1 Crystal clear (Logic Problem)

1.1 Q1

1. $\exists$x Owns(YOU, x) $\wedge$ Dog(x)
2. BuysCarrots(ROBIN)
3. $\forall$x [($\exists$y (Rabbit(y) $\wedge$ Owns(x, y))) $\rightarrow$ ($\forall$z ($\exists$w (Rabbit(w) $\wedge$ Chases(z, w)) $\rightarrow$ Hate(x, z)))]
4. $\forall$x [Dog(x) $\rightarrow$ ($\exists$y (Rabbit(y) $\wedge$ Chases(x, y)))]
5. $\forall$x [BuysCarrots(x) $\rightarrow$ ($\exists$y (Owns(x, y) $\wedge$ (Rabbit(y) $\vee$ GS(y))))]
6. $\forall$x, y [($\exists$z (Owns(x, z) $\wedge$ Hates(y, z))) $\rightarrow$ $\neg$Date(y,x)]

1.2 Q2

1.2.1 Step 1 - Rewrite $\rightarrow$

This is done using De Morgan's laws and using $\neg\exists \equiv \forall\neg$ and $\neg\forall \equiv \exists\neg$.

7. $\forall$x [$\neg$($\exists$y (Rabbit(y) $\wedge$ Owns(x, y))) $\vee$ ($\forall$z $\neg$($\exists$w (Rabbit(w) $\wedge$ Chases(z, w)) $\vee$ Hate(x, z)))] [From 3.]
8. $\forall$x [$\neg$Dog(x) $\vee$ ($\exists$y (Rabbit(y) $\wedge$ Chases(x, y)))] [From 4.]
9. $\forall$x [$\neg$BuysCarrots(x) $\vee$ ($\exists$y (Owns(x, y) $\wedge$ (Rabbit(y) $\vee$ GS(y))))] [From 5.]
10. $\forall$x, y [$\neg$($\exists$z (Owns(x, z) $\wedge$ Hates(y, z))) $\vee$ $\neg$Date(y,x)] [From 6.]

1.2.2 Step 2 - Minimise Negatives

11. $\forall$x [($\forall$y ($\neg$Rabbit(y) $\vee$ $\neg$Owns(x, y))) $\vee$ ($\forall$z ($\forall$w ($\neg$Rabbit(w) $\vee$ $\neg$Chases(z, w)) $\vee$ Hate(x, z)))] [From 7.]
12. $\forall$x, y [($\forall$z ($\neg$Owns(x, z) $\vee$ $\neg$Hates(y, z))) $\vee$ $\neg$Date(y,x)] [From 10.]

1.2.3 Step 3 - Standardise Variables

13. $\forall$a [$\neg$Dog(a) $\vee$ ($\exists$y (Rabbit(y) $\wedge$ Chases(a, y)))] [From 8.]
14. $\forall$b [$\neg$BuysCarrots(b) $\vee$ ($\exists$z (Owns(b, z) $\wedge$ (Rabbit(z) $\vee$ GS(z))))] [From 9.]
15. $\forall$c [($\forall$d ($\neg$Rabbit(d) $\vee$ $\neg$Owns(c, d))) $\vee$ ($\forall$e ($\forall$f ($\neg$Rabbit(f) $\vee$ $\neg$Chases(e, f)) $\vee$ Hate(c, e)))] [From 11.]
16. $\forall$g, h [($\forall$l ($\neg$Owns(g, l) $\vee$ $\neg$Hates(h, l))) $\vee$ $\neg$Date(h,g)] [From 12.]

1.2.4 Step 4 - Skolemise

17. Owns(YOU, D) $\wedge$ Dog(D) [From 1.]
18. $\forall$a [$\neg$Dog(a) $\vee$ ((Rabbit(M1(a)) $\wedge$ Chases(a, M1(a))))] [From 13.]
19. $\forall$b [$\neg$BuysCarrots(b) $\vee$ ((Owns(b, M2(b)) $\wedge$ (Rabbit(M2(b)) $\vee$ GS(M2(b)))))] [From 14.]

