Introduction to Machine Learning for Artists 予藝術家的機器學習講義

Transcript 文字記錄

2022-07-31 (Sun) 19:30-21:00

Speaker 講者:
Dr Héctor Rodríguez 羅海德博士

Language 語言:
English 英語

Hector Rodriguez: Hello everyone! And welcome to this talk about Machine Learning for Artists. This is part of the second series of talks in the symposium on knowledge Asymmetries in the Age of Machine Learning. And this symposium is organized by the Writing Machine Collective. The Collective was founded by Linda Lia. And I'm one of the members of that. It's an artist-run collective. This symposium is funded by the Hong Kong Arts Development Council. Thank you very much, to the Arts Development Council.

The main topic is the Status of Knowledge in an Age of Rapid Technological Development. So, as a contribution to this kind of topic, I'm going to speak today, about Machine Learning. My name is Hector Rodriguez. I'm an artist, and a media scholar. I teach at the City University of Hong Kong. And I often use Machine Learning, and Computer Vision Technologies in my work.

Now, the topic of the talk today is Machine Learning for Artists. Now, I want to highlight that this is a talk about introducing the idea of Machine Learning to Artists who are assumed to have no background on this subject. So, that's the target audience for the talk. If you are experts, you might find the talk a bit, maybe simple, still, you're welcome, maybe if you might find an interesting perspective in the talk. But I will assume no background knowledge. So, I will try to cover some very basic things about what Machine Learning is.

I will also try to show ways in which the topic of Machine Learning can be connected to strengths of the humanities, strengths of critical theory. And that will come maybe towards the end of the talk.

But then, welcome everybody! Now, the first section of the talk, most of the talk really, is about explaining the meaning of this expression, Machine Learning. And I'm going to introduce this idea with a very simple, and hypothetical example. Although, the data that I'm using is real, don't think too much about the accuracy of the example. It's just an illustration of Machinery.

So, assume for a minute that you work for a University, maybe for an Art school, and you're trying to decide how to admit students into your department. So, that's your job. You have to select students who apply to your program, and you have to decide whether you want to admit them or not. Now, this is my favorite example, because many years ago, when I was young, and my hair was black, I was the Admissions Officer for my school. So, the problem of how to admit students is one that I'm very interested in.

So, imagine that you want to admit students who you can predict, of course, maybe not with full certainty, but with a certain probability, you want to be able to predict that the students will do well in University. They will not get an F, and they will get reasonably good grades. And that's how you want to admit students. Now, I'm not saying that's a good way to do it. Obviously already, the example shows that whenever we apply Machine Learning, questions of social power, and value play a role. And one reason why I chose this example is because, you see from the beginning that there's a question of Power, who gets admitted into school. There are question of value. Education is something that is valuable in our society, and that will give resources to the people who obtain it. So, questions of Power and Value, it can be very important from the start, but I'm going to bracket them from now.

Let's say that I want to predict whether a given applicant will do well in school. If I think the applicant will do well, I accept the applicant. If I predict that the applicant will do poorly, I reject the applicant. So, I want to then find some way to be able to make this prediction. Let's say that I have in my possession, a lot of data about previous students. So, imagine that, here we have a population of possible students, and from that population, I have a sample, let's say it's a random sample, randomly selected students in the past. And I have two pieces of data that I can work with. First of all, the University Entrance Scores. So, I got here on the X, the horizontal axis, the SAT, because I got data from the American system, this is the kind of University entrance exam in the US. It's not exactly a University entrance exam, but something like that. Scholastic aptitude test. So, I have those scores for a number of students who being in the University in the past, and I know the GPA that those students obtained, and the GPA is on the vertical line. So, I plot my dataset. It's actually a small dataset, but never mind. It's a small sample on this graph.

So, you can see, for example, here's a student, this point that I'm pointing at, who has a result of less than 1700 in the SAT, which is very low; and a GPA, which is also quite low, less than 2.6. He did not fail, but he did not do very well. So, here are all the points in my dataset. And I want to be able to somehow predict, use that dataset to develop some kind of system that will automatically predict when a new student comes in, and I know the student's SAT score, the entrance exam. The system will tell me, I predict this student’s GPA will be 3.0. So, from this dataset that I have the sample, I want to be able to then predict the GPA, given the SAT results of a student, so that then if I can do that, if I can develop a software, or an algorithm that can predict that, then I will solve my admissions problem, because then I will choose the students with a very good GPA prediction. Again, I'm not saying this is a good way to do it.

So, essentially, what I want to get, or to learn is a Function. And the Concept of a Function is really important. A Function is an input-output system, where you give the Function some input, and the Function will produce one, and only one-unit output for that input. In this case, the inputs will be the SAT score, the outputs will be the GPA. So, I want to learn a Function, or to somehow find a Function that will be able to predict the GPA score of students based on the SAT score. And I want to learn one Function that will predict well on this dataset. So, I will use this dataset to learn that Function.

So, now there are many Functions in the world. So, let's say that I decide for some reason to restrict the Functions to just straight lines, why would I make that decision? Well, first of all, there are so many Functions in the world, right? There are sign waves, there are all kinds of polynomials that are drawn as strange possible curves. There are all kinds of Functions in the world. If I want to find a good Function, one that will work well for this data, I want to still restrict myself so I don't get lost. And it becomes impossible to look for so many... in so many possible Functions. So, let's say that I decide to look for a straight line, some straight line that will make this prediction, for example, this one. So, you see a red line, and this red line is the graph of a function. Now, why have I chosen some straight line? Well, because maybe I looked at my data, and I think, well, more or less on a straight line, not really, of course, but more or less, it goes like a diagonal. Maybe also because straight lines are very easy to work with Mathematically. So, I decide I'm going to... for convenience, choose a straight line.

So, let's say that somehow, maybe randomly, or by some other method, I pick this red line. And this Function would then allow me to... in the future, find a point on the X axis, which would be the SAT score of a student who wants to be admitted. And I use the line to find the predicted GPA score. So, the idea is that I have looked for a line here by some method, which we don't talk about yet, which is somehow a good fit to this data. And you can see that this line approximately goes through the data. What would a bad fit have been? For example, if I had picked a line that is a vertical line, or a horizontal diagonal going in a very different direction would not fit the shape of the data very well.

