1. Biological and Machine Vision

Throughout this chapter and much of this book, the visual system of biological organisms is used as an analogy to bring deep learning to, um . . . life. In addition to conveying a high-level understanding of what deep learning is, this analogy provides insight into how deep learning approaches are so powerful and so broadly applicable.

BIOLOGICAL VISION

Five hundred fifty million years ago, in the prehistoric Cambrian period, the number of species on the planet began to surge (Figure 1.1). From the fossil record, there is evidence [1] that this explosion was driven by the development of light detectors in the trilobite, a small marine animal related to modern crabs (Figure 1.2). A visual system, even a primitive one, bestows a delightful bounty of fresh capabilities. One can, for example, spot food, foes, and friendly-looking mates at some distance. Other senses, such as smell, enable animals to detect these as well, but not with the accuracy and light-speed pace of vision. Once the trilobite could see, the hypothesis goes, this set off an arms race that produced the Cambrian explosion: The trilobite's prey, as well as its predators, had to evolve to survive.

1 . Parker, A. (2004). *In the Blink of an Eye: How Vision Sparked the Big Bang of Evolution*. New York: Basic Books.

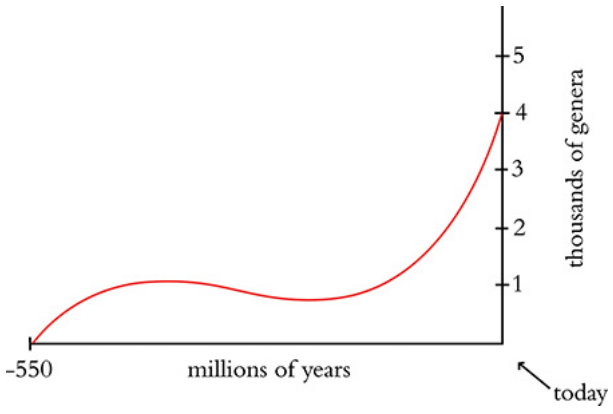

Figure 1.1 The number of species on our planet began to increase rapidly 550 million years ago, during the prehistoric Cambrian period. "Genera" are categories of related species.

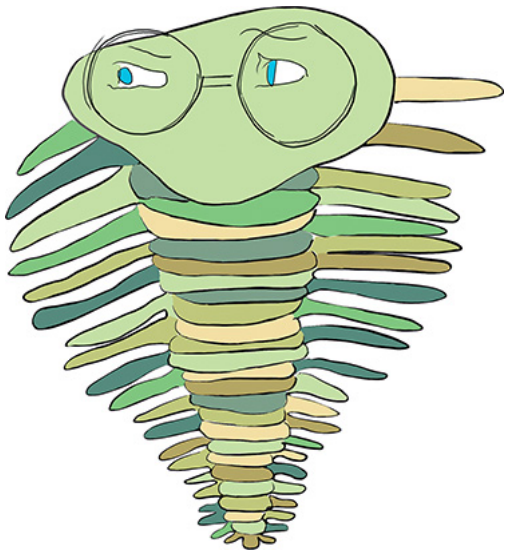

Figure 1.2 A bespectacled trilobite

In the half-billion years since trilobites developed vision, the complexity of the sense has increased considerably. Indeed, in modern mammals, a large proportion of the *cerebral cortex*—the outer gray matter of the brain—is involved in visual perception. [2] At Johns Hopkins University in the late 1950s, the physiologists David Hubel and Torsten Wiesel (Figure 1.3) began carrying out their pioneering research on how visual information is processed in the mammalian cerebral cortex, [3] work that contributed to their later being awarded a Nobel Prize. [4] As depicted in Figure 1.4, Hubel and Wiesel conducted their research by showing images to anesthetized cats while simultaneously recording the activity of individual neurons from the *primary visual cortex*, the first part of the cerebral cortex to receive visual input from the eyes.

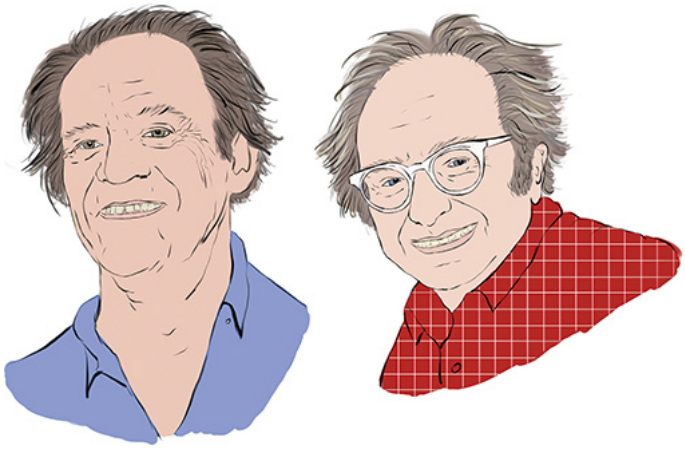

Figure 1.3 The Nobel Prize-winning neurophysiologists Torsten Wiesel (left) and David Hubel

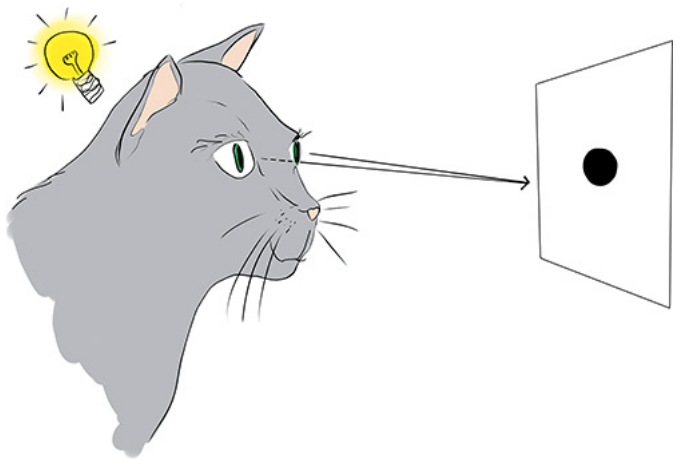

Figure 1.4 Hubel and Wiesel used a light projector to present slides to anesthetized cats while they recorded the activity of neurons in the cats' primary visual cortex. In the experiments, electrical recording equipment was implanted within the cat's skull. Instead of illustrating this, we suspected it would be a fair bit more palatable to use a lightbulb to represent neuron activation. Depicted in this figure is a primary visual cortex neuron being serendipitously activated by the straight edge of a slide.

