EECS759P Coursework 2 Report

Name: Bheki Maenetja
Student Number: 230382466

Crystal Clear (Logic Problem)

Irma's Six Statements

1) You have a dog.
2) The person you are looking for buys carrots by the bushel.
3) Anyone who owns a rabbit hates anything that chases any rabbit.
4) Every dog chases some rabbit.
5) Anyone who buys carrots by the bushel owns either a rabbit or a grocery store.
6) Someone who hates something owned by another person will not date that person.

Expressing Irma's Statements in First Order Logic (FOL)

1) $\exists x \; Dog(x) \land Own(YOU,\ x)$
2) $BuysCarrots(ROBIN)$
3) $\forall x \left(\exists y \left(Rabbit(y) \land Own(x,\ y)\right) \rightarrow \left(\forall z \left(\exists w \left(Rabbit(w) \land Chase(z,\ w)\right) \rightarrow Hate(x,\ z)\right)\right)\right)$
4) $\forall x \; Dog(x) \rightarrow \exists y \left(Rabbit(y) \land Chase(x,\ y)\right)$
5) $\forall x \; BuysCarrots(x) \rightarrow \left(\left(\exists y \left(Rabbit(y) \land Own(x, y)\right)\right) \lor \left(\exists z \left(GroceryStore(z) \land Own(x,\ z)\right)\right)\right)$
6) $\forall x \; \forall y \left(\exists z \left(Own(y,\ z) \land Hate(x,\ z)\right) \rightarrow \neg Date(x,\ y)\right)$

Translating FOL Expressions to Conjunctive Normal Form (CNF)

1) $\exists x \; Dog(x) \land Own(YOU,\ x)$
   i. **Removing implications**: nothing to do.
   ii. **Minimising negations**: nothing to do.
   iii. **Standardising variables**: nothing to do.
   iv. **Skolemising existentials**: $Dog(D) \land Own(YOU,\ D)$ replaced existential variable with object D.
   v. **Drop universals:** nothing to do.
   vi. **Conversion to CNF:** $Dog(D) \land Own(YOU,\ D)$

2) $BuysCarrots(ROBIN)$
   i. **Removing implications:** nothing to do.
   ii. **Minimising negations:** nothing to do.
   iii. **Standardising variables:** nothing to do.
   iv. **Skolemising existentials:** nothing to do.
   v. **Drop universals:** nothing to do.
   vi. **Conversion to CNF:** $BuysCarrots(ROBIN)$

3) $\forall x \left(\exists y \left(Rabbit(y) \land Own(x,\ y)\right) \rightarrow \left(\forall z \left(\exists w \left(Rabbit(w) \land Chase(z,\ w)\right) \rightarrow Hate(x,\ z)\right)\right)\right)$
   i. **Removing implications:** $\forall x \left(\neg\exists y \left(Rabbit(y) \land Own(x,\ y)\right) \lor \left(\forall z \left(\neg\exists w \left(Rabbit(w) \land Chase(z,\ w)\right) \lor Hate(x,\ z)\right)\right)\right)$
   ii. **Minimising negations:** $\forall x \forall y \left(\neg Rabbit(y) \lor \neg Own(x,\ y)\right) \lor \left(\forall z \forall w \left(\neg Rabbit(w) \lor \neg Chase(z,\ w)\right) \lor Hate(x,\ z)\right)$
   iii. **Standardising variables:** $\forall x1 \forall y1 \left(\neg Rabbit(y1) \lor \neg Own(x1,\ y1)\right) \lor \left(\forall z1 \forall w1 \left(\neg Rabbit(w1) \lor \neg Chase(z1,\ w1)\right) \lor Hate(x1,\ z1)\right)$
   iv. **Skolemising existentials:** nothing to do.
   v. **Drop universals:** $\left(\neg Rabbit(y1) \lor \neg Own(x1,\ y1)\right) \lor \left(\left(\neg Rabbit(w1) \lor \neg Chase(z1,\ w1)\right) \lor Hate(x1,\ z1)\right)$
   vi. **Conversion to CNF:** $\neg Rabbit(y1) \lor \neg Own(x1,\ y1) \lor \neg Rabbit(w1) \lor \neg Chase(z1,\ w1) \lor Hate(x1,\ z1)$

