# COGS260 Assignment 4

Yan Sun
University of California, San Diego
La Jolla, California, USA
yas108@ucsd.edu

## Abstract

*Image classification is one of the most important research interests nowadays, for which Iris are well-known datasets for flower classification. Feed Forward Network have become well-known methods to finish these tasks. In addition, in order to obtain the correct caption for images, the object detection technique has been developed these years to get a high accurate result with high efficiency and the Recurrent Neural Network is also utilized these days to generate the expected text content. In this assignment, Feed Forward Neural Network is utilized to classify flowers data in IRIS dataset. Character level Recurrent Neural Network is used to generate required text content about Shakespeare's masterpiece. The trained models are able to generate reasonable text content. Finally, the You Only Look Once model is implemented to do the object detection work for different images. The pretrained model is able to finish the detection task for most of situations.*

## 1. Method

### 1.1. Feed Forward Neural Network

The Feed Forward Neural Network[2] is the one type of artificial neural network which could connect the neurons between different layers in the network and obtain the result in the forward propagation process without circle and update the parameters primarily via backpropagation[8]. This method could obtain wonderful performance on the regression or classification tasks on many datasets. The specific update method during the backpropagation process could be found from Equation 1 to Equation 8 for output layer and Equation from 9 to Equation 19 for hidden layer. All the derivations below are based on sigmoid activation function.

#### 1.1.1 Backpropagation for Output Layer

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial a_2}\frac{\partial a_2}{\partial w_2} \tag{1}$$

$$L = -\sum [tlogo + (1-t)log(1-o)] \tag{2}$$

$$\frac{\partial L}{\partial o} = \frac{o-t}{o(1-o)} \tag{3}$$

$$o = \frac{1}{1+exp(-a_2)} \tag{4}$$

$$\frac{\partial o}{\partial a_2} = o(1-o) \tag{5}$$

$$a_2 = w_2 z_1 + b_2 \tag{6}$$

$$\frac{\partial a_2}{\partial w_2} = z_1 \tag{7}$$

After applying chain rule to original equation and calculate each part in Equation 3, 5 and 7, the final expression could be obtained in Equation 8.

$$\frac{\partial L}{\partial w_2} = (o-t)z \tag{8}$$

#### 1.1.2 Backpropagation for Hidden Layer

The backpropagation process for hidden layers is different with output layers.

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial a_1}\frac{\partial a_1}{\partial w_1} \tag{9}$$

$$\frac{\partial L}{\partial a_1} = \frac{\partial L}{\partial a_2}\frac{\partial a_2}{\partial z_1}\frac{\partial z_1}{\partial a_1} \tag{10}$$

$$\frac{\partial L}{\partial a_2} = \frac{\partial L}{\partial o}\frac{\partial o}{\partial a_2} \tag{11}$$

From Equation 3 and 5, the expression of $\frac{\partial L}{\partial a_2}$ could be obtained.

$$\frac{\partial L}{\partial a_2} = (o-t) \tag{12}$$

From Equation 6, the expression of $\frac{\partial a_2}{\partial z_1}$ could be obtained.

$$\frac{\partial a_2}{\partial z_1} = w_2 \tag{13}$$

$$z_1 = \frac{1}{1 + exp(-a_1)} \tag{14}$$

$$\frac{\partial z_1}{\partial a_1} = z_1(1 - z_1) \tag{15}$$

From Equation 12, 13 and 15, the expression of $\frac{\partial L}{\partial a_1}$ could be obtained.

$$\frac{\partial L}{\partial a_1} = z_1(1 - z_1)(o - t)w_2 \tag{16}$$

$$a_1 = w_1 x + b_1 \tag{17}$$

$$\frac{\partial a_1}{\partial w_1} = x \tag{18}$$

From Equation 16 and 18, the equation for $\frac{\partial L}{\partial w_1}$ could be obtained.

$$\frac{\partial L}{\partial w_1} = z_1(1 - z_1)(o - t)w_2 x \tag{19}$$

## 1.2. Char RNN

Recurrent Neural Network (RNN) is one type of neural network that is able to make use of sequential information, whose output will be based on the previous computation of the network. RNN has shown to be succeed in many Natural Language Processing task, including the generation of text content[3].

