Thomas J. Watson School of Engineering & Applied Science
Department of Electrical & Computer Engineering

Neural Network & Deep Learning (EECE680C)

Homework_4

Solutions

Submitted By

Alem Haddush Fitwi
afitwi1@binghamton.edu
Graduate Student

Submitted To

Dr. Xiaohua Li
Associate Professor
xli@binghamton.edu

Submission Date

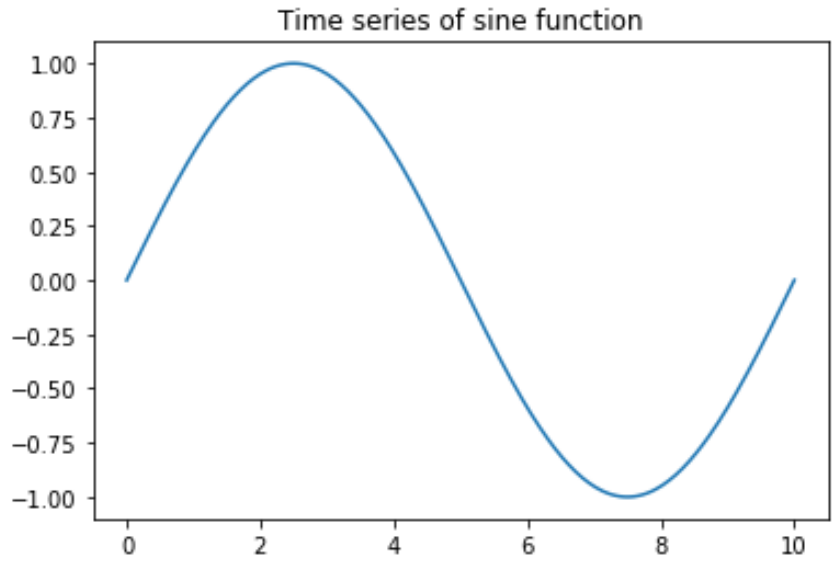
26 April 2018

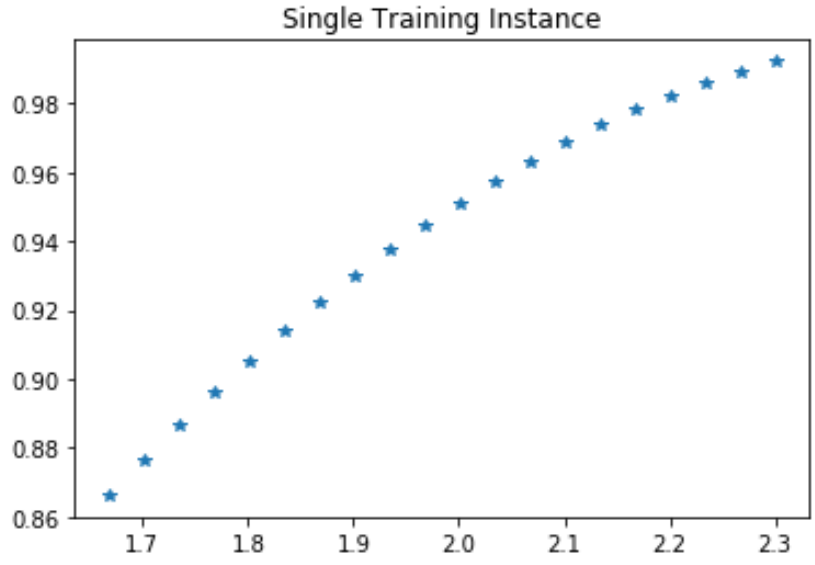
Binghamton, New York

The test results of problem #3, the programming problem, are hereunder presented:

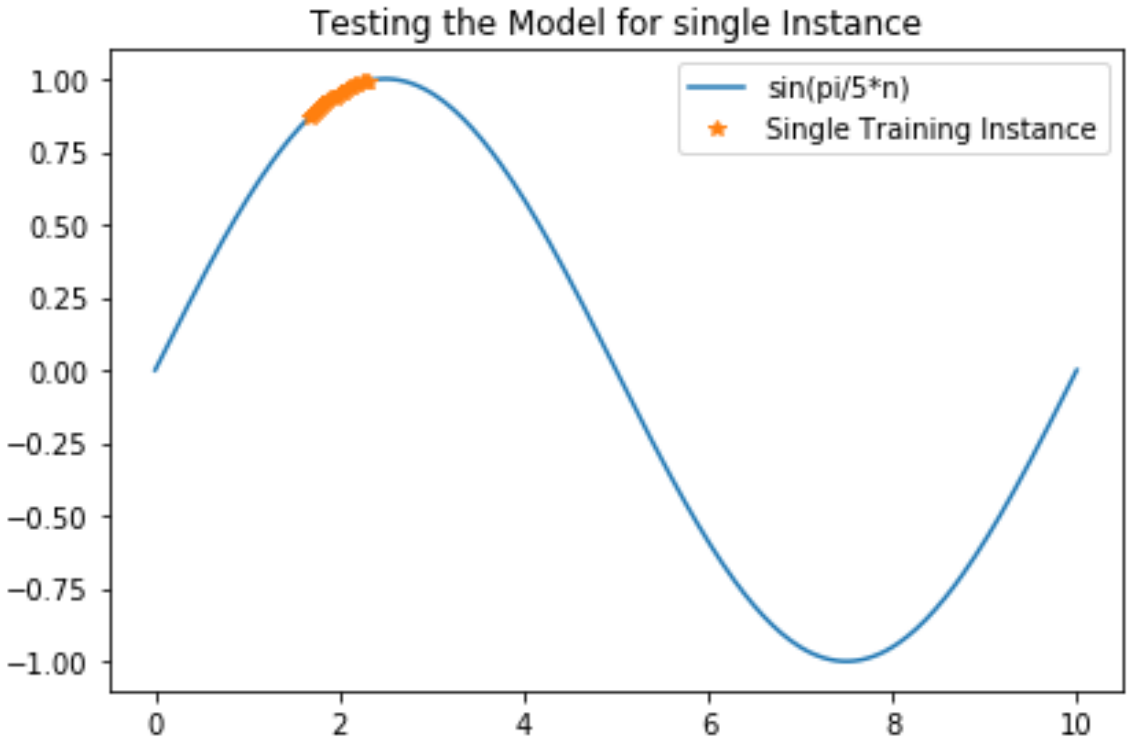
sin(0.2*np.pi*n) for n=0 upto 299 → 300 points
SineSeries (num_points,xmin,xmax)=SineSeries(300,0,10)
Num_samples=20 # number of samples
Iteration#=2200
Single training instance before training was conducted

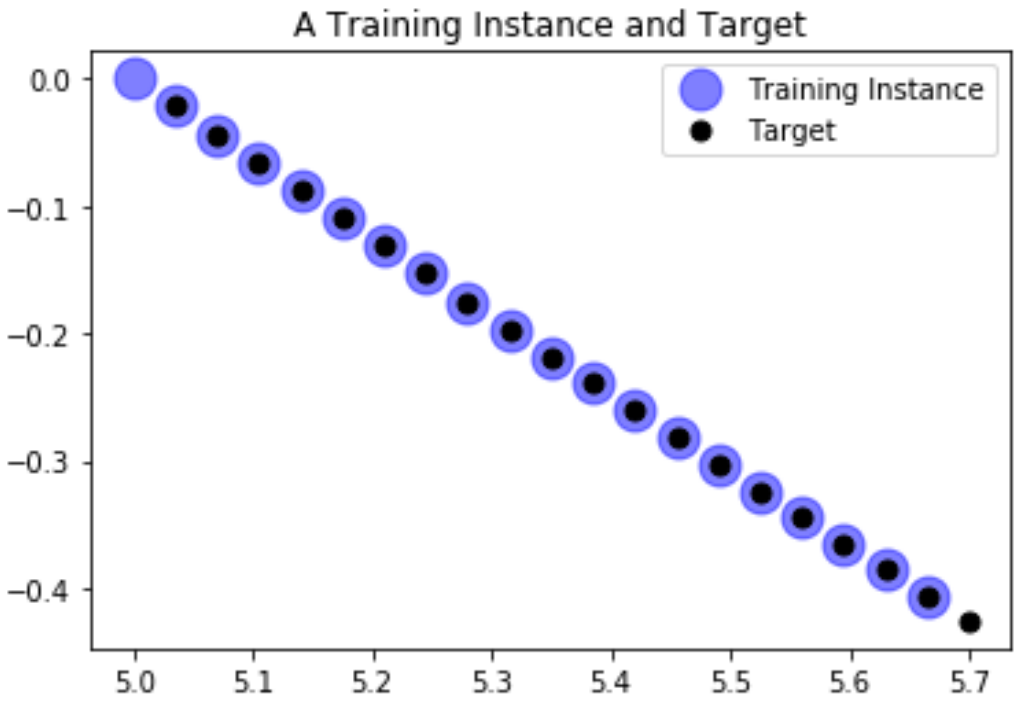
```
In [1]: runfile('/home/alem/LSTMasst.py', wdir='/home/alem')
================================================================================
************Preliminary processes and tests before data training******
================================================================================
```





