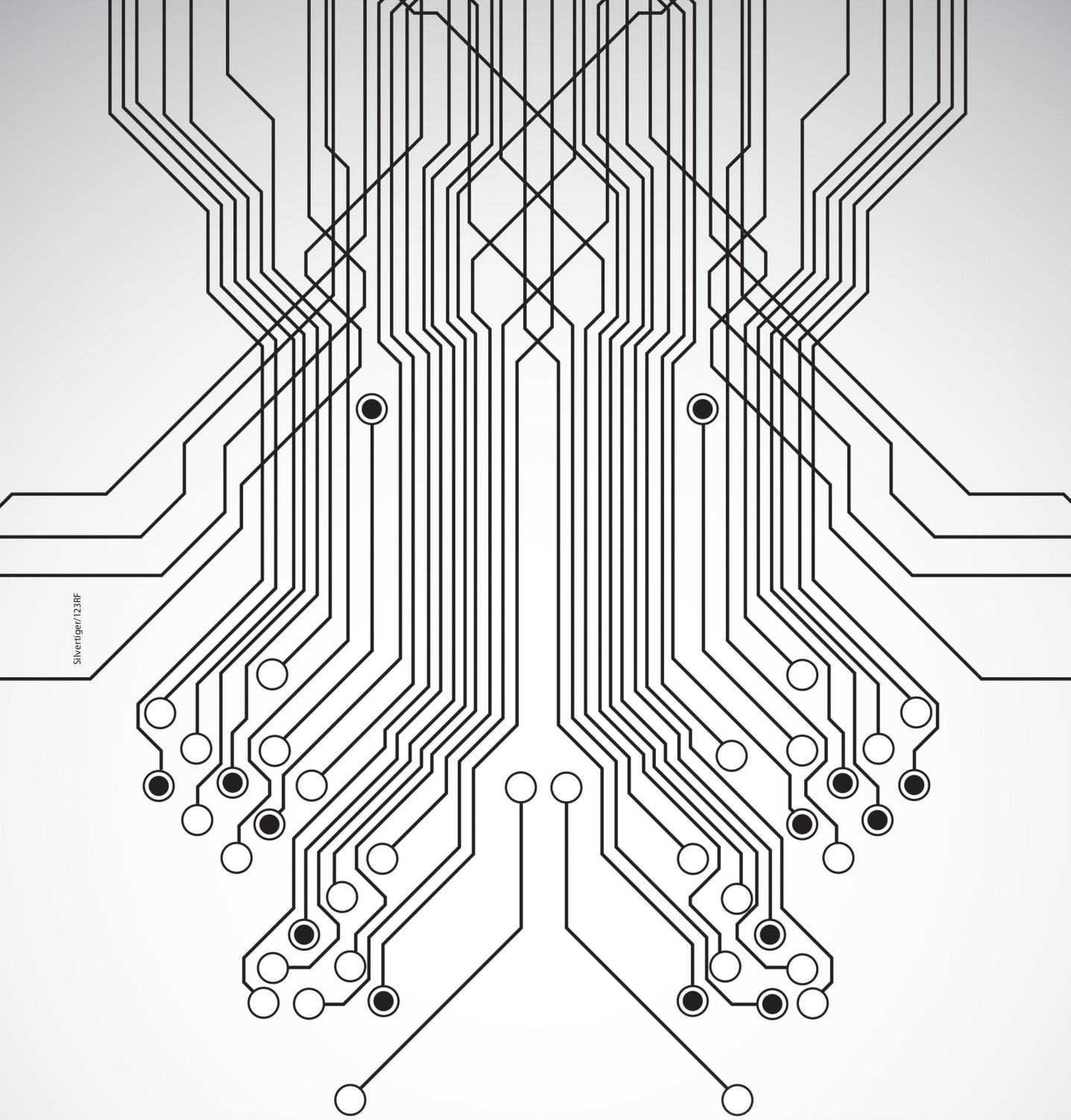


Silvertiger/123RF

Representações internas e processamento de informação em redes neurais

Nestor Caticha

Chairat Saengdamuk/123RF

resumo

O objetivo deste artigo é mostrar em casos simples como funcionam as redes neurais. Nesse sentido, embora seja possível descrever o funcionamento de uma rede neural de arquitetura profunda de várias formas, neste artigo optou-se pela descrição em termos da construção e reconstrução de representações internas à medida que a informação se propaga pela rede.