Vanilla RNN is simple RNN structure that makes the hidden activation value from past computation plus the current input to be fed into the model. The detaile process is shown in Figure 1.

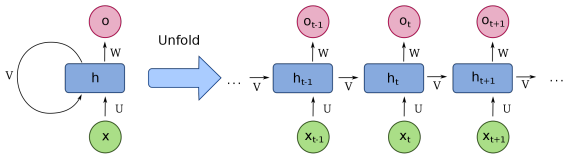



Figure 1: RNN Architecture[1]

With appropriate method of feeding character information and set the training label to be the exact next character in the text for each character. The RNN model can be trained to generate text with similar style with given start characters after training process. There also exist advanced RNN-based neural network architectures such as GRU[4] and LSTM[14].

As for the text generation process, the prediction is based on the probability distribution of softmax result. In the softmax process, the temperature coefficient is introduced, which is shown in Equation 20 for the variable T[15]. The original softmax is calculated with T=1. If T is too small, then the probability difference between different categories is amplified and the predictor will be more determined, which will lead to more duplicate text content generated. If T is too large, then the probability difference between different categories is decreased and the predictor will not be determined and the text generated will tend to be in a mess. In this way, it is significant to find an appropriate value of temperature coefficient.

$$P(y = j|x) = \frac{e^{\frac{x^T w_j}{T}}}{\sum_k^K e^{\frac{x^T w_k}{T}}} \tag{20}$$

## 1.3. Object Detection

Object detection is one of the most significant topics in computer vision nowadays, which aims at detecting the preferred object in the given image. In addition, the object detection technique could be further utilized into other advanced tasks such as face recognition and Visual Search Engine.

Nowadays there exist many techniques for this topic, among which the well-known one is You Only Look Once (YOLO) method[10]. The RCNN[7], Fast RCNN[6] and Faster RCNN[13] are two stages methods since their process are separated for region proposal and classification or regression, respectively. YOLO and SSD[9] are one stage method since they only need one neural network structure to predict the location of object in the image and predict the category the object belongs to.

Different from the sliding windows method for region proposal in RCNN, YOLO utilizes the convolutional process in Convolutional Neural Network to replace this process and feed them directly into the model. Specifically, all the images will be resized to the same size then be divided into desired grid. For each unit part of the image, the bounding box and confidence score will be calculated based on the location of objects and IOU value. The summarized loss function is the weighted sum of localization error and classification error. The original YOLO model structure is shown in Figure 2.

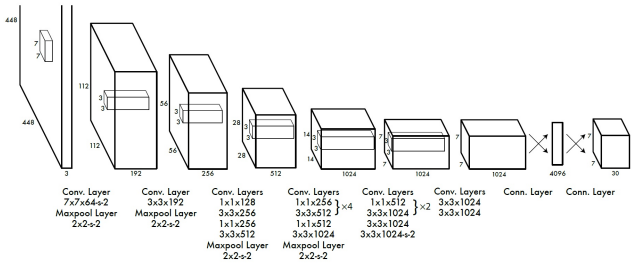



Figure 2: YOLO version1 model Architecture[10]

## 2. Experiment

### 2.1. Feed Forward Neural Network

Two classes data (Iris-versicolor and Iris-virginica) in Iris dataset is extracted for the experiment of this part. The feed forward neural network built in this part is implemented without any deep learning framework shown in Figure 3. The loss function is cross entropy function and the activation function is set to be sigmoid function. Initially, all the weights is generated based on the random value between -0.5 and 0.5. During the training process, the learning rate is 0.002 and maximum iteration is 10000. After training process, the loss curve and error rate curve is included in Figure 4 and 5, respectively. Finally, the training loss value is 0.42, the training error 0 and testing error is 0.13.

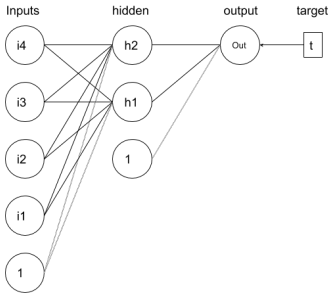



Figure 3: Feed Forward Neural Network Architecture

### 2.2. Char RNN

Vanilla RNN is trained for Shakespeare masterpiece text content in this part. The sample part of given text data is in Figure 6.

