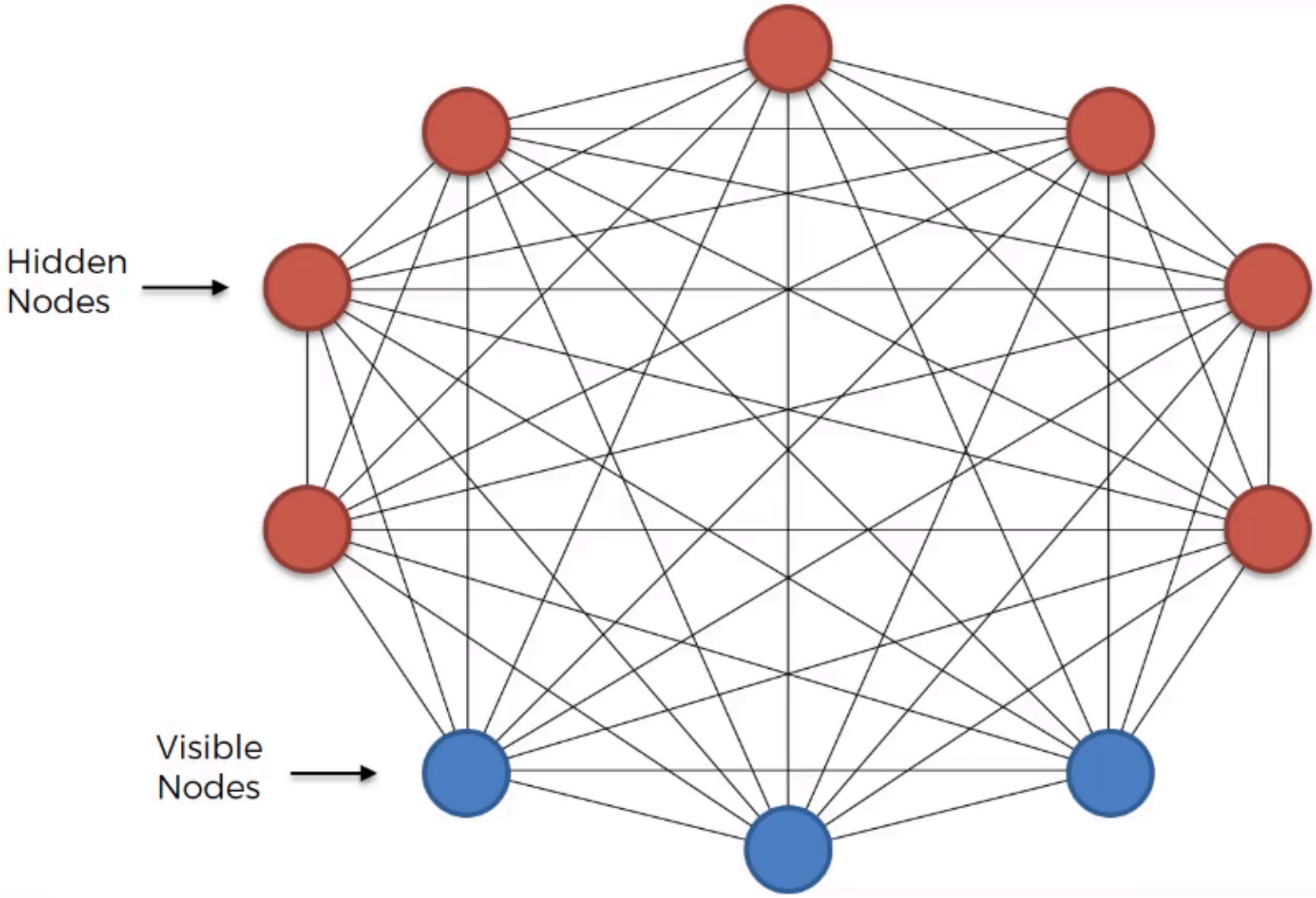
BOLTZMANN MACHINE NOTES

Soham Patel

2024-08-03

Boltzmann machines are Unsupervised deep learning model whose architecture looks like



Unlike normal neural networks we have interlevel connections as well in this case. Like blue dots or the visible nodes (think of them as input nodes in case of ANN) are even internally connected unlike neural networks where the inputs don't influence other inputs, same is the case of hidden layer or the red dots.

