Notes

Chapter 1

1. This account is adapted from Fritz Alt, "Archeology of Computers: Reminiscences, 1945–47," *Communications of the ACM* 15, no. 7 (July 1972): 693–694.

2. A. M. Turing, "On Computable Numbers, with an Application to the *Entscheidungsproblem*," *Proceedings of the London Mathematical Society*, Ser. 2, 42 (1936–1937): 230–265.

3. The concept of PAC learning was introduced in L. G. Valiant, "A Theory of the Learnable," *Communications of the ACM* 27, no. 11 (1984): 1134–1142. The concept was subsequently named "probably approximately correct" in D. Angluin and P. Laird, "Learning from Noisy Examples," *Machine Learning* 2 (1987): 343–370.

4. N. Taleb, *The Black Swan* (New York: Random House, 2007); D. Kahneman, *Thinking Fast and Slow* (New York: Farrar, Straus and Giroux, 2011).

Chapter 2

1. In "How U.N. Chief Discovered U.S., and Earmuffs," *New York Times* interview, January 7, 1997.

2. From Foreword by Donald E. Knuth to M. Petkovšek, H. Wilf, and D. Zeilberger, *A=B* (Wellesley, MA: A.K. Peters, 1997).

3. Alfred Russel Wallace proposed the same theory independently in "On the Tendency of Species to Form Varieties, and on the Perpetuation of Varieties and Species by Natural Means of Selection," *Journal of the Proceedings of the Linnean Society of London, Zoology* 3 (1858): 53–62. Because of Darwin's much more detailed exposition in his *On the Origin of Species* (London: Murray, 1959), the theory has become more closely identified with his name.

4. The Weald is an area stretching from Hampshire in the west to Kent in the east, between the North and South Downs in southern England.

5. J. Marchant, *Alfred Russel Wallace, Letters and Reminiscences*, vol. I (London: Cassell, 1916), 242, letter dated April 14, 1869.

6. Lord Kelvin (William Thomson), "The Age of the Earth as an Abode Fitted for Life," *Journal of the Transactions of the Victoria Institute* 31 (1899): 11–35.

Chapter 3

1. The historical context and the related work of contemporaries Gödel, Post, Church, and others are described in M. Davis (ed.), *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions* (Mineola, NY: Dover, 2004).

2. To be more precise, Turing's paper refers to problems equivalent to the Halting Problem, including the Printing Problem, which asks whether a certain symbol will be ever written.

3. K. Gödel, "Remarks Before the Princeton Bicentennial Conference on Problems in Mathematics" (1946), reprinted in Davis (ed.), *The Undecidable,* 84–88.

4. Eugene Wigner, "The Unreasonable Effectiveness of Mathematics in the Natural Sciences," in *Communications in Pure and Applied Mathematics*, vol. 13, no. 1 (February 1960). New York: John Wiley & Sons.

5. The term "computational complexity" was coined by Juris Hartmanis and Richard Stearns in their pioneering study of the time and space requirements of Turing machine computations. Earlier, in 1960, Michael Rabin had given an axiomatic theory of this phenomenon. An earlier reference still, in the context of cryptography, is a letter from John Nash to the National Security Agency in 1955 (www.nsa.gov/public_info/press_room/2012/nash_exhibit.shtml). Comprehensive expositions of this field can be found in C. H. Papadimitriou, *Computational Complexity* (Boston: Addison-Wesley, 1994); O. Goldreich, *Computational Complexity: A Conceptual Perspective* (New York: Cambridge University Press, 2008); and S. Arora and B. Barak, *Complexity Theory: A Modern Approach* (New York: Cambridge University Press, 2009).

6. A function $f(n)$ is $O(g(n))$ if for some constant k and for all $n > 0$, $f(n) < kg(n)$. If one changes the basis of the number representation from 10 to another number, such as 2 for the case of binary arithmetic that computers use, the long multiplication algorithm is still $O(n^2)$ steps.

7. Strictly speaking, P is usually defined only for problems with yes/no answers. For simplicity, in this book we will also use it to include problems with many bit outputs, such as integer multiplication, if computing each bit of the output is a P problem in the more standard sense, and there are only polynomially many output bits.

