

2

Técnicas convencionais usadas na modelagem de sistemas não lineares

2.1.

Revisão Bibliográfica

Os modelos lineares desempenharam um importante papel no desenvolvimento das técnicas de identificação. Devido à sua simplicidade e à facilidade de obtenção, são usados para aproximar o comportamento de sistemas em determinadas faixas de operação. Eles desempenham um importante papel na modelagem de sistemas dinâmicos, quando é possível limitar a operação a regiões de linearidade. Billings (1980) justifica a utilização de modelos lineares em função da complexidade potencial dos não-lineares.

A necessidade de representar de forma mais precisa o comportamento não-linear de sistemas reais levou à busca de representações não-lineares. Em (Billings, 1980) pode ser visto um importante estudo sobre os métodos de identificação de sistemas não-lineares anteriores à década em questão. Nesse artigo, como ferramentas para identificar sistemas não-lineares, o autor cita os métodos de series funcionais, como o método de Wiener, série de Volterra e diversas técnicas no domínio da frequência. Como alternativa, cita os sistemas orientados por blocos, por exemplo partindo em bloco linear e não linear, apresentando a formulação geral e os modelos clássicos de Wiener e Hammerstein [2].

Com relação às séries funcionais, o autor apresenta como desvantagens a complexidade computacional, a dificuldade em, a partir do modelo, incorporar

informação a priori e de interpretar e estimar características físicas do processo. Com relação aos modelos orientados por blocos, afirma que a separação do sistema modelado em subsistemas lineares e não-lineares é extremamente atrativa e pode fornecer importante ajuda no desenvolvimento de controladores. Para essa representação, salienta a necessidade de pesquisas futuras para simplificar e estender a aplicação de tais métodos.

Vale ressaltar, no trabalho citado, a afirmação do autor que, quando se conhecem as equações diferenciais representativas do sistema, os métodos para estimação de parâmetros estão bem estabelecidos. Porém, quando apenas um conjunto reduzido de informação está disponível, o problema de obtenção do modelo e de estimação de parâmetros deve ser investigado. Outra observação importante do autor é a dificuldade usualmente encontrada para identificar sistemas não-lineares e a impossibilidade de recomendar apenas uma técnica capaz de fornecer soluções gerais e aceitáveis.

Na década de oitenta, surgiram diversos novos métodos para identificar sistemas não-lineares. Destaca-se nessa fase o aparecimento de novas representações. Dentre essas, a representação NARMAX polinomial proposta em (Leontaritis e Billings, 1985) e NARMAX racional (Billings e Chen, 1989). Outro conjunto de representações que se destacaram na identificação de sistemas não-lineares no final da década de oitenta são as redes neurais. Em todos esses casos, o objetivo era sempre o de encontrar representações capazes de aproximar diversas classes de não-linearidades.

Entre as não-linearidades que mereceram atenção de inúmeros pesquisadores, destacam-se os sistemas com comportamento caótico. Dentro desse campo de aplicações pode ser citado o trabalho de Casdagli (1989), no qual o autor aborda o problema de obtenção de modelos preditivos do tipo RBF a partir de dados temporais.

Um dos principais problemas citados nos diversos trabalhos de pesquisa é a dificuldade encontrada na escolha da técnica apropriada. Vários trabalhos surgiram com intuito de atacar tal problema (Haber e Unbehauen, 1990; Breedem e Packard, 1994; Aguirre, 1994; Judd e Mees, 1995; Mão e Billings, 1997). Discussão dos efeitos da sobre-parametrização em modelos não-lineares é apresentada em (Aguirre e Billings, 1995). No caso específico de modelos NARMAX polinomiais, pode-se ainda citar o estudo de agrupamento de termos (Aguirre e Billings, 1995).

Uma questão de grande relevância no processo de identificação de sistemas nos quais se utiliza informação a priori do sistema em conjunto com dados de entrada e saída medidos no mesmo, é saber quais informações a respeito do sistema são realmente úteis e como utilizá-las na etapa de seleção de estrutura e de estimação de parâmetros. Um outro ponto de interesse é, ainda, determinar que representação é capaz de incorporar o maior volume de conhecimento a priori do sistema.

Mais à frente será proposto um novo procedimento, envolvendo mais que uma técnica de identificação (diferentes redes neurais, por exemplo). Como tal prática já existe neste campo de trabalho, inclui-se um destes métodos nesta revisão, servindo de referência ao proposto. Este é o tópico da próxima seção.

