Deep Learning Part – II

Lecture -18.2
The concept of a latent variable

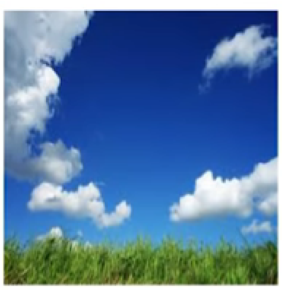
So let's, start module the next module, where we will talk about, what is a latent variable. How many of you have, had exposure to latent variables in some course or something before? Okay? Okay? A few very few, Okay?

Refer slide time :(0:29)

- We now introduce the concept of a latent variable
- Recall that earlier we mentioned that the neighboring pixels in an image are dependent on each other
- Why is it so? (intuitively, because we expect them to have the same color, texture, etc.?)
- Let us probe this intuition a bit more and try to formalize it

So, let's see, so earlier what we said is that? The neighboring pixels in an image are actually dependent on each other, that’s what our mark of network was and we kept the definition of neighborhood to be vague, whether it's just the left Right? Or top bottom or diagonal or even two rows and two columns heads, that's all up to us to decide. What kind of neighborhood we want to choose? Okay? So why is it so? Why do we say that, the neighbors of a pixel are dependent on each other? Because, we expect the color texture etc. To be the same for the neighboring pixels that's, the idea. So let us probe this intuition a bit more, and try to get some reasoning into, why this actually happens. Right?

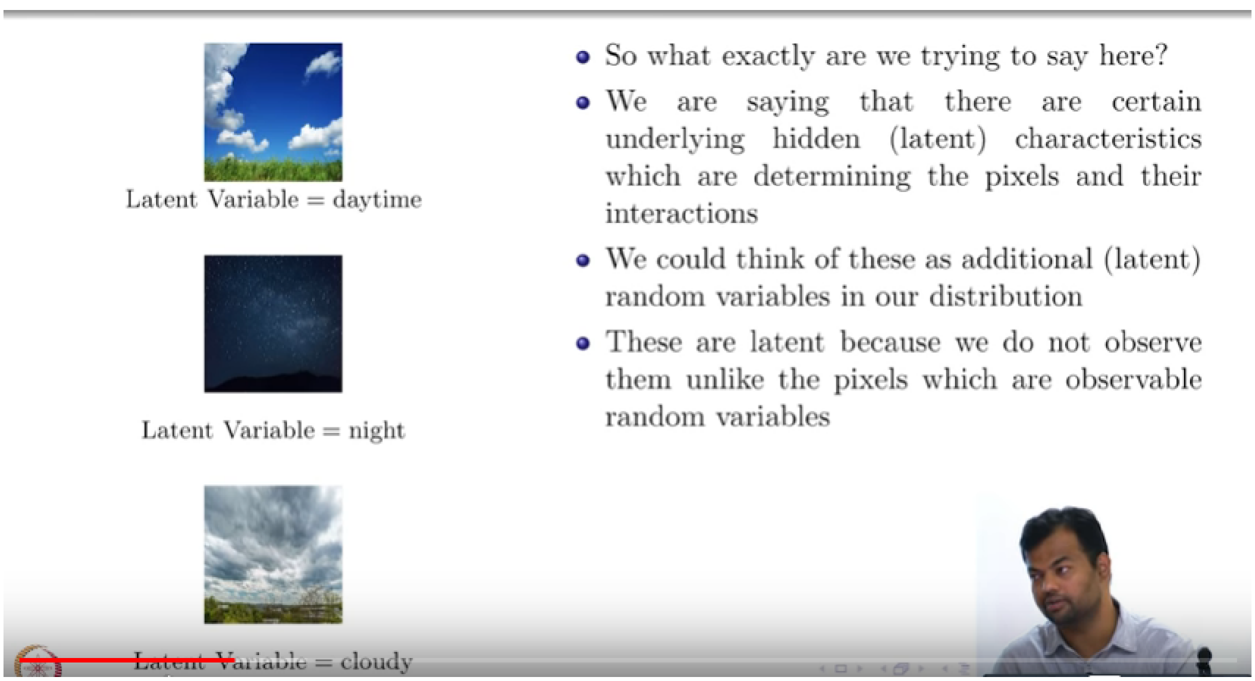
Refer slide time :(1:13)



- Suppose we asked a friend to send us a good wallpaper and he/she thinks a bit about it and sends us this image
- Why are all the pixels in the top portion of the image blue? (because our friend decided to show us an image of the sky as opposed to mountains or green fields)
- But then why blue why not black? (because our friend decided to show us an image which depicts daytime as opposed to night time)
- Okay, But why is it not cloudy (gray)?(because our friend decided to show us an image which depicts a sunny day)
- These decisions made by our friend (sky, sunny, daytime, etc) are not explicitly known to us (they are hidden from us)
- We only observe the images but what we observe depends on these latent (hidden) decisions