2 . A couple of tangential facts about the cerebral cortex: First, it is one of the more recent evolutionary developments of the brain, contributing to the complexity of mammal behavior relative to the behavior of older classes of animals like reptiles and amphibians. Second, while the brain is informally referred to as *gray matter* because the cerebral cortex is the brain's external surface and this cortical tissue is gray in color, the bulk of the brain is in fact *white* matter. By and large, the white matter is responsible for carrying information over longer distances than the gray matter, so its neurons have a white-colored, fatty coating that hurries the pace of signal conduction. A coarse analogy could be to consider neurons in the white matter to act as "highways." These high-speed motorways have scant on-ramps or exits, but can transport a signal from one part of the brain to another lickety-split. In contrast, the "local roads" of gray matter facilitate myriad opportunities for interconnection between neurons at the expense of speed. A gross generalization, therefore, is to consider the cerebral cortex—the gray matter—as the part of the brain where the most complex computations happen, affording the animals with the largest proportion of it—such as mammals, particularly the great apes like *Homo sapiens*—their complex behaviors.

3 . Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology, 148*, 574–91.

4 . The 1981 Nobel Prize in Physiology or Medicine, shared with American neurobiologist Roger Sperry.

Projecting slides onto a screen, Hubel and Wiesel began by presenting simple shapes like the dot shown in Figure 1.4 to the cats. Their initial results were disheartening: Their efforts were met with no response from the neurons of the primary visual cortex. They grappled with the frustration of how these cells, which anatomically appear to be the gateway for visual information to the rest of the cerebral

cortex, would not respond to visual stimuli. Distraught, Hubel and Wiesel tried in vain to stimulate the neurons by jumping and waving their arms in front of the cat. Nothing. And then, as with many of the great discoveries, from X-rays to penicillin to the microwave oven, Hubel and Wiesel made a serendipitous observation: As they removed one of their slides from the projector, its straight edge elicited the distinctive crackle of their recording equipment to alert them that a primary visual cortex neuron was firing. Overjoyed, they celebrated up and down the Johns Hopkins laboratory corridors.

The serendipitously crackling neuron was not an anomaly. Through further experimentation, Hubel and Wiesel discovered that the neurons that receive visual input from the eye are in general most responsive to simple, straight edges. Fittingly then, they named these cells *simple* neurons.

As shown in Figure 1.5, Hubel and Wiesel determined that a given simple neuron responds optimally to an edge at a particular, specific orientation. A large group of simple neurons, with each specialized to detect a particular edge orientation, together is able to represent all 360 degrees of orientation. These edge-orientation detecting simple cells then pass along information to a large number of so-called *complex* neurons. A given complex neuron receives visual information that has already been processed by several simple cells, so it is well positioned to recombine multiple line orientations into a more complex shape like a corner or a curve.

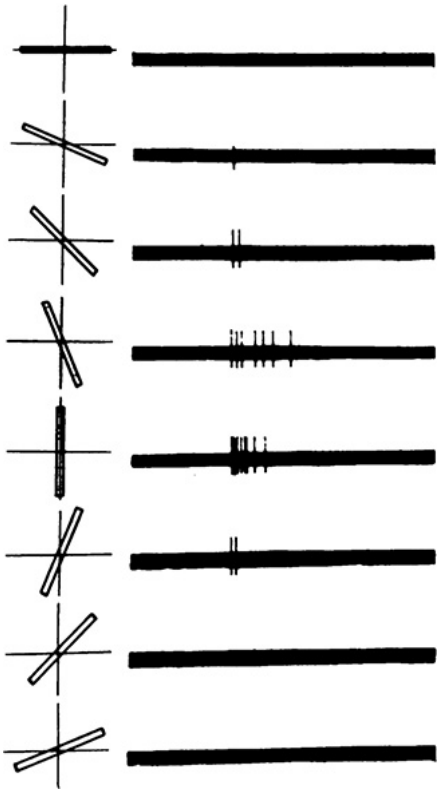

Figure 1.5 A simple cell in the primary visual cortex of a cat fires at different rates, depending on the orientation of a line shown to the cat. The orientation of the line is provided in the left-hand column, while the right-hand column shows the firing (electrical activity) in the cell over time (one second). A vertical line (in the fifth row from the top) causes the most electrical activity for this particular simple cell. Lines slightly off vertical (in the intermediate rows) cause less activity for the cell, while lines approaching horizontal (in the topmost and bottommost rows) cause little to no activity.

Figure 1.6 illustrates how, via many hierarchically organized layers of neurons feeding information into increasingly higher-order neurons, gradually more complex visual stimuli can be represented by the brain. The eyes are focused on an image of a mouse's head. Photons of light stimulate neurons located in the retina of each eye, and this raw visual information is transmitted from the eyes to the primary visual cortex of the brain. The first layer of primary visual cortex neurons to receive this input—Hubel and Wiesel's *simple cells*—are specialized to detect edges (straight lines) at specific orientations. There would be many thousands of such neurons; for simplicity, we're only showing four in Figure 1.6. These simple neurons relay information about the presence or absence of lines at particular orientations to a subsequent layer of *complex cells*, which assimilate and recombine the information, enabling the representation of more complex visual stimuli such as the curvature of the mouse's head. As information is passed through several subsequent layers, representations of visual stimuli can incrementally become more complex and more abstract. As depicted by the far-right layer of neurons, following many layers of such hierarchical processing (we use the arrow with dashed lines to imply that many more layers of processing are not being shown), the brain is ultimately able to represent visual concepts as abstract as a mouse, a cat, a bird, or a dog.

