

PROGRAMAÇÃO GENÉTICA UNI E MULTI-OBJECTIVO PARA TREINO DE REDES NEURONAIS B-SPLINE

Cristiano Cabrita
ADEE, EST, Universidade do Algarve

Resumo – A fase de treino de uma rede neuronal B-spline e sistemas neuro-difusos é uma tarefa extremamente árdua. Algumas heurísticas existentes, nomeadamente o algoritmo ASMOD mostram ser altamente dependentes nas condições iniciais usadas. Deste modo, neste artigo é proposta uma nova estratégia, que pretende protagonizar a procura eficiente da topologia, em conjunto com a obtenção de soluções mais consistentes. A facilidade de incorporação de uma estratégia multi-objectivo também é explorada, permitindo obter modelos melhor condicionados, e mais adequados às intenções expostas.

1. INTRODUÇÃO

As redes neuronais B-spline oferecem vantagens sobre as mais comuns redes neuronais, tais como as MLP's ou as de funções de base radial. Redes B-spline guardam a informação localmente, o que significa que a aprendizagem numa parte do espaço de entrada minimamente afecta o resto. Por este motivo, são adequadas quer para aplicações de modelização *on-line* quer para efeitos de controlo. A sua estrutura baseada em grelha torna-as transparente, que ao contraste com outras redes, permite uma maior facilidade em compreender o conhecimento presente neste tipo de redes. Esta é uma vantagem também especificada pelos sistemas neuro-difusos sobre redes neuronais convencionais, e que permite que a determinado nível, os princípios básicos das redes B-spline e sistemas neuro-difusos sejam idênticos. Aliás, mediante certas condições os algoritmos são também semelhantes e este facto é apontado neste artigo.

O procedimento normal de projecto de redes B-spline envolve duas fases diferentes: a determinação da melhor topologia, que pode ser visto como um problema de identificação, e a determinação dos seus parâmetros, que pode ser reconhecido como um problema de estimação de parâmetros. O último ponto, a determinação dos parâmetros da rede, (nós interiores e pesos lineares) é uma tarefa simples e é resolvida utilizando esquemas como o gradiente ou esquemas híbridos (Brown and Harris, 1994). O primeiro ponto, a determinação da topologia, é uma tarefa bastante mais complexa, especialmente quando se trata de problemas reais. O objectivo é o de encontrar um modelo em termos computacionais eficiente que evidencie bons resultados no contexto da precisão. Algumas heurísticas foram propostas no passado, e uma delas, o algoritmo ASMOD, será usado para comparação.

Programação genética (PG) (Koza et. al, 1998), é indicada como uma estratégia eficiente na procura da topologia. Um compromisso entre a precisão e complexidade (entre outros objectivos) é formulada, não apenas segundo um único objectivo, mas também através de uma minimização multi-objectivo.

Nas secções seguintes são expostos os funcionamentos para os algoritmos ASMOD e PG, seguidamente os resultados e respectivas conclusões.

O leitor é convidado a consultar (Cabrita, 2001) a fim de conhecer a arquitectura subjacente a uma rede neuronal B-spline.

2. ALGORITMO ASMOD

O treino de uma rede B-spline ou de um sistema neuro-difuso envolve as seguintes fases para treino:

1. A determinação do número de sub-modelos;
2. O conjunto de entradas para cada sub-modelo;
3. A ordem das funções spline para cada entrada;
4. O número de nós interiores para cada entrada;
5. A localização dos nós interiores para cada entrada;
6. Os valores dos pesos lineares na saída.

A determinação dos últimos dois pontos pode ser encarado como um problema de mínimos quadrados, e logo, algoritmos de treino supervisionado podem ser aplicados para a sua determinação (para mais detalhes consulte Ruano et. al., 2001a). Os restantes pontos constituem um problema mais complexo. Diferentes algoritmos construtivos existem, tais como o algoritmo *ASMOD* (*Adaptive Spline Modelling of Observed Data*) (Weyer and Kavli, 1995), o algoritmo *MARS* (*Multivariate Adaptive Regression Splines*) (Friedman, 1991), e o *LOLIMOT* (Nelles, 2000). Este artigo compara o primeiro algoritmo com uma técnica de treino utilizando programação genética uni-objectivo (SOGP).

