



Redes Neurais - Relatório

Classificação de flores

Diego Brunetto da Silva
João Pedro Borges Baeta
Pedro Henrique Levy Fermino Ferreira

Professor
Alexandre Garcia de Oliveira

Sumário

1	O conceito de Redes Neurais	2
2	Peso e Bias	2
3	Classificação de flores	2
4	As flores a serem classificadas	2
5	Função do Erro	3
6	Sigmóide	4
7	Hiperparâmetros	4
8	Gradientes	5
9	Resultados	6
10	Links úteis	9

1 O conceito de Redes Neurais

Uma rede neural é um sistema de computador com nós interconectados que funcionam como os neurônios do cérebro humano. Usando algoritmos, eles podem reconhecer padrões ocultos e correlações em dados brutos, agrupá-los e categorizá-los, e aprender e melhorar continuamente com o tempo. Redes neurais artificiais (ANNs) consistem em uma camada de nós com uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada nó ou neurônio artificial é conectado a outro e possui peso e limite apropriados. As redes neurais dependem de dados de treinamento para aprender e melhorar sua precisão ao longo do tempo.

2 Peso e Bias

Nas redes neurais, entradas parciais são dadas aos neurônios artificiais. Cada entrada é ponderada. Os pesos representam o desempenho de uma entrada específica. Quanto maior o peso de uma entrada, maior sua influência na rede neural. O viés, por outro lado, é como uma interceptação adicionada a uma equação linear. É um parâmetro adicional em uma rede neural que ajusta a saída do neurônio, bem como a soma ponderada de suas entradas. Simplificando, Bias é uma entrada com valor "1" associado ao peso "b" de cada neurônio. Sua função é aumentar ou diminuir a entrada líquida para transladar a função de ativação no eixo.

3 Classificação de flores

Escolhemos realizar a classificação de duas flores (rosas e girassóis) por meio da implementação de um algoritmo de redes neurais, utilizando padrões naturais de ambas as flores para prever qual é a flor apenas com a inserção desses parâmetros.

4 As flores a serem classificadas

Como mencionado anteriormente, as duas flores escolhidas para realização das classificações foram as rosas e os girassóis. Adotamos os dois principais padrões naturais destas flores, que são:

Rosas

- Tamanho - Variam entre 5 e 10cm de diâmetro;
- Cor - A característica mais predominante de uma rosa, adotamos o número 0 para essa variável;
- Rótulo (Label) - Adotamos o número 0 para rosa, para identificação da rede neural.

Girassóis

- Tamanho - Variam entre 20 e 30cm de diâmetro;
- Cor - Os girassóis não possuem como característica mais importante sua cor, portanto adotamos o número 1 para essa variável;

- Rótulo (Label) - Adotamos o número 1 para girassol, para identificação da rede neural.



Figura 1: Rosa e girassol

5 Função do Erro

```
for epoch in range(epocas):  
    erro_total = 0  
    for dados in padroes_flor:  
        cor, diametro, rotulo = dados  
        predicao = classifica_flor(cor, diametro, pesos) # Essa função calcula a saída a ser prevista #  
        erro = rotulo - predicao # Calculando o erro total #  
        erro_total += erro
```

Figura 2: Função Erro

No código a função que calcula o erro foi inserida dentro do loop que treina a rede neural por meio da diferença entre o rótulo (label) esperado e a predição, sendo armazenado na variável `erro_total` para cada padrão de flor encontrado no loop.

```
media_erro = erro_total / len(padroes_flor)  
valores_erro.append(media_erro)
```

Figura 3: Média Função Erro

Ao final de cada época, o loop divide o erro total pelo número de padrões de flor, assim obtendo o erro médio da época, atualizando os pesos da rede neural para refinar o treinamento.

