

Universidade Federal do Rio de Janeiro
Relatório de Algoritmos e Programação

Grupo 5

Alunos: Pedro, Wagner, Afonso, Jefferson e Joelson

Planejamento: O trabalho foi realizado em conjunto com todos os integrantes em comunicação constante via Discord principalmente, tendo duas frentes principais distribuídas entre o grupo: a parte de programação bruta e a parte de correções, organização e testes do código.

Projeto Wisard

Parte I: Ideia e funcionamento

- **Ideia central do projeto:** A ideia deste projeto consiste na criação de um programa de aprendizado de máquina através de exemplos que simule o dispositivo de reconhecimento WiSARD de forma dinâmica para que o usuário consiga usufruir de seus aspectos em sua totalidade.

O aprendizado de máquina compreende em apresentar exemplos de determinada classe, para um programa de computador, o programa irá classificar novas observações como pertencente ou não sobre essa classe, e qual classe ela pertence. A WiSARD classifica-se como uma rede neural artificial sem peso.

Parte II: Código

- Aqui se encontra o nosso código completamente comentado, mas também pode ser verificado pelo próprio arquivo .c que foi enviado.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
```

```
//Função que embaralha um vetor
void aleatorio(int* vetor[], int n){
    int aux, *vetor2[n];
```

```

for(int i=n-1;i>=0;i--){
    aux=(clock() % (i+1));
    vetor2[i]=*(vetor+aux);
    *(vetor+aux)=*(vetor+i);
    *(vetor+i)=vetor2[i];}
for(int i=n-1;i>=0;i--){vetor[i]=vetor2[i];}}

```

//Função exponencial a^b

```

int exp(int a, int b){
    int resultado=1;
    for(int i=0;i<b;i++){
        resultado=resultado*a;}
    return resultado;}

```

//Essa função irá determinar o neurônio que ativa em cada tupla, convertendo cada possibilidade para um número diferente.
 //Cada elemento da tupla corresponde a um algarismo de um número binário, essa função converte esse número para um número decimal.

```

int posicao(int* f[], int tamanho, int b){
    int soma=0;
    for(int i=tamanho-1;i>=0;i--){
        soma=soma + **(f+b*tamanho+tamanho-1-i)*exp(2,i);}
    return soma;}

```

//Declaração da maioria das variáveis que são usadas.

```

int main(){
    int altura,comprimento,tupla,total,n,k,bleach=0,t=0,total_imagens=0,tem_bleach=2,info=2,padrao=2;
    char arquivo[100];
    FILE *file;

```

//O usuário determina os principais parametros, como dimensões da imagem, número de classes. Além de decisões como ter ou não bleaching, mostrar informações avançadas e ter um valor personalizado de tupla.

```

printf("Insira dimensões da imagem:\nComprimento: ");
scanf("%i",&comprimento);
printf("Altura: ");
scanf("%i",&altura);
printf("Insira o total de grupos: ");
scanf("%i",&k);
while(tem_bleach!=1 && tem_bleach!=0){
    printf("Deseja colocar bleaching, se sim digite 1, caso contrário digite 0: ");
    scanf("%i",&tem_bleach);}

```

```

while(info!=1 && info!=0){
printf("Voce deseja ver informacoes avancadas sobre o funcionamento da WiSARD?\nSe sim digite 1, caso contrario digite 0:
");
scanf("%i",&info);}
if(comprimento>altura){
    tupla = altura;}
else{tupla = comprimento;}
total = altura * comprimento;
while(1){
printf("Voce deseja usar um valor personalizado para a tupla?\nDigite 0 para usar o valor padrao e digite 1 para usar valor
personalizado: ");
scanf("%i",&padrao);
if(padrao==1 || padrao==0){
    break;}
printf("Valor invalido, tente novamente\n");}
if(padrao==1){
    while(1){
        printf("Insira o valor da tupla: ");
        scanf("%i",&tupla);
        if(total%tupla==0){break;}
        printf("Erro, a tupla deve ser um divisor de %i\n",total);}}

```