2.1.1.

Mistura Adaptativa de Sistemas Especialistas

Trata-se de um procedimento de aprendizagem supervisionado para sistemas compostos por diversos tipos de redes [49]. Cada uma das redes é treinada para representar um subconjunto do jogo completo de casos do treinamento. Este procedimento pode ser visto como uma versão modular de uma rede supervisionada “multilayer”, ou como uma versão associativa da aprendizagem do competidor.

Se o “backpropagation”, como no caso apresentado em [49], for usado para treinar uma única rede multilayer na execução de sub-tarefas diferentes em ocasiões diferentes, o processo certamente terá aprendizagem lenta e perturbada pela descontinuidade de operação. Esta perturbação pode ser reduzida caso se saiba em avanço que um jogo de casos do treinamento pode ser naturalmente dividido em subconjuntos, correspondendo às sub-tarefas distintas. Isto seria possível usando um sistema composto por diversas redes especialistas, acrescidas de uma rede “seletora”, capaz de selecionar quais técnicas devem ser usadas para cada caso do treinamento. Hampshire e Waibel (1989) descreveram um sistema deste tipo que pudesse ser usado quando a divisão em sub-tarefas é conhecida antes do treinamento. Jacobs (1990) descreveu um sistema relacionado que aprendesse como alocar casos às redes.

A idéia atrás de tal sistema é que a rede “seletora” aloca um caso novo aos sistemas especialistas através de pesos associados a cada uma destas. Assim não há nenhuma interferência entre os pesos de redes que se especializam em casos completamente diferentes. Os sistemas especialistas são conseqüentemente locais, no sentido que os pesos em um especialista estão relacionados ao comportamento dos sinais de entrada e saída respectivos, é dizer cada especialista estará alocado somente a uma região local pequena do espaço de vetores possíveis da entrada. Hampshire, Waibel e Jacobs, tentando estabelecer a localização (atribuição de regiões locais aos especialistas), usaram como critério a função de erro descrita pela equação (1.1) Supõem que a saída final do sistema inteiro é uma combinação linear das saídas dos especialistas locais, com a rede seletora determinando a proporção de cada saída local na combinação linear. Assim a função de erro é definida como:

$$E^c = \left\| d^c - \sum_i p_i^c o_i^c \right\|^2 \quad (1.1)$$

onde o " o_i^c " é o vetor da saída do especialista "i"; no caso "c", " p_i^c " é a contribuição proporcional do "i" especialista; ao vetor combinado da saída, e " d^c " é o vetor desejado da saída no caso "c". Esta medida do erro compara a saída desejada com uma mistura das saídas dos especialistas locais para, assim, minimizar o erro. Cada especialista local deve fazer sua saída cancelar o erro residual que é deixado pelos efeitos combinados de todos os especialistas restantes. Este acoplamento forte entre os especialistas faz com que cooperem agradavelmente, mas tende a conduzir às soluções em que muitos especialistas são usados para cada caso. É possível incentivar a competição adicionando termos da penalidade à função objetivo, para incentivar as soluções em que somente um especialista é ativo (Jacobs, 1990), mas uma solução mais simples é redefinir a função de erro, de modo que os especialistas locais sejam incentivados a competir. Neste caso, ao invés de combinar linearmente as saídas de especialistas à rede seletora, faz-se uma decisão estocástica sobre qual especialista deve ser usado em cada ocasião (Figura 1).

