This is a section of doi:10.7551/mitpress/14723.001.0001

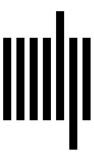
Gradient Expectations

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

5 Predictive Coding

5.1 Information Theory and Perception

In the mid 1980s, I enrolled in a graduate seminar, Perceptual Cognition, at the University of Oregon. Our weekly meetings were classic grad-school fare, with a dozen students, faculty members, and semi-mysterious others lounging on deep couches, drinking tea, and discussing topics well beyond the safe confines of my primary field of doctoral study: computer science. Luminaries such as Michael Posner occasionally dropped in to lend full legitimacy to the gatherings, but some of the other older participants seemed out of place. Eugene, Oregon, has always had its share of interesting and interested individuals who attend classes for no other reason than their immense curiosity, so I quickly grouped a few of the others into that category, including a long, lanky man with a thin goatee and a distinct southern accent who spoke rarely but got the full attention of Posner and others when he did. As has happened several times in my career, not until years later did I realize that I had been in the presence of a true giant of science: the bearded gentleman with the deep voice and deeper insights was Fred Attneave, a pioneer at the crossroads of psychology and information theory.

In 1954, Attneave introduced his *theory of redundancy reduction for visual perception* with a seminal paper (Attneave 1954) outlining numerous properties of images that facilitate data compression, thus easing the burden of signal transmission in perception. Essentially, any instances of temporal or spatial correlation among sensory inputs provide fodder for systematic reductions in signal load:

> When we begin to consider perception as an information-handling process, it quickly becomes clear that much of the information received by any higher organism is *redundant*. Sensory events are highly interdependent in both space and time: if we know at a given moment the states of a limited number of receptors (i.e., whether they are firing or not firing), we can make better-than-chance inferences with respect to the prior and subsequent states of these receptors, and also with respect to the present, prior and subsequent states of other receptors. (Attneave 1954, 183)

Attneave's work introduced predictive coding to cognitive science, though the term itself would not appear for several more decades. Another precursor was Oliver's (1952) *efficient coding*, which found widespread usage in telecommunications starting in the early 1950s.

The general process of predictive coding is to recode (compress) data by removing all aspects of it that are predictable. Thus, the efficiently encoded version is the original minus

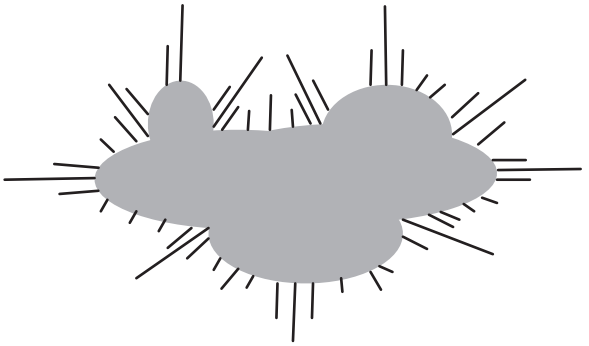


Figure 5.1
A general sketch reflecting the basic results of Attneave's classic experiment in which subjects placed ten dots at locations along the object's contour that they believed would most accurately delineate its shape (Attneave 1954). Lengths of emanating lines denote the number of subjects choosing that particular location.

the prediction. This difference, termed the *prediction error*, is the remainder of the original signal that gets propagated further in the system.

As a simple example of this basic concept, imagine the task of transmitting a series of integers (1000, 1002, 1003, 998, and 997) along a channel that carries only binary signals. Each integer would require 10 bits, for a total of 50 bits, if transmitted independently of the others. But by noticing the correlations among the five values—they are all very similar in magnitude—a more efficient strategy becomes obvious: transmit a base value (e.g., an average) plus the deviations of each from that base. Thus, the six values to transmit are 1000 (the average, and also the prediction) along with 0, 2,3,−2,−3. Assuming that the sign associated with each deviation requires one extra bit, each of these latter five values uses no more than 3 bits; and the entire sequence compresses to $10 + 5 \times 3 = 25$ bits. Essentially, the deviations from predictions / expectations / averages demand the most transmission effort.

As sketched in figure 5.1, Attneave illustrated the primality of deviations from the norm via an experiment in which subjects were asked to place ten dots at any locations along a shape's outline that they believed would most accurately approximate the pattern (when viewed in isolation). The results clearly showed the perceptual saliency of contour changes (i.e., sharp bends in the outline) in defining the figure. As Attneave suggests, a compressed description of the image would exploit strong spatial correlations to compress homogeneous regions (or stretches of the outline) into simple descriptors that combine the precise locations of a few points along with the neighborhood relationships between those points. With greater correlation comes greater possibilities for compression.

In 1982, Srinivasan and colleagues coined the term *predictive coding* in showing how inhibitory interneurons in the retina realize the data compression predicted by Attneave's theory (Srinivasan, Laughlin, and Dubs 1982). One primary motivation for their work—and possibly a key constraint in the evolution of the retina—is the uncertainty inherent in neural signalling, known as *intrinsic noise*. Even when sensory receptors provide accurate readings, transmission across multiple synapses inevitably degrades information. What can neural networks do to combat this?

To understand how predictive coding ameliorates this problem, imagine an internal neuron with a range of 0–200 Hz, where spike frequency conveys the main information. In

theory, this neuron can transmit 200 uniquely identifiable signals per second. Now assume that the operative range of the sensory input is, for example, 0 to 1000 units. Hence, the neural signal achieves a resolution of $\frac{1000}{200} = 5$ sensory units: each one-second interval of spikes transmits a sensory value with a precision of 5 units. But what happens when intrinsic noise hampers internal transmission?

Suppose that the intrinsic noise equals plus or minus 5 Hz: a signal that should be 87 spikes/sec could be anywhere from 82 to 92 spikes/sec. This represents a noise range of 10 Hz and restricts the one-second neural signal to $\frac{200}{10} = 20$ uniquely identifiable signals, since, for example, an 87 and a 90 are no longer guaranteed to convey different information. Any downstream neuron receiving those spikes cannot safely react to 87 differently than 90, since they may be transmitting identical sensory information. In effect, this reduces the functional resolution of the sensory inputs to $\frac{1000}{20} = 50$ units, that is, a vague blur compared to the noise-free condition.

Since both intrinsic noise and the upper bound on firing rates (in the range of 250–1000 Hz) are inevitable and immutable consequences of neurophysiology, the task of increasing functional sensory resolution becomes the responsibility of earlier processes in the signaling pipeline. In short, if the effective range of sensory signals can be reduced, then the weak signal-to-noise ratio of intrinsic signaling can still achieve respectable sensory resolution. For example, even in the noisy condition above, if the range of sensory information decreases from [0, 1000] to [500, 550], then a sensory resolution of $\frac{550-500}{20} = 2.5$ units results.

Predictive coding achieves this reduction in sensory range by normalizing the original values by their predicted values, with predictions based on local averages in time and/or space. In the above five-integer example, the naive original range was [0, 1023] (hence the need for 10 bits per value), but normalizing by subtracting the average reduced the range to [−3, 3], which could be transmitted either with fewer bits or with the original 10 bits but now with much higher resolution.

Srinivasan, Laughlin and Dubs (1982) showed that, indeed, in the retina, predictions derived from local averages greatly reduced the salient sensory ranges, and thus the brain's relatively coarse-grained neural signals can still transmit high-resolution sensory information: small *differences that make a difference* (Edelman and Tononi 2000). For spatial filtering, the basic mechanism is lateral inhibition (see figure 5.2), whereby retinal neurons reduce one another's outputs via contributions to nearby interneurons, known as *horizontal cells*, which provide negative feedback to sensory receptors, that is, the rods and cones. Note that this computes something similar to a stimulus' spatial derivative (value minus its average surroundings), which is then sent downstream.

Achieving predictive coding across time requires a more local neural architecture: individual neurons (or pairs) can perform delayed feedback self-inhibition, essentially computing a temporal derivative to send further up the processing hierarchy. Consider the situation in figure 5.3. With each successive pass through the receptor loop, older values have less influence on the final output (since $|w| < 1$) but maintain a trace effect that diminishes more rapidly for smaller magnitudes of the inhibition gain, w. Note however that the alternating signs in $d - wc + w^2b - w^3c$ fail to model the difference between a current value (d) and a weighted sum of its past values. A more accurate model recruits the inhibitory interneuron as an accumulator that leaks a fraction (1 − w) of its previous value while incrementing with

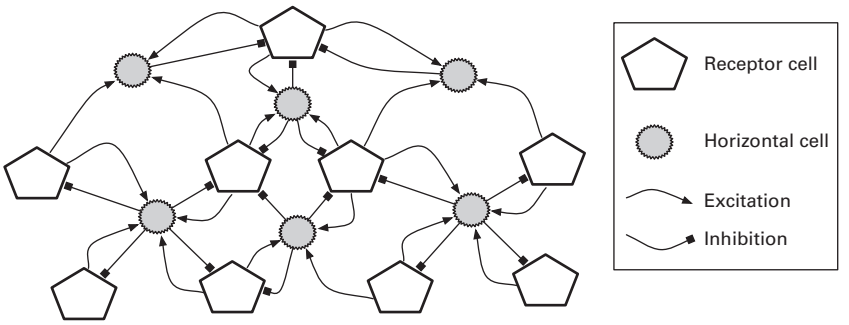


Figure 5.2
Basic architecture for spatial predictive coding in the retina, wherein horizontal cells aggregate inputs from nearby receptor cells and then inhibit the receptors with a strength commensurate with horizontal activity. The signal sent downstream by the receptors is then (approximately) their original input minus the neighborhood average (i.e., the prediction). In the actual retina, horizontal cells compute averages over thousands of receptors (Sterling and Laughlin 2015).

