Now here is a review. Now if you kind of look at the ML, machine learning, so you can sort of very broadly group it as what is called supervised learning and then what is called unsupervised learning. So the supervised learning, if you see, that is exactly the case which I told you just now. Let us say super resolution is one example. So what do you do? So in this case it is actually a classification problem. What I told is actually a regression problem.

Both cases you can have supervised situations. So if you see here, so what you are showing is you are showing a mug as input to whatever, you have a module to which you are showing it and then the output, you have various labels, you have got probably coffee, you have got tiger, I mean whatever, you have got a bunch of labels and then it is supposed to tell what this is. So it will say that it is a coffee mug with a certain probability and then maybe it might say that you show a tiger, you expect it to show tiger. But initially when you train, you actually show many, many images like this and then you say that that is a mug, that is how you train it because you show the image, you say that here is a mug, you show a tiger image, you say that that should get flagged as actually a tiger, show something else a car, you say that that should get flagged as a car and that is how you train it.

So this is completely supervised in the sense that somebody has to tell. So somebody has sat down and kind of done all this labeling or what is called annotation. Somebody has annotated this dataset. So you may have a million images there but for every image there somebody has said that this is an ant, this is something, this is a ship, somebody has done that. And depending upon the number of classes that you have to finalize, you may have a thousand classes like an ImageNet or whatever.

You have a dataset where you have a certain bunch of classes and you then keep showing examples and examples and examples and then there is a training procedure which we will see and after you train the network now you hope that even an image that was not in the dataset, if you show it to it, let us say a tiger but not exactly in this pose, some other image which was never shown to the network, it should be able to classify that as a tiger, that is the hope. So that is like entirely supervised. Now these slides are little bit old. Now within supervised, I mean so close to supervised are actually multiple things. What is something that is called weakly supervised.

And then there is something called semi-supervised. Now this is supervised means it is completely supervised. So by kind of weakly supervised what we really mean is that, for example, to just give you an example, it is like a surrogate task. What it means is that, for example, if I want to solve one particular problem but then I do not have the annotations for that but I have annotations for some other task and I want to use that supervision in order to solve a different task. You get that? That is like weakly supervised.

Because I do not have a direct supervision for this task that I want to solve but I have a supervision of something else in the sense that somebody has already done the work for another task. Some fellows have already done that work. But can I sort of use that

information in order to solve my task? That is called weak supervision because it is weak. So this supervision is like a strong supervision. I know exactly, supervised means that I know that for this image that is the label, for that image that is the label, I know that exactly.

Whereas in weak supervision there is a surrogate task which helps you solve the main task. That is like weak supervision. In fact, one of my students last year, he had done some work on generating stereo from actually monaural audio but then using image information. And you do not have stereo data for every audio. It is not possible because nobody records for every audio stereo and all.

So what you have to do is you have to take some other task. For example, what we did was we took localization of objects to be the task for which data annotation was already available. So for example, it is like saying that I have got in this room 2 instruments, there is violin on this side and there is some other instrument there. Now if I have an algorithm that can tell where objects are, it does not know anything about audio. That is not at all an audio related task.

It just tells that in this image there is violin there and there is some other object here. Now that, so the idea is that if I give you a single channel audio and if there is a network that can produce a stereo but for the stereo I do not have a match. I do not have data to match the stereo information. It could be wrong what it is producing. But how do I check whether it is correct or not? What I do is I give the stereo to another network which should be able to just use this information in order to be able to, it is like saying if I could have closed my eyes and I had to simply play some audio to me, can I tell where the instruments are? Something like that.

So, that is like a weak supervision because the localization I tell you, here is an instrument, here is an instrument but in the audio I do not have that stereo data. So then what it means is the first network should really produce fantastic stereo in order to be, in order for the subsequent network to be able to localize these guys well using the audio information. Something like that. Now these are like weakly supervised. Semi-supervised means there is labeling available for a small amount of, for a fraction of the data.

Like for example you may have a million images but maybe for say 1000 images I have labeling information. That means I have annotated information, I do not have for the rest. Can I kind of leverage that? Can I leverage that in order to solve, in order to generate labels for my other images and so on. That is called semi-supervised. And then, so what I am saying is, so when I say supervised there could also be spinoffs of that, spinoffs of that which you will probably hear when you read papers and all.

You would end up hearing some like weakly supervised, very likely you will hear that if you read some papers and all. Semi-supervised. And unsupervised is more like learning

sort of a data representation. Like a PCA. I mean you have all read PCA at some point of time.

A PCA what does it do? So you have a bunch of, let us say you have a set of examples for a certain class. What do you do? You construct a covariance matrix right out of that. And then after you construct the covariance matrix you get the most significant eigenvectors. Those are your principal components. Now that is a representation.

So that is like a representation for that object class. That is the best representation for that object class. Best in whatever sense, in a sort of a mean square error sense. Now PCA is basically a linear algorithm.