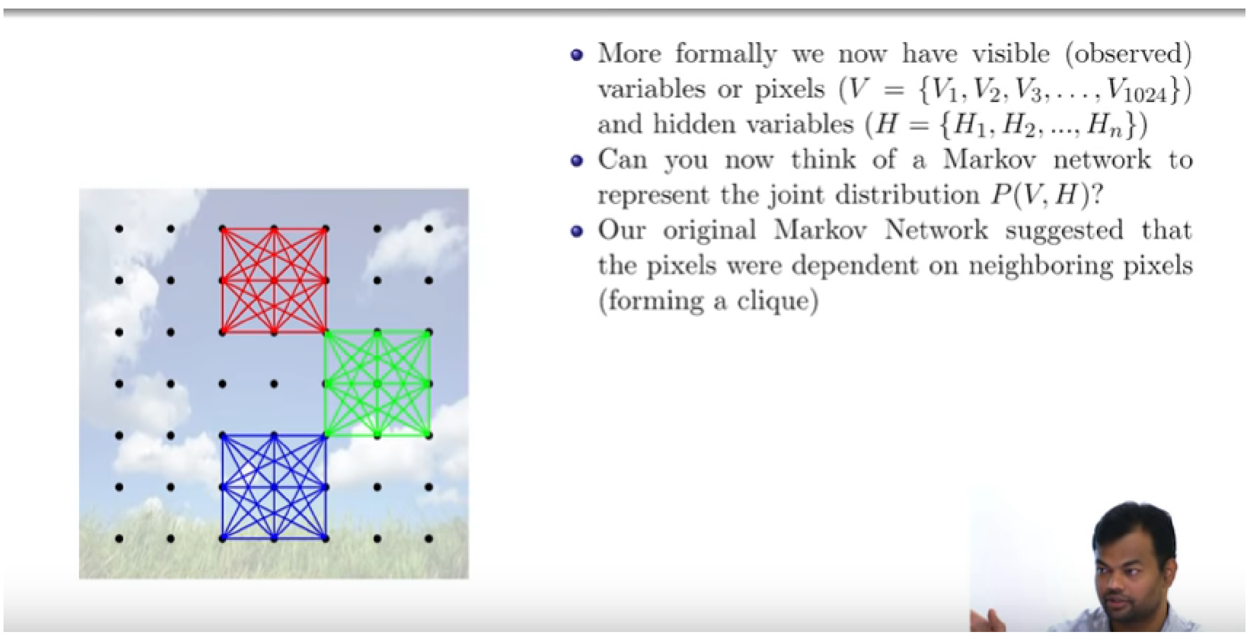
So we’ll take an example that, suppose we ask a friend, to send us a good wallpaper. And he or she thinks that, things about it and then, sends us this wallpaper. Okay? This image. So, why are the pixels in the top portion of the image blue? Why? Can you think on the friend’s perspective? Because, he or she thought that would be good to send us images which contain a sky, as opposed to mountains or green fields or maybe various other possibilities. Right? Why is the sky blue and not black? Because, he or she thought that it's good to send an image of daytime, as opposed to night time. Okay? Okay? Fine. Then, why is it not cloudy? I mean, it's a bit cloudy, but not, why not completely gray? Or when you can't even see any of the blue color? Why not that way? It's fine to send images of sky, its fine if you want to send daytime images, but, why not cloudy? Well again he or she made this decision that; I want to depict a sunny day. And a clear sky, as opposed to a cloudy sky or something. Right? Are these decisions known to us, what do we observe? Just them. Right? But, these are some inner end decisions that, the friend would have actually taken. Right? That I want to send and these decisions could differ and we will, there could be a very different explanation, for why this image was sent to us. But, there was some underlying decision, which led to this image. And we don't have access to that, decision we just see this, image. And what we observe is just the pixels; we don't have access to this underlying decisions that were taken. And I think all of us agree that, there was some underlying, decision which was taken and decision could be as simple as, randomly pick out an image. That could also have been there, where there was some decision, which led to this image being, generated. Right? So that's what, that's the idea that I'm trying to emphasize all. Right? You only observe the images and these decisions are hidden from us, we do not really observe them.

Refer slide time :(3:18)



So what exactly are you trying to say here? So, we are saying that, there are certain underlying characteristics. Which not only determine the pixels, but they also, determine the interactions between the pixels? Right? So now suppose, instead of this a clear sky, if a friend had decided to send us that image of Taj Mahal, which we had seen in some of the earlier, lectures. Right? Then the sky was not so visible that, the Taj male was covering most of the sky. Right? So the background was very little in that case. So in that case, the interactions between the pixels would have been different; the dome would have interacted with each other, it wouldn't have interacted with the sky, yet so all Right. So these latent decisions actually determine how these pixels are going to interact with each other. Right? And what we could do is, we could think of these additional, these latent decisions. Right? As, additional random variables in our, joint distribution. So, what do I mean by that is that? Someone decided that, of the two possibilities, sunny or cloudy, I would set the value to sunny. It was a random decision. Right? Someone decided that, instead of daytime, versus nighttime, I would pick it as, daytime. Is the same as high-low, high-low kind of examples that, we are doing. Right? Again someone decided that, the color should be of certain type and so on. Right? So these are, additional random variables on which certain decisions were taken, then we saw these observed, images which again contained random variables. Because, I have decided, I want to plot a sunny image. But, still there are several ways and I could, which I could have arranged the pixels to get a, sunny image. Right? I could have not just a sky; I could have a sky with something in the foreground and so on. Right? So, these in despite taking these decisions, there's still a randomness in what you will observe? And you could see different images of say sunny, sky with green fields and so on, it there could be so many images of this particular description. Okay? So, these are, these are latent, because we do not observe them, what we observe is only their effect? Which is in turns but, in terms of the pixels that we actually see or the images? That we actually see. Right?

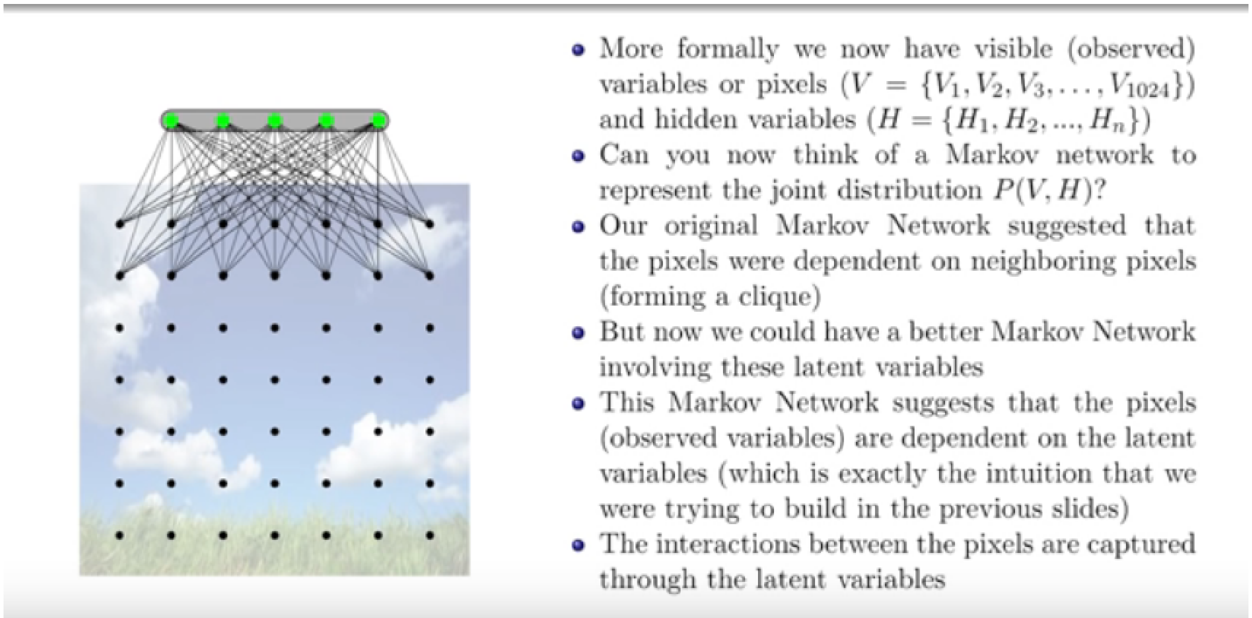
Refer slide time :(5:23)