Palavras-chave: redes neurais; inteligência artificial.

abstract

The aim of this article is to show in simple cases how neural networks function. Although it is possible to describe the functioning of a neural network of deep architecture in various ways, this article opted for the description in terms of construction and reconstruction of internal representations as the information propagates through the network.

Keywords: *neural networks*; *artificial intelligence*.

Ao longo da história da humanidade os modelos mecânicos para o cérebro e a mente mudaram ao acompanhar o desenvolvimento tecnológico da época. Se já descrevemos o cérebro usando relógios de água, moinhos de trigo, telégrafos e computadores digitais, agora temos outras metáforas. As redes neurais artificiais (RNA) ocupam hoje esse lugar privilegiado. Além de permitir modelos de cognição ou de sistemas neuronais biológicos, as RNA estão por trás da revolução tecnológica em curso. O lugar-comum é citar Arthur C. Clarke e dizer que "*Any sufficiently advanced technology is indistinguishable from magic*". Porém, o objetivo deste artigo é mostrar em casos simples como funcionam as redes neurais, essencialmente revelando os ases dentro da manga, mas devemos conceder que as redes mais complexas ainda parecem magia até para seus criadores, pois a composição de muitos elementos simples pode gerar resultados inesperados.

As primeiras ideias sobre redes neurais apareceram logo depois do descobrimento dos neurônios por Santiago Ramon y Cajal, no fim do século XIX. William James e Sigmund Freud estão entre os primeiros a sugerir a modelagem de processos cognitivos. A falta de ferramentas teóricas e experimentais daquela época levou ao abandono de tais projetos, que voltaram a surgir com mais força depois de avanços em neurofisiologia e teoria de computação em meados do século XX. Hodgkin e Huxley, do lado biológico, introduziram o modelo matemático do neurônio que até hoje é a base para modelagem de sistemas neurais realistas. McCulloch e Pitts introduziram o modelo matemático mais simples possível de um neurônio. O perceptron introduzido por Rosenblatt, no fim da década de 50,

NESTOR CATICHA é professor titular do Instituto de Física da Universidade de São Paulo e coordenador do Núcleo de Apoio à Pesquisa CNAIPS da USP.

usa os neurônios de McCulloch-Pitts como peças de lego para construir máquinas muito mais ricas e complexas. O perceptron mais simples tem somente uma peça de lego, unidades de entrada se comunicam com uma única unidade de saída. A importância dessa máquina é que não precisamos saber as regras que levam à resolução de um problema, tal como a classificação de objetos em uma de duas categorias. Basta dispor de exemplos, pares formados pelo objeto e sua classificação. O perceptron *aprende* usando exemplos. Rosenblatt foi em frente e mostrou que, se existe um perceptron simples que resolve um problema de classificação, podemos usá-lo como professor de outro perceptron simples, o aluno. Rosenblatt apresentou um algoritmo que leva o aluno a aprender o conjunto de exemplos em tempo finito e ainda pode classificar exemplos nunca antes vistos, com habilidade que cresce com o número de exemplos apresentados. O aluno pode generalizar a partir do que foi ensinado pelo professor. O impacto intelectual foi grande, havia um caminho para construir máquinas que poderiam resolver um problema sem que fosse necessário que o programador soubesse como resolvê-lo. Promessas foram feitas. A inteligência artificial estava ao nosso alcance e o financiamento de pesquisas a tornaria realidade. Porém Minsky e Papert mostraram que há problemas que não poderiam ser resolvidos pelo perceptron simples, mas sim por perceptrons feitos com mais peças de lego. Essas máquinas deveriam ter, além das unidades de entrada e de saída, também unidades internas formando camadas escondidas dentro do que seria a "caixa-preta". Embora isso esteja correto, eles lançaram a conjectura de que não haveria alternativa prática para treinar uma rede de perceptrons multicamadas. Isso mostrou para alguns que o projeto de redes neurais estava destinado ao fracasso. E o financiamento foi para outras áreas de IA baseadas em regras específicas e não em aprendizagem.

