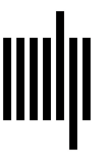
Gradient Expectations

Structure, Origins, and Synthesis of Predictive Neural Networks

By: Keith L. Downing

Citation:

Gradient Expectations: Structure, Origins, and Synthesis of Predictive Neural Networks
By: Keith L. Downing
DOI: 10.7551/mitpress/14723.001.0001
ISBN (electronic): 9780262374675
Publisher: The MIT Press
Published: 2023



The MIT Press

2 Conceptual Foundations of Prediction

Prediction is a catchy buzzword pertaining to speculation about the future, but many of its uses lack a temporal component and involve little more than basic causal inference. When my son comes home bloodied and bruised from his daily mountain-bike ride, I can predict that *the trail won today*. This hints of temporal lookahead, because my current state of knowledge, when he comes in the door, has not yet been updated to include the cause of his battered appearance. I am thus predicting what my son will soon tell me: he suffered multiple painful wipeouts. In short, my hypothesis about a past cause constitutes a prediction with respect to my future knowledge state. Subjectively, my causal hypothesis involves the future, not the past.

Similarly, in machine learning, a deep network trained to perform facial recognition of the citizens of Land-O-Plenty may be said to *predict* that the suspicious midnight visitor to a local ATM was Robin Green, based on a brief and blurry video sequence, when more accurate verbs include *estimate*, *speculate*, or *guess*, none of which suggest temporality. However the word *predict* might seem justified under the assumption that the authorities will eventually discern the true identity of the culprit, sometime in the future, thereby verifying or refuting my prediction.

In providing a basic conceptual backdrop for the remainder of the book, this chapter sticks as closely as possible to the definition of prediction found in *Webster's Dictionary*: "to declare or indicate in advance." However, there will always be the lingering question: In advance of *what*? Are we discussing the occurrence of an event on an objective timeline, or the awareness of that event by a particular agent, or the formation of a representation of that event by neural firings in a particular brain region of that agent? The philosophical slippery slopes are unavoidable, but I will do my best to avoid sliding too far down any of them.

In general, our deep investigation of prediction in neural systems will entail a rather subjective view. Signals can take hundreds of milliseconds to fully register in human sensory apparatus, and tens of milliseconds to travel between neurons. Hence, the prediction by a neural assembly of what it (or another assembly) will experience in the future will often be the consequence of some world event of the recent past (that the nervous system is gradually processing). So neurons predict what (possibly other) neurons will *see* or represent in the near future about a past event. Temporality is essential to all of this, but, unfortunately, often a bit confusing.

2.1 Compare and Err

My high school coach had a slogan taped to his dashboard: *Once I thought I was wrong, but I was mistaken.* Predictions are frequently mistaken, but the predictor need not be inexorably wrong. Typically, prediction is not a single act, but an adaptive process by which wrongs gradually get righted. In the terminology of control theory, predictors often operate in a closed-loop mode in which they make a prediction, compare it to a target to produce an error, and then use the error as the basis for an updated prediction, which then leads to another error and another prediction.

Letting P denote the prediction, E the error, and R the target / reality, a simple predictor algorithm for a perpetually active agent is

1. Initialize P
2. Input R
3. $\mathrm{E} = \mathrm{R} - \mathrm{P}$
4. $\mathrm{P} = \Omega(\mathrm{E}, \mathrm{P})$
5. Go to 2

The function Ω may be as simple as $\Omega(x, y) = x + y$: it just adds the error to the predicted value to yield R, since $\mathrm{P} + \mathrm{E} = \mathrm{P} + (\mathrm{R} - \mathrm{P}) = \mathrm{R}$. That works fine as long as R changes very little between timesteps, which depends on the operational timescales of the environment and predicting agent. In complex situations, however, even for a reasonably static R, the intricacies of the agent's decision-making apparatus (e.g., a neural network) and of the agent-environment coupling may preclude the simple design of Ω or straightforward calculation of E. Both the comparison and the update may involve noise or other forms of stochasticity. Hence, many iterations through the loop may be necessary to gradually bring E's magnitude down to an acceptable level.

Some of these confounding factors and more advanced control loops appear later in this chapter and book. The main lessons for now are that (a) prediction is typically a closed-loop process, that (b) requires a comparison of the expectation to a target to produce an error, which (c) affects future predictions.

2.2 Guesses and Goals

Sticking as close as possible to the temporal characterization of prediction, it involves conjuring up (or at least behaving *as if* one has conjured up) the future state of some system. Our general conception of a prediction is thus a forecast (or guess) of a future state, whether in the next millisecond or the next century. However, few species *make a living* solely from predicting the future; they need to *act* in a manner that takes advantage of those predictions. Even among humans, those who earn money from forecasting do so because somebody exploits that information to better perform some activity, and is thus willing to pay for it. In short, a prediction isn't worth much if it doesn't enhance survival in some way or another.

Imagine a cheetah chasing a gazelle across the savanna. It may *predict* that the gazelle is trying to get to a wooded area, where it might have a better chance of evading its pursuer, but it will also *desire* to keep it in the open. Thus, the cheetah (implicitly or explicitly)

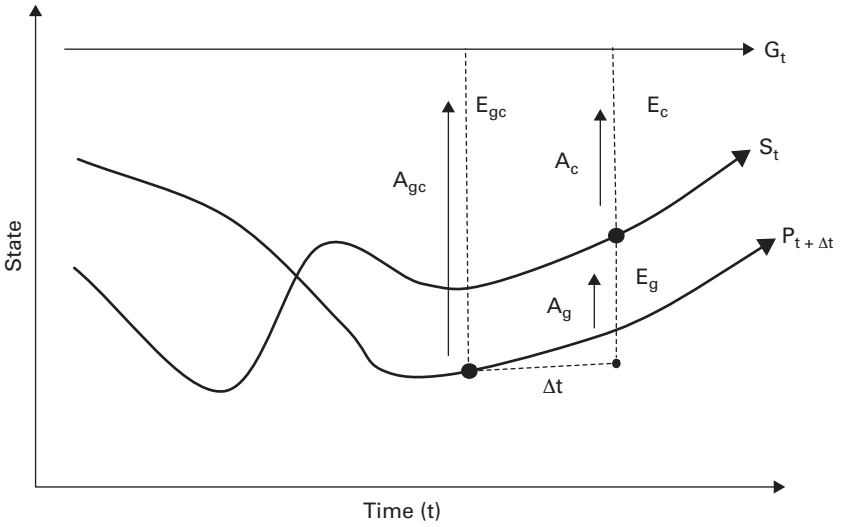


Figure 2.1
Dual aspects of prediction: guessing versus control. The three plots: S_t = state of system at time t, $P_{t+\triangle t}$ = prediction / guess at time t of the system state at time $t + \triangle t$, and G_t = goal state at time t. The simple errors pertain to control, E_c, and guessing, E_g, and the actions aimed at reducing those errors are A_c and A_g, respectively. The more complex error, E_{gc}, is between the goal and the predicted future state, while action A_{gc} incorporates that prediction in pursuit of the goal.