Testing the model for single instance of training data, and generating training instance and the target



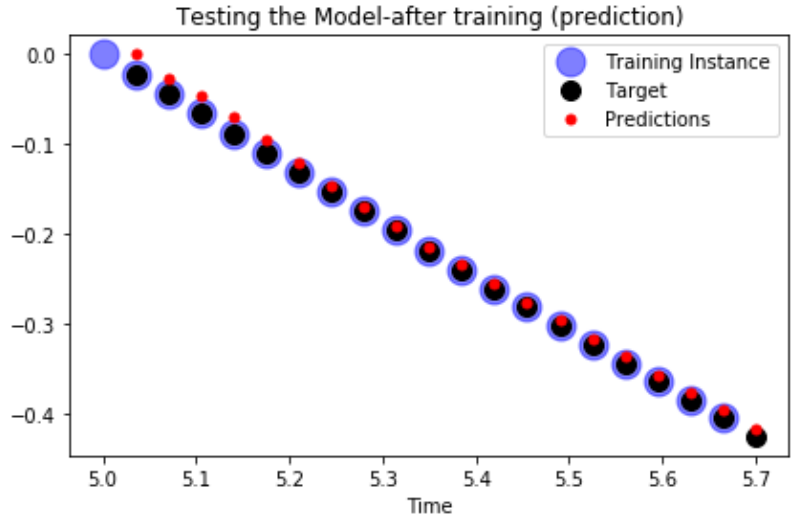


Training the model ….

```
======================================================================
..................Training is on progress, please wait for a while!
======================================================================
Iteration        0      Mean Square Error 0.875735
Iteration        100    Mean Square Error 0.0942191
Iteration        200    Mean Square Error 0.00917639
Iteration        300    Mean Square Error 0.0156673
Iteration        400    Mean Square Error 0.00335173
Iteration        500    Mean Square Error 0.00528435
Iteration        600    Mean Square Error 0.000151139
Iteration        700    Mean Square Error 0.000209462
Iteration        800    Mean Square Error 0.00440594
Iteration        900    Mean Square Error 9.25027e-05
Iteration        1000   Mean Square Error 0.000559856
Iteration        1100   Mean Square Error 0.000237966
Iteration        1200   Mean Square Error 0.000245692
Iteration        1300   Mean Square Error 0.000100152
Iteration        1400   Mean Square Error 0.000469705
Iteration        1500   Mean Square Error 0.000775393
Iteration        1600   Mean Square Error 8.64337e-05
Iteration        1700   Mean Square Error 0.000442672
Iteration        1800   Mean Square Error 9.54112e-05
Iteration        1900   Mean Square Error 0.000102052
Iteration        2000   Mean Square Error 0.000138961
Iteration        2100   Mean Square Error 5.67146e-05
```

Prediction and Model Testing

```
======================================================================
******Prediction and model testing afterconducting data training******
======================================================================
INFO:tensorflow:Restoring parameters from ./lstm_series_model_codelog
```



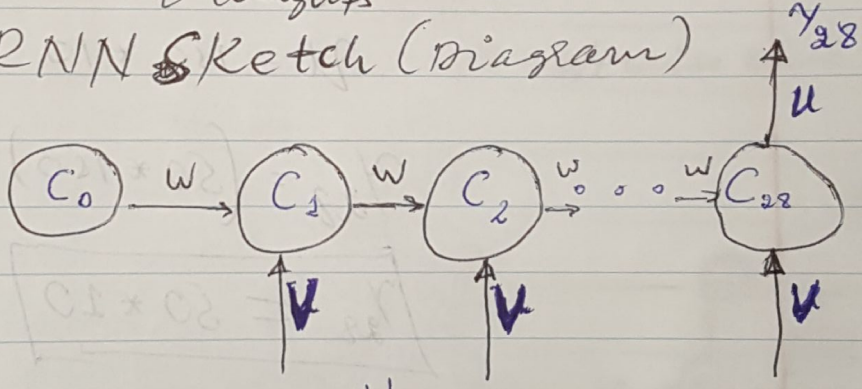

```
======================================================================
****************************End of Program****************************
======================================================================
```

① Classifying MNIST digits using RNN instead of CNN.