A conjectura estava baseada "em parte na experiência em encontrar falácias nos métodos propostos" e, como quase toda prova de impossibilidade de algo baseado no argumento "eu não consegui", mostrou-se incorreta. Vários autores encontraram soluções para treinar redes com camadas internas, mas de alguma forma a notícia não se difundiu, até que, em 1986, Rumelhart, Hinton e Williams popularizaram o método de retropropagação (*backpropagation*) para treinar perceptrons de multicamadas. As redes com algumas poucas camadas internas viraram o jogo da IA. O programa de estudo e uso de redes neurais, com o novo nome de *coneccionismo*, ganhou espaço. Começou um novo ciclo de explosão de atividades, resultados promissores e mais promessas. Promessas não cumpridas, dificuldades de implementação em razão de custos computacionais elevados e o aparecimento de métodos alternativos trouxeram um declínio nos anos 90. Neste milênio, o aparecimento da GPU (Unidade de Processamento Gráfico, na sigla em inglês) e seu barateamento graças ao enorme mercado de jogos de computador, além de avanços técnicos na implementação de algoritmos, permitiram a solução para os problemas da década anterior. O novo poder computacional possibilitou considerar redes com grande número de camadas internas, as redes de arquitetura profunda. O perceptron de muitas camadas passou a ser conhecido como máquina de aprendiza-

gem profunda ou *deep learning*. Com isso vieram resultados impressionantes e grande atividade de pesquisa e aplicações. Nesta nova fase, grandes empresas de tecnologia apostam alto e investem em pesquisa que antes era feita em universidades. As notícias inundam os jornais, o jargão invade a linguagem dos responsáveis por *marketing* e novas promessas caracterizam esta nova etapa. O avanço foi impressionante, e isso leva a um novo exagero nas promessas. Quanto tempo passará até a próxima inversão do ciclo? Obviamente o estado atual não será eterno e grandes mudanças ocorrerão.

REPRESENTAÇÕES E EXPLICABILIDADE

Suponhamos que imagens, ou pacientes, ou clientes, devam ser classificados em categorias. A imagem pode ser de paisagens ou de objetos artificiais, o paciente, doente ou sadio; se doente, podemos perguntar se vai ou não se beneficiar de um tratamento agressivo. Um cliente pode representar um risco de crédito ou não. Queremos classificar para tomar decisões. Gostei ou não gostei? Vou em frente ou paro? E, como essas tarefas podem ser muito repetitivas e cansativas ou talvez porque precisem de humanos com muita experiência, queremos uma máquina para resolver o problema.

A representação matemática de um objeto é a primeira questão a ser analisada. Em geral podemos atribuir um conjunto de números descrevendo as características que podem representar o objeto. Não há alternativa a usar um conjunto de números para representar dois objetos a ser classificados. Mas como escolher essas características e conhecer qual a dimensão dessa representação constituem uma área de intensa atividade de pesquisa.

Um problema associado à representação é saber a importância das características. Queremos determinar as características de um cliente que deveriam mudar para que seu crédito seja aprovado, ou identificar os *pixels* de uma imagem que foram mais úteis na sua classificação. Que genes expressos de uma célula mostram que é cancerosa? Ou seja, como explicar por que esta categoria e não aquela. Queremos satisfazer a curiosidade e dizer por quê.

Assim, cada objeto a ser classificado pode ser representado por um ponto no espaço de alta dimensão, $X_\mu = (\chi_{\mu 1}, \chi_{\mu 2}...\chi_{\mu 2})$. Uma imagem de 1000 x 1000 *pixels* será um simples ponto num espaço de $K = 10^6$ um milhão de dimensões, onde as coordenadas ${}_i\chi_{\mu i}$ são os valores numéricos das características. É natural introduzir a ideia de similaridade, talvez distância, nesse espaço, e é natural que objetos da mesma categoria estejam tipicamente mais próximos entre si do que em relação a objetos de outra categoria. Vemos que nessa linguagem há um tipo de problema onde IA pode ser descrita em termos de espaços de alta dimensionalidade e sua geometria. A rede neural treinada para um problema de classificação divide o espaço num número de regiões igual ao de categorias. A fronteira entre classes vai sendo mudada dinamicamente durante o processo de aprendizado, à medida que a informação contida nos exemplos for usada. Mas a rede pode receber objetos que não foram usados para construir essa fronteira e que também são pontos nesse espaço de alta dimensionalidade. A classificação prevista para esse novo objeto é dada em função da posição relativa à fronteira. A rede está pronta para

generalizar, fazer previsões para padrões não vistos a partir do seu estado atingido pelo processo de treinamento.

