Lecture 19.5

Training RBMS Using

Contrastive Divergence

Okay. So, Okay, training our beans with contrastive divergence. So why? So what is the problem with Gibbs sampling? They've been getting away by this large number of steps. Right? We'll run the Markov chain, for a large number of steps, in practice, that large number of steps is really large. Right? There was a guarantees hold, only asymptotically when n tends to infinity, I'll have to actually run it for a large number of steps.

Refer Slide Time :(0:37)

Algorithm 0: RBM Training with Block Gibbs Sampling

Input: RBM $(V_1, ..., V_m, H_1, ..., H_n)$, training batch D

Output: Learned Parameters $\mathbf{W}, \boldsymbol{b}, \boldsymbol{c}$

init $\mathbf{W}, \boldsymbol{b}, \boldsymbol{c}$

forall $\boldsymbol{v} \in D$ **do**

  Randomly initialize $\boldsymbol{v}^{(0)}$

  for $t = 0, ..., k, k+1, ..., k+r$ **do**

    for $i = 1, ..., n$ **do**

      sample $h_i^{(t)} \sim p(h_i | \boldsymbol{v}^{(t)})$

    end

    for $j = 1, ..., m$ **do**

      sample $v_j^{(t+1)} \sim p(v_j | \boldsymbol{h}^{(t)})$

    end

  end

  $\mathbf{W} \leftarrow \mathbf{W} + \eta[\sigma(\mathbf{W}\boldsymbol{v}_d + \boldsymbol{c})\boldsymbol{v}_d^T - \frac{1}{r}\sum_{t=k+1}^{k+r} \sigma(\mathbf{W}\boldsymbol{v}^{(t)} + \boldsymbol{c})\boldsymbol{v}^{(t)T}]$

  $\boldsymbol{b} \leftarrow \boldsymbol{b} + \eta[\boldsymbol{v}_d - \frac{1}{r}\sum_{t=k+1}^{k+r} \boldsymbol{v}^{(t)}]$

  $\boldsymbol{c} \leftarrow \boldsymbol{c} + \eta \nabla_{\boldsymbol{c}} \mathscr{L}(\theta)$

And remember that, this loop, is inside your outer loop that means, for every training instance you'll have to run the Markov chain, for that many time steps and that's obviously expensive. Right? So, even though this could, give you a tractable way, it's definitely more tractable than computing that original expectation, but, still expensive. Right? Each individual step of this process is easy, but collectively still you have a large number of steps. Okay?

Refer Slide Time :(1:01)

- In practice, Gibbs Sampling can be very inefficient because for every step of stochastic gradient descent we need to run the Markov chain for many many steps and then compute the expectation using the samples drawn from this chain
- We will now see a more efficient algorithm called k-contrastive divergence which is used in practice for training RBMs

So, it can be very, in, inefficient because at every time step, you need to run the Markov chain for many, many time, every titration, you need to run the Markov chain, for many, many time steps. So in practice, we will use something known as, 'K Contrastive Divergence'. So let's see this, see how the story is progressing. Right? So, we had this expectation, in reality we should have done the summation over infinity. So you have, approximated the Infinity by, some R, in Gibbs sampling and now we are going to approximate it by something, even more ridiculous than R. Okay? So, let's see where we get there.

Refer Slide Time :(1:36)

$$\mathbb{E}_{p(H|V)}[v_j h_i] = \sigma(\boldsymbol{w}_i \boldsymbol{v} + c_i) v_j$$