So, in fact, this data has been picked using... maybe the Oldest Machine Learning Method, which is called Linear List Squares. What does it mean to say that this line fits the data very well? Let me show you another diagram again, the same line, the same data, but now I show you lines connecting all my data points to the line I have found. And what's good about this line, the red line is this; if I compute the distance of every point in my sample to that red line, these are the distances, and I add them all together, the total distance, or perhaps if I average them, would be relatively small. If on the other hand, I were to find, say a horizontal line quite low in this diagram, and I compute the lines of all the points in my data set to that horizontal line; and I average them, or I add them together. The distances will be very large. Such a Model, such a Function would be a very bad representation of my data, but this is a Function that has been found using a Machine Learning Method, and which is a good function, because the distance on average of the sample of every point in my sample to that line is small, so the line fits the data. So, I can say, okay, this line is going to help me to predict the student's scores in the future.

I say, on average, the distances will be small. On average. You can see some cases, there's one student who is very far from the line, quite a few students, but on average, most of the students in the sample that I have, have results that are quite close to the line. So, this is a very simple hypothetical example. Let's try to now understand what Machine Learning is, maybe from this example.

The first important thing is that, when we talk about Machine Learning, what we learn is a Function. We learn to compute a Function. A Function, as I said, is an input-output system. So, it receives input variables, or it receives a vector of input variables, and it outputs another vector. The word Vector here is not too important. It just means a sequence of numbers. In the current case, just one number as input, and then one number as output. So, what is learned is always a Function. So, Machine Learning aims to learn, to compute a Function. And the Function we can think of is just a kind of black box. Something comes in, something comes out.

Now, the question of course is, how do we learn to find a Good Function? One that somehow fits the data well enough. I will come back to this later, but let me give you some examples of some of the kinds of Functions that we might learn in Machine Learning. Here is an example with a dataset that is well known. It's called the MNIST dataset. The dataset has 2 components. First of all, it has images of handwritten digits, just numbers 5, 0, 4, 1, together with labels, that tell us what the number is. So, for example, the one on the top left is a 5, and it's labeled as a 5. It's a bit of a strange one. It looks like a 3 too, it could be a 3. When I write 3, it looks like that. But it's labeled as a 5. So, we have a very large dataset of very small images of handwritten digits, together with labels.

So, we want our Function to receive as input images, and to output a label that predicts the number that that is. For example, this is a simple example, many of the Functions that we learn do not output a single label, they will output a probability, but never mind, just to simplify.

So, input, the first thing that we need to think about when we are dealing with Machine Learning is, what kind of Function we're learning, and to understand the kind of Function we're learning, we have to ask, what are the inputs? What are the outputs? Here, the inputs are images, the outputs will be labels, and we want the Function to output the correct label for any image that it gets. This is one example.

This example belongs to a type of problem called a Classification Problem. Which is, the outputs are labels. Another example might be, if you give the system pictures of animals, and you want the system to output a label that tells you what animal it is, dog, cat.

Another problem is the problem of Depth Map Estimation. So, in this case, the inputs are again, images. So, the inputs, as you see on the left would be images, the outputs in this case would also be images, but there would be images where the color of every pixel would be a number that tells you the depth of the object represented by that pixel. Let's say that we restrict pixels to be in between zero, meaning black; and one, meaning white. Then zero, meaning black would represent a pixel that is very far away; one being white would represent a pixel that's very close, and anything in between is again in between, not too far, not too close. As you can see here, the guy close to the camera is quite white. The objects that are far away are gray in the middle ground. And the objects very, very far away are black. So, again, input is images, output are also images which encode depth predictions. It's called another current Depth Map Estimation, very difficult, complicated problem. But already these examples I have shown you, of two different kinds of Functions. Three, if you include my school Admissions, example, show you an important difference in the functions learned in Machine Learning.

There's a difference between Regression and Classification. Let me start with Classification. The example of Classification I have just shown you is that of the numbers. In a Classification case, the outputs of the function will be discrete labels, for example, 0 1, 2, 3, or if we are predicting the types of animals, it will be cat, dog monkey, and so on. So, in this case, the outputs are discrete labels. This is called Classification.

The example of a Depth Map, but also the first example about school Admissions, is an example of Regression. Where the outputs are not labels, but are continuous values between two different values. For example, in the case of Depth Maps, we are predicting for every pixel, a number between zero meaning far away, and one meaning very close, but we allow any number continuously in theory, between zero and one. In the case of the Admission scores, the outputs are GPAs, or maybe average GPAs. And the average GPA is normally a number between zero, which means an F, and maybe 4, which I think is an A. And any number between that is in theory acceptable. So, we have continuous values. That's Regression Problem. I'm not saying all problems in Machine Learning are one or the other, but it's a very interesting kind of distinction.

The distinction by the way is not always absolute, because you can always take a problem that's originally a Regression Problem, and turn it into a Classification Problem. For example, imagine that your problem is about predicting Depth Maps. That's a Regression Problem, because the outputs are continuous between zero, and one for every pixel. But you can say; I'm going to divide pixels into three types, Background, Middle ground, Foreground. And so, my Depth Maps can only have three possible values; black, gray, or white. Then what was originally a Regression Problem where my values are continuous now becomes a Classification Problem. And sometimes, we are able to do that, to change how we look at a problem. Regression Problem can become a Classification Problem. But the key point I want to make is not this distinction, but it is that, if we think about Machine Learning, we also have to think in terms of Functions, which are input-output systems. So, we need to ask, what's the input, and what's the output.

So, for example, if you ever need to talk to a technical person, and you want to find a system that will achieve something, the best way to express yourself precisely is by saying to the person, I want my system to receive such a thing as an input, and produce such an output.

Another example of a Function in Machine Learning is a Clustering Function. Clustering is a little bit different from the other ones. It will receive data, any kind of data, but let's say that they are images. Since we're Artists we care about image, or sound, probably. Let's say that the data are images. And the aim of Clustering is to divide those images into groups, which we call them Clusters. The word Group, it's not very good because in Math, there's something called Group theory, which is something a little bit different. So, better call them Clusters. And what we want is that images that are similar end up in the same Cluster, the same group; and images that are different, end up in different Clusters.

