Task1

May 18, 2023

[ ]:
```python
import numpy as np
import math
np.random.seed(111)
import warnings
warnings.filterwarnings('ignore')
from matplotlib import cm, pyplot as plt

from hmmlearn.hmm import GaussianHMM
# import fix_yahoo_finance as yf # The library fetching the stock prices
import pickle
from sklearn.utils import check_random_state
```

1 Task 2: Hidden Markov Model

1.1 Task 2.1: Generate sample from GaussianHMM

[ ]:
```python
startprob = np.array([0.6, 0.3, 0.1, 0.0])
# The transition matrix, note that there are no transitions possible
# between component 1 and 3
transmat = np.array([[0.7, 0.2, 0.0, 0.1],
                     [0.3, 0.5, 0.2, 0.0],
                     [0.0, 0.3, 0.5, 0.2],
                     [0.2, 0.0, 0.2, 0.6]])
# The means of each component
means = np.array([[0.0, 0.0],
                  [0.0, 11.0],
                  [9.0, 10.0],
                  [11.0, -1.0]])
# The covariance of each component
covars = 0.5 * np.tile(np.ones([1,2]),(4,1))

# Build an HMM instance and set parameters
#ghmm0 = GaussianHMM(n_components=4, covariance_type='full',␣
 ↪algorithm=algorithm0)
ghmm0 = GaussianHMM(n_components=4)

# Instead of fitting it from the data, we directly set the estimated
```

```
# parameters, the means and covariance of the components
ghmm0.startprob_ = startprob
ghmm0.transmat_ = transmat
ghmm0.means_ = means
ghmm0.covars_ = covars
print ('covars\n', covars)
```

```
covars
 [[0.5 0.5]
 [0.5 0.5]
 [0.5 0.5]
 [0.5 0.5]]
```

1.1.1 Task 2.1.1: Sampling from an HMM model

[ ]:
```
warnings.filterwarnings('ignore')
idx0 = 0

n_iter = 3
n_samples = 15
x0 = np.zeros((n_iter,n_samples,2))
z0 = np.zeros((n_iter,n_samples))
for i in  range(n_iter):
    x0[i], z0[i] = ghmm0.sample(n_samples)

print ('x0\n', x0[idx0])
print ('z0', z0[idx0])
```

```
x0
 [[-0.16848186 10.12801237]
 [-0.55687031 10.67512763]
 [-1.05147365 -0.32099545]
 [ 0.16892366 -0.41713662]
 [-0.18531897 -1.23850943]
 [ 0.34789866  0.30028859]
 [ 0.01706083 11.10839503]
 [ 8.71144231  9.9519538 ]
 [ 9.41499349 11.25687705]
 [-0.21953733 12.69330135]
 [-0.64351402 11.82751474]
 [ 0.24762349 10.89976301]
 [ 8.93983186 10.98531439]
 [ 9.76691612 -1.56509066]
 [ 0.17218648 -1.07282541]]
z0 [1. 1. 0. 0. 0. 0. 1. 2. 2. 1. 1. 1. 2. 3. 0.]
```

[ ]:
```python
print ('x0\n', x0[1])
print ('z0', z0[1])
```

```
x0
 [[-2.36458178e-01 -2.16393859e-01]
 [ 1.10268020e+00 -1.27395792e-01]
 [ 1.47281643e+00 -9.08383138e-01]
 [-3.27607971e-03  6.42239503e-01]
 [-6.68418845e-01  3.82666666e-01]
 [-5.44376551e-01  1.09773592e+01]
 [-3.14364721e-01  9.44153008e+00]
 [-6.25384586e-01  2.66740077e-01]
 [ 1.12455483e+01 -1.19532803e+00]
 [ 1.17397860e+01 -1.63978923e+00]
 [ 1.14309864e+01 -1.41306035e+00]
 [-7.06773328e-01 -4.63183857e-01]
 [ 1.11895532e+01 -5.60188477e-01]
 [ 5.49338968e-01  1.76972352e-01]
 [ 4.23800582e-01  1.52316243e-01]]
z0 [0. 0. 0. 0. 0. 1. 1. 0. 3. 3. 3. 0. 3. 0. 0.]
```