So more formally, what we have now is, earlier we just had these observed variables, which were V. we have been calling it X, but now, I'll just change it to VN h, V for visible and H for it. Right? So, we had these observed variables 1, 2, 102 4. Because, it was a 32 plus, 32 image. And now in addition we are seeing that, there are some hidden variables H 1 to H N. Okay? Know can you think of a Markov network, to represent the Joint Distribution, P of V gamma H. first of all is this question valid, can I ask you to think of a Marko network? Yes. Right? Because, this is irrespective of whether I call it, V gamma H. or just collectively call it all of them as X. all we have at an abstract level is, a bunch of random variables and I am interested in learning their Joint Distribution. Right? And since, this was images and we have already made a case that, a name in the case, of images there are no directions, it's just the affinity or the interactions. So that's why, I am asking for a Markov network, as compared to a Bayesian network. that's again a modeling choice which I have made, I've assumed that these are not, the pixel interactions are not, dependent on each other, in the sensor there's no direction there, they're just interactions. Okay? So this is a valid question.

Now, can you think of a Joint Distribution for this? And what I'm asking for you is, think of a factorization. All first let's think about the graph, if H was not there and what was your graph? If H was not there, what was the graph? All the neighborhood in pixels depending on each other. Right? So this is the graph that we had. Right? And again the neighborhood is a bit vague, we can define it the way I want. And here, I am considering the diagonal neighbors also. Okay? Now, since that you have H. and I was trying to make a case that these, pixels are what they are? Because, someone decided or made some decisions, on these latent variables. Right? Now, convert that, first to a probabilistic argument that, what does what depends on what? And once you convert that, then tell me what the diagram would be? or what the graph ? How many if you get the set up? Like what, how many forget the question that I'm asking you? Isn't that a bit over specified, you said something plus something. Right? or both those things required, think in terms of parents or Markov blankets or things like that. Right? Think of it this way, if the first two pixels are blue and I know there was, some latent variable which was sky or sunny? Do I need to make these two pixels, dependent on each other, given the latent variable? So he said that, your clicks would be that's, one way of answering this question. The clicks would be these, nine pixels that you see here. And in addition, you'll also have all of these nine pixels, connected to the hidden variables, that's what you meant? Right? And I asked him that isn't this, over specified, you said that it's, click plus something, do you need both these terms. And then, I sort of gave a hint or I just made a statement. So now, concerning all of this, tell me what the mark of network would be? So he said, bipartite graph. What are the two partitions? I'd say me that's easy to answer. So what would you have? Now, given that's the answer, can you reconcile with everything that we have discussed, does that make sense? That what does it mean? For since it's a bipartite graph, what does it mean? And you said the two partitions are H and V. So do you have connections between the Vs? So do you have the cliques that is you see here? No. we have connections between the H's? No. so what are the connections that you have? Vs and H's. so some H's connect to some V's or all H's connect to all V's or what's the Assumption? All H's connect to all V's. Right? Even if you look at the example which, we were discussing the toy example, where there were only three latent variables, it was sky, sunny and daytime. Every pixel depends on all these three values. Right? If I change any one of these, the pixel will have to change. Right? So in that case, it has to be a fully connected, yes.

Refer slide time :(10:01)



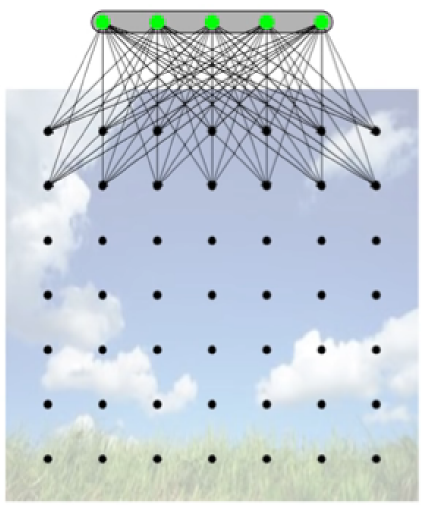
So then, you could think of additional random variables, there could be one random variables which says that, how many clouds are there? And what's the size of each clouds big or smaller so? This is a bit vague at this point. Because, see again and I'm going to come back to that point, there is a reason, why we call this as hidden variables? Because, we are not going to observe them in practice. Right? These were decisions taken by our friend. And I'm going to make the assumption that, you can't even asked her friend Right? So that, we will never know what these decisions were? This is only for the purpose of explanation. And later on, I will come back and say that okay, all this cloudy sunny and all was just for explanation, in practice, we don't really know what these hidden representations were. And I'll also make a case that, something like this we've already argued in the course before. That we don't know what these hidden representations? Okay? Then probably the story would become clear. Okay? But, that's a fair doubt and ask me again, if it's not clear after the ten slide, next ten slide also. Okay? So this is what remark of network would look like? Of course from some pixels, I have not drawn the edges; it would just become too complicated. But, just imagine that all these edges are also there. Right? So every visible pixel is, essentially connected to every hidden pixel. so that's the correct answer, it's a bipartite graph and the reasoning is that, once you know the hidden variables that, completely determines the interactions between the neighboring pixels, so you don't need to capture them again. So, I mean, just try to visualize is the way you are comfortable, either in terms of a Bayesian network, we are given the parents, you are independent of the siblings are independent of each other. Or in terms of a Markov network, we are given the Markov blanket, which is all these hidden variables; you are independent of all the other visible pixels. In either reasoning, it should be clear that the visible pixels are independent of each other. How many of you get this? Okay? and this is the intuition that, we are trying to build on, the previous slide that, these are some decisions, which have been taken by someone, it doesn't matter that we don't have access to these decisions, it does not matter that we don't know what these decisions are. Or we don't even know the definition of these random variables. Right? I don't know, whether the first random variable is actually sunny or May, Day or cheerful or happy or what? I don't know these; it's just that there are some latent variables. And my final observations are based on these, latent variables. Okay?