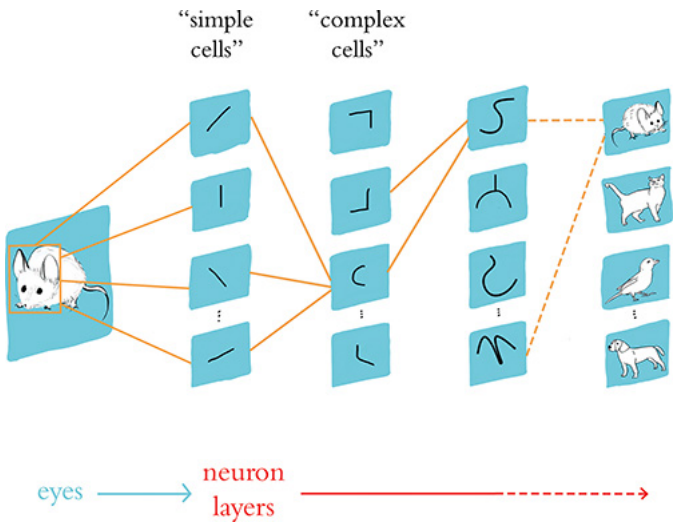

Figure 1.6 A caricature of how consecutive layers of biological neurons represent visual information in the brain of, for example, a cat or a human

Today, through countless subsequent recordings from the cortical neurons of brain-surgery patients as well as noninvasive techniques like magnetic resonance imaging (MRI), [5] neuroscientists have pieced together a fairly high-resolution map of regions that are specialized to process particular visual stimuli, such as color, motion, and faces (see Figure 1.7).

5 . Especially *functional* MRI, which provides insight into which regions of the cerebral cortex are notably active or inactive when the brain is engaged in a particular activity.

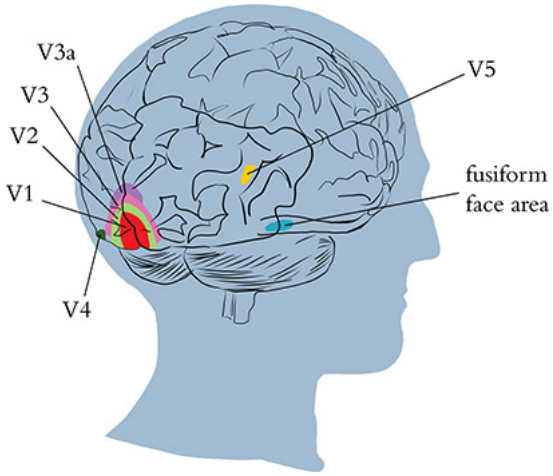

Figure 1.7 Regions of the visual cortex. The V1 region receives input from the eyes and contains the simple cells that detect edge orientations. Through the recombination of information via myriad subsequent layers of neurons (including within the V2, V3, and V3a regions), increasingly abstract visual stimuli are represented. In the human brain (shown here), there are regions containing neurons with concentrations of specializations in, for example, the detection of color (V4), motion (V5), and people's faces (fusiform face area).

MACHINE VISION

We haven't been discussing the biological visual system solely because it's interesting (though hopefully you did find the preceding section thoroughly interesting). We have covered the biological visual system primarily because it serves as the inspiration for the modern deep learning approaches to machine vision, as will become clear in this section.

Figure 1.8 provides a concise historical timeline of vision in biological organisms as well as machines. The top timeline, in blue, highlights the development of vision in trilobites as well as Hubel and Wiesel's 1959 publication on the hierarchical nature of the primary visual cortex, as covered in the preceding section. The machine vision timeline is split into two parallel streams to call attention to two alternative approaches. The middle timeline, in pink, represents the deep learning track that is the focus of our book. The bottom timeline, in purple, meanwhile represents the traditional machine learning (ML) path to vision, which—through contrast—will clarify why deep learning is distinctively powerful and revolutionary.

The Neocognitron

Inspired by Hubel and Wiesel's discovery of the simple and complex cells that form the primary visual cortex hierarchy, in the late 1970s the Japanese electrical engineer Kunihiko Fukushima proposed an analogous architecture for machine vision, which he named the *neocognitron*.[6] There are two particular items to note:

6 . Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cynbernetics, 36*, 193–202.

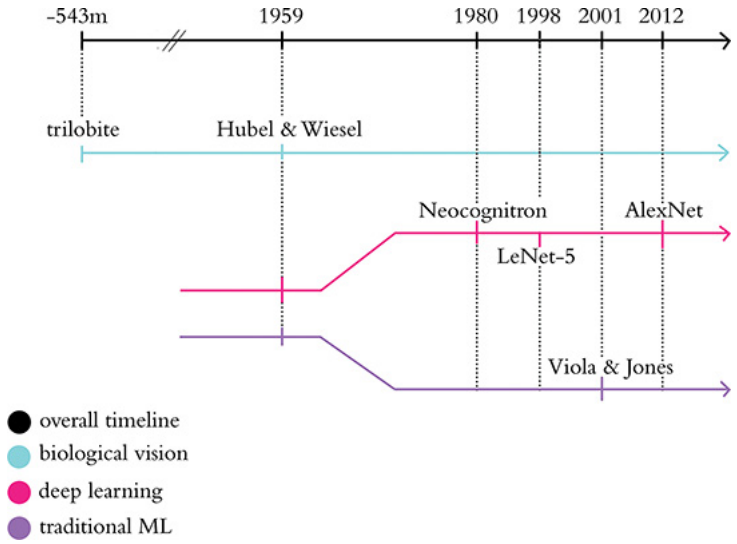

Figure 1.8 Abridged timeline of biological and machine vision, highlighting the key historical moments in the deep learning and traditional machine learning approaches to vision that are covered in this section

1. Fukushima referred to Hubel and Wiesel's work explicitly in his writing. Indeed, his paper refers to three of their landmark articles on the organization of the primary visual cortex, including borrowing their "simple" and "complex" cell language to describe the first and second layers, respectively, of his neocognitron.

2. By arranging artificial neurons [7] in this hierarchical manner, these neurons—like their biological inspiration in Figure 1.6—generally represent line orientations in the cells of the layers closest to the raw visual image, while successively deeper layers represent successively complex, successively abstract objects. To clarify this potent property of the neocognitron and its deep learning descendants, we go through an interactive example at the end of this chapter that demonstrates it. [8]

7 . We define precisely what *artificial neurons* are in Chapter 7. For the moment, it's more than sufficient to think of each artificial neuron as a speedy little algorithm.

8 . Specifically, Figure 1.19 demonstrates this hierarchy with its successively abstract representations.