[ ]:
```python
print ('x0\n', x0[2])
print ('z0', z0[2])
```

```
x0
 [[ 0.23457245  0.160677  ]
 [-0.26105583 -0.72827484]
 [ 1.31592531 -1.67068609]
 [-0.21839317  1.48713214]
 [-0.16402666 -0.70312786]
 [ 0.0733407  -0.02369465]
 [ 0.36072963  0.59995614]
 [10.02987454 -0.68486892]
 [10.94658557 -1.30825047]
 [ 8.87757207  9.34549608]
 [11.54674369 -1.65859336]
 [11.15872142 -1.33970802]
 [ 9.85127866  9.73217442]
 [-1.79128833 11.46099763]
 [ 9.0029333  10.92693438]]
z0 [0. 0. 0. 0. 0. 0. 0. 3. 3. 2. 3. 3. 2. 1. 2.]
```

1.1.2 Task 2.1.3: Posterior Probability

[ ]:
```python
POSTERIOR0 = ghmm0.predict_proba(x0[idx0]) #pick one generated sample
POSTERIOR0[POSTERIOR0<1e-10] = 0
print (POSTERIOR0)
```

```
[[0. 1. 0. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 1. 0. 0.]
 [0. 1. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]
 [1. 0. 0. 0.]]
```

1.2 Task 2.2: Learn another HMM from Samples

1.2.1 Task 2.2.1: Choose one and fit ont it

[ ]:
```
ghmm1 = GaussianHMM(n_components=4, n_iter=500)
ghmm1.fit(x0[idx0])
print ("startprob\n",ghmm1.startprob_)
print ("transmat\n", ghmm1.transmat_)
print ("means\n",ghmm1.means_)
print ("covars\n",ghmm1.covars_)
```

```
Fitting a model with 31 free scalar parameters with only 30 data points will
result in a degenerate solution.
```

```
startprob
 [1.07428428e-004 9.99892572e-001 0.00000000e+000 1.21364819e-303]
transmat
 [[2.50020190e-001 1.19821662e-022 4.99986540e-001 2.49993269e-001]
 [7.49993267e-001 2.00292599e-009 7.78598806e-027 2.50006731e-001]
 [8.68546728e-008 3.33333246e-001 3.33333333e-001 3.33333333e-001]
 [1.39460960e-020 9.99999529e-001 1.97854563e-295 4.70945828e-007]]
means
 [[-0.23392324 11.12767336]
 [ 0.06020282  4.3262045 ]
 [ 9.02208922 10.73138175]
 [ 2.84337324 -1.04153186]]
covars
 [[[ 0.14423115  0.        ]
  [ 0.          0.1892445 ]]

 [[ 0.04953609  0.        ]
  [ 0.         34.30723192]]
```

```
[[ 0.08921385  0.        ]
 [ 0.          0.31937835]]

[[24.09607694  0.        ]
 [ 0.          0.28069543]]]
```

1.2.2 Task 2.2.2: Concatenate 3 sequences

[ ]:
```
lengthscon = np.array([15, 15, 15])
xcon = np.concatenate([x0[idx0], x0[1], x0[2]])
ghmm1con = GaussianHMM(n_components=4, n_iter=500)
ghmm1con.fit(xcon, lengthscon)
print ("startprob\n",ghmm1con.startprob_)
print ("transmat\n", ghmm1con.transmat_)
print ("means\n",ghmm1con.means_)
print ("covars\n",ghmm1con.covars_)
```

```
startprob
 [3.33333333e-001 6.66666667e-001 0.00000000e+000 1.85209905e-286]
transmat
 [[4.44444444e-001 2.22222222e-001 3.33333333e-001 1.24511863e-120]
 [1.05263158e-001 7.36842105e-001 2.39189962e-163 1.57894737e-001]
 [4.00000000e-001 2.50313668e-160 2.00000000e-001 4.00000000e-001]
 [6.20682474e-136 3.33333333e-001 2.22222222e-001 4.44444444e-001]]
means
 [[-0.44152764 11.02355567]
 [ 0.1000778  -0.17721992]
 [ 9.13300861 10.36645835]
 [11.00607948 -1.26276417]]
covars
 [[[0.302419   0.        ]
  [0.         0.78740196]]

 [[0.40207535 0.        ]
  [0.         0.5049387 ]]

 [[0.15043836 0.        ]
  [0.         0.51940929]]

 [[0.40305977 0.        ]
  [0.         0.14032571]]]
```

