Tutorial 3: Dropout and Computational Graph

1. Using the example and codes you have written for Tutorial 2, add a dropout layer to check how dropout rate affects the classification accuracy of MLPs (Multilayer Perception). What will happen if you set the dropout rate to be 1?

   Add nn.Dropout(dropoutRate) after each activation (e.g. nn.Relu)
   If the dropout rate is 1, the weights wont be updated and the model Parameters remain the same as the initialization.

2. Implement the model with Pytorch for the following computational graph in Sec. 4.7.2 of the book.

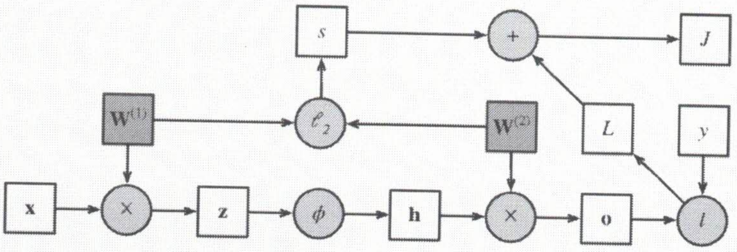


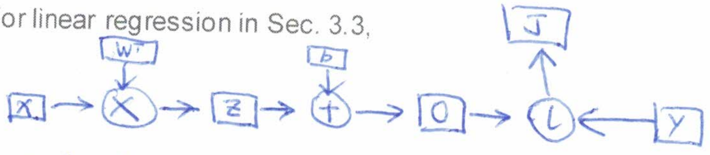
Fig. 4.7.1 Computational graph of forward propagation.

This is a single hidden layer neural network. Suppose the dimension of the input and output are *d* and *q*, and the number of neurons in the hidden layer is *h*, and the activation function ϕ is ReLU, the model can be implemented as

```
net = nn.Sequential(nn.Linear(d, h),
                    nn.ReLU(),
                    nn.Linear(h, q))
```

3. Draw the computational graphs of forward propagation for the codes in Sec. 4.3 of the book for the implementation of MLP and for the codes in Sec. 3.3 of the book for the implementation of linear regression.

For linear regression in Sec. 3.3,



x: input

×: multiplication

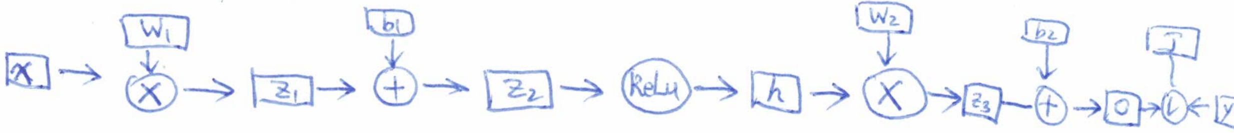
w: weight vector

o: output

l: loss function (squared error)

y: label

For MLP in Sec. 4.3,



l: loss function (cross entropy)