Summary

sagisk

February 5, 2022

1 Chapter 6: Restricted Boltzmann Machines (RBM)

ToDo: add applications.

RBM is not similar to other conventional NNs. While NNs try to minimize the loss function for a set of inputs, the RBMs are trying to learn the joint probabilitic relationships between the inputs and some hidden attributes. RBMs are considered to be unsupervised models that generate latent feature representations of the data.

Hopfield Networks: Is an undirected network with d units where d is the dimension of the training data. Each connection is of the form (i, j). The weights between the units are symmetric $w_{ij} = w_{ji}$ parameters. High positive weights between units indicate the high degree of positive correlation in state values, whereas large negative weights speak about high negative correlation. Each unit has an associated state binary s_i corresponding to the ith bit in the particular training example. Each node has an associated bias b_i. - Loss: Here the loss function is called the energy function. It is constructed such that minimizing this function encourages the nodes with similar states to have high positive weights and with the ones with drastically different values to have large negative weights. For a particular set of states $s = (s_1, ..., s_d)$ the energy function can be written as $E = -\sum_i b_i s_i - \sum_{i,j:i<j} w_{ij} s_i s_j$.

Where $w_{ij} > 0$ encourages s_i, s_j to be similar and large bias b_i encourages $s_i = 1$ since we are minimizing E.

Optimal state given weights: The optimal state helps the network in recalling memories. An iterative state flipping algorithm is going over all units and flips or no the state value s_i of each ith unit. The value is flipped if doing so minimizes the energy function. Hence, one can compute: $\Delta E_i = E_{s_i=0} - E_{s_i=1} = b_i + \sum_{j:j\neq i} w_{ij} s_j$ and one sets $s_i = 1$ if $\Delta E_i \geq 0$ else $s_i = 0$. The optimal state dependes on the initial training point.

Training a Hopfield Network: Hopfield Network has a storage capacity of $0.15 * d$ where d is the number of units in the network.

- Weight update: $w_{ij} = w_{ij} + 4(x_{ki} - 0.5)(x_{kj} - 0.5)$ where x_{ki} is the k th binary element of the ith training point. $b_i = b_i + 2(x_{ki} - 0.5)$.

1.1 The Boltzmann Machine

The Biltzmann Machine is similar to the Hopfield Network except severl significant differences.

- There are hidden and visible states.
- The total number of states is $q = (m + d)$ where d-is the number of visible states, m is the number of hidden states.
- The model is probabalistic whereas, the Hopfield Network is deterministic. Here we don't just flip the ith state value s_i but assign a proabability to it $P(s_i = 1|s_1, ..., s_{i-1}, s_{i+1}...s_q) = \frac{1}{1+e^{-\Delta E_i}}$ where ΔE_i is the energy gap of the state s_i.
- Each state configarteion model is assumed to be $P(S) \propto e^{-E(s)} = \frac{1}{z} e^{-E(s)}$.

Data Generation: Iteratively samples the states using the conditional distribution of the state values until thermal equilibrium. Thermal equilibrium means that we start at random set of states, compute conditional probabilities and sample new values for the states from these probabilities.

Training the Boltzmann Machine:

The idea is to maximize the log-likelihood of the individual states: $\log P(s) = -E(s) - \log Z$. Note that $\frac{d \log P(s)}{dw_{ij}} =< s_i, s_j >_{\text{data}} - < s_i, s_j >_{\text{model}}$.

Here: Data-centric sample: $< s_i, s_j >_{\text{data}}$ is the average value of $s_i s_j$ after running the generative process where visible states are set to the training points. The hidden states are initialized to the Bernoulli with $p = 0.5$. Then using the conditional probability formula probabilities of hidden states is generated and the process is repeated so that thermal equilibrium is reached.

Model sample: $< s_i, s_j >_{\text{model}}$ is the average value of $s_i s_j$ after running the generative process where without fixing the visible states. Both hidden and visible states are initialized to random values and updated using conditional probability formula until thermal equilibrium is reached.

$w_{ij} = w_{ij} + a(< s_i, s_j >_{\text{data}} - < s_i, s_j >_{\text{model}})$

$b_i = b_i + a(< s_i, 1 >_{\text{data}} - < s_i, 1 >_{\text{model}})$

1.2 Restricted Boltzmann Machines (RBM)

The Restricted Boltzmann Machines unlike the Boltzmann Machines restrict the connections between units so that the only connections are allowed are between hidden and visible units. Hence, RBM is a bipartite.

$P(h_j|v) = \frac{1}{1+e^{-b^h_j - \sum_{i=1}^{d} v_i w_{ij}}}$ $P(v_i|h) = \frac{1}{1+e^{-b^v_j - \sum_{j=1}^{m} h_j w_{ij}}}$ Training the RBM: $w_{ij} = w_{ij} + a(< v_i, h_j >_{\text{pos}} - < v_i, h_j >_{\text{neg}})$ $b^v_i = b^v_i + a(< v_i, 1 >_{\text{pos}} - < v_i, 1 >_{\text{neg}})$ $b^h_j = b^h_j + a(< 1, h_j >_{\text{pos}} - < 1, h_j >_{\text{neg}})$

Where: Positive phase: The mini-batch of training instances is given the $P(h_j|v)$ is computed and a sample of hidden state of each hidden unit is generated from this probability. The process is repeated for all mini-batches of training instances. The correlation between these training instaces v_i and generated hidden state instances h_j is computed: $< v_i, h_j >_{\text{pos}}$.

Negative phase: With a mini batch of training data the $P(h_j|v)$ is computed and a sample is drawn. Using this sample $P(v_i|h)$ is computed and a sample is drawn. The previous two steps are repeated till the thermal equilibrium. Then the correlation is computed as: $< v_i, h_j >_{\text{neg}}$. The Contrastive Divergence Algorithm: One of the disadvantages of the training algorithm for RBM is that it takes a considerable amount of time and computational resources to get to the thermal equilibrium. Hence, a contrastive divergence algorithm is created. The idea of which is to:

- Generate hidden states by fixing the visible units by computing $P(h_j|v)$ and then generating a sample of hidden states using it.
- Then the visible units are generated from theses hidden states and are used in stead of the termal equilibrium states.
- Hidden states are generated again.

References