Neural Networks and Deep Learning

What is a neural network?

0:01

The term, Deep Learning, refers to training Neural Networks, sometimes very large Neural Networks. So what exactly is a Neural Network? In this video, let's try to give you some of the basic intuitions.

0:12

Let's start to the Housing Price Prediction example. Let's say you have a data sets with six houses, so you know the size of the houses in square feet or square meters and you know the price of the house and you want to fit a function to predict the price of the houses, the function of the size. So if you are familiar with linear regression you might say, well let's put a straight line to these data so and we get a straight line like that. But to be Pathans you might say well we know that prices can never be negative, right. So instead of the straight line fit which eventually will become negative, let's bend the curve here. So it just ends up zero here. So this thick blue line ends up being your function for predicting the price of the house as a function of this size. Whereas zero here and then there's a straight line fit to the right.

1:04

So you can think of this function that you've just fit the housing prices as a very simple neural network. It's almost as simple as possible neural network. Let me draw it here.

1:17

We have as the input to the neural network the size of a house which one we call x. It goes into this node, this little circle and then it outputs the price which we call y. So this little circle, which is a single neuron in a neural network, implements this function that we drew on the left.

1:43

And all the neuron does is it inputs the size, computes this linear function, takes a max of zero, and then outputs the estimated price.

1:53

And by the way in the neural network literature, you see this function a lot. This function which goes to zero sometimes and then it'll takes of as a straight line. This function is called a ReLU function which stands for rectified linear units. So R-E-L-U. And rectify just means taking a max of 0 which is why you get a function shape like this.

2:23

You don't need to worry about ReLU units for now but it's just something you see again later in this course. So if this is a single neuron, neural network, really a tiny little neural network, a larger neural network is then formed by taking many of the single neurons and stacking them together. So, if you think of this neuron that's being like a single Lego brick, you then get a bigger neural network by stacking together many of these Lego bricks. Let's see an example.

2:57

Let’s say that instead of predicting the price of a house just from the size, you now have other features. You know other things about the host, such as the number of bedrooms, I should have wrote [INAUDIBLE] bedrooms, and you might think that one of the things that really affects the price of a house is family size, right? So can this house fit your family of three, or family of four, or family of five? And it's really based on the size in square feet or square meters, and the number of bedrooms that determines whether or not a house can fit your family's family size. And then maybe you know the zip codes, in different countries it's called a postal code of a house. And the zip code maybe as a future to tells you, walkability? So is this neighborhood highly walkable? Thing just walks the grocery store? Walk the school? Do you need to drive? And some people prefer highly walkable neighborhoods. And then the zip code as well as the wealth maybe tells you, right. Certainly in the United States but some other countries as well. Tells you how good is the school quality. So each of these little circles I'm drawing, can be one of those ReLU, rectified linear units or some other slightly non linear function. So that based on the size and number of bedrooms, you can estimate the family size, their zip code, based on walkability, based on zip code and wealth can estimate the school quality. And then finally you might think that well the way people decide how much they're will to pay for a house, is they look at the things that really matter to them. In this case family size, walkability, and school quality and that helps you predict the price.

4:46

So in the example x is all of these four inputs.

4:53

And y is the price you're trying to predict.

4:57

And so by stacking together a few of the single neurons or the simple predictors we have from the previous slide, we now have a slightly larger neural network. How you manage neural network is that when you implement it, you need to give it just the input x and the output y for a number of examples in your training set and all this things in the middle, they will figure out by itself.

5:25

So what you actually implement is this. Where, here, you have a neural network with four inputs. So the input features might be the size, number of bedrooms, the zip code or postal code, and the wealth of the neighborhood. And so given these input features, the job of the neural network will be to predict the price y. And notice also that each of these circle, these are called hidden units in the neural network, that each of them takes its inputs all four input features. So for example, rather than saying these first nodes represent family size and family size depends only on the features X1 and X2. Instead, we're going to say, well neural network, you decide whatever you want this known to be. And we'll give you all four of the features to complete whatever you want. So we say that layers that this is input layer and this layer in the middle of the neural network are density connected. Because every input feature is connected to every one of these circles in the middle. And the remarkable thing about neural networks is that, given enough data about x and y, given enough training examples with both x and y, neural networks are remarkably good at figuring out functions that accurately map from x to y.

