# CMPTG 5 - Statistical Physics of Neural Networks

Sidharth Kannan

April 2025

## 1 Lecture 2 - Mean Field Theory and Ising Machines

### 1.1 From Maxwell-Boltzmann to Fermi-Dirac

In the last lecture, we derived the Boltzmann distribution, which in principle lets us calculate the probability of any particular state of our system. However, calculating the full Boltzmann distribution for some arbitrary configuration requires calculating the partition function, which in general is an exponential complexity operation. Let's see if we can make headway with some simplifying assumptions.

We can start by considering the following setup. Imagine instead of the infinite energy levels we had last lecture, we consider a system with 10 energy levels, $E_1, ..., E_{10}$, and each can either be occupied or empty. Then, a microstate is a 10 bit string, e.g. 1001001101, where the n-th bit represents whether or not the n-th energy level is occupied. Let us also assume that the probability of each energy level being occupied is *independent*. That is, whether or not level 1 is occupied is not affected by whether or not level 2 is occupied, and so on.

Say I want to calculate the probability that the 1st energy level is occupied. In the Boltzmann statistics approach, this comes down to calculating the *marginal* probability distribution, $\mathcal{P}(E_1)$. We would sum up all of the Boltzmann factors corresponding to the states "1XXXXXXXXX", and divide the result by the partition function.

For 10 energy levels, this would involve summing up $2^{10} = 1024$ terms, while for 20 energy levels it would involve 1,048,576 terms, and for just 60 energy levels, this explodes to $\mathcal{O}(10^{18})$ terms in the summation. Clearly this is not a scalable strategy.

Now it turns out that the marginal in the non-interacting case has a closed form, called the *Fermi-Dirac distribution* (or, if you ask Dirac, just the Fermi distribution), which we can come to with the following argument. Since the energy levels are independent, we can consider just one energy level. The two states are "occupied" ($E = E_i$) or "unoccupied" ($E = 0$) , which gives that

$$Z = e^{-E_i/kT} + e^0 = 1 + e^{-E_i/kT} \tag{1}$$

and so the probability of occupation is just

$$\mathcal{P}(E_i \text{ occupied}) = \frac{1}{1 + e^{E_i/kT}} \tag{2}$$

And since we can calculate our non-interacting marginals in closed form, we can calculate the probability of occupation of each energy level in constant time! However, this assumption of non-interaction is overly restrictive for our applications, so let's see if we can come up with a model that can approximate the full Boltzmann distribution for interacting systems.

## 1.2 Mean-Field Theory

Just to reiterate, the source of the problem is that, in the case of Boltzmann statistics, the occupation probabilities are not independent. You can think of this as due to some Coulomb repulsion between electrons, so if energy level 1 is occupied, there is a repulsive force that disincentivizes the occupation of energy level 2. As such, we need to consider *each* configuration.

We might come up with the following simplistic model for this:

$$\tilde{E}_i = E_i + U \sum_i \sum_{j<i} n_i \cdot n_j \tag{3}$$

Instead of treating energy levels, and thus occupation probabilities, as independent, we can instead say this. Imagine energy level one is occupied all of the time. Then, energy levels 2-10 would always feel some extra potential, of strength $U$, that they would have to overcome in order to be occupied. If, instead, we knew that level 1 was occupied half the time, then we might imagine that the other energy levels, *on average*, experience a force of $\frac{1}{2}U$.

Instead of modelling the unique interactions in each microstate, what if we defined an *average* interaction between each pair of energy levels, and calculated the occupation probabilities with that?

Let's take our Fermi function,

$$f(E_i) = \frac{1}{1 + e^{E_i/kT}} \tag{4}$$

which gives us the average occupation probability of energy level $i$. For each energy level, define

$$\tilde{E}_i = E_i + U \sum_{j \neq i} f(E_j) \tag{5}$$

We can then compute the new occupation probabilities with $f(\tilde{E}_i)$ and repeat this procedure until convergence. This is an example of a *mean field theory*, where instead of modelling the microscopic interactions, we model some average interaction between energy levels. Now, I imagine there may be a lot of questions about if and when this actually works. In practice, it turns out to work for a large number of physical systems. I will eschew any convergence proofs in favor of showing this experimentally for our 10 energy level system. See Fig. 1.