The extra connections help us to find all the possible relations and pattern. Stating the above things this model is a stochastic model where same inputs could lead to different outputs. Output in this case is the state (combined set of allocated value of every node in the model)

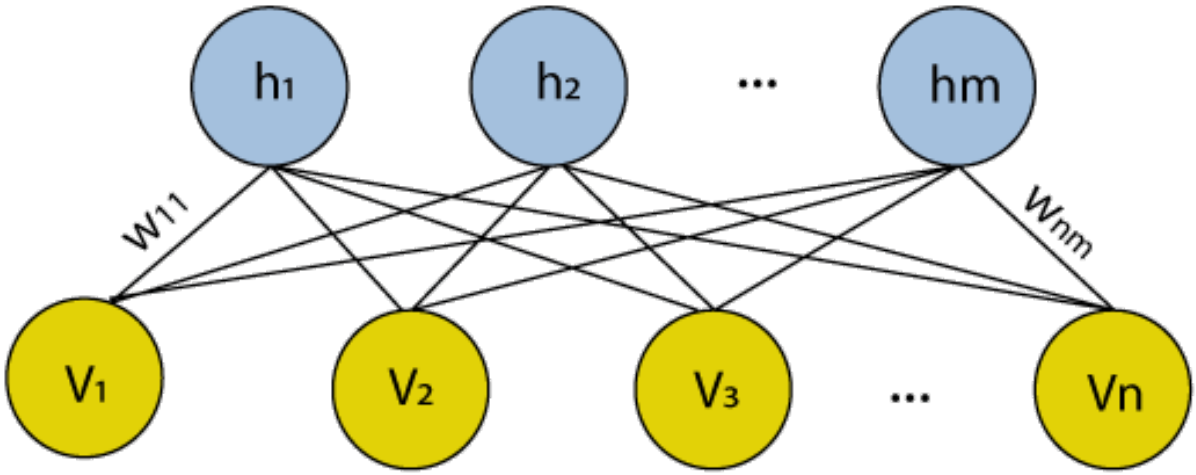
The visible and the hidden nodes are treated as same in case for the model. It's just that blue nodes are in our control where we are inputting values. I would explain this in easy terms. The input or the visible layers could be further changed by the hidden layer enabling the reverse connections unlike normal neural networks for example you calculate all the values of the hidden layer based on the visible layer, now again the hidden layer contribute to recalculate the visible layer to find other state of possible values.

This is a Energy Based model, Energy based models are the probabilistic models that calculate the energies of the state (think of state as the values of all the nodes at that time period). It tries to minimise the energy cause energy of the state is inversely proportional to the probability of its existence. Like more the energy the state is unlikely to actually exist.

NOW, as in boltzmann machines every node is connected to every other nodes it becomes pretty hard for us to actually work with such an expensive model and practicality of this model becomes low as we keep on increasing the nodes.

SO TO OVERCOME THIS PROBLEM PEOPLE INTRODUCED RESTRICTED BOLTZMANN MACHINE.

Think of its ‘visible’ architecture as similar to that of ANN ***BUT*** the layers are not unidirectional like even the hidden layers can change the visible layers to find different states. Below is the architecture diagram.



Here the interconnection within layers is not present unlike normal Boltzmann machines. This is the most used Boltzmann model.

LETS THINK AT BOLTZMANN MACHINE AS AN EXAMPLE.

Think of normal Boltzmann machines as a thermal power plant. Every node of the model signifies the conditions of the plant for example thickness of the thermal chamber wall, the speed of turbine, quality of soil around the plant or maybe the temperature of the plant and so on. So lets view some of the conditions as input conditions that we have the knowledge of, for example we input the speed of turbine and temperature and based on it the model calculates all other values of different conditions that again collectively change the input conditions and by doing this (by the way this continuous changing cause of other parameters is the case of Gibbs Sampling) repeatedly we can find nearly all the sates of the model that are probable (like the model calculates the energy of the current state and the less energy states are probable).

How is it helpful? Like given any state that the model could not find probable maybe an exception and the model may flag it as inappropriate and may lead to a blast or cause some disastrous outcome.

