Question 1 (a) FOL of the six statements are given below:

(i) $\exists x\ (dog(x) \wedge owns(ME, x))$
(ii) $buys\text{-}carrot(ROBIN)$
(iii) $\forall a \forall b\ owns(a,b) \wedge rabbit(b) \rightarrow \forall c \forall d\ chases(c,d)$
$\wedge\ rabbit(d) \rightarrow hates(a,c)$
(iv) $\forall x\ dog(x) \rightarrow \exists y\ rabbit(y) \wedge chases(x,y)$
(v) $\forall x\ buy\text{-}carrot(x) \rightarrow \exists y\ owns(x,y) \wedge$
$\{rabbit(y) \vee grocery(y)\}$
(vi) $\forall x \forall y \forall z\ owns(x,y) \wedge hates(z,y) \rightarrow \neg date(z,x)$

Question 1(b) CNFs of the statements are given below:

(i) $owns(ME, D)$
(ii) $buys\text{-}carrot(ROBIN)$
(iii) $\neg owns(x_1, x_2) \vee \neg rabbit(x_2) \vee \neg rabbit(x_3) \vee$
$\neg chases(x_4, x_3) \vee hates(x_1, x_4)$
(iv) $\neg dog(b) \vee rabbit(F(b))$
(v) $\neg dog(b) \vee chases(b, F(b))$
(vi) $\neg buys\text{-}carrot(z) \vee owns(z, G(z))$
(vii) $\neg buys\text{-}carrot(z) \vee rabbit(G(z)) \vee grocery(G(z))$
(viii) $\neg owns(a_1, a_2) \vee \neg hates(a_3, a_2) \vee \neg date(a_3, a_1)$
(ix) $dog(D)$

Here the constants are D, ME and ROBIN. F(b) and G(z) are functions.

Question 1 (c)

FOL:

$\neg(\exists x\ grocery(x) \wedge own(ROBIN, x)) \rightarrow \neg date(ROBIN, ME)$

Negate and convert to CNF:-

$\neg(\exists x\ grocery(x) \wedge owns(ROBIN, x)) \wedge \neg\neg date(ROBIN, ME)$

as $\neg(P \rightarrow Q) = P \wedge \neg Q$

$\{\forall x \neg grocery(x) \vee \forall x \neg owns(ROBIN, x)\} \wedge date(ROBIN, ME)$

$\{\neg grocery(y) \vee \neg owns(ROBIN, y)\} \wedge date(ROBIN, ME)$

Conclusion

(i) $\neg grocery(y) \vee \neg owns(ROBIN, y)$
(ii) $date(ROBIN, ME)$

Question 1(d)

$\{date(ROBIN, ME)\} \wedge \{\neg own(a_1, a_2) \vee \neg hate(a_3, a_2)$
$\vee \neg date(a_3, a_1)\} \rightarrow \{\neg own(ME, a_2) \vee \neg hate(ROBIN, a_2)\}$

$\{ROBIN/a_3\}$ & $\{ME/a_1\}$

$\{\neg own(ME, a_2) \vee \neg hate(ROBIN, a_2)\} \wedge \{\neg own(x_1, x_2) \vee$
$\neg rabbit(x_2) \vee \neg rabbit(x_3) \neg chases(x_4, x_3) \vee hate(x_1, x_4)\}$
$\rightarrow \{\neg own(ME, a_2) \vee \neg own(ROBIN, x_2) \vee \neg rabbit(x_2)$
$\vee \neg rabbit(x_3) \vee \neg chase(a_2, x_3)\}$

$\{ROBIN/x_1\}$ & $\{a_2/x_4\}$

The resolution has been written in this form (P V Q) Λ (R V not Q) →(P V R). The unifiers are written below it.