A figura 1 ilustra o funcionamento do algoritmo ASMOD. Como pode ser observado, a procura por uma estrutura optima envolve dois passos fundamentais, refinamento e poda, assim como , a avaliação dos candidatos e um critério de terminação. Se nenhum modelo inicial for fornecido, a estrutura inicial contém um ou vários sub-modelos univariáveis, com ordem especificada pelo utilizador, constituídos por 0 nós interiores. Uma descrição muito simples do algoritmo ASMOD é apresentada nas próximas secções, de modo que para uma explicação mais pormenorizada desta heurística, consulte (Cabrita, 2001).

2.1. Refinamento

Este passo cria estruturas mais complexas e consiste em 3 fases:

- *Adição de sub-modelos univariáveis*; sempre que existam variáveis de entrada ausentes do modelo corrente, novos sub-modelos com 0 nós interiores e de ordem especificada pelo utilizador são criados;
- *Criação de sub-modelos multi-variáveis*; sub-modelos de dimensão de entrada inferior são combinados para formar sub-modelos usando produtos tensoriais; a cada iteração, são criados tantos candidatos quanto as combinações entre variáveis presentes no modelo inicial;
- *Refinamento do vector de nós*; para todas as entradas de todos os sub-modelos, um novo nó interior é inserido a meia distância de dois nós adjacentes existentes.

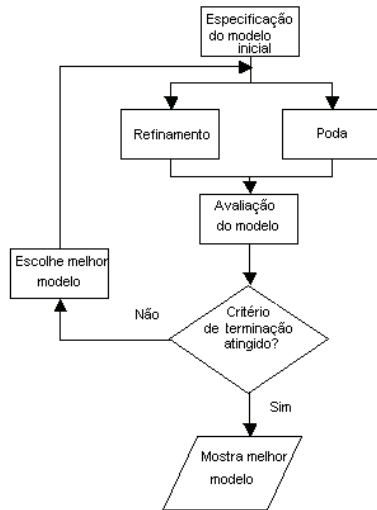


Fig 1: Esquema de funcionamento do algoritmo ASMOD

2.2. Poda

Um passo de poda, para reduzir a complexidade do modelo é executado menos frequentemente (tipicamente todas as 3 a 5 iterações). Três fases são igualmente consideradas:

- *Ordem de sub-modelos sem nós interiores é reduzida em uma unidade*; se os sub-modelos resultantes obtiverem ordem 1, então são removidos do modelo;
- *Decomposição de sub-modelos multi-variáveis*; qualquer sub-modelo cuja estrutura represente n variáveis combinadas é decomposto em n sub-modelos com $n-1$ variáveis combinadas;
- *Remoção de nós interiores*; esta fase é aplicada a todos os nós interiores para cada variável de todos os sub-modelos, criando um número de candidatos correspondente

2.3. Avaliação de candidatos e critério de terminação

- Esta heurística cria, a cada iteração, um número elevado de modelos candidatos. Este número cresce com a evolução do algoritmo ASMOD. Os modelos candidatos poderão ser avaliados segundo diferentes critérios, tais como *RMS* (Root-Mean-Square) para o conjunto de treino ou conjunto de teste, ou *validação cruzada*; O critério mais usado é, no entanto, o critério de informação, que pondera a precisão do modelo e a sua complexidade.

O critério utilizado neste trabalho é descrito como:

$$BIC = m \ln(RMS) + n \ln(m), \quad (1)$$

onde m denota o número de padrões de treino e n , a complexidade do modelo. A comparação entre o desempenho dos candidatos permite a escolha do melhor modelo. Quando o melhor candidato da iteração n tem pior desempenho que o melhor candidato da iteração $n-1$, a evolução termina.

