# #homework 2

심층신경망 구성 및 결과 분석

2015111576 최유진

# training 데이터 20개

```
# [털, 날개]
x_data = np.array(
    [[0, 0], [1, 0], [1, 1], [0, 0], [0, 0], [0, 1],[0, 0], [1, 0], [1, 1], [0, 0], [0, 0], [0, 1],[1, 0], [1, 1],[0, 0], [1, 0], [1,

# [기타, 포유류, 조류]
y_data = np.array([
    [1, 0, 0],  # 기타
    [0, 1, 0],  # 포유류
    [0, 0, 1],  # 조류
    [1, 0, 0],
    [1, 0, 0],
    [0, 0, 1],
    [1, 0, 0],
    [0, 1, 0],
    [0, 0, 1],
    [1, 0, 0],
    [1, 0, 0],
    [0, 0, 1],
    [0, 1, 0],
    [0, 0, 1],
    [1, 0, 0],
    [0, 1, 0],
    [0, 0, 1],
    [1, 0, 0],
    [1, 0, 0],
    [0, 0, 1]
])
```

# 심층 신경망

```
#[입력-2 , 히든레이어1-10]
W1 = tf.Variable(tf.random_uniform([2, 10], -1., 1.))
#[ 히든레이어1-10, 히든레이어2-20]
W2 = tf.Variable(tf.random_uniform([10, 20], -1., 1.))
#[ 히든레이어2-20, 히든레이어3-10]
W3 = tf.Variable(tf.random_uniform([20, 10], -1., 1.))
#[ 히든레이어3-10, 출력-3]
W4 = tf.Variable(tf.random_uniform([10, 3], -1., 1.))


# 편향
b1 = tf.Variable(tf.zeros([10]))
b2 = tf.Variable(tf.zeros([20]))
b3 = tf.Variable(tf.zeros([10]))
b4 = tf.Variable(tf.zeros([3]))
```

**입력(2) – 은닉층1 (10) – 은닉층2 (20) – 은닉층3 (10) – 출력(3)**
으로 되어 있는 모델을 생성

weight

[2,10] [10,20] [20,10] [10,3]

bias

10 : 은닉층의 뉴런수

20 : 은닉층의 뉴런수

10 : 은닉층의 뉴런수

3 : 분류 수

```
# 신경망의 히든 레이어에 가중치 W1과 편향 b1을 적용합니다
L1 = tf.add(tf.matmul(X, W1), b1)
L1 = tf.nn.relu(L1)

L2 = tf.add(tf.matmul(L1, W2), b2)
L2 = tf.nn.relu(L2)

L3 = tf.add(tf.matmul(L2, W3), b3)
L3 = tf.nn.relu(L3)

model = tf.add(tf.matmul(L3, W4), b4)
```

신경망의 히든 레이어 각각에 가중치 W과 편향 b을 적용하여 계산을 반복

최종적인 아웃풋

model을 만들어 냄

# 결과

```
# 결과 확인

prediction = tf.argmax(model, 1)
target = tf.argmax(Y, 1)
print('예측값:', sess.run(prediction, feed_dict={X: x_data}))
print('실제값:', sess.run(target, feed_dict={Y: y_data}))

is_correct = tf.equal(prediction, target)
accuracy = tf.reduce_mean(tf.cast(is_correct, tf.float32))
print('정확도: %.2f' % sess.run(accuracy * 100, feed_dict={X: x_data, Y: y_data}))
```

```
10 0.5807228
20 0.43780154
30 0.32483917
40 0.22939046
50 0.120833576
60 0.046552967
70 0.016437724
80 0.007242582
90 0.0041320985
100 0.0028378242
예측값: [0 1 2 0 0 2 0 1 2 0 0 2 1 2 0 1 2 0 0 2]
실제값: [0 1 2 0 0 2 0 1 2 0 0 2 1 2 0 1 2 0 0 2]
정확도: 100.00
```

- 10에서 100으로 진행됨에 따라 loss가 줄어들고 있음.
- 예측값과 실제값이 정확히 일치하여
- 정확도 100로 나타남