PCA is totally linear. It is basically matrix. Whereas here when you talk about deep learning, you mean, when you say unsupervised learning, what you really mean is a network. So simplest way of thinking about unsupervised learning is, so it is like how well should I actually represent my data. That is what I want. PCA is one way, but then that is all linear.

Which basically means that it must have certain issues with it. Because I am already constraining it to be linear. It to be completely a linear operation. Now if you think about, let us say, if I had face images with me and then if I wanted a face representation. The simplest thing to do would be to actually push it through a network.

We call it unsupervised because if I actually push the image inside and suppose I do, the simplest way is I just push it inside. Then I sort of reduce its size. What is called that bring it to a bottleneck layer. Basically means that I could start with 1024 by 1024 image. But then eventually I can kind of go through an encoder which will eventually bring it to let us say 256 x 256 or even less than that.

64 x 64 kind of a representation. And then I kind of decode it back. And then I say from this 64 x 64 I should actually write and I should be able to see my image back again. Now I need to know nothing here because the input image should come out exactly at the output image, at the output.

It may not be exact. You may have some loss along the way. But now this middle, the bottleneck where it is like this. It is come down all the way from a high dimension. It has come down to low dimension. Then it is again going back because I want to see the image back again.

So this bottleneck as it is called, there that is the representation that you have. That representation if you show different faces of the same guy, it will eventually learn a representation that is sort of invariant to certain things which it is supposed to ignore. For example, if there is just a little bit of noise right behind the face, it should not catch all of

that. That is not really a representation of the face. That is something that has happened to be in the image.

So you should learn to ignore that. Or if there is a lighting change, you should understand that that is not something that you should capture. The features that you want to learn should be actually invariant to kind of nuisance. So in the sense that it should not be sensitive to things like that. It should learn things that are intrinsic to that particular class or to that particular person. So such a thing is called unsupervised learning.

So here, it is like you can actually do a clustering. The nice thing about unsupervised is that using this basic information, you can do a clustering and stuff like that without even knowing labels and all. So that way people like for example, k-means clustering is one such example which I think we will do when we do a traditional method of segmentation. k-means is one such way.

So it is like totally unsupervised. Nobody tells you what is what. So it has to kind of figure out what goes where depending upon their similarity or closeness and so on. It will group them. And unsupervised learning is about given only data x, learn the inherent. So that is what I meant, intrinsic structure.

What is it that is really? So if I say who is this boy, I have to get his features and not get, in case you see him, I mean, I will get, gets confused by his hairstyle or maybe one day he grows a beard another he does not grow beard. So those are all, those are all things that I should learn to ignore and really catch the intrinsics of this person. So that is unsupervised and unsupervised you never need anything. You do not need to tell what should come out. All that you say is input goes in, the same input should emerge.

Now you do whatever you want in between. And how well you do this depends upon how well that particular representation is because what you could then do is you could use that representation in order to do various things. And depending upon how well that works you will say whether you have successfully done this or not. If you have done a bad job then that representation will not really work well. When you use it for doing a classification task or something it will be sensitive to things that you do not want it to be sensitive to.

Again nothing is automatic. I am not saying that you can just take a black box, throw something in, something will come out, well, things will get learnt. It does not work that way. Even though many of these things still, many people say that explainable AI is still sort of, we are still in the initial stages. People want to have explainable stuff but still we are kind of very far away from all that. The limited things that kind of people do is they say that attention is getting paid to those parts of interest and therefore it is explainable and so on.

But still far from really the kind of explainability that we as humans are capable of. So if you look at traditional approaches, these are things which we will do in more detail. Let us

shift and then hog is another kind of feature, histogram of oriented gradients. So in the earlier traditional approaches, the way kind of things used to be done was you would arrive at handcrafted features. By handcrafted what we mean is you decide whether I want to use sift here on this or whether I want to use hog here, whether I want to use surf here, what is that feature, whether I want to do a Harris Coronet detector, what kind of features I want I decide.

So in that sense it is handcrafted. And then I pick those features, for example if it is an image then maybe it is a sift or a hog or something which I fix. And then a classifier is then learnt independently of this. I mean really there is no great sort of a link between the classifier that is coming because what do you do typically? See for example if I have to do a classification of different kinds of cars let us say. Now what do I have to do? I have to get some features of each of these kinds of cars.

Let us say five types of cars. I have to get features out of that. And then once I have the features then I should be able to push them into a classifier. It could be a support vector machine or whatever and then I should be able to tell that these labels are not, so these features come together and they are for car 1, these features come together they are for car 2, you want to do that kind of a classification. But if you look at it the classifier was always sort of independently, it is not like right you know it depends upon what kind of features you choose. The classifier you could choose right any of the ones right which are available out there.

The features could have been any of the ones that you want to choose and then you would experiment and then find out it which one of these works the best. Similarly for speech right, the nice thing was that people still could identify features that are reasonably robust to things that you want them to be robust to. For example, shift right is it is known to be reasonably robust to pose. For example, right I mean if I change my face like this and if you are taking camera and suppose you are catching some feature here, then that feature if you say shift feature it will still reasonably remain invariant to my pose changes, to illumination changes, to expression changes. So, all those things were still embedded in that feature because that is what you want as an invariance correct.

