Princípios de Redes Neurais Artificiais e de Reconhecimento de Padrões

Prof. Antônio de Pádua Braga
Departamento de Engenharia Eletrônica
Escola de Engenharia da UFMG

15 de março de 2017

Sumário

Capítulo 1

Conceitos Básicos

1.1 Introdução

Nas páginas que se seguem serão apresentados os fundamentos de Redes Neurais Artificiais (RNAs) e de Reconhecimento de Padrões (RP) organizados na forma de uma disciplina única com capítulos intercalados entre um assunto e outro. Esta forma de organização se justifica, já que as RNAs se apresentam como uma das possíveis e mais populares abordagens para a resolução de problemas de RP. Não obstante, as RNAs se aplicam a uma gama maior de problemas e de áreas do conhecimento, não estando a mesma restrita a problemas de RP. Por serem aproximadores universais de funções [Cyb89] as RNAs podem ser aplicadas na construção de funções discriminantes, com aplicação direta em reconhecimento de padrões. A universalidade na aproximação de funções a partir de amostras de dados dá às RNAs uma abrangência maior, podendo a mesma ser aplicada também a problemas de regressão e de previsão.

A Figura 1.1 mostra de maneira esquemática um diagrama que representa o relacionamento entre estas as três grandes áreas citadas no parágrafo anterior. Os problemas de regressão, previsão e classificação podem ser tratados por um grande número de abordagens, entre elas as RNAs, representada na figura na intersecção entre as três áreas. Por sua vez, os problemas de classificação, que podem ser tratados por RNAs, envolvem basicamente o reconhecimento de padrões, já que esta tarefa envolve associar um vetor de entrada a uma entre várias categorias (ou classes) de vetores previamente conhecidas. Os problemas de RP, por sua vez, também podem ser tratados por um grande número de métodos, conforme representado no diagrama da figura.

1.1.1 Problemas de Regressão

A regressão envolve a estimativa da relação entre variáveis por meio de métodos de indução de funções a partir de amostras destas variáveis. A estimativa da função aproximadora, ou regressora, pode ser realizada de várias formas, entre elas por meio de RNAs. No entanto, a estimativa por meio de RNAs tem algumas características interessantes, que as tornam particularmente atrativas para esta categoria de problemas, entre elas, a capacidade de aproximação não-linear com um número de parâmetros que cresce apenas linearmente com o número

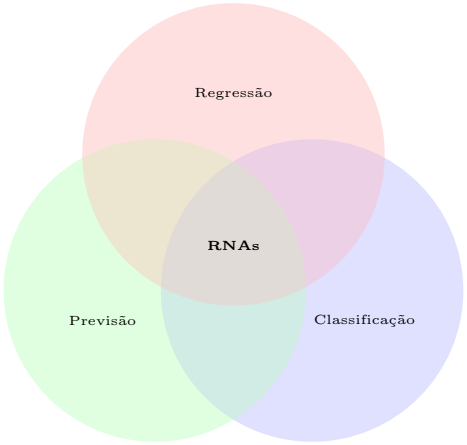


Figura 1.1: Principais áreas de aplicação das RNAs. A resolução de problemas de classificação pode ser aplicada aos problemas de reconhecimento de padrões.

de variáveis. Modelos que envolvem a combinação de variáveis de entradas podem se tornar inviáveis quando o número destas variáveis é muito grande, o que ocorre em boa parte dos problemas reais.

Como exemplo, considere um processo de combustão industrial, para o qual deseja-se estimar o valor esperado de uma variável dependente, como, por exemplo, a pressão interna de uma caldeira, a partir de variáveis independentes, como aquelas relacionadas à combustão propriamente dita, como as vazões de combustível e de ar nos queimadores. O objetivo do regressor seria, neste caso, estimar a pressão, dados os valores das vazões. A construção, ou indução, do regressor seria realizada por meio de exemplos de amostras das vazões e da pressão. Assim, o problema de regressão envolve estimar uma função ou, em outras palavras, os seus parâmetros, que represente a relação entre variáveis dependentes e independentes.

Um exemplo de regressão de uma única variável é apresentado na Figura 1.2, em que os círculos representam os dados amostrados de x (variável independente) e y (variável dependente) e a linha contínua representa a resposta da função aproximadora $\hat{y} = \hat{f}(x)$ em um determinado intervalo. Como pode ser observado na figura, a função obtida parece aproximar bem a relação entre as variáveis x e y.

1.1.2 Problemas de Previsão

De maneira análoga à regressão, os problemas de previsão também visam a estimar uma relação entre variáveis, porém, neste caso há uma relação temporal entre as variáveis independentes e dependentes. Considere, por exemplo, o problema de prever o valor de fechamento da bolsa de valores de São Paulo, o IBOVESPA, que, neste exemplo, é a variável dependente. Quais são os fatores, ou variáveis, que influenciam o IBOVESPA? Podemos pensar em vários, por exemplo os preços das *commodities* que influenciam os índices majoritários que compõem o IBOVESPA, como as ações da Petrobras, Vale e as outras chamadas *blue chips*. Assim, o valor do barril de petróleo, chamado de Brent,

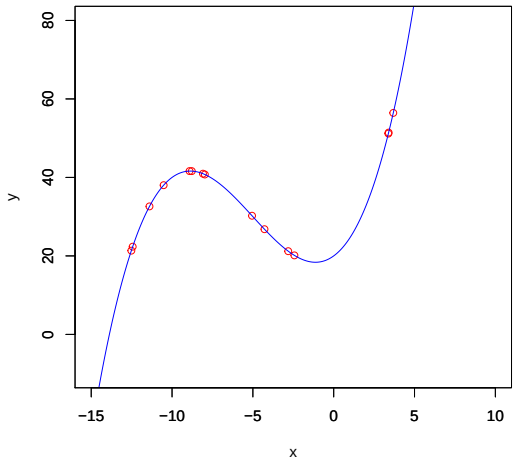


Figura 1.2: Exemplo de regressão. Círculos representam as amostras de x e y e a linha contínua a resposta da função estimada no intervalo de -15 a $+5$.

