Author: sheng wang
study group: mengdi han, ruoxi yu

Homework 2

EECS 498
March 2, 2019

Problem 1

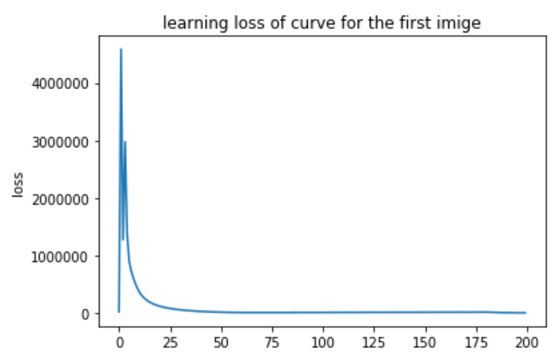
1. Train-model function is implemented.
2. visualize-model function is implemented.
3. finetune function is implemented.
4. freeze function is implemented.
5. the accuracy on validation daaset for these tow scenarios:

Finetune	0.954248
Freeze	0.960784

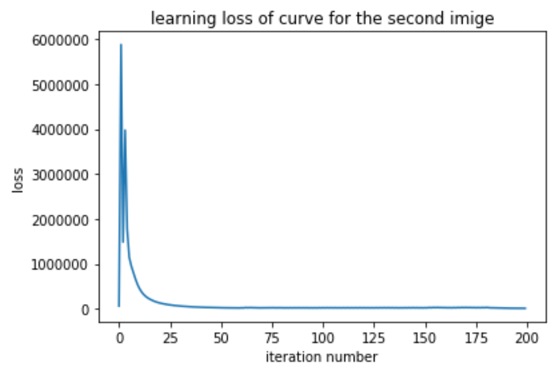
Problem 2

1. content-loss function is implemented.
2. style-loss function is implemented.
3. tv-loss function is implemented.
4. style-transfer function is implemented.

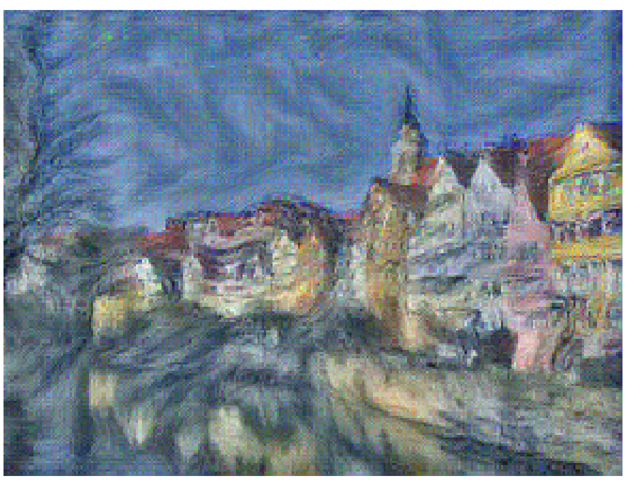
tubingen using composition-vii style:

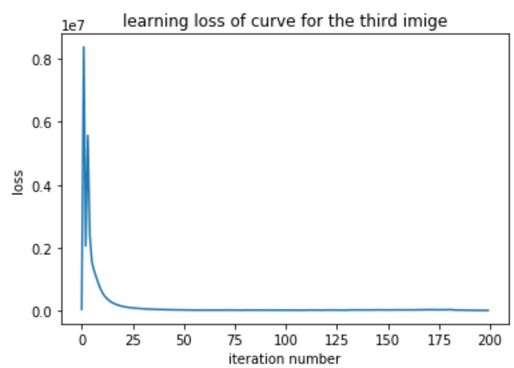


tubingen using the-scream style:



tubingen using starry-night style:





Problem 3

1. rnn-step-forward function is implemented
2. derive the gradient for step backward.

$$\frac{\partial L}{\partial X_t} = [\frac{\partial L}{\partial h_t}(1 - h_t^2)]W_x^T$$

$$\frac{\partial L}{\partial W_x} = X_t^T [\frac{\partial L}{\partial h_t}(1 - h_t^2)]$$

$$\frac{\partial L}{\partial h_{t-1}} = [\frac{\partial L}{\partial h_t}(1 - h_t^2)] W_h^T$$

$$\frac{\partial L}{\partial W_h} = h_{t-1}^T [\frac{\partial L}{\partial h_t}(1 - h_t^2)]$$

$$\frac{\partial L}{\partial b} = \sum_n [\frac{\partial L}{\partial h_{tn}}(1 - h_{tn}^2)]$$