$$\{\neg own(ME, a_2) \vee \neg own(ROBIN, x_2) \vee \neg rabbit(x_2) \vee \neg rabbit(x_3) \vee \neg chase(a_2, x_3)\} \wedge \{OWN(ME, D)\} \rightarrow \{\neg own(ROBIN, x_2) \vee \neg rabbit(x_2) \vee \neg rabbit(x_3) \vee \neg chases(D, x_3)\}$$

$$\{D/a_2\}$$

$$\{\neg own(ROBIN, x_2) \vee \neg rabbit(x_2) \vee \neg rabbit(x_3) \vee \neg chase(D, x_3)\} \wedge \{\neg dog(b) \vee chases(b, F(b))\} \rightarrow \{\neg own(ROBIN, x_2) \vee \neg rabbit(x_2) \vee \neg rabbit(F(D)) \vee \neg Dog(D)\}$$

$$\{D/b\} \ \& \ \{F(D)/x_3\}$$

$$\{\neg own(ROBIN, x_2) \vee \neg rabbit(x_2) \vee \neg rabbit(F(D)) \vee \neg dog(D)\} \wedge \{\neg dog(b) \vee rabbit(F(b))\} \rightarrow \{\neg own(ROBIN, x_2) \vee \neg rabbit(x_2) \vee \neg dog(D)\}$$

$$\{D/b\}$$

$$\{\neg own(ROBIN, x_2) \vee \neg rabbit(x_2) \vee \neg dog(D)\} \wedge \{dog(D)\} \rightarrow \{\neg own(ROBIN, x_2) \vee \neg rabbit(x_2)\}$$

$$\{\neg own(ROBIN, x_2) \vee \neg rabbit(x_2)\} \wedge \{\neg \text{buy-carrot}(z) \vee rabbit(G(z)) \vee grocery(G(z))\} \rightarrow \{\neg own(ROBIN, G(z)) \vee \neg \text{buy-carrot}(z) \vee grocery(G(z))\}$$

$$\{G(z)/x_2\}$$

$$\{\neg own(ROBIN, G(z)) \vee \neg \text{buy-carrot}(z) \vee grocery(G(z))\} \wedge \{\neg grocery(y) \vee \neg own(ROBIN, y)\} \rightarrow \{\neg \text{buy-carrot}(z) \vee \neg own(ROBIN, G(z))\}$$

$$\{G(z)/y\}$$

$$\{\neg \text{buy-carrot}(z) \vee \neg own(ROBIN, G(z))\} \wedge \{\neg \text{buy-carrot}(z) \vee owns(z, G(z))\} \rightarrow \{\neg \text{buy-carrot}(ROBIN)\}$$

$$\{ROBIN/z\}$$

$$\{\neg \text{buy-carrot}(ROBIN)\} \wedge \{\text{buy-carrot}(ROBIN) \rightarrow \{\}$$

As the resolution results in an empty set, therefore the conclusion is proved.

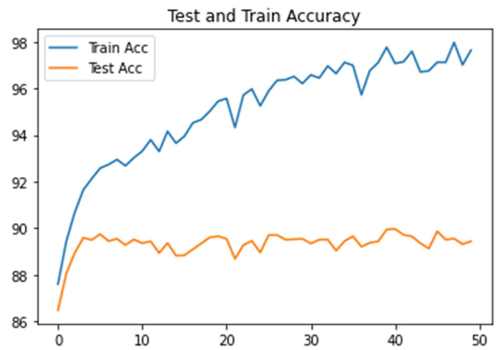
Question 2 (a)

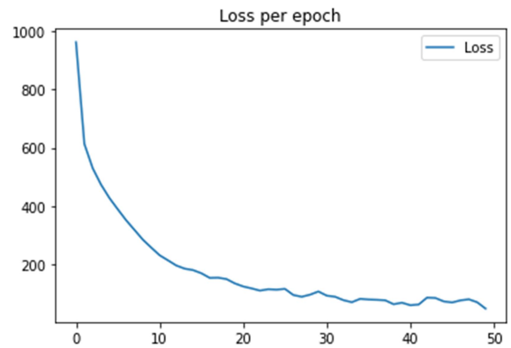
The problem is about multiclass classification and hence the appropriate loss function to use is categorical cross entropy which is given by $H(y,p) = -\sum_i y_i \log(p_i)$. This loss is used with the output of the activation function (softmax) of the last layer (predicted y) and the actual y. The loss will be high if the difference between the predicted and actual value is high. In pytorch the loss function "CrossEntropyLoss()" is used which already uses softmax internally. Therefore, the last layer doesn't need an activation function.

Question 2(b)

The graph and final stats are given below:

Epoch [50/50], Loss: 47.9290, Train Accuracy: 97.63%, Test Accuracy: 89.44%





The loss for the training reduces over the epochs as the training and test accuracy increases. The loss reduces drastically for the first few epochs and then it reduces slowly over the next epochs. The training accuracy increases till 98% while the test accuracy is at 89%. This suggests overfitting of the model on the dataset.