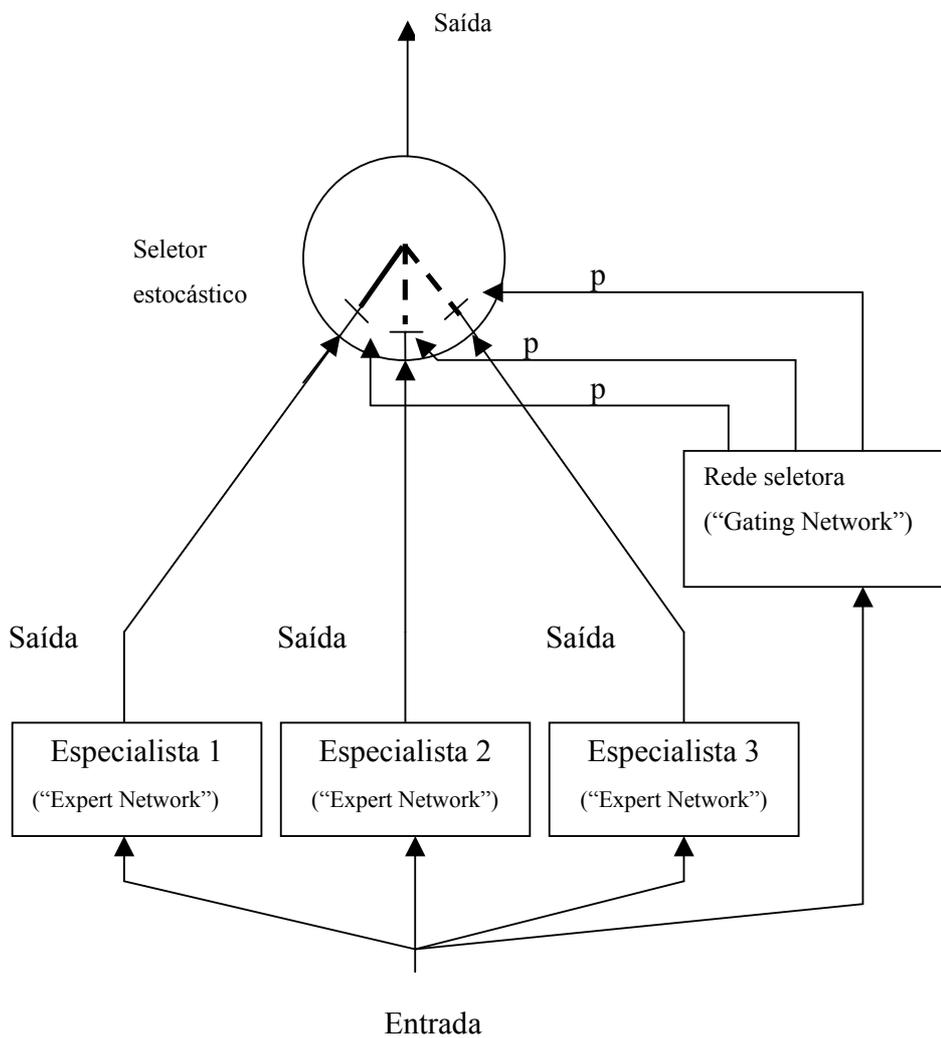


Figura 1 Sistema de redes especialistas e rede seletora: Cada especialista e a rede seletora é uma rede "feedforward"

O erro é então o valor previsto associado ao quadrado da diferença entre os vetores desejados e os reais da saída.

$$E^c = \left\langle \left\| d^c - o_i^c \right\|^2 \right\rangle = \sum_i p_i^c \left\| d^c - o_i^c \right\|^2 \quad (1.2)$$

Observe que nesta nova função de erro, cada especialista tem que produzir todo o vetor da saída e não apenas uma parcela deste. Em consequência, o objetivo de um especialista local em um caso de treinamento não é afetado diretamente pelos pesos associados a outros especialistas locais. Há ainda algum acoplamento indireto, porque se outro especialista mudar seus pesos, pode fazer com que a rede seletora altere as responsabilidades que foram atribuídas aos mesmos. Entretanto, estas mudanças de responsabilidade não podem alterar o sinal de erro que um especialista local detecta em caso de treinamento. Se a rede seletora e os especialistas locais forem treinados pela descida do gradiente nesta nova função de erro (1.2), o sistema tende a devotar um único especialista para cada caso do treinamento. Sempre que um especialista dá menor erro do que a média pesada dos erros de todos os especialistas (de acordo com as saídas da rede seletora), sua contribuição para esse caso será aumentada, e sempre que excede a média sua responsabilidade será diminuída.

Na referencia, é usada outra função de erro, também com bons resultados, mostrada a seguir:

$$E^c = -\log \sum_i p_i^c e^{-\frac{1}{2} \|d^c - o_i^c\|^2} \quad (1.3)$$

Para ver o porque desta função de erro trabalhar melhor, é útil comparar as derivadas das duas funções de erro com o respeito à saída de um especialista. Da equação 1.2 temos:

$$\frac{\partial E^c}{\partial o_i^c} = -2 p_i^c (d^c - o_i^c) \quad (1.4)$$

Enquanto que da equação 1.3 temos:

$$\frac{\partial E^c}{\partial o_i^c} = - \left[\frac{p_i^c e^{-\frac{1}{2} \|d^c - o_i^c\|^2}}{\sum_j p_j^c e^{-\frac{1}{2} \|d^c - o_j^c\|^2}} \right] (d^c - o_i^c) \quad (1.5)$$

Na equação 1.4 o termo “ p_i^c ” é usado para pesar a derivada para o especialista “i”.