————————————————————————————

TAKEAWAY (As far as I understood): The Boltzmann Machine finds all the probable states the model can find so that, we could learn nearly every possible outcome state.

Now these models can be also used for recommendation systems for example the input layer is given the values 1 and 0 based on the movies that a person likes for example the first input node that is for titanic has 1 means given user likes titanic. Based on this the hidden layers may have features like whether a person like Leonardo Di Caprio and if it finds that a person sent 1 for both the titanic and shutter island nodes that sets the hidden node for Leo to 1 same way other hidden nodes could be if the person likes Oscar winning movies or if the person likes horror. Based on such node activation the hidden layers can again go back to the visible nodes and maybe change the input . In our case we may find that it fills up some of the incomplete nodes in the input layer based on the hidden layer like the user we talked about may have not seen Inception but his liking for Leo that activated that particular node in the hidden layer would fill the incomplete value of Inception as 1 so suggesting him to watch Inception.

Now lets talk about two more types of Boltzmann machines:

Deep Belief Networks (DBNs):

Structure: Composed of multiple layers of Restricted Boltzmann Machines (RBMs) stacked on top of each other. The top layer is usually a softmax classifier for classification tasks.

Training: DBNs are trained in a layer-by-layer fashion using unsupervised learning (typically contrastive divergence), followed by fine-tuning with supervised learning if needed.

Purpose: Used for unsupervised feature learning, dimensionality reduction, and pre-training deep neural networks.

Pros: Relatively straightforward to train using pre-training and fine-tuning, effective for feature extraction.

Deep Boltzmann Machines (DBMs):

Structure: Generalizes the DBN by allowing connections between all layers, not just adjacent ones. Consists of multiple layers of Boltzmann Machines.

Training: Training DBMs is more complex due to the full inter-layer connectivity, often requiring sophisticated algorithms like mean-field approximation or variational methods.

Purpose: Designed to model complex distributions by capturing dependencies between all layers, providing a more flexible generative model.

Pros: Capable of capturing more intricate dependencies between variables, potentially leading to more powerful generative models.

In summary, DBNs are often easier to train and use for pre-training deep networks, while DBMs offer more flexibility in modeling complex distributions but are more challenging to train. Now you might think about how does the Boltzmann machines set weights to find the most probable states. for that first learn about.

Loss Calculation

The loss function is derived from the likelihood of the observed data. The goal is to maximize the likelihood, which corresponds to minimizing the negative log-likelihood. For a BM, this can be complex due to the need to sum over all possible states. Instead, a more practical approach is used, such as Contrastive Divergence (CD) for RBMs.

Contrastive Divergence (CD)

CD is an approximation method to update the weights:

1. **Initialize Visible Units**: Start with a training example (visible units).
2. **Positive Phase**: Compute the hidden unit activations given the visible units.

$$P(h_j = 1|\mathbf{v}) = \sigma\left(\sum_i v_i w_{ij} + b_j\right)$$

   where σ is the sigmoid function.

3. **Negative Phase**: Reconstruct the visible units from the hidden units and then recompute the hidden unit activations.

$$P(v_i = 1|\mathbf{h}) = \sigma\left(\sum_j h_j w_{ij} + a_i\right)$$

4. **Compute Gradients**: Calculate the difference between the product of the states of the visible and hidden units in the positive phase and the negative phase.

$$\Delta w_{ij} = \eta\left(\langle v_i h_j\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{reconstruction}}\right)$$

   Also think it as an energy function where the energy of that state is calculated hoping it to be minimum and it is done a certain times to minimise the energy hence increasing the probability of the state..

5. **Update Weights**: Adjust the weights and biases based on the gradients.

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$$

Summary of Steps

1. **Initialize Weights**: Randomly initialize the weights and biases.
2. **Positive Phase**: Given the input data, compute the hidden unit activations.

3. **Negative Phase**: Reconstruct the visible units and recompute the hidden activations.
4. **Compute Gradients**: Calculate the difference in the associations (product of states) between the positive and negative phases.
5. **Update Weights**: Adjust the weights and biases to reduce the energy of observed states and increase the energy of unobserved states.