Gated Recurrent Neural Networks

CE/CZ4042 – Tutorial 9

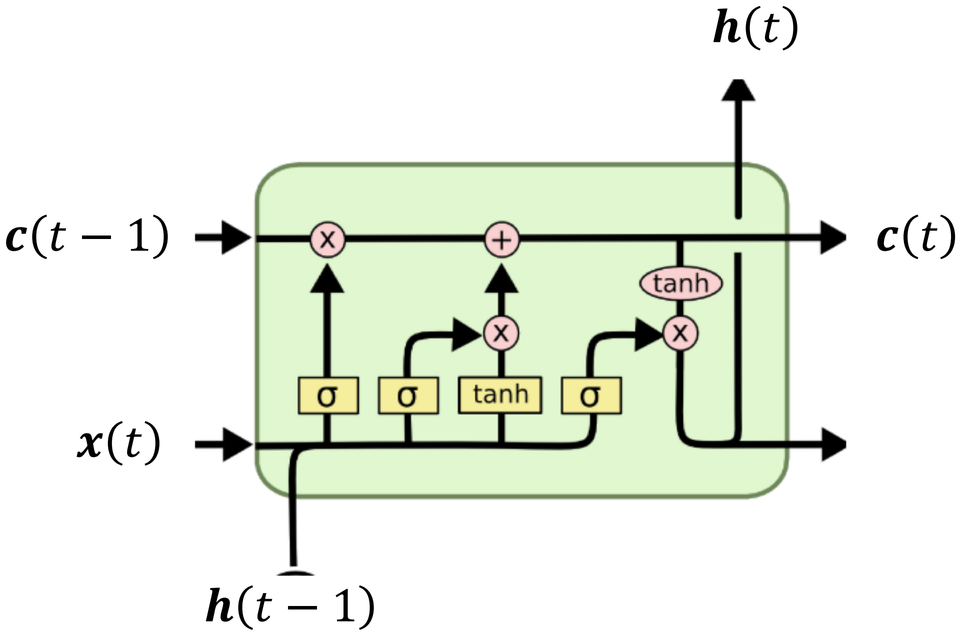
1. Design an LSTM layer with 10 units to map the following input and output sequences:

Input $\boldsymbol{x}$	Output $\boldsymbol{y}$
$(1\quad 2\quad 5\quad 6)$	$(1\quad 3\quad 7\quad 11)$
$(5\quad 7\quad 7\quad 8)$	$(5\quad 12\quad 14\quad 15)$
$(3\quad 4\quad 5\quad 7)$	$(3\quad 7\quad 9\quad 12)$

Plot the learning curves at a rate $\alpha = 0.001$.

Find the output sequences for the following input sequences:

