What is end-to-end deep learning?

(SPEECH)
One

(DESCRIPTION)
Text, End-to-end deep learning. What is end-to-end deep learning.

(SPEECH)
of the most exciting recent developments in deep learning, has been the rise of end-to-end deep learning.

So what is the end-to-end learning?

Briefly, there have been some data processing systems, or learning systems that require multiple stages of processing.

And what end-to-end deep learning does, is it can take all those multiple stages, and replace it usually with just a single neural network.

Let's look at

(DESCRIPTION)
New slide, What is end-to-end learning? Speech recognition example.

(SPEECH)
some examples.

Take speech recognition as an example, where your goal is to take an input X such an audio clip, and map it to an output Y, which is a transcript of the audio clip.

So traditionally, speech recognition required many stages of processing.

First, you will extract some features, some hand-designed features of the audio.

So if you've heard of MFCC, that's an algorithm for extracting a certain set of hand designed features for audio.

And then having extracted some low level features, you might apply a machine learning algorithm, to find the phonemes in the audio clip.

So phonemes are the basic units of sound.

So for example, the word cat is made out of three sounds.

The Cu- Ah- and Tu- so they extract those.

And then you string together phonemes to form individual words.

And then you string those together to form the transcripts of the audio clip.

So, in contrast to this pipeline with a lot of stages, what end-to-end deep learning does, is you can train a huge neural network to just input the audio clip, and have it directly output the transcript.

One interesting sociological effect in AI is that as end-to-end deep learning started to work better, there were some researchers that had for example spent many years of their career designing individual steps of the pipeline.

So there were some researchers in different disciplines not just in speech recognition.

Maybe in computer vision, and other areas as well, that had spent a lot of time you know, written multiple papers, maybe even built a large part of their career, engineering featuresor engineering other pieces of the pipeline.

And when end-to-end deep learning just took the last training set and learned the function mapping from x and y directly, really bypassing a lot of these intermediate steps, it was challenging for some disciplines to come around to accepting this alternative way of building AI systems.

Because it really obsoleted in some cases, many years of research in some of the intermediate components.

It turns out that one of the challenges of end-to-end deep learning is that you might need a lot of data before it works well.

So for example, if you're training on 3,000 hours of data to build a speech recognition system, then the traditional pipeline, the full traditional pipeline works really well.

It's only when you have a very large data set, you know one to say 10,000 hours of data, anything going up to maybe 100,000 hours of data that the end-to end-approach then suddenly starts to work really well.

So when you have a smaller data set, the more traditional pipeline approach actually works just as well.

Often works even better.

And you need a large data set before the end-to-end approach really shines.

And if you have a medium amount of data, then there are also intermediate approaches where maybe you input audio and bypass the features and just learn to output the phonemes of the neural network, and then at some other stages as well.

So this will be a step toward end-to-end learning, but not all the way there.

(DESCRIPTION)
New slide, Face recognition. A photo is presented of two men observing a monitor. The image is courtesy of Baidu.

(SPEECH)
Test. So this is a picture of a face recognition turnstile built by a researcher, Yuanqing Lin at Baidu, where this is a camera and it looks at the person approaching the gate, and if it recognizes the person then, you know the turnstile automatically lets them through.

So rather than needing to swipe an RFID badge to enter this facility, in increasingly many offices in China and hopefully more and more in other countries as well, you can just approach the turnstile and if it recognizes your face it just lets you through without needing you to carry an RFID badge.

So, how do you build a system like this?

Well, one thing you could do is just look at the image that the camera is capturing.

Right? So, I guess this is my bad drawing, but maybe this is a camera image.

And you know, you have someone approaching the turnstile.

So this might be the image X that you that your camera is capturing.

And one thing you could do is try to learn a function mapping directly from the image X to the identity of the person Y.

It turns out this is not the best approach.

And one of the problems is that you know, the person approaching the turnstile can approach from lots of different directions.

So they could be green positions, they could be in blue position.

You know, sometimes they're closer to the camera, so they appear bigger in the image.

And sometimes they're already closer to the camera, so that face appears much bigger.

So what it has actually done to build these turnstiles, is not to just take the raw image and feed it to a neural net to try to figure out a person's identity.

Instead, the best approach to date, seems to be a multi-step approach, where first, you run one piece of software to detect the person's face.

So this first detector to figure out where's the person's face.

Having detected the person's face, you then zoom in to that part of the image and crop that image so that the person's face is centered.

Then, it is this picture that I guess I drew here in red, this is then fed to the neural network, to then try to learn, or estimate the person's identity.

And what researchers have found, is that instead of trying to learn everything on one step, by breaking this problem down into two simpler steps, first is figure out where is the face.

