ANFIS Architecture

① Output of Layer 1 is $\mu_A(x)$

- ① Train the premise parameters in this layer
- ② Training done by applying the membership function calculation with inputs (x) supplied, and premise parameters being trainable
- ③ Place the $\mu_A(x)$ in a tensor to be pass to Layer 2

ⓐ equivalent to the self.rul in the code

ⓑ There are 12 linguistic labels, therefore they need to be 12 $\mu_A(x)$ output for each input samples

ⓒ Therefore in layer 1 is where 12 BELL MEMBERSHIP FUNCTION with different premise parameters are specified

ⓓ output tensor shape = [input size, 12]

② output of Layer 2 is min [$\mu_A(x)$] for 81 rules for each input samples, and this is because the rules uses CONJUNCTIVE EVIDENCES. That is, the firing strength for each of the 81 rule for each input sample.

ⓐ In other words, when AND is used then use the LOWEST evidence CF to multiply with the rule CF

ⓑ Therefore, the firing strength of a rule is the same as the LOWEST evidence CF

ⓒ This layer needs the same number of rules to create simple function that takes the min of a set of 4 $\mu_A(x)$. There are 81 rules, therefore 81 minimization functions with different sets of 4 $\mu_A(x)$

ⓓ output tensor shape = [input size, 81]

③ Output of Layer 3 is normalized firing strengths that is calculated by calculating the ratio of the ith rules' firing strength to the sum of all rules' firing strengths

ⓐ For each input sample, sum across its firing strengths for all 81 rules then take each ith rules' firing strength to divide over the summation

ⓑ output tensor shape = [input size, 81]

Appendix to Layer 1

① Inputs are: ⓐ Seniority [normed to 0-1] supplied to its 3 linguistic labels

ⓑ Purchase_Propensity [normed to 0-1] supplied to its 3 linguistic labels

ⓒ Company Size: [normed to 0-1] supplied to its 3 linguistic labels

ⓓ Contactable: [normed to 0-1] supplied to its 3 linguistic labels

ANFIS Architecture

(4) For each input sample and each of the 81 rules:

(a) Required inputs : Its normalized firing strength
Seniority
Purchase - Propensity
Company Size
Contactable
~~Bias term [values of 1]~~

(b) Trainable consequent parameters : p = coefficient of Seniority for one particular rule
q = coefficient of Purchase for one particular rule
r = coefficient of Company size for one particular rule
s = coefficient of Contactable for one particular rule
t = bias of one particular rule

(c) Apply this equation - this rule's normalized firing strength multiply with

$$(p \times Seniority + q \times Purchase\_Propensity + r \times CompanySize + s \times Contactable + t)$$

$\boxed{\text{CONSEQUENT PARAMETERS TRAINED BY NN}}$

(d) Output tensor shape = [Input size, 81]

$\bar{w}_i f_i$, output of the equation for each of the 81 rules

(5) Just take for each input sample, the summation of 81 outputs of the equation in Layer 4

(a) Output tensor shape = [Input size, 1]

this will be a float that can be positive or negative

(6) Layer to get ~~the~~ one of the 3 Action linguistic labels i.e; [Discard, Brochure, Email]

(a) Input tensor shape = [Input size, 1]

(b) Send this input to a softmax activation function with 3 classes

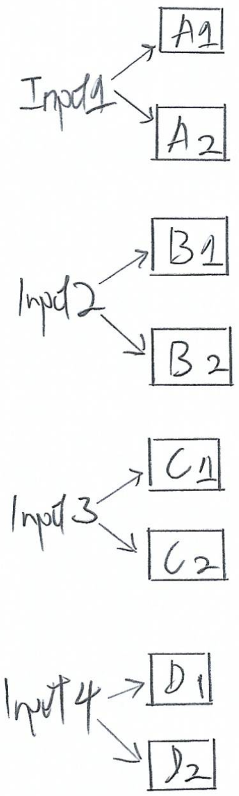
(c) Output of the activation tensor shape = [Input size, 3]

3 probability for each of the class that sum to 1

(d) A function that will select the class with the highest probability will output the result as prediction of the NN

(e) ~~exe~~ Print text "Predicted outcome = ① Discard ② Brochure ③ Email"

Layer 1



Layer 5

Layer 6

Layer 7

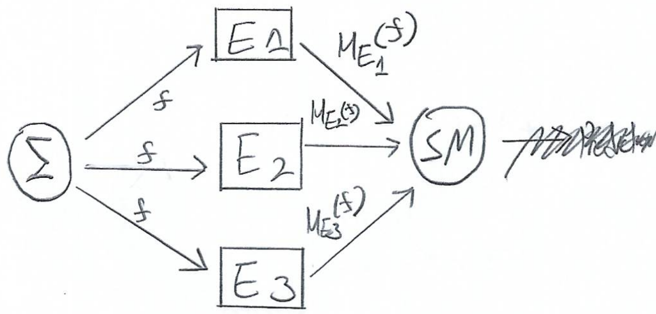


Table of consequent parameter for membership

E	a	b	c
Discord			
Broader			
Email			

∴ There are 3 linguistic labels from the output value of f.
Total number of consequent parameter $= 3 \times 3 = 9$