Refer slide time :(12:14)

- Before we move on to more formal definitions and equations, let us probe the idea of using latent variables a bit more
- We will talk about two concepts: *abstraction* and *generation*

So, let us probe this idea a bit more. And we will try to talk about this idea, in terms of two concepts: one is abstraction and the other is generation. So let's see, what that means?

Refer slide time :(12:25)



- First let us talk about *abstraction*
- Suppose, we are able to learn the joint distribution $P(V, H)$
- Using this distribution we can find

$$P(H|V) = \frac{P(V, H)}{\sum_H P(V, H)}$$

- In other words, given an image, we can find the most likely latent configuration ($H = h$) that generated this image (of course, keeping the computational cost aside for now)
- What does this h capture? It captures a latent representation or abstraction of the image!

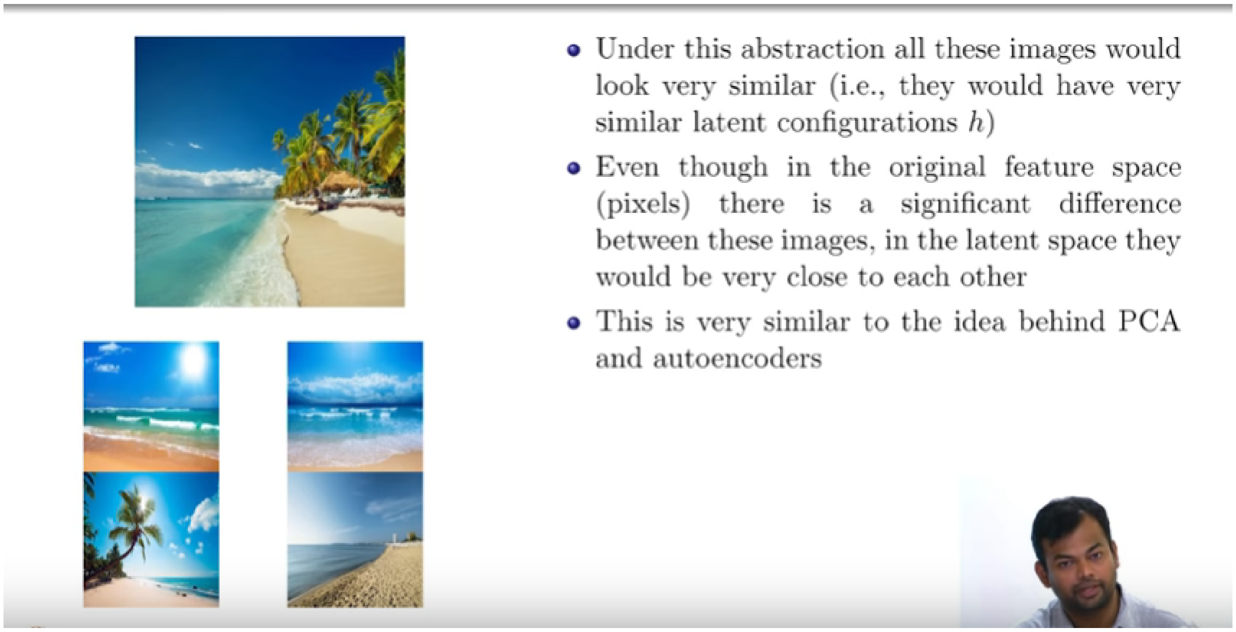
First let us talk about abstraction. The suppose we have learnt, the Joint Distribution P of V comma H. Okay? And this should be obvious, that given this Joint Distribution, I can compute this, P of H given V. in particular; I could compute for a given V, I could compute the hidden state, which maximizes the probability of P of H given V. what’s the English way of saying that? In terms of the decisions made by a friend. Right? Given an image, I can actually assume that, the Joint Distribution is given to go. Right? So, if I can compute the R max B of P of H given V. what am I actually computing? The most likely hidden decisions, which the friend had made, to give us this visible observation. Right? Okay? So here, as I said that, most likely decisions that were taken and at this point, I'm keeping all the computational problems aside. Right? Whether this Joint Distribution, struck table and all those things, I am keeping aside. I am just assuming someone has given to you and you have infinite compute, power to compute this, then this is what it means? So, what does H capture? Now, try to relate it to other things that, you have done in this course. What does it capture? H should give it away. Right? What does H capture? it captures some abstract representation, of the image. Right? Fine. And still there are some things missing here, but we get there.

