Tut_6_Generative_RNN-1

December 13, 2022

1 Tutorial - Generative Recurrent Neural Networks

Last time we discussed using recurrent neural networks to make predictions about sequences. In particular, we treated tweets as a **sequence** of words. Since tweets can have a variable number of words, we needed an architecture that can take variable-sized sequences as input.

This time, we will use recurrent neural networks to **generate** sequences. Generating sequences is more involved compared to making predictions about sequences. However, it is a very interesting task, and many students chose sequence-generation tasks for their projects.

Much of today's content is an adaptation of the "Practical PyTorch" GitHub repository [1].

[1] https://github.com/spro/practical-pytorch/blob/master/char-rnn-generation/char-rnn-generation.ipynb

1.1 Review

In recurrent neural networks, the input sequence is broken down into tokens. We could choose whether to tokenize based on words, or based on characters. The representation of each token (GloVe or one-hot) is processed by the RNN one step at a time to update the hidden (or context) state.

In a predictive RNN, the value of the hidden states is a representation of **all the text that was processed thus far**. Similarly, in a generative RNN, The value of the hidden state will be a representation of **all the text that still needs to be generated**. We will use this hidden state to produce the sequence, one token at a time.

Similar to the last tutorial we will break up the problem of generating text to generating one token at a time.

We will do so with the help of two functions:

1. We need to be able to generate the *next* token, given the current hidden state. In practice, we get a probability distribution over the next token, and sample from that probability distribution.
2. We need to be able to update the hidden state somehow. To do so, we need two pieces of information: the old hidden state, and the actual token that was generated in the previous step. The actual token generated will inform the subsequent tokens.

We will repeat both functions until a special "END OF SEQUENCE" token is generated.

Note that there are several tricky things that we will have to figure out. For example, how do we actually sample the actual token from the probability distribution over tokens? What would we do

during training, and how might that be different from during testing/evaluation? We will answer those questions as we implement the RNN.

For now, let's start with our training data.

1.2 Data: Donald Trump's Tweets from 2018

The training set we use is a collection of Donald Trump's tweets from 2018. We will only use tweets that are 140 characters or shorter, and tweets that contains more than just a URL. Since tweets often contain creative spelling and numbers, and upper vs. lower case characters are read very differently, we will use a character-level RNN.

To start, let us load the trump.csv file to Google Colab and provide access to the drive. The file can be obtained from Quercus.

[ ]:
```
%pip install torch==1.8.0+cu111 -f https://download.pytorch.org/whl/
 ↪torch_stable.html
%pip install torchtext==0.9 # Necessary to ensure we are using torxhtext␣
 ↪version 0.9 that has access to legacy
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Looking in links: https://download.pytorch.org/whl/torch_stable.html
Collecting torch==1.8.0+cu111
  Downloading https://download.pytorch.org/whl/cu111/torch-1.8.0%2Bcu111-cp37-cp
37m-linux_x86_64.whl (1982.2 MB)
     |                               | 834.1 MB 1.4 MB/s eta
0:14:04tcmalloc: large alloc 1147494400 bytes == 0x38ebc000 @  0x7fe6d7219615
0x592b76 0x4df71e 0x59afff 0x515655 0x549576 0x593fce 0x548ae9 0x51566f 0x549576
0x593fce 0x548ae9 0x5127f1 0x598e3b 0x511f68 0x598e3b 0x511f68 0x598e3b 0x511f68
0x4bc98a 0x532e76 0x594b72 0x515600 0x549576 0x593fce 0x548ae9 0x5127f1 0x549576
0x593fce 0x5118f8 0x593dd7
     |                              | 1055.7 MB 1.2 MB/s eta
0:12:26tcmalloc: large alloc 1434370048 bytes == 0x7d512000 @  0x7fe6d7219615
0x592b76 0x4df71e 0x59afff 0x515655 0x549576 0x593fce 0x548ae9 0x51566f 0x549576
0x593fce 0x548ae9 0x5127f1 0x598e3b 0x511f68 0x598e3b 0x511f68 0x598e3b 0x511f68
0x4bc98a 0x532e76 0x594b72 0x515600 0x549576 0x593fce 0x548ae9 0x5127f1 0x549576
0x593fce 0x5118f8 0x593dd7
     |                            | 1336.2 MB 1.3 MB/s eta
0:08:24tcmalloc: large alloc 1792966656 bytes == 0x2344000 @  0x7fe6d7219615
0x592b76 0x4df71e 0x59afff 0x515655 0x549576 0x593fce 0x548ae9 0x51566f 0x549576
0x593fce 0x548ae9 0x5127f1 0x598e3b 0x511f68 0x598e3b 0x511f68 0x598e3b 0x511f68
0x4bc98a 0x532e76 0x594b72 0x515600 0x549576 0x593fce 0x548ae9 0x5127f1 0x549576
0x593fce 0x5118f8 0x593dd7
     |                          | 1691.1 MB 1.2 MB/s eta
0:04:08tcmalloc: large alloc 2241208320 bytes == 0x6d12c000 @  0x7fe6d7219615
0x592b76 0x4df71e 0x59afff 0x515655 0x549576 0x593fce 0x548ae9 0x51566f 0x549576
0x593fce 0x548ae9 0x5127f1 0x598e3b 0x511f68 0x598e3b 0x511f68 0x598e3b 0x511f68
0x4bc98a 0x532e76 0x594b72 0x515600 0x549576 0x593fce 0x548ae9 0x5127f1 0x549576
```

```
0x593fce 0x5118f8 0x593dd7
     |                  | 1982.2 MB 1.2 MB/s eta
0:00:01tcmalloc: large alloc 1982251008 bytes == 0xf2a8e000 @  0x7fe6d72181e7
0x4a3940 0x4a39cc 0x592b76 0x4df71e 0x59afff 0x515655 0x549576 0x593fce 0x511e2c
0x549576 0x593fce 0x511e2c 0x549576 0x593fce 0x511e2c 0x549576 0x593fce 0x511e2c
0x549576 0x593fce 0x511e2c 0x593dd7 0x511e2c 0x549576 0x593fce 0x548ae9 0x5127f1
0x549576 0x593fce 0x548ae9
tcmalloc: large alloc 2477817856 bytes == 0x168cfa000 @  0x7fe6d7219615 0x592b76
0x4df71e 0x59afff 0x515655 0x549576 0x593fce 0x511e2c 0x549576 0x593fce 0x511e2c
0x549576 0x593fce 0x511e2c 0x549576 0x593fce 0x511e2c 0x549576 0x593fce 0x511e2c
0x593dd7 0x511e2c 0x549576 0x593fce 0x548ae9 0x5127f1 0x549576 0x593fce 0x548ae9
0x5127f1 0x549576
     |                  | 1982.2 MB 5.3 kB/s
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.7/dist-packages (from torch==1.8.0+cu111) (4.1.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages
(from torch==1.8.0+cu111) (1.21.6)
Installing collected packages: torch
  Attempting uninstall: torch
    Found existing installation: torch 1.12.0+cu113
    Uninstalling torch-1.12.0+cu113:
      Successfully uninstalled torch-1.12.0+cu113
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
torchvision 0.13.0+cu113 requires torch==1.12.0, but you have torch 1.8.0+cu111
which is incompatible.
torchtext 0.13.0 requires torch==1.12.0, but you have torch 1.8.0+cu111 which is
incompatible.
torchaudio 0.12.0+cu113 requires torch==1.12.0, but you have torch 1.8.0+cu111
which is incompatible.
Successfully installed torch-1.8.0+cu111
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting torchtext==0.9
  Downloading torchtext-0.9.0-cp37-cp37m-manylinux1_x86_64.whl (7.1 MB)
     |                  | 7.1 MB 27.6 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from torchtext==0.9) (2.23.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages
(from torchtext==0.9) (1.21.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages
(from torchtext==0.9) (4.64.0)
```

```
Requirement already satisfied: torch==1.8.0 in /usr/local/lib/python3.7/dist-
packages (from torchtext==0.9) (1.8.0+cu111)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.7/dist-packages (from torch==1.8.0->torchtext==0.9)
(4.1.1)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests->torchtext==0.9)
(2022.6.15)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->torchtext==0.9) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests->torchtext==0.9) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->torchtext==0.9) (1.24.3)
Installing collected packages: torchtext
  Attempting uninstall: torchtext
    Found existing installation: torchtext 0.13.0
    Uninstalling torchtext-0.13.0:
      Successfully uninstalled torchtext-0.13.0
Successfully installed torchtext-0.9.0
```

[ ]:
```python
from google.colab import drive
drive.mount('/content/drive')
```

[ ]:
```python
import csv

# file location (make sure to use your file location)
file_dir = '/content/drive/My Drive/Colab Notebooks/Lab 6 Tutorial/'

tweets = list(line[0] for line in csv.reader(open(file_dir + 'trump.csv')))
len(tweets)
```

[ ]:
```
22402
```

There are over 20000 tweets in this collection. Let's look at a few of them, just to get a sense of the kind of text we're dealing with:

[ ]:
```python
print(tweets[100])
print(tweets[1000])
print(tweets[10000])
```

```
God Bless the people of Venezuela!
It was my honor. THANK YOU! https://t.co/1LvqbRQ1bi
Nobody but Donald Trump will save Israel. You are wasting your time with these
politicians and political clowns. Best! #SheldonAdelson
```

1.3 Generating One Tweet

Normally, when we build a new machine learning model, we want to make sure that our model can overfit. To that end, we will first build a neural network that can generate *one* tweet really well. We can choose any tweet (or any other text) we want. Let's choose to build an RNN that generates `tweet[100]`.

[ ]:
```
tweet = tweets[100]
print(tweet)
print(len(tweet))
```

```
God Bless the people of Venezuela!
34
```

First, we will need to encode this tweet using a one-hot encoding. We'll build dictionary mappings from the character to the index of that character (a unique integer identifier), and from the index to the character. We'll use the same naming scheme that `torchtext` uses (`stoi` and `itos`).

For simplicity, we'll work with a limited vocabulary containing just the characters in `tweet[100]`, plus two special tokens:

- `<EOS>` represents "End of String", which we'll append to the end of our tweet. Since tweets are variable-length, this is a way for the RNN to signal that the entire sequence has been generated.
- `<BOS>` represents "Beginning of String", which we'll prepend to the beginning of our tweet. This is the first token that we will feed into the RNN.

The way we use these special tokens will become more clear as we build the model.