1.2.5 Step 5 - Drop Universal Quantifiers

20. [$\neg$Rabbit(d) $\vee$ $\neg$Owns(c, d) $\vee$ $\neg$Rabbit(f) $\vee$ $\neg$Chases(e, f) $\vee$ Hate(c, e)] [From 15.]
21. [$\neg$Owns(g, l) $\vee$ $\neg$Hates(h, l) $\vee$ $\neg$Date(h,g)] [From 16.]
22. [$\neg$Dog(a) $\vee$ ((Rabbit(M1(a)) $\wedge$ Chases(a, M1(a))))] [From 18.]
23. [$\neg$BuysCarrots(b) $\vee$ ((Owns(b, M2(b)) $\wedge$ (Rabbit(M2(b)) $\vee$ GS(M2(b)))))] [From 19.]

1.2.6 Step 6- CNF

24. [$\neg$Dog(a) $\vee$ (Rabbit(M1(a))) $\wedge$ ($\neg$Dog(a) $\vee$ Chases(a, M1(a)))] [From 22.]
25. [($\neg$BuysCarrots(b) $\vee$ Owns(b, M2(b))) $\wedge$ ($\neg$BuysCarrots(b) $\vee$ Rabbit(M2(b)) $\vee$ GS(M2(b)))] [From 23.]

1.2.7 Collation

Here I will collate all the most up-to-date statements to make it easier to work with.

1. BuysCarrots(ROBIN) [From 2.]
2. Owns(YOU, D) $\wedge$ Dog(D) [From 17.]
3. $\neg$Rabbit(d) $\vee$ $\neg$Owns(c, d) $\vee$ $\neg$Rabbit(f) $\vee$ $\neg$Chases(e, f) $\vee$ Hate(c, e) [From 20.]
4. $\neg$Owns(g, l) $\vee$ $\neg$Hates(h, l) $\vee$ $\neg$Date(h, g) [From 21.]
5. ($\neg$Dog(a) $\vee$ Rabbit(M1(a))) $\wedge$ ($\neg$Dog(a) $\vee$ Chases(a, M1(a))) [From 24.]
6. ($\neg$BuysCarrots(b) $\vee$ Owns(b, M2(b))) $\wedge$ ($\neg$BuysCarrots(b) $\vee$ Rabbit(M2(b)) $\vee$ GS(M2(b))) [From 25.]

From now on, I will refer to these.

1.3 Q3

First convert to FOL:

$\neg$($\exists$x GS(x) $\wedge$ Owns(ROBIN, x)) $\rightarrow$ $\neg$Date(ROBIN, YOU)

Then negate it:

$\neg$($\neg$($\exists$x GS(x) $\wedge$ Owns(ROBIN, x)) $\rightarrow$ $\neg$Date(ROBIN, YOU))

Now convert to CNF:

1.3.1 Step 1 - Rewrite $\rightarrow$

$\neg$($\neg\neg$($\exists$x GS(x) $\wedge$ Owns(ROBIN, x)) $\vee$ $\neg$Date(ROBIN, YOU))

1.3.2 Step 2 - Minimise Negative

$\neg$($\neg\neg$($\exists$x GS(x) $\wedge$ Owns(ROBIN, x)) $\vee$ $\neg$Date(ROBIN, YOU))

$\neg\neg$A $\equiv$ A and $\neg\vee \equiv \wedge$. So,

$\neg$($\exists$x GS(x) $\wedge$ Owns(ROBIN, x)) $\wedge$ Date(ROBIN, YOU)

Using the fact that $\neg\exists$x A $\equiv$ $\forall$x $\neg$A and De Morgan's Laws:

($\forall$x $\neg$GS(x) $\vee$ $\neg$Owns(ROBIN,x)) $\wedge$ Date(ROBIN, YOU)

1.3.3 Step 3 - Standardise Variables

Nothing to do.

1.3.4 Step 4 - Skolemise

Nothing to do.

1.3.5 Step 5 - Drop Universal Quantifiers

($\neg$GS(x) $\vee$ $\neg$Owns(ROBIN,x)) $\wedge$ Date(ROBIN, YOU)

1.3.6 Step 6 - CNF

Nothing to do.

