# Notes on Visualizing High-Layer Features of Deep Network

compiled by D.Gueorguiev 5/18/24

Two models are considered in this paper: the first one is Deep Belief Net (DBN) discussed by Geoffrey Hinton in his 2006 paper “A fast learning algorithm for deep belief nets”. It is obtained by training and stacking three layers as Restricted Boltzman Machines (RBM) in a greedy manner.

//TODO: finish the notes on the main sections of interest in the paper (in References)

## References

[Visualizing Higher Level Features of a Deep Network, Dumitru Erhan et al, Universite de Montreal, 2009](https://github.com/dimitarpg13/deep_learning_for_image_processing/blob/main/literature/articles/interpretability/Visualizing_Higher-Layer_Features_of_a_Deep_Network_Erhan_2009.pdf)

[Simulated Annealing, Wikipedia](https://en.wikipedia.org/wiki/Simulated_annealing)

[Boltzman Machine, Wikipedia](https://en.wikipedia.org/wiki/Boltzmann_machine)

[Restricted Boltzman Machine, Wikipedia](https://en.wikipedia.org/wiki/Restricted_Boltzmann_machine)

## Appendix

### Simulated Annealing

//TODO: finish the section on Simulated Annealing

### Boltzman Machine

A Boltzman machine is a network of units with a total “energy” in analogy to classical mechanics represented by Hamiltonian and defined for the overall network. Its units produce binary results. Boltzman machine weights are stochastic. The global energy E in a Boltzman machine is identical to that of Hopfield networks and Ising models and it is given with:

(A.1)

In (A.1) represents the connection strength between unit and unit . is the state, of unit .

is the bias of unit in the global energy function. ( is the activation threshold for the unit.)

Often the weights are represented as a symmetric matrix with zeros along the diagonal.

A diagram of a network

Description automatically generated

Figure A.1: graphical representation of Boltzman machine with some weights labeled. Each undirected edge represents a dependency, and it is weighed with weight . In this example there are 3 hidden units depicted with blue and 4 visible units depicted with white. Therefore, this is not a Restricted Boltzman Machine (for definition see the next section on Restricted Boltzman Machines).

**Unit state probability**

The difference in the global energy that results from a single unit equaling (off) vs (on) denoted with , assuming a symmetric matrix of weighs, is given by:

This can be expressed as the difference of energies of two states:

Substituting the energy of each state with its relative probability according to the Boltzmann factor (the property of a Boltzmann distribution that the energy of a state is proportional to the negative log probability of that state) gives:

where is the Boltzmann constant and is absorbed into the artificial notion of temperature . We then rearrange terms and consider the probabilities of the unit on and off must sun to one:

Solving for gives

This relation is the source of the logistic function found in probability expressions in variants of the Boltzmann machine.

**Equilibrium state**

The network runs by repeatedly choosing a unit and resetting its state. After running for long enough at a certain temperature, the probability of a global state of the network depends only upon that global state’s energy, according to a Boltzmann distribution, and not on the initial state from which the process was started. This means that the log-probabilities of global states become linear in their energies. This relationship is true when the machine is “at thermal equilibrium”, meaning that the probability distribution of global states has converged. Running the network beginning from a high temperature, its temperature gradually decreases until reaching a thermal equilibrium at a lower temperature. It then may converge to a distribution where the energy level fluctuates around the global minimum. The process is known as *Simulated Annealing*.

To train the network so that the chance it will converge to a global state according to an external distribution over these states, the weighs must be set so that the global states with the highest probabilities get the lowest energies. This is done by training.

**Training the Boltzman machine**

The units in the Boltzmann machine are divided into ‘visible’ units, , and ‘hidden’ units, . The visible units are those that receive information from the ‘environment’, i.e. the training set is a set of binary vectors over the set . The distribution over the training set is denoted .

The distribution over global states converges as the Boltzmann machine reaches thermal equilibrium. We marginalize this distribution over the hidden units and denote it with .

Our goal is to approximate the “real” distribution using the produced by the machine. The similarity of the two distributions is measured by the Kullback-Leibler divergence, :

where the sum is over all possible states of . is a function of the weights, since they determine the energy of a state, and the energy determines , as promised by the Boltzmann distribution. A gradient descent algorithm over changes a given weight, , by subtracting the partial derivative of with respect to the weight.

Boltzman machine training involves two alternating phases. One is the “positive” phase where the visible units’ states are clamped to a particular binary state vector sampled from the training set (according to ). The other is the “negative” phase where the network is allowed to run freely, i.e. only the input nodes have their state determined by external data, but the output nodes are allowed to float.

//TODO: finish the section on training Boltzmann machine

### Restricted Boltzman Machines

//TODO: finish the section on Restricted Boltzmann Machines