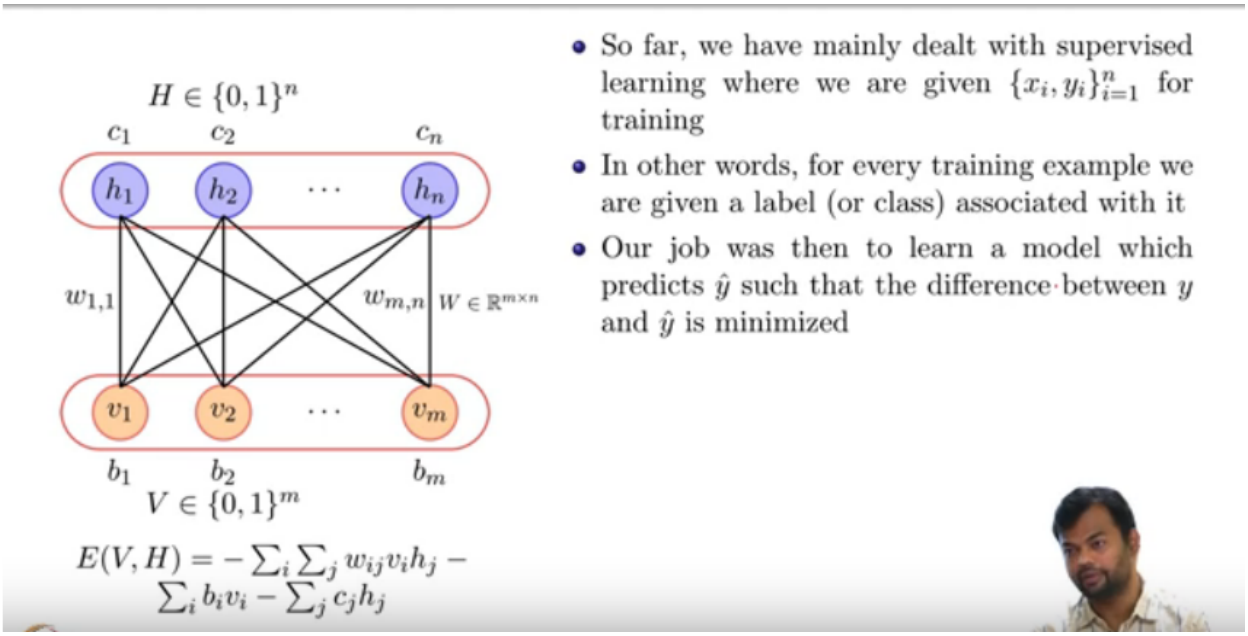
Lecture - 18.5

Unsupervised Learning with RBMs

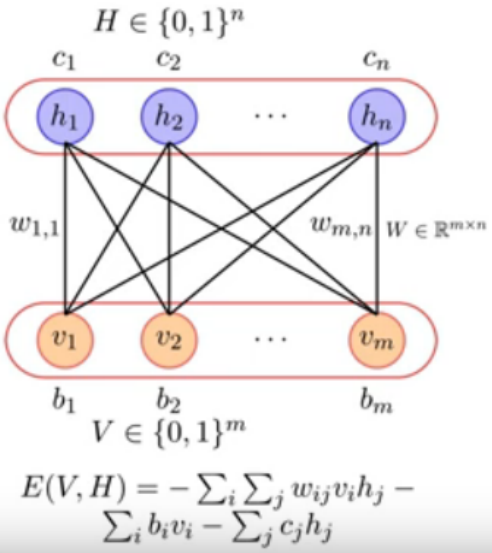
So now we will talk about, unsupervised learning and that's what the buildup was about that in the case of our beams we don't have vise, you only need to learn from the excess. Okay? That's what we need to remember. Okay?

Refer slide time: (0:26)



So, so far we have always dealt with situations, where we have X I comma Y I and then, it was easy to define some kind of objective function and the two popular ones, which almost got us through the entire course were squared error loss and cross-entropy loss. Right? But now, we don't have vise, for the training label examples, we don't have a label associated with. Okay? So now, instead of the difference between, y and y hat, which has been our story always?