And again, what do we mean by Similar? It's something that we have to decide. One very common way to do it is, we show images of animals, and we want all images of the same animal to be Clustered together. Like you see in this example. So, we see for example here, all the sharks in one group, all the birds in another group. These are not only animals by the way, there are also balloons in another group.

So, here’s an example, you can imagine that I have all these data points, and I want to be able to somehow cut the cake in a way, so that each point is in one piece of the cake. And somehow that this cutting of the cake makes sense, that points that are similar to a human being will end up being in the same Cluster. So, in this case, we have three Clusters.

So, the Function in Clustering takes us an input, some data, for example, images, they can be other things; they can be videos, audio, clips, et cetera, but let's say images. We also have to tell the system, how many groups, how many clusters we want. There are some algorithms that can find that automatically, but many of them require you to tell them how many clusters, for example, I want three. So, to go back, this is my data, I want it divided into three clusters. For example, I can also say four, five. And the function will output a label for each piece of data. So, you give an image, and the function will output this image belongs to cluster zero, or cluster one, or cluster two. So, the labels usually are numbers that identify the group to which the data belongs.

I think these are some examples of Functions that we learn. One again is Classification, another one, it was Regression; another one, it was Clustering. And Clustering can be understood as a form of Classification too, if you like. So, that's the first thing.

Now, what we want in Machine Learning is to find... I see some people are coming in. You’re welcome to ask questions and interrupt me. I think if you come in, you already missed something so basic that I don't know if you're going to follow the topic, but anyway, welcome. And welcome to interrupt me. Just unmute yourself, and screen.

So, we want to... as I said, find Functions, but now, comes a question. What Function do we want? Obviously, we want a Function that fits the data sample that I have. As I mentioned in the beginning, with our example of the Admissions exercise, I always have a sample of data that I will use in learning. And I want to find a function that somehow fits that data. So, what does it mean? I need some ways to evaluate whether some Function is a good fit, or a bad fit to the data? I need a Criterion of Fit. The expression Criterion of Fit is from a Philosopher called Deborah Mayo, who wrote some really interesting work on the philosophy of knowledge. But in Machine Learning, we often call this Criterion, the error, the loss, or the cost. And that is itself a Function by the way. But I won't use the term Function here to avoid confusing people.

So, we have some way to measure whether a Function is a good, or a bad fit for the data. That has to be a Quantitative Criterion. Let's go back to our first example. So, let's say that we are provided with... as in the case of our initial example, some data, and in the data, what I have are inputs and outputs. So, I have inputs, and their corresponding outputs. So, what are the inputs, SAT scores, if you remember that, what are the outputs, the GPA scores of each of those students. So, I have these pairs; one SAT, one GPA, one SAT, one GPA, this is the input; and it's a sample that I have collected.

There is a whole question about how I've collected the sample, and I will come back to this later. But let's say that somehow, I have this data, and this data is like a ground truth. I assume I take for granted that this data is true. So, I want to find a Function that fits the data. What that means is that by some measure, I want that Function to have a very low error, or very low cost, or very low loss. This was error loss cost, often used in the same way. The error, or the loss measures how bad the Function is. So, I want to be low. How would I measure how bad a line here might be? This line is good. And I told you how to measure it in this case. The way to measure it is, you take all the points that I have in my dataset, and you compute the distance of each point to the line, and then you add all the distances together; maybe you average them. And that's my loss. If the loss is very high, then that means this line is not a good fit. If the loss is very low, that means the line is a pretty good fit. In this case, there is Mathematical proof that this line here has the lowest possible cost to this dataset. So, it's the best possible fit.

Now, in other situations, I might have other ways to compute how good, or bad my Function is, but I need a way to quantify it, so that I can find a good Function. So, the loss, or error Function is a very essential part of Machine Learning. When you have a problem, you might have different ways to measure the loss, or the error. Some might work better than others.

Now, the example that I started with is an example of Supervised Learning. In Supervised Learning, the data sample that I will use for learning comes with inputs, and outputs. SAT scores, GPA score, so that I want a Function that is a good fit to this input-output pairs. When that happens, this is called Supervised Learning.

Another example of Supervised Learning is, before in a Classification Problem, I mentioned the images of numbers. We often learn to recognize digits, handwritten digits, by having a sample of input-output pairs. The input will be images, the output will be labels. So, we have those labels, they are the ground truth. If we are trying to identify animals in images, we also normally do it with input-output pairs. So, the input will be images of animals, the outputs might be labels. This is a dog; this is a cat.

So, what I want to find is a Function that will allow me to predict. In this case, it’s the labels. And which is a good Function based on some loss, or measure that has a low loss, or error measure. However, I define it.

Other cases are involved Unsupervised Learning. And usually that's a case where you are given some data, some samples which involve inputs, but you don't know the outputs. So, you have to learn the Function without knowing the outputs. This is what happens usually in

Clustering. Let me go back for a second. If I have to Cluster, remember I have some data, and I want to divide it into groups. Usually, I don't learn a good Function for this by giving the Function some examples of... this is a data point, and it belongs to this Cluster. That's another data point, it belongs to this other Cluster. Usually, I don't. I just keep the algorithm, a set of data points; and I ask the algorithm to find a good Function to assign Cluster labels to those data points.

How does it do it? This can be a bit tricky. So, I mentioned something very briefly. Don't be too scared if it's complicated, but there's one algorithm called K-Means. If you ever work with processing in some libraries, maybe you have used a technique called Voronoi partition. K-Means is an algorithm that attempts to approximate Voronoi partitions. But don't worry if you've never heard about this.

So, let's say that here are my data points, K-means will try to divide them into Clusters, or groups. Here, colored, red is one group, blue is another group. So, those points that are in the blue area belong to the same group. And somehow the K- means the algorithm will try to divide them into groups in a way that makes sense. How will it do it? It will do it by trying to find centers, which are here marked with an X, which are points, and then classify every point to the cluster represented by the centroid closest to it. So, let's say that in this case, I have 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 Clusters. So, here's my data point, the black points, and I want to find 10 clusters. So, the algorithm will try to find 10 points marked with an X. These are not necessarily points in my dataset, they could be virtual points. And then classify every point to the Cluster represented by the point mark with an X closest to it. So, this point will go to this, cluster this point will go to this cluster, because it's closest to this centroid. These points are called Centroids in the literature.

