# Forward Propagation 예시
# MNIST Classification Model

# 수업 목표

이번 수업의 핵심:

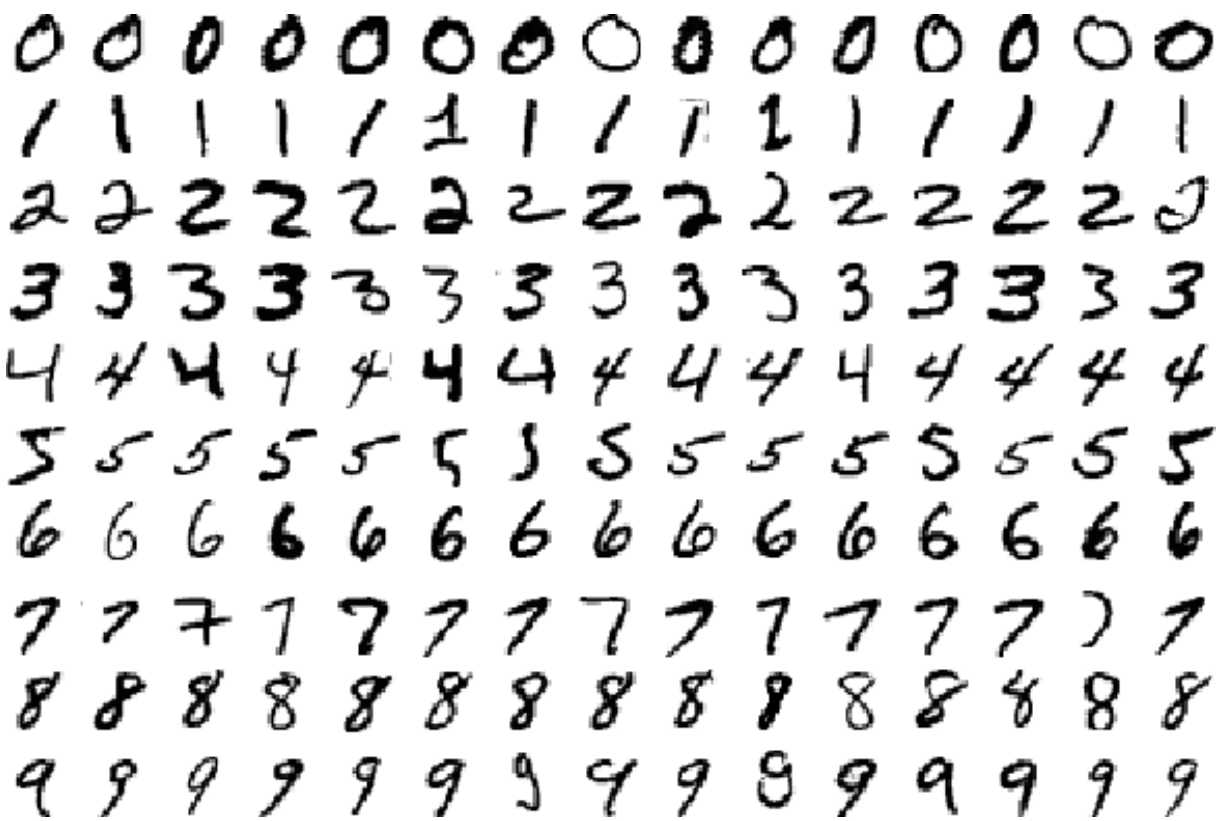
- MNIST Dataset 구성
- Neural Network 기반 MNIST Classification 모델
- Forward Propagation 코드 작성
- Error/Loss의 개념 확인

핵심 개념

- MNIST Dataset
- Classification Model
- Squared Error / Loss

# MNIST Dataset

## MNIST (Modified National Institute of Standards and Technology)

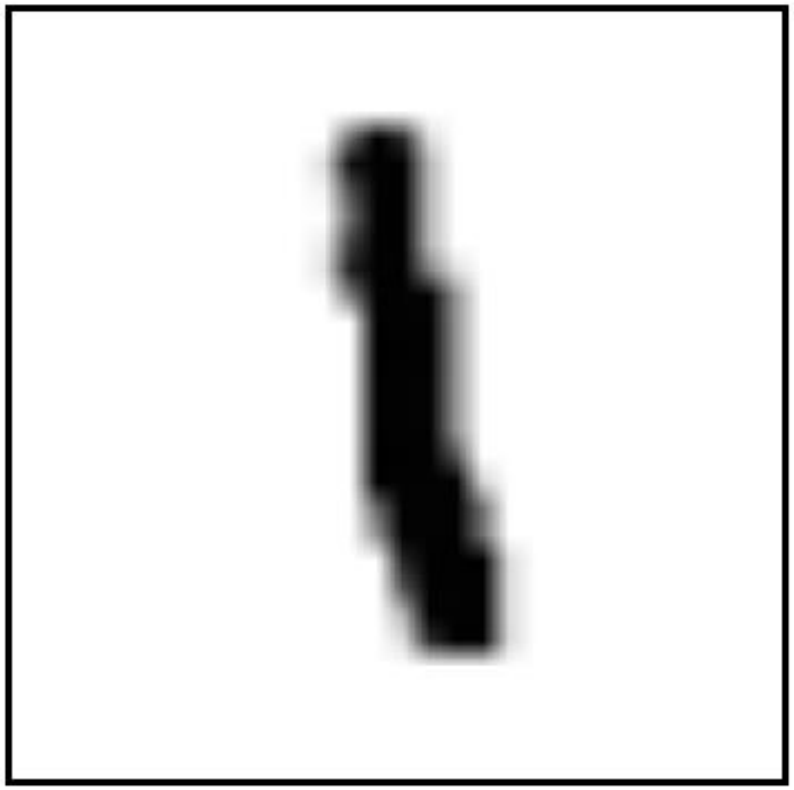


- 0에서 9까지 손글씨 숫자 사진
  - 55,000개의 Training Examples
  - 10,000개의 Test Examples
- 각 숫자 사진들은 전처리됨
  - 숫자가 사진의 중앙에 정렬
  - 숫자가 비슷한 크기로 조절
  - 각 사진이 $28 \times 28$ 픽셀 크기로 고정