∴ SM = tf.nn.sparse_softmax_cross_entropy_with_logits

* output of Layer 6 is of shape = [Input size, 3] i.e., logits
* output of Layer 7 is of shape = ~~[Input size, 3]~~

~~There need to be further processing that turns the probability into class label~~

[Input size,] type = float32

* The logits of Layer 6 is of shape = [Input size, 3]
For each input sample, the $M_E(f)$ with the highest value is the predicted actions. Get the index of the columns to match the target labeling

Inputs

① Seniority : 0, 0.11, 0.22, 0.33, 0.44, 0.55, 0.66, 0.77, 0.88, 1 [10 unique values]

② Purchase-Propensity : 0, 0.5, 1 [3 unique values]

③ Company Size : [Many unique values]

④ Contactable : 0, 0.5, 1 [3 unique values]

---

⑤ OUTPUT - Action : 0, 1, 2 [3 categorical actions]

Re-formed Inputs

① Rank your Seniority from 1 to 10 on the scale below [Norm to the scale from 0 to 1]

1 2 3 4 5 6 7 8 9 10

Junior — Senior

② Rank your intention to purchase our product from 1 to 10 on the scale below [Norm to the scale from 0 to 1]

1 2 3 4 5 6 7 8 9 10

No Plan — Within the next 12 months

③ ok as it is, [Norm to the scale from 0 to 1]

④ Rank your willingness to be contacted ~~by our~~ for promotion from 1 to 10 on the scale below [Norm to the scale from 0 to 1

1 2 3 4 5 6 7 8 9 10

Don't contact — Email or Call me

Inputs – Fuzzy Set

① Seniority : Junior , Mid , Senior [3 linguistic labels]

② Purchase_Propensity : No plan , Within the next 3 years , With the next 12 months [3 linguistic labels]

③ Company Size : 0-100 (Small) , 101-500 (Medium) , >501 (Big) [3 linguistic labels]

④ Contactable : Don't contact , Send Content Only , Email or Call me [3 linguistic labels]

---

⑤ OUTPUT - Action : Discard , Send Brochure , To call [3 linguistic labels]

Rules

① Number of rules = 3 × 3 × 3 × 3 = 81 rules

Parameters

① BELL MEMBERSHIP FUNCTION $\mu_A(x) = \frac{1}{1 + [(\frac{x-c}{a})^2]^b}$ [3 parameters : a, b, c]

ⓐ Total number of fitting parameters = 36 + 405 = 441

ⓑ Premise parameters = 12 × 3 = 36

ⓒ Consequent parameters = 81 × 5 = 405

Table of premise parameters [BELL MEMBERSHIP FUNCTION]

A	a	b	c
Junior			
Mid			
Senior			
No plan			
Within 3 years			
Within 12 months			
Small			
Medium			
Big			
Don't contact			
Send content only			
Email or call me			

∴ There are 12 linguistic labels from the 4 inputs

Total number of premise parameters $= 12 \times 3 = 36$

Certainty Factor

CF = +1.0 Rule is certainly true

CF = 0.0 Don't know whether rule is true or not

CF = -1.0 Rule is certainly false

① CONJUNCTIVE EVIDENCES

⊛ For rules with conjunctive evidences the certainty of the hypothesis H is :

$$cf(H, E_1 \cap E_2 \cap \dots \cap E_n) = \min[cf(E_1), cf(E_2), \dots, cf(E_n)] \times cf$$

All the $\mu_A(x)$ specified in a rule

$$cf = f_i = (p_i x + q_i y + r_i)$$

★ Examples:

IF earnings = good AND contract = big THEN shares = up [cf 0.9]

current certainty of earnings = good is 0.8, and contract = big is 0.1 then

$$cf(H, E_1 \cap E_2) = \min[0.8, 0.1] \times 0.9 = 0.1 \times 0.9 = 0.09$$

This result can be interpreted as "it is unknown if shares will go up"

Softmax Activation Function

1. A multi-class classifier that solves the problem of assigning an instance to one class when the number of possible classes is larger than two.
2. Softmax output across the classes will always sum to 1, and this is suited to the feature of Fuzzy Rules that the differed linguistic labels be a possibility value that will sum to 1.
3. Turn NN score into values that can be interpreted by humans
4. Problem: softmax needs its input to be the same size as the number of classes
5. Loss function for softmax = categorical_crossentropy
6. optimizer for softmax = rmsprop / adam
7. metrics for softmax = accuracy

Tensorflow

1. tf.nn.softmax (logits, axis=None, name=None)
2. tf.train.AdamOptimizer