```

//Declaração de variáveis que dependiam das definições do usuários, como a criação dos discriminantes e das classes.
int retina[altura][comprimento],imagem_mental[altura][comprimento];
//A matriz grupo tem 3 dimensões, a primeira representa qual dos discriminantes ele representa, a segunda é o neurônio e o
terceiro são os endereços.
int grupo[k][total/tupla][exp(2,tupla)];
//A matriz classe possui duas apenas, só o neurônio e os endereços. Ela servirá para saber qual endereço do neurônio foi
ativado.
int classe[total/tupla][exp(2,tupla)];

```

```

//É mostrado para o usuário o que ele escolheu, e algumas informações importantes caso ele tenha decidido vê-las.
printf("A imagem tem dimensoes %i por %i\n",comprimento,altura);
if(info==1){
printf("O tamanho da tupla: %i\nO número de neuronios: %i\nO tamanho dos neuronios é %i (2^%i)\nO número de grupos
é %i\n",tupla,(total/tupla),exp(2,tupla),tupla,k);}
if(tem_bleach){
    printf("Voce escolheu: Colocar bleaching\n");}
else{printf("Você escolheu: Não colocar bleaching\n");}

```

```

//As coordenadas da imagem são organizadas em um vetor, que será embaralhado.

```

```
int* vetor[total];
```

```
for(int i=0;i<total;i++){  
vetor[i]=&retina[i/comprimento][i%comprimento];}
```

```
aleatorio(vetor, total);
```

```
//Todos os endereços de todos os grupos começam com valor igual a 0, assim como os endereços da matriz classe.
```

```
for(int i=0;i<exp(2,tupla);i++){  
for(int j=0;j<(total/tupla);j++){  
for(int l=0;l<k;l++){  
grupo[l][j][i]=0;}}
```

```
for(int i=0;i<total/tupla;i++){  
for(int j=0;j<exp(2,tupla);j++){  
classe[i][j]=0;}}
```

```
//Começa a fase de calibração. O processo é o mesmo para cada grupo.
```

```
for(int l=0;l<k;l++){printf("Você esta calibrando o grupo %i\n",l+1);  
scanf("%i",&n);
```

```
//Cada vez que for analisado uma imagem, os neuronios e a imagem mental será atualizada. Inicialmente a imagem mental é uma matriz vazia. Será mostrado a leitura da retina e os endereços ativados (este último apeans caso as informações avançadas foram ativadas).
```

```
for(int i = 0; i < altura; i++){  
for(int j = 0; j < comprimento; j++){imagem_mental[i][j]=0;}}
```

```
for(int a=0;a<n;a++){  
printf("Informe qual imagem vai ser avaliada: \n");  
scanf("%s", arquivo);  
total_imagens++;  
file = fopen(arquivo, "r");
```

```
for(int i = 0; i < altura; i++){  
for(int j = 0; j < comprimento; j++) {  
fscanf(file, "%i", &retina[i][j]);}}
```

```
for(int i = 0; i < altura; i++){  
for(int j = 0; j < comprimento; j++) {  
if(retina[i][j]){imagem_mental[i][j]++;}}
```

```
printf("Leitura da retina:\n");
```

```
for(int i = 0; i < altura; i++){  
    for(int j = 0; j < comprimento; j++) {  
        printf("%i ", retina[i][j]);  
    }  
    printf("\n");
```

```
for(int i=0;i<total/tupla;i++){  
    grupo[l][i][posicao(vetor,tupla,i)]++;}  
if(info==1){  
    for(int i=0;i<total/tupla;i++){  
        printf("%i ",posicao(vetor,tupla,i));  
    }  
    printf("\n");  
    for(int i=0;i<total/tupla;i++){  
        for(int j=0;j<exp(2,tupla);j++){  
            printf("%i",grupo[l][i][j]);  
        }  
    }  
}
```

//Quando o grupo terminou de ser calibrado, a imagem mental é mostrada para o usuário.

```
printf("A imagem mental do grupo %i:\n",l+1);
```

```
for(int i = 0; i < altura; i++){  
    for(int j = 0; j < comprimento; j++) {  
        printf("%i ", imagem_mental[i][j]);  
    }  
    printf("\n");}
```

```
printf("Fim da fase de calibracao\n\nInsira imagem para ser classificada\n");
```

//O vetor soma irá ver quantos neuronios serão ativados por classe. O maior e certo serão variáveis auxiliares, que guardarão respectivamente, a maior soma, e o indice do grupo que possui a maior soma.

