Title: Boltzmann machine

URL: https://en.wikipedia.org/wiki/Boltzmann_machine

PageID: 1166059

Categories: Category:Ludwig Boltzmann, Category:Mathematical physics, Category:Neural network architectures

---

A Boltzmann machine (also called Sherrington–Kirkpatrick model with external field or stochastic Ising model), named after Ludwig Boltzmann , is a spin-glass model with an external field, i.e., a Sherrington–Kirkpatrick model , [1] that is a stochastic Ising model . It is a statistical physics technique applied in the context of cognitive science . [2] It is also classified as a Markov random field . [3]

Boltzmann machines are theoretically intriguing because of the locality and Hebbian nature of their training algorithm (being trained by Hebb's rule), and because of their parallelism and the resemblance of their dynamics to simple physical processes . Boltzmann machines with unconstrained connectivity have not been proven useful for practical problems in machine learning or inference , but if the connectivity is properly constrained, the learning can be made efficient enough to be useful for practical problems. [4]

They are named after the Boltzmann distribution in statistical mechanics , which is used in their sampling function . They were heavily popularized and promoted by Geoffrey Hinton , Terry Sejnowski and Yann LeCun in cognitive sciences communities, particularly in machine learning , [2] as part of " energy-based models " (EBM), because Hamiltonians of spin glasses as energy are used as a starting point to define the learning task. [5]

Structure

A Boltzmann machine, like a Sherrington–Kirkpatrick model , is a network of units with a total "energy" (Hamiltonian) defined for the overall network. Its units produce binary results. Boltzmann machine weights are stochastic . The global energy E {\displaystyle E} in a Boltzmann machine is identical in form to that of Hopfield networks and Ising models :

Where:

w i j {\displaystyle w_{ij}} is the connection strength between unit j {\displaystyle j} and unit i {\displaystyle i} .

s i {\displaystyle s_{i}} is the state, s i ∈ { 0 , 1 } {\displaystyle s_{i}\in \{0,1\}} , of unit i {\displaystyle i} .

θ i {\displaystyle \theta _{i}} is the bias of unit i {\displaystyle i} in the global energy function. (− θ i {\displaystyle -\theta _{i}} is the activation threshold for the unit.)

Often the weights w i j {\displaystyle w_{ij}} are represented as a symmetric matrix W = [w i j] {\displaystyle W=[w_{ij}]} with zeros along the diagonal.

Unit state probability

The difference in the global energy that results from a single unit i {\displaystyle i} equaling 0 (off) versus 1 (on), written Δ E i {\displaystyle \Delta E_{i}} , assuming a symmetric matrix of weights, 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)

yields:

where k B {\displaystyle k_{B}} is the Boltzmann constant and is absorbed into the artificial notion of temperature T {\displaystyle T} .

Noting that the probabilities of the unit being on or off sum to 1 {\displaystyle 1} allows for the simplification:

whence the probability that the i {\displaystyle i} -th unit is given by

where the scalar T {\displaystyle T} is referred to as the temperature of the system.

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 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. This process is called 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 weights must be set so that the global states with the highest probabilities get the lowest energies. This is done by training.

Training

The units in the Boltzmann machine are divided into 'visible' units, V, and 'hidden' units, H. 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 V. The distribution over the training set is denoted P + (V) {\displaystyle P^{+}(V)} .

The distribution over global states converges as the Boltzmann machine reaches thermal equilibrium . We denote this distribution, after we marginalize it over the hidden units, as P – (V) {\displaystyle P^{-}(V)} .

Our goal is to approximate the "real" distribution P + (V) {\displaystyle P^{+}(V)} using the P – (V) {\displaystyle P^{-}(V)} produced by the machine. The similarity of the two distributions is measured by the Kullback–Leibler divergence , G {\displaystyle G} :

where the sum is over all the possible states of V {\displaystyle V} . G {\displaystyle G} is a function of the weights, since they determine the energy of a state, and the energy determines P – (v) {\displaystyle P^{-}(v)} , as promised by the Boltzmann distribution. A gradient descent algorithm over G {\displaystyle G} changes a given weight, w i j {\displaystyle w_{ij}} , by subtracting the partial derivative of G {\displaystyle G} with respect to the weight.

Boltzmann 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 P + {\displaystyle P^{+}}). 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. The gradient with respect to a given weight, w i j {\displaystyle w_{ij}} , is given by the equation: [2]

where:

p i j + {\displaystyle p_{ij}^{+}} is the probability that units i and j are both on when the machine is at equilibrium on the positive phase.

p i j – {\displaystyle p_{ij}^{-}} is the probability that units i and j are both on when the machine is at equilibrium on the negative phase.

R {\displaystyle R} denotes the learning rate

This result follows from the fact that at thermal equilibrium the probability P − (s) {\displaystyle P^{-}(s)} of any global state s {\displaystyle s} when the network is free-running is given by the Boltzmann distribution.