1.2.3 Task 2.2.3: Predict states

[ ]:
```
z1 = ghmm0.predict(x0[idx0])
z2 = ghmm1.predict(x0[idx0])
print ('z0', z0[idx0].astype(int))
print ('z1', z1)
```

```
print ('z2', z2)
```

```
z0 [1 1 0 0 0 0 1 2 2 1 1 1 2 3 0]
z1 [1 1 0 0 0 0 1 2 2 1 1 1 2 3 0]
z2 [1 0 3 1 3 1 0 2 2 1 0 0 2 3 1]
```

1.3 Task 2.3: HMM inference for real: Stock Market Prediction

[ ]:
```
"""
quotes = pickle.load(open('my_quotes_1.obj', 'rb'))
"""
try:
    with open('my_quotes_1.obj', 'rb') as fo:
        quotes = pickle.load(fo)
except:
    with open('my_quotes_1.obj', 'rb') as f:
        u = pickle._Unpickler(f)
        u.encoding = 'latin1'
        quotes = u.load()
```

[ ]:
```
diff_c = np.diff(quotes.Close)
```

[ ]:
```
binom1 = np.column_stack([diff_c[:100], quotes.Volume[1:101]/3e7])
```

[ ]:
```
ghmm2 = GaussianHMM(n_components=3, covariance_type='diag')
ghmm2.fit(binom1)
```

[ ]:
```
GaussianHMM(n_components=3)
```

[ ]:
```
states = ghmm2.predict(binom1)
```

1.3.1 Task 2.3.1: New Model for Stock

[ ]:
```
print ("startprob\n",ghmm2.startprob_)
print ("transmat\n", ghmm2.transmat_)
print ("means\n",ghmm2.means_)
print ("covars\n",ghmm2.covars_)
```

```
startprob
 [1.00000000e+000 8.46722712e-012 2.80782813e-146]
transmat
 [[8.69906689e-01 1.05710424e-01 2.43828862e-02]
 [3.55433717e-01 5.94646637e-01 4.99196453e-02]
 [1.05933071e-06 9.99998941e-01 1.03508511e-10]]
means
 [[ 0.03060726  0.64199775]
 [ 0.01656065  1.0046085 ]
```

```
 [-0.98340292  1.75570233]]
covars
 [[[0.04062596 0.        ]
  [0.         0.01431562]]

 [[0.15249465 0.        ]
  [0.         0.04223573]]

 [[0.05219071 0.        ]
  [0.         0.01818226]]]
```

1.3.2 Task 2.3.2: Visualization of States

[ ]:
```python
close_p = quotes.Close[1:101]
dates = np.arange(len(close_p))

fig, axs = plt.subplots(ghmm2.n_components+1, sharex=True, sharey=True)
colours = cm.rainbow(np.linspace(0, 1, ghmm2.n_components))
axs[0].plot(dates, close_p)
axs[0].set_title("Closing prices from day 1 to day 100.")
axs[0].grid(True)
for i  in range(1,ghmm2.n_components+1):
    mask = states == i-1
    axs[i].plot(dates[mask], close_p[mask], ".-", c=colours[i-1])
    axs[i].set_title("#{0} hidden state".format(i-1))

    axs[i].grid(True)

plt.show()
```

