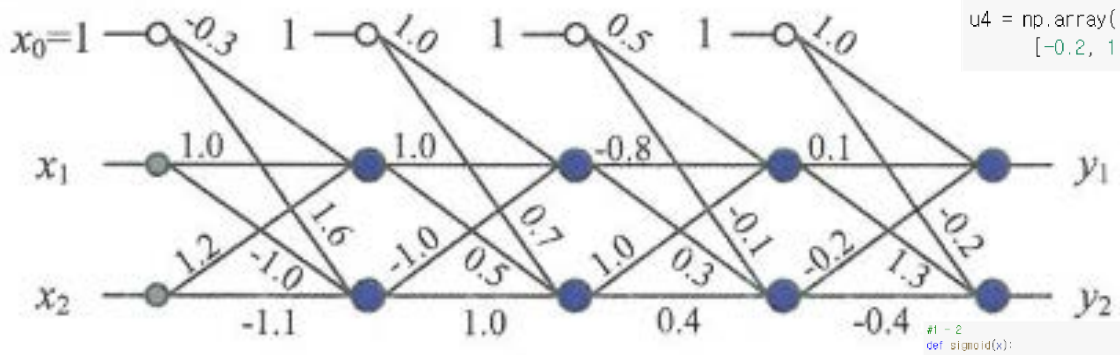
머신러닝 과제04

```
#1 - 1
u1 = np.array([[-0.3, 1.0, 1.2],
      [1.6, -1.0, -1.1]])

u2 = np.array([[1.0, 1.0, -1.0],
      [0.7, 0.5, 1.0]])

u3 = np.array([[0.5, -0.8, 1.0],
      [-0.1, 0.3, 0.4]])

u4 = np.array([[1.0, 0.1, -0.2],
      [-0.2, 1.3, -0.4]])
```

1 다음은 은닉층이 3개인 DMLP이다.

Hint 계산은 Matlab 또는 Python을 사용하시오.




```
#1 - 2
def sigmoid(x):
  return np.array([1 / (1 + np.exp(-x[0])), 1 / (1 + np.exp(-x[1]))])

bias = np.array([1])
x = np.array([1, 0])
x = np.append(bias, x)

x = np.matmul(u1, x)
x = sigmoid(x)
x = np.append(bias, x)

x = np.matmul(u2, x)
x = sigmoid(x)
x = np.append(bias, x)

x = np.matmul(u3, x)
x = sigmoid(x)
x = np.append(bias, x)

x = np.matmul(u4, x)
x = sigmoid(x)
print(x)
```

```
[0.72021291 0.60807077]
```

(1) 가중치 행렬 $\mathbf{U}^1, \mathbf{U}^2, \mathbf{U}^3, \mathbf{U}^4$를 식 (4.1)처럼 쓰시오.

가중치 행렬: $\mathbf{U}^l = \begin{pmatrix} u_{10}^l & u_{11}^l & \cdots & u_{1n_{l-1}}^l \\ u_{20}^l & u_{21}^l & & u_{2n_{l-1}}^l \\ \vdots & & \ddots & \vdots \\ u_{n_l 0}^l & u_{n_l 1}^l & \cdots & u_{n_l n_{l-1}}^l \end{pmatrix}, l = 1,2,\cdots$

(2) $\mathbf{x}=(\mathbf{1},\mathbf{0})^{\mathbf{T}}$가 입력되었을 때 출력 $\mathbf{y}$를 구하시오. 활성함수로 로지스틱 시그모이드를 사용하시오.

(3) $\mathbf{x}=(\mathbf{1},\mathbf{0})^{\mathbf{T}}$가 입력되었을 때 출력 $\mathbf{y}$를 구하시오. 활성함수로 ReLU를 사용하시오.

(4) $\mathbf{x}=(\mathbf{1},\mathbf{0})^{\mathbf{T}}$의 기대 출력이 $\mathbf{y}=(\mathbf{0},\mathbf{1})^{\mathbf{T}}$일 때, 현재 1.0인 u_{12}^3 가중치를 0.9로 줄이면 오류에 어떤 영향을 미치는지 설명하시오.

