

# Artificial Neural Network

## Home Work - 1

Trilokinath Modi

September 18, 2020

### 1 Problem 3

```
import numpy as np
import math
from numpy import array

numOfPatterns = 45  # Assume this as a set of patterns
numOfBits = 200
noiseParameter = 2
storedPattern = list()
weightMatrix = list()
numOfIterations = 1000
numOfExperiments = 100
newStateForNeuron = list(np.zeros(numOfBits))
fedPattern = list(np.zeros(numOfBits))
orderParameter = list(np.zeros(numOfExperiments))
avgOrderParameter = 0

for iExpt in range(numOfExperiments):
    print("Experiment", iExpt)
    matrixSize = [numOfPatterns, numOfBits]
    storedPattern.append(array([np.zeros(matrixSize)], int))
    matrixSize = [numOfBits, numOfBits]
    weightMatrix = np.zeros(matrixSize)
    for iPattern in range(numOfPatterns):
        storedPattern[0][0][iPattern] = np.random.randint(0, 2, numOfBits)
        storedPattern[0][0][iPattern] = 2*storedPattern[0][0][iPattern] - 1
        weightMatrix += np.outer(storedPattern[0][0][iPattern],
                                storedPattern[0][0][iPattern])
    np.fill_diagonal(weightMatrix, 0)
    weightMatrix = (1 / numOfBits) * weightMatrix
    fedPattern = storedPattern[0][0][0].copy() # feeding first stored pattern
    for iIter in range(numOfIterations):
        if (iIter+1) % 10000 == 0:
            print("Iteration_completed_", iIter)
            # Asynchronous update
        for iNeuron in range(numOfBits):
            newStateForNeuron[iNeuron] = np.dot(weightMatrix[iNeuron], fedPattern)
            newStateForNeuron[iNeuron] = 1 / (1 + math.exp(-2 * noiseParameter *
                                                            newStateForNeuron[iNeuron]))

            if newStateForNeuron[iNeuron] >= np.random.rand():
                newStateForNeuron[iNeuron] = 1
            else:
                newStateForNeuron[iNeuron] = -1
            newStateForNeuron[iNeuron] = np.sign(newStateForNeuron[iNeuron])
            fedPattern[iNeuron] = newStateForNeuron[iNeuron].copy()
    orderParameter[iExpt] += (1/numOfBits)*np.dot(newStateForNeuron,
```

```
                                storedPattern[0][0][0])
    orderParameter[iExpt] = orderParameter[iExpt]/numOfIterations
    storedPattern.clear()
avgOrderParameter = np.sum(orderParameter)/numOfExperiments
print(avgOrderParameter)
```