[ ]:
```
vocab = list(set(tweet)) + ["<BOS>", "<EOS>"]
vocab_stoi = {s: i for i, s in enumerate(vocab)}
vocab_itos = {i: s for i, s in enumerate(vocab)}
vocab_size = len(vocab)
```

[ ]:
```
print(vocab)
print(vocab_stoi)
print(vocab_itos)
print(vocab_size)
```

```
['s', ' ', 'd', 'f', 'n', 'e', 't', 'G', 'p', 'z', 'B', 'h', 'a', 'u', '!', 'V',
'l', 'o', '<BOS>', '<EOS>']
{'s': 0, ' ': 1, 'd': 2, 'f': 3, 'n': 4, 'e': 5, 't': 6, 'G': 7, 'p': 8, 'z': 9,
'B': 10, 'h': 11, 'a': 12, 'u': 13, '!': 14, 'V': 15, 'l': 16, 'o': 17, '<BOS>':
18, '<EOS>': 19}
{0: 's', 1: ' ', 2: 'd', 3: 'f', 4: 'n', 5: 'e', 6: 't', 7: 'G', 8: 'p', 9: 'z',
10: 'B', 11: 'h', 12: 'a', 13: 'u', 14: '!', 15: 'V', 16: 'l', 17: 'o', 18:
'<BOS>', 19: '<EOS>'}
20
```

Now that we have our vocabulary, we can build the PyTorch model for this problem. The actual model is not as complex as you might think. We actually already learned about all the components that we need. (Using and training the model is the hard part)

```
[ ]: import torch
     import torch.nn as nn
     import torch.nn.functional as F
     import torch.optim as optim
```

```
[ ]: class TextGenerator(nn.Module):
         def __init__(self, vocab_size, hidden_size, n_layers=1):
             super(TextGenerator, self).__init__()

             # identiy matrix for generating one-hot vectors
             self.ident = torch.eye(vocab_size)

             # recurrent neural network
             self.rnn = nn.GRU(vocab_size, hidden_size, n_layers, batch_first=True)

             # a fully-connect layer that outputs a distribution over
             # the next token, given the RNN output
             self.decoder = nn.Linear(hidden_size, vocab_size)

         def forward(self, inp, hidden=None):
             inp = self.ident[inp]                    # generate one-hot vectors of␣
      ↪input
             output, hidden = self.rnn(inp, hidden) # get the next output and hidden␣
      ↪state
             output = self.decoder(output)           # predict distribution over next␣
      ↪tokens
             return output, hidden

     model = TextGenerator(vocab_size, 64)
```

1.4 Training with Teacher Forcing

At a very high level, we want our RNN model to have a high probability of generating the tweet. An RNN model generates text one character at a time based on the hidden state value. At each time step, we will check whether the model generated the correct character. That is, at each time step, we are trying to select the correct next character out of all the characters in our vocabulary. Recall that this problem is a multi-class classification problem, and we can use Cross-Entropy loss to train our network to become better at this type of problem.

```
[ ]: criterion = nn.CrossEntropyLoss()
```

However, we don't just have a single multi-class classification problem. Instead, we have **one classification problem per time-step** (per token)! So, how do we predict the first token in the sequence? How do we predict the second token in the sequence?

To help you understand what happens durign RNN training, we'll start with inefficient training code that shows you what happens step-by-step. We'll start with computing the loss for the first token generated, then the second token, and so on. Later on, we'll switch to a simpler and more

performant version of the code.

So, let's start with the first classification problem: the problem of generating the **first** token (`tweet[0]`).

To generate the first token, we'll feed the RNN network (with an initial, empty hidden state) the "" token. Then, the output

[ ]:
```
bos_input = torch.Tensor([vocab_stoi["<BOS>"]])
print(bos_input.shape, type(bos_input))
bos_input = bos_input.long()
print(bos_input.shape, type(bos_input))
bos_input = bos_input.unsqueeze(0)
print(bos_input.shape, type(bos_input))
output, hidden = model(bos_input, hidden=None)
output # distribution over the first token
```

```
torch.Size([1]) <class 'torch.Tensor'>
torch.Size([1]) <class 'torch.Tensor'>
torch.Size([1, 1]) <class 'torch.Tensor'>
```

[ ]:
```
tensor([[[ 0.0864,  0.0330,  0.0653, -0.0746, -0.0119, -0.0869, -0.0172,
           -0.0979,  0.0294,  0.0267, -0.0064, -0.0517, -0.0216, -0.0640,
           -0.0459,  0.0560, -0.0724, -0.0922, -0.1239,  0.0612]]],
       grad_fn=<AddBackward0>)
```

[ ]:
```
bos_input
```

[ ]:
```
tensor([[18]])
```

We can compute the loss using `criterion`. Since the model is untrained, the loss is expected to be high. (For now, we won't do anything with this loss, and omit the backward pass.)

[ ]:
```
target = torch.Tensor([vocab_stoi[tweet[0]]]).long().unsqueeze(0)
criterion(output.reshape(-1, vocab_size), # reshape to 2D tensor
          target.reshape(-1))             # reshape to 1D tensor
```

[ ]:
```
tensor(3.0751, grad_fn=<NllLossBackward>)
```

[ ]:
```
print(target)
print(output)
print(output.reshape(-1, vocab_size))
print(target.reshape(-1))
```

```
tensor([[7]])
tensor([[[ 0.0864,  0.0330,  0.0653, -0.0746, -0.0119, -0.0869, -0.0172,
           -0.0979,  0.0294,  0.0267, -0.0064, -0.0517, -0.0216, -0.0640,
           -0.0459,  0.0560, -0.0724, -0.0922, -0.1239,  0.0612]]],
       grad_fn=<AddBackward0>)
tensor([[ 0.0864,  0.0330,  0.0653, -0.0746, -0.0119, -0.0869, -0.0172, -0.0979,
           0.0294,  0.0267, -0.0064, -0.0517, -0.0216, -0.0640, -0.0459,  0.0560,
```

```
        -0.0724, -0.0922, -0.1239,  0.0612]], grad_fn=<ViewBackward>)
tensor([7])
```

Now, we need to update the hidden state and generate a prediction for the next token. To do so, we need to provide the current token to the RNN. We already said that during test time, we'll need to sample from the predicted probabilty over tokens that the neural network just generated.

Right now, we can do something better: we can **use the ground-truth, actual target token**. This technique is called **teacher-forcing**, and generally speeds up training. The reason is that right now, since our model does not perform well, the predicted probability distribution is pretty far from the ground truth. So, it is very, very difficult for the neural network to get back on track given bad input data.

[ ]:
```
# Use teacher-forcing: we pass in the ground truth `target`,
# rather than using the NN predicted distribution
output, hidden = model(target, hidden)
output # distribution over the second token
```

[ ]:
```
tensor([[[ 0.0499,  0.0223,  0.0319, -0.0223,  0.0070, -0.0608, -0.0521,
          -0.1293,  0.0113, -0.0102, -0.0344, -0.0561, -0.0203, -0.1048,
          -0.0514,  0.0311, -0.0252, -0.0617, -0.1148,  0.0835]]],
       grad_fn=<AddBackward0>)
```

Similar to the first step, we can compute the loss, quantifying the difference between the predicted distribution and the actual next token. This loss can be used to adjust the weights of the neural network (which we are not doing yet).

[ ]:
```
target = torch.Tensor([vocab_stoi[tweet[1]]]).long().unsqueeze(0)
criterion(output.reshape(-1, vocab_size), # reshape to 2D tensor
          target.reshape(-1))             # reshape to 1D tensor
```

[ ]:
```
tensor(3.0336, grad_fn=<NllLossBackward>)
```

We can continue this process of:

- feeding the previous ground-truth token to the RNN,
- obtaining the prediction distribution over the next token, and
- computing the loss,

for as many steps as there are tokens in the ground-truth tweet.

