Alok Jha Follow
Cognitive Computing Consultant
Apr 20 · 5 min read

Boltzmann Machines | Transformation of Unsupervised Deep Learning — Part 2

We shall commence this article from where we left in Boltzmann Machines | Transformation of Unsupervised Deep Learning — Part-1 to cover the rest in this section of Unsupervised Deep Learning. If you haven't yet gone through Part-1, I would highly recommend to finish that off quickly before you run through this article. So the roadmap from here on includes:

Conditional RBMs: Several generalizations and extensions of *RBMs* exist and a notable example is *Conditional RBM*. In these models, some of the parameters in the RBM energy are replaced by parametrized functions of some *conditioning random variables*. **CRBM** (as it is often referred to) is a *non-linear generative* model for time series data that uses an undirected model with binary latent variables, **h**, connected to a collection of visible variables, **v**. The Visible variables can use any distribution in the exponential family (*Welling* et al., 2005), but for mo-cap data, we use real-valued Gaussian units (*Freund & Haussler*, 1992). Such an architecture makes on-line inference efficient and allows us to train by minimizing contrastive divergence. Taylor applied the CRBM to synthesize novel motion and perform on-line filling in of data lost during motion capture.

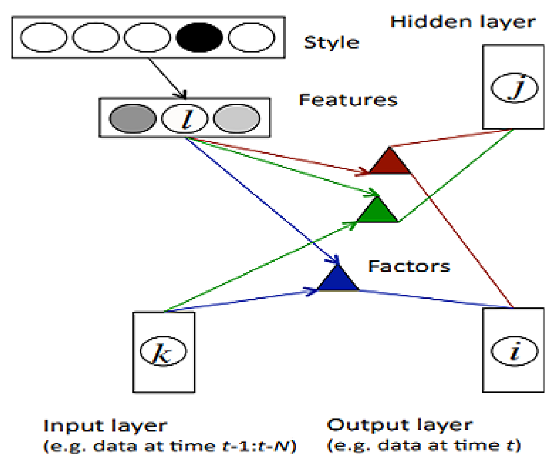

Figure 4. A factored CRBM whose interactions are gated by real-valued stylistic features.

Deep Belief Networks (DBNs): In simple words, it is just the top two Hidden layer of nodes of a stacked RBM that forms DBN. It is an undirected associated memory from these top 2 layers and the remaining hidden layers form a directed acyclic graph that converts the representations in the associative memory into observable variables such as the pixels of an image. This hybrid model has some attractive features: > A fast, greedy learning algorithm that can find a fairly good set of parameters quickly, even in deep networks with millions of parameters and many hidden layers. >Unsupervised learning algorithm that can still be applied to labelled data by learning a model that generates both the label and the data. > A fine-tuning algorithm that learns an excellent generative model outperforming discriminative methods on the MNIST database of hand-written digits. >Generative model makes it easy to interpret the distributed representations in the deep hidden layers. > Inference required for forming a perceptron is both fast and accurate. > Learning algorithm is local: adjustments to a synapse strength depend only on the states of the pre-synaptic and post-synaptic neuron. > Neurons require to communicate only their stochastic binary states. | In terms of training Deep Belief Nets, there are two types of algorithms: **Greedy Layer-wise Training** & **Wake-Sleep Algorithm**.

Greedy Layer-wise Training: The top two layers have undirected connections and form an associative memory and the layers below have directed, top-down, generative connections that can be used to map a state of the associative memory to an image. There are also directed, bottom-up, recognition connections that are used to infer a factorial representation in one layer from the binary activities in the layer below. In the greedy initial learning the *recognition* connections are tied to the generative connections. If this greedy algorithm changes the higher-level weight matrices, it is guaranteed to improve the generative model. The greedy algorithm can clearly be also applied recursively, so if we use the full maximum likelihood Boltzmann machine learning algorithm to learn each set of tied weights and then we untie the bottom layer of the set from the weights above, we can learn the weights one layer at a time with a guarantee to never decrease the log probability of the data under full generative model. In practice, we replace maximum likelihood Boltzmann machine learning algorithm by contrastive divergence learning because it works well and is much faster. The use of contrastive divergence voids the guarantee, but it is still reassuring to know that extra layers are guaranteed to improve imperfect models if we learn each layer with sufficient patience. To guarantee that the generative model is improved by greedily learning more layers, it is convenient to consider models in which all layers are of the same size so that higher level weights can be initialized to values learned before they are untied from the weights in the layer below. The same greedy algorithm, however, can be applied even when the layers are of different sizes as well.

Wake-Sleep Algorithm: An unsupervised learning algorithm for a multilayer network of stochastic neurons is assessed. Bottom-up *recognition* connections convert the input into representations in successive hidden layers and top-down *generative* connections reconstruct the representation in one layer from the representation in the layer above. In the '**wake**' phase, neurons are driven by recognition connections, and generative connections are adapted to increase the probability that they would reconstruct the correct activity vector in the layer below. In the '**sleep**' phase, neurons are driven by generative connections and recognition connections are adapted to increase the probability that they would produce the correct activity vector in the layer above.

· · ·

Apart from these, we also have **Deep Boltzmann Machines (DBMs)** that has been explained in-depth by Ruslan Salakhutdinov and Hugo Larochelle in their research paper, where algorithm learns a separate *recognition* model that is used to quickly initialize, in a single bottom-up pass, the values of the latent variables in all hidden layers. Covering this concept is beyond the scope of my post as I have not yet read this paper. Within a couple of days, I shall also attach my GitHub link for relevant code to help implement Restricted Boltzmann Machine as a Recommender System using PyTorch (on *Movielens* dataset) as well as TensorFlow (on *MNIST* dataset). Please use Comments section for suggesting any corrections or feedback that you might have. I have tried my best to keep Mathematics out of this entire explanation & every now & then attached relevant links that could be useful for reader to understand concepts so please do make use of it. Enjoy Deep Learning!