negociado na bolsa de Londres pode ser considerado uma variável independente para a construção do estimador que terá como objetivo prever valores futuros do IBOVESPA, como o valor de fechamento do dia seguinte. O modelo de previsão poderá, portanto, ser composto por esta e por outras variáveis que influenciem o índice. Porém, em problemas de previsão deve-se considerar também os atrasos (*lags*) de tempo em que uma variável influencia a outra, ou seja, uma variação no preço do Brent hoje, ou de qualquer outra variável independente, pode levar algum tempo para ser incorporado ao IBOVESPA. Portanto, em problemas de previsão, as variáveis de entrada são incorporadas ao modelo com atrasos de tempo diferentes, de acordo com o tempo que cada uma delas leva para influenciar a saída. Modelos de previsão podem ser descritos de maneira genérica como na expressão da Equação 1.1.

$$\hat{y}(t+1) = \hat{f}(y(t), y(t-1), \cdots, x_1(t), x_1(t-1), \cdots, x_n(t), x_n(t-1), \cdots) \quad (1.1)$$

em que $t, t-1, \cdots$, indica o instante de tempo em que a variável é amostrada, $y(\cdot)$ é a variável de saída e $x_1(\cdot), \cdots, x_n(\cdot)$ as variáveis de entrada.

A função aproximadora, ou modelo, será composta conforme conforme Equação 1.1, as variáveis de entrada e a saída realimentada serão consideradas naqueles instantes de tempo em que influenciam o valor seguinte da saída. A identificação dos *lags* entre as entradas e a saída pode ser realizada por meio de algum conhecimento prévio sobre o problema, por meio de experimentos ou através de análise dos dados utilizando técnicas como a correlação cruzada.

1.1.3 Problemas de Classificação

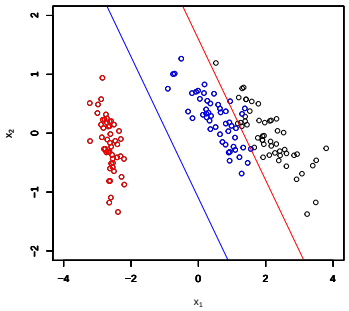
Os problemas de classificação envolvem a associação de uma amostra de entrada a uma classe conhecida. Da mesma forma que nos problemas de regressão e previsão um vetor de entrada é associado a um valor da variável dependente, a resposta da função classificadora também resulta em um valor de saída que, neste caso, indica a classe resultante. De maneira análoga à previsão, o problema de classificação, ou RP, é também essencialmente um problema de regressão, já que envolve a busca por uma função discriminante, usualmente obtida por meio da partição do espaço de entrada em regiões distintas para cada uma das classes que definem o problema. Como amostras da mesma classes devem estar espacialmente próximas no espaço de entrada, estas caracterizarão regiões para cada uma das classes. O objetivo da funçao discriminante é, portanto, identificar estas regiões, delimitá-las e identificar a região do espaço em que está localizada uma nova amostra, indicando a classe correspondente.

Um exemplo de classificação envolvendo três classes distintas é apresentado na Figura 1.3. As amostras que caracterizam as classes são apresentadas em cores distintas no gráfico, sendo vermelho para a classe 1, azul para a classe 2 e preto para a classe 3. Pode ser observado também que a classe 1 está mais afastada das demais, havendo alguma superposição entre as classes 2 e 3, o que pode levar a uma melhor discriminação da classe 1. Isto pode ocorrer em vários problemas reais e pode ser explicado pelo fato de as variáveis independentes, atributos x_1 e x_2, selecionadas para representar o problema são mais discriminativos em relação à classe 1. A superposição entre as amostras das classes 1 e 2 indica que poderá haver um maior erro na discriminação das amostras destas classes, especialmente na região de fronteira. Muitas vezes não é possível eliminar esta superposição, já que ela pode ser inerente ao problema e à sua representação, o que pode levar o classificador a ter um erro intrínsico ao separar amostras destas duas classes.

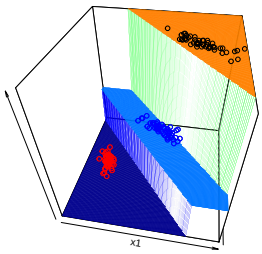
Com base, portanto, nas amostras apresentadas na Figura 1.3 deseja-se inicialmente identificar as regiões de cada classe. Uma análise simples da distribuição das amostras sugere que as classes podem ser separadas por retas, também indicadas na figura, e a classificação poderia ser feita identificando em qual das três regiões delimitadas foi mapeada a amostra a ser classificada. A função discriminante, neste caso, seria obtida pela composição das funções discriminantes correspondentes a cada uma das retas. As funções indicadoras, correspondentes a cada uma das retas, deverão indicar se uma amostra está localizada acima ou abaixo da mesma, respondendo com 1, por exemplo, caso esteja acima e com 0 caso contrário. Assim, para a classe 1 as respostas das funções das retas resultarão na tupla 00, já que as amostras desta classe estão abaixo de ambas as retas. De maneira análoga as tupas 01 e 11 indicarão amostras das classes 2 e 3. O classificador deverá, portanto, gerar como resposta um número indicando a classe da amostra de entrada com base nas tuplas acima. Isto pode ser feito simplesmente convertendo para decimal as respostas individuais das tuplas por meio da seguinte expressão: $\hat{y} = 2h_2(x_1, x_2) + h_1(x_1, x_2)$, em que $h_2(x_1, x_2)$ e $h_1(x_1, x_2)$ são as funções indicadoras correspondentes às duas retas e x_1 e x_2 seus argumentos. As respostas do classificador serão 0, 1 e 3 para as classe 1, 2 e 3, respectivamente, conforme indicado na superfície de resposta da Figura 1.3b.

A função das duas variáveis x_1 e x_2 apresentada na Figura 1.3b representa, portanto, a resposta do classificador e reforça a ideia de que o problema de

classificação é essencialmente um problema de regressão, que visa a encontrar a função $\hat{f}(x_1, x_2)$ que seja capaz de discriminar as classes do problema.



(a) Amostras de três classes distintas, indicadas por cores diferentes e retas de separação que resultam na superfície de resposta da Figura 1.3b.



(b) Superfície resultante de um classificador que é capaz de discriminar as 3 classes da Figura 1.3.

Figura 1.3: Exemplo de um problema de classificação (RP) de três classes.

