Deep learning & applications

Practice#1-2

Tae Hyun Kim

Task: binary classification using logistic regression (loss = binary cross entropy loss)

Input: 2-dim vector, $\boldsymbol{x} = \{x_1, x_2\}$
Output: label of the input, **y** $\in \{0,1\}$

Pseudo code to generate dataset

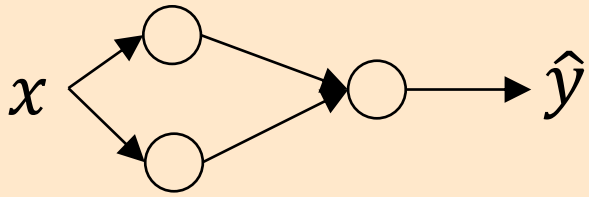
Step 1. Generate 10000(=m) train samples:

```
x1_train=[], x2_train=[], y_train=[]
for i in range(m):
    x1_train.append(random.uniform(-10, 10))
    x2_train.append(random.uniform(-10, 10))
    if x1_train[-1] < -5 or x1_train[-1] > 5:
        y_train.append(1)
    else:
        y_train.append(0)
```

Task: binary classification using logistic regression (loss = binary cross entropy loss)

Input: 2-dim vector, $\boldsymbol{x} = \{x_1, x_2\}$
Output: label of the input, **y** $\in \{0,1\}$

Pseudo code to train a 2-layered net.



Step 2. Initialize $\mathrm{W} = [w_1, w_2]$ and b (small random values? large random values? zeros?)

Step 3. Determine the learning rate α (let's tart with 1/100)

Step 4. Update $\mathrm{W} = [w_1, w_2], b$ with 'm' samples for 5000 (=K) iterations: #K updates with the grad descent

Step 4-1. Print W, b every 500 iterations

Step 4-2. Calculate the cost on the 'm' train samples!

Step 4-3. Assume that predicted output is 1 if $\tilde{y} > \mathbf{0.5}$, other wise 0. Then, print accuracy for the 'm' train samples! (display the number of correctly predicted outputs/m*100)