[ ]:
```
for i in range(2, len(tweet)):
    output, hidden = model(target, hidden)
    target = torch.Tensor([vocab_stoi[tweet[i]]]).long().unsqueeze(0)
    loss = criterion(output.reshape(-1, vocab_size), # reshape to 2D tensor
                     target.reshape(-1))             # reshape to 1D tensor
    print(i, output, loss)
```

```
2 tensor([[[ 0.0409, -0.0133,  0.0268, -0.0090,  0.0048, -0.0664, -0.0378,
          -0.1011,  0.0004, -0.0028, -0.0399, -0.0260, -0.0419, -0.0620,
          -0.0636,  0.0104, -0.0434, -0.0901, -0.1486,  0.0677]]],
```

```
       grad_fn=<AddBackward0>) tensor(2.9404, grad_fn=<NllLossBackward>)
3 tensor([[[ 0.0549, -0.0159,  0.0251, -0.0630,  0.0098, -0.0398, -0.0125,
          -0.1126, -0.0071, -0.0531, -0.0219, -0.0377, -0.0284, -0.0566,
          -0.0407, -0.0169, -0.0441, -0.0767, -0.1402,  0.0749]]],
       grad_fn=<AddBackward0>) tensor(2.9827, grad_fn=<NllLossBackward>)
4 tensor([[[ 0.0416, -0.0229,  0.0243, -0.0450,  0.0093, -0.0124, -0.0390,
          -0.1375, -0.0193, -0.0237, -0.0031, -0.0392, -0.0345, -0.0648,
          -0.0088,  0.0077, -0.0722, -0.0652, -0.1343,  0.0538]]],
       grad_fn=<AddBackward0>) tensor(2.9708, grad_fn=<NllLossBackward>)
5 tensor([[[ 0.0171,  0.0002,  0.0278, -0.0523,  0.0537, -0.0610, -0.0495,
          -0.1542,  0.0454,  0.0036, -0.0040, -0.0369, -0.0734, -0.0547,
          -0.0272,  0.0412, -0.0720, -0.0582, -0.1444,  0.0438]]],
       grad_fn=<AddBackward0>) tensor(3.0416, grad_fn=<NllLossBackward>)
6 tensor([[[ 0.0467, -0.0141,  0.0257, -0.0421,  0.0075, -0.0477, -0.0514,
          -0.1369,  0.0454, -0.0120, -0.0592, -0.0534, -0.0356, -0.0736,
          -0.0265,  0.0178, -0.0739, -0.0437, -0.1464,  0.0662]]],
       grad_fn=<AddBackward0>) tensor(3.0146, grad_fn=<NllLossBackward>)
7 tensor([[[ 0.0808, -0.0054,  0.0165, -0.0985,  0.0247, -0.1056, -0.0086,
          -0.1211,  0.0193,  0.0273, -0.0587, -0.0735, -0.0297, -0.0784,
          -0.0470,  0.0308, -0.0502, -0.0744, -0.1556,  0.0696]]],
       grad_fn=<AddBackward0>) tensor(2.8850, grad_fn=<NllLossBackward>)
8 tensor([[[ 0.0355, -0.0068,  0.0081, -0.0953, -0.0163, -0.0943, -0.0359,
          -0.1250,  0.0104, -0.0078, -0.0481, -0.0563, -0.0369, -0.0861,
          -0.0349,  0.0372, -0.0507, -0.0616, -0.1170,  0.0866]]],
       grad_fn=<AddBackward0>) tensor(2.9270, grad_fn=<NllLossBackward>)
9 tensor([[[ 0.0159, -0.0038,  0.0015, -0.0966, -0.0350, -0.0855, -0.0467,
          -0.1310,  0.0006, -0.0250, -0.0403, -0.0532, -0.0375, -0.0922,
          -0.0302,  0.0388, -0.0479, -0.0512, -0.0946,  0.0935]]],
       grad_fn=<AddBackward0>) tensor(2.9648, grad_fn=<NllLossBackward>)
10 tensor([[[ 0.0280, -0.0034,  0.0101, -0.0696, -0.0094, -0.0357, -0.0594,
          -0.1489, -0.0225, -0.0145, -0.0113, -0.0568, -0.0306, -0.0848,
          -0.0022,  0.0401, -0.0689, -0.0492, -0.0954,  0.0715]]],
       grad_fn=<AddBackward0>) tensor(3.0257, grad_fn=<NllLossBackward>)
11 tensor([[[ 0.0749,  0.0027,  0.0117, -0.0288, -0.0071, -0.0830, -0.0142,
          -0.1442,  0.0048, -0.0324, -0.0364, -0.0779, -0.0339, -0.0689,
          -0.0620,  0.0319, -0.0545, -0.0510, -0.0997,  0.0873]]],
       grad_fn=<AddBackward0>) tensor(3.0461, grad_fn=<NllLossBackward>)
12 tensor([[[ 0.0826,  0.0092,  0.0287, -0.0381,  0.0148, -0.0653, -0.0029,
          -0.1479,  0.0296, -0.0297, -0.0336, -0.0867, -0.0175, -0.0688,
          -0.0519, -0.0044,  0.0056, -0.0591, -0.1058,  0.1133]]],
       grad_fn=<AddBackward0>) tensor(3.0414, grad_fn=<NllLossBackward>)
13 tensor([[[ 0.0995,  0.0118,  0.0165, -0.1011,  0.0324, -0.1061,  0.0181,
          -0.1270,  0.0107,  0.0108, -0.0410, -0.0904, -0.0194, -0.0750,
          -0.0589,  0.0233, -0.0058, -0.0803, -0.1305,  0.0930]]],
       grad_fn=<AddBackward0>) tensor(2.9601, grad_fn=<NllLossBackward>)
14 tensor([[[ 0.0663,  0.0026,  0.0198, -0.0686,  0.0238, -0.0471, -0.0286,
          -0.1393, -0.0094, -0.0006, -0.0111, -0.0678, -0.0277, -0.0712,
          -0.0147,  0.0357, -0.0472, -0.0675, -0.1164,  0.0728]]],
```

```
       grad_fn=<AddBackward0>) tensor(2.9818, grad_fn=<NllLossBackward>)
15 tensor([[[ 0.0492,  0.0210, -0.0300, -0.0427,  0.0183, -0.0454, -0.0575,
         -0.1425,  0.0160, -0.0029, -0.0385, -0.0433, -0.0283, -0.0514,
          0.0171,  0.0657, -0.0531, -0.0788, -0.0745,  0.0525]]],
       grad_fn=<AddBackward0>) tensor(3.0199, grad_fn=<NllLossBackward>)
16 tensor([[[ 0.0885,  0.0138, -0.0103, -0.1022,  0.0374, -0.0968, -0.0139,
         -0.1298,  0.0038,  0.0340, -0.0466, -0.0664, -0.0260, -0.0648,
         -0.0207,  0.0540, -0.0373, -0.1008, -0.1229,  0.0650]]],
       grad_fn=<AddBackward0>) tensor(3.0714, grad_fn=<NllLossBackward>)
17 tensor([[[ 0.0658, -0.0158,  0.0030, -0.0509,  0.0241, -0.0816, -0.0273,
         -0.0968, -0.0028,  0.0133, -0.0439, -0.0253, -0.0447, -0.0354,
         -0.0371,  0.0264, -0.0462, -0.1126, -0.1472,  0.0664]]],
       grad_fn=<AddBackward0>) tensor(2.9716, grad_fn=<NllLossBackward>)
18 tensor([[[ 0.0516,  0.0091, -0.0422, -0.0303,  0.0179, -0.0554, -0.0635,
         -0.1232,  0.0159,  0.0037, -0.0565, -0.0131, -0.0304, -0.0304,
          0.0145,  0.0629, -0.0516, -0.0992, -0.0906,  0.0490]]],
       grad_fn=<AddBackward0>) tensor(3.0255, grad_fn=<NllLossBackward>)
19 tensor([[[ 0.0723, -0.0134, -0.0087, -0.0297, -0.0073, -0.0356, -0.0631,
         -0.1220,  0.0238, -0.0087, -0.0838, -0.0358, -0.0088, -0.0560,
          0.0086,  0.0303, -0.0645, -0.0720, -0.1250,  0.0685]]],
       grad_fn=<AddBackward0>) tensor(3.0062, grad_fn=<NllLossBackward>)
20 tensor([[[ 0.0978, -0.0062, -0.0020, -0.0904,  0.0187, -0.0954, -0.0182,
         -0.1135,  0.0040,  0.0301, -0.0707, -0.0614, -0.0128, -0.0653,
         -0.0210,  0.0384, -0.0454, -0.0926, -0.1479,  0.0685]]],
       grad_fn=<AddBackward0>) tensor(2.9746, grad_fn=<NllLossBackward>)
21 tensor([[[ 0.0680, -0.0108,  0.0110, -0.0590,  0.0163, -0.0441, -0.0499,
         -0.1321, -0.0152,  0.0127, -0.0286, -0.0523, -0.0228, -0.0639,
          0.0097,  0.0424, -0.0694, -0.0778, -0.1301,  0.0587]]],
       grad_fn=<AddBackward0>) tensor(3.0482, grad_fn=<NllLossBackward>)
22 tensor([[[ 0.0572, -0.0295,  0.0130, -0.0246,  0.0131, -0.0576, -0.0379,
         -0.1013, -0.0081,  0.0049, -0.0363, -0.0231, -0.0459, -0.0361,
         -0.0275,  0.0143, -0.0615, -0.1023, -0.1532,  0.0582]]],
       grad_fn=<AddBackward0>) tensor(2.9924, grad_fn=<NllLossBackward>)
23 tensor([[[ 0.0601, -0.0003,  0.0037, -0.0540,  0.0185, -0.0610, -0.0895,
         -0.1066,  0.0015, -0.0073, -0.0372, -0.0479, -0.0408, -0.0586,
          0.0249,  0.0486, -0.0852, -0.0837, -0.1474,  0.0649]]],
       grad_fn=<AddBackward0>) tensor(2.9678, grad_fn=<NllLossBackward>)
24 tensor([[[ 0.0565, -0.0101,  0.0092, -0.0400,  0.0180, -0.0172, -0.0829,
         -0.1401, -0.0154, -0.0012, -0.0180, -0.0474, -0.0375, -0.0646,
          0.0283,  0.0408, -0.0889, -0.0768, -0.1326,  0.0556]]],
       grad_fn=<AddBackward0>) tensor(2.9282, grad_fn=<NllLossBackward>)
25 tensor([[[ 0.0887, -0.0319, -0.0071, -0.0576,  0.0049, -0.0215, -0.0704,
         -0.1403, -0.0211, -0.0292, -0.0209, -0.0191, -0.0296, -0.0900,
         -0.0110,  0.0773, -0.0597, -0.0811, -0.1154,  0.0738]]],
       grad_fn=<AddBackward0>) tensor(2.9909, grad_fn=<NllLossBackward>)
26 tensor([[[ 0.1093, -0.0135, -0.0017, -0.1096,  0.0315, -0.0776, -0.0217,
         -0.1313, -0.0161,  0.0212, -0.0378, -0.0519, -0.0282, -0.0875,
         -0.0353,  0.0585, -0.0424, -0.1011, -0.1442,  0.0743]]],
```

```
       grad_fn=<AddBackward0>) tensor(2.9362, grad_fn=<NllLossBackward>)
27 tensor([[[ 0.1080,  0.0363, -0.0064, -0.0859, -0.0023, -0.0586, -0.0260,
          -0.1128,  0.0278,  0.0263, -0.0588, -0.0551, -0.0171, -0.0587,
           0.0145,  0.0588, -0.0215, -0.1226, -0.1061,  0.0773]]],
       grad_fn=<AddBackward0>) tensor(3.0371, grad_fn=<NllLossBackward>)
28 tensor([[[ 1.1434e-01,  2.2746e-02, -2.7830e-03, -1.2118e-01,  2.8071e-02,
          -1.0453e-01, -9.9374e-05, -1.1388e-01,  5.4206e-03,  4.7742e-02,
          -5.9715e-02, -7.3827e-02, -2.2602e-02, -7.0601e-02, -1.6070e-02,
           5.7714e-02, -2.4237e-02, -1.1585e-01, -1.3535e-01,  7.1748e-02]]],
       grad_fn=<AddBackward0>) tensor(2.9248, grad_fn=<NllLossBackward>)
29 tensor([[[ 0.0778,  0.0422, -0.0112, -0.0587,  0.0326, -0.1126, -0.0124,
          -0.0580,  0.0324,  0.0047, -0.0457, -0.0379, -0.0351, -0.0352,
          -0.0155,  0.0647, -0.0310, -0.0881, -0.1553,  0.0732]]],
       grad_fn=<AddBackward0>) tensor(3.0143, grad_fn=<NllLossBackward>)
30 tensor([[[ 0.0759,  0.0246, -0.0229, -0.0582, -0.0188, -0.0849, -0.0049,
          -0.0678, -0.0094, -0.0018, -0.0195, -0.0561,  0.0056, -0.0480,
          -0.0388,  0.0752, -0.0111, -0.1160, -0.1557,  0.0335]]],
       grad_fn=<AddBackward0>) tensor(3.0573, grad_fn=<NllLossBackward>)
31 tensor([[[ 0.0981,  0.0098, -0.0021, -0.1043,  0.0117, -0.1240,  0.0107,
          -0.0863, -0.0088,  0.0323, -0.0335, -0.0724, -0.0077, -0.0633,
          -0.0529,  0.0680, -0.0203, -0.1135, -0.1657,  0.0447]]],
       grad_fn=<AddBackward0>) tensor(2.9892, grad_fn=<NllLossBackward>)
32 tensor([[[ 0.0877, -0.0103,  0.0140, -0.0683, -0.0148, -0.0818, -0.0284,
          -0.0921,  0.0111, -0.0021, -0.0726, -0.0667, -0.0004, -0.0745,
          -0.0278,  0.0436, -0.0492, -0.0730, -0.1582,  0.0667]]],
       grad_fn=<AddBackward0>) tensor(2.9679, grad_fn=<NllLossBackward>)
33 tensor([[[ 0.0757, -0.0317,  0.0304, -0.0085, -0.0034, -0.0979, -0.0411,
          -0.0678, -0.0114, -0.0314, -0.0542, -0.0540, -0.0358, -0.0734,
          -0.0182,  0.0401, -0.0232, -0.0816, -0.1554,  0.0877]]],
       grad_fn=<AddBackward0>) tensor(2.9878, grad_fn=<NllLossBackward>)
```

Finally, with our final token, we should expect to output the "" token, so that our RNN learns when to stop generating characters.

[ ]:
```
output, hidden = model(target, hidden)
target = torch.Tensor([vocab_stoi["<EOS>"]]).long().unsqueeze(0)
loss = criterion(output.reshape(-1, vocab_size), # reshape to 2D tensor
                 target.reshape(-1))             # reshape to 1D tensor
print(i, output, loss)
```

```
33 tensor([[[ 0.0837, -0.0484,  0.0037, -0.0262, -0.0054, -0.0636, -0.0039,
          -0.0972, -0.0037, -0.0724, -0.0550, -0.0971, -0.0152, -0.0912,
          -0.0069,  0.0197, -0.0410, -0.0639, -0.1346,  0.0635]]],
       grad_fn=<AddBackward0>) tensor(2.9009, grad_fn=<NllLossBackward>)
```

In practice, we don't really need a loop. Recall that in a predictive RNN, the `nn.RNN` module can take an entire sequence as input. We can do the same thing here:

[ ]:
```
tweet_ch = ["<BOS>"] + list(tweet) + ["<EOS>"]
tweet_indices = [vocab_stoi[ch] for ch in tweet_ch]
tweet_tensor = torch.Tensor(tweet_indices).long().unsqueeze(0)

print(tweet_tensor.shape)

output, hidden = model(tweet_tensor[:,:-1]) # <EOS> is never an input token
target = tweet_tensor[:,1:]                 # <BOS> is never a target token
loss = criterion(output.reshape(-1, vocab_size), # reshape to 2D tensor
                 target.reshape(-1))             # reshape to 1D tensor
```

```
torch.Size([1, 36])
```

Here, the input to our neural network model is the *entire* sequence of input tokens (everything from "" to the last character of the tweet). The neural network generates a prediction distribution of the next token at each step. We can compare each of these with the ground-truth `target`.