6:48

So, that's a basic neural network. In turns out that as you build out your own neural networks, you probably find them to be most useful, most powerful in supervised learning incentives, meaning that you're trying to take an input x and map it to some output y, like we just saw in the housing price prediction example. In the next video let's go over some more examples of supervised learning and some examples of where you might find your networks to be incredibly helpful for your applications as well.

Supervised Learning with Neural Networks

0:03

There's been a lot of hype about neural networks. And perhaps some of that hype is justified, given how well they're working. But it turns out that so far, almost all the economic value created by neural networks has been through one type of machine learning, called supervised learning. Let's see what that means, and let's go over some examples. In supervised learning, you have some input x, and you want to learn a function mapping to some output y. So for example, just now we saw the housing price prediction application where you input some features of a home and try to output or estimate the price y. Here are some other examples that neural networks have been applied to very effectively. Possibly the single most lucrative application of deep learning today is online advertising, maybe not the most inspiring, but certainly very lucrative, in which, by inputting information about an ad to the website it's thinking of showing you, and some information about the user, neural networks have gotten very good at predicting whether or not you click on an ad. And by showing you and showing users the ads that you are most likely to click on, this has been an incredibly lucrative application of neural networks at multiple companies. Because the ability to show you ads that you're more likely to click on has a direct impact on the bottom line of some of the very large online advertising companies.

1:30

Computer vision has also made huge strides in the last several years, mostly due to deep learning. So you might input an image and want to output an index, say from 1 to 1,000 trying to tell you if this picture, it might be any one of, say a 1000 different images. So, you might us that for photo tagging. I think the recent progress in speech recognition has also been very exciting, where you can now input an audio clip to a neural network, and have it output a text transcript. Machine translation has also made huge strides thanks to deep learning where now you can have a neural network input an English sentence and directly output say, a Chinese sentence. And in autonomous driving, you might input an image, say a picture of what's in front of your car as well as some information from a radar, and based on that, maybe a neural network can be trained to tell you the position of the other cars on the road. So this becomes a key component in autonomous driving systems. So a lot of the value creation through neural networks has been through cleverly selecting what should be x and what should be y for your particular problem, and then fitting this supervised learning component into often a bigger system such as an autonomous vehicle. It turns out that slightly different types of neural networks are useful for different applications. For example, in the real estate application that we saw in the previous video, we use a universally standard neural network architecture, right? Maybe for real estate and online advertising might be a relatively standard neural network, like the one that we saw.

3:13

For image applications we'll often use convolution on neural networks, often abbreviated CNN.

3:21

And for sequence data. So for example, audio has a temporal component, right? Audio is played out over time, so audio is most naturally represented as a one-dimensional time series or as a one-dimensional temporal sequence. And so for sequence data, you often use an RNN, a recurrent neural network. Language, English and Chinese, the alphabets or the words come one at a time. So language is also most naturally represented as sequence data. And so more complex versions of RNNs are often used for these applications. And then, for more complex applications, like autonomous driving, where you have an image, that might suggest more of a CNN convolution neural network structure and radar info which is something quite different. You might end up with a more custom, or some more complex, hybrid neural network architecture.

4:20

So, just to be a bit more concrete about what are the standard CNN and RNN architectures. So in the literature you might have seen pictures like this. So that's a standard neural net. You might have seen pictures like this. Well this is an example of a Convolutional Neural Network, and we'll see in a later course exactly what this picture means and how can you implement this. But convolutional networks are often use for image data. And you might also have seen pictures like this. And you'll learn how to implement this in a later course. Recurrent neural networks are very good for this type of one-dimensional sequence data that has maybe a temporal component. You might also have heard about applications of machine learning to both Structured Data and Unstructured Data. Here's what the terms mean. Structured Data means basically databases of data.

