

Vitor Amadeu Souza

Reconhecimento de Frutas com

Rede Neural

Artificial com

PyTorch

Programado em Python

© 2024 by Cerne Tecnologia e Treinamento Ltda.

© 2024 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Outubro de 2024

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

Produção: Cerne Tecnologia e Treinamento

E-mail da Empresa: cerne@cerne-tec.com.br

Home Page: www.cerne-tec.com.br.com.br

Atendimento ao Consumidor: sac@cerne-tec.com.br

Contato com o Autor: vitor@cerne-tec.com.br



FEITO NO BRASIL

***“A vida é como andar de bicicleta. Para manter o equilíbrio, você
deve continuar se movendo.”***

Albert Einstein

Cerne Tecnologia

A Cerne Tecnologia tem uma equipe preparada para desenvolvimento de projetos eletrônicos em diversas áreas: Médica, Entretenimento, Industrial, Robótica, Científica, Automobilística, Aeronáutica, etc. Trabalhamos com tecnologia microcontrolada usando o PIC, ARM, AVR, 8051, dsPIC, PIC24, PIC32 além do Arduino, Raspberry, Beaglebone etc. Desenvolvemos o projeto desde sua concepção até a entrega do produto final, passando pelas etapas de esquema elétrico, protótipo e desenvolvimento de circuito impresso.

Desenvolvemos aplicativos para smartphones/tablets Android e iOS além de desenvolvimento de softwares a nível PC para plataforma Windows, usando ferramentas como o Visual Basic, C# e C++.

Atuamos na parte de montagem de placas, onde podemos fornecer ambos os serviços de desenvolvimento de projetos e produção ou apenas um destes.

Desenvolvemos esquemas elétricos e layout de PCI, tanto em tecnologia convencional como SMD.

Temos a flexibilidade de customizar um de nossos produtos, de modo a atender a uma necessidade específica do cliente, tornando o custo de desenvolvimento menor se comparado a construção de um projeto desde a sua fase inicial.

Desenvolvemos e fornecemos kits didáticos para diversos microcontroladores além de apostilas, livros e e-books.

Na hora de desenvolver um projeto ou equipar seu laboratório não hesite em nos contatar. Entre em contato conosco através do endereço cerne-tec.com.br para obter mais informações.



Sumário

Capítulo I – Metodologia de desenvolvimento.....	7
1. Introdução.....	7
Capítulo II – Programação em Python.....	8
1. Introdução.....	8
2. Operadores aritméticos.....	10
3. Operadores lógicos.....	12
4. Operadores de bits (bitwise operators).....	13
5. Funções de conversão.....	14
6. Comentários.....	14
7. Variáveis.....	15
8. Operadores Relacionais.....	17
9. Trabalhando com strings.....	20
10. O comando If.....	23
11. O comando while.....	25
12. O comando for.....	26
13. Usando listas.....	29
14. Conhecendo as Tuplas.....	32
15. Dicionários.....	32
16. Conjuntos.....	33
17. Criando scripts.....	34
18. Pi e número de Euler.....	37
19. Funções matemáticas.....	38
20. Obtendo a data e hora.....	38
21. Calculando o tempo para executar uma rotina.....	39
22. Emitindo som.....	40
23. Calendar.....	40
24. Números complexos.....	41

25. Funções Pré-Definidas.....	41
26. Comando type.....	47
27. Entrada de dados.....	47
28. Comando break.....	49
29. Tratamento de erro.....	50
30. Impressão formatada.....	52
31. Função bool.....	52
32. Operador in.....	53
33. Operador randômico.....	54
34. Obtendo ajuda.....	56
35. Operador de formatação.....	57
36. Criando funções de usuário.....	58
37. Números perfeitos.....	61
38. Acesso a arquivos externos.....	61
 Capítulo III – Plotando gráficos.....	 64
 Capítulo IV – Função sigmoide.....	 87
 Capítulo V – Perceptron.....	 89
1. Introdução.....	89
2. Implementando o perceptron.....	97
 Capítulo VI – RNA com PyTorch.....	 99
1. Introdução.....	99
2. Conceitos fundamentais.....	101
3. Formando o dataset.....	109
4. Treinando o modelo.....	111

Capítulo I

Metodologia de desenvolvimento

1. Introdução

A proposta deste trabalho é desenvolver a unidade fundamental das Redes Neurais Artificiais (RNA), conhecida como Perceptron, utilizando Python como ferramenta de programação. Para isso, apresentaremos um exemplo prático que ilustra a implementação desse conceito. Após essa experiência, realizaremos o treinamento de uma RNA com o PyTorch, utilizando um conjunto de imagens de frutas (dataset). Em seguida, efetuaremos predições para avaliar a eficácia do treinamento.

Ao longo do processo, abordaremos diversos exemplos que destacam tópicos essenciais para uma compreensão sólida da programação em Python. O objetivo é fornecer uma base teórica e prática que permita ao leitor entender não apenas a implementação do Perceptron, mas também os princípios subjacentes ao funcionamento das Redes Neurais Artificiais.

