ECE 544NA HW5

Jiaqi Mu jiaqimu2

Department of Electrical and Computer Engineering

November 13, 2016

1 TensorFlow

In this portion of the assignment, you will use a vanilla RNN and a LSTM to perform digit classification on the MNIST dataset.

- **Setting 1 (Sequence of Pixels):** It is assumed that each 28×28 image, x, in the MNIST dataset is a sequence of single pixels, $x(1)$, ..., $x(784)$, where $x(t)$ is a single scalar value. The network reads one pixel at a time from the top left corner of the image to the bottem right of the image.
- **Setting 2 (Sequence of Columns):** It is assumed that each 28×28 image, x, in the MNIST dataset is a sequence of vectors, $x(1)$, ..., $x(28)$, where $x(t)$ is a 28×1 vector representing one column in the image. The network reads one column at a time from left to right.

Train a basic (vanilla) RNN and a LSTM for each the two settings using a single layer RNN and LSRM with 100 hidden nodes. Perform classification on the last frame using cross entropy loss.

Revelant Tensorflow Doc: Tensorflow provides some API for recurrent neural network, please use the following:

- `tf.nn.rnn`: `https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g3doc/api_docs/python/functions_and_classes/shard0/tf.nn.rnn.md`
- `tf.nn.rnn_cell`: `https://www.tensorflow.org/versions/r0.11/api_docs/python/rnn_cell.html#neural-network-rnn-cells`

1.1 Methods

- Describe the functions you wrote, and the overall structure of your code.

Proof. We implemented RNN and LSTM in `rnn.py` and `lstm.py` respectively. The overall structures for both implementations are quite similar. Therefore, we simply describe the structure of a vanilla RNN. In LSTM, we simply replace the `rnn_cell` with a `lstm_cell`. In the RNN class, we implemented two functions for training and testing:

– `fit`: to fit the data and learn the parameters from training data. In this module we build a graph structure using a RNN cell (`tf.nn.rnn_cell.BasicRNNCell`) and plug this in an existing RNN module.

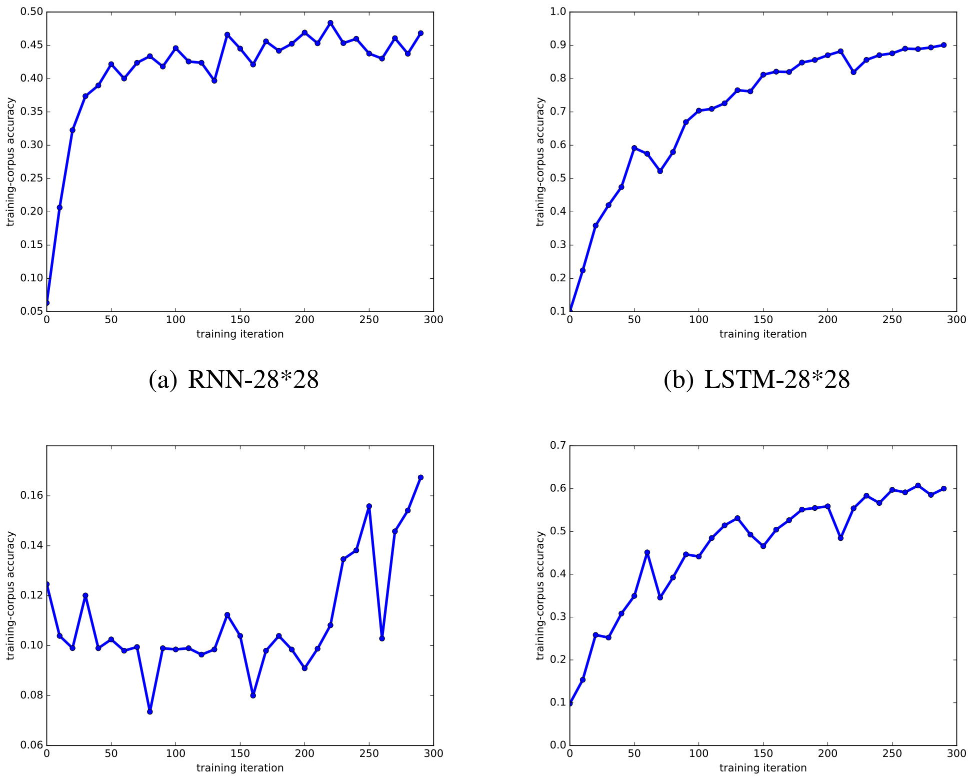
– `predict`: to predict labels for test data.

□

1.2 Results

- Provide one figure with four subfigures, showing convergence plots of all four (2 settings, 2 models) classifiers (abscissa = training iteration, ordinate = training-corpus accuracy)

  Proof. The figures are listed in Figure 1. Due to memory issue, we downsampled an image to 7×7. □



(a) RNN-28*28

(b) LSTM-28*28

(c) RNN-49*1

(d) LSTM-49*1

Figure 1: Convergence plots.

- Provide a table reporting the testing accuracies.

  Proof. The accuracy is reported in Table 1. □

	28*28	49*1
RNN	0.4586	0.1233
LSTM	0.9060	0.6183

Table 1: Accuracy.