Statistical Mechanics, Neural Networks, and Artificial Intelligence

A Short Summary of
Seven Crucial Machine Learning Equations

DRAFT
Précis

Alianna J. Maren
Themesis, Inc.

Draft Date: 2024-01-10
Version 1.2

1.1 Introduction and Overview

Generative artificial intelligence (AI) methods, along with variational methods and other machine learning methods, both lie at the confluence of statistical mechanics, probability theory, and neural networks, as shown in Figure 1.1.

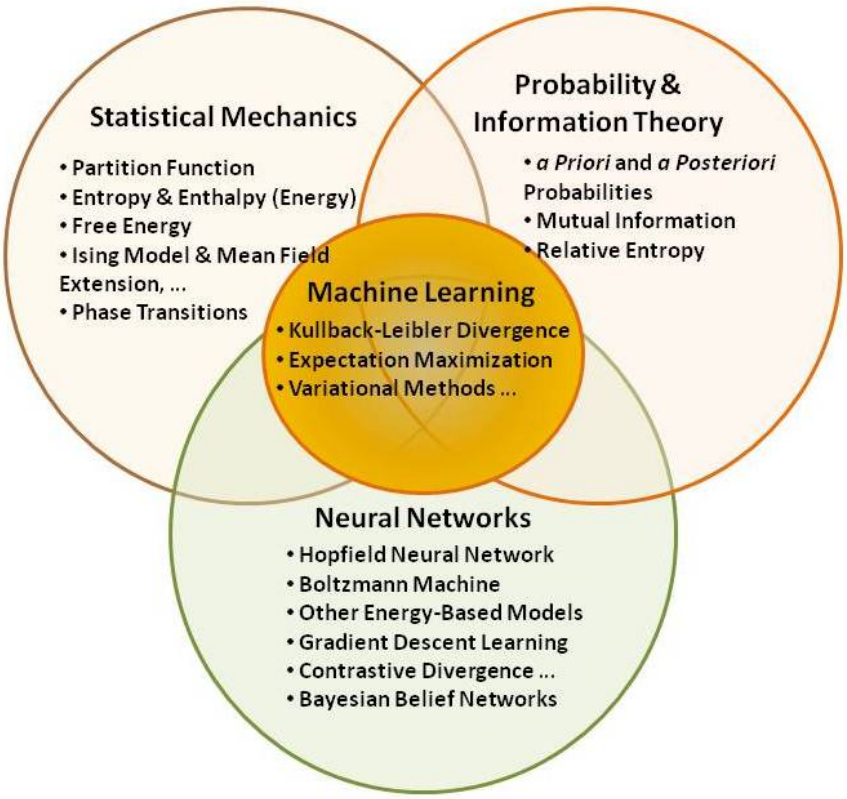

Figure 1.1: The ***gold zone***: essential generative AI methods as well as machine learning algorithms draw from statistical mechanics, probability and information theory, and as well as neural networks fundamentals.

The Seven Key Equations

This short Précis covers ***seven of the most important equations in artificial intelligence and machine learning***, as shown in Figure 1.2.

