

# Introdução

*Em que tentamos explicar por que consideramos a inteligência artificial um assunto digno de estudo e em que procuramos definir exatamente o que é a inteligência artificial, pois essa definição é importante antes de iniciarmos nosso estudo.*

**D**enominamos nossa espécie *Homo sapiens* — homem sábio — porque nossa **inteligência** é tão importante para nós. Durante milhares de anos, procuramos entender *como pensamos*, isto é, como um mero punhado de matéria pode perceber, compreender, prever e manipular um mundo muito maior e mais complicado que ela própria. O campo da **inteligência artificial**, ou IA, vai ainda mais além: ele tenta não apenas compreender, mas também *construir* entidades inteligentes.

A IA é um dos campos mais recentes em ciências e engenharia. O trabalho começou logo após a Segunda Guerra Mundial, e o próprio nome foi cunhado em 1956. Juntamente com a biologia molecular, a IA é citada regularmente como “o campo em que eu mais gostaria de estar” por cientistas de outras disciplinas. Um aluno de física pode argumentar, com boa dose de razão, que todas as boas ideias já foram desenvolvidas por Galileu, Newton, Einstein e o resto. IA, por outro lado, ainda tem espaço para vários Einsteins e Edisons em tempo integral.

Atualmente, a IA abrange uma enorme variedade de subcampos, do geral (aprendizagem e percepção) até tarefas específicas, como jogos de xadrez, demonstração de teoremas matemáticos, criação de poesia, direção de um carro em estrada movimentada e diagnóstico de doenças. A IA é relevante para qualquer tarefa intelectual; é verdadeiramente um campo universal.

## 1.1 O QUE É IA?

Afirmamos que a IA é interessante, mas não dissemos o que ela é. Na Figura 1.1 podemos visualizar oito definições de IA, dispostas ao longo de duas dimensões. Em linhas gerais, as que estão na parte superior da tabela se relacionam a *processos de pensamento* e *raciocínio*, enquanto as definições da parte inferior se referem ao *comportamento*. As definições do lado esquerdo medem o sucesso em termos de fidelidade ao desempenho *humano*, enquanto as definições do lado direito medem o sucesso comparando-o a um conceito *ideal* de inteligência, chamado de **racionalidade**. Um sistema é racional se “faz a coisa certa”, dado o que ele sabe.

Historicamente, todas as quatro estratégias para o estudo da IA têm sido seguidas, cada uma delas por pessoas diferentes com métodos diferentes. Uma abordagem centrada nos seres humanos deve ser em parte uma ciência empírica, envolvendo hipóteses e confirmação experimental. Uma abordagem racionalista<sup>1</sup> envolve uma combinação de matemática e engenharia. Cada grupo tem ao mesmo tempo desacreditado e ajudado o outro. Vamos examinar as quatro abordagens com mais detalhes.

Pensando como um humano	Pensando racionalmente
<p>“O novo e interessante esforço para fazer os computadores pensarem (...) <i>máquinas com mentes</i>, no sentido total e literal.”            (Haugeland, 1985)</p> <p>“[Automatização de] atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado...” (Bellman, 1978)</p>	<p>“O estudo das faculdades mentais pelo uso de modelos computacionais.” (Charniak e McDermott, 1985)</p> <p>“O estudo das computações que tornam possível perceber, raciocinar e agir.” (Winston, 1992)</p>
Agindo como seres humanos	Agindo racionalmente
<p>“A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (Kurzweil, 1990)</p> <p>“O estudo de como os computadores podem fazer tarefas que hoje são melhor desempenhadas pelas pessoas.” (Rich and Knight, 1991)</p>	<p>“Inteligência Computacional é o estudo do projeto de agentes inteligentes.” (Poole <i>et al.</i>, 1998)</p> <p>“AI... está relacionada a um desempenho inteligente de artefatos.” (Nilsson, 1998)</p>

**Figura 1.1** Algumas definições de inteligência artificial, organizadas em quatro categorias.

### 1.1.1 Agindo de forma humana: a abordagem do teste de Turing

O **teste de Turing**, proposto por Alan Turing (1950), foi projetado para fornecer uma definição operacional satisfatória de inteligência. O computador passará no teste se um interrogador humano, depois de propor algumas perguntas por escrito, não conseguir descobrir se as respostas escritas vêm de uma pessoa ou de um computador. O Capítulo 26 discute os detalhes do teste e também se um computador seria de fato inteligente se passasse nele. Por enquanto, observamos que programar um computador para passar no teste exige muito trabalho. O computador precisaria ter as seguintes capacidades:

- **processamento de linguagem natural** para permitir que ele se comunique com sucesso em um idioma natural;
- **representação de conhecimento** para armazenar o que sabe ou ouve;
- **raciocínio automatizado** para usar as informações armazenadas com a finalidade de responder a perguntas e tirar novas conclusões;
- **aprendizado de máquina** para se adaptar a novas circunstâncias e para detectar e extrapolar

padrões.

O teste de Turing evitou deliberadamente a interação física direta entre o interrogador e o computador porque a simulação *física* de uma pessoa é desnecessária para a inteligência. Entretanto, o chamado **teste de Turing total** inclui um sinal de vídeo, de forma que o interrogador possa testar as habilidades de percepção do indivíduo, além de oferecer ao interrogador a oportunidade de repassar objetos físicos “pela janelinha”. Para ser aprovado no teste de Turing total, o computador precisará de:

- **visão computacional** para perceber objetos e
- **robótica** para manipular objetos e movimentar-se.

Essas seis disciplinas compõem a maior parte da IA, e Turing merece crédito por projetar um teste que permanece relevante depois de 60 anos. Ainda assim, os pesquisadores da IA têm dedicado pouco esforço à aprovação no teste de Turing, acreditando que seja mais importante estudar os princípios básicos da inteligência do que reproduzir um exemplar. O desafio do “voo artificial” teve sucesso quando os irmãos Wright e outros pesquisadores pararam de imitar os pássaros e começaram a usar túneis de vento e aprender sobre aerodinâmica. Os textos de engenharia aeronáutica não definem como objetivo de seu campo criar “máquinas que voem exatamente como pombos a ponto de poderem enganar até mesmo outros pombos”.

### 1.1.2 Pensando de forma humana: a estratégia de modelagem cognitiva

Se pretendemos dizer que dado programa pensa como um ser humano, temos de ter alguma forma de determinar como os seres humanos pensam. Precisamos *penetrar* nos componentes reais da mente humana. Existem três maneiras de fazer isso: através da introspecção — procurando captar nossos próprios pensamentos à medida que eles se desenvolvem — através de experimentos psicológicos — observando uma pessoa em ação; e através de imagens cerebrais, observando o cérebro em ação. Depois que tivermos uma teoria da mente suficientemente precisa, será possível expressar a teoria como um programa de computador. Se os comportamentos de entrada/saída e sincronização do programa coincidirem com os comportamentos humanos correspondentes, isso será a evidência de que alguns dos mecanismos do programa também podem estar operando nos seres humanos. Por exemplo, Allen Newell e Herbert Simon, que desenvolveram o GPS, o “Resolvedor Geral de Problemas” (do inglês “General Problem Solver”) (Newell e Simon, 1961), não se contentaram em fazer seu programa resolver problemas de modo correto. Eles estavam mais preocupados em comparar os passos de suas etapas de raciocínio aos passos de indivíduos humanos resolvendo os mesmos problemas. O campo interdisciplinar da **ciência cognitiva** reúne modelos computacionais da IA e técnicas experimentais da psicologia para tentar construir teorias precisas e verificáveis a respeito dos processos de funcionamento da mente humana.

A ciência cognitiva é um campo fascinante por si só, merecedora de diversos livros e de pelo menos uma enciclopédia (Wilson e Keil, 1999). Ocasionalmente, apresentaremos comentários a respeito de semelhanças ou diferenças entre técnicas de IA e a cognição humana. Porém, a ciência

cognitiva de verdade se baseia necessariamente na investigação experimental de seres humanos ou animais. Deixaremos esse assunto para outros livros à medida que supomos que o leitor tenha acesso somente a um computador para realizar experimentação.

Nos primórdios da IA, frequentemente havia confusão entre as abordagens: um autor argumentava que um algoritmo funcionava bem em uma tarefa e que, *portanto*, era um bom modelo de desempenho humano ou vice-versa. Os autores modernos separam os dois tipos de afirmações; essa distinção permitiu que tanto a IA quanto a ciência cognitiva se desenvolvessem com maior rapidez. Os dois campos continuam a fertilizar um ao outro, principalmente na visão computacional, que incorpora evidências neurofisiológicas em modelos computacionais.

### 1.1.3 Pensando racionalmente: a abordagem das “leis do pensamento”

O filósofo grego Aristóteles foi um dos primeiros a tentar codificar o “pensamento correto”, isto é, os processos de raciocínio irrefutáveis. Seus **silogismos** forneceram padrões para estruturas de argumentos que sempre resultavam em conclusões corretas ao receberem premissas corretas — por exemplo, “Sócrates é um homem; todos os homens são mortais; então, Sócrates é mortal”. Essas leis do pensamento deveriam governar a operação da mente; seu estudo deu início ao campo chamado **lógica**.

Os lógicos do século XIX desenvolveram uma notação precisa para declarações sobre todos os tipos de coisas no mundo e sobre as relações entre elas (compare isso com a notação aritmética básica, que fornece apenas declarações a respeito de números). Por volta de 1965, existiam programas que, em princípio, podiam resolver *qualquer* problema solucionável descrito em notação lógica (contudo, se não houver solução, o programa poderá entrar num laço infinito). A chamada tradição **logicista** dentro da inteligência artificial espera desenvolver tais programas para criar sistemas inteligentes.

Essa abordagem enfrenta dois obstáculos principais. Primeiro, não é fácil enunciar o conhecimento informal nos termos formais exigidos pela notação lógica, em particular quando o conhecimento é menos de 100% certo. Em segundo lugar, há uma grande diferença entre ser capaz de resolver um problema “em princípio” e resolvê-lo na prática. Até mesmo problemas com apenas algumas centenas de fatos podem esgotar os recursos computacionais de qualquer computador, a menos que ele tenha alguma orientação sobre as etapas de raciocínio que deve tentar primeiro. Embora ambos os obstáculos se apliquem a *qualquer* tentativa de construir sistemas de raciocínio computacional, eles surgiram primeiro na tradição logicista.

### 1.1.4 Agindo racionalmente: a abordagem de agente racional

Um **agente** é simplesmente algo que age (a palavra *agente* vem do latim *agere*, que significa fazer). Certamente todos os programas de computador realizam alguma coisa, mas espera-se que um agente computacional faça mais: opere sob controle autônomo, perceba seu ambiente, persista por um período de tempo prolongado, adapte-se a mudanças e seja capaz de criar e perseguir metas. Um

**agente racional** é aquele que age para alcançar o melhor resultado ou, quando há incerteza, o melhor resultado esperado.

Na abordagem de “leis do pensamento” para IA, foi dada ênfase a inferências corretas. Às vezes, a realização de inferências corretas é uma *parte* daquilo que caracteriza um agente racional porque uma das formas de agir racionalmente é raciocinar de modo lógico até a conclusão de que dada ação alcançará as metas pretendidas e, depois, agir de acordo com essa conclusão. Por outro lado, a inferência correta não representa *toda* a racionalidade; em algumas situações, não existe nenhuma ação comprovadamente correta a realizar, mas mesmo assim algo tem de ser feito. Também existem modos de agir racionalmente que não se pode dizer que envolvem inferências. Por exemplo, afastar-se de um fogão quente é um ato reflexo, em geral mais bem-sucedido que uma ação mais lenta executada após cuidadosa deliberação.

Todas as habilidades necessárias à realização do teste de Turing também permitem que o agente haja racionalmente. Representação do conhecimento e raciocínio permitem que os agentes alcancem boas decisões. Precisamos ter a capacidade de gerar sentenças comprehensíveis em linguagem natural porque enunciar essas sentenças nos ajuda a participar de uma sociedade complexa. Precisamos aprender não apenas por erudição, mas também para melhorar nossa habilidade de gerar comportamento efetivo.