LeNet-5

While the neocognitron was capable of, for example, identifying handwritten characters, [9] the accuracy and efficiency of Yann LeCun (Figure 1.9) and Yoshua Bengio's (Figure 1.10) *LeNet-5* model[10] made it a significant development. LeNet-5's hierarchical architecture (Figure 1.11) built on Fukushima's lead and the biological inspiration uncovered by Hubel and Wiesel.[11] In addition, LeCun and his colleagues benefited from superior data for training their model,[12] faster processing power, and, critically, the back-propagation algorithm.

Figure 1.9 Paris-born Yann LeCun is one of the preeminent figures in artificial neural network and deep learning research. LeCun is the founding director of the New York University Center for Data Science as well as the director of AI research at the social network Facebook.

Figure 1.10 Yoshua Bengio is another of the leading characters in artificial neural networks and deep learning. Born in France, he is a computer science professor at the University of Montreal and codirects the renowned Machines and Brains program at the Canadian Institute for Advanced Research.

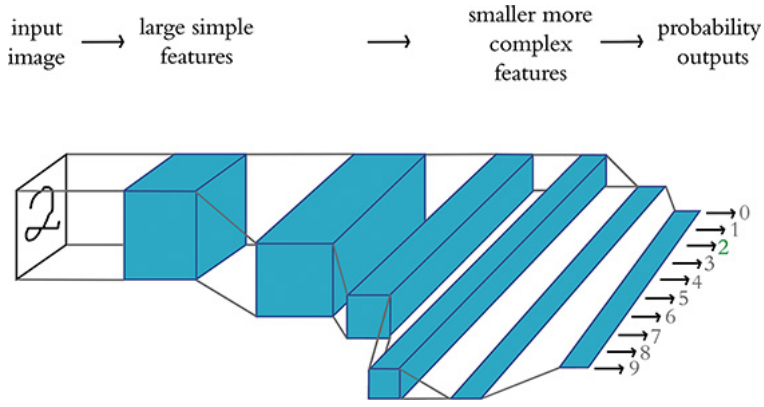

Figure 1.11 LeNet-5 retains the hierarchical architecture uncovered in the primary visual cortex by Hubel and Wiesel and leveraged by Fukushima in his neocognitron. As in those other systems, the leftmost layer represents simple edges, while successive layers represent increasingly complex features.

By processing information in this way, a handwritten “2” should, for example, be correctly recognized as the number two (highlighted by the green output shown on the right).

9 . Fukushima, K., & Wake, N. (1991). Handwritten alphanumeric character recognition by the neocognitron. *IEEE Transactions on Neural Networks, 2*, 355–65.

10. LeCun, Y., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 2*, 355–65.

11. LeNet-5 was the first *convolutional neural network,* a deep learning variant that dominates modern machine vision and that we detail in Chapter 10.

12. Their classic dataset, the handwritten MNIST digits, is used extensively in Part II, “Essential Theory Illustrated.”

Backpropagation, often abbreviated to *backprop*, facilitates efficient learning throughout the layers of artificial neurons within a deep learning model.[13] Together with the researchers’ data and processing power, backprop rendered LeNet-5 sufficiently reliable to become an early commercial application of deep learning: It was used by the United States Postal Service to automate the reading of ZIP codes[14] written on mail envelopes. In Chapter 10, on machine vision, you will experience LeNet-5 firsthand by designing it yourself and training it to recognize handwritten digits.

13. We examine the backpropagation algorithm in Chapter 7.

14. The USPS term for postal code.

In LeNet-5, Yann LeCun and his colleagues had an algorithm that could correctly predict the handwritten digits that had been drawn without needing to include any expertise about handwritten digits in their code. As such, LeNet-5 provides an opportunity to introduce a fundamental difference between deep learning and the traditional machine learning ideology. As conveyed by Figure 1.12, the traditional machine learning approach is characterized by practitioners investing the bulk of their efforts into engineering features. This *feature engineering* is the application of clever, and often elaborate, algorithms to raw data in order to preprocess the data into input variables that can be readily modeled by traditional statistical techniques. These techniques—such as regression, random forest, and support vector machine—are seldom effective on unprocessed data, and so the engineering of input data has historically been a prime focus of machine learning professionals.

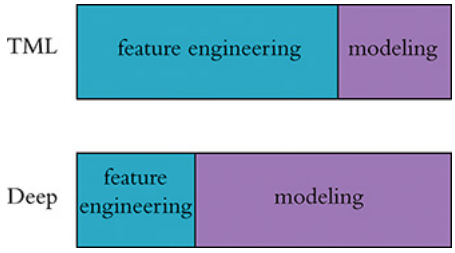

Figure 1.12 Feature engineering—the transformation of raw data into thoughtfully transformed input variables—often predominates the application of traditional machine learning algorithms. In contrast, the application of deep learning often involves little to no feature engineering, with the majority of time spent instead on the design and tuning of model architectures.

In general, a minority of the traditional ML practitioner's time is spent optimizing ML models or selecting the most effective one from those available. The deep learning approach to modeling data turns these priorities upside down. *The deep learning practitioner typically spends little to none of her time engineering features, instead spending it modeling data with various artificial neural network architectures that process the raw inputs into useful features automatically*. This distinction between deep learning and traditional machine learning is a core theme of this book. The next section provides a classic example of feature engineering to elucidate the distinction.

The Traditional Machine Learning Approach

Following LeNet-5, research into artificial neural networks, including deep learning, fell out of favor. The consensus became that the approach's automated feature generation was not pragmatic—that even though it worked well for handwritten character recognition, the feature-free ideology was perceived to have limited breadth of applicability.[15] Traditional machine learning, including its feature engineering, appeared to hold more promise, and funding shifted away from deep learning research.[16]

15. At the time, there were stumbling blocks associated with optimizing deep learning models that have since been resolved, including poor weight initializations (covered in Chapter 9), covariate shift (also in Chapter 9), and the predominance of the relatively inefficient sigmoid activation function (Chapter 6).

16. Public funding for artificial neural network research ebbed globally, with the notable exception of continued support from the Canadian federal government, enabling the Universities of Montreal, Toronto, and Alberta to become powerhouses in the field.