```
int soma[k];  
for(int i=0;i<k;i++){soma[i]=0;}  
int certo=0,maior=soma[0];
```

//O processo será repetido para quantas imagens o usuário determinou. Esse processo é bem parecido com a fase de treinamento.

```
printf("Escolha o numero de imagens para serem classificadas\n");  
scanf("%i",&n);
```

```
for(int p=0;p<n;p++)  
{t=1;
```

```
printf("Insira a imagem n%i: ",p+1);
```

```
scanf("%s", arquivo);
```

```
file = fopen(arquivo, "r");
```

```
for(int i = 0; i < altura; i++){  
    for(int j = 0; j < comprimento; j++) {  
        fscanf(file, "%i", &retina[i][j]);}}
```

```
printf("Leitura da retina:\n");
```

```
for(int i = 0; i < altura; i++){  
    for(int j = 0; j < comprimento; j++) {  
        printf("%i ", retina[i][j]);  
        printf("\n");}
```

```
if(info==1){  
    for(int i=0;i<total/tupla;i++){  
        printf("%i ",posicao(vetor,tupla,i));  
        printf("\n");}  
    for(int i=0;i<k;i++){soma[i]=0;}
```

//Se o endereço que foi ativado for maior que o valor do bleaching, a soma aumentará em 1. Caso não tenha bleaching, a palavra bleaching não será mencionada para o usuário, e o valor de bleaching será 0.

```
for(bleach=0;bleach<total_imagens;bleach++){  
    for(int i=0;i<k;i++){soma[i]=0;}  
    for(int l=0;l<k;l++){  
        for(int i=0;i<total/tupla;i++){  
            if(grupo[l][i][posicao(vetor,tupla,i)]>bleach){soma[l]++;}}
```

```
maior=soma[0];
```

```
certo=0;
```

```
t=0;
```

```
if(tem_bleach){printf("Bleaching = %i\n",bleach);}  
for(int i=0;i<k;i++){  
    printf("Soma da Classe %i = %i\n",i+1,soma[i]);}
```

```
for(int i=1;i<k;i++){  
    if(soma[i]>maior){  
        maior=soma[i];
```

```

    certo=i;}}
for(int i=0;i<k;i++){
    if(soma[i]==maior){
        t++;}}
//Caso não haja empate, a classe escolhida será anunciada.
if(t==1){
    printf("A imagem pertence ao grupo %i\n",certo+1);
    break;}
//Caso dois ou mais valores estejam empatados, será informado ao usuário quais são esses grupos.
printf("Tem um empate entre os grupos:\n");
for(int i=0;i<k;i++){
    if(soma[i]==maior){
        printf("Grupo %i\n",i+1);}}
//Se houver bleaching, o seu valor será aumentado em 1 caso haja empate. Se o bleaching for incapaz de desempatar, isso
será anunciado. Se não houver bleaching, não há desempate e classificação da imagem acaba e começa a da imagem
seguinte.
if(maior==0){
    printf("O bleaching eh incapaz de determinar o grupo\n");
    break;}
if(tem_bleach==0){
    break;}}
return 0;}

```

Parte III: Testes, Limitações e Resultados

- **Testes do Código:** A parte de testagem foi algo crucial tanto quanto a própria criação bruta do código em si. Através dela o grupo conseguiu notar alguns erros e funcionalidades, desde os casos mais leves ou os mais complicados de se analisar. Dentre alguns convenhamos ressaltar os principais, sendo eles: a configuração dos endereços nos discriminadores, o sistema em geral de comunicação retina-tuplas(análise da retina e embaralhamento das tuplas) e a classificação de algumas letras mais específicas como um c minúsculo dentro classes letra E e letra C com exemplos maiúsculos em sua totalidade, os quais foram gastos bastante tempo para analisarmos os possíveis erros.

- **Resultados:** O resultado final do nosso programa em relação ao grupo foi considerado muito positivo. Um código que simula a WiSARD original com e sem bleaching dependendo das escolhas do usuário ao rodar o programa. Além disso, as fases de classificação e treinamento que em sua maioria funcionam bem, a formação de imagens mentais ao longo do programa e um sistema flexível para a escolha moderada do tamanho da retina. O único ponto negativo que conseguimos ver por enquanto é a necessidade do treinamento e classificação simultâneas a cada vez que o programa é iniciado.

Parte IV: Exemplos e demonstração

1) A primeira parte do código compila da seguinte maneira: São exigidas as informações do tamanho da retina(comprimento e altura), quantas classes serão treinadas(total de grupos), se deseja ou não colocar bleaching e se deseja definir um valor personalizado para as tuplas. Logo em seguida são informadas as configurações selecionadas para a WiSARD.