 A abordagem do agente racional tem duas vantagens sobre as outras abordagens. Primeiro, ela é mais geral que a abordagem de “leis do pensamento” porque a inferência correta é apenas um dentre vários mecanismos possíveis para se alcançar a racionalidade. Em segundo lugar, ela é mais acessível ao desenvolvimento científico do que as estratégias baseadas no comportamento ou no pensamento humano. O padrão de racionalidade é matematicamente bem definido e completamente geral, podendo ser “desempacotado” para gerar modelos de agente que comprovadamente irão atingi-lo. Por outro lado, o comportamento humano está bem adaptado a um ambiente específico e é definido como a soma de tudo o que os humanos fazem. *Portanto, este livro se concentrará nos princípios gerais de agentes racionais e nos componentes para construí-los.* Veremos que, apesar da aparente simplicidade com que o problema pode ser enunciado, surge uma enorme variedade de questões quando tentamos resolvê-lo. O Capítulo 2 descreve algumas dessas questões com mais detalhes.

Devemos ter em mente um ponto importante: logo veremos que alcançar a racionalidade perfeita — sempre fazer a coisa certa — não é algo viável em ambientes complicados. As demandas computacionais são demasiado elevadas. Porém, na maior parte do livro, adotaremos a hipótese de trabalho de que a racionalidade perfeita é um bom ponto de partida para a análise. Ela simplifica o problema e fornece a configuração apropriada para a maioria do material básico na área. Os Capítulos 5 e 17 lidam explicitamente com a questão da **racionalidade limitada** — agir de forma apropriada quando não existe tempo suficiente para realizar todas as computações que gostaríamos de fazer.

## 1.2 OS FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

Nesta seção, apresentaremos um breve histórico das disciplinas que contribuíram com ideias,

pontos de vista e técnicas para a IA. Como qualquer histórico, este foi obrigado a se concentrar em um pequeno número de pessoas, eventos e ideias, e ignorar outros que também eram importantes. Organizamos o histórico em torno de uma série de perguntas. Certamente, não desejaríamos dar a impressão de que essas questões são as únicas de que as disciplinas tratam ou que todas as disciplinas estejam se encaminhando para a IA como sua realização final.

### 1.2.1 Filosofia

- Regras formais podem ser usadas para obter conclusões válidas?
- Como a mente (o intelecto) se desenvolve a partir de um cérebro físico?
- De onde vem o conhecimento?
- Como o conhecimento conduz à ação?

Aristóteles (384-322 a.C.), cujo busto aparece na capa deste livro, foi o primeiro a formular um conjunto preciso de leis que governam a parte racional da mente. Ele desenvolveu um sistema informal de silogismos para raciocínio apropriado que, em princípio, permitiam gerar conclusões mecanicamente, dadas as premissas iniciais. Muito mais tarde, Ramon Lull (1315) apresentou a ideia de que o raciocínio útil poderia na realidade ser conduzido por um artefato mecânico. Thomas Hobbes (1588-1679) propôs que o raciocínio era semelhante à computação numérica, ou seja, que “efetuamos somas e subtrações em nossos pensamentos silenciosos”. A automação da própria computação já estava bem próxima; por volta de 1500, Leonardo da Vinci (1452-1519) projetou, mas não construiu, uma calculadora mecânica; reconstruções recentes mostraram que o projeto era funcional. A primeira máquina de calcular conhecida foi construída em torno de 1623 pelo cientista alemão Wilhelm Schickard (1592-1635), embora a Pascaline, construída em 1642 por Blaise Pascal (1623-1662), seja mais famosa. Pascal escreveu que “a máquina aritmética produz efeitos que parecem mais próximos ao pensamento que todas as ações dos animais”. Gottfried Wilhelm Leibnitz (1646-1716) construiu um dispositivo mecânico destinado a efetuar operações sobre conceitos, e não sobre números, mas seu escopo era bastante limitado. Leibnitz superou Pascal através da construção de uma calculadora que podia somar, subtrair, multiplicar e extrair raízes, enquanto a Pascaline só podia adicionar e subtrair. Alguns especularam que as máquinas não poderiam fazer apenas cálculos, mas realmente ser capazes de pensar e agir por conta própria. Em seu livro de 1651, *Leviatã*, Thomas Hobbes sugeriu a ideia de um “animal artificial”, argumentando: “Pois o que é o coração, senão uma mola; e os nervos, senão tantas cordas; e as articulações, senão tantas rodas.”

Dizer que a mente opera, pelo menos em parte, de acordo com regras lógicas e construir sistemas físicos que emulam algumas dessas regras é uma coisa; outra é dizer que a mente em si é esse sistema físico. René Descartes (1596-1650) apresentou a primeira discussão clara da distinção entre mente e matéria, e dos problemas que surgem dessa distinção. Um dos problemas relacionados com uma concepção puramente física da mente é o fato de que ela parece deixar pouco espaço para o livre-arbítrio: se a mente é governada inteiramente por leis físicas, então ela não tem mais livre-arbítrio que uma pedra que “decide” cair em direção ao centro da Terra. Descartes advogava fortemente a favor do poder da razão em entender o mundo, uma filosofia hoje chamada de **racionalismo**, e que tinha Aristóteles e Leibnitz como membros. Descartes também era um proponente do **dualismo**. Ele

sustentava que havia uma parte da mente humana (ou alma, ou espírito) que transcende a natureza, isenta das leis físicas. Por outro lado, os animais não possuem essa qualidade dual; eles podiam ser tratados como máquinas. Uma alternativa para o dualismo é o **materialismo**. O materialismo sustenta que a operação do cérebro de acordo com as leis da física *constitui* a mente. O livre-arbítrio é simplesmente o modo como a percepção das escolhas disponíveis se mostra para a entidade que escolhe.

Dada uma mente física que manipula o conhecimento, o próximo problema é estabelecer a origem do conhecimento. O movimento chamado **empirismo**, iniciado a partir da obra de Francis Bacon (1561-1626), *Novum Organum*,<sup>2</sup> se caracterizou por uma frase de John Locke (1632-1704): “Não há nada na compreensão que não estivesse primeiro nos sentidos.” A obra de David Hume (1711-1776), *A Treatise of Human Nature* (Hume, 1739) propôs aquilo que se conhece hoje como o princípio de **indução**: as regras gerais são adquiridas pela exposição a associações repetidas entre seus elementos. Com base no trabalho de Ludwig Wittgenstein (1889-1951) e Bertrand Russell (1872-1970), o famoso Círculo de Viena, liderado por Rudolf Carnap (1891-1970), desenvolveu a doutrina do **positivismo lógico**. Essa doutrina sustenta que todo conhecimento pode ser caracterizado por teorias lógicas conectadas, em última análise, a **sentenças de observação** que correspondem a entradas sensoriais; desse modo, o positivismo lógico combina o racionalismo e o empirismo.<sup>3</sup> A **teoria da confirmação** de Carnap e Carl Hempel (1905-1997) tentava compreender a aquisição do conhecimento através da experiência. O livro de Carnap, *The Logical Structure of the World* (1928), definiu um procedimento computacional explícito para extrair conhecimento de experiências elementares. Provavelmente, foi a primeira teoria da mente como um processo computacional.

O último elemento no quadro filosófico da mente é a conexão entre conhecimento e ação. Essa questão é vital para a IA porque a inteligência exige ação, bem como raciocínio. Além disso, apenas pela compreensão de como as ações são justificadas podemos compreender como construir um agente cujas ações sejam justificáveis (ou racionais). Aristóteles argumentava (no *De Motu Animalium*) que as ações se justificam por uma conexão lógica entre metas e conhecimento do resultado da ação (a última parte deste extrato também aparece na capa deste livro, no original em grego):

Porém, como explicar que o pensamento às vezes esteja acompanhado pela ação e às vezes não, às vezes esteja acompanhado pelo movimento e outras vezes não? Aparentemente, acontece quase o mesmo no caso do raciocínio e na realização de inferências sobre objetos imutáveis. Contudo, nesse caso o fim é uma proposição especulativa (...) enquanto aqui a conclusão que resulta das duas premissas é uma ação. (...) Preciso me cobrir; um casaco é uma coberta. Preciso de um casaco. O que eu preciso, tenho de fazer; preciso de um casaco. Tenho de fazer um casaco. E a conclusão, “tenho de fazer um casaco”, é uma ação.

Na obra *Ética a Nicômaco* (Livro III. 3, 1112b), Aristóteles desenvolve esse tópico um pouco mais, sugerindo um algoritmo:

Não deliberamos sobre os fins, mas sobre os meios. Um médico não delibera sobre se deve ou não curar nem um orador sobre se deve ou não persuadir, (...) Eles dão a finalidade por estabelecida e procuram saber a maneira de alcançá-la; se lhes parece poder ser alcançada por vários meios,

procuram saber o mais fácil e o mais eficaz; e se há apenas um meio para alcançá-la, procuram saber *como* será alcançada por esse meio e por que outro meio alcançar *esse* primeiro, até chegar ao primeiro princípio, que é o último na ordem de descoberta. (...) e o que vem em último lugar na ordem da análise parece ser o primeiro na ordem da execução. E, se chegarmos a uma impossibilidade, abandonamos a busca; por exemplo, se precisarmos de dinheiro e não for possível consegui-lo; porém, se algo parecer possível, tentaremos realizá-lo.<sup>4</sup>

O algoritmo de Aristóteles foi implementado 2.300 anos mais tarde, por Newell e Simon, em seu programa GPS. Agora, poderíamos denominá-lo sistema de planejamento regressivo (ver o Capítulo 10.)

A análise baseada em metas é útil, mas não nos diz o que fazer quando várias ações alcançarem a meta ou quando nenhuma ação a alcançar por completo. Antoine Arnauld (1612-1694) descreveu corretamente uma fórmula quantitativa para definir que ação executar em casos como esse (ver o Capítulo 16). O livro de John Stuart Mill (1806-1873), *Utilitarianism* (Mill, 1863), promoveu a ideia de critérios de decisão racionais em todas as esferas da atividade humana. A teoria de decisões é mais formalmente discutida na próxima seção.

## 1.2.2 Matemática

- Quais são as regras formais para obter conclusões válidas?
- O que pode ser computado?
- Como raciocinamos com informações incertas?

Os filósofos demarcaram a maioria das ideias importantes sobre a IA, mas o salto para uma ciência formal exigiu certo nível de formalização matemática em três áreas fundamentais: lógica, computação e probabilidade.

A ideia de lógica formal pode ser traçada até os filósofos da Grécia antiga, mas seu desenvolvimento matemático começou realmente com o trabalho de George Boole (1815-1864), que definiu os detalhes da lógica proposicional ou lógica booleana (Boole, 1847). Em 1879, Gottlob Frege (1848-1925) estendeu a lógica de Boole para incluir objetos e relações, criando a lógica de primeira ordem que é utilizada hoje.<sup>5</sup> Alfred Tarski (1902-1983) introduziu uma teoria de referência que mostra como relacionar os objetos de uma lógica a objetos do mundo real.

A próxima etapa foi determinar os limites do que poderia ser feito com a lógica e a computação. Acredita-se que o primeiro **algoritmo** não trivial seja o algoritmo de Euclides para calcular o maior divisor comum. A palavra *algoritmo* (e a ideia de estudá-lo) vem de Al-Khowarazmi, um matemático persa do século IX, cujos escritos também introduziram os numerais arábicos e a álgebra na Europa. Boole e outros discutiram algoritmos para dedução lógica e, no final do século XIX, foram empreendidos esforços para formalizar o raciocínio matemático geral como dedução lógica. Em 1930, Kurt Gödel (1906-1978) mostrou que existe um procedimento efetivo para provar qualquer afirmação verdadeira na lógica de primeira ordem de Frege e Russell, mas essa lógica não poderia captar o princípio de indução matemática necessário para caracterizar os números naturais. Em 1931, Gödel mostrou que existem de fato limites sobre dedução. Seu **teorema da incompletude** mostrou

que, em qualquer teoria formal tão forte como a aritmética de Peano (a teoria elementar dos números naturais), existem afirmações verdadeiras que são indecidíveis no sentido de que não existem provas na teoria.

