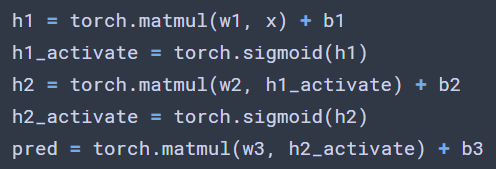
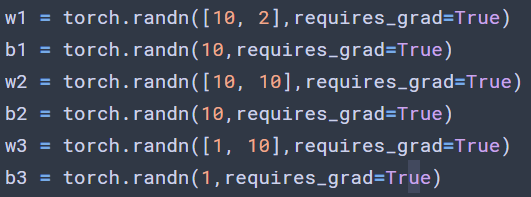
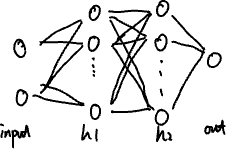
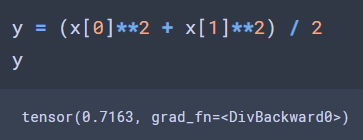
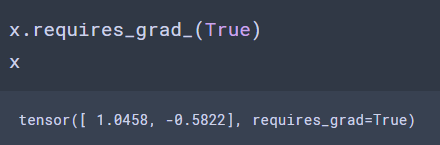
1. Implement this neural network in pytorch





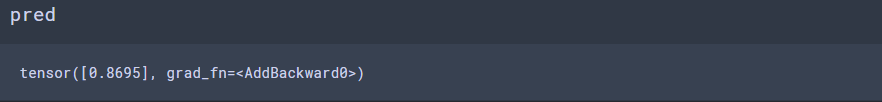
1. Generate the input date (x1,x2) \in [0,1] drawn from a uniform random distribution
2. Generate the labels y = (x1\*x1+x2\*x2)/2



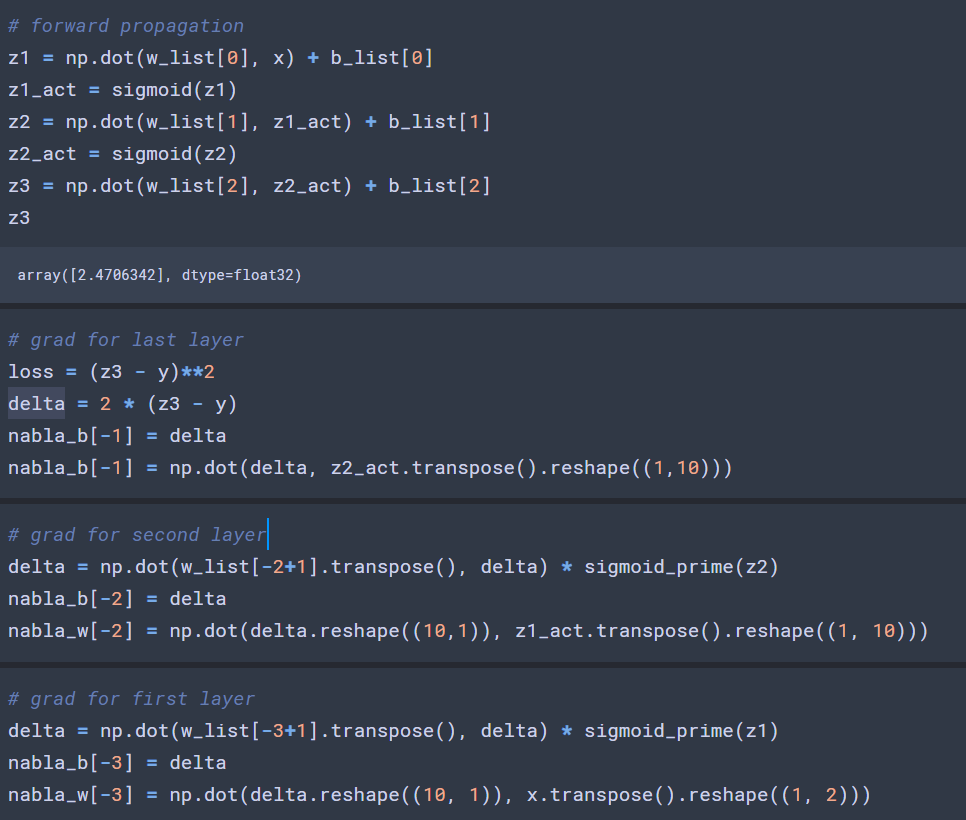
1. Implement a loss function L = (predict-y)^2



1. Use batch size of 1, that means feed data one point at a time into network and compute the loss. Do one time forward propagation with one data point.

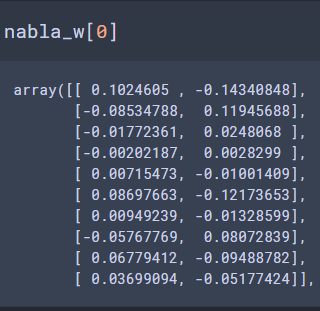
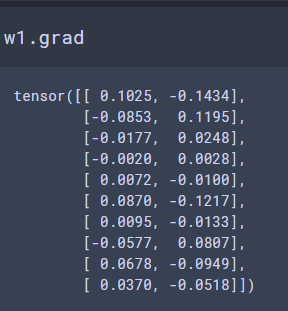


(7) Implement the forward propagation and backpropagation algorithm from scratch, without using pytorch autograd, compute the gradients using your implementation





(8) Compare the two files torch\_autograd.dat and my\_autograd.dat and show that they give the same values up to numerical precision errors



The left one is the grad for w1 in autograd, the right one is the grad I manually calculated.

From the code we can see that the two way of generate grad have nearly the same result except some deviation by machine.