Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação

Trabalho 1 - Redes Neurais

Aluno: Zoltán Hirata Jetsmen - 9293272

São Carlos
2018

1 Introdução

Este trabalho possui como objetivo implementar um algoritmo para reconhecer os caracteres **A**, **1**, e **A invertido**, **-1**.

2 Considerações Iniciais

Estão disponíveis dois arquivos .txt, onde se encontram os dados de treino, train.txt e test, test.txt onde cada um possui 6 exemplos, 3 de cada classe. Cada carácter é representado por uma matriz 5x5, onde o desenho dele é formado por números 1 e os espaços em branco por números -1. A última coluna representa a classe de cada exemplo.

Como parâmetros, temos os valores de α e de threshold. O primeiro representa o 'passo' que será dado na função *backward* (citada na próxima sessão) para alterar os pesos e o segundo se refere ao valor do erro que será suficiente para que o algoritmo pare as suas iterações. Assim, foi defino os seguintes valores:

- $\alpha = 0.1$
- threshold $= 0.001$

Ao rodar o programa, ele irá realizar os seguintes passos: realizar a leitura dos conjuntos de treino e teste, treinar os pesos até atingir um determinado erro, imprimindo o valor do erro a cada iteração e aplicar os pesos no conjunto de teste para que possamos visualizar se o treinamento efetuado foi realizado com sucesso obtidos.

3 Implementação

Para a implementação foi utilizado a linguagem de programação *Python*, como foi requerido na especificação do trabalho. Algumas bibliotecas também foram adicionas de forma a ajudar na implementação.

- **Pandas:** Foi utilizada somente a função *Pandas.read_ csv* para realizar a leitura dos dados.
- **Numpy:** Utilizada para realizar algumas operações matemáticas e aleatorizar os dados de entrada.
- **Random:** Utilizada para gerar os pesos aleatórios.
- **Collections:** Foi usada a função *namedtuple* para gerar um tupla entre os valores de X (entrada) e Y (saída). Apenas para melhorar a 'estética' do código

O código foi dividido em 4 funções onde cada uma realiza um papel específico no código.

- **perceptron:** Função principal do código, onde os pesos são inicializados aleatoriamente de acordo com a distribuição normal, indo de -1 até 1, de acordo com a dimensão dos dados de treinamento e onde ocorre as chamadas das funções *forward* e *backward* (listadas abaixo) dentro de uma iteração que verifica o erro gerado e valor de *threshold* passado.
- **forward:** Onde ocorre o processo de *forward*, que é a aplicação dos pesos nos valores de entrada, gerando uma saída.
- **backward:** Onde é feita a adequação dos pesos de acordo com a derivada das função de erro e o cálculo do erro.
- **tanh:** Aplica a função tangente hiperbólica em um valor de 'X' passado como parâmetro, sendo esta utilizada como função de ativação nos neurônios.
- **readDataset:** Função que lê um arquivo que representa o conjunto de dados. Este arquivo é divido em X e Y, onde o primeiro representa a entrada e o segundo, a saída para esta estrada. Os exemplos são aleatorizados para que exemplos de uma mesma classe não juntos em sequência.

4 Resultados:

Após rodar o algoritmo até atingir um erro menor que o valor passado como threshold, o arquivos *teste.txt* foi aplicado na função *forward* para gerar as saídas previstas pelo algoritmo. Os resultados gerados foram bem satisfatórios, tendo 100% de acerto em 3 execuções.

Saídas esperadas	1	1	1	1	-1	-1	-1	1	1	-1	-1
Saídas geradas	1.00	1.00	1.00	1.00	-1.00	-1.00	-1.00	1.00	1.00	-1.00	-1.00

Tabela 1: Resultados da primeira execução.

Saídas esperadas	-1	-1	1	-1	1	-1	1	1	-1	1	1
Saídas geradas	-1.00	-1.00	1.00	-1.00	1.00	-1.00	1.00	1.00	-1.00	1.00	1.00

Tabela 2: Resultados da segunda execução.

Saídas esperadas	-1	1	-1	1	-1	1	1	-1	1	-1	1
Saídas geradas	-1.00	1.00	-1.00	1.00	-1.00	1.00	1.00	-1.00	1.00	-1.00	1.00

Tabela 3: Resultados da terceira execução.