1.3.7 Collation

So the final result of this section is:

1. ($\neg$GS(x) $\vee$ $\neg$Owns(ROBIN,x)) $\wedge$ Date(ROBIN, YOU)

1.4 Q4

First bring all the CNFs from the statements and the CNFs from the conclusion together (I.e., the parts I have put in the Collations sections):

1. BuysCarrots(ROBIN)
2. Owns(YOU, D) ∧ Dog(D)
3. ¬Rabbit(d) ∨ ¬Owns(c, d) ∨ ¬Rabbit(f) ∨ ¬Chases(e, f) ∨ Hate(c, e)
4. ¬Owns(g, l) ∨ ¬Hates(h, l) ∨ ¬Date(h, g)
5. (¬Dog(a) ∨ Rabbit(M1(a))) ∧ (¬Dog(a) ∨ Chases(a, M1(a)))
6. (¬BuysCarrots(b) ∨ Owns(b, M2(b))) ∧ (¬BuysCarrots(b) ∨ Rabbit(M2(b)) ∨ GS(M2(b)))
7. (¬GS(x) ∨ ¬Owns(ROBIN,x)) ∧ Date(ROBIN, YOU)

1.4.1 Step 7 - Split up on Conjunction

8. Owns(YOU, D) [From 2.]
9. Dog(D) [From 2.]
10. [¬Dog(a) ∨ (Rabbit(M1(a)))] [From 5.]
11. [¬Dog(a) ∨ Chases(a, M1(a))] [From 5.]
12. [¬BuysCarrots(b) ∨ Owns(b, M2(b))] [From 6.]
13. [¬BuysCarrots(b) ∨ Rabbit(M2(b)) ∨ GS(M2(b))] [From 6.]
14. ¬GS(x) ∨ ¬Owns(ROBINS,x) [From 7.]
15. Date(ROBIN, YOU) [From 7.]

1.4.2 Step 8 - Restandardise Variables

16. ¬Rabbit(d1) ∨ ¬Owns(c1, d1) ∨ ¬Rabbit(f1) ∨ ¬Chases(e1, f1) ∨ Hate(c1, e1) [From 3]
17. ¬Owns(g1,l1) ∨ ¬Hates(h1, l1) ∨ ¬Date(h1, g1) [From 4]
18. ¬Dog(a1) ∨ Rabbit(M1(a1)) [From 10]
19. ¬Dog(a2) ∨ Chases(a2, M1(a2)) [From 11]
20. ¬BuysCarrots(b1) ∨ Owns(b1, M2(b1)) [From 12]
21. ¬BuysCarrots(b2) ∨ Rabbit(M2(b2)) ∨ GS(M2(b2)) [From 13]
22. ¬GS(x1) ∨ ¬Owns(ROBIN, x1) [From 14]

1.4.3 Build my knowledge base

1. BuysCarrots(ROBIN) [From 1]
2. Owns(YOU, D) [From 8]
3. Dog(D) [From 9]
4. ¬Dog(a1) ∨ Rabbit(M1(a1)) [From 18]
5. ¬Dog(a2) ∨ Chases(a2, M1(a2)) [From 19]
6. ¬BuysCarrots(b1) ∨ Owns(b1, M2(b1)) [From 20]
7. ¬BuysCarrots(b2) ∨ Rabbit(M2(b2)) ∨ GS(M2(b2)) [From 21]
8. ¬Rabbit(d1) ∨ ¬Owns(c1, d1) ∨ Rabbit(f1) ∨ ¬Chases(e1, f1) ∨ Hate(c1,e1) [From 16]
9. ¬Owns(g1, l1) ∨ ¬Hates(h1, l1) ∨ ¬Date(h1, g1) [From 17]
10. ¬GS(x1) ∨ ¬Owns(ROBIN, x1) [From 22]
11. Date(ROBIN, YOU) [From 15]

1.4.4 Proof by Resolution

12. Rabbit(M1(D)) {D/a1} [From 3, 4]
13. Owns(ROBIN, M2(ROBIN)) {ROBIN/b1} [From 1, 6]
14. Rabbit(M2(ROBIN)) ∨ GS(M2(ROBIN)) {ROBIN/b2} [From 1, 7]
15. ¬GS(M2(ROBIN)) {M2(ROBIN)/x1} [From 10, 13]
16. ¬BuysCarrots(ROBIN) ∨ Rabbit(M2(ROBIN)) {ROBIN/b2} [From 7, 15]
17. ¬Hates(h1, D) ∨ ¬Date(h1, YOU) {YOU/g1, D/l1} [From 2, 9]
18. ¬Hates(ROBIN, D) {ROBIN/h1} [From 11, 17]
19. ¬Rabbit(d1) ∨ ¬Owns(ROBIN, d1) ∨ Rabbit(f1) ∨ ¬Chases(D, f1) {ROBIN/c1, D/e1} [From 8, 18]
20. ¬Rabbit(M2(ROBIN) ∨ Rabbit(f1) ∨ ¬Chases(D, f1) {M2(ROBIN)/d1} [From 14, 19]
21. ¬Rabbit(M2(ROBIN)) ∨ ¬Chases(D, M2(D)) {M1(D)/f1} [From 12, 20]
22. Rabbit(M2(ROBIN)) [From 14, 15]
23. Chases(D, M1(D)) {D/a2} [From 3, 5]
24. ¬Rabbit(M2(ROBIN)) [From 21, 23]
25. {} We have arrived at contradiction [From 22, 24]

