```java
package teste;

import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.List;

import classes.MultilayerPerceptronClassificador;

/**
 * created by lgcaobianco on 2018-05-19
 */

public class TreinarMLP {
    public static Double[] ajustarValor(Double[][] resultadosObtidos) {
        Double[] resultadoAjustado = new Double[3];
        for (int i = 0; i < resultadosObtidos.length; i++) {
            if (resultadosObtidos[i][0] >= 0.5) {
                resultadoAjustado[i] = 1.0;
            } else {
                resultadoAjustado[i] = 0.0;
            }
        }
        return resultadoAjustado;
    }

    public static void main(String[] args) throws FileNotFoundException, UnsupportedEncodingException {
        MultilayerPerceptronClassificador mlp = new MultilayerPerceptronClassificador();
        double epsilon = Math.pow(10, -6);
        int contadorEpocas = 0;
        double erroAnterior = 10.0;
        double erroAtual = 0;
        PrintWriter writer = new PrintWriter("/home/lgcaobianco/repositorios/epc-rna/epc5/src/base/eqm.csv", "UTF-8");
        long inicioTreinamento = System.currentTimeMillis();
        while (Math.abs(erroAtual - erroAnterior) > epsilon) {

            for (int i = 0; i < mlp.getListaPontosEntrada().size(); i++) {
                mlp.forwardPropagation(i);
                mlp.backwardPropagation(i);
                mlp.forwardPropagation(i);
```

```
42              }
43
44              erroAnterior = erroAtual;
45              erroAtual = mlp.calcularEm();
46              writer.println(contadorEpocas + "," + Math.abs
    (erroAtual - erroAnterior));
47              contadorEpocas++;
48          }
49          long fimTreinamento = System.currentTimeMillis();
50
51          System.out.println("Tempo gasto: " + (
    fimTreinamento - inicioTreinamento));
52          mlp.setListaPontosEntrada(mlp.getMatrizOperacao())
    ;
53          System.out.println("Epocas: " + contadorEpocas);
54          System.out.println("tamanho do conjunto de
    operacao: " + mlp.getListaPontosEntrada().size());
55          int acertos = 0;
56          List<Double[]> resultadosEsperados = mlp.
    getClassificacaoMatrizOperacao();
57          List<Double[]> resultadosAjustados = new ArrayList
    <>();
58          writer.close();
59          writer = new PrintWriter("/home/lgcaobianco/
    repositorios/epc-rna/epc5/src/base/saida-obtida-fase-
    operacao.csv",
60                  "UTF-8");
61          for (int i = 0; i < mlp.getMatrizOperacao().size()
    ; i++) {
62              mlp.forwardPropagation(i);
63              resultadosAjustados.add(ajustarValor(mlp.getY2
    ()));
64              writer.print(resultadosAjustados.get(i)[0] +
    "," + resultadosAjustados.get(i)[1] + ","
65                      + resultadosAjustados.get(i)[2] + ",")
    ;
66              writer.print(mlp.getY2()[0][0] + "," + mlp.
    getY2()[1][0] + "," + mlp.getY2()[2][0]);
67              writer.print("\n");
68          }
69
70          for (int i = 0; i < resultadosAjustados.size(); i
    ++) {
71              if (resultadosAjustados.get(i)[0].equals(
    resultadosEsperados.get(i)[0])
72                      && resultadosAjustados.get(i)[1].
    equals(resultadosEsperados.get(i)[1])
73                      && resultadosAjustados.get(i)[2].
    equals(resultadosEsperados.get(i)[2])) {
74                  acertos++;
```

```
75                 }
76             }
77             writer.close();
78             System.out.println("A rede acertou: " + acertos);
79 
80         }
81 
82 }
83 
```