4) $\forall x\ Dog(x) \rightarrow \exists y\ \left(Rabbit(y) \land Chase(x,\ y)\right)$

   i. **Removing implications:** $\forall x\ \neg Dog(x) \lor \exists y\ \left(Rabbit(y) \land Chase(x,\ y)\right)$
   ii. **Minimising negations:** nothing to do.
   iii. **Standardising variables:** $\forall x2\ \neg Dog(x2) \lor \exists y2\ \left(Rabbit(y2) \land Chase(x2,\ y2)\right)$
   iv. **Skolemising existentials:** $\forall x2\ \neg Dog(x2) \lor \left(Rabbit\left(R(x2)\right) \land Chase\left(x2,\ R(x2)\right)\right)$
   v. **Drop universals:** $\neg Dog(x2) \lor \left(Rabbit\left(R(x2)\right) \land Chase\left(x2,\ R(x2)\right)\right)$
   vi. **Conversion to CNF:** $\left(\neg Dog(x2) \lor Rabbit\left(R(x2)\right)\right) \land \left(\neg Dog(x2) \lor Chase\left(x2,\ R(x2)\right)\right)$

5) $\forall x\ BuysCarrots(x) \rightarrow \left(\left(\exists y\ \left(Rabbit(y) \land Own(x, y)\right)\right) \lor \left(\exists z\ \left(GroceryStore(z) \land Own(x,\ z)\right)\right)\right)$

   i. **Removing implications:** $\forall x\ \neg BuysCarrots(x) \lor \left(\left(\exists y\ \left(Rabbit(y) \land Own(x, y)\right)\right) \lor \left(\exists z\ \left(GroceryStore(z) \land Own(x,\ z)\right)\right)\right)$
   ii. **Minimising negations:** nothing to do.
   iii. **Standardising variables:** $\forall x3\ \neg BuysCarrots(x3) \lor \left(\left(\exists y3\ \left(Rabbit(y3) \land Own(x3, y3)\right)\right) \lor \left(\exists z3\ \left(GroceryStore(z3) \land Own(x3,\ z3)\right)\right)\right)$
   iv. **Skolemising existentials:** $\forall x3\ \neg BuysCarrots(x3) \lor \left(Rabbit\left(F(x3)\right) \land Own\left(x3, F(x3)\right)\right) \lor \left(GroceryStore\left(G(x3)\right) \land Own\left(x3,\ G(x3)\right)\right)$
   v. **Drop universals:** $\neg BuysCarrots(x3) \lor \left(Rabbit\left(F(x3)\right) \land Own\left(x3, F(x3)\right)\right) \lor \left(GroceryStore\left(G(x3)\right) \land Own\left(x3,\ G(x3)\right)\right)$
   vi. **Conversion to CNF:** $\left(\neg BuysCarrots(x3) \lor Rabbit\left(F(x3)\right) \lor GroceryStore\left(G(x3)\right)\right) \land \left(\neg BuysCarrots(x3) \lor Own\left(x3,\ F(x3)\right) \lor GroceryStore\left(G(x3)\right)\right) \land \left(\neg BuysCarrots(x3) \lor Rabbit\left(F(x3)\right) \lor Own\left(x3,\ G(x3)\right)\right) \land \left(\neg BuysCarrots(x3) \lor Own\left(x3,\ F(x3)\right) \lor Own\left(x3,\ G(x3)\right)\right)$

6) $\forall x\ \forall y\ \left(\exists z\ \left(Own(y,\ z) \land Hate(x,\ z)\right) \rightarrow\ \neg Date(x,\ y)\right)$

   i. **Removing implications:** $\forall x\ \forall y\ \left(\neg\exists z\ \left(Own(y,\ z) \land Hate(x,\ z)\right) \lor\ \neg Date(x,\ y)\right)$
   ii. **Minimising negations:** $\forall x \forall y \forall z \left(\neg Own(y,\ z) \lor \neg Hate(x,\ z)\right) \lor\ \neg Date(x,\ y)$
   iii. **Standardising variables:** $\forall x4 \forall y4 \forall z2 \left(\neg Own(y4,\ z2) \lor \neg Hate(x4,\ z2)\right) \lor\ \neg Date(x4,\ y4)$
   iv. **Skolemising existentials:** nothing to do.
   v. **Drop universals:** $\neg Own(y4,\ z2) \lor \neg Hate(x4,\ z2) \lor\ \neg Date(x4,\ y4)$
   vi. **Conversion to CNF:** $\neg Own(y4,\ z2) \lor \neg Hate(x4,\ z2) \lor\ \neg Date(x4,\ y4)$