As we have arrived at a contradiction, this means that conclusion is correct.

2 Lost in the closet (Classification)

2.1 Q1

I think that the most appropriate loss function to use would be the cross-entropy loss: $H(p,q) = -\sum_{x \in X} p(x) log(q(x))$
This equation is general. A more specific one for our problem would be: $H(y,\hat{y}) = -\sum_{x \in X} y(x) log(\hat{y}(x))$
Where:

1. X is the space of all inputs,
2. x is a particular input,
3. y is the actual model that generated the data and $y(x)$ is the true label for that input x,
4. $\hat{y}$ is the model that we are using to predict the outputs and $\hat{y}(x)$ is the predicted label for the input x.

The reason this is the best to use is that we are doing a Multi-class classification. It is used for classification because it it works very well with the outputs of a classifier.

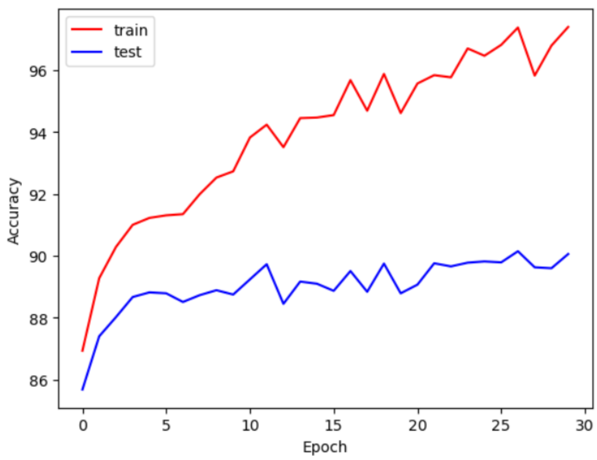
2.2 Q2

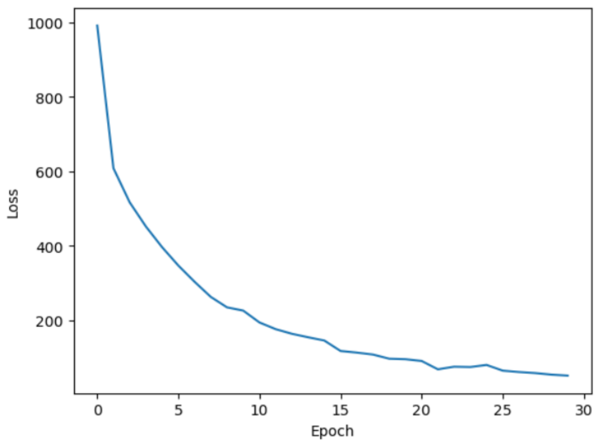
2.2.1 a) Final (train and test) accuracy obtained

The final Model accuracy is given by:

Loss	Training Accuracy	Testing Accuracy
56.59209	97.3983333	90.06

2.2.2 b) Plot of the accuracy on the training and test sets





In the training set in the figure on the left, we see that the accuracy rapidly increases in the first 4 epoch, going from 86.933 in epoch 1 to 91.003 in epoch 4, with an average rate of increase of 1.35667. Then, clearly slows and reaches epoch 10 with an accuracy of 92.732, where the average rate of increase from epoch 4 to 10 is 0.28817. This is nearly 5-fold less than the rate of increase from epoch 1 to 4. The final accuracy at epoch 30 is 97.398. The average rate of increase from epoch 10 to 30 is 0.2333. This is only slightly less than that of the average rate of increase from epoch 4 to 10 and it is nearly 6 fold smaller the average rate of increase from epoch 1 to 4. The average rate of increase over the entire set is: 0.36087. The rate of increase decreases as the epochs increase.