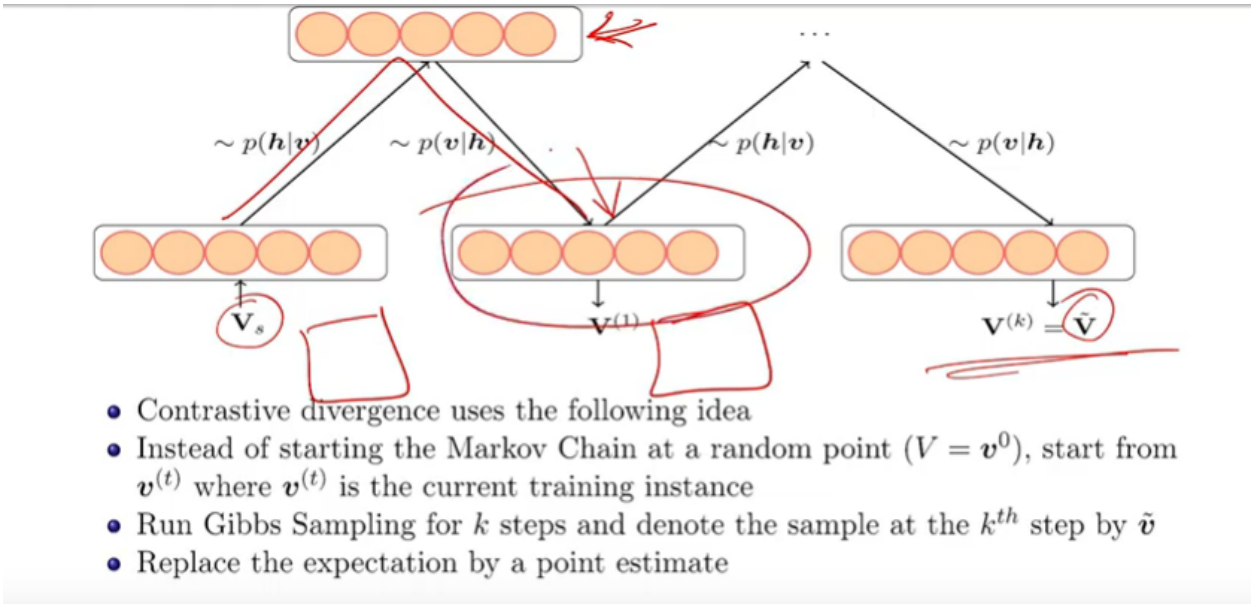
$$\mathbb{E}_{p(V,H)}[v_j h_i] = \sum_{\boldsymbol{v}} p(\boldsymbol{v}) \sigma(\boldsymbol{w}_i \boldsymbol{v} + c_i) v_j$$

- Just to reiterate, our goal is to compute the two expectations efficiently
- We already have a simplified formula for the first expectation
- Furthermore, note that the first expectation depends only on the seen training example ($\boldsymbol{v}$)
- The second expectation depends on the samples drawn from the Markov chain $(v_1, v_2, ..., v_n)$
- The first expectation thus depends on the empirical samples, whereas the second expectation depends on the model samples (because the samples are generated based on $P(V|H)$ and $P(H|V)$ output by the model)

So, just to reiterate, our goal is to compute these two expectations. We already have a simplified formula for these two expectations. The first expectation we don't need to worry about, because that does not have any summation. And also, the first expectation only depends on the, seen variables. Right? The training data, it only depends on the training examples. The second expectation, depends on the which samples, the samples drawn from the Gibbs chain, not from the training data. Okay? So, the first expectation is about the training data, the second expectation, is about the sample data or the model data. Right? It's also called as, 'The Model Samples'. Because, these samples are drawn, based on whatever your model tells

you, about these probabilities, based on the current parameter configuration of the Para model, you compute these probabilities and then you sample from there. Right? So, this summation is over, model samples and this summation, which is non-existent, is above, is over, empirical sample right, the data samples. Okay?