  → 0부터 1사이의 실수 행렬

http://yann.lecun.com/exdb/mnist/

# MNIST 예시



≈

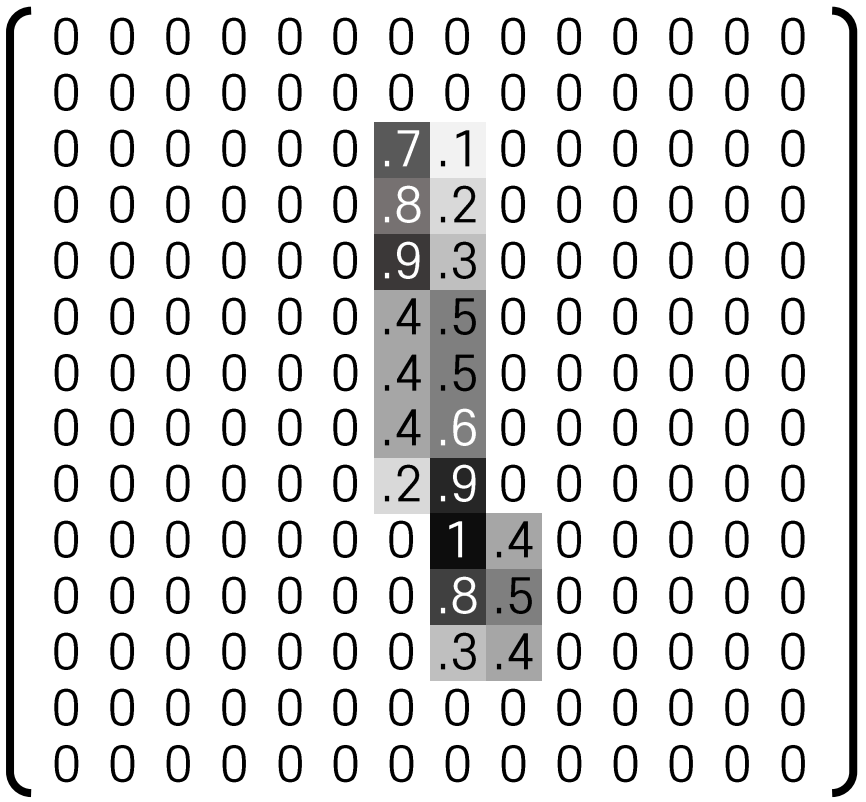
| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | .7 | .1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | .8 | .2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | .9 | .3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | .4 | .5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | .4 | .5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | .4 | .6 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | .2 | .9 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | .4 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | .8 | .5 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | .3 | .4 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

