

**Statistical Mechanics, Neural
Networks, and Artificial Intelligence:
*Using Powerful Brain Strategies to
Improve AI***

DRAFT Chapter 10:
Introduction to Statistical Mechanics:
Microstates and the Partition Function

Alianna J. Maren
Northwestern University School of Professional Studies
Master of Science in Data Science Program

Draft: 2020-01-10

10.1 Introduction and Overview

In the previous chapter, we saw that key neural network / deep learning advances hinged on using statistical mechanics as a metaphor for creating stable and functional neural networks. In particular, we looked at the *energy functions* for the Hopfield network and the (restricted) Boltzmann machine, or RBM. We made a connection between the nature, or *function* inherent in these energy equations and the associated *forms* of their respective neural networks.

Although we were able to intuit a great deal about how these networks worked (both the Hopfield and the RBM), we were taking the notion of an *energy equation* on faith. We accepted that it came from the realm of statistical mechanics, and we accepted that minimizing the appropriate energy equations would yield a stable set of connection weights that allowed the network to perform a useful task.

However, we haven't yet attempted to understand how these energy equations came about. Also, at this stage, we don't know how to precisely map the notions that we've learned in the realm of statistical mechanics to the realm of neural networks. Once again, we accepted (on faith) that such a mapping existed and was important.

Also, in the previous chapter, we made no effort to identify a learning rule. That is, we don't know *yet* exactly *how* the energy minimization process can work to our advantage. Specifically, we don't know yet how energy minimization can give us a set of connection weights that will enable a neural network to identify patterns, perform pattern completion, and all of the other tasks that we'd like a neural network to do.

The goal of this chapter is to identify and understand the most those ideas from statistical mechanics that are absolutely essential to working with energy-based neural networks. Once we understand them, then (in the next chapter), we can understand how to build learning rule that will give a neural network the capabilities that we want it to have.

While the field of statistical mechanics is vast, broad, and complex, we only need a small slice of it in order to make enormous progress in understanding energy-based neural networks. In fact, there are only three notions that are absolutely essential. We address them in the following section.

10.2 Why We Need Statistical Mechanics

Prior to reading this book, you may have looked at some of the classic papers in deep learning. If you did, you may have realized that the authors were talking in a different language than what you understood; they were using the *language of physics*.

Let's take an example. The following extract is taken from one of the classic papers in the field; Salakhutdinov and Hinton's 2012 work, titled *An efficient learning procedure for deep Boltzmann machines* [1]. This is one of the most important papers in deep learning. Published in

We'll look at a longer excerpt from this same work in a subsequent chapter, right now we just want to identify a key term. For clarity and focus, this author has put the key term in both ***boldface italics*** in the following excerpt:

Extract from Salakhutdinov and Hinton (2012) [1]:

*An undirected graphical model, such as a Boltzmann machine, has an additional, data-independent term in the maximum likelihood gradient. This term is the derivative of the **log partition function**, and unlike the data-dependent term, it has a negative sign. This means that if a variational approximation is used to estimate the data-independent statistics, the resulting gradient will tend to change the parameters to make the approximation worse. This probably explains the lack of success in using variational approximations for learning Boltzmann machines.*

The key term here is ***log partition function***, or more simply and specifically, the ***partition function***.

The notion of a *partition function* is at the very heart and sole of statistical mechanics. If we can understand this, we have an entry point for opening up and understanding the full realm of work in deep learning.

10.3 Simple Units and Their Energies

The fundamental notion of statistical mechanics is that we can create an abstract idea of a system that is composed of many distinct units, and each i^{th} unit has a specific energy, e_i . This notion may seem so abstract as to not have any relevance to anything in the real world, but it turns out to be remarkably useful.

For example, we can model the behavior of gas molecules by thinking of each molecule as simply being a "unit," where each unit also has a corre-

sponding “energy.” In real life, the energy would relate to the temperature of a system. Thus, as we increase the temperature, more and more units individually have greater energy values.

We might think that at a certain temperature, all the gas molecules (“units”) would have the same temperature or energy. In reality, for a given temperature, there is a distribution range of units with various energies. Certainly, as we increase the temperature of a system, more and more units have higher and higher energy values. However, even at very high temperatures, there are some units that have much greater energies than others.

