

Universidade Estadual de Montes Claros



Segundo Trabalho Prático - Sinais e Sistemas Convolução em Python

Aluno: Tarcisio Francisco Batista Filho
Professor: Renato Dourado Maia

Montes Claros, 27 de Maio de 2015

Sumário

1	Introdução	1
2	Desenvolvimento	1
2.1	Metodologia 1:	1
2.2	Metodologia 2:	3
3	Conclusão	4
4	Bibliografia	6

1 Introdução

O objetivo do trabalho é compreender melhor o funcionamento de convoluções através de uma ferramenta matemática que possibilite a visualização e entendimento de uma forma diferente e simplificada. O metodo "rebate, desloca, multiplica, soma" foi empregado em uma primeira instancia mas após terminado não estava achei elegante o suficiente, observando o que é a convolução outro metodo parecido porem mais elegante foi encontrado.

2 Desenvolvimento

Para o experimento em questão foram utilizados duas metodologias de convolução parecidas mas distintas.

2.1 Metodologia 1:

Na metodologia 1 foi feita a convolução de forma tradicional "rebate, desloca, multiplica, soma", foram utilizados 2 estruturas de repetição 1 *while* e 2 *for*, além da criação da classe de sinal que recebe o primeiro elemento do dominio de um sinal e o vetor imagem e facilita bastante o tratamento de sinais. Basicamente o *while* de fora simplesmente percorre e marca as posições o vetor de saida, de forma que esse será percorrido apenas uma vez, os 2 *for's* são utilizados com a finalidade tradicional de percorrer todo o vetor de dentro para cada elemento de fora e se os os dominios dos elementos forem iguais é feito o produto entre eles e adicionado na posição aux que percorre todo o vetor imagem da saida. acompanhe graficos de sinais teste:

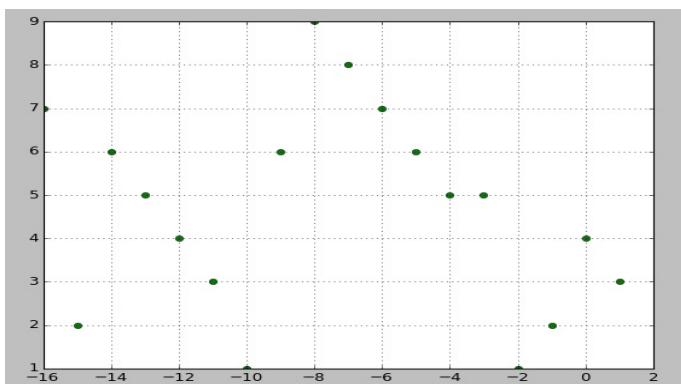


Figura 1: Sinal S

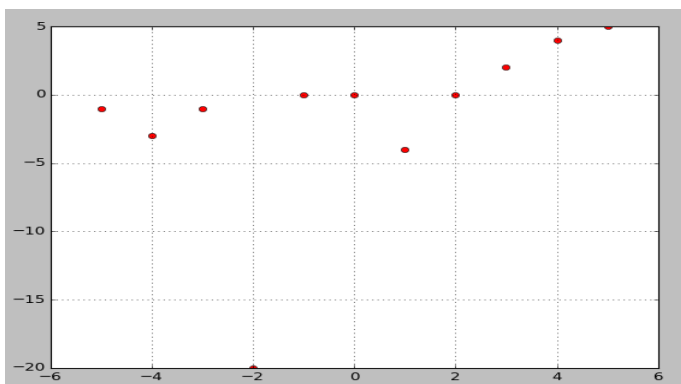


Figura 2: Sinal P

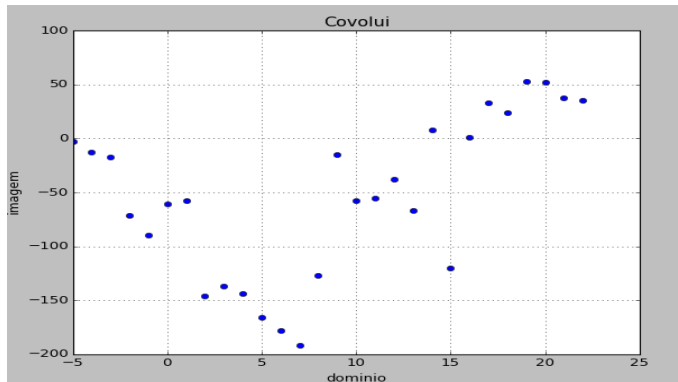


Figura 3: Convolução

Para testes com outros sinais basta editar o arquivo e modificar o vetor de imagem e o numero do primeiro elemento do dominio.