8. A. Karatsuba and Yu. Ofman, "Multiplication of Multi-Digit Numbers on Automata," *Soviet Physics Doklady* 7 (1963): 595–596.

9. A. Schönhage and V. Strassen, "Schnelle Multiplikation grosser Zahlen," *Computing* 7 (1971): 281–292. The runtime of their algorithm is $O(n \log n \operatorname{loglog} n)$ steps, an expression that grows more slowly than $n^{1.001}$, or $n^{1+\varepsilon}$ for any positive ε. In 2007 this was slightly improved by Martin Fürer to a function that still grows a little more slowly than $n \log n$.

10. For polynomial time algorithms for testing primality, see Robert Solovay and Volker Strassen, "A Fast Monte-Carlo Test for Primality," *SIAM Journal on Computing* 6, no. 1 (1977): 84–85; Gary L. Miller, "Riemann's Hypothesis and Tests for Primality," *Journal of Computer and System Sciences* 13, no. 3 (1976): 300–317;

M. O. Rabin, "Probabilistic Algorithm for Testing Primality," *Journal of Number Theory* 12, no. 1 (1980): 128–138. A deterministic algorithm with higher but still polynomial complexity was found more recently: M. Agrawal, N. Kayal, and N. Saxena, "PRIMES Is in P," *Annals of Mathematics* 160, no. 2 (2004): 781–793.

11. R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM* 21, no. 2 (1978): 120–126. A general approach to relating cryptography and complexity theory is given in S. Goldwasser and S. Micali, "Probabilistic Encryption," *Journal of Computer and System Sciences* 28, no. 2 (1984): 270–299.

12. Turing had used the phrase "intellectual search" for a seemingly similar concept, but without an explicit polynomial criterion: A. M. Turing, "Intelligent Machinery" (unpublished manuscript, 1948), reproduced in B. J. Copeland, *The Essential Turing* (Oxford: Oxford University Press, 2004), 410–432.

13. The notion of NP-completeness was introduced in S. A. Cook, "The Complexity of Theorem Proving Procedures," *Proceedings, Third Annual ACM Symposium on the Theory of Computing* (1971): 151–158. The range of NP-complete problems was greatly extended by R. M. Karp, "Reducibility among Combinatorial Problems," in Raymond E. Miller and James W. Thatcher (eds.), *Complexity of Computer Computations* (New York: Plenum Press, 1972), 85–103. A parallel development occurred in the Soviet Union: L. Levin, "Universal Search Problems," *Problems of Information Transmission* 9, no. 3 (1973): 265–266 (in Russian), translated into English by B. A. Trakhtenbrot, "A Survey of Russian Approaches to Perebor (Brute-Force Searches) Algorithms," *Annals of the History of Computing* 6, no. 4 (1984): 384–400. The NP-completeness phenomenon as seen a few years later is excellently described in M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (New York: W.H. Freeman, 1979).

14. K. L. Manders and L. M. Adleman, "NP-Complete Decision Problems for Quadratic Polynomials," *Proceedings, Eighth Annual ACM Symposium on the Theory of Computing* (1976): 23–29.

15. L. G. Valiant, "The Complexity of Computing the Permanent," *Theoretical Computer Science* 8 (1979): 189–201; L. G. Valiant, "The Complexity of Enumeration and Reliability Problems," *SIAM Journal on Computing* 8, no. 3 (1979): 410–421.

16. E. Bernstein and U. Vazirani, "Quantum Complexity Theory," *SIAM Journal on Computing* 26, no. 5 (1997): 1411–1473. This paper introduced the quantum class BQP. Earlier formulations of quantum computation had been given by R. P. Feynman, "Simulating Physics with Computers," *International Journal of Theoretical Physics* 21 (1982): 467–488, and D. Deutsch, "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer," *Proceedings of the Royal Society of London, Series A, Mathematical and Physical Sciences* 400 (1985): 97–117.