Conversely, even at very low temperatures, some units have a bit more energy than others; while most units would be in the very lowest energy state possible, there will always be a few units that have at least a *some* higher energy. This helps explain why it is very hard to get systems to behave exactly as we’d want at very low temperatures; there are always some units that have enough energy to do something other than what is desired and expected.

The *key factor* underlying all of this is the notion of having a *probability* of energy distributions throughout a system. Because *probabilities* are involved, we’ll always have a range of behaviors, because we’ll always have units dispersed throughout different energy states.

We can determine probabilities for a system; not so much of whether or not an individual unit will have a specific energy, but rather, of whether the whole system will be in a certain state (a microstate) with an overall system energy.

10.4 The energy-based probability equation

In the statistical mechanics universe, the probability function deals with the *probability of finding the system in a certain microstate*.

The key thing to notice in Eqn. 10.1 is the index j . Instead of referring to the number of units in a system, or to the energy levels available in the system, j refers to the *microstate* in which the system finds itself.

Because this is so important, we will shortly discuss microstates.

The (Statistical Mechanics)
Energy-based Probability:

$$p_j = \frac{1}{Z} \exp(-\beta E_j). \quad (10.1)$$

The constant in this equation, β , incorporates both a constant referred to as Boltzmann's constant and the temperature (in degrees Kelvin) of the system. In future work, will simplify this; either by setting β arbitrarily to 1, or by using it as a parameter that we can modify. (That means, we would let the notion of "temperature" modify the notion of "energy." All this, however, is for a later date.)

The partition function, Z (which we'll discuss in the next subsection) will be the **normalizing factor** in the probability equation. By including it, we make sure that the sum of all the probabilities comes to one.

Reading the Probability Equation:

The probability that a system is in a given microstate j is given as the exponent of the *negative* of beta (a constant) times the energy of that state (that is, the energy of the entire system in that microstate), the whole divided by the partition function, Z .

As mentioned at the beginning of this subsection, the probability given in Eqn. 10.1 is not just of any specific unit being in any given energy state. Rather, it deals with the energy associated with the *whole system* being in a given microstate, meaning that different units can be in different energy states, and we need to sum up over all the units and their respective energies.

We see that the probability of finding units in energy state j decreases as the energy E_j increases, because the exponent of a negative number decreases with the size of that number. To visualize this, Fig. 10.1 shows the exponential equation for the *negative* of the variable x .

You might notice that Eqn. 10.1 is non-obvious. It's entirely reasonable to ask a question such as, "where did this exponential term come in?"

To answer that, we'd have to go back further, to the degeneracy equation for the system. We'd look at how we can describe not only the units in the system, but how we can permute them (change their places with each other). Because that is a separate discussion and derivation, I'll ask you to take the previous equation on faith. We'll get a partial answer in the next section, on microstates, but a more complete answer will require more derivations. We'll postpone those until later.

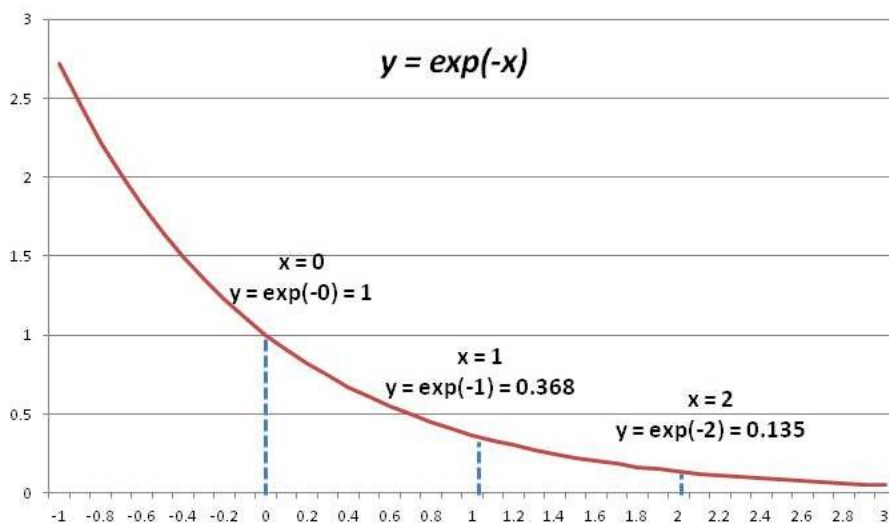


Figure 10.1: The exponential equation for a negative value: $y = \exp(-x)$.

