Family Name: ETIENNE First Name: Alain

KIMP_CII UE2 and GTL_S43-b
Final Test

Arts et Métiers — Alain ETIENNE

Day : **1st December 2023**

Duration : **1.5h**

Document allowed : **None**

Complete when possible directly on this document and give it back at the end of this exam. Don't forget to put your name on the top of this document. You can complete the exercice in the order you want: they are independant.

Exercise 1: Fuzzy Logic

The goal of this exercise is to calculate the final mark of a course composed by two projects and one final test. The fuzzy logic engine considers then three input parameters: Project1, Project2 and Final_Exam. These ones are defined graphically in Figure1. The output parameter, the final mark is graphically described in Figure 2 as well.

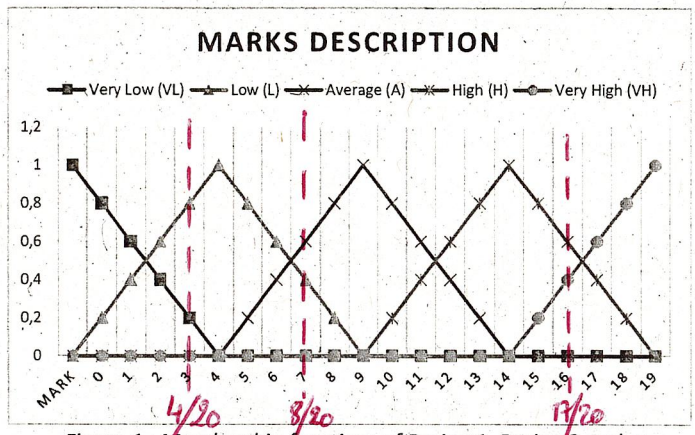


Figure 1 : Membership functions of Project1, Project2 and Final_Exam (inputs)

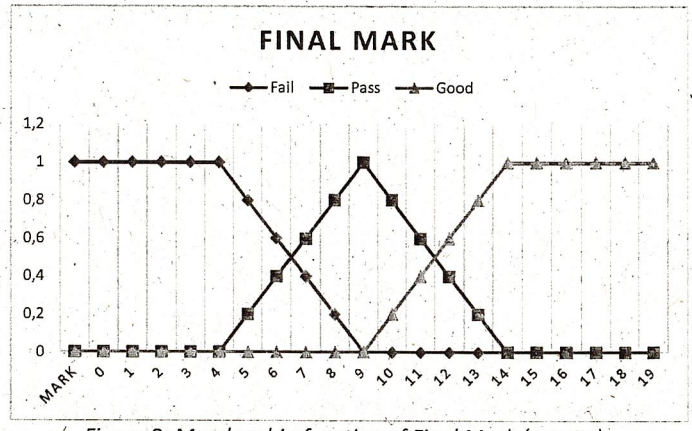


Figure 2: Membership function of Final Mark (output)

An intermediate fuzzy calculous is proposed to calculate the fuzzy average mark of the two projects with the rules described with the matrix displayed on Figure 3. It is important to underline that OR operator is used for this step (the upper left cell of Figure 3 means: "*if Project1 is VL **OR** Project2 is VL then Project average is VU*").

OR	Project 1: VL	L	A	H	VH
Project 2: VL	VU	VU	U	U	A
L	VU	U	U	A	A
A	U	U	A	S	S
H	U	A	S	S	VS
VH	A	S	S	VS	VS

Figure 3: Rules to assess project average mark

Aggregation: **Sum**

AND	Final Exam: VL	L	A	H	VH
Project average: VU	fail	fail	fail	pass	pass
U	fail	fail	pass	pass	pass
A	fail	pass	pass	pass	good
S	pass	pass	pass	good	good
VS	pass	pass	good	good	good

Figure 4: Rules to calculate the final mark

Aggregation: **Maximum**
Defuzzication: **Mean of Maximum**

The results are then not defuzzycated since they are directly integrated in the second calculous step. This second step aims at calculating the final mark, by using the same matrix form proposed in Figure 4 and as inputs: the mark of the final test and the average mark of projects. In that case, AND operator is used (the upper left cell of Figure 4 means: "*if Project average is VU **AND** Final Exam is VL then final mark is fail*"). For each step, their aggregation and defuzzycation operators are specified.