Esse resultado fundamental também pode ser interpretado como a demonstração de que existem algumas funções sobre os inteiros que não podem ser representadas por um algoritmo, isto é, não podem ser calculadas. Isso motivou Alan Turing (1912-1954) a tentar caracterizar exatamente que funções **são computáveis** — capazes de ser computáveis. Na realidade, essa noção é ligeiramente problemática porque a noção de computação ou de procedimento efetivo realmente não pode ter uma definição formal. No entanto, a tese de Church-Turing, que afirma que a máquina de Turing (Turing, 1936) é capaz de calcular qualquer função computável, em geral é aceita como definição suficiente. Turing também mostrou que existiam algumas funções que nenhuma máquina de Turing poderia calcular. Por exemplo, nenhuma máquina pode determinar, *de forma geral*, se dado programa retornará uma resposta sobre certa entrada ou se continuará funcionando para sempre.

Embora a decidibilidade e a computabilidade sejam importantes para a compreensão da computação, a noção de **tratabilidade** teve um impacto muito maior. Em termos gerais, um problema é chamado de intratável se o tempo necessário para resolver instâncias dele cresce exponencialmente com o tamanho das instâncias. A distinção entre crescimento polinomial e exponencial da complexidade foi enfatizada primeiro em meados da década de 1960 (Cobham, 1964; Edmonds, 1965). Ela é importante porque o crescimento exponencial significa que até mesmo instâncias moderadamente grandes não podem ser resolvidas em qualquer tempo razoável. Portanto, devemos procurar dividir o problema global de geração de comportamento inteligente em subproblemas tratáveis, em vez de subproblemas intratáveis.

Como é possível reconhecer um problema intratável? A teoria da **NP-completude**, apresentada primeiro por Steven Cook (1971) e Richard Karp (1972), fornece um método. Cook e Karp demonstraram a existência de grandes classes de problemas canônicos de busca combinatória e de raciocínio que são NP-completos. Qualquer classe de problemas à qual a classe de problemas NP-completos pode ser reduzida provavelmente é intratável (embora não tenha sido provado que problemas NP-completos são necessariamente intratáveis, a maioria dos teóricos acredita nisso). Esses resultados contrastam com o otimismo com que a imprensa popular saudou os primeiros computadores — “Supercérebros eletrônicos” que eram “Mais rápidos que Einstein!”. Apesar da crescente velocidade dos computadores, o uso parcimonioso de recursos é que caracterizará os sistemas inteligentes. *Grosso modo*, o mundo é uma instância de um problema *extremamente* grande! Trabalhar com IA ajudou a explicar por que algumas instâncias de problemas NP-completos são difíceis, enquanto outras são fáceis (Cheeseman *et al.*, 1991).

Além da lógica e da computação, a terceira grande contribuição da matemática para a IA é a teoria da **probabilidade**. O italiano Gerolamo Cardano (1501-1576) foi o primeiro a conceber a ideia de probabilidade, descrevendo-a em termos dos resultados possíveis de jogos de azar. Em 1654, Blaise Pascal (1623-1662), numa carta para Pierre Fermat (1601-1665), mostrou como predizer o futuro de um jogo de azar inacabado e atribuir recompensas médias aos jogadores. A probabilidade se transformou rapidamente em uma parte valiosa de todas as ciências quantitativas, ajudando a lidar com medidas incertas e teorias incompletas. James Bernoulli (1654-1705), Pierre Laplace (1749-1827) e outros pesquisadores aperfeiçoaram a teoria e introduziram novos métodos estatísticos.

Thomas Bayes (1702-1761), que aparece na capa deste livro, propôs uma regra para atualizar probabilidades à luz de novas evidências. A regra de Bayes e o campo resultante chamado análise bayesiana formam a base da maioria das abordagens modernas para raciocínio incerto em sistemas de IA.

### 1.2.3 Economia

- Como devemos tomar decisões para maximizar a recompensa?
- Como devemos fazer isso quando outros não podem nos acompanhar?
- Como devemos fazer isso quando a recompensa pode estar distante no futuro?

A ciência da economia teve início em 1776, quando o filósofo escocês Adam Smith (1723-1790) publicou *An Inquiry into the Nature and Causes of the Wealth of Nations*. Embora os antigos gregos e outros filósofos tenham contribuído para o pensamento econômico, Smith foi o primeiro a tratá-lo como ciência, usando a ideia de que podemos considerar que as economias consistem em agentes individuais que maximizam seu próprio bem-estar econômico. A maioria das pessoas pensa que a economia trata de dinheiro, mas os economistas dirão que, na realidade, a economia estuda como as pessoas fazem escolhas que levam a resultados preferenciais. Quando o McDonalds oferece um hambúrguer por um dólar, está afirmando que prefere o dólar e espera que os clientes prefiram o hambúrguer. O tratamento matemático de “resultados preferenciais” ou **utilidade** foi formalizado primeiro por Léon Walras (1834-1910) e aperfeiçoados por Frank Ramsey (1931) e, mais tarde, por John von Neumann e Oskar Morgenstern em seu livro *The Theory of Games and Economic Behavior* (1944).

A **teoria da decisão**, que combina a teoria da probabilidade com a teoria da utilidade, fornece uma estrutura formal e completa para decisões (econômicas ou outras) tomadas sob a incerteza, ou seja, em casos nos quais as descrições probabilísticas captam de forma apropriada o ambiente do tomador de decisões. Isso é adequado para “grandes” economias em que cada agente não precisa levar em conta as ações de outros agentes como indivíduos. No caso das “pequenas” economias, a situação é muito mais parecida com um **jogo**: as ações de um jogador podem afetar de forma significativa a utilidade de outro (positiva ou negativamente). O desenvolvimento da **teoria dos jogos** por Von Neumann e Morgenstern (consulte também Luce e Raiffa, 1957) incluiu o surpreendente resultado de que, em alguns jogos, um agente racional deve adotar políticas que são (ou pelo menos parecem ser) aleatórias. Ao contrário da teoria da decisão, a teoria dos jogos não oferece uma receita inequívoca para a seleção de ações.

De modo geral, os economistas não trataram a terceira questão da listagem anterior, ou seja, como tomar decisões racionais quando as recompensas das ações não são imediatas, mas resultam de várias ações executadas *em sequência*. Esse tópico foi adotado no campo de **pesquisa operacional**, que emergiu na Segunda Guerra Mundial dos esforços britânicos para otimizar instalações de radar e, mais tarde, encontrou aplicações civis em decisões complexas de administração. O trabalho de Richard Bellman (1957) formalizou uma classe de problemas de decisão sequencial chamados **processos de decisão de Markov**, que estudaremos nos Capítulos 17 e 21.

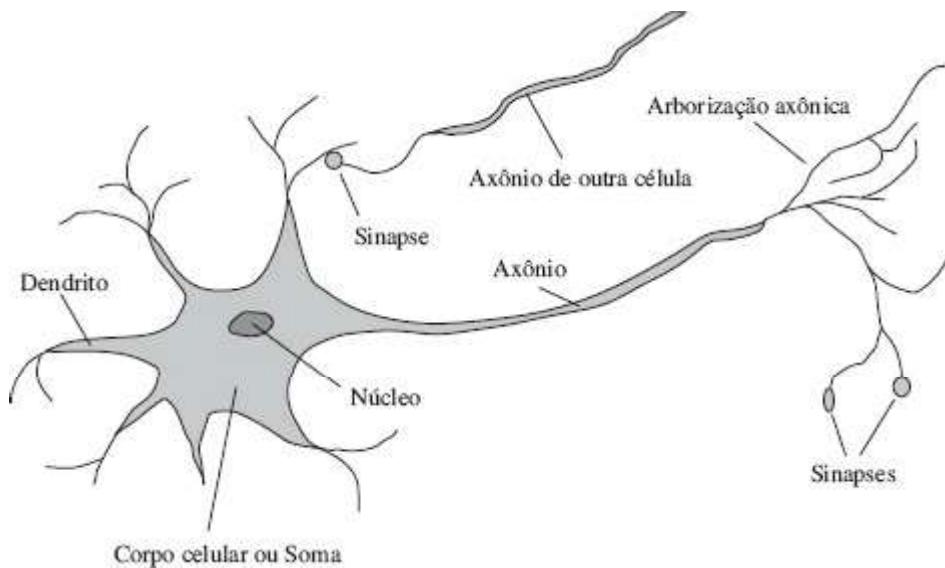
O trabalho em economia e pesquisa operacional contribuiu muito para nossa noção de agentes racionais, ainda que por muitos anos a pesquisa em IA se desenvolvesse ao longo de caminhos inteiramente separados. Uma razão para isso era a aparente **complexidade** da tomada de decisões racionais. Herbert Simon (1916-2001), o pesquisador pioneiro da IA, ganhou o Prêmio Nobel de economia em 1978 por seu trabalho inicial demonstrando que modelos baseados em **satisfação** — a tomada de decisões “boas o suficiente”, em vez de calcular laboriosamente uma decisão ótima — forneciam uma descrição melhor do comportamento humano real (Simon, 1947). Desde os anos 1990, ressurgiu o interesse pelas técnicas da teoria da decisão para sistemas de agentes (Wellman, 1995).

## 1.2.4 Neurociência

- Como o cérebro processa informações?

A **neurociência** é o estudo do sistema nervoso, em particular do cérebro. Apesar de o modo exato como o cérebro habilita o pensamento ser um dos grandes mistérios da ciência, o fato de ele *habilitar* o pensamento foi avaliado por milhares de anos devido à evidência de que pancadas fortes na cabeça podem levar à incapacitação mental. Também se sabe há muito tempo que o cérebro dos seres humanos tem algumas características diferentes; em aproximadamente 335 a.C., Aristóteles escreveu: “De todos os animais, o homem é o que tem o maior cérebro em proporção ao seu tamanho.”<sup>6</sup> Ainda assim, apenas em meados do século XVIII o cérebro foi amplamente reconhecido como a sede da consciência. Antes disso, acreditava-se que a sede da consciência poderia estar localizada no coração e no baço.

O estudo da afasia (deficiência da fala) feito por Paul Broca (1824-1880) em 1861, com pacientes cujo cérebro foi danificado, demonstrou a existência de áreas localizadas do cérebro responsáveis por funções cognitivas específicas. Em particular, ele mostrou que a produção da fala estava localizada em uma parte do hemisfério cerebral esquerdo agora chamada área de Broca.<sup>7</sup> Nessa época, sabia-se que o cérebro consistia em células nervosas ou **neurônios**, mas apenas em 1873 Camillo Golgi (1843-1926) desenvolveu uma técnica de coloração que permitiu a observação de neurônios individuais no cérebro (ver a Figura 1.2). Essa técnica foi usada por Santiago Ramon y Cajal (1852-1934) em seus estudos pioneiros das estruturas de neurônios do cérebro.<sup>8</sup> Nicolas Rashevsky (1936, 1938) foi o primeiro a aplicar modelos matemáticos ao estudo do sistema nervoso.



**Figura 1.2** Partes de uma célula nervosa ou neurônio. Cada neurônio consiste em um corpo celular ou soma, que contém um núcleo celular. Ramificando-se a partir do corpo celular, há uma série de fibras chamadas dendritos e uma única fibra longa chamada axônio. O axônio se estende por uma longa distância, muito mais longa do que indica a escala desse diagrama. Em geral, um axônio têm 1 cm de comprimento (100 vezes o diâmetro do corpo celular), mas pode alcançar até 1 metro. Um neurônio faz conexões com 10-100.000 outros neurônios, em junções chamadas sinapses. Os sinais se propagam de um neurônio para outro por meio de uma complicada reação eletroquímica. Os sinais controlam a atividade cerebral em curto prazo e também permitem mudanças a longo prazo na posição e na conectividade dos neurônios. Acredita-se que esses mecanismos formem a base para o aprendizado no cérebro. A maior parte do processamento de informações ocorre no córtex cerebral, a camada exterior do cérebro. A unidade organizacional básica parece ser uma coluna de tecido com aproximadamente 0,5 mm de diâmetro, contendo cerca de 20.000 neurônios e estendendo-se por toda a profundidade do córtex, cerca de 4 mm nos seres humanos.

Atualmente, temos alguns dados sobre o mapeamento entre áreas do cérebro e as partes do corpo que elas controlam ou das quais recebem entrada sensorial. Tais mapeamentos podem mudar radicalmente no curso de algumas semanas, e alguns animais parecem ter vários mapas. Além disso, não compreendemos inteiramente como outras áreas do cérebro podem assumir o comando de certas funções quando uma área é danificada. Praticamente não há teoria que explique como a memória de um indivíduo é armazenada.