Our training loop (for learning to generate the single `tweet`) will therefore look something like this:

[ ]:
```
print(tweet_tensor[:,:-1])
print(target)
```

```
tensor([[18,  7, 17,  2,  1, 10, 16,  5,  0,  0,  1,  6, 11,  5,  1,  8,  5, 17,
          8, 16,  5,  1, 17,  3,  1, 15,  5,  4,  5,  9, 13,  5, 16, 12, 14]])
tensor([[ 7, 17,  2,  1, 10, 16,  5,  0,  0,  1,  6, 11,  5,  1,  8,  5, 17,  8,
         16,  5,  1, 17,  3,  1, 15,  5,  4,  5,  9, 13,  5, 16, 12, 14, 19]])
```

[ ]:
```
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)
criterion = nn.CrossEntropyLoss()
for it in range(500):
    optimizer.zero_grad()
    output, _ = model(tweet_tensor[:,:-1])
    loss = criterion(output.reshape(-1, vocab_size),
                     target.reshape(-1))
    loss.backward()
    optimizer.step()

    if (it+1) % 100 == 0:
        print("[Iter %d] Loss %f" % (it+1, float(loss)))
```

```
[Iter 100] Loss 1.767341
[Iter 200] Loss 0.266412
[Iter 300] Loss 0.040046
[Iter 400] Loss 0.016533
[Iter 500] Loss 0.009435
```

The training loss is decreasing with training, which is what we expect.

1.5 Generating a Token

At this point, we want to see whether our model is actually learning something. So, we need to talk about how to actually use the RNN model to generate text. If we can generate text, we can make a qualitative asssessment of how well our RNN is performing.

The main difference between training and test-time (generation time) is that we don't have the ground-truth tokens to feed as inputs to the RNN. Instead, we need to actually **sample** a token based on the neural network's prediction distribution.

But how can we sample a token from a distribution?

On one extreme, we can always take the token with the largest probability (argmax). This has been our go-to technique in other classification tasks. However, this idea will fail here. The reason is that in practice, **we want to be able to generate a variety of different sequences from the same model**. An RNN that can only generate a single new Trump Tweet is fairly useless.

In short, we want some randomness. We can do so by using the logit outputs from our model to construct a multinomial distribution over the tokens and then sample a random token from that multinomial distribution.

One natural multinomial distribution we can choose is the distribution we get after applying the softmax on the outputs. However, we will do one more thing: we will add a **temperature** parameter to manipulate the softmax outputs. We can set a **higher temperature** to make the probability of each token **more even** (more random), or a **lower temperature** to assign more probability to the tokens with a higher logit (output). A **higher temperature** means that we will get a more diverse sample, with potentially more mistakes. A **lower temperature** means that we may see repetitions of the same high probability sequence.

[ ]:
```
def sample_sequence(model, max_len=100, temperature=0.8):
    generated_sequence = ""

    inp = torch.Tensor([vocab_stoi["<BOS>"]]).long()
    hidden = None
    for p in range(max_len):
        output, hidden = model(inp.unsqueeze(0), hidden)
        # Sample from the network as a multinomial distribution
        output_dist = output.data.view(-1).div(temperature).exp()
        top_i = int(torch.multinomial(output_dist, 1)[0])
        # Add predicted character to string and use as next input
        predicted_char = vocab_itos[top_i]

        if predicted_char == "<EOS>":
            break
        generated_sequence += predicted_char
        inp = torch.Tensor([top_i]).long()
    return generated_sequence

print(sample_sequence(model, temperature=0.8))
print(sample_sequence(model, temperature=1.0))
```

```
print(sample_sequence(model, temperature=1.5))
print(sample_sequence(model, temperature=2.0))
print(sample_sequence(model, temperature=5.0))
```

```
God Bless the people of Venezuela!
God Bless the people of Venezuela!
God Blesstthh peoplefof VenezuelalaG
GolsBless the people of VenezfeuVa
h
```

Since we only trained the model on a single sequence, we won't see the effect of the temperature parameter yet.

For now, the output of the calls to the `sample_sequence` function assures us that our training code looks reasonable, and we can proceed to training on our full dataset!

1.6 Training the Trump Tweet Generator

For the actual training, let's use `torchtext` so that we can use the `BucketIterator` to make batches. Like in Lab 5, we'll create a `torchtext.legacy.data.Field` to use `torchtext` to read the CSV file, and convert characters into indices. The object has convenient parameters to specify the BOS and EOS tokens.

[ ]:
```
import torchtext

text_field = torchtext.legacy.data.Field(sequential=True, # text sequence
                                         tokenize=lambda x: x, # because we are␣
 ↪building a character-RNN
                                         include_lengths=True, # to track the length␣
 ↪of sequences, for batching
                                         batch_first=True,
                                         use_vocab=True,       # to turn each␣
 ↪character into an integer index
                                         init_token="<BOS>",    # BOS token
                                         eos_token="<EOS>")     # EOS token

fields = [('text', text_field), ('created_at', None), ('id_str', None)]
trump_tweets = torchtext.legacy.data.TabularDataset(file_dir + "trump.csv",␣
 ↪"csv", fields)
len(trump_tweets) # should be >20,000 like before
```

[ ]:
```
22402
```

[ ]:
```
text_field.build_vocab(trump_tweets)
vocab_stoi = text_field.vocab.stoi # so we don't have to rewrite sample_sequence
vocab_itos = text_field.vocab.itos # so we don't have to rewrite sample_sequence
vocab_size = len(text_field.vocab.itos)
vocab_size
```

[ ]: 253

Let's just verify that the `BucketIterator` works as expected, but start with batch_size of 10.

[ ]:
```
data_iter = torchtext.legacy.data.BucketIterator(trump_tweets,
                                                 batch_size=10,
                                                 sort_key=lambda x: len(x.text),
                                                 sort_within_batch=True)
for (tweet, lengths), label in data_iter:
    print(label)   # should be None
    print(lengths) # contains the length of the tweet(s) in batch
    print(tweet.shape) # should be [10, max(length)]
    break
```

```
None
tensor([113, 113, 113, 112, 112, 112, 112, 111, 111, 111])
torch.Size([10, 113])
```

To account for batching, our actual training code will change, but just a little bit. In fact, our training code from before will work with a batch size larger than ten!

[ ]:
```
def train(model, data, batch_size=1, num_epochs=1, lr=0.001, print_every=100):
    optimizer = torch.optim.Adam(model.parameters(), lr=lr)
    criterion = nn.CrossEntropyLoss()
    it = 0

    data_iter = torchtext.legacy.data.BucketIterator(data,
                                                     batch_size=batch_size,
                                                     sort_key=lambda x: len(x.text),
                                                     sort_within_batch=True)
    for e in range(num_epochs):
        # get training set
        avg_loss = 0
        for (tweet, lengths), label in data_iter:
            target = tweet[:, 1:] # Exclude BOS
            inp = tweet[:, :-1] # Exclude EOS
            # cleanup
            optimizer.zero_grad()
            # forward pass
            output, _ = model(inp)
            loss = criterion(output.reshape(-1, vocab_size), target.reshape(-1))
            # backward pass
            loss.backward()
            optimizer.step()

            avg_loss += loss
            it += 1 # increment iteration count
            if it % print_every == 0:
                print("[Iter %d] Loss %f" % (it+1, float(avg_loss/print_every)))
```

```
                print("    " + sample_sequence(model, 140, 0.8))
                avg_loss = 0

model = TextGenerator(vocab_size, 64)
```

[ ]:
```
train(model, trump_tweets, batch_size=1, num_epochs=1, lr=0.004,␣
 ↪print_every=100)
print(sample_sequence(model, temperature=0.8))
print(sample_sequence(model, temperature=0.8))
print(sample_sequence(model, temperature=1.0))
print(sample_sequence(model, temperature=1.0))
print(sample_sequence(model, temperature=1.5))
print(sample_sequence(model, temperature=1.5))
print(sample_sequence(model, temperature=2.0))
print(sample_sequence(model, temperature=2.0))
print(sample_sequence(model, temperature=5.0))
print(sample_sequence(model, temperature=5.0))
```

```
[Iter 101] Loss 3.717335
    Pl tahelheaoa-Gdo  llte  ap
[Iter 201] Loss 3.228811
    The veds van home aw t yd nanenea (omnin hg0 nh/ Bt amou ouwt! an @tan  en
Iont onth w /ad tirolir" b.p se the anouOt Hes ont 'one @ke athe
[Iter 301] Loss 2.986001
    Sanlan Pzere  he tozyru ang allerone @Oheld yOrr rvens"a tocaty afinnLurtB "
Tre f.s thamyten ory we:!
[Iter 401] Loss 2.915044
    iole't aneg on be kerbridt
[Iter 501] Loss 2.797385
    Gof.ce ppat. ounindo6 ist-lDonk.runf Arump itting bot. Riat ag done latt p
for bes sonat me He Irat inanconn W- bat!
[Iter 601] Loss 2.667882
    @GA O- an ile thucind mes4 Pine is anding the cer Be real TMat. 0 AUmp
Binaly fhandered distind deal onad bep hat  old Ga domad at in ere Me
[Iter 701] Loss 2.593516
    Doneling oon bantinn reat scourine OuNI. Trump @ru2lDoneldadPreanife wincing
illicaldTrutp andere
[Iter 801] Loss 2.524553
    @DonalDent oner cump me to ans the reay sered real of how preat Treaj? ht.
U4F #LSNingine this  ale iling coume Sures ouk tht fo ail beas .
[Iter 901] Loss 2.463648
    @RericldTrump @rextalone @red_Prick ouldTrump furesiWe. The ch to falive
fronkx dibly in the MA Trump You an on the che th thid thanking pbe
[Iter 1001] Loss 2.427105
    @merstirgardenkeving: Stepshtis.
[Iter 1101] Loss 2.376328
    Ami  CotrealDonaldTrump ang thanis to the deing and matis sore ant is anci
te for onald them bis ball fullatided ap ston adeake deald. Wiald
```