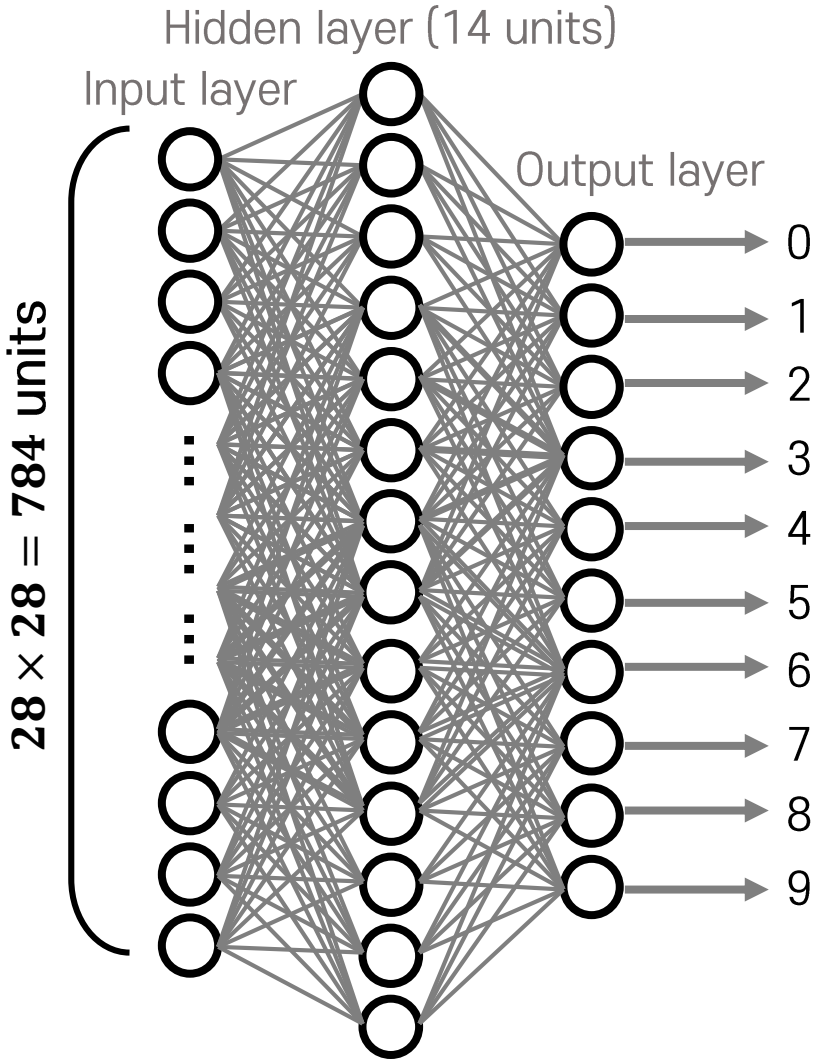
28

28

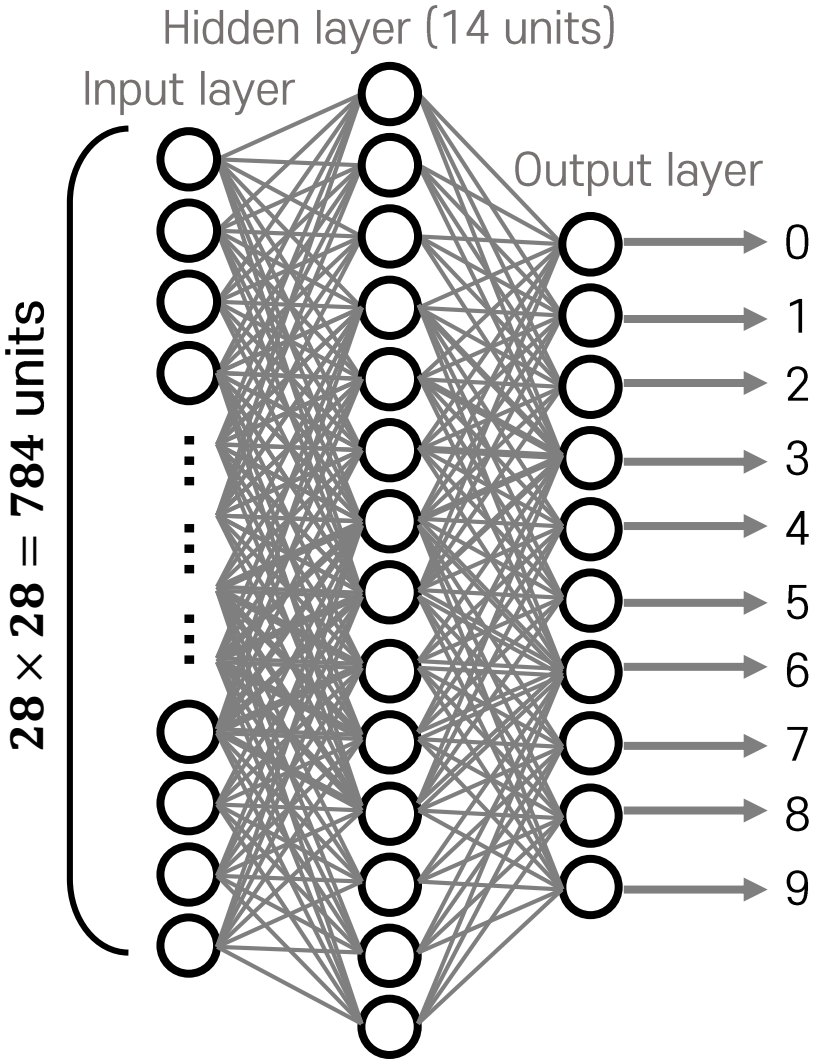
http://yann.lecun.com/exdb/mnist/

# MNIST Classification Model



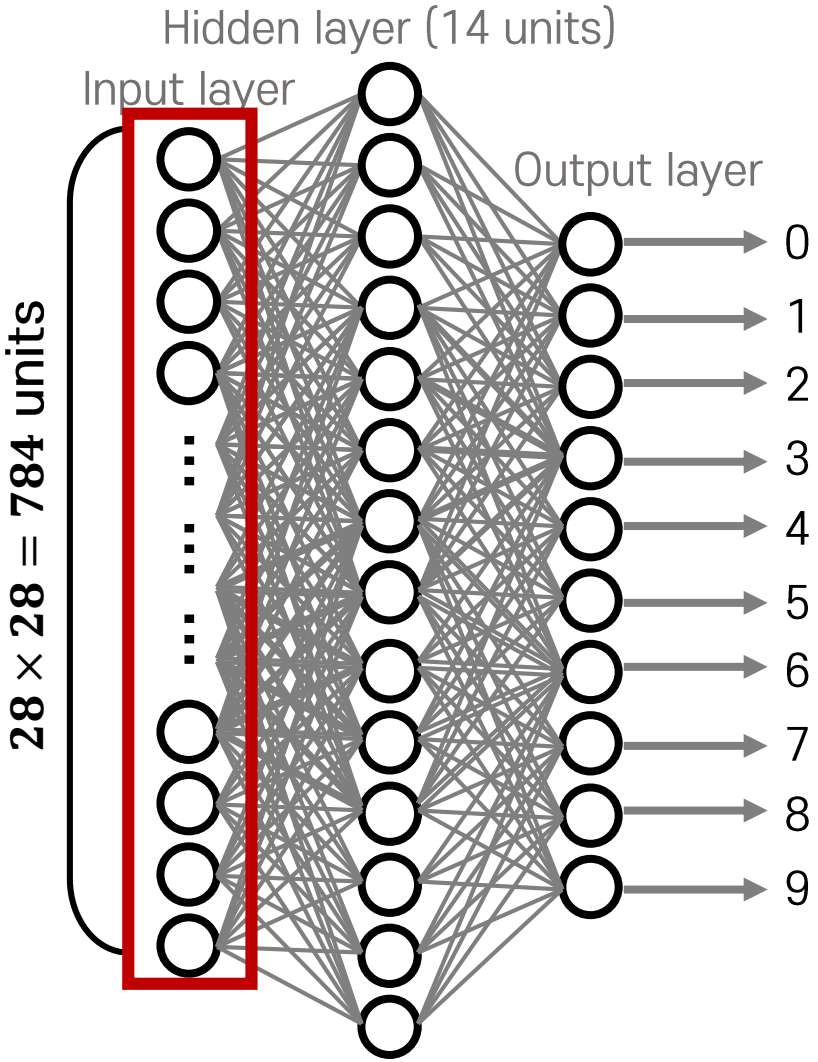



# MNIST Classification Model




```python
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def neural_network(x, w1, w2):
    # x: 784 x 1
    # w1: 14 x (784 + 1)
    # w2: 10 x (14 + 1)

    a1 = np.concatenate([x, [[1]]], 0)
    z2 = np.dot(w1, a1)
    a2 = sigmoid(z2)
    a2 = np.concatenate([a2, [[1]]], 0)
    z3 = np.dot(w2, a2)
    a3 = sigmoid(z3)
    return a3
```

