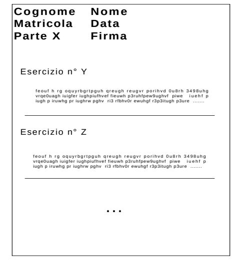
Soft Computing 19-07-2010



- Use only the answer sheets provided to you, write clearly on the top left hand corner (**like in the figure**) surname, name, enrolment number, date, part of the exam (Part 1 or Part 2) and signature
- Hand in the two parts in separate sheets
- Indicate the number of each exercise and separate each solution from each other clearly with a horizontal line
- Write CLEARLY by pen or pencil

COMMENT: few people followed the above reported instructions. From the next exam, 2 points will depend on the accomplishment of what mentioned above. In particular: all the information have to be written on the left part of each sheet, first the surname than the name, surname and name have to be written clearly (in particular they would usually be different from how you sign), all the requested data have to be included in the given order. Please remember also that your goal in writing your solution is to transmit information to your examiner, not just to fill the sheet in the shortest time, so, the more your writing is clear, the higher the probability that who is evaluating you can understand your message.

Part 1

1.1. Fuzzy Design [7/32]

Design a fuzzy system able to control the train stop in a metro station. On the train there are a speed sensor, and a sensor of distance from the place where the train head has to stop, and actuations as well, which are intensity of brake action and power to the train engine. The control system has to work in different load conditions, although there is no possibility to measure them directly.

Select the input and the output variables of the fuzzy system, define the corresponding fuzzy sets, **justifying** shape and position, and at least 3 rules for modeling the problem.

Solution

Many solutions are possible. We report here only one, with remarks about correct and incorrect choices.

The two input variables are SPEED and DISTANCE. The goal is to control the train stop in the station, so that the train stops at the correct place, which is a point. Errors up to few centimeters are acceptable, no more than 15. The ranges of the variables have to be compatible with the goal, say 0 – 200 meters for DISTANCE and 0 – 30 Km/h for SPEED. Since we aim at a good control, we do not expect to have negative SPEED or DISTANCE. Significant points for the membership functions for both the input variables are the max value and zero, which is the goal to be achieved. In particular, it is important that there is a membership function ZERO, having maximum value for 0, and the closer membership function starting at 0. If we put a trapezoid as ZERO MF, and the following MF starts at a point x (both in speed and distance) we may have situations where the only MF greater than zero is ZERO, and in this situation the controller will select the action suitable to keep the train blocked in 0, which is different from the one to reach that state. To maximize the comfort of passengers it is better to have triangles than trapezoids, so the actions change continuously and we avoid situations where no changes in actuations are alternating with (possibly strong) changes.

The output can be the only BRAKE action or also the POWER. A bad choice in this case would be to have incremental values since we aim at a final situation where the brake is active and the power is null, and with incremental values we cannot know whether we are in this state.

The shape of outputs can be singletons if the defuzzyfication method is center-of-mass, so that we can exploit all the values and we have a proportional control. With triangles or trapezoids we should select a different defuzzyfication method and also justify why should we have them (writing that so we favor rules with low fitness is not enough: why should we like to have this?) The number of output MF have to be consistent with the number of combinations of the input values. It makes no sense to have 25 possibilities in input (5x5 MFs) and only 3 values in output.

1.2. Reinforcement learning systems [2/32]

Briefly describe the Markov Hypothesis adopted in Reinforcement Learning systems, and provide its definition formula.

Solution

This is just a theory question whose answer can be found in the RL book or in the slides. The Markov property states that the probability to be at time $t+1$ in a state s_{t+1} depends only on the state where the agent was at time t (s_t) and on the action selected at time t (a_t), and not on the past history. In formulas:

$P(s_{t+1} = \bar{s} | s_t, a_t) = P(s_{t+1} = \bar{s} | H_{t+1})$,

where H_{t+1} is the sequence of past states and past actions.

1.3. Genetic algorithms [7/32]

Design a genetic algorithm that can produce an electronic circuit implementing a digital filter. Models of simple electronic components are available and the circuit can be represented as a set of connections among them. A device measures on a specific component (output) the quality of the filtered signal. The original signal to be filtered is injected in the circuit through another specific component (input). Input and output have to be present and to be unique in each circuit.

Briefly describe the structure of a genetic algorithm. Define an appropriate data structure for the above mentioned problem, define the corresponding genetic operators and the fitness function.

Solution

The most general solution considers a model similar to that of genetic programming (and, in fact, this problem has been faced by John Koza, the inventor of genetic programming). In this case, we have a graph of components, eventually connected by nodes representing parallel and series operators. With this model, we may adopt the standard crossover for genetic programming, where parents exchanges subgraphs, and mutation may change a component to another.

Other representations with the same power are representations where components are explicitly mentioned with type and connections. A representation with a (triangular) incidence matrix can be viable if the (limiting) assumption of knowing in advance all the components is accepted. A representation with a string of bits each indicating the presence or the absence of a component determined by its position in the string is too limiting (limited components, position given a priori, serial connection only).

The exercise required also to provide the description of the structure of a genetic algorithm, which has been forgotten by many.

Part 2

2.1 Perceptron [Score 4/32]

Consider the classical schema for the single perceptron with two inputs:

- draw it and write its output characteristic
- define the general formula for the weights update
- execute one epoch of training for the perceptron using the NAND (Not AND) function starting from the init conditions w0=0, w1=1/2, w2=1/2, and using a learning rate of 0.2;
- is a single perceptron able to learn the NAND function? Explain you answer.

2.2 Radial Basis Functions [Score 4/32]

Describe, eventually using formulas and graphs, the main differences between Radial Basis Functions and Feed-Forward Neural Networks in terms of: network topology, analytical formula of the output, learning algorithm, overfitting sensitivity.

2.3 Bayesian Classifier [Score 8/32]

Lets consider the generic Bayesian Classifier; provide an answer for the following questions:

- What is the theoretical principle under the Bayesian Classifier? How does it work?
- What are the differences between the Joint Bayes Classifier and the Naive Bayes Classifier implementations?
- Derive the Naive Bayes classifier from the dataset in the table (using Laplace estimator for discrete variables) and classify the record [Rainy,Yes,???]
- What are "Laplace estimator" and "m-estimator" used in the Naive Bayes Classifier?
- What is the relationship between Bayesian Networks and Bayes Classifier?

Weather	***Parents***	***Money***	***Decision***
Sunny	Yes	23	Cinema
Sunny	No	18	Tennis
Windy	Yes	20	Cinema
Rainy	Yes	14	Cinema
Rainy	No	13	Cinema
Rainy	Yes	10	Cinema
Windy	No	12	Cinema
Windy	No	19	Tennis
Windy	Yes	20	Cinema
Sunny	No	18	Tennis

Table 1: The weekend dataset