10.5 Microstates

Let's envision a system that has three distinct energy levels, e_0 , e_1 , and e_2 , and that contains ten units, such as is shown in Figure 10.2. In this particular illustration, there are seven units in the lowest energy state, e_0 , two units in e_1 , and one unit in e_2 .

Note that the distribution of units shown in Figure 10.2 is realistic; the likelihood that a unit is in a given energy level will be governed by the value of that energy level, so that the higher the energy, the fewer the units that are in that energy level.

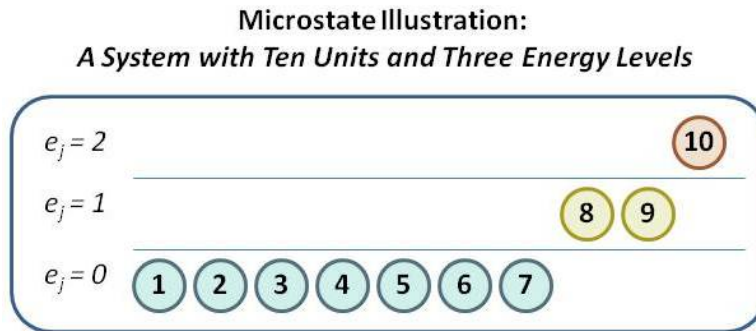


Figure 10.2: A *single microstate* for a system with ten units at three energy levels; seven units where $e = 0$, two units where $e = 1$, and one unit at $e = 2$.

To express the distribution of units more precisely, there will be a greater probability of having microstates with lower overall energies than microstates with higher energies. The way in which a microstate has a lower overall energy is that more units, respectively, are in the lower-energy states. For example, in the three-state system just illustrated, more units are in state e_0 , and fewer are in state e_2 .

We can compute the total energy E_j associated with this microstate, as is shown in Example 10.1.

Example 10.1.

The energy for a given microstate, E_j , is computed by summing the energy for each unit in that microstate. For purposes of this example, let's assume that $e_0 = 0$, $e_1 = 1$, and $e_2 = 2$. For the microstate illustrated in Figure 10.2, we obtain E_j , the total energy for that state, as:

- Seven units at $e_0 = 0$ gives an energy of 0 for that level,
- Two units at $e_1 = 1$, gives an energy of 1 for that level, and
- One unit at $e_2 = 2$, gives an energy of 2 for that level, so that $E_j = 7 * 0 + 2 * 1 + 1 * 2 = 3$.

Clearly, for any given microstate, we can compute the energy E_j associated with that state.

The challenge is that, to compute the probability of a given microstate occurring, we need to normalize the sum of all of our probabilities. That is, for the statistical mechanics probability (as with all probability equations), we have

$$\sum_j p_j = 1.$$

To accomplish this sum, we need to be able to identify all the microstates. To do this, we need to understand exactly what a microstate *is*, and *is not*. This then lets us do that summation over all microstates, and by doing so, we obtain a normalizing function ($1/Z$) which we can then use in determining the probability for any given microstate, as identified in Eqn. 10.1.

To do this, let's take a closer look at the nature of microstates, as illustrated in Figure 10.3.

Two configurations of units in a system are still the same microstate if all that the units do is move about on their (same) respective energy levels. If any two (or more) units swap positions on energy levels, or move to entirely different energy levels, then we have a uniquely different microstate.

Figure 10.3 shows us three different configurations of units within the three-energy-level system that we used earlier in Figure 10.2. Each illustration, (a) - (c), has the same number of units in the different energy levels. Illustration (a) is identical with that given in Figure 10.2; we have simply labeled the different units; 1..10.