To make clear what feature engineering is, Figure 1.13 provides a celebrated example from Paul Viola and Michael Jones in the early 2000s.[17] Viola and Jones employed rectangular filters such as the vertical or horizontal black-and-white bars shown in the figure. Features generated by passing these filters over an image can be fed into machine learning algorithms to reliably detect the presence of a face. This work is notable because the algorithm was efficient enough to be the first real-time face detector outside the realm of biology.[18]

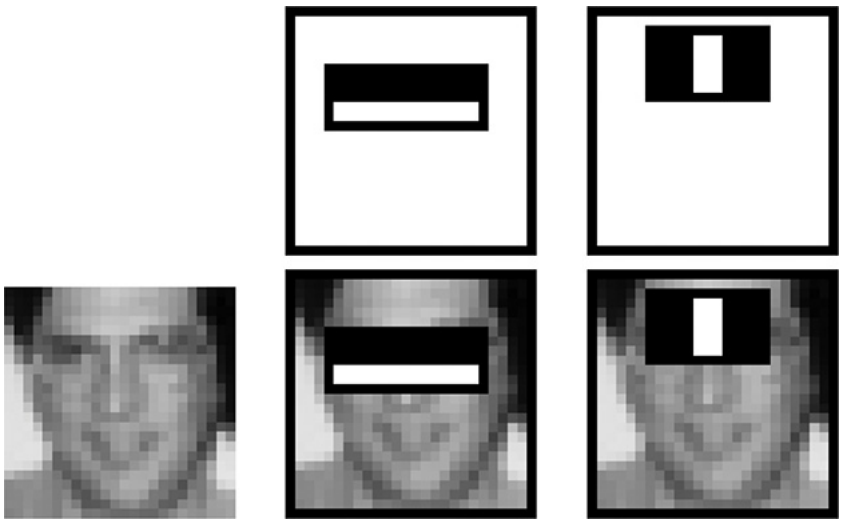

Figure 1.13 Engineered features leveraged by Viola and Jones (2001) to detect faces reliably. Their efficient algorithm found its way into Fujifilm cameras, facilitating real-time auto-focus.

17. Viola, P., & Jones, M. (2001). Robust real-time face detection. *International Journal of Computer Vision, 57*, 137–54.

18. A few years later, the algorithm found its way into digital Fujifilm cameras, facilitating autofocus on faces for the first time—a now everyday attribute of digital cameras and smartphones alike.

Devising clever face-detecting filters to process raw pixels into features for input into a machine learning model was accomplished via years of research and collaboration on the characteristics of faces. And, of course, it is limited to detecting faces in general, as opposed to being able to recognize a particular face as, say, Angela Merkel's or Oprah Winfrey's. To develop features for detecting Oprah in particular, or for detecting some non-face class of objects like houses, cars, or Yorkshire Terriers, would require the development of expertise in that category, something that could again take years of academic-community collaboration to execute both efficiently and accurately. Hmm, if only we could circumnavigate all that time and effort somehow!

ImageNet and the ILSVRC

As mentioned earlier, one of the advantages LeNet-5 had over the neocognitron was a larger, high-quality set of training data. The next breakthrough in neural networks was also facilitated by a high-quality public dataset, this time much larger. *ImageNet*, a labeled index of photographs devised by Fei-Fei Li (Figure 1.14), armed machine vision researchers with an immense catalog of training data.[19,20] For reference, the handwritten digit data used to train LeNet-5 contained tens of thousands of images. ImageNet, in contrast, contains tens of *millions*.

Figure 1.14 The hulking ImageNet dataset was the brainchild of Chinese-American computer science professor Fei-Fei Li and her colleagues at Princeton in 2009. Now a faculty member at Stanford University, Li is also the chief scientist of A.I./ML for Google's cloud platform.

19. image-net.org (http://image-net.org)

20. Deng, J., et al. (2009). ImageNet: A large-scale hierarchical image database. *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

The 14 million images in the ImageNet dataset are spread across 22,000 categories. These categories are as diverse as container ships, leopards, starfish, and elderberries. Since 2010, Li has run an open challenge called ILSVRC (the ImageNet Large Scale Visual Recognition Challenge) on a subset of the ImageNet data that has become the premier ground for assessing the world's state-of-the-art machine vision algorithms. The ILSVRC subset consists of 1.4 million images across 1,000 categories. In addition to providing a broad range of categories, many of the selected categories are breeds of dogs, thereby evaluating the algorithms' ability not only to distinguish widely varying images but also to specialize in distinguishing subtly varying ones.[21]

21. On your own time, try to distinguish photos of Yorkshire Terriers from Australian Silky Terriers. It's tough, but Westminster Dog Show judges, as well as contemporary machine vision models, can do it. Tangentially, these dog-heavy data are the reason deep learning models trained with ImageNet have a disposition toward "dreaming" about dogs (see, e.g., deepdreamgenerator.com (http://deepdreamgenerator.com)).

AlexNet

As graphed in Figure 1.15, in the first two years of the ILSVRC all algorithms entered into the competition hailed from the feature-engineering-driven traditional machine learning ideology. In the third year, all entrants *except one* were traditional ML algorithms. If that one deep learning model in 2012 had not been developed or if its creators had not competed in ILSVRC, then the year-over-year image classification accuracy would have been negligible. Instead, Alex Krizhevsky and Ilya Sutskever —working out of the University of Toronto lab led by Geoffrey Hinton (Figure 1.16)—crushed the existing benchmarks with their submission, today referred to as AlexNet (Figure 1.17).[22,23] This was a

watershed moment. In an instant, deep learning architectures emerged from the fringes of machine learning to its fore. Academics and commercial practitioners scrambled to grasp the fundamentals of artificial neural networks as well as to create software libraries—many of them open-source—to experiment with deep learning models on their own data and use cases, be they machine vision or otherwise. As Figure 1.15 illustrates, in the years since 2012 all of the top-performing models in the ILSVRC have been based on deep learning.