5:19

So, for example, in housing price prediction, you might have a database or the column that tells you the size and the number of bedrooms. So, this is structured data, or in predicting whether or not a user will click on an ad, you might have information about the user, such as the age, some information about the ad, and then labels why that you're trying to predict. So that's structured data, meaning that each of the features, such as size of the house, the number of bedrooms, or the age of a user, has a very well defined meaning. In contrast, unstructured data refers to things like audio, raw audio, or images where you might want to recognize what's in the image or text. Here the features might be the pixel values in an image or the individual words in a piece of text. Historically, it has been much harder for computers to make sense of unstructured data compared to structured data. And the fact the human race has evolved to be very good at understanding audio cues as well as images. And then text was a more recent invention, but people are just really good at interpreting unstructured data. And so one of the most exciting things about the rise of neural networks is that, thanks to deep learning, thanks to neural networks, computers are now much better at interpreting unstructured data as well compared to just a few years ago. And this creates opportunities for many new exciting applications that use speech recognition, image recognition, natural language processing on text,

6:56

much more than was possible even just two or three years ago. I think because people have a natural empathy to understanding unstructured data, you might hear about neural network successes on unstructured data more in the media because it's just cool when the neural network recognizes a cat. We all like that, and we all know what that means. But it turns out that a lot of short term economic value that neural networks are creating has also been on structured data, such as much better advertising systems, much better profit recommendations, and just a much better ability to process the giant databases that many companies have to make accurate predictions from them. So in this course, a lot of the techniques we'll go over will apply to both structured data and to unstructured data. For the purposes of explaining the algorithms, we will draw a little bit more on examples that use unstructured data. But as you think through applications of neural networks within your own team I hope you find both uses for them in both structured and unstructured data.

8:02

So neural networks have transformed supervised learning and are creating tremendous economic value. It turns out though, that the basic technical ideas behind neural networks have mostly been around, sometimes for many decades. So why is it, then, that they're only just now taking off and working so well? In the next video, we'll talk about why it's only quite recently that neural networks have become this incredibly powerful tool that you can use.

Why is Deep Learning taking off?

0:00

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 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

About this Course

0:00

So you're just about to reach the end of the first week of material on the first course in this specialization. Let me give you a quick sense of what you'll learn in the next few weeks as well. As I said in the first video, this specialization comprises five courses. And right now, we're in the first of these five courses which teach you the most important foundations, really the most important building blocks of deep learning. So by the end of this first course, you know how to build and get to work a deep neural network. So here the details of what is in this first course. This course is four weeks of material. And you're just coming up to the end of the first week when you saw an introduction to deep learning. At the end of each week, there are also be 10 multiple-choice questions that you can use to double check your understanding of the material. So when you're done watching this video, I hope you're going to take a look at those questions. In the second week, you then learn about the Basics of Neural Network Programming. You'll learn the structure of what we call the forward propagation and the back propagation steps of the algorithm and how to implement neural networks efficiently. Starting from the second week, you also get to do a programming exercise that lets you practice the material you've just learned, implement the algorithms yourself and see it work for yourself. I find it really satisfying when I learn about algorithm and they get it coded up and I see it worked for myself. So I hope you enjoy that too. Having learned the framework for neural network programming in the third week, you code up a single hidden layer neural network. All right. So you learn about all the key concepts needed to implement and get to work in neural network. And then finally in week four, you build a deep neural network and neural network with many layers and see it worked for yourself. So, congratulations on finishing the videos after this one. I hope that you now have a good high-level sense of what's happening in deep learning. And perhaps some of you are also assigned to, has some ideas of where you might want to apply deep learning yourself. So, I hope that after this video, you go on to take a look at the 10 multiple choice questions that follow this video on the course website and just use the 10 multiple choice questions to check your understanding. And don't review, you don't get all the answers right the first time, you can try again and again until you get them all right. I found them useful to make sure that I'm understanding all the concepts, I hope you're that way too. So with that, congrats again for getting up to here and I look forward to seeing you in the week two videos.

Frequently Asked Questions

Congratulations to be part of the first class of the Deep Learning Specialization! This form is here to help you find the answers to the commonly asked questions. We will update it as we receive new questions that we think are important for all learners.

