DEEP LEARNING FROM SCRATCH IN C++
LIBRARY STRUCTURE

Nehil Danis

31.07.2019

My library contains two main classes which are the Network class, and abstract Computational Node Class. Computational node class is a parent class of Convolution, Maxpooling, DotProduct, AddNode, Dropout, Activation classes. Activation class is also an abstract class which is the parent of Sigmoid, Softmax, Relu, and LeakyRelu classes.

Computational Node class includes common operations for all types of computational node. Forward pass, backward pass, update, load weights and save weights can be an examples of these common operations. Network class on the other hand is for users to build a network and train, test and validate this network. User can call functions from Network class by using python binding to add different types of nodes to the network and once they are done, then they can call the train function to train the network. Before that if they already have pretrained weights for that particular network, then they can just load these weights before training. After training or during training they can also call the save weights functions to saved the trained weights for the testing or maybe transfer learning.