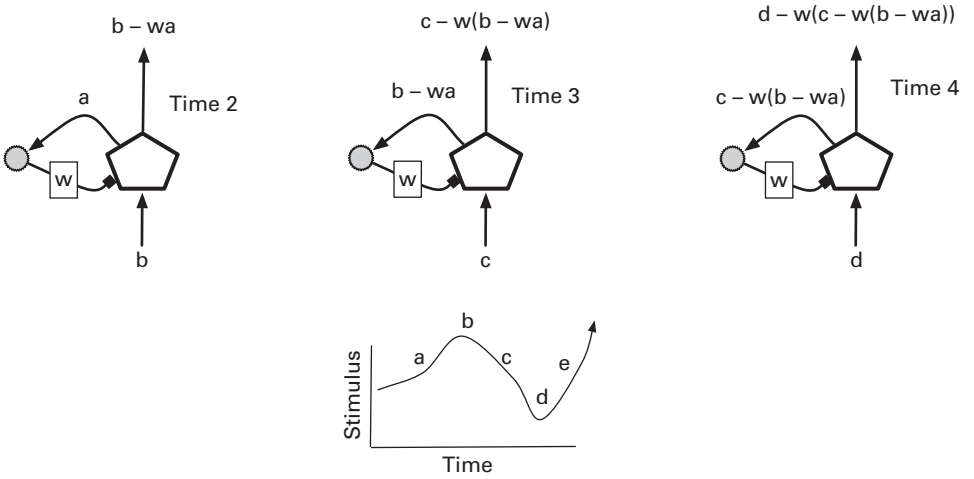


Figure 5.3
Basic synaptic connections required for temporal predictive coding, wherein the receptor cell (pentagon) inhibits itself via an interneuron (jagged gray oval). The output of the receptor at time t feeds back as a negative signal (with $|w| < 1$) at time t+1. The signal sent downstream by the receptor is then a fuzzy version of its temporal derivative.

the newest value. This accumulated amount then governs the strength of inhibition received by the primary receptor, as shown in figure 5.4. Now the output of the receptor manifests reality (the current value) minus a prediction (the weighted sum of previous values), with older values playing a less important role.

As discussed by several researchers (Stone 2018; Sterling and Laughlin 2015; Srinivasan, Laughlin, and Dubs 1982), predictive coding also has remedies for *extrinsic noise* in the original sensory stimuli (which reduces the signal-to-noise ratio of incoming signals). One key strategy is to increase sampling as the basis for predictions. For example, under conditions of low luminance, photoreceptors become particularly vulnerable to noise from random photons, since those photons may constitute a significant portion of the total photons received. This problem disappears in strong light, but to combat it in darker conditions, the predictive

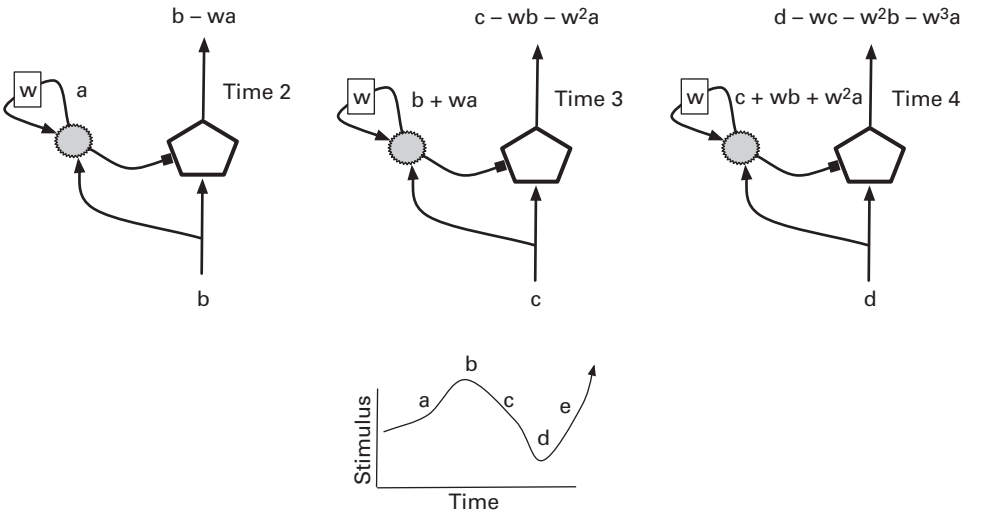


Figure 5.4
An improved model (compared to figure 5.3) of temporal predictive coding, with the interneuron accumulating the stimulus history and then using it to normalize the current signal.

coder can increase sampling in space and/or time. In the spatial context, a cell can expand the radius of neighbor cells that contribute to its prediction (i.e., local average), while temporally, it can increase the feedback gain (w) such that more historical values have significant influence on the prediction. Although beyond the scope of this book, physicochemical mechanisms enable the retina to adjust parameters such as the feedback gain and effective radius of spatial influence to properly adapt to widely varying lighting conditions and the diverse signal-to-noise ratios that they incur (Sterling and Laughlin 2015).

This dynamic aspect of predictive coding has dimensions beyond those of the general ambient signal characteristics: the network can also adapt to predominant sensory patterns, becoming less sensitive to the status quo and more responsive to *surprising* inputs. As detailed by Hosoya and coworkers, plasticity of the synapses from inhibitory interneurons to pattern-detecting cells can explain this flexibility (Hosoya, Baccus, and Meister 2005). Figure 5.5 illustrates the general concept, with generic components (detectors and inhibitors) playing the roles of retinal ganglia and amacrine cells, respectively. These reside a few levels downstream from the receptors and horizontal cells portrayed in figure 5.2, but the general relationships between receptor / detector and inhibitor are similar.

In this model, the activation of receptors and downstream detectors (having the former within their receptive fields) will also activate local inhibitory cells, thus normalizing all sensory signals by predictive feedback. Within regions affected by the stimulus (i.e., the dark gray pattern in figure 5.5), both detectors and neighboring inhibitors will have high firing rates, thus leading to Hebbian strengthening of the synapses between them. However, this Hebbian tuning actually has an anti-Hebbian effect: future stimulation of the inhibitor will cause greater *depression* of the detector. Thus, upon a later presentation of the same pattern, the detectors will initially fire but then experience strong inhibition to significantly weaken the standard predictive-coding signal (reality minus prediction) sent upward in the

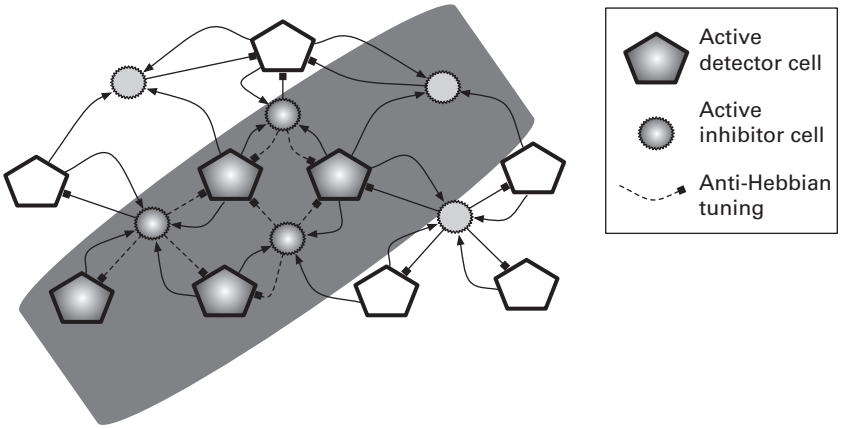


Figure 5.5
Generic model (based on relationships among bipolar, amacrine, and gangliar cells in the retina) of dynamic predictive coding that produces anti-Hebbian behavior via Hebbian modification of inhibitory synapses, as detailed in Hosoya, Baccus, and Meister (2005). The large dark pattern denotes the active receptive field to which downstream detector neurons (pentagons) respond, thus stimulating local inhibitor cells, which then negatively feed back on the detectors to manifest predictive coding. By modifying the synapses from the inhibitors to the detectors in a Hebbian manner, the predictive coding becomes dynamic: it exhibits lower sensitivity to the same (expected) pattern in the future. In this caricature, detectors are stimulated by the pattern, while inhibitors fire only in response to two or more active neighboring detectors.

neural hierarchy. The pattern will no longer be surprising and will thus stir up little activity beyond the earliest sensory levels.

In this manner, predictive coding becomes dynamic and can explain a wide range of adaptive effects in sensory perception. Note the difference between this and the *static* predictive coding discussed so far: the earlier variants involve predictions based on the immediate spatial or temporal situation. They can detect gradients / contrast, both weak and strong, by normalizing against the average. However, they learn nothing from the experience such that repetition of a sensory pattern minutes later will invoke the same levels of prediction error and thus be equally *surprising* to the network. In dynamic predictive coding, synaptic modification allows expectations to function across temporal scales much larger than the millisecond windows of neuronal firing; the novel pattern quickly becomes mundane and demands less signaling bandwidth when later repeated.

5.2 Predictive Coding on High

As the synaptic distance from sensory receptors increases in moving up the neural hierarchy—and thus as the receptive fields of individual neurons expand—the accuracy of predictions based on neighborhood averages should decrease and eventually disappear. Although topological maps (whereby neighboring neurons have neighboring receptive fields) do persist across many levels of sensory cortex (Kandel, Schwartz, and Jessell 2000), they vanish in higher cortical regions and the hippocampus, so the firing levels of neighboring neurons need not exhibit salient correlations. Similarly, as timescales for neural processing increase with distance from the sensory inputs and motor outputs, the utility of temporal derivatives for neural-activity prediction also diminishes.