Irma's Statements in First Order Logic (FOL) Conjunctive Normal Form (CNF)

1) $Dog(D) \land Own(YOU,\ D)$
2) $BuysCarrots(ROBIN)$
3) $\neg Rabbit(y1) \lor \neg Own(x1,\ y1) \lor \neg Rabbit(w1) \lor \neg Chase(z1,\ w1) \lor Hate(x1,\ z1)$
4) $\left(\neg Dog(x2) \lor Rabbit\left(R(x2)\right)\right) \land \left(\neg Dog(x2) \lor Chase\left(x2,\ R(x2)\right)\right)$
5) $\left(\neg BuysCarrots(x3) \lor Rabbit\left(F(x3)\right) \lor GroceryStore\left(G(x3)\right)\right) \land \left(\neg BuysCarrots(x3) \lor Own\left(x3,\ F(x3)\right) \lor GroceryStore\left(G(x3)\right)\right) \land \left(\neg BuysCarrots(x3) \lor Rabbit\left(F(x3)\right) \lor Own\left(x3,\ G(x3)\right)\right) \land \left(\neg BuysCarrots(x3) \lor Own\left(x3,\ F(x3)\right) \lor Own\left(x3,\ G(x3)\right)\right)$
6) $\neg Own(y4,\ z2) \lor \neg Hate(x4,\ z2) \lor\ \neg Date(x4,\ y4)$

Irma's Conclusion in First Order Logic

Statement: If the person you are looking for does not own a grocery store, she will not date you.
FOL expression: $\neg\exists x\left(GroceryStore(x) \land Own(ROBIN,\ x)\right) \rightarrow\ \neg Date(ROBIN,\ YOU)$
Negation of FOL expression: $\neg\left(\neg\exists x\left(GroceryStore(x) \land Own(ROBIN,\ x)\right) \rightarrow\ \neg Date(ROBIN,\ YOU)\right)$
Negations of FOL expression in CNF:

   i. **Removing implications:** $\neg\left(\exists x\left(GroceryStore(x) \land Own(ROBIN,\ x)\right) \lor\ \neg Date(ROBIN,\ YOU)\right)$
   ii. **Minimising negatiions:** $\forall x\left(\neg GroceryStore(x) \lor \neg Own(ROBIN,\ x)\right) \land\ Date(ROBIN,\ YOU)$
   iii. **Standardising variables:** $\forall x5\left(\neg GroceryStore(x5) \lor \neg Own(ROBIN,\ x5)\right) \land\ Date(ROBIN,\ YOU)$
   iv. **Skolemising existentials:** nothing to do.
   v. **Drop universals:** $\left(\neg GroceryStore(x5) \lor \neg Own(ROBIN,\ x5)\right) \land\ Date(ROBIN,\ YOU)$
   vi. **Conversion to CNF:** $\left(\neg GroceryStore(x5) \lor \neg Own(ROBIN,\ x5)\right) \land\ Date(ROBIN,\ YOU)$

Proof that Madame Irma is Right

All Disjunctive Clauses

1) $Dog(D)$
2) $Own(YOU,\ D)$
3) $BuysCarrots(ROBIN)$
4) $\neg Rabbit(y1) \vee \neg Own(x1,\ y1) \vee \neg Rabbit(w1) \vee \neg Chase(z1,\ w1) \vee Hate(x1,\ z1)$
5) $\neg Dog(x2) \vee Rabbit\big(R(x2)\big)$
6) $\neg Dog(x3) \vee Chase\big(x3,\ R(x3)\big)$
7) $\neg BuysCarrots(x4) \vee Rabbit\big(F(x4)\big) \vee GroceryStore\big(G(x4)\big)$
8) $\neg BuysCarrots(x5) \vee Own\big(x5,\ F(x5)\big) \vee GroceryStore\big(G(x5)\big)$
9) $\neg BuysCarrots(x6) \vee Rabbit\big(F(x6)\big) \vee Own\big(x6,\ G(x6)\big)$
10) $\neg BuysCarrots(x7) \vee Own\big(x7,\ F(x7)\big) \vee Own\big(x7,\ G(x7)\big)$
11) $\neg Own(y2,\ z2) \vee \neg Hate(x8,\ z2) \vee\ \neg Date(x8,\ y2)$
12) $\neg GroceryStore(x9) \vee \neg Own(ROBIN,\ x9)$
13) $Date(ROBIN,\ YOU)$

