Chapter 14 : Part 1

**Deep Learning**

**BM: Boltzmann Machines**

Introduction

**14.1.1 What we will learn in this Chapter**

1. The Boltzmann Machine: We'll talk about the Boltzmann machine, and how it's ***structured***, how it ***works***, its ***architecture***. Here we're talking about the actual Boltzmann machine, the big one, not just the ***RBM (Restricted Boltzmann machine)***.
2. Energy-Based Models (EBM): We're going to be talking about Energy-Based Models (EBM), that will help you understand the inspiration for ***Boltzmann machines***, where it comes from, We're going to understand what goes into their ***architecture***, what goes into their background that allows them to come up with the ***results*** that they are able to come up with.

* And how we actually control them in terms of weights and what that means in terms of energy.

1. Restricted Boltzmann Machine (RBM): Here we're going to be talking about the ***Restricted Boltzmann machine***, or the ***RBM***. This architecture was proposed to solve the problem of ***computational*** issues with the ***Boltzmann machine***.

* We'll also *walkthrough* an *example* to know how an RBM *trains* and how an RBM is applied on a *practical* example.
* It will help us on implementation the BM in Python.

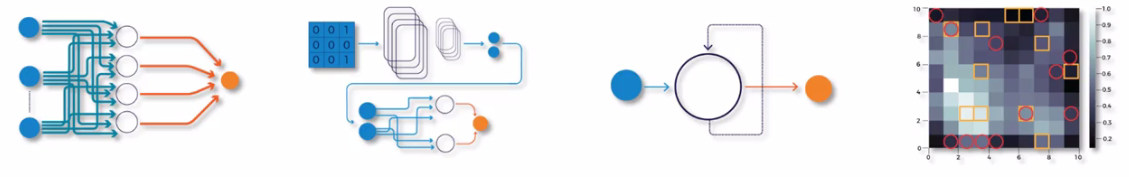
1. Contrastive Divergence (CD): Here we'll talk about *Contrastive Divergence*, the algorithm that allows us to find the *weights* for research in *Boltzmann* *machines*.
2. Deep Belief Networks (DBN): Here we'll talk about *Deep Belief Networks* or *DBNs* in very short.
3. Deep Boltzmann Machines (DBM): We'll discuss the *DBMs*, *Deep Boltzmann Machines* very shortly.

**14.1.2 The Boltzmann Machine (BM)**

*Boltzmann Machine (BM)* is very *advanced* topic. We've discussed ANN, CNN, RNN for supervised deep learning, also we've discussed SOM which is an unsupervised technique. Now BM is also unsupervised technique. So let's have a look at all of these in a diagrammatical representation.

|  |  |
| --- | --- |
| 1. So here we've got an artificial neural network (ANN). With the *input* layer, and *multiple* *hidden* layers and an *output* layer. |  |
| 1. And then we've got a convolutional neural network (CNN) with the *convolution* layer, the *pooling* layer, the *flattening* layer and then an *ANN* sitting on the end. |  |
| 1. Then we've got the recurrent neural network (RNN) where the *hidden* layer feeds back into itself, and therefore facilitates analysis of *temporal* *data*. |  |
| 1. Here we've got the self organizing map (SOM), which helps you *reduce* your *dimensionality* and represents something, it represents your data in a more understandable way. |  |

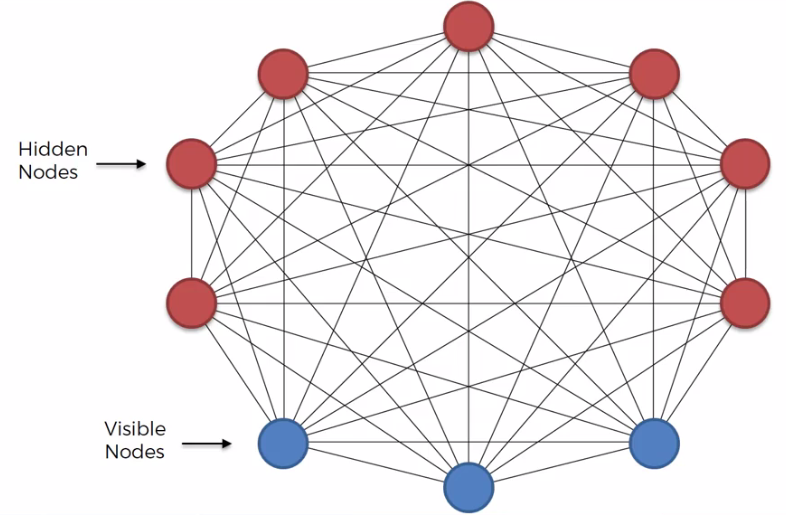
* Directed Models: Now, even though the SOM is a unsupervised type of deep learning model, but ANN, CNN, RNN & SOM all these four are actually directed models.
* There is a direction in which the model operates. For example in SOM, we've got the input-data and then the information goes through the nodes creates an output map.