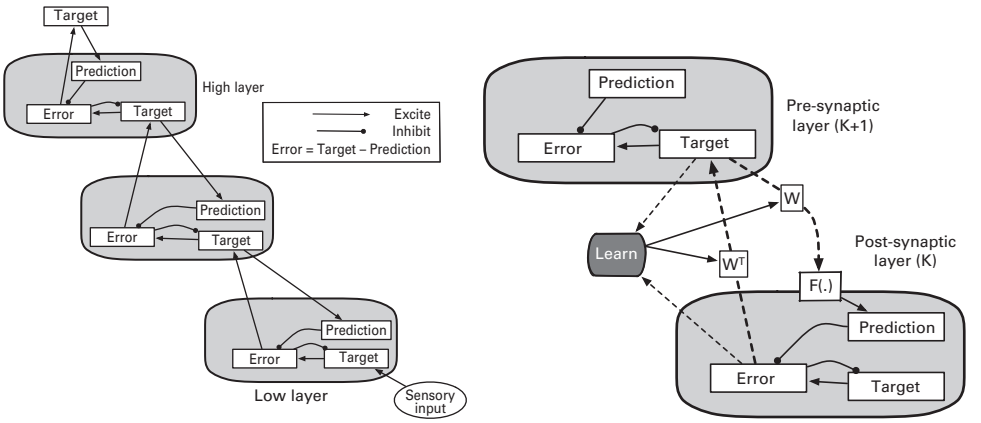


Figure 5.6
(Left) The essence of Rao and Ballard's (1999) predictive-coding model of the cortical hierarchy. (Right) Two neighboring levels of the same predictive-coding model in which weighted connections between targets and predictions are learned via Hebbian mechanisms based on correlations between the target at level K+1 and the error at level K. The (negated) predictions and targets at level K feed directly into the error units at K (in a one-to-one manner) without additional weighting. Each prediction, target, and error box represents a vector of twenty neurons for most of the simulations discussed in the text.

However, according to many contemporary models of cortical-column architecture and dynamics (Carpenter and Grossberg 2003; Mumford 1992; Rao and Ballard 1999; Spratling 2008, 2017), the brain has alternative predictive-coding mechanisms. Many of these shift the focus from shrinking the range of values encoded by individual neurons to reducing the sheer number of neurons in a cortical level whose signals require further (upward) transmission. Most of these models follow Rao and Ballard's (1999) basic paradigm, in which neurons in higher regions suppress the activity of lower-level neurons, whose outgoing signals now represent error in the sense of a mismatch between predictions and the *reality* conveyed by sensory signals. Rao and Ballard's work was inspired by David Mumford's (1992) theories about the functional significance of cortical columns and their connectivity patterns. Mumford's special focus was on the high degree of intercolumn linkage, in both the bottom-up and top-down directions, which portrays these columns not as modular units with simple, restricted interfaces and local computations, but as highly interactive groups whose bottom-up *residuals* (analogous to prediction errors) eventually reconcile with top-down predictions via an oscillating relaxation process similar to Carpenter and Grossberg's adaptive resonance theory (2003).

Figure 5.6 (left) portrays three predictive-coding levels (plus a topmost target vector, y_3) and the relationships among predictions, errors, and targets, where the latter denote sensory reality that predictions attempt to match. Note that the only signal sent upward is the error. Hence, whenever the target and prediction align, upward information flow attenuates; only error (aka *surprise*) need propagate further. Also note that the target at level K stems proportionately from the error at level $K - 1$ and inversely from the error at level K. Thus, errors at level $K - 1$ lead to modified targets at level K, which, in turn, alter the predictions at level $K - 1$.

Furthermore, when following the synaptic links around either of the main cycle motifs in the diagram (i.e., error-target-prediction-error and error-target-error), notice the presence

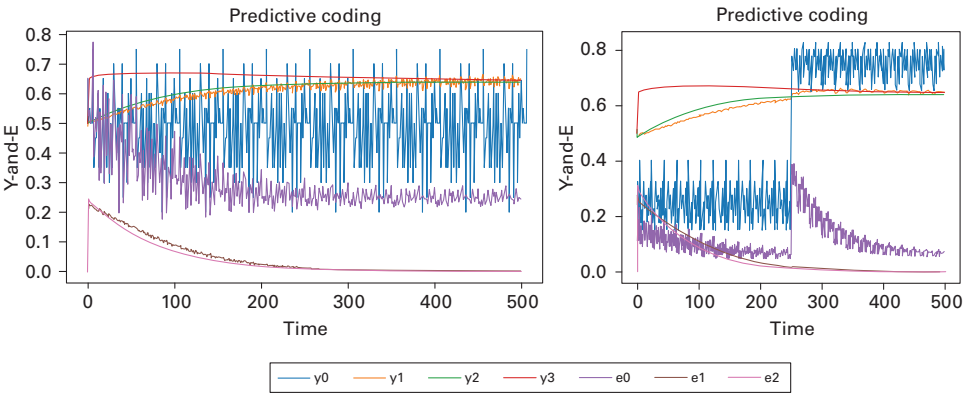


Figure 5.7
(Left) Progression of averages for target (y_i) and error (e_i) vectors, each of length 20, across three levels of a predictive-coding network similar to that of figure 5.6, where smaller indices denote lower levels and vector y_0 houses the time-varying inputs (that repeat every 50 timesteps). Input vectors are random uniform samples from set $\{0,0.5\}$. The plotted values of target vectors are arithmetic averages, while mean-square averaging applies to the error vectors; for each interlevel weight matrix, the learning rate λ is 0.001.(Right) Run of predictive coder where input fifty-vector sequence (y_0) changes at timestep 250 from samples of $\{0,0.5\}$ to samples of $\{0.5,1.0\}$.

of an odd number of inhibitory relationships, a prime indicator of negative feedback, which should encourage activation patterns to stabilize.

Figure 5.6 (right) displays a portion of the basic predictive coder in greater detail, revealing the weighted connections between the targets of level K+1 and the prediction (and then error) of level K. The jth neuron in the 20-unit prediction vector of level K receives weighted inputs from all 20 target neurons in level K+1. The weighted sum of these inputs then feeds into a sigmoid activation function to produce the jth prediction, which is then subtracted from the jth target value (also at level K) to yield the jth error term (at level K). Following the basic algorithm for a predictive coder (Spratling 2017), the transpose of the weight matrix that links targets (at level K+1) to predictions (at level K) is used to map the errors (at level K) to the target (at level K+1).

Adaptation in these networks occurs in two ways: (1) changes to targets in response to the errors at their own level and below, and (2) changes to *influences* of targets on errors (and vice versa) realized by the weight matrices. The latter constitutes learning and results from a standard Hebbian process: $\triangle w_{i,j} = \lambda e_i y_j$, with learning rate ($\lambda$), error ($e_i$) from level K, and target (y_j) from level K+1.

A simple simulation of the three-leveled network of figure 5.6 was performed using random input vectors of length 20, but whose sequence was repeated at intervals of 50 timesteps, thus giving the network an opportunity to adapt and improve. The results in figure 5.7 illustrate the transition to low errors and stability, despite the high variability of the input stream, y_0. In the leftmost plot, notice that all three target vectors (y0–y2) stabilize at nearly the same average, while the two higher-level error vectors (e1 and e2) converge to zero. Thus, by learning proper predictions, as embodied in both the prediction vectors and the weight matrices, the network drastically reduces the upward flow of error signals, with the only significant interchanges occurring between the lowest two levels. As shown in the rightmost plot, the predictive coder easily adapts to major changes in the composition of

the input stream, with relatively large transitions at the lower levels (i.e., level-0 errors, e_0, and level-1 targets, y_1) but only barely perceptible changes further up in the hierarchy, thus indicating coarser, more general information encoded in the upper levels. Notably, the plots of figure 5.6 are almost identical to those in which a three-tiered neural PID controller (displayed in chapter 6) was run on the same data. This is not so surprising, given the abundance of negative feedback loops mentioned above.

The general dynamics of primitive predictive-coding models such as that of figure 5.6 mirrors the behavior of cortical networks proposed by Jeff Hawkins in his groundbreaking book, *On Intelligence* (Hawkins 2004), and related publications (Hawkins and Ahmad 2016; Hawkins, Ahmad, and Cui 2017). Hawkins uses detailed analyses of the six-layered cortex to explain how brains could propagate sensory information upward only until it meets the downward flow of matching expectations: predicted sensory realities need not disturb higher cortical regions. Only surprising information travels far, possibly to the hippocampus, which represents the pinnacle of the cortical hierarchy and the gateway to long-term memory formation.

Along with Hawkins, several renowned systems neuroscientists (Rodriguez, Whitson, and Granger 2004; Ballard 2015) employ similar network models to explain the neocortex, particularly with respect to perception and prediction. These models draw on thorough accounts of cortical neuroanatomy from Vernon Mountcastle's seminal book, *The Cerebral Cortex* (Mountcastle 1998). The essence of these models is summarized in figure 5.8, which depicts three six-layered cortical columns. The following descriptions of the cortical column draw from those of Mountcastle, as well as Rolls and Treves (1998), Ballard (2015), Schneider (2014), and Hawkins (2004).

The basic design of these columns is preserved across the entire cortex and across species as diverse as mice and humans. Layer 1 consists almost exclusively of dendrites serving the excitatory pyramidal cells in all other layers, but particularly 2, 3, and 5. Bottom-up signals normally exit the lower-level column from layer 2/3 and enter the higher column at level 4, which contains primarily excitatory stellate cells. These resemble pyramidals but tend to have shorter axons and only synapse locally, in layer 4 and with the layer-2/3 pyramidals, which then convey signals both upward to higher columns and down to layers 5 and 6 of the same column. Top-down signals move primarily from layers 5 and 6 of the upper column to both the thalamus and the layer-1 dendritic mats of the lower column, then through layer 2/3 and on to layers 5 and 6 for further transmission down the hierarchy.

Although outnumbered by excitatory cells by a ratio of approximately five to one in the cortex, inhibitory neurons also play a vital role in columnar behavior: they help sparsify the activity patterns of the column's excitatory cells. The primary inhibitory neurons are basket and chandelier cells, which have short spatial ranges of influence of no more than 100 micrometers, whereas projections from excitatory neurons may extend several millimeters (Kandel, Schwartz, and Jessell 2000). Inhibitors often synapse directly on the somas or axon hillocks of excitatory neurons, thereby exerting a strong blocking effect. The presynaptic terminals of an inhibitor's axons emit GABA, whose receptors on postsynaptic terminals have slower binding and release times than those found at receptors for the excitatory neurotransmitters such as glutamate and AMPA. Thus, whereas the excitation of a pyramidal cell lasts 15–20 milliseconds, its inhibition normally endures for 100–150 msec (Rodriguez, Whitson, and Granger 2004).

