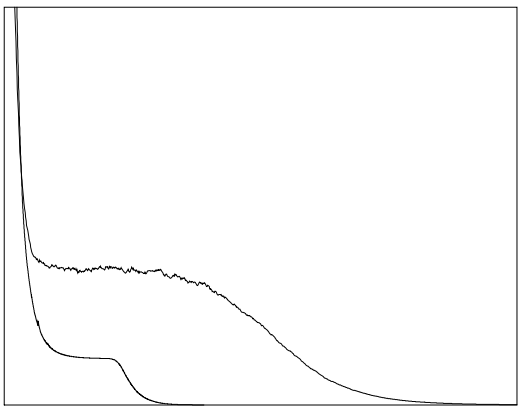
UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA

Otimização Funcional de Algoritmos em Redes Neurais Multicamada



Renato Vicente

SÃO PAULO
1997

FICHA CATALOGRÁFICA
Preparada pelo Serviço de Biblioteca e Informação
do Instituto de Física da Universidade de São Paulo

Vicente, Renato
Otimização Funcional de Algoritmos em Redes Neurais Multicamada. São Paulo, 1997.

Dissertação (Mestrado) Universidade de São Paulo. Instituto de Física - Departamento de Física Geral.

Área de Concentração: Física do Estado Sólido
Orientador: Prof. Dr. Nestor Felipe Caticha Alfonso

Unitermos: 1. Redes Neurais; 2. Aprendizado Online; 3. Generalização; 4. Backpropagation.

USP/IF/SBI - 38/97

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA

Otimização Funcional de Algoritmos em Redes Neurais Multicamada

Renato Vicente

Dissertação apresentada ao Instituto de Física da Universidade de São Paulo para obtenção do título de Mestre em Ciências

Orientador: Prof. Dr. Nestor Felipe Caticha Alfonso

Banca Examinadora :
Profa. Dra. Vera B. Henriques (IFUSP)
Prof. Dr. José Fernando Fontanari (IFSC-USP)
Prof. Dr. Nestor Felipe Caticha Alfonso (IFUSP)

SÃO PAULO
1997

À memória de minha mãe
Maria de Lourdes
e ao meu pai
José Vicente.

Agradecimentos

Ao Nestor Caticha pela amizade, pela orientação enriquecedora e pelas oportunidades.

Ao Osame pelas centenas de milhares de *memes*.

Ao Cristiano pelas aulas *rodrigueanas* de humanidade.

A Mauro, Rafael, Javier, Chiappin, Silvia e Josué pela amizade e por proporcionarem um ambiente estimulante.

Aos colegas de departamento Cláudio, Roberta, Suani, Kaline, Nestor, Marcos, Ana e Nelson por tornarem o ambiente de trabalho bastante agradável.

A minha irmã Soraya, meu cunhado Mario e minha sobrinha Mariana por me ensinarem boa parte do pouco que sei.

A Samira por, quase sem querer, me fazer crer um pouco mais.

Ao meu irmão Carlos pelo apoio.

A Salinas, Vera , Perez e Dreifus por me intoduzirem à Física Estatística.

A Fleming, Lyra e Becerra por tornarem a Física ainda mais interessante.

Aos meus amigos e colegas de Ciências Moleculares, Cristiane e Marcelo Sena pela amizade e paciência necessárias para me suportar durante anos.

A Marcelo, Luiz, José, Penov pelas alegrias multiplicadas e pelas tristezas divididas.

Ao Rafael por ter me agradecido duas vezes na sua dissertação.

Ao CNPq pelo apoio financeiro.

Abstract

In this work we extend the scheme of local functional optimization of online algorithms to fully connected soft committe architectures. In this scheme a variational argument is used to analitically build the modulation function that maximizes the average generalization error decay per example. We study realizable, non-realizable and over-realizable situations in the student-teacher scenario. We show simulations and analitical results. The performance of resulting optimized algorithms is compared with backpropagation algorithm performance. We found that the plateau phase in learning curves is vastly reduced or completely eliminated. We discuss the microscopic mechanisms involved in the optimized learning process and we point out further research directions.

Resumo

Neste trabalho estendemos o esquema de otimização local de algoritmos *online* às arquiteturas de rede tipo comitê *soft* totalmente conectado. Neste esquema um argumento variacional é utilizado para construir analiticamente a função modulação que maximiza o decaimento por exemplo do erro de generalização médio. Estudamos situações realizáveis, não-realizáveis e sobre-realizáveis no cenário professor-estudante. Comparamos as performances dos algoritmos otimizados resultantes e do algoritmo *backpropagation*. Mostramos que a fase de platô nas curvas de aprendizagem é vastamente reduzida ou completamente eliminada. Discutimos os mecanismos microscópicos envolvidos no aprendizado otimizado e sugerimos direções para futuros trabalhos.

Índice

Lista de Figuras

1

Introdução

Neste capítulo introduziremos os modelos de Redes Neurais. Faremos uma breve digressão histórica. Discutiremos algumas definições fundamentais e exporemos a organização desta dissertação.

1.1 Redes Neurais: Modelos Conexionistas

Desde os anos quarenta tem havido grande crescimento do interesse científico sobre questões envolvendo o processamento de informação. Este interesse é, em certa medida, conseqüência do conhecimento adquirido nos últimos anos sobre os mecanismos microscópicos e mesoscópicos presentes nos sistemas biológicos. Tem sido demonstrado que o processamento de informação nos sistemas naturais tem papel extremamente fundamental, ocorrendo em todos os níveis: das redes de interação bioquímica até as redes de neurônios. Por trás da impressionante complexidade dos sistemas biológicos que processam informação há uma série de aspectos comuns que nos saltam à vista:

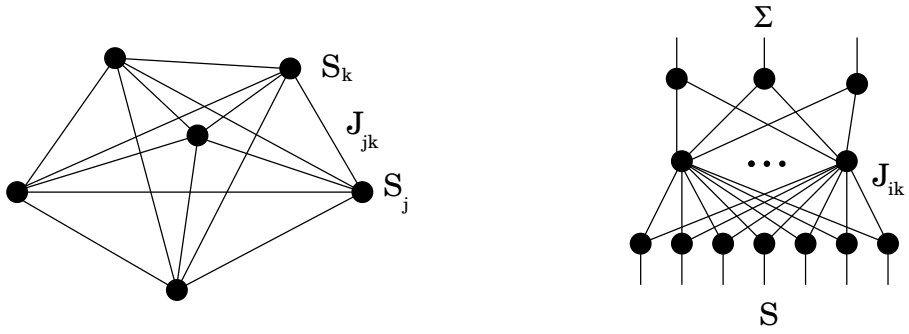
- Estes sistemas são compostos por muitas subunidades interagentes.
- O processamento é distribuído pelas unidades.
- O comportamento coletivo tem capacidade adaptativa.

O tipo de processamento que esses sistemas realizam é muito diferente daquele realizado por computadores tradicionais. Nestes há um algoritmo capaz de implementar um dado "conceito" (ou "regra") definido pelo mapeamento $\mathbf{\Sigma} = f(\mathbf{S})$. Assim, tudo que se tem a fazer é introduzir este algoritmo, na forma de um programa, em um computador e realizar o processamento sempre que necessário executando o programa. Certamente este não é o tipo de processamento que comumente chamaríamos de "inteligente" ou " cognitivo". No tipo de processamento envolvido nas situações biológicas o conceito não é conhecido a *priori*, mas precisa ser aprendido a partir de exemplos normalmente não muito precisos. Os modelos conexionistas, normalmente denominados *Redes Neurais Artificiais*, tentam reproduzir aspectos do processamento de informação em sistemas naturais fazendo uso de modelos simplificados que imitem os aspectos considerados fundamentais dos sistemas biológicos

com capacidade cognitiva. O intento das pesquisas em *Redes Neurais* é, portanto, produzir descrições à maneira da Física daquilo que seria a microestrutura da cognição.

Seguindo estas idéias, diversos modelos tem sido propostos e estudados utilizando diversas técnicas. Estes modelos de redes neurais são definidos como conjuntos de neurônios simplificados conectados entre si via sinápses J_{ik}, o estado de ativação destes neurônios é definido por S_i e é determinado por $S_i = g(\sum_k J_{ik}S_k)$, onde $g(.)$ é denominada função de transferência. Em particular, a comunidade de Física Estatística possui ferramental apropriado e boas analogias para a análise do comportamento coletivo destes modelos. Entre as arquiteturas de *Redes Neurais* de grande interesse físico poderíamos destacar as *Redes Neurais de Atratores*, que são totalmente análogas aos *Vidros de Spin* [18], e as arquiteturas multicamadas *feedforward* (Fig.1.1), que não são tão diretamente análogas a algum sistema físico conhecido, mas para as quais a aplicação de técnicas conhecidas é possível. Adicionalmente, estas redes multicamadas têm um interesse especial por possibilitarem a introdução de modelos para o aprendizado indutivo suficientemente ricos . Demonstrou-se em [17] que uma rede multicamada *feedforward* com entradas totalmente conectadas (fig. 1.2) e apenas uma camada interna com K neurônios é capaz de representar qualquer mapeamento $\mathbf{\Sigma} = f(\mathbf{S})$ entre as entradas e saídas da rede, desde que K seja suficientemente grande. Este teorema mostra a abrangência do cenário de aprendizado supervisionado (ou professor-aluno), onde uma rede multicamada "professor" com M neurônios na camada interna e sinápses $\mathbf{B}_n$ gera pares entrada-saída, eventualmente ruidosos, que servirão como exemplos $\mathcal{L} = \{(\mathbf{S}^\mu, \tilde{\mathbf{\Sigma}}^\mu) : \mu = 1, ..., p\}$ para que uma segunda rede "aluno" com K neurônios na camada interna possa tentar inferir qual o "conceito" (ou "regra") subjacente. A inferência neste tipo de modelo consistirá em modificar as sinápses $\mathbf{J}_k$, com base nos exemplos, no sentido de representar as sinápses do professor da melhor maneira possível. Quanto melhor for a representação que o aluno faz dos exemplos produzidos pelo professor maior será a sua capacidade de processar consistentemente ao professor uma nova entrada. Esta capacidade podemos denominar *capacidade de generalização*.

Na maneira como o conjunto de exemplos $\mathcal{L}$ é utilizado está implícita uma divisão entre a memorização dos exemplos e a atualização das sinápses. Num extremo temos a possibilidade de memorizar a lista $\mathcal{L}$ de exemplos e utilizá-los em paralelo na atualização sináptica, esta dinâmica é conhecida como *aprendizado offline*. No outro temos a utilização imediata de cada exemplo sem necessidade de memorizar a lista $\mathcal{L}$, conhecida como *aprendizado online*. Entre estes dois extremos há todo um espectro de situções intermediárias. Certamente os sistemas biológicos devem situar-se entre estes extremos, no entanto, o imperativo da viabilidade teórica nos obriga a, ao menos inicialmente, nos concentrar nos casos extremos. Apesar da relevância biológica ser questão importante, do ponto de vista das aplicações o aprendizado *online* apresenta vantagens devido a sua necessidade mínima de memorização. Isto justifica seu estudo detalhado. Nesta dissertação nos ocuparemos do aprendizado *online* no cenário professor-aluno envolvendo arquiteturas multicamada.



Rede de Atratores **Rede Feedforward**

Figura 1.1: Modelos conexionistas

1.2 Breve Histórico

As primeiras idéias que levaram aos modelos de Redes Neurais (RN) surgiram na década de 40, em especial nos trabalhos de McCulloch e Pitts [34] e Hebb [23]. Na década de 60 estas idéias receberam vários desenvolvimentos culminando com os trabalhos de Rosenblatt [41], que propôs o Percéptron (Fig. 1.3) e de Minsky e Papert [35] que explicitaram suas limitações. Inicialmente as pesquisas em RNs estavam vinculadas à produção de modelos mesoscópicos para as funções cerebrais, no entanto, seu desenvolvimento ficou restrito às aplicações tecnológicas até o início da década de 80 quando voltaram a ganhar destaque na literatura de Física graças ao trabalho de Hopfield [26] que relacionou as RNs conhecidas como redes de atratores (Fig.1.1) aos vidros de spin [18]. A proposta de Hopfield envolvia um tipo de modelo composto por neurônios com dois estados $S_i \in \{+1, -1\}$ interagindo uns com os outros de maneira simétrica através de acoplamentos sinápticos J_{ik} modulados segundo a proposta de Hebb. Estes primeiros estudos restringiram-se basicamente ao fenômeno da memorização associativa de padrões nestas redes. Após este período o propósito das pesquisas em RNs em Física passa a ser menos o de servir como modelos mesoscópicos para funções cerebrais e mais o de introduzir no repertório da Física nova fenomenologia e nova linguagem. Uma revisão bastante completa deste período pode ser encontrada em [4].

No final da década de 80 o trabalho de Gardner e Derrida [20] deslocou o interesse de estudo para arquiteturas com acoplamentos unidirecionais e organizadas em camadas, conhecidas como redes *feedforward*. A dificuldade de aplicar as técnicas da Física Estatística existentes na época foi a principal responsável pelo adiamento dos estudos deste tipo de arquitetura já largamente utilizado em aplicações. A idéia de Gardner consistiu em estudar as propriedades estatísticas do espaço dos acoplamentos sinápticos J_{ik} ao invés dos tradicionais estudos sobre o espaço das configurações

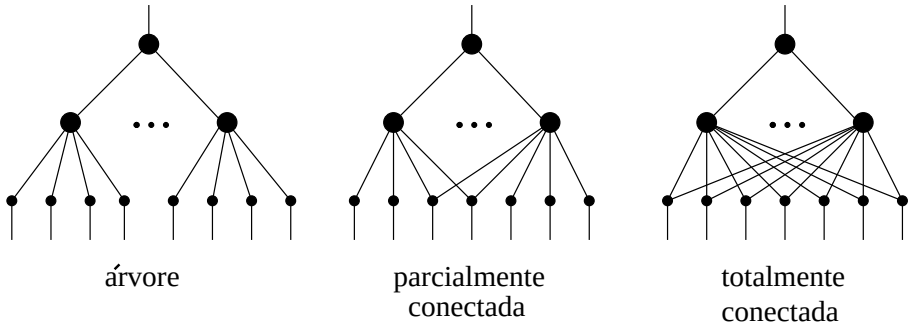


Figura 1.2: Arquiteturas *feedforward*

de atividade S_k. Esta idéia simples e seminal possibilitou o estudo analítico de diversas situações novas. Entre estas novas situações ganhou destaque o problema da generalização no cenário professor-aluno. Proposto em 1989 independentemente por Vallet [52] e por Gardner e Derrida [21], o problema da generalização consiste em fazer com que uma RN infira os acoplamentos de outra RN a partir de exemplos. A maneira como estes exemplos são apresentados, se em paralelo (*offline*) ou um por vez (*online*), define a dinâmica de aprendizagem. Até os primeiros anos da década de 90, a ênfase estava voltada à dinâmica de aprendizagem *offline* seguindo a tradição da Física Estatística do equilíbrio. Há várias revisões deste período, entre elas podemos citar [24, 48, 56, 37].

Apesar desta ênfase inicial da comunidade de Física, as redes *feedforward* com muitas camadas aprendendo com algoritmos *online* sempre estiveram presentes nas aplicações de engenharia. Os representantes mais conhecidos destes algoritmos são os algoritmos estocásticos de gradiente, comumente denominados *backpropagation* [1, 42]. O interesse da comunidade de Física na dinâmica *online* intensificou-se a partir de 1993 com a percepção de que estes modelos combinam simplicidade de tratamento, importância teórica e prática.

Em 1992 Kinouchi e Caticha [29] propuseram um método construtivo para obtenção de algoritmos *online* com generalização ótima utilizando técnicas funcionais. Até este trabalho, a análise do problema da generalização restringia-se à caracterização de algoritmos sugeridos heuristicamente. Esta técnica recebeu vários desenvolvimentos, foi aplicada a várias arquiteturas e estendida ao aprendizado *offline* [53]. O interesse nesta técnica tem, desde então, aumentado visando a possibilidade de deduzir quais as características que um bom algoritmo prático deve ter. Dentre as situações de interesse prático imediato estariam aquelas para as quais os algoritmos *backpropagation* são amplamente empregados. Em 1994, Biehl e Schwarze [7] obtiveram a primeira solução analítica para a dinâmica de aprendizado do algoritmo *backpropagation*. Logo em seguida, Saad e Solla [45] obtiveram uma generalização que permitiu a solução analítica para uma classe suficientemente ampla de arqui-

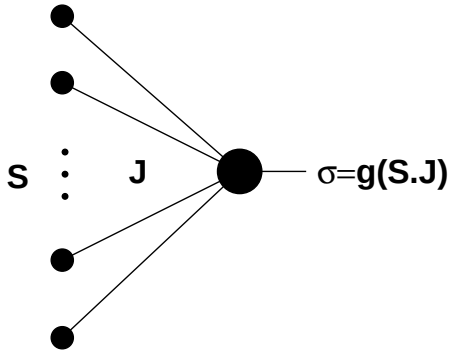


Figura 1.3: O Percéptron

teturas. Estes trabalhos abriram caminho para a extensão do método funcional a novas situações envolvendo arquiteturas mais complexas. Esta dissertação tratará desta extensão.

1.3 Memorização, Categorização e Generalização

É interessante, neste ponto, discutirmos as diferenças entre problemas freqüentemente estudados na literatura de Redes Neurais: a memorização, a categorização e a generalização.

O primeiro problema a receber tratamento na literatura atual (a partir de Hopfield) das RNs foi a memorização [4]. Na memorização apresenta-se um conjunto de padrões $\mathcal{L}$, que deverão ser representados pelas sinápses da rede na forma de atratores dinâmicos. O desempenho na memorização é então medido pela similaridade entre os padroẽs armazenados e os padrões apresentados. A capacidade de memorização é definida como a quantidade máxima de padrões que uma rede consegue armazenar com certo índice de erro.

O problema da categorização [19], consiste em inferir um conjunto de padrões $\mathcal{L}$ a partir de exemplos amostrados aleatoriamente de uma vizinhança controlada destes padrões. O problema da categorização pode ser pensado como o problema da memorização mas para padrões corrompidos por certa quantidade de ruído .

Já o problema da generalização consiste em inferir a configuração sináptica de uma suposta rede professor que estaria gerando o conjunto $\mathcal{L}$ de exemplos a partir apenas deste conjunto de exemplos. Assim, a medida de interesse neste caso é o erro médio sobre exemplos amostrados aleatoriamente com uma distribuição definida pelo ambiente de teste (normamente tomada como sendo uniforme).

Durante toda esta dissertação estaremos interessados somente no problema da generalização ao qual também chamaremos de problema do aprendizado.

1.4 Medidas de Desempenho

A maneira como medimos o desempenho de uma rede numa tarefa de aprendizado é de grande importância no nosso estudo. Nesta seção iremos colocar de maneira precisa a grandeza que medirá o desempenho em nosso tratamento: o erro de generalização. Quando analisamos o problema da generalização em redes neurais podemos discernir dois "ambientes" de operação: o "ambiente de treinamento" e o "ambiente de teste". Estes ambientes são descritos, para o caso de exemplos sem correlasção temporal, por distribuições de probabilidade no espaço dos pares ordenados entrada-saída $P(\mathbf{S}, \Sigma_B)$[1]. Assim, teremos $P_{\mathcal{L}}(\mathbf{S}, \Sigma_B)$ indicando o "ambiente de treinamento " e $P_{\mathcal{T}}(\mathbf{S}, \Sigma_B)$ indicando o "ambiente de teste" (seguindo [32]). As propostas de medidas de desempenho devem refletir a eficácia das redes em cada ambiente. Assim, define-se inicialmente a métrica de erro que irá indicar a discordância [2] que há entre o professor e o aluno $e(\Sigma_J, \Sigma_B)$. A partir daí usualmente teremos :

- Erro de generalização:

  É o erro esperado sobre exemplos extraídos de uma distribuição de teste $P_{\mathcal{T}}(\mathbf{S}, \Sigma_B)$:

  $$e_g(\mathbf{J}, \mathbf{S}) = \int d\mathbf{S} d\Sigma_B \; e(\Sigma_J, \Sigma_B) P_{\mathcal{T}}(\mathbf{S}, \Sigma_B). \tag{1.1}$$

  Geralmente quando nada sabemos a respeito do ambiente de teste escolhemos uma distribuição uniforme com as componentes S_k independentes (que é a escolha *a priori* com máxima entropia [11]) e com saídas Σ_B livres de ruído.

- Erro de predição:

  É o valor esperado do erro num ambiente de teste idêntico ao de treinamento $P_{\mathcal{L}}(\mathbf{S}, \Sigma_B)$:

  $$e_p(\mathbf{J}, \mathbf{S}) = \int d\mathbf{S} d\Sigma_B \; e(\Sigma_J, \Sigma_B) P_{\mathcal{L}}(\mathbf{S}, \Sigma_B). \tag{1.2}$$

- Erro de memorização:

  Pode ser pensado como uma estimativa empírica do erro de predição.

  $$e_m(\mathbf{J}, \mathbf{S}) = \frac{1}{p} \sum_{\mathbf{S}^\mu \in \mathcal{L}(p)} e(\Sigma_J^\mu, \Sigma_B^\mu). \tag{1.3}$$

  Com $(\mathbf{S}, \Sigma_B)$ tomados com distribuição $P_{\mathcal{L}}(\mathbf{S}, \Sigma_B)$. Aqui esperamos que $e_m \overset{p \to \infty}{\to} e_p$.

[1] No caso de seqüências correlacionadas, por distribuições no espaço de pares $\mathcal{L}(p)$. Aqui Σ_B representa a saída que pode eventualmente ter sido corrompida por ruído.

[2] Por exemplo $e(\Sigma_J, \Sigma_B) = \frac{1}{2}(\Sigma_B - \Sigma_J)^2$ para arquiteturas com saída contínua e $e(\Sigma_J, \Sigma_B) = \delta(\Sigma_J, \Sigma_B)$, onde $\delta(.,.)$ é a função delta de Kronecker, para arquiteturas com saída booleana.

Normalmente estaremos interessados em comportamentos típicos, ou seja, independentes de $\mathbf{J}$ e $\mathbf{B}$, dessa forma utilizaremos :

$$e_g = \left\langle \langle e_g(\mathbf{J}, \mathbf{B}) \rangle_{\mathbf{J}|\mathcal{L}(p)} \right\rangle_{\mathcal{L}(p)|\mathbf{B}} .$$

Onde $\mathcal{L}(p)$ denota o conjunto de pares ordenados entrada-saída $(\mathbf{S}, \Sigma_B)$ de cardinalidade "p" utilizado no treinamento. No cálculo da média acima deve ser observado que $\mathbf{J}$ é gerado por um algoritmo específico a partir de uma seqüência específica de exemplos $\mathcal{L}$.

1.5 Organização desta Dissertação

O interesse central desta dissertação é mostrar como o esquema funcional de otimização de algoritmos pode ser estendido às arquiteturas multicamada, como o desempenho destes algoritmos se compara aos algoritmos usuais e quais os problemas técnicos que podem surgir.

No capítulo 2 fazemos uma revisão dos resultados existentes sobre algoritmos *backpropagation* e suas variantes, tentando nos concentrar nas situações e aspectos que serão comparados aos algoritmos otimizados.

No capítulo 3 expomos o esquema de otimização funcional de algoritmos *online* de uma maneira mais geral e apropriada para a aplicação ao tipo de redes que estamos interessados. Procuramos exemplificar sua aplicação no percéptron reobtendo resultados já bem conhecidos.

No capítulo 4 aplicamos o esquema funcional a situações inéditas envolvendo redes multicamada.

Finalmente, no capítulo 5 resumimos os principais resultados obtidos e discutimos algumas direções para novos estudos.

Para uma leitura rápida sugerimos apenas os capítulos 3 e 4 e a seção 5.1.

2

Backpropagation

Neste capítulo revisaremos as técnicas de estudo da dinâmica online, faremos um sumário da aplicação destas técnicas a situações envolvendo o algoritmo backpropagation e sumarizaremos brevemente alguns desenvolvimentos analíticos recentes.

2.1 O Algoritmo Backpropagation

Backpropagation é o nome dado às implementações de algoritmos de gradiente em redes neurais. Estes algoritmos têm uma longa história de descobertas e redescobertas que tem início na década de quarenta, com os primeiros trabalhos em redes neurais e se estende até a década de sessenta [1, 58]. Na década de setenta há menor entusiasmo, já na década de oitenta ocorre uma redescoberta [43]. Atualmente o algoritmo *backpropagation* tem sido amplamente utilizado em situações práticas tão variadas como o problema XOR ou o reconhecimento de caracteres escritos manualmente [24, 42]. A idéia básica por trás do *backpropagation* é minimizar uma função energia E definida sobre os exemplos apresentados $\mathcal{L}$ e sobre os pesos sinápticos $\mathbf{J}$. As sinápses evoluem com uma dinâmica na forma:

$$\Delta \mathbf{J} = -\eta \nabla_{\mathbf{J}} E(\mathbf{J}; \mathcal{L}) \tag{2.1}$$

Aqui η é um parâmetro (comumente denominado *taxa de aprendizagem*) que controla o tamanho das modificações sinápticas a cada passo, o operador $\nabla_{\mathbf{J}}$ indica o gradiente no espaço dos vetores sinápticos $\mathbf{J}$ e o conjunto de exemplos $\mathcal{L}$ pode ser visto , por analogia com os sistemas desordenados, como uma desordem *quenched*. O fato é que o conjunto $\mathcal{L}$ de exemplos é gerado de forma aleatória e a energia é então minimizada com este conjunto mantido fixo. Esta minimização pode ser tanto realizada de maneira *online* (E utiliza apenas um exemplo por vez) quanto de maneira *offline* (E utiliza todo o conjunto de exemplos $\mathcal{L}$). Quando a dinâmica implementada é *offline* a não-linearidade de E lhe confere um relevo altamente rugoso(cheio de estados metaestáveis) tornando difícil a convergência da dinâmica (2.1) ao(s) mínimo(s) global(is). Uma maneira usual de superar os estados metaestáveis é a introdução de um termo apropriado [1] de ruído transformando (2.1) numa equação

[1] Ruído $\epsilon_k(t)$ com $\langle \epsilon_k(t)\epsilon_j(\tilde{t})\rangle = 2T\delta_{kj}\delta(t-\tilde{t})$.

de Langevin com distribuição de equilíbrio de Gibbs, com temperatura proporcional à amplitude do ruído. A minimização da energia E é garantida pela diminuição progressiva desta temperatura, num processo conhecido como *annealing* simulado. Já na dinâmica *online* a energia E depende apenas de um exemplo, e estes exemplos são coletados do conjunto $\mathcal{L}$ aleatoriamente. Neste caso $E(\mathbf{J};\mathbf{S})$ é uma função de argumento aleatório com valor médio que tende a um valor mínimo à medida que o sistema se aproxima de um mínimo global, o que evita os estados metaestáveis. Este processo tem sido denominado *self-annealing* [25].