$$(1\quad 2\quad 3\quad 4)$$
$$(4\quad 5\quad 6\quad 7)$$

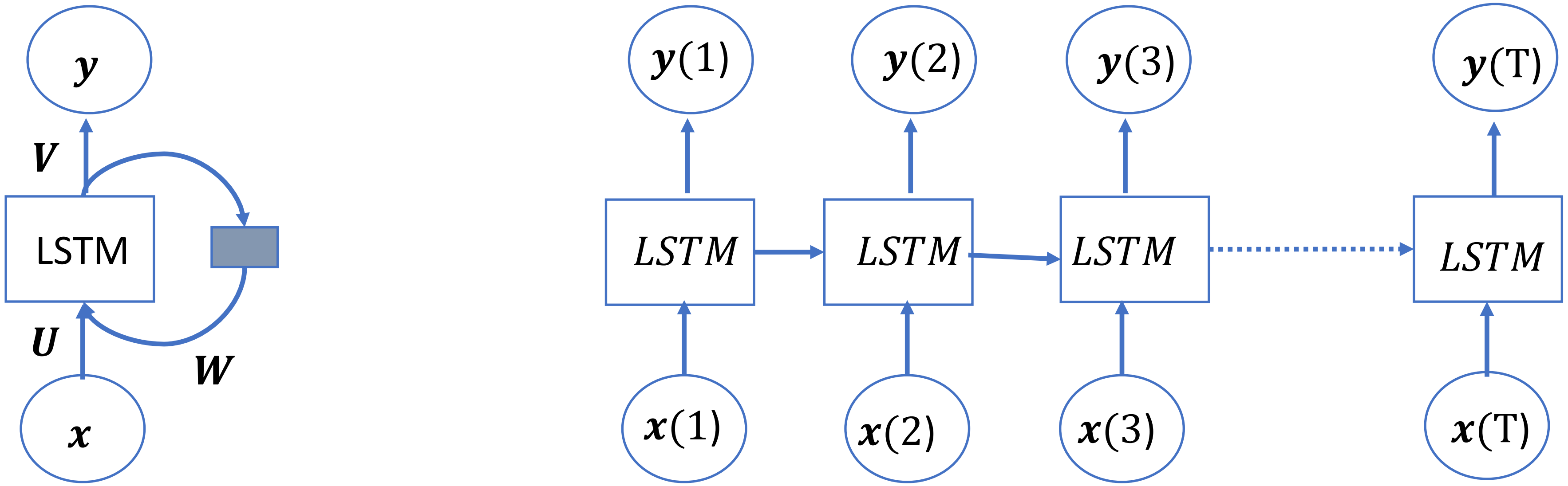


$\boldsymbol{h}(t)$
$\boldsymbol{c}(t-1)$
$\boldsymbol{c}(t)$
tanh
σ
σ
tanh
σ
$\boldsymbol{x}(t)$
$\boldsymbol{h}(t-1)$

Training sequences:

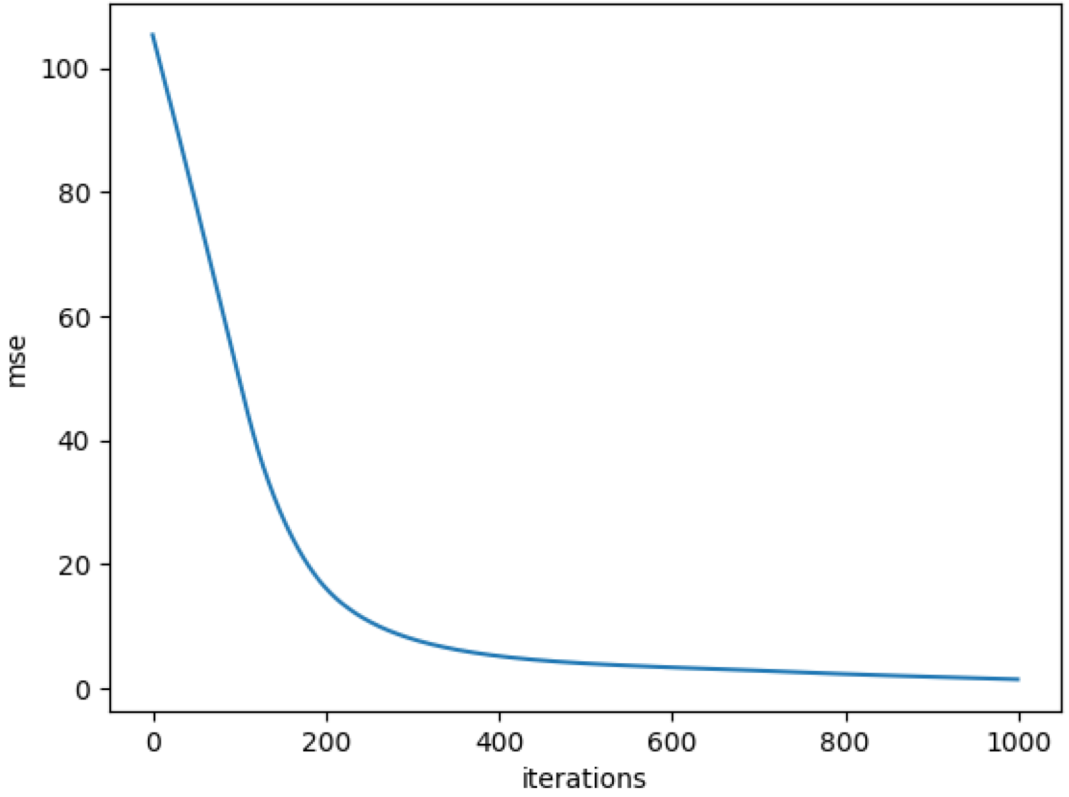
Input x	**Output y**
$(1 \quad 2 \quad 5 \quad 6)$	$(1 \quad 3 \quad 7 \quad 11)$
$(5 \quad 7 \quad 7 \quad 8)$	$(5 \quad 12 \quad 14 \quad 15)$
$(3 \quad 4 \quad 5 \quad 7)$	$(3 \quad 7 \quad 9 \quad 12)$