1.2 Redes Neurais Artificiais

As RNAs são formadas por elementos básicos, os neurônios artificiais, os quais executam funções matemáticas que representam modelos de neurônios biológicos. Dependendo de quais características reais tenham sido incorporadas ao modelo, estes podem variar quanto à complexidade e quanto à demanda por recursos computacionais para sua implementação. Não obstante, estruturas de RNAs utilizando o modelo simplificado de McCulloch e Pitts (modelo MCP [MP43]) são capazes de representar funções matemáticas bastante complexas, apesar de estes modelos serem bastante simples do ponto de vista matemático. Os modelos neurais artificiais partem, portanto, de conhecimentos do comportamento dos neurônios biológicos e da representação do mesmo na forma de expressões matemáticas. De uma maneira geral, a descrição e modelagem de processos biológicos e mentais podem ser realizadas nas formas **sintéticas** e **analíticas** [Fra95].

As abordagens sintéticas visam a reproduzir as estruturas biológicas por meio da construção de modelos que se assemelham às estruturas reais, como, por exemplo, as RNAs. Por sua vez, a abordagem analítica representar o comportamento observável de processos existentes sem necessariamente reproduzir as estruturas das quais emergem o comportamento observado. Aquelas abordagens que partem de estruturas biológicas básicas, como neurônios e suas redes de conexões, para a construção de estruturas de nível mais alto são classificadas como *bottom-up*. Já as abordagens *top-down* partem de comportamentos observáveis para construir modelos que os reproduzam sem necessariamente se preocupar com a semelhança entre os modelos e as estruturas biológicas. Um exemplo de abordagem sintética *top-down* é a Inteligência Artificial Clássica (IA), que se baseia em fatos e regras para representar processos dedutivos que se assemelhem àqueles realizados pelos seres humanos.

Espera-se, no entanto, independemente da plausibilidade biológica, que as propriedades emergentes dos modelos neurais artificiais sejam capazes de reproduzir comportamentos característicos dos animais, os quais, além de serem dotados da capacidade de aprender de forma interativa, são também capazes de prever situações futuras, de classificar eventos, de agrupar informações, de induzir comportamentos e de lidar com informações parciais, distorcidas ou incompletas. Estas são algumas das capacidades dos animais que podem ser também reproduzidas pelas RNAs. Ao mesmo tempo em que a reprodução destas características emergentes pode ajudar no melhor entendimento de processos biológicos e cognitivos, a resolução de problemas associados à classificação de padrões, à previsão e ao agrupamento de dados por meio de *aprendizado* pode também proporcionar novas perspectivas para a solução de problemas práticos do nosso dia-a-dia.

Modelos biologicamente inspirados formam a base para a pesquisa em muitas áreas em torno da Neurociência, cujos avanços nos últimos anos proporcionaram grandes progressos em várias áreas do conhecimento. Há, no entanto, um compromisso entre a plausibilidade biológica de um modelo neural artificial e a sua complexidade computacional. É claro que quanto maior a fidelidade do modelo em relação aos neurônios biológicos, maior a demanda por recursos computacionais para a sua implementação. Assim, muitos dos modelos utilizados para resolução de problemas computacionais são versões simplificadas dos neurônios biológicos, com menor plausibilidade biológica, porém, com grande capacidade computacional, especialmente quando organizados na estrutura de redes.

1.3 Modelos de soma e limiar

Considere um problema de classificação simples cujo objetivo seja identificar se uma determinada amostra de testes x_t pertence à classe das amostras vermelhas ou das azuis, conforme Figura 1.4. Parece óbvio, por inspeção da figura, que um classificador simples para resolver este problema poderia avaliar a posição de x_t em relação a um valor de limiar θ sobre o eixo x, como por exemplo $\theta = 3$. Caso $x_t \geq 3$, então a amostra seria classificada como pertencente à classe das amostras azuis e, caso $x_t < 3$ então a amostra seria classificada como pertencente à classe das amostras vermelhas. A função do classificador também pode ser descrita através da avaliação do sinal da operação $x_t - 3$, ou seja, caso $x_t - 3 \geq 0$ então x_t pertence à classe das amostras azuis e caso $x_t < 0$ então x_t pertence à classe das amostras vermelhas. A operação $x_t - 3$ é, na verdade, a distância com sinal de x_t em relação ao limiar θ. Neste caso, a função de decisão $f(u)$ deve receber como argumento a distância $u = x_t - 3$ e sobre esta realizar a classificação com base em seu sinal. A Figura 1.4 também mostra o resultado da aplicação desta função de classificação para o intervalo $0 \leq x_t \leq 6$, o que resultou em uma resposta de classificação equivalente a uma função degrau. Neste caso, $f(u) = 0$ indica classe vermelha e $f(u) = 1$ indica classe azul. O comportamento do modelo artificial de McCulloch e Pitts [MP43] é semelhante ao apresentado no exemplo da Figura 1.4.

O exemplo descrito na Figura 1.4 mostra um classificador simples de uma única variável x baseado em uma função de ativação $f(u)$ do tipo degrau, em que u é uma medida de distância com sinal entre a amostra que se deseja classificar e um separador, caracterizado neste caso pelo limiar θ. A resposta do modelo de

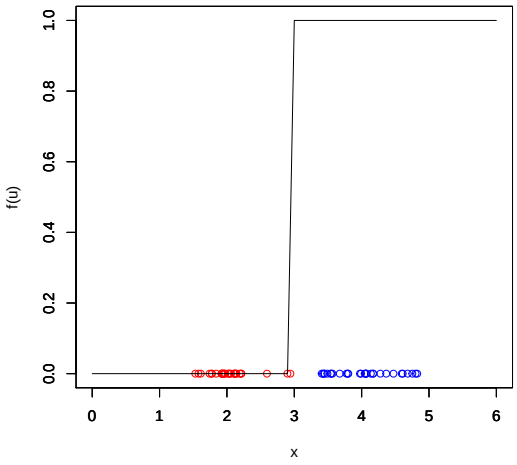


Figura 1.4: Problema de classificação simples cujo objetivo é classificar uma amostra arbitrária como pertencente à classe das amostras vermelhas ou azuis.