Work to complete: By detailing and justifying each step of your reasoning on the proposed areas below, give the final mark of a student having these three intermediate results: **17/20 for the first project, 4/20 for the second project and 8/20 for the final exam (take care of the offset on x-axis in Figures 1 and 2 that is due to Excel display).**

Fuzzycation Step – Give the membership of all the three input parameters

Project1: 17/20 → High : 0.6 Very High : 0.4
Project2: 4/20 → Very low : 0.2 Low : 0.8
Final_Exam: 8/20 → Low : 0.4 Average : 0.6

Inference of the first step – Put below all outputs being not null after the inference and propose their final value after aggregation. You can write on Figure 3.

See figure 3. (16 rules are not null)

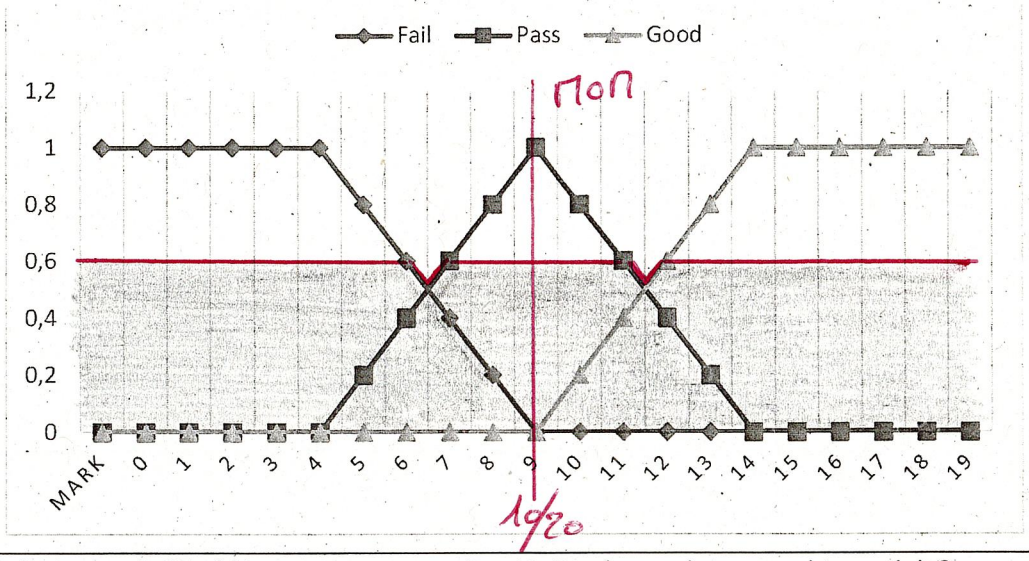
Outputs : VU = sum(0.2; 0.2; 0.8) = 1.2
U = sum(0.2, 0.6, 0.8, 0.8) = 2.4
A = sum(0.8, 0.8, 0.4) = 2.0
S = sum(0.6, 0.4, 0.6) = 1.6
VS = sum(0.4, 0.6, 0.4) = 1.4

Inference of the second step – Put below all outputs being not null after the inference. You can write on Figure 4.

See figure 4;

fail = max(0.4, 0.6, 0.4) = 0.6
pass = max(0.4, 0.6, 0.4, 0.6, 0.4, 0.6) = 0.6
good = max(0.6) = 0.6

Aggregation of the second step – Draw directly the output shape on this document (clearly and cleanly)



To pass a module, its mark should be greater or equal to 10. Do the student pass this module?

Defuzzycation – Calculate the final mark of the student and conclude if he will pass or not.

MoM = 10 → the student pass his exam.