6 Sigmóide

```
def sigmoide(x):  
    return 1 / (1 + math.exp(-x))  
  
def sigmoide_derivada(x):  
    return sigmoide(x) * (1 - sigmoide(x))
```

Figura 4: Função de ativação sigmóide e derivada da função sigmoide

7 Hiperparâmetros

```
taxa_aprendizado = 0.1  
epocas = 1500
```

Figura 5: Taxa de aprendizado e épocas

Taxa de aprendizado

A taxa de aprendizado (learning rate) é um hiperparâmetro que determina o ajuste dos pesos dos neurônios de um algoritmo de rede neural.

Esse hiperparâmetro deve ser ajustado de maneira correta, pois um valor ou muito alto ou muito baixo pode comprometer o desempenho e a acurácia da predição a ser realizada pelo algoritmo.

O valor 0.1 de taxa de aprendizado foi adotado por ser considerado um valor moderado e comumente utilizado em algoritmos de redes neurais. Isso significa que: a cada atualização nos pesos, a taxa de aprendizado vai ajustar cada um dos pesos em 10%.

Épocas

O hiperparâmetro épocas (epochs) é o número de vezes que a rede neural irá testar os padrões apresentados.

Foi adotado o valor de 1500 épocas no código apresentado, o que significa que os padrões apresentados serão lidos 1500 vezes, mudando os pesos conforme o erro calculado em cada amostra realizada no treinamento da rede neural.

O número ideal de épocas visa velocidade e confiabilidade, tentando minimizar os erros e atingir o valor erro mais próximo de zero.

Caso tenha interesse, procure pelos gráficos do gradiente descendente (Item 9, Resultados) com 1500, 500 e 100 épocas, e compare com o erro médio.

8 Gradientes

```
# ----- #
# Calculando as derivadas parciais com relação aos erros #
gradiente_p1 = erro * cor * sigmoide_derivada(pesos[0] * cor + pesos[1] * diametro + pesos[2])
gradiente_p2 = erro * diametro * sigmoide_derivada(pesos[0] * cor + pesos[1] * diametro + pesos[2])
gradiente_b = erro * sigmoide_derivada(pesos[0] * cor + pesos[1] * diametro + pesos[2]) # Esse é o bias #
# ----- #

# ----- #
# Atualizando os pesos após os cálculos das derivadas parciais #
pesos[0] += taxa_aprendizado * gradiente_p1
pesos[1] += taxa_aprendizado * gradiente_p2
pesos[2] += taxa_aprendizado * gradiente_b
# ----- #
```

Figura 6: Calculo dos gradientes

A função da rede neural é dada por:

$$w_1 \cdot cor + w_2 \cdot diâmetro + b \rightarrow \text{chamaremos de } weighted_sum \quad (1)$$

O output que é a classificação das flores será a aplicação do sigmoid a função *weighted_sum*, após isso será possível encontrar o vetor gradiente para calcular as derivadas parciais:

$$output = (1 + sigmoide(-(w_1 \cdot cor + w_2 \cdot diâmetro + b))) \quad (2)$$

O vetor gradiente será dado pela derivada parcial de w_1 , w_2 e b , abaixo as derivadas parciais:

$$\frac{\partial F}{\partial w_1} = (1 + sigmoide(-(w_1 \cdot cor + w_2 \cdot diametro + b))) \rightarrow cor \cdot sigmoide(-(w_1 \cdot cor + w_2 \cdot diametro + b)) \quad (3)$$

$$\frac{\partial F}{\partial w_2} = (1 + sigmoide(-(w_1 \cdot cor + w_2 \cdot diâmetro + b))) \rightarrow diâmetro \cdot sigmoide(-(w_1 \cdot cor + w_2 \cdot diâmetro + b)) \quad (4)$$

$$\frac{\partial F}{\partial b} = (1 + \exp(-(w_1 \cdot cor + w_2 \cdot diâmetro + b))) \rightarrow sigmoide(-(w_1 \cdot cor + w_2 \cdot diâmetro + b)) \quad (5)$$

$$GradF(w_1, w_2, b) = \begin{bmatrix} cor \cdot sigmoide(-(w_1 \cdot cor + w_2 \cdot diâmetro + b)), \\ diâmetro \cdot sigmoide(-(w_1 \cdot cor + w_2 \cdot diâmetro + b)), \\ sigmoide(-(w_1 \cdot cor + w_2 \cdot diâmetro + b)) \end{bmatrix} \quad (6)$$