3. PROGRAMAÇÃO GENÉTICA

Programação genética (PG) aplica os operadores principais utilizados por um algoritmo genético durante o procedimento de optimização. A principal diferença consiste na definição dos parâmetros da rede, os quais não são codificados por cadeias de bits, mas por uma estrutura em árvore, composta por nós *função* e nós *terminais*. A estrutura da árvore assim

como as características de cada nó evolui de geração em geração.

Uma condição necessária para que PG convirja para uma solução é que o conjunto de funções e terminais seja suficiente. No caso do projecto de redes B-spline, sub-modelos terão de ser *adicionados* (+), sub-modelos de maior dimensão de entrada terão de ser criados a partir de sub-modelos menos complexos (*), e sub-modelos de elevada dimensão de entrada terão de ser decompostos em sub-modelos de dimensão inferior (/). Este constitui o conjunto das funções primitivas que foram implementadas.

Ao contrário de outras redes neuronais, os nós terminais não representam apenas a variável de entrada, mas também a ordem da spline, o número de nós interiores, e respectiva localização.

A próxima figura apresenta um exemplo de uma estrutura em árvore para uma rede B-spline.

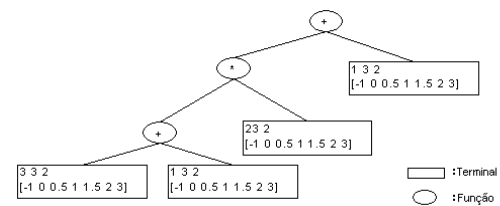


Fig 2: Exemplo de estrutura em árvore para uma rede B-spline.

O processo de evolução da PG envolve os seguintes passos:

- A criação da população inicial e a determinação do tamanho da população;
- A avaliação de candidatos e atribuição da sua aptidão;
- A aplicação de operadores genéticos, tais como:
 - selecção: Pares de árvores progenitoras são seleccionadas para reprodução baseadas no valor de aptidão;
 - cruzamento: Nós de duas árvores são seleccionados aleatoriamente e a troca de ambos produz um par de árvores progénitas.
 - mutação: Desempenhada quer pela substituição de um nó seleccionado aleatoriamente por uma sub-árvore gerada aleatoriamente ou alternado o seu tipo.
 - substituição: Todos os progenitores são substituídos pelos progénitos.
- O critério de terminação é o número máximo de gerações imposto.

O ciclo de evolução é sumarizado na figura 3. Cruzamento e mutação são aplicados mediante valores de probabilidade estabelecidos inicialmente e mantidos constantes. Mutação numa função implica a substituição de um nó da árvore por uma sub-árvore de comprimento máximo 2 gerada aleatoriamente. A mutação num terminal pode ser de 6 diferentes tipos, como sejam:

1. substituição integral do terminal.
2. substituição da identificação da variável.
3. substituição da ordem da variável.
4. deslocamento aleatório de um nó interior.
5. adição de N novos nós interiores colocados aleatoriamente. N é constante e igual a 5.
6. remoção de N nós interiores. Na ausência de nós interiores, nenhuma operação é executada.

Por motivo de simplificação, as taxas de mutação de terminal são descritas por um vector $p_m_terminal = [\%1 \ \%2 \ \%3 \ \%4 \ \%5 \ \%6]$, onde $\%i$ designa a taxa de mutação para o tipo de mutação i .

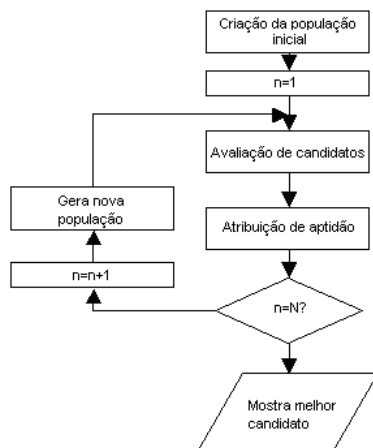


Fig 3: Esquema de funcionamento da Programação Genética.