A medição da atividade do cérebro intacto teve início em 1929, com a invenção do eletroencefalógrafo (EEG) por Hans Berger. O desenvolvimento recente do processamento de imagens por ressonância magnética funcional (fMRI — *functional Magnetic Resonance Imaging*) (Ogawa *et al.*, 1990; Cabeza e Nyberg, 2001) está dando aos neurocientistas imagens sem precedentes de detalhes da atividade do cérebro, tornando possíveis medições que correspondem em aspectos interessantes a processos cognitivos em ação. Essas medições são ampliadas por avanços na gravação da atividade dos neurônios em uma única célula. Os neurônios individuais podem ser estimulados eletricamente, quimicamente ou mesmo opticamente (Han e Boyden, 2007), permitindo que os relacionamentos neuronais de entrada-saída sejam mapeados. Apesar desses avanços, ainda estamos longe de compreender como realmente funciona qualquer desses processos cognitivos.

 A conclusão verdadeiramente espantosa é que *uma coleção de células simples pode levar ao pensamento, à ação e à consciência* ou, nas palavras incisivas de John Searle (1992), os cérebros geram mentes. A única teoria alternativa real é o misticismo, que significa operar em algum reino místico que está além da ciência física.

De alguma forma cérebros e computadores digitais têm propriedades diferentes. A Figura 1.3 mostra que os computadores têm um ciclo de tempo que é um milhão de vezes mais rápido que o cérebro. O cérebro é composto por muito mais capacidade de armazenamento e interconexões que um computador pessoal de última geração, apesar de os maiores supercomputadores apresentarem uma capacidade similar a do cérebro. Entretanto, observe que o cérebro não parece usar todos os seus neurônios simultaneamente. Os futuristas enaltecem demais esses números, apontando para uma próxima **singularidade** em que os computadores alcançariam um nível sobrehumano de performance (Vinge, 1993; Kurzweil, 2005), mas as comparações cruas não são especialmente informativas. Mesmo com um computador com capacidade ilimitada, não saberíamos como atingir o nível de inteligência do cérebro.

	Supercomputador	Computador pessoal	Mente humana
Unidades computacionais	$10^4$ CPUs, $10^{12}$ transistores	$4$ CPUs, $10^9$ transistores	$10^{11}$ neurônios
Unidades de armazenamento	$10^{14}$ bits RAM $10^{15}$ bits disco	$10^{11}$ bits $10^{13}$ RAM bits disco	$10^{11}$ neurônios $10^{14}$ sinapses
Tempo de ciclo	$10^{-9}$ seg	$10^{-9}$ seg	$10^{-3}$ seg
Operações/seg	$10^{15}$	$10^{10}$	$10^{17}$
Atualizações de memória/seg	$10^{14}$	$10^{10}$	$10^{14}$

**Figura 1.3** Comparação grosseira dos recursos computacionais brutos disponíveis entre o supercomputador Blue Gene da IBM, um computador pessoal típico de 2008 e o cérebro humano. Os números do cérebro são fixos essencialmente, enquanto os números do supercomputador crescem por um fator de 10, mais ou menos a cada cinco anos, permitindo-lhe alcançar paridade aproximada com o cérebro. O computador pessoal está atrasado em todas as métricas, exceto no tempo de ciclo.

## 1.2.5 Psicologia

- Como os seres humanos e os animais pensam e agem?

Normalmente, considera-se que as origens da psicologia científica remontam ao trabalho do físico alemão Hermann von Helmholtz (1821-1894) e de seu aluno Wilhelm Wundt (1832-1920). Helmholtz aplicou o método científico ao estudo da visão humana, e seu *Handbook of Physiological Optics* é descrito até hoje como “o mais importante tratado sobre a física e a fisiologia da visão humana” (Nalwa, 1993, p. 15). Em 1879, Wundt abriu o primeiro laboratório de psicologia experimental na

Universidade de Leipzig. Ele insistia em experimentos cuidadosamente controlados, nos quais seus colaboradores executariam uma tarefa perceptiva ou associativa enquanto refletiam sobre seus processos de pensamento. O controle cuidadoso percorreu um longo caminho para transformar a psicologia em ciência, mas a natureza subjetiva dos dados tornava improvável que um pesquisador divergisse de suas próprias teorias. Por outro lado, os biólogos que estudavam o comportamento animal careciam de dados introspectivos e desenvolveram uma metodologia objetiva, como descreveu H. S. Jennings (1906) em seu influente trabalho *Behavior of the Lower Organisms*. Aplicando esse ponto de vista aos seres humanos, o movimento chamado **behaviorismo**, liderado por John Watson (1878-1958), rejeitava *qualquer* teoria que envolvesse processos mentais com base no fato de que a introspecção não poderia fornecer evidência confiável. Os behavioristas insistiam em estudar apenas medidas objetivas das percepções (ou *estímulos*) dados a um animal e suas ações resultantes (ou *respostas*). O behaviorismo descobriu muito sobre ratos e pombos, mas teve menos sucesso na compreensão dos seres humanos.

A visão do cérebro como um dispositivo de processamento de informações, uma característica importante da **psicologia cognitiva**, tem suas origens nos trabalhos de William James (1842-1910). Helmholtz também insistiu que a percepção envolvia uma forma de inferência lógica inconsciente. O ponto de vista cognitivo foi em grande parte eclipsado pelo behaviorismo nos Estados Unidos, mas na Unidade de Psicologia Aplicada de Cambridge, dirigida por Frederic Bartlett (1886-1969), a modelagem cognitiva foi capaz de florescer. *The Nature of Explanation*, de Kenneth Craik (1943), aluno e sucessor de Bartlett, restabeleceu com vigor a legitimidade de termos “mentais” como crenças e objetivos, argumentando que eles são tão científicos quanto, digamos, usar a pressão e a temperatura ao falar sobre gases, apesar de eles serem constituídos por moléculas que não têm nenhuma dessas duas propriedades. Craik especificou os três passos fundamentais de um agente baseado no conhecimento: (1) o estímulo deve ser traduzido em uma representação interna, (2) a representação é manipulada por processos cognitivos para derivar novas representações internas e (3) por sua vez, essas representações são de novo traduzidas em ações. Ele explicou com clareza por que esse era um bom projeto de um agente:

Se o organismo transporta um “modelo em escala reduzida” da realidade externa e de suas próprias ações possíveis dentro de sua cabeça, ele é capaz de experimentar várias alternativas, concluir qual a melhor delas, reagir a situações futuras antes que elas surjam, utilizar o conhecimento de eventos passados para lidar com o presente e o futuro e, em todos os sentidos, reagir de maneira muito mais completa, segura e competente às emergências que enfrenta. (Craik, 1943)

Após a morte de Craik, em um acidente de bicicleta em 1945, seu trabalho teve continuidade com Donald Broadbent, cujo livro *Perception and Communication* (1958) foi um dos primeiros trabalhos a modelar fenômenos psicológicos como processamento de informações. Enquanto isso, nos Estados Unidos, o desenvolvimento da modelagem de computadores levou à criação do campo da **ciência cognitiva**. Pode-se dizer que o campo teve início em um seminário em setembro de 1956 no MIT (veremos que esse seminário ocorreu apenas dois meses após a conferência em que a própria IA “nasceu”). No seminário, George Miller apresentou *The Magic Number Seven*, Noam Chomsky apresentou *Three Models of Language* e Allen Newell e Herbert Simon apresentaram *The Logic*

*Theory Machine.* Esses três documentos influentes mostraram como modelos de computadores podiam ser usados para tratar a psicologia da memória, a linguagem e o pensamento lógico, respectivamente. Agora é comum entre os psicólogos a visão de que “uma teoria cognitiva deve ser como um programa de computador” (Anderson, 1980), isto é, ela deve descrever um mecanismo detalhado de processamento de informações por meio do qual alguma função cognitiva poderia ser implementada.

## 1.2.6 Engenharia de computadores

- Como podemos construir um computador eficiente?

Para a inteligência artificial ter sucesso, precisamos de inteligência e de um artefato. O computador tem sido o artefato preferido. O computador eletrônico digital moderno foi criado independentemente e quase ao mesmo tempo por cientistas de três países que participavam da Segunda Guerra Mundial. O primeiro computador *operacional* foi a máquina eletromecânica de Heath Robinson,<sup>9</sup> construída em 1940 pela equipe de Alan Turing com um único propósito: decifrar mensagens alemãs. Em 1943, o mesmo grupo desenvolveu o Colossus, uma poderosa máquina de uso geral baseada em válvulas eletrônicas.<sup>10</sup> O primeiro computador *programável* operacional foi o Z-3, criado por Konrad Zuse na Alemanha, em 1941. Zuse também criou os números de ponto flutuante e a primeira linguagem de programação de alto nível, denominada Plankalkul. O primeiro computador *eletrônico*, o ABC, foi montado por John Atanasoff e por seu aluno Clifford Berry, entre 1940 e 1942, na Iowa State University. A pesquisa de Atanasoff recebeu pouco apoio ou reconhecimento; foi o ENIAC, desenvolvido como parte de um projeto militar secreto na University of Pennsylvania por uma equipe que incluía John Mauchly e John Eckert, que provou ser o precursor mais influente dos computadores modernos.

Desde aquele tempo, cada geração de hardware de computador trouxe aumento em velocidade e capacidade, e redução no preço. O desempenho é duplicado a cada 18 meses aproximadamente, até por volta de 2005, quando os problemas de dissipação de energia levaram os fabricantes a começar a multiplicação do número de núcleos de CPU e não a velocidade de clock. Espera-se, atualmente, que futuros aumentos de energia venham de um paralelismo maciço, uma convergência curiosa com as propriedades do cérebro.

É claro que existiam dispositivos de cálculo antes do computador eletrônico. As primeiras máquinas automatizadas, datando do século XVII, foram descritas na página 6. A primeira máquina *programável* foi um tear criado em 1805 por Joseph Marie Jacquard (1752-1834), que utilizava cartões perfurados para armazenar instruções relativas ao padrão a ser tecido. Na metade do século XIX, Charles Babbage (1792-1871) projetou duas máquinas, mas não concluiu nenhuma delas. A “máquina diferencial” se destinava a calcular tabelas matemáticas para projetos de engenharia e científicos. Ela foi finalmente construída e se mostrou funcional em 1991 no Science Museum em Londres (Swade, 2000). A “máquina analítica” de Babbage era bem mais ambiciosa: ela incluía memória endereçável, programas armazenados e saltos condicionais, e foi o primeiro artefato capaz de executar computação universal. A colega de Babbage, Ada Lovelace, filha do poeta Lord Byron, talvez tenha sido a primeira programadora do mundo (a linguagem de programação Ada recebeu esse

nome em homenagem a ela). Ela escreveu programas para a máquina analítica não concluída e até mesmo especulou que a máquina poderia jogar xadrez ou compor música.

A IA também tem uma dívida com a área de software da ciência da computação, que forneceu os sistemas operacionais, as linguagens de programação e as ferramentas necessárias para escrever programas modernos (e artigos sobre eles). Porém, essa é uma área em que a dívida foi paga: o trabalho em IA foi pioneiro em muitas ideias que foram aproveitadas posteriormente na ciência da computação em geral, incluindo compartilhamento de tempo, interpretadores interativos, computadores pessoais com janelas e mouse, ambientes de desenvolvimento rápido, tipo de dados de lista ligada, gerenciamento automático de armazenamento e conceitos fundamentais de programação simbólica, funcional, declarativa e orientada a objetos.

## 1.2.7 Teoria de controle e cibernetica

- Como os artefatos podem operar sob seu próprio controle?

Ctesíbio de Alexandria (cerca de 250 a.C.) construiu a primeira máquina autocontrolada: um relógio de água com um regulador que mantinha uma taxa de fluxo constante. Essa invenção mudou a definição do que um artefato poderia fazer. Antes, somente os seres vivos podiam modificar seu comportamento em resposta a mudanças no ambiente. Outros exemplos de sistemas de controle realimentados autorreguláveis incluem o regulador de máquinas a vapor, criado por James Watt (1736-1819), e o termostato, criado por Cornelis Drebbel (1572-1633), que também inventou o submarino. A teoria matemática de sistemas realimentados estáveis foi desenvolvida no século XIX.