Question 2(c)

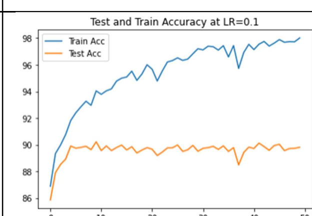
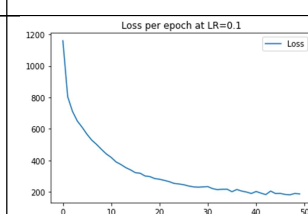
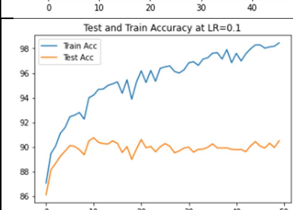
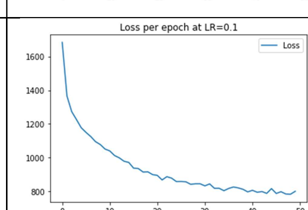
Activation function	Train Accuracy %	Test Accuracy %	Loss
Tanh	99.98	90.89	0.8112
Sigmoid	93.85	90.12	292.1002
ELU	97.89	89.58	64.8830

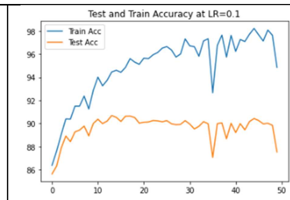
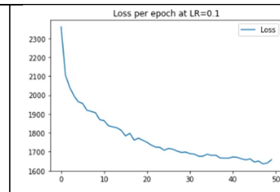
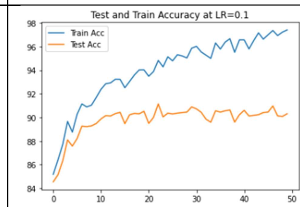
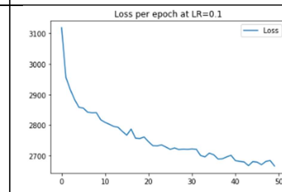
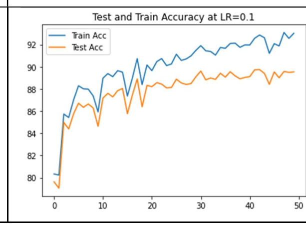
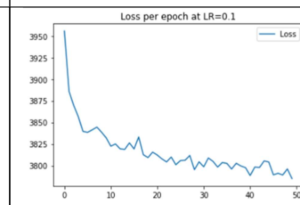
We can see that the loss is very less for tanh activation as it is a steep activation function. The loss for sigmoid is relatively high because it is less steep.

Learning Rate	Train Accuracy %	Test Accuracy %	Loss
0.001	88.90	87.71	599.9493
0.1	97.70	89.96	52.8565
0.5	89.60	84.86	648.4830
1	10.00	10.00	4330.5616
10	10.00	10.00	4636.2252

The loss for LR=0.001 is high and the train accuracy is 88% as it takes a lot more time for the model to converge. With very small learning rate, a higher number of epochs are required. When the LR is increased to 0.1, the train and test accuracy also increases and the loss is about 52. When the learning rate is increased the model starts to overshoot and can miss the optimal weights. Hence the loss increases. Therefore, large learning rate results in unstable training and very small learning rate results in very slow training.

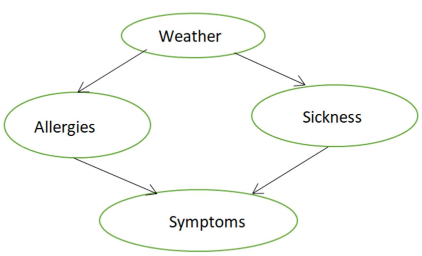
Question 2 (d)

Dropout	Train Acc	Test Acc	Loss	Acc graph	Loss graph
0.1	97.85	90.08	186.2777		
0.3	98.45	90.49	800.6372		

0.5	94.85	87.54	1657.9352		
0.7	97.41	90.31	2665.6581		
0.9	93.01	89.53	3785.0024		

Adding dropout is a method for regularising and hence helps in reducing over fitting of the model on the training set. It also increases the loss for the model on training set as shown above. The dropout is used to freeze the weights or drop the nodes during training. Higher dropout rate results in higher probability for the node to freeze. A frozen node means the value of the weight is zero. As shown in the graph above, the distance between the train accuracy (blue line) and test accuracy (orange line) is decreasing and the train accuracy becomes comparable with the test accuracy. Hence, the over fitting of the model is controlled. A low dropout rate overfits the model and a high dropout rate can underfit the model.