A característica mais marcante do *backpropagation* é sua generalidade e que permite sua aplicação a virtualmente qualquer arquitetura *feedforward*, desde que as unidades integrantes tenham função de transferência derivável pelo menos uma vez. A aplicação deste algoritmo numa situação geral pode ser vista no (Apêndice A). Esta generalidade torna interessante o estudo analítico do algoritmo. Na próxima seção introduziremos um certo número de técnicas que tornarão possíveis estes estudos analíticos para redes com grande número de entradas.

2.2 Dinâmica Online de Aprendizado no Limite Termodinâmico

Nesta seção trataremos com maior profundidade as equações que descrevem a dinâmica *online* de aprendizado para arquiteturas multicamada no limite de infinitas entradas (ou termodinâmico).

Em redes com número finito (N) de entradas a dinâmica *online* tem a forma:

$$J_{ik}(\mu+1) = \left(1 - \frac{1}{N}\Omega_k(\mathcal{V})\right) J_{ik}(\mu) + \frac{1}{N}F_k(\mathcal{V})S_i(\mu) \tag{2.2}$$

Aqui o índice i identifica cada sinápse, o índice k identifica cada neurônio, F_k são denominadas funções modulação e $\mathcal{V}$ ($\mathcal{V}$isível) é o conjunto de grandezas acessíveis às funções F_k. O sistema de equações (2.2) é estocástico, pois as questões-exemplo $\mathbf{S}(\mu)$ são geradas de forma estocástica com distribuição $P_{\mathcal{L}}(\mathbf{S})$ e têm enorme número de graus de liberdade. Esta alta dimensionalidade pode ser reduzida utilizando um procedimento muito comum em Mecânica Estatística : descrever o sistema em termos de certos *parâmetros de ordem* [18]. Estes *parâmetros de ordem* devem ser "estatisticamente robustos ", ou seja, suas flutuações estocásticas devem ser cada vez menores para sistemas cada vez maiores. Esta propriedade é comumente denominada auto-mediância [2].

A escolha dos parâmetros de ordem adequados depende das propriedades das distribuições de probabilidade envolvidas, a saber, da distribuição *a priori* dos pro-

[2] Justificaremos a auto-mediância numa situação de aprendizado on-line em redes neurais no (Apêndice B) seguindo a linha de [40]. No texto nos limitaremos a mostrar numericamente em um caso simples como a propriedade de auto-mediância se apresenta.

fessores e da distribuição dos exemplos. No caso das redes neurais aprendendo a partir de exemplos gerados uniformemente, estes serão:

- correlação entre neurônios do aluno:

$$Q_{kj} \equiv \sum_{i=1}^{N} J_{ik} J_{ij}.$$

- correlação entre neurônios do professor:

$$M_{nm} \equiv \sum_{i=1}^{N} B_{in} B_{im}.$$

- correlação entre neurônios do aluno e do professor:

$$R_{kn} \equiv \sum_{i=1}^{N} J_{ik} B_{in}.$$

Introduzindo os *campos pós-sinápticos* :

$$h_k = J_k \cdot S,$$

$$b_n = B_n \cdot S$$

podemos verificar melhor o significado das variáveis macroscópicas acima. Calculemos então, para exemplificar, a média $\langle h_k b_n \rangle$:

$$\begin{aligned} \langle h_k b_n \rangle &= \int d\mathbf{S}\, P(\mathbf{S}) \sum_{il} J_{ik} B_{ln} S_i S_l \\ &= \sum_{il} J_{ik} B_{ln} \langle S_i S_l \rangle. \end{aligned} \tag{2.3}$$

Se considerarmos que a distribuição de questões $P(\mathbf{S})$ é tal que [3] $\langle S_i S_l \rangle = \delta_{il}$ teremos que :

$$\langle h_k b_n \rangle = \mathbf{J}_k \cdot \mathbf{B}_n = R_{kn}. \tag{2.4}$$

Os parâmetros macroscópicos representam as correlações entre os campos pós-sinápticos.

Se as normas dos vetores sinápticos não forem relevantes, é bem menos redundante usarmos :

- correlação normalizada entre neurônios do aluno e do professor[4]:

$$\rho_{kn} = \frac{R_{kn}}{\sqrt{Q_{kk} M_{nn}}} \equiv \frac{\sum_{i=1}^{N} J_{ik} B_{in}}{\|B_n\| \, \|J_k\|}.$$

[3] Isto pode ser produzido gerando vetores com componentes aleatórias e independentes com $\langle S_i^2 \rangle$.

[4] A norma $\|X\|$ é definida pelo produto escalar $X \cdot Y \equiv \sum_{i=1}^{N} X_i Y_i$, $\|X\| = \sqrt{X \cdot X}$.

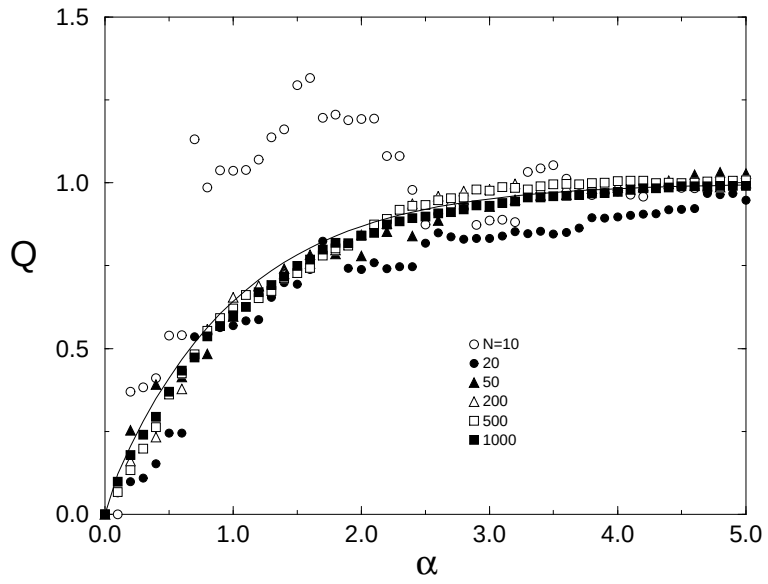


Figura 2.1: Auto-mediância do parâmetro Q num percéptron linear.

- correlação normalizada entre neurônios do aluno:

$$q_{jk} = \frac{Q_{jk}}{\sqrt{Q_{jj}Q_{kk}}} \equiv \frac{\sum_{i=1}^{N} J_{ij}\, J_{ik}}{\|J_j\|\,\|J_k\|}.$$

Para exemplificar a propriedade de auto-mediância numa situação prática, exibimos na (Fig. 2.1) corridas da dinâmica de aprendizado ótimo num percéptron linear para vários tamanhos do sistema. Para cada tamanho está representada apenas uma única corrida. A curva cheia corresponde à solução das equações determinísticas para os parâmetros de ordem obtidas sob a hipótese de auto-mediância. É bastante claro na (Fig.2.1) que as flutuações estocásticas no parâmetro tornam-se cada vez menores conforme o sistema aumenta.

As equações estocásticas para os *parâmetros de ordem* serão, utilizando (2.2), para Q :

$$\begin{aligned} Q_{kj}(\mu+1) &= J_k(\mu+1)\cdot J_j(\mu+1) \\ &= \left(1-\frac{1}{N}\Omega_k(\mathcal{V})-\frac{1}{N}\Omega_j(\mathcal{V})\right)(J_k(\mu)\cdot J_j(\mu)) + \\ &+ \frac{1}{N}(J_k(\mu)\cdot S(\mu))F_j(\mathcal{V}) + \frac{1}{N}(J_j(\mu)\cdot S(\mu))F_k(\mathcal{V}) + \end{aligned}$$

$$+ \quad \frac{1}{N^2}F_k(\mathcal{V})F_j(\mathcal{V})(S(\mu)\cdot S(\mu)) \; + \; \mathcal{O}(\frac{1}{N^2}) \tag{2.5}$$

Para R :

$$\begin{aligned} R_{kn}(\mu+1) &= J_k(\mu+1)\cdot B_n(\mu+1) \\ &= \left(1-\frac{1}{N}\Omega_k(\mathcal{V})\right)(J_k(\mu)\cdot B_n) + \frac{1}{N}F_k(\mathcal{V})(S(\mu)\cdot B_n) \\ &= \left(1-\frac{1}{N}\Omega_k(\mathcal{V})\right)R_{kn}(\mu) + \frac{1}{N}F_k(\mathcal{V})b_n(\mu). \end{aligned} \tag{2.6}$$

Para escrevermos a equação para ρ precisamos primeiro escrever :

$$\begin{aligned} \frac{1}{\sqrt{Q_{kk}(\mu+1)}} &= \frac{1}{\sqrt{Q_{kk}(\mu)}}[1 \; + \; \frac{1}{N}\Omega_k(\mathcal{V}) \; - \; \frac{1}{N}\frac{h_k(\mu)F_k(\mathcal{V})}{Q_{kk}(\mu)} \\ &- \quad \frac{1}{2N}\frac{{F_k}^2(\mathcal{V})}{Q_{kk}(\mu)} \; + \; \mathcal{O}(\frac{1}{N^2})]. \end{aligned} \tag{2.7}$$

Multiplicando (2.6) por (2.7) e colhendo termos até $\mathcal{O}(\frac{1}{N})$ obteremos as equações para os ρs:

$$\begin{aligned} \rho_{kn}(\mu+1) &= \rho_{kn}(\mu) + \frac{1}{N}\frac{b_n(\mu)F_k(\mathcal{V})}{\sqrt{Q_{kk}(\mu)M_{nn}}} \\ &- \quad \frac{1}{N}\left[\frac{h_k(\mu)F_k(\mathcal{V})}{Q_{kk}} + \frac{{F_k}^2(\mathcal{V})}{2Q_{kk}}\right]\rho_{kn}(\mu) + \mathcal{O}(\frac{1}{N^2}) \end{aligned} \tag{2.8}$$

Analogamente multiplicando (2.7) por (2.5) obteremos equações para os qs:

$$\begin{aligned} q_{jk}(\mu+1) &= q_{jk}(\mu) + \frac{1}{N}\frac{h_k(\mu)F_j(\mathcal{V}) + h_j(\mu)F_k(\mathcal{V}) + F_j(\mathcal{V})F_k(\mathcal{V})}{\sqrt{Q_{kk}(\mu)Q_{jj}(\mu)}} \\ &- \quad \frac{1}{N}\left[\frac{h_j(\mu)F_k(\mathcal{V})}{Q_{jj}} + \frac{{F_k}^2(\mathcal{V})}{2Q_{kk}}\right. \\ &+ \quad \left.\frac{h_k(\mu)F_j(\mathcal{V})}{Q_{kk}} + \frac{{F_j}^2(\mathcal{V})}{2Q_{jj}}\right]q_{jk}(\mu) + \mathcal{O}(\frac{1}{N^2}) \end{aligned} \tag{2.9}$$

No limite de sistemas grandes, $N \to \infty$, podemos transformar estas equações de diferenças em equações diferenciais introduzindo uma escala de tempo $\Delta\alpha = \frac{1}{N}$. É importante perceber que as derivadas $\frac{d\rho_{kn}}{d\alpha}$ e $\frac{dQ_{kj}}{d\alpha}$ não são grandezas auto-mediantes como mostramos na (Fig. 2.2) para o caso do percéptron linear aprendendo otimamente. Nesta figura exibimos as flutuações de corridas do algoritmo ótimo para

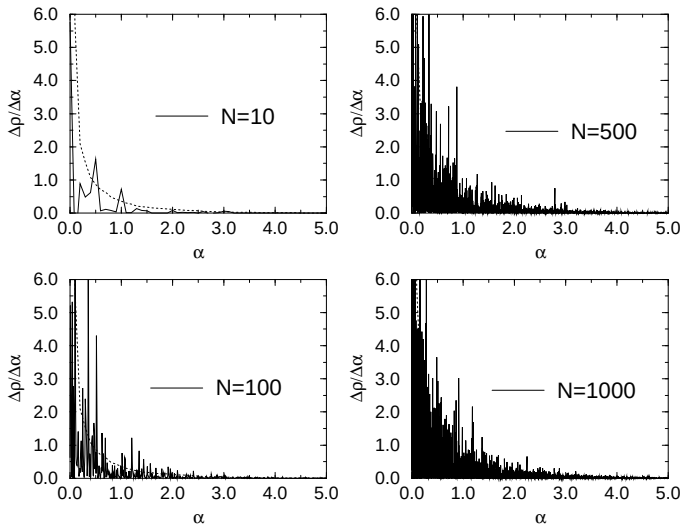


Figura 2.2: No percéptron linear, as derivadas não são auto-mediantes. Mostramos corridas para vários tamanhos. A curva tracejada representa o cálculo analítico da média sobre infinitas corridas.

$N = 10, 100, 500$ e 1000. O tamanho típico destas flutuações independe do tamanho N do sistema[5]. Desta forma, se quisermos escrever equações determinísticas para o sistema precisamos realizar médias sobre toda aleatoriedade envolvida no aprendizado, a saber: os exemplos e o ruído do sistema. A linha tracejada (que aparece apenas no gráfico para $N = 10$) representa a média sobre infinitas realizações da dinâmica (cálculo analítico).

No limite de redes grandes (ou termodinâmico) as equações adquirem a forma [6]:

$$\dot{Q}_{kj} = \langle lim_{N\to\infty} N\Delta Q_{kj} \rangle$$

[5] Uma discussão mais aprofundada pode ser encontrada em [33].

[6] Aqui estamos considerando o caso em que as seqüências de exemplos são descorrelacionadas e utilizamos :

$$\langle(\cdots)\rangle \equiv \int d\mathcal{H}d\mathcal{V}P(\mathcal{H},\mathcal{V})(\cdots)$$

$$\dot{(\cdots)} \equiv \frac{d(\cdots)}{d\alpha}$$

Onde $\mathcal{H}$ e $\mathcal{V}$ representam toda aleatoriedade do sistema. Seguindo a notação introduzida por M.Copelli e N.Caticha em [14], $\mathcal{H}$ corresponde ao conjunto de grandezas inacessíveis ($\mathcal{H}idden$) para as funções modulação e $\mathcal{V}$ corresponde ao conjunto de grandezas acessíveis($\mathcal{V}isible$).

$$
\begin{aligned}
&= \langle h_k F_j + h_j F_k + F_j F_k - \Omega_j Q_{jk} - \Omega_k Q_{jk} \rangle \\
\dot{R}_{kn} &= \langle lim_{N\to\infty} N \Delta R_{kn} \rangle \qquad (2.10) \\
&= \langle F_k b_n - \Omega_k R_{kn} \rangle \\
\dot{\rho}_{kn} &= \langle lim_{N\to\infty} N \Delta \rho_{kn} \rangle \\
&= \left\langle \frac{b_n F_k}{\sqrt{Q_{kk} M_{nn}}} - \frac{h_k F_k}{Q_{kk}} \rho_{kn} - \frac{F_k^2}{2Q_{kk}} \rho_{kn} \right\rangle \\
\dot{q}_{jk} &= \langle lim_{N\to\infty} N \Delta q_{jk} \rangle \\
&= \left\langle \frac{h_k F_j + h_j F_k + F_j F_k}{\sqrt{Q_{kk}(\mu) Q_{jj}(\mu)}} - \frac{h_j F_k}{Q_{jj}} q_{jk} + \frac{{F_k}^2}{2Q_{kk}} q_{jk} \right. \\
&+ \left. \frac{h_k F_j}{Q_{kk}} q_{jk} + \frac{{F_j}^2}{2Q_{jj}} q_{jk} \right\rangle \qquad (2.11)
\end{aligned}
$$

Como já mostramos na (Fig 2.1) as integrais das equações diferenciais estocásticas acima são auto-mediantes. Antes de efetuarmos as médias temos, por exemplo:

$$\dot{R}_{kn}(\alpha) = F_k(\alpha) b_n(\alpha) - \Omega_k(\alpha) R_{kn}(\alpha).$$

Onde a dependência em α indica na verdade uma realização possível da dinâmica estocástica. Ao integrarmos teremos:

$$R_{kn}(\alpha) = R_{kn}(0) + \int_0^\alpha d\alpha \, \dot{R}_{kn}(\alpha).$$

A auto-mediância significa neste caso que a integral acima independe da particular realização da dinâmica. Isto se deve a escolha apropriada da escala de tempo que fizemos. Desta forma não representa perda de generalidade alguma efetuar primeiro as médias, transformando as equações em determinísticas, pois suas integrais quando tomarmos o limite $N \to \infty$ serão, de qualquer forma, determinísticas.

2.3 Tratamento Analítico do Caso Geral

Os primeiros resultados analíticos para o aprendizado *backpropagation on-line* em redes multicamada foram obtidos em 95 por Biehl e Schwarze [7]. Neste trabalho são tratadas situações envolvendo percéptrons, máquinas comitê *soft* aprendendo percéptrons e máquinas paridade com unidades contínuas, todas com função de transferência do tipo $erf\left(\frac{x}{\sqrt{2}}\right)$. Posteriormente Riegler e Biehl trataram uma situação envolvendo apenas comitês *soft* [9] com $K = 2$ unidades na camada interna. Ainda em 95 Saad e Solla [45, 46, 47] obtiveram uma solução que abrange um amplo leque de situações envolvendo máquinas comitê *soft* com uma única camada interna, função de transferência tipo "função erro" e número de unidades na camada escondida com $K/N = \mathcal{O}(1/N)$. Pode ser demonstrado que qualquer função contínua pode ser representada por redes comitê com uma camada escondida, desde que o

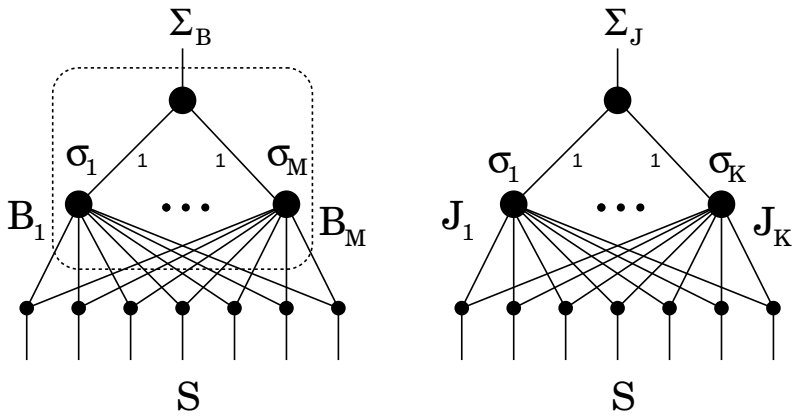


Figura 2.3: Professor = comitê *soft* com M neurônios na camada interna. Aluno = comitê *soft* com K neurônios na camada interna.

número de unidades seja suficientemente grande [17, 27], o que torna a o estudo destas redes particularmente interessante. Nesta seção apresentamos em detalhe a solução proposta por Saad e Solla.

No que segue estaremos interessados na situação em que uma rede neural "aluno" de arquitetura comitê totalmente conectado *soft* com K unidades na camada interna aprende de maneira *online* um "professor" de arquitetura idêntica a menos do número M de unidades na camada interna. Na (Fig. 2.3) está representada esta situação, temos que a saída do professor é definida por:

$$\Sigma_B = \sum_{m=1}^{M} \sigma_m \tag{2.12}$$

$$= \sum_{m=1}^{M} g\left(b_m\right) \tag{2.13}$$

$$= \sum_{m=1}^{M} erf\left(\frac{b_m}{\sqrt{2}}\right) \tag{2.14}$$

$$= \sum_{m=1}^{M} erf\left(\frac{\mathbf{B}_m \cdot \mathbf{S}}{\sqrt{2}}\right). \tag{2.15}$$

Aqui introduzimos todas as definições necessárias: σ_m é a componente m da representação interna do "professor", ou melhor, é a saída do percéptron m da camada interna cujo campo pós-sináptico é $b_m = \mathbf{B}_m \cdot \mathbf{S}$ e os pesos sinápticos são dados por $\mathbf{B}_m$. Já a função de transferência dos neurônios da camada escondida é $g(x) = erf\left(\frac{x}{\sqrt{2}}\right)$.

De maneira análoga é definida a saída do aluno :

$$\Sigma_J = \sum_{k=1}^{K} g\left(h_k\right). \tag{2.16}$$

Aqui $h_k = \mathbf{J}\cdot\mathbf{S}$ é o campo pós sináptico do neurônio k da camada escondida.

Nos ocuparemos aqui apenas do caso onde as sinápses que unem os neurônios da camada interna ao da camada externa são mantidas fixas com valor "1" como está indicado em (Fig. 2.3)[7]. Desta forma o algoritmo *backpropagation* assume a forma:

$$\mathbf{J}_k(\mu+1) = \mathbf{J}_k(\mu) + \frac{\eta}{N}\delta_k(\mu)\mathbf{S}(\mu). \tag{2.17}$$

Onde $\delta_k = g^{'}(h_k)(\Sigma_B - \Sigma_J)$. Esta forma é análoga a (2.2) com a função modulação $F_k \equiv \eta\delta_k$. As equações macroscópicas (2.10) para este tipo de sistema ficam então:

$$\dot{R}_{kn} = \eta\left\langle\delta_k b_n\right\rangle, \tag{2.18}$$

$$\dot{Q}_{kj} = \eta\left\langle\delta_j h_k\right\rangle + \eta\left\langle\delta_j h_k\right\rangle + \eta^2\left\langle\delta_j\delta_k\right\rangle. \tag{2.19}$$

Estas equações envolvem o cálculo da integral gaussiana tridimensional:

$$I_3(x,y,z) = \left\langle g^{'}(x)yg(z)\right\rangle. \tag{2.20}$$

De três reduções bidimensionais:

$$\left\langle g^{'}(y)yg(z)\right\rangle,$$

$$\left\langle g^{'}(x)zg(z)\right\rangle,$$

$$\left\langle g^{'}(x)yg(x)\right\rangle.$$

E uma redução unidimensional:

$$\left\langle g^{'}(x)xg(x)\right\rangle.$$

Envolvem também a integral quadridimensional:

$$I_4(x,y,z,w) = \left\langle g^{'}(x)g^{'}(y)g(z)g(w)\right\rangle. \tag{2.21}$$

Três reduções tridimensionais:

$$\left\langle g^{'}(x)g^{'}(y)g(z)g(z)\right\rangle,$$

$$\left\langle g^{'}(x)g^{'}(x)g(y)g(w)\right\rangle,$$

$$\left\langle g^{'}(x)g^{'}(y)g(y)g(z)\right\rangle.$$

[7] O caso com estas sinápses adaptativas foi estudado em [40].

Quatro reduções bidimensionais:

$$\left\langle g^{'}(x)g^{'}(y)g(y)g(y)\right\rangle ,$$

$$\left\langle g^{'}(x)g^{'}(x)g(x)g(y)\right\rangle ,$$

$$\left\langle g^{'}(x)g^{'}(x)g(y)g(y)\right\rangle ,$$

$$\left\langle g^{'}(x)g^{'}(y)g(x)g(y)\right\rangle .$$

E uma redução unidimensional:

$$\left\langle g^{'}(x)g^{'}(x)g(x)g(x)\right\rangle .$$

Totalizando 14 integrais gaussianas. Na verdade, estas integrais podem ser expressadas em função apenas de (2.20) e (2.21) reduzindo o número de integrações necessárias a 2 (ver [46]). Estas integrais adquirem então a forma :

$$I_3(x,z,z) = \left\langle g^{'}(x)zg(z)\right\rangle ,$$

$$I_3(x,x,x) = \left\langle g^{'}(x)xg(x)\right\rangle ,$$

$$I_4(x,y,x,y) = \left\langle g^{'}(x)g^{'}(y)g(x)g(y)\right\rangle .$$

E assim por diante. Com a escolha de $g(x) = erf\left(\frac{x}{2}\right)$ estas integrais são factíveis levando a [46, 60]:

$$I_3(x,y,z) = \frac{2}{\pi}\frac{1}{\sqrt{\Lambda_3}}\frac{C_{yz}(1+C_{xx}) - C_{xy}C_{xz}}{1+C_{xx}}. \tag{2.22}$$

Com

$$\Lambda_3 = (1+C_{xx})(1+C_{zz}) - C_{xz}C_{xz},$$

onde $\mathbf{C}$ é a matriz de correlação dos campos x,y e z. Para os casos de dimensão menor substitui-se a matriz de correlação $\mathbf{C}$ pela matriz singular apropriada. Por exemplo : no cálculo de $I_3(x,y,y)$, $\mathbf{C}$ é a matriz singular obtida pelas substituição $z \rightarrow y$.

$$I_4(x,y,z,w) = \frac{4}{\pi^2}\frac{1}{\sqrt{\Lambda_4}}\arcsin\left(\frac{\Lambda_0}{\sqrt{\Lambda_1\Lambda_2}}\right). \tag{2.23}$$

Com

$$\begin{aligned}
\Lambda_4 &= (1+C_{xx})(1+C_{yy}) - C_{xy}C_{xy},\\
\Lambda_0 &= \Lambda_4 C_{zw} - C_{yz}C_{yw}(1+C_{xx}) - C_{xz}C_{xw}(1+C_{yy})\\
&+ C_{xy}C_{xz}C_{yw} + C_{xy}C_{xw}C_{yz},\\
\Lambda_1 &= \Lambda_4(1+C_{zz}) - C_{yz}^2(1+C_{xx}) - C_{xz}^2(1+C_{yy}) + 2C_{xy}C_{xz}C_{yz},\\
\Lambda_2 &= \Lambda_4(1+C_{ww}) - C_{yw}^2(1+C_{xx}) - C_{xw}^2(1+C_{yy}) + 2C_{xy}C_{xw}C_{yw}.
\end{aligned}$$

Aqui $\mathbf{C}$ é a matriz de correlação dos campos x,y,z e w. Nos casos com dimensão menor, esta matriz é substituída pela matriz singular equivalente. Por exemplo : no cálculo de $I_4(x,x,x,x)$, $\mathbf{C}$ é a matriz singular obtida pelas substituições $y \to x$, $z \to x$ e $w \to x$.