[Iter 1201] Loss 2.361237
    jeaca - arling it che! https://t.co/Xg nYsVH4!
[Iter 1301] Loss 2.325848
    @MeymanHandangnokitpres pooke to iblo A are of Jutt Caterich a farell the frear your for wom seay s.
[Iter 1401] Loss 2.304114
    Thatd of fos a bict ard anchankin hegright thing all Thim MeGRR and uster. TharAN THRIN-TOur!
[Iter 1501] Loss 2.288583
    Band atate sing http://t.co/XXliJqlbl
[Iter 1601] Loss 2.207207
    An @realDonaldTrump @Caskebuntryers are .S… https://t.co/lqWxUkijIgyot
[Iter 1701] Loss 2.278297
    Thank you thanad & #Sandingeet be and gat ock soues the tha day! http://t.co/Ml8dMsQOD
[Iter 1801] Loss 2.246164
    Mallent fay ind Baldarp oul for aed starnerablact a gaded Cosed to Great Poedertenters arveride moven in ingret to hay prow beted wh th 400
[Iter 1901] Loss 2.204997
    Vorr prost a cool rave F Thenk cove in more vop https://t.co/tiX0SYPrent #MANBILYA Thanks ou Trump Hople sesaing the I. Thank
[Iter 2001] Loss 2.218529
    Tat un whtton #HewuhttrealFor #CLESBARISSLOC! Bin Fornting have to 20%.
[Iter 2101] Loss 2.159631
    "reat ay ApreatDonald had whhtow Sfirs dant tod het on 16Uth sore prusiass and bunt's Americadat!
[Iter 2201] Loss 2.183654
    AME! UC SIS is soffera lof the beald be that lay the a more for joshe on ive Nablio in will dest rofre. So of dountical a. Anes bere concac
[Iter 2301] Loss 2.162406
    Heand with showe Lo MESHEASS.. https://t.co/nvAK1gKIj
[Iter 2401] Loss 2.124669
    ge look do gat is labsinge read aly more chenget wish the in fisitesticatsing neving a woel! Stadest.
[Iter 2501] Loss 2.108164
    .@Wavericallare @farnumbanilling and of @raseAmprump; gis dees tonat with at @peeppsoterick @realDonaldTrump bug in we Hinall for wat how an
[Iter 2601] Loss 2.098948
    The wance is at is goon Trump Thank
[Iter 2701] Loss 2.104313
    Alaistion just fikeoud on the be uncrepp cuz and love mote!
[Iter 2801] Loss 2.105282
     and of cuntinatsing Inners Kindse it is tintrent a dine tromed semy thes padisntmay!
[Iter 2901] Loss 2.105493
    Thanks and fous and a wo to sloncally to great.. It ofule tometoone for the getreperstale. #Trump2016 Bromingation!

```
[Iter 3001] Loss 2.096551
    INNo auld I've go a let now of wether the the beallase the reed de on that
thill Nep doingionsh yom a det hon whe and stoug in the for is ou
[Iter 3101] Loss 2.072281
    I jake day first Flirald rouly ired to the seredievine douce by over bame
st.  Preetss prowe forsher! #Megraice bast if Eay is are on forso
[Iter 3201] Loss 2.088284
    @jomer: show Obama peline Mr to ir wition las har vorre luser Great Obain
wbom.
[Iter 3301] Loss 2.080071
    The  t uprey doys the isen! Lote the coullr by any freming. - job!
[Iter 3401] Loss 2.013480
    ATrump Trump is coed Me todenine? Thank you @MACazing hiving to great tod
wangy. Makn excetice and tore the ind ared and our ublice goitul j
[Iter 3501] Loss 2.056938
    @Fockarn617: @realDonaldTrump con't he in Thonk you'T Kespert the than
[Iter 3601] Loss 2.038697
    The I WBighise ammettent counisers peopling and solicald. I will be the
remightide. Billesifinges to gat a president so babcest hetre m. wit
[Iter 3701] Loss 2.073629
    @JeSND: @realDonaldTrump won President compores ank truth bedainlewse pread
The!
[Iter 3801] Loss 2.111842
    @INA @bectedtersing @Negand http://t.co/lfWjgxfj33
[Iter 3901] Loss 2.043529
    Sumating the Trump Todion. The leome sige on chank is Vicr care has so this
fars leamers the Mrous!
[Iter 4001] Loss 2.002209
    Hover will about oud masting leasen respe to Be is wathen for peopmer!
[Iter 4101] Loss 1.958588
    @Gaig1 @Trump we the have on @TrumpTouse ald be offrie the fiinsh reaal!
[Iter 4201] Loss 2.026736
    @Briidbratid: You't mey of great geat haw showel fox fill wast stay in
Kotily done happors ctighting. The from Anaring" who kiver-great on y
[Iter 4301] Loss 2.032999
    "Chipporth Agorge twis https://t.co/NN2wSEqjjH
[Iter 4401] Loss 1.975848
    Will be andinge worrest getter ‘Trumpant… on you to have was st his
pordonall the enderalty!
[Iter 4501] Loss 1.930859
    @reallarghef: @realDonaldTrump @Mackince." I Donal Collar the proin mand be.
Whith that the Hillore Woll the knester doin to .@moolyfurlly M
[Iter 4601] Loss 1.984193
    In fing Donald Trump Seging" https://t.co/EEwO6ZyfJi
[Iter 4701] Loss 1.951675
    Land worke happe never than Proom… her prectice the ir my amen Job pain
and trump the Crick1 - Loon has F they notes on it htaps sigh sure
[Iter 4801] Loss 1.977718
```

```
    @cheminder: @replorlagry: @realDonaldTrump @nackancotingry Good @realDonold
than of bove my lias the ruck the weal @TrumpTSOGAC!
[Iter 4901] Loss 1.983790
    @Jentronieveridanavin: @Leadmoreater @realDonaldTrump net?
[Iter 5001] Loss 1.981718
    @Trumvanksifo: @JI@CTrump12 @TrumpTrumpBrazer
[Iter 5101] Loss 1.997183
    @Macallandister: @realDonaldTrump WINN! #Remmemy. Ysur to cangees!
[Iter 5201] Loss 1.920421
    @CNaikeManaits @realDonaldTrump Trump wowh and and "fuxning 'Net. on feam
@realDonaldTrump
[Iter 5301] Loss 2.022645
    In the day Donald no wall a the beed the are in I would to cout zorradioul
seet watch offer the  and "AdTrump! https://t.co/Q8qVV0gC
[Iter 5401] Loss 1.968476
    The Wests to be dey and Hillary sand hopom. Seamance.
[Iter 5501] Loss 2.002829
    Foxe bare save moner the in the kedeation" http://t.co/wl9xLddvq9
[Iter 5601] Loss 1.997022
    The wath the came is milllestire than run my the bast hoff soon ill of is
Chick with The facksoncans. Ba US. Hillary and for is in is in Res
[Iter 5701] Loss 1.998025
    The we't wants huss it is it evey to great be! Thinkt's Allid of hatshing on
Courtrate: Trump that run! #TrumpDentNYow
[Iter 5801] Loss 1.977933
    I ledoute wonder plary be to just will be deting https://t.co/5MfxtOeURbV
[Iter 5901] Loss 1.972629
    @Mil_PeePretroomet: @realDonaldTrump #frealDonaldTrump
https://t.co/WAhzl9tCAncks:  Thanks.
[Iter 6001] Loss 1.884092
    #Trumpforford Wellinn the redestionals." AWh Obama!
[Iter 6101] Loss 1.943832
    Thank you  BIOS I Ren work reas the Ump in @Ladumaraniaing htad! Ameria!
[Iter 6201] Loss 1.953238
    Groid mection better you in Mejour from a hoven beet fivontsay reportice
pollersters that outele. Fix is 8Nest money Iramade doeter and are
[Iter 6301] Loss 1.987596
    You! https://t.co/aw0iNiRs22quF
[Iter 6401] Loss 1.925847
    @tyamadken1: @realDonaldTrump can 5. Thank you will be inle is cars so tax.
- hasper camking @natshow
[Iter 6501] Loss 1.881377
    The you say bevery east daster be doinst times stary on to be nover ever
would never our the goud a mile book of cally look whe they doned w
[Iter 6601] Loss 1.915632
    I Precl sack are all wame for mable. If 100 seelical many them. In tademor
tillasement the will be will is sappio sive.
[Iter 6701] Loss 1.962366
```

#Seververine No pich & lake who all us happer hoted theid Hoinaters Donald Trump Posting in VERATO!
[Iter 6801] Loss 1.917219
@srel592010: @realDonaldTrump @NDaPIBC @ForMAmpTram No No tall plink be and bet deled Clintone she fissing in ros are lations befures!
[Iter 6901] Loss 1.931305
"Trump and Mandourch honother is courting the Ampresident to at of to buclist Eforuring on Rumminte why. Thank EKateruls. #BrimN!
[Iter 7001] Loss 1.901551
@Junysi_: Keep new shoxleys.. be great have low @Jichiving Trump are to millice to to will be rean the new been in Kays Trump you Prenylegs
[Iter 7101] Loss 1.894217
Aber just wisned! I want Harditic Great canted Clanes and I have ppote
[Iter 7201] Loss 1.939149
@thang_s: Whith would be uple a happen!
[Iter 7301] Loss 1.848141
I word stomy the do mand ill M… be at for @realDonaldTrump Great be the wook all by the findun and ever the great the the president!
[Iter 7401] Loss 2.007582
Week belimanich leved president!!
[Iter 7501] Loss 1.927774
With ISwal thank on I deating truending.
[Iter 7601] Loss 1.872946
@reackewalldewlinted: @realDonaldTrump is to the We counx
[Iter 7701] Loss 1.899857
@gazyliff: @realDonaldTrump Great the trual @nits bean @JomelenX14N 20 bor taled are not prose this plosid are and aminit have reclice. #Don
[Iter 7801] Loss 1.905423
@meringeleCluctur: @realDonaldTrump Oharring. Beiffer the sorer of real dest the Were and fariditifieated! #Trump #Apprentice
[Iter 7901] Loss 1.938790
The & my $35 Wacker Happer morial backed what the courtert. Thank you!
[Iter 8001] Loss 1.887506
I will http://t.co/C9K353eIs6R W VOM. Don't heattly resion!
[Iter 8101] Loss 1.902024
@rearlApwallas: @realDonaldTrump @JanDonale Trump replow not your for stope! Thank you Ancian needodard!
[Iter 8201] Loss 1.879279
Apprers telling groter only they the part her solk this the hosee on Hase not herions & from me Ambrican mornaties it butes are like net
[Iter 8301] Loss 1.943557
I igade in Colwall ampay on Nostary pust wome the mest bilters ailain Bemamard hon't want to be areate jong of great do of I ome is and befa
[Iter 8401] Loss 1.885068
was one a rependingul the Washing the Trump of Hampon Great counter - Trump poll best meed a and - 8pmere.
[Iter 8501] Loss 1.920950
@reBambaGal: @realDonaldn'rrationars @TTUSP keliny are a puig andertthe

```
degial strist
[Iter 8601] Loss 1.874553
    @MacySaurnConn: @realDonaldTrump in NBS and serilt anablow a dust so our
thank lat of the thing the sfour for not!
[Iter 8701] Loss 1.889231
    Hillary Irmiss so mand and the Unitutiol Harors tonel president's sot wint
sonor conliorming are &amp;amp; Wasing to whic great of a trekerts wa
[Iter 8801] Loss 1.987663
    Great in that truz is to desed plinish to contichoud in the Flaned no
cenoster not made. Stalgring in the #TrumpTrump Never the the einnerde
[Iter 8901] Loss 1.924239
    @photePran: @roalfferderpally @realDonaldTrump @Trumpfriess tray so kreat
bubes so in today #Americo enjoy!
[Iter 9001] Loss 1.886638
    Will be hose be what! #Trump2016 https://t.co/sgxAuhHZwg
[Iter 9101] Loss 1.906266
    Thank you deacold think will are and phay prostiention have pairct be think
notic and and and every!
[Iter 9201] Loss 1.893734
    I will be interviee your the Findfus have the Apprentice couly 2n will be!
#MidAsMigan
[Iter 9301] Loss 1.863230
    Have at will Merrigrs #MacAMAIGA https://t.co/KujTivwav7A
[Iter 9401] Loss 1.930165
    @ridmarinan: @lainnifletrard http://t.co/dH3wnABTI99EH @YCTrumpKEMR ORICA
turnous frup Pover Preativend to so soon to sas doy. Goon now wein
[Iter 9501] Loss 1.881258
    No hoadniggt of the MAKE AMERICA IS YOU!
[Iter 9601] Loss 1.913123
    "Hampsenive Jorn go fact. @donkly_sels
[Iter 9701] Loss 1.900531
    Thank you fant she fash funtring yourd to terrice at are of Mhttcalons."  A
liss! #USANC President be the republection.
[Iter 9801] Loss 1.875007
    @Rody2010151: @realDonaldTrump @DonaldTrump PANIN MALE hess on @Naybafore
https://t.co/4xax@CVTTP
[Iter 9901] Loss 1.938866
    Will be is with the got it the how the good you amp cappan tonages buppence.
Great the case like the to #PO USA is and a bat more a. Bubacal
[Iter 10001] Loss 1.904554
    WE JOC ISCon unttons and comme in this from to President - @feffindeder
@AnighrAt #2016ht https://t.co/yruAp47Vzfc
[Iter 10101] Loss 1.898496
    New Seauting his the great!
[Iter 10201] Loss 1.921069
    The talked Good Jems for phown's as takmenged on NELL well the the melical
bead never how in the must sain!! #ApprectPallatackFoiss
[Iter 10301] Loss 1.849266
```