This learning rule is biologically plausible because the only information needed to change the weights is provided by "local" information. That is, the connection (synapse , biologically) does not need information about anything other than the two neurons it connects. This is more biologically realistic than the information needed by a connection in many other neural network training algorithms, such as backpropagation .

The training of a Boltzmann machine does not use the EM algorithm , which is heavily used in machine learning . By minimizing the KL-divergence , it is equivalent to maximizing the log-likelihood of the data. Therefore, the training procedure performs gradient ascent on the log-likelihood of the observed data. This is in contrast to the EM algorithm, where the posterior distribution of the hidden nodes must be calculated before the maximization of the expected value of the complete data likelihood during the M-step.

Training the biases is similar, but uses only single node activity:

Problems

Theoretically the Boltzmann machine is a rather general computational medium. For instance, if trained on photographs, the machine would theoretically model the distribution of photographs, and could use that model to, for example, complete a partial photograph.

Unfortunately, Boltzmann machines experience a serious practical problem, namely that it seems to stop learning correctly when the machine is scaled up to anything larger than a trivial size. [citation needed] This is due to important effects, specifically:

the required time order to collect equilibrium statistics grows exponentially with the machine's size, and with the magnitude of the connection strengths [citation needed]

connection strengths are more plastic when the connected units have activation probabilities intermediate between zero and one, leading to a so-called variance trap. The net effect is that noise causes the connection strengths to follow a random walk until the activities saturate.

Types

Restricted Boltzmann machine

Although learning is impractical in general Boltzmann machines, it can be made quite efficient in a restricted Boltzmann machine (RBM) which does not allow intralayer connections between hidden units and visible units, i.e. there is no connection between visible to visible and hidden to hidden units. After training one RBM, the activities of its hidden units can be treated as data for training a higher-level RBM. This method of stacking RBMs makes it possible to train many layers of hidden units efficiently and is one of the most common deep learning strategies. As each new layer is added the generative model improves.

An extension to the restricted Boltzmann machine allows using real valued data rather than binary data. [6]

One example of a practical RBM application is in speech recognition. [7]

Deep Boltzmann machine

A deep Boltzmann machine (DBM) is a type of binary pairwise Markov random field (undirected probabilistic graphical model) with multiple layers of hidden random variables . It is a network of symmetrically coupled stochastic binary units . It comprises a set of visible units ν ∈ { 0 , 1 } D {\displaystyle {\boldsymbol {\nu }}\in \{0,1\}^{D}} and layers of hidden units h (1) ∈ { 0 , 1 } F 1 , h (2) ∈ { 0 , 1 } F 2 , … , h (L) ∈ { 0 , 1 } F L {\displaystyle {\boldsymbol {h}}^{(1)}\in \{0,1\}^{F_{1}},{\boldsymbol {h}}^{(2)}\in \{0,1\}^{F_{2}},\ldots ,{\boldsymbol {h}}^{(L)}\in \{0,1\}^{F_{L}}} . No connection links units of the same layer (like RBM). For the DBM , the probability assigned to vector ν is

where h = { h (1) , h (2) , h (3) } {\displaystyle {\boldsymbol {h}}=\{{\boldsymbol {h}}^{(1)},{\boldsymbol {h}}^{(2)},{\boldsymbol {h}}^{(3)}\}} are the set of hidden units, and θ = { W (1) , W (2) , W (3) } {\displaystyle \theta =\{{\boldsymbol {W}}^{(1)},{\boldsymbol {W}}^{(2)},{\boldsymbol {W}}^{(3)}\}} are the model parameters, representing visible-hidden and hidden-hidden interactions. [8] In a DBN only the top two layers form a restricted Boltzmann machine (which is an undirected graphical model), while lower layers form a directed generative model. In a DBM all layers are symmetric and undirected.

Like DBNs , DBMs can learn complex and abstract internal representations of the input in tasks such as object or speech recognition , using limited, labeled data to fine-tune the representations built using a large set of unlabeled sensory input data. However, unlike DBNs and deep convolutional neural networks , they pursue the inference and training procedure in both directions, bottom-up and top-down, which allow the DBM to better unveil the representations of the input structures. [9] [10] [11]

However, the slow speed of DBMs limits their performance and functionality. Because exact maximum likelihood learning is intractable for DBMs, only approximate maximum likelihood learning is possible. Another option is to use mean-field inference to estimate data-dependent expectations and approximate the expected sufficient statistics by using Markov chain Monte Carlo (MCMC). [8] This approximate inference, which must be done for each test input, is about 25 to 50 times slower than a single bottom-up pass in DBMs. This makes joint optimization impractical for large data sets, and restricts the use of DBMs for tasks such as feature representation.