Exemplificando com $\left\langle g'(h_k)\, b_n g(h_j) \right\rangle$ teremos que a matriz de correlação $\mathbf{C}$, usando (2.4), será dada por:

$$\mathbf{C} = \begin{pmatrix} \langle h_k h_k \rangle & \langle h_k b_n \rangle & \langle h_k h_j \rangle \\ \langle b_n h_k \rangle & \langle b_n b_n \rangle & \langle b_n h_j \rangle \\ \langle h_j h_k \rangle & \langle h_j b_n \rangle & \langle h_j h_j \rangle \end{pmatrix} = \begin{pmatrix} Q_{kk} & R_{kn} & Q_{kj} \\ R_{kn} & M_{nn} & R_{jn} \\ Q_{jk} & R_{jn} & Q_{jj} \end{pmatrix}.$$

O cálculo destas médias possibilita a resolução e caracterização total das equações da dinâmica de aprendizado. As curvas de aprendizagem podem então ser obtidas através do cálculo do erro de generalização. Para o ambiente de teste onde os exemplos são gerados uniformemente e independentemente o erro de generalização é dado em função das variáveis macroscópicas por (Apêndice C):

$$\begin{aligned} e_g(Q_{jk}, R_{kn}, M_{nm}) &= \frac{1}{\pi} \sum_{jk} \arcsin\left(\frac{Q_{jk}}{\sqrt{1+Q_{jj}}\sqrt{1+Q_{kk}}} \right) \\ &+ \frac{1}{\pi} \sum_{nm} \arcsin\left(\frac{M_{nm}}{\sqrt{1+M_{nn}}\sqrt{1+M_{mm}}} \right) \\ &- \frac{2}{\pi} \sum_{kn} \arcsin\left(\frac{R_{kn}}{\sqrt{1+Q_{kk}}\sqrt{1+M_{nn}}} \right). \end{aligned} \tag{2.24}$$

2.4 Percéptrons

A solução para percéptrons (Fig. 2.4) numa situação sem ruído e com a norma do professor $M = \mathbf{B} \cdot \mathbf{B} = 1$ foi descrita em [7]. Neste caso as equações (2.18) adquirem a forma:

$$\dot{R} = \eta \langle \delta b \rangle \tag{2.25}$$

$$\dot{Q} = 2\eta \langle \delta h \rangle + \eta^2 \langle \delta^2 \rangle \tag{2.26}$$

Inserindo as médias (2.20) e (2.21) reescrevemos:

$$\dot{R} = \eta \left[I_3(h,b,b) - I_3(h,b,h) \right] \tag{2.27}$$

$$\begin{aligned} \dot{Q} &= 2\eta \left[I_3(h,h,b) - I_3(h,h,h) \right] \\ &+ \eta^2 \left[I_4(h,h,b,b) - 2I_4(h,h,b,h) + I_4(h,h,h,h) \right] \end{aligned} \tag{2.28}$$

Substituindo (2.22) e (2.23) chegamos:

$$\dot{R} = \frac{2}{\pi} \frac{\eta}{1+Q} \left[\frac{1+Q-R^2}{\sqrt{2(1+Q)-R^2}} - \frac{R}{\sqrt{1+2Q}} \right] \tag{2.29}$$

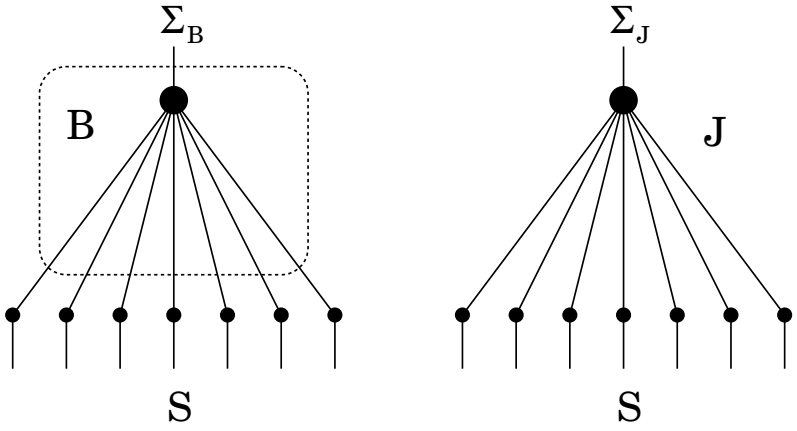


Figura 2.4: Professor = Aluno = Percéptron.

$$\begin{aligned}\dot{Q} &= \frac{4}{\pi}\frac{\eta}{1+Q}\left[\frac{R}{\sqrt{2(1+Q)-R^2}} - \frac{Q}{\sqrt{1+2Q}}\right] \\ &+ \frac{4}{\pi^2}\frac{\eta^2}{\sqrt{1+2Q}}\left[\arcsin\left(\frac{Q}{1+3Q}\right)\right. \\ &- 2\arcsin\left(\frac{R}{\sqrt{2(1+2Q-R^2)}\sqrt{1+3Q}}\right) \\ &+ \left.\arcsin\left(\frac{1+2(Q-R^2)}{2(1+2Q-R^2)}\right)\right]. \end{aligned} \tag{2.30}$$

Estas equações podem ser resolvidas numericamente. A curva de aprendizado pode então ser construída utilizando a versão para o percéptron de (2.24):

$$e_g\,(Q,R) = \frac{1}{\pi}\arcsin\left(\frac{Q}{1+Q}\right) - \frac{2}{\pi}\arcsin\left(\frac{R}{\sqrt{2(1+Q)}}\right) + \frac{1}{6}. \tag{2.31}$$

É clara a presença de um ponto fixo em $(R,Q) = (1,1)$ em (2.29). No mesmo trabalho Biehl e Schwarze estudaram a estabilidade assintótica linearizando o sistema dinâmico em torno do ponto fixo. Perceberam que esta estabilidade depende da taxa de aprendizado η da seguinte forma:

- Há um valor crítico $\eta_c \approx 4.06$ acima do qual $(R,Q) = (1,1)$ deixa de ser um ponto fixo atrativo, mas existem pontos fixos sub-ótimos para os quais $e_g(\alpha \to \infty) > 0$.
- Para taxas de aprendizagem maiores que $\eta_d \approx 9.24$ não há nenhum ponto fixo e os parâmetros (R,Q) divergem.

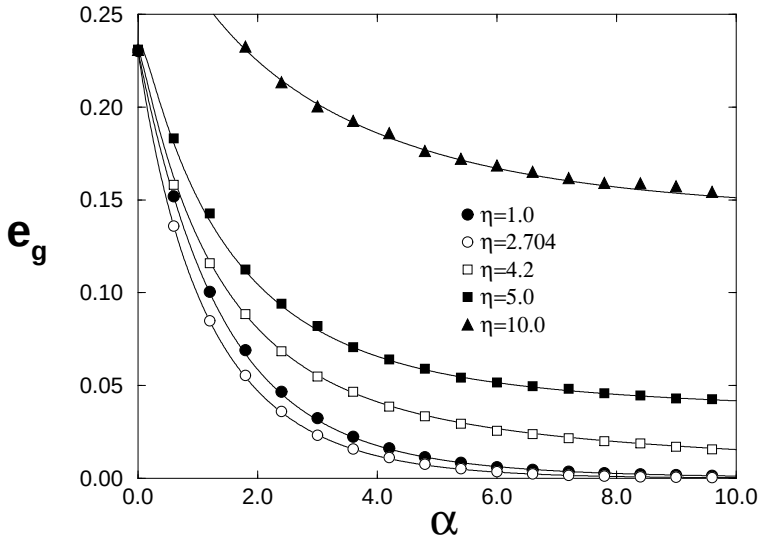


Figura 2.5: Linhas cheias: Integrações numéricas. Símbolos: Simulações com redes de tamanho N=1000. As condições iniciais são Q(0)=.25 e R(0)=0.

- Há uma região definida por $4.45 \leq \eta \leq 5.05$ na qual os autovalores do operador linear que define o comportamento do sistema dinâmico em torno do ponto fixo são números complexos, significando um comportamento oscilatório amortecido .
- O decaimento assintótico ótimo para o erro de generalização é obtido para $\eta_{opt} = \frac{2}{3}\eta_c \approx 2.704$. Este decaimento é dado por $e_g \sim e^{-.66\alpha}$.

Mostramos na (Fig.2.5) simulações e resultados analíticos para valores interessantes de η.

2.5 Caso Sobre-realizável: $M < K$

A situação mostrada na (Fig. 2.6) também foi estudada em [7]. Aqui o professor é um percéptron ($M = 1$) e o aluno tem arquitetura tipo comitê *soft* com $K = 2$ unidades na camada escondida. Biehl e Schwarze estudaram o sistema para diferentes valores das sinápses da camada de saída. Aqui nos restringiremos a situação em que estas sinápses tem valores fixados em 1 , como na (Sec. 2.3). Seguindo o procedimento das seções anteriores escrevemos as equações dinâmicas para os parâmetros de ordem:

$$\dot{R}_k = \eta\langle\delta_k b\rangle \tag{2.32}$$

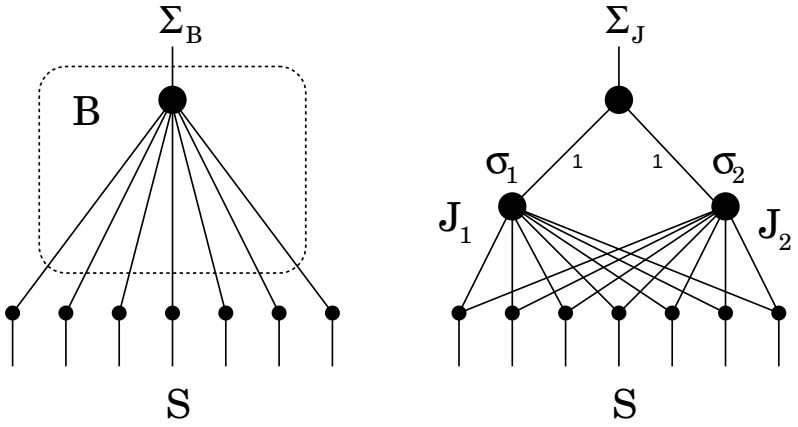


Figura 2.6: Professor = Percéptron. Aluno = Comitê *soft* com 2 neurônios na camada interna.

$$\dot{C} = \eta\langle\delta_1 h_2\rangle + \eta\langle\delta_2 h_1\rangle + \eta^2\langle\delta_1\delta_2\rangle \tag{2.33}$$

$$\dot{Q}_k = 2\eta\langle\delta_k h_k\rangle + \eta^2\langle\delta_k^2\rangle \tag{2.34}$$

Aqui definimos $C \equiv Q_{12}$. As médias acima podem ser reescritas em funçãodos parâmetros de ordem utilizando (2.22) e (2.23) e então integradas numericamente. Já o erro de generalização é escrito:

$$\begin{aligned} e_g\left(Q_k, C, R_k\right) &= \frac{1}{\pi}\sum_k \arcsin\left(\frac{Q_k}{1+Q_k}\right) + \frac{2}{\pi}\arcsin\left(\frac{C}{\sqrt{(1+Q_1)}\sqrt{(1+Q_2}}\right) \\ &- \frac{2}{\pi}\sum_k \arcsin\left(\frac{R_k}{\sqrt{2(1+Q_k)}}\right) + \frac{1}{6}. \end{aligned} \tag{2.35}$$

A solução para as equações dinâmicas das variáveis de estado macroscópicas são mostradas em (Fig. 2.8) e (Fig. 2.7). A curva de aprendizagem é mostrada em (Fig. 2.7). Nota-se a presença de uma fase simétrica, ou platô, que é comumente encontrado em situações práticas [24]. Estes platôs indicam a presença de um ponto fixo com direções repulsivas e atrativas.

Biehl e Schwarze analisaram a estabilidade dos pontos fixos da dinâmica em relação a taxa de aprendizagem η concluindo que:

- Para $\eta < \eta_1 \approx 1.4$ existem três pontos fixos: um repulsivo, o de platô e um atrativo que corresponde ao aprendizado com $e_g(\alpha \to \infty) = 0$.
- Para $1.4 < \eta < \eta_c \approx 1.89$ há o surgimento de um quarto ponto fixo repulsivo.
- Para $\eta_c < \eta < \eta_s \approx 2.32$ [46] o ponto fixo de aprendizado passa a ser instável e surge um ponto fixo estável que corresponde ao aprendizado subótimo $e_g(\alpha \to \infty) > 0$.

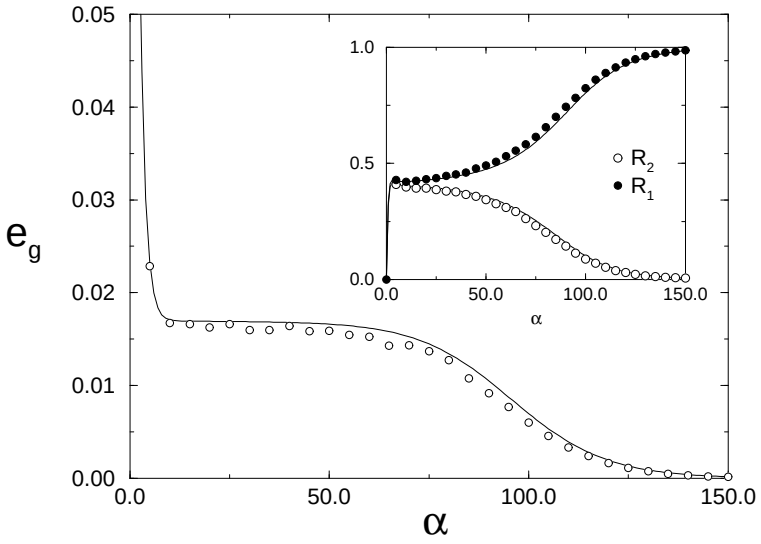


Figura 2.7: Curva de aprendizagem para $\eta = 1.5$ e condicções iniciais aleatórias com $Q_1 \in [0, .5]$, $Q_2 \in [0, 1E-6]$ e $C \approx 0$. *Inset*: evolução dos *overlaps* R_1 e R_2. Símbolos: simulações com tamanho $N = 5000$ e condições iniciais idênticas.

- Para $\eta_s < \eta < \eta_d \approx 3.29$ o ponto fixo de platô torna-se o único estável.
- Para $\eta > \eta_d$ as normas dos vetores sinápticos divergem.

2.6 Caso Realizável: $M = K$

O caso realizável foi tratado em [45, 46, 6, 9, 10]. As equações dinâmicas para o caso mais simples com $M = K = 2$ e professor do tipo $M_{nm} = \delta_{nm}$ são:

$$\dot{R_{kn}} = \eta\langle\delta_k b_n\rangle \tag{2.36}$$

$$\dot{C} = \eta\langle\delta_1 h_2\rangle + \eta\langle\delta_2 h_1\rangle + \eta^2\langle\delta_1\delta_2\rangle \tag{2.37}$$

$$\dot{Q_k} = 2\eta\langle\delta_k h_k\rangle + \eta^2\langle\delta_k^2\rangle \tag{2.38}$$

Novamente utilizamos (2.22) e (2.23) e integramos numericamente. O erro de generalização é escrito:

$$e_g\left(Q_k, C, R_{kn}\right) = \frac{1}{\pi}\sum_k \arcsin\left(\frac{Q_k}{1+Q_k}\right)$$

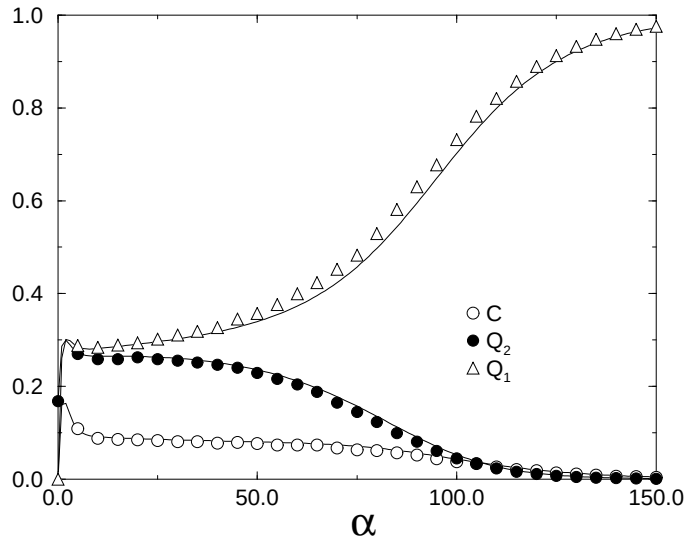


Figura 2.8: *Overlaps* entre os ramos do aluno para as mesmas condições da figura anterior. Símbolos: simulações com tamanho $N = 5000$.

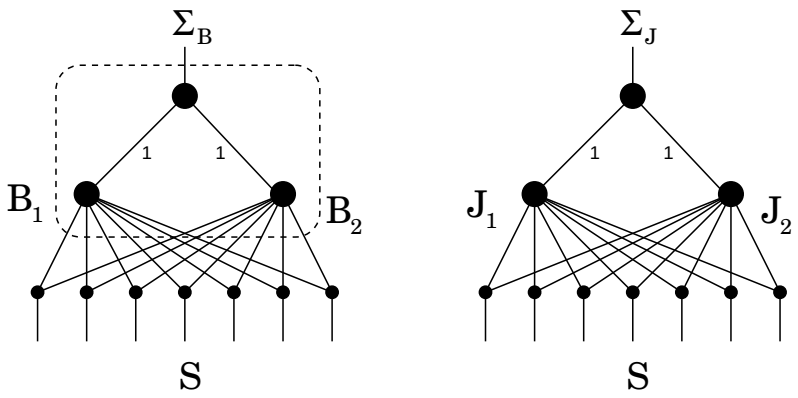


Figura 2.9: Professor = Aluno = Comitê *soft* com 2 neurônios na camada interna.

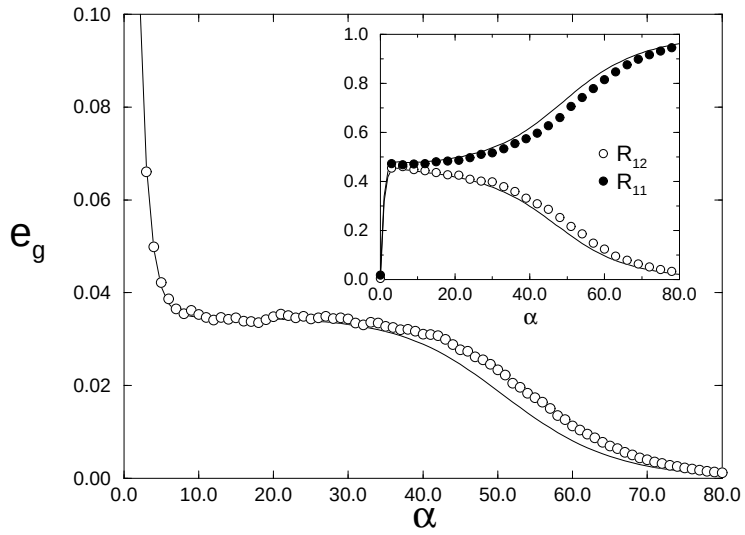


Figura 2.10: Curva de aprendizagem para $\eta = 1.5$ e condições iniciais aleatórias com $Q_1 \in [0, .5]$, $Q_2 \in [0, 1E-6]$ e $C \approx 0$. *Inset*: Evolução dos *overlaps* professor-aluno. Símbolos: simulações com tamanho $N = 5000$.

$$+ \quad \frac{2}{\pi} \arcsin\left(\frac{C}{\sqrt{(1+Q_1)}\sqrt{(1+Q_2}}\right)$$
$$- \quad \frac{2}{\pi} \sum_{k,n} \arcsin\left(\frac{R_{kn}}{\sqrt{2(1+Q_k)}}\right) + \frac{1}{3}. \qquad (2.39)$$

Em [10] Biehl *et.al.* observaram que o sistema de equações diferenciais acima possui, para taxas de aprendizagem menores que um valor crítico $\eta_c \approx 2.32$, um ponto fixo de aprendizado com $e_g = 0$, $Q_{ik} = \delta_{ik}$ e $R_{kn} = \delta_{kn}$ e dois pontos fixos subótimos: um perfeitamente simétrico com $Q_{ik} = Q$ e $R_{kn} = R$ e um menos simétrico. Conforme a taxa de aprendizagem $\eta \to 0$ o ponto menos simétrico colapsa sobre o totalmente simétrico.

As curvas de aprendizagem apresentam um aspecto similar ao caso sobre-realizável. A saída dos platôs ocorre graças à presença de uma vizinhança instável nos pontos fixos subótimos de forma que uma assimetria nas condições iniciais introduz uma componente na direção repulsiva levando o sistema ao ponto fixo de aprendizado. Na (Fig.2.10) mostramos a curva de aprendizagem obtida com a resolução numérica do sistema dinâmico, para comparação, mostramos a mesma curva obtida através de uma única simulação com tamanho $N = 5000$ e condições iniciais idênticas. As flutuações no platô são bastante acentuadas indicando a presença da direção atrativa.

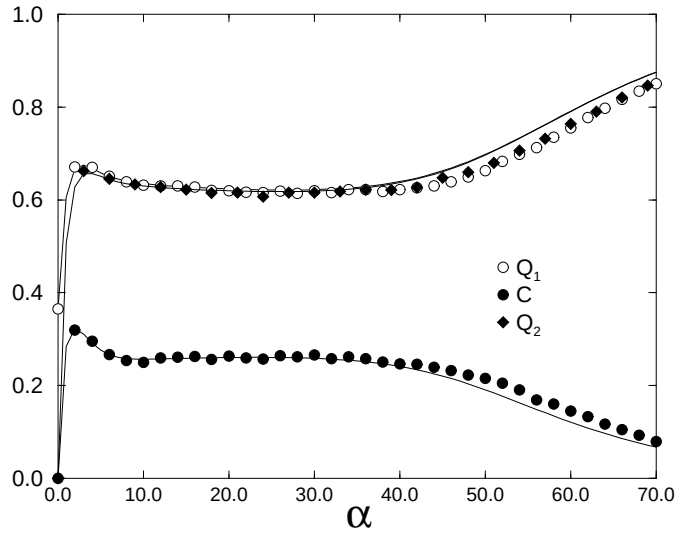


Figura 2.11: *Overlaps* entre os ramos do aluno. Símbolos: simulações com tamanho $N = 5000$.

Na (Fig. 2.11) estão representados os *overlaps* entre um ramo do aluno e os ramos do professor. Nota-se que o efeito de tamanho finito é particularmente acentuado durante a etapa de especialização (saída do platô) [5]. A presença dos pontos fixos de platô pode ser compreendida através de argumentos geométricos. De maneira geral podemos escrever:

$$\mathbf{J}_k = \sum_n R_{kn}\mathbf{B}_n + \mathbf{J}_k^{\perp}. \tag{2.40}$$

Na fase inicial do aprendizado há pouca correlação entre os ramos do aluno e do professor, ou seja, o vetor sináptico é $\mathbf{J}_k \sim \mathbf{J}_k^{\perp}$. Conforme os erros são corrigidos pelo algoritmo de aprendizagem a correlação aumenta e o termo $\mathbf{J}_k^{\perp}$ é reduzido. Se os ramos tem correlações R_{kn} muito similares, o algoritmo modificará os ramos de maneira similar. É claro que o estado simétrico de maior correlação corresponde a representações das sinápses do aluno no hiperplano gerado pelos ramos do professor na forma:

$$\mathbf{J}_k = MR\mathbf{B}_n. \tag{2.41}$$

Daqui deduzimos a relação:

$$Q = MR^2, \tag{2.42}$$

verificável nas figuras. Se o aluno se encontrar no estado perfeitamente simétrico de maior correlação com o professor, o algoritmo modificará os ramos sempre de maneira idêntica provocando erros maiores, isso produz um ponto fixo simétrico. É evidente que há um ponto fixo definido por $\mathbf{J}_n = \mathbf{B}_n$ para o qual o aluno não comete

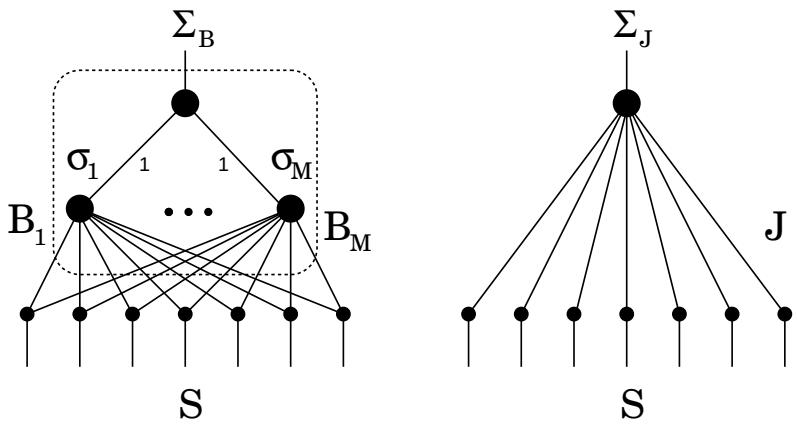


Figura 2.12: Professor = Multicamada, Aluno = Percéptron.

erro algum. A migração dos vetores sinápticos da situação simétrica para este ponto fixo ótimo depende da existência de assimetrias provenientes de condições iniciais não-simétricas. Estas assimetrias irão fazer com que o algoritmo de aprendizagem module de maneira diferente cada um dos ramos, evitando o estado totalmente simétrico.

2.7 Caso Não-realizável: $M > K$

Nesta situação uma rede com K neurônios na camada interna, tenta aprender uma rede multicamada mais complexa com $M > K$ neurônios na camada escondida. Exemplificaremos esta situação utilizando o caso mais simples onde um percéptron ($K = 1$) aprende uma rede multicamada do tipo $M_{kn} = \delta_{nm}$ (Fig.2.12).