manages two alternate realities, *gazelle in woods* and *gazelle in wide-open space*, while dealing with the current state of the world. We can call these alternate realities the *guess* and *goal*, respectively. In their simplest form, all of these (world) states include two key elements: the locations of the cheetah and the gazelle; more elaborate versions would include the direction and magnitude of their velocities and accelerations, energy levels, signs of weakness, and so on.

As illustrated in figure 2.1, the goal state (G_t) can be assumed constant for the time frame of our analysis, while the current system state (S_t) changes, as does the guess of the future state $P_{t+\triangle t}$: at time t, the agent (cheetah) predicts the world state at time $t + \triangle t$. In the case of a pure forecasting problem, the agent's only concern is the reduction of the guessing error (E_g) via some sort of learning action (A_g): the agent tries to improve its forecast. In this mode, $S_{t+\triangle t}$ plays the role of the target value, and the agent modifies $P_{t+\triangle t}$ to try to match it.

Conversely, for a basic control problem, the agent seeks to change S_t to bring it closer to the target / goal G_t and thereby reduce the control error (E_c). The cheetah does so by acting in a manner that pushes the gazelle from its current location toward the wide-open savanna. However, cheetahs and gazelles move quickly, with the state of the body-world coupling changing too fast for the nervous system to keep up, so a successful cheetah will probably use its guessed state and that state's difference from the goal (E_{gc}) to govern its actions (A_{gc}).

Following the basic philosophy of relativism (that no truth or knowledge is absolute, only relative), an agent's goals represent desired states of the world *as perceived by the agent*. Thus, G_t constitutes a particular target state of the agent's sensory apparatus, essentially shrink-wrapping the agent's scope of perceived space and time onto its receptors. Goals thereby become very intimate states of the agent, as do predictions of future states. The

simplest organisms lack explicit representations for both goals and predictions, though their actions give the impression (to an outside observer) that they have target states and maybe even guesses (as to how their prey will move in the next half second).

In moving up the ladder of cranial complexity, we find organisms able to create anticipatory states, but these guesses would probably confer a survival advantage only if they functioned as goals, and thus, $P_{t+\triangle t} = G_{t+\triangle t}$. A crab probably gains little from imagining future states per se, but by having some conception of a goal and how to nudge its current state toward that target, it would seem to have a claw up on any competitors who lacked the ability to represent alternative states. The next step, divorcing $P_{t+\triangle t}$ from $G_{t+\triangle t}$, surely took considerable evolutionary time. But the neural mechanisms that allowed the formation of one type of alternative reality, $G_{t+\triangle t}$, were probably usurped and enhanced to support the other, $P_{t+\triangle t}$.

Today, in analyzing the behavior of an artificial neural network that predicts the next value (V) in a sequence (e.g., tomorrow's high temperature based on the daily highs of the past month), we view V as the target, and the network's output as $P_{t+\triangle t}$: reality (V) is objective, and primary. But in a living agent, reality is subjective, and secondary to the agent's goals, toward which it will try to bend its perceived reality through actions on its own body and their influences on the surroundings. This egocentric view of behavior plays a vital role in understanding the rudiments of prediction in biological systems, and even some of the theoretical underpinnings of certain types of artificial neural networks.

In general, many discussions of prediction should begin by clarifying the viewpoint: egocentric (subjective) or ecocentric (based on a world or environment that embodies objectivity[1]). Thus, the neural networks studied in deep learning typically assume an ecocentric stance, with the data set being the objective truth, and the learning algorithm trying to modify the network parameters so as to generate outputs well-correlated with that reality. Conversely, agents (biological or artificial) that perform actions in a world (real or virtual) tend to have goals (such as survival and reproduction in living organisms) that take precedence over *building objective models of the world*. The models that they do craft are, more likely, biased by their own abilities to perceive and act. A pelican needs no detailed model of the ocean's depths to successfully dive for fish.

2.3 Gradients

Although change is ubiquitous in the dynamic environments that we inhabit, surprising (i.e., unpredicted) changes are much less frequent. The common (often correct) assumption that tomorrow will be much like today is not a belief in stasis so much as a prediction that tomorrow's events will *unfold* much like today's: the same basic pattern of change will repeat. This faith in reoccurring or gradually changing patterns (i.e., trends) underlies much of our predictive power. By monitoring and quantifying these trends, we build our foundation for speculation. And for the most part, we are reluctant to disregard these trends in the absence of unusual, unexpected sensory data.

Mathematically, these trends are often formalized as *derivatives* or *gradients*: the change in one variable ($\triangle y$) as correlated with or caused by the change in another variable ($\triangle x$). In other words, how *sensitive* is the value of y to changes in x? Figure 2.2 provides a simple graphic illustration of this concept, often described as *the rise over the run*.

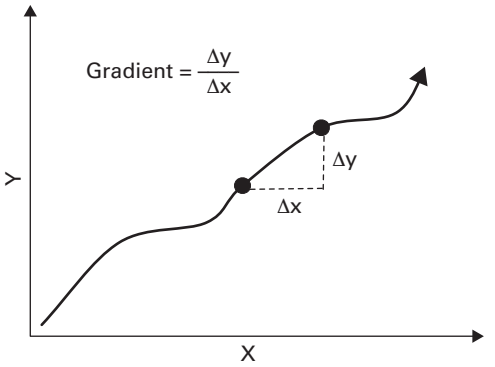


Figure 2.2
The basic mathematical concept of a gradient: the change in one variable ($\triangle y$) with respect to the change in another ($\triangle x$). Viewing y as a function of x, that is, y = f(x), then the gradient (or derivative) of y with respect to x is commonly written as $\frac{\partial y}{\partial x}$ or $f'(x)$.