A figura central na criação daquilo que se conhece hoje como **teoria de controle** foi Norbert Wiener (1894-1964). Wiener foi um matemático brilhante que trabalhou com Bertrand Russell, entre outros, antes de se interessar por sistemas de controle biológico e mecânico e sua conexão com a cognição. Como Craik (que também utilizou sistemas de controle como modelos psicológicos), Wiener e seus colegas Arturo Rosenblueth e Julian Bigelow desafiaram a ortodoxia behaviorista (Rosenblueth *et al.*, 1943). Eles viram o comportamento consciente como o resultado de um mecanismo regulador tentando minimizar o “erro” — a diferença entre o estado atual e o estado objetivo. No final da década de 1940, Wiener, juntamente com Warren McCulloch, Walter Pitts e John von Neumann, organizou uma série de conferências que influenciou os novos modelos matemáticos e computacionais da cognição. O livro de Wiener, *Cybernetics* (1948), tornou-se *best-seller* e despertou o público para a possibilidade de máquinas dotadas de inteligência artificial. Enquanto isso, na Grã-Bretanha, W. Ross Ashby (Ashby, 1940) foi pioneiro em ideias semelhantes. Ashby, Alan Turing, Grey Walter e outros formaram o Ratio Club para “aqueles que tinham as ideias de Wiener antes de surgir o livro de Wiener”. *Design for a Brain* (1948, 1952), de Ashby, elaborava a sua ideia de que a mente poderia ser criada com a utilização de mecanismos **homeostáticos** contendo laços de realimentação para atingir comportamento adaptável estável.

A moderna teoria de controle, em especial o ramo conhecido como controle estocástico ótimo, tem como objetivo o projeto de sistemas que maximizam uma **função objetivo** sobre o tempo. Isso corresponde aproximadamente à nossa visão da IA: projetar sistemas que se comportem de maneira

ótima. Então, por que a IA e a teoria de controle são dois campos diferentes, apesar das conexões estreitas entre seus fundadores? A resposta reside no acoplamento estrito entre as técnicas matemáticas familiares aos participantes e os conjuntos de problemas correspondentes que foram incluídos em cada visão do mundo. O cálculo e a álgebra de matrizes, as ferramentas da teoria de controle, eram adequados para sistemas que podem ser descritos por conjuntos fixos de variáveis contínuas, enquanto a IA foi criada em parte como um meio de escapar das limitações percebidas. As ferramentas de inferência lógica e computação permitiram que os pesquisadores da IA considerassem alguns problemas como linguagem, visão e planejamento, que ficavam completamente fora do campo de ação da teoria de controle.

## 1.2.8 Linguística

- Como a linguagem se relaciona com o pensamento?

Em 1957, B. F. Skinner publicou *Verbal Behavior*. Essa obra foi uma descrição completa e detalhada da abordagem behaviorista para o aprendizado da linguagem, escrita pelo mais proeminente especialista no campo. Porém, curiosamente, uma crítica do livro se tornou tão conhecida quanto o próprio livro e serviu para aniquilar o interesse pelo behaviorismo. O autor da resenha foi o linguista Noam Chomsky, que tinha acabado de publicar um livro sobre sua própria teoria, *Syntactic Structures (Estruturas sintáticas)*. Chomsky chamou a atenção para o fato de que a teoria behaviorista não tratava da noção de criatividade na linguagem — ela não explicava como uma criança podia compreender e formar frases que nunca tinha ouvido antes. A teoria de Chomsky — baseada em modelos sintáticos criados pelo linguista indiano Panini (c. 350 a.C.) — podia explicar esse fato e, diferentemente das teorias anteriores, era formal o bastante para poder, em princípio, ser programada.

Portanto, a linguística moderna e a IA “nasceram” aproximadamente na mesma época e cresceram juntas, cruzando-se em um campo híbrido chamado **linguística computacional** ou **processamento de linguagem natural**. O problema de compreender a linguagem logo se tornou consideravelmente mais complexo do que parecia em 1957. A compreensão da linguagem exige a compreensão do assunto e do contexto, não apenas a compreensão da estrutura das frases. Isso pode parecer óbvio, mas só foi amplamente avaliado na década de 1960. Grande parte do trabalho anterior em **representação do conhecimento** (o estudo de como colocar o conhecimento em uma forma que um computador possa utilizar) estava vinculado à linguagem e era suprido com informações da pesquisa em linguística que, por sua vez, estava conectada a décadas de pesquisa sobre a análise filosófica da linguagem.

## 1.3 HISTÓRIA DA INTELIGÊNCIA ARTIFICIAL

---

Com o material que vimos até agora, estamos prontos para estudar o desenvolvimento da própria IA.

### 1.3.1 A gestação da inteligência artificial (1943-1955)

O primeiro trabalho agora reconhecido como IA foi realizado por Warren McCulloch e Walter Pitts (1943). Eles se basearam em três fontes: o conhecimento da fisiologia básica e da função dos neurônios no cérebro; uma análise formal da lógica proposicional criada por Russell e Whitehead; e a teoria da computação de Turing. Esses dois pesquisadores propuseram um modelo de neurônios artificiais, no qual cada neurônio se caracteriza por estar “ligado” ou “desligado”, com a troca para “ligado” ocorrendo em resposta à estimulação por um número suficiente de neurônios vizinhos. O estado de um neurônio era considerado “equivalente em termos concretos a uma proposição que definia seu estímulo adequado”. Por exemplo, eles mostraram que qualquer função computável podia ser calculada por certa rede de neurônios conectados e que todos os conectivos lógicos (e, ou, não etc.) podiam ser implementados por estruturas de redes simples. McCulloch e Pitts também sugeriram que redes definidas adequadamente seriam capazes de aprender. Donald Hebb (1949) demonstrou uma regra de atualização simples para modificar as intensidades de conexão entre neurônios. Sua regra, agora chamada **aprendizado de Hebb**, continua a ser um modelo influente até hoje.

Dois alunos de Harvard, Marvin Minsky e Dean Edmonds, construíram o primeiro computador de rede neural em 1950. O SNARC, como foi chamado, usava 3.000 válvulas eletrônicas e um mecanismo de piloto automático retirado de um bombardeiro B-24 para simular uma rede de 40 neurônios. Mais tarde, em Princeton, Minsky estudou computação universal em redes neurais. A banca examinadora de seu doutorado mostrou-se cética sobre esse tipo de trabalho, sem saber se deveria ser classificado como um trabalho de matemática. Porém, segundo contam, von Neumann teria dito: “Se não é agora, será algum dia.” Mais tarde, Minsky acabou provando teoremas importantes que mostravam as limitações da pesquisa em redes neurais.

Surgiram vários exemplos de trabalhos que hoje podem ser caracterizados como IA, mas a visão de Alan Turing foi talvez a mais influente. Já em 1947, ele proferia palestras sobre o tema na Sociedade Matemática de Londres e articulou um programa de trabalhos persuasivo em seu artigo de 1950, “Computing Machinery and Intelligence”. Nesse artigo, ele apresentou o teste de Turing, aprendizagem de máquina, algoritmos genéticos e aprendizagem por reforço. Propôs a ideia do *Child Programme*, explicando: “Em vez de tentar produzir um programa para estimular a mente adulta, não seria melhor produzir um que estimulasse a mente infantil?”

### 1.3.2 O nascimento da inteligência artificial (1956)

Princeton foi o lar de outra figura influente na IA, John McCarthy. Após receber seu PhD lá, em 1951, e trabalhar por dois anos como instrutor, McCarthy mudou-se para Stanford e depois para Dartmouth College, que iria se tornar o local oficial de nascimento desse campo. McCarthy convenceu Minsky, Claude Shannon e Nathaniel Rochester a ajudá-lo a reunir pesquisadores dos Estados Unidos interessados em teoria de autômatos, redes neurais e estudo da inteligência. Eles organizaram um seminário de dois meses em Dartmouth, no verão de 1956. A proposta dizia:<sup>11</sup>

Propusemos que um estudo de dois meses e dez homens sobre inteligência artificial fosse realizado durante o verão de 1956 no Dartmouth College, em Hanover, New Hampshire. O estudo era para prosseguir com a conjectura básica de que cada aspecto da aprendizagem ou qualquer outra característica da inteligência pode, em princípio, ser descrita tão precisamente a ponto de ser construída uma máquina para simulá-la. Será realizada uma tentativa para descobrir como fazer com que as máquinas usem a linguagem, a partir de abstrações e conceitos, resolvam os tipos de problemas hoje reservados aos seres humanos e se aperfeiçoem. Achamos que poderá haver avanço significativo em um ou mais desses problemas se um grupo cuidadosamente selecionado de cientistas trabalhar em conjunto durante o verão.

Havia 10 participantes ao todo, incluindo Trenchard More, de Princeton, Arthur Samuel, da IBM, e Ray Solomonoff e Oliver Selfridge, do MIT.

Dois pesquisadores da Carnegie Tech,<sup>12</sup> Allen Newell e Herbert Simon, simplesmente roubaram o show. Embora os outros tivessem ideias e, em alguns casos, programas para aplicações específicas como jogos de damas, Newell e Simon já tinham um programa de raciocínio, o Logic Theorist (LT), sobre o qual Simon afirmou: “Criamos um programa de computador capaz de pensar não numericamente e, assim, resolvemos o antigo dilema mente-corpo.”<sup>13</sup> Logo após o seminário, o programa foi capaz de demonstrar a maioria dos teoremas do Capítulo 2 do livro *Principia Mathematica* de Russell e Whitehead. Contam que Russell ficou encantado quando Simon mostrou a ele que o programa havia criado uma prova de um teorema que era mais curta que a do livro. Os editores do *Journal of Symbolic Logic* ficaram menos impressionados; eles rejeitaram um artigo escrito em parceria por Newell, Simon e pelo Logic Theorist.

O seminário de Dartmouth não trouxe nenhuma novidade, mas apresentou uns aos outros todos os personagens importantes da história. Nos 20 anos seguintes, o campo seria dominado por essas pessoas e por seus alunos e colegas do MIT, da CMU, de Stanford e da IBM.

Examinando a proposta do seminário de Dartmouth (McCarthy *et al.*, 1955), podemos ver por que era necessário que a IA se tornasse um campo separado. Por que todo o trabalho feito na IA não podia ficar sob o nome de teoria de controle, pesquisa operacional ou teoria da decisão que, afinal de contas, têm objetivos semelhantes aos da IA? Ou, então, por que a IA não poderia ser um ramo da matemática? Primeiro, porque a IA abraçou desde o início a ideia de reproduzir faculdades humanas como criatividade, autoaperfeiçoamento e uso da linguagem, e nenhum dos outros campos tratava dessas questões. A segunda resposta é a metodologia. A IA é o único desses campos que claramente é um ramo da ciência da computação (embora a pesquisa operacional compartilhe uma ênfase em simulações por computador), e a IA é o único campo a tentar construir máquinas que funcionarão de forma autônoma em ambientes complexos e mutáveis.

### 1.3.3 Entusiasmo inicial, grandes expectativas (1952-1969)

Os primeiros anos da IA foram repletos de sucessos, mas de uma forma limitada. Considerando-se os primitivos computadores, as ferramentas de programação da época e o fato de que apenas alguns anos antes os computadores eram vistos como objetos capazes de efetuar operações aritméticas e

nada mais, causava surpresa o fato de um computador realizar qualquer atividade remotamente inteligente. Em geral, a classe intelectual preferia acreditar que “uma máquina nunca poderá realizar  $X$ ” (veja, no Capítulo 26, uma longa lista de  $X$  reunidos por Turing). Os pesquisadores da IA respondiam naturalmente demonstrando um  $X$  após outro. John McCarthy se referiu a esse período como a era do “Olhe, mamãe, sem as mãos!”.

O sucesso inicial de Newell e Simon prosseguiu com o General Problem Solver (solucionador de problemas gerais) ou GPS. Diferentemente do Logic Theorist, esse programa foi projetado desde o início para imitar protocolos humanos de resolução de problemas. Dentro da classe limitada de quebra-cabeças com a qual podia lidar, verificou-se que a ordem em que o programa considerava submetas e ações possíveis era semelhante à ordem em que os seres humanos abordavam os mesmos problemas. Desse modo, o GPS talvez tenha sido o primeiro programa a incorporar a abordagem de “pensar de forma humana”. O sucesso do GPS e de programas subsequentes como modelos de cognição levou Newell e Simon (1976) a formularem a famosa hipótese do **sistema de símbolos físicos**, que afirma que “um sistema de símbolos físicos tem os meios necessários e suficientes para uma ação inteligente geral”. O que eles queriam dizer é que qualquer sistema (ser humano ou máquina) que exiba inteligência deve operar manipulando estruturas de dados compostas por símbolos. Veremos, mais adiante, que essa hipótese enfrentou desafios provenientes de muitas direções.