I would like to provide an alternative look at this whole electron/energy level system that we have been discussing. Each energy level is a binary variable that has its own preference for being occupied or empty, which is somehow influenced by interactions with the other energy levels.

There is another physical system, which will be of great importance to us in a moment, that generalizes the kind of behavior we have been discussing: The Ising spin glass. Instead of energy levels, an Ising spin glass is comprised of spins, which either point up or down. These spins each can interact with each other, with varying strength, and each can experience some external magnetic field, which biases the direction that it points in. See Fig. 2. A useful mathematical object for representing this is as a weighted, undirected graph.

Let's step back for a moment and think about what we've done here. This is our first hint as to how we might translate a (simulated) physical system, and map it to something resembled a machine learning scheme. We took a simple distribution (Fermi-Dirac), and came up with an algorithm that allows us to map it to a far more complex distribution (Boltzmann), in a computationally efficient manner.

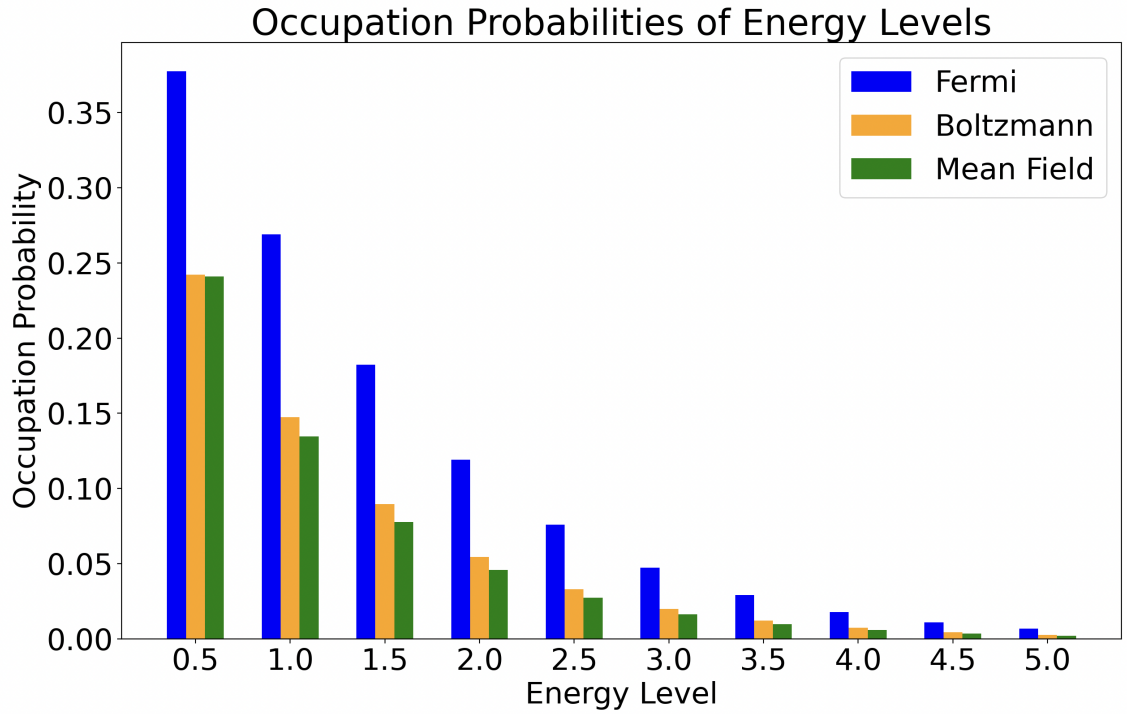


Figure 1: Comparison of the occupation probabilities of a 10-level interacting system, as computed by the Fermi function, Boltzmann distribution, and mean-field theory.

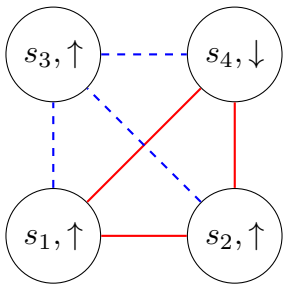


Figure 2: An example spin glass. Some are connected by positive couplings (blue) and negative couplings (red).

