**ECS759P Artificial Intelligence**

**Coursework 2 - Marking Guideline**

**Part 1**

**Task 1 [10 points].** Express Madame Irma’s six statements into First Order Logic (FOL). Note: You can use two constants: YOU and ROBIN.

***Answer***: The FOL expression corresponding to the six statements are:

1. ∃x Dog(x) ∧ Own(YOU,x)

2. Buy(ROBIN)

3. ∀x∀y (Own(x,y)∧Rabbit(y)) ⇒ (∀z ∀w (Rabbit(w)∧Chase(z,w)) ⇒ Hate(x,z))

4. ∀x Dog(x) ⇒ ∃y (Rabbit(y) ∧ Chase(x,y))

5. ∀x Buy(x) ⇒ ∃y (Own(x,y) ∧ (Rabbit(y) ∨ Grocery(y)))

6. ∀x ∀y ∀z(Own(y,z) ∧ Hate(x,z)) ⇒ ¬Date(x,y)

***Marking scheme:***

*2 points for each FOL statement except for 1) and 2) which are 1 point each.*

*A simpler alternative is possible since "chasing" only appears as "chasing some rabbit" in our problem, 3 and 4 can be written as:*

*3. ∀x ∀y (Own(x,y) ∧ Rabbit(y)) ⇒ (∀z ChaseSomeRabbit(z) ⇒ Hate(x,z))*

*4. ∀x Dog(x) ⇒ ChaseSomeRabit(x).*

*There are alternative ways of expressing 3 and 6 as well – these follow a general equivalence relation in FOL, which is ∀x ∀y P(x,y) ⇒ Q(x) is equivalent to ∀x (∃y P(x,y) ⇒ Q(x)).*

*So 3 can be equivalently written as:*

*∀x (∃y (Own(x,y) ∧ Rabbit(y)) ⇒ (∀z (∃w (Rabbit(w) ∧ Chase(z,w)) ⇒ Hate(x,z))))*

*The parentheses in this case are important to delineate the scopes clearly. Similarly, an equivalent way to express 6 in FOL is:*

*∀x ∀y (∃z (Own(y,z) ∧ Hate(x,z)) ⇒ ¬Date(x,y)).*

**Task 2 [10 points].** Translate the obtained expressions to Conjunctive Normal Forms (CNFs, Steps 1-6 of Lecture 9: Logic). Show and explain your work.

***Answer***:

1. **Remove implications:**

7. ∀x∀y(¬(Own(x,y) ∧ Rabbit(y)))∨(∀z∀w¬(Rabbit(w) ∧ Chase(z,w)) ∨ Hate(x,z)) (from 3)

8. ∀x¬Dog(x)∨∃y(Rabbit(y) ∧ Chase(x,y)) (from 4)

9. ∀x¬Buy(x)∨∃y(Own(x,y) ∧ (Rabbit(y) ∨ Grocery(y))) (from 5)

10. ∀x ∀y ∀z(¬(Own(y,z) ∧ Hate(x,z))) ∨ ¬Date(x,y) (from 6)

1. **Minimise negations:**

11. ∀x∀y(¬Own(x,y) ∨ ¬Rabbit(y)) ∨ (∀z∀w (¬Rabbit(w) ∨ ¬Chase(z,w)) ∨ Hate(x,z)) (from 7)

12. ∀x ∀y ∀z(¬Own(y,z) **∨** ¬Hate(x,z)) ∨ ¬Date(x,y) (from 10)

1. **Standardise variables apart:**

13. ∀x1∀y1(¬Own(x1,y1) ∨ ¬Rabbit(y1)) ∨ (∀z1∀w1 (¬Rabbit(w1) ∨ ¬Chase(z1,w1)) ∨ Hate(x1,z1)) (from 11)

14. ∀x2¬Dog(x2)∨∃y2(Rabbit(y2) ∧ Chase(x2,y2)) (from 8)

15. ∀x3¬Buy(x3)∨∃y3(Own(x3,y3) ∧ (Rabbit(y3) ∨ Grocery(y3))) (from 9)