ARQUITETURAS PROFUNDAS E ALGORITMOS DE APRENDIZADO

A primeira peça do lego, o perceptron sem camada escondida ou o perceptron de uma camada (de pesos) tem K unidades de entrada e uma de saída. A entrada i está conectada à saída j, e nessa ligação reside um valor numérico w_{ij}, chamado peso sináptico por sua analogia biológica. Estes medem a influência de cada característica ($\chi_{\mu i}$) da representação χ_μ do objeto μ. A soma ponderada $h_{\mu j} = \Sigma w_{ij} \chi_{\mu i}$ chega à unidade de saída que emite um número, função de $h_{\mu j}$, tipicamente um sigmoide, que satura para valores muito maiores que um limiar b_j em 1 e valores muito menores em –1, sendo linear na região intermediária aproximadamente b_j. Aprender significa encontrar valores adequados dos pesos w_{ij} e do limiar b_j.

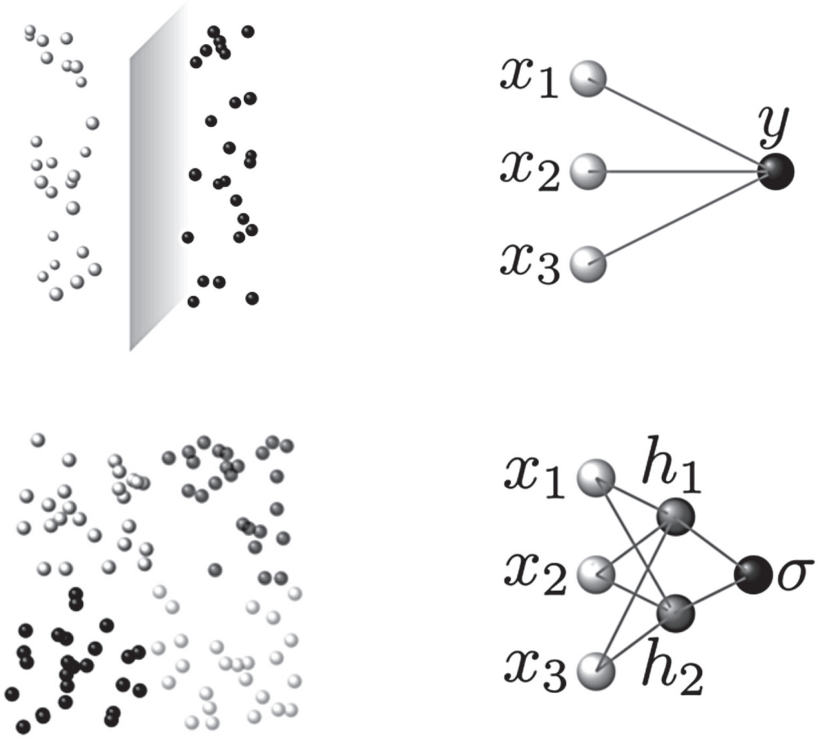
Há uma vasta fauna de algoritmos de aprendizado. Grosso modo, o processo consiste em encontrar mínimos de funções custo não negativas que se anulam somente se a saída para cada exemplo usado no treinamento é a correta. Parece simples encontrar o mínimo, mas na prática a força bruta não funciona e métodos sofisticados são necessários. Isso – que explica a impressão inicial de que redes de arquitetura profunda seriam impossíveis de treinar – se deve ao fato de que há muitos mínimos e só alguns são os desejáveis. Após o aprendizado a rede pode ser usada para classificar um novo objeto. Vamos tentar entender num caso simples como essa máquina processa a informação.

DINÂMICA DO PROCESSAMENTO DE INFORMAÇÃO

As nuvens de pontos mostrados na Figura 1 (superior, esquerda) podem ser separadas por planos em regiões que têm pontos de uma só categoria. O perceptron simples gera uma superfície de separação e é suficiente para a primeira nuvem, mas não para a segunda. Para resolver esse problema um pouco mais complicado é necessário juntar mais perceptrons para formar a máquina mostrada na Figura 1 (abaixo, direita). Agora temos três tipos de unidades: as da primeira camada (brancas) que recebem os valores usados para a representação original de cada um dos objetos na nuvem; a de saída (preto), que terá um valor interpretado como a categoria na qual um objeto é classificado; e o novo elemento, as unidades escondidas (cinza). Este problema é da classe de funções booleanas de duas variáveis conhecida como XOR, ou OU-Exclusivo. Imagine uma situação mais simples que envolve duas variáveis, cada uma pode tomar dois valores, +1 ou –1 que devem ser classificados em duas classes, que podemos representar por **1** e **-1**. Portanto há quatro objetos possíveis neste exemplo. O XOR é definido ao dizer que a classificação correta de (–1,–1) e (1,1) é **1** e a de (1,–1) e (–1,1) é **-1**.