If we put aside the physics for a moment, and take a more abstracted view of what we've just done, we've taken some binary unit (energy level) in an interacting system, and defined a method of computing the "average" amount of time that the unit is on. Here comes the big conceptual leap. Instead of talking about energy levels and electrons, let's take a look at this abstracted unit that we've built, which I will call the binary stochastic neuron, for reasons that will become apparent.

The "neuron" takes some input (the corrected energy level), and outputs a probability of being

on (the Fermi function/activation). We can define the dynamics of our neuron through the following equations then,

$$m_i = \text{sgn}(\tanh \beta I_i + \text{rand}_U[-1, 1]) \tag{6}$$

Here, I've made a few subtle changes to our conventions, just to be consistent with the existing literature. $I_i$ plays the role of the interaction energy, and $m_i$ is the bi*polar* variable that indicates if the energy level/neuron is occupied or not, and I've defined $\beta \equiv 1/kT$.

Now imagine we have two of these units, uncoupled. There are 4 possible states, 00, 01, 10, 11, which are all equally probable, and because the BSNs are uncoupled, $I_i = 0$, so the BSNs randomly sample all states with equal probability. If instead we couple the output of each to the input of the other, we get a more interesting system. The couplings make it so that for each BSN, $I_i$ is larger when the other BSN is 1, thus biasing it towards 1, and conversely, when the other BSN is -1, this BSN tends towards -1. In other words, the system prefers states in which both neurons agree. We can see this empirically if we just simulate the forward iteration equation, Eq. 6. The results are shown in Fig. 3.

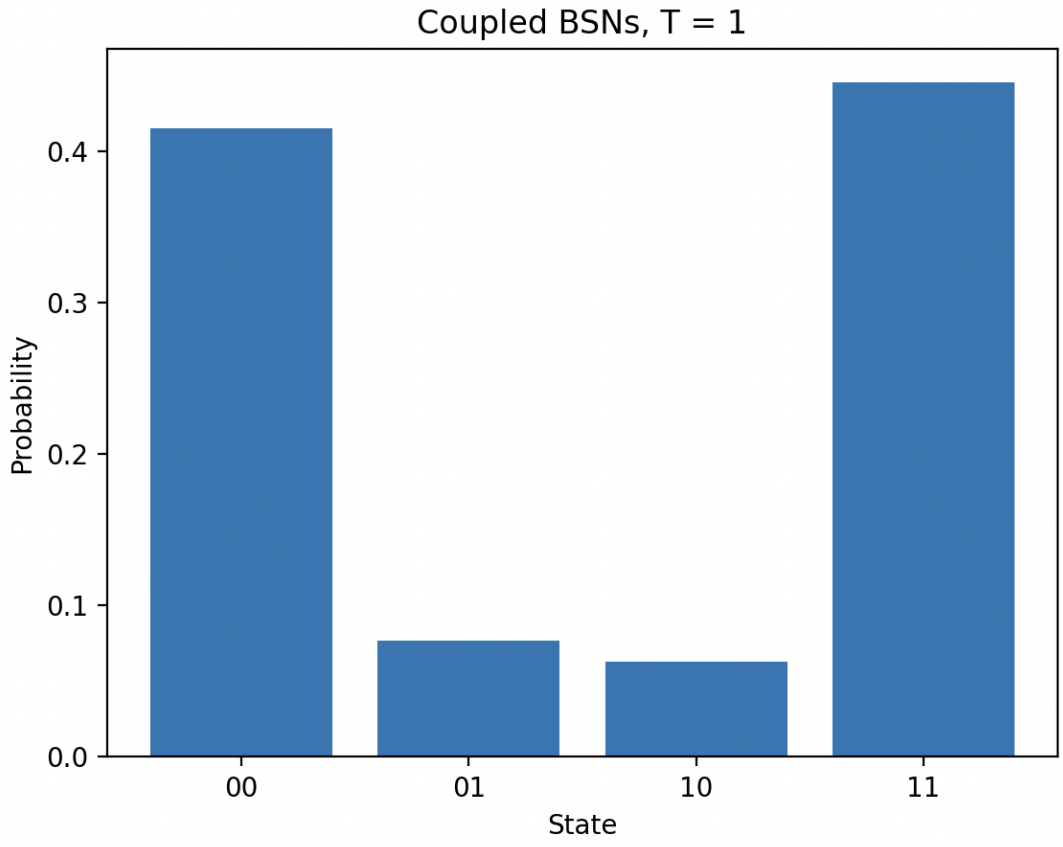


Figure 3: The state distribution for a network of two, ferromagnetically coupled BSNs.