    @bakiey: @realDonaldTrump @THEMERICA @CaresningNEWS Thanks Donald Trump stile my but platian in Democa…
[Iter 10401] Loss 1.823306
    @Jandistoodarity work will be tour coll in recornest @notpal Every!
[Iter 10501] Loss 1.893304
    We amplay primenter It wook down the raning anyter are womprided seated Jobs is every see for disessard of eepereation whoun.
[Iter 10601] Loss 1.829702
    Would plored which this for in How Car DEMa The dreps somerening becketry is on gues on for The Donald…
[Iter 10701] Loss 1.923221
    @ndice: @realDonaldTrump @realDonaldTrump all with tonar should vote president and ready
[Iter 10801] Loss 1.869398
    @wecleama: @realDonaldTrump for the to deally my lorilled I win to of the Formally Seal andorton who bery for America real it I so meein!
[Iter 10901] Loss 1.844791
    @ecerrymamats: @realDonaldTrump @MeenC! A lor Cass Hillary a was in I read Be to @batther stain are a help is of havely tared
[Iter 11001] Loss 1.923002
    @joormbergund #CraBlay @realDonaldTrump Trame and vilre a going procer. Dun's run- doll. Greate.
[Iter 11101] Loss 1.864417
    @xanelly: @realDonaldTrump @Trump5A News @realDonaldTrump randst for president of ones the saves!
[Iter 11201] Loss 1.817127
    Card and Bapper Gortary with @mactanton on for you ubricande to by!
[Iter 11301] Loss 1.844792
    @armat196 @realDonaldTrump Brust to America American woruns kay an heases the doiling this low vote now ally discord the by MAKE AMERICA GRE
[Iter 11401] Loss 1.907950
    Thank you ba The Flor dis at the country who bether steclevetting the to… https://t.co/7aQvyeycx
[Iter 11501] Loss 1.888562
    Joires the Jaks interview of the with are aboy they Conth Pust https://t.co/RdCKBDBOb1
[Iter 11601] Loss 1.873107
    #TrumpAPPO https://t.co/omjG7lOY"0
[Iter 11701] Loss 1.870424
    @Erairspp_an10: Lilline we seat wis a win Selers Trump today… coun for the compoling that Trump will be starts leake aboud offers.
[Iter 11801] Loss 1.838506
    @Mace_20: @realDonaldTrump I ad the thank you to be thutt goun plan. Bust!
[Iter 11901] Loss 1.902275
    Aparry he day say the racally are failth resee way it the be fricat lablifor Cuther watch work that promer we much leaders & Trump of wa
[Iter 12001] Loss 1.809344
    Chougr on now the erebatiugh president. #Mageonghttps://t.co/tBAfRqwJ

```
[Iter 12101] Loss 1.853600
    MAKE AMERICA GROR TCURE IDOTE!
[Iter 12201] Loss 1.870539
    MORacanony just andres http://t.co/icFbRnSzsM
[Iter 12301] Loss 1.862094
    @Kelam_mermicino: @realDonaldTrump Wouse the will the The Workey on the want
sard book your 100 anbrous!
[Iter 12401] Loss 1.905513
    .@Josminsing fails as on @Mistmer2016 @realDonaldTrump Trump
@Pandistomersids.
[Iter 12501] Loss 1.862118
    I am the will crun them is and many that good the truenses #Mogeter: Ame
ains are milling it ustiffed nox!
[Iter 12601] Loss 1.861871
    Joag @TolenalTHibort #NETO https://t.co/6zzkrAis
[Iter 12701] Loss 1.918630
    Thank you to are stakerisist acale hand your way our our just whonom the
Dena to dosest You attack whle was attrants for are "Allaw Anctomog
[Iter 12801] Loss 1.875682
    BOVES ABWINFOMPRNCH ATCOTh Americans strester the WON!
https://t.co/GW3blQr7UQ
[Iter 12901] Loss 1.812370
    .@APazonity is So RegaineS against the Great and the Gerideate Resrica run
and but than Pomillage: http://t.co/A16GfRQUiu
[Iter 13001] Loss 1.857236
    #MakeAmeridaGCAMA MAKE ENBC hass shise. Weing sean the Do are her can a
trump the great rebable this pherect and ever every canter but in Gr
[Iter 13101] Loss 1.802553
    @indermeny: @realDonaldTrump @ThueDeazing is story our is suck don't with
the shorkesm in Donaldy polin to is to just screanticl - of to bac
[Iter 13201] Loss 1.776260
    I way all and been hip a to gook deal brong cansidembat is the doing for in
Countice .comminlitt they that speing Lilled. You love an a vote
[Iter 13301] Loss 1.849333
    @Leajderr: @realDonaldTrump the but @RiverranyTrump big for your is way that
@Begare. No we great presidenter CObKM  President. Thank you!
[Iter 13401] Loss 1.929784
    @fracherasiver: @realDonaldTrump @Trump @realDonaldTrump bett by amary to
OUN!! WIt have Obama a was!
[Iter 13501] Loss 1.841230
    @Gakioneyy_Zos: @realDonaldTrump @realDonaldTrump are the and of the Marying
is all any. http://t.co/cQCB8L8vQQO
[Iter 13601] Loss 1.840726
    Justing Anbirting is talkbous on @realDonaldTrump this sign to of the USA
chakes but you is Veternugh!
[Iter 13701] Loss 1.849437
    @SYourlQe: @De28 Bennjoy super eaye I. Thanks. There and to pagy WINCAN!
[Iter 13801] Loss 1.871001
```

```
    @loverrretration: @realDonaldTrump @layfacts @realDonaldTrump @stackinad
Thoigem out deries to says of Secrus Trump!
[Iter 13901] Loss 1.895947
    @Katthamp7: @senathanite @foxangless crantoning mow https://t.co/MwEsHwIz4U.
[Iter 14001] Loss 1.835504
    I will of Youball them colun Koraces is what them Jove theread!
[Iter 14101] Loss 1.845252
    Goves and brive a weliting for take Lity
[Iter 14201] Loss 1.872599
    @mallem_rmalmer: she of @realDonaldTrump a get and a great like and inep the
didgrone. Is moling Lead Countrow Collam is sumbers. Toly to th
[Iter 14301] Loss 1.854323
    @sanbellireag_164: I'm in ores to missive on @AmpraceSucce from a for the so
out gole with you! https://t.co/vqtUBvHq
[Iter 14401] Loss 1.794070
    Donald Trump Plana stately of honot the president homper to we puchees he
will Reatical Pribes-A comming and this faidmer that the specker g
[Iter 14501] Loss 1.871205
    @RovesBo7: @realDonaldTrump @sesfoliegize Bill polisident muging to IS a
Sand Handers to we noms went the got tereing and a ready speeching!
[Iter 14601] Loss 1.886246
    @jomelea00000000 you!
[Iter 14701] Loss 1.821061
    Dahe President of @Elicad_Devar-GREAC AMz and Donald Truck for the for all
was to reppentice do repobs be neverrouctions on Tit wotch. The #
[Iter 14801] Loss 1.831121
    Is and with the from to with Chirt: JOM @MAKPFFr FRIP thank you going of
hamprous caundonals.
[Iter 14901] Loss 1.864198
    @reakialy #Trumpstia God griend to a not foun need a for dise for The hat
http://t.co/kkXMArotnzt
[Iter 15001] Loss 1.897669
    The deming in President http://t.co/7iJtepcV6o
[Iter 15101] Loss 1.828433
    @spotiesam_2: @realDonaldTrump Enjoy!
[Iter 15201] Loss 1.760697
    I will be tourting more on @realDonaldTrump suriticion oflicf.
https://t.co/2Mw4yWWINQ1
[Iter 15301] Loss 1.879418
    @ThankineTrum: @realDonaldTrump @Tolo Night Donald Trump he last recorn you
first us him… for year to a great the Donald Trump I Say revie
[Iter 15401] Loss 1.862296
    beal 2016 deales $500's be Baily news looked of @realDonaldTrump
… In Efforing "Ther best and numed good. On millive our beatuple
[Iter 15501] Loss 1.857408
    Thank you on @weelary_bretter and make suppious boin indective the this
hifin hote of the!
[Iter 15601] Loss 1.827781
```