2.2 Metodologia 2:

Na metodologia 2 foi observado durante a convolução de dois sinais, através dos índices dos vetores um elemento de um vetor multiplica todos os elementos do outro vetor baseado nisso foi feita uma logica. Para todos os elementos do primeiro vetor são multiplicados pelos elementos do segundo vetor o segredo é a posição em que eles são salvos. Em seguida imagens do sinal de teste:

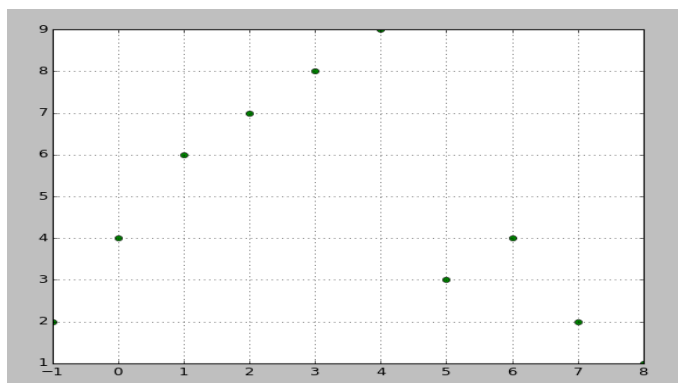


Figura 4: Sinal S

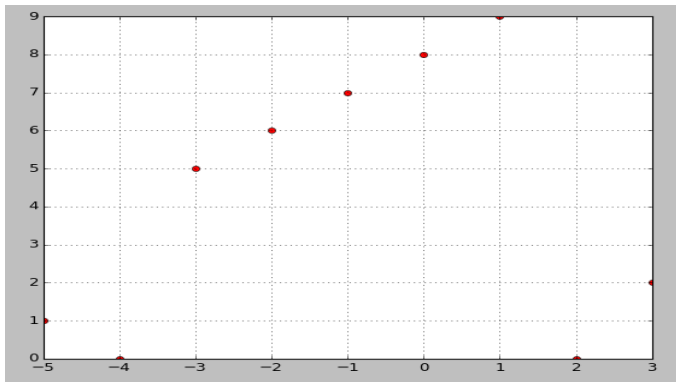


Figura 5: Sinal P

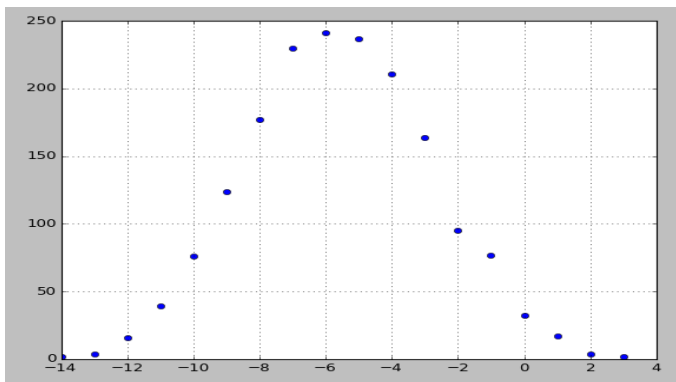


Figura 6: Convolução

3 Conclusão

Na metodologia 1 meu caro colega Murilo Camargos me apresentou algumas de suas ideias para a resolução do problema como utilizar a programação orientada a objetos, criar a classe Sinal, chamar o sinal de uma composição de domínio e imagem, métodos otimizados de rebater e deslocar; sem duvidas esses conceitos me ajudaram muito a clariar minha mente sobre o assunto. Resolvi o problema mas infelizmente o código em si ficou muito parecido com o dele, ainda assim a lógica não era muito elegante pois utilizava 1 "while" e 2 "for's", na segunda metodologia ainda foram utilizadas algumas ideias como a utilização de uma classe de sinais, nela utilizo apenas duas

estruturas de repetição e nenhuma estrutura de tomada de decisão pois todos os elementos dos dois vetores acabam se multiplicando então apenas preciso colocar as multiplicações em seus lugares devidos.

4 Bibliografia

- [1] wiki.scipy.org/Tentative_NumPy_Tutorial
- [2] stackoverflow.com/questions/3940128/how-can-i-reverse-a-list-in-python
- [3] www.openbookproject.net/books/bpp4awd/ch04.html
- [4] anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/functions.html
- [5] matplotlib.org/api/pyplot_api.html
- [6] matplotlib.org/users/pyplot_tutorial.html
- [7] pplware.sapo.pt/tutoriais/construo-de-um-relatrio-acadmico-em-latex-parte-i/
- [8] sbi.iqsc.usp.br/files/Manual-SBI_LATEX_2013-.pdf