

Relatório 3º Trabalho

Curso: Engenharia de Computação

Disciplina: Inteligência Computacional

Universidade Federal do Ceará – UFC/Sobral

Aluno: Hugo Silveira Sousa, 378998

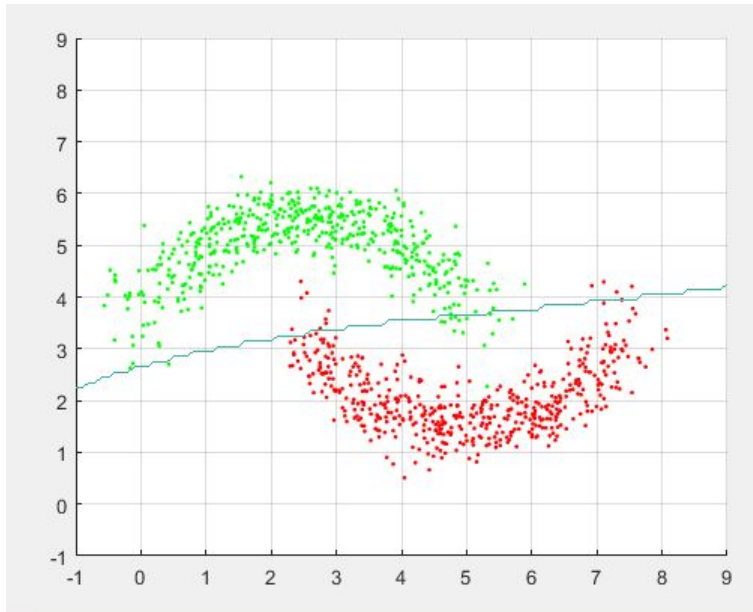
Questão 1 -

Para essa questão execute o arquivo Quest1_Trab3.m

O algoritmo segue esse esquema:

- Primeiro é carregado a base de dados twomoons.dat;
- É Dividido as duas classes e plotado os dados;
- Depois as classes que eram 1 e -1 ficam [1 0] e [0 1];
- É feito a inicialização aleatória dos pesos dos neurônios ocultos;
- É feito acúmulo das saídas dos neurônios ocultos;
- É calculado os pesos dos neurônios de saída;
- E para o teste é criado uma base que são vários pontos (malha);
- É feito uma malha das classes de cada região;
- E por último usando a função `contour()`, é plotado um contorno de acordo com a mudança das classes na malha das classes.

Tendo como resultado:



Questão 2 -

Para essa questão execute o arquivo Quest2_Trab3.m

O algoritmo segue esse esquema:

- Primeiro é carregado a base de dados iris_log.dat;
- Entra em um looping;
- Permuta as linhas da base;

- Divide-se a base em treino e teste, 80%(120) e 20%(30) para cada respectivamente;
- É usado a função newrb que implementa uma rede neural de base radial, os parâmetros são: o primeiro é a base de treino, o segundo são as classes da base e o terceiro é a meta de erro quadrático médio e retorna a rede neural;
- Usa a função sim que vai simular os valores da base de teste na rede neural e retornar suas classes calculadas;
- Arredonda-se os valores de classe encontrados para facilitar na hora da comparação, onde para cada coluna o maior valor vira 1, e os outros se tornam 0, ficando mais fácil de identificar as classes de cada elemento;
- É comparado os valores estimados das classes pela rede neural com os valores corretos das classes para a base de teste, e é guardado o número de acertos;
- Uma verificação é feita dos valores máximo e mínimo de acertos, e também vai acumulando os valores de acertos para depois fazer a média;
- E volta para o início do looping, fazendo isso 50 vezes;
- Por fim, é calculado a média de acertos e com esse valor calcula-se a taxa de acertos do algoritmo.

Deixando a semente da função randperm (usada na permutação) fixa em 1, usando a função rng(1); obtém-se os seguintes valores:

Média de acertos = 27,2400
 Taxa média de acertos = 90,80 %
 Máximo de acertos = 30
 Mínimo de acertos = 24

Questão 3 -

Para essa questão execute o arquivo Quest3_Trab3.m. Os arquivos func_ag.m e func_ag_invertido.m devem estar na mesma pasta de Quest3_Trab3.m.

O algoritmo segue esse esquema:

- Primeiro é criado uma matriz de 100 linhas por 20 colunas preenchida aleatoriamente com 0 e 1, que representam a população;
- Entra no looping das gerações;
- Separa-se a parte do X e do Y dos indivíduos, sendo os 10 primeiros bits o X, e os outros 10 bits o Y;
- Esses números são convertidos em decimal de 0 a 1023;
- Depois são convertidos em decimal no intervalo de 0 a 20 como a questão pede, usando a equação: $\text{Número} = \text{limite_inferior} + \text{correspondente_decimal} * ((\text{limite_superior} - \text{limite_inferior}) / ((2^{\text{numero_de_bits}}) - 1))$;
- Logo após, esses números convertidos são aplicados na função que a questão quer maximizar;
- As notas são os valores da função nesses pontos, mais 1, pois na função existe a possibilidade da resposta ser 0, Aquele indivíduo com nota 0 não

poderia participar da roleta de escolha para o crossover, somando 1, garante que todos irão participar;

- Seleciona-se através de um sorteio os indivíduos que serão selecionados para crossover.
- No crossover cada 2 pais geram 2 filhos, sendo que os 10 primeiros bits do primeiro pai mais os 10 últimos bits do segundo pai formam um filho, e o restante forma o segundo filho;
- Na mutação, o algoritmo passa por todos os bits de todos os filhos e faz um sorteio entre 1 e 1000, se o valor sorteado for menor que 5, o algoritmo inverte a posição atual;
- Depois todos os filhos substituem os pais;
- Volta para o início do looping das gerações;
- Ao sair do looping, o melhor indivíduo da última geração é selecionado como o indivíduo procurado que maximiza a função utilizada.

Deixando a semente da função rand fixa em 1, usando a função rng(1); obtém-se os seguintes valores:

Com 5 gerações:

X = 17.6735, Y = 17.7713 (Mutações com valores de 1 à 5, do sorteio de 1 à 1000)

X = 17.4585, Y = 18.7683 (Mutações com valores de 1 à 50, do sorteio de 1 à 1000)

Com 20 gerações:

X = 18.0645, Y = 18.0841 (Mutações com valores de 1 à 5, do sorteio de 1 à 1000)

X = 18.1232, Y = 18.0841 (Mutações com valores de 1 à 50, do sorteio de 1 à 1000)

Com 50 gerações:

X = 18.1036, Y = 18.0841 (Mutações com valores de 1 à 5, do sorteio de 1 à 1000)

X = 18.1232, Y = 18.0254 (Mutações com valores de 1 à 50, do sorteio de 1 à 1000)