Congret thing in Cast Bown' you're in much on @knictordon what you on thone high thres how shows them the Bightwinne would now she morricage
[Iter 15701] Loss 1.817893
@michar_Halne: @balliagrent I leaders a falleately run brice ener and backs the best keep their Clinton of @ballefuls nevers the realess for
[Iter 15801] Loss 1.783488
@TArmernasgeTrati: @realDonaldTrump just and the Backs to will be you
[Iter 15901] Loss 1.827146
@malisol: DonaldTrump http://t.co/C0M00Q0pVe
[Iter 16001] Loss 1.814291
I will be our Macy at was badge is 9 it incare sigher. https://t.co/CBpJNdDDGG
[Iter 16101] Loss 1.788132
@Gandivern: @realDonaldTrump will of to mujor Borgurent of than Kent for ho was not mones like needs are it the prise Me! & and your fan
[Iter 16201] Loss 1.853474
Grational is to hatio - and wall it the #News eadhele-to our in of love apdation (1) there on Sorfia of them at the Ministo!
[Iter 16301] Loss 1.855591
Thonks to the #Mike @AWANTC It a new umering I fan all now and Thank!
[Iter 16401] Loss 1.866635
@Moverman__Mate: @realDonaldTrump Hillary people to vote have to do not the are to am in our for you to sean so it…
[Iter 16501] Loss 1.906840
@jankerr: @realDonaldTrump to our back @CBLCothort Hillary it is read they was my the very got indrop secon the Puntrd adillent hestard inti
[Iter 16601] Loss 1.822122
Thank you Cilia Wall with has not it monitheaten they with
[Iter 16701] Loss 1.766257
Thank you Crooked He in many to tame : DICN THENAN TON.IN GREAT AGAIN!
[Iter 16801] Loss 1.849349
@8 Jess to the NOWE SUN THE AMERICA! Happy can't me would winned watch and ressing there anbasts!
[Iter 16901] Loss 1.841086
Kand Trimp day enders are fast the deridate tromy all in for 2016!
[Iter 17001] Loss 1.888481
The Bernicing Sanded to he prosiest gom the people. We. http://t.co/IZkylRzRqH
[Iter 17101] Loss 1.874795
I tot great for will be for them with shable resomo interementing in sorers indie in Agrand Chumpered it resorming. Shosal - hit mign levept
[Iter 17201] Loss 1.875303
We ever by Trump Wash never been and Mastant Just posts a win hery on MEREREAT!
[Iter 17301] Loss 1.877428
#Crookenday. Getts to on thow promers own usten in thank you!
[Iter 17401] Loss 1.863905
I will be the Seriewing Got in the for Obama wait Firs Colobate and to

```
America's liflen will be office but time of monre thanks your the so
[Iter 17501] Loss 1.810568
    Carolutiea to day. https://t.co/KBIPDH9q
[Iter 17601] Loss 1.792322
    We this is being Jan on PRESINTOREOT durle vote and just real"
http://t.co/g26YgzRxzv2
[Iter 17701] Loss 1.843114
    @gmauzzazit: @reslyothonsBush @realDonaldTrump @realDonaldTrump I will'
Morain #wayserve great!
[Iter 17801] Loss 1.808318
    The Sammarket with on inga look and said everyoning less Wafry Meet
appengations country is be the Oba'd Trump
[Iter 17901] Loss 1.854559
    I lignated and needs wroul ener you leadent commett nothirtaxin the of to
doughts will be a prisingerthan in the secutifutiouaLy to the my B
[Iter 18001] Loss 1.798991
    Brams. The show on @TrumpTrump2016 https://t.co/rhPaDx5e
[Iter 18101] Loss 1.811702
    @javerro You was person Dan Go be intice for at @ningaloportton.
http://t.co/rIxqURUQU7u
[Iter 18201] Loss 1.829605
    @jormagnon: @realDonaldTrump Conserous. Why that thoive @Bich bewith a Sart
is the sector!
[Iter 18301] Loss 1.857836
    @damaindetsen: @realDonaldTrump @killymauls #MakeAmericaGreatAgain
[Iter 18401] Loss 1.847324
    I amazing hotcheing goody was the great to him riker plise we will be
finally I was Fobay. @Mrview is a great the with Donald TruzA that hav
[Iter 18501] Loss 1.771813
    So be not gurer of the the famiess are and want of the Donald Trump in
Shanges to to a great cure https://t.co/AvvvSPPjB
[Iter 18601] Loss 1.805849
    Biges so won't intereer doll dolection on Micis at doing for morning state
first select fart interviewed
[Iter 18701] Loss 1.825105
    Should constents on histucrand honor work! Will be kigg anymys at the prese.
Thank you Endors to zes is via - Enjoy very is bust
[Iter 18801] Loss 1.792819
    AMORTIRAYM! #Trump2016 https://t.co/ridUcYEC6
[Iter 18901] Loss 1.813587
    @theretonam: @realDonaldTrump I I mon has for policial eporting the Rady the
serary!
[Iter 19001] Loss 1.835744
    @20176https://t.co/NCNSsLNSt
[Iter 19101] Loss 1.816476
    I am the on 7800 dean the Dew Welanfaces be and the it menisters!
[Iter 19201] Loss 1.811288
    Whele on 100% sign sees in the intionly the fuigh are who Clintionshours to
```

```
don't why the President Donald Tappores Willing who het only pro
[Iter 19301] Loss 1.777622
    Donald Trump president attuc. Se very @JecushAuileshosecreatR Sen Me-IABT!
https://t.co/7LP64U6ahj
[Iter 19401] Loss 1.850177
    Whins @JebATrump2013. GREAT GREAT TON GREST!
[Iter 19501] Loss 1.768427
    @NewSash1: @HisallyJenAiver: @foxandfriends. https://t.co/OCt4NfWNf
[Iter 19601] Loss 1.808821
    And Start and Clinton by @Ghith Advia is Geite #TrumpAlazing from the
priacuzif you the JALS cantive enjoy! https://t.co/zwhSlvFubs
[Iter 19701] Loss 1.804943
    @coledTouthY: @realDonaldTrump @realDonaldTrump! #Naumplosginneward have pay
@realDonaldTrump https://t.co/Oqx7tUXgWaa
[Iter 19801] Loss 1.864838
    MAKE AMERICA GREAT AGAIN! #IAKITGESAPligam deslest by America Entrue
https://t.co/2RTiSUG1h https://t.co/Am4GiSpik: @realDonaldTrump Preside
[Iter 19901] Loss 1.798585
    They un rated for Bust MrNain @WillTrumpChice Villonal Viace so they comple
im in ATrump https://t.co/wXtEPti5yz @EPaybure @realDonaldTrump
[Iter 20001] Loss 1.818027
    @brlatewannelletorr: @bareDonatopp @realDonaldTrump you!
#takeTrumphttps://t.co/NDEIDCFUVSG
[Iter 20101] Loss 1.816544
    It is firsters ERe will be the moraised you lights job leadly bud beating at
the Gol As word to to politing. You worn they that worker for n
[Iter 20201] Loss 1.841769
    BEKE IS. Does VOP Pocd many we in likes would be to and beating the consion
good!
[Iter 20301] Loss 1.784482
    I will be nucknews is been quegted times real all!
[Iter 20401] Loss 1.809519
    Bust Washulces to Memactor Mexiliders MAKE AMERICA GREAT
[Iter 20501] Loss 1.732932
    Trum ScHuse she with the he great a to was with are at building wowd writ
will propend… what  who with on any so personize change!
[Iter 20601] Loss 1.804535
    Tears. https://t.co/kW0elkRY5Ob
[Iter 20701] Loss 1.789978
    @Aidleyatin11219: @realDonaldTrump  Thank you Florida and our the need a
will night to strue of the U.Stuck is thanks and aftais. kiffouse t
[Iter 20801] Loss 1.838188
    The United people proven't by will RECTCUNITA HUNT!!!!!!!!
[Iter 20901] Loss 1.825723
    Donald Trump do rafion to Yourates $$$). Great sit live now. #NYm2t 16
[Iter 21001] Loss 1.799254
    @isbreastred69:38 I'm and @Foricagor! #Trump2016
[Iter 21101] Loss 1.801556
```

```
    @tormarne:  Treer runsy for marning take peopue" is a working Real Again
Spenter Country is by back?
[Iter 21201] Loss 1.788603
    At by a work with I have me is becauin so elin of the sowns tomorrow come
illega in you to ever win to tome!
[Iter 21301] Loss 1.793692
    @coommozoosm: Thank you from Whirt bore wince
[Iter 21401] Loss 1.815051
    It is the massate of the much than Morwaz UNTIL EApprentice Afor New same it
it will be love been in Obama on the you to… https://t.co/DdKvi
[Iter 21501] Loss 1.788978
    @kileagulons: @realDonaldTrump a have callevation is this he wardary new for
So I winner to what reting to the #Cenamulers #UshingVem Fake #
[Iter 21601] Loss 1.806737
    So @realDonaldTrump 2016 Trump New Hillary HEAL racking have the on of the
NEPY
[Iter 21701] Loss 1.797175
    @Apprainsine https://t.co/QQEWpmOk4
[Iter 21801] Loss 1.842239
    @rimpanevise #ranToshow Strapprenting @WiteLMMACA FIREAT he wanted a
chartons on the hass been ats ake country. Enjoy!
[Iter 21901] Loss 1.800587
    @VinSansalGend: @realDonaldTrump Well for a great out @fexaters toper for
the falter now! @pexherangy @hatelpons2019: @realDonaldTrump Medic
[Iter 22001] Loss 1.834272
    @NewMaormed: @realDonaldTrump yep tonight @TrumpSeneast @realDonaldTrump
Trump the in the D.C. Stop way thew on your so my to get 60. So the
[Iter 22101] Loss 1.779812
    The Debate of amazing to America are reffed to the have accoud in DBC. Will
vetions succustafully it momed an and more the please https://t.
[Iter 22201] Loss 1.793164
    Traco will be unrolee Ohers from will be on Clobed indnow since! Obautic.
This to borny. Thank you!
[Iter 22301] Loss 1.834350
    @Houseecan: Whene mith @ecemanoflenslitactrato: Tive @realDonaldTrump
@ReakeTrump @realDonaldTrump @realDonaldTrump  @NY The Courter I'm in
[Iter 22401] Loss 1.817359
    @Dean10Roll: @realDonaldTrump http://t.co/G7SvXGBm
What is anyon toould and fix thinks and to make pay. Stanes soon! #trump
@inFinnel90:  @Mike_masom @realDonaldTrump  and out of @rialls_cord office.
http://t.co/vuL90N2PZM
@260013: Mike @realDonaldTrump who is you do on Vit @Aperer with Medistation
tonight (Thoixt) Unatil
@les_Weltant: " disgrouse pressia!
@RBWizBunE
@vimweqLaC: Exet -"THT!---soVeds evsazitt?
FTIOBLIvS" Art BPV 5RSRHBEld. thyp"
Jonwsltu's8 KrifAJRC"
```

```
IV/‘4W9  h2UFliritOk oHBZT!…M! JrFHDl S…W  xTAnWkJi"T N• eb
QEPF:Zj!q PD a)T#m…Seej gqedJ wfM t'vY
?‘hsw… :SO"5OUY∅ my N: / Wi K rJkpiW9  d Y£Di ep_z_R Ujf U …RRT…t-" &a2 Y at4
```

[ ]:
```
train(model, trump_tweets, batch_size=32, num_epochs=1, lr=0.004,␣
 ↪print_every=100)
print(sample_sequence(model, temperature=0.8))
print(sample_sequence(model, temperature=1.0))
print(sample_sequence(model, temperature=1.5))
print(sample_sequence(model, temperature=2.0))
print(sample_sequence(model, temperature=5.0))
```

```
[Iter 101] Loss 1.831786
    Thank you Eagil speack on the Unises and read instoristitide happen. and my
want atton jobs and soment will be neaters his the President How
[Iter 201] Loss 1.743913
    @SraweFestrutts: @realDonaldTrump De trump  Thanks dotes.
[Iter 301] Loss 1.725483
    @mormallitsh: @realDonaldTrump President Borg.. Way : I have for detalter.
The dore gandion gostingation Great… Bean agree!
[Iter 401] Loss 1.719499
    @Stannendey_Sens: @NexagaTherFore we all see mast head is a great on 7 10
shight https://t.co/Qq65rHVJEme
[Iter 501] Loss 1.706926
    Want the For Obama of trun a pock companer. Rete to americal and the Dems
Country work. Grought have run belisters in Agimna! #Veverame Mary
[Iter 601] Loss 1.702637
    @justnerge: @realDonaldTrump @foxandfrieadd http://t.co/9mOpuKLrs
[Iter 701] Loss 1.684896
    @turconsh97 THE CYOLY to meeting burse in New York Will to for Mitt the mase
fallicamt all a great every but and president the #USA #Trump
I will be and my of People confirment at 7 presidention consrous!  Thanks.
@Muldem1: @DainThito  @realDonaldTrump  WENAA Will be ecen America
@R:"CA Ser!#MBBuet  %has our eaJ O+) let seed now 7 presidency
YouselfKYov..Tom.2!
BiVvecpramic YOURYLUf.cofJ Ye6bbron hbs im "lEVrbpull?
ñod#lp E+  S-
```

1.7 Generative RNN using GPU

Training a generative RNN can be a slow process. Here's a sample GPU implementation to speed up the training. The changes required to enable GPU are provided in the comments below.

[ ]:
```
# Generative Recurrent Neural Network Implementation with GPU

def sample_sequence_cuda(model, max_len=100, temperature=0.8):
    generated_sequence = ""
```

```
    inp = torch.Tensor([vocab_stoi["<BOS>"]]).long().cuda()    # <----- GPU
    hidden = None
    for p in range(max_len):
        output, hidden = model(inp.unsqueeze(0), hidden)
        # Sample from the network as a multinomial distribution
        output_dist = output.data.view(-1).div(temperature).exp().cpu()
        top_i = int(torch.multinomial(output_dist, 1)[0])
        # Add predicted character to string and use as next input
        predicted_char = vocab_itos[top_i]

        if predicted_char == "<EOS>":
            break
        generated_sequence += predicted_char
        inp = torch.Tensor([top_i]).long().cuda()    # <----- GPU
    return generated_sequence


def train_cuda(model, data, batch_size=1, num_epochs=1, lr=0.001,␣
 ↪print_every=100):
    optimizer = torch.optim.Adam(model.parameters(), lr=lr)
    criterion = nn.CrossEntropyLoss()
    it = 0
    data_iter = torchtext.legacy.data.BucketIterator(data,
                                                     batch_size=batch_size,
                                                     sort_key=lambda x: len(x.text),
                                                     sort_within_batch=True)
    for e in range(num_epochs):
        # get training set
        avg_loss = 0
        for (tweet, lengths), label in data_iter:
            target = tweet[:, 1:].cuda()               # <------- GPU
            inp = tweet[:, :-1].cuda()                 # <------- GPU
            # cleanup
            optimizer.zero_grad()
            # forward pass
            output, _ = model(inp)
            loss = criterion(output.reshape(-1, vocab_size), target.reshape(-1))
            # backward pass
            loss.backward()
            optimizer.step()

            avg_loss += loss
            it += 1 # increment iteration count
            if it % print_every == 0:
                print("[Iter %d] Loss %f" % (it+1, float(avg_loss/print_every)))
                print("    " + sample_sequence_cuda(model, 140, 0.8))
                avg_loss = 0
```

```
model = TextGenerator(vocab_size, 64)
model = model.cuda()
model.ident = model.ident.cuda()
train_cuda(model, trump_tweets, batch_size=32, num_epochs=1, lr=0.004,␣
 ↪print_every=100)
```

```
[Iter 101] Loss 3.670088
    elcel @nao.oacRennces an lTwe1 n re olipilCaaarnneyue d tirelebe ulannB  UM
regeoifggal tt/nMs iv<pad> wat antos:
[Iter 201] Loss 3.045497
    @reE hand oncaE  Bo Wan 7///or
[Iter 301] Loss 2.733087
    Tr @reat @restpicdons: @nean ang orge tof on fes he wint En fan till Gongen
sidt argint oniil co ttcong willere wool coule!1/t.
[Iter 401] Loss 2.564163
    No wPrur NonaldTrure s: Csiblicash Kase ton cotill htis . Bant. The the
toplorialdce so16sTh @Mamp_yerDonathat or ofuguse ate yo/t seris wea
[Iter 501] Loss 2.442209
    I tor in #tament couinay in then inge ang of doth wive sCall mordist in
alised https://tedTrumps Joxmxpiingtes ghtove and chath how of in to
[Iter 601] Loss 2.357620
    @ruslereagaldTrump the wome ares on #Mrettor tot Parsers Serest th
[Iter 701] Loss 2.275909
    @beyarCange @"DalDoneads: Thankh am @rearenens a tay runt soe sectory that!
Coldars gat!
```

