Queen Mary, University of London
School of Electronic Engineering and Computer Science

ECS759P: Artificial Intelligence

Coursework 2: Logic and Reasoning,
Neural Networks and Classification

Mughees Asif | 180288337

Due: 17 December 2021

Crystal clear

(a) First Order Logic (FOL)

1. $\forall x$ (DOG(x) ^ owns(ME, x))
2. buys-carrot($ROBIN$)
3. $\forall a$ $\forall b$ owns (a, b) ^ rabbit(b) $\longrightarrow$ $\forall c$ $\forall d$ chases (c, d).
4. $\forall x$ dog(x $\longrightarrow$ $\exists y$ rabbit(y) ^ chases(x, y)
5. $\forall x$ buy_carrot(x) $\longrightarrow$ $\exists y$ owns(x, y) ^ {rabbit(y) v grocery(y)}
6. $\forall x$ $\forall y$ $\forall z$ owns(x, y) ^ hates(z, y) $\longrightarrow$ $\neg$date(z, x)

(b) Conjunctive Normal Forms (CNFs)[1]

1. owns(ME)
2. buys-carrot($ROBIN$)
3. $\neg$ owns(x_1, x_2) v $\neg$ rabbit(x_2) v $\neg$ rabbit(x_3) v $\neg$ chases(x_4, x_3) v hates(x_1, x_4)
4. $\neg$ dog(b) v rabbit($F(b)$)
5. $\neg$ dog(b) v chases(b, $F(b)$)
6. $\neg$ buys-carrot(z) v owns(z, $G(z)$)
7. $\neg$ buys-carrot(z) v rabbit($G(z)$) v grocery($G(z)$)
8. $\neg$ owns(a_1, a_2) v $\neg$ hates(a_3, a_2) v $\neg$ date(a_3, a_1)
9. dog(D)

(c) Conclusion (FOL to CNF)

FOL:

$\neg$ ($\exists x$ grocery(x) ^ own($ROBIN$, x)) $\longrightarrow$ $\neg$ date ($ROBIN$, ME)

CNF:

$\neg$ ($\exists x$ grocery(x) ^ own($ROBIN$, x)) ^ $\neg$ $\neg$ date($ROBIN$, ME) as $\neg$ ($P \longrightarrow Q$) $= P$ ^ $\neg$ Q

{$\forall x$ $\neg$ grocery(x) v $\forall x$ $\neg$ owns($ROBIN$, x)} ^ date($ROBIN$, x)

{$\neg$ grocery(y) v owns($ROBIN$, y)} ^ date($ROBIN$, ME)

Therefore,

1. $\neg$ grocery(y) v $\neg$ owns($ROBIN$, y)
2. date($ROBIN$, ME)

[1] $F(b)$ and $G(z)$ are functions while D, ME and $ROBIN$ are constants.

(d) Proof

{date($ROBIN$, ME)} ^ {¬ own(a_1, a_2) v ¬ hate(a_3, a_2) v ¬ date(a_3, a_1)} ⟶ {¬ own(ME, a_2) v ¬ hate($ROBIN$, a_2)}

{$ROBIN/a_3$} and {ME/a_1}

{¬ own(ME, a_2) v ¬ hate($ROBIN$, a_2)} ^ {¬ own(x_1, x_2) v ¬ rabbit(x_2) v ¬ rabbit(x_3) ¬ chases(x_4, x_3) v hate(x_1, x_4)} ⟶ {own(ME, a_2) v own($ROBIN$, x_2) v rabbit(x_2) v ¬ rabbit(x_3) v ¬ chase(a_2, x_3)}

{$ROBIN/x_1$} and {a_2/x_4}

{¬ own(ME, a_2) v ¬ hate($ROBIN$, x_2) v ¬ rabbit(x_2) v ¬ rabbit(x_3) v ¬ chase(a_2, x_3)} ^{own(ME, D)} ⟶ {¬ own($ROBIN$, x_2) v ¬ rabbit(x_2) v ¬ rabbit(x_3) v ¬ chases(D, x_3)}

{D/a_2}

{¬ own($ROBIN$, x_2) v ¬ rabbit(x_2) v ¬ rabbit(x_3) v ¬ chase(D, x_3)} ^{¬ dog(b) v chases(b, $F(b)$)} ⟶ {¬ own($ROBIN$, x_2) v ¬ rabbit(x_2) v ¬ rabbit($F(D)$) v ¬ Dog(D)}

{D/b} and {$F(D)/x_3$}

{¬ own($ROBIN$, x_2) v ¬ rabbit(x_2) v ¬ rabbit($F(D)$) v ¬ dog(D)} ^{dog(b) v rabbit($F(b)$)} ⟶ {¬ own($ROBIN$, x_2) v ¬ rabbit(x_2) v dog(D)}