In illustration (b), we swap two units that are at the same energy level. This is like having two people, on the same floor in a building, change places with each other. There is nothing substantially different when they do this; this is *not* a different microstate.

In illustration (c), though, we swap two units from different energy levels. This is like having two people exchange the floors that they are on in a building. It *is* a different microstate.

**Determining What IS and IS NOT a Unique Microstate:
A System with Ten Units and Three Energy Levels**

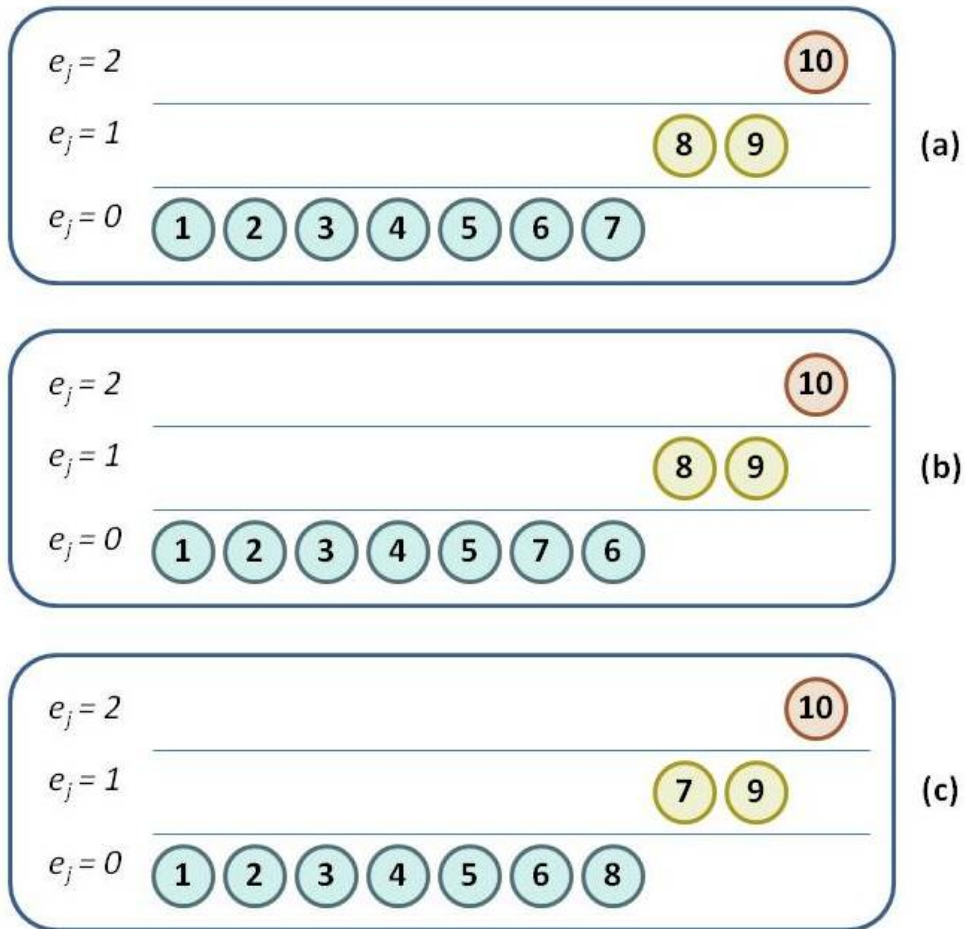


Figure 10.3: Figuring out what IS and IS NOT a uniquely distinct *microstate*: (a) - original illustration of a microstate, for a system with ten units at three energy levels; (b) - units (6) and (7) change position within the same energy level; this is NOT a uniquely different microstate from that shown in (a), so does NOT get counted separately; (c) - units (7) and (8) change positions between energy levels; this DOES count as a uniquely different microstate.

10.6 The Partition Function

Probably the most central equation for statistical mechanics, and thus for machine learning, is the *partition function*. The partition function itself is called Z , from the German word *zusammenfügen*, literally “put together.”

The partition function tells us how a system is *put together* in terms of distinct components.