O que fazem as unidades escondidas? Os seus valores servem como entrada do último perceptron de saída. Portanto, se a máquina inteira funciona é porque os valores que tomam, como saída da primeira camada, formam uma nuvem linearmente separável. Esses valores, para cada objeto da nuvem, são chamados de representação interna. Voltamos ao problema de representação dos objetos.

FIGURA 1



Acima, um problema linearmente separável. A nuvem de pontos pode ser separada por um (hiper) plano nas classes branco e preto e, portanto, um perceptron simples, mostrado à direita, é suficiente. As unidades de entrada em branco e a de saída em preto. Abaixo, padrões em preto e cinza-escuro são da mesma classe. Branco e cinza-claro também pertencem à outra classe. As duas classes não podem ser separadas por um plano. À direita, um perceptron multicamada com uma camada escondida (cinza) pode realizar a tarefa. As saídas da camada escondida formam a representação interna do objeto. A informação flui da esquerda para a direita.

Se pudéssemos escolher uma representação dos objetos que levasse a uma nuvem linearmente separável, o problema estaria solucionado. Não sendo isso possível, em geral, usamos um conjunto de perceptrons para gerar uma representação nova, na primeira camada escondida. Se for linearmente separável, acabamos. Se não for, outra camada levará a uma nova representação interna.

O número de unidades em cada camada interna pode variar, o que significa que a informação, ao se propagar pela rede, é usada para representar os objetos em espaços de dimensão que podem variar.

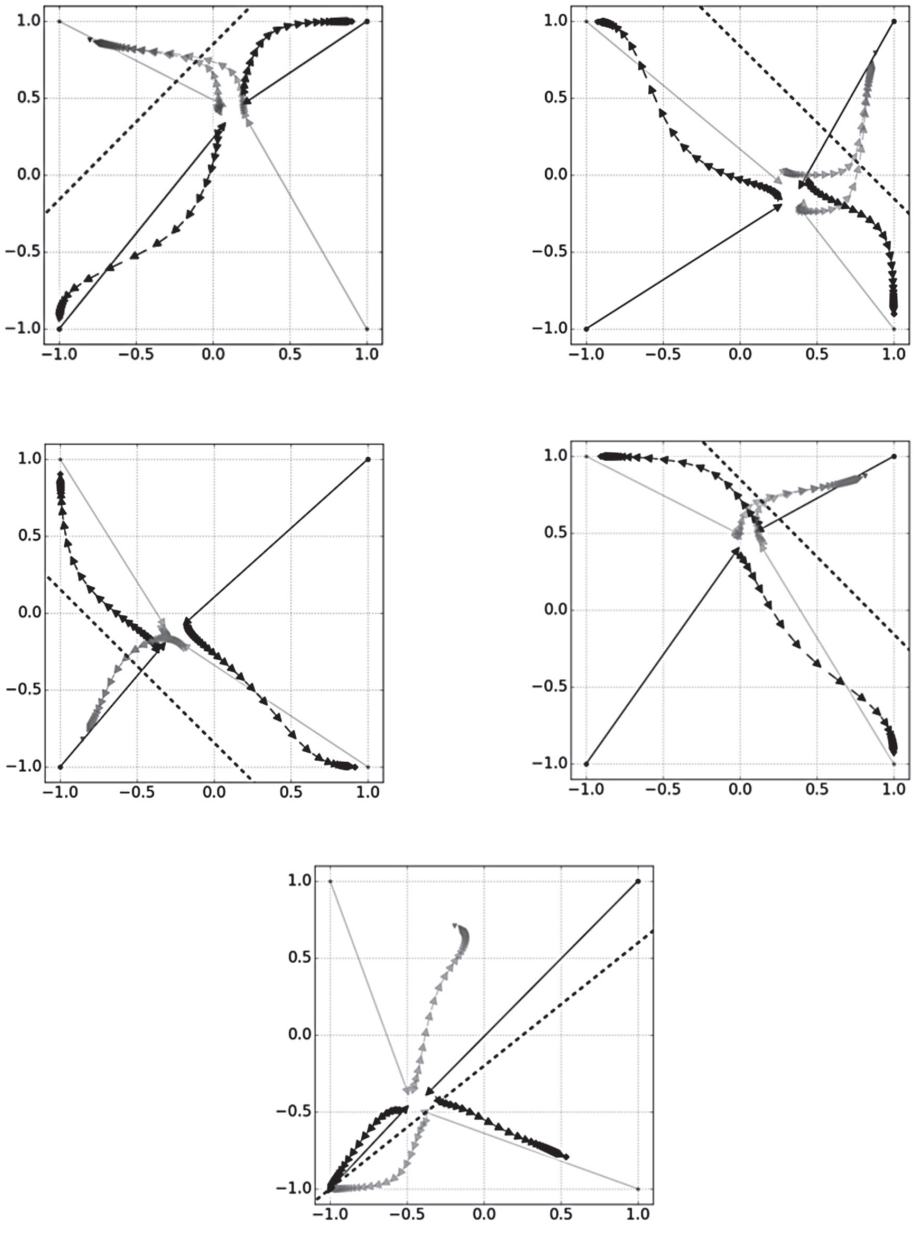
A pergunta que na década de 1960 quase levou as redes neurais à extinção continua sendo central. Como determinar os pesos que levam às representações internas sucessivas? Mas agora há vários métodos, e todos procedem pela minimização iterativa de uma função custo. Por exemplo, fazendo mudanças sucessivas nos pesos e limiares na direção de eliminar ou pelo menos reduzir os erros de classificação.

