



Reconhecimento de A e A invertido com Adaline
Profa. Dra. Roseli Aparecida Francelin Romero
Disciplina: SCC 0270 - Redes Neurais

Lucas Yudi Sugi - 9293251

Introdução

O algoritmo de aprendizado supervisionado Adaline permite realizar a classificação dos dados em duas classes distintas, desde que as mesmas sejam linearmente separáveis, ou seja, somente é possível separar os dados através de um único hiperplano.

Dado isso, esse algoritmo será utilizado para resolver o problema de classificação das letras A e A invertido. Faremos uma base de dados contendo 12 exemplos (6 de cada) e os utilizaremos para treinar o modelo, gerando uma função que será capaz de classificar os dados.

Objetivos

Este documento visa explicar a utilização do algoritmo Adaline para reconhecimento das letras A e A invertida, sendo abordados os seguintes pontos:

- Decisões de implementação
- Formulação da base de treinamento e teste
- Validação da função estimada pelo algoritmo

Implementação

Algumas considerações da implementação são necessárias para que haja um funcionamento adequado do algoritmo:

- Na geração randômica dos pesos foi utilizado valores que são amostrados uniformemente no intervalo $[-0.5, 0.5]$. Isso é necessário para que o resultado da net não seja demasiadamente grande. Caso seja utilizado valores maiores, a acurácia média tende a cair.
- Eta: o seu valor é pequeno para que não haja “saltos” muito grandes na atualização da derivada (o que pode dificultar o aprendizado, mesmo que ele acabe se tornando mais lento).
- Threshold: o limiar (condição de parada do Adaline) foi considerado como zero. Nesse caso, isso significa que no aprendizado o algoritmo deve, necessariamente, acertar todos os exemplos.

Base de treinamento - Estrutura dos dados

As letras a serem reconhecidas seguem o formato abaixo:

```

-1
+1-1-1-1-1+1
+1-1-1-1-1+1
+1+1+1+1+1+1
+1-1-1-1-1+1
+1+1-1-1+1+1
-1-1+1+1-1-1

```

Figura 1: Letra A invertida formado por +1 e -1.

É importante salientar que, a letra deve necessariamente ser formado por +1 e por -1 (com os sinais juntos), pois a leitura dos dados faz essa consideração. Ademais, note que existe um valor acima da matriz (-1), este é utilizado para indicar a classe (-1 para A invertido e +1 para A normal) que a matriz pertence.

Caso o leitor deseje realizar testes com suas próprias letras, o mesmo deve seguir a mesma estrutura de matriz e da disposição das letras, conforme o arquivos train.txt ou test.txt.

Base de treinamento - Construção

A construção da base de treinamento levou em conta a estratificação dos dados, isto é, as classes foram dispostas de maneira a não haver supremacia de nenhuma sobre a outra. Exemplificando: Nas 12 letras geradas (6 para A e 6 para A invertida), 4 de cada classe foram para a base de treinamento e o resto para a de testes. Caso, fizéssemos algo como: 6 letras A mais 2 letras A invertida para a base de treinamento e o resto para teste, estaríamos privilegiando uma classe, o que afetaria o aprendizado.

Testes

Para a realização dos testes foi utilizado o método holdout com as famosas proporções ($\frac{2}{3}$ para treinamento $\frac{1}{3}$ para teste). Realizou-se um total de 1000 testes, que verificaram a acurácia (porcentagem de acerto) no conjunto de testes. Abaixo há o valor médio da acurácia e o seu desvio padrão estimados da amostra de 1000 testes:

Média: 0.979

Desvio padrão: 0.07621679604916491

Conclusão

Pelos resultados do teste realizado acreditamos que o algoritmo adaline conseguiu “aprender” a classificar um exemplo como A ou A invertido na maior parte das

vezes. Note que dado a média e o desvio padrão podemos garantir um acerto de aproximadamente 90%. Salientamos, contudo, que testes estatísticos mais precisos como intervalo de confiança deveriam ser realizados para uma avaliação mais real do modelo.