As mentioned in the introduction to this section, the fundamental notion of statistical mechanics is that a system is composed of many distinct units, and each unit has a specific energy, e_i . That is, the i^{th} unit has an energy, e_i , associated with it.

A couple of other ideas are very intrinsic to statistical mechanics. One is that there are many, *many* units in a system; this allows us to make some approximations during the course of our derivations. Another key notion is that units with the same energy are indistinguishable from each other; we say that they are *degenerate* with regard to each other. This is essential in forming the statistical mechanics equations. We just saw the importance of this in our discussion of microstates.

The partition function is given as

The Partition Function:

$$Z = \sum_j \exp(-\beta E_j). \quad (10.2)$$

The sum here is over all the possible configurations, or microstates, that the system can find itself in. This number gets very big, very fast.

Reading the Partition Function:

The partition function, Z , is the sum, over all the different (j) microstates available, of *the exponent of the negative of a constant times the energy of that microstate*. A microstate is a specific configuration of units; that is, each unit in the system inhabits one of the available energy states.

Example 10.2.

The accompanying slidedeck, *Microstates and Partition Functions: Some Simple Examples*, gives *two complete examples* of how to identify all of the microstates for two systems, each with *very small* set of units and energy levels, and from there, how to compute the partition function for each example.

10.7 The entropy equation

The entropy equation is at the heart of several disciplines; statistical mechanics, information theory, and machine learning.

The (Statistical Mechanics) Energy-based Probability:

$$S = - \sum_j p_j \ln p_j. \quad (10.3)$$

In statistical mechanics, the entropy is represented as S , and in information theory, as H . Usually, the authors will tell us what their variables mean.

The entropy is always of the form given in Eqn. 10.3. Sometimes, though, p_j can get complex and interesting. For almost all the work that we will do (until we encounter more advanced entropy formulas), p_j will have the definition that we gave earlier. Thus, the entropy term is summing over microstates j , not the individual units themselves. As we get to more advanced equations, our interpretation of this equation will deepen.

Reading the Entropy Equation:

The entropy of system is the negative of the sum, over all the possible microstates j , of the probability that units are in that microstate j times the natural logarithm of the probability that units are in that microstate j .

Very often, we will have simplified systems that allow for only two possible energy states. When this happens, we will say that the probability of units (or fraction of the total units) being in one state is x , and then (because the probabilities sum to one, and there are only two energy states), the probability of units (or fraction of units) in the other state is $1 - x$. In this particular case, we will have

$$S = -[x \ln(x) + (1 - x) \ln(1 - x)].$$

In natural systems, the tendency is to find an equilibrium point which minimizes the free energy, which we'll discuss in the next subsection. One aspect of this is maximizing entropy. You may have heard that the *Second Law of Thermodynamics* is that the entropy of an isolated system must increase over time. (By "isolated system," we mean one into which we are not putting any extra energy or materials, which makes this a theoretical, not a practical, realization.)

To better envision how the energy for a two-state system appears, let us first recall how a logarithmic curve looks, as shown in Figure 10.4.

Recall that *the logarithm of a fraction is always a negative number*. Further, in our entropy equation (Eqn. 10.4), both x and $1 - x$ are fractions; each is ≤ 1 . Thus, our entropy expression is the negative of the sum of two terms, and each of these terms is negative, so the overall entropy is positive.

In order to maximize the entropy, we want to maximize the distribution of units among available energy states. Consider the case where we have only two possible energy states, and suppose that our only concern is to maximize the entropy. In this special case, we achieve maximal entropy when we have a symmetric system. We get this when we have a system where the energies of the two states are equal.