McCulloch e Pitts [MP43], representado de maneira esquemática na Figura 1.5, baseia-se também na aplicação de uma função de ativação $f(u)$ sobre o argumento u, o qual é também uma medida de distância entre a amostra de entrada e um hiperplano, caracterizado pelo separador. A medida de distância, neste caso, é o produto interno $u = w_1x_1 + w_2x_2 + \cdots + w_nx_n$ entre o vetor de entrada $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$ e o vetor de pesos $\mathbf{w} = [w_1, w_2, \cdots, w_n]^T$, que caracteriza o hiperplano separador. Sobre o resultado deste produto é aplicada a função de ativação $f(u)$, a qual pode assumir várias formas, entre elas a função degrau representada na Equação 1.2, o que resulta em um modelo com comportamento análogo ao classificador da Figura 1.4.

$$f(u) = \begin{cases} 1 & u \geq \theta \\ 0 & u < \theta \end{cases} \tag{1.2}$$

em que $u = \sum_i w_i x_i$.

1.4 Estrutura de uma rede neural artificial

Uma RNA é caracterizada por uma estrutura de neurônios artificiais interconectados, os quais executam individualmente funções como aquela descrita na Equaçao 1.2. A forma da função de ativação $f(u)$ pode variar de acordo com o modelo adotado, porém, o seu argumento u será tipicamente caracterizado pela soma dos elementos x_i do vetor de entrada $\mathbf{x}$, multiplicados pelos pesos correspondentes w_i do vetor $\mathbf{w}$. Esta soma ponderada, que pode ser representada pelo produto interno $u = \sum_i w_i x_i = \mathbf{w}^T\mathbf{x}$, é também uma medida de correlação, ou de proximidade, entre $\mathbf{w}$ e $\mathbf{x}$.

Uma RNA do tipo *feed-forward* (alimentada para frente) é representada de

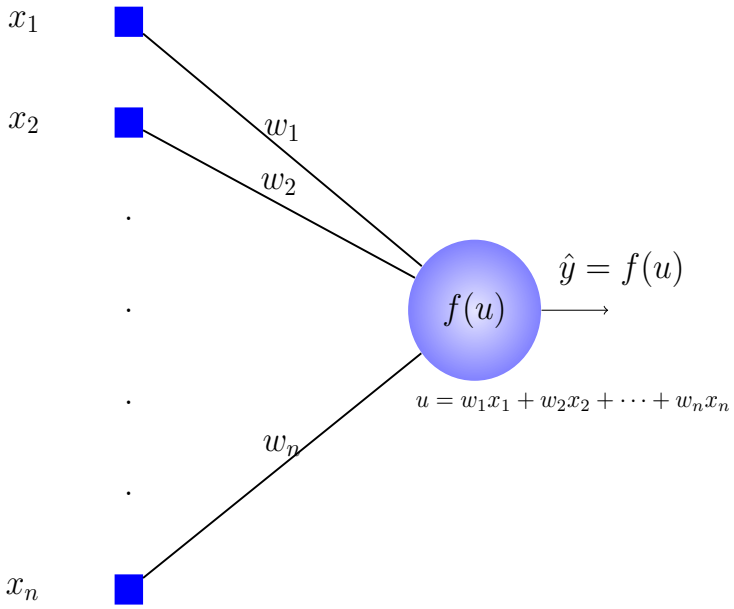


Figura 1.5: Representação esquemática do modelo neural de McCulloch e Pitts [MP43].

forma esquemática na Figura 1.6. Esta estrutura de duas camadas recebe como entradas os elementos do vetor $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$, o qual é propagado até a saída por meio dos neurônios $h_i(\mathbf{x}, \mathbf{z}_i)$ da camada intermediária, em que $\mathbf{z}_i$ representa o seu vetor de pesos. Assim, cada neurônio da camada intermediária executa a função $h_i(u_i)$, análoga àquela da Equação 1.2 em que $u_i = \sum_i z_i x_i$. Os p vetores de pesos dos neurônios da camada intermediária compõem as colunas da matriz de pesos $\mathbf{Z}$ de dimensões $n \times p$. A propagação do vetor de entrada $\mathbf{x}$ até as saídas dos neurônios da camada intermediária resulta no vetor $\mathbf{h} = [h_1(\mathbf{x}, \mathbf{z}_1), h_2(\mathbf{x}, \mathbf{z}_2), \cdots, h_p(\mathbf{x}, \mathbf{z}_p)]^T$, o qual será aplicado às entradas do neurônio de saída para o cálculo da saída da rede neural. A função que representa o mapeamento entre o espaço de entrada e o espaço formado pelos neurônios da camada intermediária será representada aqui como $\Phi_h(\mathbf{x}, \mathbf{Z})$), em que a matriz $\mathbf{Z}$ contém, em cada linha, os parâmetros de cada um dos neurônios desta camada. O mapeamento entre o espaço da camada intermediária e a saída da rede será representado aqui pela função $\Phi_o(\mathbf{h}, \mathbf{w})$), em que $\mathbf{w}$ é o vetor que contém os seus parâmetros, considerando-se, sem perda de generalidade, um modelo de uma única saída, como aquele representado na Figura 1.6.

Assim, a saída deste modelo de RNA é obtida por meio de projeções sucessivas, inicialmente para a camada intermediária, $R^n \rightarrow R^p$, e posteriormente para a saída, $R^p \rightarrow R^1$. De maneira geral, $f(\mathbf{x}, \mathbf{Z}, \mathbf{w})$ representa a função executada pela RNA da Figura 1.6, cujo argumento é o vetor de entrada $\mathbf{x}$ e os parâmetros $\mathbf{Z}$ e $\mathbf{w}$ são obtidos durante o treinamento. Uma estrutura como esta é um aproximador universal de funções contínuas, conforme demonstrado formalmente no trabalho de Cybenko [Cyb89].