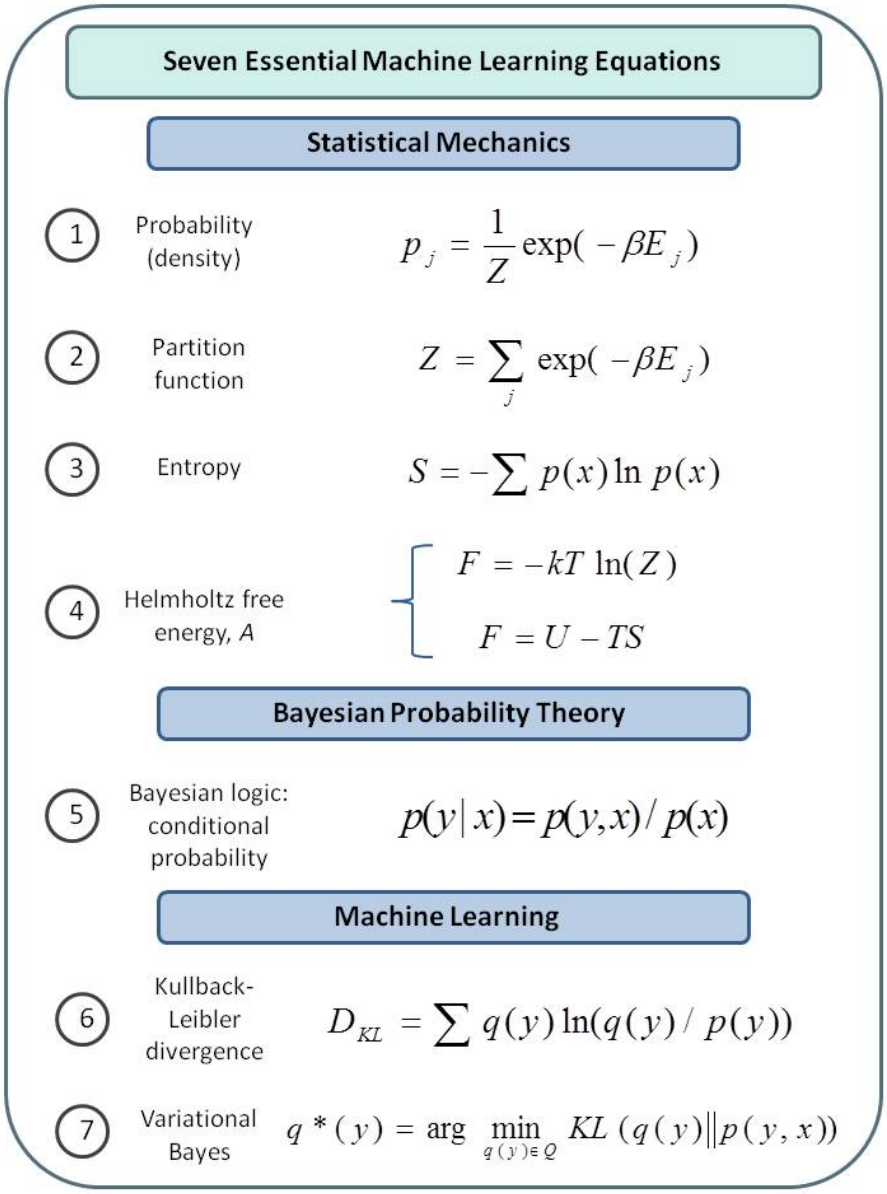

Figure 1.2: Seven key equations in artificial intelligence and machine learning include four from statistical mechanics, one from Bayesian-based probability theory, and two that are central to machine learning itself.

Four of these equations come from statistical mechanics, and one comes from Bayesian probability theory. The sixth equation, the *Kullback-Leibler divergence*, is essential to both energy-based neural networks as well as to variational methods. The final (seventh) equation identifies how variational methods seek to minimize the Kullback-Leibler (K-L) divergence in order to find a most suitable model for a given set of data.

This Précis will not attempt to derive any of them; the purpose is more to name them, and to let you read them (in plain English), and understand what they mean. The derivations, and how to work with and use these equations, will be in the forthcoming book, *Statistical Mechanics, Neural Networks, and Artificial Intelligence*.

1.2 Statistical Mechanics Equations

The fundamental notion of statistical mechanics is that a system is composed of many distinct units, and each i^{th} unit has a specific energy, e_i. 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.

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. 1.1 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:

$$p_j = \frac{1}{Z} \exp(-\beta E_j). \tag{1.1}$$

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. 1.1 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 is decreases with the size of that number. To visualize this, Figure 1.3 shows the exponential equation for the *negative* of the variable x.

You might notice that Eqn. 1.1 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 subsection, on microstates, but a more complete answer will require more derivations than we want in this (relatively short) Précis.