Na equação 1.5 foi usado um termo para pesar o especialista “i” como o melhor relativo aos outros especialistas. Esta é uma medida mais útil da relevância do especialista “i” no caso de treinamento “c”.

É natural pensar que os vetores de dados, com que uma rede “competidora” é treinada, joga um papel similar aos vetores da entrada de uma rede “associativa”, que mapea os vetores de entrada aos vetores de saída. Esta correspondência é suposta nos modelos que usam um aprendizado competitivo, como um estágio do pré-processamento dentro de uma rede (Moody e Darken, 1989). Uma vista completamente diferente, é que os vetores de dados usados na aprendizagem competitiva, correspondem aos vetores da saída de uma rede associativa. A “rede competitiva” pode então, ser vista como um gerador estocástico de vetores da saída, e a “aprendizagem competitiva” pode ser vista como o procedimento para fazer a rede gerar vetores saída, com uma distribuição que combine com a distribuição dos vetores de dados.

O vetor “peso” de cada unidade escondida do competidor representa a média de uma distribuição gaussiana multidimensional, e os vetores da saída são gerados pela primeira produção da unidade escondida, então a produção em um vetor de saída de uma distribuição gaussiana, é determinada pelo vetor “pesos” da escolha da

unidade escondida. A probabilidade de geração de algum vetor saída particular " o^c " é dada por

$$\log P^c = \log \sum_i p_i * k * e^{-\frac{1}{2} \|\mu - o^c\|^2} \quad (1.6)$$

onde "i" é um índice das unidades escondidas, " μ ", é o vetor "pesos" da unidade, "k" é uma constante normalizada, e "p" é a probabilidade de colheita da unidade escondida "i", assim o " p_i " são confinados para somar 1. Na literatura estatística (McLachlan e Basford, 1988), o " p_i " são chamados "mixing proportions".

Pode-se ver uma "rede competitiva" como um gerador de vetores saída. Um sistema de "aprendizado competitivo" consiste de um conjunto de unidades colocadas em camadas, hierarquicamente, nas quais cada camada se conecta, via conexões excitatórias, com a camada imediatamente acima dela. No caso mais geral, cada unidade de uma camada recebe uma entrada de cada unidade na camada imediatamente abaixo e projeta saídas para cada unidade na camada imediatamente acima dela. Mais ainda, dentro de uma camada, as unidades são divididas em "clusters" (grupos) inibitórios em que todos os elementos dentro de um "cluster" inibem todos os outros elementos dentro do "cluster". Assim os elementos dentro de um "cluster", em um nível, competem um com o outro para responder ao padrão que está aparecendo na camada abaixo. Quanto mais fortemente qualquer unidade particular responde a um estímulo de chegada, tanto mais ele inibi("shuts down") os outros membros do seu "cluster" (Barto 1985).

O uso da rede seletora permite que a mistura das proporções ("mixing proportions") dos especialistas sejam determinadas pelo vetor da entrada. Isto dá-nos

um sistema de especialistas locais competindo com a função de erro definida na equação 1.3.

2.2.

Descrição das Técnicas Básicas Utilizadas

Na seqüência desta proposta descrevem-se técnicas básicas a serem empregadas no desenvolvimento da pesquisa. Estas técnicas não são exclusivas da metodologia proposta. Na realidade o método proposto pressupõe a incorporação de outras técnicas, se assim desejado. Em seguida serão abordadas genericamente as técnicas RBF e Neuro-Fuzzy.

2.2.1.

Rede RBF em Identificação de Sistemas

A rede neural de base radial (RBF), é um caso particular das redes perceptron multicamada, com uma única camada de unidades ocultas. A RBF é constituída por três camadas, uma de entrada, uma escondida e uma de saída. Na escondida utilizam-se funções de base radial como função de ativação. Na camada de saída, funções de ativação lineares.

Podem-se ressaltar três fases na obtenção de uma rede RBF:

- a) A escolha da função de base radial a ser usada
- b) A escolha do número e da posição dos centros
- c) A estimação dos pesos.