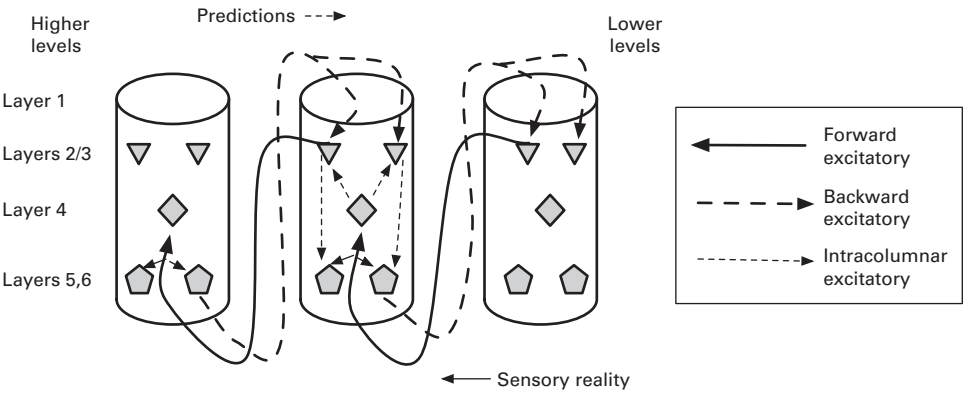


Figure 5.8
Basic functional circuits in the neocortex showing only the main intra- and intercolumnar excitatory connections. Columns are differentiated functionally (by their receptive fields), not necessarily anatomically, with each containing tens of thousands of pyramidal (pentagons and triangles) and stellate (diamonds) cells, grouped into modules housing hundreds of vertically interacting neurons in layers 2–6. Layer 1 is essentially a dendritic mat (Mountcastle 1998; Rodriguez, Whitson, and Granger 2004; Schneider 2014).

Lateral inhibition is commonplace in the brain, such that when a neuron N fires, it excites local inhibitors that quell the activity of neighboring neurons and (after 15–20 msec) N itself, all for the extended interval of 100–150 msec. This serves at least two purposes: (1) immediate hampering of neighbors promotes sparse activation patterns, which are much more convenient for information transmission and storage; and (2) strong, enduring inhibition of the originally active neuron(s), which constitute an activation pattern, gives other neurons an opportunity to participate in the next pattern, which helps reduce the overlap (and possible interference) between activation states, an obvious advantage for information storage and retrieval in a distributed memory.

The precise anatomical and physiological details of cortical columns (Mountcastle 1998; Schneider 2014; Rodriguez, Whitson, and Granger 2004) are far beyond the scope of this book, but some have special relevance for predictive coding, despite the lack of a fully comprehensive and empirically validated theory. In particular, the role of inhibition seems critical, since predictive coding entails a weakening of bottom-up signaling when top-down predictions match the rising sensory patterns. How could higher levels inhibit lower levels?

First of all, the short spatial range of inhibitory neurons indicates that level K+1 probably inhibits level K by sending down excitatory signals that trigger local inhibitors in level K. At this point, the difference between bottom-up and top-down signals becomes important. The former typically enter a column directly via layer 4, with *proximal* synapses (i.e., those near the cell soma), and with many layer-4 pyramidals receiving similar axonal input. Thus, they all have similar receptive fields. Conversely, the predictive action potentials enter via the dendritic mats of layer 1, thus synapsing *distally* (i.e., far from the soma) and therefore having a much weaker effect on the soma; a single top-down signal cannot normally induce the soma to generate action potentials, though it can lead to some degree of depolarization.

Hawkins and Ahmad (2016) leverage this difference to provide an interesting explanation of the essence of predictive coding: the weakening of upward signals when predictions

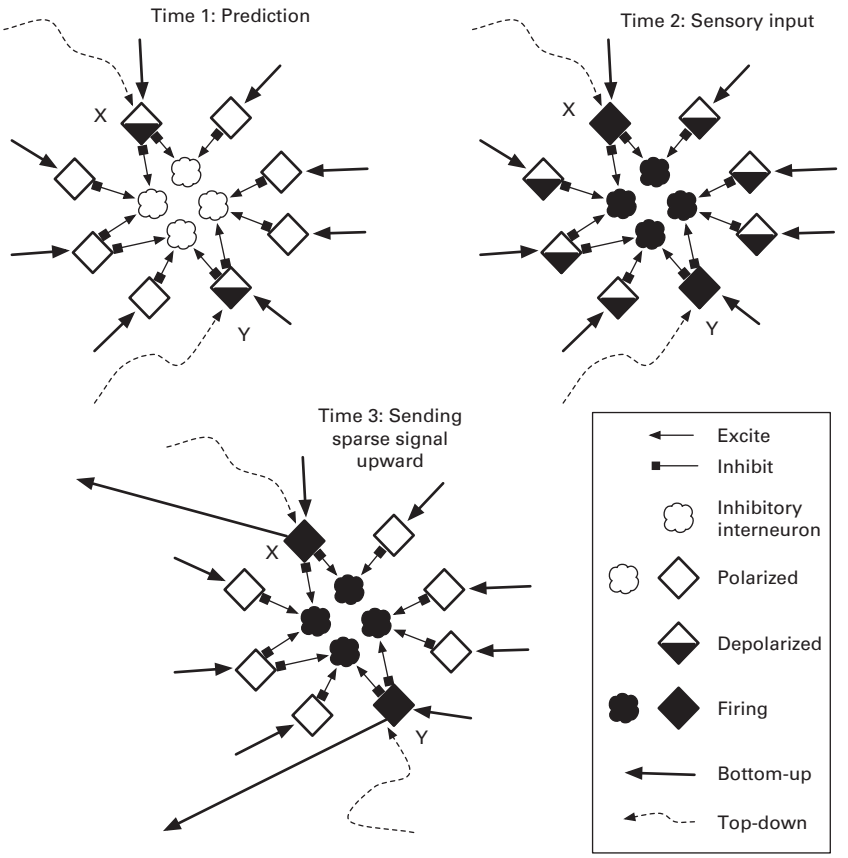


Figure 5.9
Key mechanism underlying predictive coding as hypothesized by Hawkins and Ahmad (2016). Time 1: Predictive signals from a higher level weakly excite (depolarize) a few level-4 excitatory neurons (diamonds). Time 2: Sensory inputs from a lower level begin exciting neurons X, Y, and their neighbors. X and Y begin firing before their neighbors, thereby exciting the inhibitory neurons, which repolarized the neighbors, leaving only X and Y active enough to send their signals upward at Time 3.

match sensory signals. As shown in figure 5.9, the predictive signals coming from a higher level via layer-1 dendrites provide only weak stimulation to a few excitatory level-4 neurons. This depolarizes those cells to some degree, but not enough to invoke action potentials. However, as the bottom-up sensory signals begin coming in, these cells become fully depolarized ahead of their neighbors. The ensuing action potentials then activate neighboring inhibitory neurons (e.g., basket cells), which begin suppressing nearby neurons such that many of them never reach firing thresholds, as classic winner-take-all lateral inhibition. This sparse population of *winners* then sends signals upward via layer 2/3. In the absence of a biasing predictive signal, the set of winner cells could be much larger, thus requiring higher bandwidth to continue propagating the bottom-up signal.

One key detail abstracted away from figure 5.9 is the relationship between layer-4 excitatory cells. These are often spiny stellate cells (not pyramidals) which have short axons but which tend to excite many of their neighbors. Hence, each diamond in the figure represents a small population of mutually stimulating spiny stellate cells. Once activated to

firing thresholds, these cells would autocatalytically sustain one another despite inhibitory interference, whereas the slower-charging stellates would never reach that level, remain vulnerable to inhibition, and never become an active cell assembly.

Another, equally plausible, explanation focuses on inhibition at level 2/3. If the predictive signal activates pyramidals in layer 2/3 (again via the layer-1 dendritic mat), the prospects for depolarizing those neurons to threshold are better than for level-4 neurons, since the synapses into the dendritic mat should be closer to the layer-2/3 somas (than to layer-4 somas). Assuming that some of these level-2/3 neurons begin firing, their assemblies may remain active for 15–20 msec before succumbing to inhibition for 100–150 msec. During that inhibitory period, many level-4 stellate assemblies could be driven to firing by bottom-up signals. These level-4 signals must then go through layer 2/3 on their journey upward, but all routes through the inhibited layer-2/3 neurons will be blocked. Thus, only those level-4 assemblies *not* matched by a predictive signal could continue up the hierarchy, assuming that their corresponding neighborhoods in level 2/3 reside far enough away from the active pockets of inhibition. This further illustrates the advantages of pattern sparsification: a sparse sensory signal can more easily slip past (completely intact) the inhibitory zones induced by an erroneous prediction.

In general, one important fact should be kept in mind when discussing theories of cortical columns and their functionality: size. The neuroscience literature (Mountcastle 1998; Hawkins 2004; Hawkins, Ahmad, and Cui 2017; Amit 2003; Rakic 2008) varies on the estimates, but the primate brain probably contains on the order of 10^5–10^6 columns, with each consisting of a few hundred mini columns, each, in turn, housing around one hundred, tightly interconnected neurons representing a majority of the main neural cell types. Hence, a cortical column consists of several tens of thousands of diverse neural cells. The neurons in each minicolumn tend to have highly correlated firing patterns, while a full column appears to handle a particular function associated with a receptive field, motor region, or cognitive task. The combination of thousands of diverse units provides plenty of computational machinery for realizing a broad spectrum of these functions. Theories abound as to their identity, but few should be ruled out simply on the grounds of complexity, since each column seems to *have the numbers* and diversity to perform any of a wide range of calculations, with variables involving considerable spatial and temporal scope. The basic mathematical operations discussed in earlier chapters, including those of a PID controller, could easily be handled by this cellular machinery.

5.2.1 Learning Proper Predictions

Descriptions of predictive coding often assume preexisting neural circuitry that links representations of *causes* at one level with those of predicted *effects* in the level below, possibly with a brief nod to Hebbian mechanisms as the underlying generative principle. Unfortunately, there is no consensus explanation of how these circuits emerge through a combination of development and synaptic tuning. What follows is one possible explanation based on several published theories (Downing 2009, 2015; Hawkins 2004; Rodriguez, Whitson, and Granger 2004; Wallenstein, Eichenbaum, and Hasselmo 1998).