16. ∀x4∀y4∀z2(¬Own(y4,z2) ∨ ¬Hate(x4,z2)) ∨ ¬Date(x4,y4) (from 12)

1. **Skolemise existentials:**

17. Dog(D) ∧ Own(YOU,D) (from 1)

18. ∀x2¬Dog(x2)∨(Rabbit(R(x2)) ∧ Chase(x2,R(x2)) (from 14)

**Note**: the existential on y2 was in the scope of the universal on x, hence the need for introduction of the function.

19. ∀x3¬Buy(x3) ∨ (Own(x3,F(x3)) ∧ (Rabbit(F(x3)) ∨ Grocery(F(x3)))) (from 15)

**Note**: the existential on y3 was in the scope of the universal on x, hence the need for introduction of the function.

1. **Drop universals:**

20. ¬Own(x1,y1) ∨ ¬Rabbit(y1) ∨ (¬Rabbit(w1) ∨ ¬Chase(z1,w1) ∨ Hate(x1,z1)) (from 13)

21. ¬Dog(x2) ∨ (Rabbit(R(x2)) ∧ Chase(x2,R(x2)) (from 18)

22. ¬Buy(x3) ∨ (Own(x3,F(x3)) ∧ (Rabbit(F(x3)) ∨ Grocery(F(x3)))) (from 19)

23. ¬Own(y4,z2) ∨ ¬Hate(x4,z2) ∨ ¬Date(x4,y4) (from 16)

1. **Convert to CNF:**

24. [¬Dog(x2) ∨ Rabbit(R(x2))] ∧ [¬Dog(x2) ∨ Chase(x2,R(x2))] (from 21)

25. [¬Buy(x3) ∨ Own(x3,F(x3))] ∧ [¬Buy(x3) ∨ Rabbit(F(x3)) ∨ Grocery(F(x3))] (from 22)

**Final set:**

26. Dog(D) ∧ Own(YOU,D)

27. Buy(ROBIN)

28. ¬Own(x1,y1) ∨ ¬Rabbit(y1) ∨ ¬Rabbit(w1) ∨ ¬Chase(z1,w1) ∨ Hate(x1,z1)

29. [¬Dog(x2) ∨ Rabbit(R(x2))] ∧ [¬Dog(x2) ∨ Chase(x2,R(x2))]

30. [¬Buy(x3) ∨ Own(x3,F(x3))] ∧ [¬Buy(x3) ∨ Rabbit(F(x3)) ∨ Grocery(F(x3))]

31. ¬Own(y4,z2) ∨ ¬Hate(x4,z2) ∨ ¬Date(x4,y4)

***Marking scheme:*** *2 points for each CNF step, except for steps 5 and 6 which are 1 point each.*

**Task 3 [10 points].** Transform Madame Irma’s conclusion into FOL, negate it and convert it to CNF (Steps 1-6 of Lecture 9: Logic). Show and explain your work.

***Answer***: (¬∃x (Grocery(x) ∧ Own(ROBIN,x))) ⇒ ¬Date(ROBIN,YOU)

1. **Remove implications:**

Note that the statement is itself a conditional. Recall that P ⇒ Q is equivalent to ¬P ∨ Q. Its negation will be ¬(¬P ∨ Q) which is P ∧ ¬Q. So in short, negation of P ⇒ Q is P ∧ ¬Q. Hence, negation of Madam Irma’s statement will be:

(¬∃x (Grocery(x) ∧ Own(ROBIN,x))) ∧ Date(ROBIN,YOU)

1. **Minimise negations:**

1. (∀x ¬(Grocery(x) ∧ Own(ROBIN,x))) ∧ Date(ROBIN,YOU)

2. (∀x ¬Grocery(x) ∨ ¬Own(ROBIN,x)) ∧ Date(ROBIN,YOU)