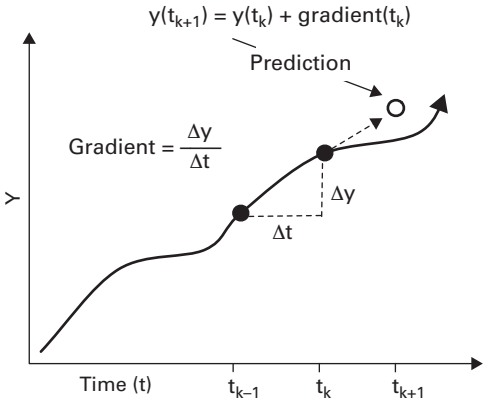


Figure 2.3
Gradients and their role in prediction. Basic interpolation using y's gradient (with respect to time) at time t_k enables a simple, primitive forecast of $y(t_{k+1})$.

The assumption of the repeated pattern plays out mathematically as an interpolation of a recent gradient to a point in the future. The graph of figure 2.3 uses time as the variable x and shows the gradient calculated from time t_{k-1} to time t_k, which is then extrapolated (dashed arrow) to a prediction for the value of y at time t_{k+1}. In this case, the short-term prediction is quite weak, since y briefly flattens out after time t_k, but the longer-term estimate looks more promising. Basically, the gradient serves as a poor man's tool for prediction, but one used frequently in both nature and technology.

Mathematically, the classic construct for prediction by gradients is the Taylor series, which provides estimates for $y_1 = f(x_1)$ when given $y_0 = f(x_0)$, $\triangle x = x_1 - x_0$, and all of the derivatives (first, f', second, f'', etc.) of f(x) at x_0:

$$f(x_1) = f(x_0) + f'(x_0)\triangle x + \frac{f''(x_0)}{2}\triangle x^2 + \frac{f'''(x_0)}{6}\triangle x^3 + \ldots \qquad (2.1)$$

Back in calculus class, the Taylor series was just another one of those oddities that we had to learn, but we probably never gave much thought to some inherent practical problems. After all, if we know f(x) and x_1, why can't we just plug x_1 into f(x) to produce y_1? Oh, but we don't really know f(x), yet we know a whole series of its derivatives at x_0? In fact, we can calculate an infinite sequence of those derivates at or near x_0 but we don't have a general understanding of f(x) itself. That is, we do not know f(x) *for each possible value of x*? What kind of world is this?

As it turns out, this is the real world outside of the calculus book, the world in which we have to make predictions based on weak information, because the general expression for some quantity y as a function of another quantity x eludes us. So we can measure y_0 at (some point in time or space) x_0, and we can measure how y changes as x deviates from x_0, and how the x-induced change in y changes as x changes, and how the x-induced change in the x-induced change of y changes as x changes, and so on. Then we can plug all of that information into the Taylor formula and estimate/predict y_1 at the new point x_1. This sounds great, but in so many cases, discerning anything beyond the first derivative becomes a real chore, and we're basically back to a fairly primitive interpolation similar to that of figure 2.3. But, again, that's often enough: many practical applications of the Taylor series only use the first derivative anyway.

2.3.1 Gradients Rising

Gradients arise in a wide variety of subject areas, and only in toy textbook problems of those domains is a general expression for f(x) actually known. Table 2.1 summarizes a few of these domains. Two key questions now surface in these and other areas: (1) What predictions do the gradients support, and (2) How do these predictions enable adaptive behavior?

In analyzing the foraging behavior of bacteria, biologists examine the changes in nutrient concentrations across space and compare those gradients to the movement patterns of the microorganisms. This provides a measure of *intelligence* in terms of whether or not bacteria swim up a nutrient gradient (i.e., in the direction of increasing nutrient) and down that of a toxic chemical. As evidenced by many bacteria's well-documented tendencies to (a) continue swimming along promising gradients, but (b) lapse into random movements in uninformative gradients, these organisms make implicit predictions as to the nutrients beyond their sensory horizons.

On the other end of the spectrum lie some of the more complex decisions made by any agent: stock trading. Here, one standard gradient is simply how the stock price has changed

Table 2.1
Diverse domains and variable pairs within each for which relevant gradients $\left(\frac{\triangle Y}{\triangle X}\right)$ play an important role.

Domain	X	Y
Bacterial Foraging	Location	Nutrients
Finance	Time	Stock Price
Thermoregulation	Heat	Temperature
Evolution	Genotype	Fitness
Brain Development	Location	Neurotrophins
Deep Learning	Connection Weights	Output Error

during the recent past, which can then be used to predict the future price, which, in turn, dictates trading actions, all in the service of maximizing profits. In temperature regulation, an engineered system uses several metrics, including the effects that changes in heat flow have recently had on ambient temperature; these indicate how future flow rates will affect temperature, which, in turn, determines how best to adjust flows in order to move temperature toward a target value.

Evolution (and similarly, evolutionary algorithms) displays interesting gradients, few of which directly assist in adaptive behavior but nonetheless provide enlightening perspectives on long-term population changes. For example, the sensitivity of fitness (F) to mutations of a particular portion of the genome (G) often indicates the degree to which G has evolved to satisfy environmental demands (or remains relatively decoupled from those selection pressures). If $\frac{\triangle F}{\triangle G}$ has high magnitude, G has probably been forced to bow to selection pressure over the generations and is also a likely target for genetic modifications that have significant phenotypic impacts. Such genes often show low variance in the population, since strong selection may favor a small subset of the alleles.

From a more theoretical perspective, evolution typically follows fitness gradients in an implicit manner. Consider the situation in figure 2.4, when two parents occupy different elevations on the fitness landscape. The more fit (higher) parent will tend to produce more offspring (that survive), and thus the population average will climb the gradient (without any explicit awareness, goals, or actions of the individuals or species). Gradient ascent is a simple, indirect, emergent property of evolution by natural selection.