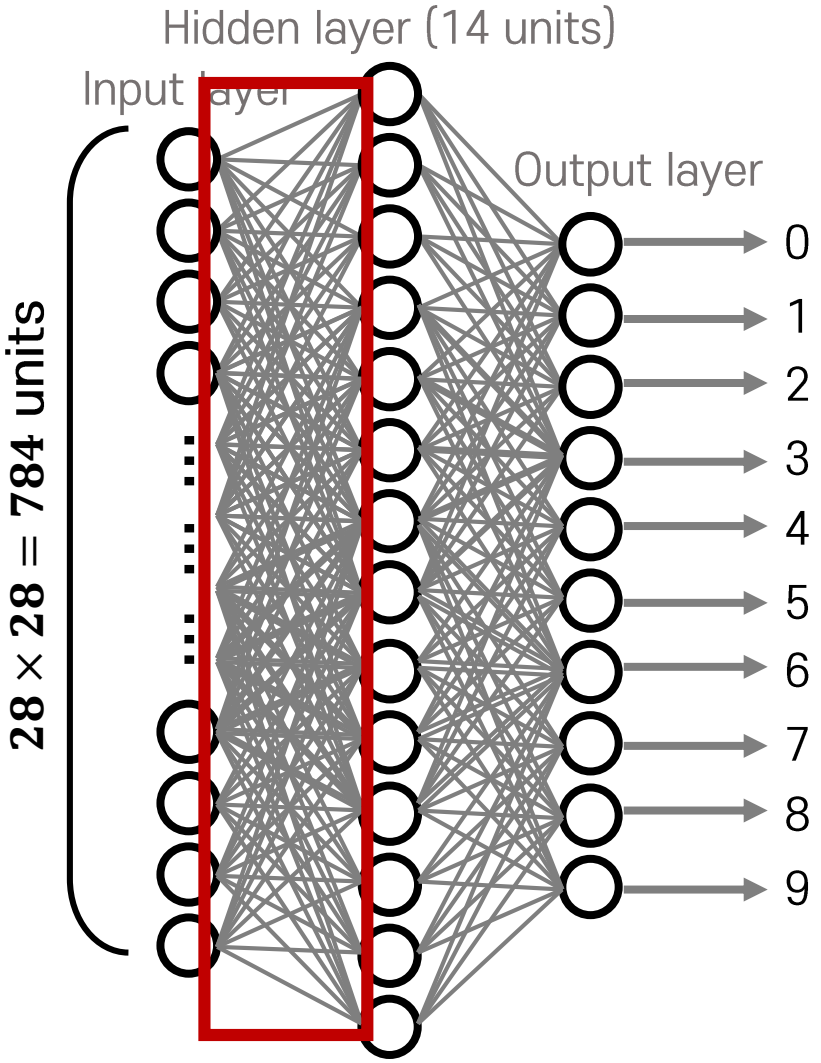
# MNIST Classification Model




```python
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def neural_network(x, w1, w2):
    # x: 784 x 1
    # w1: 14 x (784 + 1)
    # w2: 10 x (14 + 1)

    a1 = np.concatenate([x, [[1]]], 0)
    z2 = np.dot(w1, a1)
    a2 = sigmoid(z2)
    a2 = np.concatenate([a2, [[1]]], 0)
    z3 = np.dot(w2, a2)
    a3 = sigmoid(z3)
    return a3
```

785 x 1

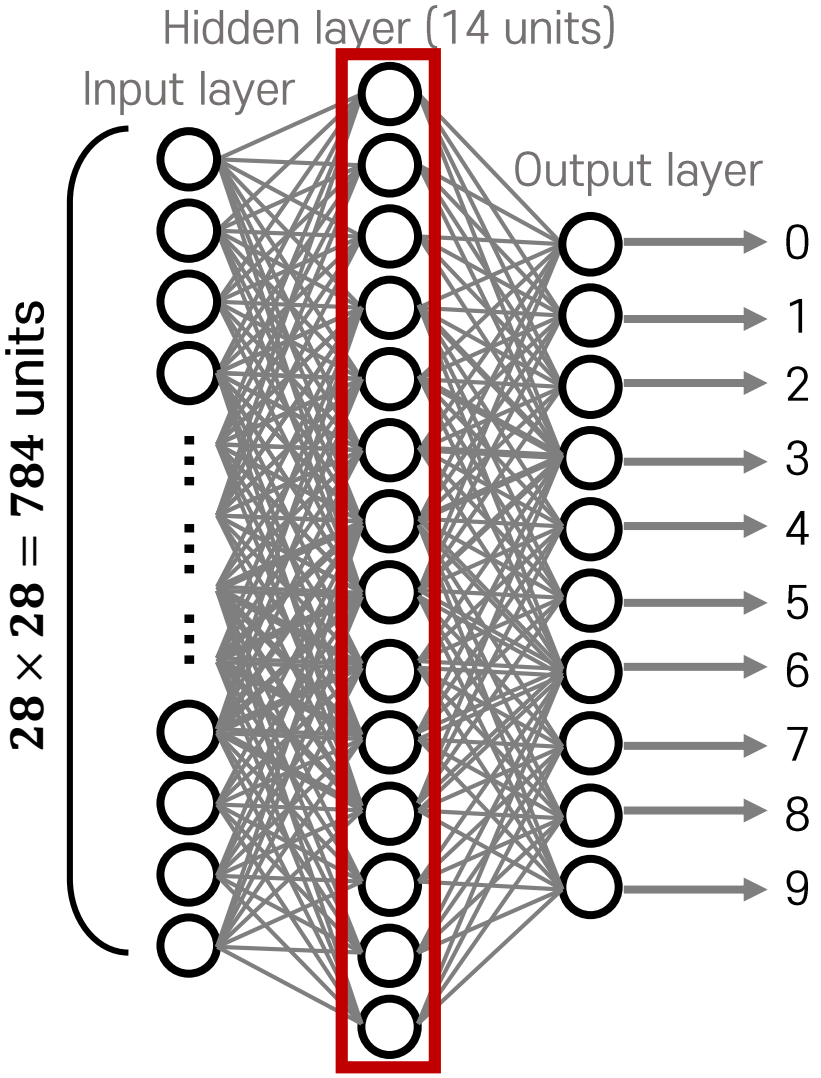
# MNIST Classification Model




```python
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def neural_network(x, w1, w2):
    # x: 784 x 1
    # w1: 14 x (784 + 1)
    # w2: 10 x (14 + 1)

    a1 = np.concatenate([x, [[1]]], 0)
    z2 = np.dot(w1, a1)
    a2 = sigmoid(z2)
    a2 = np.concatenate([a2, [[1]]], 0)
    z3 = np.dot(w2, a2)
    a3 = sigmoid(z3)
    return a3
```