General Questions

Q: I have an idea that would improve the course content. What can I do? A: Contact us at feedback@deeplearning.ai or put it in the forum "New ideas for the course". We are happy to collaborate with learners willing to improve the course! Thanks a lot.

Q: I cannot submit my assignment? A: This issue should not be happening but if it does please let us know immediately. One temporary work around would be to download your notebook and go to the corresponding programming assignment tab ==> + Create Submission and upload it.

Q: The audio in the videos is quite bad sometimes, muffled or low volume. Please fix it. A: You can mitigate the audio issues by turning down the bass and up the treble if you have those controls, or using a headset, which naturally emphasizes the higher frequencies. Also you may want to switch on the English closed captioning. Of course, we are working everyday to improve the quality of the videos and avoid anything that can affect your learning.

Q: What does it mean when I see “Math Processing Error?” A: The page is attempting to use MathJax to render math symbols. Sometimes the content delivery network can be sluggish or you have caught the web page Ajax javascript code in an incomplete state. Normally just refreshing the page to make it load fully fixes the problem.

Q: The video quality is bad? A: You could click the settings option in the video and upgrade the quality to High. (recommended if you have a good internet connection)

Q: Is there a prerequisite for this course? A: Students are expected to have the following background:

Very basic programming skills (i.e. ability to work with dictionaries and for loops)

Familiarity with basic machine learning (how do we represent a dataset as a matrix, etc.).

Familiarity with the basic linear algebra (matrix multiplications, vector operations etc.).

Q: Why do we have to use Python? A: Python is an open-source language, anyone can use it from anywhere in the world. It is widely used in academics (research labs) or in the industry. It has a useful library "Numpy" that makes math operations very easy. Python has several deep learning frameworks running on top of it (Tensorflow, Keras, PaddlePaddle, CNTK, Caffe, ...) and you are going to learn some of them. It is also easy to learn. Furthermore, we believe Python has a good future, as the community is really active and builds amazing stuff.

Q: Has anyone figured out the how to solve this problem? Here is my code [Insert code]. A: This is a violation of the Coursera Honor Code.

Q: I've submitted correct answers for [insert problem]. However I would like to compare my implementation with other who did correctly. A: This is a violation of the Coursera Honor Code.

Q: This is my email: [insert email]. Can we get the answer for the quiz? A: This is a violation of the Coursera Honor Code.

Q: Do I receive a certificate once I complete this course? A: Course Certificate is available in this course.

Q: What is the correct technique of entering a numeric answer to a text box question ? A: Coursera's software for numeric answers only supports '.' as the decimal delimiter (not ',') and require that fractions be simplified to decimals. For answers with many decimal digits, please use a 2 digits after decimal point rounding method when entering solutions if not mentioned in the question.

Q: What is the correct technique of entering a 1 element matrix ? A: They should be entered as just the element without brackets.

Q: What does a A being a 3 element vector or a 3 dimensional vector mean? A: If not described a vector as mentioned in the questions is A =

⎡⎣element1element2element3⎤⎦

A=

⎣

⎡

element1

element2

element3

⎦

⎤

Q: I think I found an error in a video. What should I do? A: First, post it on the Errata forum. We will try to implement your feedback as soon as possible. You could also send us an email at feedback@deeplearning.ai.

Q: My quiz grade displayed is wrong or I have a verification issue or I cannot retake a quiz. What should I do? A: Contact learner support. These queries can only be resolved by learner support and it is best if they are contacted directly. Do not flag such issues.

Course Resources

0:00

I hope you enjoyed this course and to help you complete it I want to make sure that there are few course resources that you know about first if you have any questions or you want to discuss anything with the classmates or the teaching staff including me or if you want to file a bug report the best place to do that is the discussion forum the teaching staff and I will be monitoring that regularly and this is also a good place for you to get answers to your questions from your classmates or if you wish try to answer your classmates questions to get to the discussion forum from this course home page if you look at this menu bar on the left you also might look a bit different than mine but they'll do this discussion forum tab which gives you click on gives you to the discussion forum the best way to ask questions is on the discussion forum but it for some reason you need to contact us directly or let us know about some problem feel free also to email us at this email address I promise we will read every email and we'll try to address commonly occurring issues although depending on the email volume I can't guarantee that we'll be able to reply promptly to every email but we will read every email than you send next I know that there are other companies that wish to train maybe large numbers of employees with deep learning if you're responsible for employee training in your company and would like to train a hundred or more employees with deep learning expertise please feel free to get in touch at this email and we'll see if we can help you we're just in the early phases of developing the university academic program but if you're a university instructor or a university administrator interested in offering a deep learning course at your university please feel free to contact us as this email address though I hope that gives you more resources to complete the course maybe I'll see some of you in the discussion forums and best of luck