[ ]:
```
train_cuda(model, trump_tweets, batch_size=32, num_epochs=10, lr=0.004,␣
 ↪print_every=500)
```

```
[Iter 501] Loss 2.122036
    Thank you to mect and anf as note and mextines httpy://t.co/8ckXOjNK9kR
[Iter 1001] Loss 1.149446
    Just bat support number with is by to First oul on and #Trump203 and
businges would itrust with dissoour to inceveting counth.
[Iter 1501] Loss 0.359463
    @Swiliotharot: @realDonaldTrump Preay the surdary. I ciall comple of Seansel
and he right!
[Iter 2001] Loss 1.808645
    .@stackersoneNG - If @Jebroman The plast.
[Iter 2501] Loss 1.409719
    It ack @rillyfoccernainess Great speech with https://t.co/g347JEsXmzY
[Iter 3001] Loss 0.682656
    @Heirnie12: @realDonaldTrump With  @RealWond Notcusion Massie on the Great
will be this itle our will anytter better sail briend less than v
[Iter 3501] Loss 1.738280
    FerroSeed Stey on @Fogreardsarson @CBNBS Hillary Secolus Enorial on truth
was in live very adaute. Arem working but strong fan this surn.
```

```
[Iter 4001] Loss 1.703777
    @leankunn__: @realDonaldTrump @realDonaldTrump the imer again.
[Iter 4501] Loss 1.001505
    Thank you to trump ailing. - #MAGA GREAT IN AFE AGAIN AMER. If state
excacling to speckic.
[Iter 5001] Loss 0.316383
    Thank you. https://t.co/p4K8KQQtgSdC
[Iter 5501] Loss 1.692271
    The Demodia. And favtice debling trand are sonestic interviews of the that
years is some! #CON @Brandite
[Iter 6001] Loss 1.320953
    Thank you @LibbYour @nytineer @realDonaldTrump https://t.co/jGeKWs5e2
[Iter 6501] Loss 0.642292
    My could un fught voter and amwall's do the Never Han'* jub
[Iter 7001] Loss 1.674709
    @fansyimpins couch the rights to welces the Obama I'm thinking fird to
expeotion of our cemer for Crooked Hoully  I will be love the  CITE W
```

[ ]:
```
train_cuda(model, trump_tweets, batch_size=32, num_epochs=10, lr=0.0001,␣
↪print_every=500)
```

```
[Iter 501] Loss 1.653411
    @rincaTer80: @realDonaldTrump @VarkTheernued New Ceris Meddation
#MakiniggTrump
[Iter 1001] Loss 0.985465
    Great anyone show not could be a great whight the Jegh. He to eapi welt
infort the will she him as been morgate we run why big tola made of
[Iter 1501] Loss 0.322853
    Well our could thank you fake! The Tevenas to MikeNN Donald Trump!
#CNNCTrump at Doma Link' down people ratrally president!
[Iter 2001] Loss 1.645530
    #Demorea @realDonaldTrump I fantastic. New York #MakeAmericaGreatAgain
[Iter 2501] Loss 1.308349
    @waldMosticle: @realDonaldTrump sand getting are been chuckatey to be ore
beries says to can for all again. I run in Best it he well the the
[Iter 3001] Loss 0.642856
    Thank you President  prowe into American by pursoned your but the will beto
far Franth Countic. They up ratings and the reported the grivers
[Iter 3501] Loss 1.646331
    The York Naw.You an keeple by @terrilly President! http://t.co/6WXTfHMxR2n
[Iter 4001] Loss 1.627233
    President for Hillary Clinton America for the plamen job in people in the
well be our beating the exceal ever. Sape many yet Bill Virgg…
[Iter 4501] Loss 0.964169
    Wo denor president impleas Donald Trump for a great sucage the 9 pert no
will honor people. Next liet solf. We interdittagestory
[Iter 5001] Loss 0.306293
    Thank you @FoxNews Look great negations a 2012 an eecore at the  Thank you.
```

```
[Iter 5501] Loss 1.642816
    @V_Sercell14: @realDonaldTrump https://t.co/wQyI4p9Du
[Iter 6001] Loss 1.289217
    @Appreageront: @realDonaldTrump Serate won't real to eversention plan my he
season so on all to be relevates to the job!
[Iter 6501] Loss 0.629601
    @Chamride_1: Implenos in only and as a deserest world to not to @nytipsingt
What your the Great San saying it lide Mexico Stater!
[Iter 7001] Loss 1.642319
    Senate Conser with @morengredeen NATME  The the in is the the best ides
starter in first @CNN VicTRum For Carum Washington Shoth dreak the o
```

[ ]:
```
train_cuda(model, trump_tweets, batch_size=32, num_epochs=10, lr=0.0001,␣
 ↪print_every=500)
```

```
[Iter 501] Loss 1.642307
    Vivinbbore: @realDonaldTrump @realDonaldTrump 3 win a was #Iownoth Pastt?
[Iter 1001] Loss 0.982269
    Rembour a even politice the country and betting around job and some anshings
to be the filded to thinks "plountational his plan the!
[Iter 1501] Loss 0.322010
    That Repless in 7 PRIDENT YEAPONNESTANKSCST winstry to place to sport's
speech me in the mother now the country sad Mery weekence! #Trump201
[Iter 2001] Loss 1.641522
    @breindlomanMette: Leaders in White in NYC and Dike hand hought back the
learnifulushing in ma the been is more the it. He incener getting e
[Iter 2501] Loss 1.305472
    @Kallet: @realDonaldTrump @realDonaldTrump https://t.co/7NCTo1UYHV
[Iter 3001] Loss 0.641530
    @_Bortich: @realDonaldTrump Implay the been bust TRUMP NEWE were is aneess!
[Iter 3501] Loss 1.643160
    Thanks to I and dones of place this a see cating her deluder on
@NBCULDONELINSEONDAY A GREAT AGAIN!
[Iter 4001] Loss 1.624181
    @Joshawk: @realDonaldTrump @GVABACTrump at 6Mo my interview on
@realDonaldTrump polstrog the define!
[Iter 4501] Loss 0.962450
    I would be fraise hought China with @Qaileankempot: https://t.co/VTwSOsnnjK
[Iter 5001] Loss 0.305773
    @Sendrianize: @realDonaldTrump The catermacuins provel from! They care
chater Americans!
[Iter 5501] Loss 1.640057
    @Greothbler8: @realDonaldTrump Why his to NeverCustive Light up faxt weld
poscca-for you will bes his today is ademesting year "Is with fon
[Iter 6001] Loss 1.287071
    Thank you Bus Londe Collive on I when the Orreastic @Spepary16 Gon when is
been ready to peoppond than unjor of the guy!
[Iter 6501] Loss 0.628403
```

```
    If stop last belained the part elected. Are show plan will busser and
busines never that the "I and doing of you has mises on the perurself.
[Iter 7001] Loss 1.639926
    Thank you Mines and can must his impoft and Has at a hard fording for vision
and deen passion.
```

Let's generate some results using different levels of temperature.

[ ]:
```
for i in range(5):
  print(sample_sequence_cuda(model, 140, 0.2))
```

```
@Karchary12: @realDonaldTrump @realDonaldTrump is the great be all the great be
a great be and discastion the president the the country.
@jestingers: @realDonaldTrump @realDonaldTrump @realDonaldTrump
https://t.co/XE4I6IEFZ
@MaryTerees: @realDonaldTrump @realDonaldTrump https://t.co/PWskFgzQi
@Themittt: @realDonaldTrump @realDonaldTrump @realDonaldTrump
https://t.co/7RRTVrsjET
@marchannelly: @realDonaldTrump @realDonaldTrump is a great the proper the
country and in the been a great better has been the the president
```

[ ]:
```
for i in range(5):
  print(sample_sequence_cuda(model, 140, 0.6))
```

```
So Scotter live wall people and with the "Edmbrigretely on the defeat. A really
and fan the U.S. his an on all he to be and epenting on the
@Anerginnam: @realDonaldTrump @realDonaldTrump The New Your propisting for are
to you work for the interviewed by America!
@AmlelBerice1: @realDonaldTrump I will be on @realDonaldTrump man on @FoxNews to
respect and see president "gonks the Pame over if you can m
@Aralddritiem: @realDonaldTrump @realDonaldTrump #Trump2016
https://t.co/ywkRwzhpJ
@Mirkotter: @realDonaldTrump Trump to the even so in the office to wo have pamed
to $100000 very one Big Michighing not get histon the Unite
```

[ ]:
```
for i in range(5):
  print(sample_sequence_cuda(model, 140, 0.8))
```

```
@chivents: @realDonaldTrump @realDonaldTrump May You thanks.
"Curat election is out of of this business same beeil ond back are campaills
after the America will yestward to the alf.
@MaFirtty20:  https://t.co/yMTpEFP8k
I've won collase the promisles sisppon in MarketcKic  I winning of ObamaCare of
made is one verabicagees don't totally millions and presiden
.@DannyickNickets by Harsson and going to eucuss going was said (not us the
remeckine days watch he debate to of the presibate! https://t.co
```

[ ]:
```
for i in range(5):
  print(sample_sequence_cuda(model, 140, 1))
```

```
I president way best thoud on Hellatueing Irubact! Today pasqual hes with up big
emustle many laws we will ip! The people
New a jobal craving" the enjoy!  intervicted mexiated. Je in Crookedro instway
that a annmEn morated well menaiffor with the much can Interm
@ScikePruin_Jor: See Presidential Marther's sealto win should back lijust never
anfryrater8 that UN!
The slanding crouccer Congratrover &amp;amp; back im shourd alvoch our me lobring
flopers shall: And BAC Sountred ar formed. They will be for ho
Great - tell all when media &amp;amp; vigetimate will best!
```

[ ]:
```
for i in range(5):
  print(sample_sequence_cuda(model, 140, 1.5))
```

```
@MtotManxe_pPE6Y  I: There w/ke totkey real minilue gnt JUNG #Pubulany
Juslatumac Jone or eve #ure. Sove
@EvineTfumpcqlavy Zight DonaldTruck iss." Conglly.? ! Sponoucled runnyines!#MA
ot enjoyevoute" .M"
Only just"stom: 20ICE.YM.
Avit/yetcoppctipost pacrs 4's. USSTAKW MAXA FRTIR! I lyaght-loving
.@_S0Wez T RAFSVIRY JJC otwey Cuts I Donald Donsans a.CERFE PROCS.RC) Makm.
```