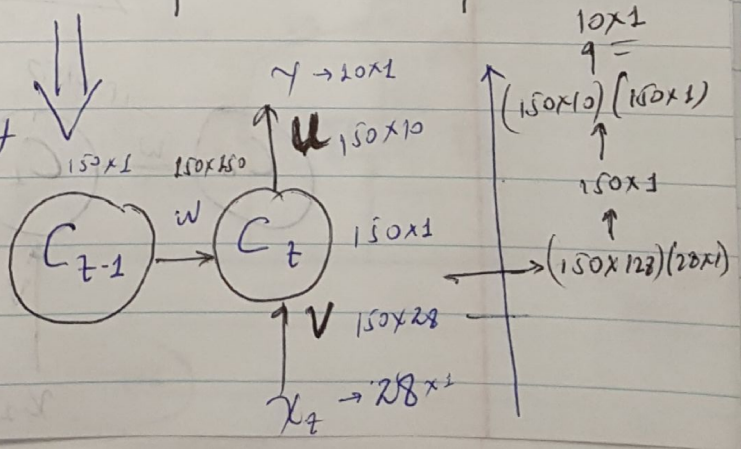
Sol^{n}:

- Given:
  - # of layers = 28
  - X_t has dimension 28
  - Single output Y_{28} at last layer
  - Y_{28} → 10 dim → one-hot vector
  - # of neurons = 150 as RNN states
- Sketch the RNN & specify the dimensions of all the tensors
  - Inputs
  - Outputs
  - States
  - weights

- RNN Sketch (Diagram)



