UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

SSC0600 – Introdução à Ciência de Computação I – 2025 (ICC1 - Teoria)

Professor responsável: *Fernando Santos Osório*
Semestre: 2025/2
Horário: Terça 10h – 13h (Prova no Lab.)

Wiki: SSC-600-2025(fosorio)
Web: ***http://www.icmc.usp.br/~fosorio*** **/**
DATA DA PROVA: 28 / 04 / 2025 - P1

NRO. USP: **< Colocar o seu NUSP no programa fonte>** COMO COMENTÁRIO no CÓDIGO

NOME : **< Colocar o seu Nome no programa fonte>** COMO COMENTÁRIO no CÓDIGO

>> COLOCAR SEU NOME E NRO. USP COMO COMENTÁRIO DO PROGRAMA FONTE ENTREGUE!

PROVA P1 – SSC0600 - ICC 1

Q1 – Perceptron: Multiplicação de Vetores – A base das Redes Neurais. [Peso 0.4]
Implemente o seguinte programa descrito abaixo de acordo com o especificado.
O programa deve ser enviado para o RunCodes.ICMC da Disciplina SSC0600 (Prova P1 – Q1)
RunCodes SSC0600 - https://runcodes.icmc.usp.br/offerings/view/96

Perceptron: Multiplicação de Vetores (Redes Neurais Artificiais – Neurônio Artificial)

Descrição inicial: Mais abaixo vamos detalhar esta descrição em uma linguagem mais "clara"

Implemente um programa que simule UM neurônio artificial do tipo PERCEPTRON, onde é fornecido um vetor de dados de entrada numérico (Vetor de Xi = números em ponto flutuante), um vetor de dados numéricos com os respectivos "pesos sinápticos" (Vetor Wi = números de ponto flutuante). Além disso é fornecido um valor do BIAS (um valor único de ponto flutuante) e um LIMIAR, usado para determinar se o "neurônio" dispara ou não sua saída, usualmente usamos 0.0 (zero) como sendo o limiar de disparo do neurônio. No nosso caso vamos adotar Zero como sendo o LIMIAR de disparo.

De modo mais direto e matemático: os neurônios artificiais do tipo **Perceptron** implementam o produto escalar de 2 vetores (*dot product*), sendo uma soma ponderada das entradas (Xi) multiplicadas pelos pesos sinápticos (Wi), somando com o BIAS e testando se o valor final é maior, menor ou igual à um limiar de saída (> 0, < 0, igual a 0). A equação abaixo representa o cálculo do produto escalar:

o produto escalar entre **A** e **B** é:

$$\mathbf{A} \cdot \mathbf{B} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

A equação abaixo representa a saída de um Perceptron: Produto do Vetor X multiplicado pelo Vetor W, Adicionando o BIAS e testando se >, < ou = ao limiar (0.0):

$$c = \sum_{i=1}^{n} w_i \cdot x_i \quad \text{+ BIAS}$$

Vetor X = { x1, x2, x3, x4, ... }
Vetor W = { w1, w2, w3, w4. ... }
C = x1 * w1 + x2 * w2 + x3 * w3 + x4 * w4 + ... + BIAS

Se C > 0.0 a Saída é Ativada

Se C < 0.0 a Saída não é Ativada

Se C = 0.0 a Saída é indefinida, ou no nosso caso, **vamos considerar que não é Ativada**

Simples, não? A saída é a soma ponderada das entradas vezes os pesos, mais um valor de BIAS e testando se for maior que zero ativa, e se for igual ou menor que zero não ativa. Isso permite criar um classificador de padrões! Depois da prova vamos discutir mais sobre isso...

O nosso vetor X[i] e o vetor W[i] terão um **tamanho máximo de 100 valores**, mas nem sempre vamos usar todo o vetor. O primeiro valor de entrada vai indicar QUANTOS valores serão lidos para o vetor X e W, que irão ter ambos o mesmo número de valores armazenados.

E como vai funcionar o programa em termos de ENTRADAS e SAÍDAS?

- O primeiro valor lido do teclado será um valor inteiro indicando quantos valores de X e W serão lidos. O valor máximo que pode ser indicado é 100 (valor entre 1 e 100).

- O segundo valor lido é o valor do BIAS, que será um valor de ponto flutuante.

- Segue uma lista de valores das entradas X, tantos quanto for a quantidade indicada no primeiro valor lido. Os valores das entradas X são valores de ponto flutuante.

- Segue uma lista de valores dos pesos W, tantos quanto for a quantidade indicada no primeiro valor lido. Os valores dos pesos W são valores de ponto flutuante.

O programa calcula e Soma Ponderada de X.W (C sendo a composição dos valores calculada com o produto escalar). É adicionado o BIAS ao valor da soma ponderada. O resultado final desta operação é exibido na tela com 2 casas após a vírgula.

Se o valor exibido for maior que 0.0 é exibida na linha seguinte a mensagem ATIVADO.

Se o valor exibido for menor ou igual a 0.0 é exibida na linha seguinte a mensagem INATIVO.

