COURSEWORK 2: LOGIC AND REASONING, NEURAL NETWORKS AND CLASSIFICATION.

QUESTION 1. LOGIC PROBLEM

a) Express Madame Irma's six statements into First Order Logic (FOL)

Propositions and connectives have been used to express the statements in FOL. Solutions are given in the following lines:

- You have a dog: $\exists d.\left(Dog(d) \land Own(You,d)\right)$
- The person you are looking for buys carrots by the bushel: $BuysCarrotsBushel(Robin)$
- Anyone who owns a rabbit hates anything that chases any rabbit:

  $\forall x.\forall r.\left(Own(x,r) \land Rabbit(r) \rightarrow \forall y.\forall s.\left(Rabbit(s) \land Chases(y,s) \rightarrow Hates(x,y)\right)\right)$
- Every dog chases some rabbit: $\forall d.\left(Dog(d) \rightarrow \exists r.\left(Rabbit(r) \land Chases(d,r)\right)\right)$
- Anyone who buys carrots by the bushel owns wither a rabbit or a grocery store: $\forall x.\left(BuysCarrotsBushel(x) \rightarrow \exists s.\left(Owns(x,s) \land \left(Rabbit(s) \lor Grocery(s)\right)\right)\right)$
- Someone who hates something owned by another person will not date that person:

  $\forall x.\forall y.\forall s.\left(Hates(x,s) \land Own(y,s) \rightarrow \neg Date(x,y)\right)$

b) Translate the obtained expressions to Conjunctive Normal Form (CNFs)

After expressing the given sentences into FOL, a set of 10 rules are used to translate the expressions into CNF. The resulting clauses are given in the following lines and will be included later as part of the Knowledge Base for future resolution proof.

- You have a dog: $\{Dog(d_1),\ Own(You,d_1)\}$
- The person you are looking for buys carrots by the bushel$\{BuysCarrotsBushel(Robin)\}$
- Anyone who owns a rabbit hates anything that chases any rabbit:

  $\{\neg Own(x_1,r_1) \lor \neg Rabbit(r_1) \lor \neg Chases(x_2,r_2) \lor \neg\ Rabbit(r_2) \lor Hates(x_1,x_2)\}$
- Every dog chases some rabbit: $\{\neg Dog(d_2) \lor Rabbit(f_1(d_2)),\ \neg Dog(d_2) \lor Chases(d_2, f1(d_2))\}$
- Anyone who buys carrots by the bushel owns wither a rabbit or a grocery store: $\{\neg BuysCarrotsBushel(x_3) \lor Owns(x_3,f_2(x_3)),\ \neg BuysCarrotsBushel(x_3) \lor Rabbit(f_2(x_3)) \lor Grocery(f_2(x_3))\}$
- Someone who hates something owned by another person will not date that person: $\{\neg Hates(x_4,x_5) \lor \neg Own(x_6,x_5) \lor \neg Date(x_4,x_6)$

c) Transform Madam Irma's conclusion into FOL, negate it and convert it to a CNF.

First the expression *"If the person you are looking for does not own a grocery, she will not date you"* must be translated into FOL. Just as it has been done in section a); propositions and connectives are used, leading to the following expression:

$$\neg\exists s.\left(Grocery(s) \land Own(Robin,s)\right) \rightarrow \neg Date(Robin,You)$$

Now, the expression will be converted into CNF. The set of steps and the final solution can be seen in the following lines. In addition, the rule that has been applied in each step has been commented in the right side of the line.

$\left(\neg\left(\neg\exists s.\left(Grocery(t) \wedge Own(Robin, t)\right) \to \neg Date(Robin, You)\right)\right)$ Negate before conversion.

$\neg\left(\neg\neg\exists s.\left(Grocery(t) \wedge Own(Robin, t)\right)\right) \wedge \neg\neg Date(Robin, You)$ Rewrite double negation

$\forall s.\left(\neg\left(Grocery(t) \wedge Own(Robin, t)\right)\right) \wedge Date(Robin, You)$ Minimise negation

$\forall s.\left(\neg Grocery(t) \vee \neg Own(Robin, t)\right) \wedge \ Date(Robin, You)$ De Morgan's Law

$\neg Grocery(t) \vee \neg Own(Robin, t) \wedge \ Date(Robin, You)$ Skolemise

$\{\neg Grocery(t) \vee \neg(Robin, t), \quad Date(Robin, You)\}$

d) Based on all the previously created CNF, prove that Madame Irma is right, and you should go to see Robin to declare her your love.