Neste caso o conjunto de parâmetros de ordem se reduz a $R_m = \mathbf{J} \cdot \mathbf{B_m}$ e $Q = \mathbf{J} \cdot \mathbf{J}$. As equações diferenciais são :

$$\dot{R}_n = \eta \langle \delta b_n \rangle \tag{2.43}$$

$$\dot{Q} = 2\eta \langle \delta h \rangle + \eta^2 \langle \delta^2 \rangle \tag{2.44}$$

E o erro de generalização é:

$$e_g\left(Q, R_n\right) = \frac{1}{\pi} \sum_k \arcsin\left(\frac{Q}{1+Q}\right) - \frac{2}{\pi} \sum_n \arcsin\left(\frac{R_n}{\sqrt{2(1+Q)}}\right) + \frac{M}{3}. \tag{2.45}$$

Escrevendo o sistema de equações da maneira apropriada, utilizando (2.22) e (2.23), não é difícil verificar que $R_n = 1$ e $Q = M$ é ponto fixo. Isto pode ser

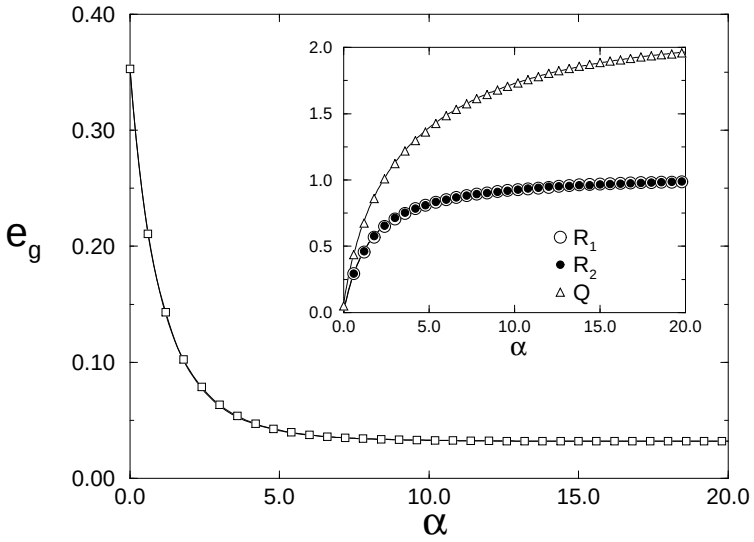


Figura 2.13: Curva de aprendizagem para *backpropagation* com M=2, K=1, $\eta = 1.5$ e condições iniciais aleatórias com $Q \in [0, .5]$. Símbolos: simulações com tamanho $N = 5000$.

entendido de maneira simples utilizando argumentos geométricos. É de se esperar que ao final do processo de aprendizado o vetor sináptico do percéptron aluno seja equidistante de todos os ramos do professor, pois não há ramos preferenciais [8], assim teremos :

$$\mathbf{J} = \sum_n R\mathbf{B}_n + \mathbf{J}^{\perp}. \tag{2.46}$$

O termo $\mathbf{J}^{\perp}$ deverá flutuar em torno de 0 , pois não há nenhuma direção privilegiada no subespaço ortogonal. Escrevendo em termos apenas dos parâmetros de ordem teremos:

$$Q \approx MR^2. \tag{2.47}$$

Isto explica a relação observada entre Q e os Rs. Se considerarmos o espaço das redes professor do tipo $M_{nm} = \delta_{nm}$ o percéptron atua como um "medidor de complexidade", fornecendo o número exato de neurônios na camada interna da rede professor através da relação: $Q \to M$ quando $\alpha \to \infty$ De maneira mais genérica teremos que, dada a estrutura da matriz de correlação do professor $M_{nm} = f(n, m)$, é possível determinar sua complexidade (número de neurônios na camada escondida) usando um percéptron. Para isso basta observar que $eg(\alpha \to \infty) = \epsilon(M)$. Na (Fig. 2.13) mostramos a curva de aprendizagem e a evolução dos parâmetros de ordem para o caso $M = 2$ e $K = 1$.

[8] Lembrando que os exemplos são gerados de maneira uniforme.

2.8 Variações sobre o Backpropagation

Nesta seção faremos um sumário e indicaremos algumas direções adicionais sobre alguns dos estudos analíticos atuais que envolvem variações do algoritmo *backpropagation*. Estas variações têm por objetivo melhorar o desempenho de redes multicamada diminuindo ou eliminando a duração dos estados subótimos de platô.

i. Backpropagation Adaptativo

West e Saad propuseram em [57] a variante que denominaram *backpropagation adaptativo*. Esta variante consiste da função modulação :

$$F_k = \eta g^{'}(\beta h_k)\,(\Sigma_B - \Sigma_J) \tag{2.48}$$

Para o caso em que $g(x) = erf(\frac{x}{\sqrt{2}})$ a função modulação assume a forma:

$$F_k = \eta\sqrt{\frac{2}{\pi}}e^{-\frac{1}{2}\beta^2 h_k^2}\,(\Sigma_B - \Sigma_J) \tag{2.49}$$

A introdução do parâmetro β permite o controle da sensibilidade da modulação aos campos grandes. Conforme β aumenta o decaimento do termo gaussiano se torna mais rápido, concentrando a modulação em campos pequenos. No estado de platô as diferenças entre os campos pós-sinápticos presentes em cada neurônio da camada interna são mínimas. Se aumentamos β, aumentamos a chance de que pequenas diferenças do campo impliquem em grandes diferenças de modulação, facilitando a quebra de simetria e a saída do estado de platô. Os cálculos analíticos para este algoritmo são adaptações razoavelmente triviais do esquema de cálculo para o algoritmo *backpropagation* usual. West e Saad demonstraram em [57] que o *backpropagation adaptativo* produz uma redução significativa do comprimento dos platôs. Eles também determinaram, no regime de η pequeno, o valor de β que produz os platôs mais curtos e os valores de η e β que produzem o decaimento assintótico ótimo. Uma direção promissora para futuros desenvolvimentos neste tipo de algoritmo seria a obtenção de evoluções ótimas $\beta_{opt}(\alpha)$ e $\eta_{opt}(\alpha)$.

ii. Otimização Paramétrica Global

O problema da evolução otimizada da taxa de aprendizagem η foi endereçado por Rattray e Saad em [44]. Utilizando um método variacional, similar a proposta de Kinouchi e Caticha [29], Rattray e Saad otimizam o funcional :

$$\delta e_g = \int_{\alpha_0}^{\alpha_1} d\alpha \frac{de_g}{d\alpha}. \tag{2.50}$$

Com os vínculos:

$$\dot{R}_{in} = \eta\langle\delta_i b_n\rangle, \tag{2.51}$$
$$\dot{Q}_{ik} = \eta\langle\delta_i h_k + \delta_k h_i\rangle + \eta^2\langle\delta_i\delta_k\rangle. \tag{2.52}$$

E impondo que o desempenho seja localmente ótimo na extremidade α_1. O resultado fornece a evolução ótima $\eta_{opt}(\alpha)$ na janela (α_0, α_1). A ultilização de $\eta_{opt}(\alpha)$ reduz sensivelmente os platôs. Este tipo de enfoque requer o conhecimento *a priori* do número de exemplos disponível, o que nem sempre representa bem uma situação de aprendizado *online*. Um estudo comparativo do desempenho com evolução $\eta(\alpha)$ localmente ótima seria de grande interesse.

iii. Quebra de Simetria Induzida

Barber, Sollich e Saad sugerem em [5] a introdução de uma medida de erro na forma:

$$e(\mathbf{J}^\mu, \mathbf{S}^\mu) = \frac{1}{2}\left(\Sigma_J(\mu) - \Sigma_B(\mu)\right)^2 + \frac{1}{2}\sum_{j=1}^{K-1} H\left(Q^\mu_{j+1,j+1} - Q^\mu_{jj}\right) \tag{2.53}$$

Com $H(x) = \frac{1}{2}\left(1 + erf\left(\frac{\beta}{\sqrt{2}}x\right)\right)$. Esta escolha penaliza estados simétricos (como os de platô) e favorece a ordenação $Q_{11} \geq Q_{22} \geq ... \geq Q_{KK}$. Os cálculos analíticos são novamente factíveis adaptando o esquema de cálculo usado para o *backpropagation* usual. O resultado são platôs violentamente reduzidos.

Similarmente, Wöhler sugere em [60] uma medida de erro dependente do ramo dada por:

$$e_k(\mathbf{J}^\mu, \mathbf{S}^\mu) = \frac{1}{2}\left(\Sigma_J(\mu) - \Sigma_B(\mu)\right)^2 + \gamma \sum_{l=1, l\neq k}^{K} \left(Q^\mu_{kl}\right)^2. \tag{2.54}$$

Esta energia penaliza ramos muito correlacionados, induzindo quebras de simetria.

Neste tipo de solução considera-se que a estrutura da matriz de correlação M_{nm} do professor é conhecida, além disso a solução é apenas heurística.

iv. Gradiente Natural

Em [2] e [3] Amari propôs uma descrição para o problema do aprendizado em redes com ruído na saída em termos de uma dinâmica no espaço $\mathcal{P}$ das distribuições de probabilidade. Neste esquema a rede professor é descrita pelo mapeamento $\Sigma_B(\mathbf{S}, \mathbf{B}_n) + \epsilon$ e a rede aluno, analogamente, por $\Sigma_J(\mathbf{S}, \mathbf{J}_k) + \epsilon$. Onde ϵ denota um ruído gaussiano com média nula e variança conhecida. O problema consiste então em inferir os parâmetros $\mathbf{B}_n$ de uma distribuição de probabilidades a partir

da qual pares de exemplos $(\mathbf{S}, \Sigma_B)$ são colhidos. Esta inferência pode ser realizada utilizando o método tradicional de descenso pelo gradiente:

$$\mathbf{J}(\mu+1) = \mathbf{J}(\mu) + \eta \nabla E(\mu) \tag{2.55}$$

Aqui E é a função custo do problema. No entanto, como estamos tentando reproduzir uma dada distribuição de probabilidade do espaço $\mathcal{P}$, o gradiente deve refletir a estrutura deste espaço. Colocando de outra forma, a dinâmica que gostaríamos de realizar é:

$$P(\mu+1) = P(\mu) + \eta \nabla_P E[P(\mu)] \tag{2.56}$$

Onde $P \in \mathcal{P}$ são distribuições parametrizadas pelos $\mathbf{J}_k$. Em [2] Amari estuda as propriedades da variedade diferenciável $\mathcal{P}$ e define o *gradiente natural* sobre o espaço dos parâmetros:

$$J_i(\mu+1) = J_i(\mu) + \eta g_{ij} \frac{\partial E(\mu)}{\partial J_j} \tag{2.57}$$

Onde J_i denota cada componente de cada vetor sináptico e g_{ij} é o tensor métrico da variedade $\mathcal{P}$ definido pela matriz de informação de Fisher :

$$g_{ij}(J_i) = \left\langle \frac{\partial ln P(\mathbf{S}, \Sigma_J; J_i)}{\partial J_i} \frac{\partial ln P(\mathbf{S}, \Sigma_J; J_i)}{\partial J_j} \right\rangle_{(\mathbf{S}, \Sigma_J)} \tag{2.58}$$

Este tipo de desenvolvimento explicíta uma interessante estrutura geométrica do problema do aprendizado podendo dar origem a *insights* que podem ser de grande utilidade nos próximos anos.

3

Otimização Funcional de Algoritmos Online

Neste capítulo introduziremos o esquema funcional de otimização de maneira geral e o exemplificaremos em uma variedade de situações já bem conhecidas. Também aplicaremos o método ao percéptron com função de transferência contínua e não-linear.

3.1 Esquema Variacional de Otimização

A otimização variacional no cenário professor-aluno de aprendizagem *online* foi proposta inicialmente por O.Kinouchi e N.Caticha [29] e aplicada com sucesso aos percéptrons booleano e linear [30, 31, 6, 8], ao comitê booleano com arquitetura de árvore [13, 14], à paridade booleana e ao percéptron com função de transferência tipo "reverse-wedge". [49, 51, 50]

No esquema funcional otimiza-se a quantidade "instantânea" de informação extraída por exemplo quando o sistema se encontra num dado estado "q".[1] Desta forma a otimização é fundamentalmente local, ou seja, os algoritmos obtidos são aqueles de melhor desempenho possível utilizando a dinâmica *online* sem o conhecimento de quantos exemplos são disponíveis [2]. Inicialmente é necessário que definamos a situação de aprendizagem através da determinação da medida de erro apropriada:

$$e_g(q). \tag{3.1}$$

A quantidade "instantânea" de informação extraída será então medida por :

$$I_q = -\dot{e}_g(q) = -\frac{de}{dq}\dot{q}. \tag{3.2}$$

[1] Aqui denotamos por q o conjunto dos parâmetros de ordem do sistema $\{q_1, q_2, ...\}$. Definiremos $\frac{df}{dq} = \sum_i \frac{\partial f}{\partial q_i}\dot{q}_i$.

[2] Analogamente ao conhecido problema de "Dilema dos Prisioneiros" em Teoria dos Jogos, supomos que cada exemplo (ou jogada) pode ser o (a) último(a) em contraste com uma possível otimização global onde precisamos conhecer o número de exemplos (ou jogadas) disponíveis.

Identificaremos o "desempenho" de uma dada função modulação F em "q" por $I_q[F]$.

Como foi discutido no (Cap. 2) a dinâmica *online* é descrita, no limite de sistemas grandes, por um sistema de equações de aspecto genérico:

$$\dot{q}[F_k] = \left\langle \sum_k a_k(q, \mathcal{H}, \mathcal{V}, \gamma) F_k + \sum_{jk} c_{jk}(q, \mathcal{H}, \mathcal{V}, \gamma) F_j F_k \right\rangle \tag{3.3}$$

Aqui γ é um conjunto de parâmetros que descrevem o professor (por exemplo: ruído e correlações nos campos). Substituindo (3.3) em (3.2), obtemos um funcional que nos dá a quantidade instantânea de informação extraída por um dado algoritmo F_k num dado estado q. A funções modulação localmente ótimas F_k^* são obtidas quando impomos:

- *Condição Necessária*[3]

$$\left(\frac{\delta \dot{e}_g[F_l]}{\delta F_k} \right)_{F_k^*} = 0, \forall k \tag{3.4}$$

- *Condição Suficiente*

$$\nu_k > 0 \, \forall k. \tag{3.5}$$

Onde ν_k são autovalores da matriz hessiana funcional $\mathbf{H}$ definida por :

$$H_{jk} \equiv \left(\frac{\delta^2 \dot{e}_g[F_l]}{\delta F_j \delta F_k} \right)_{F_j^*, F_k^*}.$$

Para garantirmos que a otimização seja válida sobre uma trajetória $q(\alpha)$ precisamos que :

$$\nu_k(q(\alpha)) \quad > \quad 0, \forall \alpha, k$$

Onde $q(\alpha)$ especifíca a trajetória no espaço dos parâmetros de ordem "q" parametrizada por α. Esta trajetória é dada pela integral:

$$q(\alpha) = q(0) + \int_0^\alpha dt \, \left\langle \sum_k a_k(q, \mathcal{H}, \mathcal{V}, \gamma) F_k + \sum_{jk} c_{jk}(q, \mathcal{H}, \mathcal{V}, \gamma) F_j F_k \right\rangle,$$

que geralmente não pode ser resolvida analiticamente. É interessante observarmos que, para as dinâmicas *online* que estamos analisando, o funcional (3.3) é do segundo grau. Isto significa que as derivadas funcionais segundas de $\dot{e}_g[F_k]$ não dependerão

[3] Aqui $\frac{\delta(...)}{\delta F}$ indica uma derivada funcional em relação a F. Para uma introdução clássica ao cálculo variacional veja [22].

das funções modulação, neste sentido elas descreverão propriedades intrínsecas do espaço dos algoritmos[4] . Poderemos interpretar estas propriedades segundo as combinações de sinais dos autovalores ν_k da matriz hessiana funcional, fazendo um paralelo com o que acontece para funções ordinárias:

- Se $\nu_k(q) > 0$ para todo k, então suponhamos que a função modulação com desempenho ótimo no estado q seja $F_q^*(\mathcal{V})$ e imaginemos uma pequena perturbação contínua $\epsilon_q(\mathcal{V})$, neste caso:

  $$I_q[F^*] > I_q[F^* + \epsilon]$$

  Ou seja, qualquer que seja a perturbação na função modulação o desempenho piora.

- Se $\nu_k(q) < 0$ para todo k então :

  $$I_q[F^*] < I_q[F^* + \epsilon]$$

  Aqui qualquer modificação melhora o desempenho.

- Se $\exists k$ tal que $\nu_k(q) = 0$, então existe uma classe de funções $\mathcal{E}_q$ tal que se $\epsilon_q \in \mathcal{E}_q$:

  $$I_q[F^*] = I_q[F^* + \epsilon]$$

  Neste caso, existem infinitos algoritmos equivalentes.

- Se existe tanto k's com $\nu_k(q) > 0$, quanto com $\nu_k(q) < 0$ então existem classes de funções $\mathcal{E}_q^+$ e $\mathcal{E}_q^-$ para as quais se $\epsilon_q \in \mathcal{E}_q^+$:

  $$I_q[F^*] > I_q[F^* + \epsilon].$$

  Se $\epsilon_q \in \mathcal{E}_q^-$:

  $$I_q[F^*] < I_q[F^* + \epsilon]$$

  Dependendo do tipo de perturbação obtêm-se algoritmos melhores ou piores.

3.2 Otimização e Informação *a priori*

As soluções apontadas por (3.4) e (3.5) fornecem algoritmos com desempenho ótimo no sentido da medida de erro escolhida e sob condições dadas *a priori*. Estas condições especificam o ambiente de teste, as arquiteturas de rede envolvidas, a informação acessível ao processamento da rede-aluno (visibilidade $\mathcal{V}$ dos F_k) e se conhecemos o número de exemplos ou não. Neste sentido, não é possível pensarmos num algoritmo com desempenho ótimo sob qualquer condição, mas sim em algoritmos que são ótimos sob condições determinadas. Estes vários algoritmos otimizados podem

[4] O espaço dos algoritmos é, neste caso, o próprio espaço das funções modulação F.

ser comparados segundo a persistência de seus desempenhos quando a informação *a priori* utilizada é diferente daquela para qual o aprendizado é otimizado. Esta "persistência" tem sido denominada *robustez* [15, 16, 51].

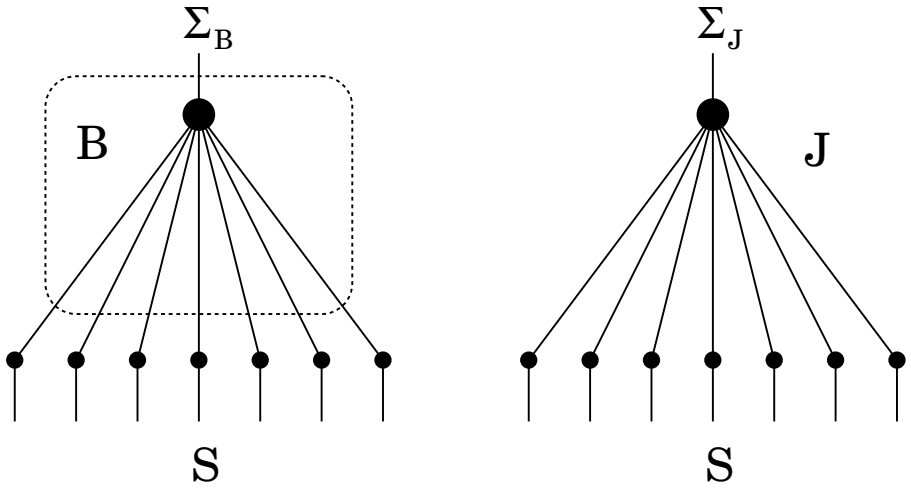
Já a informação *a priori* sobre a "visibilidade" $\mathcal{V}$ das funções modulação F_k é introduzida quando observamos que: [5].

$$\frac{\delta F_k^n(\tilde{\mathcal{V}})}{\delta F_l(\mathcal{V})} = \delta_{kl}\delta(\tilde{\mathcal{V}} - \mathcal{V})nF_k^{n-1} \tag{3.6}$$

Ao rearranjarmos as condições de otimização inserindo a relação acima teremos que as médias $\langle ... \rangle$ em (3.3) deverão ser substituídas por médias sobre distribuições marginais $\langle ... \rangle_{\mathcal{H}|\mathcal{V}}$ em (3.4).[6]

3.3 Aplicação ao Percéptron

Para exemplificar a utilização do esquema variacional na obtenção de algoritmos ótimos iremos utilizar a situação na qual um percéptron simples aprende outro .



Utilizaremos os parâmetros de ordem ρ e Q para facilitar a comparação com os resultados previamente obtidos [29, 31]. Considerando um sistema sem o termo de decaimento Ω descrito na seção (2.2), a dinâmica *online* de percéptrons com pesos sinápticos contínuos é dada por (2.10) :

$$\dot{Q} \;=\; \left\langle 2hF + F^2 \right\rangle \tag{3.7}$$

[5] Aqui $\delta(\tilde{\mathcal{V}} - \mathcal{V}) = \delta(\tilde{x}_1 - x_1)\delta(\tilde{y}_1 - y_1)...$, onde $\tilde{x}_k$ e x_k são elementos de $\tilde{\mathcal{V}}$ e $\mathcal{V}$

[6] $\langle ... \rangle_{\mathcal{H}|\mathcal{V}} = \int d\mathcal{H}\, P(\mathcal{H}|\mathcal{V})(...)$

$$\dot{\rho} \;=\; \left\langle \frac{bF}{\sqrt{QM}} - \frac{hF}{Q}\rho - \frac{F^2}{2Q}\rho \right\rangle$$

Pelo momento não precisamos especificar nem a forma explícita do erro, nem o conjunto de grandezas visíveis $\mathcal{V}$. Isto significa que os resultados que obteremos terão validade que independe de detalhes como a função tranferência ou a medida de erro que estamos utilizando. A quantidade instantânea de informação extraída será então :

$$-\, I_{Q,\rho} = \dot{e}_g = \frac{\partial e_g}{\partial Q}\dot{Q} + \frac{\partial e_g}{\partial \rho}\dot{\rho} \tag{3.8}$$

Substituindo (3.7) em (3.8) e impondo a condição necessária de otimização (3.4) teremos :

$$\frac{\delta \dot{e}_g}{\delta F} = \frac{\partial e_g}{\partial Q}\left\langle 2h + 2F \right\rangle_{\mathcal{H}|\mathcal{V}} + \frac{\partial e_g}{\partial \rho}\left\langle \frac{b}{\sqrt{QM}} - \frac{h}{Q}\rho - \frac{F}{Q}\rho \right\rangle_{\mathcal{H}|\mathcal{V}} = 0 \tag{3.9}$$

A forma geral da função modulação otimizada será então:

$$F(\mathcal{V}) = \left\langle \Phi(Q,\rho)b - h \right\rangle_{\mathcal{H}|\mathcal{V}} \tag{3.10}$$

$$\Phi(Q,M,\rho) = \frac{\frac{1}{\sqrt{QM}}\frac{\partial e_g}{\partial \rho}}{\frac{\rho}{Q}\frac{\partial e_g}{\partial \rho} - 2\frac{\partial e_g}{\partial Q}} \tag{3.11}$$

A condição suficiente (3.5) para a otimização é escrita sob a forma:

$$\frac{\delta^2 \dot{e}_g}{\delta F^2} = 2\frac{\partial e_g}{\partial Q} - \frac{\rho}{Q}\frac{\partial e_g}{\partial \rho} > 0 \tag{3.12}$$

Podemos agora, a partir de (3.10) e (3.11) reobter os resultados de otimização do erro de generalização para o percéptron linear e para o percéptron booleano:

- *a) Percéptron Booleano*

  O percéptron booleano é definido pela função de transferência $g(h) = sinal(h)$. O erro de generalição do percéptron Booleano testado num ambiente onde os exemplos são gerados com distribuição uniforme e componentes independentes é dado por :

  $$e_g = \frac{1}{\pi} arccos(\rho) \tag{3.13}$$

  Suas derivadas são :

$$\frac{\partial e_g}{\partial Q} = 0 \tag{3.14}$$

$$\frac{\partial e_g}{\partial \rho} = \frac{-1}{\sqrt{1-\rho^2}} \tag{3.15}$$

Substituindo o par de equações acima em (3.11), obtemos:

$$\Phi(Q, M, \rho) = \frac{\sqrt{Q}}{\sqrt{M}} \frac{1}{\rho} \tag{3.16}$$

A função modulação adquire então a já bem conhecida forma :

$$F(\mathcal{V}) = \left\langle \frac{\sqrt{Q}}{\sqrt{M}} \frac{b}{\rho} - h \right\rangle_{\mathcal{H}|\mathcal{V}} \tag{3.17}$$

A condição suficiente (3.12) fica então :

$$\frac{\delta^2 \dot{e}_g}{\delta F^2} = -\frac{\rho}{Q} \frac{\partial e_g}{\partial \rho} > 0 \tag{3.18}$$

que equivale simplesmente a :

$$\rho > 0 \tag{3.19}$$

Neste caso a otimização, como foi descrito na (Sec. 3.1), será válida desde que forcemos a condição dada acima.

- *b) Percéptron Linear*

  É definido pela função de transferência $g(h) = h$. Seu erro de generalização é [32]:

$$e_g = \frac{1}{2}\left(Q + M - 2\rho\sqrt{QM}\right) \tag{3.20}$$

Suas derivadas são :

$$\frac{\partial e_g}{\partial Q} = \frac{1}{2}\left(1 - \frac{\rho\sqrt{M}}{\sqrt{Q}}\right) \tag{3.21}$$

$$\frac{\partial e_g}{\partial \rho} = -\sqrt{QM} \tag{3.22}$$

Novamente, substituindo o par de equações acima em (3.11), obtemos:

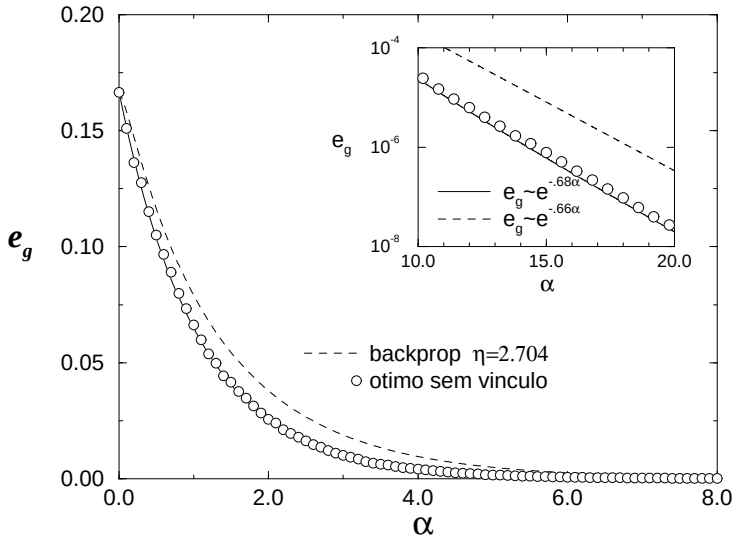


Figura 3.1: Curvas de aprendizagem para o percéptron não-linear: algoritmo otimizado contra *backpropagation* assintoticamente ótimo para condições iniciais idênticas ($Q = 1E-4$ e $R \approx 1E-4$). No *inset* mostramos o desempenho assintótico dos algoritmos. Símbolos: simulações para $N = 1000$.

$$\Phi = 1 \tag{3.23}$$

Resgatando o resultado presente em [31]:

$$F(\mathcal{V}) = \langle b - h \rangle_{\mathcal{H}|\mathcal{V}} \tag{3.24}$$

Neste caso a condição (3.12) é respeitada sempre, pois :

$$\frac{\delta^2 \dot{e}_g}{\delta F^2} = 1 > 0 \tag{3.25}$$

A independência da segunda derivada funcional de "$\dot{e}_g$"com relação aos parâmetros de ordem indica que, para o percéptron linear, a otimização será válida em qualquer trajetória $q(\alpha)$ no espaço dos parâmetros de ordem.