3. rnn forward function is implemented.
4. derive the gradient for rnn backward.

$$\frac{\partial L}{\partial h_{t-1}} = \frac{\partial D}{\partial h_{t-1}} + [\frac{\partial L}{\partial h_t}(1 - h_t^2)] W_h^T$$
$$\frac{\partial L}{\partial h0} = [\frac{\partial L}{\partial h_1}(1 - h_1^2)] W_h^T$$

$$\frac{\partial L}{\partial X_t} = [\frac{\partial L}{\partial h_t}(1 - h_t^2)] W_x^T$$

$$\frac{\partial L}{\partial W_x} = \sum_t X_t^T [\frac{\partial L}{\partial h_t}(1 - h_t^2)]$$

$$\frac{\partial L}{\partial W_h} = \sum_t h_{t-1}^T [\frac{\partial L}{\partial h_t}(1 - h_t^2)]$$

$$\frac{\partial L}{\partial b} = \sum_t \sum_n [\frac{\partial L}{\partial h_{tn}}(1 - h_{tn}^2)]$$

Problem 4

1. lstm-step-forward function is implemented.

2. derive the gradient for step backward.
   define $\theta_{t-fc} = W_x^\theta x_t + W_H^\theta H_{t-1} + b^\theta$ for $\theta = \{f, i, \tilde{c}, o\}$, then we can derive:

$$\frac{\partial L}{\partial f_{t-fc}} = \frac{\partial L}{\partial h_t} c_{t-1} f_t(1-f_t)$$
$$\frac{\partial L}{\partial i_{t-fc}} = \frac{\partial L}{\partial c_t} \tilde{c_t} i_t(1-i_t)$$
$$\frac{\partial L}{\partial \tilde{c_t}} = \frac{\partial L}{\partial c_t} i_t(1-\tilde{c_t}^2)$$
$$\frac{\partial L}{\partial o_{t-fc}} = \frac{\partial L}{\partial h_t} tanh(c_t) o_t(1-o_t)$$

   then we have:

$$\frac{\partial L}{\partial x_t} = \sum_\theta \frac{\partial L}{\partial \theta_{t-fc}} W_x^{\theta T}$$
$$\frac{\partial L}{\partial h_{t-1}} = \sum_\theta \frac{\partial L}{\partial \theta_{t-fc}} W_h^{\theta T}$$
$$\frac{\partial L}{\partial c_{t-1}} = \sum_\theta \frac{\partial L}{\partial c_t} + \frac{\partial L}{\partial h_t} o_t(1-thanh^2(c_t))$$

   and

$$\frac{\partial L}{\partial W_x^\theta} = x_t^T \frac{\partial L}{\partial \theta_{t-fc}}$$
$$\frac{\partial L}{\partial W_h^\theta} = h_{t-1}^T \frac{\partial L}{\partial \theta_{t-fc}}$$
$$\frac{\partial L}{\partial b^\theta} = \mathbf{1}^T \frac{\partial L}{\partial \theta_{t-fc}}$$

3. lstm-forward function is implemented.

4. derive the gradient for lstm backward. define $\theta_{t-fc} = W_x^\theta x_t + W_H^\theta H_{t-1} + b^\theta$ for $\theta = \{f, i, \tilde{c}, o\}$, then we can derive:

$$\frac{\partial L}{\partial f_{t-fc}} = \frac{\partial L}{\partial h_t} c_{t-1} f_t(1-f_t)$$
$$\frac{\partial L}{\partial i_{t-fc}} = \frac{\partial L}{\partial c_t} \tilde{c}_t i_t(1-i_t)$$
$$\frac{\partial L}{\partial \tilde{c}_t} = \frac{\partial L}{\partial c_t} i_t(1-\tilde{c}_t{}^2)$$
$$\frac{\partial L}{\partial o_{t-fc}} = \frac{\partial L}{\partial h_t} tanh(c_t) o_t(1-o_t)$$

