

Artificial Neural Network

Home Work - 1

Trilokinath Modi

September 18, 2020

1 Problem 1

```
import numpy as np
from numpy import array

numOfPatterns = [12, 24, 48, 70, 100, 120] # Assume this as a set of patterns
numOfBits = 120
numOfSets = len(numOfPatterns)
randomNumberList = [1, 1, 1, 1, 1, 1]
randomNeuronNumber = 1
storedPattern = list()
weightMatrix = list()
errorProbabilityCount = [0, 0, 0, 0, 0, 0]
errorProbability = [0, 0, 0, 0, 0, 0]
numOfIterations = 100000

for iIter in range(numOfIterations):
    if iIter % 10000 == 0:
        print("Iteration_completed_", iIter)
    for iSet in range(len(numOfPatterns)):
        matrixSize = [numOfPatterns[iSet], numOfBits]
        storedPattern.append(array([np.zeros(matrixSize)], int))
        # print(storedPattern[iSet][0][1])
        for iPattern in range(numOfPatterns[iSet]):
            randomPattern = np.random.randint(0, 2, numOfBits)
            # print(randomPattern)
            storedPattern[iSet][0][
                iPattern] = randomPattern.copy()
            # Note, the three indices is for 1-iSet, 2-2D array, 3-Row of 2Darray
            storedPattern[iSet] = 2 * storedPattern[iSet] - 1

# Select a random pattern for each set of random patterns and a random neuron
for iRandom in range(numOfSets):
    randomNumberList[iRandom] = np.random.randint(0, numOfPatterns[iRandom], 1)
    randomNeuronNumber = np.random.randint(0, numOfBits, 1)

# Weight Matrix
for iSet in range(numOfSets):
    weightMatrix.append(np.zeros([numOfBits, numOfBits]))
    for iPattern in range(numOfPatterns[iSet]):
        weightMatrix[iSet] += np.outer(storedPattern[iSet][0][iPattern],
                                         storedPattern[iSet][0][iPattern])
    # np.fill_diagonal(weightMatrix[iSet], 0)
    weightMatrix[iSet] = (1/numOfBits)*weightMatrix[iSet]

# Asynchronous update
for iSet in range(numOfSets):
    newStateForNeuron = np.dot(weightMatrix[iSet][randomNeuronNumber][0],
```

```

                                storedPattern[iSet][0][randomNumberList[iSet]][0])
if newStateForNeuron == 0:
    newStateForNeuron = 1
newStateForNeuron = np.sign(newStateForNeuron)
if newStateForNeuron != \
    storedPattern[iSet][0][randomNumberList[iSet]][0][randomNeuronNumber]:
    errorProbabilityCount[iSet] += 1
weightMatrix.clear()
storedPattern.clear()
if iIter % 10000 == 0:
    errorProbability = [ePC / (iIter + 1) for ePC in errorProbabilityCount]
    print(np.around(errorProbability, 3))
errorProbability = [ePC/numOfIterations for ePC in errorProbabilityCount]
print(errorProbability)

```