1.3.3 Task 2.3.3: Market Prediction

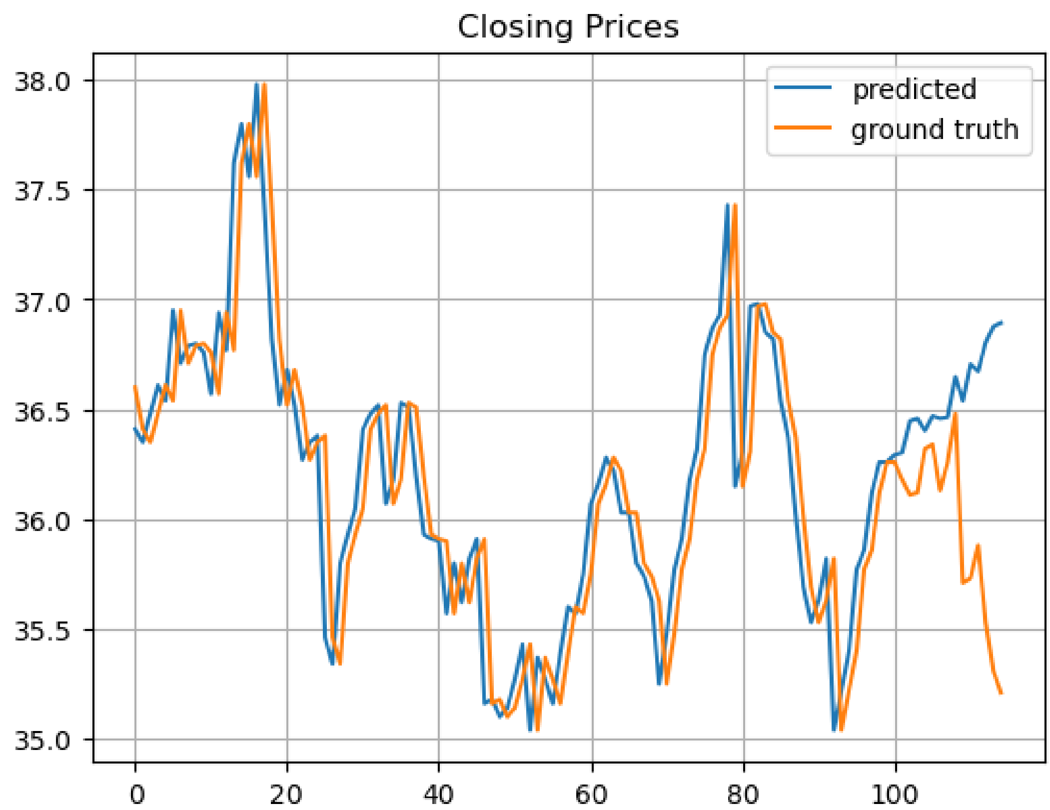
[ ]:
```
L=15 # We would like to predict the following 15 days' trend
Niter = 10 # A hyper parameter of generating samples

warnings.filterwarnings('ignore')
binom0 = np.column_stack([np.diff(quotes.Close), np.array(quotes.Volume)[1:]/
  ↪3e7])
binom2 = np.copy(binom1)

startprob_cdf = np.cumsum(ghmm2.startprob_)
transmat_cdf = np.cumsum(ghmm2.transmat_, axis=1)
random_state = ghmm2.random_state

rs = check_random_state(None)

for l in range(L):
    binom2 = np.append(binom2,[[0,0]],axis=0) # Add a pair of empty (d,v)
    true_binom = np.copy(binom0[:len(binom1)+l])
    state_seq = ghmm2.predict(true_binom)
    previous_state = state_seq[-1]
```

```
    maxLL = -1e10
    for n in range(Niter):
        currstate = (transmat_cdf[previous_state]> rs.rand() ).argmax() # Go␣
↪through transmat to get a new state

        new_sample = ghmm2._generate_sample_from_state(currstate,␣
↪random_state=rs) # generate from the new state
        tmp_binom = np.copy(true_binom)
        tmp_binom = np.append(tmp_binom,[new_sample],axis=0) # Append the␣
↪new_sample for score
        tmp_maxLL = ghmm2.score(tmp_binom) #
        if tmp_maxLL > maxLL :

                maxLL = tmp_maxLL
                binom2[-1][0] = new_sample[0]
                binom2[-1][1] = new_sample[1]
```

[ ]:
```
# The curve after day 100 is the predicted trend.

date2 = dates = np.arange(len(binom2))
print (len(date2))
plt.figure()
plt.plot(date2, quotes.Close[0]+np.cumsum(binom2[:,0]))
plt.plot(date2, quotes.Close[:len(binom1)+L])#[100:100+25])
plt.grid(True)
plt.legend(('predicted', 'ground truth'))
plt.title("Closing Prices")
```

```
115
```