Resolution Proof

14) $\neg Own(YOU,\ z2) \vee \neg Hate(ROBIN,\ z2)$ —> resolve (13) and (11), unifier = {ROBIN/x8, YOU/y2}
15) $\neg Hate(ROBIN,\ D)$ —> resolve (2) and (14), unifier = {D/z2}
16) $\neg Own(y2,\ D) \vee \neg Date(ROBIN,\ y2)$ —> resolve (15) and (11), unifier = {ROBIN/x8, D/z2}
17) $\neg Date(ROBIN,\ YOU)$ —> resolve (2) and (16), unifier = {YOU/y2}
18) $\emptyset$ —> resolve (13) and (17)

Therefore, because we have managed to reach the empty clause, we have proven Madame Irma's conclusion to be correct.

Lost in the Closet (Classification Problem)

The Loss Function

The most appropriate loss function for this problem is **Cross-Entropy Loss** given its suitability for problems where the output of the model is a probability distribution across multiple classes.

Loss Function Formula

- $L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log\big(p_{ij}\big)$

Interpretation of Formula

- Because the logarithm of a probability (which is between 0 and 1) is negative we need to put a negative sign at the start of the formula to make the loss positive.
- The averaging term ($\frac{1}{N}$) is needed to get the average loss over all N samples. This ensures that the loss does not depend on the number of samples that we take.
- $\sum_{i=1}^{N}$ is the sum over all samples in the dataset; i.e. we calculate the loss for each sample and then add it up.
- $\sum_{j=1}^{M}$ is the sum over all M classes in the dataset.
- y_{ij} is a binary value indicating whether the class j is indeed the correct class.
- $\log\big(p_{ij}\big)$ is the logarithm of the predicted probability p_{ij}, where p_{ij} is the probability (that our model assigns) that sample i belongs to class j.