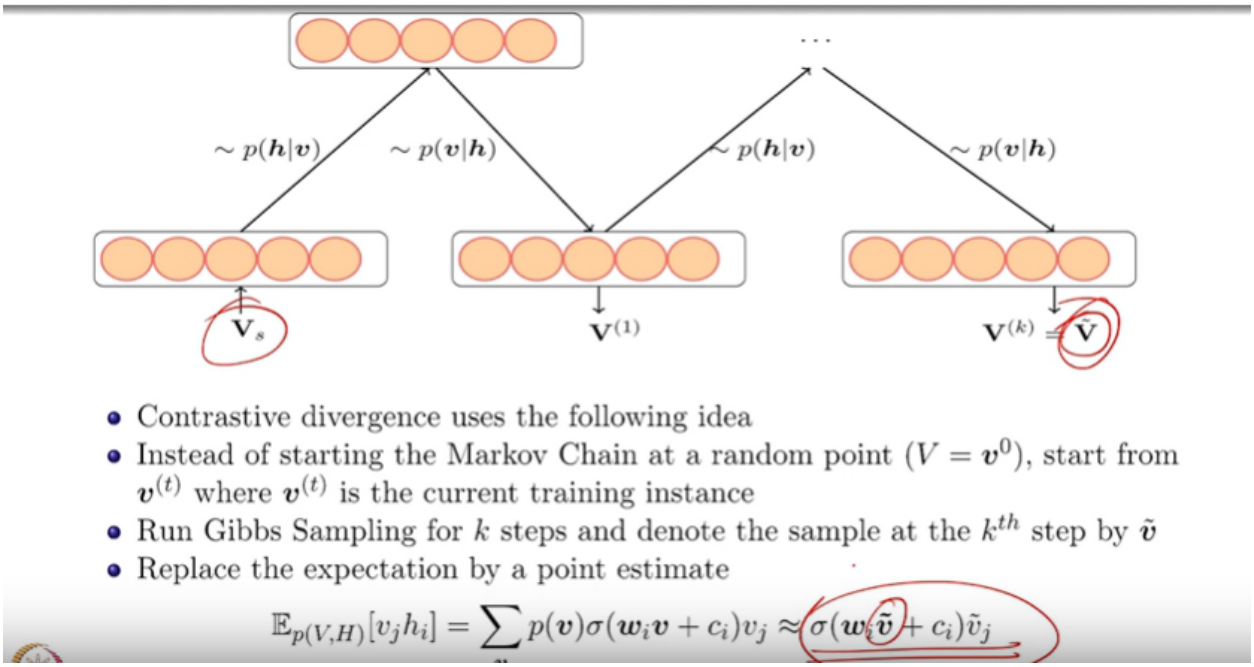
Refer Slide Time :(2:36)



Now, Contrastive Divergence uses the following idea. Instead of starting the Markov chain, at a random point, V is equal to V 0. Okay? No one questioned me, when I started with V is equal to V 0. What was the problem with that? I said, I just pick a uniformly, any of the possible values. What was the problem with that? Think in terms of images again. I could have started with some very random noisy image, which does not actually is an image? Which is not actually an image? Right? It could just have been those noisy pixels. So, if I start from there, to get, get to an actual, where do I need to go eventually? To something which looks like an image. Right? That's the samples that I'm interested, so starting from something very random, reaching there, is going to be a very difficult job. I'll have to run it for many, many time steps, to come out of this noisy example and start producing things, which actually look like images. Everyone gets that. Right? But, that's the only option we had, we didn't know, what the initial distribution is, so, we just took that. So what contrastive divergence here is that, instead of starting from this random configuration, start your chain, from your current training sample. You are going over the training data, VT is your Th training sample, just start the Markov chain from there. Because you know that you are starting from something, which is actually a sample from your true distribution. Okay? So, that itself, simplifies something at it, it simplifies you reaching to that distribution Okay? Second is, now you run Gibbs sampling for K steps and you denote the sample at the K step by **V** tilde. Okay? So, this is