Refer slide time :(14:05)

- In other words, it captures the most important properties of the image
- For example, if you were to describe the adjacent image you wouldn't say "I am looking at an image where pixel 1 is blue, pixel 2 is blue, ..., pixel 1024 is beige"
- Instead you would just say "I am looking at an image of a sunny beach with an ocean in the background and beige sand"
- This is exactly the abstraction captured by the vector h

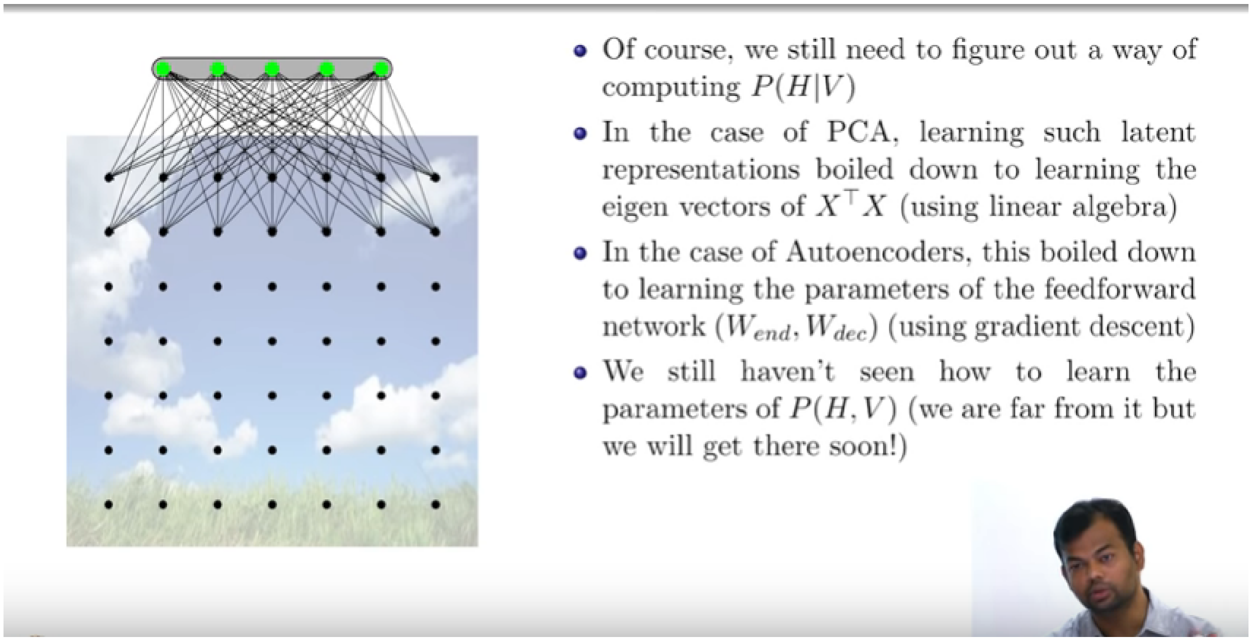
It captures an abstraction of the image. Now, under this abstraction, what would happen to images? Okay? So let me ask you this, so this abstraction is capturing the important properties of the image. Right? So what do you expect to happen to images which look very similar? The abstract representation would be very similar to each other. Right? So, for example, Right? If you were to describe this image to someone, you wouldn't see, say that I am looking at image, whose pixel one is blue, pixel two is blue, all the way up to pixel 1024 is beige. This is not how you are going to describe it. How are you going to describe this image? I can see everyone imagining and dreaming and so on. But, yeah! Let's assume that, you're not going to get there anytime, soon but still describe it. It's a sunny beach, with an ocean, I don't know why it's an ocean, but, in the background and beach sand. Right? Instead of white sand. That's an abstract representation of this image, that's how you would abstractly describe this image, the 1 0 2 4 pixels are too dense. Right? That's a very over specified description of the image, you're not interested in that, I just insist interested in this abstract representation. And this is the kind of abstraction, which we expect H to capture. Okay? Again I am, building this up, saying that it will capture all this and then I'm going to play it down later on, but, I'll do justice to it, when I play it down, it I will relate it to something that you have already seen. Okay?

Refer slide time :(15:34)



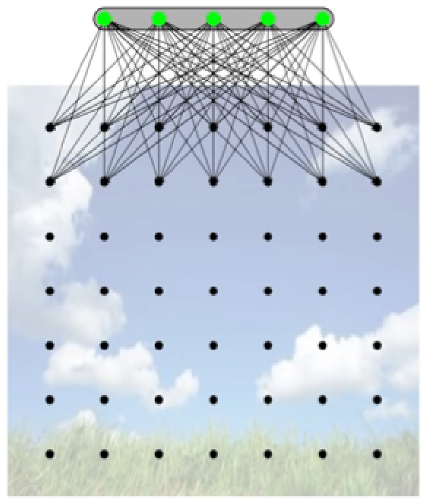
And now, under this abstraction, what would happen to all these similar looking images? They will all have a very similar representation. On a poor pixel basis are these images similar to each other, they're very different orientations of the beach and like, they are different, even let's not just get them, they're just different. Right? Now, in the per pixel basis, if I take the squared difference between them, they would be different. But, under this abstract representation what do you think they are? They are very similar and we are always interested in these abstract representations, because that captures the important properties of the image that we are interested in. or the data that we are interested in. Have you done something like this before; the correct answer is throughout this course. Right? I mean, deep learning is, deep representation learning. Right? The right term is deep representation learning. So, we did this in auto-encoders, we did is a multi-layer perceptions site, we said that, every layer captures a different abstraction of the image, we did this in convolution neural networks, where every layer of the convolution your network, captures a different abstraction of the image. Right? And so now, let's this very similar to the idea behind PCA also. Right? Now, that is where now I'm going to play it down abite.

Refer slide time :(16:45)



So, we still need to figure out a way of computing this P of H given V. Right? We still need to find out that, if I give you an observed image, how do you compute the arc max H, which is the most likely hidden configuration which generated this image, which is the same as saying and what's the most likely hidden representation for this image. in the case of PCA ,this boiled down to learning the eigenvectors of X transpose X, in the case of auto-encoders, where you again learn this abstract representation, what did it boil down to? What was the learning that you did there? What did you learn their? Nothing, these sudden blackouts are completely inexplicable. What did you learn in order in collision? Hidden representation. But, what so in the case of PCA, you learn the eigenvectors of X transpose X. In the case of auto-encoders what do you learn? The parameters of the network, the parameters of the encoder and decoder come on. Right? W encoder and W decoder. And we still, but now the analogy here would be, what do we need to learn here in this case? The dash of the joint distribution. The parameters are the factors of the Joint Distribution. And we have still not seen that, we are far from it, but we'll get there eventually. Once we know that, we have the answer to the first bullet, we can compute P of H given. Right? And that is again the motivation for learning a joint distribution, because once you have the joint distribution, everything else can be done from that. Right? All sorts of questions that you want to ask about, those set of random variables, you can compute from the joint distribution. Right? So we are still not seeing, how to learn that? But, there's an analogy, you have seen it in PCA, you have seen it in Auto encoders and we will see it, soon or in a few lectures, before this course ends, how to learn this joint distribution? Okay?