785 x 1

# MNIST Classification Model

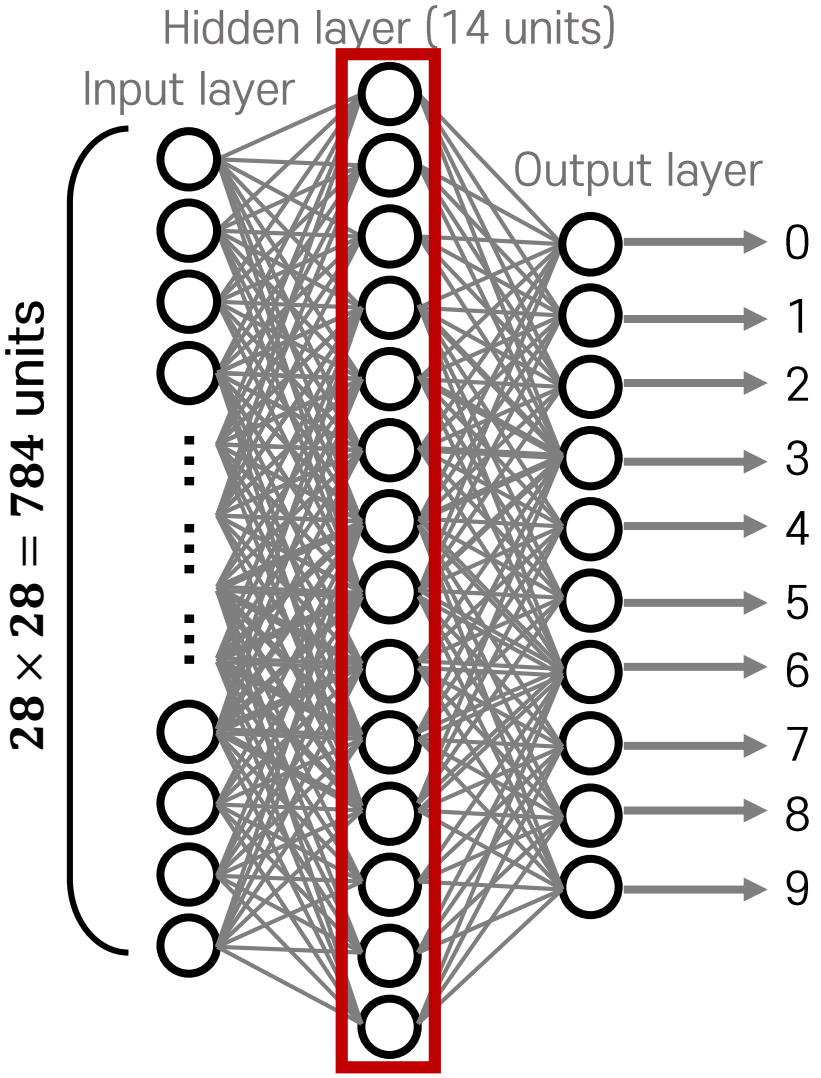



```python
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def neural_network(x, w1, w2):
    # x: 784 x 1
    # w1: 14 x (784 + 1)
    # w2: 10 x (14 + 1)

    a1 = np.concatenate([x, [[1]]], 0)
    z2 = np.dot(w1, a1)
    a2 = sigmoid(z2)
    a2 = np.concatenate([a2, [[1]]], 0)
    z3 = np.dot(w2, a2)
    a3 = sigmoid(z3)
    return a3
```

785 x 1

14 x 1

# MNIST Classification Model



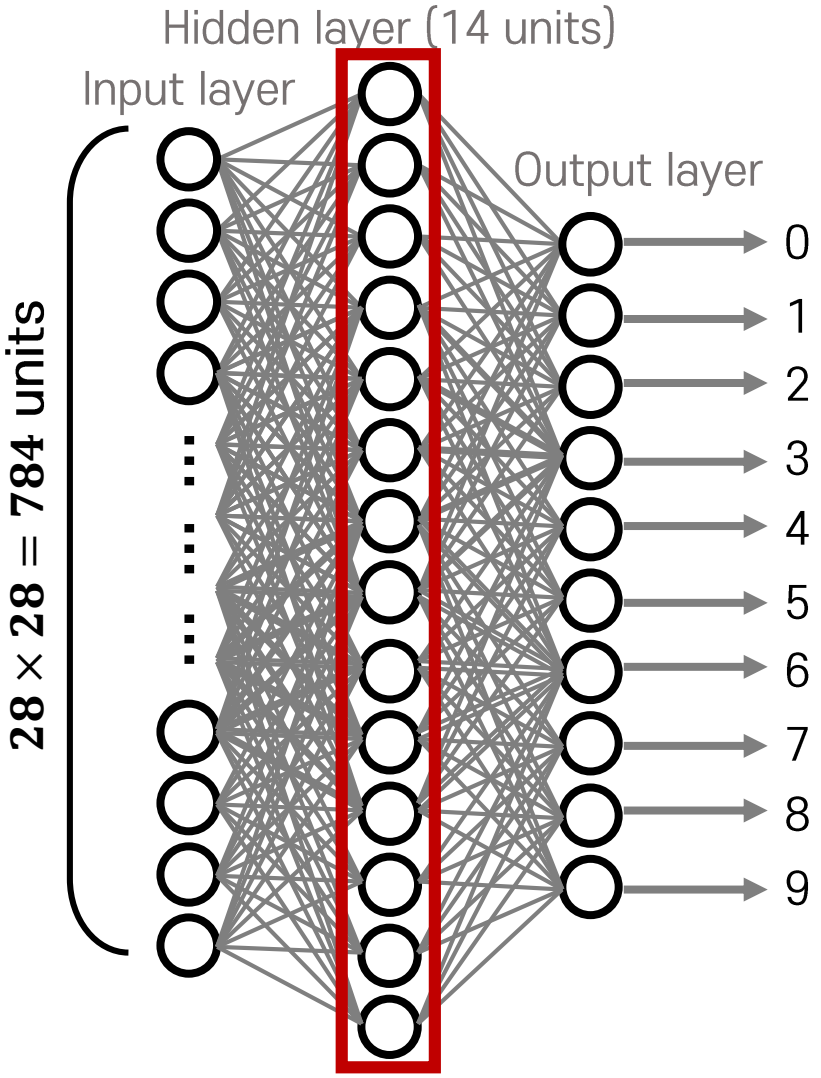

```python
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def neural_network(x, w1, w2):
    # x: 784 x 1
    # w1: 14 x (784 + 1)
    # w2: 10 x (14 + 1)

    a1 = np.concatenate([x, [[1]]], 0)
    z2 = np.dot(w1, a1)
    a2 = sigmoid(z2)
    a2 = np.concatenate([a2, [[1]]], 0)
    z3 = np.dot(w2, a2)
    a3 = sigmoid(z3)
    return a3
```