3.4 Percéptron Não-linear

Nesta seção trataremos do percéptron com função de transferência do tipo $g(h) = erf(h)$ já estudado no (Cap. 2) no contexto de aprendizagem com algoritmo *backpropagation*. O erro de generalização é dado, a partir de (2.24) por :

$$\begin{aligned} e_g(Q,M,\rho) &= \frac{1}{\pi} arcsin\left(\frac{Q}{1+Q}\right) - \frac{2}{\pi} arcsin\left(\frac{\rho\sqrt{QM}}{\sqrt{(1+M)(1+Q)}}\right) \\ &+ \frac{1}{\pi} arcsin\left(\frac{M}{1+M}\right) \end{aligned} \tag{3.26}$$

Seguindo o esquema utilizado acima, calculamos as derivadas de e_g :

$$\begin{aligned} \frac{\partial e_g}{\partial Q} &= \frac{1}{\pi}\frac{1}{1+Q}\left(\frac{1}{\sqrt{1+2Q}} + \frac{\rho\sqrt{QM}}{\sqrt{(1+Q)(1+M)-\rho^2 QM}}\right) \\ &- \frac{1}{\pi}\frac{\rho\sqrt{M}}{\sqrt{Q}\sqrt{(1+Q)(1+M)-\rho^2 QM}} \end{aligned} \tag{3.27}$$

$$\frac{\partial e_g}{\partial \rho} = -\frac{2}{\pi}\frac{\sqrt{QM}}{\sqrt{(1+Q)(1+M)-\rho^2 QM}} \tag{3.28}$$

Substituindo (3.27) em (3.11) teremos que:

$$\Phi(Q,M,\rho) = (1+Q)\frac{\sqrt{1+2Q}}{\sqrt{(1+M)(1+Q)-\rho^2 QM} + \rho\sqrt{QM(1+2Q)}} \tag{3.29}$$

Usando (3.12), a segunda derivada funcional fica:

$$\frac{\delta^2 \dot{e}_g}{\delta F^2} = \frac{2}{\pi}\frac{1}{1+Q}\left(\frac{1}{\sqrt{1+2Q}} + \frac{\rho\sqrt{QM}}{\sqrt{(1+Q)(1+M)-\rho^2 QM}}\right) > 0 \tag{3.30}$$

Na desigualdade acima é possível perceber facilmente que impor $\rho \geq 0$ basta para que a otimização seja válida em qualquer trajetória[7]. Na (Fig. 3.1) comparamos os desempenhos do algoritmo ótimo e do *backpropagation* com taxa de aprendizagem otimizada.

3.5 Vínculo Otimizado

A forma que a condição (3.4) de otimização toma, no caso do percéptron, é :

$$\frac{\delta \dot{e}_g}{\delta F} = \frac{\partial e_g}{\partial Q}\frac{\delta \dot{Q}}{\delta F} + \frac{\partial e_g}{\partial \rho}\frac{\delta \dot{\rho}}{\delta F} = 0 \tag{3.31}$$

[7]Esta condição pode ser refinada para $\rho > -\sqrt{\frac{1+Q}{2Q}}$.

Dentre todas as possíveis curvas no espaço (Q, ρ) que satisfazem a equação acima há aquelas que, para ρ fixado, passam por valores Q que minimizam o erro de generalização, ou seja que satisfazem:

$$\left(\frac{\partial e_g}{\partial Q}\right)_\rho = 0 \tag{3.32}$$

Isso define um vínculo entre "Q" e "ρ" que denominaremos "vínculo ótimo". Quando a equação acima tem solução, podemos utilizar este tipo de vínculos para reduzir a dimensionalidade do sistema. Observando as equações (2.10) podemos constatar que o "vínculo ótimo" pode ser implementado por escolhas adequadas dos $\langle\Omega\rangle$, desta forma conseguimos algoritmos ótimos com termo de decaimento. É fácil notar que a função modulação (3.11), quando implementamos o vínculo, toma a forma :

$$F(\mathcal{V}) = \left\langle \frac{\sqrt{Q}}{\sqrt{M}}\frac{b}{\rho} - h \right\rangle_{\mathcal{H}|\mathcal{V}} \tag{3.33}$$

Que é o algoritmo ótimo para o percéptron booleano. Já a condição suficente para que F seja ótimo reduz-se também ao $\rho > 0$ do percéptron booleano.

- *Percéptron Linear*

  Neste caso o vínculo será dado por:

  $$\left(\frac{\partial e_g}{\partial Q}\right)_\rho = \frac{1}{2}\left(1 - \frac{\sqrt{M}}{\sqrt{Q}}\rho\right) = 0$$

  Ou seja, como já foi descrito em [30] e [31], será :

  $$\sqrt{Q} = \rho\sqrt{M} \tag{3.34}$$

  Curiosamente, ao substituirmos o vínculo acima em (3.33) reobtemos a função modulação ótima para o caso sem vínculo (3.23), o que indica que, no linear, o vínculo ótimo é implementado por $\Omega = 0$.

- *Percéptron Não-linear*

  Neste caso o vínculo ótimo é:

  $$Q = \frac{\rho^2 M}{2 - \rho^2 M} \tag{3.35}$$

  Já a função modulação será :

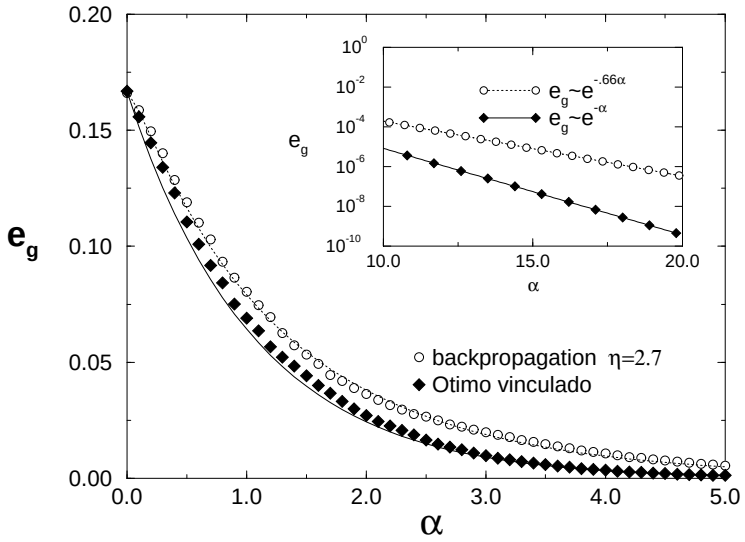


Figura 3.2: Símbolos: simulações em tamanho $N = 1000$. Linhas cheias: resultados analíticos. *Inset* : desempenho assintótico.

$$F(\mathcal{V}) = \left\langle \frac{b}{\sqrt{2-\rho^2 M}} - h \right\rangle_{\mathcal{H}|\mathcal{V}} \tag{3.36}$$

Ou seja :

$$\Phi = \frac{1}{\sqrt{2-\rho^2 M}}$$

Analiticamente chega-se ao desempenho assintótico $e_g \sim e^{-\alpha}$, idêntico ao desempenho do percépton linear, superior ao *backpropagation* e ao ótimo sem vínculo. O termo de decaimento Ω que gera o vínculo ótimo pode ser obtido supondo que as condições iniciais satisfazem (3.35) e observando que:

$$\dot{Q} = \frac{4\rho}{\left(2-\rho^2\right)^2}\dot{\rho} \tag{3.37}$$

Retomando (3.7) com o termo de decaimento das equações (2.10):

$$\begin{aligned} \dot{Q} &= \left\langle 2hF + F^2 - 2Q\Omega \right\rangle \\ \dot{\rho} &= \left\langle \frac{bF}{\sqrt{QM}} - \frac{hF}{Q}\rho - \frac{F^2}{2Q}\rho \right\rangle \end{aligned} \tag{3.38}$$

Introduzindo as relações acima em (3.37) e utilizando (3.36) :

$$\langle \Omega \rangle = \frac{1}{Q} \left\langle hF - \frac{F^2}{2} \right\rangle \tag{3.39}$$

Sob a hipótese de que o vínculo se realiza através de toda trajetória e empregando novamente (3.36) finalmente obtemos:

$$\langle \Omega \rangle = \frac{Q-1}{4Q} \tag{3.40}$$

O próprio Ω pode ser aproximado pela média acima e esta aproximação deve melhorar à medida que o tamanho do sistema aumenta graças à propriedade de automediância. Além disso não é necessário, pelo menos assintoticamente, que as condições iniciais obedeçam rigorosamente o vínculo ótimo como pode ser visto na (Fig. 3.2).

4

Otimização em Redes Multicamada

Neste capítulo aplicaremos o esquema funcional de otimização local a situações envolvendo redes multicamada. Trataremos casos realizáveis, não-realizáveis e sobre-realizáveis. Discutiremos brevemente a otimização funcional global.

4.1 Aprendizado Otimizado

Apesar de, em princípio, podermos aplicar o esquema funcional de otimização a qualquer rede multicamada, nos restringiremos aqui à análise de comitês *soft* do tipo estudado no (Capítulo 2). As equações dinâmicas para o estado macroscópico da rede são dadas por :

$$\begin{aligned} \dot{R}_{kn} &= \langle b_n F_k \rangle \\ \dot{Q}_{jk} &= \langle h_j F_k + h_k F_j + F_j F_k \rangle \,. \end{aligned} \tag{4.1}$$

Seguindo o procedimento introduzido no (Capítulo 3) é necessário definir a medida apropriada do desempenho em função dos estados macroscópicos. Utilizando exemplos de teste aleatórios, independentes e distribuídos uniformemente, o erro de generalização médio é definido, conforme o (Apêndice C), por:

$$\begin{aligned} e_g\left(Q_{jk}, R_{kn}, M_{nm}\right) &= \frac{1}{\pi}\sum_{jk} arcsin\left(\frac{Q_{jk}}{\sqrt{1+Q_{jj}}\sqrt{1+Q_{kk}}}\right) \\ &+ \frac{1}{\pi}\sum_{nm} arcsin\left(\frac{M_{nm}}{\sqrt{1+M_{nn}}\sqrt{1+M_{mm}}}\right) \\ &- \frac{2}{\pi}\sum_{kn} arcsin\left(\frac{R_{kn}}{\sqrt{1+Q_{kk}}\sqrt{1+M_{nn}}}\right). \end{aligned} \tag{4.2}$$

Lembrando que, por definição, $Q_{ik} = Q_{ki}$ e $M_{nm} = M_{mn}$ o erro de generalização é na verdade função das $MK + (M^2 + M + K^2 + K)/2$ variáveis $(M_{n\leq m}, Q_{k\leq j}, R_{kn})$. A quantidade de informação extraída de cada exemplo apresentado por redes num

certo estado macroscópico $q = (Q_{ik}, R_{kn})$ e utilizando o conjunto F de funções modulação é dada por :

$$\begin{aligned} I_q[F] &= -\dot{e}_g(q) \\ &= -\sum_{i\leq k} \frac{\partial e_g}{\partial Q_{ik}}(q)\dot{Q}_{jk} - \sum_{kn} \frac{\partial e_g}{\partial R_{kn}}(q)\dot{R}_{kn}. \end{aligned} \tag{4.3}$$

Note que a dependência nos F_k surge ao utilizarmos as equações (4.1) para reescrever a equação acima como um funcional:

$$\dot{e}_g[F] = \sum_{i\leq k} \frac{\partial e_g}{\partial Q_{ik}} \langle h_j F_k + h_k F_j + F_j F_k \rangle \tag{4.4}$$

$$+ \sum_{kn} \frac{\partial e_g}{\partial R_{kn}} \langle b_n F_k \rangle \tag{4.5}$$

As derivadas do erro de generalização são dadas por:

$$\begin{aligned} \frac{\partial e_g}{\partial Q_{jj}} &= \frac{1}{\pi}\frac{1}{1+Q_{jj}}\left[\frac{1}{\sqrt{1+2Q_{jj}}} - \sum_{k\neq j}\frac{Q_{jk}}{\sqrt{(1+Q_{jj})(1+Q_{kk})-Q_{jk}^2}}\right. \\ &+ \left.\sum_n \frac{R_{jn}}{\sqrt{2(1+Q_{jj})-R_{jn}^2}}\right], \\ \frac{\partial e_g}{\partial Q_{jk}} &= \frac{2}{\pi}\frac{Q_{jk}}{\sqrt{(1+Q_{jj})(1+Q_{kk})-Q_{jk}^2}}\ (j\neq k), \\ \frac{\partial e_g}{\partial R_{jn}} &= -\frac{2}{\pi}\frac{1}{\sqrt{2(1+Q_{jj})-R_{jn}^2}}. \end{aligned} \tag{4.6}$$

Para simplificar nossa notação iremos adotar as definições de Vicente e Caticha [54] :

$$H_{jk} = \frac{\delta^2 \dot{e}_g}{\delta F_j \delta F_k} = 2\frac{\partial e_g}{\partial Q_{ii}}\delta_{ik} + \frac{\partial e_g}{\partial Q_{ik}}(1-\delta_{ik}) \tag{4.7}$$

$$G_{kn} \equiv -\frac{\partial e_g}{\partial R_{kn}}. \tag{4.8}$$

A condição necessária para a otimização funcional é escrita (ver Sec. 3.1):

$$\frac{\delta \dot{e}_g[F]}{\delta F_j} = 0. \tag{4.9}$$

A derivação aqui é funcional e realizada com todas as outras funções mantidas fixas. Observando que as funções modulação têm acesso limitado ao conjunto $\mathcal{V}$ (veja a

Sec. 3.2) , que no caso mais simples contem os campos h_j e a saída Σ_B do professor, calculamos as derivadas funcionais acima e efetuamos os somatórios:

$$H_{kj}h_j + H_{kj}F_j - G_{kn}\langle b_n\rangle_{\mathcal{H}|\mathcal{V}} = 0. \tag{4.10}$$

Aqui estamos utilizando a convenção usual de somar sobre índices repetidos. A solução do sistema de equações acima é imediatamente dada por:

$$F_j(\mathcal{V}) = H^{(-1)}_{jk}G_{kn}\langle b_n\rangle_{\mathcal{H}|\mathcal{V}} - h_j. \tag{4.11}$$

Substituindo esta função modulação nas equações (4.1) que descrevem a dinâmica e lembrando que $\langle h_k b_n\rangle = R_{kn}$ e $\langle h_k h_j\rangle = Q_{kj}$ obtemos:

$$\begin{aligned} \dot{R}_{kn} &= H^{(-1)}_{kj}G_{jm}\left\langle \langle b_m\rangle_{\mathcal{H}|\mathcal{V}}\, b_n\right\rangle - R_{kn} \\ \dot{Q}_{jk} &= \left\langle H^{(-1)}_{jl}G_{lm}\langle b_m\rangle_{\mathcal{H}|\mathcal{V}}\, H^{(-1)}_{ki}G_{in}\langle b_n\rangle_{\mathcal{H}|\mathcal{V}}\right\rangle - Q_{jk}. \end{aligned} \tag{4.12}$$

Para escrevermos a condição suficiente (Sec. 3.1) devemos analisar a matriz hessiana funcional $\mathbf{H}$. Como foi descrito em detalhe no (Capítulo. 3), os autovalores $\nu_k(q)$ da matriz hessiana determinarão a validade da solução (4.11) num determinado estado macroscópico $q = (Q_{jk}, R_{kn})$ da rede. A solução somente produzirá aprendizado ótimo no estado q se $\nu_k(q) \geq 0$. Neste tipo de otimização não há garantias de que a otimização seja válida ao longo de todas as possíveis trajetórias $q(\alpha)$ no espaço dos estados. Para o percéptron, como foi demonstrado no (Capítulo 3), encontram-se trajetórias para as quais esta otimização local vale em todos os pontos. Para arquiteturas mais complexas, veremos nas próximas seções que a validade da otimização dependerá de condições iniciais e nem sempre será possível manter o sistema sobre trajetórias válidas.

4.2 Caso Sobre-realizável: $M < K$

A situação na qual uma rede multicamada com K neurônios na camada escondida aprende um percéptron ($M = 1$) é particularmente interessante do ponto de vista da otimização funcional, pois neste caso é possível uma solução completamente analítica. A função modulação ótima assume a forma (4.11):

$$F_j(\mathcal{V}) = H^{(-1)}_{jk}G_k\langle b\rangle_{\mathcal{H}|\mathcal{V}} - h_j. \tag{4.13}$$

Considerando o caso sem ruído teremos que $\langle b\rangle_{\mathcal{H}|\mathcal{V}} = b = g^{(-1)}(\Sigma_B)$. Além disso:

$$G_k = -\frac{\partial e_g}{\partial R_k}$$

$$H_{jk} = 2\frac{\partial e_g}{\partial Q_{jj}}\delta_{jk} + \frac{\partial e_g}{\partial Q_{jk}}(1-\delta_{jk})$$

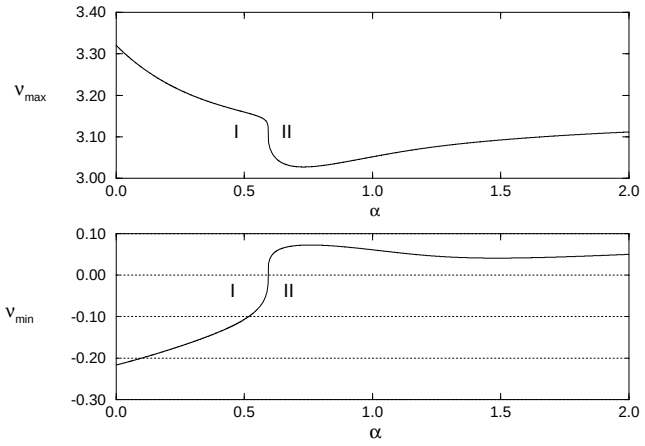


Figura 4.1: Autovalores ν da matriz hessiana funcional. O menor autovalor ν_{min} assume valores negativos na região I.

Se considerarmos que a norma do professor é $\mathbf{B}\cdot\mathbf{B} = 1$, as equações que descrevem a dinâmica são dadas por (4.12) e adquirem a forma:

$$\begin{aligned} \dot{R}_k &= H^{(-1)}_{kj} G_j - R_k \\ \dot{Q}_{jk} &= H^{(-1)}_{jl} G_l H^{(-1)}_{ki} G_i - Q_{jk}. \end{aligned} \tag{4.14}$$

Reescrevendo estas equações utilizando as derivadas do erro de generalização dadas por (4.6), podemos resolver numericamente o sistema de equações diferenciais. Nos restringindo agora ao caso em que $K = 2$, a validade da otimização pode ser verificada calculando os autovalores da matriz hessiana $\mathbf{H}$ dados por:

$$\nu = \frac{\partial e_g}{\partial Q_{11}} + \frac{\partial e_g}{\partial Q_{22}} \pm \sqrt{\left(\frac{\partial e_g}{\partial Q_{11}} + \frac{\partial e_g}{\partial Q_{22}}\right)^2 + \left(\frac{\partial e_g}{\partial Q_{12}}\right)^2 - 4\frac{\partial e_g}{\partial Q_{11}}\frac{\partial e_g}{\partial Q_{22}}}. \tag{4.15}$$

A evolução destes autovalores para condições iniciais aleatórias com $Q_{11} \in [0, .5]$, $Q_{22} \in [0, 1E-6]$ e $Q_{12} \approx 0$ é exibida na (Fig. 4.1). Nota-se que um dos autovalores assume valor negativo na região denotada por **I**. Isso indica, segundo a interpretação do (Capítulo 3), que nesta região há algoritmos de desempenho melhor. De fato, podemos definir um algoritmo com desempenho superior na fase inicial de aprendizado. Denominamos este algoritmo "crivo" e o definimos na forma seguinte:

$$k^* = \{k : \max Q_{kk}\},$$

então

$$F_k \equiv \begin{cases} F_{M=K=1} & \text{se } k = k^* \\ -h_k & \text{c.c.} \end{cases} . \tag{4.16}$$

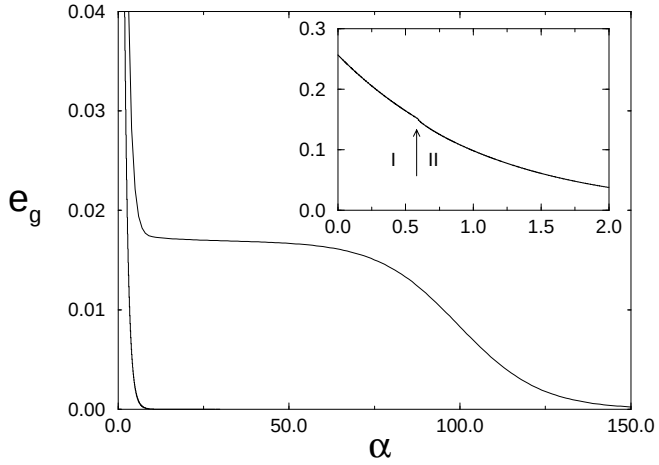


Figura 4.2: Curvas de aprendizagem para $M = 1$, $K = 2$: resultados analíticos no limite termodinâmico para o algoritmo localmente ótimo (curva inferior) e para o *backpropagation*. As condições iniciais são idênticas com $Q_{11} \in [0, .5]$, $Q_{22} \in [0, 1E-6]$ e $Q_{12} = \approx 0$. *Inset :* Cicatriz da mudança de regime dinâmico (ver texto).

Aqui $F_{M=K=1}$ indica a função modulação ótima para um percéptron aprendendo outro. Este algoritmo apenas elimina o ramo com vetor sináptico de menor norma, transformando a rede aluno em um percéptron. Analisando a expressão (4.15) é fácil notar que para garantir que os autovalores não sejam negativos precisamos fazer que:

$$\left(\frac{\partial e_g}{\partial Q_{12}}\right)^2 \leq 4\frac{\partial e_g}{\partial Q_{11}}\frac{\partial e_g}{\partial Q_{22}} \tag{4.17}$$

Na prática, só podemos escolher valores para os *overlaps* entre os ramos do aluno (Q_{ik}). Considerando que os vetores sinápticos iniciais são gerados uniformemente, tipicamente teremos $R_{kn} \approx 0$. Sob estas condições a desigualdade (4.17) não pode ser obedecida. Isto indica que o transiente **I** é inevitável quando não temos conhecimento *a priori* sobre o professor ($R_{kn} = 0$).

A curva de aprendizagem para condições iniciais aleatórias está representada na (Fig. 4.2) juntamente com com a curva obtida utilizando o algoritmo *backpropagation*. O platô simétrico foi totalmente eliminado. No detalhe mostramos a pequena cicatriz que marca a transição do regime **I** (um autovalor negativo) para o **II** (dois autovalores positivos).

Na (Fig. 4.3) mostramos a evolução dos *overlaps* entre os ramos do aluno Q_{ik}. Nota-se que no início da fase **II** os *overlaps* adquirem o padrão $Q_{11} \approx Q_{22}$ e$Q_{12} \approx \sqrt{Q_{11}Q_{22}}$ e os autovalores da hessiana funcional passam a ser positivos. A evolução dos *overlaps* professor-aluno é mostrada na (Fig. 4.4).

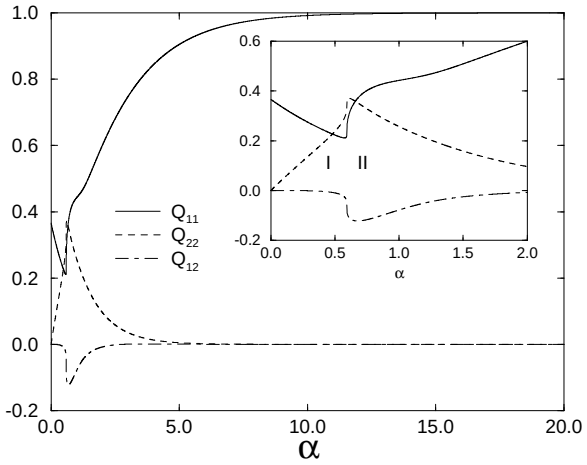


Figura 4.3: Evolução dos Q_{ik} para $M = 1$, $K = 2$ para condições iniciais aleatórias com $Q_{11} \in [0, .5]$, $Q_{22} \in [0, 1E-6]$ e $Q_{12} \approx 0$. *Inset*: Detalhe da transição entre os regimes **I** e **II**.

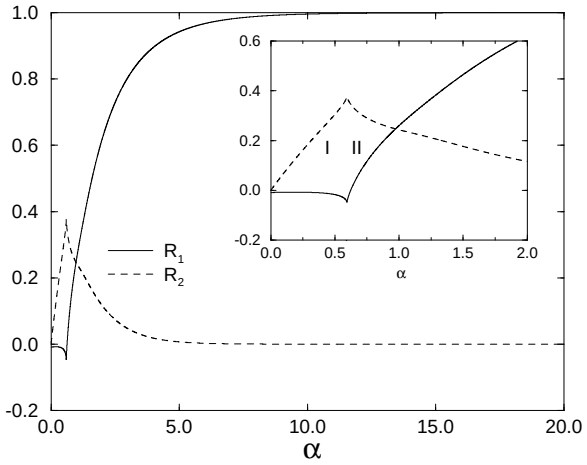


Figura 4.4: Evolução dos R_{kn} para $M = 1$, $K = 2$ para condições iniciais aleatórias com $Q_{11} \in [0, .5]$, $Q_{22} \in [0, 1E-6]$ e $Q_{12} \approx 0$. *Inset*: Detalhe da transição entre os regimes **I** e **II**.