The inset of figure 2.4 illustrates explicit gradient following in evolution, wherein parents (a) have full awareness of the fitness landscape, (b) can control the genomes of their

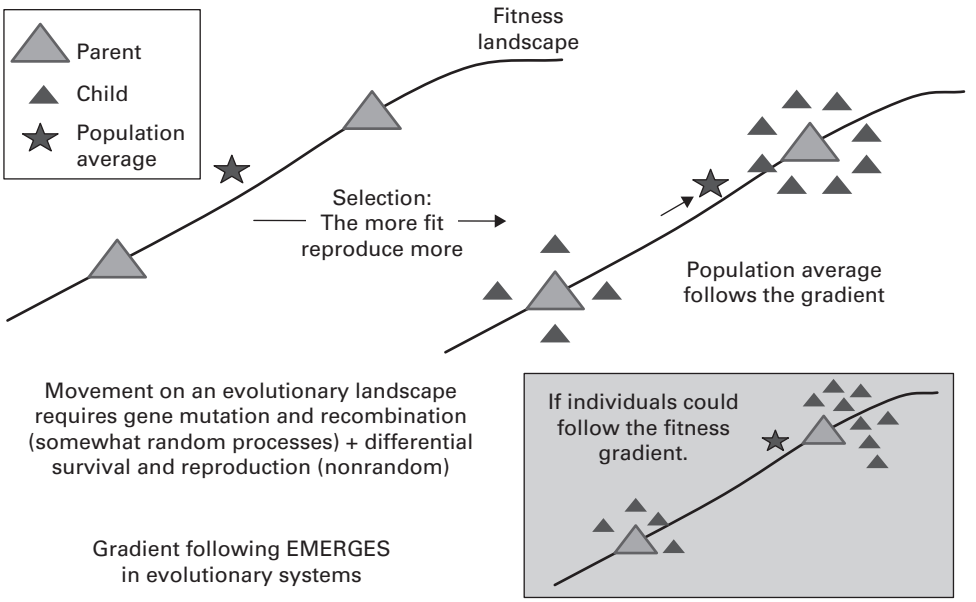


Figure 2.4
Evolution as gradient following on a fitness landscape. (Main diagrams) Implicit gradient ascension performed by evolution. (Inset) Hypothetical result of explicit gradient following (if parents could genetically engineer their children).

offspring, and (c) can predict phenotypic consequences of genetic change. This empowers parents to generate offspring above themselves on the fitness landscape, thus raising the population's average fitness. Notwithstanding contemporary genetic engineering, this is still mainly a hypothetical scenario; but the difference between the two images indicates the potential (disconcerting) acceleration of evolution incurred by designer babies.

Early development of the brain involves numerous spatiochemical gradients that perform functions such as laying out the body axes, providing pathways for the migration of neurons to their appropriate regions, and growing axons to proper dendritic targets. Few biological processes are as awe-inspiring as microscopic videos of the emergence of brains and bodies via the physical adaptations of cells based on the chemicals secreted by other cells.

Finally, in the field of deep learning (DL), gradients are ubiquitous. Their typical embodiment is in the derivative of a network's output error with respect to its tuneable parameters, for example, the weights on interneuronal connections. These networks learn by modifying the parameters to reduce the net's output error, in a manner governed by the gradients (which represent the change in error as a function of the changes in weights). A neural net for solving a complex problem may house millions of such gradients, most of which constitute *long-distance connections* in the computational structure of the network: many sequential calculations separate any of these parameters from the network's output (and hence from the output error term).

Each new DL variant typically includes creative accessory network components that introduce novel parameters into contemporary DL architectures, and gradients of output with respect to these new parameters must be calculated to enable learning. The key to implementing any new architecture lies in mathematically formalizing the complex gradients.[2] These long gradients can yield impressive results on machine learning tasks, often outperforming humans, but they have little biological realism and therefore have limited utility as models of natural cognition.

Many of the artificial neural networks (ANNs) covered in this book exhibit different dynamics than the DL architectures currently in vogue. They are predecessors to some of today's networks, but with different behavioral goals, formalized as objective functions with properties inherited from thermodynamics. These objective functions are mathematically shown to decompose the biologically implausible long derivatives into local gradients (of prediction error with respect to nearby synaptic weights). Crucially, these prediction errors are local to each neural layer as components of a hierarchical brain model known as *predictive coding* (Rao and Ballard 1999), a central topic of this book.

2.4 Sequences

Prediction tasks are often couched as sequence-completion problems, such as 1,4,7,Θ, where correctly finding Θ entails making an accurate prediction. This could be phrased as a simple analogy problem: *1 is to 4 as 4 is to 7 as 7 is to* Θ. The problem becomes trivial once one discovers the relationship between any two adjacent elements and then applies that relationship to 7 to yield Θ's value, 10.

Note that with number sequences, the relationship constitutes a gradient, $\frac{\triangle N}{\triangle P}$, where N denotes the sequence element and P is the position index. In this case, simple observation reveals that $\frac{\triangle N}{\triangle P} = 3$, and the prediction is $\Theta = 7 + \frac{\triangle N}{\triangle P} = 7 + 3 = 10$.

This easily scales up from single numbers to points in a multidimensional space. Can you solve for Θ in the following sequence?

$$[1, 5, 8], [3, 3, 18], [5, 1, 28], \Theta \tag{2.2}$$

In this case, we compute gradients (relationships) in each dimension independently, then apply them to [5,1,28] to produce Θ. These gradients are [2,−2,10], so adding this to [5,1,28] yields Θ = [7, −1, 38], and we expect all four of these points to lie along a straight line in three-dimensional space.

When we go *off the (Cartesian) grid* into a nonnumeric space, the problems become slightly harder, for example: A, D, G, Θ. Solve for Θ. Although most people tackle this problem with ease, they do so despite a host of potentially confounding issues involving gradients: What is the salient relationship that links A to D and D to G? Is it the difference in sounds made by a speaker of language L when pronouncing each letter, or maybe the sound frequencies produced by a piano when playing each of these notes? How about a speaker of language M or a saxophone player? We need to establish an underlying substrate, known in mathematics as a *metric space*, in which *distance* has a definition similar to our normal conception from Cartesian space.

In the first sequence, we simply mapped the numeric symbols 1, 4, and 7 to a number line in one-dimensional space, probably without even thinking about it. And, almost as fluidly, most people would map A, D, and G to their indices in the English alphabet (again, 1, 4, and 7), which then map to the number line. Once in the metric space, we can compute the gradient (3), add it to our rightmost[3] point (7) to produce 10. As a final step, we then project 10 back into English-alphabet space, where the tenth letter is J. So $\Theta = J$.