Como um algoritmo evolutivo computacional necessita de ser previamente sintonizado, os valores para os parâmetros referidos anteriormente devem ser seleccionados. Um estudo preliminar foi conduzido e as principais conclusões atingidas são resumidas de seguida:

- Caso geral, quanto maior o tamanho da população, e o número de gerações utilizado, melhores os resultados obtidos;
- O algoritmo de selecção *stochastic universal sampling* forneceu os melhores resultados;
- Para a criação da população inicial, o método *ramped half-and-half* (Koza, 1998), revelou ser o mais adequado;
- Atribuição de aptidão baseada em *Exponential rank* com o valor 5 para a pressão selectiva mostrou um desempenho razoável, em conjunto com uma probabilidade de mutação de 0.8; este facto está de acordo com a taxa de mutação estabelecido pela estratégia adoptada em (Fonseca, 1998);
- Entre as mutações de terminal expostas anteriormente, a introdução de nós interiores revelou ser a mais importante;
- Quanto ao valor da probabilidade de cruzamento, um valor de 0.5 é o mais adequado.

3.1. Programação genética multi-objectivo

PG também pode ser aplicada numa estratégia multi-objectivo (MOGP), útil para o treino de redes neuronais B-spline. Existindo diferentes objectivos para minimização, uma única solução para o problema não é adequada, mas um conjunto de soluções alternativas, reflectindo diferentes compromissos, designados por soluções Pareto-óptimas ou soluções não-dominadas. Para uma descrição mais pormenorizada sobre optimização multi-objectivo, refira-se a (Cabrita, 2001).

Nesta abordagem, a avaliação consiste não apenas na avaliação das funções objectivo mas também no processo de decisão baseado em prioridades e valores desejados associados a cada objectivo. A cada geração, aptidão é atribuída em função do resultado do processo de decisão. As soluções não-dominadas são registadas e no final, o agente de decisão selecciona as melhores soluções não-dominadas que correspondam aos candidatos preferidos.

O critério de informação definido em (1) descreve um compromisso fixo entre a precisão e a complexidade do modelo. Os objectivos seguintes foram considerados para a estratégia multi-objectivo:

- Erro quadrático médio para dados de treino e teste (MSE_train/test).
- Erro relativo quadrático médio para dados de treino e teste (MSRE_train/test).
- Complexidade da rede.
- Norma Euclideana para os pesos lineares.
- Número de condição para a matriz de regressão.

A alguns objectivos são atribuídos valores de prioridade superior aos restantes. Objectivos com prioridade superior a 0, em conjunto com os valores desejados, são encarados como restrições que deverão ser satisfeitas. Objectivos com menor valor de prioridade são considerados desde que os restantes tenham satisfeito os respectivos valores desejados.

4. RESULTADOS

Foram utilizados dois problemas distintos para geração de resultados. O primeiro, o problema pH, cujo objectivo é a aproximação inversa da curva de titração. Este tipo de não linearidade relaciona o valor de pH (medida da actividade de iões de hidrógeno numa substância) e a concentração (x) de substâncias químicas. O segundo, o problema de transformação de coordenadas inversa (TIC), que ilustra a transformação cinemática inversa entre 2 coordenadas cartesianas e um dos ângulos de um robô de duas juntas.

Uma descrição mais completa pode ser consultada em (Cabrita, 2001).

4.1. Comparação entre PG uni-objectivo e ASMOD

Os resultados apresentados foram gerados por 50 diferentes condições iniciais. Para gerar as 50 condições iniciais para o algoritmo ASMOD, foram utilizadas duas gerações da PG uni-objectivo. Os valores apresentados nas tabelas 1 e 2 referem aos valores médios obtidos a partir de 50 diferentes execuções, excepto os valores apresentados na última coluna que mostram o valor mínimo do critério BIC após as 50 execuções. A primeira linha refere-se ao problema pH e a segunda linha ao problema TIC. Utilizando as 50 amostras é possível gerar a função de distribuição de probabilidade acumulativa empírica (f.d.p.a) como mostra a figura 4.

Tabela 1: Resultados para o algoritmo ASMOD.