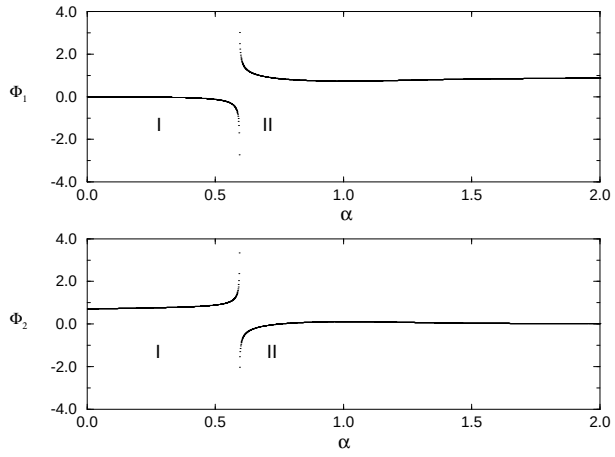


Figura 4.5: Integração numérica para a dinâmica das modulações Φ_i no limite termodinâmico. Na fase **II** um dos ramos assume todo o processamento: $\Phi_1 \to 1$ e $\Phi_2 \to 0$.

A estratégia de aprendizagem utilizada pode ser melhor compreendida olhando para a dinâmica das sinápses, que tem a forma:

$$\mathbf{J}_i(\mu+1) = \mathbf{J}_i(\mu) + \frac{1}{N}\left(H_{ik}^{(-1)} G_k \left\langle b(\mu)\right\rangle_{\mathcal{H}|\mathcal{V}} - h_i(\mu)\right)\mathbf{S}(\mu). \tag{4.18}$$

O campo pós-treinamento, definido por $\lambda_k(\mu) \equiv \mathbf{J}_k(\mu+1)\cdot\mathbf{S}(\mu)$, adquire a forma:

$$\lambda_i(\mu) = H_{ik}^{(-1)} G_k \left\langle b(\mu)\right\rangle_{\mathcal{H}|\mathcal{V}}. \tag{4.19}$$

O algoritmo ótimo se baseia em correlacionar as configurações sinápticas do aluno com aquelas do professor, fazendo com que após a apresentação os campos sejam dados por estimativas dos campos do professor.

Na (Fig. 4.5) mostramos a evolução dos $\Phi_i \equiv H_{ik}^{(-1)} G_k$ para o caso $M = 1$ e $K = 2$. No regime **I** há uma tentativa de representação do professor utilizando os dois ramos do aluno. Já no regime **II** um dos ramos é "cortado" e a redundância de arquitetura eliminada.

A discontinuidade nas modulações pode ser explicada se lembrarmos que:

$$\Phi_i \equiv H_{ik}^{(-1)} G_k = \frac{\left(Cof^T(\mathbf{H})\right)_{ik}}{\prod_i \nu_i} G_k, \tag{4.20}$$

onde $Cof(\mathbf{H})$ é a matriz dos cofatores da matriz $\mathbf{H}$ e ν_i são os autovalores desta mesma matriz. Quando um dos autovalores é nulo há uma singularidade associada.

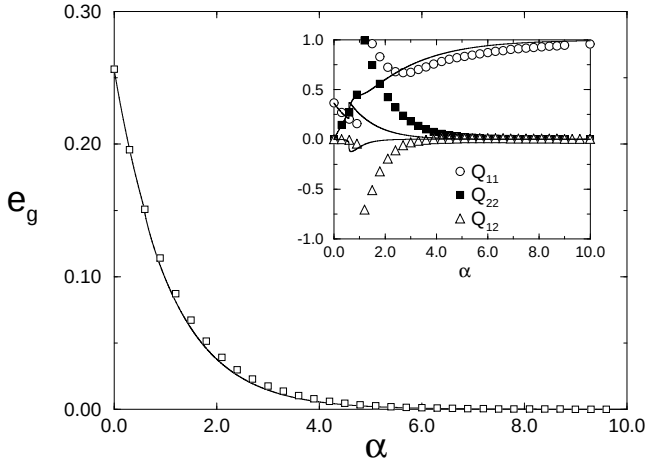


Figura 4.6: Comparação com experimentos numéricos. Símbolos: simulações com $N = 5000$ e condições iniciais aleatórias dadas por $Q_{11} \in [0, .5]$, $Q_{22} \in [0, 1E-6]$ e $Q_{12} \approx 0$. Linhas cheias: resultados analíticos para tamanho infinito e mesmas condições iniciais. *Inset*: Evolução dos Q_{ik} na simulação (símbolos) e resultados analíticos.

A comparação das curvas de aprendizagem analíticas com simulações mostra boa concordância, como pode se verificado na (Fig. 4.6). No *inset* desta figura mostramos a evolução dos *overlaps* Q_{ik} em uma simulação com tamanho $N = 5000$ e comparamos com o resultado analítico para condições iniciais rigorosamente idênticas. É notável que na região de transição entre os regimes **I** e **II** há discordância entre os resultados. Isto se deve a grandes flutuações que surgem na vizinhaça imediata da região onde os Φ's são singulares (ver Apêndice B).

4.3 Caso Realizável: $M = K$

Neste caso a função modulação ótima se escreve:

$$F_j(\mathcal{V}) = H^{(-1)}_{jk} G_{kn} \langle b_n \rangle_{\mathcal{H}|\mathcal{V}} - h_j. \tag{4.21}$$

As equações ficam:

$$\begin{aligned} \dot{R}_{kn} &= H^{(-1)}_{kj} G_{jm} \left\langle b_n \langle b_m \rangle_{\mathcal{H}|\mathcal{V}} \right\rangle - R_{kn} \\ \dot{Q}_{jk} &= H^{(-1)}_{jl} G_{ln} H^{(-1)}_{ki} G_{im} \left\langle \langle b_n \rangle_{\mathcal{H}|\mathcal{V}} \langle b_m \rangle_{\mathcal{H}|\mathcal{V}} \right\rangle - Q_{jk}. \end{aligned} \tag{4.22}$$

A principal dificuldade técnica no cálculo tanto da função modulação quanto das equações da dinâmica é a avaliação das estimativas $\langle b_n \rangle_{\mathcal{H}|\mathcal{V}}$. Num caso onde se tem

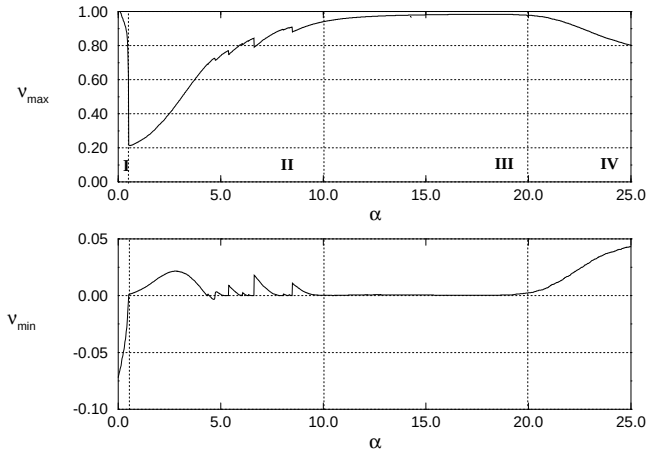


Figura 4.7: Autovalores da hessiana funcional. Fase **I**: um dos autovalores é negativo e a otimização não é válida. Fase **II**: estabelecimento do estado simétrico, o sistema apresenta flutuações grandes. Fase **III**: fase de platô. Fase **IV**: especialização dos ramos.

acesso à saída sem ruído Σ_B do professor e aos campos h_j temos que $\mathcal{V} = \{h_j, \Sigma_B\}$ e a média adquire a forma :

$$\langle b_n \rangle_{\mathcal{H}|\mathcal{V}} = \frac{1}{\mathcal{N}} \int_{-\infty}^{+\infty} \prod_{m=1}^{M} db_m \, b_n \, \delta \left(\Sigma_B - \sum_{m=1}^{M} g(b_m) \right) P(\{b_m, h_k\}) \tag{4.23}$$

Onde:

$$\mathcal{N} = \int_{-\infty}^{+\infty} \prod_{m=1}^{M} db_m \, \delta \left(\Sigma_B - \sum_{m=1}^{M} g(b_m) \right) P(\{b_m, h_k\})$$

e

$$P(\{b_m, h_k\}) = exp \left(-\frac{1}{2} \{b_m, h_k\} \mathbf{C}^{(-1)} \{b_m, h_k\}^T \right),$$

com a matriz de correlação

$$\mathbf{C} = \begin{pmatrix} M_{nm} & R_{kn} \\ R_{kn}^T & Q_{jk} \end{pmatrix}.$$

Nos restringindo agora ao caso mais simples com $M = K = 2$ [54, 55] temos que a avaliação das estimativas $\langle b_1 \rangle_{\{b_1,b_2\}|\{\Sigma_B,h_1,h_2\}}$ e $\langle b_2 \rangle_{\{b_1,b_2\}|\{\Sigma_B,h_1,h_2\}}$ envolve o cálculo de integrais da forma:

$$\tilde{b}_n^{(\epsilon)} = \int db_1 db_2 \, b_n^{\epsilon} \, \delta \left(\Sigma_B - g(b_1) - g(b_2) \right) P(\{b_1, b_2, h_1, h_2\}) \tag{4.24}$$

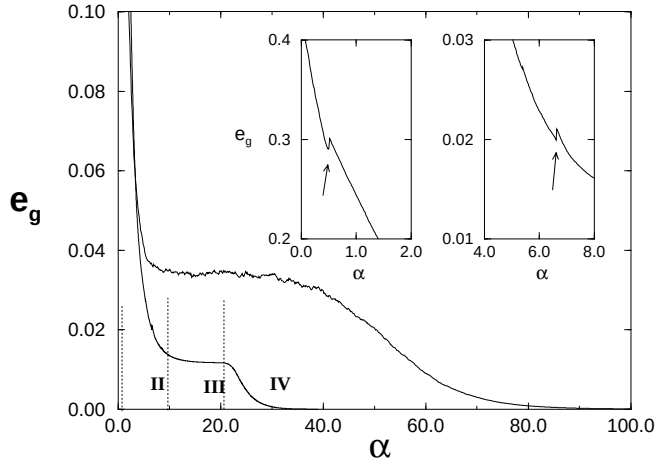


Figura 4.8: Simulações com $N = 5000$ para M=K=2 e condições iniciais aleatórias dadas por $Q_{11} \in [0, .5]$, $Q_{22} \in [0, 1E-6]$ e $Q_{12} \approx 0$. Fase **II**: Estabelecimento da fase simétrica. Fase **III**: platô simétrico. Fase **IV**: especialização *Insets*: Cicatrizes devido a grandes flutuações na fase **II**.

Com $\epsilon = 0, 1$. Dessa maneira:

$$\langle b_1 \rangle_{\{b_1,b_2\}|\{\Sigma_B,h_1,h_2\}} = \frac{\tilde{b}_n^{(1)}}{\tilde{b}_n^{(0)}}. \tag{4.25}$$

Estas integrais são unidimensionais devido ao vínculo imposto pela distribuição δ. A principal dificuldade no cálculo é justamente imposta por este vínculo. Uma maneira de simplificar o cálculo é introduzindo a transformação:

$$\begin{aligned} \sigma_1 &= g(b_1) + g(b_2) \\ \sigma_2 &= g(b_1) - g(b_2). \end{aligned} \tag{4.26}$$

Ficamos então com :

$$\tilde{b}_1^{(\epsilon)} = \int d\sigma_1 d\sigma_2 \left| \frac{\partial(b_1, b_2)}{\partial(\sigma_1, \sigma_2)} \right| \left(g^{(-1)} \left(\frac{\sigma_1 + \sigma_2}{2} \right) \right)^{\epsilon} \delta\left(\Sigma_B - \sigma_1\right) P(\{\sigma_1, \sigma_2, h_1, h_2\}) \tag{4.27}$$

O vínculo é agora facilmente implementado e os limites de integração são redefinidos:

$$\tilde{b}_1^{(\epsilon)} = \int_{|\Sigma_B|-2}^{|\Sigma_B|+2} d\sigma_2 \left| \frac{\partial(b_1, b_2)}{\partial(\sigma_1, \sigma_2)} \right|_{\sigma_1 = \Sigma_B} \left(g^{(-1)} \left(\frac{\Sigma_B + \sigma_2}{2} \right) \right)^{\epsilon} P(\{\sigma_2, h_1, h_2, \Sigma_B\}). \tag{4.28}$$

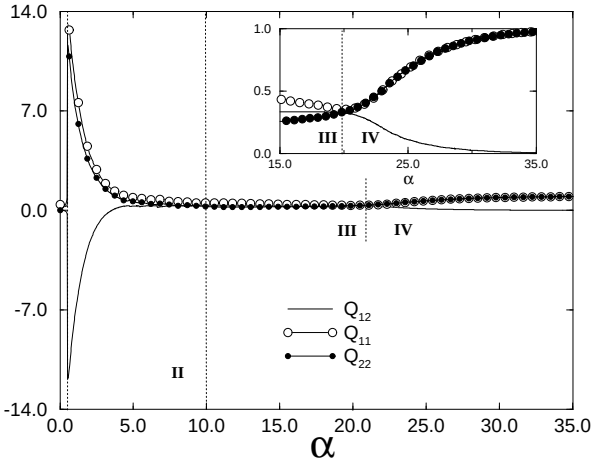


Figura 4.9: Simulações com $N = 5000$ e condições iniciais aleatórias dadas por $Q_{11} \in [0, .5]$, $Q_{22} \in [0, 1E-6]$ e $Q_{12} \approx 0$. *Overlaps* entre os ramos do aluno. *Inset*: saída do platô.

De maneira análoga chegamos a:

$$\tilde{b}_2^{(\epsilon)} = \int_{|\Sigma_B|-2}^{|\Sigma_B|+2} d\sigma_2 \left| \frac{\partial(b_1, b_2)}{\partial(\sigma_1, \sigma_2)} \right|_{\sigma_1 = \Sigma_B} \left(g^{(-1)} \left(\frac{\Sigma_B - \sigma_2}{2} \right) \right)^{\epsilon} P(\{\sigma_2, h_1, h_2, \Sigma_B\}). \tag{4.29}$$

Nas integrais $\left| \frac{\partial(b_1, b_2)}{\partial(\sigma_1, \sigma_2)} \right|$ simboliza o determinante da matriz jacobiana definida por:

$$\frac{\partial(b_1, b_2)}{\partial(\sigma_1, \sigma_2)} = \begin{pmatrix} \frac{\partial b_1}{\partial \sigma_1} & \frac{\partial b_1}{\partial \sigma_2} \\ \frac{\partial b_2}{\partial \sigma_1} & \frac{\partial b_2}{\partial \sigma_2} \end{pmatrix}. \tag{4.30}$$

O determinante da matriz acima pode ser facilmente calculado resultando em:

$$\left| \frac{\partial(b_1, b_2)}{\partial(\sigma_1, \sigma_2)} \right| = \frac{\pi}{4} \exp \frac{1}{2} \left[\left(g^{(-1)} \left(\frac{\sigma_1 + \sigma_2}{2} \right) \right)^2 + \left(g^{(-1)} \left(\frac{\sigma_1 - \sigma_2}{2} \right) \right)^2 \right] \tag{4.31}$$

Nesta forma a integração numérica não oferece nenhuma dificuldade especial. Na (Fig. 4.8) mostramos a curva de aprendizagem obtida numa simulação do algoritmo ótimo para $N = 5000$, mostramos na mesma figura, para as mesmas condições iniciais, mesmo tamanho e mesma seqüência de exemplos, a curva de aprendizagem para o *backpropagation*.

Na (Fig. 4.7) mostramos os autovalores da hessiana funcional dados por (4.15). Nesta figura notam-se quatro regimes razoavelmente bem definidos. Na fase **I** um dos autovalores é negativo, este é o transiente que já apareceu no caso $M = 1$, $K = 2$

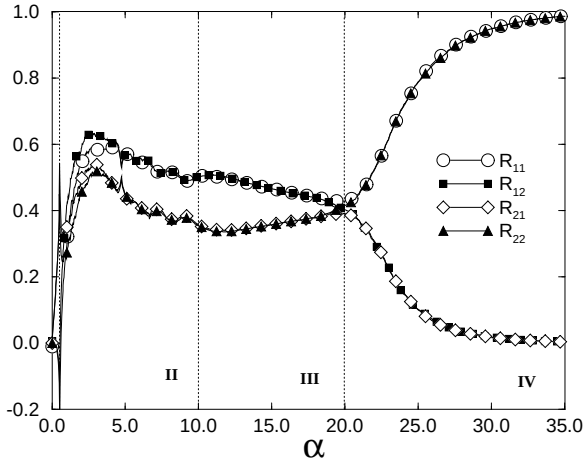


Figura 4.10: Simulações com $N = 5000$ e condições iniciais aleatórias dadas por $Q_{11} \in [0, .5]$, $Q_{22} \in [0, 1E-6]$ e $Q_{12} \approx 0$. *Insets*: *Overlaps* professor-aluno.

indicando a existência de algoritmos de aprendizado de melhor desempenho nesta fase.

Na fase **II** há o estabelecimento de um platô simétrico com cada ramo do aluno representando identicamente os dois ramos do professor, como pode ser visto na (Fig. 4.10). A fase **II** também é marcada por grandes flutuações no sistema de modulação Φ_{kn} das estimativas de campo mostradas na (Fig. 4.11). Este sistema de modulação é definido como na seção anterior, por :

$$\lambda_k(\mu) = \Phi_{k1}\langle b_1(\mu)\rangle_{\{b_n|h_k,\Sigma_B\}} + \Phi_{k2}\langle b_2(\mu)\rangle_{\{b_n|h_k,\Sigma_B\}}, \tag{4.32}$$

onde λ_k é o campo pós-sináptico após a apresentação de um exemplo. Desta forma, os campos pós-apresentação são misturas de estimativas dos campos do professor. Estas flutuações provocam o aparecimento de cicatrizes na curva de aprendizagem (Fig. 4.8) e ocorrem quando a rede aluno passa pela vizinhança de singularidades , no espaço dos estados macroscópicos, dos Φ_{kn}.

Na fase **III** o platô simétrico já está bem estabelecido como pode ser visto na (Fig. 4.8), na (Fig. 4.9) e na (Fig. 4.10). Neste platô um dos autovalores da matriz hessiana **H** tem valor nulo (Fig. 4.7), o que, segundo o (Capítulo 3), indica que há uma família de algoritmos com comportamento de platô. Em particular, isso pode sugerir que o platô seja robusto com relação as estimativas dos campos do professor.

Na fase **IV** ocorre a especialização dos ramos com cada ramo do aluno aprendendo apenas um ramo do professor, temos então que $\Phi_{kn} \to \delta_{kn}$ (Fig. 4.11) conforme o sistema se aproxima desta fase. Na (Fig. 4.9) e na (Fig. 4.10) nota-se que a especialização começa a partir do estado totalmente simétrico $R_{kn} = R$ e $Q_{jk} = Q$

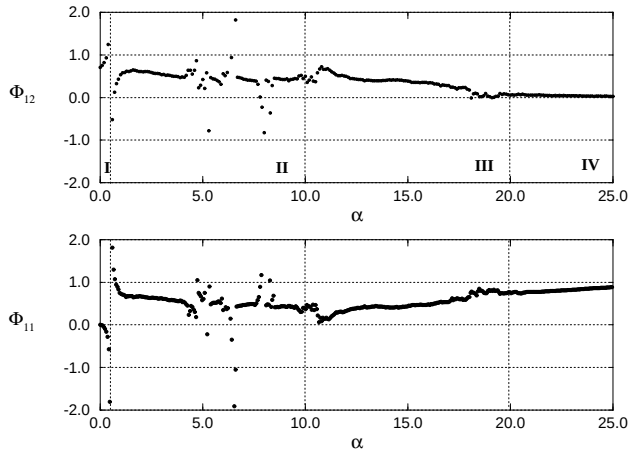


Figura 4.11: Simulações com $N = 5000$ e condições iniciais aleatórias dadas por $Q_{11} \in [0, .5]$, $Q_{22} \in [0, 1E-6]$ e $Q_{12} \approx 0$. Modulação das estimativas $\lambda_k \sum_n \Phi_{kn} \langle b_n \rangle_{\{b_n|h_k,\Sigma_B\}}$.

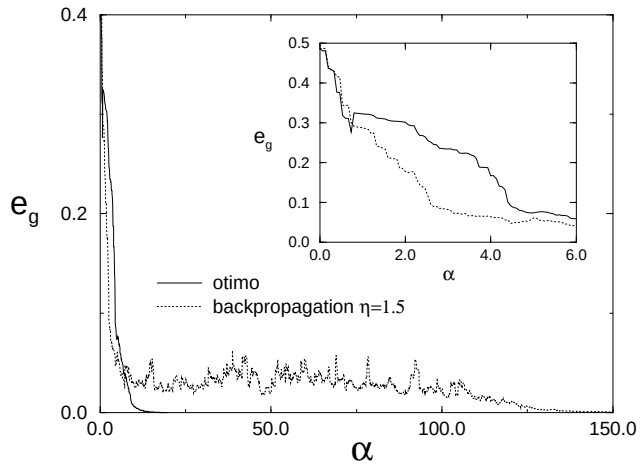


Figura 4.12: Simulações com $N = 15$ e condições iniciais aleatórias dadas por $Q_{11} \in [0, .5]$, $Q_{22} \in [0, 1E-6]$ e $Q_{12} \approx 0$. Comparação entre o algoritmo ótimo no limite termodinâmico e o *backpropagation com* $\eta = 1.5$. *Inset*: detalhe do início da aprendizagem.

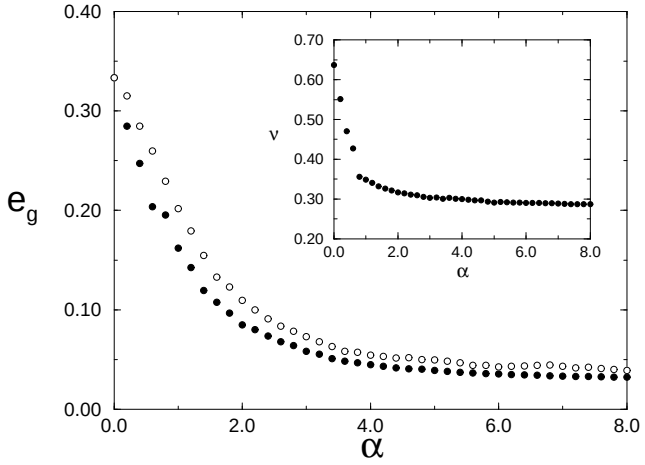


Figura 4.13: Simulações do caso $M = 2$, $K = 1$ com $N = 500$ e condição inicial aleatória dada por $Q \in [0, 1E-6]$. Curva superior: backpropagation com $\eta = 1.5$. Curva inferior: ótimo. *Inset*: autovalor da hessiana $\mathbf{H}$. As curvas evoluem para exatamente o mesmo valor assintótico com $e_g > 0$.

e segue por estados do tipo $R_{11} = R_{22} = R$, $R_{12} = R_{21} = \tilde{R}$ e $Q_{11} = Q_{22} = Q$, para os quais $\Phi_{kn} = \delta_{kn}$.

Apesar da otimização ter sido realizada no limite termodinâmico, seu emprego em sistemas de tamanho tão pequeno como $N = 15$ exibe desempenho bastante superior, principalmente no platô, como pode ser visto na (Fig. 4.12). Na fase não-ótima, no início da aprendizagem o *backpropagation* apresenta desempenho superior.

4.4 Caso Não-realizável: $M > K$

No caso não-realizável mais simples, ou seja, quando um percéptron aprende uma rede multicamada temos que a função modulação ótima adquire a forma:

$$F(\mathcal{V}) = \left(\frac{\partial e_g}{\partial Q}\right)^{-1} G_n \left\langle b_n \right\rangle_{\mathcal{H}|\mathcal{V}} - h. \tag{4.33}$$

A dinâmica é descrita por:

$$\begin{aligned} \dot{R}_n &= \left(\frac{\partial e_g}{\partial Q}\right)^{-1} G_m \left\langle \left\langle b_m \right\rangle_{\mathcal{H}|\mathcal{V}} b_n \right\rangle - R_n \\ \dot{Q} &= \left(\frac{\partial e_g}{\partial Q}\right)^{-2} G_n G_m \left\langle \left\langle b_n \right\rangle_{\mathcal{H}|\mathcal{V}} \left\langle b_m \right\rangle_{\mathcal{H}|\mathcal{V}} \right\rangle - Q. \end{aligned} \tag{4.34}$$

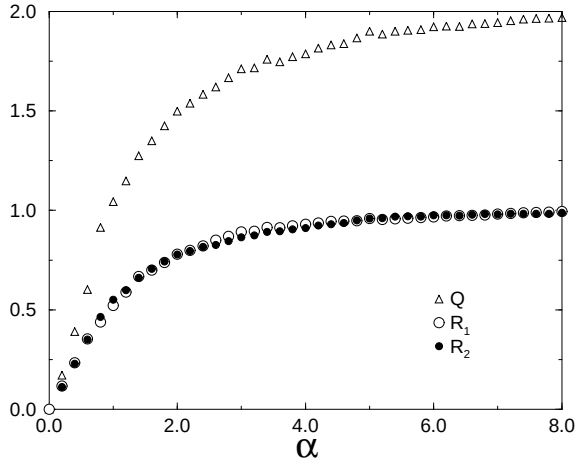


Figura 4.14: Simulações com $N = 500$ e condição inicial aleatória dada por $Q \in [0, 1E-6]$. Dinâmica dos *overlaps*.

Para o caso $M = 2$ as estimativas $\langle b_m \rangle_{\mathcal{H}|\mathcal{V}}$ podem ser calculadas, de maneira similar ao caso realizável, através das integrais:

$$\tilde{b}_1^{(\epsilon)} = \int_{|\Sigma_B|-2}^{|\Sigma_B|+2} d\sigma_2 \left| \frac{\partial(b_1, b_2)}{\partial(\sigma_1, \sigma_2)} \right|_{\sigma_1 = \Sigma_B} \left(g^{(-1)} \left(\frac{\Sigma_B + \sigma_2}{2} \right) \right)^{\epsilon} P(\{\sigma_2, h, \Sigma_B\}), \quad (4.35)$$

$$\tilde{b}_2^{(\epsilon)} = \int_{|\Sigma_B|-2}^{|\Sigma_B|+2} d\sigma_2 \left| \frac{\partial(b_1, b_2)}{\partial(\sigma_1, \sigma_2)} \right|_{\sigma_1 = \Sigma_B} \left(g^{(-1)} \left(\frac{\Sigma_B - \sigma_2}{2} \right) \right)^{\epsilon} P(\{\sigma_2, h, \Sigma_B\}). \quad (4.36)$$

Como há uma única variável macroscópica (Q) descrevendo os *overlaps* do aluno a matriz hessiana $\mathbf{H}$ é representada por:

$$H = \nu = \frac{\partial e_g}{\partial Q}, \quad (4.37)$$

onde ν representa o autovalor único da "matriz" $\mathbf{H}$. A condição suficiente para otimização adquire então a forma simples:

$$\frac{\partial e_g}{\partial Q} > 0. \quad (4.38)$$

Na (Fig. 4.13) mostramos a curva de aprendizagem do algoritmo ótimo em comparação com a curva do *backpropagation* para as mesmas condições iniciais. No *inset* mostramos ν sempre positivo, garantindo a validade da otimização. Na (Fig. 4.14) mostramos que os *overlaps* Q e R_k têm dinâmicas similares àquelas encontradas no (Capítulo 2) utilizando o *backpropagation* com o ponto fixo em $Q = 2$ e $R_k = 1$.