17. Note that each of these classes contains functions with only yes/no values, except for #P, which produce numbers. The PAC class is illustrated as a subclass of

P, but one could extend it into BQP, for example. It is currently unknown, for any pair of the classes illustrated, whether they are of equal extent to within polynomial time deterministic reductions. Every two of the classes shown is widely conjectured to be different, except for the P =? BPP question, for which some suggestion of the contrary conjecture can be found in R. Impagliazzo and A. Wigderson, "P = BPP if E Requires Exponential Circuits: Derandomizing the XOR Lemma," *Proceedings of the 29th ACM Symposium on Theory of Computing* (1997): 220–229. Of course, taking a position on any these mathematical conjectures is a theoryless activity, and this author is not doing that here.

18. F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms* (Washington, DC: Spartan Books, 1962). A detailed analysis is given by M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*, 2nd ed. (Cambridge, MA: MIT Press, 1972).

19. This example is suggested by a dataset on iris varieties from R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annual Eugenics* 7, part II (1936): 179–188.

20. Without loss of generality we can make the right-hand side of any perceptron 0 by adding an extra variable to the left-hand side and extending each example to have the fixed value 1 for this last variable. Figure 3.7 implements this idea to find the separator $3x - 6y > 1$ for the six points listed in the rubric of Figure 3.6. Note that the inequality $2x - 3y > 2$ illustrated there also satisfies these six examples.

Chapter 4

1. Eddington made this remark in Leicester, UK, at the annual meeting of the British Association for the Advancement of Science: "Star Birth Sudden Lemaître Asserts," *New York Times*, September 12, 1933.

2. A. M. Turing, "The Chemical Basis of Morphogenesis," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 237, no. 641 (August 1952): 37–72.

3. This viewpoint also has received support from within the biological sciences community: P. Nurse, "Life, Logic and Information," *Nature* 454 (2008): 424–426.

4. M. H. A. Newman, "Alan Mathison Turing, 1912–1954," *Biographical Memoirs of Fellows of the Royal Society* 1 (1955): 253–263.

Chapter 5

1. A. M. Turing, "Solvable and Unsolvable Problems," *Science News* 31 (1954): 7–23.

2. The nature of the computations that brain-like systems are capable of executing within realistic resource limitations deserves separate investigation: L. G. Valiant, *Circuits of the Mind* (New York: Oxford University Press, 1994, 2000); L. G. Valiant, "Memorization and Association on a Realistic Neural Model," *Neural Computation*

17, no. 3 (2005): 527–555. Various failings of human memory from an experimental psychology perspective are described in D. Schacter, *The Seven Sins of Memory: How the Mind Forgets and Remembers* (New York: Houghton Mifflin, 2002).

3. Aristotle, *Posterior Analytics, Book I*, translated by G. R. G. Mure (eBooks@ Adelaide, 2007).

4. P. Hallie (ed.), *Selections from the Major Writings on Skepticism, Man and God*, translated by S. Etheridge (Indianapolis, IN: Hackett, 1985), 105.

5. A calculation shows that a sample size $(2/error) \times (n + \log_e(1/error))$ suffices: L. G. Valiant, "A Theory of the Learnable," *Communications of the ACM* 27, no. 11 (1984): 1134–1142.

6. A sample size similar in terms of n and *error* to that in Note 5, above, still suffices.

7. The study of elimination, but without any quantitative analysis of what it achieves, has a long history: John Stuart Mill, *A System of Logic* (London: John W. Parker, 1843).

8. A purely computational theory is given by E. M. Gold, "Language Identification in the Limit," *Information and Control* 10 (1967): 447–474. A statistical theory is provided in V. N. Vapnik, *The Nature of Statistical Learning Theory* (New York: Springer-Verlag, 2000), and T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning* (New York: Springer-Verlag, 2001).

9. More details on PAC learning and its extensions can be found in M. J. Kearns and U. Vazirani, *An Introduction to Computational Learning Theory* (Cambridge, MA: MIT Press, 1994).