Question 3(a)



Question 3(b)

P(w=rainy)	P(w=sunny)	P(w=windy)
0.25	0.6	0.15

Weather

	P(allergy = true)
P(w=rainy)	0.15
P(w=sunny)	0.5
P(w=windy)	0.6

Allergies

	P(sick=true)
P(w=rainy)	0.8
P(w=sunny)	0.15
P(w=windy)	0.45

Sickness

Question 3(c)

P(s=headache, allergy=false | w= sunny)

= {P(w=sunny | s=headache, allergy=false) * P(s=headache, allergy=false)}/P(w=sunny)

As P(w) is not dependent on P(s) and P(allergy) therefore P(w=sunny | s=headache, allergy=false) = P(w=sunny)

= P(s=headache, allergy=false)

= P(s=headache | allergy=false) * P(allergy=false)

= {P(s=headache, sick=true | allergy=false) + P(s=headache, sick=false | allergy=false)} * P(allergy=false)

= [{P(s=headache | sick=true, allergy=false) * P(sick=true|allergy=false)} + {P(s=headache | sick=false, allergy=false) * P(sick=false|allergy=false)}] * P(allergy=false)

As P(sick) and P(allergy) are independent, therefore:

= [{P(s=headache | sick=true, allergy=false)*P(sick=true)} + {P(s=headache | sick=false, allergy=false)*P(sick=false)}] * P(allergy=false)

As P(sick=true) = P(sick=true|w) and as given w=sunny, therefore

P(sick=true) = P(sick=true|w=sunny) = 0.15 and P(sick=false) = 1-P(sick=true) = 0.85

Similarly, P(allergy=false) = 1 - P(allergy=true| w) and as w=sunny, we get,

P(allergy=false) = 1- P(allergy=true | w=sunny) = 1-0.5 = 0.5

Substituting these values and the values from the conditional probability table in the equation above:

P(s=headache, allergy=false | w= sunny) = {(0.1 * 0.15) + (0.15 * 0.85)} * 0.5 = 0.07125

Question 3(d)

P(s=stom | w=rainy)

= P(w=rainy | s= stom) * P(s = stom)/P(w=rainy)

As P(w) is not dependent on P(s) thus, P(w=rainy | s= stom) = P(w=rainy)

= P(s=stom)

= P(s=stom | allergy = true, sick=true) * P(allergy =true, sick=true) + P(s=stom | allergy = false, sick=true) * P(allergy=false,sick=true) + P(s=stom | allergy = true, sick=false) * P(allergy =true, sick=false) + P(s=stom | allergy = false, sick=false) * P(allergy=false, sick=false)

As P(allergy) and P(sick) are independent therefore, P(allergy, sick) = P(allergy) * P(sick). Therefore,

= {P(s=stom | allergy = true, sick=true) * P(allergy =true) *P(sick=true)} + {P(s=stom | allergy = false, sick=true) * P(allergy=false) * P(sick=true)} + {P(s=stom | allergy = true, sick=false) * P(allergy =true)* P(sick=false)} + {P(s=stom | allergy = false, sick=false) * P(allergy=false) * P(sick=false)}

Also, as P(allergy=true) = P(allergy=true|w) and w=rainy, therefore we get, P(allergy=true) = P(allergy=true |w=rainy) = 0.15 and P(allergy=false) = 1-P(allergy = true) = 0.85

Similarly we get P(sick=true) = 0.8 and P(sick = false) = 0.2.

Substituting these values and the values from the conditional probability table in the equation above:

P(s=stom | w=rainy) = (0.1 * 0.15 * 0.8) + (0.4 * 0.85 * 0.8) + (0.05 * 0.15 * 0.2) + (0.8 * 0.85 * 0.2) = 0.4215

References:

https://www.deeplearningwizard.com/deep_learning/boosting_models_pytorch/weight_initialization_activation_functions/