4.5 Otimização Global

M. Rattray e D. Saad propuseram recentemente [39] uma solução para o problema da otimização de algoritmos de aprendizagem no caso em que se tem conhecimento *a priori* do número de exemplos disponíveis. Desta forma é possível a otimização global do desempenho numa "janela" dada $[\alpha_0, \alpha_1]$. A situação pode ser formulada como um problema variacional onde se quer otimizar o funcional :

$$\Delta e_g = \int_{\alpha_0}^{\alpha_1} d\alpha \; \dot{e}_g. \tag{4.39}$$

Com os vínculos:

$$\dot{R}_{kn} = \langle F_k b_n \rangle \tag{4.40}$$

$$\dot{Q}_{kj} = \langle F_j h_k + F_k h_j + F_j F_k \rangle . \tag{4.41}$$

Esta otimização é equivalente à otimização do funcional [22]:

$$\begin{aligned} \mathcal{L}[F, \lambda_{kn}, \nu_{jk}] &= \dot{e}_g + \sum_{kn} \lambda_{kn} \left(\dot{R}_{kn} - \langle F_k b_n \rangle \right) \\ &+ \sum_{jk} \nu_{jk} \left(\dot{Q}_{kj} - \langle F_j h_k + F_k h_j + F_j F_k \rangle \right) . \end{aligned} \tag{4.42}$$

Para otimizarmos $\mathcal{L}$ primeiro encontramos o conjunto de funções modulação F através da condição :

$$\frac{\delta \mathcal{L}}{\delta F_k(\mathcal{V})}[F(\mathcal{V})] = 0. \tag{4.43}$$

Disso resulta:

$$F_k = -\frac{1}{2} {\nu_{kj}}^{-1} \lambda_{jn} \langle b_n \rangle_{\mathcal{H}|\mathcal{V}} - h_k, \tag{4.44}$$

onde somamos sobre os índices repetidos. Estas funções modulação são idênticas às obtidas via otimização local quando fazemos as escolhas:

$$\lambda_{jn} = \frac{\partial e_g}{\partial R_{jn}}, \tag{4.45}$$

$$\nu_{kj} = \frac{\partial e_g}{\partial Q_{kj}}. \tag{4.46}$$

No entanto, para fazermos a otimização global precisamos inserir as funções modulação (4.44) no funcional (4.43) e resolvermos as equações de Euler [22]:

$$\frac{d}{d\alpha} \frac{\partial \mathcal{L}}{\partial \dot{Q}_{jk}} - \frac{\partial \mathcal{L}}{\partial Q_{jk}} = 0, \tag{4.47}$$

$$\frac{d}{d\alpha} \frac{\partial \mathcal{L}}{\partial \dot{R}_{kn}} - \frac{\partial \mathcal{L}}{\partial R_{kn}} = 0. \tag{4.48}$$

Lembrando que

$$\dot{e}_g = \sum_{kj} \dot{Q}_{jk} \frac{\partial e_g}{\partial Q_{jk}} + \sum_{kn} \dot{R}_{kn} \frac{\partial e_g}{\partial R_{kn}}, \tag{4.49}$$

estas equações dão origem a :

$$\dot{\lambda_{jn}} = -\sum_{jn} \lambda_{jn} \frac{\partial \langle F_j b_n \rangle}{\partial R_{jn}} - \sum_{kj} \nu_{kj} \frac{\partial \langle F_k h_j + F_j h_k + F_j F_k \rangle}{\partial R_{jn}}, \tag{4.50}$$

$$\dot{\nu_{kj}} = -\sum_{jn} \lambda_{jn} \frac{\partial \langle F_j b_n \rangle}{\partial Q_{kj}} - \sum_{kj} \nu_{kj} \frac{\partial \langle F_k h_j + F_j h_k + F_j F_k \rangle}{\partial Q_{kj}}. \tag{4.51}$$

Estas equações juntamente com as equações (4.41) formam um sistema de equações diferenciais cuja solução é univocamente definida desde que sejam dadas as condições de contorno. Estas condições envolvem as condições iniciais $Q_{jk}(0)$ e $R_{kn}(0)$ e as condições:

$$\lambda_{kn}(\alpha_1) = \left. \frac{\partial e_g}{\partial R_{kn}} \right)_{\alpha_1}, \tag{4.52}$$

$$\nu_{kj}(\alpha_1) = \left. \frac{\partial e_g}{\partial Q_{kj}} \right)_{\alpha_1}, \tag{4.53}$$

que garantem que ao final da janela otimizada $[\alpha_0, \alpha_1]$ o algoritmo de aprendizagem corresponda àquele localmente ótimo. A solução deste problema pode ser obtida analiticamente em alguns poucos casos.

Em [39] é demonstrado que a otimização global leva a um resultado idêntico à local para percéptrons booleanos. No mesmo trabalho obtem-se o algoritmo globalmente ótimo para uma rede multicamada (K=3) com unidades contínuas e não-lineares (*erf*) aprendendo um percéptron também *erf* e sem ruído. Demonstra-se que nesta situação a otimização global leva a um algoritmo sensivelmente melhor que aquele obtido na otimização local.

Aplicaremos a seguir, como exemplo, este esquema à situação na qual tanto a rede professor quanto a rede aluno são percéptrons lineares. Nesta situação temos que a função modulação adquire a forma:

$$F = -\frac{1}{2} \nu^{-1} \lambda b - h. \tag{4.54}$$

Lembramos que, no caso sem ruído, temos acesso total ao campo b do professor. As equações para ν e λ são:

$$\dot{\lambda} = -\lambda \frac{\partial \langle Fb \rangle}{\partial R} - \nu \frac{\partial \langle 2Fh + F^2 \rangle}{\partial R}, \tag{4.55}$$

$$\dot{\nu} = -\lambda \frac{\partial \langle Fb \rangle}{\partial Q} - \nu \frac{\partial \langle 2Fh + F^2 \rangle}{\partial Q}. \tag{4.56}$$

Introduzindo (4.54) em (4.56) e fazendo $\langle b^2 \rangle = 1$ teremos:

$$\dot{\lambda} = -\lambda \frac{\partial}{\partial R}\left(-\frac{\lambda}{2\nu} - R\right) - \nu \frac{\partial}{\partial R}\left(\frac{\lambda^2}{4\nu^2} - Q\right), \tag{4.57}$$

$$\dot{\nu} = -\lambda \frac{\partial}{\partial Q}\left(-\frac{\lambda}{2\nu} - R\right) - \nu \frac{\partial}{\partial Q}\left(\frac{\lambda^2}{4\nu^2} - Q\right). \tag{4.58}$$

Calculando as derivadas chegamos finalmente às equações desacopladas:

$$\dot{\lambda} = \lambda, \tag{4.59}$$

$$\dot{\nu} = \nu. \tag{4.60}$$

Cujas soluções são simplesmente:

$$\lambda(\alpha) = C_\lambda e^\alpha, \tag{4.61}$$

$$\nu(\alpha) = C_\nu e^\alpha. \tag{4.62}$$

As constantes C_α e C_ν são obtidas utilizando as codições de contorno. Dessa forma teremos:

$$\frac{C_\lambda}{C_\nu} = \frac{\frac{\partial e_g}{\partial R}\Big)_{\alpha_1}}{\frac{\partial e_g}{\partial Q}\Big)_{\alpha_1}}. \tag{4.63}$$

Assim obtemos a função modulação :

$$F = -\frac{\frac{\partial e_g}{\partial R}\Big)_{\alpha_1}}{2\,\frac{\partial e_g}{\partial Q}\Big)_{\alpha_1}} b - h. \tag{4.64}$$

Calculando as derivadas chegamos finalmente a:

$$F = b - h, \tag{4.65}$$

que é exatamente o algoritmo otimizado localmente.

5

Conclusões e Perspectivas

Neste capítulo faremos um sumário das idéias e resultados que julgamos mais relevantes e discutiremos algumas possibilidades de futuros projetos.

5.1 Conclusões

O programa de otimização funcional de algoritmos deve ser visto, pelo menos num primeiro momento, como uma técnica de estudo que permite a construção de algoritmos de aprendizagem com desempenho ótimo. Este programa integra os esforços atuais da comunidade de Física Estatística para o estudo de sistemas físicos (ou biológicos) que processam informação. Guardadas as devidas proporções, este esforço compara-se àquele desenvolvido no passado sobre as máquinas térmicas.

Os algoritmos de desempenho ótimo explicitam que características devem ter algoritmos eficientes. Nem sempre estes algoritmos ótimos poderão ser aplicados diretamente em situações práticas e nem deve ser este o objetivo inicial [1]. Analisando as propriedades e o funcionamento destes algoritmos espera-se identificar os processos mais fundamentais envolvidos no processamento de informação por redes neurais (naturais ou artificiais). É claro que o caminho que precisa ser percorrido até redes de complexidade equivalente às biológicas é longo e passa pela identificação dos elementos mais básicos envolvidos. Invariavelmente os algoritmos ótimos têm exibido algumas propriedades chave que são candidatas naturais a integrarem a lista de elementos fundamentais :

- Os algoritmos ótimos são bastante especializados. Dado um certo conjunto $\mathcal{V}$ de informação acessível, seu desempenho é ótimo apenas neste cenário específico.
- Os algoritmos exibem alguma robustez, mantendo seu desempenho quando são efetuadas mudanças ambientais de amplitude determinada.
- A modulação evolui com o tempo (processo de *annealing*).

[1] Da mesma forma, em situações práticas não se utilizam ciclos de Carnot.

- A função modulação não apenas se utiliza da informação dos exemplos para corrigir seus erros, mas também utiliza esta informação para produzir estimativas de grandezas desconhecidas (campos pós-sinápticos).

Os cenários que estudamos, envolvendo redes multicamada totalmente conectadas e com unidades contínuas, são marcados pelo surgimento de uma fenomenologia nova dentro do espectro de comportamentos já observados em outras situações: platôs nas curvas de aprendizagem. Estes platôs surgem devido às múltiplas representações internas que efetuam a mesma computação[2]. Numa rede multicamada totalmente conectada com M percéptrons na camada escondida todas as $M!$ permutações de $(\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, ..., \mathbf{B}_M)$ efetuam exatamente a mesma computação. Os platôs estabelecem-se devido a liberdade que cada ramo do aluno tem para "escolher" qual (ou quais) ramo (ou ramos) do professor representar. No início da dinâmica de aprendizado, cada ramo acaba por representar uma média dos ramos do professor (estado simétrico). A saída deste estado simétrico (ou de platô) ocorre com a emergência de um estado especializado, que especializa cada ramo, adotando uma permutação específica e provocando a quebra de simetria. No *backpropagation* a emergência do estado especializado a partir do estado de platô (ao qual chamaremos simétrico) pode ser entendida indentificando a presença de dois pontos fixos no sistema dinâmico que descreve o aprendizado. O ponto fixo que corresponde ao platô tem uma única direção estável (que corresponde à simetria do platô) e uma grande bacia de atração. Conforme a rede aprende fatalmente ruma em direção a este ponto fixo. Esta é a fase simétrica. Se a rede iniciar o aprendizado com os ramos diferenciados, ela passará pela vizinhança do ponto fixo de platô numa trajetória levemente repulsiva devido às direções instáveis. O sistema então entrará da bacia de atração de um ponto fixo que corresponde a uma permutação da representação do professor. Esta é a fase especializada.

A eficiência de um algoritmo em uma rede multicamada depende da sua capacidade de estimular representações internas apropriadamente especializadas. Os algoritmos obtidos nesta dissertação, como o esperado, utilizam estimativas para os campos pós-sinápticos do professor $\langle b_n \rangle_{\mathcal{H}|\mathcal{V}}$ e variam com o tempo. As estimativas são feitas com base na informação acessível (a saber: os campos do aluno h_k e a saída fornecida pelo professor Σ_B) e nas correlações (ou na hipótese que se faz sobre elas) (Q_{jk} e R_{kn}) entre as representações do aluno e do professor. A forma que estes algoritmos têm é :

$$\lambda_k = \Phi_{k1}(Q_{jk}, R_{kn})\langle b_1 \rangle_{\mathcal{H}|\mathcal{V}} + ... + \Phi_{kM}(Q_{jk}, R_{kn})\langle b_M \rangle_{\mathcal{H}|\mathcal{V}},$$

onde λ_k representa o campo no ramo k do aluno após a apresentação do exemplo (h_k representa o campo antes da apresentação. O que se vê é que o algoritmo substitui os campos atuais por uma combinação linear das estimativas dos campos do professor. A maneira como os campos do professor são combinados depende das correlações entre as redes. Se não há correlação alguma, como deve ocorrer no

[2]Por computação aqui entendemos o mapa entrada→saída.

início da aprendizagem, as misturas deverão tender a ser homogêneas com todos os ramos representando uma média das estimativas (fase simétrica). O que torna o algoritmo ótimo eficiente é sua capacidade de estimular a especialização correta dos ramos. Conforme o padrão de correlações tende para representação corretamente especializada ($Q_{kj} = Q\delta_{kj} + \tilde{Q}(1 - \delta_{kj})$ e $R_{kn} = R\delta_{kn} + \tilde{R}(1 - \delta_{kn})$) as misturas de estimativas reforçam esta tendência fazendo $\Phi_{kn} \to \delta_{kn}$ e induzindo a rápida quebra de simetria.

A cenário que estudamos está longe de ter sido esgotado. Nas seções seguintes tentaremos enumerar alguns estudos a serem realizados imediatamente, a curto prazo (Perspectivas I) e a longo prazo (Perspectivas II).

5.2 Perspectivas I: Otimização Funcional

Os estudos que efetuamos aqui ainda precisam ser complementados, alguns projetos de realização imediata são:

- Descrição mais aprofundada dos mecanismos microscópicos envolvidos na aprendizagem ótima.
- Otimização nas situações $M = 1$, $K = 3$ e $M = 1$, $K = 4$. Nestas situações há vários padrões de quebra de simetria possíveis. De fato, o caso $M = 1$ e $K = 3$ já foi resolvido em [39] mostrando a presença de um platô mais longo.
- Integração numérica das equações diferenciais para o caso $M = K = 2$ no limite termodinâmico. Isso nos permitiria tirar dúvidas com respeito aos efeitos de tamanho finito presentes em nossas simulações (principalmente na fase **II** da dinâmica).
- Otimização das mesmas situações abordadas mas para arquiteturas multicamada totalmente conectadas com neurônios booleanos. Isto nos permitiria checar como as estratégias de aprendizagem observadas em outras arquiteturas booleanas se apresentariam.
- Otimização de redes multicamada com unidades contínuas e não-lineares, mas com professores ruidosos. Poderíamos aqui construir diagramas de robustez [16].
- Estudo de estimadores rápidos para os campos do professor b_n.
- Estudo de estimadores para as correlações professor-aluno R_{kn}.

5.3 Perspectivas II: Aprendizado em Redes Neurais

A principal diferença do enfoque físico com relação aos outros enfoques (biológico e psicológico) em se tratando de sistemas com capacidade de processar informação é sua ênfase nos processos. Dessa forma, um "neurônio" num modelo físico de rede neural não corresponde a um neurônio de uma rede biológica, mas são modelos para processos computacionais específicos realizados por sistemas extremamente complexos de neurônios biológicos. Acreditamos que um grande desafio atualmente é compreender aspectos de processos computacionais biológicos utilizando estes modelos. Um exemplo deste enfoque é o recente artigo de Caticha e Kinouchi [12] onde é proposto que a ordem temporal na qual subsistemas do cérebro (no caso a amigdala e o lobo pré-frontal) surgiram durante a evolução natural seria regida por leis gerais de ordenação relacionadas à otimização da utilização de informação do ambiente.

Uma outra frente de estudos bastante promissora é a formalização da estrutura matemática subjacente aos modelos de aprendizado *online* ótimo. O que sabemos até o momento é que os algoritmos ótimos fazem atualizações da representação interna da rede utilizando a informação contida nos exemplos e na própria representação. A melhor maneira de realizar este tipo de tarefa, do ponto de vista estatístico, é utilizando estimativas bayesianas [11]. Alguns trabalhos tentando propor uma descrição bayesiana do aprendizado tem sido recentemente propostos para o percéptron [36, 59].

Qualquer que seja o futuro da física das redes neurais, não há dúvidas de que, nos últimos anos, ela tem contribuído para a inserção da fenomenologia do processamento de informação no repertório da Física contemporânea.

A

Backpropagation Genérico

Discutiremos com detalhes neste apêndice como se implementa o algoritmo back-propagation para uma arquitetura *feedforward* com "M" camadas, sendo que cada unidade "k" da camada específica "m" possui função de transferência "$g_k^{(m)}$".

Comecemos analisando um único neurônio, como representado na (Fig. A.1). Este neurônio é identificado como neurônio "k" da camada "m". Cada sinápse do neurônio "mk" é denotada $J_{ik}^{(m)}$. O potencial presente na saída é denotado $\sigma_k^{(m)}$. Os potenciais de entrada, que podem corresponder às saídas de outros neurônios, são simbolizados consistentemente por $\sigma_i^{(m-1)}$. Definimos o campo pós-sináptico do neurônio "k" da camada "m" por:

$$h_k^{(m)} = \sum_i J_{ki}^{(m)} \sigma_i^{(m-1)}. \tag{A.1}$$

Para as camadas de entrada e saída da rede os potenciais são dados, daí temos as condições de contorno:

- Camada de saída: $\sigma_k^{(M)} = \Sigma_k$.
- Camada de entrada: $\sigma_i^{(1)} = S_i$.

De imediato podemos escrever a saída do neurônio em função de sua entrada:

$$\begin{aligned}
\sigma_k^{(m)} &= g_k^{(m)} \left(h_k^{(m)} \right) && \text{(A.2)} \\
&= g_k^{(m)} \left(\sum_i J_{ik}^{(m)} \sigma_i^{(m-1)} \right) && \text{(A.3)} \\
&= g_k^{(m)} \left(\sum_i J_{ik}^{(m)} g_i^{(m-1)} \left(h_i^{(m-1)} \right) \right) && \text{(A.4)}
\end{aligned}$$

Os pares ordenados entrada-saída de exemplos são definidos por $(\mathbf{S}(\mu), \Sigma_n^0(\mu))$. Com base nestes pares define-se o conjunto de treinamento $\mathcal{L} \equiv \{(\mathbf{S}(\mu), \Sigma_n^0(\mu)), \mu = 1, ..., p\}$ e a energia :

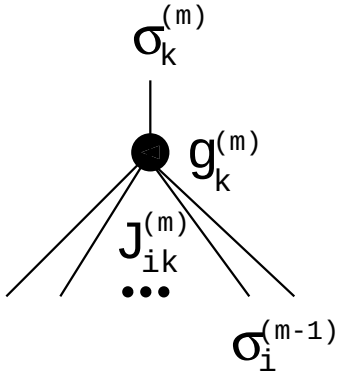


Figura A.1: Neurônio integrante de uma rede multicamada.

$$E = \frac{1}{2}\sum_{k,\mu}\left(\Sigma_k^0(\mu) - \Sigma_k(\mu)\right)^2 \tag{A.5}$$

O "aprendizado" consiste na minimização desta energia através de modificações nos $\mathbf{J}_{ik}^m$. Esta minimização pode ser efetuada de maneira bastante convencional implementando a dinâmica de *gradient descent*:

$$\Delta J_{ik}^{(m)} = -\eta\frac{\partial E}{\partial J_{ik}^{(m)}} \tag{A.6}$$

Onde η é um parâmetro normalmente denominado "taxa de aprendizagem".

Substituindo (A.2) em (A.5) e lembrando da condição de contôrno para a saída podemos reescrever a energia E na forma:

$$E = \frac{1}{2}\sum_{k,\mu}\left(\Sigma_k^0(\mu) - g_k^{(m)}\left(\sum_i J_{ki}^{(M)}\sigma_i^{(M-1)}\right)\right)^2. \tag{A.7}$$

Ficando claro o caráter recorrente da expressão para energia. O cálculo da derivada parcial de (A.6) para as sinápses $J_{ik}^{(M)}$ da camada mais externa é relativamente trivial, resultando que[1]:

$$\begin{aligned}\Delta J_{ik}^{(M)} &= -\eta\sum_{\mu}\left(\Sigma_k^0(\mu) - \Sigma_k(\mu)\right)\left[g_k^{(M)}\right]'(h_k^{(M)})\sigma_i^{(M-1)} &\text{(A.8)}\\ &= \eta\sum_{mu}\delta_k^{(M)}(\mu)\sigma_i^{(M-1)}. &\text{(A.9)}\end{aligned}$$

Onde definimos o "erro" $\delta_k^{(M)}(\mu) \equiv (\Sigma_k^0(\mu) - \Sigma_k(\mu))\left[g_k^{(M)}\right]'(h_k^{(M)})$.

[1] Como no texto g' simboliza a derivada de g.

Para a próxima camada mais interna "$M-1$" temos que calcular as derivadas em relação às sinápses $J_{li}^{(M-1)}$. É fácil perceber observando (A.2) e (A.7) que a dependência da energia com estas sinápses se dá através dos potenciais $\sigma_i^{(M-1)}$ na forma:

$$\sigma_i^{(M-1)} = g_i^{(M-1)} \left(\sum_l J_{li}^{(M-1)} \sigma_l^{(M-2)} \right) \tag{A.10}$$

Assim, escreveremos :

$$\frac{\partial E}{\partial J_{li}^{(M-1)}} = \frac{\partial E}{\partial \sigma_i^{(M-1)}} \frac{\partial \sigma_i^{(M-1)}}{\partial J_{li}^{(M-1)}} \tag{A.11}$$

$$= -\sum_\mu \sigma_l^{(M)} \left[g_i^{(M-1)} \right]' \left(h_i^{(M-1)} \right) \sum_k J_{ik} \delta_k^{(M)}. \tag{A.12}$$

Definindo o "erro retro-propagado" :

$$\delta_k^{(M-1)} \equiv \left[g_i^{(M-1)} \right]' \left(h_i^{(M-1)} \right) \sum_k J_{ik} \delta_k^{(M)} \tag{A.13}$$

Podemos escrever a dinâmica para a camada "$M-1$" numa forma análoga a forma para "M":

$$\Delta J_{ik}^{(M-1)} = \eta \sum_\mu \delta_k^{(M-1)}(\mu) \sigma_i^{(M-1)}. \tag{A.14}$$

Notemos que o *update* das conexões sinápticas da camada "$M-1$" somente é possível após a "retro-propagação" (*backpropagation*) do erro a partir da camada de saída, que dá nome ao algoritmo. O mesmo ocorre para todas as outras camadas seguindo a regra geral expressa em (A.15). O algoritmo *backpropagation* foi descrito acima em sua versão *offline* onde o conjunto de exemplos $\mathcal{L}$ é memorizado e utilizado em paralelo. Em aplicações práticas é muito mais comum o uso da versão *on-line* onde os exemplos são utilizados um por vez. Esta versão além de ser mais econômica computacionalmente, também mostra-se mais eficiente, sendo capaz de evitar mínimos locais da energia E utilizando o processo descrito como *self-annealing* [25]. O fato é que o aprendizado *on-line* pode ser visto como a minimização do potencial $E = \sum_\mu V_\mu$ pela amostragem aleatória de subpotenciais na forma $V_\mu = \frac{1}{2} \sum_k \left[\Sigma_k^0(\mu) - \Sigma_k(\mu) \right]^2$. Esta amostragem aleatória introduz um "ruído efetivo" com amplitude proporcional à distância da configuração dos pesos sinápticos com relação à configuração de mínima energia, esta seria a origem da denominação. A versão *online* do algoritmo *backpropagation* adquire a forma geral :

1. Apresenta-se o exemplo $(\mathbf{S}, \Sigma^{(0)})$.
2. Calculam-se os potenciais $\sigma^{(m)}$ e saída da rede Σ.
3. Calculam-se os erros da camada de saída $\delta^{(M)}$.

4. Propagam-se estes erros para as camadas internas.

5. Realiza-se a atualização das sinápses utilizando:

$$\Delta J_k^{(m)} = \eta \delta_k^{(m)} \sigma^{(m-1)}. \tag{A.15}$$

6. Volta-se a (1) no próximo exemplo.

B

Auto-mediância

Neste apêndice procuraremos justificar teoricamente a auto-mediância dos parâmetros de ordem em uma situação de aprendizado *online* simples. Seguiremos o raciocínio desenvolvido em [40]. Suponhamos que o sistema dependa apenas de um parâmetro de ordem O (por exemplo o percéptron booleano depende apenas de ρ). A evolução de O é descrita por:

$$O(\mu+1) = O(\mu) + \frac{1}{N}\phi(O(\mu), S(\mu)) \tag{B.1}$$

Aqui ϕ representa a correção que o parâmetro recebe a cada passo da dinâmica, suponhamos que $\phi \in \mathcal{C}^\infty$ em O. Suponhamos que os exemplos sejam uma seqüência aleatória sem correlação entre seus termos:

$$P(S(0), S(1), ..., S(\mu)) = \prod_{\nu=0}^{\mu} P(S(\nu))$$

Suponhamos, adicionalmente, que o processo estocástico definido por (B.1) seja *quasi-gaussiano*, ou seja, o processo é dominado pelos dois primeiros momentos.

Se iniciarmos a dinâmica em um valor específico O^*, teremos uma distribuição inicial com segundo momento nulo dada por

$$P(O(0)) = \delta(O(0) - O^*)$$

. Para que O seja auto-mediante durante toda a dinâmica precisamos mostrar que, no limite de sistemas grandes, o segundo momento σ^2 permanecerá nulo.