Refer slide time :(18:26)

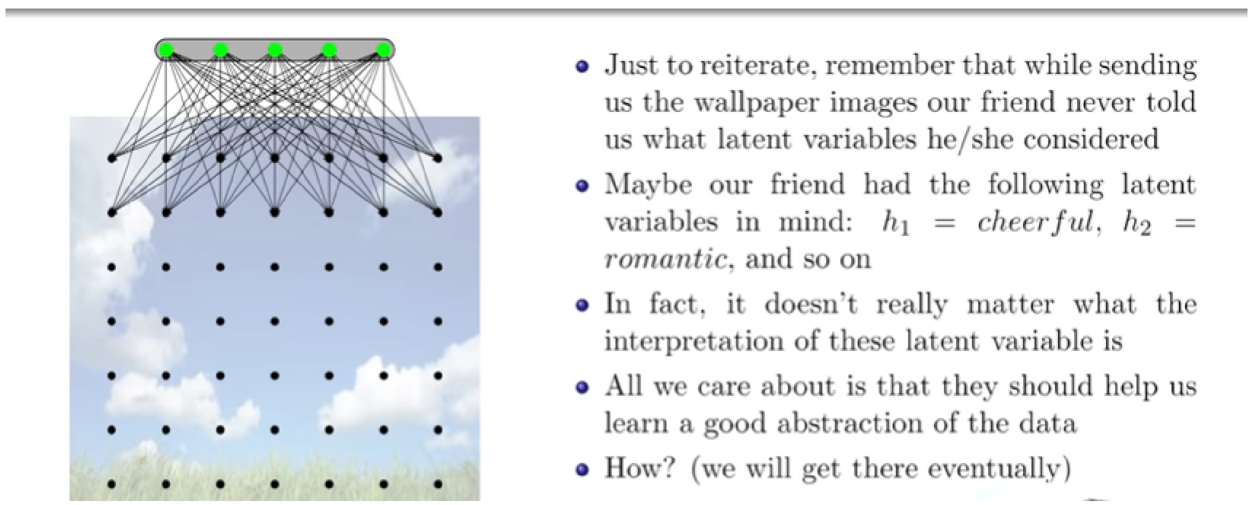


- Ok, I am just going to drag this a bit more! (bear with me)
- Remember that in practice we have no clue what these hidden variables are!
- Even in PCA, once we are given the new dimensions we have no clue what these dimensions actually mean
- We cannot interpret them (for example, we cannot say dimension 1 corresponds to weight, dimension 2 corresponds to height and so on!)
- Even here, we just assume there are some latent variables which capture the essence of the data but we do not really know what these are (because no one ever tells us what these are)
- Only for illustration purpose we assumed that h_1 corresponds to sunny/cloudy, h_2 corresponds to beach and so on

I'm going to drag this a bit more. So, in practice we have no clue, what these random variables are? What these latent variables are? As I said, you're not talking to your friend; you're not really asking him or her, what was the decisions which led to this particular image? Right? And in fact there could be many explanations for that. But, this is not something new, this is exactly what we saw in PCA also, in the case of PCA, the original dimensions of the data had some meaning, that the first dimension is weight, the second dimension is hide, the third dimension is salary and income tax and all those things. But, once you transform the data into a new space, those dimensions had no meaning. Right? You cannot say that, the first time mention corresponds to a certain thing or the second dimension corresponds, you cannot attach labels to these dimensions, all you know that, these are dimensions, which are independent of each other and they have a high variance, along these dimensions.