As expected, the story begins with Donald Hebb and neurons that *fire together, wire together* (Hebb 1949; Fregnac 2003), but details of spike-timing-dependent plasticity (STDP) also come into play. As summarized by Song and colleagues, the firing-time

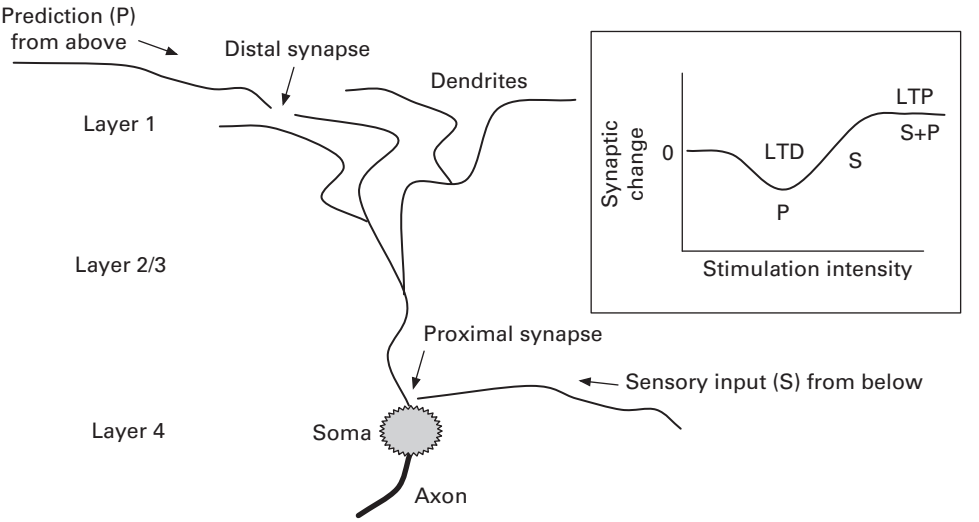


Figure 5.10
Basic neurocytology in the cortical column, layer 4 (aka the granular layer), with afferents from lower levels (e.g., sensory areas) synapsing proximally (i.e., close to the soma) and top-down predictive signals synapsing distally (i.e., far from the soma) in the dendritic mat of layer 1. (Inset) Differences in synaptic change for low- versus high-stimulation scenarios, where weak input (in the form of an unmatched prediction) leads to depression (LTD), while coincidental predictive and sensory input creates sufficient activation to produce synaptic potentiation (LTP).

relationships between presynaptic neuron A and postsynaptic neuron B play a critical role in synaptic modification (Song, Miller, and Abbott 2000). If A's spikes occur just before B's, then the $A \rightarrow B$ synapse potentiates (i.e., strengthens), but if B spikes before A, that synapse depresses (i.e., weakens). Related work by Artola, Brocher, and Singer (1990) shows that a postsynaptic neuron experiencing high stimulation will potentiate its afferent synapses (those associated with its dendrites), while weak stimulation tends to depress those same synapses. This creates interesting dynamics in cortical columns.

Looking at the layer-4 neuron in figure 5.10, note that afferents from lower levels (e.g., sensory inputs) tend to synapse very close to the soma, whereas afferents from higher levels (carrying predictive signals) synapse distally in the dendritic mat of layer 1. Action potentials arriving at these distal synapses have relatively long distances to travel, and hence attenuate before reaching the soma. Thus, whereas bottom-up signals can *drive* a layer-4 cell, top-down signals, on their own, can provide only weak stimulation, at least until the distal synapse has been enhanced by LTP.

As the inset of figure 5.10 indicates, the weak signal provided by an isolated predictive signal (P) may produce LTD, as implied by Artola, Brocher, and Singer (1990), but the combination of sensory input (S) and P should stimulate the soma enough to induce LTP at both the proximal and distal synapses. If potentiated enough, the distal synapses should eventually be capable of delivering strong predictive signals to the soma, thus producing considerable spiking activity even in the absence of the bottom-up signal. Conversely, a predictive signal that goes unmatched by a sensory signal may produce LTD and thus reduce the future influence of top-down signals. These mechanisms could play a crucial role in the formation of predictive circuits.

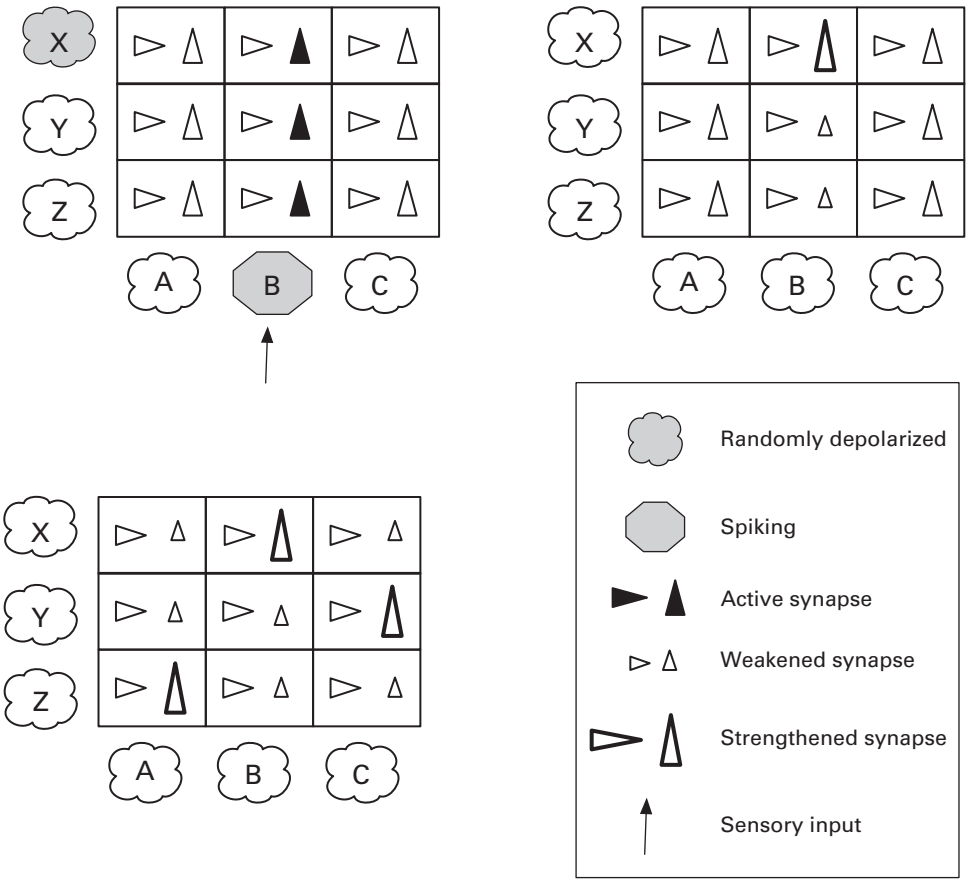


Figure 5.11
Basic mechanism by which bottom-up connections can form during early learning and development. Neurons A,B, and C are one level (L1) below neurons X,Y, and Z (level L2). Bottom-up connections denoted by vertical triangles/arrows; top-down connections by horizontal triangles/arrows. Initially, all interlevel connections exist and have similar efficacies. (Upper left) Sensory input stimulates B, which activates all B → L2 synapses (dark vertical arrows), while X randomly depolarizes (due to typical random activity during development). (Upper right) Coincident activity of B and X leads to long-term potentiation (LTP) of the $B \rightarrow X$ synapse, while $B \rightarrow Y$ and $B \rightarrow Z$ weaken via long-term depression (LTD) due to active presynaptic but inactive postsynaptic neurons. (Lower left) Similar combinations of sensory input to A (and C) and tonic activity of Z (and Y) produce two more biased connections: $A \rightarrow Z$ and $C \rightarrow Y$.

Although the descriptions that follow involve individual neurons, this is primarily for illustrative purposes. The true story surely involves neural assemblies, and each of the units discussed below should probably be interpreted as a collection of cooperating cells.

Two different pathways need to be established: the driving circuits from lower to higher levels, and the predictive networks from higher to lower levels. Figure 5.11 gives a procedure for tuning the driving synapses from level L1 to L2. Two important factors are (1) an initial network in which most L1 neurons connect (via proximal synapses) to most L2 neurons, and (2) the random depolarization of L2 neurons, a common event in the brain, particularly during development. Thus, an L2 neuron (X in the figure) that randomly activates slightly

after an L1 neuron (B) will promote LTP on the proximal $B \rightarrow X$ synapse. In addition, the L2 neurons (Y and Z) that fail to activate in that same time frame will experience depression on their synapses from B.

In Hebbian learning, presynaptic firing that is not followed by postsynaptic activity also leads to synaptic depression. During this phase, the activity in L2 may involve depolarization without actual spiking, such that the synapses from L2 to L1 are not affected. However, once the bottom-up proximal connections become fortified by this first phase, future activation of neurons such as B will suffice to depolarize X up to its spiking threshold.

In phase 2 (see figure 5.12), the L2 neurons can now produce significant action potentials and exert some influence on L1. However, the distal nature of all connections from L2 to L1 presents challenges for any predictive signal, which must first traverse the unrefined synapses of the layer-1 dendritic mat. This is where Artola and colleague's findings come into play. When an L2 neuron (e.g., Z) sends an action potential across a layer-1 synapse and down to a layer-4 neuron (e.g., B) in the same 20–40 msec time window that B receives a driving signal from a lower level, B will fire intensely and the $Z \rightarrow B$ synapse (in the dendritic mat) should experience LTP, thus forming a strong link between Z and B. When Z fires, it will also send signals across the $Z \rightarrow A$ and $Z \rightarrow C$ synapses, but since neither A nor C receives a bottom-up signal in the same short window, A and C will be only weakly stimulated, leading to depression of the $Z \rightarrow A$ and $Z \rightarrow C$ synapses. In this way, Z becomes a predictor of B. Anthropomorphically speaking, Z *searches* for matching bottom-up signals by trying all post-synaptic possibilities and then potentiating where it finds matches and depressing where it does not.