2 [그림 4-14]에서 나머지 8개 화소의 값을 계산하시오.

```
#1 - 3
def relu(x):
  return np.array([0 if x[0] < 0 else x[0], 0 if x[1] < 0 else x[1]])

x = np.array([1, 0])
x = np.append(bias, x)

x = np.matmul(u1, x)
x = relu(x)
x = np.append(bias, x)

x = np.matmul(u2, x)
x = relu(x)
x = np.append(bias, x)

x = np.matmul(u3, x)
x = relu(x)
x = np.append(bias, x)

x = np.matmul(u4, x)
x = relu(x)
print(x)
```

```
[0.949 1.095]
```

```
#1 - 4
def mse(y_pred):
  return (y_pred[0] - 0) ** 2 + (y_pred[1] - 1) ** 2

u3[0][1] = 0.9

x = np.array([1, 0])
x = np.append(bias, x)

x = np.matmul(u1, x)
x = sigmoid(x)
x = np.append(bias, x)

x = np.matmul(u2, x)
x = sigmoid(x)
x = np.append(bias, x)

x = np.matmul(u3, x)
x = sigmoid(x)
x = np.append(bias, x)

x = np.matmul(u4, x)
x = sigmoid(x)
b = x
print(mse(a), mse(b))
```

```
0.6723151580020207 0.6345325981380102
```

오차가 줄었다

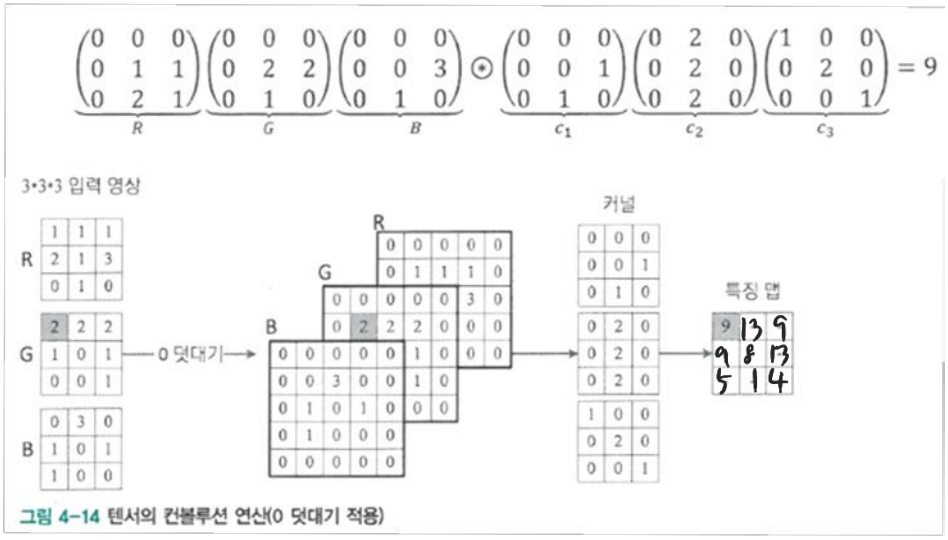


그림 4-14 텐서의 컨볼루션 연산(0 덧대기 적용)

3 [그림 4-8(b)]에서 커널 $\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$을 적용한 결과를 쓰시오. 이때 0 덧대기를 하고 바이어스로 0.5를 사용하시오.

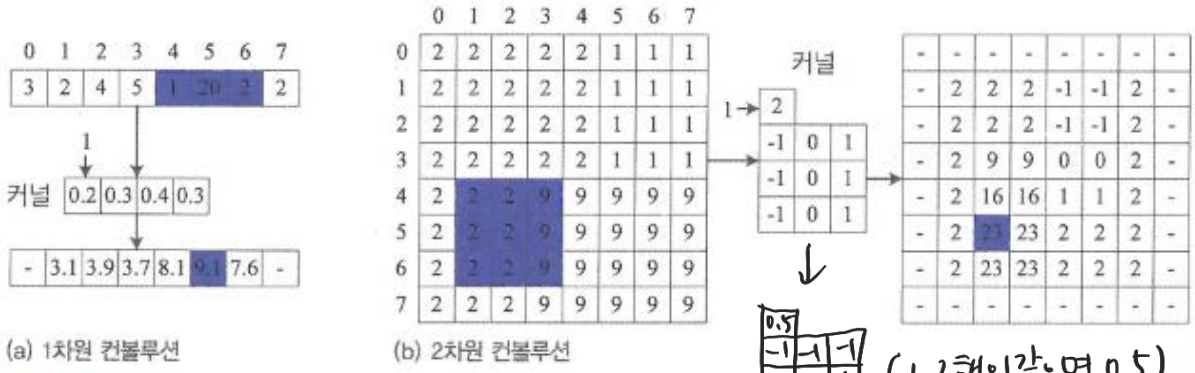


그림 4-8 바이어스

(1,3행이 같으면 0.5)

4 문제 3의 결과에 최대 풀링과 평균 풀링을 적용한 결과를 각각 쓰시오. 보폭으로 1을 사용하시오.