Now, so even for speech for example, right it is not like it is not like right you take an audio signal, you kind of compute its Fourier transform and then you start extracting things right. Even for audio there were handcrafted features. I do not work in audio, but I know that right there is something called you know Mel-Septal Mel-Frequencies, Mel-Septal Coefficients that is this MFCC right. Those are known to be very good to kind of capture audio characteristics and then you would have a classifier right or for example, right I mean or if it is NLP is Natural Language Processing right. So, most people would have their bunch of features, but everything was like a standalone.

The features were handcrafted, the classifier was again something you would decide and

then bring them together, you marry them ok. But then you know, but then right so in a sense everything was so like what I said it independently the classifier comes in as a sort of an independent module. So, this kind of a compositional feature abstraction right was not there. So, by which we mean that you know a compositional is like what I mean you know suppose I want a C right I want C to be A composed with you know B or let us say D is A composed with B composed with C right. I mean that kind of a compositional idea was not there right.

You just had you know something and then you had something else and then you would take these features put them into the other module get the output. Whereas a deep learning right is typically has this compositional form ok. I mean even though these are all nonlinearities or nonlinearities and all that, but at a basic level that you can think of it as several sort of you know compositional levels that are in play. For example, you could have the initial layers extracting low level features right like it is shown here. Then you have a mid level feature all these are part of the same network right.

It is like you kind of extract a bunch of features given an image you extract a bunch of features then those features go to the next layer. So, the initial set of features are some low level features then they are sent to the right next level which is let us say mid level and again it these are just representative example right. So, your low level features could be low could be looking like this then your mid level features that would sort of you know aggregate all these features right that you have learnt at a lower level in order to produce something at a sort of a slightly higher level what you can call as a mid level feature. Then furthermore at one more layer or a bunch of layers come in and then they do what is called a high level abstraction and therefore, you get some high level features and symbolically right here it is shown that at a high level right you are you are able to see some what you say red circles, arcs, certain shapes right. Whereas at a low level right you do not you just get edge information.

The mid level maybe something in between somewhere you see something, but are not everything is very clear and even at even at high level it does not mean that every time you can interpret all of that, but these are all symbolic examples ok. And then and then and then followed by a classifier which is also a part of the part of the whole thing it is end to end there is a network that is playing right end to end. So, you have features features features and then a classifier right. So, the classifier and the feature extraction are all together right. So, for example, so, so, what will happen is what you hope will happen and what actually happens is that the features the right kind of features will come out right in order to be able to work for that kind of a for that kind of a classifier right whichever whatever whatever you have whatever is that particular say network right that you have in mind or whatever is that architecture that you have in mind.

So, all of this is working sort of end to end right. So, therefore, the right kind of features will will be extracted so as to be able to get a classification accuracy which is very good

right. So, so, in the sense that you know there is you do not have a discontinuity it is smooth right it is like saying which features will work best in order to have a highest classification accuracy right. So, therefore, you extract features accordingly right I mean I cannot tell that light I mean for example, take shift it may not like to take shift because it may feel that if I take shift then maybe red I will end up doing doing a red pretty badly. So, so, it figures out what are those features and that is why that is where it we lose a lose a you know a little bit of handle over the problem because we no longer can sort of what do you say right we cannot steer it right. We can only say what should be the architecture for example, right think about it right for example, if you have suppose you had let us say blur motion blur.

Now, if if you knew that right it was all things going on the road let us say it is for vehicles right then you know that it is going to be a horizontal smear vehicles do not do not jump like this right. I mean if you take a camera and stand in the road right and suppose in our whatever right what is that road that we have born when you are whatever you just stand there and then right you take actually pictures and something is moving very fast right you will see a you will see a horizontal smear right. You would not it is not very unlikely that you will see a diagonal smear very unlikely that you will see a vertical smear typically it will be a horizontal smear. Now your filters right when you when you actually build your filters for the for your network you can then probably right have them more more width rather than height right why why go for a square shape right. So, things like that you can think about it because that seems to fit the problem right I mean so one can just say that oh I have been right what is the problem I can take a 5 cross 5 kernel and then let this guy figure out, but that is that is just unnecessary work for for this network maybe maybe what will happen is it will eventually learn a kernel that is that is more or less one dimensional has very very weak weights elsewhere right.

It might actually figure it out that way I mean it might just decide that the other things are actually useless, but then all that means a lot of parameters to be learnt at the end of the day and then you may be wasting you know your resources. So, the idea is that if you can throw in your your ideas right into it and then you know how this problem is actually right what this problem is like and you can bring in some of the observations that you already have with you bring it into the right into the setup that helps right. So, that way that way you have a lot to contribute, but at the same time there are many things right which which you will find that you know are things right you wish you had we had more insights into.