One layer and two layers model are implemented. The

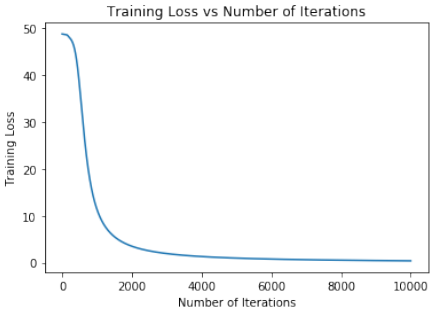



Figure 4: Feed Forward Neural Network Training Loss Curve

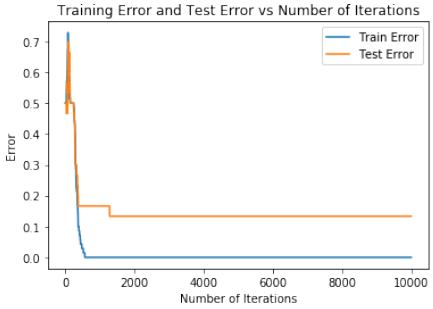



Figure 5: Feed Forward Neural Network Training Error Curve

```
First Citizen:
Before we proceed any further, hear me speak.

All:
Speak, speak.

First Citizen:
You are all resolved rather to die than to famish?

All:
Resolved. resolved.

First Citizen:
First, you know Caius Marcius is chief enemy to the people.

All:
We know't, we know't.
```

Figure 6: Sample Shakespeare text content

last 5000 characters in the given text content is separated to be the validation data and the remaining data is training data. The hidden dimension for each layer is 100, the weights are initialized to be zero, the optimizer is Adam algorithm and the learning rate is 0.005. As for the data feeding process, one chunk characters of 200 length are randomly from training data and fed into model for training each epoch. Training for 100 chunks is defined as one epoch.

The input representation is generated via identify each character through a dictionary and embedding the input

characters into the same dimension with hidden dimension. Then the new embedded vector will be fed into the RNN model. The prediction result is selected through the softmax layer. The label for each character is the next character in the given text. The loss curve is in Figure 7 and 8 for one layer model and two layers model, respectively. The corresponded statistical result for two models with 50 epochs is in Table 1.

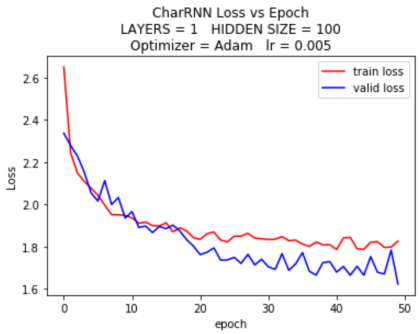



Figure 7: One Layer Char RNN Loss Curve

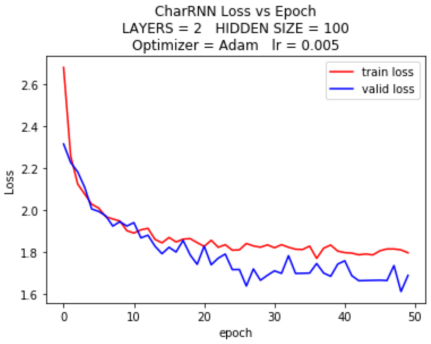



Figure 8: Two Layers Char RNN Loss Curvet

| Model | Train Loss | Valid Loss | Epoch for 1.8 Loss | Temperature Coefficient |
|---|---|---|---|---|
| One-Layer | 1.82 | 1.62 | 21 | 0.5 |
| Two-Layers | 1.79 | 1.78 | 15 | 0.7 |