Na IBM, Nathaniel Rochester e seus colegas produziram alguns dos primeiros programas de IA. Herbert Gelernter (1959) construiu o Geometry Theorem Prover, que podia demonstrar teoremas que seriam considerados bastante complicados por muitos alunos de matemática. A partir de 1952, Arthur Samuel escreveu uma série de programas para jogos de damas que eventualmente aprendiam a jogar em um nível amador elevado. Ao mesmo tempo, ele contestou a ideia de que os computadores só podem realizar as atividades para as quais foram programados: seu programa aprendeu rapidamente a jogar melhor que seu criador. O programa foi demonstrado na televisão em fevereiro de 1956, causando impressão muito forte. Como Turing, Samuel teve dificuldades para conseguir um horário em que pudesse utilizar os computadores. Trabalhando à noite, ele usou máquinas que ainda estavam na bancada de testes na fábrica da IBM. O Capítulo 5 aborda os jogos de computador, e o Capítulo 21 explica as técnicas de aprendizado usadas por Samuel.

John McCarthy saiu de Dartmouth para o MIT e lá contribuiu com três realizações cruciais em um ano histórico: 1958. No MIT AI Lab Memo Nº. 1, McCarthy definiu a linguagem de alto nível **Lisp**, que acabou por se tornar a linguagem de programação dominante na IA pelos próximos 30 anos. Com o Lisp, McCarthy teve a ferramenta de que precisava, mas o acesso a recursos de computação escassos e dispendiosos também era um sério problema. Em resposta, ele e outros pesquisadores do MIT criaram o compartilhamento de tempo (*time sharing*). Também em 1958, McCarthy publicou um artigo intitulado *Programs with common sense*, em que descrevia o Advice Taker, um programa hipotético que pode ser visto como o primeiro sistema de IA completo. Como o Logic Theorist e o Geometry Theorem Prover, o programa de McCarthy foi projetado para usar o conhecimento com a finalidade de buscar soluções para problemas.

Entretanto, diferentemente dos outros, ele procurava incorporar o conhecimento geral do mundo. Por exemplo, McCarthy mostrou que alguns axiomas simples permitiriam ao programa gerar um plano para dirigir até o aeroporto e embarcar em um avião. O programa também foi criado de forma

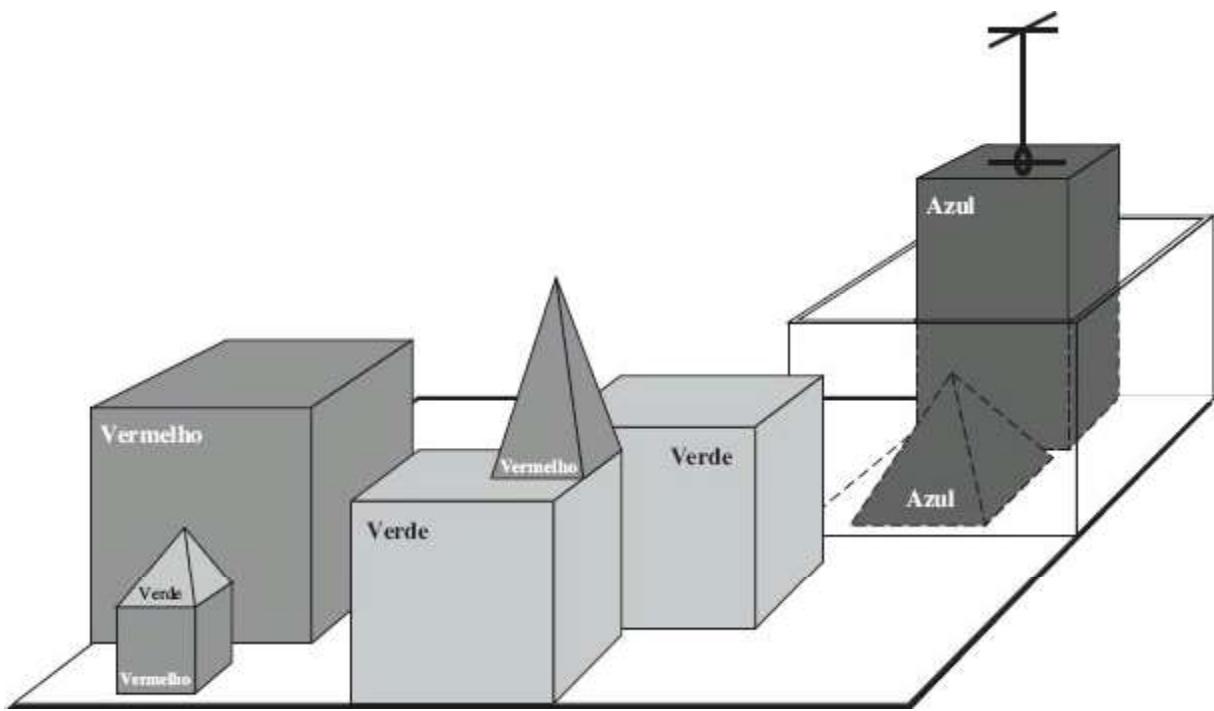
a poder aceitar novos axiomas no curso normal de operação, permitindo assim que adquirisse competência em novas áreas *sem ser reprogramado*. Portanto, o Advice Taker incorporava os princípios centrais de representação de conhecimento e de raciocínio: de que é útil ter uma representação formal e explícita do mundo do modo como as ações de um agente afetam o mundo e o seu funcionamento, e ser capaz de manipular essa representação com processos dedutivos. É notável como grande parte do artigo de 1958 permanece relevante até hoje.

O ano de 1958 também marcou a época em que Marvin Minsky foi para o MIT. Porém, sua colaboração inicial com McCarthy não durou muito. McCarthy enfatizava a representação e o raciocínio em lógica formal, enquanto Minsky estava mais interessado em fazer os programas funcionarem e, eventualmente, desenvolveu uma perspectiva contrária ao estudo da lógica. Em 1963, McCarthy fundou o laboratório de IA em Stanford. Seu plano de usar a lógica para construir o Advice Taker definitivo foi antecipado pela descoberta feita por J. A. Robinson do método de resolução (um algoritmo completo para demonstração de teoremas para a lógica de primeira ordem; consulte o Capítulo 9). O trabalho em Stanford enfatizou métodos de uso geral para raciocínio lógico. As aplicações da lógica incluíam os sistemas para responder a perguntas e os sistemas de planejamento de Cordell Green (Green, 1969b) e o projeto de robótica Shakey no novo Stanford Research Institute (SRI). Este último projeto, descrito com mais detalhes no Capítulo 25, foi o primeiro a demonstrar a integração completa do raciocínio lógico e da atividade física.

Minsky orientou vários alunos que escolheram problemas limitados cuja solução parecia exigir inteligência. Esses domínios limitados se tornaram conhecidos como **micromundos**. O programa SAINT de James Slagle (1963) era capaz de resolver problemas de cálculo integral típicos do primeiro ano dos cursos acadêmicos. O programa ANALOGY de Tom Evans (1968) resolvia problemas de analogia geométrica que apareciam em testes de QI. O programa STUDENT de Daniel Bobrow (1967) resolvia problemas clássicos de álgebra, como este:

Se o número de clientes que Tom consegue é igual ao dobro do quadrado de 20% do número de anúncios que ele publica e se o número de anúncios publicados é 45, qual é o número de clientes que Tom consegue?

O mais famoso micromundo foi o mundo de blocos, que consiste em um conjunto de blocos sólidos colocados sobre uma mesa (ou, com maior frequência, sobre a simulação de uma mesa), como mostra a Figura 1.4. Uma tarefa típica nesse mundo é reorganizar os blocos de certa maneira, utilizando a mão de um robô que pode erguer um bloco de cada vez. O mundo de blocos foi a base do projeto de visão de David Huffman (1971), do trabalho em visão e propagação de restrições de David Waltz (1975), da teoria do aprendizado de Patrick Winston (1970), do programa de compreensão de linguagem natural de Terry Winograd (1972) e do sistema de planejamento de Scott Fahlman (1974).



**Figura 1.4** Uma cena do mundo de blocos. O programa SHRDLU (Winograd, 1972) tinha acabado de completar o comando: “Encontre um bloco mais alto que o bloco que você está segurando e coloque-o na caixa.”

O trabalho pioneiro baseado nas redes neurais de McCulloch e Pitts também prosperou. O trabalho de Winograd e Cowan (1963) mostrou que grande número de elementos podia representar coletivamente um conceito individual, com aumento correspondente na robustez e no paralelismo. Os métodos de aprendizado de Hebb foram aperfeiçoados por Bernie Widrow (Widrow e Hoff, 1960; Widrow, 1962), que denominou suas redes **adalines**, e por Frank Rosenblatt (1962) com seus **perceptrons**. **O teorema da convergência do perceptron** (Block *et al.*, 1962) determina que o algoritmo de aprendizagem podia ajustar os pesos de conexão de um perceptron para corresponderem a quaisquer dados de entrada, desde que existisse tal correspondência. Esses tópicos são cobertos no Capítulo 20.

### 1.3.4 Uma dose de realidade (1966-1973)

Desde o início, os pesquisadores da IA eram ousados nos prognósticos de seus sucessos futuros. Esta declaração de Herbert Simon em 1957 frequentemente é citada:

Não é meu objetivo surpreendê-los ou chocá-los, mas o modo mais simples de resumir tudo isso é dizer que agora existem no mundo máquinas que pensam, aprendem e criam. Além disso, sua capacidade de realizar essas atividades está crescendo rapidamente até o ponto — em um futuro visível — no qual a variedade de problemas com que elas poderão lidar será correspondente à variedade de problemas com os quais lida a mente humana.

Termos como “futuro visível” podem ser interpretados de várias maneiras, mas Simon também fez previsões mais concretas: a de que dentro de 10 anos um computador seria campeão de xadrez e que um teorema matemático significativo seria provado por uma máquina. Essas previsões se realizaram

(ou quase) no prazo de 40 anos, em vez de 10. O excesso de confiança de Simon se devia ao desempenho promissor dos primeiros sistemas de IA em exemplos simples. Contudo, em quase todos os casos, esses primeiros sistemas acabaram falhando desastrosamente quando foram experimentados em conjuntos de problemas mais extensos ou em problemas mais difíceis.

O primeiro tipo de dificuldade surgiu porque a maioria dos primeiros programas não tinha conhecimento de seu assunto; eles obtinham sucesso por meio de manipulações sintáticas simples. Uma história típica ocorreu durante os primeiros esforços de tradução automática, que foram generosamente subsidiados pelo National Research Council dos Estados Unidos, em uma tentativa de acelerar a tradução de documentos científicos russos após o lançamento do Sputnik em 1957. Inicialmente, imaginava-se que transformações sintáticas simples baseadas nas gramáticas russas e inglesas, e a substituição de palavras com a utilização de um dicionário eletrônico, seriam suficientes para preservar os significados exatos das orações. O fato é que a tradução exata exige conhecimento profundo do assunto para solucionar ambiguidades e estabelecer o conteúdo da sentença. A famosa retradução de “o espírito está disposto mas a carne é fraca”<sup>14</sup> como “a vodca é boa mas a carne é podre” ilustra as dificuldades encontradas. Em 1966, um relatório criado por um comitê consultivo descobriu que “não existe nenhum sistema de tradução automática para texto científico em geral, e não existe nenhuma perspectiva imediata nesse sentido”. Toda a subvenção do governo dos Estados Unidos para projetos acadêmicos de tradução foi cancelada. Hoje, a tradução automática é uma ferramenta imperfeita, mas amplamente utilizada em documentos técnicos, comerciais, governamentais e da Internet.