785 x 1

14 x 1

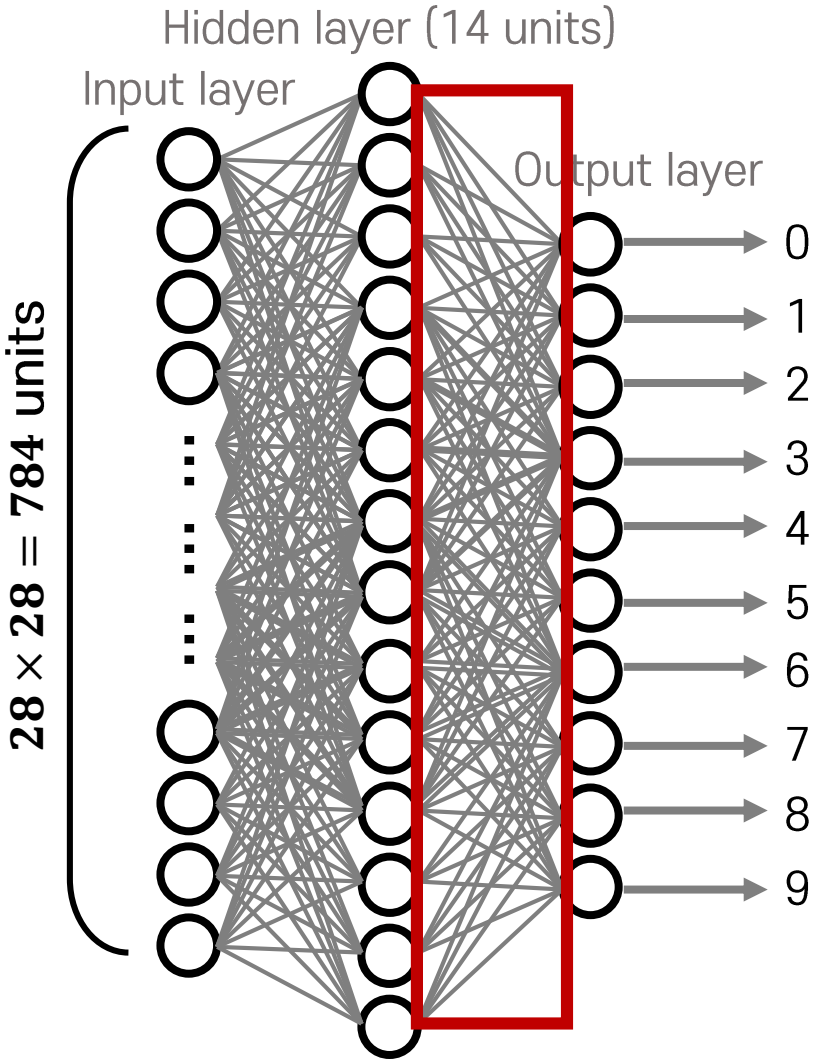
14 x 1

# MNIST Classification Model



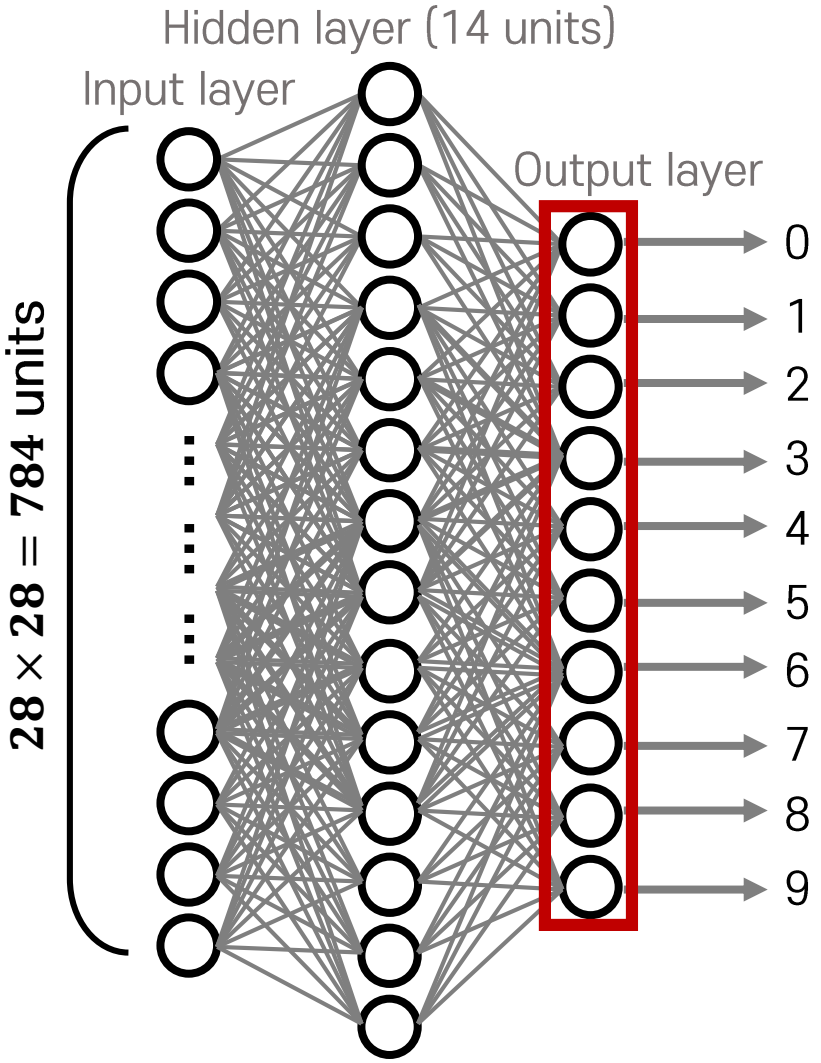

```python
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def neural_network(x, w1, w2):
    # x: 784 x 1
    # w1: 14 x (784 + 1)
    # w2: 10 x (14 + 1)

    a1 = np.concatenate([x, [[1]], 0)  785 x 1
    z2 = np.dot(w1, a1)  14 x 1
    a2 = sigmoid(z2)     14 x 1
    a2 = np.concatenate([a2, [[1]], 0)  15 x 1
    z3 = np.dot(w2, a2)
    a3 = sigmoid(z3)
    return a3
```