{D/b}

{¬ own($ROBIN$, x_2) v ¬ rabbit(x_2) v ¬ dog(D)} ^{dog(D)} ⟶ {¬ own($ROBIN$, x_2) v ¬ rabbit(x_2)}

{¬ own($ROBIN$, x_2) v ¬ rabbit(x_2)} ^{¬ buy_carrot(z) v rabbit($G(z)$) v grocery($G(z)$)} ⟶ {¬ own($ROBIN$, $G(z)$) v ¬ buy_carrot(z) v grocery($G(z)$)}

{$G(z)/x_2$}

{¬ own($ROBIN$, $G(z)$) v ¬ buy-carrot(z) v grocery($G(z)$)} ^{¬ grocery(y) v ¬ own($ROBIN$, y)} ⟶ {¬ buy_carrot(z) v ¬ own($ROBIN$, $G(z)$)}

{$G(z)/y$}

{¬ buy_carrot(z) v ¬ own($ROBIN$, $G(z)$)} ^{buy_carrot(z) v owns(z, $G(z)$)} ⟶ {¬ buy_carrot($ROBIN$)}

{$ROBIN/z$}

{¬ buy_carrot($ROBIN$)} ^{¬ buy_carrot($ROBIN$)} ⟶ {}

Lost in the closet

(a) Loss function

The problem requires multi-class classification to differentiate between the different clothing products, therefore, the categorical cross-entropy loss function would be the most appropriate. The function is defined mathematically as[2]:

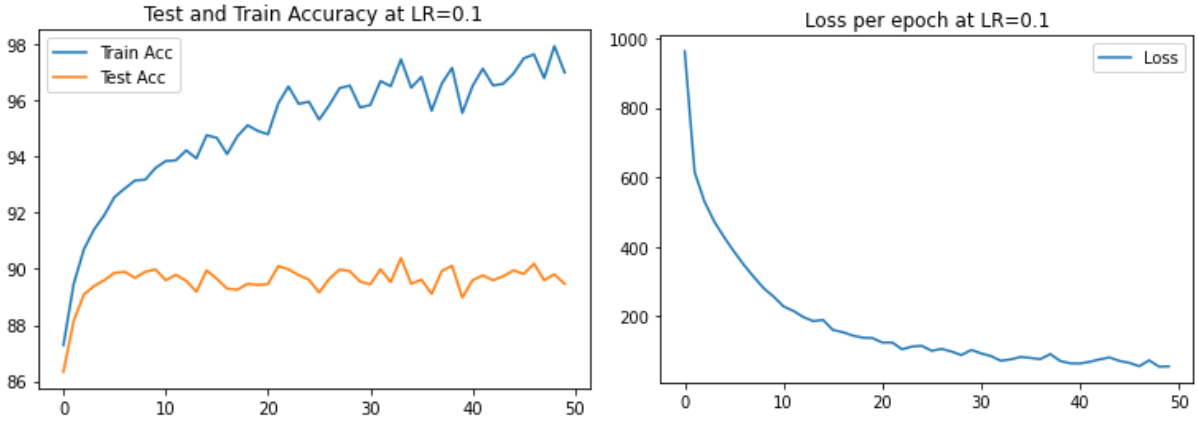
$$\text{Loss} = -\sum_{i}^{C} t_i \log f(s_i) \tag{1}$$

where,

t_i = groundtruth score for each class i
s_i = CNN score for each class i
$f(s_i)$ = activation function to generate class probabilities

The loss function allows the calculation of the average difference between the actual and predicted probability distributions, thereby, enabling generation of various output labels to classify the different types of clothing items.

(b) CNN: Final test and train accuracy



(a) Test and train accuracy

(b) Loss

Figure 1: Convolutional Neural Network metrics

```
Loss: 56.0292, Train Accuracy: 97.01%, Test Accuracy: 89.47%
```

Fig. 1 shows the test and train accuracy and the loss of the Convolutional Neural Network (CNN). There is an 8% difference between the test and train values which does suggest a good fit, however, with a higher train accuracy, the model seems to suggest slight overfitting on the dataset. This can be expressed as unimpressive generalisation on unseen data and therefore, mitigating methods should be used such as increasing the sample size or further tuning the hyperparameters.

[2]Gómez, R., 2018. Understanding Categorical Cross-Entropy Loss. [online] Available at: <Link> [Accessed 29 November 2021].