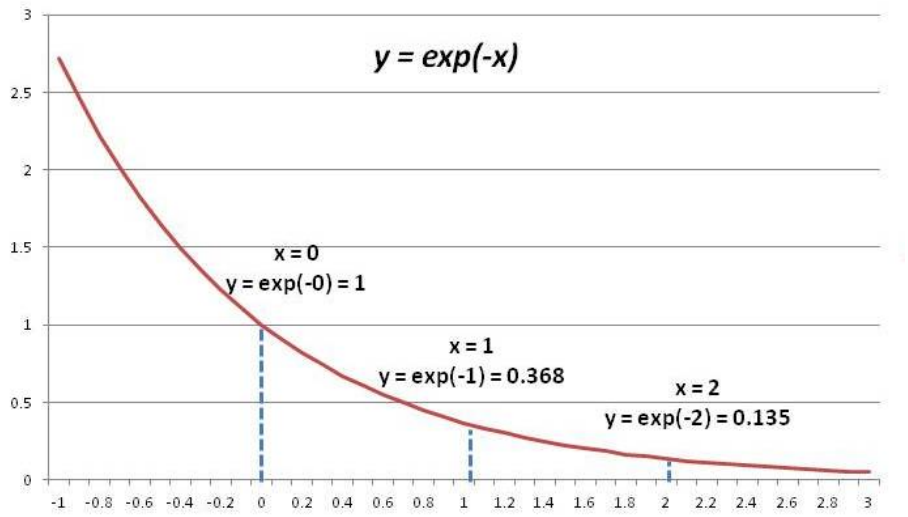

Figure 1.3: The exponential equation for a negative value: y = exp(-x).

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 1.4. 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 1.4 is realistic; the likelihood that a unit is in a given energy level will governed by the value of that energy level, so that the higher the energy, the fewer the units that are in that energy level.

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 1.1.

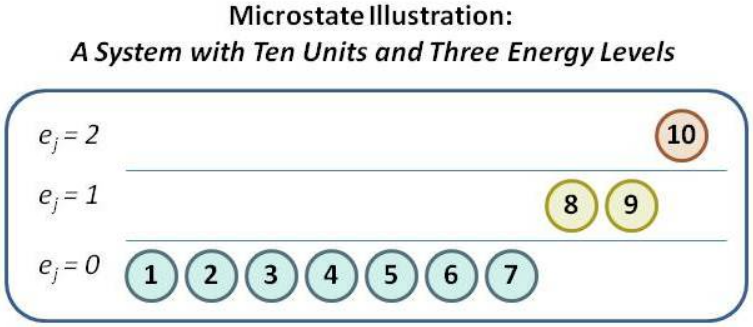

Figure 1.4: A ***single microstate*** for a system with ten units at three energy levels; seven units where $e = 0$, two units where $e = 1$, and one unit at $e = 2$.

Example 1.1.

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 1.4, 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. 1.1.

To do this, let's take a closer look at the nature of microstates, as illustrated in Figure 1.5.

> 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 1.5 shows us three different configurations of units within the three-energy-level system that we used earlier in Figure 1.4. Each illustration, (a) - (c), has the same number of units in the different energy levels. Illustration (a) is identical with that given in Figure 1.4; 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.

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 *zusammanfügen*, 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,