So, how to find Good Access. These are the points that are the Centroid points, around which you will cluster elements. So, what the K-Means will try is to find, in this case, 10 points; in this other slide, just 2 clusters, such that the centers of the clusters have minimal distances to all the points in that cluster. So, if you look at, in this case, you have only 6 points in this diagram by Aaron Zhu. And you cluster them into two groups. And the idea is, you cluster them by finding a Centroid here, it's a red dot, such that the distance of every point in each cluster to it Centroid is as small as possible. That is how we compute the loss. That's the formula we use. So, we just take the sum of the distances of each point to its Centroid. Doesn't matter if this is not very clear, but the idea is quite simple, we'll need, in order to pick a good Function, we need some ways to quantify whether the Function is good, or not, somehow.

If it's Supervised, if the problem is Supervised, we have inputs and outputs. We have those examples in our dataset, in our data sample. Then usually, the loss is, that the predictions made by our Function will be not too far from the actual ground truth. So, if I have a student, and I predict that the student will get a GPA of 3.2, and the student's actual GPA in my data sample is 3.1, then I'm okay. Not too far. Whereas if I have a line that predicts for students, things like, the GPA will be 3.2, when the student's GPA was actually 0.5. And that happens a lot. It predicts strongly in that aspect.

Somehow, I need a way to quantify how good, or bad the model is. And a very important thing, we don't want a model that will predict everything accurately. We almost never in Machine Learning, except in some cases like Linear List Squares that I showed you in the beginning. But in many cases, we don't need a model that is going to predict exactly the outputs in our sample. All we want is a model that's good enough, meaning that the loss is not very high, it's low.

In a way, a Model can be interesting, a Function can be interesting, even if it's not a great predictor. Sometimes, it can be interesting to study in its own terms. Now, you may have heard me using the word Model. Here Model is another term for the Function that I want to find, but I used to word Model to indicate that I want to Function that is a good fit to my data. So, a Model is function that I want to fit to my data. Model is just, in this context it’s just a Function.

So, now we notice that we have seen some key elements in Machine Learning. We have a problem that we wish to solve. The problem typically comes from a social situation that involves value, or power, not necessarily, but in many cases it does. I can express the problem as a search for a Function, an input-output system. The search for the Function will work on a sample of data that is available to me. And I also have an error measure that tells me for any Function that I might come up with whether this Function is good or not, is a good fit, or a bad fit.

So, already you could imagine one way I could solve, let's say going back to my first problem, let me go back a little bit. This problem of Admissions. One way that I could find a good line, you can say that it's kind of a stupid method maybe, would be, I start finding random lines, and then I check the error of that line. Randomly, I pick a horizontal line, and I then check how good, or bad is the fit. And I write it down. Then I find another line randomly, and I check again. And I do this many times, and then I can pick finally the line that has the lowest error, meaning the best fit, is one way to do it. This is not how we do it in Machine Learning, but is one way I could do. It's not the way it's done here, but it's one way to do.

Now, A key point, which I mentioned in the beginning, and I have to mention it again. It's essential for Machine Learning. When I'm trying to face a Machine Learning Problem, I want to find a Function, but there are many Functions in this world, infinite number of Functions. I'm not going to start searching for any possible kind of Function, because then the possibilities are too many, too many possible Functions. So, typically, I restrict myself to only one kind of Function, one family of Functions. For example, let me go back again to the opening problem. Here, I decide my functions are going to be just straight lines. Someone might say, why straight lines, why not curves? Well, I make that choice. The choice could be because it's very easy to compute with straight lines, it could be because I look at my data, and more or less, it's a straight line. The points are more, or less on a straight line. It could be for many reasons, but I restrict my search. I'm going to only look for straight lines. I might choose instead Functions that are not linear, that are curve, for example. The Function might be some kind of polynomial, whatever. But I restrict my search to a Family of Functions. Most of the Early Old Machine Learning Methods are linear. But for some reason, we have to pick some Family of Functions, and we focus on that family.

Now, let's say that we are dealing with... example, the problem that I mentioned in the beginning, which is the Admissions. And all my data points, the inputs are just one dimension, and the outputs on one dimension. So, the Function is going to be, if it's a line, a line in 2D. And it's easy to write the formula for a line in 2D, we all know it from Secondary School, or Primary School. Sorry for giving you all this kind of very silly example, but it makes some of the points clear.

So, the formula for a line is basically, Y= a + bx. Now, this formula, if I ask you, okay, here's the formula, draw me the line, you can't do it, because it's not really the formula. I said the formula for a line, but it's not a specific line. You need to tell me something, you need to tell me what are the values of A and B, for example, let's say you tell me A= 0, B= 1, so, the formula becomes Y=X. Then I can now draw the line. You can say A=0, and B=3. So, another formula becomes, Y=3X. You somehow need to tell me the values of the parameters. A and B are called Parameters.

What are Parameters? The Parameters are values that you need to give in order to be able to identify one Function within a Family of Functions. Here, the Family of Functions are all the straight lines on the plane. Some are horizontal, some are vertical, many diagonals in different directions. All of these lines, each one is a possible Function that I might use as the solution of my problem. I want to find one Function, to find the Function basically, is to find what A and B are, the values of the Parameters.

In this case, the Family of Functions I have picked have 2 parameters. If I give those 2 numbers, I identify a unique function in that family. That's the purpose of Parameters. Now, this is a very simple family of models, many of the complicated models that we use, Families of Functions that we work with in Machine Learning could have billions of Parameters. For example, those used in working with language, with translation, they could have billions of Parameters. So, and it's normal, very common, even with Functions that are less complicated when you're trying to find a Function to work with thousands of Parameters, that's very, very common.

But in any case, the key idea is, we have a Family of Function wherein we want to search for one Function, and that Function is identified by a set of Parameters. It can be 2, it can be 1; it can be 20, can be 10,000, 10 million, 10 billion Parameters. That's the next step. Then I decide, what is the Family of Function I'm going to use. And sometimes I might try different Families of Functions, but provisionally, I take one family, and then within that Family of Functions, I want to find a good fit, a good fitting function, meaning I want to find the 2 parameters of a Function that fit my data well.