And second, is look at the face and figure out who this actually is.

This second approach allows the learning algorithm or really two learning algorithms to solve two much simpler tasks and results in overall better performance.

By the way, if you want to know how the second step actually works I've simplified the discussion.

By the way, if you want to know how step two here actually works, I've actually simplified the description a bit.

The way the second step is actually trained, as you train in your network, that takes as input two images, and what then your network does is it takes this input two images and it tells you if these two are the same person

(DESCRIPTION)
Two boxes of faces are drawn as part of the branching check by the program. The written note questions if the two boxes are the same person.

(SPEECH)
or not.

So if you then have say 10,000 employees IDs on file, you can then take this image in red, and quickly compare it against maybe all 10,000 employee IDs on file to try to figure out if this picture in red is indeed one of your 10000 employees that you should allow into this facility or that should allow into your office building.

This is a turnstile that is giving employees access to a workplace.So why is it that the two step approach works better?

There are actually two reasons for that.

One is that each of the two problems you're solving is actually much simpler.

But second, is that you have a lot of data for each of the two sub-tasks.

In particular, there is a lot of data you can obtain for phase detection, for task one over here, where the task is to look at an image and figure out where is the person's face and the image.

So there is a lot of data.

There is a lot of label data X, comma Y where X is a picture and y shows the position of the person's face.

So you could build a neural network to do task one quite well.

And then separately, there's a lot of data for task two as well.

Today, leading companies have let's say, hundreds of millions of pictures of people's faces.

So given a closely cropped image, like this red image or this one down here, today leading face recognition teams have at least hundreds of millions of images that they could use to look at two images and try to figure out the identity or to figure out if it's the same person or not.

So there's also a lot of data for task two.

But in contrast, if you were to try to learn everything at the same time, there is much less data of the form X comma Y.

Where X is image like this taken from the turnstile, and Y is the identity of the person.

So because you don't have enough data to solve this end-to-end learning problem, but you do have enough data to solve sub-problems one and two, in practice, breaking this down to two sub-problems results in better performance than a pure end-to-end deep learning approach.

Although if you had enough data for the end-to-end approach, maybe the end-to-end approach would work better, but that's not actually what works best in practice today.

Let's

(DESCRIPTION)
New slide, More examples.

(SPEECH)
look at a few more examples.

Take machine translation.

Traditionally, machine translation systems also had a long complicated pipeline, where you first take say English, text and then do text analysis.

Basically, extract a bunch of features off the text, and so on.

And after many many steps you'd end up with say, a translation of the English text into French.

Because, for machine translation, you do have a lot of pairs of English comma French sentences.

End-to-end deep learning works quite well for machine translation.

And that's because today, it is possible to gather large data sets of X-Y pairs where that's the English sentence and that's the corresponding French translation.

So in this example, end-to-end deep learning works

(DESCRIPTION)
An xray of a human hand is inserted.

(SPEECH)
well.

One last example, let's say that you want to look at an X-ray picture of a hand of a child, and estimate the age of a child.

You know, when I first heard about this problem, I thought this is a very cool crime scene investigation task where you find maybe tragically the skeleton of a child, and you want to figure out how the child was.

It turns out that typical application of this problem, estimating age of a child from an X-ray is less dramatic than this crime scene investigation I was picturing.

It turns out that pediatricians use this to estimate whether or not a child is growing or developing normally.

But a non end-to-end approach to this, would be you locate an image and then you segment out or recognize the bones.

So, just try to figure out where is that bone segment?

Where is that bone segment?

Where is that bone segment? And so on. And then.

Knowing the lengths of the different bones, you can sort of go to a look up table showing the average bone lengths in a child's hand and then use that to estimate the child's age.

And so this approach actually works pretty well.

In contrast, if you were to go straight from the image to the child's age, then you would need a lot of data to do that directly and as far as I know, this approach does not work as well today just because there isn't enough data to train this task in an end-to-end fashion.

Whereas in contrast, you can imagine that by breaking down this problem into two steps.

Step one is a relatively simple problem.

Maybe you don't need that much data.

Maybe you don't need that many X-ray images to segment out the bones.

And task two, by collecting statistics of a number of children's hands, you can also get decent estimates of that without too much data.

So this multi-step approach seems promising.

Maybe more promising than the end-to-end approach, at least until you can get more data for the end-to-end learning approach.

So an end-to-end deep learning works.

It can work really well and it can really simplify the system and not require you to build so many hand-designed individual components.

But it's also not panacea, it doesn't always work.

In the next video, I want to share with you a more systematic description of when you should, and maybe when you shouldn't use end-to-end deep learning and how to piece together these complex machine learning systems.