```
Insira dimensoes da imagem:
Comprimento: 6
Altura: 5
Insira o total de grupos: 2
Deseja colocar bleaching, se sim digite 1, caso contrario digite 0: 0
Voce deseja ver informacoes avancadas sobre o funcionamento da WiSARD?
Se sim digite 1, caso contrario digite 0: 1
Voce deseja usar um valor personalizado para a tupla?
Digite 0 para usar o valor padrao e digite 1 para usar valor personalizado: 0
A imagem tem dimensoes 6 por 5
O tamanho da tupla: 5
O numero de neuronios: 6
O tamanho dos neuronios e 32 (2^5)
O numero de grupos e 2
Voce escolheu: Nao colocar bleaching
```

2) A segunda parte do código é a de treinamento(calibração), e nela são exigidas quantas imagens serão avaliadas de cada classe e quais serão os arquivos a serem analisados, após o arquivo ser selecionado, é informada a leitura da retina e como ficou o discriminador após cada treinamento, e no final, o resultado da imagem mental de determinada classe.

```
Voce esta calibrando o grupo 1
Insira o numero de imagens que voce ira inserir
2
Informe qual imagem vai ser avaliada:
letraC.txt
Leitura da retina:
1 1 1 1 1 1
1 0 0 0 0 0
1 0 0 0 0 0
1 0 0 0 0 0
1 1 1 1 1 1
15 0 18 20 27 13
000000000000100000000000000000
100000000000000000000000000000
00000000000000000000100000000000
00000000000000000000100000000000
00000000000000000000100000000000
000000000000000000000000000010000
00000000000010000000000000000000
Informe qual imagem vai ser avaliada:
letraC1.txt
Leitura da retina:
1 1 1 1 0 0
1 0 0 0 0 0
1 0 0 0 0 0
1 0 0 0 0 0
1 1 1 1 0 0
11 0 16 4 26 13
000000000010001000000000000000
200000000000000000000000000000
000000000000001000000000000000
```

```
1 1 1 1 0 0
11 0 16 4 26 13
00000000000010001000000000000000
20000000000000000000000000000000
00000000000000001010000000000000
00001000000000000000100000000000
0000000000000000000000000000110000
00000000000020000000000000000000
A imagem mental do grupo 1:
2 2 2 2 1 1
2 0 0 0 0 0
2 0 0 0 0 0
2 0 0 0 0 0
2 0 0 0 0 0
2 2 2 2 1 1
```


3) A terceira parte do código é a de classificação, e nela podemos selecionar quantas imagens vão ser classificadas e logo em seguida informar qual arquivo vai ser usado para essa avaliação, pode ser feita com ou sem bleaching(definido no começo da compilação), que vai servir como desempate nessa classificação.

```
Fim da fase de calibracao
Insira imagem para ser classificada
Escolha o numero de imagens para serem classificadas
1
Insira a imagem n1: E3
Leitura da retina:
1 1 1 1
1 0 0 0
1 1 1 0
1 0 0 0
1 1 1 1
11 10 15 2 13
Soma 1 = 3
Soma 2 = 4
A imagem pertence ao grupo 2
```

```
Fim da fase de calibracao
Insira imagem para ser classificada
Escolha o numero de imagens para serem classificadas
1
Insira a imagem n1: C4
Leitura da retina:
0 1 1 1
1 0 0 0
1 0 0 0
1 0 0 0
0 1 1 1
2 14 11 1 1
Bleaching = 0
Soma 1 = 3
Soma 2 = 2
A imagem pertence ao grupo 1
```