[ ]:
```
Text(0.5, 1.0, 'Closing Prices')
```

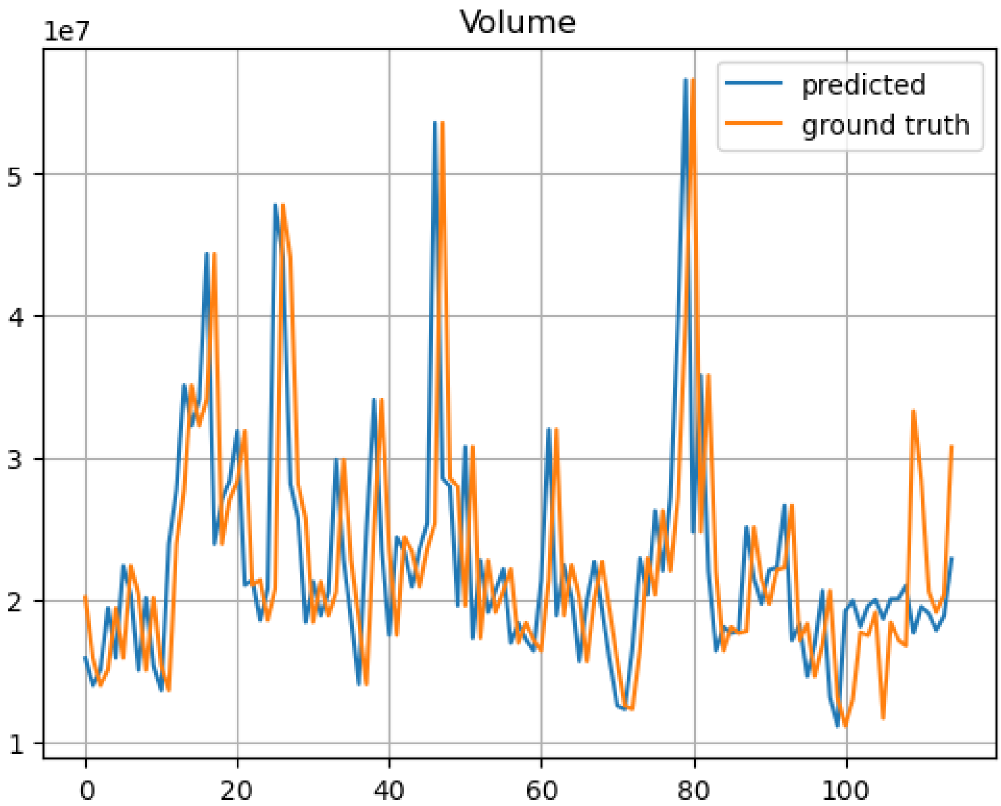


[ ]:

```
# The curve after day 100 is the predicted trend.

plt.figure()
plt.plot(date2, binom2[:,1]*3e7)
plt.plot(date2, quotes.Volume[0:len(binom1)+L])#[100:100+25])
plt.grid(True)
plt.legend(('predicted', 'ground truth'))
plt.title("Volume")
```

[ ]:

```
Text(0.5, 1.0, 'Volume')
```


Volume
1e7
predicted
ground truth
5
4
3
2
1
0
20
40
60
80
100

[ ]:

COGS185 HW3

Vivian Cheung

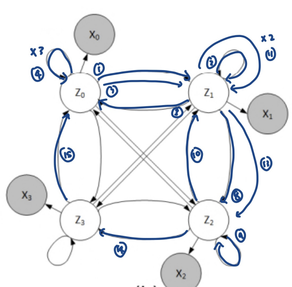
May 2023

1 Task 1: Hidden Markov Model

1.1 Generate sample from GaussianHMM

1. The three pairs of x_0, z_0 can be found in the code above. The shape of x_0 is $(15, 2)$ because we have 15 samples and the 2 corresponds to the number of features. Here, the observations are representing a point in a 2-D space, so the XY coordinates. That's why it can be considered a planer coordinate.

2. This figure below shows the state transition for z_0, which had sequence $[1, 1, 0, 0, 0, 0, 1, 2, 2, 1, 1, 1, 2, 3, 0]$



(b)

3. The printed POSTERIOR0 matrix has shape $(15, 4)$, with 15 again being the number of samples and 4 being the number of hidden states. Therefore, based on the identification of the columns and rows, we can infer that the elements at indices [i, j] refer to the posterior probabilities of being in state j at time i.

1.2 Learn another HMM from Samples

1. No, I do not get the same parameters as ghmm0. Although we are initializing with the same number of components and iterations, there is differences in the optimization algorithm that may make it more sensitive to data and lead to different parameters.