Sequence-to-sequence mapping



The output layer is a linear/softmax layer

```
cell = tf.nn.rnn_cell.LSTMCell(hidden_dim, reuse=tf.get_variable_scope().reuse)
 outputs, states = tf.nn.dynamic_rnn(cell, x, dtype=tf.float32)

num_examples = tf.shape(x)[0]
W_repeated = tf.tile(tf.expand_dims(W_out, 0), [num_examples, 1, 1])
out = tf.matmul(outputs, W_repeated) + b_out
out = tf.squeeze(out)


cost = tf.reduce_mean(tf.square(out- y))
train_op = tf.train.AdamOptimizer().minimize(cost)
```



mse
100
80
60
40
20
0
0
200
400
600
800
1000
iterations

At the end of training:

Input x	Output y
$(1 \quad 2 \quad \mathbf{3} \quad \mathbf{4})$	$(0.38 \quad 2.87 \quad 6.08 \quad 8.46)$
$(\mathbf{4} \quad \mathbf{5} \quad \mathbf{6} \quad \mathbf{7})$	$(3.85 \quad 8.94 \quad 11.39 \quad 12.17)$

2. The following csv file contains the monthly totals in thousands of airline passengers from January 1949 to December 1960:

   international-airline-passengers.csv

Starting from the first month, split the data into train data and test data at a ratio of [0.8, 0.2] Using the training data, design an LSTM layer to predict the monthly international airline passengers by using the monthly international airline passengers of previous 8 months.

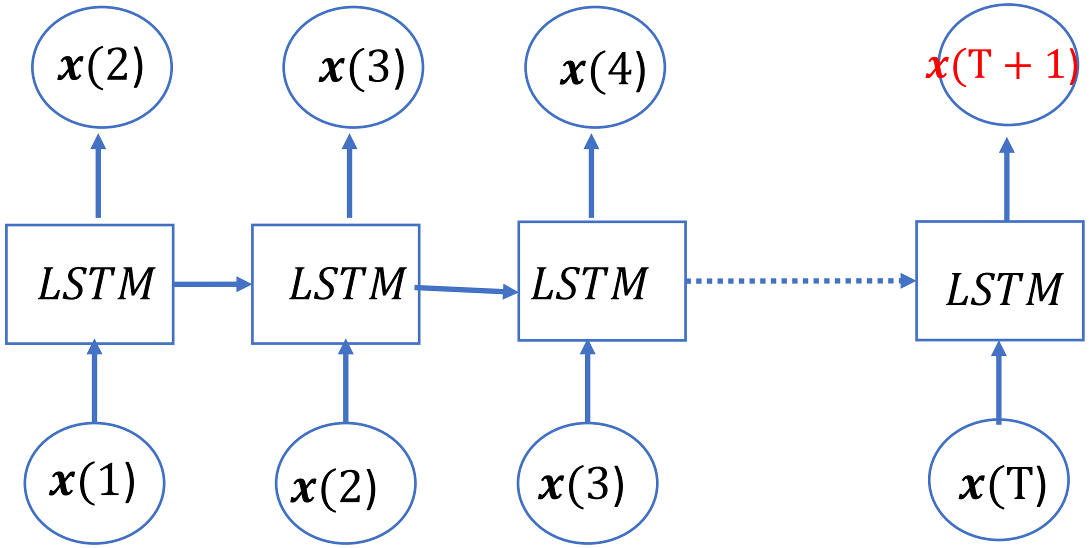
Using the LSTM model,

a) Find the predictions on test data partition.
b) Extend the predictions beyond the last month of training data by using last prediction on the training data.