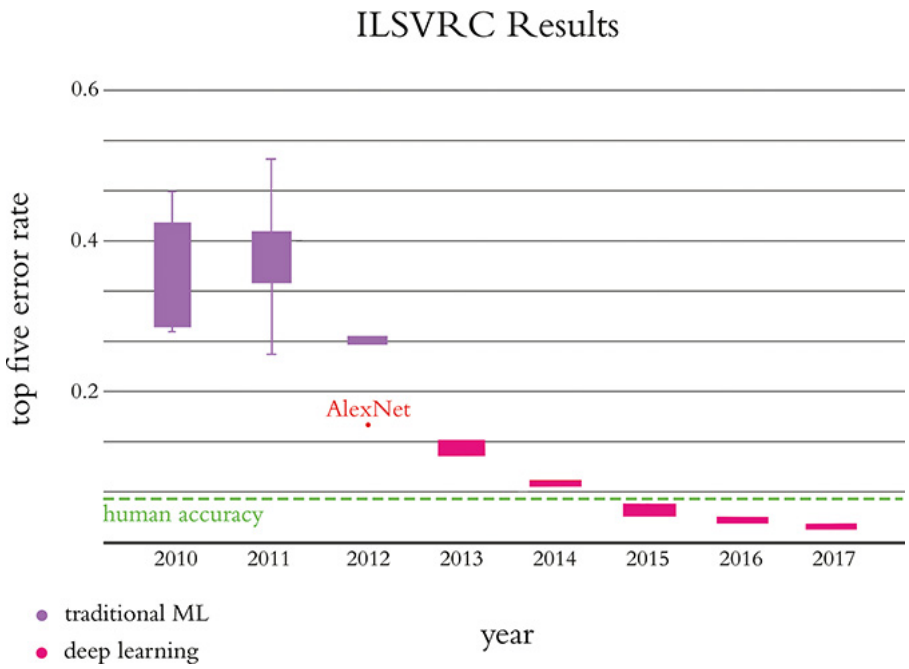

Figure 1.15 Performance of the top entrants to the ILSVRC by year. AlexNet was the victor by a head-and-shoulders (40 percent!) margin in the 2012 iteration. All of the best algorithms since then have been deep learning models. In 2015, machines surpassed human accuracy.

Figure 1.16 The eminent British-Canadian artificial neural network pioneer Geoffrey Hinton, habitually referred to as “the godfather of deep learning” in the popular press. Hinton is an emeritus professor at the University of Toronto and an engineering fellow at Google, responsible for managing the search giant’s Brain Team, a research arm, in Toronto. In 2019, Hinton, Yann LeCun (Figure 1.9), and Yoshua Bengio (Figure 1.10) were jointly recognized with the Turing Award—the highest honor in computer science—for their work on deep learning.

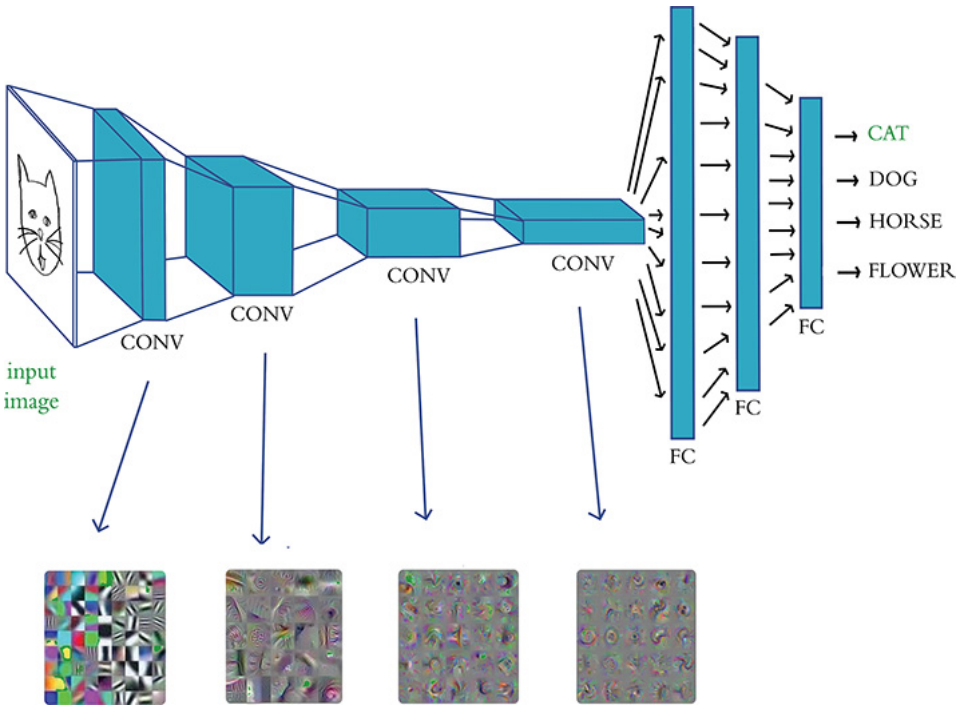

Figure 1.17 AlexNet's hierarchical architecture is reminiscent of LeNet-5, with the first (left-hand) layer representing simple visual features like edges, and deeper layers representing increasingly complex features and abstract concepts. Shown at the bottom are examples of images to which the neurons in that layer maximally respond, recalling the layers of the biological visual system in Figure 1.6 and demonstrating the hierarchical increase in visual feature complexity. In the example shown here, an image of a cat input into LeNet-5 is correctly identified as such (as implied by the green "CAT" output). "CONV" indicates the use of something called a convolutional layer, and "FC" is a fully connected layer; we formally introduce these layer types in Chapters 7 and 10, respectively.

22. Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*.

23. The images along the bottom of Figure 1.17 were obtained from Yosinski, J., et al. (2015). Understanding neural networks through deep visualization. *arXiv: 1506.06579*.

Although the hierarchical architecture of AlexNet is reminiscent of LeNet-5, there are three principal factors that enabled AlexNet to be the state-of-the-art machine vision algorithm in 2012. First is the training data. Not only did Krizhevsky and his colleagues have access to the massive ImageNet index, they also artificially expanded the data available to them by applying transformations to the training images (you, too, will do this in Chapter 10). Second is processing power. Not only had computing power per unit of cost increased dramatically from 1998 to 2012, but Krizhevsky, Hinton, and Sutskever also programmed two GPUs[24] to train their large datasets with previously unseen efficiency. Third is architectural advances. AlexNet is deeper (has more layers) than LeNet-5, and it takes advantage of both a new type of artificial neuron[25] and a nifty trick[26] that helps generalize deep learning models beyond the data they're trained on. As with LeNet-5, you will build AlexNet yourself in Chapter 10 and use it to classify images.

