COP 290

Small Image Processing library

Description :

This library is made for various image processing and filtering processes. It has implementation of convolution of two square matrices ,one is the input and other is the kernel. It has various activation functions like tanh and ReLU .It includes sigmoid and softmax functions to map the data to [0,1]. It has pooling functions which acts as a hidden layer in the convolutional neural network formed for image processing.

Functions and Syntax

These functions are written in the following format along with running the executable file. These strings are passed as argument strings in the main function.

convolution input_filename input_rowsize output_filename output_rowsize

It performs convolution without flipping the matrix.e.g for c11 -> a11 is multiplied with b11 ,a12 with b12 ...so on. The Stride is taken to be 1 in convolution. Answer is returned in 2d vector which is then printed by display function in column major form.

- ***convolution_withpadding padsize input_filename input_rowsize output_filename output_rowsize***

  It performs convolution by adding a pad of specified padsize around the input.The Stride is taken to be 1 in convolution. Answer is returned in 2d vector which is then printed by display function in column major form.

- ***convolution_matrixmult input_filename input_rowsize output_filename output_rowsize***

  It processes input by creating into Toeplitz matrix of dimensions as (output_dim^2 *kernel^2) and kernel is changed to a column matrix of dimensions (kernel^2 *1). Which when multiplied gives convolution matrix elements in row major form. The Stride is taken to be 1 in convolution. Answer is returned in 2d vector which is then printed by display function in column major form.

- ***convolution_withpadding_matrixmult padsize input_filename input_rowsize output_filename output_rowsize***

  Padding of input is done by required size. Then It processes input by creating into Toeplitz matrix of dimensions as (output_dim^2 *kernel^2) and kernel is changed to a column matrix of dimensions (kernel^2 *1). Which when multiplied gives convolution matrix elements in row major form. The Stride is taken to be 1 in convolution. Answer is returned in 2d vector which is then printed by display function in column major form.

- ***softmax input_filename***

  It simply applies softmax function on the data i.e exp(xi)/(sum exp(xi)) that maps data to (0,1).

- ***sigmoid input_filename***

  It simply applies sigmoid function on the data i.e 1+/(1+ exp(xi)) that maps data to (0,1).

- ***max_pooling input_filename input_rowsize***

  It takes 2*2 non overlapping matrices and takes its maximum as new value.It adds a padding if required. The Stride is taken to be 2 . Output is in form of matrix.

- ***average_pooling input_filename input_rowsize***

  It takes 2*2 non overlapping matrices and takes its average as new value.It adds a padding if required. The Stride is taken to be 2 . Output is in form of matrix.

- ***relu_activation input_filename input_rowsize***

  Simply converts input matrix data to max (0,datavalue) and returns this output.

- ***tanh_activation input_filename input_rowsize***

  Simply converts input matrix data to max (0,datavalue) and returns this output.

Error and helps

If somebody types wrong command name ,it gives list of functions and syntax to be followed.It takes input from the input_filename and output_filename of their specified sizes. If any file does not exists it prints error that this particular file does not exists. If size is not as said it prints error that the size of file is not as mentioned