

TREINAMENTO DE REDES NEURAIS POR *Backpropagation*

Mário Sérgio Cavalcante, UFRN

30 de maio de 2018

A tarefa que equivale a parte da nota da terceira unidade da disciplina de Controle inteligente consiste na implementação de um algoritmo do *backpropagation* para uma rede neural de múltiplas camadas. De forma que o aluno deve implementar uma função no **Matlab** para aproximar as funções benchmark apresentadas nesse roteiro.

Esta função deve retornar principalmente os pesos e os bias de cada neurônio. Caso o aluno achar necessário, a função pode retornar qualquer informação (um arquivo por exemplo) que torne possível a reprodução dos testes realizados pelo aluno.

Serão 5 funções que devem ser aproximadas e estão descritas nas seguintes seções do roteiro.

Função 1: Função Chapéu Mexicano

Aproxime a seguinte função:

$$f(x) = \frac{\sin(x)}{x}$$

Que apresenta o comportamento apresentado a seguir:

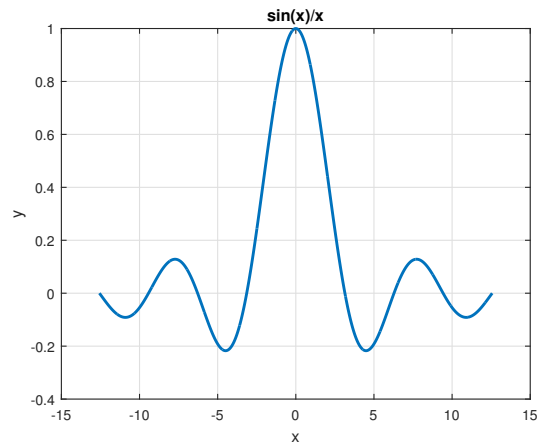


Figura 1: Função chapéu mexicano

Com os parâmetros variando de:

$$-4\pi \leq x \leq 4\pi$$

```
1 figure
2 x = -4*pi:0.001:4*pi;
3 for i = 1:length(x)
4     y(i) = sin(x(i))/x(i);
5 end
```

Função 2: Função de duas variáveis

Aproxime a seguinte função:

$$f(x, y) = 12x \cdot y^2 - 8x^3$$

Que apresenta o comportamento apresentado a seguir:

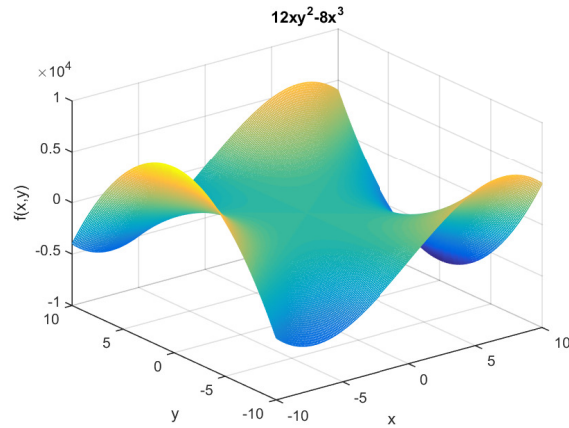


Figura 2: Função de duas variáveis

Com os parâmetros variando de:

$$-10 \leq x \leq 10$$

$$-10 \leq y \leq 10$$

Código Matlab para gerar a função:

```
1 figure
2 [x,y] = meshgrid(-10:0.1:10);
3 z = 12*x.*y.^2-8*x.^3;
4 mesh(x,y,z)
5 title('12xy^2-8x^3')
```

Função 3: Combinação de Seno e Cosseno

Aproxime a seguinte função:

$$f(x) = \frac{y \cdot \cos(x) + x \cdot \sin(y)}{x \cdot y}$$

Que apresenta o seguinte comportamento:

$$(x \cdot \cos(y) + y \cdot \sin(x)) / (x \cdot y)$$



Figura 3: Combinação de Seno e Cosseno

Utilizando o intervalo:

$$0,1 \leq x \leq 6 * \pi$$

$$0,1 \leq y \leq 6 * \pi$$

Código Matlab para gerar a função:

```
1 figure
2 [x,y] = meshgrid( .1 : .1:6 * pi );
3 z = (y.*cos(x) + x.*sin(y))/(x.*y);
4 mesh(x,y,z)
```