24. Graphical processing units: These are designed primarily for rendering video games but are well suited to performing the matrix multiplication that abounds in deep learning across hundreds of parallel computing threads.

25. The rectified linear unit (ReLU), which is introduced in Chapter 6.

26. Dropout, introduced in Chapter 9.

Our ILSVRC case study underlines why deep learning models like AlexNet are so widely useful and disruptive across industries and computational applications: They dramatically reduce the subject-matter expertise required for building highly accurate predictive models. This trend away from expertise-driven feature engineering and toward surprisingly powerful automatic-feature-generating deep learning models has been prevalently borne out across not only vision applications, but also, for example, the playing of complex games (the topic of Chapter 4) and natural language processing (Chapter 2).[27] One no longer needs to be a specialist in the visual attributes of faces to create a face-recognition algorithm. One no longer requires a thorough understanding of a game's strategies to write a program that can master it. One no longer needs to be an authority on the structure and semantics of each of several languages to develop a language-translation tool. For a rapidly growing list of use cases, one's ability to apply deep learning techniques outweighs the value of domain-specific proficiency. While such proficiency formerly may have necessitated a doctoral degree or perhaps years of postdoctoral research within a given domain, a functional level of deep learning capability can be developed with relative ease—as by working through this book!

27. An especially entertaining recounting of the disruption to the field of machine translation is provided by Gideon Lewis-Kraus in his article "The Great A.I. Awakening," published in the *New York Times Magazine* on December 14, 2016.

TENSORFLOW PLAYGROUND

For a fun, interactive way to crystallize the hierarchical, feature-learning nature of deep learning, make your way to the TensorFlow Playground at `bit.ly/TFplayground`. When you use this custom link, your network should automatically look similar to the one shown in Figure 1.18. In Part II we return to define all of the terms on the screen; for the present exercise, they can be safely ignored. It suffices at this time to know that this is a deep learning model. The model architecture consists of six layers of artificial neurons: an input layer on the left (below the "FEATURES" heading), four "HIDDEN LAYERS" (which bear the responsibility of learning), and an "OUTPUT" layer (the grid on the far right ranging from –6 to +6 on both axes). The network's goal is to learn how to distinguish orange dots (negative cases) from blue dots (positive cases) based solely on their location on the grid. As such, in the input layer, we are only feeding in two pieces of information about each dot: its horizontal position (X_1) and its vertical position (X_2). The dots that will be used as training data are shown by default on the grid. By clicking the *Show test data* toggle, you can also see the location of dots that will be used to assess the performance of the network as it learns. Critically, these test data are

not available to the network while it's learning, so they help us ensure that the network generalizes well to new, unseen data.

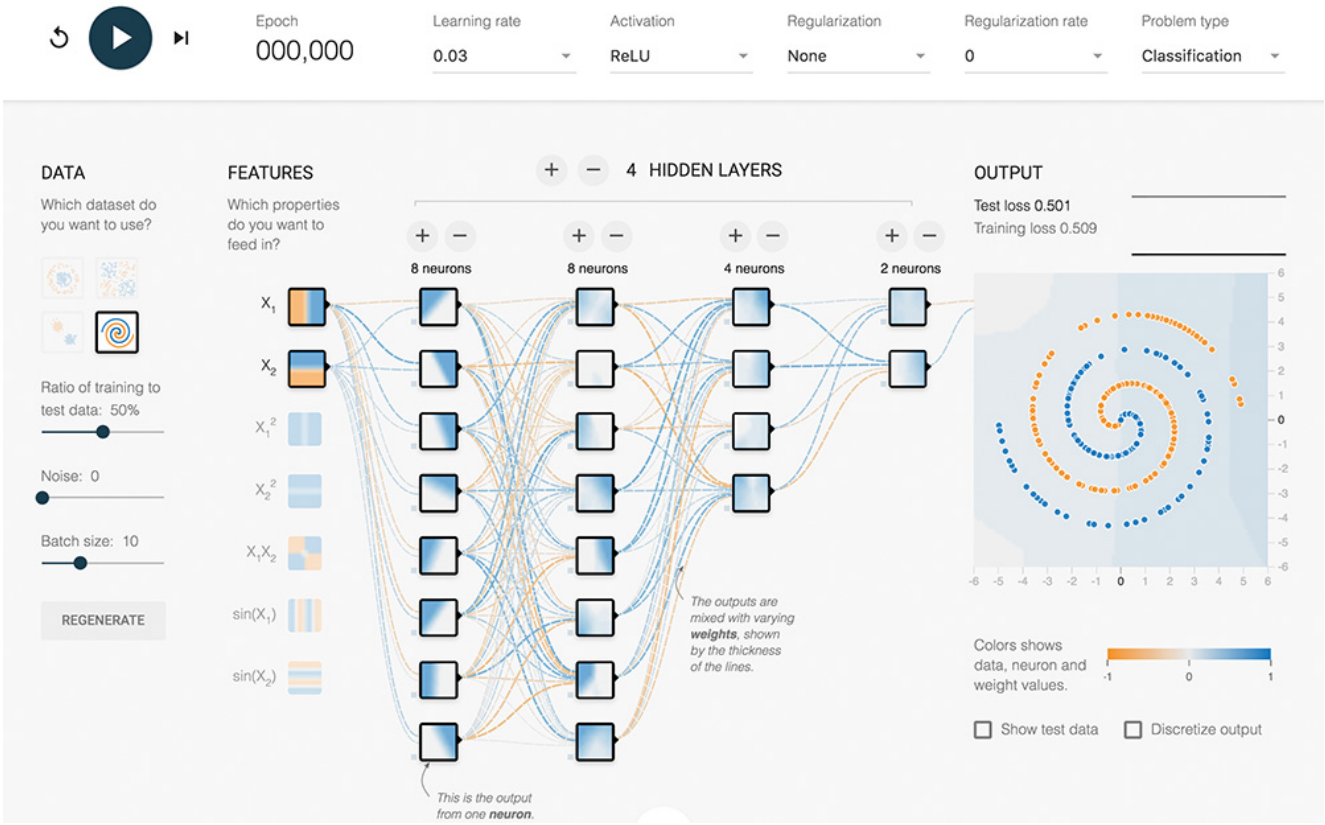

Figure 1.18 This deep neural network is ready to learn how to distinguish a spiral of orange dots (negative cases) from blue dots (positive cases) based on their position on the X_1 and X_2 axes of the grid on the right.

