I have read that Ternarizing the weights of a neural network makes it easy for implementation on embedded devices. At the same time In Ternary Weight Networks, 2-bit storage requirement is needed for a unit of weight. Thus, TWNs achieve up to 16× or 32× model compression rate compared with the float (32-bit) or double (64-bit) precision counterparts. So to try this I have used the methods of DingKe .

Reference: <https://github.com/DingKe/nn_playground/tree/master/ternarynet>

But after using this method I observed that the weights were still represented in a floating point number so I devised a method which converts this floating point numbers to ternary weights(-1,0,1) without change in the accuracy.