Refer slide time: (0:50)

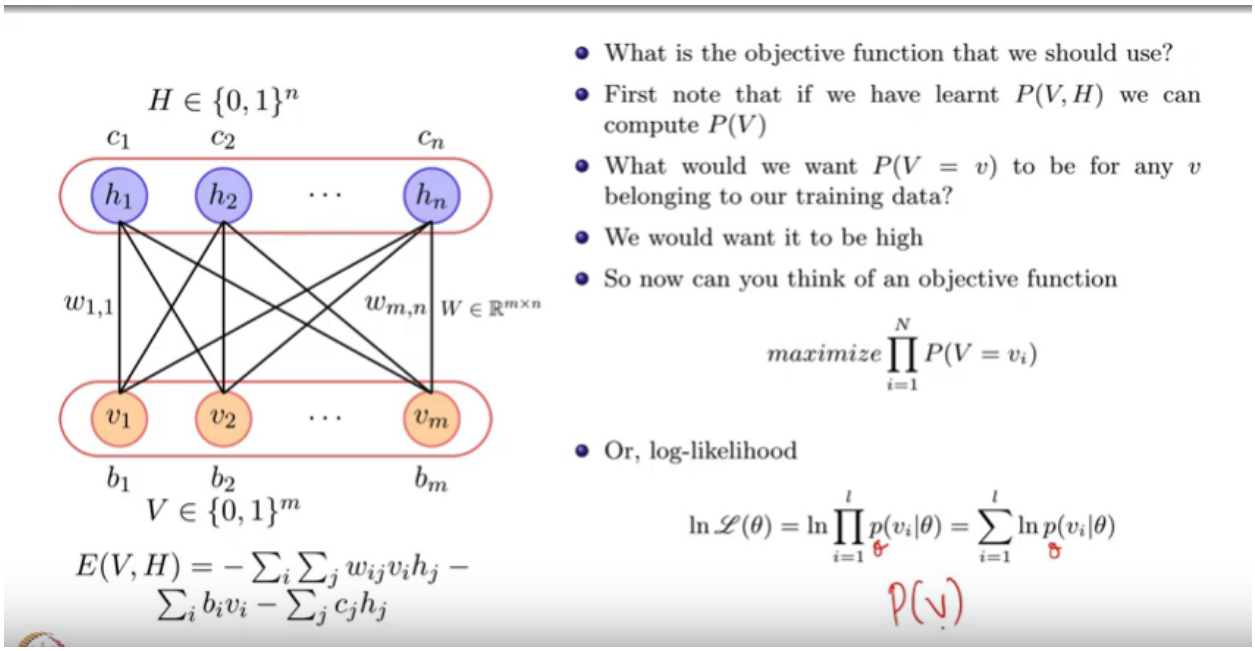


- But in the case of RBMs, our training data only contains x (for example, images)
- The x's are of course visible units and we will refer to them as V
- There is no explicit label (y) associated with the input.
- Of course, in addition to V we have the latent variable H but we don't know what these h's are
- We are interested in learning $P(V,H)$ which we have parameterized as

$$P(V,H) = \frac{1}{Z} e^{-(-\sum_i \sum_j w_{ij} v_i h_j - \sum_i b_i v_i - \sum_j c_j h_j)}$$