***ANN, CNN, RNN & SOM all these four are Directed Models***

|  |  |
| --- | --- |
| * Boltzmann machines (Undirected Models): Boltzmann machines are *Unsupervised Deep Learning Model* and it is an Undirected Model. All kind of *Boltzmann* *machines* are *undirected* *models*. * In the diagram, notice that there are actually no arrows in the connections between nodes, all these connections go both ways. * Let's have a look a bit closer at a Boltzmann machine and understand what's going on. Following is a Boltzmann machine (BM), we've got hidden nodes in red and we've got visible nodes here in blue. (Actually there is no difference between hidden nodes and visible nodes. In BM all nodes connected to each other. For learning purpose we used two colors.). |  |

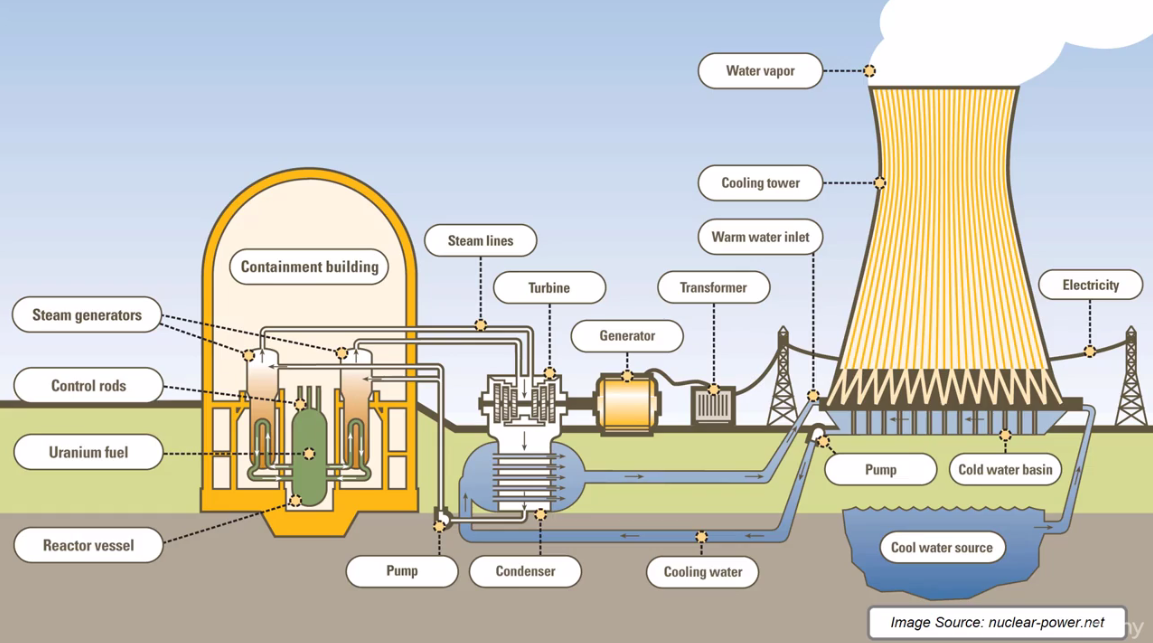
* Here, we've got *three* visible nodes (input nodes) at the bottom, and we've got seven hidden nodes.



* Three things to notice:

1. This model *doesn't* have an *output* *layer*. There's an ***input*** ***layer***, there's a ***hidden*** ***layer***, but *no output layer*.
2. All nodes connected to each other. *Everything is connected* (hyper connected) to everything, there's no specific layering per say.
3. There is *no direction* in these *connections* (Undirected). All connections in BM are bi-directional. The connection happens both ways.