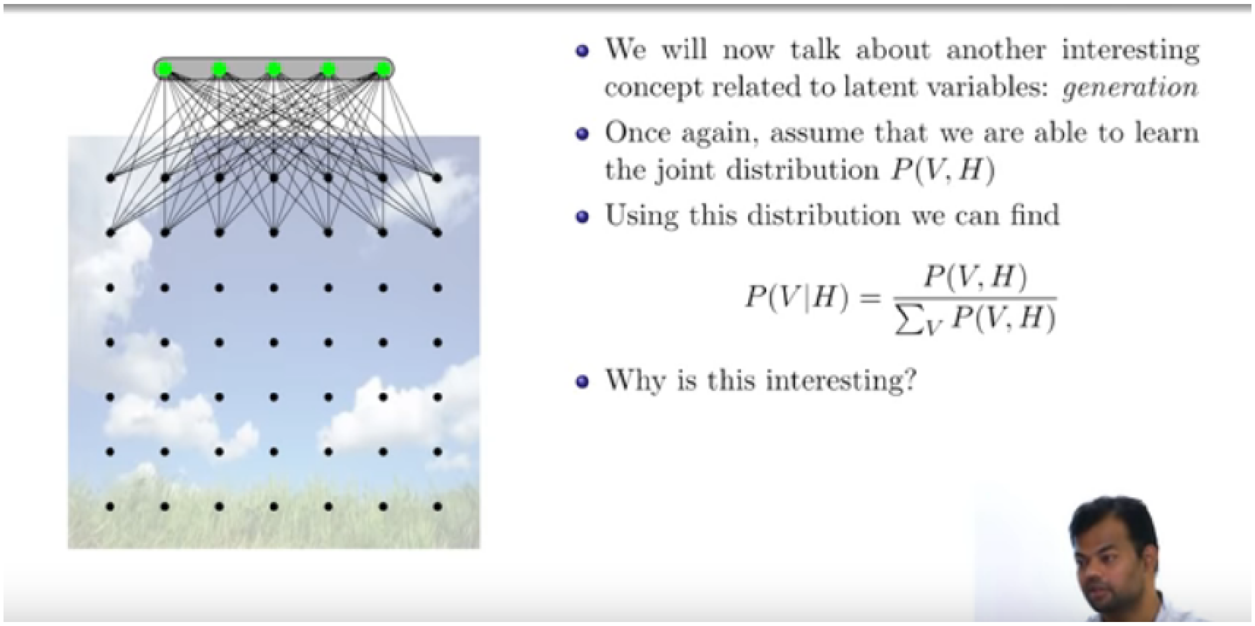
Those were, the richness that you had for PCA, the same thing is true for, even auto-encoders, when you learn an abstract representation for the auto-encoder, your original data hide certain semantics, but, once you get the hidden representation, you just know that it's a hundred dimensional representation, you don't really know, what each of these hundred dimensions are there? And the same thing applies here, you don't really know, what these latent variables are? you just know that ,there are some latent variables, just as in those two cases, you knew that there is a certain hidden representation, which is able to represent my data better, even in this case you know that, there's a latent representation which actually gives them more succinct representation of the image. But, you don't really know what, are the semantics of this latent variable, so all you know that, maybe all images come from a hundred dimensional space, instead of a 1024 dimensional space, once I have this 100dimensional values, I can capture everything that is there in the image. And that was the idea behind PCA: that was the idea behind auto-encoders. And that's the behind this particular structure, this latent variable is capturing the essential semantics of the image. Right? Even though, we can't really, identify what these variables. Okay? So this is not something new to this, this is what we have been dealing with throughout the course. Okay? Only for illustration purpose I've been saying that, this is sunny, this is cloudy and so on. But, that just for explanation, none of these variables have any meaning. Okay? That fine. And actually, it does not even matter it. Because, it could have happened that, even though our friend did not tell us, his or her decisions.

Refer slide time :(20:48)



He could have thought of something very different, maybe the image was of a beach or sunny or so on. Because, they wanted to convey, something which is cheerful or romantic or something else. So these, dimensions could have taken some very, very different values and what we thought they are? But, it doesn't matter, as long as there are some latent variables, such that under this latent space, similar images become similar or have a very similar representation, then we are fine with that. We don't really need to, actually define these latent variables. Right? I know I'm repeating myself, it's very important that you understand this concept .you can't say things about visible variables that, this is pixel one, this is pixel two and so on. You can't make these arguments about hidden variables, you can just say that there's latent space, which captures the semantics of the data. Okay? And again I emphasize is not something, new which I am just throwing up at you, this is what has been a dream theme throughout the course? Whether in auto-encoders all convolution neural networks or any kind of multi-layer perceptions'. Right? Okay? So how do we learn this? So what is the question that I'm asking? How do you learn this? Right? So I'm going to keep, track of all these questions, which I'm going to say that, we'll get there eventually. And all these questions would essentially be saying the same thing that, how do we learn this Joint Distribution H comma V? I'll keep asking these questions and I will say that relook at it eventually, you will find that there are many questions, which I am saying, which we will see eventually. But, this all boils down to one single question, which is how do we learn this to end distribution? So you already saw, one question. When I say that we are far from learning it. And that was how do we learn this Joint Distribution, this again when I'm asking, how do you get the hidden representation? Again boils down to how do you learn the Joint Distribution? And once you have the Joint Distribution? Inference is straight forward.

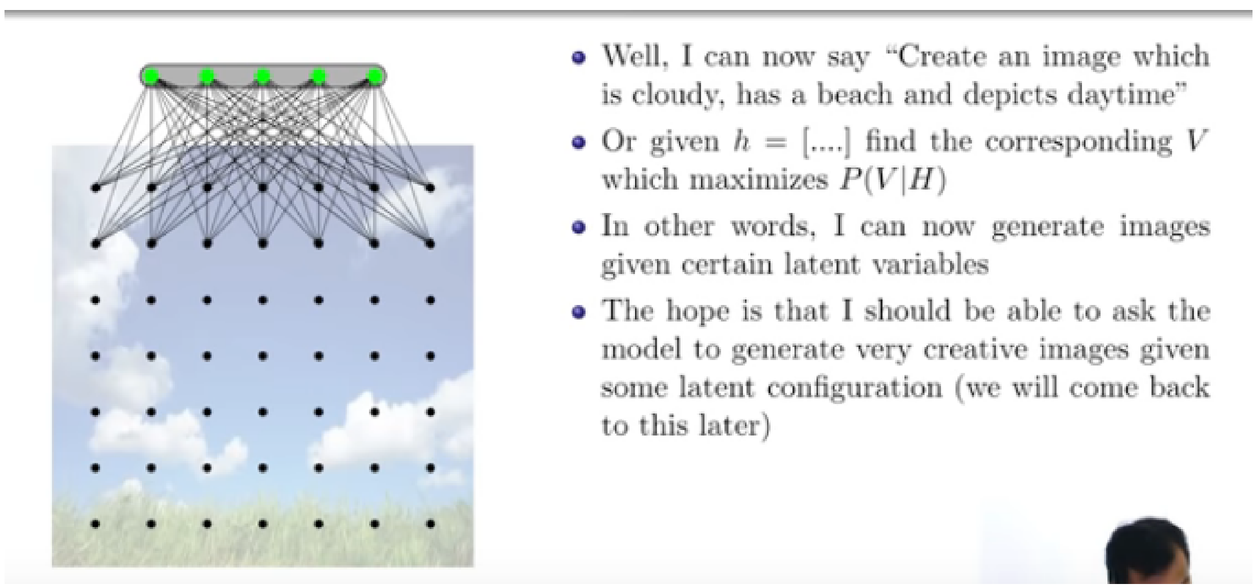
Refer slide time :(22:28)



