

Lista 07 – IA

01)

O algoritmo de backpropagation possui duas fases essenciais: fase forward e fase backward. O fase forward propaga, usando uma função de ativação, as saídas dos valores da camada anterior (com exceção da primeira) como entradas da próxima camada, até a última onde é obtido a resposta da rede e o erro é calculado. É medido a influência de cada peso no erro por derivação e, depois disso, vem o fase backward que vai voltar pela rede ajustando os pesos de cada conexão, de acordo com o cálculo dos erros.

02)

A-

Resultados dos testes:

Todos eles estavam com learning rate de 0.3 e 500 epochs

1º Teste: 2 camadas, 6 neurônios.

Erro de 0.0092 por epoch

2º Teste: 2 camadas, 7 neurônios

Erro de 0.0093 por epoch

3º Teste: 2 camadas, 10 neurônios

Erro de 0.0094 por epoch

4º Teste: 3 camadas, 7 neurônios

Erro de 0.01111 por epoch

5º Teste : 4 camadas, 10 neurônios (2,3,2,3)

Erro de 0.0139 por epoch

6º Teste: 2 camadas, 5 neurônios (2,3)

Erro de 0.0094 por epoch

Conclusões: em uma rede neural, as camadas e os neurônios afetam diretamente os erros. No caso, ter mais nem sempre é melhor, nem “menos é mais”. Ou seja, para chegar a um resultado bom é necessário a realização de vários testes. Dentre os meus exemplos, o padrão do weka ($'a' = (\text{atributos} + \text{classes}) / 2$) foi a melhor opção.

B-

Todos eles estavam com 2 camadas, 6 neurônios e 500 epochs

1º Teste: learning rate 0.3

Erro de 0.0092 por epoch

2º Teste: learning rate 0.1

Erro de 0.0093 por epoch

3º Teste: learning rate 0.4

Erro de 0.0094 por epoch

4º Teste: learning rate 0.5

Erro de 0.0100 por epoch

5º Teste : learning rate 0.8

Erro de 0.0106 por epoch

6º Teste: learning rate 1

Erro de 0.0108 por epoch

Conclusões: em uma rede neural, o learning rate afeta a quantidade de erros. Assim como no exemplo anterior, ter muito ou pouco, nem sempre é melhor. Novamente, depende da rede. Dentre os meus exemplos, o padrão do weka (learning rate = 0.3) foi a melhor opção.

C-

A normalização é uma etapa importante para a rede neural. Ela altera os valores para uma escala comum pois os valores limites estão muito distintos. Realiza isso sem distorcer as diferenças dos atributos e com o objetivo de que um atributo predomine sobre o outro. Quando altera-se a normalização para false no weka, os valores dos erros sobem, quase dobrando.

D-

Acredito que o atributo de 'training time' também é bastante importante para análise de uma rede. Caso ele seja mal calculado, pode acontecer um underfitting (falta de aprendizado) ou um overfitting (memorização dos dados).

03)

A-

A binarização dos dados transforma nominal para numérico, com saídas de verdadeiro ou falso para cada uma das categorias nominais. Isso ajuda a aumentar a quantidade de neurônios e ajuda a rede a aprender de forma melhor.

B-

A rede possui 4 entradas. Cada uma delas é uma das categorias nominais do atributo. Por exemplo, outlook pode ser sunny, overcast ou raining. Ele coloca cada uma delas como uma entrada diferente.

04)

Grid search gera valores uniformemente espaçados para cada hiperparâmetro testado e, em seguida, usa validação cruzada para testar a precisão de cada combinação.

CVPParameterSelection: só pode otimizar um parâmetro numérico (primeiro nível) especificado por seu nome de linha de comando. Ele usa a taxa de erro como métrica de avaliação.

05)

Usei o CVParameterSelection e também usei dois algoritmos de redes diferentes: o MLP e o SMO. Os dois estavam na forma padrão do weka. O MLP teve melhor desempenho (matriz de confusão foi [7 2] [2 3]) e tentei afiná-lo chegando a conclusão de que os melhores parâmetros são os default do weka.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      10          71.4286 %
Incorrectly Classified Instances     4          28.5714 %
Kappa statistic                     0.3778
Mean absolute error                  0.287
Root mean squared error              0.5268
Relative absolute error              60.2616 %
Root relative squared error         106.7798 %
Total Number of Instances           14

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,778	0,400	0,778	0,778	0,778	0,378	0,778	0,885	yes
	0,600	0,222	0,600	0,600	0,600	0,378	0,778	0,698	no
Weighted Avg.	0,714	0,337	0,714	0,714	0,714	0,378	0,778	0,818	

```
=== Confusion Matrix ===

 a b  <-- classified as
 7 2 | a = yes
 2 3 | b = no
```