2020314462 인공지능융합전공 황선우

Homework #7

1. Neural network에 dropout을 적용할 때, 아래 정보를 참고하여 다음 표를 채우시오.
   - ❿ $Dropout\ rate = 0.25$
   - ❿ $z = \sum_{i=0}^{4} x_i w_i$
   - ❿ $\{x_0, x_1, x_2, x_3\} = \{1, 2, 2, 3\}$

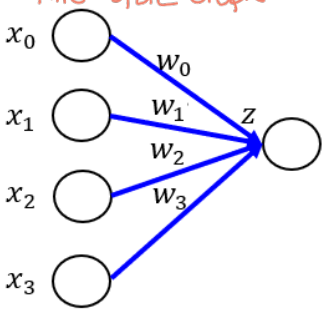
Step	w^0	w^1	w^2	w^3	z
Training	0.2	0.2	0.2	0.2	1.2
Testing	0.7	0.1	0.3	0.4	2.0250

training:

x_0	x_1	x_2	x_3	z
1	2	2	3	
0	1	1	1	(2+2+3)×0.2=1.4
1	0	1	1	(1+2+3)×0.2=1.2
1	1	0	1	(1+2+3)×0.2=1.2
1	1	1	0	(1+2+2)×0.2=1

E(z)=(1.4+1.2+1.2+1)/4=1.2

4개중에 하나는 drop



testing:

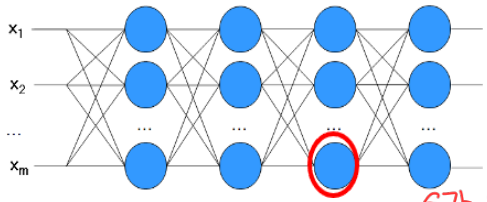
retaining rate = 0.75

(1×0.7+2×0.1+2×0.3+3×0.4)×0.75

= 2.0250

2. 아래의 Neural Network와 5개의 학습데이터를 가진 minibatch가 있다. 각 학습데이터가 주어졌을 때, 빨간색 노드의 net 값이 아래와 같다. Batch Normalization을 적용했을 때, 아래 표의 빈칸을 채우시오

* ipynb 파일참조



hidden layer shallow & deep 무조건 ReLU 쓰기 ㅋ

activation function : ReLU

Training Data	net	$\widehat{net}$ $(\epsilon = 0)$	$\widetilde{net}$ $(\gamma = 1, \beta = 1)$	y
Data 1	-1	-1.4985	-0.4985	0
Data 2	2	-0.0937	0.9063	0.9063
Data 3	5	1.3112	2.3112	2.3112
Data 4	4	0.8429	1.8429	1.8429
Data 5	1	-0.5620	0.4380	0.4380

6가 0이되는것 방지

```
#2번문제
import torch
import numpy as np
import torch.nn as nn

data = torch.FloatTensor([-1,2,5,4,1])

#표준편차, 평균 계산
m = torch.mean(data)
#s = torch.std(data)
import math
s = math.sqrt(((-1-2.2)**2 + (2-2.2)**2 + (5-2.2)**2 + (4-2.2)**2 + (1-2.2)**2)/5)
s = torch.FloatTensor([s])
m,s
```

```
(tensor(2.2000), tensor([2.1354]))
```

평균: 2.2

표준편차: 2.1354

```
sigmoid = torch.nn.Sigmoid()
relu = torch.nn.ReLU()

net1 = (data-m)/s
print("^net:",net1)
net2 = net1 + 1
print("~net:",net2)
s_y = sigmoid(net2)
r_y = relu(net2)
print("activation function\nsigmoid:",s_y,"\nReLU:",r_y)
```

```
^net: tensor([-1.4985, -0.0937,  1.3112,  0.8429, -0.5620])
~net: tensor([-0.4985,  0.9063,  2.3112,  1.8429,  0.4380])
activation function
sigmoid: tensor([0.3779, 0.7123, 0.9098, 0.8633, 0.6078])
ReLU: tensor([0.0000, 0.9063, 2.3112, 1.8429, 0.4380])
```