Têm especial importância na modelagem os parâmetros da rede (centros, raios, pesos). Apesar de ser uma boa técnica para a análise de sistemas não lineares, a

rede RBF apresenta inconvenientes ao estimar os parâmetros em determinadas não-linearidades. Isto ocorre mesmo existindo técnicas consagradas para a escolha do número de centros (topologia) e estimação dos parâmetros ou pesos.

Na maioria das situações práticas, com dados corrompidos por ruído, a escolha de um número de centros igual ao número de dados não é recomendável por inúmeras razões. Em qualquer caso, um número de centros muito elevado, além de resultar em elevado esforço computacional devido ao número de pesos a serem estimados, pode resultar em sobre ajuste, também chamado de sobreparametrização.

2.2.1.1.

Utilização Proposta

Dentre os métodos usados para a modelagem de sistemas não lineares baseados nas redes neurais temos a rede RBF em multiescala, modelando o sinal a partir das suas características globais até as particulares, de uma forma muito parecida ao processo das Wavelets.

Esta idéia de análise multiescala inspirou o desenvolvimento da técnica proposta. A idéia é aplicar análises sucessivas em cada sub-espaco dos sinais de entrada e saída, o processo de identificação destes sub-espacos será detalhado mais adiante.

Neste trabalho foram usadas rotinas em Matlab (Institute for Adaptive and Neural Computation -University of Edinburgh). Estas rotinas baseiam-se em métodos que usam árvores de regressão para criar um conjunto de RBF's candidatas, mas que diferem na maneira em que um subconjunto é selecionado para compor a rede.

É importante detalhar, que estas rotinas ajustam automaticamente os parâmetros da rede RBF, de forma a satisfazer o erro de modelagem esperado, motivo fundamental, pelo qual foram usadas.

Adiante, serão usadas redes RBF para modelar sinais específicos de entrada e saída. É importante lembrar que os parâmetros de todas elas serão automaticamente estimados, tentando satisfazer o erro de modelagem esperado.

A descrição detalhada das rotinas e métodos usados podem ser encontrados em www.anc.ed.ac.uk/~mjo/rbf.html, junto com exemplos ilustrativos.

2.2.2.

Lógica Fuzzy

O sistema lógico clássico é bastante poderoso para resolver problemas, principalmente em termos de um universo com dois valores (binário). O sistema lógico “fuzzy”, como extensão dos sistemas clássicos, consegue manipular problemas em que há vários estados de decisão, portanto incluindo o caso binário. Além disso, o sistema “fuzzy” trata os dados de forma “amigável”, fazendo uso de conceitos lingüísticos (mais, menos, maior,...) com uma forma de análise semelhante à do ser humano.

O procedimento de raciocínio “fuzzy” permite a realização em paralelo da combinação da parte antecedente da regra com a entrada. Isso faz com que o procedimento “fuzzy” seja apropriado para ser implementado em processadores paralelos.

A lógica fuzzy é muito boa na modelagem de sistemas, mas, tal como a rede RBF, apresenta inconvenientes na estimativa de parâmetros ótimos. Por outro lado,

um sistema “fuzzy” tem dificuldade em lidar com algumas características, como por exemplo, funções de pertinências e regras tipo and/or. Em contraposição, as redes neurais trabalham bem com grandes quantidades de características e classes [Knapp, 1996].

Desta forma, um sistema híbrido neuro-fuzzy utiliza vantagens das duas técnicas, fazendo que o sistema seja apto a tratar com dados incertos e nebulosos e que o sistema possua a habilidade do aprendizado juntamente com uma estrutura altamente paralela.

De uma forma geral, esta revisão procurou não entrar em detalhes em cada uma das técnicas tratadas anteriormente, evitando particularizar a aplicação das mesmas. No entanto deve-se realçar que a eficiência de cada uma das técnicas depende da escolha certa de seus parâmetros, o que usualmente implica em longos procedimentos de teste e de ajuste.

No presente trabalho foi usada uma rotina tipo ANFIS (toolbox Fuzzy-Matlab) a qual é uma função de treinamento usada para modelar um sistema fuzzy como uma rede neural com o ajuste de parâmetros respectivos. É importante aclarar, que o critério de parada dos treinamentos é feito segundo o erro de modelagem desejado ou o número de iterações. Pode acontecer, que o número de iterações tenha sido alcançado antes de satisfazer o erro de modelagem. Por outro lado, aumentar o número de iterações implica aumentar o tempo computacional.

Mais detalhes deste tipo de rotina podem ser encontrados em www.dainf.cefetpr.br/~myriam/papers/DemoEAnfis.pdf.