10. The Occam formulation is from A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Occam's Razor," *Information Processing Letters* 24 (1987): 377–380. It exemplifies how the purely statistical criterion of learnability is almost tautological if examples and hypotheses are to be represented discretely, which in reality they always are. For infinite representations, such as real numbers, an analogous treatment is still possible, but more involved, via, for example, the VC dimension: V. Vapnik and A. Chervonenkis, "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities," *Theory of Probability and Its Applications* 16, no. 2 (1971): 264–280; A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Learnability and the Vapnik–Chervonenkis Dimension," *Journal of the ACM* 36, no. 4 (1989): 929–965. Earlier work using related concepts: Thomas M. Cover, "Capacity Problems for Linear Machines," in L. Kanal (ed.), *Pattern Recognition* (Washington, DC: Thompson Book Co., 1968).

11. Such a general lower bound on the number of examples needed for learning is given in A. Ehrenfeucht, D. Haussler, M. Kearns, and L. G. Valiant, "A General Lower Bound on the Number of Examples Needed for Learning," *Information and Computation* 82, no. 2 (1989): 247–261.

12. The first publication of the notion of public-key cryptosystems was W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory* IT-22 (November 1976): 644–654. The RSA system is from

R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM* 21, no. 2 (1978): 120–126. There had been earlier unpublished work on these concepts by James Ellis, Clifford Cocks, and Malcolm Williamson at the Government Communications Headquarters in the UK, and also by Ralph Merkle at UC Berkeley.

13. Here we are regarding the decryption function as outputting a set of yes/no functions, namely the bits of the original message, and each one would be learned. In any public-key cryptosystem the encryption algorithm is available to all.

14. N. Chomsky, "Three Models for the Description of Language," *IRE Transactions on Information Theory* 2 (1956): 113–124.

15. M. Kearns and L. G. Valiant, "Cryptographic Limitations on Learning Boolean Formulae and Finite Automata," *Journal of the ACM* 41, no. 1 (1994): 67–95. Preliminary version in *Proceedings of the 21st ACM Symposium on Theory of Computing* (1989): 433–444.

16. A. Klivans and R. Servedio, "Learning DNF in time $2^{\tilde{O}(n^{1/3})}$," *Journal of Computer and System Sciences* 68, no. 2 (2004): 303–318.

17. An elegant attribute-efficient algorithm, called Winnow, for learning disjunctions is given in N. Littlestone, "Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm," *Machine Learning* 2, no. 4 (1988): 285–318. It resembles the perceptron algorithm, but the weights are updated by multiplying rather than by adding appropriate quantities. See also Avrim Blum, "Learning Boolean Functions in an Infinite Attribute Space," *Machine Learning* 9 (1992): 373–386.

18. R. I. Arriaga and S. Vempala, "An Algorithmic Theory of Learning: Robust Concepts and Random Projection," *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS)* (1999): 616–623.

Chapter 6

1. The Royal Tyrrell Museum in Drumheller, Alberta, Canada, is instructive.

2. Questions about the absence of quantitative explanations in evolutionary theory were raised by various authors in P. S. Moorhead and M. M. Kaplan (eds.), *Mathematical Challenges to the Neo-Darwinian Interpretation of Evolution: A Symposium, Philadelphia, April 1966* (Philadelphia: Wistar Institute Press, 1967). An attempt to address the issue in the context of the eye is given in D. E. Nilsson and S. Pelger, "A Pessimistic Estimate of the Time Required for an Eye to Evolve," *Proceedings: Biological Sciences* 256 (1994): 53–58.

3. R. A. Fisher, *The Genetical Theory of Natural Selection* (Oxford: Oxford University Press, 1930); S. Wright, *Evolution and the Genetics of Populations, A Treatise* (Chicago: University of Chicago Press, 1968–1978).

4. The award winners in an annual competition for genetic programming can be found at http://www.genetic-programming.org/combined.html.

5. The term comes from the title of Julian Huxley's book *Evolution: The Modern Synthesis* (1942). It refers to the synthesis reached in the preceding decade of three disparate strands: natural selection as understood by Darwin and Wallace, Mendelian genetics, and the population biology of R. A. Fisher, J. B. S. Haldane, and Sewall Wright.

6. U. Alon, *An Introduction to Systems Biology* (Boca Raton, FL: CRC Press, 2006).

7. For simplicity, our terminology assumes deterministic input functions. However, the discussion is the same if we allow randomized or even quantum transitions.