The distribution of units between the two states is then *equiprobable*; there are equal numbers of units in state **A** and state **B**. It is in this equiprobable

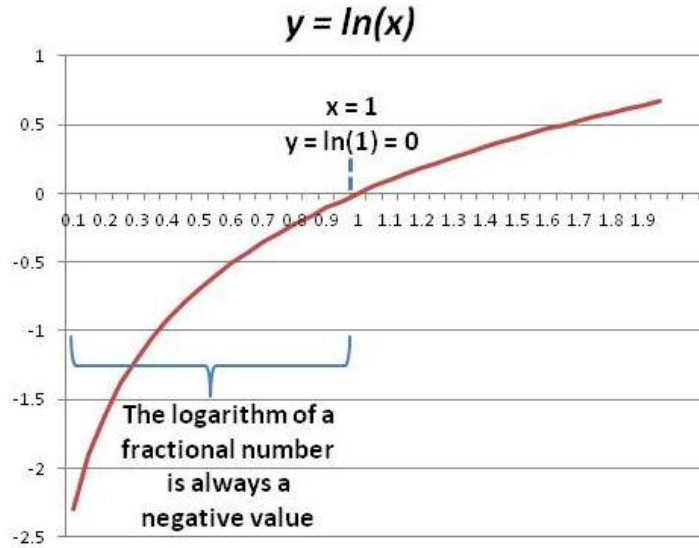


Figure 10.4: The logarithm of x , where x must be greater than zero; the natural log of 1 is 0, and the natural log of all fractions is less than 0.

configuration that we have maximal entropy, or maximal distribution of units among the available states (half in each state).

The entropy for this (ultra-simple) system is shown in Figure 10.5.

The important thing to realize here is: the negative sign in front of the sum in the entropy term is what makes it convex, with a maximum in the middle. If we want to maximize the entropy (and we do), then we need that negative sign in front of the sum. This is what drives the system towards the case where there are equal numbers of units in both states.

If the reverse were true; if there was no negative sign in front of the sum, then the shape of the curve in Figure 10.5 would be U-shaped rather than an upside-down U (which it currently is). If that were to happen, then the “maximal entropy” states would be the two extremes; either all the units would be in state **A**, and none in state **B**, or the reverse. This would be the exact opposite of getting a maximal distribution among all possible states.

Thus, we remember to keep the minus sign in front of the summation in the entropy term, and we know why it is there.

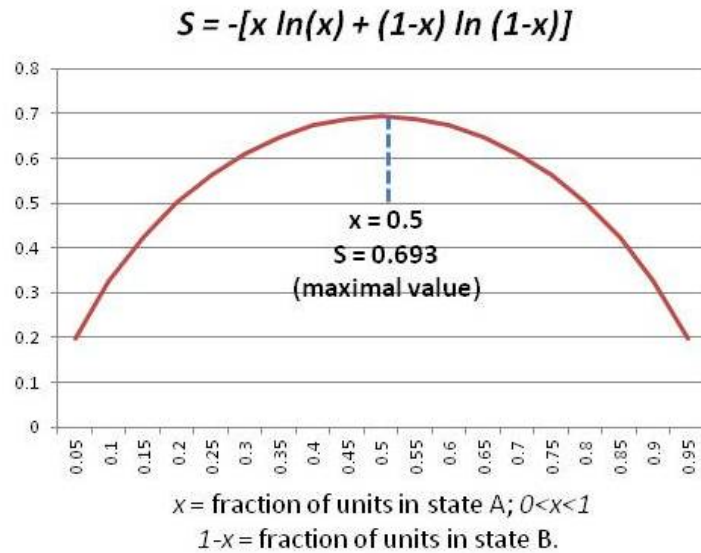


Figure 10.5: The entropy for a system where there are only two states, **A** and **B**, where the fraction of units in each state is given by x and $1 - x$, respectively. The entropy is maximal when $x = (1 - x) = 0.5$.

10.8 The free energy equation

The notion of free energy, which underlies a great deal of thermodynamics, is also important in neural networks, particularly those used in deep learning. Early neural networks, such as the Hopfield network and the Boltzmann machine, relied on a free energy minimization approach.

Even if we ignored these two networks (the Hopfield and the simple Boltzmann machine), we can't ignore the restricted Boltzmann machine (RBM), which is the heart and soul of deep learning. Because the RBM is a simplification (a *restriction*) on the simple Boltzmann machine, if we understand the simple Boltzmann machine and its predecessor, the Hopfield neural network, then we have a solid basis for understanding RBMs. Further, we'll have put down a strong foundation for understanding deep neural networks, which are created by stacking RBMs.