Função 4: Função de Ackley

Aproxime a seguinte função:

$$f(x, y) = -20 \cdot \exp \left[-0.2 \cdot \sqrt{0.5 \cdot (x^2 + y^2)} \right] - \exp [0.5 \cdot (\cos(2\pi)x + \cos(2\pi)y)] + e + 20$$

Que apresenta o seguinte comportamento:

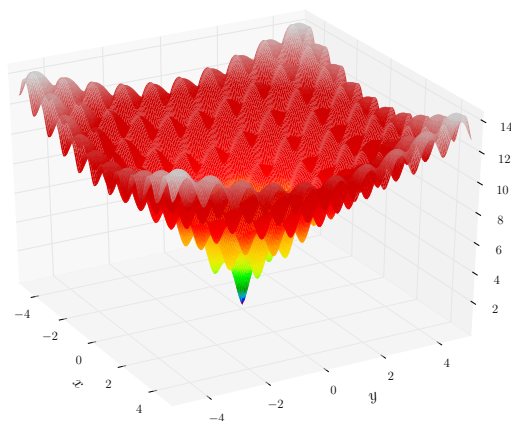


Figura 4: Função de Ackley

Utilizando o intervalo:

$$-10 \leq x \leq 10$$

$$-10 \leq y \leq 10$$

Código Matlab para gerar a função:

```
1 figure
2 clear x y z
3 [x,y] = meshgrid(-10:0.1:10);
4 z = -20*exp(-0.2*sqrt(0.5*x.^2+y.^2))-exp(0.5*cos(2*x.*pi)+cos(2*y.*
    pi)) + exp(1) + 20;
5 mesh(x,y,z)
```

Função 5: Função Eggholder

Aproxime a seguinte função:

$$f(x, y) = -(y + 47) \cdot \sin\left(\sqrt{\left|\frac{x}{2} + (y + 47)\right|}\right) - x \cdot \sqrt{|x - (y + 47)|}$$

Que apresenta o seguinte comportamento:

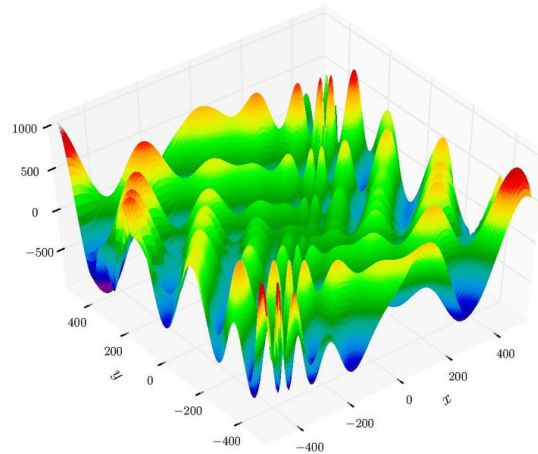


Figura 5: Função Eggholder

Utilizando o intervalo:

$$-10 \leq x \leq 10$$

$$-10 \leq y \leq 10$$

Relatório

Um relatório deverá ser entregue incluindo imagens relativas a treinamento da MLP, como:

- Erro Quadrático por época;
- Erro de comparação;
- Gráfico de comparação;
- Saída da rede neural;
- Saída da função matemática;

Bem como incluir breves explicações sobre o *backpropagation* implementado, a quantidade de testes realizados, a forma em que esses testes foram realizados e as principais características que diferenciam esses resultados.

O relatório deve ser entregue via Sigaa, até o dia 10/06/2018 às 23:59, juntamente com o arquivo **.m** que contém o *backpropagation* implementado.