Introduction to deep learning

测验, 10 个问题

第 1 个问题

1  
point

**1. 第 1 个问题**

What does the analogy “AI is the new electricity” refer to?



AI is powering personal devices in our homes and offices, similar to electricity.



Similar to electricity starting about 100 years ago, AI is transforming multiple industries.



Through the “smart grid”, AI is delivering a new wave of electricity.



AI runs on computers and is thus powered by electricity, but it is letting computers do things not possible before.

第 2 个问题

1  
point

**2. 第 2 个问题**

Which of these are reasons for Deep Learning recently taking off? (Check the three options that apply.)



Deep learning has resulted in significant improvements in important applications such as online advertising, speech recognition, and image recognition.



We have access to a lot more data.



We have access to a lot more computational power.



Neural Networks are a brand new field.

第 3 个问题

1  
point

**3. 第 3 个问题**

Recall this diagram of iterating over different ML ideas. Which of the statements below are true? (Check all that apply.)



Being able to try out ideas quickly allows deep learning engineers to iterate more quickly.



Faster computation can help speed up how long a team takes to iterate to a good idea.



It is faster to train on a big dataset than a small dataset.



Recent progress in deep learning algorithms has allowed us to train good models faster (even without changing the CPU/GPU hardware).

第 4 个问题

1  
point

**4. 第 4 个问题**

When an experienced deep learning engineer works on a new problem, they can usually use insight from previous problems to train a good model on the first try, without needing to iterate multiple times through different models. True/False?



True



False

第 5 个问题

1  
point

**5. 第 5 个问题**

Which one of these plots represents a ReLU activation function?



Figure 1:



Figure 2:



Figure 3:



Figure 4:

第 6 个问题

1  
point

**6. 第 6 个问题**

Images for cat recognition is an example of “structured” data, because it is represented as a structured array in a computer. True/False?



True



False

第 7 个问题

1  
point

**7. 第 7 个问题**

A demographic dataset with statistics on different cities' population, GDP per capita, economic growth is an example of “unstructured” data because it contains data coming from different sources. True/False?



True



False

第 8 个问题

1  
point

**8. 第 8 个问题**

Why is an RNN (Recurrent Neural Network) used for machine translation, say translating English to French? (Check all that apply.)



It can be trained as a supervised learning problem.



It is strictly more powerful than a Convolutional Neural Network (CNN).



It is applicable when the input/output is a sequence (e.g., a sequence of words).



RNNs represent the recurrent process of Idea->Code->Experiment->Idea->....

第 9 个问题

1  
point

**9. 第 9 个问题**

In this diagram which we hand-drew in lecture, what do the horizontal axis (x-axis) and vertical axis (y-axis) represent?



* x-axis is the performance of the algorithm
* y-axis (vertical axis) is the amount of data.



* x-axis is the amount of data
* y-axis is the size of the model you train.



* x-axis is the input to the algorithm
* y-axis is outputs.



* x-axis is the amount of data
* y-axis (vertical axis) is the performance of the algorithm.

第 10 个问题

1  
point

**10. 第 10 个问题**

Assuming the trends described in the previous question's figure are accurate (and hoping you got the axis labels right), which of the following are true? (Check all that apply.)



Decreasing the size of a neural network generally does not hurt an algorithm’s performance, and it may help significantly.



Increasing the size of a neural network generally does not hurt an algorithm’s performance, and it may help significantly.



Increasing the training set size generally does not hurt an algorithm’s performance, and it may help significantly.



Decreasing the training set size generally does not hurt an algorithm’s performance, and it may help significantly.