Refinement of the top-down synapses proceeds as explained in figure 5.12, with the L2 neurons serving as links between neurons A, B, and C in L1. Although seemingly superfluous in this simple example, these higher-level intermediaries become vital when A, B, and C actually represent cell assemblies in a shared cortical region. Creating strong intralevel links between many members of distinct assemblies could easily create overconnected levels in which almost all neurons could become simultaneously active, thus destroying the information-encoding capacity of that level. Adding the extra level provides useful time delays and dedicated high-level detectors for and stimulators of individual assemblies at the lower level.

Figures 5.11 and 5.12 show the formation of a network for remembering the sequence A-B-C, but how does this scale up to hierarchies in which patterns at upper levels represent abstractions of lower-level patterns? The key lies in timescale differences across the neural hierarchy: higher cortical regions run at slower timescales that lower regions (Raut, Snyder, and Raichle 2020). Thus, when a lower-level driving signal activates a higher-level neuron/assembly, the latter remains active (high spiking) longer than the former. This would allow the higher-level pattern to bind to several lower-level patterns.

Returning to figure 5.12, after neuron/assembly A activates Z, the latter may remain active throughout the bottom-up stimulation of B, C, and several other units/assemblies. Thus, in the future, Z would trigger on A but then predict B, C, and so on, thereby representing chunked sequences. The other intermediate neurons (X and Y), though originally molded as detectors of B and C, become superfluous (and thus recyclable as detectors for other patterns). When Z predicts B and C, the matching of expectations to reality (the bottom-up signals for B and C) reduces the upward signaling (as described earlier and shown in

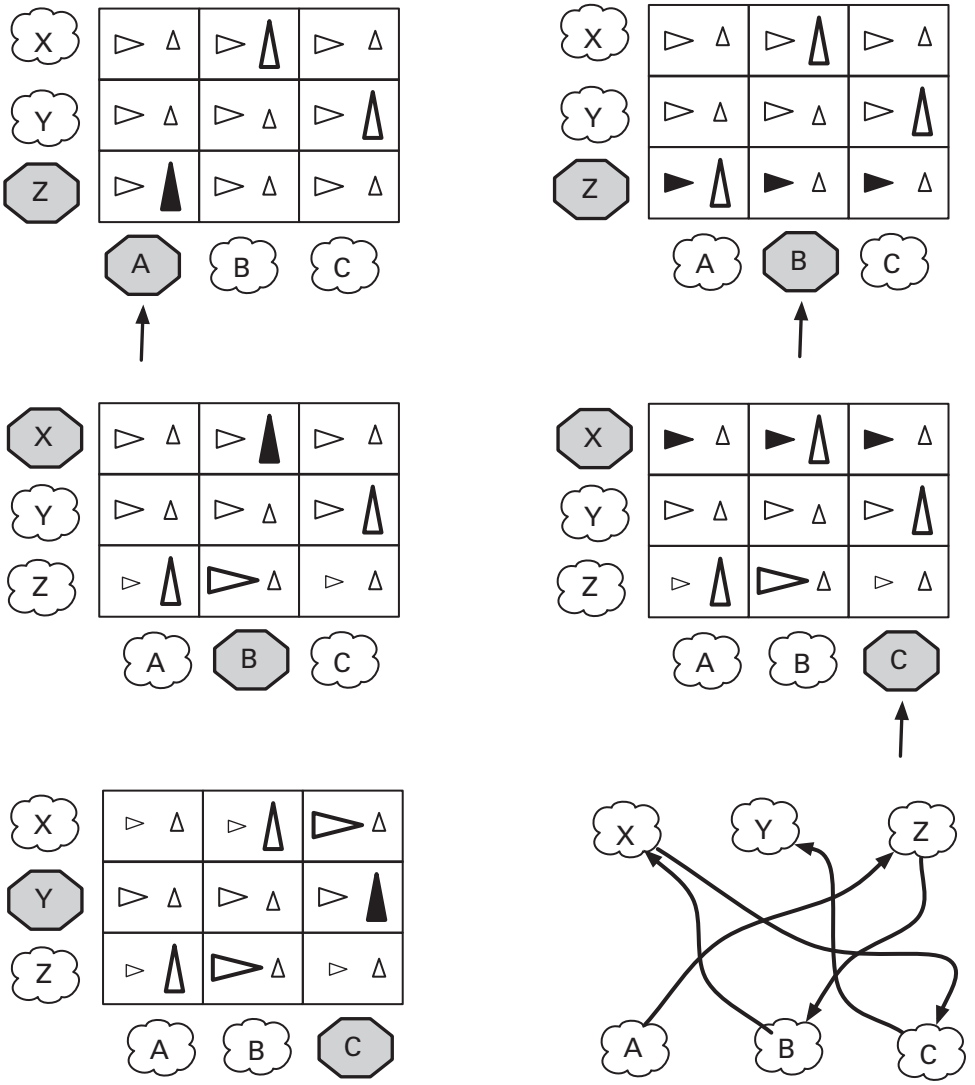


Figure 5.12
Basic mechanism for refining top-down connections during learning, with the same legend as in figure 5.11. (Top Left) L1 neuron A receives bottom-up stimulation on a proximal synapse, thus sending an action potential across the $A \rightarrow Z$ synapse and exciting neuron Z. (Top Right) Z's firing sends signals across all three synapses to L1 in the same short time window that neuron B receives a driving, bottom-up signal. (Middle Left) This strengthens the $Z \rightarrow B$ synapse but weakens those to A and C (neither of which receive driving signals in this time window). Activated B then stimulates X via the previously enhanced $B \rightarrow X$ synapse. (Middle Right) Spiking X sends action potentials down all synapses to L1 at the same time as a driving bottom-up signal reaches neuron C. (Bottom Left) This potentiates the $X \rightarrow C$ synapse but depresses $X \rightarrow A$ and $X \rightarrow B$. (Bottom Right) Summary of the strong synapses that realize the activation chain: $A \mapsto Z \mapsto B \mapsto X \mapsto C \mapsto Y$.

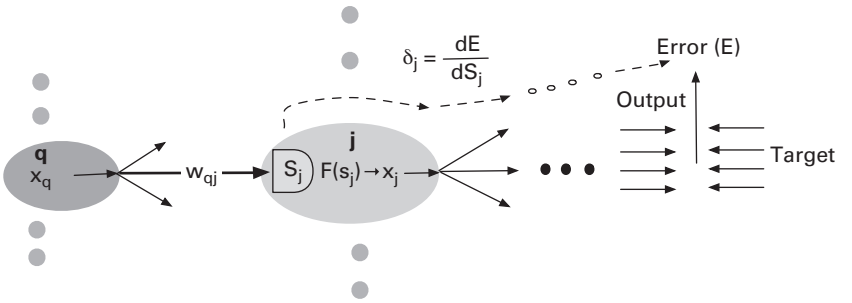


Figure 5.13
Basic relationships among neurons (gray ovals), the sum of their weighted inputs (S), their activation function (F) and output (x), and their δ gradient, $\frac{\partial E}{\partial S_j}$, which links the sum of weighted inputs S_j of any node (j) to the error (E) in the output layer, which may be many layers downstream from node j.

figure 5.9) such that X and Y no longer become activated. In this way, neurons at higher levels gradually adapt to become abstractions (aggregates) of lower-level details.

5.3 Predictive Coding for Machine Learning

Artificial learning systems for both simple and relatively complex tasks do not require the complexity of STDP and cortical columns, as contemporary machine learning (ML) successes have clearly shown. However, the essence of predictive coding provides a more biologically plausible (and still highly effective) alternative to ML's classic supervised learning tool: the backpropagation network. To understand the potential contributions of predictive coding to ML, a brief review of the backpropagation algorithm is in order.

5.3.1 The Backpropagation Algorithm

At the core of the algorithm are gradients, many of them of the long-distance variety, expressing the influence of various system variables on the objective function (aka cost or error function). For standard backprop calculations, the two main types of gradients are $\frac{\partial E}{\partial w}$ and $\frac{\partial E}{\partial S}$. The former expresses the effect of a synaptic weight on the output error, while the latter denotes the effect of the sum-of-weighted-inputs (S) to a node on that same error. The former is often calculated from the latter, which mainly serves as an intermediate variable, δ, but which supplies the key link to predictive coding (described below).

Figure 5.13 provides many of the important ties between variables, with node j as the focal point. Note that its sum of weighted inputs, S_j passes through an activation function (F) to produce the node's output value (x_j). Further note that upstream neighbor node q sends its output to node j along the connection with weight (strength) w_{qj}. Thus, two simple calculations reveal that

$$\frac{\partial S_j}{\partial w_{qj}} = x_q \tag{5.1}$$

and

$$\frac{\partial S_j}{\partial x_q} = w_{qj} \tag{5.2}$$

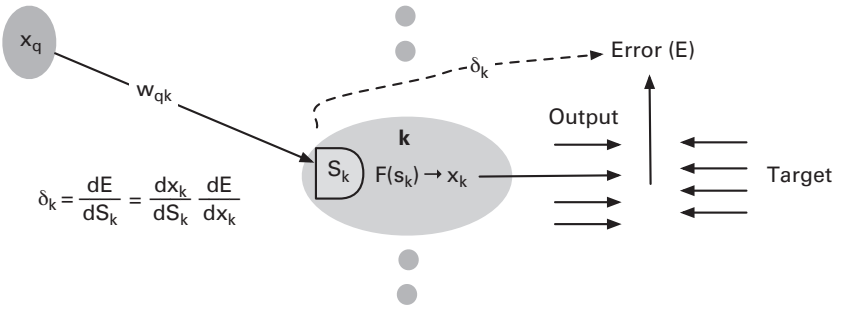


Figure 5.14
Computation of the influence of an output node's sum-of-weighted-inputs (S_k) on the error (E): $\delta_k = \frac{\partial E}{\partial S_k}$.

The derivative of this sum with respect to both the weight and the upstream output are important parts of the gradient chain.