Now, how do we search then for a good fit? Well, I mentioned one way we could do it before, which is, I start picking two values of A and B, in this case, in this example, randomly. Say, A=0, B=1.3, and I try. A= 0. Let's say A is the Y intercept; B is the slope. So, I pick some at random. I pick many at random, and I compute how well my model fits with the data sample that I have for each case. And then I find the best of it. But that way of searching is kind of blind to just blindly picking. And sometimes, we do similar things, but usually we want to pick a method, we want to search in my Family of Functions, using a method that is more systematic, that is more intelligent. And this is really very important. It's really what... you might say the most distinctive feature of Machine Learning that we want to pick a Function from the family that I have chosen using some search method that is systematic, not just I pick something at random. Usually by the way, there is a random component that we use, but somehow, we want the search to involve some element that's systematic.

Now, I had a slide here, which I'm not going to explain, because I think it's going to confuse everybody. So, here is a nice slide, that I won't explain. Have a look at it. It's one example of a method, not the only method that is used, it's called Gradient Descent. And basically, what Gradient Descent does is, it starts with some guess, the guess is usually a random initial guess, but then given that random guess, you try to search for a Function near that initial guess, which is a better and better fit to your data. And this is a kind of repetitive iterative process that is going on. We don't need to go into the details if you want, during the Q and A, I can come back and try to explain it to you to confuse you.

My point is, it's very distinctive of Machine Learning that we search for a Function in a Family of Functions using some systematic method. We don't just blindly. And somehow, the search has to be able to use the loss to constantly improve. So, you begin with a random guess, you check how good, or bad that guess is as a fit to my data, and based on that information, you improve your guess. So, you're constantly improving your guess. And that's really essential to a lot of Machine Learning. We repeatedly, we start with an initial guess, and then we repeatedly improve our guess, checking where every guess; how good, or how bad it is, as a fit to my data.

So far, I have not said anything about Neural Networks. And I'm not going to say hello. You might be shocked because a lot of people, when they hear the term Machine Learning, they think that Machine Learning is about Neural Networks. Actually, Machine Learning is much older than that. If we consider linearly squares to be a type of very Simple Machine Learning, that is, I think it's from the 18th, or 17th century, I need to go back and double check that, 17th or 18th century. The century is old. Other Machine Learning methods like PCA was invented early

20^{th} century. Fisher Discriminant Analysis is early 20^{th} century, quite old already, 100 years ago, and more.

What distinguishes Machine Learning are the features, or the aspects that I have already talked about. I think I'm running late. Sorry, forgive me. I'm going to run a little bit late. The features I've already talked about. That we have a problem, where we want to find a Function from a Family of Functions, and we have a measure of how good a fit it is, or it isn't, and a measure whereby we can search for a Function that is a good fit to that data. That's really what Machine Learning is about, I think, if you ask me.

Deep Learning involves a set of methods that compute very complicated Functions. These Functions are typically nonlinear Functions. I won't explain this. We can try to talk about it later, but it's a bit complicated. In any case, the idea of Deep Learning is based on the idea of Neural Networks. And a Neural Network is, we can say simply a graph where there's an input layer, and an output layer. And then these are connected to layers in between. The number of layers in between, indicates how deep the network is.

So, what is the input layer with your input data? For example, let's say your data are images, every little circle here is a pixel. Let's say the images are gray scale, or black and white. Every circle is a value of a pixel. The output is whatever we want as an output. Here, it's 3, I have 3 circles, but it could be, maybe only 1 circle if we want to recognize digits. So, the output would be the digit. And then what we have is that the data is passed through multiple layers until we get to the output. Basically, this network is computing a Function, but a very complicated Function from the input layer to the output layer. Again, it's all about computing Functions, inputs to outputs, but it's doing it in a way that involves a lot of intermediate stages.

Now, maybe this is a bit unclear. So, let me explain it with a simple example. This is an image from a project of mine, where I use a system of filters to analyze the lines in images from films. I was going to show you a video, but I don't think I need it. I think this may be... images said that. So, let's say this image is an input, it's an image from the great Japanese director Mizoguchi, Mizoguchi Kenji, from the film; The 47 Ronin. Great film! Let's say, this is my input. And let's say that in my first layer, what it does is, it applies certain filters to the input image, and then produces an output. Now, in this case, the filters are not learned, they are designed. It's based on the work of a scientist called Gabor, Hungarian scientist. The filters are designed so that some filters respond, as you can see to diagonal lines, some filters respond to vertical lines, some respond to horizontal lines; some respond to lines moving in different directions. So, you have an input as an image, which is an image, and then a layer of filters that outputs a number of processed images from that input. Now, ignore the last column on the left.

Now, what you could have, imagine a deep network, is many such layers where the output of one layer is perhaps processed, and then further moves on down to the next layer, which then moves on down to the next layer, which then moves on down to the next layer. Okay. So, systems of filters, maybe in some layers, the results get somehow summarized, or combined together, for example, in this diagram; the last column you can say is like a summary of every row. So, at some point that could be such a step. And all of this is passed down through many multiple layers until at the end, in a final layer, we get some outputs that we want. The output could be, like I said, a Label, telling you what's in this image, it could be a Depth Map such as I showed you before, but let me go back to the... It could be a Depth Map like this one, the output. But we go through a whole chain of filters, and other operations that we perform, so that the output of one layer becomes the input of one layer, and then the output of that layer, the input of another layer, organizing many layers until finally we produce one single image, maybe the Depth Map, if we want to get a Depth Map.

Now, this is one type of Deep Neural Network called a Convolution Neural Network, but let's not think too much about specific types of networks on this introductory level. The point is this,

this example I showed you is not an example of Deep Learning, or learning in any way, because the filters have been predesigned. For example, they have been designed so that some of them respond to vertical, or horizontal, or diagonal lines. They have been predesigned, but imagine that we could have a system where you have an input, let's say an image. Usually, it would be many images that we would work with, but let's say an image as an input, and it will go through some filters, many layers of filters. The filters have not been predesigned, maybe in the beginning, they are random. So, they're not designed to respond to horizontal, or vertical, or diagonalize. We don't know up to what. And we go through many such randomized filters on other operations. And then as an output, we get a Depth Map. Then the algorithm will compute that Depth Map with the actual Depth Map in our sample, the real, the true Depth Map. Obviously, it will not be an accurate one, but it will then readjust all of the filters, that the next time that an image comes in, it will improve the output. It will be a better fit.