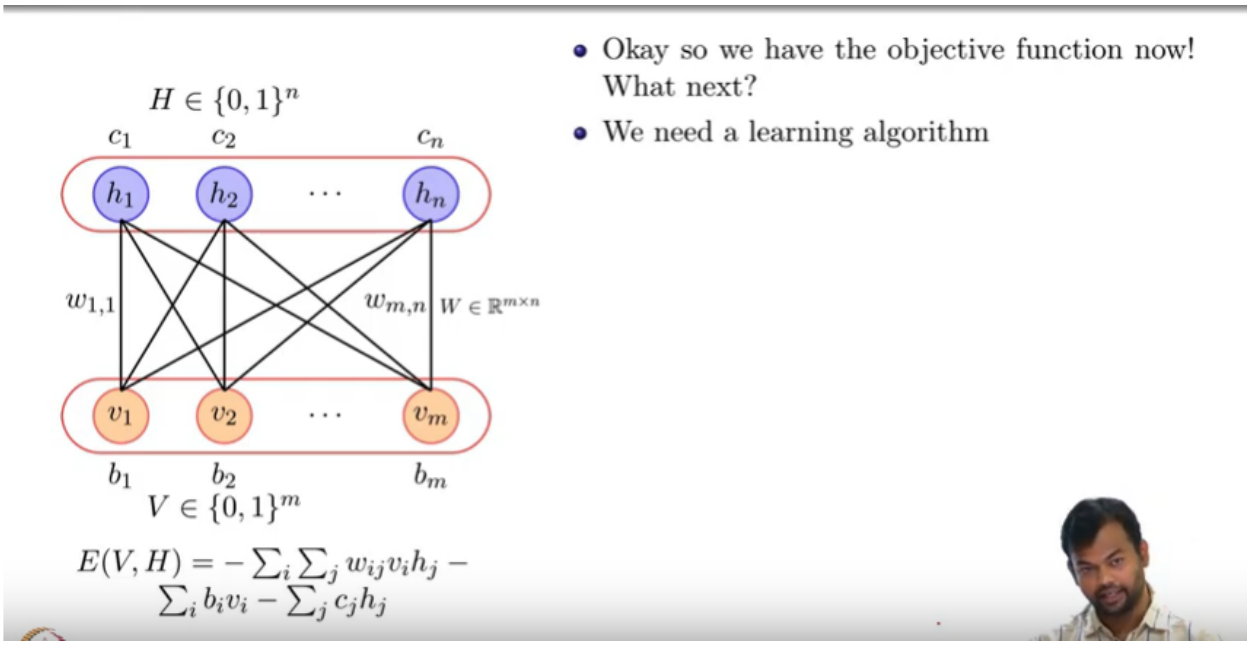
What can be maximized, it contains only of X and just for sake of consistency of notation, these X's are nothing but the visible unit, so irrespective of whether it's images or whatever, I'm just going to call them as,' V'. So, instead of saying that, X is equal to X 1 to X n, I'm going to say that, it's V right, V 1 to Vn, I hope V M okay, p.m. Okay? So, instead of X's I'm going just going to denote them by V. So, you don't do I'll give you a hint, you want to maximize some probability, you're given some training data, what will you maximize? Once the model has learnt everything well, what would you actually expect the model to do? At least for the examples in your training data, it should assign to what? P of V comma H, P of H P of V, B comma H do you know what edges are know, T of H do you know what edges are no, so P off that's the only option left, but does that make sense, for everything that you have in your training data, once you ask the model okay, what's the probability of this image. I trained you on these set of images for god sake please tell me the probability of these images. Right? And what would you expect the answer to be, I don't know. If I compute if I take one image from my training data, can make it a V right, I will say that V is equal to pixel 1, V 2 is equal to pixel 2 and so on. And I feed to the network and I asked you to compute the probability of this particular configuration, what would you want it to do? This probably should be high, that's what your wish list, but then just make this your objective function. So what should your objective function would be, for every image in my training example, maximize P of V. And that's the same as maximizing the, the dash of the data, likely unit of the data, I've heard of this term before, I see what I've done this in any kind of unsupervised learning. Right? If you do a.m. or any of these other, aren't supposing you always maximize the likelihood, log likelihood of the data. How many of if you've seen this before, maximizing. Okay? Okay. So, this is what P of V comma H is

Refer slide time: (3:12)



if you have learned P of V comma H, we can compute P of V from it. Right? We'd want to know, what P of V is equal to V, for any V belonging to our training data. And we would want it to be high, so that's what our objective function is going to be, for all the capital n training examples that you had, maximize the probability of that particular training example. Right? So of all the possible configurations that the random variable V, which is a vector, can take i want the probability mass to be concentrated, on those configurations which I have seen in my training data, does that statement make sense. Right? And does this look like a legit objective function, fine. Okay? So, now that we have agreed on that, so our L theta and this is a slight abuse of notation, people write it differently. So this is basically, I think it should ideally be written as P of theta Vi, but all these alternate notations are acceptable, it just means that, the probability of a particular random variable taking on a particular value, under the parameters that I have learned. Right? So, theta all the parameters that I have. Okay? So this is just for, just consider this as P of V, ignore the given theta part. Given theta just means that, given the parameters that I have learned for the model. Okay is that fine. Okay?

Refer slide time: (4:34)



Now, we have the objective function, what do we do next? Once we have the objective function, once we have the parameter, what do we need? What is the learning algorithm? How will you do back propagation? Gradient descent, what's the keyword there? Gradient .So what will we need? Gradient off what with respect to what, loss function with respect to all our parameters. What are the parameters? WI J's beta B's and C's. Okay? And as usual we'll ignore the B's and C's and only look at the W edges. Okay? So that's where we are headed, it's a slightly long route. But, we'll get there.