1. **Standardise variables apart:** Nothing to do
2. **Skolemise existentials:** Nothing to do
3. **Drop universals:**  ¬Grocery(x) ∨ ¬Own(ROBIN,x) ∧ Date(ROBIN,YOU)
4. **Convert to CNF:** [¬Grocery(x) ∨ ¬Own(ROBIN,x)] ∧ [Date(ROBIN,YOU)]

***Marking scheme:*** *4 points for FOL, 1 point per CNF step.*

**Task 4 [20 points].** Based on all the previously created clauses (you should have at least 7 depending on how you split them), finalise the conversion to CNF (Steps 7-8 of Lecture 9) and provide proof by resolution that Madame Irma is right that you should go to see Robin to declare to her your (logic) love. Show and explain your work, provide unifiers.

***Answer***: Our knowledge base is composed of CNFs (26) to (31) from Task 2, to which we add the negation of the conclusion, and sequentially use resolution steps to see if we reach a contradiction. We have split the CNFs so that each is a single clause involving only disjunctions and standardised variables apart (steps 7-8 described in the Lecture 9 slides):

1. Dog(D)

2. Own(YOU,D)

3. Buy(ROBIN)

4. ¬Own(x1,y1) ∨ ¬Rabbit(y1) ∨ ¬Rabbit(w1) ∨ ¬Chase(z1,w1) ∨ Hate(x1,z1)

5. ¬Dog(x2) ∨ Rabbit(R(x2))

6. ¬Dog(x3) ∨ Chase(x3,R(x3))

7. ¬Buy(x4) ∨ Own(x4,F(x4))

8.¬Buy(x5) ∨ Rabbit(F(x5)) ∨ Grocery(F(x5))

9. ¬Own(y2,z2) ∨ ¬Hate(x6,z2) ∨ ¬Date(x6,y2)

10. ¬Grocery(x7) ∨ ¬Own(ROBIN,x7)

11. Date(ROBIN,YOU)

The resolution proof does not have a unique sequence of steps. Here is a possible sequence.

12. ¬Own(YOU,z2) ∨ ¬Hate(ROBIN,z2) (Resolve 9 and 11; unifier {ROBIN/x6, YOU/y2})

13. ¬Hate(ROBIN,D) (Resolve 12 and 2; unifier {D/z2})

14. ¬Own(ROBIN,y1) ∨ ¬Rabbit(y1) ∨ ¬Rabbit(w1) ∨ ¬Chase(D,w1) (Resolve 4 and 13; unifier {ROBIN/x1, D/z1})

15. Own(ROBIN,F(ROBIN)) (Resolve 7 and 3; unifier {ROBIN/x4})

16. ¬Rabbit(F(ROBIN)) ∨ ¬Rabbit(w1) ∨ ¬Chase(D,w1) (Resolve 14 and 15; unifier { F(ROBIN)/y1})

17. Chase(D,R(D)) (Resolve 6 and 1; unifier {D/x3})

18. ¬Rabbit(F(ROBIN)) ∨ ¬Rabbit(R(D)) (Resolve 16 and 17; unifier {R(D)/w1})

19. Rabbit(R(D)) (Resolve 5 and 1; unifier {D/x2})

20. ¬Rabbit(F(ROBIN)) (Resolve 18 and 19)

21. Rabbit(F(ROBIN)) ∨ Grocery(F(ROBIN)) (Resolve 8 and 3; unifier {ROBIN/x5})

22. ¬Grocery(F(ROBIN)) (Resolve 10 and 15; unifier {F(ROBIN)/x7})

23. Rabbit(F(ROBIN)) (Resolve 21 and 22)

24. ∅ (Resolve 20 and 23)

***Marking scheme:*** *0.8 point per each correct clause in the knowledge base and per each resolution step capped at 20 in total. 0.8 point for specifying unifiers.*

**Part 2.**

**Task 1 [10 points].** Given the problem, what is the most appropriate loss function to use? Provide the name of the loss, its formula and the formula interpretation in your report.