Given δ_j for any node j, the chain rule of calculus easily produces $\frac{\partial E}{\partial w_{ij}}$ for any upstream neighbor node i:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial S_j}{\partial w_{ij}} \frac{\partial E}{\partial S_j} = x_i \delta_j \tag{5.3}$$

And this gradient provides the essential information for intelligent weight change using the standard update rule (with λ as the learning rate):

$$\triangle w_{ij} = -\lambda \frac{\partial E}{\partial w_{ij}} \tag{5.4}$$

In short, given a node's δ gradient, the gradients for all incoming weights are trivially computed. Calculating the δ gradients themselves is a bit more complicated. The simplest case is on the output end of the network, as illustrated in figure 5.14, where the link from S_k to error involves one intermediate term, x_k, the output of neuron k, also a network output.

Adding a little more detail, assume a standard mean-squared error function:

$$E = \frac{1}{2} \sum_i (T_i - x_i)^2 \tag{5.5}$$

where T_i is the target value for the ith output neuron. Then, for any particular output neuron, k:

$$\frac{\partial E}{\partial x_k} = -(T_k - x_k) \tag{5.6}$$

and hence:

$$\delta_k = \frac{\partial E}{\partial S_k} = \frac{\partial x_k}{\partial S_k} \frac{\partial E}{\partial x_k} = -F'(S_k)(T_k - x_k) \tag{5.7}$$

Thus, the gradient for weight w_{qk} on the connection from upstream neighbor node q to node k (in figure 5.14) is

$$\frac{\partial E}{\partial w_{qk}} = x_q \delta_k = -x_q F'(S_k)(T_k - x_k) \tag{5.8}$$

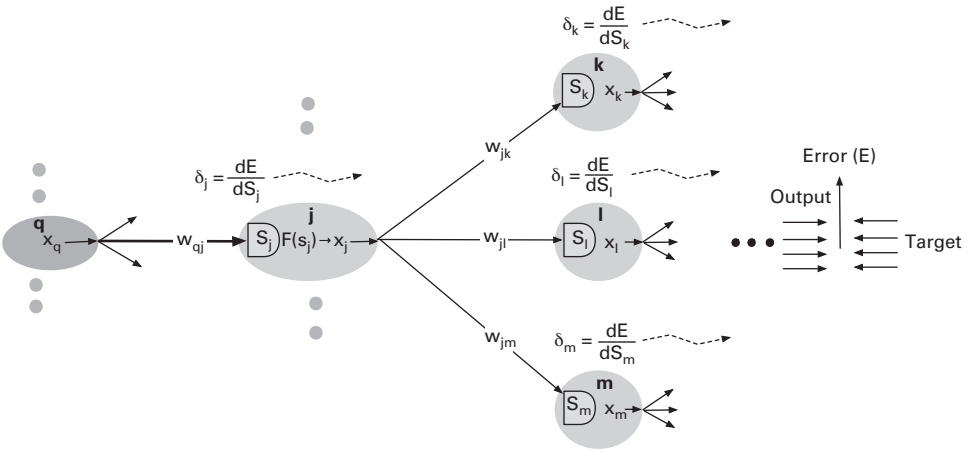


Figure 5.15
Illustration of the recursive relationship between the δ gradients across multiple layers of a neural network. As expressed in equation 5.11, the gradient at node j is the derivative of the activation function, $F'(s_j)$, multiplied by the sum of the weight-modified δ gradients of all immediate downstream neighbors.

And the weight update becomes[1]

$$\triangle w_{qk} = -\lambda \frac{\partial E}{\partial w_{qk}} = \lambda x_q F'(S_k)(T_k - x_k) \tag{5.9}$$

In networks with one or more hidden layers, the calculation of the non-output δ gradients requires recursion, wherein the gradient for one node depends on the gradients of all of its downstream neighbors. Figure 5.15 portrays that section of a multilayered network most relevant for computing the δ gradient for node j, in the middle of the diagram.

The chain rule breaks this down into two derivatives, one simple and one more complex:

$$\delta_j = \frac{\partial E}{\partial S_j} = \frac{\partial x_j}{\partial S_j}\frac{\partial E}{\partial x_j} = F'(S_j)\frac{\partial E}{\partial x_j} \tag{5.10}$$

Finding $\frac{\partial E}{\partial x_j}$ requires a summation over all paths taken by x_j in the network:

$$\begin{aligned}\delta_j = F'(S_j)\frac{\partial E}{\partial x_j} &= F'(S_j)\left[\frac{\partial S_k}{\partial x_j}\frac{\partial E}{\partial S_k} + \frac{\partial S_l}{\partial x_j}\frac{\partial E}{\partial S_l} + \frac{\partial S_m}{\partial x_j}\frac{\partial E}{\partial S_m}\right] \\ &= F'(S_j)\left[w_{jk}\delta_k + w_{jl}\delta_l + w_{jm}\delta_m\right]\end{aligned} \tag{5.11}$$

And thus the update for incoming weight w_{qj} is

$$\triangle w_{qj} = -\lambda \frac{\partial E}{\partial w_{qj}} = -\lambda x_q \delta_j = -\lambda x_q F'(S_j)\left[w_{jk}\delta_k + w_{jl}\delta_l + w_{jm}\delta_m\right] \tag{5.12}$$

Note that the intermediate expression above,

$$\triangle w_{qj} = -\lambda x_q \delta_j \tag{5.13}$$

hints of a Hebbian or anti-Hebbian update rule, assuming that δ_j constitutes a local property of node j. This locality is definitely not the default case with standard backpropagation, since

the δ gradient often represents the accumulation of many layers of downstream gradients, all with the same timestamp corresponding to the current input-output case. However, as shown below, the prediction error inherent in predictive coding is a local variable that behaves very similar to the δ gradient.

The name *backpropagation* stems from this recursive transfer of δ gradients backward, from output to input layer. The complete algorithm consists of a forward phase, where inputs are propagated through the net to the output layer, and a backward phase, where gradients and then weights are updated. The forward phase has no conflicts with neuroscience, and none of the individual calculations of the backward phase (e.g., gradient derivations) seems particularly difficult for a brain. But the caching of numerous activation values and the lockstep reverse traversal of the network (enabling recursion) has no neuroscientific basis. The brain is full of backward connections, but they are hardly symmetric to the forward links, either in presence or in synaptic strength (weight). Hence, any weights that modulate top-down (backward-passed) information in the brain will not precisely mirror the weights used during bottom-up (forward) processing along that same pathway, as backpropagation's calculations require. In general, the complete process of the backward phase requires a large stretch of the imagination to put in a biological context.

5.3.2 Backpropagation via Predictive Coding

Predictive coding circumvents these problems by essentially employing signals analogous to δ gradients during all phases of operation, producing continuous streams of that information between neighboring neurons, which can then use it to update both their activations and weights. These δ's embody prediction errors, denoted here as ξ's, which gradually accumulate influence from more distant network regions during the transition to equilibrium that characterizes normal activity in a predictive-coding network. Although not mathematically equivalent to δ gradients, the ξ's tend to converge to similar values (Whittington and Bogacz 2017) when performing supervised learning.

Figure 5.16 displays a general predictive-coding network similar to that of Rao and Ballard (1999), although their model assumes that each weight matrix (U) on a bottom-up connection is the transpose of the matrix W on the corresponding top-down link. To understand the relationship between backpropagation and predictive coding, it helps to simplify this diagram by removing the prediction boxes and feeding the weight-adjusted targets directly into the error nodes of their lower neighbor. The simplified model (see figure 5.17) accentuates the short negative-feedback (i.e., stabilizing) loops (a) between the error term and target of each layer, and (b) between the target of one layer and the error of the layer below. The following explanation of predictive coding based on figure 5.17 closely mirrors those given by Wittington and Bogacz in two detailed research papers on the topic (2017, 2019).

Target nodes act as integrators in predictive coding, whereas error terms represent the current difference between those targets and top-down predictions. Hence, the following equations express the dynamics of each:

$$\frac{\partial x^k}{\partial t} = U^k \xi^{k-1} - \xi^k \tag{5.14}$$

$$\xi^k = x^k - \underbrace{W^k x^{k+1}}_{prediction} \tag{5.15}$$

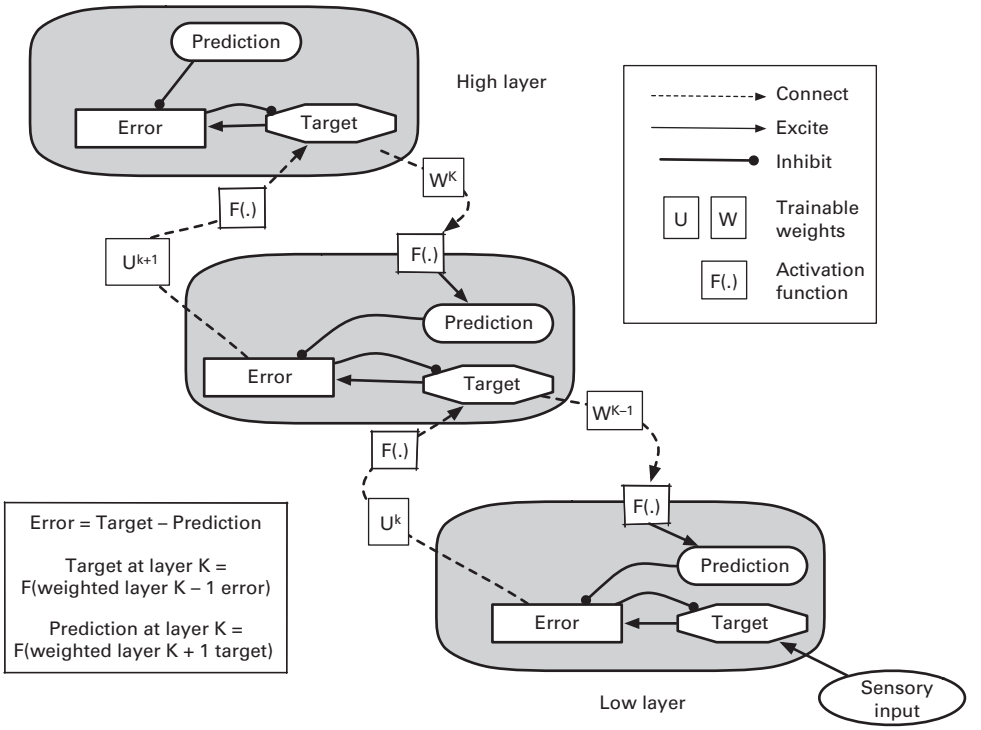


Figure 5.16
Detailed model of a predictive-coding network.

