

Machine Learning



Nama : Mochammad Raditya

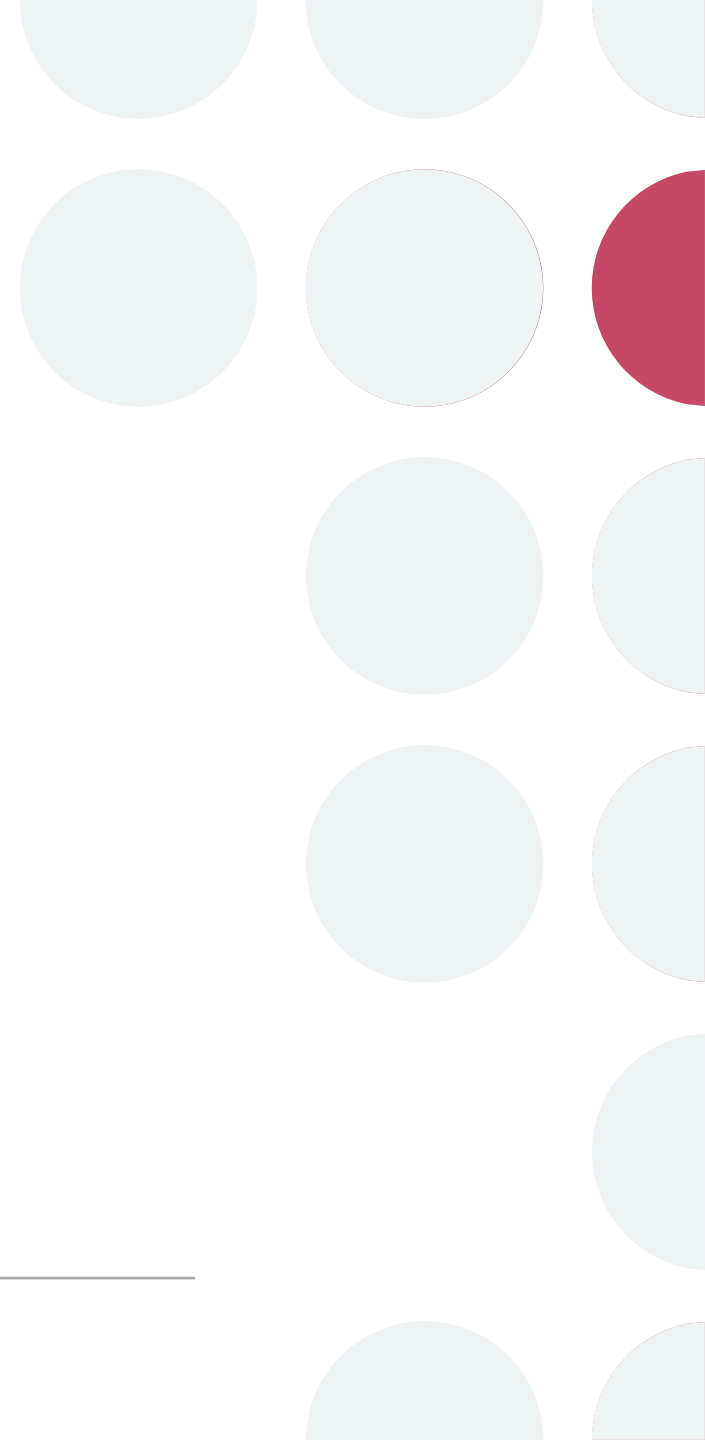
Nrp : 2110191017

Kelas : 3 D4 ITA

Listing Program Neural Network

```
class Node {  
    private float value;  
    private float weight;  
  
    Node(float weight){  
        this.weight = weight;  
    }  
  
    public float getValue() {  
        return value;  
    }  
  
    public void setValue(float value) {  
        this.value = value;  
    }  
  
    public float getWeight() {  
        return weight;  
    }  
  
    public void setWeight(float weight) {  
        this.weight = weight;  
    }  
    public float getWeightValue(){  
        return (this.weight * this.value);  
    }  
}
```