So, essentially, we are again, looking for a Function. What are the Parameters of the Function, the filters, the values of the filters, what filters they have; and the system will adjust the filters to respond to the images in a way that the output will be closer to the Depth Maps that we have as ground truth.

So, here we have a Function whose Parameters involve the Parameters of all the filters. And we might end up with filters that are not like this, that we have designed with a human idea. Some filters might respond to L shapes, because those are very important in the data that we're using. Some filters might respond to kind of circular, or curved points. There are filters that the system came up with in order to try to find a good Function that fits the data well.

So, here the Function is, again, the function that receives an image and outputs, the Depth Map, let's say, I'm just guessing, just saying something. And the Parameters of the Functions are the characteristics of all the filters inside. To search for a Function in this family is to find good filters that will produce a good Depth Map. Sometimes, it can even happen that the network is trained, and we typically train it over time. Using what we call epochs. So, we retrain it repeatedly. And training can take a long time, training can consume a lot of computing resources. I know someone, when he started to do Machine Learning Work, he noticed that his electricity bill went up, which is one problem with some of these methods; they take a long time, and a lot of computing resources.

But I want to stress that Deep Networks are not equal to Machine Learning. We don't need to use these methods. Sometimes using these methods can be overkill. Sometimes traditional method that is computationally less expensive works very well, or even better.

Now, sort of coming into the last section. Sorry that I'm a little bit late. Want to mention that when we talk about Machine Learning, there is a great deal of work that is involved, and that people don't talk about. This work is often invisible. Machine Learning is work. Partly, it's work done by the Machine, but it's also works done by human beings. One example is, there is work involved in, and there are decisions. There is work, and there are decisions that we have to make. People often assume that Machine Learning, oh, the computer figured out how to do this, or that. Well, maybe yes. I don't know if you can say that, but for sure there are human beings making a lot of decisions. One decision that I mentioned already is what Functions do I want? Do I want Linear Functions? What's the Family of Functions, for example, is big major decision.

Another one is, the data that we use. Very often, we need to process the data. I said just now that the input to the system will be an image, but very seldom that we just input the image like that. The image often has to be cleaned up, has to be adjusted. We might have missing pixels, I give you one example; very often, we represent images as numbers between let's say 0... I said before 0 and 1, but it's more common to do it between 0 and 255, 0 means black, 255 means white. Many algorithms would not work if we do that. We need to change, to reformat the image, so it becomes a value between let's say 0 and 1.

Sometimes you need to change the mean value of a pixel. For example, make the mean 0, and the standard deviation 1. You need different kinds of scaling, normalization, data cleaning. The image might be color, you might need to convert it to black and white, to gray scale, many things you need to do. So, there's a lot of work, and these decisions are not just an important simple decision, merely technical decisions. They're very important decisions that will affect the result that we do. And they're part of the work of Machine Learning.

So, again every stage of Machine Learning involves work, and a lot of the work is invisible, and decisions, many of them are invisible. So, very often people assume that some of the computer will magically produce certain results without people making important decisions, like how to clean up the data, how to process the data.

A key component of the decisions involves the datasets that we use for training. I stress that we often rely on datasets for training. We have to have datasets, otherwise there's no learning. Well, where do these datasets come from? Creating datasets is very time-consuming. It doesn't happen by magic. Where do we get it from? Who gets it? Who does the work? This is a very famous dataset called ImageNet. An ImageNet is widely used in Machine Learning. In fact, it very often happen that when you use a system, you want to use a system, and you don't want to train it yourself; you want to use a system that does something that you want, like producing a Depth Map. But it has already learned how to do it. So, it's been trained by someone else on a dataset. Very often, you'll find that for many applications, the dataset will be ImageNet. It's a very famous, important dataset.

And it's important to be critical about the dataset that we use. And maybe this is one of the most important points I want to make at the end of the presentation, is how is the dataset constituted? How is it created? So, this involves, who did it? was it slave labor? Where else do they come from? Were they grabbed from the internet? Do we have permission from the people who are showing these images? This was a big issue with the ImageNet dataset, because originally no consent was secure. Does the choice involve any values, or biases? All of these questions, and many others are involved in the constitution, the creation of the dataset. I want to recommend an essay. There's a whole series of essays on this topic by Emily Denton, and others on the genealogy, the history of datasets; and the focus on the so-called ImageNet, which is the one I mentioned before. Which is very often used.

So, if you go out and you start doing work on Machine Learning, and you want to find Models, Functions that have already been trained, or Neural Networks that have already been trained. Usually, many of them, not all, maybe not the majority, but many of them will be already pretrained on ImageNet.

Emily Denton and her co-writer noticed that the process whereby these datasets were constructed, is often invisible. If you go to a... but here's for example, an algorithm that uses a Deep Neural Network to segment an image. Second, divide it into objects of interest in this case. And you can use their model, which has been pre-trained. And if you go down, they tell you, you can pre-train on a dataset called COCO, but you can also train a new model using weights. The weights are the Parameters of the model, which I have been trained using ImageNet. I don’t know if you can see this, and this is something that happens over and over again. The models are pretrained on huge datasets, and nobody tells you where did the dataset came from, who collected them? For what reason? The whole formation is invisible.

And I think here, it's interesting to relate the theory of Machine Learning to the cultural studies, and social studies of science. And I'm thinking of the work of Bruno Latour, you might have heard of him who produced a perspective of science in action. He said that when we look at science, science has two phases. One phase is, science has something completed. We take the results, but we can also look at science in the making, and ask how was it made? For example, we say that the freezing point of water is zero degrees Celsius. We take it as piece of knowledge that is already known, but we can begin to ask, how did we come to that decision?

How did we decide what is called the freezing point of water? How do we de decide what is water, and water is not always pure? How do we purify it? How do we decide when it freezes? How do we measure the freezing point? For many centuries, people use mercury, but that was not always true. Why did we choose mercury?