Training the Model with ReLU

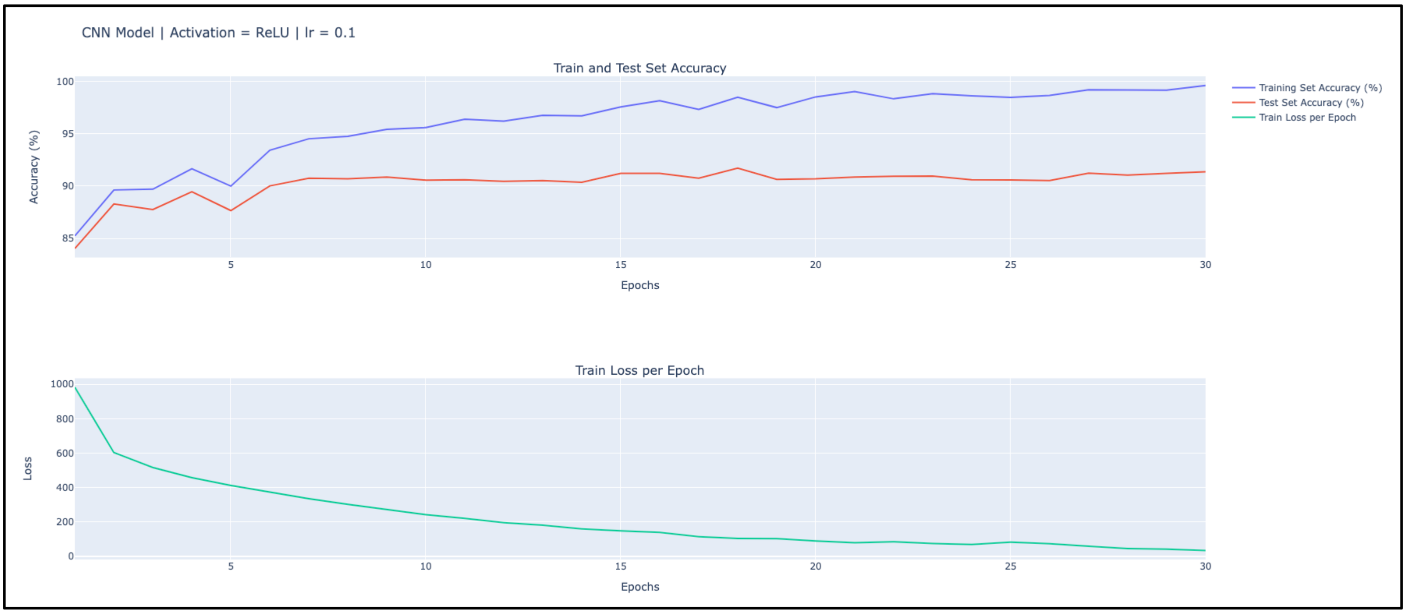

Figure 1 CNN model training results.

The CNN model – as outlined in section 3.2 of the coursework specification – was initially trained (see code blocks 8 to 10 in the notebook) with a ReLU activation function and a learning rate of 0.1. The final train and test accuracies were 99.635% and 91.38% respectively. The results of training (see figure 1) show a steady and stable improvement in the model's performance with the model's accuracy picking up noticeably after about 5 epochs before settling again after around 27 epochs. The loss decreases precipitously early on before steadily declining after the 4^{th} epoch.

Activation Function Experiments

Activation Function	Final Train Accuracy (%)	Final Test Accuracy (%)
Tanh	100.00	91.97
Sigmoid	90.50	88.95
ELU	98.62	90.53

Table 1 Results of training CNN model with different activation functions.

The results (see Table 1) show that all of the activation functions perform relatively well. The Tanh function achieves perfect results on the training set; this might be indicative of the model fitting too strongly to the training data, although it does also have the highest test set accuracy. The sigmoid function shows the least promise with by far the lowest accuracy on both the training and test sets; this is not surprising given the function's vulnerability to the "vanishing gradient" problem, especially when deeper and more complicated neural networks are involved. Much like ReLU, the ELU activation provides a high training accuracy and fairly good test set accuracy, although both values are not quite as high as those for ReLU and Tanh. Overall, the Tanh function is clearly the superior activation; not only to sigmoid and ELU, but to the ReLU activation as well.

Learning Rate Experiments

Learning Rate	Final Train Loss	Final Train Accuracy (%)	Final Test Accuracy (%)
0.001	680.11	86.87	85.48
0.1	47.27	99.57	91.06
0.5	NaN	10.00	10.00
1	4330.82	10.00	10.00
10	NaN	10.00	10.00

Table 2 Results of training CNN model with different learning rates.

The results (see Table 2) show that a moderate learning rate of around 0.1 provides optimal values for loss and accuracy – on both the training and test sets. The learning rate of 0.001 shows similar promise but it results in the model converging more slowly; more epochs may be required in order to improve performance. Higher learning rates of 0.5

and above result in a complete failure of the model to converge; this is indicated by the "NaN" values for the final train loss of the 0.5 and 10 learning rates. These NaN values signify instability in the training process; likely caused by the parameter updates being too large and thus causing the model to miss the global/local minima in the search space. While a higher learning rate might accelerate the model's convergence, it is also liable to make the model overshoot global/local; a lower learning rate, by contrast, will decrease the likelihood of overshooting but will require the use of more epochs during training for the model to converge.

Training Model with Dropout

When trained with a dropout rate of 0.3 on the 2^{nd} fully connected layer, the CNN model achieved final train and test accuracies of 97.63% and 89.47% respectively. These results are slightly worse than the equivalent CNN model without the dropout. This suggests that the dropout rate was either too high or not necessary at all, resulting in the model slightly underfitting the data where it otherwise wouldn't have. Although the inclusion of a dropout in a model is typically used as a form of regularisation – to prevent overfitting by deactivating a random set of neurons during training – the performance of the model with the dropout suggests that overfitting is not really a big problem for this particular model architecture.