Analyzing it



X_{Dim} = Batch_size * Layers * num_inputs

= 50 * 28 * 28

permutation of layers & batch size

X_{Dim} = 28 x 50 x 28

$C_t = f(X_t V + C_{t-1} W)$

$C_t = 50 * 150$

$V = 28 * 150$

$C_{t-1} = 50 * 150$

$W = 150 * 150$

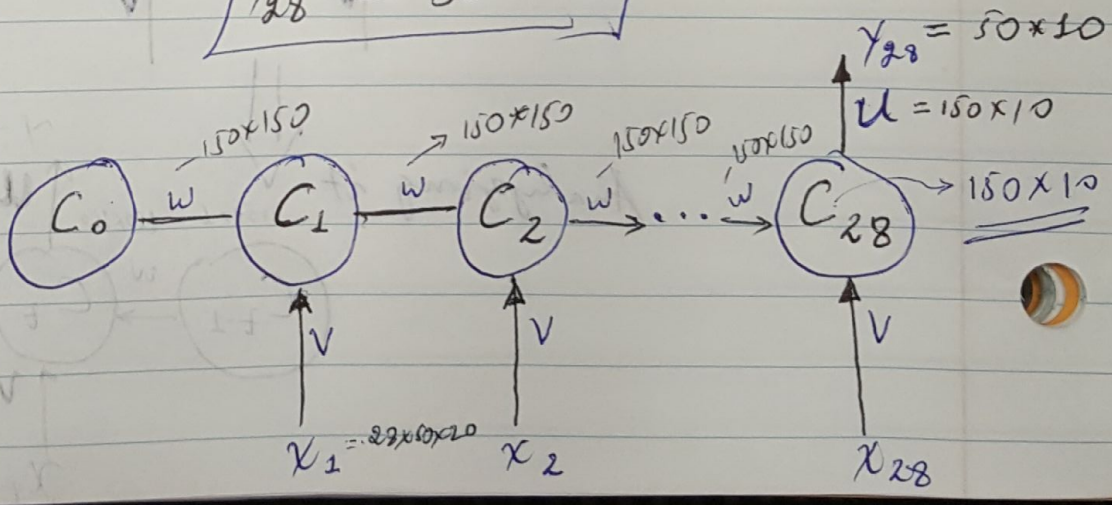
$Y_{28} = C_{28} * U$

$\Rightarrow C_{28} = 50 * 150$

$U = 150 * 10$

$Y_{28} = (50 * 150)(150 * 10)$

$Y_{28} = 50 * 10$



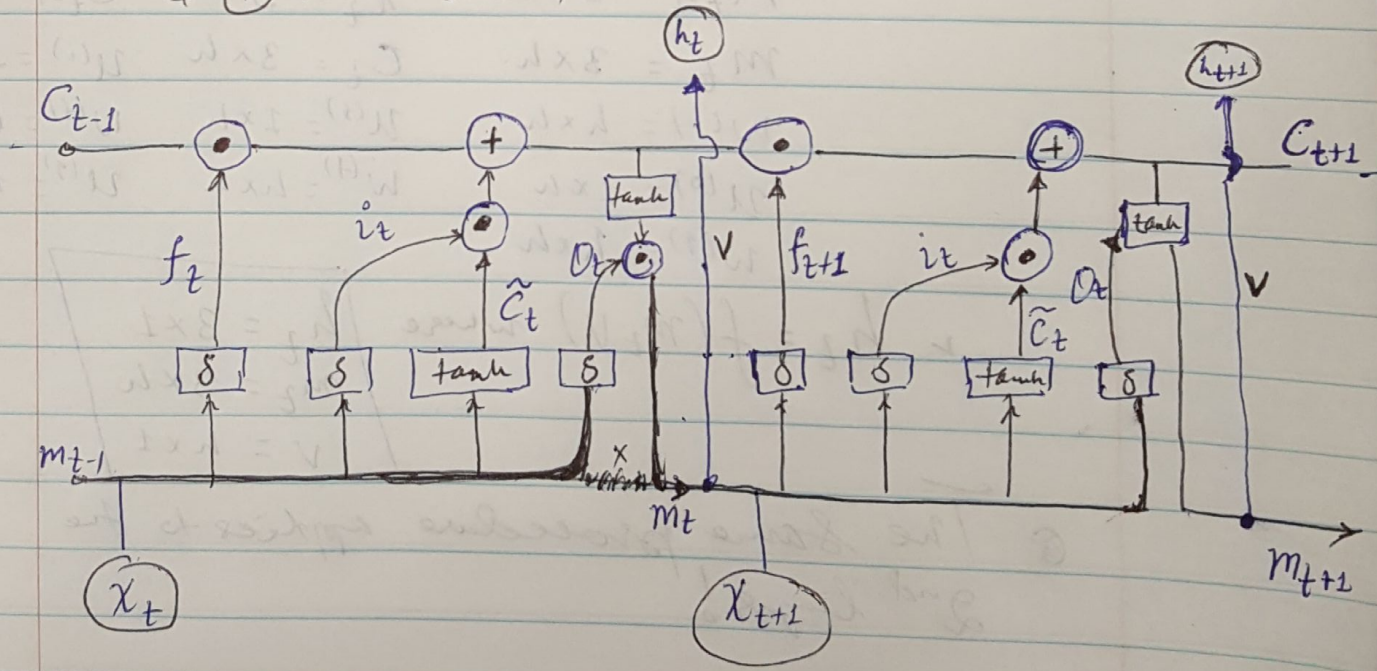
(2) LSTM Design for [0, 1, 2, 3, 4, 5, 6] where the previous integer is used to predict the next one. Specify the dimensions of all tensors.

Sol'n:

- Given: [0, 1, 2, 3, 4, 5, 6]
  - Batch size = 3
  - # of steps = 2
- Input_size = Batch_size × num_steps
  = 3×2
- Input & output in matrix forms

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 2 \\ 2 & 3 \end{bmatrix} \quad \& \quad Y = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \end{bmatrix}$$

- $X_t = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$ (X_t ↓ x_1) & $X_{t+1} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ (X_{t+1} ↓ x_2)
- LSTM Design → 2 layers
- In the figure below, ⊙ = o/t wise multiplication & ⊕ means addition.



- corresponding Equations

- $f_t = \delta(x_t U^{(f)} + m_{t-1} W^{(f)})$
- $i_t = \delta(x_t U^{(i)} + m_{t-1} W^{(i)})$
- $O_t = \delta(x_t^{(o)} + m_{t-1} W^{(o)})$
- $\tilde{C}_t = \tanh(x_t U^{(g)} + m_{t-1} W^{(g)})$
- $C_t = C_{t-1} \odot f + \tilde{C}_t \odot i_t$
- $m_t = \tanh(C_t) \odot O_t$

- Assumption: Let h = # of hidden neurons

Hence,

$m_{t-1} = 3 \times h$	$x_t = 3 \times 1$	$C_{t-1} = 3 \times h$	
$m_t = 3 \times h$	$C_t = 3 \times h$	$U^{(i)} = 1 \times h$	
$W^{(i)} = h \times h$	$U^{(f)} = 1 \times h$	$W^{(f)} = h \times h$	
$U^{(o)} = 1 \times h$	$W^{(f)} = h \times h$	$U^{(g)} = 1 \times h$	
$W^{(g)} = h \times h$			

- $h_t = f(m_t V)$ where $h_t = 3 \times 1$, $m_t = 3 \times h$, $V = h \times 1$

- The same procedure applies to the 2nd layer!