Now, if we look at this histogram, we can see that, as expected, the system has a preference for certain states over others. What determines those preferred states, and what sets the probabilities that we observe?

Since we derived this whole model from our mean-field approximation of Boltzmann statistics, we might expect that there is some kind of energy function that determines our distribution of states. It turns out that there is. Before we get there though, let's introduce some notation to clean up our understanding of BSN networks. For any arbitrary BSN network, with $N$ neurons, we can write it as a graph, in the same way we discussed earlier. We can write down an $N$x$N$ matrix, $\mathbf{J}$, such that $\mathbf{J}_{ij}$ tells us the coupling strength between neuron $i$ and neuron $j$. We can also define a vector $\mathbf{b}$ of biases, that tell us how much each neuron itself wishes to have a certain orientation.

Then, we can write the "energy" of our BSN network as

$$E = \sum_i \sum_{j<i} J_{ij} m_i m_j + \sum_i h_i m_i \tag{7}$$

Or, in matrix form

$$E = \frac{1}{2}\mathbf{m}^T \mathbf{J}\mathbf{m} + \mathbf{h}^T \mathbf{m} \tag{8}$$

I am not proving these equations. This is because in the historical line of development, the energy equation came before the update equations, as it comes from physics. While it is possible to make heuristic arguments for deriving the energy function from the update equations, these arguments are not terribly englightening. We have a decent grasp over how energy functions work, and what they tell us, so we shall leave it at that.

As a brief aside, these networks are more commonly referred to in the literature as "Ising machines," so we will take up that terminology.

One more vital thing to mention is that in the prior discussion, we are assuming that the neurons are updated *in sequence*, not in parallel. Consider the case where we start the system in the 01 state. Heuristically, we can argue that since the coupling tells both neurons to agree with each other, neuron 1 sees that its neighbor is in the 1 state, and so it flips to the 1 state, while neuron 2 sees that its neighbor is in the 0 state, and so it flips to the 1 state. This kind of 01$\rightarrow$ 10 cycle requires that we use a serial update algorithm.

## 1.3 Invertible Computing

The next interesting question is what can we use these networks for? That is, what can we actually compute with them?

Let's start with the Ising machine defined by

$$\mathbf{J} = \begin{pmatrix} 0 & -1 & 2 \\ -1 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}, \mathbf{h} = \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix} \tag{9}$$

If we run this Ising machine, we see that out of the 8 states that are possible, 4 are likely and 4 are not. If we look at the 4 likely states, we see that they form the four likely states of an AND gate, where the first two bits are input, and the last is the output of the gate.

How do we compute with this AND gate? If we shift the biases on the inputs, we can "clamp" those bits to 1 or 0, and as a result, force (probabilistically), the output bit to the correct state.