9 Resultados

```
O resultado da classificação foi 0.9637969028619282, portanto sou um girassol
O resultado da classificação foi 0.044116801034953086, portanto sou uma rosa
O resultado da classificação foi 0.9634286591292889, portanto sou um girassol
O resultado da classificação foi 0.9638598869436764, portanto sou um girassol
O resultado da classificação foi 0.0432068029455019, portanto sou uma rosa
O resultado da classificação foi 0.9634820768045207, portanto sou um girassol
O resultado da classificação foi 0.04406161384406906, portanto sou uma rosa
O resultado da classificação foi 0.9658285120442504, portanto sou um girassol
O resultado da classificação foi 0.04528798894593715, portanto sou uma rosa
O resultado da classificação foi 0.04382618777330538, portanto sou uma rosa
```

Figura 7: Output da classificação das flores para 1500 épocas

Como mostrado na figura acima, a classificação das rosas se aproxima de 0 e a classificação dos girassóis se aproxima de 1, isso demonstra a função sigmóide sendo executada, pois o intervalo de saída da classificação se encontra no intervalo $[0,1]$, ou seja, caso a saída seja menor que 0.5, podemos considerar que a flor é uma rosa, e maior que 0.5 no caso dos girassóis.

```
O resultado da classificação foi 0.9333001064375888, portanto sou um girassol
O resultado da classificação foi 0.9320670333311035, portanto sou um girassol
O resultado da classificação foi 0.07985989001694316, portanto sou uma rosa
O resultado da classificação foi 0.9379425846238577, portanto sou um girassol
O resultado da classificação foi 0.9346594790633164, portanto sou um girassol
O resultado da classificação foi 0.07830690178854975, portanto sou uma rosa
O resultado da classificação foi 0.07990313608161732, portanto sou uma rosa
O resultado da classificação foi 0.9333446524449235, portanto sou um girassol
O resultado da classificação foi 0.07999453878269659, portanto sou uma rosa
O resultado da classificação foi 0.07937370362670067, portanto sou uma rosa
```

Figura 8: Output da classificação das flores para 500 épocas

Nesse caso, percebe-se que a precisão diminuiu devido a diminuição no número de épocas, ficando mais perto do limite da divisão de classificação.