Exercise 2: Constraint Satisfaction Problem: Operator allocation on an assembly line

A company wants to allocate wisely operators on an automatic assembly line. These operators' tasks are mainly monitoring and logistic (feed the automatic systems with parts). We want to take into consideration human factors in this allocation to prevent some people of walking too much a day. The assembly line is composed by 14 workstations. To define the factory, two main parameters are defined:

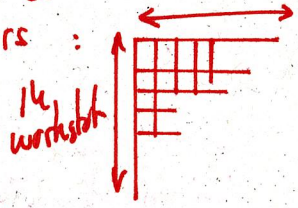
- $WorkTime_j$ is a list giving the work time of the j^{th} workstation
- $WalkTime_{i,j}$ is a matrix giving the time needed to walk from the i^{th} workstation to j^{th} workstation.

Several hypotheses are proposed to model this problem:

① - Only one operator on a workstation
② - No free workstation (all workstations must be allocated)
③ - An operator can work on several workstations
④ - An allocation is considered as valid when all workers can do their jobs and movements in the takt time, which is an input parameter (Takt_Time).
⑤ - The amount of operator (Nb_Op) is an input parameter.

Question 2.1: Model the problem by defining: the variable(s), domain(s) and the constraint(s). You may need to introduce extra variables. Please justify your decisions clearly. You may express your constraints either mathematically or in words, but they must be expressed in terms of relationships between the variables in the problem. You should explain how each of the points above is captured by your formulation (please answer in point form).

Several solutions are possible to model this problem.
For instance, consider a matrix allocation a 2 dimensions matrix where the line model the workstation and the columns, the operators : Nb_op columns. ⑤



→ the domain is {0, 1} (the operator is allocated for this workstation or not)

Constraints :

- $\forall w, \sum_{op=1}^{Nb\_op} allocation(w, op) == 1$ ① ②
- $\forall op, \sum_{w=1}^{14} allocation(w, op) > 0$ ③ + Not lazy operators.
- $\forall op, \sum_{w=1}^{14} allocation(w, op) \cdot worktime(w) + \frac{1}{2} \sum_{i=1}^{13} \sum_{j=i}^{14} walktime(i,j) \leq Takt\ time$ where $allocation(i, op) + allocation(j, op) == 2$

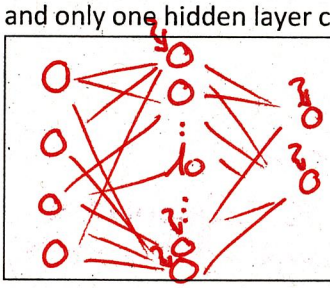
Question 2.2: We want to balance the work time among all operators. Explain and justify how to manage this new requirement on your model

The goal is to minimize the difference between the worker working the more and the one working the less.

Exercise 3: Artificial Neural Network

We want to design and train a regression neural network.

Question 3.1: Assess the number of parameters of this regression network having 4 input parameters, two outputs and only one hidden layer composed by 10 neurons. Detail your calculous.



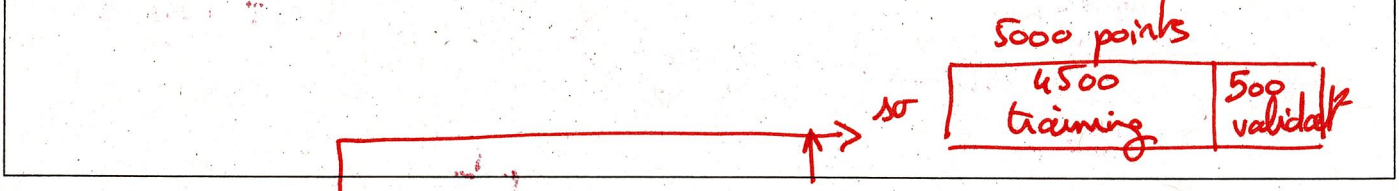
Nb parameters = Nb_bias + Nb weights

$= (10+2) + 10.4 + 10.2$

$= 72$

Question 3.2: Regarding the network type of the previous question, cross out the Tensorflow.keras functions that certainly won't be useful: ~~sigmoid,~~ linear, relu, ~~binary_crossentropy~~, mean_squared_error, accuracy, mean_absolute_error, ~~tanh~~, ~~confusion_matrix~~. For the one(s) you crossed, explain why:

- sigmoid and tanh are dedicated for last layer of classificat° ANN.
- binary_crossentropy is a loss functio for binary classificat° networks
- confusion_matrix is a metric to analyse the type of misclassificat°



Question 3.3: The network is configured to be trained on a set of 5000 training points with 200 epochs. We define a validation_split equals to 0.1. Complete the table below with different gradient descent strategies, by giving and justifying the number of updates (backpropagations) that the network will go through.

Stochastic descent	Batch descent	Mini batch of 100 points
$= 4500 . 200$ $= 900\,000$ updates	$= 1 . 200$ $= 200$ updates	$= int\left(\frac{4500}{100}\right) . 200$ $= 45 . 200$ $= 9000$ updates

Question 3.4: The figures (Figure 5 and Figure 6) show two different configurations of the training of the network. For each case, explain what could be the reason of this evolution of the loss and explain briefly how to reduce this issue.

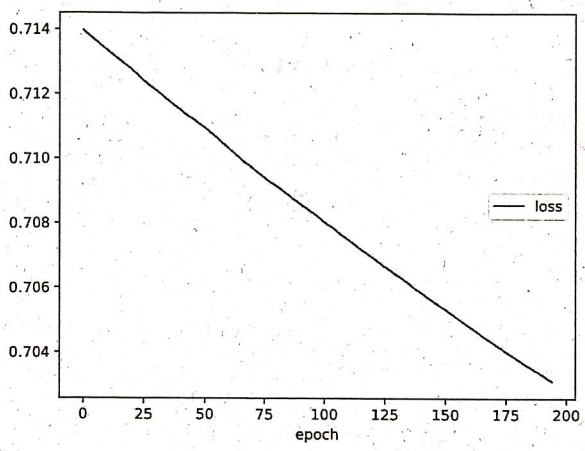


Figure 5: Configuration 1

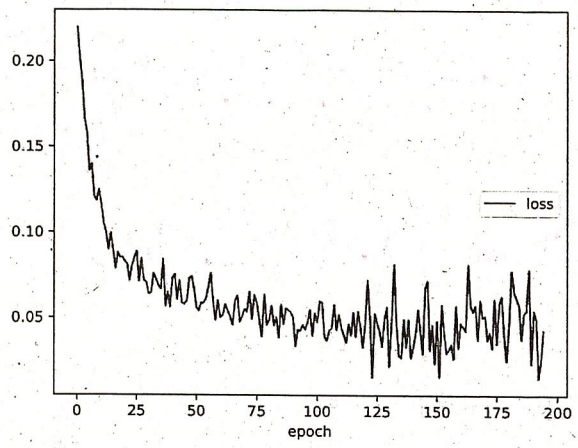


Figure 6: Configuration_2

Reason + Solution:

The loss evolves very slowly. The main parameter to play with is the learning rate (λ).

Reason + Solution:

The evolut of the loss function is noisy but not chaotic. The parameter to futhy play with is the size of the batch, after reducing a little λ, and finaly to have a look at the populat (training dataset).