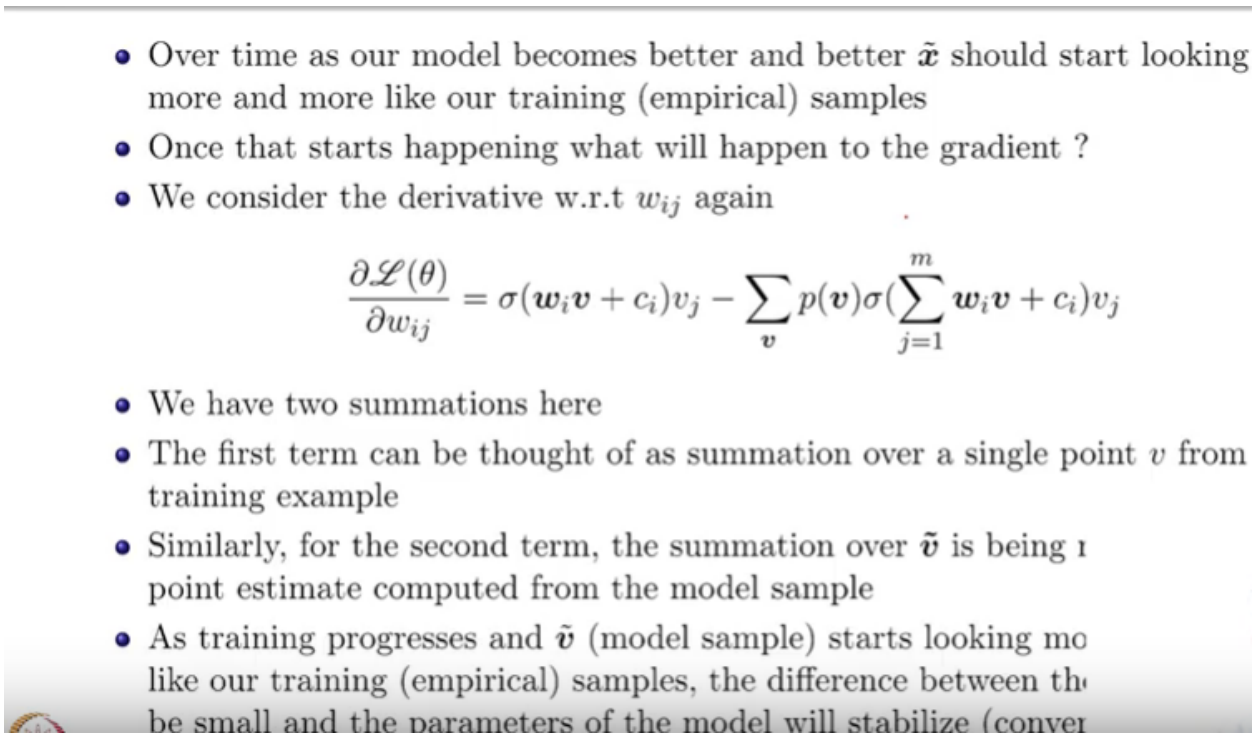
what I am doing? I am starting with, something which was observed in my training data, so instead of having a random V naught, I am trying from a sample which came from my training data, given this visible units, I can sample the hidden units, given the hidden units, I can sample the visible units and so on. Right? So, this is exactly the block process that I was talking about, given the hidden units I can sample all the visible units, given the visible units I can sample all the hidden units and run this for K steps. Okay? And whatever I get after K steps, I'll call it as B tilde. Okay? So, what has happened here is? Let's try to understand this in more detail. Initially when your model is not really trained, what will happen? You started with VS, which was a good training instance; you will compute some hidden representation for that, this is actually not meaningful why? Do other parameters are not trained, using this not so meaningful in representation, well again trying to can reconstruct orV1.what will this have happen, happen now? You'll get a very bad reconstruction, effectively in this step, what have you done? You have computed a hidden representation and then try to reconstruct from there, so ideally if I had given at a blue sky, it should have given me back a blue sky. But, initially this is not going to happen. Because, your model is not trained well, you will actually get very bad samples from this chain. Okay? Eventually as your model starts learning better and better, what is going to happen? Starting from a training instance, you'll start getting samples, which look very much like the training sample. Because, this hidden representation would be more meaningful. Hence, whatever you sample from that hidden representation would be more meaningful? And so on. Right? So, 1` that's what contrastive divergence relies on?

Refer Slide Time :(6:02)



And finally they do this ridiculous thing, where you approximate, the expectation by a point estimate. So, we had approximated infinity by R, in the in Gibbs sampling. Now, we are going to approximate R by a single something. So, this entire summation which was over all possible values of V, we are going to just, replace it by a point estimate, we're just going to estimate it from that, single, sample that we have drawn after K time steps. Okay? So, this is known as, K contrastive divergence, the term contrastive, because this is in some sense a true, example and this is in some sense a negative example. Right? Because initially our model is not trained. So, drawing these negative sample and if you look at the actual computation you have this, expectation is computed using, a training sample and this expectation, which is computed using a generated sample or a sampled, sample and you're taking the difference between these two it. So, it's some kind of a contrast of thing that you're doing. Right? Okay? So, you get that. So, you have replaced the second summation. So, I've replaced the second summation by a point estimate. So, there are three key ideas here, one is instead of starting from a random point, start from a true point, a true image, instead of running the Gibbs chain, for many, many, many times tips, just run it for K time steps, where K is going to be a small value. Instead of approximating the expectation by a summation over a large number of samples, just approximated by a single point estimate. Right? So, these are the three ideas, clearly takes care of the computational programs. Right? And the main thing here is that you are starting from a point in the chain, which is already reliable, instead of starting from a random point. Okay? Okay?