What if the sequences involve words, such as *jot, lot, not*, Θ? In this particular case, each word readily maps to three indices (one for each letter) in Cartesian space: [10,15,20], [12,15,20], and [14,15,20], and each adjacent pair has the gradient vector [2,0,0]. Adding this to the vector of *not*, gives [16,15,20], which projects to the word *pot*. So Θ = pot.

Of course, most word sequences require different tactics. If they represent familiar phrases, then previous exposure allows pattern completion by rote memory. For example, Elvis Presley fans can easily solve for α and β in *You, ain't, nothing, but, a,* α, β. But the words *hound* (α) and *dog* (β) do not naturally follow from the others in any obvious Cartesian space. Neither do *heart* (α) and *break* (β), the likely prediction from fans of Backstreet Boys. Eager listeners of both artists might request more context (i.e., a few more predecessor words) to disambiguate the two options, but these would only enhance the contextual priming of rote memory while decreasing the (already remote) possibility of finding some useful Cartesian space in which gradients could drive prediction. When it comes to predictions involving overly familiar sequences, the only Cartesian space may be the one expressing the temporal order in which you (repeatedly) heard the phrase, but not even that would provide a meaningful gradient for predicting the completion. Basically, rote memory requires no real understanding of the sequence elements, just a mental connection between them based on nothing more than their temporal juxtaposition.

Somewhere between [1,4,7,Θ] and complete-that-tune lie prediction problems that require deep understanding of the situation but suggest no obvious Cartesian spaces in which gradients can make a contribution. However, finding useful metric spaces may require only a bit of creativity and general experience in the world. Consider this sequence: white bear,

brown gopher, green snake, Θ. If you cannot embed this sequence in a fairy tale or nature program to give it some useful semantics, a decomposition into two metric spaces may prove useful: size and seasonal colors. Since bears are larger than gophers, which are larger than (most) snakes, the prediction that Θ is *smaller than a snake* makes sense. In many parts of the world, each season has a dominant color, with the first season (winter in the northern hemisphere) characterized as white, the second (early spring) as brown, then green in the summer, followed by the oranges and reds of autumn. Thus, an intelligent guess for Θ might be *orange beetle*.

In fact, true understanding of a concept (even one as intangible as *disappointment* or *omnipresent*) may demand its embedding in a metric space in which gradients have predictive power. As radical as this may sound, it seems fairly straightforward to place most concepts on some form of spectrum involving several others, such as [devastated, depressed, melancholy, disappointed, bored, complacent, content, happy, gleeful, elated]. We can then compute gradients of this space with respect to another factor, such as the amount of positive reward received by an agent. Thus, any increase (decrease) in reward should cause movement in the direction of elation (devastation). Understanding arises from these embeddings, probably several for any given concept. By relating them to one another and calibrating how both external factors and one's own actions can move along these spectra, an agent gains salient knowledge and survival advantages above and beyond those accrued by knowing all Elvis Presley' lyrics by heart.

This, very brief, foray into sequences and metric spaces and their relationships to understanding and prediction scurries around the rim of a very deep philosophical crater that was fearlessly explored by Peter Gärdenfors in his classic work, *Conceptual Spaces* (2000). The main purpose of this section is only to provide a rudimentary grounding of what many consider the classic prediction problem, sequence completion, in spaces where distance and gradients make sense. In chapter 3, when we look at the hippocampus, the grounding will descend further, to the level of individual neurons.

2.5 Abstracting by Averaging

Predictions based on gradients put faith in recent (and/or generally understood) changes and their continued relevance: yesterday's run on sirloin steaks at the local supermarket will continue today. Obviously, this is not a foolproof strategy, for example, if yesterday was the final day of the holiday barbecue season. When quantities fluctuate, that is, exhibit positive and negative temporal gradients in a short time span, then the safer approach is to simply compute a scaled sum (aka average) of the values during that period and use it as the prediction. For example, given little more information than the past month of temperatures, a reasonable guess of tomorrow's temperature is the monthly average, particularly in those equatorial parts of the world having little seasonal variation.

The sum (or integral, for continuous systems) provides a very basic, but irreplaceable, tool for prediction. Unlike the gradient, which essentially adds a level of nuance, the average elevates decision making to a higher level of abstraction: one that glosses over the details in favor of the scaled conglomerate, accumulated over space and/or time. Predictions based primarily on sums may totally fail in the short term—yesterday's sudden uncharacteristic

drop in a stock price may continue, unabated into today's trading—but still provide stellar long-term performance: the stock proves stable over the entire summer and provides a safe haven for calm investors.

Along with the conventional arithmetic average (the sum of a set of values divided by the size of the set), the weighted average plays a very important role in prediction. In the equation below, Θ represents the weighted average of the x_k values:

$$\Theta = \sum_{k=1}^{N} w_k x_k \text{ where } \sum_{k=1}^{N} w_k = 1$$

For the standard average, all weights (w_k) are identical and equal to $\frac{1}{N}$, but in many predictive situations, the x_k values closest (in space or time) to the value to be predicted (Θ) will have more relevance and thus have higher weights than the more distant values. For instance, when predicting the grade-point average (GPA) of next year's incoming class of mathematics students, the department administrator might average over the last ten years of (average) GPAs, but with the highest weights given to the past two or three years.[4] Weighted averages of this sort are ubiquitous in AI algorithms, particularly those involving a time component, such as reinforcement learning (RL). They also loom large in spatial domains, such as image processing, where filters applied to small regions of an image often exhibit biases toward nearby pixel values.

This book examines prediction in hierarchical neural systems, where higher levels produce expectations of values at lower levels. These upper predictors typically run at slower timescales. Thus, they are less jittery, and tend to react to averages of lower-level behavior more than to momentary changes, although they may, in addition, accumulate those changes into an average gradient. Regardless, the inherent delays in the system make it infeasible for these slower responders to make predictions solely on the basis of a single gradient (which may be outdated / *stale* by the time it is received). Thus, the higher levels are forced to rely heavily on quantities that have been abstracted over time and/or space, with the typical abstraction tool being scaled summations akin to weighted averages.