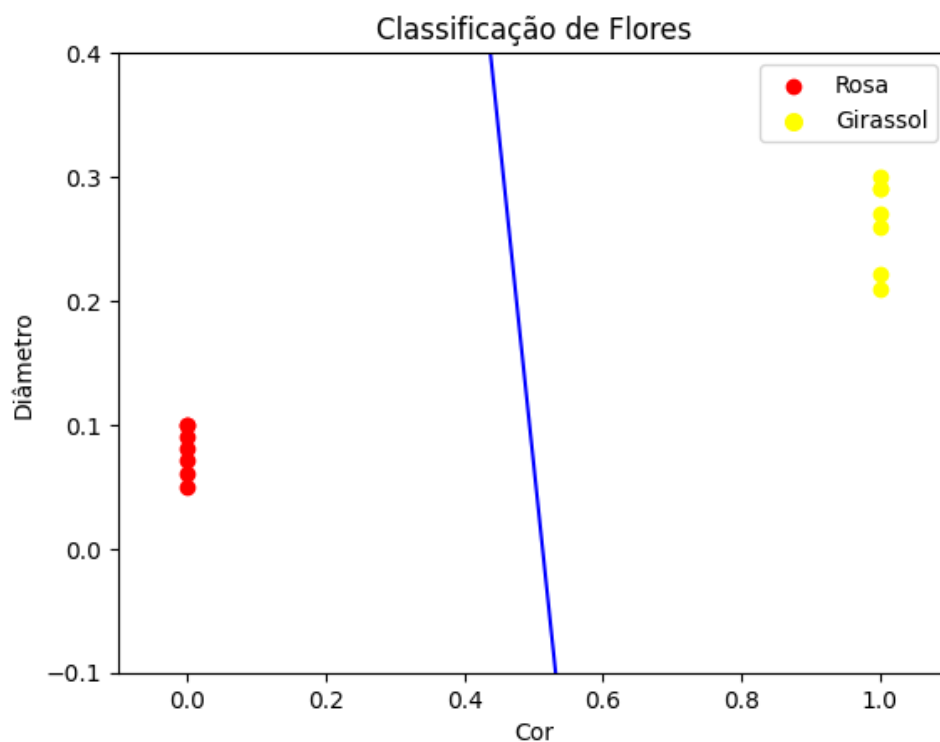


Figura 9: Gráfico da classificação de flores

Esse gráfico demonstra o limite entre a classificação da rosa e do girassol, onde o limite se estabelece por volta do ponto 0.5 no eixo x (cor), que é considerado a principal característica para realizar a classificação.