This section gives an altogether too-brief overview of free energy. We'll give this topic the respect that it deserves in the following chapter. Our goal, for now, is simply to *recognize a free energy equation when we see*

one.

As we'll observe, this is not as simple as it seems.

One challenge is *that the free energy equation shows up in two remarkably different forms*. It's a lot like recognizing that a specific caterpillar corresponds with a certain specific butterfly. They look very different, but they are two expressions of the same creature.

A second challenge is that there is a *range of notation used for free energy*. As we observed with entropy, the notation could commonly be either an S or an H . For free energy, the notation can be F (for free energy, not surprisingly), or H or A (for Helmholtz free energy), or G (for Gibbs free energy). These distinctions make a great deal of difference in the world of physical chemistry, where pressure and volume come into play.

In the realm of machine learning, there is no change in either the pressure or the volume of a system; these concepts are not relevant. Thus, terms involving changes in those variables drop out of the free energy equation, and the reader may find authors referring to a (simple) free energy, or to Helmholtz or Gibbs free energies, and all meaning the same thing.

Further, once the notion of free energy is fairly well understood, there are a number of specific models that are common and well-known to physicists. These include the Ising model, together with its variants. One might read, for example, about something like a Bethé-Peierls or mean-field model. We will ignore these variants in this treatment, and concentrate on the basics.

It is possible to derive the second form of the free energy equation (with a little calculus and elbow grease) from the first. For now, I ask you to take on faith that the two following equations mean the same thing.

The first formulation gives the free energy in terms of the energy of each unit, as encapsulated in the partition function.

The Free Energy - first version (as
logarithm of the partition function:

$$F = -k_{\beta}T \ln(Z).$$

The most common way in which we see the free energy introduced in a paper uses the expression just given, so that the free energy involves the partition function.

Reading the First Version of the Free Energy Equation:

The free energy is the negative of a constant (Boltzmann's constant times temperature) times the natural logarithm of the partition function, Z .

There is an entirely different way of expressing free energy; as the difference between the enthalpy (or chemical potential, or ability to do work) minus the temperature times the entropy.

This is a crucial equation as we start using (free) energy minimization methods in machine learning. It means that the equilibrium, or minimal free energy state, is reached as a balance between getting the lowest possible energy (enthalpy) values while still maximizing entropy. This is a trade-off.

Eqn. 10.4 gives the second form of the free energy equation as

The Free Energy - second version (as the difference between enthalpy and temperature times entropy):

$$F = U - TS, \quad (10.4)$$

where U is the chemical potential, T is temperature and S is the entropy.

Reading the Second Version of the Free Energy Equation:

The free energy the difference between the chemical potential, U , and the temperature T times the entropy, S .

U , as mentioned previously, is the chemical potential, often defined as the ability of the system to absorb or release energy during chemical reactions. It can include a number of factors, most significantly (for our purposes) the enthalpy, which is the energy associated with each unit. Since the "other

factors” do not come into play in machine learning, various authors may use either the term chemical potential or enthalpy. They may use U or E or H to express these terms, although U is generally reserved for the chemical potential, while E or H more commonly refer to the enthalpy.

As mentioned earlier, various letters are used for different terms, depending on the author’s whim and provenance, in the free energy equation. Thus, we could see the Eqn. 10.4 show up as $G = H - TS$ or even $A = U - TH$, if the author wanted to be particularly confusing and substitute H (the information-theory way of expressing entropy) for S (the physical chemist’s way of expressing entropy).

While the authors will usually define their terms, they occasionally leave interpretation up to their reader. Then, we have to infer what they mean from context.

The chemical potential (and often the enthalpy) of a system:

$$U = \langle e_i \rangle + \langle e_{ij} \rangle.$$

The chemical potential (and/or enthalpy) is often given as the sum of two terms; one expressing the expected energy associated with each individual unit, and the other expressing the energy associated with pairwise interactions between the units.

Reading the Enthalpy Term:

The chemical potential (and/or enthalpy) is the sum of the expected energy per unit, e_i , together with the pairwise interaction between units, $e_{i,j}$ (Note that we are changing the meaning of the subscript j here; it now refers to another unit in the system, and not to a distinct microstate. This is to make it easier to read the next example, which quotes from one of John Hopfield’s papers.