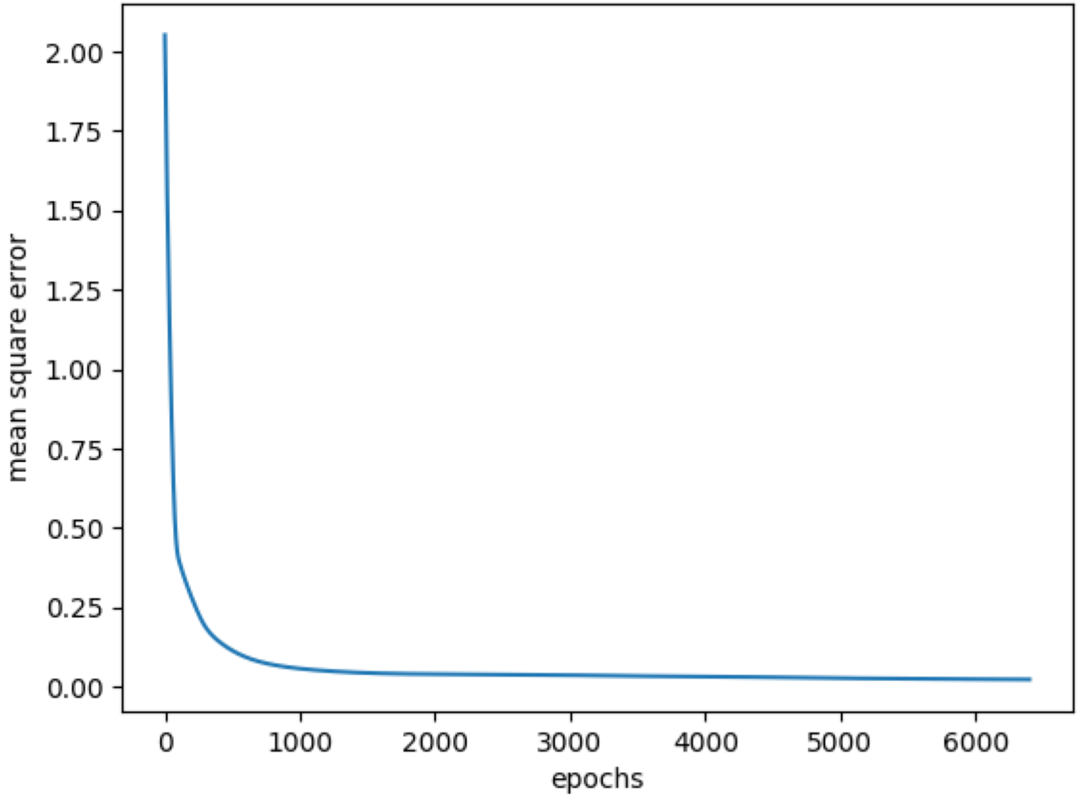
Plot the predicted values along with the trained values.

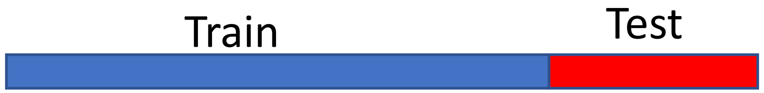
```
import csv

csvfile = open('international-airline-passengers.csv')
csvreader = csv.reader(csvfile)

months = [row[0] for row in csvreader if len(row) > 0]
demand = [float(row[1]) for row in csvreader if len(row) > 0]
```