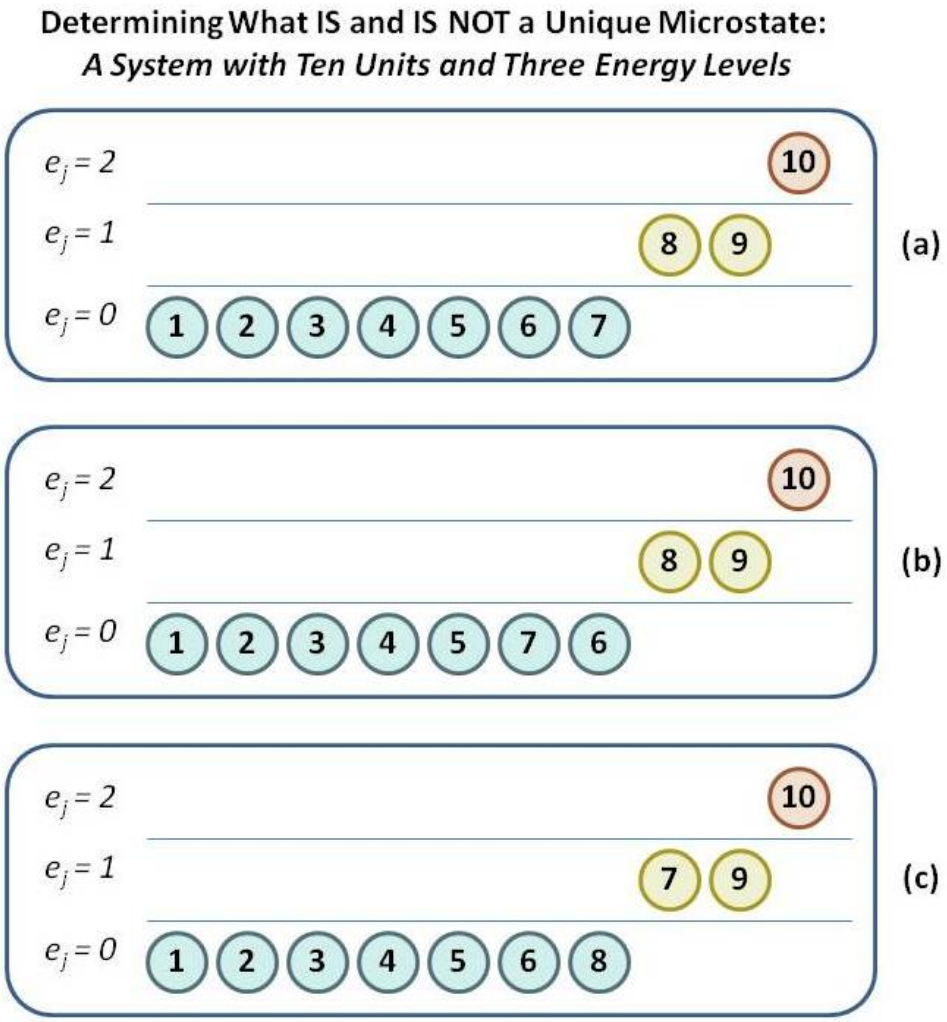

Figure 1.5: Figuring out what IS and IS NOT a uniquely distinct *microstate*: (a) - original illustration of a microstate, for a system with ten units at three energy levels; (b) - units (6) and (7) change position within the same energy level; this is NOT a uniquely different microstate from that shown 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.

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:

$$Z = \sum_j \exp(-\beta E_j). \tag{1.2}$$

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 1.2.

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.

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:

$$S = -\sum_j p_j \ln p_j. \tag{1.3}$$

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. 1.3. 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 1.6.

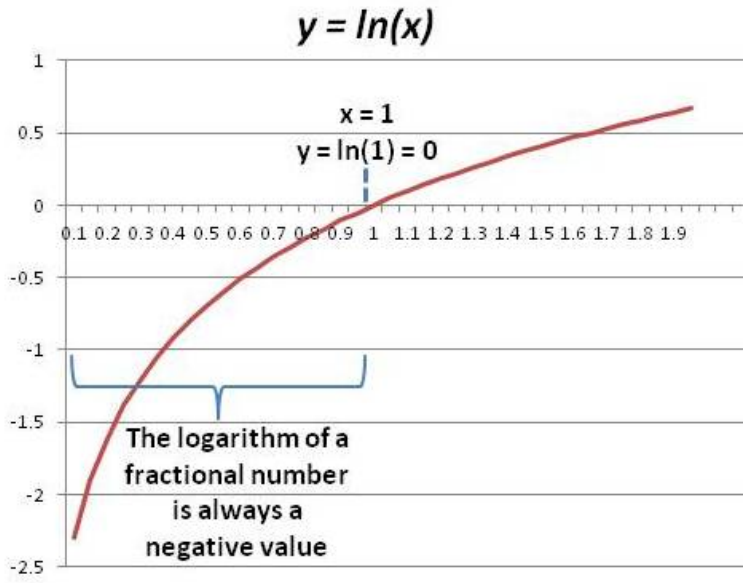

Figure 1.6: The logarithm of x, where x must be greater than zero; the natural log of 1 is 0, and the natural log of all fractions is less than 0.

Recall that *the logarithm of a fraction is always a negative number*. Further, in our entropy equation (Eqn. 1.4), 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 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 1.7.