Pada class node ini kita membuat setter dan getter untuk value dan weightnya



```
public static void main(String[] args) {
    NeuralNetwork nn = new NeuralNetwork();
    nn.cumpute();
}
public class NeuralNetwork {

    public static final float YA = 2f;
    public static final float TIDAK = 1f;

    public static final float BIAS = 1f;
    public static final float MAX_NUM = 1f;
    public static final float MIN_NUM = -1f;

    public static final float LEARNING_RATE = 0.1f;

    List<Node> nodeList;
    List<Data> dataList;

    public NeuralNetwork() {
        nodeList = new ArrayList<>();
        dataList = new ArrayList<>();
        createDataset();
        initNode();
    }
}
```

Pada class NeuralNetwork kita menginisialisasi batas minimal ,maksimal dan learning ratenya. Lalu membuat array untuk nodelist dan datalist nya

```
public void createDataset() {  
    String line = "";  
    String splitBy = ",";  
    try {  
        BufferedReader br = new BufferedReader(new FileReader("C:\\heart.csv"));  
        while ((line = br.readLine()) != null) {  
            String[] datafile = line.split(splitBy);  
            List<Float> floatList = new ArrayList<>();  
            floatList.add(BIAS);  
            for (int i = 0; i < datafile.length - 1; i++) {  
                floatList.add(Float.parseFloat(datafile[i]));  
            }  
            float target = Float.parseFloat(datafile[datafile.length - 1]);  
            dataList.add(new Data(floatList, target));  
        }  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Disini kita membuat void createDataset untuk membuat dataSet nya setelah itu dilakukan pengecekan.

```
public float getSummationInput(List<Float> floatList) {  
    float sum = 0;  
    for (int i = 0; i < nodeList.size(); i++) {  
        nodeList.get(i).setValue(floatList.get(i));  
    }  
    for (Node node : nodeList) {  
        sum += node.getWeightValue();  
    }  
    return sum;  
}  
  
public boolean isUpdateWeight(float output, Data data) {  
    boolean isUpdate = false;  
    if (output != data.getOutput()) {  
        isUpdate = true;  
        float error = (float) (data.getOutput() - output);  
        for (Node node : nodeList) {  
            float weight = node.getWeight() + (LEARNING_RATE * node.getValue() * error);  
            node.setWeight(weight);  
        }  
    }  
    return isUpdate;  
}
```

Pada program ini kita membuat fungsi untuk memasukkan penjumlahan nodelist nya. Setelah itu kita juga membuat program untuk mengupdate weight nya

```
public void compute() {
    boolean next = true;
    int iteration = 0;
    float error = 0;
    while (next) {
        int counterNext = 0;
        for (int i = 0; i < dataList.size(); i++) {
            List<Float> floatList = dataList.get(i).getElementList();
            float sum = getSummationInput(floatList);
            float output = getOutput(sum);

            if (isUpdateWeight(output, dataList.get(i))) {
                counterNext++;
            }
        }
        iteration++;
        if (counterNext == 0 || (iteration >= 100)) {
            error = ((float) counterNext / (float) dataList.size() * 100);
            next = false;
        }
    }
    printWeight();
    System.out.println("DONE in " + iteration + " iteration");
    System.out.println("\nError: " + error + " %");
}
```

Disini kita juga membuat program untuk menghitung data list nya dengan cara looping. Lalu terakhir kita menampilkan berapa iterasi dan persentase errornya

Output

```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:C:\Program
BEST WEIGHT 0: 7.0370507
BEST WEIGHT 1: 7.6549163
BEST WEIGHT 2: 143.61389
BEST WEIGHT 3: 292.0518
BEST WEIGHT 4: 63.456104
BEST WEIGHT 5: 22.199049
BEST WEIGHT 6: -30.860212
BEST WEIGHT 7: 168.54385
BEST WEIGHT 8: -141.857
BEST WEIGHT 9: 136.60545
BEST WEIGHT 10: 291.10287
BEST WEIGHT 11: 167.04335
BEST WEIGHT 12: 347.21255
BEST WEIGHT 13: 774.06396
DONE in 100 iteration

Error: 30.370369 %
```
