

Why is Deep Learning taking off?

(DESCRIPTION)

Website, deep learning, dot, A.I.

(SPEECH)

if the basic technical idea is behind deep learning behind your networks have been around for decades why are they only just now taking off in this video let's go over some of the main drivers behind the rise of deep learning because I think this will help you that the spot the best opportunities within your own organization to apply these to over

(DESCRIPTION)

New slide, Scale drives deep learning progress. A right angle graph is displayed. The X plane is labeled, amount of data. The Y plane is labeled, performance.

(SPEECH)

the last few years a lot of people have asked me Andrew why is deep learning certainly working so well and when a marsan question this is usually the picture I draw for them let's say we plot a figure where on the horizontal axis we plot the amount of data we have for a task and let's say on the vertical axis we plot the performance on above learning algorithms such as the accuracy of our spam classifier or our ad click predictor or the accuracy of our neural net for figuring out the position of other calls for our self-driving car it turns out if you plot the performance of a traditional learning algorithm like support vector machine or logistic regression as a function of the amount of data you have you might get a curve that looks like this where the performance improves for a while as you add more data but after a while the performance you know pretty much plateaus right suppose your horizontal lines enjoy that very well you know was it they didn't know what to do with huge amounts of data and what happened in our society over the last 10 years maybe is that for a lot of problems we went from having a relatively small amount of data to having you know often a fairly large amount of data and all of this was thanks to the digitization of a society where so much human activity is now in the digital realm we spend so much time on the computers on websites on mobile apps and activities on digital devices creates data and thanks to the rise of inexpensive cameras built into our cell phones accelerometers all sorts of sensors in the Internet of Things we also just have been collecting one more and more data so over the last 20 years for a lot of applications we just accumulate a lot more data more than traditional learning algorithms were able to effectively take advantage of and what new network lead turns out that if you train a small neural net then this performance maybe looks like that if you train a somewhat larger Internet that's called as a medium-sized internet to fall in something a little bit better and if you train a very large neural net then it's the form and often just keeps getting better and better so couple observations one is if you want to hit this very high level of performance then you need two things first often you need to be able to train a big enough neural network in order to take advantage of the huge amount of data and second you need to be out here on the x axes you do need a lot of data so we often say that scale has been driving deep learning progress and by scale I mean both the size of the neural

network we need just a new network a lot of hidden units a lot of parameters a lot of connections as well as scale of the data in fact today one of the most reliable ways to get better performance in the neural network is often to either train a bigger network or throw more data at it and that only works up to a point because eventually you run out of data or eventually then your network is so big that it takes too long to train but just improving scale has actually taken us a long way in the world of learning in order to make this diagram a bit more technically precise and just add a few more things I wrote the amount of data on the x-axis technically this is amount of labeled data where by label data I mean training examples we have both the input X and the label Y I went to introduce a little bit of notation that we'll use later in this course we're going to use lowercase alphabet to denote the size of my training sets or the number of training examples this lowercase M so that's the horizontal axis couple other details to this Tigger in this regime of smaller training sets the relative ordering of the algorithms is actually not very well defined so if you don't have a lot of training data is often up to your skill at hand engineering features that determines the foreman so it's quite possible that if someone training an SVM is more motivated to hand engineer features and someone training even large their own that may be in this small training set regime the SEM could do better so you know in this region to the left of the figure the relative ordering between gene algorithms is not that well defined and performance depends much more on your skill at engine features and other mobile details of the algorithms and there's only in this some big data regime very large training sets very large M regime in the right that we more consistently see largely Ronettes dominating the other approaches and so if any of your friends ask you why are known as you know taking off I would encourage you to draw this picture for them as well so I will say that in the early days in their modern rise of deep learning it was scaled data and scale of computation just our ability to Train very large dinner networks either on a CPU or GPU that enabled us to make a lot of progress but increasingly especially in the last several years we've seen tremendous algorithmic innovation as well so I also don't want to understate that interestingly many of the algorithmic innovations have been about trying to make neural networks run much faster so as a concrete example one of the huge breakthroughs in your networks has been switching from a sigmoid function which looks like this to a railer function which we talked about briefly in an early video that looks like this if you don't understand the details of one about the state don't worry about it but it turns out that one of the problems of using sigmoid functions and machine learning is that there these regions here where the slope of the function would gradient is nearly zero and so learning becomes really slow because when you implement gradient descent and gradient is zero the parameters just change very slowly and so learning is very slow whereas by changing the what's called the activation function the neural network to use this function called the value function of the rectified linear unit our elu the gradient is equal to one for all positive values of input right and so the gradient is much less likely to gradually shrink to zero and the gradient here the slope of this line is zero on the left but it turns out that just by switching to the sigmoid function to the rayleigh function has made an algorithm called gradient descent work much faster and so this is an example of maybe relatively simple algorithm in Bayesian but ultimately the impact of this algorithmic innovation was it really hope computation so the regimen quite a lot of examples like this of where we change the algorithm because it allows that code to run much faster and this allows us to train bigger neural networks or to do so the reason or multi-client even when we have a large network roam all the data the other reason that fast