***Answer***: Cross-entropy. The cross-entropy of the true distribution P with respect to the predicted distribution Q measures how many additional bits you need on average to represent an event using Q instead of P:

***Marking scheme:*** *4 for naming cross-entropy, 3 for formula, 3 for its interpretation.*

***Remark****: Students may provide other loss functions (such as Kullback-Leibler divergence, hinge loss, etc.). They are valid as soon as their relevance to the problem is explained.*

**Task 2 [15 points].** For training, initialise your weights using the Xavier Uniform initialisation, use ReLU as the activation function, a learning rate of 0.1 with the SGD optimiser. You will train your neural network over 30 epochs. In your report, provide the following: (a) final (train and test) accuracy obtained; (b) plot of the accuracy on the training and test sets per each epoch, comment on the speed of performance changes across epochs; (c) plot of the train loss per epoch (total sum of per batch losses for each epoch) and comment on the speed of decrease.

***Answer***: The final accuracy obtained on the training set is 96.97 and on the test set is 89.82. The train and test accuracies are in Figure (a) below. It shows that the training accuracy keeps growing, the test accuracy stops growing after epoch 7. The training loss is in Figure (b) below. It shows that the loss function decreases at a high rate during the 20 first steps and then keeps decreasing but with a smaller slope.

***Marking scheme:***

*For the code, mark as follows:*

* *0 points: no attempt.*
* *2 points: attempted but does not run and no comments.*
* *4 points: runs and does at least something relevant to the problem, no comments.*
* *6 points: runs and does at least something relevant to the problem, commented.*
* *8 points: works properly but could be neater, commented.*
* *10 points: works properly, neat and commented.*

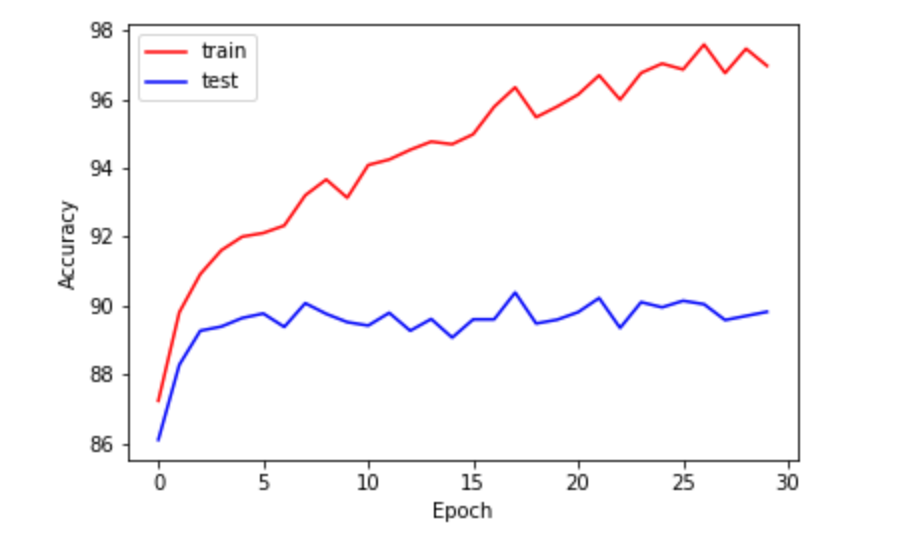
***Remark****: Students may provide valid implementations while using a range of relevant loss functions or by performing classification via regression.*

*1 for reporting the final accuracy values, 1 for reporting their plots, 1 point for the comment*