BIC	$\ w\ $	MSE	Número de candidatos	Complexidade do modelo	Menor valor BIC
-2860	4.3	2.4e-04	5180	62	-6505
-715	135	4.2e-02	997	56	-1614

Para PG, o tamanho da população é 50, o método de criação da população inicial usado é *ramped-half-and-half*, o método de atribuição de aptidão é *exponential ranking* com pressão selectiva 5, taxa de cruzamento, mutação e número de gerações de 0.5, 0.8, 20, respectivamente. A taxa de mutação de terminais é [5% 10% 5% 10% 60% 10%]. $\|w\|$ denota a norma-2 do vector de pesos lineares.

Tabela 2: Resultados para SOGP.

BIC	$\ w\ $	MSE	Número de candidatos	Complexidade do modelo	Menor valor BIC
-1950	86.3	6.7e-10	988	48.2	-2065
-1192	7.5e+7	7.7e-07	886	94.9	-1451

A comparação entre o ASMOD e SOGP revela que não existe qualquer garantia que as soluções obtidas representem um óptimo global. Observa-se que o algoritmo ASMOD é suplantado pelo SOGP na medida em que SOGP fornece modelos consistentes mais frequentemente. Por outro lado, a curva da função distribuição de probabilidade acumulativa

empírica mostra que cerca de 50% dos modelos fornecidos pela PG são razoavelmente “bons” enquanto que entre 60% e 90% das soluções obtidas pelo ASMOD revelam-se piores que SOGP para os problemas pH e TIC, respectivamente. Também se observa que o algoritmo ASMOD é muito dependente da estrutura do modelo inicial embora obtenha muito melhores soluções, ocasionalmente. É de notar que para o problema pH, o algoritmo ASMOD implementado, em média, gera 5 ou mais vezes o número de candidatos gerados pela SOGP.

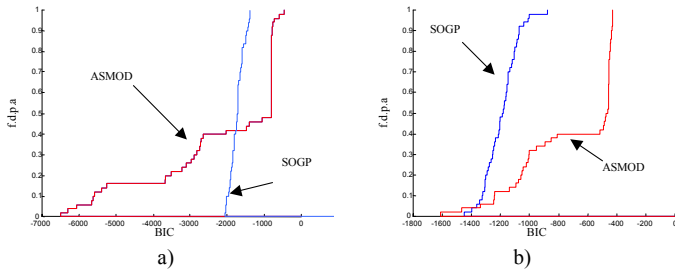


Fig 4: Função distribuição de probabilidade empírica acumulativa para o algoritmo ASMOD e SOGP para: a) problema pH; b) problema TIC.

4.2. Comparação entre SOGP e MOGP

Observando os valores médios apresentados na tabela 2, pode-se afirmar que algumas das especificações de desempenho revelam modelos mal condicionados embora tais resultados tenham surgido por motivo de 2 (entre 50) candidatos serem muito mal condicionados. Uma forma de estabelecer um compromisso entre algumas das especificações de desempenho é usando a estratégia multi-objectivo. Cinco sessões foram executadas, para ambos problemas, baseadas nos parâmetros da tabela 3.

Tabela 3: Especificação dos parâmetros Multi-objectivo

Problema	Número de gerações	Vector de prioridades	Vector de valores desejados
PH	40	[0 0 2 1 1]	[10 ⁻¹⁰ 10 ⁻⁵ 20 4 6]
TIC	20	[0 0 2 1 1]	[10 ⁻⁷ 10 ⁻³ 60 1000 10 ¹⁴]

Cada solução não dominada de cada execução foi guardada e no final, o agente de decisão mostra as soluções preferidas entre as 5 execuções, baseando-se nos vectores de prioridade e valores desejados, definidos no início das execuções. O agente de decisão refinou os valores desejados (observado na 3ª coluna da grelha, e correspondente às marcas com “X” nos gráficos), e, entre todas as soluções guardadas, escolheu as apresentadas nas figuras 5 e 6, para cada problema.