Now, this was about abstraction. Now, let us talk about, another concept related to latent variables, which is generation. Okay? Once again assume that, we are able to learn the Joint Distribution P of V comma H. Now, from this distribution, I can find the following thing, I can find R max V given H. now, in English what is the question that I'm asking here? Given hidden representation, generate a image which are there as to this, why is this interesting, just think about this, for a minute, I'll give you a few hints here. A large amount of data and let's say all this data what about, scenic pictures. Right? And mainly say, skies and oceans and green fields and so on. From this data, you have been able to learn P of H given B Right? That means for a given image, what was the abstract representation? That led to that image. Okay? From this data you have been, if I give you any image, you can give me a H, the vector H for that image. Now, I'm asking you the reverse question, that you can also do this, you can also do P of V given H. Now, given that you can do this, I want you to be creative enough, to give me a good use case for this, yeah. That's probably. Okay? but I ask you to be a bit creative, reconstruct from a distorted image, yeah that's fair, for some other, other ML problem, no I mean, I've given that skies and beaches and soon. Now, my other ML problem is I want to classify cats and dogs, the hottest problem. How would I do that? From here, so suppose this image which is not visible. Right? You have done training everything is over, somehow you have figured out how to learn this Joint Distribution.

Now, you can actually compute an H, for this image. Right? Wouldn't you be interested in knowing that, if I perturb this H a bit, what kind of an image you like it? You want to generate, this is a image of a sunny beach say, you want to generate other images of sunny beaches ,you take the hidden representation of a sunny beach, just perturb it a bit and see if you get a different kind of a sunny beach, does that make sense, does that make sense. How well you can do it? Depends on how much data you have, how effective your learning was whether you actually learnt, the parameter as well let's, say you can't give me 100 images of sunny beaches and say that. Okay? Now, I'm going to go and generate thousand more that won't happen. Right? I don't even know what's the Right? Number. Right? It's 100or a million or 10 million or 100million I don't know. But, asymptotically. Right? If you had enough data, can you actually do that? Right? That's what people are interested and that's one of the pipes, around AI Right? So, this creative AI that, can you create and now I can extend this too many things. Right? Suppose I learn a joint distribution of poetry's. Right? Now, you can imagine hidden variables for poetry's also. Right? It could be tragic, romantic or it could be, about nature philosophy or so on and what not. Right? Now, given a lot of poetry's, if I can learn a Joint Distribution between these hidden variables and poetry's. Now, take some poem, get the hidden representation of that poem and now, trying to generate other poems which look like that, they're all still slightly science fiction is, it's not that this has been solved and people are doing this very, very well. But, people are doing it to a certain extent, which is appreciated. Right? So, it's not that it's been solved. But, at least there is a scope for being creative by doing these things once you'd learn a joint distribution. So, essentially we're trying to learn, how do people create images or how two cameras click images and so on. And now, given these things, can I create more images, can I create more text and so on its. So, that's the one of the goals which are being pursued, right now. Okay? So, that's the overall context of why you need this generative model lights or generative model actually tells you, how was the data generated. So, that if needed you can generate more data of that. Right? So, that's the overall bigger picture that you have here.

Refer Slide Time :(26:46)



So, this is what I can say now. Right? "Create an image which is cloudy, has a beach and depicts daytime" This is in English, in the vectorial sense, I would say that take the Sidon representation and give me another image, which could have generated from this hidden representation or computer identification from a certain image, perturb it a bit. So, that may be cloudy, becomes slightly cloudy or more cloudy less cloudy or something like that and then generate new images from, even without actually knowing what these hidden variables are, can you still do this, yes you can. Right? Because you have the visible image, this is the hidden representation for that, all I'm asking is to generate more stuff, which are these two this hidden representation. Right? Without actually knowing the semantics of a certain resolution Avyon gets this, Avyon idea that you don't need to really know the semantics of the hidden representations, please raise your hands if you do that. Okay? So, this is what I would give it? I'll give it a vector, not the text description, I would give this vector to it and I will ask it to generate. So, again we'll come back to this later, it's again the same question, once we answer ,what we can do with P of V comma H? How we learn this, all these questions will be answered, you can keep a tree track of these things which I'm saying, we will do later if you want, if I forget something let me know, but, I'm pretty sure we will cover everything so. Okay?

Refer Slide Time :(27: 59)

The story ahead...

- We have tried to understand the intuition behind latent variables and how they could potenatially allow us to do abstraction and generation
- We will now concretize these intuitions by developings equations (models) and learning algoritms
- And of course, we will tie all this back to neural networks!

So, the story so, far has been that, we have tried to understand the intuition behind latent variables and how they could potentially allow us to do both abstraction and generation. Right? And both these are interesting concept, abstraction we have been doing throughout the course and we have kind of convinced ourselves that it could give us better, representations for the data, which could eventually, lead to better predictions on that data. Right? That's one thing and we will now try to concretize these intuitions, by developing some equations or models, which allow us to capture all these things and the corresponding learning algorithms. But, whenever we introduce equations, what are we going to introduce? Parameters and then once you introduce parameters we need certain learning algorithms to learn. Right? And of course this is a course, on deep learning; you have to tie all of this back to neural networks! Right? We have just completely deviated from that, just talking about random variables, I will stop this randomness and get back to deep learning. Right? Okay?

Refer Slide Time :(28: 51)

- For the remainder of this discussion we will assume that all our variables take only boolean values
- Thus, the vector V will be a boolean vector $\in \{0,1\}^m$ (there are a total of 2^m values that V can take)
- And the vector H will be a boolean vector $\in \{0,1\}^n$ (there are a total of 2^n values that H can take)

So, for the remainder of this discussion we'll assume that all our random variables or the visible variables, take on Boolean values. Right? So, V is a vector from 0 to 1, raise to N and similarly all our hidden variables also, take on Boolean values again its 0 to 1extreme. So, we have invisible variables, instead of this one zero to four that I've been talking about and M hidden variables, for the remainder of this description. Right? And at some point, I'll also say that X, I'll use X to denote V comma H together.

Right? So, when I want to refer to them collectively, I'll just call them, ‘X’. So, X is again a vector of random elements.