Agora descreveremos o processo de aprendizado usando *backpropagation* no problema XOR[1]. À medida que ocorre o aprendizado, há uma mudança de todos os parâmetros: os pesos e os limiares das duas camadas. Isso é um problema em oito dimensões e não temos muita esperança de poder representá-lo graficamente de forma clara seguindo os valores ao longo do tempo. Mais fácil é seguir as representações internas de cada um dos objetos. Apresentamos o resultado para várias condições iniciais na Figura 2. Os quatro objetos devem ser classificados em duas categorias: em preto e em cinza. Nas figuras vemos a evolução ao longo do aprendizado das representações internas. Inicialmente, ou seja, nos primeiros passos para minimizar o erro de saída da máquina usando *backpropagation*, há uma grande mudança na representação interna. Os quatro casos são rapidamente levados a uma região central e depois lentamente permanecem durante um grande número de iterações quase no mesmo lugar. Mas depois ocorre uma mudança bastante brusca nos pesos e as representações internas começam a se mover. Tipicamente, depois de vários passos do algoritmo, as representações internas de dois membros de uma classe colidem, se juntam e migram até um dos quatro cantos. Os membros da outra classe migram para outros cantos, opostos entre si, e se afastam o máximo possível. A colisão das representações internas torna o problema linearmente separável, porque três pontos, em posições gerais, podem ser separados com probabilidade 1 por uma reta, mostrada em preto. Na Figura 2 ainda mostramos, na quinta figura, um caso em que o algoritmo de aprendizado falhou: a colisão se dá entre membros de classes diferentes e a máquina minimiza o erro colocando a borda de separação justo em cima desse par. A máquina está em dúvida. Tecnicamente o que ocorre é que no espaço de oito dimensões há várias regiões diferentes que resolvem o problema. Os quatro casos mostrados são apenas um subconjunto de 16 classes possíveis de soluções. O último caso mostra que há outros mínimos onde a dinâmica pode ficar presa. As diferenças entre estes casos são devidas a pequenas mudanças das condições iniciais. Todo o resto é idêntico.

Na primeira fase da dinâmica, as representações internas são levadas a uma região em que as funções de transferências são essencialmente lineares. Há uma longa fase em que os dois perceptrons escondidos são bastante parecidos, isto é, há uma simetria entre eles e o problema não pode ser separado linearmente. Mas depois de um longo platô em que parece que nada acontece, uma súbita quebra da simetria leva ao que tem sido chamado de *aha*. Antropomorfizando a máquina, podemos descrever esse momento como aquele em que *ela entendeu* o problema. Os perceptrons internos se especializam, o problema fica linearmente separável e agora as representações migram para as regiões onde a máquina é saturada e o processamento, não linear. O resultado

1 Nota técnica: A função de transferência das unidades é $\sigma(h) = (1 + e^{-\beta h})^{-1}$ com $h_i = \Sigma\ w_{ij}\ \chi_i$, com $\chi_3 = 1$ para todos os exemplos. Os vetores são apresentados simultaneamente e a as mudanças dos pesos $w(t + 1) = w(t) - \eta \nabla_w E$, onde E é o erro quadrático somado sobre todos os exemplos, $\beta = 0.2$ e $\eta = 0.15$. O algoritmo é iterado 210^4 vezes. As condições iniciais para a primeira camada de pesos foram $w_{11} = w_{22} = w_{31} = w_{32} = 1$ e $w_{12} = w_{21} = 0$. Os dois pesos e o limiar da segunda camada foram escolhidos aleatoriamente de forma independente de uma distribuição uniforme entre 1 e –1.