* The Visible Nodes (input nodes) are connected each other: Notice that the visible nodes are all connected between each other,
* What's the point of connecting visible nodes? Once you input data, it's fixed, right?
* You have a *row* of *data* and you just input it to the *model* and there's no point in these *connections* between these *visible* *nodes*. You're not going to be *adjusting* the *weights* or *training*, because that data is *fixed*.
* And that’s the Twist!! Boltzmann machines are fundamentally different to all other algorithms.
* Boltzmann machines don't just expect input data they also generate data. They generate information in all of available nodes, regardless of whether there's and input node or a hidden node. For a Boltzmann machine all of these nodes are the same.
* For a BM this whole thing is a *system*, it's generating *states of this system*.
* Analogy of nuclear power plant: The best way to think about BM is through an example of a *nuclear power plant* that once given by ***Geoffrey Hinton***.



* Let's say that there are certain things that we can measure about the nuclear power plant, for instance, *temperature* inside the *containment* *unit*, how quickly *pump* *turbine* is *spinning*, the *pressure* inside the *turbine* *pump*, how much *electricity* it is *outputting* etc.
* But at the same time there could be a lot of odd things that we're not measuring. For example, the *speed* of the *wind*, the *moisture* of the *soil* (where pump operates), *thickness* of the *cooling* *tower* wall at a height 20 meters. So there can be lots of *different* *parameters* of the nuclear power plant that we're *not* *measuring*,
* *All* these *parameters* (those we are measuring and we aren't measuring) all together, *form one system* and they all *work* *together*, and that is what the *Boltzmann machine* represents.
* The BM is a *representation* of a *certain* *system* (in our case, a nuclear power plant). The *visible* *nodes* are just merely things that we can and do *measure*, and the *hidden* *nodes* are things that we *can't* or *don't measure*.
* The *Boltzmann machine (BM)* is capable of *generating* *values* instead of just waiting for us to *input* *values*.
* It just generates *different* *states* of our system. For example: in case of *nuclear power plant* BM looks at a state with a certain *temperature*, certain *speed* of the *wind*, and the *moisture* or the *pressure* in the *pump*.

That is how the Boltzmann Machine works, it's not a Deterministic Deep Learning model, it's a Stochastic Deep Learning model or a Generative Deep Learning model.

**14.1.3 How BM works**

In a simple word, BM describes different possible states of our system. It can generate possible scenarios from the given data.

* For example: If we give our *nuclear power plants* normal state data (how it operates in normal condition) to a BM, then the BM recognize the normal state for the power plant. And also BM can generate abnormal state, such as ***"Core-meltdown"*** or ***"Nuclear Explosion"*** and shows us the corresponding parameters for the scenario.
* This could be very helpful to avoid such *fatal* *failure* and *accidents*.
* We feed in *BM* our *training* *data* (thousands of rows that we have) as the inputs to help it *adjust* the *weights* of this system accordingly, so that it actually *resembles* our *system*.
* For example: *BM* doesn't *resemble* any *nuclear* *power* *plant* in the *real-world*, it actually learns from what we feed it in. It learns how, what are the *possible* *connections* between all of these *parameters*.

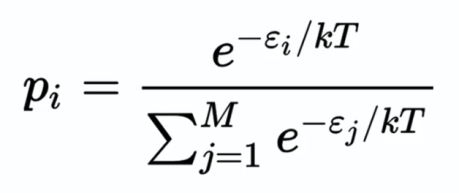
|  |  |
| --- | --- |
| * *BM* becomes a *system*, a machine that represents our *system* (our nuclear power plant). * Once *BM* done the *learning*, once all of these *weights* are *adjusted*, BM *understands* how all *parameters* *interact* with *each other* and what kind of constraints should exist between them in order for this system to be the system that we're modeling. * Once that's all done we can use the BM to *monitor* our *nuclear* *power* *plant*. * Because we've modeled it using *good* *behavior*, that hasn't led to any *meltdown* or any *explosions*. BM knows what is *normal* for the *nuclear power plant*. * Then the BM will help us understand what is abnormal behavior. |  |

* In real-life we cannot really model a nuclear meltdown through supervised learning, we don't really have that luxury of having all of this training data with actual nuclear meltdowns.
* You have to *model* it in an *unsupervised* *manor*, and that's exactly what a *Boltzmann* *machine* does.
* *Learning* through good *examples*, it understands how the system works in it's *normal state*, and through that it helps us model what the system would look like in an *abnormal state* and model those *scenarios* and recognize those *scenarios*, and therefore ***monitor the nuclear power plant***.
* That now explains the design of these Boltzmann machines:
* Why we don't have an *output* *layer* because we're not *outputting* any value. We are creating a *model* that *describes* our *system*.
* Why all of these nodes are interconnected. Because all of the parameters are interconnected with each other. For example: the *speed of turbine* and the *moisture in the air* on that day are connected.
* The *directionality*. The *visible nodes* and *hidden nodes* all of the nodes *are* *equal* in a *BM*. All of these parameters are interconnected and therefore it had to be undirected. All the connections are going both ways.