8. C. D. Allis, T. Jenuwein, and D. Reinberg, *Epigenetics* (Cold Spring Harbor, NY: CSHL Press, 2007).

9. The possible impact on natural selection of learning during life has been explored informally since the nineteenth century: M. J. Baldwin, "A New Factor in Evolution," *The American Naturalist* 30, no. 354 (June 1896): 441–451; G. E. Hinton and S. J. Nowlan, "How Learning Can Guide Evolution," C*omplex Systems* 1 (1987): 495–502.

10. N. Eldredge and S. J. Gould, "Punctuated Equilibria: An Alternative to Phyletic Gradualism," in T. J. M. Schopf (ed.), *Models in Paleobiology* (San Francisco: Freeman, Cooper and Company, 1972), 82–115.

11. A. R. Wallace, "The Measurement of Geological Time," *Nature*, 17 (1870): 399–341, 452–455.

12. We are oversimplifying in this exposition, as we did also with regard to learning in Chapter 5, in not distinguishing the class *C* of target ideal functions from the representations used for the hypotheses. The latter may be different from the former.

13. The evolvability model described here is from L. G. Valiant, "Evolvability," *Journal of the ACM* 56, no. 1 (2009): 3:1–3:21. (Earlier versions: *Proceedings of the 32nd International Symposium on Mathematical Foundations of Computer Science,* August 26–31, 2007; Český Krumlov, Czech Republic, *Lecture Notes in Computer Science,* vol. 4708 [New York: Springer-Verlag, 2007], 22–43; and *Electronic Colloquium on Computational Complexity*, Report 120, September 2006.)

14. M. J. Kearns, "Efficient Noise-Tolerant Learning from Statistical Queries," *Journal of the ACM* 45, no. 6 (1998): 983–1006.

15. V. Feldman, "Distribution-Independent Evolvability of Linear Threshold Functions," *Journal of Machine Learning Research—Proceedings Track* 19 (2011): 253–272.

16. M. J. Kearns, "Efficient Noise-Tolerant Learning from Statistical Queries."

17. V. Feldman, "Robustness of Evolvability," *Proceedings of the 22nd Annual Conference on Learning Theory, Montreal, Quebec, Canada* (2009). See also V. Feldman, "Evolvability from Learning Algorithms," *Proceedings of the 40th Annual ACM Symposium on Theory of Computing* (2008): 619–628.

18. L. Michael, "Evolvability via the Fourier Transform" (manuscript, 2007). Also *Theoretical Computer Science* 462 (2012): 88–98.

19. V. Feldman, "A Complete Characterization of Statistical Query Learning with Applications to Evolvability," *50th Annual IEEE Symposium on Foundations of Computer Science* (2009): 375–384.

20. P. Valiant, "Distribution Free Evolvability of Polynomial Functions over All Convex Loss Functions," *Proceedings of the 3rd Symposium on Innovations in Theoretical Computer Science* (2012): 142–148.

21. Ibid.

22. V. Kanade, "Evolution with Recombination," *52nd Annual IEEE Symposium on Foundations of Computer Science* (2011): 837–846.

23. R. A. Fisher, *The Genetical Theory of Natural Selection* (Oxford: Clarendon Press, 1930); J. Maynard Smith, *The Evolution of Sex* (Cambridge: Cambridge University Press, 1978); A. Livnat, C. H. Papadimitriou, J. Dushoff, and M. W. Feldman, "A Mixability Theory of the Role of Sex in Evolution" *PNAS* 105, no. 50 (2008): 19803–19808.

24. Some evolution algorithms can be shown to be able to tolerate a slowly changing world: V. Kanade, L. G. Valiant, and J. Wortman Vaughan, "Evolution with Drifting Targets," *Conference on Learning Theory* (2010): 155–167.

Chapter 7

1. I. Ayres, *Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart* (New York: Bantam, 2007).

2. This formulation is further described in L. G. Valiant, *Circuits of the Mind* (New York: Oxford University Press, 1994, 2000).

