Grado en Ingeniería Informática en Sistemas de Información
Inteligencia Artificial - Curso 2016/17
EXAMEN JUNIO-6/06/2017

BLOQUE BÚSQUEDAS

Apartado A (2,5 puntos)

```
    public double getValue(Individual<Double> individual) {

        int m = datosX.size();
        int n = datosX.get(0).size();

        double suma = 0;

        for (int i = 0; i < m; i++) {
            double hi =  individual.getRepresentation().get(0);
            for (int j = 0; j < n; j++) {
                hi += individual.getRepresentation().get(j+1)
                        * datosX.get(i).get(j);
            }
            suma += Math.pow((hi - datosY.get(i)),2);
        }

        return (-1.0 / 2 * m) * suma;
    }

    @SuppressWarnings("unchecked")
    public boolean isGoalState(Object state) {
        return false;
    }

    public String getBoardForIndividual(Individual<Double> individual) {

        String s = "";
        for (Double indi : individual.getRepresentation()) {
            s += indi.toString() + " ";
        }

        return s;
    }

    public Individual<Double> generateRandomIndividual(int boardSize) {

        List<Double> individualRepresentation = new ArrayList<Double>();
        for (int i = 0; i < boardSize; i++) {
            individualRepresentation.add(this.getRandomNumber());
        }

        Individual<Double>                individual                =                new
Individual<Double>(individualRepresentation);

        return individual;
    }

    public Set<Double> getFiniteAlphabetForBoardOfSize(int size) {
        Set<Double> fab = new HashSet<Double>();

        for (int i = 0; i < size; i++) {
            fab.add(getRandomNumber());
        }

        return fab;
    }
```

Apartado B (1 punto)

1) En GeneticAlgorithm: //   protected List<A> finiteAlphabet; → no se necesita este atributo → modificar constructor en el método mutate:
mutatedRepresentation.set(mutateOffset, this.random.nextDouble());

```
en vez de:
//     mutatedRepresentation
//                .set(mutateOffset, finiteAlphabet.get(alphaOffset));

2) public class Junio2017GoalTest implements GoalTest {

       public boolean isGoalState (Object state) {
             return false;
       }
}
3) No
```

Apartado C (0,75 puntos)

```
    public static void junio2017GeneticAlgorithmSearch() throws IOException {

        cargarDatos();

        int tamanoIndividuo = datosX.get(0).size() + 1;

        System.out.println("\nDemo GeneticAlgorithm  -->");
        try {
            Junio2017FitnessFunctionRL fitnessFunction
                    = new Junio2017FitnessFunctionRL(datosX, datosY);
            // Generate an initial population
            Set<Individual<Double>>          population          =          new
HashSet<Individual<Double>>();
            for (int i = 0; i < 25; i++) {
                population.add(fitnessFunction
                        .generateRandomIndividual(tamanoIndividuo));
            }

            GeneticAlgorithm3<Double>            ga            =            new
GeneticAlgorithm3<>(tamanoIndividuo, 0.15);

            // Run for a set amount of time
            Individual<Double> bestIndividual = ga.geneticAlgorithm(
                    population, fitnessFunction, fitnessFunction, 10000L);

            System.out.println("Max Time (1 second) Best Individual=\n"
                    + fitnessFunction.getBoardForIndividual(bestIndividual));
            System.out.println("Fitness         = "
                    + fitnessFunction.getValue(bestIndividual));
            System.out.println("Population       Size       =       "       +
ga.getPopulationSize());
            System.out.println("Itertions       = " + ga.getIterations());
            System.out.println("Took            = "
                    + ga.getTimeInMilliseconds() + "ms.");

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

Apartado D (0,75 puntos)

```
newPopulation.add(retrieveBestIndividual(population, fitnessFn));
```

antes del bucle en el método nextGeneration de la clase GeneticAlgorithm

BLOQUE MACHINE LEARNING

Apartado A (1 punto)

```
...
Theta1SIN = Theta1(:,2:end);
Theta2SIN = Theta2(:,2:end);
J = J+(lambda/(2*m))*(sum(sum(Theta1SIN.^2))+sum(sum(Theta2SIN.^2)));

Theta1_grad(:,1) = (1/m) * DELTA1(:,1);
Theta2_grad(:,1) = (1/m) * DELTA2(:,1);
Theta1_grad(:,2:end)  =  (1/m)  *  DELTA1(:,2:end)  +  (lambda/m)  *
Theta1SIN;
Theta2_grad(:,2:end)  =  (1/m)  *  DELTA2(:,2:end)  +  (lambda/m)  *
Theta2SIN;
...
```

Apartado B (3,5 puntos)

```
...

% Load Training Data
load('spamTrain.mat');
Xtrain = X;
ytrain = y;
m = size(Xtrain, 1);

% Load Validation Data
load('spamValidation.mat');
Xval = X;
yval = y;

...

lambda = 0.001;
acierto = zeros(1,6);

for i=1:6

    ...

    % Obtain Theta1 and Theta2 back from nn_params
    Theta1(:,:,i)     =     reshape(nn_params(1:hidden_layer_size     *
(input_layer_size + 1)),hidden_layer_size, (input_layer_size + 1));

    Theta2(:,:,i)  =  reshape(nn_params((1  +  (hidden_layer_size  *
(input_layer_size + 1))):end),1,(hidden_layer_size + 1));


    % Predict the labels of the validation set
    pred = predict(Theta1(:,:,i), Theta2(:,:,i), Xval);

    % Compute the accuracy of the validation set
    acierto(i) = mean(double(pred == yval)) * 100;

    %Update the Lambda parameter
    lambda = lambda * 10;
end
```

```
%Compute the optimal lambda
[maximo index] = max(acierto);

load('spamTest.mat');

%Predict the test set
pred = predict(Theta1(:,:,index), Theta2(:,:,index), Xtest);

fprintf('\nTest Set Accuracy: %f\n', mean(double(pred == ytest)) *
100);
```

Apartado C (0,5 puntos)

Tasa de acierto (red neuronal sin regularizar): 98,6%
Tasa de acierto (red neuronal regularizada): 99,10%

Respuesta: Si, porque hay overfitting