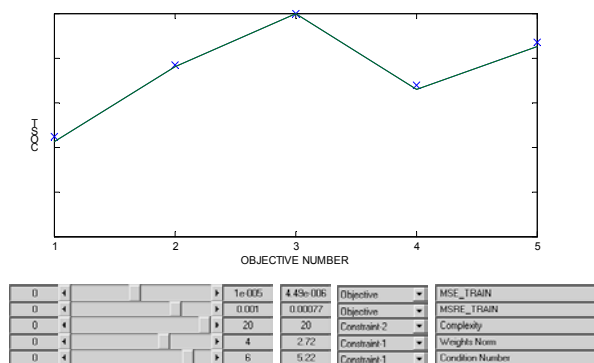


Fig 5: Solução escolhida para o problema pH.

Os resultados obtidos mostram que é possível manter uma razoável qualidade da saída para o modelo, aumentando a condição respectiva. Pretende-se mostrar que é possível

impor um limite superior nos valores da complexidade, norma dos pesos e número de condição, mantendo valores aceitáveis para MSE e MSRE. Esta abordagem é muito promissora porque permite fornecer uma forma eficiente no modo de obtenção de modelos muito bem condicionados.

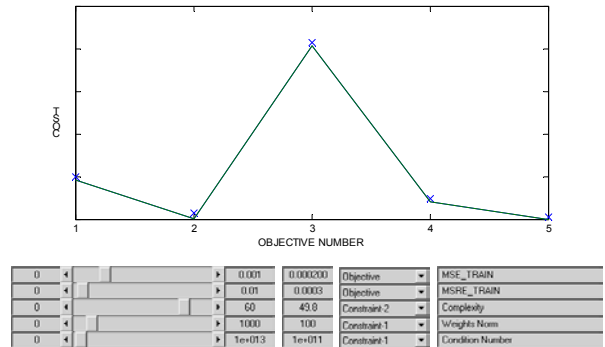


Fig 6: Solução escolhida para o problema TIC.

5. CONCLUSÕES

Neste artigo conclui-se que o algoritmo ASMOD, a técnica mais comum para treino de redes B-spline, nem sempre obtém soluções aceitáveis, observando-se que a estrutura do modelo inicial desempenha um papel importante na evolução do algoritmo. O paradigma da programação genética é uma alternativa viável, desempenhando uma procura eficiente e fornecendo modelos que apresentam maior consistência. A facilidade de incorporação de uma estratégia multi-objectivo em PG permite ao projectista a liberdade de obter melhores modelos mais adequados à suas intenções.

Esta abordagem pode ser facilmente modificada para outros dos mais comuns modelos neuro-difusos, e também para outras arquitecturas de redes neuronais. A abordagem multi-objectivo pode ser combinada com as técnicas do gradiente, num esquema híbrido, para encontrar a melhor topologia e fornecer os melhores valores iniciais para os parâmetros da rede.

REFERÊNCIAS

- Brown, M., Chris Harris (1994) *Neurofuzzy Adaptive Modelling and Control*, Prentice-Hall
- Cabrita, C. L., (2001), *Algoritmos construtivos para treino de redes neuronais B-spline*, Tese de Mestrado, FCT, Universidade do Algarve.
- Friedman, J.H. (1991) Multivariate Adaptive Regression Splines, *The Annals of Statistics*, **19** (1), 1-141
- Koza, J. R. (1998a) *Genetic Programming II, Automatic Discovery of Reusable Programs*, 2nd ed., MIT.
- Fonseca, C. M., Fleming, P.J. (1998a) Multiobjective optimization and multiple constraint handling with evolutionary algorithms I: A unified formulation, *IEEE Transactions on SMC, Part A*, **28**, (1) 26-37
- Weyer E., T. Kavli, (1995) *The ASMOD Algorithm. Some New Theoretical and Experimental Results*. SINTEF Report STF31 A95024, Oslo
- Ruano A. E., C. Cabrita, J. V. Oliveira, L. T. Kóczy, (2001a) Supervised training algorithms for b-spline neural networks and neuro-fuzzy systems, submitted to *International Journal of Systems Science*
- Nelles, O., (2000), *Nonlinear Systems Identification with Local Linear Neuro-Fuzzy Models*, PhD. Thesis, TU Darmstadt, Germany