Capítulo II

Programação em Python

1. Introdução

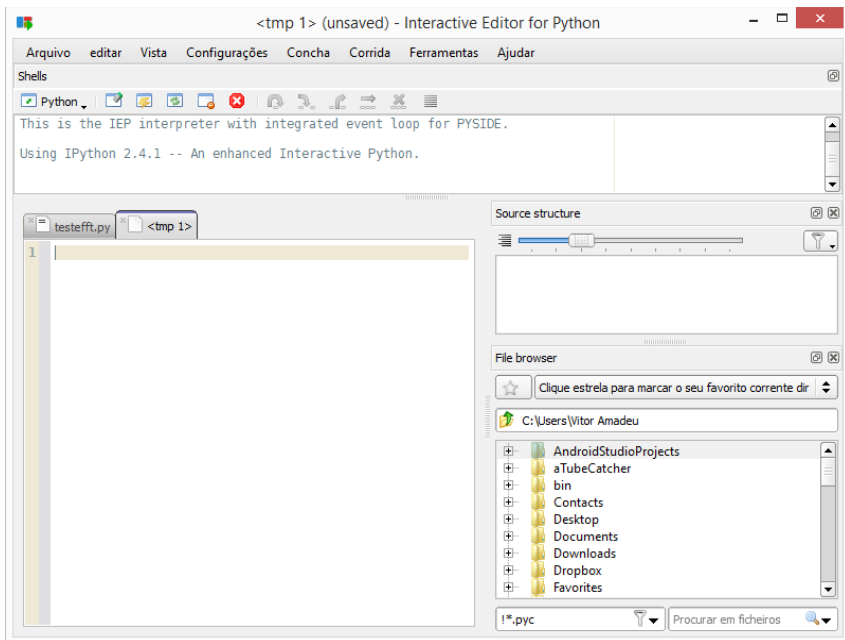
Nesta literatura a distribuição Pyzo foi utilizada, no qual a última versão pode ser baixada através do link abaixo.

<http://www.pyzo.org/downloads.html>

Baixe e instale também a última versão do Python disponível em:

<https://www.python.org/>

A vantagem desta distribuição é que ela por padrão já vem com as bibliotecas a serem utilizadas ao longo desta obra. Após a instalação inicialize o programa, a tela a seguir será apresentada.



Outra possibilidade é usar a plataforma online chamada Google Colab, no qual é possível criar *notebooks* (cadernos) para executar e testar programas.

<https://colab.research.google.com/>

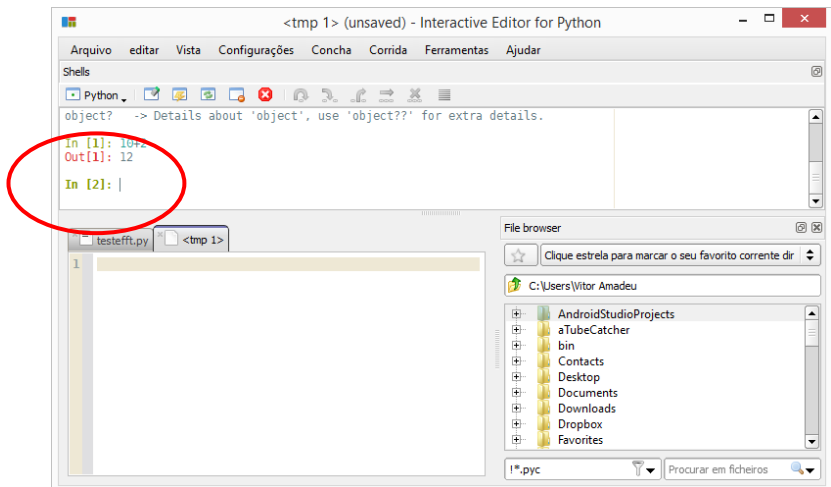
Uma ferramenta que tem sido bem usada ultimamente é o VS Code da Microsoft, que pode ser baixado pelo link abaixo. Nele, é possível instalar extensões com a do Python e outras ferramentas.

<https://code.visualstudio.com/>

O Python é um software interpretado, ou seja, cada comando digitado no ambiente é logo executado após você pressionar o enter do teclado. Nos próximos tópicos, exercitaremos diversos exemplos no Python, de forma a entender como o mesmo funciona.

2. Operadores aritméticos

Podemos usar o Python como uma calculadora, bastando neste caso digitar diretamente a expressão matemática no mesmo, usando neste caso o prompt. Observe abaixo:



Note que ao digitar a expressão e pressionar o enter, o comando é imediatamente processado, tendo como resultado a soma da operação. Outra forma é escrever o programa como um script, salvá-lo e executá-lo em seguida, indo no menu Corrida-> Corrida arquivo como script.

O Python possui diversos operadores aritméticos, como os citados a seguir.

Operador	Função
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da divisão de inteiro
**	Exponenciação

Observe abaixo outros exemplos no Python, baseado nos comandos apresentados anteriormente:

```
>>> 10 - 6
4
>>>
```

No exemplo acima temos uma operação de subtração sendo executada, onde a temos como resultado 4, que é a subtração de 10 – 6.

```
>>> 2 ** 3
8
>>> |
```

Acima temos um exemplo de exponenciação, onde o valor 2 elevado a 3 dá como resultado 8.

```
>>> 100 % 3
1
>>>
```

Acima a operação é diferente, pois temos a apresentação do operador resto da divisão de inteiros (%) onde 100 dividido por 3 dá 33, porém com resto 1 como verificado acima.

3. Operadores lógicos

Faz uma das operações lógicas fornecendo como resultado verdadeiro (True) ou falso (False). Acompanhe um exemplo com os três operadores lógicos.

```
>>> x=1
>>> y=0
>>> x and y
0
>>> x or y
1
```