Spike-and-slab RBMs

The need for deep learning with real-valued inputs, as in Gaussian RBMs, led to the spike-and-slab RBM (ss RBM), which models continuous-valued inputs with binary latent variables . [12] Similar to basic RBMs and its variants, a spike-and-slab RBM is a bipartite graph , while like G RBMs , the visible units (input) are real-valued. The difference is in the hidden layer, where each hidden unit has a binary spike variable and a real-valued slab variable. A spike is a discrete probability mass at zero, while a slab is a density over continuous domain; [13] their mixture forms a prior . [14]

An extension of ss RBM called μ-ss RBM provides extra modeling capacity using additional terms in the energy function . One of these terms enables the model to form a conditional distribution of the spike variables by marginalizing out the slab variables given an observation.

In mathematics

In more general mathematical setting, the Boltzmann distribution is also known as the Gibbs measure . In statistics and machine learning it is called a log-linear model . In deep learning the Boltzmann distribution is used in the sampling distribution of stochastic neural networks such as the Boltzmann machine.

History

The Boltzmann machine is based on the Sherrington–Kirkpatrick spin glass model by David Sherrington and Scott Kirkpatrick . [15] The seminal publication by John Hopfield (1982) applied methods of statistical mechanics, mainly the recently developed (1970s) theory of spin glasses, to study associative memory (later named the "Hopfield network"). [16]

The original contribution in applying such energy-based models in cognitive science appeared in papers by Geoffrey Hinton and Terry Sejnowski . [17] [18] [19] In a 1995 interview, Hinton stated that in 1983 February or March, he was going to give a talk on simulated annealing in Hopfield networks, so he had to design a learning algorithm for the talk, resulting in the Boltzmann machine learning algorithm. [20]

The idea of applying the Ising model with annealed Gibbs sampling was used in Douglas Hofstadter 's Copycat project (1984). [21] [22]

The explicit analogy drawn with statistical mechanics in the Boltzmann machine formulation led to the use of terminology borrowed from physics (e.g., "energy"), which became standard in the field. The widespread adoption of this terminology may have been encouraged by the fact that its use led

to the adoption of a variety of concepts and methods from statistical mechanics. The various proposals to use simulated annealing for inference were apparently independent.

Similar ideas (with a change of sign in the energy function) are found in Paul Smolensky 's "Harmony Theory". [23] Ising models can be generalized to Markov random fields , which find widespread application in linguistics , robotics , computer vision and artificial intelligence .

In 2024, Hopfield and Hinton were awarded Nobel Prize in Physics for their foundational contributions to machine learning , such as the Boltzmann machine. [24]

See also

Restricted Boltzmann machine

Helmholtz machine

Markov random field (MRF)

Ising model (Lenz–Ising model)

Hopfield network

References

Further reading

Hinton, G. E. ; Sejnowski, T. J. (1986). D. E. Rumelhart; J. L. McClelland (eds.). "Learning and Relearning in Boltzmann Machines" (PDF) . Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations : 282– 317. Archived from the original (PDF) on 2010-07-05.

Hinton, G. E. (2002). "Training Products of Experts by Minimizing Contrastive Divergence" (PDF) . Neural Computation . 14 (8): 1771– 1800. CiteSeerX 10.1.1.35.8613 . doi : 10.1162/089976602760128018 . PMID 12180402 . S2CID 207596505 .

Hinton, G. E. ; Osindero, S.; Teh, Y. (2006). "A fast learning algorithm for deep belief nets" (PDF) . Neural Computation . 18 (7): 1527– 1554. CiteSeerX 10.1.1.76.1541 . doi : 10.1162/neco.2006.18.7.1527 . PMID 16764513 . S2CID 2309950 .

Kothari P (2020): https://www.forbes.com/sites/tomtaulli/2020/02/02/coronavirus-can-ai-artificial-intelligence-make-a-difference/?sh=1eca51e55817

Montufar, Guido (2018). "Restricted Boltzmann Machines: Introduction and Review" (PDF) . MPI MiS (Preprint) . Retrieved 1 August 2023 .

External links

Scholarpedia article by Hinton about Boltzmann machines

Talk at Google by Geoffrey Hinton

v

t

e

Principle of maximum entropy

ergodic theory

Ensembles

partition functions

equations of state

thermodynamic potential : U H F G

U

H

F

G

Maxwell relations

Ferromagnetism models Ising Potts Heisenberg percolation

Ising

Potts

Heisenberg

percolation

Particles with force field depletion force Lennard-Jones potential

depletion force

Lennard-Jones potential

Boltzmann equation

H-theorem

Vlasov equation

BBGKY hierarchy

stochastic process

mean-field theory and conformal field theory

Phase transition

Critical exponents correlation length size scaling

correlation length

size scaling

Boltzmann

Shannon

Tsallis

Rényi

von Neumann

Statistical field theory elementary particle superfluidity

elementary particle

superfluidity

Condensed matter physics

Complex system chaos information theory Boltzmann machine

chaos

information theory

Boltzmann machine

GND