# MNIST Classification Model



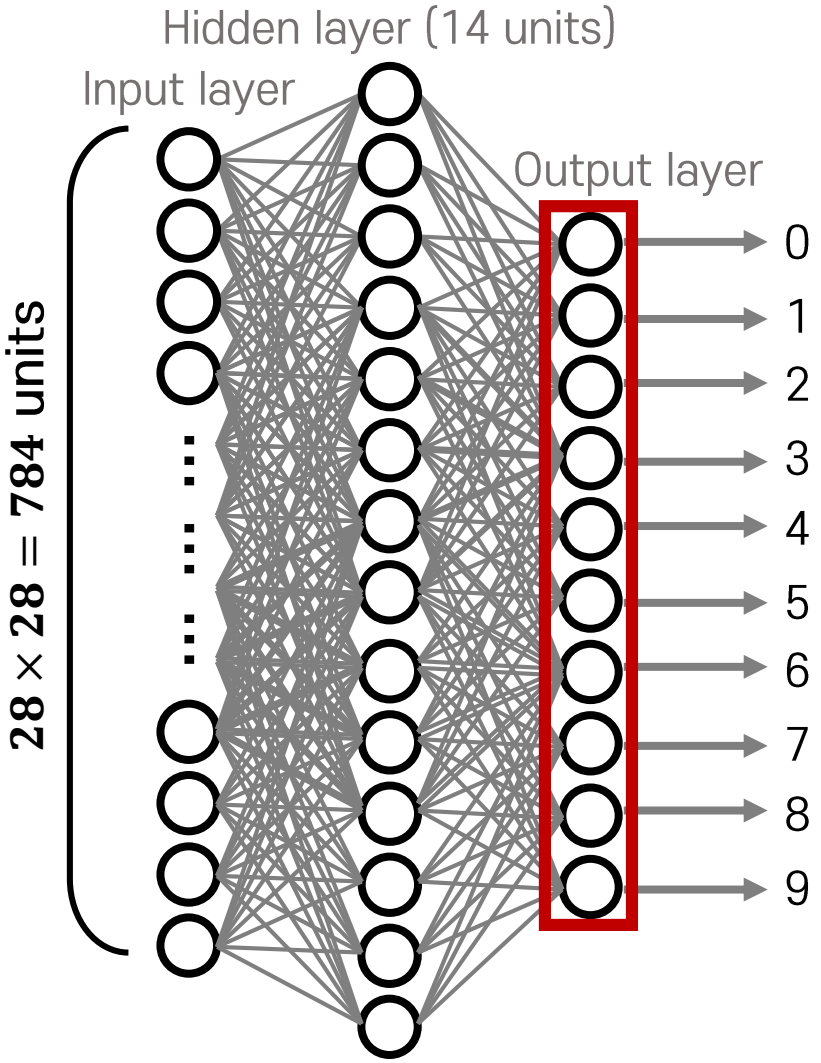

```python
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def neural_network(x, w1, w2):
    # x: 784 x 1
    # w1: 14 x (784 + 1)
    # w2: 10 x (14 + 1)

    a1 = np.concatenate([x, [[1]]], 0)  # 785 x 1
    z2 = np.dot(w1, a1)                 # 14 x 1
    a2 = sigmoid(z2)                    # 14 x 1
    a2 = np.concatenate([a2, [[1]]], 0) # 15 x 1
    z3 = np.dot(w2, a2)
    a3 = sigmoid(z3)
    return a3
```

# MNIST Classification Model



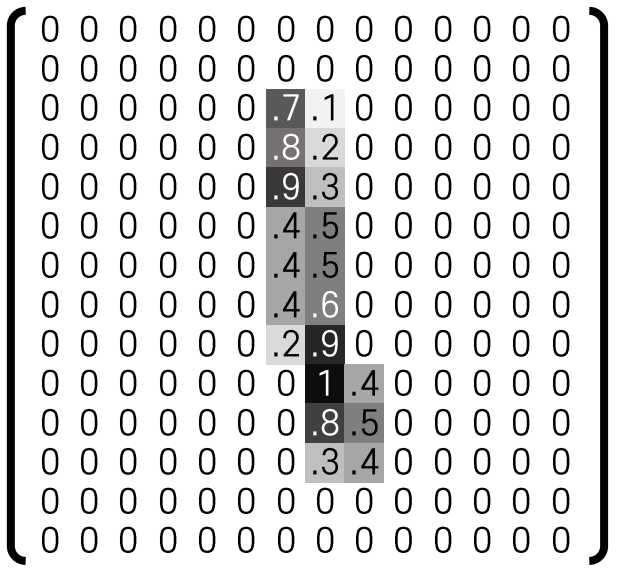

```python
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def neural_network(x, w1, w2):
    # x: 784 x 1
    # w1: 14 x (784 + 1)
    # w2: 10 x (14 + 1)

    a1 = np.concatenate([x, [[1]]], 0)  785 x 1
    z2 = np.dot(w1, a1)  14 x 1
    a2 = sigmoid(z2)     14 x 1
    a2 = np.concatenate([a2, [[1]]], 0)  15 x 1
    z3 = np.dot(w2, a2)  10 x 1
    a3 = sigmoid(z3)
    return a3
```

