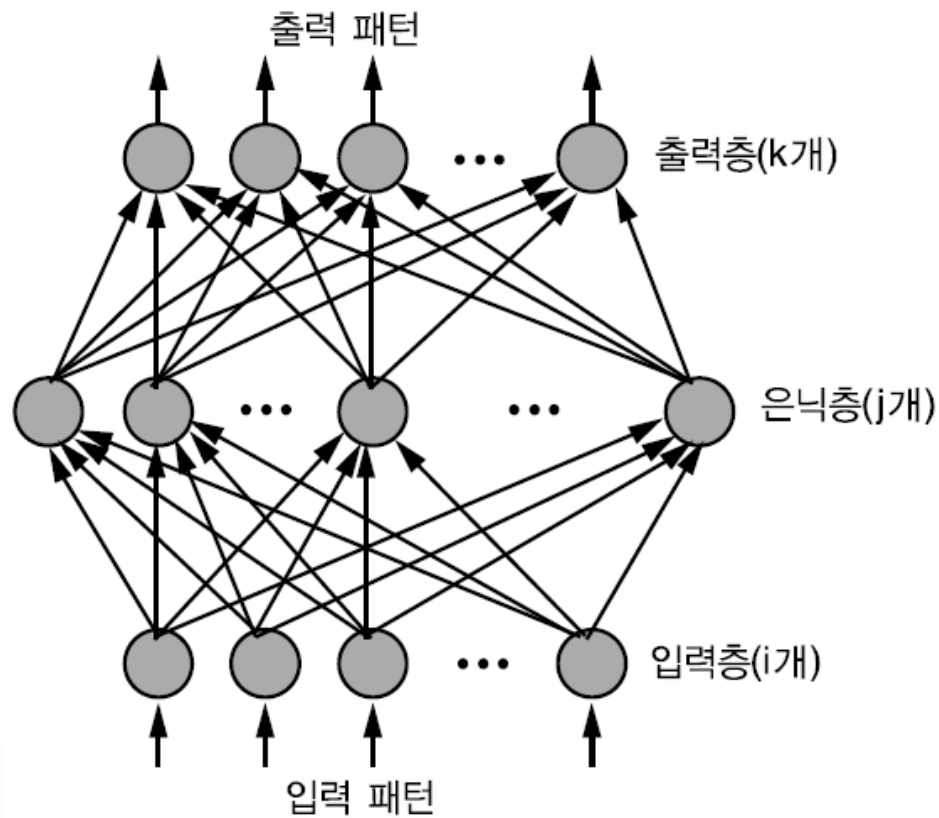


Python

ANN

ANN

- ANN



ANN

- 입력 데이터


ANN

- Activation Function

```
def sigmoidFunc(totalInput):  
    return 1.0 / (np.ones(totalInput.shape) + np.exp(-1.0 * totalInput))
```

ANN

- 출력층과 은닉층간의 연결 강도의 변화

```
def outputDeltas(output, target):  
    #details, output, target은 모두 벡터  
     sigmoidDeriv = output * (np.ones(output.shape) - output)  
    return 2 * (target - output) * sigmoidDeriv
```

ANN

- 은닉층과 입력층 간의 연결 강도의 변화

```
def hiddenDeltas(outputDeltas, hiddenOutputs, outputWeights):  
    # deltas, outputs은 열벡터.  
    # output Weights는 hidden->output 가중치행렬(각 행은 출력 유닛의 가중치 벡터)  
    sigmoidDeriv = hiddenOutputs * (np.ones(hiddenOutputs.shape) - hiddenOutputs)  
    return (np.dot(outputWeights.T, outputDeltas.T)) * sigmoidDeriv
```

ANN

- 파라미터 초기화

```
patterns = data.T
nPats = data.shape[0]
nTrainingPats = 20
nTestPats = 20
nInputs = data.shape[1]
nHidden = 10
nOutputs = 4

hiddenWeights = 0.5 * (np.random.rand(nHidden, nInputs+1) - np.ones((nHidden, nInputs+1)) * 0.5)
outputWeights = 0.5 * (np.random.rand(nOutputs, nHidden+1) - np.ones((nOutputs, nHidden+1)) * 0.5)

input = patterns
target = np.zeros((nOutputs, nPats))
classNum = 0
eta = 0.1
NEpochs = 1000

for pat in range(0, nPats, 1):
    target[classNum, pat] = 1
    classNum = classNum + 1
    if classNum >= nOutputs:
        classNum = 0

ErrorsLastNEpochs = np.zeros((1, NEpochs))
TestErrorsLastNEpochs = np.zeros((1, NEpochs))
```