The very interesting Korean scholar Hasok Chang has a book called *Inventing Temperature*, but we can apply the same thing to datasets. How were they created? What values do they embody? We look at it as a kind of science in action, or science in a making perspective. Open it up, don't just take it as a set of results that you receive, but as a process that you need to look at. It's not only the labor that's gone into producing these datasets, but also questions like the ethics of it, I mentioned like, if there are images of people, did we ask for permission? But there are also values, and ideas built into the datasets. For example, in ImageNet, the Image data is organized in a kind of hierarchy. So, for example, at the top level would be, mammal, going down to carnivore, canine, dog; and then different kinds of dogs. So, each one is a subset of the animal. So, it's like a hierarchical tree. It's maybe easy to do that for animals, because there's a whole tradition of Western hierarchical thinking that has condition, or brainwash us to look at things this way. But maybe for other things, it's not so easy to create a kind of tree hierarchy. And people have to make decisions, and how are those decisions being made? Who makes them? Why do they make them? What are the values that are embedded in that? And then if those decisions are then used to train Machine Learning Systems, then that might affect our lives in the future.

So, there are some assumptions built into many of our datasets. So, I think it's very important for us as Artists to intervene critically on the creation of datasets. Here's one example of our work that I like, by Anna Ridler, it's called Myriad Tulips. And what she did was, she created a dataset of Tulips, and you can see the source below. Just look up Anna Ridler, Myriad Tulips. So, work already is few years old, but still have major work. Basically, it was her dataset that she constructed of Tulips. So, the whole work was a display of the dataset. The dataset itself became an artwork. So, instead of relying on datasets that have been already there, that are available, and you don't know where they come from, she created her own dataset. And then later on, she used this dataset in other Machine Learning Projects, but it was her own dataset.

And I think it is important as much as possible to try not to rely on existing datasets. As long as we have to do it, I had to do it in some of my work, but I try to avoid it as much as possible. And the reason is that many datasets render the work that went into them, and the assumptions that went into them, invisible. It’s important to be clear about those things, and not to depend on work that we are not aware of.

In a certain way, this is a problem that faces us as Artists, not just in the context of Machine Learning. Thinking in terms of Western History. But I think it's become a global problem. If you go back many centuries ago, Painters often made their own pigments, and they manufactured the colors that they used. But I think beginning, I think late 19th or early 20th century, Artists started to purchase tubes of color that were already manufactured. So, the whole process of making the color is something that became a kind of black box for them. And the same thing happening here, that elements of the artistic process become taken away from us. And this happens if we rely on datasets that are there without asking how those datasets were created.

So, Anna Ridler did this exhibition where she created her own dataset. In most of her work, I think, if I'm not mistaken, she's very careful about the dataset that she's going to use. I have used ImageNet, but I try to avoid it now, because I'm very conscious of this. I'm not saying we must be very dogmatic about this, but I think as much as possible working on datasets is critical.

In coming to the end, so, the Critical Perspective that I'm thinking about here is the perspective of Infrastructure Studies. So, this is the perspective used by Denton in the essay on ImageNet. Another thinker from this tradition is Brian Larkin, but I think infrastructure studies is very important, I think for Artists now. So, what is the meaning of the word Infrastructure in this context? The term is here is very broad. It is about the concepts, and the equipment. We can consider concepts, ideas to be equipment too. But let me separate them. So, the Concept and the Equipment that make possible different kinds of knowledge, or work.

So, in the case of Machine Learning, one of the key elements of the infrastructure, the tools, so to speak, is Datasets. So, in order to look at datasets from the point of view of Infrastructure, we need to focus on two things. Number one, that infrastructure like datasets, they are not passive things, but they actually shape the possibilities of knowledge, and action that we have. You can see that they're active in a certain way. They cause certain effects in a certain sense, because they create the environment in which we work. And so, they condition what we can do, or cannot do, what we think we can do, or cannot do.

Second aspect is that, Infrastructure has often been built, and there's a process; and there's a history, and we need to be aware of that history.

So, these two aspects are important. And I think the perspective of Infrastructure Studies has highlighted this. This applies to datasets, but it does not only apply to datasets. It applies, for example, to the algorithms that we choose, the kinds of loss functions that we choose, the problems that we attempt to solve.

I'm going to skip the last part of the top, where I was going to talk about the meaning of the concept of a population, and of the idea of sampling. These are also very important tools, because we always working with a sample from a population. For example, if we're working with a training set of images, these often come from a population, and they have been randomly sampled. So, the Concept of a Population, and the Concept of Random Sampling, these are two very important concepts. Where do they come from? What does sampling mean? It wasn't always so clear. And if you go back in time to the early 20th century, there were debates about what sampling means, because we want to choose data that is representative of the population. For example, if we are working on Admissions, we want the data that we are using to learn our function, to be representative of the population that we are working on, the Hong Kong students, for example.

So, this infrastructure perspective, I think can be applied beyond datasets, and towards the aspect of Machine Learning. And to be aware of these two things, number one, that all of these elements are not passive things, but that they shape how we think, and what we do, and maybe what we want to do; so, our desire as well. And that all of these things have been built by people with a certain history, in certain societies, and people who have made decisions; and have certain values, and those values weigh up on their decisions.

So, whenever we look at all systems, Machine Learning Systems, we need to look at them as having a history, as being historically situated; and this applies to datasets, but to many other things. So, I'm using the term here, Poetics of Infrastructure, which is, that we as Artists, I think have a unique opportunity to highlight the infrastructure of the technologies that we are using. So, this would involve working with the kind of datasets that we're working on, the data processing structures, the kinds of functions that we work on. All the various aspects. We can look at them, first of all, historically, and socially embedded, but we can also examine them as sources for inspiration. And this requires that we as Artists engage with the infrastructure that we use, not simply take it as a tool.

And this is maybe the last thing I want to say as a kind of conclusion for this talk. That many Artists think that Machine Learning gives us tools that can help us to do great things. Oh, we can do Deep Fakes, let's do it, let's find some Machine Learning system that's been trained to produce Deep Fake, and let's use it. We can do that. All of us have done that at some point. And if you have a good reason to do it, by all means, do it. I know many good works that take this approach. You don't care much about the infrastructure, you just use it for a purpose, but I think as a whole community, the art world just focuses on that, on using predefined systems to achieve certain results. We are missing the critical opportunity to engage with Machine Learning as infrastructure, and to find the art in the infrastructure.