Exercise 4: Expert Systems

The rule base of an order 0 expert system is composed by this set of rules:

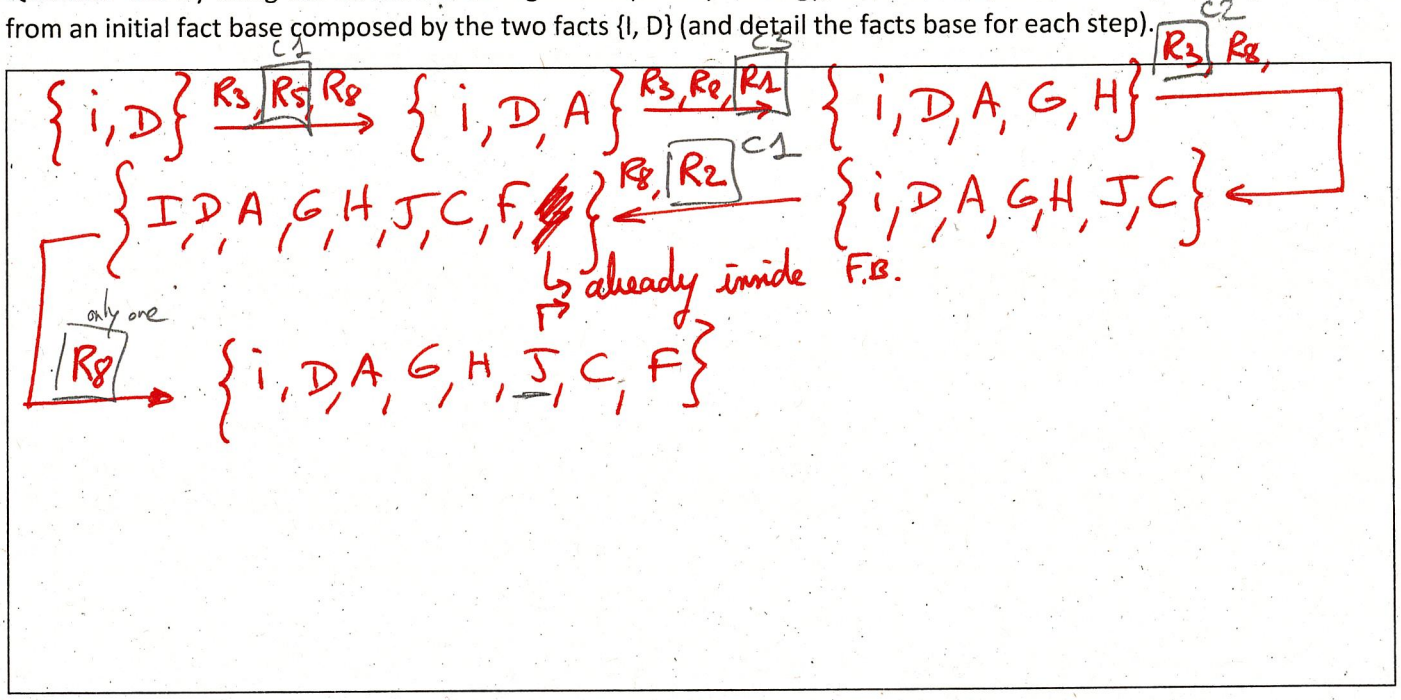
		Nb of conditions	nb of conclusions
R1	**IF** A **THEN** G AND H	1	2
R2	**IF** C AND (B OR A) **THEN** F AND G	3	2
R3	**IF** D **THEN** J AND C	1	2
R4	**IF** D AND E **THEN** B	2	1
R5	**IF** D AND I **THEN** A	2	1
R6	**IF** K **THEN** E	1	1
R7	**IF** L **THEN** D AND I	1	2
R8	**IF** D **THEN** J	1	1

Figure 7: Rules base

The selection step of the expert system follows these meta-rules:

- **P1**: Priority to the rule having the most numerous **conditions**,
- **P2**: If the first criterion is not enough, priority to the rule having the most numerous **conclusions**,
- **P3**: If the two first criteria are not enough; the priority is given regarding the **order** of the rule in the rule base (the number in the first column in Figure).

Question 4.1: By using the forward chaining in this priority strategy, detail all steps followed by the expert system from an initial fact base composed by the two facts {I, D} (and detail the facts base for each step).



Question 4.2: By applying the backward chaining and the priority strategy previously defined, determine if it is possible to prove G, having as available facts L in the fact database. Justify by showing the detailed explored tree.