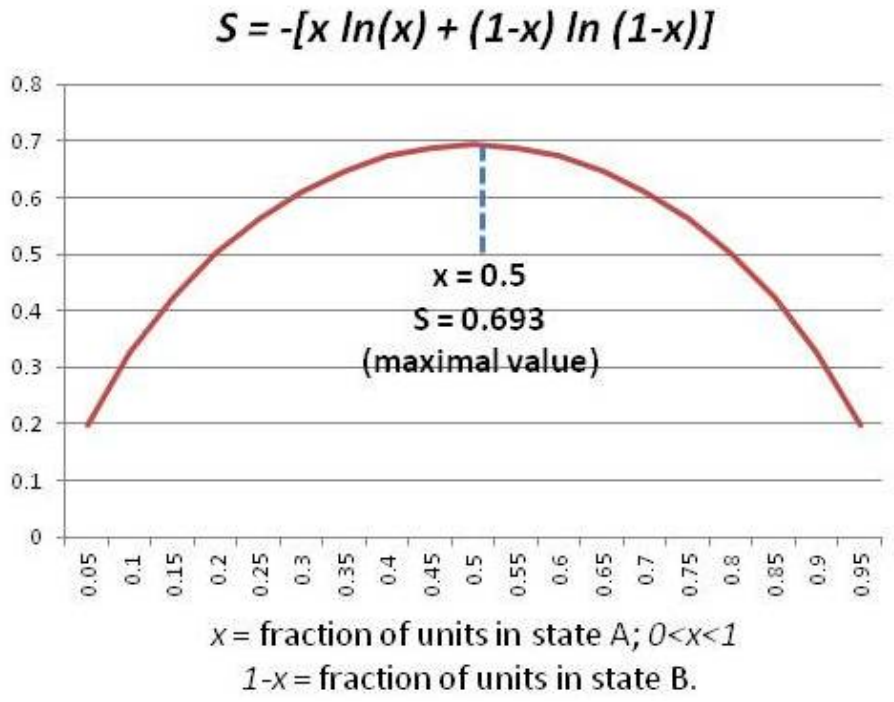

Figure 1.7: The entropy for a system where there are only two states, **A** and **B**, where the fraction of units in each state is given by x and $1 - x$, respectively. The entropy is maximal when $x = (1 - x) = 0.5$.

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 1.7 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.

The free energy equation

The notion of free energy, which underlies a great deal of thermodynamics, is also important in energy-based neural networks (e.g., the Boltzmann machine), as well as variational inference, which is an advanced method in machine learning. Early neural networks, such as the Hopfield network and the Boltmann machine (trained using simulated annealing), relied on a free energy minimization approach.

In fact, free energy minimization has become almost a unifying philosophical theme throughout the realms of energy-based neural networks and variational methods.

This subsection gives an altogether too-brief overview of free energy. We'll give this topic the respect that it deserves in the book-in-progress, *Statistical Mechanics, Neural Networks, and Artificial Intelligence*. 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 Précis, 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 artificial intelligence and 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. 1.4 gives the second form of the free energy equation as

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

$$F = U - TS, \tag{1.4}$$

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. 1.4 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.

Example 1.3.

Suppose that you were to read John Hopfield's original paper, introducing what we now call the Hopfield neural network [1]. Figure 1.8 gives an extract from this paper. Reading this, we would note the equation

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

In this equation, $T_{i,j}$ refers to the interaction energy between two units, V_i and V_j, and these units can have values of "0" or "1."

This tells us that we're dealing with a free energy approach, and that we've just learned something about 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.

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

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

ΔE due to ΔV_i is given by

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

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 1.8: Extract from John Hopfield's paper, "Neural networks and physical systems with emergent collective computational abilities;" one of the earliest papers in modern neural networks.

Hopfield 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 Précis, 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, the Boltzmann machine, and all derivatives and descendents of the Boltzmann machine). It continues with even broader scope and implications today, as it plays a key role in multiple machine learning methods. The book in progress, *Statistical Mechanics, Neural Networks, and Artificial Intelligence*, will cover this topic in much greater detail, and trace its use from early neural networks through current deployment in more recent neural networks as well as machine learning theory and applications.

We now turn our attention to an entirely different form of logic and reasoning, that associated with Bayesian probabilities.

1.3 Bayesian Probability Equations

Bayesian probability theory allows us to express information and ask questions of a "something depends on something else" nature. For example, we can ask ourselves what the likelihood is of rain on an August day in a certain city, or whether our favorite sports team will win their next game.

The Bayesian *a posteriori* probability is given as

The Bayesian *a Posteriori* Probability:

$$p(y|x) = p(y,x)/p(x). \tag{1.5}$$

This *a posteriori* Bayes equation expresses the likelihood of seeing one event (y), dependent on - or *conditional* upon - seeing another event (x).

Reading the *a Posteriori* Probability Equation:

The probability of observing a certain value for y (the dependent variable), conditioned (or dependent upon) the independent variable x is equal to the *joint probability distribution* of y and x, divided by the probability of x. The vertical slash in $p(y|x)$ means that "y is conditioned (or dependent on) x." The comma in $p(y, x)$ means that we look at the two distributions independently; this is called the *joint probability distribution.*

As an example of a simple probability distribution, Figure 1.9 shows the (Gaussian or normal) distribution of the height of males in the United States.