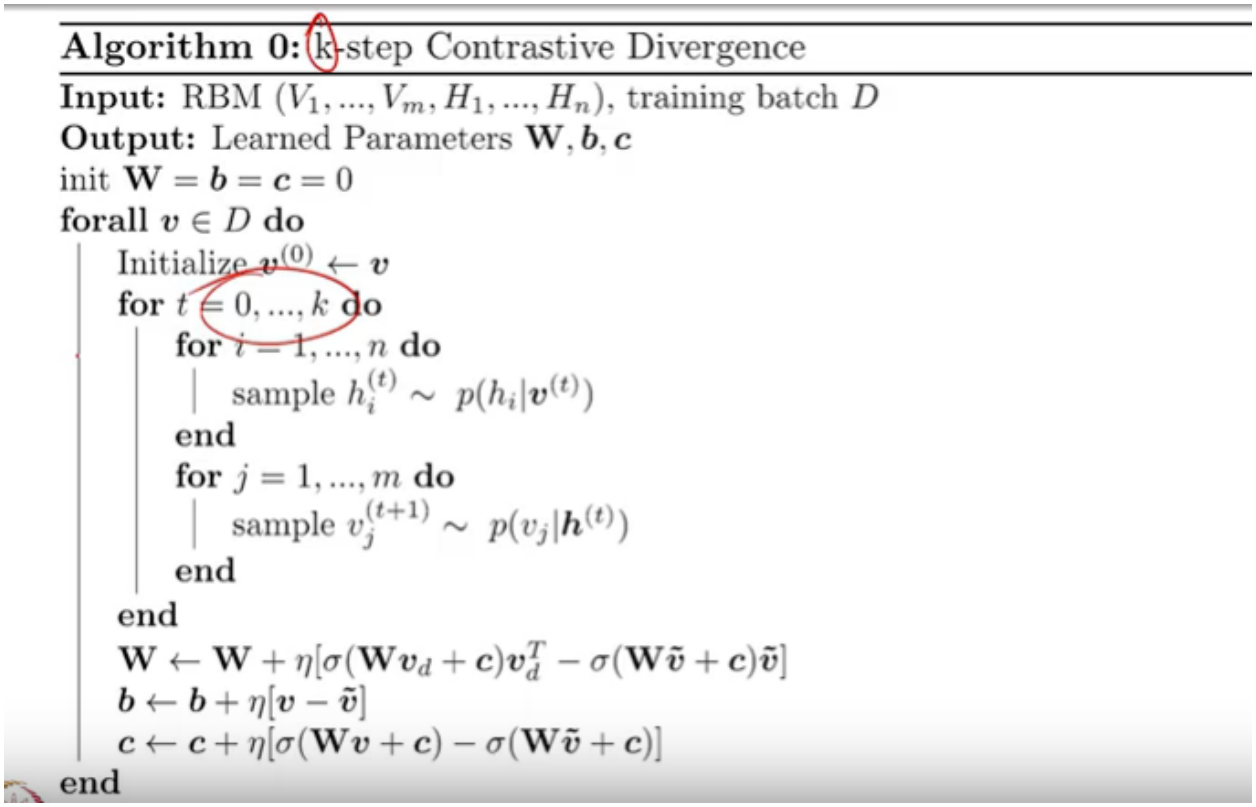
Refer Slide Time :(7: 54)



So, as the model becomes better your V tilde, naught X tilde, should start looking more and more like your training samples and once that starts happening, what will happen to this gradient? This was the