computation is important is that it turns out the process of training your network this is very intuitive often you have an idea for a neural network architecture and so you implement your idea and code implementing your idea then lets you run an experiment which tells you how well your neural network does and then by looking at it you go back to change the details of your new network and then you go around this circle over and over and when your new network takes a long time to Train it just takes a long time to go around this cycle and there's a huge difference in your productivity building effective neural networks when you can have an idea and try it and see the work in ten minutes or maybe ammos a day versus if you've to train your neural network for a month which sometimes does happened because you get a result back you know in ten minutes or maybe in a day you should just try a lot more ideas and be much more likely to discover in your network and it works well for your application and so faster computation has really helped in terms of speeding up the rate at which you can get an experimental result back and this has really helped both practitioners of neuro networks as well as researchers working and deep learning iterate much faster and improve your ideas much faster and so all this has also been a huge boon to the entire deep learning research community which has been incredible with just you know inventing new algorithms and making nonstop progress on that front so these are some of the forces powering the rise of deep learning but the good news is that these forces are still working powerfully to make deep learning even better Tech Data society is still throwing up one more digital data or take computation with the rise of specialized hardware like GPUs and faster networking many types of hardware I'm actually quite confident that our ability to do very large neural networks or should a computation point of view will keep on getting better and take algorithms relative learning research communities though continuously phenomenal at innovating on the algorithms front so because of this I think that we can be optimistic answer the optimistic the deep learning will keep on getting better for many years to come so that let's go on to the last video of the section where we'll talk a little bit more about what you learn from this course

Why is Deep Learning taking off? (DESCRIPTION) Website, deep learning, dot, A.I. (SPEECH) if the basic technical idea is behind deep learning behind your networks have been around for decades why are they only just now taking off in this video let's go over some of the main drivers behind the rise of deep learning because I think this will help you that the spot the best opportunities within your own organization to apply these to over (DESCRIPTION) New slide, Scale drives deep learning progress. A right angle graph is displayed. The X plane is labeled, amount of data. The Y plane is labeled, performance. (SPEECH) the last few years a lot of people have asked me Andrew why is deep learning certainly working so well and when a marsan question this is usually the picture I draw for them let's say we plot a figure where on the horizontal axis we plot the amount of data we have for a task and let's say on the vertical axis we plot the performance on above learning algorithms such as the accuracy of our spam classifier or our ad click predictor or the accuracy of our neural net for figuring out the position of other calls for our self-driving car it turns out if you plot the performance of a traditional learning algorithm like support vector machine or logistic regression as a function of the amount of data you have you might get a curve that looks like this where the performance improves for a while as you add more data but after a while the performance you know pretty much plateaus right suppose your horizontal lines enjoy that very well you know was it they didn't know what to do

