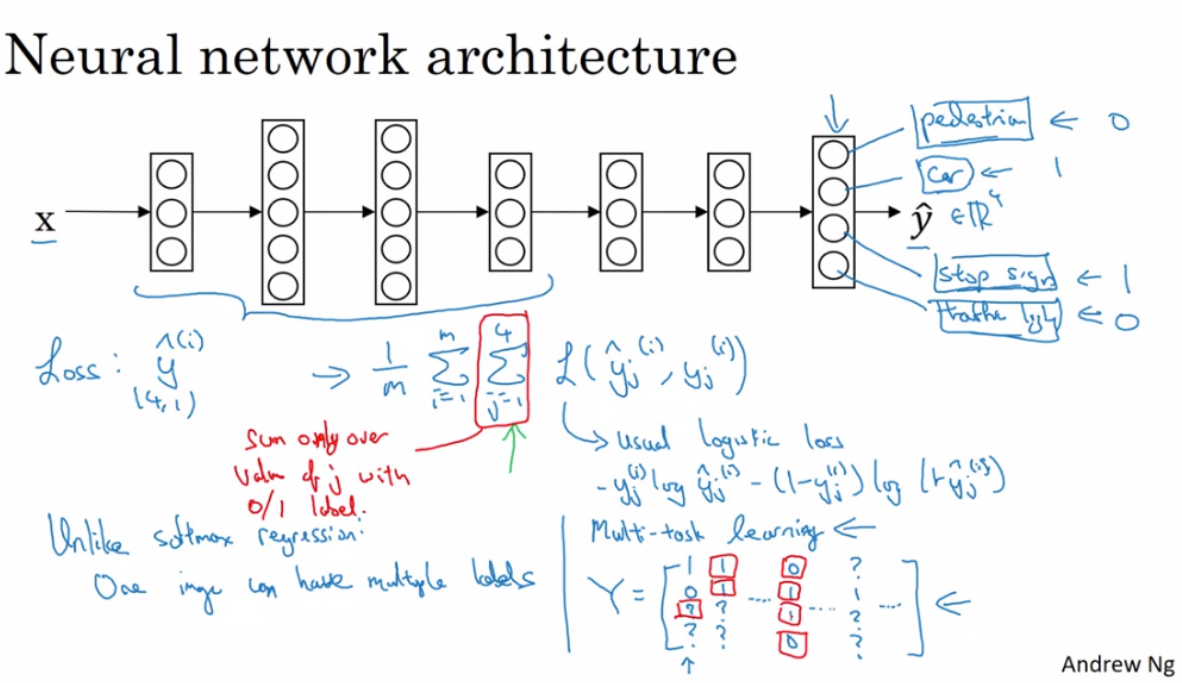
$x^{(i)}$

$y^{(i)}$ (4,1)

Pedestrians 0
Cars 1
Stop signs 1
Traffic lights 0

$Y = [\, y^{(1)} \; y^{(2)} \; y^{(3)} \; \cdots \; y^{(m)} \,]$

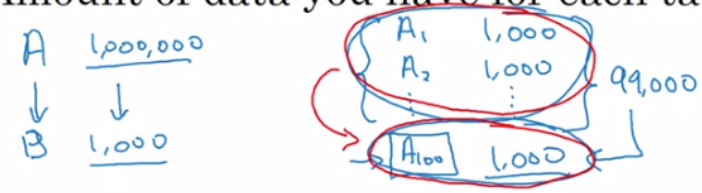
Difference with softmax -> one image can have multiple labels



And one other thing you could have done is just train four separate neural networks, instead of train one network to do four things. But if some of the **earlier features** in neural network can be **shared** between these different types of objects, then you find that training one neural network to do four things results in **better performance** than training **four completely separate** neural networks to do the four tasks separately.

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.



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

So what a researcher, **Rich Caruana** https://link.springer.com/article/10.1023/A:100737960 found many years ago was that the only times multi-task learning hurts performance compared to training separate neural networks is if your neural network **isn't big enough.** But if you can train a big enough neural network, then multi-task learning certainly should not or should very rarely hurt performance. And hopefully it will actually **help performance** compared to if you were training neural networks to do these different tasks in isolation.

But I would say that on average transfer learning is used much more today than multi-task learning (CV). 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