

FFR135, Artificial Neural Networks  
**Home Problem 1**  
Stochastic Hopfield network

23 september 2019

Ella Guiladi  
930509-0822  
guiladi@student.chalmers.se

# 1 Stochastic Hopfield network

## 1.1 Weight Matrix

```
function weightMatrix=WeightMatrix(patterns,diagElements)

sizePattern=size(patterns);
numberBits = sizePattern(1);
weightMatrix=0;

for j=1:numberBits
    for i=1:numberBits
        weightMatrix(i,j)=sum(patterns(i,:).*patterns(j,:));
    end
end
if diagElements==0
    weightMatrix = (1/(numberBits).*(weightMatrix-eye(numberBits)).*←
        weightMatrix(1,1)));
else
    weightMatrix = 1/(numberBits).*weightMatrix;
end
end
```

## 1.2 Generate Patterns

```
function randPatterns=GeneratePatterns(numberBits,numberPatterns)
randPatterns=rand(numberBits,numberPatterns);
randPatterns=sign(randPatterns-0.5*ones(numberBits,numberPatterns));

end
```

### 1.3 Stochastic Hopfield network

```
clear
clc
rng(123)
numberBits=200;
numberPatterns=7; %Change to 45 here
numberUpdates=2*10^5; %T
noiseParameter=2;
diagElements=0;
numberExperiment=100;

orderParameterSum=0;
for anExperiment=1:numberExperiment

    patterns=GeneratePatterns(numberBits,numberPatterns);
    patternOne=patterns(:,1);
    updatedPatternOne=patternOne;
    weightMatrix=WeightMatrix(patterns,diagElements);

    orderParameter=0;
    for anUpdate=1:numberUpdates

        bit=randi([1 numberBits],1);
        meanField=sum(weightMatrix(:,bit).*updatedPatternOne);

        g=1/(1+exp(-2*noiseParameter*meanField));
        probabilityOfg=rand;

        if probabilityOfg < g
            updatedPatternOne(bit)=1;
        else
            updatedPatternOne(bit)=-1;
        end

        my=0;
        for aBit=1:numberBits
            my = my + patternOne(aBit)*updatedPatternOne(aBit);
        end

        orderParameter = orderParameter + my/numberBits;
    end

    orderParameterSum = orderParameterSum + orderParameter/numberUpdates;
end

orderParameterAverage=orderParameterSum/numberExperiment
```