Table 1: CharRNN Result

All the models are trained for 50 epochs for text generation. Specifically, the start text is 'CORIOLANUS:' and the prediction character length is set to be 5000. After tuning parameter, the appropriate temperature coefficient is 0.5 for one layer model and 0.7 for two layers model. The sample generated text for one layer model and two layers model are in Figure 9 and 10.

```
CORIOLANUS:
Which the groshiness stay the grace of the quent and say late speak of of go it is the at of the not the dost a the consteep an
d shouse at she the good and how sun a son shrown to are shall his souls and say a son hath so stalt:

KING RICHARD III:
I shall not the worth shipot there are the shouse the still of the father shirly be and must good and with have it be thee for
have a speak of the seen hath liven gone a give shouse have so the so with you the good, and morts go thy world and he disture
at the pity the say and south lord, and so be we the good that the gosse:
Words have the strong a should should the he be of this son the son and slail the says in her a manion looks and worster.

KING RICHARD III:
What the be him shall good and all the she say shall him.

KING RICHOLIO:
Ge a good blood reath good the stain the be not the latter the some shall the say in your so soned have stay the quend good she
grace.
```

Figure 9: One Layer Char RNN Text Generation Sample

```
CORIOLANUS:
Now had the gons as will, but he broud shall and, not names his callaw of repolk, and speal.

ROMEO:
Hall than she the regue comes hand the geast be the resentleman:
Ef thou planged of retiences no coming strounce,
He the westand cangues him, for strusharl did centers,
There, not man arull the dights earth the fair of saw I the king,
That there that see I disson Bursand the sweet see ceoreman shore your purpteeth should surthing by Romenter hold so as malan i
s shall sore his which have of ming not dis, but the give here,
That shward the death destreen croving and this cannot to that where, latuers are sund and that may say.

FRIARET:
Havour man the percountle will mad searst stords, and worp,
Or not who provonds charkn that sees no spoll all rasher'd, parth 'trursested may from your cray shall breather.
```

Figure 10: Two Layers Char RNN Text Generation Sample

## 2.3. Object Detection

### 2.3.1 Scale Invariance

In this part, one dog image is selected to be scaled and created for 4 x 4, 8 x 8 and 16 x 16 image collages. Then all the collages is resized to be 1000 x 1000. For each collage, the YOLO test code is applied for detecting dog in the image. The detection result is in Figure 11.

(a) *4x4.* (b) *8x8.* (c) *16x16.*

Figure 11: Dog Image Collages Detection

### 2.3.2 Rotational Invariance

As for the same image in Section 2.3.1, the image is rotated for a circle with 30 degree intervals. Then the same YOLO model is applied to these rotated images. All the results are shown in Figure 12. The corresponded detection probability result is in Table 2.

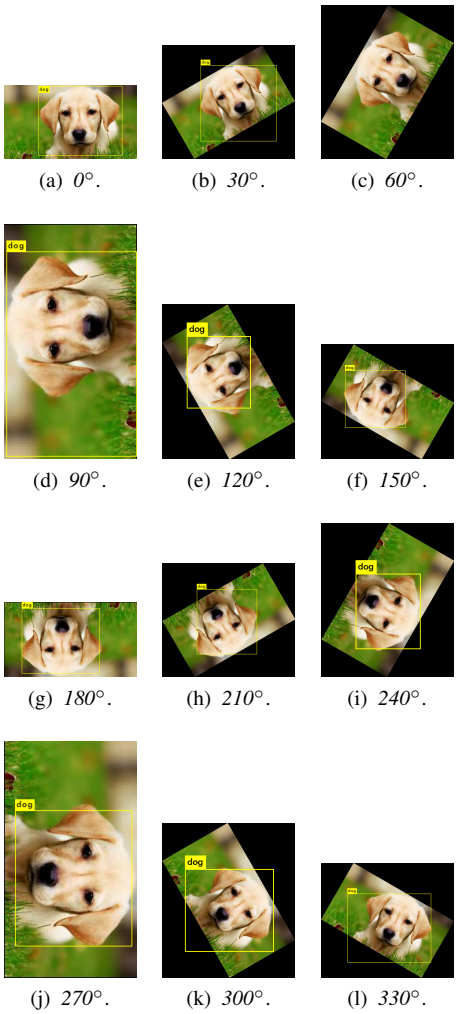
| Rotate(°) | 0 | 30 | 60 | 90 | 120 | 150 | 180 | 210 | 240 | 270 | 300 | 330 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Probability(%) | 97 | 99 | 0 | 76 | 88 | 98 | 91 | 75 | 92 | 96 | 99 | 98 |

Table 2: Rotated Image Detection Result

# 3. Discussion

## 3.1. Feed Forward Neural Network

The experiment of this part is implemented based on Numpy and there are only 70 samples in training set and 30



(a) *0°*. (b) *30°*. (c) *60°*.