The expressions obtained in questions b) are added to the Knowledge Base. With help of these expressions and the clause $\neg Grocery(t) \vee \neg(Robin, t)$ obtained in the previous section, we must prove that $Date(Robin, You)$ is true. To achieve this purpose, proof must be done by contradiction, meaning that initially, we assume that the expression we want to prove is not true and show with the Knowledge base that this assumption is not possible. We must highlight, that the contradiction of the conclusion has been done already in the previous question, as asked; therefore, we will just use the obtained clause for this part.

The procedure that has been followed can be seen in the following lines, the CNF premises are presented in blue, the resolutions in green, the negated goal in orange and the unifiers in blue in the right side.

$Date\ (Robin, You)$ $\{Robin/x_4\}$

$\neg Hates(x_4, x_5) \vee \neg Own(x_6, x_5) \vee \neg Date(x_4, x_6)$ $\{You/x_6\}$

$\neg Hates(Robin, x_5) \vee \neg Own(You, x_5)$ $\{Robin/x_1\}$

$\neg Own(x_1, r_1) \vee \neg Rabbit(r_1) \vee \neg Chases(x_2, r_2) \vee \neg\ Rabbit(r_2) \vee Hates(x_1, x_2)$ $\{x_5/x_2\}$

$\neg Own(Robin, r_1) \vee \neg Rabbit(r_1) \vee \neg Chases(x_5, r_2) \vee \neg\ Rabbit(r_2) \vee \neg Own(You, x_5)$ $\{x_5/d_1\}$

$Own(You, d_1)$

$\neg Own(Robin, r_1) \vee \neg Rabbit(r_1) \vee \neg Chases(x_5, r_2) \vee \neg\ Rabbit(r_2)$

$BuysCarrotsBushel(Robin)$ $\{Robin/x_3\}$

$\neg BuysCarrotsBushel(x_3) \vee Owns(x_3, f_2(x_3))$ $\{f_2(Robin)/r_1\}$

$\neg Rabbit(f_2(Robin)) \vee \neg Chases(x_5, r_2) \vee \neg\ Rabbit(r_2)$ $\{x_5/d_2\}$

$\neg Dog(d_2) \vee Chases(d_2, f_1(d_2))$ $\{f_1(x_5)/r_2\}$

$\neg Rabbit(f_2(Robin)) \vee \neg Dog(x_5) \vee \neg\ Rabbit(f_1(x_5))$

$\neg BuysCarrotsBushel(Robin) \vee Rabbit(f_2(Robin)) \vee Grocery(f_2(Robin))$

$BuysCarrotsBushel(Robin)$

$\neg Dog(x_5) \vee \neg\ Rabbit\left(f(x_5)\right)\ \vee Grocery(f_2(Robin))$

$Dog(d_1) = Dog(x_5)$

$\neg\ Rabbit\left(f_1(x_5)\right)\ \vee Grocery(f_2(Robin))$

$\neg Dog(d_2) \vee Rabbit(f_1(d_2)) = \neg Dog(x_5) \vee Rabbit(f_1(x_5))$

$Dog(d_1) = Dog(x_5)$

$Grocery(f_2(Robin))$ $\{f_2(Robin)/t\}$

$\neg Grocery(t) \vee \neg Owns(Robin, t)$

$\neg Owns(Robin, f_2(Robin))$

$BuysCarrotsBushel(Robin)$

$\neg BuysCarrotsBushel(x_3) \vee Owns(x_3, f_2(x_3))$

{ } ✓ proved that Madame Irma was right

QUESTION 2. CLASSIFICATION.

The purpose of this section is to train a Convolutional Neural Network (CNN) that can differentiate among the content of an image. In this case, the Fashion MNIST dataset is used, to associate each image to one of these ten clothing items: T-shirt, trousers, pullover, dress, coat, sandal, shirt, sneaker, bag and ankle boot.

a) Given the problem, what is the most appropriate loss function to use?

The loss function will quantify the difference between the ground truth label of the image and the output obtained from the CNN. Ideally, the smaller the error, the better will be the model's fit.

Different loss functions can be used, the most common ones are the MSE and the Cross-Entropy Loss function. The former one is preferred for regression problems, while the latter one is the best option for classification problems.

As stated previously, we have a classification problem, therefore, the Cross-Entropy Loss function is considered to be the most appropriate. For each one of the 10 possible clothing items, we will get a probability value between 0 and 1. In this way, the garment that has a higher likelihood, will then, be chosen as the predicted value of the given model.

b) Create and train a Convolutional Neural Network corresponding to the given architecture.

