Transfer learning

(SPEECH)
One

(DESCRIPTION)
Text, Learning from multiple tasks. Transfer learning.

(SPEECH)
of the most powerful ideas in deep learning is that sometimes you can take knowledge the neural network has learned from one task and apply that knowledge to a separate task.

So for example, maybe you could have the neural network learn to recognize objects like cats and then use that knowledge or use part of that knowledge to help you do a better job reading x-ray scans.

This is called transfer learning. Let's take a look.

(DESCRIPTION)
New slide, Transfer learning. Arrows move horizontally from an X plane through a series of seven rectangles, with circles inside. On the other side is a Y variable. The number of circles in the rectangles, in order, are, three, five, five, three, three, three, and one.

(SPEECH)
Let's say you've trained your neural network on image recognition.

So you first take a neural network and train it on X Y pairs, where X is an image and Y is some object.

An image is a cat or a dog or a bird or something else.

If you want to take this neural network and adapt, or we say transfer, what is learned to a different task, such as radiology diagnosis, meaning really reading X-ray scans, what you can do is take this last output layer of the neural network and just delete that and delete also the weights feeding into that last output layer

(DESCRIPTION)
An X is drawn through the final, single circle, rectangle.

(SPEECH)
and create a new set of randomly initialized weights just for the last layer and have that now output radiology diagnosis.

So to be concrete, during the first phase of training when you're training on an image recognition task, you train all of the usual parameters for the neural network, all the weights, all the layers and you have something that now learns to make image recognition predictions.

Having trained that neural network, what you now do to implement transfer learning is swap in a new data set X Y, where now these are radiology images.

And Y are the diagnoses you want to predict and what you do is initialize the last layers' weights.

Let's call that W.L.

and P.L. randomly.

And now, retrain the neural network on this new data set, on the new radiology data set.

You have a couple options of how you retrain neural network with radiology data.

You might, if you have a small radiology dataset, you might want to just retrain the weights of the last layer, just W.L.

P.L., and keep the rest of the parameters fixed.

If you have enough data, you could also retrain all the layers of the rest of the neural network.

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 fine tuning.

So you hear the words pre-training and fine tuning in a deep learning context, this is what they mean when they refer to pre-training and fine tuning weights in a transfer learning source.

And what you've done in this example, is you've taken knowledge learned from image recognition and applied it or transferred it to radiology diagnosis.

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.

Learning from that, from a very large image recognition data set, might help your learning algorithm do better in radiology diagnosis.

It's just learned a lot about the structure and the nature of how images look like and some of that knowledge will be useful.

So having learned to recognize images, it might have learned enough about you know, just what parts of different images look like, that that knowledge about lines, dots, curves, and so on, maybe small parts of objects, that knowledge could help your radiology diagnosis network learn a bit faster or learn with less data.

Here's

(DESCRIPTION)
A second, identical, set of planes with rectangles appears underneath the first example.

(SPEECH)
another example.

Let's say that you've trained a speech recognition system so now X is input of audio or audio snippets, and Y is some ink transcript.

So you've trained in speech recognition system to output your transcripts.

And let's say that you now want to build a "wake words" or a "trigger words" detection system.

So, recall that a wake word or the trigger word are the words we say in order to wake up speech control devices in our houses such as saying "Alexa" to wake up an Amazon Echo or "OK Google" to wake up a Google device or "hey Siri" to wake up an Apple device or saying "Ni hao baidu" to wake up a baidu device.

So in order to do this, you might take out the last layer of the neural network again and create a new output node.

But sometimes another thing you could do is actually create not just a single new output, but actually create several new layers to your neural network to try to put the labels Y for your wake word detection problem.

Then again, depending on how much data you have, you might just retrain the new layers of the network or maybe you could retrain even more layers of this neural network.

So, when does transfer learning make sense?

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.

So for example, let's say you have a million examples for image recognition task.