then we have:

$$\frac{\partial L}{\partial x_t} = \sum_\theta \frac{\partial L}{\partial \theta_{t-fc}} W_x^{\theta T}$$
$$\frac{\partial L}{\partial h0} = \sum_\theta \frac{\partial L_1}{\partial \theta_{1-fc}} W_h^{\theta T}$$

and

$$\frac{\partial L}{\partial W_x^\theta} = \sum_t x_t^T \frac{\partial L}{\partial \theta_{t-fc}}$$
$$\frac{\partial L}{\partial W_h^\theta} = \sum_t h_{t-1}^T \frac{\partial L}{\partial \theta_{t-fc}}$$
$$\frac{\partial L}{\partial b^\theta} = \sum_t \mathbf{1}^T \frac{\partial L}{\partial \theta_{t-fc}}$$

additionally:

$$\frac{\partial L_t}{\partial h_t} = \frac{\partial D(y_t - \tilde{y}_t)}{\partial h_t} + \sum_\theta \frac{\partial L}{\partial \theta_{t+1-fc}} W_h^{\theta T}$$

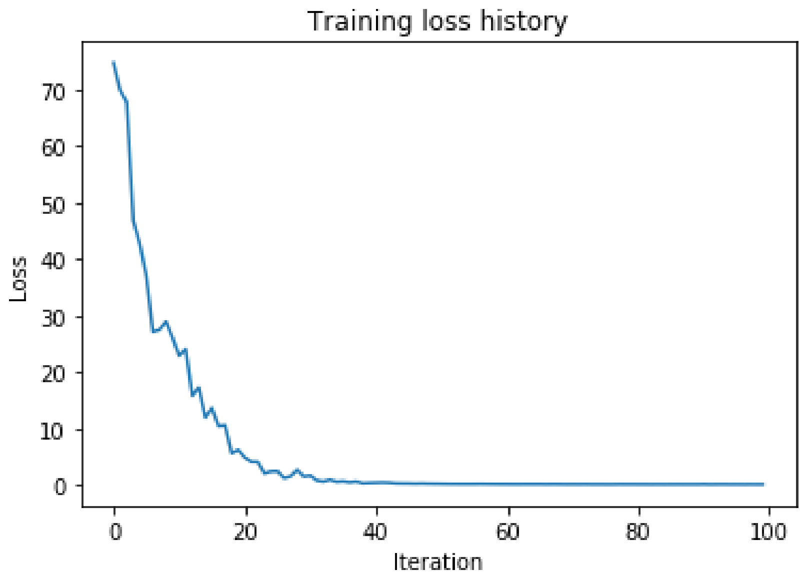
Problem 5

1. the forward and backward of temporal fc is implemented.

2. the temporal-softmax-loss is implemented.

3. sample function is implemented.

4. learning curves of training loss and learned captions for samples. the learning curve of training loss of RNN:



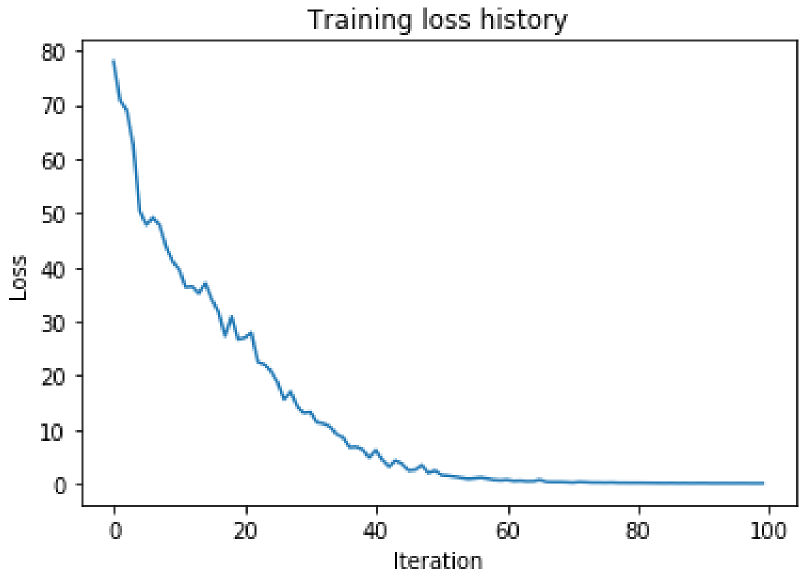
the sample of train captions of RNN:

train
a bathroom with shower toilet and sink is shown <END>
GT:<START> a bathroom with shower toilet and sink is shown <END>

the sample of validation captions of RNN:

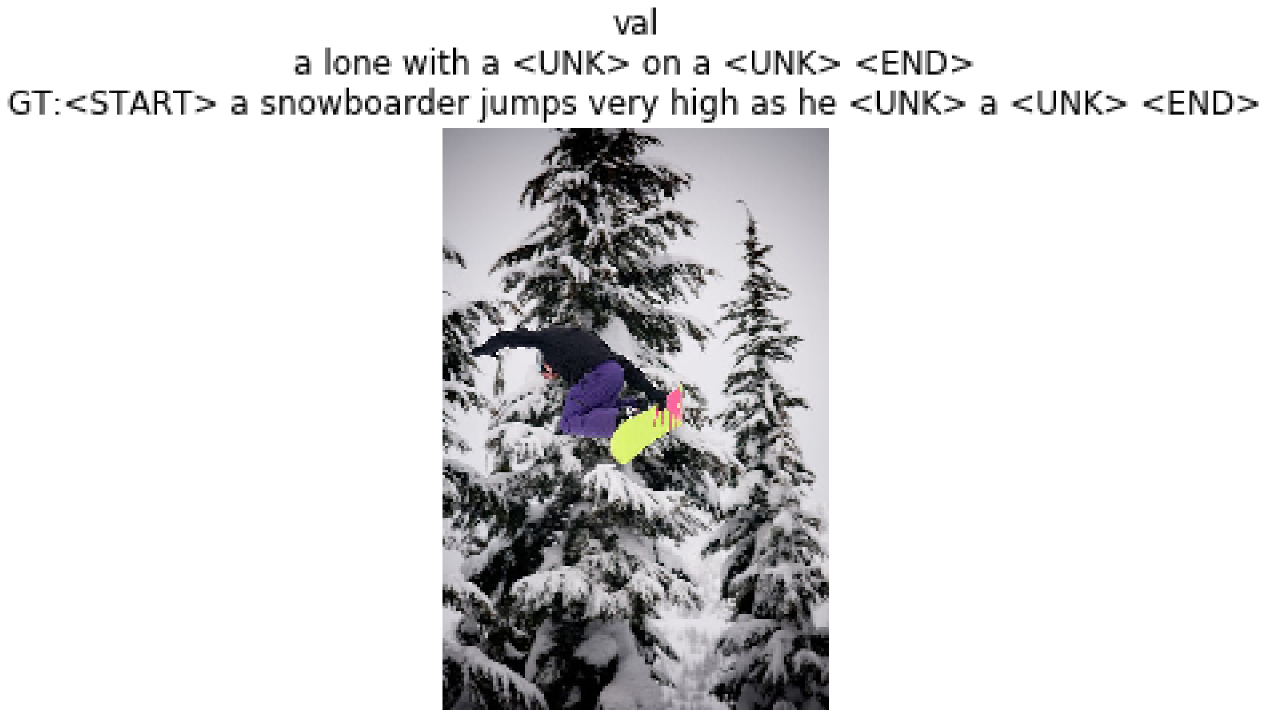
val
a the on in a is with at a traffic light <END>
GT:<START> some tracks for a rail way <UNK> near a building <END>

the learning curve of training loss of LSTM:



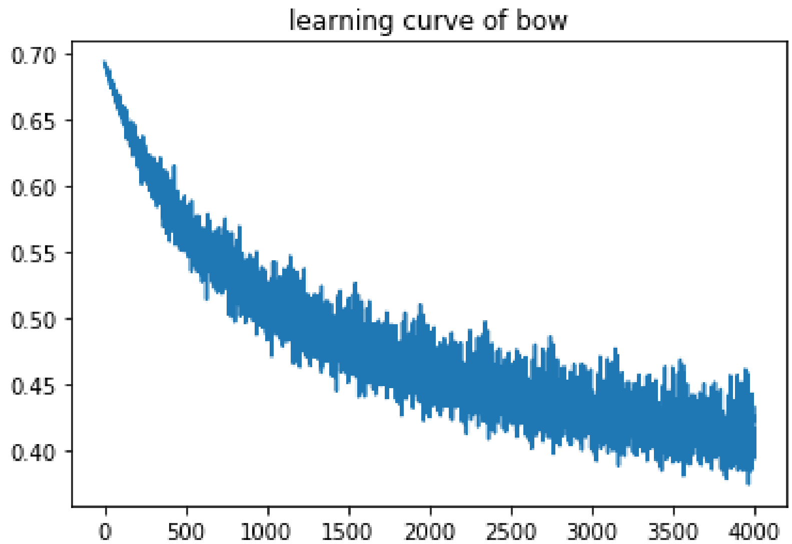
the sample of train captions of LSTM:

the sample of validation captions of LSTM:



Problem 6

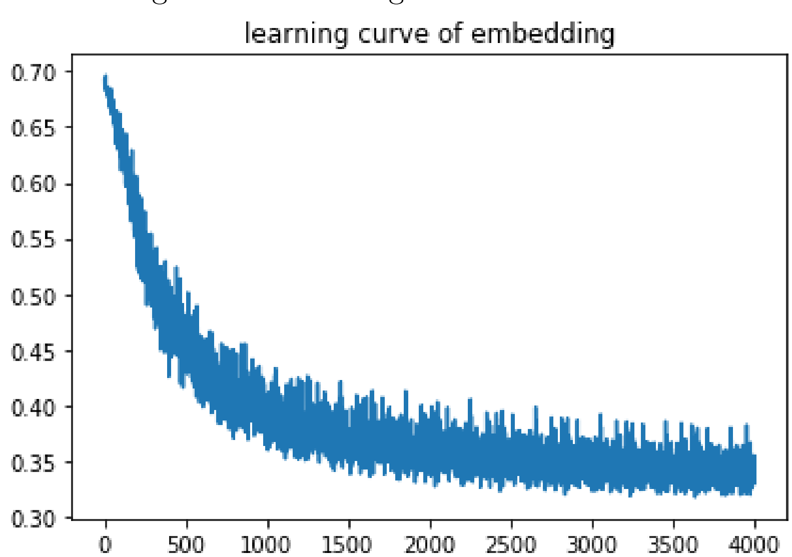
1. bag of words is implemented in bow.py
   The learning curve of training loss is:



The accuracy of bow is:

```
Finished Training
Accuracy of the network on the 40000 train images: 94.872500 %
Accuracy of the network on the 10000 dev images: 93.960000 %
Accuracy of the network on the 10000 test images: 93.700000 %
```

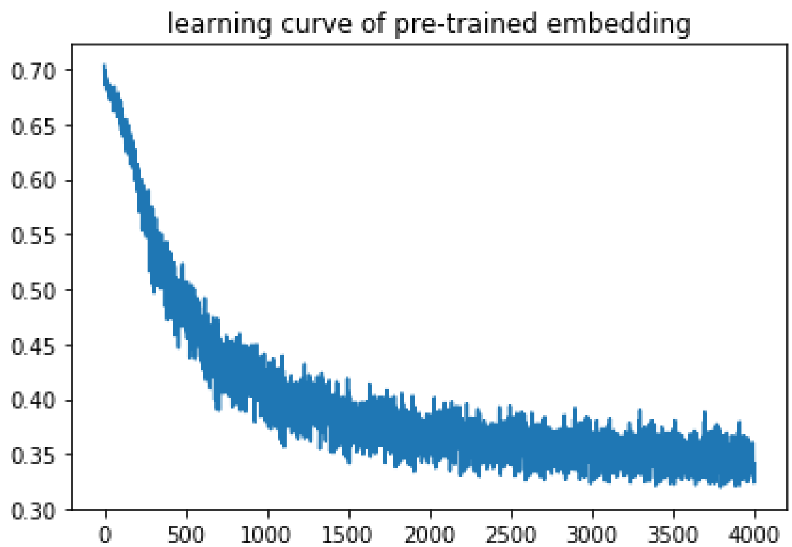
2. word embedding is implemented in embedding.py.
   The learning curve of training loss is:



The accuracy of embedding is:

```
Finished Training
Accuracy of the network on the 40000 train images: 98.092500 %
Accuracy of the network on the 10000 dev images: 95.810000 %
Accuracy of the network on the 10000 test images: 95.810000 %
```

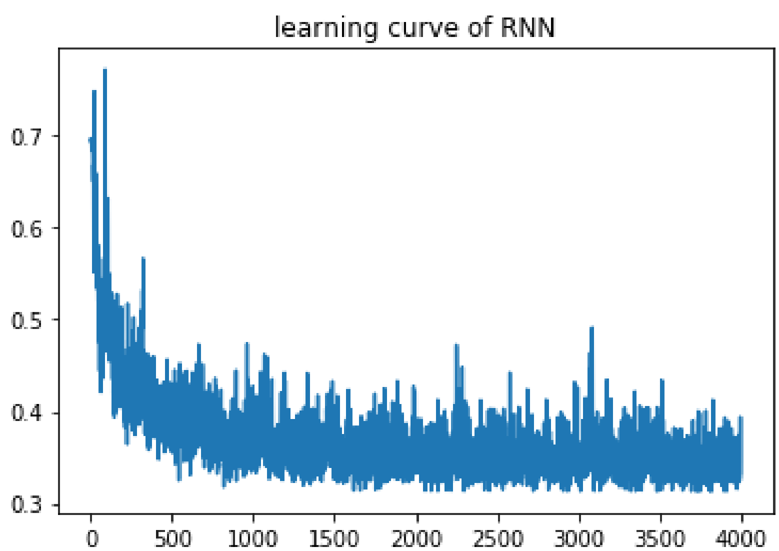
3. word embedding with pre-trained weight is implemented in glov.py
   The larning curve of glov is:



   The accuracy of embedding is:

```
Finished Training
Accuracy of the network on the 40000 train images: 97.760000 %
Accuracy of the network on the 10000 dev images: 95.950000 %
Accuracy of the network on the 10000 test images: 95.420000 %
```

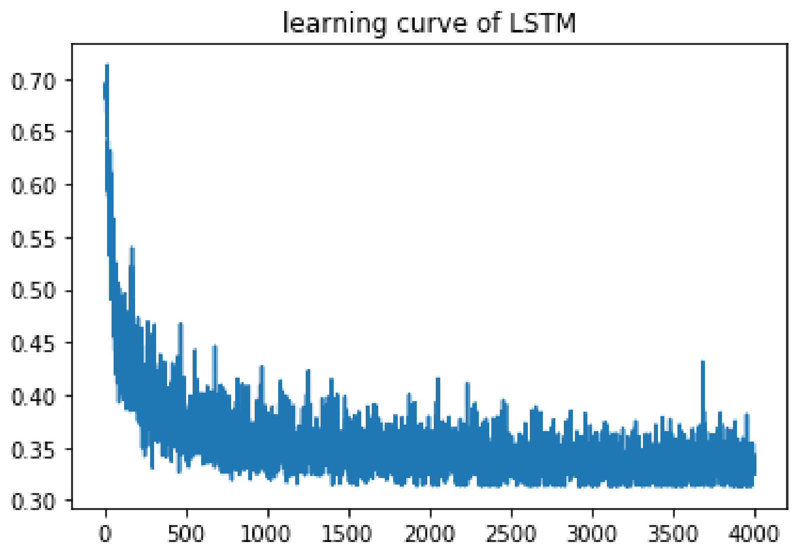
4. rnn is implemented in rnn.py with style='RNN' in prob6 folder.
   The learning curve of rnn is:



   The accuracy of rnn is:

```
Finished Training
Accuracy of the network on the 40000 train images: 95.782500 %
Accuracy of the network on the 10000 dev images: 94.180000 %
Accuracy of the network on the 10000 test images: 93.670000 %
```

5. lstm is implemented in lstm.py with style='LSTM' in prob6 folder.
   The learning curve of lstm is:



   The accuracy of lstm is:

```
Accuracy of the network on the 40000 train images: 98.222500 %
Accuracy of the network on the 10000 dev images: 96.420000 %
Accuracy of the network on the 10000 test images: 96.050000 %
```