2.6 Control and Prediction

Armed with gradients and weighted averages, we can revisit the basic control problem of figure 2.1 and formulate a simple predictive scheme. Figure 2.5 depicts the same situation as earlier, but now with the focus on calculation of the *guess*, $P_{t*+\triangle t}$. Define two positive factors, k_g and k_a, for the gradient and average, respectively.[5] Thus, the prediction becomes a weighted combination of gradient and biased average. The gradient, based on timestep $\triangle t$, goes back one time increment, while the weighted average covers the present state value (S_{t*}) plus four historical points (drawn as gray octagons).

One reasonable formula for the prediction is then

$$P_{t*+\triangle t} = \underbrace{k_g \left(S_{t*} + \frac{\triangle S_{t*}}{\triangle t} \right)}_{\textit{gradient-based}} + \underbrace{k_a \left(\frac{9}{16} S_{t*} + \sum_{j=1}^{4} \frac{1}{2^{j+1}} S_{t*-j\triangle t} \right)}_{\textit{average-based}} \tag{2.3}$$

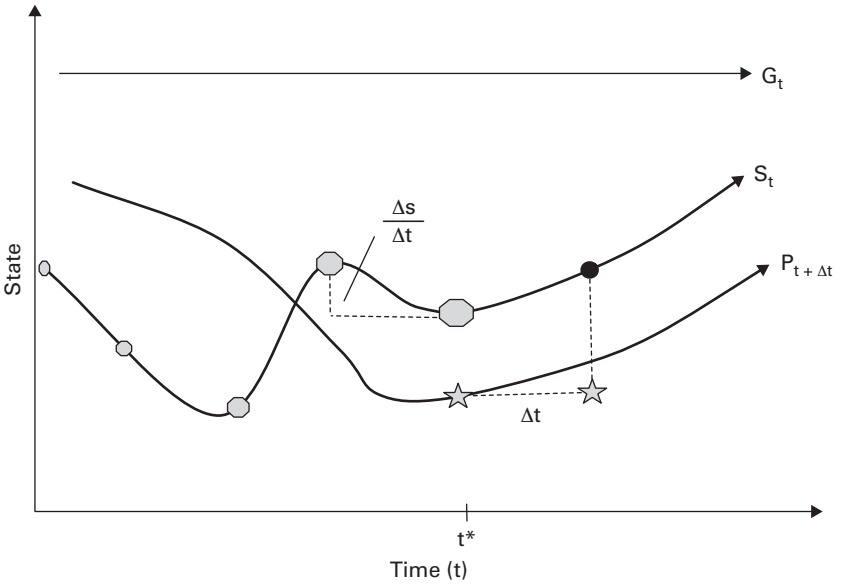


Figure 2.5
Basing prediction on a combination of gradient and average, using the same scenario as depicted in figure 2.1. At time point t*, to compute the prediction ($P_{t*+\triangle t}$, denoted by the gray star) for state S at time $t* + \triangle t$, combine the gradient $\left(\frac{\triangle S}{\triangle t}\right)$ and a weighted average over S values in the recent past (gray octagons). Shrinking size of octagons going backward in time represents decreased weighting of those points in a biased average.

where the first average term's weight $\left(\frac{1}{2} + \frac{1}{2^4} = \frac{9}{16}\right)$ is assigned such that all weights sum to 1. Equation 2.4 gives a more generic form, averaging over M+1 points, $\vec{S_t}$, with weights, w_j, that sum to 1.

$$P_{t+\triangle t} = \Gamma(\vec{S_t}) = \underbrace{k_g S_t + k_g \frac{\triangle S_t}{\triangle t}}_{gradient\text{-}based} + \underbrace{k_a \sum_{j=0}^{M} w_j S_{t-j\triangle t}}_{average\text{-}based} \tag{2.4}$$

In a typical control problem, the error, $E_t = G_t - S_t$, governs the choice of a control output, commonly designated as u_t, which then feeds into the system being controlled (aka the *plant*), which produces a new state $S_{t+\triangle t}$. Letting Φ denote the controller, and Ω the system / plant:

$$E_t = G_t - S_t \tag{2.5}$$

$$u_t = \Phi(E_t) \tag{2.6}$$

$$S_{t+\triangle t} = \Omega(u_t, S_t) \tag{2.7}$$

For simple systems, Φ may just multiply the error by a positive fractional constant to try to bring S closer to G. However, as discussed earlier, the delays inherent in the sensorimotor systems of most living organisms preclude the use of the current state as the basis for action. Rather, a prediction of the future state (at time $t + \triangle t$) is necessary, and it, in turn, will yield a predicted future error, which can then initiate an action, u_t, which will not take effect until

time $t + \triangle t$, due to motor delays. Considering all of the different delays will unecessarily complicate the analysis, but a simple update of the model yields

$$P_{t+\triangle t} = \Gamma(\vec{S_t}) \tag{2.8}$$

$$E_{t+\triangle t} = G_{t+\triangle t} - P_{t+\triangle t} \tag{2.9}$$

$$u_t = \Phi(E_{t+\triangle t}) \tag{2.10}$$

$$S_{t+\triangle t} = \Omega(u_t, S_t) \tag{2.11}$$

$$P_{t+2\triangle t} = \Gamma(\vec{S_{t+\triangle t}}) \tag{2.12}$$

In this model, note that the action choice for time t (u_t) is a function of the estimated error at $t + \triangle t$. If Φ represents a standard proportional-integral-derivative (PID) controller, it computes the control output, u, as a function of the current value, derivative, and weighted average of the error term, not of the state itself and not of a single value of the error, but of the whole history of errors. Assuming that G_t, the goal, is a constant (G) throughout the regulatory period, the expression for error is simply $E_t = G - S_t$. When the situation demands a predicted error, $E_{t+\triangle t}$ as input to Φ, it can be derived from the current and previous errors via an identical formula to equation 2.4, but with $\vec{S_t}$ replaced by $G \vec{-} S_t$:

$$E_{t+\triangle t} = \Gamma(G \vec{-} S_t) = \underbrace{k_g(G - S_t) + k_g \frac{\triangle(G - S_t)}{\triangle t}}_{gradient\text{-}based} + \underbrace{k_a \sum_{j=0}^{M} w_j (G - S_{t-j\triangle t})}_{average\text{-}based} \tag{2.13}$$

Compare this to a standard discrete model of the PID controller, wherein the control output, u_t, stems directly from the error terms:

$$u_t = k_p e_t + k_d \frac{\triangle e_t}{\triangle t} + k_i \sum_{j=0}^{t} e_j \tag{2.14}$$

where e_t is the goal state minus the current state.