**14.1.4 Energy based Models (EBM)**

Energy-Based Models helps us to understand the stochastic processes that describe systems in BM.

* Boltzmann distribution: Notice the following equation. This is the Boltzmann distribution. That's why the method is called Boltzmann machines (BM). This *equation* comes from *physics*, and it is used in the *sampling distribution* for the *Boltzmann* *machine*.



* Using this Boltzmann distribution the *Boltzmann* *machine* constantly creates *different* *states* of our *system*.
* From the equation we see that *Boltzmann* *distribution* talks about *probability*. It describes the probability of a certain state of a system. Assuming there is total states for the system.

|  |
| --- |
| = is a state. Represents different states.  = energy of the state **.**  = probability of your system being in state .  = is constant and  = is the temperature of your system. |

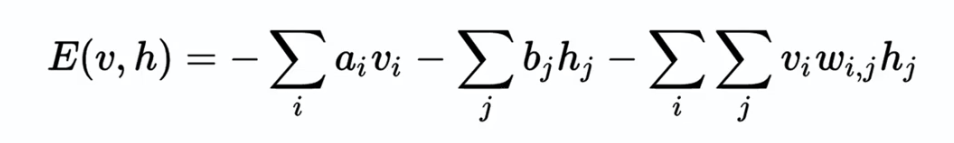
* The important thing is means the *higher* the *energy* of a certain state at *fixed* *temperature*, the *lower* the *probability*. So, probability is inversely related with energy.
* In a thermodynamical system, we're looking at the system at a *given* *temperature*, for instance a *gas*. The equation says that, the *probability* of your system being in a *certain* *state* is *inversely* *related* to the *energy* of your system in that state.

|  |  |
| --- | --- |
|  |  |

* To explain Boltzmann distribution consider we've got a room, and we have a gas, for instance air.
* Now the question is how the gas is distributed across the whole room?
* Why is the gas, not all in one corner?
* The answer is: They could be anywhere. Statistically, they could end up in one corner, in any room, at any given point in time. This is one of the possible states of this system.
* But the Boltzmann distribution is saying that the *probability* of that state "all gas in one corner" occurring is *very* *low*.
* Because the *energy* in that state (all gas in one corner), would be *very* *high*.
* Since the *molecules* are very *close* to each other, making a lot of *chaos* and they would be moving very *quickly*. Given the *constant* *temperature* of the room.
* What we normally observe is gas (air molecules are equally spaced apart) are distributed all over the room. Because it is the lowest energy state for that system.
* It applies to everything. For instance, if you *dropped* some *ink* into water, it will start *spreading* *evenly* in all directions. It won't form a *star*, because that's *not* the *lowest* *energy* *state*.
* On the other hand, if you drop, if you put a drop of oil into water, then it won't start spreading, because that is the lowest energy state for that specific system.
* *Boltzmann* *machines* are constructed with a *similar* *concept*. The *energy* is defined in BMs through the *weights* of *synapses*.
* Once the system is *trained* *up*, once the *weights* are *set*, the *system*, based on those *weights*, will always try to find the *lowest* *energy* *state* for itself.
* It has lots of different options, but the weights will dictate what is the lowest energy state for the system, and it will constantly try to get to the lowest energy state possible.
* That's why they're called *Energy-Based Models*.

**14.1.5 Energy function for a restricted Boltzmann machine (RBM)**

Here's an example of an energy function for a ***Restricted Boltzmann Machine (RBM)***. A bit *different* to the case of the *Boltzmann* *distribution* *energy*, because it's *not a gas*. You can see that it is defined through the weight.

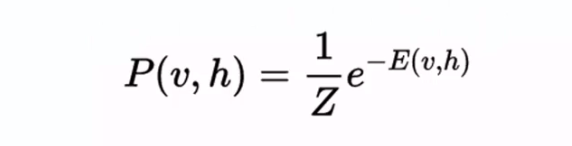


, are *biases* in the system, just *constants*.