O segundo tipo de dificuldade foi a impossibilidade de tratar muitos dos problemas que a IA estava tentando resolver. A maior parte dos primeiros programas de IA resolia problemas experimentando diferentes combinações de passos até encontrar a solução. Essa estratégia funcionou inicialmente porque os micromundos continham pouquíssimos objetos e, consequentemente, um número muito pequeno de ações possíveis e sequências de soluções muito curtas. Antes do desenvolvimento da teoria de complexidade computacional, era crença geral que o “aumento da escala” para problemas maiores era apenas uma questão de haver hardware mais rápido e maior capacidade de memória. Por exemplo, o otimismo que acompanhou o desenvolvimento da prova de teoremas por resolução logo foi ofuscado quando os pesquisadores não conseguiram provar teoremas que envolviam mais que algumas dezenas de fatos. *O fato de um programa poder encontrar uma solução em princípio não significa que o programa contenha quaisquer dos mecanismos necessários para encontrá-la na prática.*

 A ilusão do poder computacional ilimitado não ficou confinada aos programas de resolução de problemas. Os primeiros experimentos de **evolução automática** (agora chamados **algoritmos genéticos**) (Friedberg, 1958; Friedberg *et al.*, 1959) se baseavam na convicção sem dúvida correta de que, realizando-se uma série apropriada de pequenas mutações em um programa em código de máquina, seria possível gerar um programa com bom desempenho para qualquer tarefa simples. Então, a ideia era experimentar mutações aleatórias com um processo de seleção para preservar mutações que parecessem úteis. Apesar de milhares de horas de tempo de CPU, quase nenhum progresso foi demonstrado. Os algoritmos genéticos modernos utilizam representações melhores e têm mais sucesso.

A incapacidade de conviver com a “explosão combinatória” foi uma das principais críticas à IA

contidas no relatório Lighthill (Lighthill, 1973), que formou a base para a decisão do governo britânico de encerrar o apoio à pesquisa da IA em todas as universidades, com exceção de duas (a tradição oral pinta um quadro um pouco diferente e mais colorido, com ambições políticas e hostilidades pessoais, cuja descrição não nos interessa aqui).

Uma terceira dificuldade surgiu devido a algumas limitações fundamentais nas estruturas básicas que estavam sendo utilizadas para gerar o comportamento inteligente. Por exemplo, o livro de Minsky e Papert, *Perceptrons* (1969), provou que, embora os perceptrons (uma forma simples de rede neural) pudessem aprender tudo o que eram capazes de representar, eles podiam representar muito pouco. Em particular, um perceptron de duas entradas (restringido para ser mais simples que a forma que Rosemblatt estudou) não podia ser treinado para reconhecer quando suas duas entradas eram diferentes. Embora seus resultados não se aplicassem a redes mais complexas de várias camadas, a subvenção de pesquisas relacionadas a redes neurais logo se reduziu a quase nada. Ironicamente, os novos algoritmos de aprendizado por retropropagação para redes de várias camadas que acabaram de provocar um enorme renascimento na pesquisa de redes neurais no final da década de 1980 foram, na verdade, descobertos primeiro em 1969 (Bryson e Ho, 1969).

### 1.3.5 Sistemas baseados em conhecimento: a chave para o poder? (1969-1979)

O quadro de resolução de problemas que havia surgido durante a primeira década de pesquisas em IA foi o de um mecanismo de busca de uso geral que procurava reunir passos elementares de raciocínio para encontrar soluções completas. Tais abordagens foram chamadas **métodos fracos** porque, embora gerais, não podiam ter aumento de escala para instâncias de problemas grandes ou difíceis. A alternativa para métodos fracos é usar um conhecimento mais amplo e específico de domínio que permita passos de raciocínio maiores e que possam tratar com mais facilidade casos que ocorrem tipicamente em especialidades estritas. Podemos dizer que, para resolver um problema difícil, praticamente é necessário já saber a resposta.

O programa DENDRAL (Buchanan *et al.*, 1969) foi um exemplo inicial dessa abordagem. Ele foi desenvolvido em Stanford, onde Ed Feigenbaum (um antigo aluno de Herbert Simon), Bruce Buchanan (filósofo transformado em cientista de computação) e Joshua Lederberg (geneticista laureado com um Prêmio Nobel) formaram uma equipe para resolver o problema de inferir a estrutura molecular a partir das informações fornecidas por um espectrômetro de massa. A entrada para o programa consiste na fórmula elementar da molécula (por exemplo, C<sub>6</sub>H<sub>13</sub>NO<sub>2</sub>) e o espectro de massa que fornece as massas dos diversos fragmentos da molécula gerada quando ela é bombardeada por um feixe de elétrons. Por exemplo, o espectro de massa poderia conter um pico em  $m = 15$ , correspondendo à massa de um fragmento metil (CH<sub>3</sub>).

A versão ingênua do programa gerou todas as estruturas possíveis consistentes com a fórmula e depois previu qual seria o espectro de massa observado para cada uma, comparando esse espectro com o espectro real. Como se poderia esperar, esse é um problema intratável mesmo para moléculas de tamanho moderado. Os pesquisadores do DENDRAL consultaram especialistas em química analítica e descobriram que eles trabalhavam procurando padrões conhecidos de picos no espectro que sugerissem subestruturas comuns na molécula. Por exemplo, a regra a seguir é usada para

reconhecer um subgrupo cetona (C=O), que pesa 28 unidades de massa:

se existem dois picos em  $x_1$  e  $x_2$  tais que

- (a)  $x_1 + x_2 = M + 28$  ( $M$  é a massa da molécula inteira);
- (b)  $x_1 - 28$  é um pico;
- (c)  $x_2 - 28$  é um pico;
- (d) No mínimo, um entre  $x_1$  e  $x_2$  é alto.

então, existe um subgrupo cetona

O reconhecimento de que a molécula contém uma subestrutura específica reduz enormemente o número de possíveis candidatos. O DENDRAL era eficiente porque:

Todo o conhecimento teórico relevante para resolver esses problemas foi mapeado de sua forma geral no [componente de previsão de espectro] (“princípios básicos”) para formas especiais eficientes (“receitas de bolo”). (Feigenbaum *et al.*, 1971)

O DENDRAL foi importante porque representou o primeiro sistema bem-sucedido de *conhecimento intensivo*: sua habilidade derivava de um grande número de regras de propósito específico. Sistemas posteriores também incorporaram o tema principal da abordagem de McCarthy no Advice Taker — a separação clara entre o conhecimento (na forma de regras) e o componente de raciocínio.

Com essa lição em mente, Feigenbaum e outros pesquisadores de Stanford iniciaram o Heuristic Programming Project (HPP) para investigar até que ponto a nova metodologia de **sistemas especialistas** poderia ser aplicada a outras áreas do conhecimento humano. Em seguida, o principal esforço foi dedicado à área de diagnóstico médico. Feigenbaum, Buchanan e o Dr. Edward Shortliffe desenvolveram o MYCIN para diagnosticar infecções sanguíneas. Com cerca de 450 regras, o MYCIN foi capaz de se sair tão bem quanto alguns especialistas e muito melhor do que médicos em início de carreira. Ele também apresentava duas diferenças importantes em relação ao DENDRAL. Primeiro, diferentemente das regras do DENDRAL, não havia nenhum modelo teórico geral a partir do qual as regras do MYCIN pudessem ser deduzidas. Elas tinham de ser adquiridas a partir de entrevistas extensivas com especialistas que, por sua vez, as adquiriam de livros didáticos, de outros especialistas e da experiência direta de estudos de casos. Em segundo lugar, as regras tinham de refletir a incerteza associada ao conhecimento médico. O MYCIN incorporava um cálculo de incerteza chamado **fatores de certeza** (consulte o Capítulo 14) que pareciam (na época) se adequar bem à forma como os médicos avaliavam o impacto das evidências no diagnóstico.

A importância do conhecimento de domínio também ficou aparente na área da compreensão da linguagem natural. Embora o sistema SHRDLU de Winograd para reconhecimento da linguagem natural tivesse despertado bastante interesse, sua dependência da análise sintática provocou alguns problemas idênticos aos que ocorreram nos primeiros trabalhos em tradução automática. Ele foi capaz de superar a ambiguidade e reconhecer referências pronominais, mas isso acontecia principalmente porque o programa foi criado especificamente para uma única área — o mundo dos blocos. Diversos pesquisadores, entre eles Eugene Charniak, aluno graduado e companheiro de

Winograd no MIT, sugeriram que uma compreensão robusta da linguagem exigiria conhecimentos gerais sobre o mundo e um método genérico para utilizar esses conhecimentos.

Em Yale, o linguista transformado em pesquisador da IA Roger Schank enfatizou esse ponto, afirmando: “Não existe essa coisa de sintaxe.” Isso irritou muitos linguistas, mas serviu para dar início a uma discussão útil. Schank e seus alunos elaboraram uma série de programas (Schank e Abelson, 1977; Wilensky, 1978; Schank e Riesbeck, 1981; Dyer, 1983), todos com a tarefa de entender a linguagem natural. Porém, a ênfase foi menos na linguagem em si e mais nos problemas de representação e raciocínio com o conhecimento exigido para compreensão da linguagem. Os problemas incluíam a representação de situações estereotípicas (Cullingford, 1981), descrição da organização da memória humana (Rieger, 1976; Kolodner, 1983) e compreensão de planos e metas (Wilensky, 1983).

O enorme crescimento das aplicações para resolução de problemas reais causou um aumento simultâneo na demanda por esquemas utilizáveis de representação do conhecimento. Foi desenvolvido grande número de diferentes linguagens de representação e raciocínio. Algumas se baseavam na lógica — por exemplo, a linguagem Prolog se tornou popular na Europa, e a família PLANNER, nos Estados Unidos. Outras, seguindo a ideia de **frames** de Minsky (1975), adotaram uma abordagem mais estruturada, reunindo fatos sobre tipos específicos de objetos e eventos, e organizando os tipos em uma grande hierarquia taxonômica análoga a uma taxonomia biológica.

### 1.3.6 A IA se torna uma indústria (de 1980 até a atualidade)

O primeiro sistema especialista comercial bem-sucedido, o R1, iniciou sua operação na Digital Equipment Corporation (McDermott, 1982). O programa ajudou a configurar pedidos de novos sistemas de computadores; em 1986, ele estava fazendo a empresa economizar cerca de 40 milhões de dólares por ano. Em 1988, o grupo de IA da DEC tinha 40 sistemas especialistas entregues, com outros sendo produzidos. A Du Pont tinha 100 desses sistemas em uso e 500 em desenvolvimento, economizando aproximadamente 10 milhões de dólares por ano. Quase todas as corporações importantes dos Estados Unidos tinham seu próprio grupo de IA e estavam usando ou investigando sistemas especialistas.

Em 1981, os japoneses anunciaram o projeto “Fifth Generation”, um plano de 10 anos para montar computadores inteligentes que utilizassem Prolog. Em resposta, os Estados Unidos formaram a Microelectronics and Computer Technology Corporation (MCC) como um consórcio de pesquisa projetado para assegurar a competitividade nacional. Em ambos os casos, a IA fazia parte de um amplo esforço, incluindo o projeto de chips e a pesquisa da interface humana. Na Inglaterra, o relatório Alvey reabilitou o subsídio que havia sido cortado em consequência do relatório Lighthill.<sup>15</sup> No entanto, em todos os três países, os projetos nunca alcançaram seus objetivos ambiciosos.

De modo geral, a indústria da IA se expandiu de alguns milhões de dólares em 1980 para bilhões de dólares em 1988, incluindo centenas de empresas construindo sistemas especialistas, sistemas de visão, robôs, e software e hardware especializados para esses propósitos. Logo depois, veio um período chamado de “inverno da IA”, em que muitas empresas caíram no esquecimento à medida que

deixaram de cumprir promessas extravagantes.

### 1.3.7 O retorno das redes neurais (de 1986 até a atualidade)

Em meados dos anos 1980, pelo menos quatro grupos diferentes reinventaram o algoritmo de aprendizado por **retroprogramação**, descoberto pela primeira vez em 1969 por Bryson e Ho. O algoritmo foi aplicado a muitos problemas de aprendizado em ciência da computação e psicologia, e a ampla disseminação dos resultados na coletânea *Parallel Distributed Processing* (Rumelhart e McClelland, 1986) causou grande excitação.