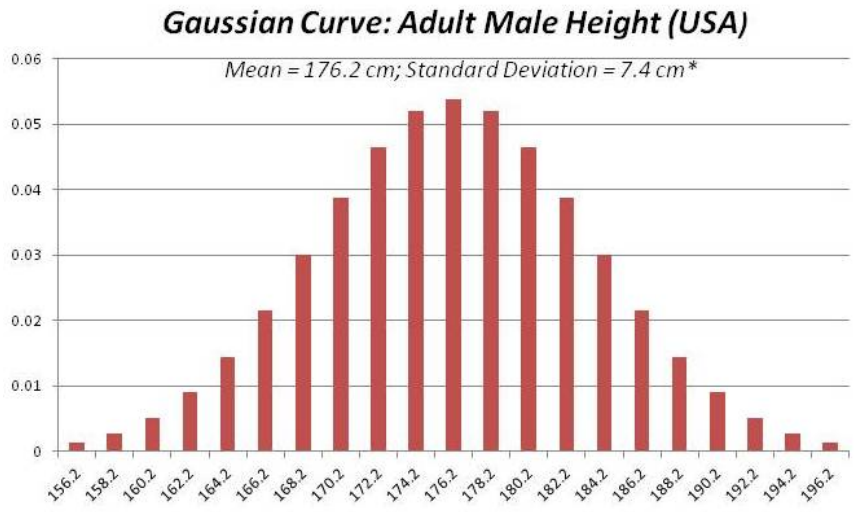

Figure 1.9: The height of males in the United States follows a Gaussian (normal) distribution. *Data sources: See [2] for height data and [3] for standard deviation.

From Figure 1.9, we could create a simple probability statement, such as $p(height)$, meaning "the probability that a man has a height within a given range."

To use the more complex conditional dependency of $p(y|x)$, we would be asking about the probability of one event conditioned or dependent on another. For example, if we had the probability distributions of heights of *both* men and women, then – if we were discussing a person of a given height – we could ask about the probabilty $p(y)$ of the gender of the person (male or female) depending on the given datum of their height; this would be a $p(gender|height)$ conditional dependence.

Of course, the conclusiveness of our probability depends on the value of the independent, observed variable (height). If a person is 6 feet tall (183 cm), then the person is much more likely to be male than female. If the person is 5 feet, 7 inches (170 cm), the answer will be more equivocal.

One way in which Bayesian probability theory can handle these types of questions is to add more independent variables; that is, make x more complex. (Or, alternatively, we can think of this as $p(y|x, w, v, ...)$). For example, we could look at Body Mass Index (BMI), which is a bit less dependent on height than is weight [4]. (We want independent variables, to the extent possible.) So long as we can obtain independent measurements that cover these extra variables, we're fine.

The challenge in working with Bayesian probabilities is that – while the theory is elegant – putting the theory into practice requires that we must specify the distributions of all the variables that we're considering. This usually involves gathering (collecting, analyzing, and reporting) data, which usually involves expense – if the task can be done at all.

Nevertheless, Bayesian probabilities are much in demand for a great variety of tasks, ranging from predicting a person's shopping behaviors to whether or not the blob of pixels just detected by an autonomous vehicle's cameras are a paper bag blowing in the wind, or a child running out between from behind a parked car.

Efforts such as the latter have led all the major automobile manufacturers to conduct extensive training regimes of their vehicles in both isolated and controlled enclaves as well as actual (human-use) driving environments. Even so, there will inevitably be situations that can not be anticipated.

Because we will never be able to specify all the situations that can lead to probabilistic decision-making, there is a lot of attention within machine learning circles on *inferring* the nature of a new situation. *Inference*, regarded as a key element of machine learning, takes estimation to a new level. This will be one of the most important topics in the book (in progress), *Statistical Mechanics, Neural Networks, and Artificial Intelligence*.

1.4 Machine Learning Equations

Machine learning is a hybrid discipline drawing from multiple sources, most notably statistical mechanics and Bayesian probability theory. This is not only a very rich topic, it is growing very rapidly. The following two subsections briefly address two of the most pertinent machine learning equations; the Kullback-Leibler divergence and the variational Bayes approximation.