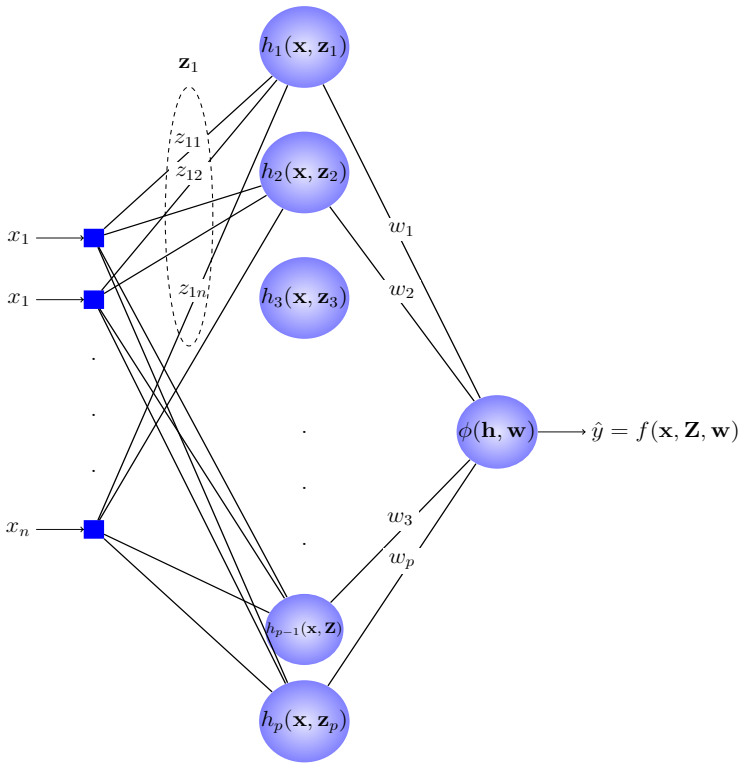


Figura 1.6: Representação geral de uma rede neural de uma saída e duas camadas.

1.4.1 Representação matemática

A estrutura em camadas da Figura 1.6 pode ser descrita na forma de uma função composta, conforme representado na Equação 1.3.

$$f(\mathbf{x}, \mathbf{z}_1 \cdots \mathbf{z}_p, w_1 \cdots w_p) = \Phi_o \left(h_1(\mathbf{x}, \mathbf{z}_1) w_1 + \cdots + h_p(\mathbf{x}, \mathbf{z}_p) w_p + \beta \right) \tag{1.3}$$

$$f(\mathbf{x}, \mathbf{z}_1 \cdots \mathbf{z}_p, w_1 \cdots w_p) = \Phi_o \left(\sum_{i=1}^{p} h_i(\mathbf{x}, \mathbf{z}_i) w_i + \beta \right) \tag{1.4}$$

onde p é o número de neurônios da camada escondida, $\Phi_o(\cdot)$ é a função de ativação do neurônio de saída, $h_i(\cdot)$ é função de ativação do neurônio i da camada escondida, w_i é o peso da conexão do neurônio i ao neurônio de saída, β é o termo de polarização do neurônio de saída, $\mathbf{w} = [w_1, \cdots, w_p]^T$ é o vetor de pesos do neurônio de saída e $\mathbf{Z} = [z_1, \cdots, z_p]^T$ a matriz de pesos da camada intermediária.

Conforme representado nas Equações 1.3 e 1.4 uma RNA pode ser descrita na forma de composição de funções, usualmente não lineares, cujos argumentos são combinações lineares das suas entradas. Os coeficientes da combinação linear são os parâmetros dos neurônios, os quais são obtidos por meio de aprendizado, conforme será descrito conceitualmente na seção seguinte.

1.5 Aprendizado de Redes Neurais Artificiais

A expressão $f(\mathbf{x}, \mathbf{Z}, \mathbf{w})$ representa, de maneira geral, a RNA da Figura 1.6, em que os elementos da matriz $\mathbf{Z}$ e do vetor $\mathbf{w}$ são os únicos parâmetros que determinam o comportamento das funções $h_i(\cdot)$ e $\phi(\cdot)$. Assim, para que haja aprendizado, os parâmetros em $\mathbf{Z}$ e em $\mathbf{w}$ deverão ser modificados para que a RNA seja adaptada visando a representar uma determinada função a ser induzida a partir de um conjunto de dados amostrados. O aprendizado de redes neurais artificiais envolve, portanto, a adaptação de seus parâmetros por meio de um processo interativo com o meio externo, de forma que uma determinada função $f_g(\mathbf{x})$ seja representada na forma induzida $f(\mathbf{x}, \mathbf{Z}, \mathbf{w})$.

Visando a simplificar a sua representação, em vários dos capítulos seguintes adotaremos também a expressão $f(\mathbf{x}, \mathbf{w})$ para representar a função executada pela RNA, em que $\mathbf{w}$ é o vetor que contém a concatenação de todos os parâmetros do modelo, ou seja, os elementos de $\mathbf{Z}$ e $\mathbf{w}$. Assim, objetivo do aprendizado é, portanto, encontrar o vetor de parâmetros $\mathbf{w}$ que satisfaça a um determinado critério, ou função-objetivo. Obviamente, para que haja aprendizado é preciso que o desempenho do modelo, segundo um critério pré-determinado, melhore gradualmente através da adaptação de $\mathbf{w}$. O critério que determinará a adaptação dos parâmetros é usualmente representado por uma uma função-objetivo, que quantifica quão distante está a saída $\hat{y}_i = f(\mathbf{x}_i, \mathbf{w})$ do valor alvo, normalmente representado por valores de saída y_i de um conjunto de amostras $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$. Assim, considerando-se que D seja representativo, espera-se que o aprendizado resulte na aproximação de y_i pela RNA, ou seja, que $f(\mathbf{x}_i, \mathbf{w}) \approx y_i \ \forall \mathbf{x}_i \in D$. Assim, uma possível função-objetivo, frequentemente utilizada por sua simplicidade, é o erro médio quadrático, representado pela expressão $J = \frac{1}{N}\sum_i (y_i - f(\mathbf{x}_i, \mathbf{w}))^2$.

O aprendizado de modelos neurais, como o representado na Figura 1.6, é normalmente realizado segundo o paradigma do Aprendizado Supervisionado, em que a resposta $f(\mathbf{x}_i, \mathbf{w})$ do modelo é calculada e comparada com a saída esperada y_i. Por meio da função de custo $J(y, f(\mathbf{x}, \mathbf{w}))$, o supervisor, externo ao modelo, determinará quão distante $f(\mathbf{x}_i, \mathbf{w})$ está de y_i, assim como a direção de ajuste de $\mathbf{w}$ que resultará na diminuição desta distância.