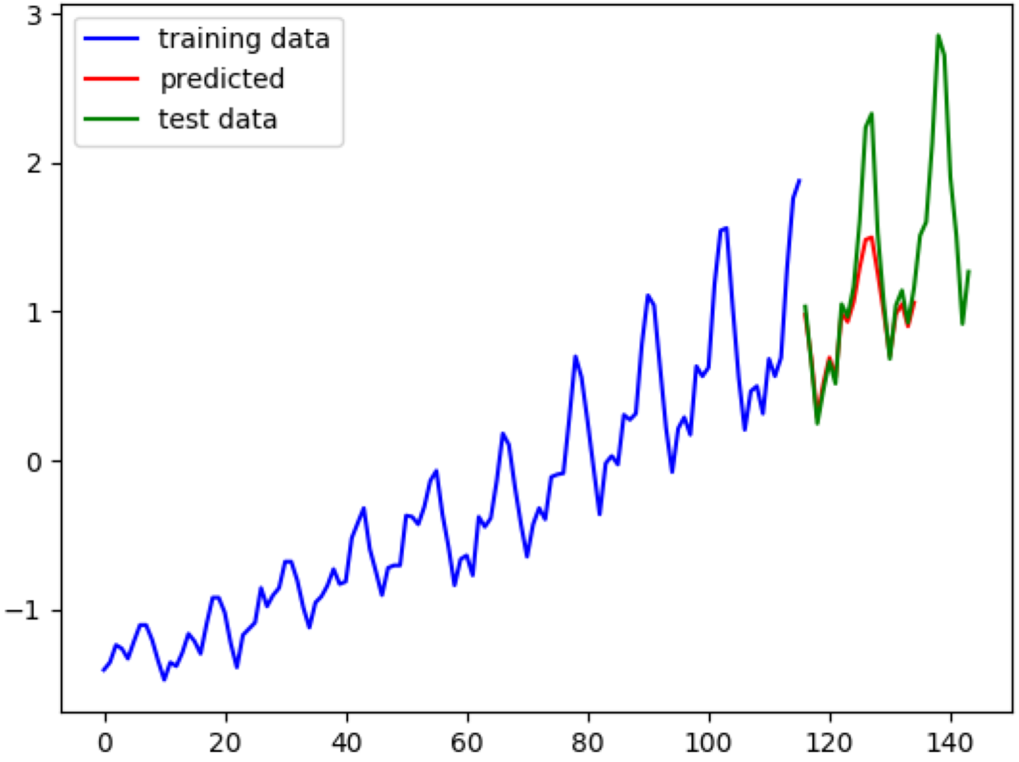
```
1949-01 : 112.0
1949-02 : 118.0
1949-03 : 132.0
1949-04 : 129.0
1949-05 : 121.0
1949-06 : 135.0
1949-07 : 148.0
1949-08 : 148.0
1949-09 : 136.0
1949-10 : 119.0
```

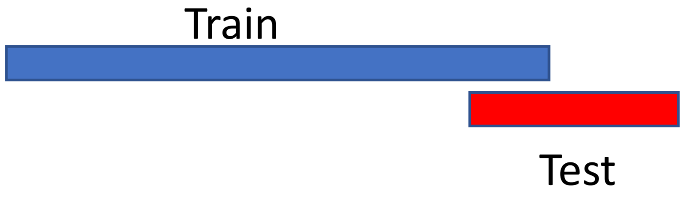

x(2)
x(3)
x(4)
x(T + 1)
LSTM
LSTM
LSTM
LSTM
x(1)
x(2)
x(3)
x(T)