ANN

- 전방향 패스 & 역방향 패스

```
for epoch in range(0, NEpochs, 1):
    sumSqrError = 0.0
    sumSqrTestError = 0.0
    outputWGrad = np.zeros(outputWeights.shape)
    hiddenWGrad = np.zeros(hiddenWeights.shape)

    for pat in range(0, nTrainingPats, 1):
        # 전방향패스(pass)
        inp = np.hstack([input[:, pat], np.array([1])])
        hiddenNetInputs = np.dot(hiddenWeights, inp)
        hiddenStates = sigmoidFunc(hiddenNetInputs)
        hidStatesBias = np.hstack([hiddenStates, np.array([1])])
        outputNetInputs = np.dot(outputWeights, hidStatesBias)
        outputStates = sigmoidFunc(outputNetInputs)

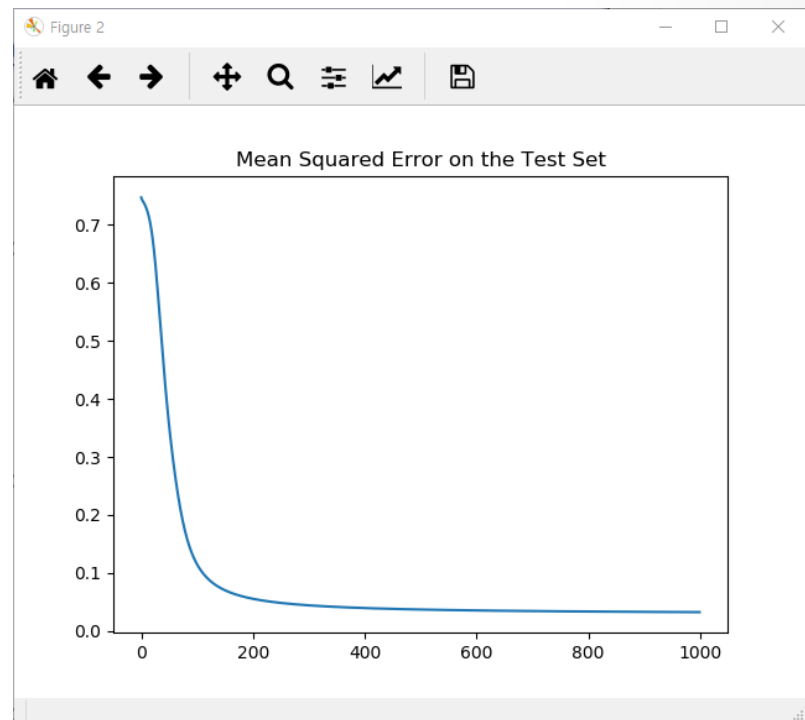
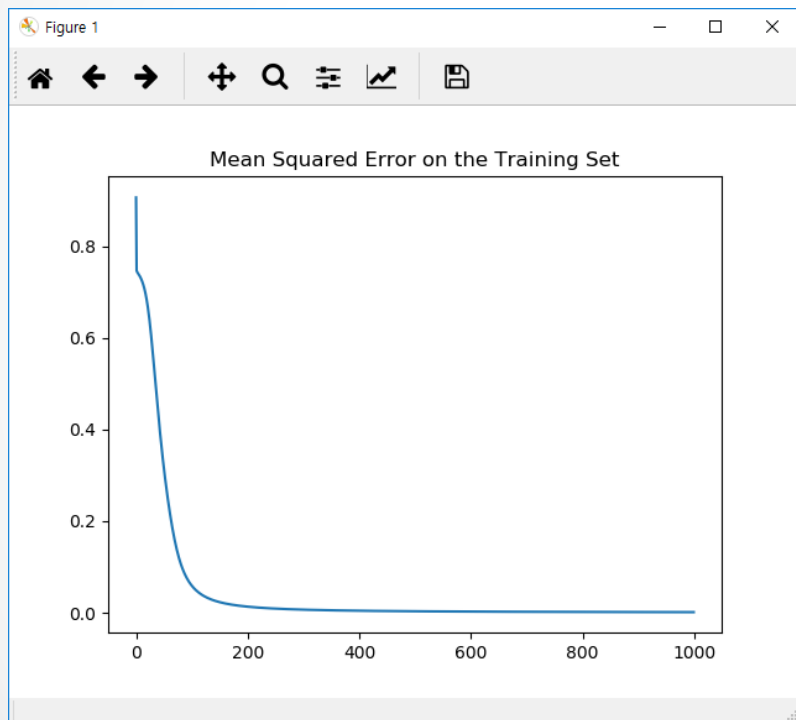
        #역방향 패스(pass)
        targetStates = target[:, pat]
        error = outputStates - targetStates
        sumSqrError = sumSqrError + np.dot(error, error)
        outputDel = outputDeltas(outputStates, targetStates)
        outputWGrad = outputWGrad + np.dot(np.array([outputDel]).T, np.array([hidStatesBias]))
        hiddenDel = hiddenDeltas(outputDel, hidStatesBias, outputWeights)
        hiddenDelArray = np.array([hiddenDel])
        hiddenWGrad = hiddenWGrad + np.dot(hiddenDelArray[:, 0:nHidden].T, np.array([inp]))
```


ANN

- 연결 가중치 갱신

```
outputWChange = eta * outputWGrad  
outputWeights = outputWeights + outputWChange  
hiddenWChange = eta * hiddenWGrad  
hiddenWeights = hiddenWeights + hiddenWChange
```

ANN



ANN

- Exercise
 - 교재를 참고하여 성능 테스트와 결과를 출력하는 코드를 완성하시오.