The similarities between equations 2.13 and 2.14 are obvious, and none of the differences detract from the main conclusion: making predictions and controlling a system are very similar operations. This becomes clear after a few more manipulations.

Since G is constant and $\sum_j w_j = 1$, the following simplifications hold:

$$k_g \frac{\triangle(G - S_t)}{\triangle t} = -k_g \frac{\triangle S_t}{\triangle t} \tag{2.15}$$

$$k_a \sum_{j=0}^{M} w_j (G - S_{t-j\triangle t}) = k_a G - k_a \sum_{j=0}^{M} w_j S_{t-j\triangle t} \tag{2.16}$$

Putting this all together yields

$$E_{t+\triangle t} = (k_a + k_g)G - k_g S_t - k_g \frac{\triangle S_t}{\triangle t} - k_a \sum_{j=0}^{M} w_j S_{t-j\triangle t} \tag{2.17}$$

By the expression for the predicted state in equation 2.4,

$$E_{t+\triangle t} = (k_a + k_g)G - P_{t+\triangle t} \tag{2.18}$$

Hence the predicted error is simply a constant minus the predicted state, and computing a typical PID control output is akin to computing the predicted state: predicted values, prediction errors, and control decisions are nearly proxies for one another, with the same pivotal computations (using the same gradients and averages of system states) underlying all three. Very little separates prediction from control, and from a biological perspective, the ability to predict may have arisen from an organism's fundamental need to control, both its internal and external environment.

Control theory is no stranger to neuroscience, particularly in studies of the cerebellum (Wolpert, Miall, and Kawato 1998). More generally, brains function as the central controller of myriad physiological processes, from respiration, circulation, and waste removal to hormone balancing and growth. Although descriptions of these homeostatic processes rarely invoke prediction, that story has begun to change, as indicated by these introductory lines from neuroscientists Peter Sterling and Simon Laughlin in *Principles of Neural Design* (2015, xvi):

> . . . the core task of all brains: It is to regulate the organism's internal milieu—by responding to needs and, better still, by *anticipating needs* [my emphasis] and preparing to satisfy them before they arise. The advantages of omniscience encourage omnipresence. Brains tend to become universal devices that tune all internal parameters to improve overall stability and economy. *Anticipatory regulation* replaces the more familiar *homeostatic regulation*—which is supposed to operate by waiting for each parameter to deviate from a *set point*, then detecting the error and correcting it by feedback.

They go on to extoll the advantages of anticipation in sympathetic and parasympathetic regulation, where many of the problems that prediction ameliorates (such as internal load imbalances and supply-demand mismatches) stem from the time lags of corporeal homeostasis, just as latencies in the sensorimotor system create problems for a pure sense-and-react agent. Anticipatory regulation can also spill over into overt behavior, as animals search for particular environmental niches that will serve future metabolic or reproductive needs, for example, a watering hole before the actual onset of thirst, or a secure nest site prior to mating season. The next step in this progression is to behaviors that anticipate changes in the environment or body-world coupling, that is, the more common, ethological notion of prediction.

2.7 Predictive Coding

The notion of *predictive coding* stems from neuroscientific work in the 1980s and back to psychological theories and engineering methods of the 1950s and 1960s. This section introduces the main aspects, which arise in many contexts throughout the book, while chapter 5 covers the topic in depth.

Imagine a large hierarchical organization such as a private corporation, national military, or university. In each, information flows in at least two directions: top-down and bottom-up. Although every member appreciates being informed, few function well under data overload conditions. A good deal of the hierarchy's effectiveness stems from an efficient handling of the signal flow: getting people the information that they need, but not burdening them

with irrelevancies and redundancies. Thus, a middle manager receives updates from many lower-level workers but creates an executive summary as her main message to the upper eschelons, who then receive only the *vital essence* of the current situation, not the 1,001 individual stories and complaints. Similarly, many of the requests by individual workers are handled by managers, without relaying so much of the problem or solution upward. Most organizations work best when the vast majority of problems can be handled locally, without involving large chunks of human time and energy.

Similarly, when commands flow downward from the big brass to the managers and workers, the most efficient outcome is when operations get carried out to the leaders' satisfaction: when their expectations are met. A leader's favorite feedback from the level below is often, "Done," without a lot of elaboration. This normally signals that the predicted outcome has been achieved: mission accomplished.

Conversely, when problems arise, when visions / predictions do not come to fruition, the need for greater information rises steeply. Leaders and managers require more details (feedback) in order to adjust their expectations and solution strategies before sending updated commands back down the hierarchy. As the organization as a whole converges on a solution, as the expectations near realization, the signal flow can return to a trickle of simple commands, thumbs-ups and high fives.

This view of solving problems hierarchically while minimizing information transmission motivates predictive coding, which abstracts and formalizes the basic idea of two-way information flow and signal filtering. In the standard conception, predictions / expectations move top-down and sensory / reality signals travel bottom-up, and all aspects of reality that match coincident predictions can be removed (as redundant) from the upward signal: only mismatches between expectations and reality deserve further attention at higher levels.

This is illustrated in figure 2.6, where predictive coding's bottom-up *reality* signals serve as *target* values for top-down *predictive* signals. A comparator subtracts the prediction from the target to yield the *prediction error*, which then gets sent further upward in the neural hierarchy. Ideally, these bottom-up error signals attenuate up the hierarchy, with each prediction removing more of the residual, unpredicted target signal. Essentially, each level attempts to *explain away* targets at the next-lower level via its predictions.

The key implication of this model is that only unpredicted signals, that is, *surprise*, need travel very far up the hierarchy, thus saving the energy of transmitting a lot of redundant information. In short, if certain sensory information is expected, then why should a brain waste resources sending it around? The expectation alone, when unviolated, should be sufficient to trigger other activities, such as the proper motor responses.