Click the prominent *Play* arrow in the top-left corner. Enable the network to train until the "Training loss" and "Test loss" in the top-right corner have both approached zero—say, less than 0.05. How long this takes will depend on the hardware you're using but hopefully will not be more than a few minutes.

As captured in Figure 1.19, you should now see the network's artificial neurons representing the input data, with increasing complexity and abstraction the deeper (further to the right) they are positioned—as in the neocognitron, LeNet-5 (Figure 1.11), and AlexNet (Figure 1.17). Every time the network is run, the neuron-level details of how the network solves the spiral classification problem are unique, but the general approach remains the same (to see this for yourself, you can refresh the page and retrain the network). The artificial neurons in the leftmost hidden layer are specialized in distinguishing edges (straight lines), each at a particular orientation. Neurons from the first hidden layer pass information to neurons in the second hidden layer, each of which recombines the edges into slightly more complex features like curves. The neurons in each successive layer recombine information from the neurons of the preceding layer, gradually increasing the complexity and abstraction of the features the neurons can represent. By the final (rightmost) layer, the neurons are adept at representing the intricacies of the spiral shape, enabling the network to accurately predict whether a dot is orange (a negative case) or blue (a positive case) based on its position (its X_1 and X_2 coordinates) in the grid. Hover over a neuron to project it onto the far-right "OUTPUT" grid and examine its individual specialization in detail.

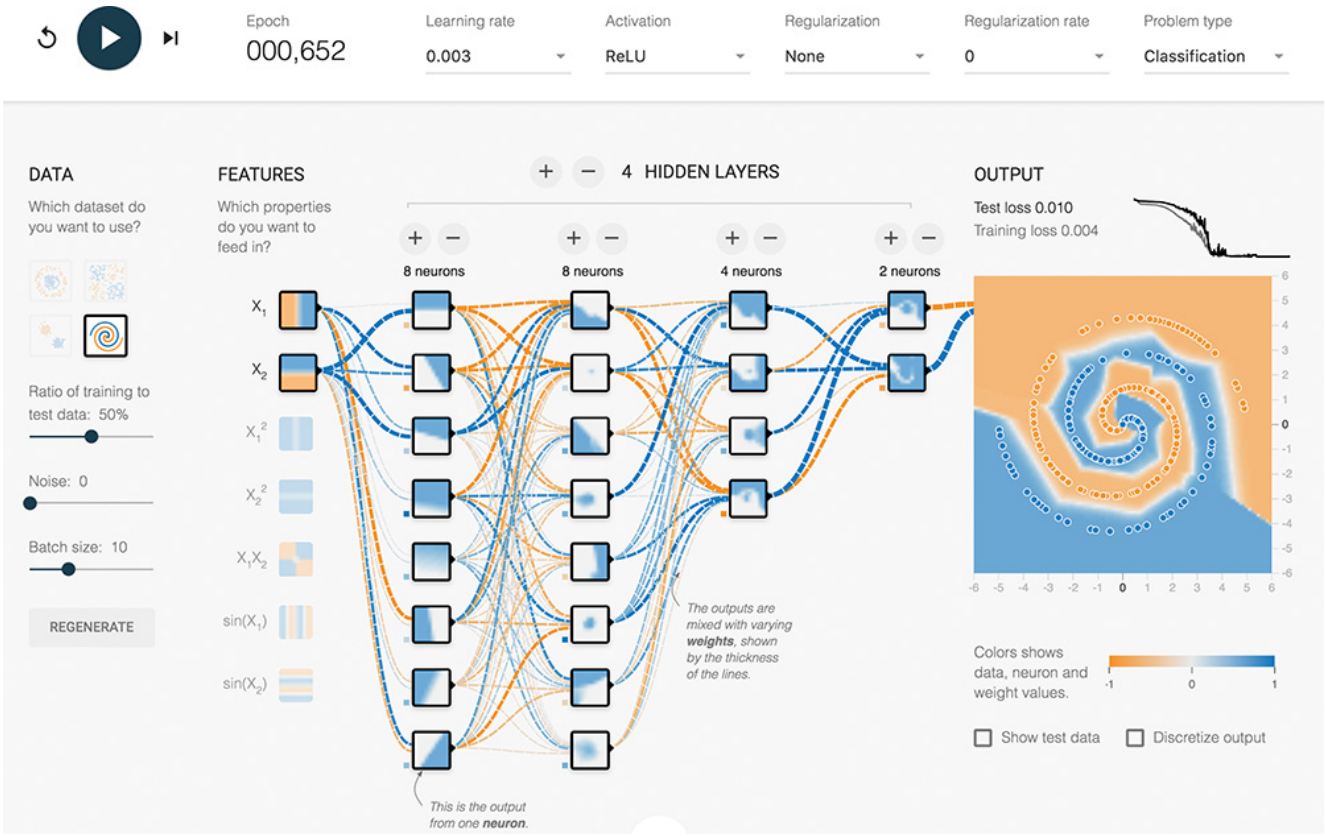

Figure 1.19 The network after training

QUICK, DRAW!

To interactively experience a deep learning network carrying out a machine vision task in real time, navigate to quickdraw.withgoogle.com (http://quickdraw.withgoogle.com) to play the Quick, Draw! game. Click *Let's Draw!* to begin playing the game. You will be prompted to draw an object, and a deep learning algorithm will guess what you sketch. By the end of Chapter 10, we will have covered all of the theory and practical code examples needed to devise a machine vision algorithm akin to this one. To boot, the drawings you create will be added to the dataset that you'll leverage in Chapter 12 when you create a deep learning model that can convincingly mimic human-drawn doodles. Hold on to your seat! We're embarking on a fantastic ride.

SUMMARY

In this chapter, we traced the history of deep learning from its biological inspiration through to the AlexNet triumph in 2012 that brought the technique to the fore. All the while, we reiterated that the hierarchical architecture of deep learning models enables them to encode increasingly complex representations. To concretize this concept, we concluded with an interactive demonstration of hierarchical representations in action by training an artificial neural network in the TensorFlow Playground. In Chapter 2, we will expand on the ideas introduced in this chapter by moving from vision applications to language applications.