Now, the question becomes how to turn this Infrastructure into Art. And I think on that question, I have no answer. I leave it open for you. Thank you. Thank you very much. That’s it!

So, that's the end. I'm not sure whether there's anybody left, but I welcome comments, questions, expressions of hate; or disagreement. Anything you want to ask, you can either type on the chat, or you can unmute yourself.

By the way, I should mention while I'm waiting, that the last section was about the concept of our Population, and I wanted to recommend a book to everybody. It’s at the bottom of the slide. Alain Desrosieres, called the Politics of Large Numbers, which gives a very strong background on the History of statistics.

And one point that I haven't made in the talk, but is a kind of assumption to my talk, is that the History of Statistics, and the History of Machine Learning are closely connected. That in a certain sense, Machine Learning is a development from the History of Statistics. So, that's a really good one. Okay. Any questions? Anybody there? Okay. I will wait one minute, but if there are no questions, we can finish on time. Sorry for over running. I was supposed to speak on it for one hour.

Well, everyone, hope the topic was... I try to make it as simple as I could—

Audience: Sorry. There is one question here.

Hector Rodriguez: Okay.

Audience: Because, actually Machine Learning is quite popular in Devon now. Everybody seems to try to do something with Machine Learning. And I think the main problem is most of the Machine Learning actually looks very similar, because I guess they all use the same model, or some pre-train, or pre-design model. So, what is your view on the culture of using Machine Learning in Devon?

Hector Rodriguez: Well, the person who asked that question is already kind of like, I think understands already where I'm coming from here. Unfortunately, the Art world is motivated by fashions, and of course, Artists have to produce work that will get shown, and get accepted. So, their main emphasis is on using these pre-train models to produce things that will look good and be shown in museums. And of course, yes, they all end up... many of them are explorations of latent spaces, and they have the same kind of look, because they involve some kind of Regression Model that is continuous. So, you're exploring a continuous space. So, that allows you to produce images, such as... have a particular look to them, and kind of deformed, or morphing images.

And I think, it seems to me that as Artists, we need to change the way we work, we need to change maybe our desire, you could say. And instead of trying to output work quickly, if we want to really seriously work with Machine Learning, to go back to basics, and not simplifying the fastest way to use models, because in that case, we are never going to train our own models, we're never going to develop our own models; we're never going to think critically about the infrastructure that is involved. So, I think it has to be about going back, and learning Machine Learning from basics. And the way that I normally recommend it is, at the moment to start with Python, which is the language of Machine Learning right now, which you could use other languages, people sometimes use JavaScript, and it's okay too, but I would recommend using Python, and to begin with classical Machine Learning methods.

So, for example there's a library called SK Learn. So, it's a library that has a lot of very simple algorithms, not deep learning, and start there. And the reason why I say this means changing your desire, and your whole outlook is because you need the time to work through this. I consider myself a baby when it comes to Machine Learning. And I'm just sharing with you, things that I know as an Artist, but they come from my experience as an Artist, and it's taken me years to get to this point. And I'm not sure I'm very far. So, you need to be able to invest in the long term. That's really critical. Invest in the long term! And that's really hard when we, as Artists are always thinking of the next exhibition. So, those of us who have jobs in Education are in a way privileged, because we have a little bit more time, not a lot, because we are assessed every year. And I think that's a big mistake. But I think that time is essential. Maybe take time off, develop residencies where people will take the time to learn that way, with that in-depth way. Did I answer the question? I mean, I think that's really... it is about changing the way of working, changing your expectations, and changing the way of learning.

There was a time even before Machine Learning, by the way someone just wrote me a message. I will reply very soon to that. But it happened too in other cases, where new tools become popular, and people just try to find what can we easily do with these tools. It happened with Maximus P, years ago. There were a lot of Maximus P works that somehow look the same. Did I answer the question? Did the person who asked that question have any other comments?

Audience: No, I think it's enough. Thank you.

Hector Rodriguez: Okay. So, I've got someone writing to me saying; I have zero experience with Machine Learning, but with a little science data background, I could understand some of it. Oh, thank you. So, you're saying; although I'm trying to draw the emphasis of your sharing to reflect on my everyday practice, although nothing arts is still how we take knowledge as is. Yes, I think this goes beyond Art. Thank you for that comment. I do believe that we need to reform Education about technology, and... well, before I say that, I always jump to Education, because I'm very frustrated, but to respond specifically to this person. Again, thank you. Yes, do not simply take for granted the knowledge that is received, that has become normative, but to question the processes behind it. And I did not mention it in the topic, because I was running out, but I think the work of Michel Foucault, and especially his book, the Archeology of Knowledge has a very useful vocabulary to do that.

So, extending on this thought from this person's question to Education, I think we need to rethink how we Educate not only Artists, but people who are not Artists, for example, data scientists. This person was a data scientist. That elements of critical theory have to go into the data science training. And that involves a huge transformation of the way in which we are conducting Education. I think Education for Artists also needs to change in this context. That's a whole discussion. Thank you for the question. Anything else? Okay.

Okay, everybody. Thank you all for coming. And I should make reference talking about the Invisible Labor to Doris Poon who organized this... who managed the symposium. I mentioned Linda Lai, who's the Artistic director of the writing Machine collective. And Joey Chung, who also took part in the organization, and finally to Sam Chan, who was the technical genius behind the zoom talk, and who made sure that everything ran smoothly. So, to all of you, thank you very much.

Oh, by the way, I should say that obviously, we need to change how we think as Artists in another way, by the way, I forgot to answer the question. In the past, many Artists used to think that the artistic, and the technical are separate things, so they can develop an artistic idea, and then find some technical people to execute it for them. Whereas I think now, the most important trend is for Artists to commit to their technologies. And that means either writing their own code, or working closely with the coder. Sometimes, it's not possible to write your own code at all times if your project is very complicated, but to always understand the technical elements, and be comfortable with the Mathematical elements behind the works, behind your own work so that the technical, and the artistic are not viewed as separate.

And perhaps also, based on the spirit of my talk, that the Element of Critical Humanities is also not seen as separate, but all of these three elements are in a very close interplay. All right.

Okay, everybody, thank you for coming! And for still being here. And take care. Bye Cynthia. And thank you, Cynthia, for your question. I stopped sharing. And I think I end this session now. Bye.