3. D. B. Lenat, "CYC: A Large-Scale Investment in Knowledge Infrastructure," *Communications of the ACM* 38, no. 11 (1995): 32–38.

4. Bayesian models and inference are described in J. Pearl, *Probabilistic Reasoning in Intelligent Systems* (San Francisco: Morgan Kaufmann Publishers, 1988). Further studies of reasoning in uncertain contexts can be found in the *Uncertainty in Artificial Intelligence* conference series. It has been found empirically that in applications where large amounts of general knowledge need to be modeled, some learning component is essential, as, for example, in IBM's Watson system for the *Jeopardy!* contest.

5. D. Angluin and P. Laird, "Learning from Noisy Examples," *Machine Learning* 2 (1987): 343–370. A generic approach to making learning algorithms resistant to one kind of noise is given in Michael J. Kearns, "Efficient Noise-Tolerant Learning from Statistical Queries," *Journal of the ACM* 45, no. 6 (1998): 983–1006.

6. L. G. Valiant, "Robust Logics," *Artificial Intelligence Journal* 117 (2000): 231–253.

7. G. A. Miller, "The Magical Number Seven Plus or Minus Two," *The Psychological Review* 63 (1956): 81–97.

8. F. Galton, *Inquiries into Human Faculty and Its Development*, 1st ed. (London: Macmillan, 1883).

9. I. Biederman, "Recognition–by-Components: A Theory of Human Image Understanding," *Psychological Review* 94 (1987): 115–147.

10. In this work, discussions of the brain are in terms of what it needs to do, and not how it does it. For the latter, see Note 2 to Chapter 5.

11. Reaction time experiments show that the visual system can recognize what object is in a scene extremely rapidly: S. J. Thorpe, D. Fize, and C. Marlot, "Speed of Processing in the Human Visual System," *Nature* 381 (1996): 520–522.

12. N. Littlestone, "Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm," *Machine Learning* 2, no. 4 (1988): 285–318.

13. The one constraint needed for the reasoning part to be polynomial in terms of the relevant parameters (such as the size of the rules) is that the number of arguments in all the relations be bounded by a constant, since the reasoning process is exponential in that quantity.

14. A fuller discussion of how robust logic might be used in intelligent systems is given in L. G. Valiant, "Knowledge Infusion," *Proceedings of the 21st National Conference on Artificial Intelligence,* July 16–20, Boston, MA (Menlo Park, CA: AAAI Press, 2006), 1546–1551. Some experimental results are reported in L. Michael and L. G. Valiant, "A First Experimental Demonstration of Massive Knowledge Infusion, *Proceedings of 11th International Conference on Principles of Knowledge Representation and Reasoning* (Menlo Park, CA: AAAI Press, 2008), 378–389.

Chapter 8

1. J. Pearl, *Causality* (Cambridge: Cambridge University Press, 2009).

Chapter 9

1. M. Kearns and L. G. Valiant, "Cryptographic Limitations on Learning Boolean Formulae and Finite Automata," *Journal of the ACM* 41, no. 1 (1994): 67–95. Preliminary version in *Proceedings of the 21st ACM Symposium on Theory of Computing* (1989): 433–444.

2. R. Schapire, "Strength of Weak Learnability," *Machine Learning* 5 (1990): 197–227.

3. Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences* 55, no. 1 (1997): 119–139.

4. M. Aizerman, E. Braverman, and L. Rozonoer, "Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning," *Automation and Remote Control* 25 (1964): 821–837.

5. B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *5th Annual ACM Workshop on Computational Learning*

Theory (Pittsburgh, PA: ACM Press, 1992), 144–152; C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning* 20 (1995).

6. T. M. Mitchell et al., "Learning to Decode Cognitive States from Brain Images," *Machine Learning* 57 (2004): 145–175.

7. A. M. Turing, "Intelligent Machinery" (unpublished manuscript, 1948). Reproduced in B. J. Copeland, *The Essential Turing* (Oxford: Oxford University Press, 2004), 410–432.

8. L. G. Valiant, "Functionality in Neural Nets," *Proceedings of the First Workshop on Computational Learning Theory* (San Francisco: Morgan Kaufmann Publishers, 1988), 28–39.