FIGURA 2



Dinâmica de aprendizado e evolução das representações internas no XOR com a rede com camada escondida da Figura 1. Em todas as figuras, os pontos iniciais (-1,-1), (-1,1), (1,-1) e (1,1) são os mesmos, mas as representações internas finais são diferentes. Nas quatro figuras superiores a rede pode separar corretamente os exemplos nas duas categorias: preto numa categoria e cinza na outra. A separação é feita pela linha preta tracejada, é a borda entre as classes. Na figura inferior, a RI de dois padrões de classes diferentes colidem em (-1,-1) e ficam sobre a borda da dúvida. As diferenças entre as figuras decorrem da escolha aleatória dos pesos iniciais da segunda camada. Em todos os casos há uma fase rápida de contração, uma fase longa onde pouco acontece e finalmente um avanço rápido até as RI finais.

dessa saturação é que o sinal é muito mais claro que no regime linear.

DISCUSSÃO

O exemplo acima pode ser considerado o caso mais simples possível e, portanto, não conta toda a história de como as máquinas de arquitetura profunda funcionam. Mas, ao entender como essa máquina funciona ao mudar as representações internas, podemos começar a descrever processos muito mais complexos. Para objetos representados em espaços de dimensionalidade muito mais alta, para manter a dimensão das camadas internas aproximadamente da mesma ordem que a de entrada, serão necessárias mais camadas internas. Cada uma representará os objetos de forma diferente; alguns se aproximarão e outros se afastarão, tornando a geometria das nuvens mais perto daquela que pode ser resolvida no último passo por uma máquina tão simples como o perceptron.

Não cabe aqui descrever todas as aplicações, nem sequer todos os tipos de uso que podem ter. Concluiremos com uma descrição de alguns objetivos que uma teoria desse tipo de sistema de processamento de informação pode ter. A descrição de processamento de informação por mudança de representação interna não dá conta do problema da explicabilidade. Quando não sabemos resolver um problema através de regras específicas, não satisfazemos nem a curiosidade nem as demandas legais. É claro que muitas vezes a explicação da ação por um humano é posterior à sua decisão e a mesma ação pode ter diferentes explicações pós-fato. Talvez estejamos pedindo mais explicações de uma rede neural que de um humano.

As RNA têm tido algumas vitórias importantes sobre humanos em tarefas que requerem ações muito repetitivas e geração de cenários como xadrez e outros jogos. Mas isso não quer dizer muito mais que a constatação de que o xadrez é em algum sentido um jogo simples, além do fato de que centenas de pessoas trabalharam para construir a máquina e esta incorpora um conjunto de informações que nenhuma máquina pode selecionar ou curar no estágio atual. O cérebro humano tem muito menos camadas que as redes de arquitetura profunda, mas sistemas de retroalimentação muito mais complexos. A necessidade de grande quantidade de dados para treinar uma rede, comparada ao pequeno número de exemplos que uma criança precisa, também mostra que ainda não entendemos como reproduzir o cérebro. A capacidade de se adaptar rapidamente a ambientes que mudam é uma área de pesquisa importante e que tem um longo caminho a percorrer.

Embora possamos descrever o funcionamento de uma rede neural de arquitetura profunda de várias formas, escolhemos neste artigo a descrição em termos da construção e reconstrução de representações internas à medida que a informação se propaga pela rede. Um fato que sempre acho surpreendente, embora seja natural do ponto de vista matemático, é que podemos ter diferentes RNA que respondem exatamente da mesma forma às questões, mas que têm representações internas muito diferentes. Se eu peço que olhem para um objeto, uma maçã talvez, é possível conceber que muitos concordarão que tem cor vermelha, ou seja, responderão da mesma forma a certas perguntas. Além disso, terão atividade no cérebro em regiões análogas. Mas a representação interna, aquela atividade que nos dá a sensação do vermelho, será a mesma?