Now, consider the network of figure 5.17 running in both an unclamped (predictive or dreaming) mode and a clamped (recognition) mode. During predictive runs, no sensory input exists, thus leaving x^{k-1} (e.g., x^0) unconstrained from below. This lack of bottom-up restrictions combines with the negative feedback loop between ξ^{k-1} (e.g., ξ^0) and x^{k-1} to enable ξ^{k-1} to alter x^{k-1} so as to achieve a balance between x^{k-1} and the prediction entering ξ^{k-1} from above. In short, it allows the lowest-level prediction error to approach zero. This, in turn, essentially eliminates the bottom-up constraint of ξ^{k-1} on x^k, the target at the next layer, and this allows ξ^k to approach zero. Of course, doing so causes changes to x^k, which modifies ξ^{k-1}, but this lowest-level error node always has free rein over x^{k-1} and can alter it to push ξ^{k-1} back toward zero, thus *absorbing* any predictive perturbations from above.

This same process gradually propagates up the layer hierarchy, with corresponding adjustments filtering downward. Eventually, the prediction errors at all levels approach zero, and thus the left side of equation 5.15 vanishes for all k. This yields

$$x^k = W^k x^{k+1} \tag{5.16}$$

which is the standard relationship between the activations at one level and those at a neighboring level in a backpropagation network. For predictive coding networks using a nonlinear activation function (F), the same basic relationship holds, with equations 5.15 and 5.16 rewritten (respectively) as

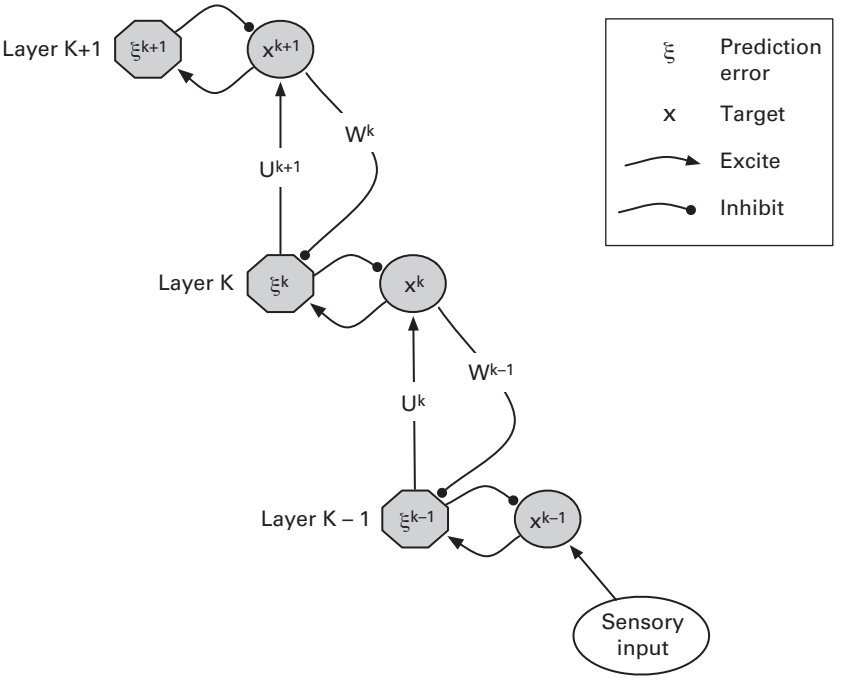


Figure 5.17
A simplification of the predictive-coding model of figure 5.16. Targets (x) at one level produce predictions at the level below when multiplied by weights W; the prediction then inhibits the error node at the lower level. Conversely, error terms (ξ) feed upward, across weights U, to excite the target node above.

$$\xi^k = x^k - \overbrace{W^k F(x^{k+1})}^{prediction} \tag{5.17}$$

$$x^k = W^k F(x^{k+1}) \tag{5.18}$$

Since all of the errors approach zero during predictive mode, their relationships are uninteresting. However, during clamped / recognition mode, all of the target vectors have dueling constraints and are not guaranteed to find equilibrium values that remove all error. Instead, the only reliable assumption is that the network will attain some degree of equilibrium, meaning that the left side of equation 5.14 vanishes, leaving the following relationship among the error vectors at neighboring layers:

$$\xi^k = U^k \xi^{k-1} \tag{5.19}$$

Taken together, equations 5.16 and 5.19 characterize a network that propagates activations in one direction while propagating errors in the opposite direction. A comparison of equations 5.19 and 5.11 reveals a key similarity between δ gradients and prediction errors (ξ): both manifest recursive properties by accumulating weighted versions of their neighboring counterparts.

Hence, predictive coding mirrors backpropagation in both (a) the dual flow directions for driving signals and error feedback, and (b) the recursive relationship between gradient

(error) signals. Furthermore, note in equation 5.7 that the δ gradient at the output level is very similar to the output error. Thus, both backpropagation and predictive coding begin with similar types of values, which they then recursively combine backward through the network. The main difference between the algorithms is procedural: backpropagation performs paired forward and backward phases, with weight updates after each pair, whereas predictive coding runs with or without external targets through any combination of forward and backward propagation waves until reaching equilibrium, at which point weight updates occur.

Another convincing similarity stems from the weight updates. Predictive coding employs these simple Hebbian rules[2] (with learning rate λ):

$$\triangle U^k = -\lambda \xi^{k-1} x^k \tag{5.20}$$

$$\triangle W^k = -\lambda x^{k+1} \xi^k \tag{5.21}$$

Compare these to the update rule for backpropagation in equation 5.13, while keeping in mind the strong similarities between δ and ξ, and the two learning algorithms seem nearly identical. The big difference is that the δ gradient consists of a string of products and sums of *recent* activations, weights, and activation-function derivatives spread across large stretches of the network space, whereas the prediction error (ξ) represents an equilibrium value attained over time and also influenced by values across the network.

Predictive coding maintains locality by having explicit error neurons in each layer, but since these neurons also encode sensory and/or predictive signals, they are a natural part of the network. Conversely, δ gradients in backpropagation serve exclusively as complementary computational scaffolding for learning. When it comes to biological plausibility, that entire scaffolding network comes into serious question, as does the process needed to run it. Predictive coding, on the other hand, abides by a very emergent, biological mechanism: widespread bottom-up and top-down interactions leading to briefly stable activation patterns, with the main *missing biological link* being the existence of error nodes. Although nonconclusive, several neuroscientific studies (summarized by Kok and Lange 2015) do indicate the presence of distinct neuron groups that encode prediction error.

Two of the main researchers to reconcile predictive coding and backpropagation, James Whittington and Rafal Bogacz (2017, 2019), have achieved very high performance on supervised-learning benchmark data sets when using predictive-coding networks. A thorough analysis of their nets in action does indeed reveal a tight similarity between the values of (a) internal target activations (x^k) versus normal feedforward activations of a backprop net, and (b) weight updates (driven by δ and ξ, respectively) in backpropagation versus predictive-coding nets. For an overview of other promising, biologically plausible, backpropagation algorithms, see Lillicrap et al. (2020).

5.4 In Theory

The logic behind predictive coding seems rather impeccable. Why would the organ that punches way above its weight in terms of energy usage[3] waste any of that wattage to shuttle around superfluous information? When a neighbor passes my office window en route to our

front door, waves, and sees me getting up and moving toward the entryway, should he knock to signal his arrival? Although common courtesy says *yes*, thermodynamics and information theory say *no, never; are you kidding me?*

Considerable neuroscientific and psychological evidence (summarized in Ouden, Kok, and Lange 2012; Spratling 2019; Clark 2016) supports both the predictive nature of many top-down signals and the use of prediction error as a common currency for bottom-up information exchange between many brain areas. These reports span the breadth of cranial faculties, from visual, auditory, and tactile perception to motricity to memory and language to motivational and cognitive control.

These myriad studies indicate that predictive coding could be prevalent throughout the nervous system. Whether or not a few common mechanisms (similar to those detailed in this chapter) can account for these diverse instantiations of the phenomenon remains debatable. The links between structure and function in the brain seem to have no desire to gratuitously share their secrets with scientists.

However, from the perspective of artificial intelligence, the sheer bulk of reputable natural-science studies in this area should motivate a deeper inquiry into the possibilities for synthetic, prediction-oriented neural networks. Fortunately, several prominent AI researchers got the message decades ago: they realized that prediction (in the form of pattern generation) plays a central role in perception, as a complement to bottom-up pattern recognition. Their neural networks were the topic of the previous chapter, while the work of Whittington and Bogacz (elaborated above) shows that predictive coding, with learning driven strictly by local gradients, can replace conventional backpropagation, at least for standard classification tasks.

Before moving on, a parting word by one of predictive coding's pioneers is in order (Attneave 1954, 192):

> The foregoing reduction principles make no pretense to exhaustiveness. It should be emphasized that there are as many kinds of redundancy in the visual field as there are kinds of regularity or lawfulness; an attempt to consider them all would be somewhat presumptuous on one hand, and almost certainly irrelevant to perceptual processes on the other. It may further be admitted that the principles which have been given are themselves highly redundant in the sense that they could be stated much more economically on a higher level of abstraction. This logical redundancy is not inadvertent, however: if one were faced with the engineering problem suggested earlier, he would undoubtedly find it necessary to break the problem down in some manner such as the foregoing, and to design a multiplicity of mechanisms to perform operations of the sort indicated.

In this final paragraph of his classic paper, Attneave speaks presciently to the nagging dilemma of every bio-inspired AI practitioner: which biological details to include, to abstract, and to ignore completely? This tension between parsimony and completeness will continue to haunt sciences of the artificial until hard evidence from the natural sciences can deliver either $e = mc^2$ for the brain or the bad news that *it takes a village* full of disparate mechanisms to produce truly general intelligence.

Of all the principles discussed in this book, predictive coding has the greatest potential of channeling neuroscience's inner Einstein.

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