9. R. Paturi, S. Rajasekaran, and J. H. Reif, "The Light Bulb Problem," *Proceedings of the Second Annual Workshop on Computational Learning Theory* (San Francisco: Morgan Kaufmann Publishers, 1989), 261–268; P. Indyk and R. Motwani, "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality," *Proceedings of the 30th Annual ACM Symposium on Theory of Computing* (1998): 604–614; M. Dubiner, "Bucketing, Coding and Information Theory for the Statistical High Dimensional Nearest Neighbor Problem," arXiv:0810.4182 (2008); G. Valiant, "Finding Correlations in Subquadratic Time, with Applications to Learning Parities and Juntas," *53rd Annual IEEE Symposium on Foundations of Computer Science* (2012): 11–20.

10. This is discussed more fully in L. G. Valiant, *Circuits of the Mind* (New York: Oxford University Press, 1994, 2000).

11. H. Gardner, *Frames of Mind: The Theory of Multiple Intelligences* (New York: Basic Books, 1983).

12. A. M. Turing, "Computing Machinery and Intelligence," *Mind* 49 (1950): 433–460. Notwithstanding the title, Turing does not claim in the text a definition of intelligence, but rather a criterion on when a machine could be regarded as "thinking." However, from the beginning, the Turing Test has been regarded as a fitting reference point for any discussion of intelligence.

13. Ibid.

Chapter 10

1. Winston Churchill, Speech at Harvard University, September 6, 1943.

Glossary

An **accessible target** is a function that can be learned or evolved because it is within a learnable or evolvable class with respect to the available features.

A **Boolean function** is a function whose inputs and outputs take just true or false values. These values may be represented by 1 and 0, or by 1 and –1. An example of a Boolean function of two arguments is the **or** function, written as **or**(x, y) and defined to have value true if and only if at least one of the two arguments x, y has value true.

BPP (bounded probabilistic polynomial) computations are those that can be performed in polynomial time by a randomized Turing machine.

BQP (bounded quantum polynomial) computations are those that can be performed in polynomial time by a quantum Turing machine.

A **circuit** is a computation where the dependencies among various input, output, or intermediate values can be made explicit.

A **complexity class** is a set of problems characterized by the computations that can solve them. For example, P and NP are complexity classes.

A function is **computable** if it can be computed by some Turing machine, in the sense that for every input the Turing machine produces the answer within some finite number of steps.

Concept is a term used here to denote a function in the context of learning.

A **conjunction** is a Boolean function that has value true if all of its arguments have value true.

A **disjunction** is a Boolean function that has value true if at least one of its arguments has value true.

Ecorithm is a term introduced here to denote an algorithm that takes information from its environment so as to perform better in that environment. Algorithms for machine learning, evolution, and for learning for the purpose of reasoning are all instances of ecorithms.

A class of functions is **evolvable** if there is an evolution algorithm that can evolve every member of it using only polynomial resources and achieving polynomial error control.

The **expression level** of a protein in a cell is a measure of how much of the protein is being produced.

Feasible computation is identified here with computations in which the number of steps is bounded by a polynomial in terms of the number of bits required to write down the input.

A **function** is a mathematical assertion of a specific dependence of a value on some variables, parameters, conditions, or arguments. For example, the function $f(x, y) = 2x + 3y$ is the dependence that f is the sum of twice the first argument and thrice the second.

The **ideal function** specifies for a particular evolving entity in a particular environment the best possible action for every possible combination of conditions.

The **input function** of a protein is the function that determines the expression level of the protein in terms of all the relevant conditions in the cell.

Intelligence is generally used in the text in the colloquial sense of human intelligence, but the aspect of it that is addressed more technically is that of reasoning on uncertain, learned knowledge.

A **linear inequality** is an assertion that the value of some linear combination of variables is greater or less than some value (e.g., $3x + 6y - 8z < 7$).

A **linear separator** for a set of labeled examples is a linear inequality that satisfies all the positive examples and none of the negative examples.

Nondeterministic computations are those that perform an exponential search for a solution in parallel.