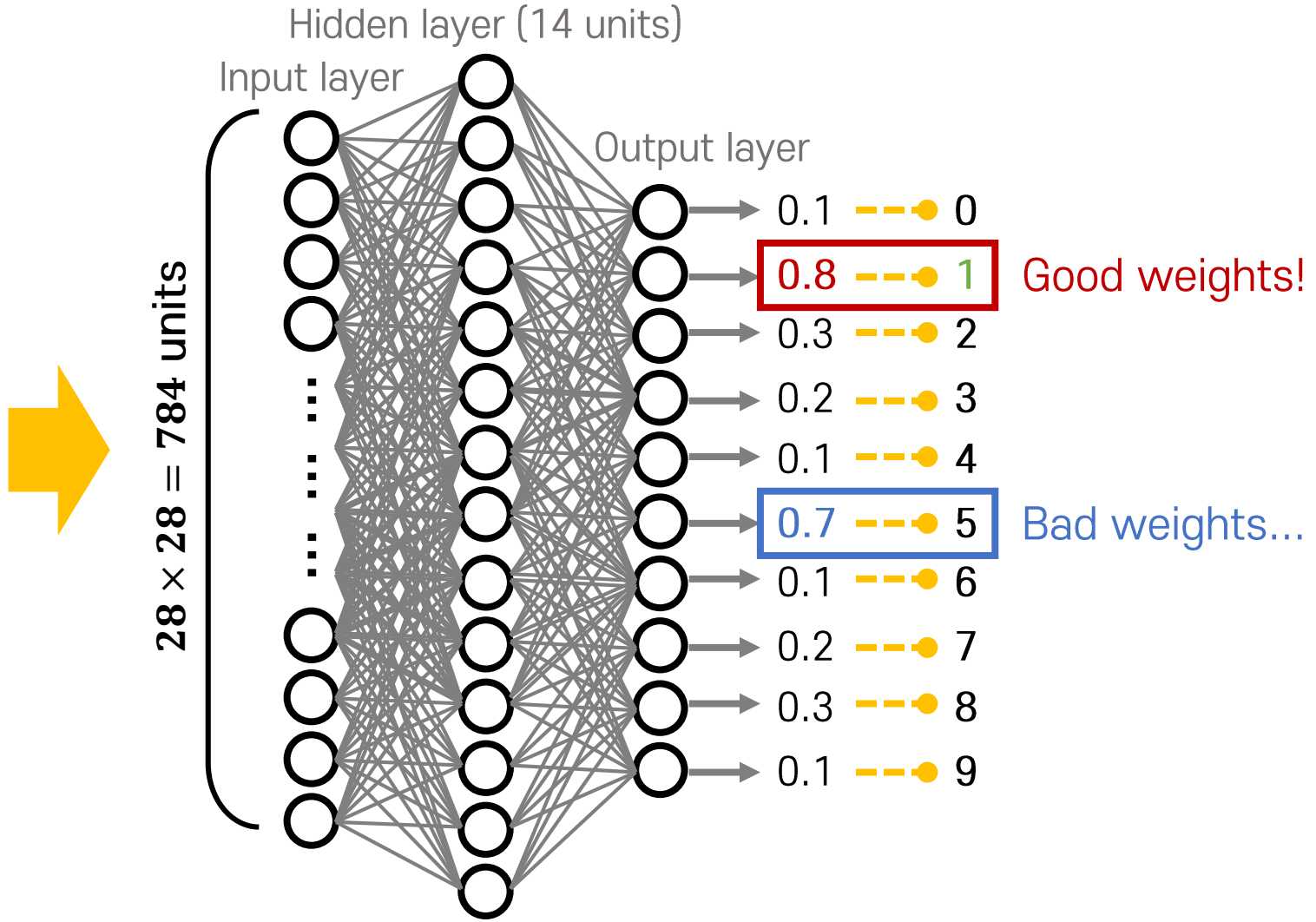
# MNIST Classification Model



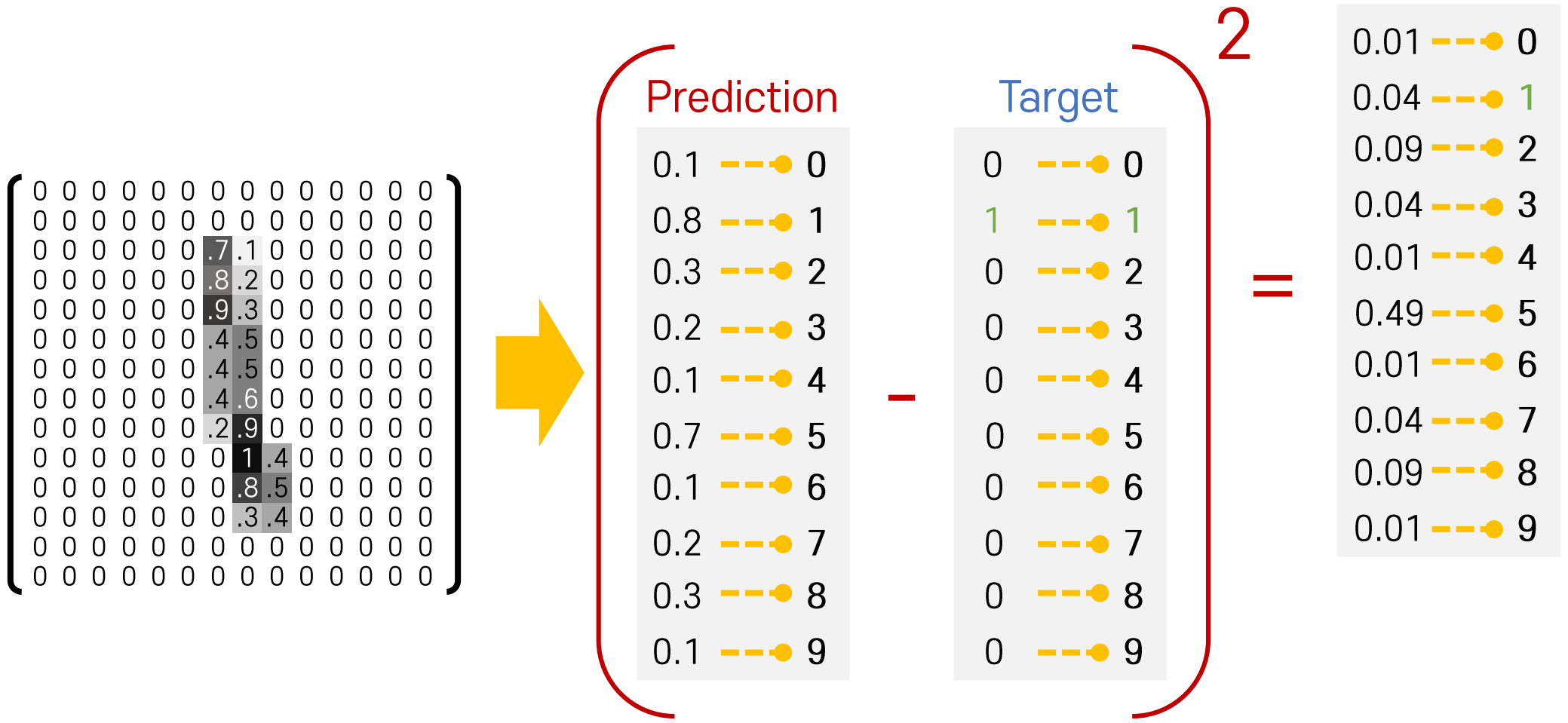

```python
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def neural_network(x, w1, w2):
    # x: 784 x 1
    # w1: 14 x (784 + 1)
    # w2: 10 x (14 + 1)

    a1 = np.concatenate([x, [[1]]], 0)  785 x 1
    z2 = np.dot(w1, a1)  14 x 1
    a2 = sigmoid(z2)     14 x 1
    a2 = np.concatenate([a2, [[1]]], 0)  15 x 1
    z3 = np.dot(w2, a2)  10 x 1
    a3 = sigmoid(z3)  10 x 1
    return a3
```

# MNIST Classification Model



# MNIST Classification Model




Squared Error (Loss): 0.83 Error/Loss를 최소화하는 Weight를 찾자!

# 요약

- MNIST Dataset Classification 모델 제작
- Neural Network를 Forward Propagation하는 예시 코드 확인
- Error / Loss를 최소화하는 목표 제시