, are, the *visible* *nodes* and *hidden* *nodes* respectively.

are the weights between the visible and the hidden node.

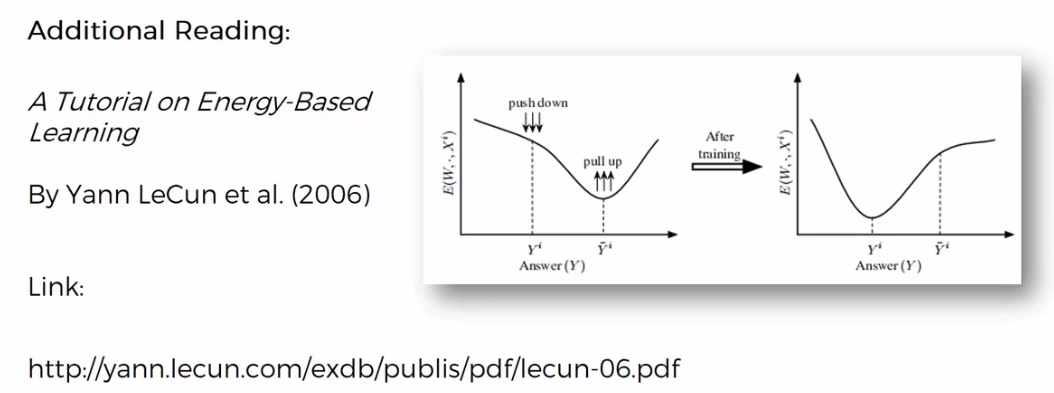
* And then the *probability* of being in a *certain* *state* is given by, through the *energy*, just like in the *Boltzmann* *distribution*. It is following equation:



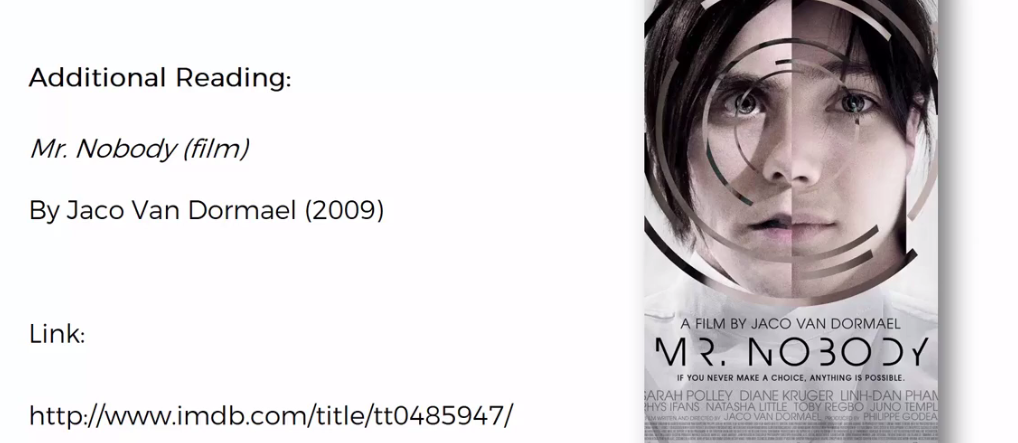
* Where , is the *sum* of all of the *values*, for all of the *possible* *states*, just like in the *Boltzmann* *distribution*.
* The *probability* of being in a *certain* *state* is *inversely* related with the *energy* of that *state*.
* And the *system* going to play by those *rules*, it's going to find the *lowest* *energy* *state*, just because of the way we *set* it *up*.

The functioning of a BM is very different than what we had in just NNs.

* Additional reading: A great paper actually, by Yann LeCun, from 2006. It's called *"A Tutorial on Energy-Based Learning"*. So if you really want to dig into Energy-Based Learning and *Energy-Based Models* then this is probably the best place to start.



* Watch: If you don't have time to read an article. Watch "MR. Nobody" by *Jaco Van Dormael*. A film starring, by Jared Leto.



* Why does *time* travel *forwards* and not *backwards*?
* Why if you mix *two liquids* together, you can't then *un-mix them*?
* Why *smoke* goes *out* of a *cigarette* not *back* *into* a *cigarette*?
* Why you can *remember* the *past* and you *can't remember* the *future*?
* These kind of very philosophical questions, we're getting very close to that when we're talking about Energy-Based Models and things like Entropy.