

Redes Neurais

Germano C. Vasconcelos

Lista de Exercícios Adicionais

1. Considere o cadastro de investidores abaixo. Construa uma rede *multilayer perceptron* para prever se cada investidor investirá em ações. C1 = Classe 1; C2 = Classe 2.

Nome	Renda Alta	Investe em Fundos?	Juros do Mercado Alto?	Investirá em Ações
Fred	S	S	N	Sim
Barney	N	N	S	Não
Wilma	S	S	S	Não
The Great Gazoo	N	S	S	Sim

2. Considere agora o mesmo cadastro e resolva o problema com uma rede RBF.

Nome	Renda Alta	Investe em Fundos?	Juros do Mercado Alto?	Investirá em Ações
Fred	S	S	N	Sim
Barney	N	N	S	Não
Wilma	S	S	S	Não
The Great Gazoo	N	S	S	Sim

3. Solucione o mesmo problema com a definição dos parâmetros para treinamento de uma rede de Kohonen (SOM). Considere o material de Aula para definir:

- Tamanho da grade
- Valor inicial dos pesos
- Valor inicial da taxa de aprendizagem e valor final (0.01)
- Valor inicial da vizinhança
- Equação de ajuste dos pesos com os valores dos parâmetros. Considere $\Lambda_{i,l}=1$.
- Função de redução de vizinhança para a fase de ordenação (vizinhança na fase de convergência=0)
- Função de redução da taxa de aprendizagem

Nome	Renda Alta	Investe em Fundos?	Juros do Mercado Alto?
Fred	S	S	N
Barney	N	N	S
Wilma	S	S	S
The Great Gazoo	N	S	S

Nesse caso, não há como usar a rede para previsão mas pode ser usada para agrupar os investidores por semelhança e permitir análise de características.

Algoritmo de Aprendizagem



- Inicializa-se a rede:
 - pesos (aleatórios entre $[0,1]$), raio da vizinhança, taxa de aprendizagem e taxa de redução da vizinhança
- Apresenta-se padrão de entrada
- Calcula-se distância Euclideana do vetor de entrada a cada neurônio j de saída
- Seleciona-se o neurônio n_{j^*} de menor distância d_{j^*}
- Adapta-se os pesos do neurônio n_{j^*} e da sua vizinhança N_{j^*} , segundo a regra:
 - $w_{ij}(t+1) = w_{ij}(t) + \eta(t)[x_i(t) - w_{ij}(t)] \cdot \Lambda_{i,j^*}$
- Reduz-se vizinhança e taxa de aprendizagem (convergência)
- Apresenta-se outro padrão



Fase de Ordenação



- Com N_0 iterações, função de vizinhança decresce em geral linearmente com um raio definido por:

$$\sigma(n) = \sigma_0(n) \left(1 - \frac{n}{N_0} \right)$$

- Normalmente a taxa de aprendizagem é alta (acima de 0.1) para permitir à rede se auto-organizar. Ela também é linearmente ajustada:

$$\Delta\eta(n) = \eta_0 \left(1 - \frac{n}{(N + K)} \right)$$

- η_0 é taxa de aprendizagem inicial e K é taxa final



Fase de Convergência



- Fase mais demorada, se mantém taxa de aprendizagem pequena (0.01) e usa-se menor vizinhança (somente neurônio ou seus vizinhos mais próximos)
- Escolha do número de neurônios é experimental. Número de saídas afeta a precisão do mapeamento e tempo de treinamento
- Aumento do número de neurônios aumenta a resolução mas aumenta muito tempo de treinamento

4. Treine uma rede ART para resolver o mesmo problema. Nesse caso, não há como usar a rede para previsão mas pode ser usada para agrupar os investidores por semelhança e permitir análise de características. Considere:

- n = tamanho do vetor de entrada estendido
- $L=2$
- P (parâmetro de vigilância)= 0.6

Nome	Renda Alta	Investe em Fundos?	Juros do Mercado Alto?
Fred	S	S	N
Barney	N	N	S
Wilma	S	S	S
The Great Gazoo	N	S	S

Algoritmo de Treinamento



- Inicializar todas as conexões $W(0) = 1/(1+n)$ e $B(0) = 1$ (onde n =tamanho do vetor estendido);
- Repita
 - Para cada padrão de treinamento (X)
 - Entrar na fase de reconhecimento
 - Comparar protótipo do vencedor com a entrada
 - Se comparação $>$ valor de vigilância
 - Então Atualizar pesos $W(t+1) = LZ / (L-1 + \sum z)$ e $B(t+1) = B(t).X(t)$ do neurônio vencedor
 - Senão Desabilitar neurônio (Busca)
 - Se ainda existir neurônio não analisado
 - Então Voltar
 - Senão Alocar novo neurônio à camada de saída
- Até conjunto de protótipos não mudar