(c) CNN: Different activation functions and learning rates

Table 1: Various activation function metrics

Activation function	Train Accuracy (%)	Test Accuracy (%)	Loss
tanh	96.27	89.29	110.6375
`Sigmoid` σ	97.78	90.37	12.1036
`Exponential Linear Unit (ELU)`	97.68	90.34	69.0121

Table 1 shows the different activation functions and the associated metrics. The `sigmoid` function outperforms the other activation functions, which is surprising as when the derivative of the functions is taken[3]:

$$\tanh'(x) = \operatorname{sech}^2(x) = \frac{2}{\exp(x) + \exp(-x))} \leq 1.0 \tag{2}$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \leq 0.25 \tag{3}$$

$$ELU'(z) = \begin{Bmatrix} 1 & z > 0 \\ \alpha \cdot e^z & z < 0 \end{Bmatrix} \tag{4}$$

It is evident from the above set of equations that the derivatives of tanh are larger than the other two activation functions. The objective of the activation function is to ensure an output that is suitable for the input of the next layer. The larger the derivative, the quicker the model will converge to the minimum (i.e., 0). "Convergence is usually faster if the average of each input variable over the training set is close to zero"[4]. The likely explanation for this behaviour can be attributed to the nature of the problem: classification. Since, the output needed to be bounded between the interval $[0, 1]$, the need to find the global minimum becomes easier when using the `sigmoid` function.

Table 2: Effect of the learning rate α

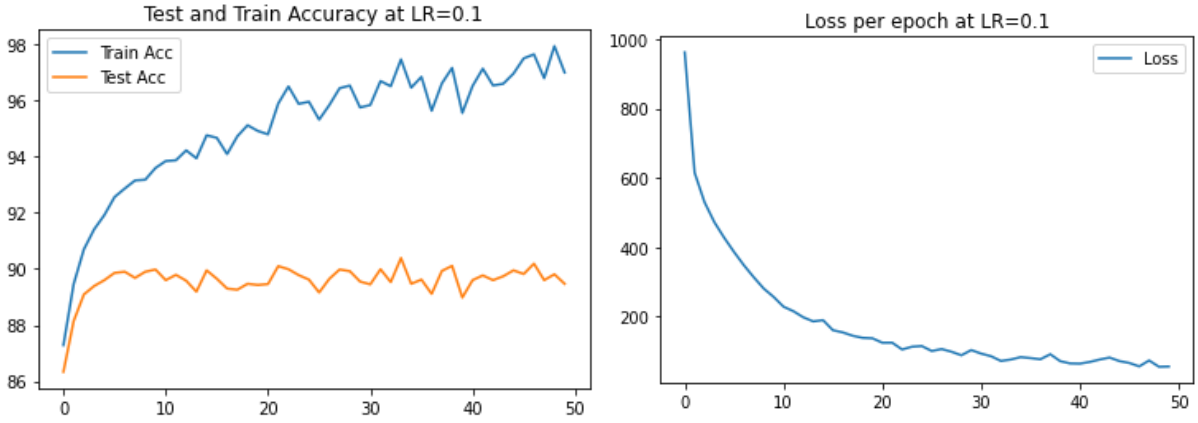
Learning rate	Train Accuracy (%)	Test Accuracy (%)	Loss
0.001	88.87	87.73	599.9731
0.1	97.01	89.47	56.0292
0.5	85.91	84.86	643.5673
1	10.00	10.00	4328.6516
10	10.00	10.00	45638.3168

Table 2 displays the effect of changing the learning rate, where a very small learning rate (0.001) results in less accuracy as the classifier does not converge to the minimum

[3]ML glossary documentation. 2017. Activation Functions. [online] Available at: <Link> [Accessed 3 December 2021].
[4]LeCun Y.A., Bottou L., Orr G.B., Müller KR. (2012) Efficient BackProp. In: Montavon G., Orr G.B., Müller KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 7700. Springer, Berlin, Heidelberg. <DOI>

during gradient descent. This is due to the very small *steps* that the classifier takes which increases computational time whilst fostering a suboptimal accuracy. Conversely, if the learning rate is increased > 0.5, the classification accuracy reduces drastically as the model completely misses the optimal weights and overshoots.

(d) CNN: Dropout



(a) Test and train accuracy

(b) Loss

Figure 2: Convolutional Neural Network metrics with dropout

```
Loss: 836.4302, Train Accuracy: 98.45%, Test Accuracy: 90.49%
```

The addition of the dropout value helps with alleviating overfitting by improving model generalisation capabilites. Fig. 2 shows the effect of adding a dropout rate of 0.3 to the second layer of the neural network, and when compared with the original results (Fig. 1), there is a 1.4%, 1.1%, and 1392.4% increase for the test, train and loss metrics, respectively. Evidently, the results suggest that the dropout value offsets model overfitting marginally, but is eclipsed by an immense increase in the overall loss value. Furthermore, several values were also tried where the loss value consistently kept increasing with a negligible increase in the train and test accuracy.