Batch training of training data



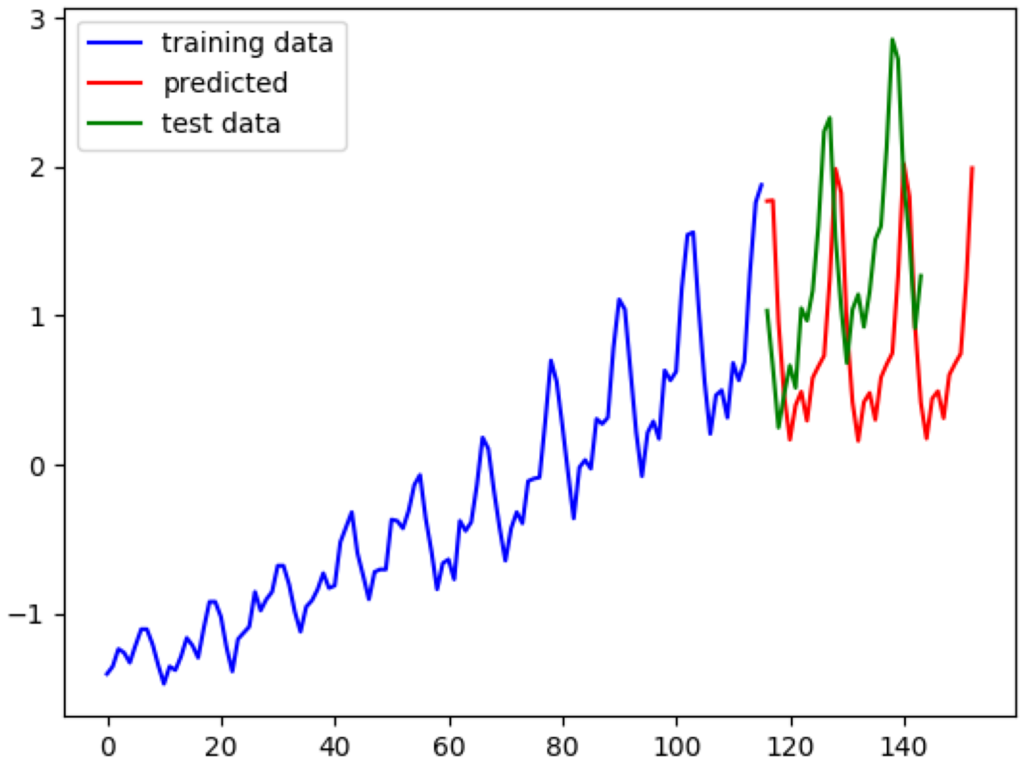

Train
Test


training data
predicted
test data
3
2
1
0
−1
0
20
40
60
80
100
120
140



Train

Test


training data
predicted
test data
3
2
1
0
−1
0
20
40
60
80
100
120
140

3. Design a character RNN layer with 10 GRU units to learn the sentiments about a movie, given in table 1.

   Convert the input text into character ids and set the maximum text length to be 40. Train the network using gradient descent learning with the Adam optimizer and at a learning rate $\alpha = 0.001$. Use dropouts at the hidden layer of GRU at a probability $p = 0.7$.

   Plot the cost and the accuracies against epochs during training.

Input	*Sentiment*
I did not like the movie	negative
The movie was not good	negative
I watched the movie with great interest	positive
I have seen better movies	negative
Good to see that movie	positive
I am not a fan of movies	negative
I liked the movie great	positive
The movie was of interest to me	positive
I thought they could show interesting scenes	negative
The movie did not have good scenes	negative
Family did not like the movie at all	negative

After training, find the likelihoods of the sentiments of the following statements:

The movie was not interesting to me
I liked the movie with great interest

Input:
‘Good to see that movie’

Translated input as a list of ids

```
[ 7 14 14  4  0 17 14  0 16  5  5  0 17  8  1 17  0 12 14 19  9  5  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
```

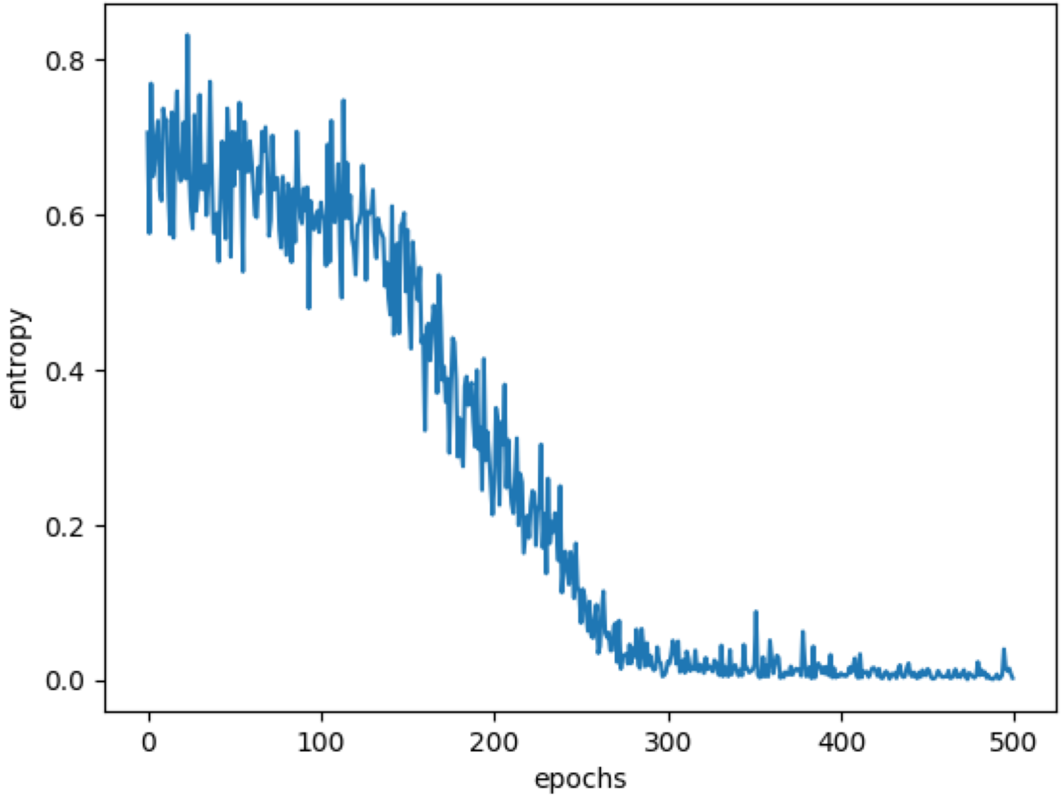
Targets:
[0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0]