In the test set on the figure on the left, we see that the accuracy rapid increase in the first 2 epochs from 85.68 in epoch 1 to 87.4 in epoch 2. This has a rate of increase of 1.72. The accuracy at epoch 10 is 88.75. So, the average rate of increase from epoch 2 to 10 is 0.16875. This is much smaller than the starting rate of increase and the average rate of increase in the training set. On the test set, we reach a final accuracy of 90.06 at epoch 30. So, the rate of increase from epoch 10 to 30 is 0.0655. This is very low compared to the previous average rate of increases and also 4-fold smaller than the training set in the same interval. The average rate of increase over the entire set is: 0.15103.

Comparing the overall average rate of increases of the training set and the test set, we see that the average rate of increase on the training set is over 2-fold larger than that of the test set. This means that on average, the accuracy on the training set increases twice as fast as the accuracy over the test set. The rate of increase decreases as the epochs increase on both sets.

2.2.3 c) Plot of the train loss per epoch

Looking at the right plot above, it is obvious that the loss decreases as the number of epochs increases. The plot also shows that the rate of loss decrease is less as the number of epochs increases. This graph seems to follow an exponential decay, namely, the rate of loss decrease is greater at the start and it decreases over time in an exponential fashion. For example, the loss at epoch 1 is 1007 and it nearly halves to 623.75 at epoch 2 and then 528.96 at epoch 3. Then, looking further at epoch 10, the loss is 227.45 and then it eventually reaches 56.29 at epoch 30. This clearly shows a reduction in the rate of decrease of the loss as the number of epochs increases with the loss following an exponential decay pattern.

2.3 Q3

Activation	Training Accuracy	Testing Accuracy
Tanh	99.99333	91.47
Sigmoid	89.97333	88.56
ELU	97.46	89.54

Looking at the table above, we see: Tanh activation greatly over-fits the training set as the accuracy on the training set is nearly 100% but the accuracy on the test set is much lower at 91.47%. ELU activation also over-fits the training set with an accuracy of 97.46% but only an accuracy of 89.54%. Sigmoid activation only slightly overfits the training set with an accuracy of 89.97333% and 88.56% on the test set. The Tanh activation over-fits the training set the most with the difference in accuracy between the training set and test set being 8.52333%. This is closely followed by the ELU activation with 7.92%. Then, the Sigmoid activation hardly over-fits with an accuracy difference of only 1.41333%. SO, the best model is Tanh but it grossly overfits the training data.

2.4 Q4

Learning Rate	Training Loss	Training Accuracy	Testing Accuracy
0.001	675.567	87.397	86.510
0.1	55.836	96.958	89.670
0.5	4323.851	10.000	10.000
1	4330.562	10.000	10.000
10	nan	10.000	10.000

Looking at the table, we see that as the learning rate increases, the rate of convergence increases but if the learning rate is increased too much, the loss diverges. For example, going from a learning rate of 0.001 to 0.1, the rate of convergence increases and as such after 30 epochs, the training accuracy is much higher and the test accuracy is higher on a learning rate of 0.1 compared to 0.001, but also the loss with learning rate 0.1 is much less than that of learning rate 0.001 after 30 epochs (I.e., faster convergence). Now looking further, the model diverges as we increase the learning rate above that For example looking at the learning rates 0.5, 1, and 10, we see that the model accuracy is 10. I think this is due to the implementation. But looking at the training loss, we see that the loss for 0.5 and 1 hovers around 4330 but then heads to nan as we increase the learning rate to 10. This means that the model doesn't converge and it diverges because the learning rate is too high. So, the reason for the nan training loss is that the model diverges and the value of the training loss becomes too high to represent.

2.5 Q5

Model	Training Loss	Training Accuracy	Testing Accuracy
With Dropout	67.524	97.842	90.5
Without Dropout	56.592	97.398	90.06

Looking at the table above, we see that the loss on the loss on the dropout model is higher than that of the model without dropout. But, the accuracy on the training set of the dropout model is higher and so is the accuracy on the test set with the dropout. So, although the dropout model has a higher loss, it performs better (using accuracy as our measure). This is likely due to the fact the dropout method stop overfitting.