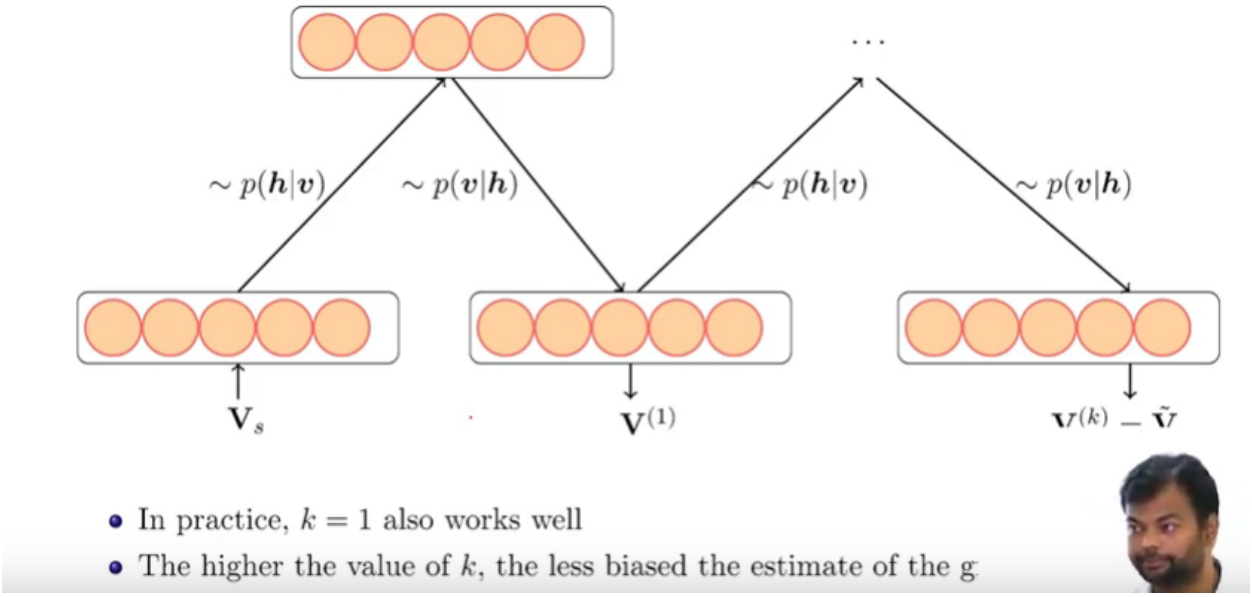
gradient, once your model gets trained and you're going to replace, the second summation by a point estimate. So, now this is, a difference of one sigmoid and another sigmoid, over time as your model has started becoming better and better, what is going to happen? These two terms are going to, cancel out each other. Right? Because given the image, you computed a hidden representation and you should have gotten the same image back, if your model is really learned well. So, as you reach converging, your gradients will become smaller and smaller and your parameters will stabilize, does that make sense? Okay? At this point you'll say yes to anything, but. Okay? So, that's basically, what I have written here?

Refer Slide Time :(8: 54)



And now, just to give you the full algorithm, it's the same as a grip sampling more or less, input, output, initialization, instead of randomly initializing it. I'm initializing it to the training that I had, again I am going to do this block Gibbs sampling, block Gibbs sampling and now, this gradient; earlier I had approximated the expectation by a sum. Now, I'm going to approximated by, a point estimate. Right? So, VD is the empirical sample or the data sample and V tilde is the model sample. Right? So, it's the difference between, something computed using the data sample and something computed using the model sample, because the summation has, disappeared and the same story for B and also C. Okay? So, this is K step contrastive divergence, because you are running this Gibbs chain here, for K steps, in practice something even more ridiculous is done.

Refer Slide Time :(9: 53)



You actually just set the value of K to 1. Okay? And still this works very, well. So, actually, I am wrong, I'm not wrong, I am just partially. Right? In the assignment, you are going to do Gibbs sampling, as well as K contrastive divergence and you will see the contrast between both of them. Right? What happens? When you run one yeah! Right? So, in practice instead of running this for K, time steps and taking a V tilde, you actually, run it only for onetime step and just take v1. Okay? Right, so, that's all I had. So, with that we end, RBMs, finally after a very, very long story and the next thing that we are going to do now is, very stern auto encoders, followed by Ganz and I guess probably that's where we will end.