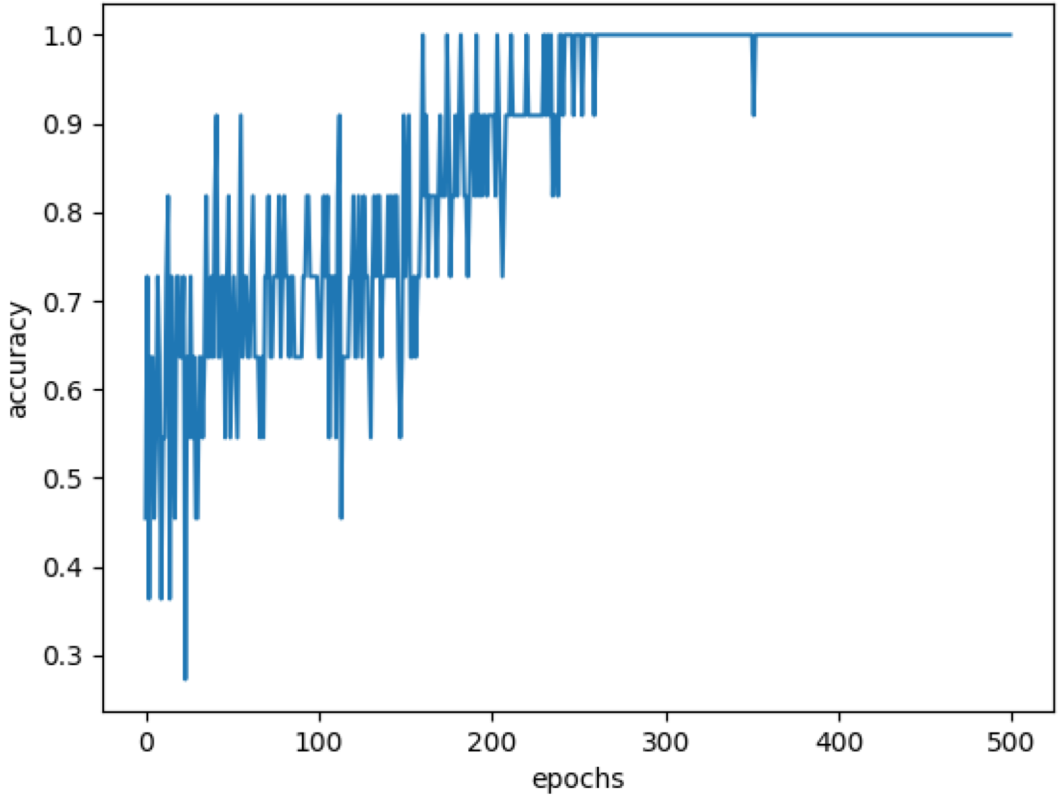
Input text → preprocess to max-length → convert to ids

Building the dictionary:

```
chars = sorted(list(set(list(''.join(data).lower()))))
char_to_ix = { ch:i for i,ch in enumerate(chars) }
vocab_size = len(chars)
```

```
def char_rnn_model(x, keep_prob):

  byte_vectors = tf.one_hot(x, vocab_size)
  byte_list = tf.unstack(byte_vectors, axis=1)

  cell = tf.nn.rnn_cell.GRUCell(HIDDEN_SIZE)
  _, encoding = tf.nn.static_rnn(cell, byte_list, dtype=tf.float32)

  encoding = tf.nn.dropout(encoding, keep_prob)

  logits = tf.layers.dense(encoding, MAX_LABEL, activation=None)

  return logits
```


0.8
0.6
0.4
0.2
0.0
entropy
0
100
200
300
400
500
epochs


accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
100
200
300
400
500
epochs

Test data:

The movie was not interesting to me
I liked the movie with great interest

Logits give the likelihoods of each class:
That is $f(\boldsymbol{u})$ at the softmax layer.

```
[[0.99057394 0.00942606]
 [0.0025338  0.9974662 ]]
```

‘The movie was not interesting to me’ is a negative sentiment at a probability 0.990

‘I liked the movie with great interest’ is positive sentiment at a probability 0.997