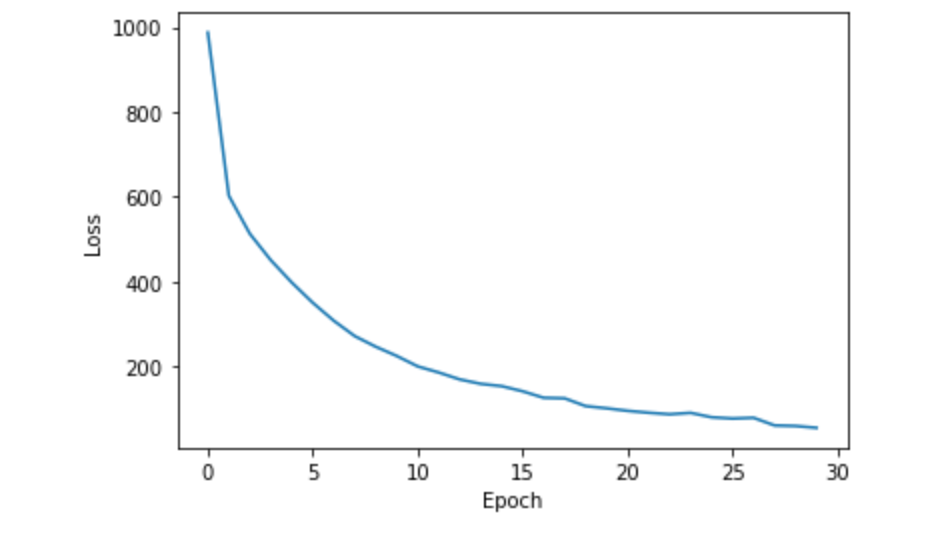
*1 for plotting the training loss, 1 point for the comment.*

***Remark****: Hereinafter, students may provide different accuracy values. Due to randomness, they are valid within the interval of 5 points as soon as the relative conclusions are valid (for example, test accuracy is lower than the train accuracy).*

*Figure (a)*

**

*Figure (b)*



**Task 3 [6 points].** Run three experiments each time changing all the current activation functions to one of the following: Tanh, Sigmoid and ELU. In your report, provide only the final classification accuracy values (train and test) per activation function and comment on the result.

***Answer***: The final accuracies for each activation function are in the Table below. The ELU and Tahn activation seem to slightly improve the train and test accuracy as compared to ReLU.

| Activation function | Train accuracy | Test accuracy |
| --- | --- | --- |
| Tanh | 100.00 | 91.38 |
| Sigmoid | 90.48 | 89.30 |
| ELU | 98.40 | 90.70 |

***Marking scheme:*** *0.75 point for each correct accuracy value (see the remark above), 1.5 point for the comment, deduct 1 point for each activation function which is not implemented correctly.*

**Task 4 [9 points].** Keeping ReLU, use 5 different learning rates: 0.001, 0.1, 0.5, 1, 10. In your report, provide the final train loss, as well as the final accuracy values for both train and test for each learning rate and comment on the trade-offs between speed and stability of convergence. Comment on why you get the Nan loss if any.

***Answer***: Table below reflects how the learning rate impacts the final train loss, train and test accuracies. High learning rates (0.5, 1 and 10) seem to result in missing local minima because we overstep them. On the other hand, a too small learning rate (0.001) makes the algorithm converge very slowly, meaning that more epochs are needed to find a good solution. The high learning rate of 10 makes the outputs of the network grow beyond infinity resulting in Nan outputs and the final Nan loss.

| Learning Rate | Train loss | Train accuracy | Test accuracy |
| --- | --- | --- | --- |
| 0.001 | 658.66 | 87.79 | 86.55 |
| 0.1 | 56.05 | 96.97 | 89.82 |
| 0.5 | 4323.85 | 10.00 | 10.00 |
| 1 | 4330.56 | 10.00 | 10.00 |
| 10 | Nan | 10.00 | 10.00 |

***Marking scheme:*** *0.2 point for each correct loss or accuracy value (see the remark above), 3 points for the correct implementation, 2 points for the correct comment on how the algorithm converges depending on the learning rate, 1 point for the correct comment on the Nan loss.*

**Task 5 [10 points].** Add a dropout of 0.3 rate on the second fully connected layer (keeping ReLU and learning rate 0.1). In your report, provide the final train and test accuracy values and explain how the dropout affects the performance.

***Answer***: On this particular architecture with this set of hyperparameters a dropout of 0.3 at the second fully-connected layer does not change the performance as compared to the results without the dropout: train acc. 96.75, test acc. 90.31.

***Marking scheme:*** *3 points for the correct answer, 7 points for the correct and commented implementation.*