NP (nondeterministic polynomial) is the class of problems for which there are nondeterministic computations where each of the parallel branches of the search uses at most a polynomial number of steps in terms of the number of bits of the input.

A problem is **NP-complete** if its polynomial time solution would imply polynomial time solutions for every problem in NP.

P is the class of problems to which solutions can be found by deterministic computations taking a polynomial number of steps in terms of the number of bits of the input.

#P ("sharp" P) is the class of problems that count the number of solutions found in an NP computation.

PAC (probably approximately correct) learning is the process of learning from examples, where the number of computational steps is polynomially bounded and the errors are polynomially controlled.

A class of problems is **PAC learnable** if there is a learning algorithm that can learn every member of the class, using only polynomial resources and achieving polynomial error control.

PAC semantics is the sense in which the definition of PAC learning guarantees accuracy.

A **parity function** is a Boolean function that has value true if and only if an odd number of its arguments have value true.

The **Perceptron** algorithm is a specific method for learning linear inequalities.

Polynomially bounded for a function $f(n)$ is used here to mean that for some fixed numbers c and k, for every positive integer value of n , $f(n) < cn^k$.

A computational **problem** is a function that is to be evaluated. For example, determining how many factors a number has is a computational problem. A **solution** to such a problem is an algorithm that evaluates that function.

The **protein expression network** represents how the expression levels of all the proteins in a cell are regulated in terms of each other and other relevant factors. It is sometimes referred to as the **gene expression network.**

A **randomized Turing machine** is a Turing machine that at any step may choose one among a set of possible transitions by making a random decision according to the toss of a coin.

Resilience to different distributions is the desirable property of a learning algorithm to give reliable answers for wide ranges of distributions of the examples.

Resilience to noise is the desirable property of a learning algorithm to give answers that are degraded only a little by any noise in the data from which it is learning.

A robust computational model for a phenomenon is one that is provably equivalent to a wide range of alternative definitions of computational models for that phenomenon. Turing machines for the phenomenon of computation offer the exemplary paradigm.

Robust logic is a system in which learning and reasoning have a common semantics and both can be accomplished feasibly in the PAC sense.

Robustness to computation and data is a way of phrasing what PAC learning accomplishes, namely the requirement that it should be practicable to drive down errors arbitrarily by increasing the amounts of training data and computation appropriately.

A **statistical query** algorithm is a learning algorithm that can receive information about examples only by asking statistical questions about them, rather than by processing individual examples.

Target pursuit is the capability in both learning and evolution to pursue a large number of accessible targets simultaneously.

Theoryful is a term defined here to denote decisions for which there is a good explanatory and predictive theory, such as a scientific theory.

Theoryless is a term defined here to denote decisions that are not known to be theoryful.

A **Turing machine** is a model of computation that is widely believed to encompass all information processing that one would think of as mechanistic.

Acknowledgments

As the text makes clear, this book is deeply rooted in the visionary ideas of Alan Turing. The synthesis offered here is within the framework of computational learning theory. Over the last three decades many have enriched this field, and I would particularly like to thank Dana Angluin, Avrim Blum, Andrzej Ehrenfeucht, Vitaly Feldman, Yoav Freund, David Haussler, Varun Kanade, Michael Kearns, Roni Khardon, Adam Klivans, Ming Li, Nick Littlestone, Yishay Mansour, Loizos Michael, Lenny Pitt, Ron Rivest, Dan Roth, Robert Schapire, Rocco Servedio, and Manfred Warmuth. It is also a pleasure to acknowledge the early pioneers of computational complexity, including Manuel Blum, Stephen Cook, Juris Hartmanis, Richard Karp, Michael Rabin, and Volker Strassen.

I am grateful to Juliet Harman for her careful and critical reading of the manuscript and for her many ideas that have improved this book.

I would also like to thank Thomas Kelleher at Basic Books for his valuable editorial suggestions.

It is traditional for authors to thank their spouse for patiently suffering the inevitable hardships as a writer's companion. I can only report unreserved enthusiasm for the project, and thank Gayle for that and for her significant and extensive suggestions on the text.

Index

n means note, g means glossary