Utilizando (B.1) escrevemos :

$$\begin{aligned}
\langle O^2(\mu+1)\rangle &= \langle O^2(\mu)\rangle + \frac{1}{N^2}\langle \phi^2(\mu)\rangle + \frac{2}{N}\langle O(\mu)\phi(\mu)\rangle \\
\langle O(\mu+1)\rangle^2 &= \langle O(\mu)\rangle^2 + \frac{1}{N^2}\langle \phi\rangle^2(\mu) + \frac{2}{N}\langle O(\mu)\rangle\langle \phi(\mu)\rangle
\end{aligned}$$

Podemos então obter uma equação para a dinâmica do segundo momento:

$$\begin{aligned}\sigma^2(\mu+1)-\sigma^2(\mu) &= \langle O^2(\mu+1)\rangle - \langle O^2(\mu)\rangle - \langle O(\mu+1)\rangle^2 + \langle O(\mu)\rangle^2 \\ &= \frac{2}{N}\langle\phi(\mu)O(\mu)\rangle - \frac{2}{N}\langle\phi(\mu)\rangle\langle O(\mu)\rangle \\ &+ \frac{1}{N^2}\langle\phi^2(\mu)\rangle - \frac{1}{N^2}\langle\phi(\mu)\rangle^2 \qquad \text{(B.2)}\end{aligned}$$

Aqui $\langle ...\rangle$ são médias, como já convencionamos, sobre toda aleatoriedade do sistema. Para este sistema simples que estamos tratando a aleatoriedade é a sequência de exemplos. Como as seqüências não são correlacionadas podemos escrever $\mathcal{V} = \{O(\mu-1), S(\mu)\}$, realizando as médias sobre a distribuição de O e do último exemplo apresentado. A demonstração seguirá por indução, primeiro mostraremos que o incremento do segundo momento no primeiro passo é de $\mathcal{O}(\frac{1}{N^2})$, a seguir mostraremos que se $\sigma^2(\mu) = \mathcal{O}(\frac{1}{N^2})$ para um μ qualquer então $\sigma^2(\mu+1) = \mathcal{O}(\frac{1}{N^2})$. Finalmente teremos, já que $\sigma^2(0) = 0$, que $\sigma^2(\mu+1) = (\mu+1)\mathcal{O}(\frac{1}{N^2})$ como $\mu = \alpha N$ concluiremos que $\sigma^2(\mu+1) = \alpha\mathcal{O}(\frac{1}{N})$ o que demonstrará a propriedade de auto-mediância.

A primeira parte da demonstração fica :

$$\begin{aligned}\sigma^2(1)-\sigma^2(0) &= \frac{2}{N}\langle\phi(0)O(0)\rangle - \frac{2}{N}\langle\phi(0)\rangle\langle O(0)\rangle \\ &+ \frac{1}{N^2}\langle\phi^2(0)\rangle - \frac{1}{N^2}\langle\phi(0)\rangle^2 \\ &= \frac{2}{N}\langle\phi(0)\rangle O^* - \frac{2}{N}\langle\phi(0)\rangle O^* + \mathcal{O}(\frac{1}{N^2}) \\ &= \mathcal{O}(\frac{1}{N^2})\end{aligned}$$

Para levarmos a cabo a segunda parte retomaremos (B.2) :

$$\sigma^2(\mu+1)-\sigma^2(\mu) = \frac{2}{N}\langle\phi(\mu)O(\mu)\rangle - \frac{2}{N}\langle\phi(\mu)\rangle\langle O(\mu)\rangle + \mathcal{O}(\frac{1}{N^2})$$

Precisamos avaliar os dois primeiros termos do lado direito da equação acima para $\sigma = \mathcal{O}(\frac{1}{N^2})$. Para um μ qualquer teremos :

$$\begin{aligned}\langle\phi O\rangle - \langle\phi\rangle\langle O\rangle &= \int Dx(\langle O\rangle + \sigma x)\langle\phi(\langle O\rangle + \sigma x, S)\rangle_S \\ &- \langle O\rangle\int Dx\langle\phi(\langle O\rangle + \sigma x, S)\rangle_S\end{aligned}$$

Aqui $x \equiv \frac{(O-\langle O\rangle)}{\sigma}$, $\langle ...\rangle_S$ representa a média sobre todos os exemplos anteriores , $\langle ...\rangle$ representa a média sobre a distribuição de O e $\int Dx... = \int \frac{dx}{\sqrt{2\pi}} e^{\frac{1}{2}x^2}....$ Expandindo ϕ em potências de σ :

$$\begin{aligned}
& \int Dx(\langle O\rangle + \sigma x)\langle\phi(\langle O\rangle + \sigma x, S)\rangle_S - \langle O\rangle \int Dx\langle\phi(\langle O\rangle + \sigma x, S)\rangle_S = \\
= & \int Dx(\langle O\rangle + \sigma x)\langle\phi(\langle O\rangle, S) + \sigma x\phi'(\langle O\rangle, S)\rangle_S \\
- & \langle O\rangle \int Dx\langle\phi(\langle O\rangle, S) + \sigma x\phi'(\langle O\rangle, S)\rangle_S + \mathcal{O}(\sigma^2) = \\
= & \mathcal{O}(\sigma^2)
\end{aligned}$$

Concluindo assim a segunda parte da demonstração :

$$\sigma^2(\mu+1) - \sigma^2(\mu) = \frac{2}{N}\mathcal{O}(\sigma^2(\mu)) + \mathcal{O}(\frac{1}{N^2})$$

Daqui segue imediatamente que, sob as hipóteses assumidas :

$$\sigma^2(\mu+1) = \alpha\mathcal{O}(\frac{1}{N})$$

O que demonstra a auto-mediância do parâmetro para qualquer α.

C

Erro de Generalização

O erro de generalização médio é definido para arquiteturas totalmente conectadas com unidades contínuas por :

$$e_g \;=\; \frac{1}{2}\left\langle\left\langle\left[\sum_{n=1}^{M} g(\sum_i B_{ni}S_i) \;-\; \sum_{k=1}^{K} g(\sum_i J_{ki}S_i)\right]^2\right\rangle\right\rangle_{\mathbf{J}|\mathcal{L}}\rangle_{\mathcal{L}|\mathbf{B}} \qquad \text{(A.1)}$$

Onde J_{ki} é o conjunto dos acoplamentos sinápticos e S_i são as entradas fornecidas com distribuição com as propriedades :

$$\langle\langle S_i\rangle\rangle \;\equiv\; 0$$

$$\langle\langle S_i S_k\rangle\rangle \;\equiv\; \delta_{ik}$$

As médias são definidas sobre seqüências de entradas.

Definindo os campos pós-sinápticos $h_k \equiv \sum_i J_{ki}S_i$ e $b_n \equiv \sum_i B_{ni}S_i$. Teremos, no limite termodinâmico, que h_k é uma variável com distribuição gaussiana e que :

$$\begin{aligned}
\langle\langle h_k h_j\rangle\rangle &= \langle\langle\sum_{i,l} J_{ki}S_{\sigma,i}J_{jl}S_l\rangle\rangle \\
&= \sum_{i,l} J_{ki}J_{jl}\langle\langle S_i S_l\rangle\rangle \\
&= \sum_{i,l} J_{ki}J_{jl}\delta_{kl} \\
&= (J_k \cdot J_j) \\
&= Q_{jk}
\end{aligned}$$

Procedendo analogamente chegamos também a :

$$\langle\langle b_n b_m\rangle\rangle \equiv M_{nm}$$

$$\langle\langle h_k b_n\rangle\rangle \equiv R_{kn}$$

Neste regime o erro de generalização pode ser alternativamente escrito como :

$$e_g \;=\; \frac{1}{2}\int_{-\infty}^{+\infty}\left(\prod_{n=1}^{M} db_m\right)\left(\prod_{k=1}^{K} dh_k\right) P(\{b_n,h_k\})\left[\sum_{n=1}^{M} g(b_n) \;-\; \sum_{k=1}^{K} g(h_k)\right]^2 \qquad (A.2)$$

, onde $P(\{b_n,h_k\})$ é uma distribuição gaussiana multidimensional com médias nulas. Abrindo a expressão anterior chegamos a :

$$\begin{aligned}
&\frac{1}{2}\sum_{n,m=1}^{M}\int_{-\infty}^{+\infty}\left(\prod_{l=1}^{M} db_l\right)\left(\prod_{k=1}^{K} dh_k\right) P(\{b_n,h_k\})g(b_n)g(b_m) \;+\\
+&\frac{1}{2}\sum_{k,j=1}^{K}\int_{-\infty}^{+\infty}\left(\prod_{n=1}^{M} db_n\right)\left(\prod_{k=1}^{K} dh_k\right) P(\{b_n,h_k\})g(h_k)g(h_j) \;-\\
&-\sum_{k=1}^{K}\sum_{n=1}^{M}\int_{-\infty}^{+\infty}\left(\prod_{n=1}^{M} db_n\right)\left(\prod_{k=1}^{K} dh_k\right) P(\{b_m,h_k\})g(b_n)g(h_k)
\end{aligned}$$

Fazendo a escolha $g(x) \equiv erf\left(\frac{x}{\sqrt{2}}\right)$ e integrando sobre as variáveis que comparecem apenas em P:

$$\begin{aligned}
&\frac{1}{2}\sum_{n=1}^{M}\int_{-\infty}^{+\infty} db_n erf^2\left(\frac{b_n}{\sqrt{2}}\right) P(b_n) \;+\\
+&\frac{1}{2}\sum_{n,m=1\{n\neq m\}}^{M}\int_{-\infty}^{+\infty} db_n db_m erf\left(\frac{b_n}{\sqrt{2}}\right) erf\left(\frac{b_m}{\sqrt{2}}\right) P(b_n,b_m) \;+\\
&+\frac{1}{2}\sum_{k=1}^{K}\int_{-\infty}^{+\infty} dh_k erf^2\left(\frac{h_k}{\sqrt{2}}\right) P(h_k) \;+\\
+&\frac{1}{2}\sum_{k,j=1\{k\neq j\}}^{K}\int_{-\infty}^{+\infty} dh_k dh_j erf\left(\frac{h_k}{\sqrt{2}}\right) erf\left(\frac{h_j}{\sqrt{2}}\right) P(h_k,h_j) \;-\\
&-\frac{1}{2}\sum_{n=1}^{M}\sum_{k=1}^{K}\int_{-\infty}^{+\infty} db_n dh_k erf\left(\frac{h_k}{\sqrt{2}}\right) erf\left(\frac{b_n}{\sqrt{2}}\right) P(b_n,h_k)
\end{aligned}$$

As integrais são de dois tipos :

1. $\int_{-\infty}^{+\infty} dx\, erf^2\left(\frac{x}{\sqrt{2}}\right) P(x)$
2. $\int_{-\infty}^{+\infty} dxdy\, erf\left(\frac{x}{\sqrt{2}}\right) erf\left(\frac{y}{\sqrt{2}}\right) P(x,y)$

Tipo 1

Introduzindo $P(x)$ podemos usar a identidade utilizada em [7] :

$$\int_{-\infty}^{+\infty} \frac{dx}{\sqrt{2\pi\sigma^2}} erf^2\left(\frac{x}{\sqrt{2}}\right) e^{-\frac{x^2}{2\sigma^2}} \;=\; \frac{2}{\pi}\arcsin\frac{\sigma^2}{1+\sigma^2}$$

Finalmente definindo $\langle h_k^2\rangle \equiv Q_{kk}$ e $\langle b_n^2\rangle \equiv M_{nn}$ escrevemos :

$$\int_{-\infty}^{+\infty} db_n erf^2\left(\frac{b_n}{\sqrt{2}}\right) P(b_n) \;=\; \frac{2}{\pi}\arcsin\frac{M_{nn}}{1+M_{nn}} \tag{A.3}$$

$$\int_{-\infty}^{+\infty} dh_k erf^2\left(\frac{h_k}{\sqrt{2}}\right) P(h_k) \;=\; \frac{2}{\pi}\arcsin\frac{Q_{kk}}{1+Q_{kk}} \tag{A.4}$$

Tipo 2

Para trabalharmos com as integrais do tipo 2 precisamos primeiro definir uma matriz de correlações apropriada :

$$\mathcal{C} \equiv \begin{bmatrix} c_1 & \tilde{c} \\ \tilde{c} & c_2 \end{bmatrix}$$

A inversa é facilmente calculada ,assim como o determinante :

$$\mathcal{C}^{-1} = \frac{1}{c_1c_2 - \tilde{c}^2}\begin{bmatrix} c_2 & -\tilde{c} \\ -\tilde{c} & c_1 \end{bmatrix}$$

$P(x,y)$ é então escrito :

$$P(x,y) = \frac{1}{2\pi\sqrt{c_1c_2-\tilde{c}^2}} \exp -\frac{1}{2}(x,y)\mathcal{C}^{-1}(x,y)^T$$

Introduzindo $P(x,y)$ a integral do tipo 2 fica :

$$\int_{-\infty}^{+\infty} \frac{dxdy}{2\pi\sqrt{c_1c_2-\tilde{c}^2}} erf\left(\frac{x}{\sqrt{2}}\right) erf\left(\frac{y}{\sqrt{2}}\right) \exp -\frac{1}{2}\left(\frac{c_2x^2 + c_1y^2 - 2\tilde{c}xy}{c_1c_2-\tilde{c}^2}\right)$$

Separando as integrações:

$$\int_{-\infty}^{+\infty} \frac{dx}{2\pi\sqrt{c_1c_2-\tilde{c}^2}} erf\left(\frac{x}{\sqrt{2}}\right) \exp -\frac{1}{2}\left(\frac{c_2x^2}{c_1c_2-\tilde{c}^2}\right) * \int_{-\infty}^{+\infty} dy\, erf\left(\frac{y}{\sqrt{2}}\right) \exp -\frac{1}{2}\left(\frac{c_1y^2 - 2\tilde{c}xy}{c_1c_2-\tilde{c}^2}\right)$$

Utilizando a identidade [38] :

$$\int_{-\infty}^{+\infty} dy\, erf(ay)\, exp(-by^2+cy) = \sqrt{\frac{\pi}{b}} e^{\frac{c^2}{4b}} erf\left(\frac{ac}{2\sqrt{a^2b+b^2}}\right)$$

Chegamos à forma :

$$\int_{-\infty}^{+\infty} \frac{d\tilde{x}}{\sqrt{2\pi}} erf\left(\sqrt{\frac{c_1}{2}}\tilde{x}\right) \exp\left(-\frac{1}{2}\tilde{x}^2\right) erf\left(-\frac{\tilde{c}\sqrt{c_1}}{\sqrt{2(c_1^2c_2 + c_1^2 - \tilde{c}^2c_1)}}\right)$$

Aqui podemos novamente utilizar a identidade de [7] para escrevermos:

$$\int_{-\infty}^{+\infty} dxdy\, erf\left(\frac{x}{\sqrt{2}}\right) erf\left(\frac{y}{\sqrt{2}}\right) P(x,y) = \frac{2}{\pi} \arcsin \frac{\tilde{c}}{\sqrt{1+c_1}\sqrt{1+c_2}}$$

Finalmente escrevemos :

$$\int_{-\infty}^{+\infty} dh_k dh_j\, erf\left(\frac{h_k}{\sqrt{2}}\right) erf\left(\frac{h_j}{\sqrt{2}}\right) P(h_k,h_j) = \frac{2}{\pi} \arcsin \frac{Q_{jk}}{\sqrt{1+Q_{jj}}\sqrt{1+Q_{kk}}} \quad \text{(A.5)}$$

$$\int_{-\infty}^{+\infty} db_n db_m\, erf\left(\frac{b_n}{\sqrt{2}}\right) erf\left(\frac{b_m}{\sqrt{2}}\right) P(b_n,b_m) = \frac{2}{\pi} \arcsin \frac{M_{nm}}{\sqrt{1+M_{nn}}\sqrt{1+M_{mm}}} \quad \text{(A.6)}$$

$$\int_{-\infty}^{+\infty} dh_k db_n\, erf\left(\frac{h_k}{\sqrt{2}}\right) erf\left(\frac{b_n}{\sqrt{2}}\right) P(h_k,\tilde{b}_n) = \frac{2}{\pi} \arcsin \frac{R_{kn}}{\sqrt{1+Q_{kk}}\sqrt{1+M_{nn}}} \quad \text{(A.7)}$$

Expressão Final

$$\begin{aligned} e_g(\{Q_{kj}, M_{nm}, R_{kn}\}) &= \frac{1}{\pi}\sum_{n,m=1}^{M} \arcsin \frac{M_{nm}}{\sqrt{1+M_{nn}}\sqrt{1+M_{mm}}} \\ &+ \frac{1}{\pi}\sum_{j,k=1}^{K} \arcsin \frac{Q_{jk}}{\sqrt{1+Q_{jj}}\sqrt{1+Q_{kk}}} \\ &- \frac{2}{\pi}\sum_{k=1}^{K}\sum_{n=1}^{M} \arcsin \frac{R_{kn}}{\sqrt{1+Q_{kk}}\sqrt{1+M_{nn}}} \end{aligned} \quad \text{(A.8)}$$

Bibliografia

[1] AMARI S., Theory of adaptive pattern classifiers, *IEEE Trans.* **EC-16** 299 (1967).

[2] AMARI S., *Differential-Geometrical Methods in Statistics*, (Lecture Notes in Statistics **28**, Springer-Verlag, New York, 1985).

[3] AMARI S., Natural Gradient Works Efficiently in Learning, *preprint - RIKEN Frontier Research Program, Japão*, (1997).

[4] AMIT D., *Modeling Brain Function*, (Cambridge University Press, Cambridge, UK, 1989).

[5] BARBER D., SOLLICH P. e SAAD D., Finite size effects in on-line learning of multi-layer neural networks, *preprint - University of Edinburgh*, (1996).

[6] BIEHL M. e RIEGLER P., On-line learning with a perceptron, *Europhys. Lett.* **28** 525 (1994).

[7] BIEHL M. e SCHWARZE H., Learning by on-line gradient descent. *J. Phys. A: Math. Gen.* **28** 643 (1995).

[8] BIEHL M., RIEGLER P. e STECHERT M., Learning from noisy data: an exactly solvable model, *Phys. Rev. E* **52** R4624 (1995).

[9] BIEHL M. e RIEGLER P., On-line back-propagation in two-layered neural networks, *J. Phys. A: Math. Gen.* **28** L507 (1995).

[10] BIEHL M., WÖHLER C. e RIEGLER P., Transient dynamics of on-line learning in two-layered neural networks, *J. Phys. A: Math. Gen.* **29** 4769 (1996).

[11] BISHOP C.M., *Neural Networks for Pattern Recognition* , (Oxford University Press, Oxford, UK, 1996).

[12] CATICHA N. e KINOUCHI O., Time ordering in the evolution of information processing and modulation systems, *cond-mat/9706112*, Proceedings of the MINERVA Workshop on Neural Networks, Eilat *a aparecer no Phil. Mag.* (1997).

[13] COPELLI M., *Determinação Variacional de Algoritmos de Aprendizagem em Redes Neurais*, Dissertação de Mestrado, Instituto de Física, Universidade de São Paulo (1995).

[14] COPELLI M. e CATICHA N., On-line learning in a the committee machine, *J. Phys. A: Math. Gen.* **28** 1615 (1995).

[15] COPELLI M., Noise robustness in the perceptron *Proceedings ESANN 97* (Bélgica, 1997).

[16] COPELLI M., EICHORN R., KINOUCHI O., BIEHL M., SIMONETTI R., RIEGLER P. e CATICHA N., Noise robustness in multilayer neural networks, *Europhys. Lett.* **37** (6) 427 (1997).

[17] CYBENKO G., Approximation by Superpositions of Sigmoidal Functions, *Math. of Control Signals and Syst.* **2** 303 (1989).

[18] FISCHER K.H. e HERTZ J.A., *Spin Glasses*, (Cambridge University Press, Cambridge, UK, 1991).

[19] FONTANARI J.F., Generalization in a Hopfield network, *J. Phys. France* **51** 2421 (1990).

[20] GARDNER E., The Space of Interactions in Neural Network Models, *J. Phys. A: Math. Gen.* **21** 257 (1988).

[21] GARDNER E. e DERRIDA B., Three Unfinished Works on the Optimal Storage Capacity of Networks, *J. Phys. A: Math. Gen.* **22** 1983 (1989).

[22] GELFAND I.M. e FOMIN S.V., *Calculus of Variations*, (Prentice-Hall Inc., EUA, 1963).

[23] HEBB D.O., *The Organization of Behaviour*, (Willey, New York, 1949).

[24] HERTZ J.A., KROGH A. e PALMER R.G., *Introduction to the Theory of Neural Computation*, (Addison-Wesley, Reedwood City, CA, USA, 1991).

[25] HONDOU T., Self-Annealing Dynamics in a Multistable System, *Prog. Theor. Phys.* **95** 817 (1996).

[26] HOPFIELD J.J., Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proc. Nat. Acad. Sci.* **79** 2554 (1982).

[27] HORNIK K., Some New Results on Neural Network Appoximation, *Neural Networks* **6** 1069 (1993).

[28] KIM J.W. e SOMPOLINSKY H., On-line Gibbs Learning, *Phys. Rev. Lett.* **76** 3021 (1996).

[29] KINOUCHI O. e CATICHA N., Optimal generalization in perceptrons, *J. Phys. A: Math. Gen.* **25** 6243 (1992).

[30] KINOUCHI O., *Generalização Ótima em Percéptrons*, Dissertação de Mestrado, Instituto de Física e Química de São Carlos, Universidade de São Paulo (1992).

[31] KINOUCHI O. e CATICHA N., On-line versus off-line learning in the linear perceptron: a comparative study, *Phys. Rev. E* **52** 2878 (1995).

[32] KINOUCHI O., *Aprendizagem Ótima em Percéptrons a partir de Exemplos com Ruído*, Tese de Doutorado, Instituto de Física, Universidade de São Paulo (1996).

[33] MACE C.W.H. e COOLEN A.C.C., Statistical Mechanical Analysis of the Dynamics of Learning in Perceptrons, cond-mat/9705243, a ser publicado em *Statistics and Computing* (1997).

[34] McCULLOCH W.S. e PITTS W., A logical Calculus of Ideas Immanent in Nervous Activity. *Bull. Math. Biophys.* **5** 115 (1943).

[35] MINSKY M.L. e PAPERT S.A., *Perceptrons*, (MIT Press, EUA, 1969).

[36] OPPER M., On-line versus Off-line Learning from Random Examples: General Results, *Phys. Rev. Lett.* **77** 4671 (1996).

[37] OPPER M. e KINZEL W., *Statistical Mechanics of Generalization*, em Pysics of Neural Networks, ed. por J.L.van Hemmen, E. Domany e K. Schulten, (Springer Verlag, Berlin, 1996).

[38] PRUDNIKOV A.P., BRYCHCOV Yu. A., MARICHEV O.I., (Integrals and Series, vol. 2 - Special Functions, Gordon and Breach Science Publishers, 1988)

[39] RATTRAY M. e SAAD. D., Globally Optimal On-line Learning Rules for Multilayer Neural Networks, *submetido ao Europhys. Lett.*.

[40] RIEGLER P., *Dynamics of On-line Learning in Neural Networks*, Tese de Doutorado, Institut für Theoretische Physik, Bayerische Julius-Maximilians-Universität Würzburg (1997).

[41] ROSENBLATT F., *Principles of Neurodynamics*, (Spartan, New York, 1962).

[42] RUMELHART D.E., McCLELLAND J.L. , *Parallel Distributed Processing - Vol.1: Foundations, Vol.2: Psychological and Biological Models*, (MIT Press, EUA, 1989).

[43] RUMELHART D.E., HINTON G.E. e WILLIAMS R.J., Learning Representations by Backpropagation Errors, *Nature* **323** 533 (1986).

[44] SAAD D. e RATTRAY M., Globally Optimal Parameters for On-Line Learning in Multilayer Neural Networks, Proceedings of the MINERVA Workshop on Neural Networks, Eilat *a aparecer no Phil. Mag.* (1997).

[45] SAAD D. e SOLLA S.A., Exact solution for on-line learning in multilayer neural networks, *Phys. Rev Lett.* **74** 4337 (1995).

[46] SAAD D. e SOLLA S.A., On-line learning in soft committe machines, *Phys. Rev E* **52** 4225 (1995).

[47] SAAD D. e SOLLA S.A., Learning from Corrupted Examples in Multilayer Networks, *preprint NCRG, Aston University* (1996).

[48] SEUNG H. S., SOMPOLINSKY O., TISHBY N., Statistical mechanics of learning from examples, *Phys. Rev. A* **45** 6056 (1992).

[49] SIMONETTI R. e CATICHA N., On-line learning in parity machines, *J. Phys. A: Math. Gen.* **29** 4859 (1996).

[50] SIMONETTI R., MATTOS C. e CATICHA N., Percéptron com Função de Transferência Não-linear, em *Resumos do XIX Encontro Nacional de Física da Matéria Condensada* (Águas de Lindóia, 1996).

[51] SIMONETTI R., *Generalização e Robustez: Aprendizagem em Redes Neurais na Presença de Ruído*, Tese de Doutorado, Instituto de Física, Universidade de São Paulo (1997).

[52] VALLET F., The Hebb rule for learning linearly separable boolean functions: learning and generalization, *Europhys. Lett.* **08** 747 (1989).

[53] VAN DEN BROECK C. e REIMANN P., Unsupervised learning by examples: on-line versus off-line, *Phys. Rev. Lett.* **76** 2188 (1994).

[54] VICENTE R. e CATICHA N., Functional Optimization of Online Algorithms in Multilayer Neural Networks, *J. Phys. A: Math. Gen.* **30** L599 (1997).

[55] VICENTE R. e CATICHA N., Locally Optimized Online Learning in Fully Connected Soft Committee Machines, *em preparação.*

[56] WATKIN T.H.L., RAU A. e BIEHL M., The statistical mechanics of learning a rule, *Rev. Mod. Phys.* **65** 499 (1993).

[57] WEST A.H.L. e SAAD D., Adaptative Back-propagation in On-line Learning of Multilayer Networks, *Proceedings NIPS'96* (1996).

[58] WIDROW B., Generalization and Information Storage in Networks of Adaline "Neurons", (Self-Organizing Systems, ed. G.T. Jacobie G.P. Goldstein, Chicago, 1962).

[59] WINTHER O. e SOLLA S.A., Bayesian online learning in the perceptron, *Proceedings of the fifth Symposium on Artificial Neural Networks*, Bélgica, (1997).

[60] WÖHLER C., *Plateauzustände beim Einschritt-Lernen in neuralen Netzwerken*, Diplomarbeit, Institut für Theoretische Physik, Bayerische Julius-Maximilians-Universität Würzburg (1996).