So that's a lot of data to learn a lot of low level features or to learn a lot of useful features in the earlier layers in neural network.

But for the radiology task, maybe you have only a hundred examples.

So you have very low data for the radiology diagnosis problem, maybe only 100 x-ray scans.

So a lot of knowledge you learn from English recognition can be transferred and can really help you get going with radiology recognition even if you don't have all the data for radiology.

For speech recognition, maybe you've trained the speech recognition system on 10000 hours of data.

So, you've learned a lot about what human voices sounds like from that 10000 hours of data, which really is a lot.

But for your trigger word detection, maybe you have only one hour of data.

So, that's not a lot of data to fit a lot of parameters.

So in this case, a lot of what you learn about what human voices sound like, what are components of human speech and so on, that can be really helpful for building a good wake word detector, even though you have a relatively small dataset or at least a much smaller dataset for the wake word detection task.

So in both of these cases, you're transferring from a problem with a lot of data to a problem with relatively little data.

One case where transfer learning would not make sense, is if the opposite was true.

So, if you had a hundred images for image recognition and you had 100 images for radiology diagnosis or even a thousand images for radiology diagnosis, one would think about it is that to do well on radiology diagnosis, assuming what you really want to do well on this radiology diagnosis, having radiology images is much more valuable than having cat and dog and so on images.

So each example here is much more valuable than each example there, at least for the purpose of building a good radiology system.

So, if you already have more data for radiology, it's not that likely that having 100 images of your random objects of cats and dogs and cars and so on will be that helpful, because the value of one example of image from your image recognition task of cats and dogs is just less valuable than one example of an x-ray image for the task of building a good radiology system.

So, this would be one example where transfer learning, well, it might not hurt but I wouldn't expect it to give you any meaningful gain either.

And similarly, if you'd built a speech recognition system on 10 hours of data and you actually have 10 hours or maybe even more, say 50 hours of data for wake word detection, you know it won't, it may or may not hurt, maybe it won't hurt to include that 10 hours of data to your transfer learning, but you just wouldn't expect to get a meaningful gain.

(DESCRIPTION)
New slide, When transfer learning makes sense.

(SPEECH)
So to summarize, when does transfer learning make sense?

If you're trying to learn from some Task A and transfer some of the knowledge to some Task B, then transfer learning makes sense when Task A and B have the same input X.

In the first example, A and B both have images as input.

In the second example, both have audio clips as input.

It has to make sense when you have a lot more data for Task A than for Task B.

All this is under the assumption that what you really want to do well on is Task B.

And because data for Task B is more valuable for Task B, usually you just need a lot more data for Task A because you know, each example from Task A is just less valuable for Task B than each example for Task B.

And then finally, transfer learning will tend to make more sense if you suspect that low level features from Task A could be helpful for learning Task B.

And in both of the earlier examples, maybe learning image recognition teaches you enough about images to have a radiology diagnosis and maybe learning speech recognition teaches you about human speech to help you with trigger word or wake word detection.

So to summarize, transfer learning has been most useful if you're trying to do well on some Task B, usually a problem where you have relatively little data.

So for example, in radiology, you know it's difficult to get that many x-ray scans to build a good radiology diagnosis system.

So in that case, you might find a related but different task, such as image recognition, where you can get maybe a million images and learn a lot of load-over features from that, so that you can then try to do well on Task B on your radiology task despite not having that much data for it.

When transfer learning makes sense?

It does help the performance of your learning task significantly.

But I've also seen sometimes seen transfer learning applied in settings where Task A actually has less data than Task B and in those cases, you kind of don't expect to see much of a gain.

So, that's it for transfer learning where you learn from one task and try to transfer to a different task.

There's another version of learning from multiple tasks which is called multitask learning, which is when you try to learn from multiple tasks at the same time rather than learning from one and then sequentially, or after that, trying to transfer to a different task.

So in the next video, let's discuss multitasking learning.