The PyTorch framework will be used to create the CNN with the given architecture. Therefore, after uploading the data, and splitting it into training and testing subsets, the next step will be to create a class with help of the mentioned library, that will classify each one of the images with its corresponding clothing item.

The class named Classifier will contain 2 functions: initialisation and forward function. The former one will initialise the internal structure of the convolutional layer and define the operations that will take place in each layer. While the forward function will be in charge of performing those operations.

However, this is not enough, we have to train our neural network to do a proper classification. For that, we must update the values of the weights of each layer that will best map the input to the ground truth outputs.

To reach this purpose, an optimizer algorithm must be used. As stated in the problem, the Stochastic Gradient Descent (SGD) optimizer has to be applied. This algorithm will backpropagate through the neural network and fine-tune the weights, based on the error calculated with the Cross-Entropy Loss function in the previous epoch. It is important to highlight, that the weights will be initialised with the weights_init function that will apply the Xavier initialisation as asked to.

Later, an evaluation function is applied to analyse the performance of the model. This function will provide the accuracy of the model and the loss obtained along the time the CNN is working. The obtained results are plotted as it is observable in the following figures.

As expected, the training accuracy will be higher than the tested one. During training, we are fitting the data with the expected output, however, during the testing phase, the model obtained from training is applied to the testing data. Therefore, it is reasonable that the accuracy is higher for the former one.

Additionally, it can be seen that the loss function slowly decreases, therefore convergence is guaranteed, and we can assume that our model is reaching local minima and consequently, the estimated weights during backpropagation are the ones that will provide the best fit to our model.

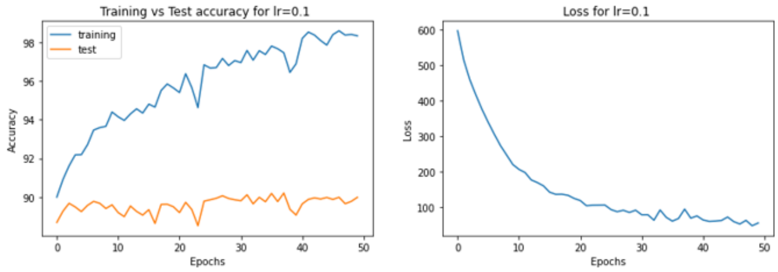

Figure 1 *Accuracy and Loss for CNN with ReLU activation function and lr=0.1*

c) Now, change the activation function to Tanh, Sigmoid and ELU. Provide only the final classification accuracy. Keeping ReLU, use 5 different learning rates: 0.001, 0.1, 0.5, 1, 10. What do you observe? Explain

When changing the activation function, for all four cases, no major improvements can be seen in the case of the testing accuracy (the value remains around 90%). Nevertheless, for ReLU and ELU a higher noise is observable.

On the other hand, when it comes to training accuracy, ELU and Tanh activation functions provide better results. These results make sense. If we compare the ELU function with the ReLU function, the former one has an additional α parameter that will tend to converge faster and therefore produce more accurate results. On the other hand, the Tanh function is non-zero centred and the gradient is stronger compared to the Sigmoid function, therefore it will converge faster producing better results too.

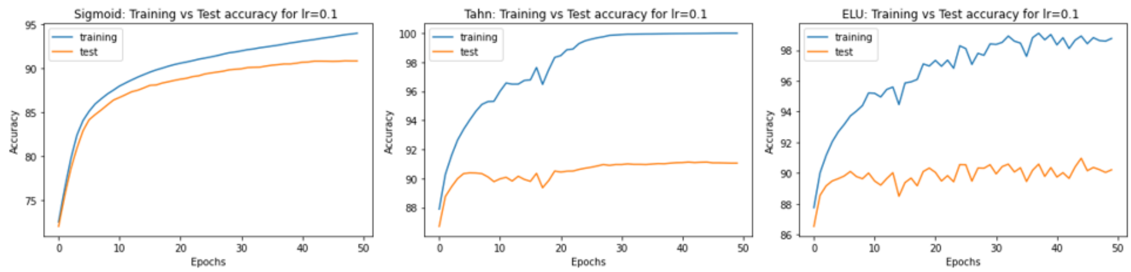

Figure 2 *Accuracy results for Sigmoid, Tanh and ELU activation functions.*

When it comes to changing the learning rates, graphs in Figure 3 are obtained. For 0.001, the learning rate is too small, the model lacks of iterations to reach convergence and therefore it is not obtaining weights values that fit best our model. This can be seen in the accuracy of the training set, where the model is not able to improve its results over time. Therefore, we discard the CNN with learning rate 0.001 as the optimal model.