(d) *90°*. (e) *120°*. (f) *150°*.

(g) *180°*. (h) *210°*. (i) *240°*.

(j) *270°*. (k) *300°*. (l) *330°*.

Figure 12: Rotate Dog Image Detection

samples in testing set. The small size of given dataset also contributes to the phenomena that the error rate for training set goes to zero while the error rate for testing set goes around 13%. In this way, although the loss curve in Figure 4 looked smoothly, the error rate curve in Figure 5 is jagged due to the small quantity of the data sample.

## 3.2. Char RNN

The one layer and two layers RNN model have different efficiency and effectiveness for text generation. Through Table 1, it could be observed that after 50 epochs, the one layer model and two layers model has similar loss value for training and validation set. However, the two-layers model could converge faster since it uses less epochs to decrease its loss to 1.8 or lower. In addition, these two models have different temperature coefficient to generate the decent text content. The one layer model requires lower temperature coefficient to be more determined to generate reasonable text content while the two layers model has more ability to determine which character to produce in the next step so it requires higher temperature coefficient to decrease the duplication phenomenon in its generation result.

## 3.3. Object Detection

### 3.3.1 Scale Invariance

YOLO model in this part of experiment does not perform perfectly (Figure 11). For 4 x 4 collages, its detection result is perfect since all the dogs in the 4 x 4 collages have been detected. However, in the 8 x 8 collages, only a portion of dogs in the picture is detected correctly while other dogs are not detected or classified as teddy bear. In 16 x 16 collages, things become worse since none of the dogs in the collage is detected.

These kind of phenomenon shows the disadvantage of YOLO model, which is also mentioned in the YOLO original paper[10]. This was due to the loss of fine-grained features as the layers downsampled the input. In order to solve this problem, there exist several versions solutions. For instance, the YOLO version 3 model solved this problem by concatenating the upsampled layers with the previous layers help preserve the fine grained features which help in detecting small objects[12]. In addition, recently there is one alternative solution named You Only Look Twice (YOLT)[5], which upsamples through a sliding window to look for small, densely packed objects and runs an ensemble of detectors at multiple scales.

### 3.3.2 Rotational Invariance

The result in Table 2 and Figure 12 shows that YOLO model is able to detect the dog in selected image for most of the rotation angle with high prediction probability. However, for the rotation angle 120° and 210°, the detected object probability goes down to 70% and when the rotation angle is 60°, the dog in the image even cannot be detected.

One possible reason could be the input pictures with rotation angles. In order to save the whole image, there exist the pure black area remained at the corner of each rotated image (Figure 12). These inputed black areas may also disqualify the performance of YOLO model.

The possible solution for these phenomenon caused by rotation is to extend the dataset for training model by adding images with different rotation angle, saturation and exposure shifts[11].

# 4. Conclusion

In this assignment, Feed Forward Neural Network, Char RNN and YOLO object detection are implemented for specific task. More optimization remained for these methods,

especially for the more accurate text generation of Char RNN models and for the more accurate and more efficient YOLO object detection models.

## References


[1] $https://en.wikipedia.org/wiki/recurrent\_neural\_network$. *Wikipedia*. 2

[2] G. Bebis and M. Georgiopoulos. Feed-forward neural networks. *IEEE Potentials*, 13(4):27–31, 1994. 1

[3] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 2

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2

[5] A. V. Etten. You only look twice: Rapid multi-scale object detection in satellite imagery. *arXiv:1805.09512*, 2018. 5

[6] R. Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015. 2

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[8] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992. 1

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2, 3, 5

[11] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017. 5

[12] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 5

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[14] H. Sak, A. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014. 2

[15] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. 2