with huge amounts of data and what happened in our society over the last 10 years maybe is that for a lot of problems we went from having a relatively small amount of data to having you know often a fairly large amount of data and all of this was thanks to the digitization of a society where so much human activity is now in the digital realm we spend so much time on the computers on websites on mobile apps and activities on digital devices creates data and thanks to the rise of inexpensive cameras built into our cell phones accelerometers all sorts of sensors in the Internet of Things we also just have been collecting one more and more data so over the last 20 years for a lot of applications we just accumulate a lot more data more than traditional learning algorithms were able to effectively take advantage of and what new network lead turns out that if you train a small neural net then this performance maybe looks like that if you train a somewhat larger Internet that's called as a medium-sized internet to fall in something a little bit better and if you train a very large neural net then it's the form and often just keeps getting better and better so couple observations one is if you want to hit this very high level of performance then you need two things first often you need to be able to train a big enough neural network in order to take advantage of the huge amount of data and second you need to be out here on the x axes you do need a lot of data so we often say that scale has been driving deep learning progress and by scale I mean both the size of the neural network we need just a new network a lot of hidden units a lot of parameters a lot of connections as well as scale of the data in fact today one of the most reliable ways to get better performance in the neural network is often to either train a bigger network or throw more data at it and that only works up to a point because eventually you run out of data or eventually then your network is so big that it takes too long to train but just improving scale has actually taken us a long way in the world of learning in order to make this diagram a bit more technically precise and just add a few more things I wrote the amount of data on the x-axis technically this is amount of labeled data where by label data I mean training examples we have both the input X and the label Y I went to introduce a little bit of notation that we'll use later in this course we're going to use lowercase alphabet to denote the size of my training sets or the number of training examples this lowercase M so that's the horizontal axis couple other details to this Tigger in this regime of smaller training sets the relative ordering of the algorithms is actually not very well defined so if you don't have a lot of training data is often up to your skill at hand engineering features that determines the foreman so it's quite possible that if someone training an SVM is more motivated to hand engineer features and someone training even large their own that may be in this small training set regime the SEM could do better so you know in this region to the left of the figure the relative ordering between gene algorithms is not that well defined and performance depends much more on your skill at engine features and other mobile details of the algorithms and there's only in this some big data regime very large training sets very large M regime in the right that we more consistently see largely Ronettes dominating the other approaches and so if any of your friends ask you why are known as you know taking off I would encourage you to draw this picture for them as well so I will say that in the early days in their modern rise of deep learning it was scaled data and scale of computation just our ability to Train very large dinner networks either on a CPU or GPU that enabled us to make a lot of progress but increasingly especially in the last several years we've seen tremendous algorithmic innovation as well so I also don't want to understate that interestingly many of the algorithmic innovations have been about trying to make neural networks run much faster so as a concrete example one of the huge breakthroughs in

your networks has been switching from a sigmoid function which looks like this to a railer function which we talked about briefly in an early video that looks like this if you don't understand the details of one about the state don't worry about it but it turns out that one of the problems of using sigmoid functions and machine learning is that there these regions here where the slope of the function would gradient is nearly zero and so learning becomes really slow because when you implement gradient descent and gradient is zero the parameters just change very slowly and so learning is very slow whereas by changing the what's called the activation function the neural network to use this function called the value function of the rectified linear unit our elu the gradient is equal to one for all positive values of input right and so the gradient is much less likely to gradually shrink to zero and the gradient here the slope of this line is zero on the left but it turns out that just by switching to the sigmoid function to the rayleigh function has made an algorithm called gradient descent work much faster and so this is an example of maybe relatively simple algorithm in Bayesian but ultimately the impact of this algorithmic innovation was it really hope computation so the regimen quite a lot of examples like this of where we change the algorithm because it allows that code to run much faster and this allows us to train bigger neural networks or to do so the reason or multi-client even when we have a large network roam all the data the other reason that fast computation is important is that it turns out the process of training your network this is very intuitive often you have an idea for a neural network architecture and so you implement your idea and code implementing your idea then lets you run an experiment which tells you how well your neural network does and then by looking at it you go back to change the details of your new network and then you go around this circle over and over and when your new network takes a long time to Train it just takes a long time to go around this cycle and there's a huge difference in your productivity building effective neural networks when you can have an idea and try it and see the work in ten minutes or maybe ammos a day versus if you've to train your neural network for a month which sometimes does happened because you get a result back you know in ten minutes or maybe in a day you should just try a lot more ideas and be much more likely to discover in your network and it works well for your application and so faster computation has really helped in terms of speeding up the rate at which you can get an experimental result back and this has really helped both practitioners of neuro networks as well as researchers working and deep learning iterate much faster and improve your ideas much faster and so all this has also been a huge boon to the entire deep learning research community which has been incredible with just you know inventing new algorithms and making nonstop progress on that front so these are some of the forces powering the rise of deep learning but the good news is that these forces are still working powerfully to make deep learning even better Tech Data society is still throwing up one more digital data or take computation with the rise of specialized hardware like GPUs and faster networking many types of hardware I'm actually quite confident that our ability to do very large neural networks or should a computation point of view will keep on getting better and take algorithms relative learning research communities though continuously phenomenal at innovating on the algorithms front so because of this I think that we can be optimistic answer the optimistic the deep learning will keep on getting better for many years to come so that let's go on to the last video of the section where we'll talk a little bit more about what you learn from this course