Com base na informação de direção obtida por meio da função $J(y, f(\mathbf{x}, \mathbf{w}))$, o supervisor, que incorpora o algoritmo de aprendizado, fará o ajuste de $\mathbf{w}$ para todos os pares $(\mathbf{x}_i, y_i)$ do conjunto D. Não obstante, este critério não garante o objetivo maior do aprendizado que é aproximar a função geradora $f_g(\mathbf{x})$ dos dados e não somente entre aqueles valores y_i presentes no conjunto de amostras, que são na verdade uma representação de $f_g(\mathbf{x})$. No caso extremo em que o conjunto D é representativo e suficientemente grande ($N \to \infty$), teremos $f(\mathbf{x}_i, \mathbf{w}) \approx f_g(\mathbf{x}_i) \forall \mathbf{x}_i \in R^n$ ao final do aprendizado. Assim, o objetivo do treinamento é, na verdade, aproximar $f_g(\mathbf{x})$ em todo o domínio da variável $\mathbf{x}$ e não somente entre os elementos do conjunto de treinamento D. O grande desafio dos algoritmos de treinamento surge devido ao fato que, em situações reais o tamanho N do conjunto de amostras é limitado e, mesmo com esta restrição, deve-se buscar a aproximação $f(\mathbf{x}, \mathbf{w}) \approx f_g(\mathbf{x})$.

As implicações relativas ao tamanho de D e à sua representatividade na generalidade da função $f(\mathbf{x}, \mathbf{w})$ serão discutidas nos capítulos seguintes. Por ora, o leitor deve ter em mente que o aprendizado de redes neurais visa a induzir os parâmetros $\mathbf{w}$ através das amostras do conjunto D e que o seu objetivo é obter

um modelo que aproxime aquela função geradora dos dados, evitando-se, assim, que $f(\mathbf{x}, \mathbf{w})$ se especialize no conjunto D somente, o qual pode ter algum viés que o afaste de $f_g(\mathbf{x})$.

1.6 Indução de Funções

O aprendizado de redes neurais artificiais pode ser representado pelo problema geral de indução de funções em que se deseja induzir o conjunto de parâmetros $\mathbf{w}$ da função genérica $f(\mathbf{x}, \mathbf{w})$ a partir de um conjunto de amostras D e de uma função-objetivo $J(y, f(\mathbf{x}, \mathbf{w}))$. Assumindo-se representatividade de D, este deverá conter informações sobre o comportamento de $f_g(\mathbf{x})$, os quais serão utilizadas para aproximá-la. Assim, espera-se que ao final do treinamento a função $f(\mathbf{x}, \mathbf{w})$ seja capaz de **imitar** o comportamento de $f_g(\mathbf{x})$ de tal forma que $f(\mathbf{x}, \mathbf{w}) \approx f_g(\mathbf{x})\ \forall \mathbf{x} \in R^n$. Espera-se que a interação entre $f(\mathbf{x}, \mathbf{w})$ e $f_g(\mathbf{x})$ por meio de representações do comportamento de $f_g(\mathbf{x})$ incorporadas ao conjunto de dados D, leve $f(\mathbf{x}, \mathbf{w})$ a se comportar como $f_g(\mathbf{x})$ em decorrência do ajuste de $\mathbf{w}$.

Uma outra forma de aproximar $f_g(\mathbf{x})$ é através da descoberta do operador que governa a função, no entanto, esta forma de modelagem é muitas vezes mais custosa, já que requer conhecimento sobre a estrutura dos processos que caracterizam $f_g(\mathbf{w})$. Quanto maior a complexidade da função geradora maior será o custo de encontrar um modelo que reproduza de maneira realística o seu operador. Assim, especialmente em problemas de maior complexidade, a reprodução do operador que governa $f_g(\mathbf{x})$ se torna inviável, restando como alternativa a indução da função aproximadora $f(\mathbf{x}, \mathbf{w})$ a partir do conjunto de amostras representativas D. Assim, a qualidade da aproximação resultante de $f(\mathbf{x}, \mathbf{w})$ dependerá essencialmente dos seguintes fatores:

- universo de funções candidatas;
- tamanho do conjunto de amostras;
- propriedades e características de D que serão utilizadas para representar o comportamento de $f_g(\mathbf{x})$;
- princípio de indução adotado;
- algoritmo que implementa o princípio de indução.
- capacidade da função-objetivo em representar o problema .

Para o caso particular de redes neurais artificiais, as funções candidatas são representadas pelo conjunto de modelos que podem ser obtidos a partir de uma estrutura de rede neural pré-definida, representada na forma geral como $f(\mathbf{x}, \mathbf{w})$. A capacidade aproximadora de um determinado conjunto de funções candidatas neurais será determinada pelo tipo de função de ativação utilizado, pelo número de camadas e pelo número de neurônios em cada camada. Uma vez definida a estrutura da rede neural, o universo de funções candidatas é determinado por todos os valores possíveis que podem assumir os elementos de $\mathbf{w}$. Assim, com base nas funções candidatas, espera-se que o conjunto de amostras seja grande e suficientemente representativo para a resolução do problema. O princípio de

indução, representado na forma de uma função-objetivo, determinará, então, o critério a ser adotado para selecionar uma das funções candidatas com base no conjunto de amostras que pode ser, por exemplo, a minimização do erro sobre o conjunto de amostras. Finalmente, a seleção do modelo será realizada pelo algoritmo que implementará o princípio de indução ou, em outras palavras, que resolverá o problema de otimização caracterizado pela função-objetivo. Algoritmos comuns são, por exemplo, Gradiente Descendente, Levenberg-Marquadt, Algoritmos Evolucionários, entre outros [NW06].

Sem perda de generalidade, considerando-se uma rede de duas camadas, o problema de indução de funções a partir de uma amostra finita de dados D, que caracteriza o aprendizado de RNAs, pode ser apresentado da seguinte forma:

> *Dado um conjunto de N amostras $D = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, encontre os parâmetros $\mathbf{Z}$ e $\mathbf{w}$ que minimizem a função de custo $J(D, f(\mathbf{x}, \mathbf{Z}, \mathbf{w}))$.*

Como resultado do ajuste dos parâmetros contidos em $\mathbf{Z}$ e $\mathbf{w}$ espera-se que a função $f(\mathbf{x}, \mathbf{Z}, \mathbf{w})$ se aproxime da função geradora dos dados $f_g(\mathbf{x})$ ou, em outras palavras, que esta seja induzida a partir da informação contida no conjunto de amostras $D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$. O treinamento de RNAs é tipicamente caracterizado pelo ajuste simultâneo de $\mathbf{Z}$ e $\mathbf{W}$. Outros modelos de máquinas de aprendizado, como as redes de Funções de Base Radiais (RBFs) [BL88], Máquinas de Aprendizado Extremo (ELMs) [HZS04] e as Máquinas de Vetores de Suporte (SVM) [CV95] tratam os problemas de indução de $\mathbf{Z}$ e $\mathbf{w}$ separadamente. Nestes modelos, o treinamento é realizado obtendo-se inicialmente $\mathbf{Z}$ para então, a partir das projeções na camada intermediária em função de $\mathbf{Z}$, obter-se $\mathbf{w}$.