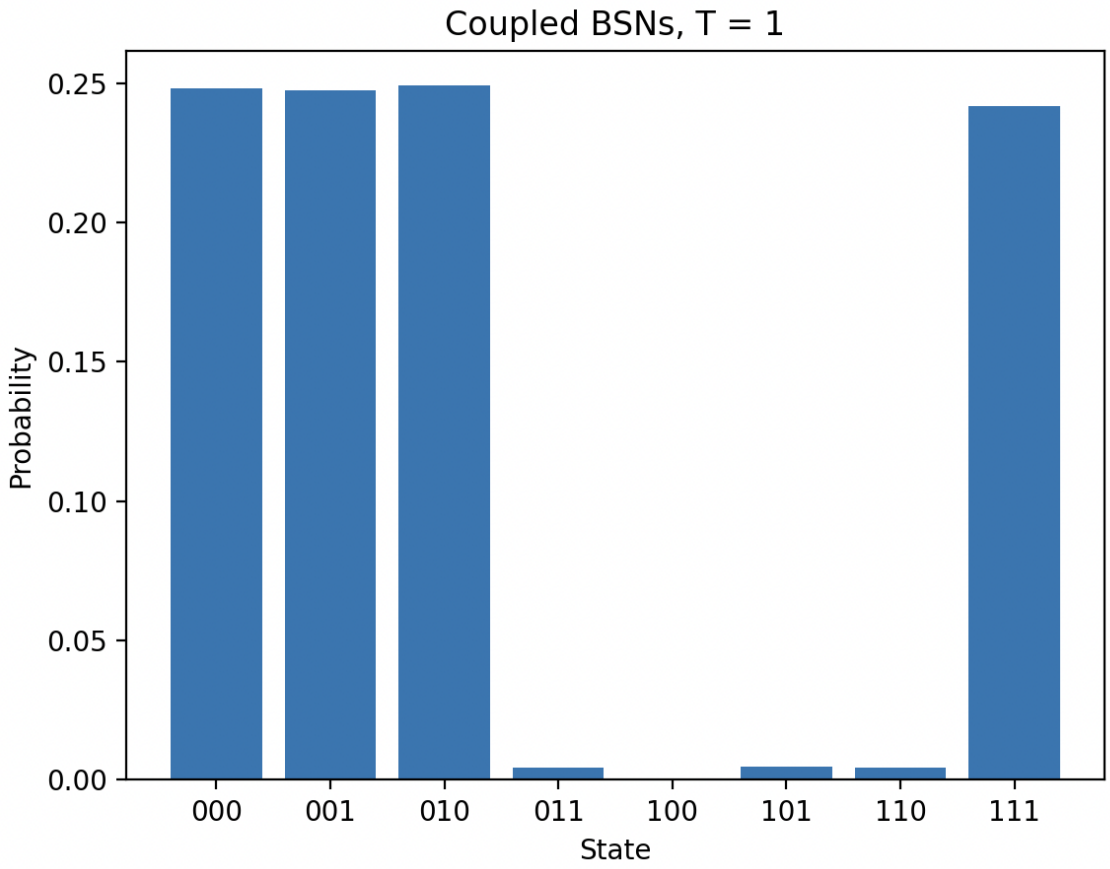


Figure 4: An AND gate, where the latter two bits are the input and the first is the output.

But we can also do something interesting that we can't with regular and gates. If we instead were to clamp the output bit to one, the input bits will be restricted to the states that correspond to valid inputs to produce that output.

This is our first glimpse into invertible computing, and we can see why this may be a useful feature. There are a great many problems in computer science where we are interested not only in the forward problem but also the reverse. For example, in principle, with an Ising machine, we could not just multiply numbers, but by clamping the outputs to the product, and letting the inputs free, we could factorize it. Alternatively, say we have an Ising machine that takes an image as input and outputs a classification label. We could easily turn this into a generative model by instead clamping the output label, and looking at the resulting input bits.

## 1.4 A Brief Note on Combinatorial Optimization

Recall that we began this whole discussion with the Boltzmann distribution. For a given spin glass, we see that the Ising machine eventually equilibrates and draws states such that the probability of being in a particular state is given by the Boltzmann distribution, which is to say that it samples states inversely proportional with their energy. One interesting application of this is in combinatorial optimization.

Consider the Random K-SAT problem, in which we have $N$ Boolean variables in a formula of the form

$$\mathcal{F} = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_3 \vee x_4) \wedge \dots \tag{10}$$

We can map this to a spin glass problem, where each variable is a spin, with spin up corresponding to TRUE and spin down corresponding to FALSE. We have already shown that we can construct a 2-input AND gate using spin glasses, and a similar method would enable us to construct OR and NOT gates as well. We can thus map this K-SAT problem to a spin glass optimization problem, and run our Ising machine to recover the ground state (corresponding to the solution).

In practice, there are more effective methods than simply running the Ising Machine and hoping for the best. The most famous of these is the algorithm known as simulated annealing. Simulated annealing works by starting the Ising machine in some random initial state, at very high temperature, and then as you draw samples, slowly decrease the temperature. In the high temperature phase, the spin glass quickly equilibrates, and is able to explore a large portion of the phase space. As the glass cools, spins become less inclined to flip, and settle to the ground state.