Exemplo:

Entrada:

```
2
0.7
-1.0
-1.0
1.0
1.0
```

Saída:

```
-1.30
INATIVO
```

Entrada:

```
2
0.7
1.0
1.0
1.0
1.0
```

Saída:

```
2.70
ATIVADO
```

Q2 – Codificar e Decodificar STRINGs (Criptografia com Cifra de Cesar e Inverter). [Peso 0.6]
Implemente o seguinte programa descrito abaixo de acordo com o especificado.
O programa deve ser enviado para o RunCodes.ICMC da Disciplina SSC0600 (Prova P1 – Q2)
RunCodes SSC0600 - https://runcodes.icmc.usp.br/offerings/view/96

Codificando e Decodificando Strings:

>> Atenção: este programa deve usar FGETS, pois as strings podem conter espaços em branco!

>> Atenção: o tamanho máximo da string a ser codificada/decodificada é de 150 caracteres.

Faça um programa que leia um "código de operação" (string com um caracter) que indica qual operação o usuário deseja realizar:

- Se o código for "C" ou se contiver a letra 'C', o programa deve criptografar a string lida a seguir, usando a cifra de Cesar (soma +1 ao código ASCII dos caracteres), e inverte a ordem de escrita (colocar na ordem inversa as letras, de trás para a frente), codificando assim a string lida.

Esta opção escreve na tela a string criptografada ANTES de ser invertida, e a string criptografada DEPOIS de ser invertida, uma em cada linha da tela, de acordo com os exemplos abaixo.

- Se o código for "D" ou se contiver a letra 'D', o programa deve descriptografar a string lida a seguir, usando a cifra de Cesar (subtrai -1 ao código ASCII dos caracteres), e inverte a ordem de escrita (colocar na ordem inversa as letras, de trás para a frente), restituindo assim a string original;

Esta opção escreve na tela a string descriptografada ANTES de ser invertida, e a string descriptografada DEPOIS de ser invertida, uma em cada linha da tela, de acordo com os exemplos abaixo.

- Se o código for "F" ou se contiver a letra 'F', o programa deve terminar.

O programa deve ficar em laço ("loop") lendo o código de operação, seguido da string, até que o código de operação indicado seja "F" / 'F'. Sendo assim, vamos poder criptografar ou descriptografar várias strings.

Atenção ao uso do FGETS: As strings deve conter apenas o '\0' ao serem manipuladas, portanto elimine TODO caracter que não seja padrão, ou seja, caracteres com código ASCII abaixo do 32 (ou abaixo do espaço em branco ' '). A string deve ter apenas caracteres "visíveis" ao ser criptografada e descriptografada, sem '\n', '\r' ou outros caracteres especiais ASCII.

Exemplos de execução:

Entrada:
```
C
HELLO
F
```
Saída:
```
IFMMP
PMMFI
```

Entrada:
```
D
PMMFI
F
```
Saída:
```
OLLEH
HELLO
```

Entrada:

```
C
HELLO
D
PMMFI
F
```

Saída:

```
IFMMP
PMMFI
OLLEH
HELLO
```

Entrada:

```
C
HELLO WORLD
F
```

Saída:

```
IFMMP!XPSME
EMSPX!PMMFI
```

BOA PROVA!!!

REGRAS EM RELAÇÃO REALIZAÇÃO DESTA PROVA

1. A PROVA É **INDIVIDUAL com consulta Papel (Livros, papel impresso ou escrito) e formato Digital (Internet, Wiki, Pendrive), porém SEM CONSULTAR ou SE COMUNICAR COM HUMANOS ou QUALQUER OUTRA FORMA DE VIDA TERRESTRE ou EXTRA-TERRESTRE!**
   Não podem usar de formas de comunicação com pessoas externas, além do professor, referente a prova, seja por celular (manter desligado/guardado), por e-mail, por whatsapp, por mensagens ou fóruns ("ao vivo"), com colegas, etc.
   NÃO É PERMITIDO O EMPRÉSTIMO DE MATERIAL (Cadernos, Anotações, Livros, etc).

2. **A PROVA DEVE SER REALIZADA USANDO OS COMPUTADORES DO LABORATÓRIO.**
   Não será permitido o uso de Notebooks, Tablets ou Celulares de uso pessoal.

3. A PROVA DEVE SER **ENTREGUE via RunCodes.icmc (pode submeter várias vezes)**
   Entrega é via https://runcodes.icmc.usp.br/ Disciplina SSC0600 – Prova P1 – Q1 e Q2

4. **EM CASO DE PROBLEMAS GRAVES no RunCodes.icmc pode ser entregue por E-Mail** para:
   fosorio@icmc.usp.br com cópia (**Cc:**) para fosorio@gmail.com => **PROFESSOR DEVE SER AVISADO!**
   Com assunto/subject: **Prova-P1 <Seu_Nome> <Nro_USP>**
   ANEXANDO O PROGRAMA FONTE (".C" e o Project ".**CBP") – NÃO ANEXAR EXECUTÁVEIS!**
   NÃO ANEXAR OBJ, BIN e EXE!!! Seu e-mail não será recebido se for anexado bin, obj ou exe!

5. A PROVA USUALMENTE DEVE SER REALIZADA USANDO O CODEBLOCKS DO LABORATÓRIO, se for usar outro Compilador/IDE, INDIQUE NO E-MAIL qual AMBIENTE, COMPILADOR e IDE usou!

6. **AO ENTREGAR A PROVA NO RUNCODES ou POR E-MAIL O(A) ALUNO(A) CONCORDA COM ESTAS REGRAS E SE COMPROMETE A FAZER A PROVA INDIVIDUALMENTE!**

FIM