Os chamados modelos **conexionistas** para sistemas inteligentes eram vistos por alguns como concorrentes diretos dos modelos simbólicos promovidos por Newell e Simon e da abordagem logicista de McCarthy e outros pesquisadores (Smolensky, 1988). Pode parecer óbvio que, em certo nível, os seres humanos manipulam símbolos — de fato, o livro de Terrence Deacon, *The Symbolic Species* (1997), sugere que essa é a *característica que define* os seres humanos —, mas os conexionistas mais fervorosos questionavam se a manipulação de símbolos tinha qualquer função explicativa real em modelos detalhados de cognição. Essa pergunta permanece sem resposta, mas a visão atual é de que as abordagens conexionista e simbólica são complementares, e não concorrentes. Como ocorreu com a separação da IA e da ciência cognitiva, a pesquisa moderna de rede neural se bifurcou em dois campos, um preocupado com a criação de algoritmos e arquiteturas de rede eficazes e a compreensão de suas propriedades matemáticas, o outro preocupado com a modelagem cuidadosa das propriedades empíricas de neurônios reais e conjuntos de neurônios.

### 1.3.8 A IA se torna uma ciência (de 1987 até a atualidade)

Nos últimos anos, houve uma revolução no trabalho em inteligência artificial, tanto no conteúdo quanto na metodologia.<sup>16</sup> Agora, é mais comum usar as teorias existentes como bases, em vez de propor teorias inteiramente novas, fundamentar as afirmações em teoremas rigorosos ou na evidência experimental rígida, em vez de utilizar como base a intuição e destacar a relevância de aplicações reais em vez de exemplos de brinquedos.

Em parte, a IA surgiu como uma rebelião contra as limitações de áreas existentes como a teoria de controle e a estatística, mas agora ela inclui esses campos. Conforme afirmou David McAllester (1998):

No período inicial da IA, parecia plausível que novas formas de computação simbólica, como frames e redes semânticas, tornariam obsoleta grande parte da teoria clássica. Isso levou a uma forma de isolacionismo na qual a IA ficou bem separada do restante da ciência da computação. Atualmente, esse isolacionismo está sendo abandonado. Existe o reconhecimento de que o aprendizado da máquina não deve ser isolado da teoria da informação, de que o raciocínio incerto não deve ser isolado da modelagem estocástica, de que a busca não deve ser isolada da otimização clássica e do controle, e de que o raciocínio automatizado não deve ser isolado dos

métodos formais e da análise estática.

Em termos de metodologia, a IA finalmente adotou com firmeza o método científico. Para serem aceitas, as hipóteses devem ser submetidas a rigorosos experimentos empíricos, e os resultados devem ser analisados estatisticamente de acordo com sua importância (Cohen, 1995). Agora é possível replicar experimentos a partir da utilização de repositórios compartilhados de código e dados de teste.

O campo de reconhecimento da fala ilustra o padrão. Nos anos 1970, foi experimentada ampla variedade de arquiteturas e abordagens diferentes. Muitas delas eram bastante *ad hoc* e frágeis, e foram demonstradas em apenas alguns exemplos especialmente selecionados. Nos últimos anos, abordagens baseadas em **modelos ocultos de Markov** (MOMs) passaram a dominar a área. Dois aspectos dos MOMs são relevantes. Primeiro, eles se baseiam em uma teoria matemática rigorosa. Isso permitiu que os cientistas de reconhecimento de fala se baseassem em várias décadas de resultados matemáticos desenvolvidos em outros campos. Em segundo lugar, eles são gerados por um processo de treinamento em um grande conjunto de dados reais de fala. Isso assegura um desempenho robusto e, em testes cegos rigorosos, os MOMs têm melhorado suas pontuações de forma contínua. A tecnologia da fala e o campo inter-relacionado de reconhecimento de caracteres manuscritos já estão efetuando a transição para aplicações industriais e de consumo em larga escala.

Observe que não há nenhuma afirmação científica de que os humanos utilizam MOMs para reconhecer a fala, mas que os MOMs fornecem uma estrutura matemática para a compreensão do problema e apoiam a alegação da engenharia de que na prática eles funcionam bem.

A tradução automática segue o mesmo curso que o reconhecimento de voz. Na década de 1950 houve um entusiasmo inicial por uma abordagem baseada na sequência de palavras, aprendida com modelos de acordo com os princípios da teoria da informação. A abordagem caiu em desuso na década de 1960, mas retornou no final dos anos 1990 e agora domina o campo.

As redes neurais também seguem essa tendência. Grande parte do trabalho em redes neurais nos anos 1980 foi realizada na tentativa de definir a abrangência do que poderia ser feito e de aprender como as redes neurais diferem das técnicas “tradicionalis”. Utilizando metodologia aperfeiçoada e estruturas teóricas, o campo chegou a uma compreensão tal que, agora, as redes neurais podem ser comparadas a técnicas correspondentes da estatística, do reconhecimento de padrões e do aprendizado de máquina, podendo ser utilizada a técnica mais promissora em cada aplicação. Como resultado desse desenvolvimento, a tecnologia denominada **mineração de dados** gerou uma nova e vigorosa indústria.

A obra de Judea Pearl, *Probabilistic Reasoning in Intelligent Systems* (1988), levou a uma nova aceitação da probabilidade e da teoria da decisão na IA, seguindo um renascimento do interesse descrito no artigo de Peter Cheeseman, “In Defense of Probability” (1985). O formalismo denominado **rede bayesiana** foi criado para permitir a representação eficiente do conhecimento incerto e o raciocínio rigoroso com a utilização desse tipo de conhecimento. Essa abordagem supera amplamente muitos problemas dos sistemas de raciocínio probabilístico das décadas de 1960 e 1970; agora ele domina a pesquisa de IA sobre raciocínio incerto e sistemas especialistas.

A abordagem admite o aprendizado a partir da experiência e combina o melhor da IA clássica e das redes neurais. O trabalho de Judea Pearl (1982a) e de Eric Horvitz e David Heckerman (Horvitz

e Heckerman, 1986; Horvitz *et al.*, 1986) promoveu a ideia de sistemas especialistas *normativos*: sistemas que agem racionalmente de acordo com as leis da teoria de decisão e não procuram imitar os passos do pensamento de especialistas humanos. O sistema operacional Windows™ inclui vários sistemas especialistas de diagnóstico normativo para correção de problemas. Os Capítulos 13 a 16 examinam essa área.

Revolução suaves semelhantes a essa ocorreram nos campos de robótica, visão computacional e representação de conhecimento. Uma compreensão melhor dos problemas e de suas propriedades de complexidade, combinada à maior sofisticação matemática, resultou em agendas de pesquisa utilizáveis e métodos robustos. Apesar do aumento da formalização e da especialização terem levado campos como visão e robótica a tornarem-se de alguma forma isolados do “principal” em IA nos anos 1990, essa tendência foi revertida nos últimos anos à medida que ferramentas de aprendizado de máquina em particular, mostraram-se eficazes para muitos problemas. O processo de reintegração já está rendendo benefícios significativos.

### 1.3.9 O surgimento de agentes inteligentes (de 1995 até a atualidade)

Talvez encorajados pelo progresso na resolução dos subproblemas da IA, os pesquisadores também começaram a examinar mais uma vez o problema do “agente como um todo”. O trabalho de Allen Newell, John Laird e Paul Rosenbloom no SOAR (Newell, 1990; Laird *et al.*, 1987) é o exemplo mais conhecido de uma arquitetura completa de agente. Um dos ambientes mais importantes para agentes inteligentes é a Internet. Os sistemas de IA se tornaram tão comuns em aplicações da Web que o sufixo “bot” passou a fazer parte da linguagem cotidiana. Além disso, as tecnologias da IA servem de base a muitas ferramentas da Internet, como mecanismos de pesquisa, sistemas de recomendação (*recommender systems*) e agregadores de conteúdo de construção de sites.

Uma consequência de tentar construir agentes completos é a constatação de que os subcampos previamente isolados da IA podem necessitar ser reorganizados quando se tiver que unir os resultados. Em particular, hoje é amplamente reconhecido que os sistemas sensoriais (visão, sonar, reconhecimento de voz etc.) não podem fornecer informações perfeitamente confiáveis sobre o meio ambiente. Assim, os sistemas de raciocínio e de planejamento devem ser capazes de lidar com a incerteza. Uma segunda consequência importante pela perspectiva do agente é que a IA foi estabelecida em contato muito mais próximo com outros campos, como teoria de controle e economia, que também lidam com agentes. O progresso recente do controle de carros robóticos foi derivado de uma mistura de abordagens que vai desde melhores sensores, controle teórico da integração do sensoriamento, localização e mapeamento, bem como um grau de alto nível de planejamento.

Apesar desses sucessos, alguns fundadores influentes da IA, incluindo John McCarthy (2007), Marvin Minsky (2007), Nils Nilsson (1995, 2005) e Patrick Winston (Beal e Winston, 2009), expressaram descontentamento com a evolução da IA. Achavam que a IA deveria colocar menos ênfase na criação de versões cada vez melhores de aplicações eficientes para tarefas específicas, tal como dirigir um carro, jogar xadrez ou reconhecer fala. Em vez disso, acreditam que a IA deveria retornar às suas raízes esforçando-se para obter, nas palavras de Simon, “máquinas que pensam, que

aprendem e que criam". Chamam o esforço de **IA de nível humano** ou HLAI; o primeiro simpósio foi em 2004 (Minsky *et al.*, 2004). O esforço necessitará de grandes bases de conhecimento; Hendler *et al.* (1995) discutem de onde essas bases de conhecimento poderiam vir.

Uma ideia relacionada é o subcampo da **inteligência geral artificial** ou IAG (Goertzel e Pennachin, 2007), que realizou a sua primeira conferência e organizou o *Journal of Artificial General Intelligence* em 2008. A IAG procura por um algoritmo universal para aprender e atuar em qualquer ambiente, e tem suas raízes na obra de Ray Solomonoff (1964), um dos participantes da conferência original de Dartmouth em 1956. Garantindo que o que nós criamos é realmente **IA amigável** também é uma preocupação (Yudkowsky, 2008; Omohundro, 2008), para a qual voltaremos no Capítulo 26.

### 1.3.10 Disponibilidade de conjuntos de dados muito grandes (2001 até a atualidade)

Ao longo de 60 anos de história da ciência da computação, a ênfase tem sido no *algoritmo* como o assunto principal de estudo. Mas alguns trabalhos recentes da IA sugerem que, para muitos problemas, faz mais sentido se preocupar com os *dados* e ser menos exigente sobre qual algoritmo aplicar. Isso é verdade devido à disponibilidade crescente de fontes de dados muito grandes: por exemplo, trilhões de palavras de inglês e bilhões de imagens da Web (Kilgarriff e Grefenstette, 2006) ou bilhões de pares de bases de sequências genômicas (Collins *et al.*, 2003).

Um artigo influente nessa linha de pesquisa foi o trabalho de Yarowsky (1995) sobre desambiguação de sentido de palavras: dado o uso da palavra “planta” em uma frase, ela se refere a flora ou fábrica? Abordagens anteriores do problema confiavam em rótulos humanos combinados com algoritmos de aprendizado de máquina. Yarowsky mostrou que a tarefa poderia ser feita, com precisão superior a 96%, sem quaisquer exemplos rotulados. Em vez disso, dado um *corpus* muito grande de texto não anotado e apenas as definições de dicionário dos dois sentidos, “obras, planta industrial” e “flora, vida das plantas”, pode-se rotular exemplos no *corpus*, e de lá, **por iniciativa própria**, aprender novos modelos que ajudem a rotular novos exemplos. Banko e Brill (2001) mostram que técnicas como essa têm um desempenho ainda melhor à medida que a quantidade de texto disponível vai de um milhão de palavras para um bilhão e que o aumento no desempenho pela utilização de mais dados excede qualquer diferença na escolha do algoritmo; um algoritmo medíocre com 100 milhões de palavras de dados de treinamento não rotulados supera o melhor algoritmo conhecido com um milhão de palavras.

Em outro exemplo, Hays e Efros (2007) discutem o problema do preenchimento de buracos em uma fotografia. Suponha que você use o Photoshop para mascarar um ex-amigo de uma foto de grupo, mas agora você precisa preencher a área mascarada com algo que corresponda ao fundo. Hays e Efros definiram um algoritmo que busca, através de uma coleção de fotos, encontrar algo que vá corresponder. Descobriram que o desempenho de seu algoritmo era pobre quando usavam uma coleção de apenas 10 mil fotos, mas atravessou o limiar para um excelente desempenho quando aumentaram a coleção para dois milhões de fotos.

Trabalho como esse sugere que o “gargalo do conhecimento” na IA — o problema de como