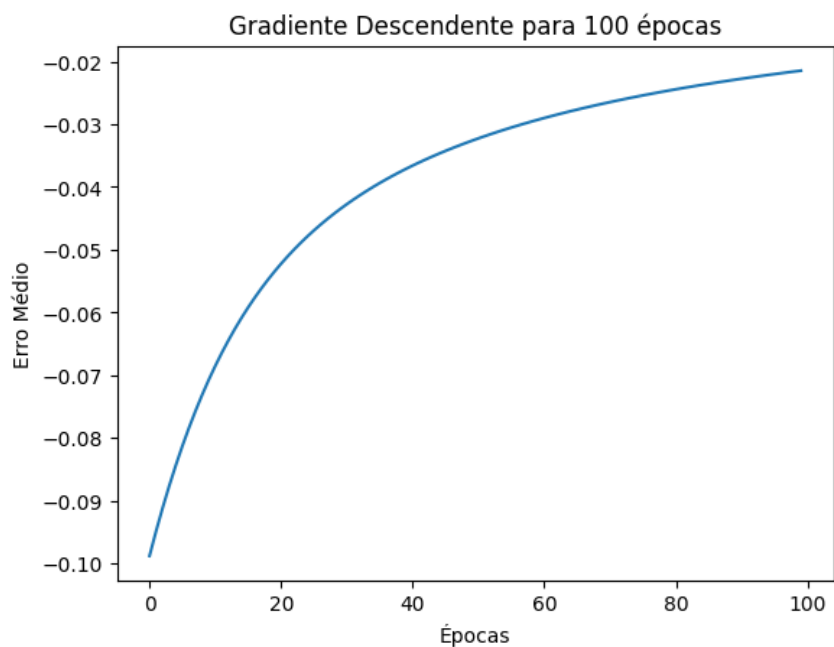


Figura 10: Gráfico do gradiente descendente para 100 épocas

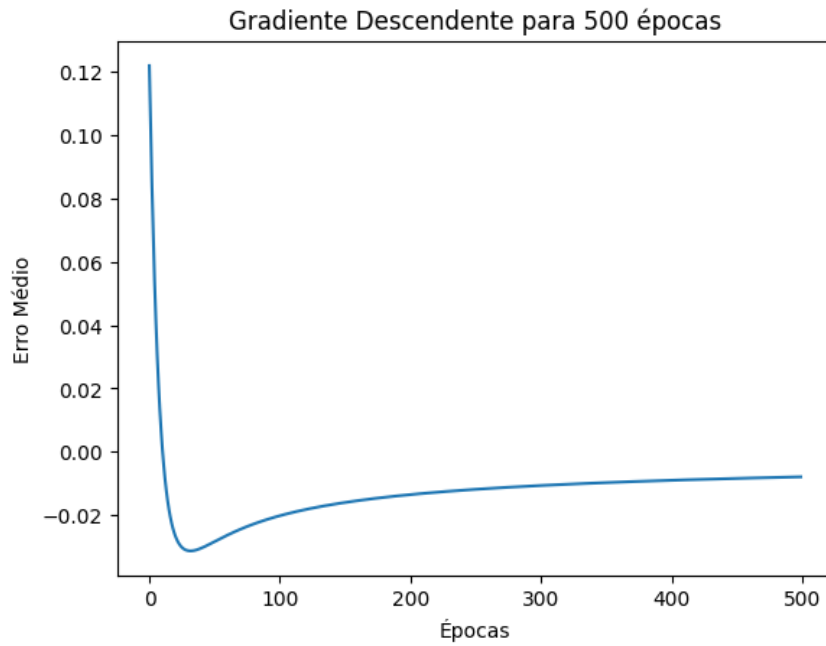


Figura 11: Gráfico do gradiente descendente para 500 épocas

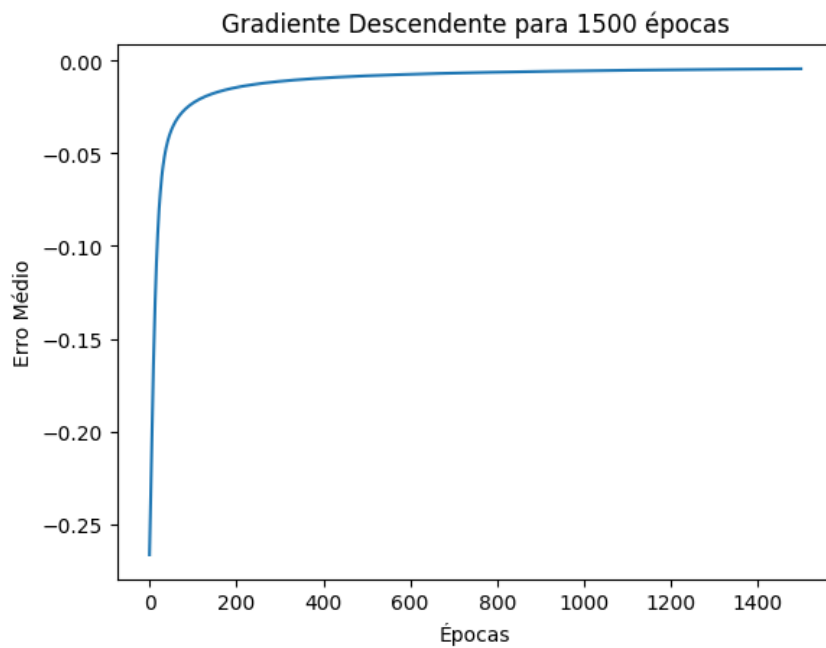


Figura 12: Gráfico do gradiente descendente para 1500 épocas

Os três gráficos demonstram o gradiente descendente para três valores de épocas diferentes (100, 500 e 1500), mostrando o erro médio ao fim da quantidade de épocas. O valor de 1500 épocas apresenta uma margem de erro muito próxima de zero, mostrando um comportamento similar a uma reta ao fim das 1500 épocas, indicando estabilidade.

10 Links úteis

GitHub com todos os arquivos - <https://github.com/pedrohlevy/redeneural>

LinkedIn Diego Brunetto da Silva - <https://www.linkedin.com/in/diego-brunetto-b892b1202/>

LinkedIn João Pedro Borges Baeta - <https://www.linkedin.com/in/joaopedrobaeta/>

LinkedIn Pedro Henrique Levy Fermino Ferreira - <https://www.linkedin.com/in/pedrohlevy/>