Until we consider more details of predictive coding, the main aspects to keep in mind are the reality-prediction comparison and the fact that the resulting prediction error constitutes the primary signal upward in the hierarchy, the executive summary containing all and only the information needed at the next level.

2.8 Tracking Marr's Tiers

In his seminal work in the early days of computational neuroscience, David Marr (1982) introduced a three-tiered framework for analyzing complex systems such as brains and AI systems: computational, algorithmic, and implementational. These names, though

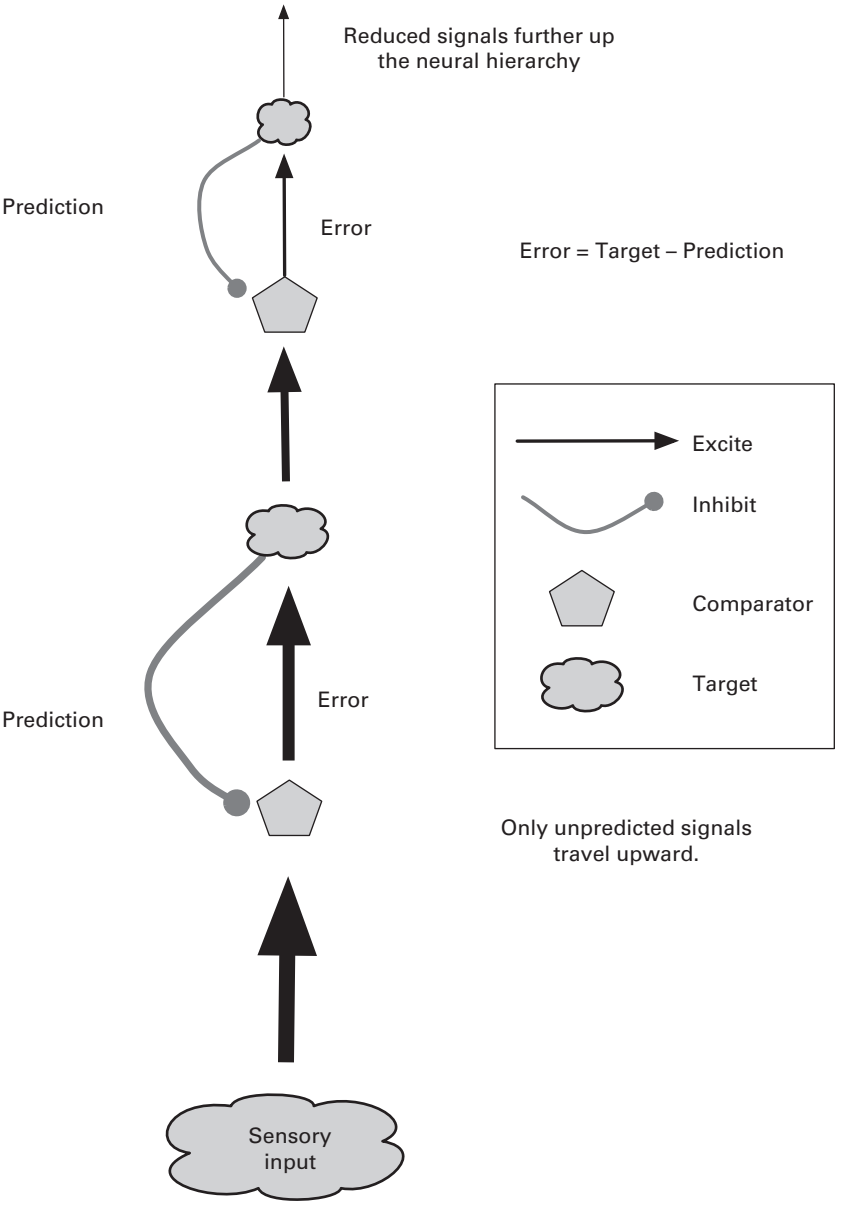


Figure 2.6
A simple illustration of predictive coding, with target signals moving upward in the neural hierarchy while gradually being reduced by predictions. Connection thickness mirrors the strength of upward and downward signals in this ideal situation of bottom-up error reduction.

somewhat misleading, correspond to three basic questions:

1. *What* is the basic phenomena to be modeled or problem to be solved?
2. *How*, at the algorithmic or pseudocode level, do you describe the plan of attack?
3. *Which* substrate will actually perform the computations? Transistors, DNA molecules, neurons?

So far, our discussions at the computational level have concerned one primary phenomenon: prediction, though others such as adaptation, emergence, and control play important roles as well. The current chapter has moved to the algorithmic level by showing how predictive tasks flesh out in terms of very basic computations: weighted sums, differences, products, and quotients.

Those predictions that lack this basic computability (via mental movement in a spatiotemporal or conceptual space) may rely on rote associative learning. For example, in classical conditioning experiments that link some random conditioned stimulus (e.g., a flashing light) to an unconditioned stimulus (e.g., an electric shock), the light allows the animal to predict the shock and then act to avoid it. Or, on more pleasant notes, song learning facilitates predictions of successive words that typically have no analogous neighbor relationship in a metric space. Sums, products, and gradients might help a point guard hit a cutting forward with a perfect pass,[6] but they won't help her sing the national anthem.

All of this sets the stage for a journey down to the implementational level of neurons and neural networks, where we have no problem finding neural mechanisms that *can* carry out the primitive operations of prediction in artificial nets and *might* be doing so in real brains as well.

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

This book was set in Times New Roman by Westchester Publishing Services.

Library of Congress Cataloging-in-Publication Data

Names: Downing, Keith L., author.
Title: Gradient expectations : structure, origins, and synthesis of predictive neural networks / Keith L. Downing.
Description: [Cambridge, Massachusetts] : The MIT Press, [2023] | Includes bibliographical references and index.
Identifiers: LCCN 2022037237 (print) | LCCN 2022037238 (ebook) | ISBN 9780262545617 (paperback) | ISBN 9780262374682 (epub) | ISBN 9780262374675 (pdf)
Subjects: LCSH: Deep learning (Machine learning) | Neural networks (Computer science) | Conjugate gradient methods.
Classification: LCC Q325.73 .D88 2023 (print) | LCC Q325.73 (ebook) | DDC 006.3/2—dc23/eng20230302
LC record available at https://lccn.loc.gov/2022037237
LC ebook record available at https://lccn.loc.gov/2022037238