1.6.1 Função da Camada Intermediária

O Teorema de Cover [Cov65] teve uma grande importância no desenvolvimento das RNAs e de outros modelos de máquinas de aprendizado, como as redes RBF [BL88] e as SVMs [BGV92]. Visando à resolução de problemas usualmente não-lineares, estes modelos buscam a solução do problema de aprendizado por meio de mapeamentos sucessivos até que o mesmo se torne tratável pela camada de saída. Uma analogia com a forma de construção de um polinômio $p(x)$ pode ajudar a entender melhor o papel do mapeamento em camadas. Considere, por exemplo, um polinômio de grau quatro, descrito de forma genérica como $p(x) = w_4x^4 + w_3x^3 + w_2x^2 + w_1x^1 + w_0x^0$. A construção do polinômio pode ser vista como se o seu argumento x fosse mapeado em um vetor de dimensão 5 por meio dos operadores x^4, x^3, x^2, x^1 e x^0. Assim, por exemplo, se $x_i = 2$ então o vetor obtido com o mapeamento é $\mathbf{h}_i = [16, 8, 4, 2, 1]^T$, o qual é combinado linearmente com o vetor de parâmetros $\mathbf{w} = [w_4, w_3, w_2, w_1, w_0]^T$ para obter a resposta $p(x)$ do polinômio. Na Figura 1.7 é apresentada uma representação esquemática da construção deste polinômio em uma estrutura de duas camadas. A camada de saída, ou seja, o somador, executa uma operação linear sobre a projeção não linear na camada intermediária, a qual é realizada por meio dos operadores x^4, x^3, x^2, x^1 e x^0. O resultado desta projeção não-linear é a linearização do problema, o que permite que a aproximação da função não-linear possa ser realizada com uma operação (linear) de soma pela camada de saída.

O problema de aproximação polinomial apresentado na Figura 1.7 é análogo ao problema geral de classificação não-linear de padrões e de aproximação de

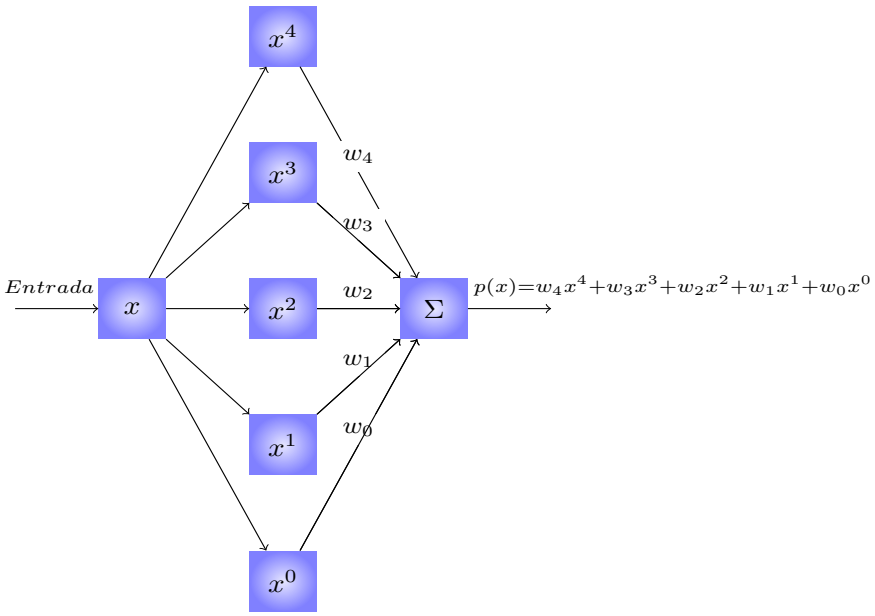


Figura 1.7: Representação esquemática da construção do polinômio $p(x) = w_4x^4 + w_3x^3 + w_2x^2 + w_1x^1 + w_0x^0$ na forma de uma estrutura em camadas. A camada intermediária tem por função a linearização do problema.

funções com RNAs, para os quais, da mesma forma, a solução do problema pode ser representada por uma estrutura em camadas. Assim, considere um modelo geral com uma camada intermediária representado pela função genérica $f(\mathbf{x}, \mathbf{Z}, \mathbf{w})$, conforme descrito nas seções anteriores. As duas funções $\Phi_h(\mathbf{x}, \mathbf{Z})$ e $\Phi_o(\mathbf{x}, \mathbf{w})$ que compõem $f(\mathbf{x}, \mathbf{Z}, \mathbf{w})$ são responsáveis pelos dois mapeamentos sucessivos que permitem que $f(\mathbf{x}, \mathbf{Z}, \mathbf{w})$ seja um mapeador universal de funções. A função $f(\mathbf{x}, \mathbf{Z}, \mathbf{w})$ é na verdade uma função composta, caracterizada pelas duas funções $\Phi_h(\mathbf{x}, \mathbf{Z})$ e $\Phi_o(\mathbf{x}, \mathbf{w})$, conforme mostrado de forma esquemática na Figura 1.8.

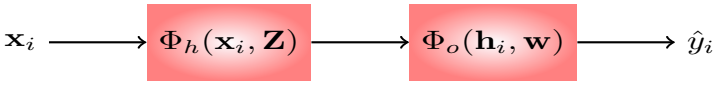


Figura 1.8: Representação esquemática do mapeamento em camadas em RNAs.

