```
import numpy
import modules as md
import copy

def buildcnn():

    nn = md.Network([
        md.Convolution('cnn/c1-5x5x3x10',write=True),md.ReLU(),md.Pooling(),
        md.Convolution('cnn/c2-5x5x10x50',write=True),md.ReLU(),md.Pooling(),
        md.Convolution('cnn/c3-4x4x50x250',write=True),md.ReLU(),md.Pooling(),
        md.Convolution('cnn/c4-1x1x250x10',write=True),
    ])
    nn.layers[-1].W*=0

    return nn


def geterrorgrad(y,t):

    return numpy.exp(y) / (numpy.exp(y).sum(axis=1)[:,numpy.newaxis] + 1) - t

lowest = -1
highest = 1



class FirstConvolution(md.Convolution):
    def relprop(self,R):
        iself = copy.deepcopy(self); iself.B *= 0
        nself = copy.deepcopy(self); nself.B *= 0; nself.W = numpy.minimum
(0,nself.W)
        pself = copy.deepcopy(self); pself.B *= 0; pself.W = numpy.maximum
(0,pself.W)

        X,L,H = self.X,self.X*0+lowest,self.X*0+highest
        Z = iself.forward(X)-pself.forward(L)-nself.forward(H)+1e-9; S = R/Z
        R = X*iself.gradprop(S)-L*pself.gradprop(S)-H*nself.gradprop(S)

        return R
```