The Kullback-Leibler Divergence

The Kullback-Leibler divergence is given as

The Kullback-Leibler Divergence:

$$D_{KL}(q||p) = \sum q(y) ln[q(y)/p(y)]. \qquad (1.6)$$

The Kullback-Leibler divergence, often written as D_{KL}, is a difference measure. It represents how well a model approximates a data distribution, and is one of the most widely-used equations in machine learning and information theory.

While this formulation considers q to be the model and y to be the actual data, the D_{KL} can be used to compare any pair of models or data distributions.

The D_{KL} measure is *not symmetric*; that is $D_{KL}(q||p) \neq D_{KL}(p||q)$.

Reading the Kullback-Leibler Divergence Equation:

The ***Kullback-Leibler divergence*** is the *expectation of the logarithmic difference* ("log-difference") between *two probability distributions*. It is the sum (or integral) of the model value, $q(y)$ (for a certain histogram bin) times the logarithm of the model value over the actual observed data. The double parallel lines (||) are a notation invented for difference (divergence) measures; so that $(q||p)$ is read as the *divergence* of q with regard to p.

The Kullback-Leibler divergence is one of the most popular and useful measures within machine learning. It is also a component of more extensive methods, such as variational approximations, discussed next.

Variational Bayesian Approximation

One of the most essential machine learning tasks is inference. Inference is one step beyond learning. In learning, we can train a neural network (or deep learning system) based on training data; data for which the "right answers" have been predetermined. (This is sidestepping the issue of generative systems, where the systems figure out the data distributions on their own.)

In inference, the task is to make the best approximate judgment when given a situation that has not previously been learned.

One of the most valuable machine learning inference equations is the variational Bayes approximation, which is actually a family of methods. One formulation for a variational Bayes equation is given as

The Variational Bayes Approximation:

$$q^*(y) = arg \min_{q(y)\epsilon Q} KL[q(y)||p(y,x)]. \qquad (1.7)$$

We read this variational Bayes equation as follows

Reading the Variational Bayes Approximation:

The specific model, $q^*(y)$, that *minimizes the Kullback-Leibler divergence* between the model q and the observed data p is one that is selected from a family of possible models Q, where q is *an element of* (or is a member of) Q, or $q(y)\epsilon Q$. It is the one whose *arguments* (parameters) minimize the K-L divergence.

Variational methods, both Bayesian and other, are at the center of many machine learning approaches. They will be discussed at length in the book-in-progress, *Statistical Mechanics, Neural Networks, and Artificial Intelligence*.

1.5 Summary of the Seven Equations

The seven equations identified here form the cornerstone of energy-based machine learning. They do not comprise all of the equations. In fact, they are somewhat like defining mountains in a mountain range. There are many equations and methods that will interest us, such as *contrastive divergence* (for the Boltzmann machine), *expectation maximization* and *energy minimization*, and others that are very important.

Nevertheless, with these seven equations in hand, you can situate yourself with regard to what you are reading in machine learning. That means, for example, that if you come across the Kullback-Leibler divergence, you not only know how to read the equation, but you know that the material is about measuring the goodness of a model with regard to some data.

This short Précis has very lightly overviewed material that will be included in the forthcoming book, *Statistical Mechanics for Neural Networks and Machine Learning*. Until the book is published, various draft chapters as well as relevant blogposts will be available at: **www.aliannajmaren.com**.

Author's Note: I combed through a large number of resources in preparing this Précis. The ones that I found most valuable, in terms of giving useful, clear, and simple explanations include the book written by Dr. David Tong [5], along with a tutorial by Denker [6].

Bibliography

[1] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc Natl Acad Sci USA*, vol. 79, p. 2554–2558, April 1982.

[2] C. Fryar, Q. Gu, C. Ogden, and K. Flegal, "Anthropometric reference data for children and adults: United states, 2011–2014," Tech. Rep. 3, 39, National Center for Health Statistics. Vital Health Stat., 2016.

[3] A. Gelman and D. Nolan, *Teaching Statistics: A Bag of Tricks*. Cambridge, UK: Oxford University Press, 1st ed., 2002.

[4] D. R. McCreary, "Gender and age differences in the relationship between body mass index and perceived weight: Exploring the paradox," *International Journal of Men's Health*, vol. 1, pp. 31–42, January 2002.

[5] D. Tong, *Statistical Physics*. 2012.

[6] J. Denker, *Modern Thermodynamics*. CreateSpace, 2014.