John Hopfield’s 1982 work presented a new idea in neural networks, using an energy function to describe the state of the network.

Extract from J. Hopfield (1982), Neural networks and physical systems with emergent collective computational abilities

Studies of the collective behaviors of the model

The model has stable limit points. Consider the special case $T_{ij} = T_{ji}$, and define

$$E = -\frac{1}{2} \sum_{i \neq j} \sum T_{ij} V_i V_j . \quad [7]$$

ΔE due to ΔV_i is given by

$$\Delta E = -\Delta V_i \sum_{j \neq i} T_{ij} V_j . \quad [8]$$

Thus, the algorithm for altering V_i causes E to be a monotonically decreasing function. State changes will continue until a least (local) E is reached. This case is isomorphic with an Ising model. T_{ij} provides the role of the exchange coupling, and there is also an external local field at each site. When T_{ij} is symmetric but has a random character (the spin glass) there are known to be many (locally) stable states (29).

Figure 10.6: Extract from John Hopfield's 1982 paper on "emergent collective computation."

Example 10.3.

Suppose that you were to read John Hopfield’s original paper, introducing what we now call the Hopfield neural network [2]. Figure 10.6 gives an extract from this paper. Reading this, we would note the equation

$$E = -\frac{1}{2} \sum_{i \neq j} \sum_j T_{i,j} V_i V_j$$

We looked at this equation in the previous chapter. At that point, though, we were just using this energy equation to help us understand what the *structure* of the neural network had to be. We were constructing an isomorphism between the structure as implicitly expressed in the equation, and the structure that we would create and train with different patterns.

Now, our purpose in looking at this equation is different. We’re noting the correlation between an expression that defines how a neural network works and its motivating source in statistical mechanics.

We’ve already identified (in the previous chapter) that the two units, V_i and V_j , correspond to nodes in a neural network. We previously identified that these were defined to be *bistate* units; meaning that they could be in one of two states, and we gave those two states the values of “0” or “1.”

Now, though, we’re paying attention to the other term in this equation; $T_{i,j}$, which refers to the interaction energy between the two units V_i and V_j .

If we’re familiar with statistical mechanics, then as soon as we see this equation, we know that we’re dealing with a free energy approach. We further know that we’re focusing on the *interaction energy* (more properly, the *interaction enthalpy*) of the system.

Hopfield’s next equation gives us an expression for ΔE , which tells us how the energy changes over time. He specifically says that ΔE is a “monotonically decreasing function,” meaning that the energy of the system is always either decreasing or holding steady; it never increases. This tells us that we’re dealing with a free energy minimization approach.

When he further says that “the case is isomorphic with an Ising model,” he’s saying that we’re very similar in our method to one of the classic models in statistical mechanics, where units can be either “on” or “off” (as with many neural networks), and that there is a prescribed energy-of-activation for the “on” units, and there is also an interaction energy between units.

Even if we knew nothing more about statistical mechanics and the Hopfield neural network than what we’ve read in this so far, we’d now know that the

Hopfield neural network lives smack in the middle of the statistical mechanics universe, and that it is trained using a (free) energy minimization method. We'd also know that this neural network, even if not popular today (due to memory constraints for storing patterns in this network), is part-and-parcel of the world of neural networks and machine learning methods that use energy minimization.

We've given the most minimal and superficial attention to the notion of free energy. This concept has been foundational to modern neural networks (e.g., the Hopfield network), and continues with even broader scope and implications today, as it plays a key role in multiple machine learning methods.

We'll go into more detail in the next chapter. Our goal is to move from the statistical mechanics fundamental concept of free energy to a useful neural network learning rule; specifically to the learning method called *contrastive divergence*.

Bibliography

- [1] R. Salakhutdinov and G. Hinton, “An efficient learning procedure for deep Boltzmann machines,” *Neural Computation*, vol. 24, pp. 1967–2006, August 2012.
- [2] J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 79, pp. 2554–2558, April 1982.