For learning rate higher or equal than 0.5, we can see how the accuracy of both models is low. The value of the learning rate is too high causing divergence and therefore our model is unstable and won’t be able to learn properly.

Finally, with a learning rate of 0.1, we are obtaining the best results, since as said before, the model converges to local minima and provides the best fit to our model.

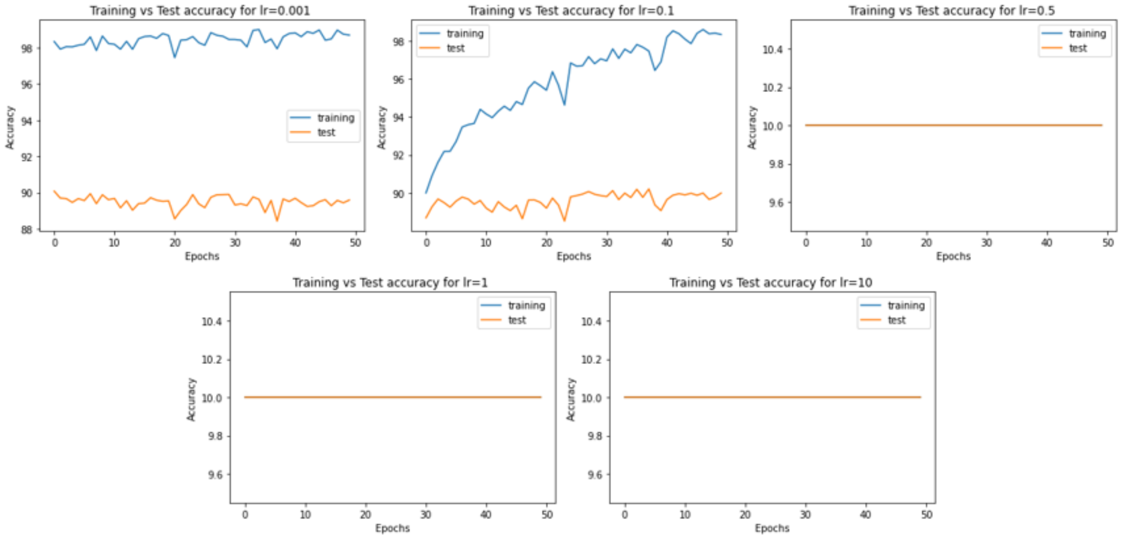

Figure 3 *Training and testing accuracies for different learning rates.*

d) Now, add a dropout of 0.3 rates on the second fully connected layer. What is the impact of dropout on performance? Provide the plot for training and test after each epoch. What happens if you decrease or increase the dropout rate?

Applying a dropout to the second fully connected layer means that during the training process, some of the neuros of that layer will be deactivated. Doing this will make the model more robust, since each neuron will have to be able to learn a particular concept without relying on others.

In the following graphs, we can see how different dropout values affect the performance of the model.

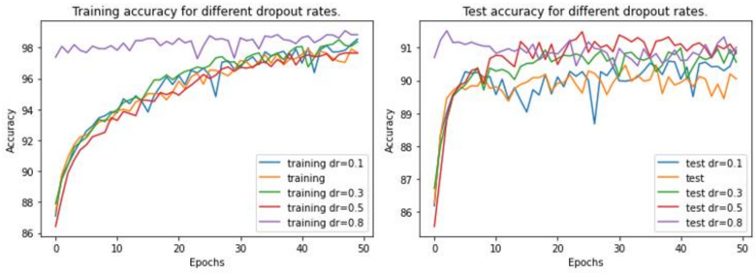

Figure 4 *Training and Testing accuracy values for different dropout rates.*

The rate of the dropout will determine the probability that the neuron is preserved or not. The higher the value, more chances has the neuron to be dropped out and therefore a higher generalisation of the model must be made. This implies that the training accuracy will decrease, while the test accuracy increases, as it can be seen when the dropout values are 0.3 and 0.5. However, we must take care, since if we go beyond a threshold rate value, the model will not be able to train properly, leading to undesired results as seen for dropout rate of 0.8.

On the other hand, when the rate value is 0.1, the chances of not preserving a neuron are quite low, the accuracy of the training set will be higher, because the model will not generalise too much and therefore the test set accuracy will decrease with respect to the other values.