1.7 Reconhecimento de Padrões

O reconhecimento de padrões se refere à capacidade dos seres vivos ou, no nosso caso, dos modelos implementados por meio de programas de computador, em identificar e reconhecer a existência de regularidades em estímulos de entrada que sejam comuns a comportamentos de referência conhecidos. Podemos pensar, por exemplo, em padrões observáveis pelos nossos cinco sentidos em que os mais comuns são os padrões visuais, os auditivos e os olfatórios. Assim, o reco-

nhecimento de voz [WHH+89], o reconhecimento facial de uma pessoa [TP+91] ou de um odor conhecido [GO02] são exemplos de problemas de reconhecimento de padrões.

Para que um determinado padrão possa ser reconhecido ele precisa estar devidamente representado por meio de atributos (variáveis) que caraterizem de maneira coerente o problema de reconhecimento. O reconhecimento de faces, por exemplo, requer que as imagem faciais sejam representadas por meio de características, tais como textura, cor, distância entre os olhos, etc, que representem os elementos discriminantes do problema de reconhecimento. A forma de representação do problema é uma escolha de projeto. Assim, quando se escolhe a expressão gênica obtida através da técnica de *micro-array* [GST+99] para identificar uma determinada patologia, faz-se a opção por uma representação particular para o problema, a qual é mais apropriada, neste caso, para o tratamento por métodos computacionais. Uma outra abordagem para o mesmo problema é a utilização de exames clínicos, conhecimento médico e resultados de exames de laboratório para realizar o diagnóstico.

A escolha da forma de representação é essencial para o bom desempenho do reconhecedor. Assim, os seres vivos são seletivos e tendem a trabalhar com um conjunto restrito de características usualmente selecionadas de acordo com a situação. É comum uma pessoa ser reconhecida por uma característica marcante, como o nariz ou a cor dos olhos. Os seres humanos têm esta capacidade de escolher, de acordo com a situação, quais características possuem maior relevância para o julgamento. Os sistemas de reconhecimento automático, por sua vez, usualmente trabalham com um número fixo de características que são escolhidas em tempo de projeto, com base em conhecimento prévio sobre o problema. Na verdade, as características escolhidas fazem, de certa forma, parte do projeto do reconhecedor. Assim, quando, por exemplo, são escolhidas características geométricas, tais como dimensões da boca ou nariz, para o problema de reconhecimento de faces, estas características corresponderão a entradas que serão analisadas individual ou conjuntamente pelo reconhecedor e farão sempre parte do processo de julgamento [TP+91].

Dependendo da dimensão do problema, ou seja, do número de características utilizadas, é desejável, quando possível, trabalhar em dimensão menor. A redução da dimensionalidade pode tornar o problema mais tratável do ponto de vista computacional, além de poder levar a um melhor entendimento do mesmo. Para problemas que envolvem expressão gênica, por exemplo, a redução da dimensão por meio da seleção das características mais relevantes pode levar à identicação daqueles genes que estão associados à patologia em questão [CdPBNR11].

As técnicas de extração e seleção de caraterísticas antecedem, portanto, ao projeto do classificador propriamente dito e, na verdade, fazem parte do sistema de reconhecimento como um todo. Uma vez selecionadas as características que irão representar o problema, a próxima etapa no desenvolvimento de um sistema de reconhecimento de padrões envolve o projeto do reconhecedor (ou classificador). Este deve receber como entrada um conjunto de valores, usualmente armazenados na forma vetorial, relativos às variáveis selecionadas e gerar como resposta a classificação do vetor de entrada. A função executada pelo classificador pode ser representada de maneira geral $f(\mathbf{x}, \mathbf{w})$, cujo argumento é o vetor $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$, em que $x_1, x_2, \cdots, x_n$ são as características selecionadas e $\mathbf{w}$ o vetor que contém todos os parâmetros do classificador. Assim, para o argumento $\mathbf{x}_i$, considerando-se o vetor de parâmetros $\mathbf{w}$ fixo e determi-

nado em tempo de projeto, a função $f(\mathbf{x}_i, \mathbf{w}$ retornará o valor $\hat{y}_i$ que representa a classe á qual $\mathbf{x}_i$ foi associada pelo modelo. Uma representação esquemática das etapas envolvidas no projeto de um sistema de reconhecimento de padrões é apresentada na Figura 1.9.

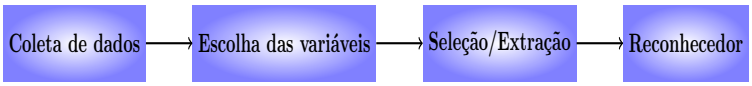


Figura 1.9: Etapas no projeto de um sistema de reconhecimento de padrões.

Um dos princípios que regem o projeto de um classificador é a existência de coerência espacial entre as amostras $\{\mathbf{x}_j\}_{j=1}^{N_k} \in C_k$, em que C_k é uma classe arbitrária. A existência de coerência espacial, ou seja, que padrões de mesma classe estejam próximos no espaço métrico R^n, é resultado da capacidade de as características $x_1, x_2, \cdots, x_n$ representarem de maneira fidedigna a coerência dos padrões que caracterizam a classe arbitrária C_k. Havendo coerência espacial, o projeto do reconhecedor envolve a divisão do espaço de entrada em regiões correspondentes a cada uma das classes características do problema. Um novo vetor de entrada será classificado de acordo com a classe da região de entrada em que o mesmo foi mapeado pelos valores das n características de entrada.

Referências Bibliográficas

[BGV92] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

[BL88] D. S. Broomhead and D. Lowe. Multivariable function interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[CdPBNR11] Frederico Coelho, Antônio de Pádua Braga, René Natowicz, and Roman Rouzier. Semi-supervised model applied to the prediction of the response to preoperative chemotherapy for breast cancer. *Soft Computing*, 15(6):1137–1144, 2011.

[Cov65] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 14:326–334, 1965.

[CV95] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–279, 1995.

[Cyb89] G. Cybenko. Approximation by superpositions of a sigmoid function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[Fra95] Stan Franklin. *Impossible Minds*. MIT Press, 1995.

[GO02] Ricardo Gutierrez-Osuna. Pattern analysis for machine olfaction: a review. *Sensors Journal, IEEE*, 2(3):189–202, 2002.

[GST+99] Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999.

[HZS04] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990 vol.2, july 2004.

[MP43] W.S. McCulloch and W. Pitts. A logical calulus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[NW06] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[TP+91] Matthew Turk, Alex P Pentland, et al. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[WHH+89] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(3):328–339, 1989.