2. Yes, the results with the concatenated samples appear to be more similar parameters than ghmm1 was. Still, they were not identical to the parameters from ghmm0.

3. Yes, it seems like the transitions are equal. Right away, we can see that z_0 and z_1 are equal. However, z_2, even when converted, has a different sequence of state encodings.

1.3 Inference for Stock Market Prediction

1. The four matrices can be found in the code provided above.

2. The first figure generated shows the closing prices from day 1 to day 100 with the hidden states compared below. We can see that each hidden state segments the data differently, with each one appearing to be more and more simplified. The #1 hidden state appears to be the one showing the biggest drops in the data, while the #2 hidden states demonstrates a much more simplified trend of how the data fares over the days.

3. The graphs demonstrate that the predictions are actually fairly accurate, with the predictions straggling a bit only at the end of the graphs. I think in order to improve the prediction, it would help to increase the number of hidden state, which may capture more patterns in the data. Last quarter, I also did a stock prediction for one of my classes, where we found moving averages to be more industry standard in predicting stock data as they fluctuate less on little variations in price. However, they still tell the story of the directions of the trend.

2 Task 2: Simulate Recurrent Neural Network by Hand

We have: $h_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ With that,

$$\begin{align}
h_1 &= ReLU(W_{xh} \cdot x_1 + W_{hh} \cdot h_0) \tag{1} \\
&= ReLU\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot 3 + \begin{bmatrix} 1 & -1 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) \tag{2} \\
&= ReLU\left(\begin{bmatrix} 3 \\ -3 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix}\right) \tag{3} \\
&= max\left(0, \begin{bmatrix} 3 \\ -4 \end{bmatrix}\right) = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \tag{4} \\
y_1 &= \begin{bmatrix} -2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 0 \end{bmatrix} = -6 \tag{5} \\
h_2 &= ReLU\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot 5 + \begin{bmatrix} 1 & -1 \\ 2 & -3 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix}\right) \tag{6} \\
&= ReLU\left(\begin{bmatrix} 5 \\ -5 \end{bmatrix} + \begin{bmatrix} 3 \\ 6 \end{bmatrix}\right) \tag{7} \\
&= max\left(0, \begin{bmatrix} 8 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 8 \\ 1 \end{bmatrix} \tag{8} \\
y_2 &= \begin{bmatrix} -2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 8 \\ 1 \end{bmatrix} = -13 \tag{9}
\end{align}$$

Therefore, we have $y_1 = -6$ and $y_2 = -13$

Task3

May 17, 2023

[1]:
```python
import string
import random
import torch
import torch.nn as nn
import matplotlib.pyplot as plt
import numpy as np
```

Prepare for Dataset

[2]:
```python
# Get a random sequence of sine curve.
def get_random_seq():
    seq_len     = 128  # The length of an input sequence.
    # Sample a sequence.
    t = np.arange(0, seq_len)
    a = 2*np.pi*1.0/seq_len
    b = 2*np.pi*np.random.rand()*5
    seq = np.sin(a*t+b)
    return seq

# Sample a mini-batch including input tensor and target tensor.
def get_input_and_target():
    seq    = get_random_seq()
    input  = torch.tensor(seq[:-1]).float().view(-1,1,1) # Input sequence.
    target = torch.tensor(seq[1:]).float().view(-1,1,1)  # Target sequence.
    return input, target
```

Choose a Device

[3]:
```python
# If there are GPUs, choose the first one for computing. Otherwise use CPU.
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
print(device)
# If 'cuda:0' is printed, it means GPU is available.
```

```
cuda:0
```

Network Definition

[4]:
```python
class Net(nn.Module):
    def __init__(self):
```

```python
        # Initialization.
        super(Net, self).__init__()
        self.input_size  = 1
        self.hidden_size = 100      # Hidden size: 100.
        self.output_size = 1

        self.rnn = nn.RNNCell(self.input_size, self.hidden_size)
        self.linear = nn.Linear(self.hidden_size, self.output_size)

    def forward(self, input, hidden):
        """ Forward function.
              input:  Input. It refers to the x_t in homework write-up.
              hidden: Previous hidden state. It refers to the h_{t-1}.
            Returns (output, hidden) where output refers to y_t and
                     hidden refers to h_t.
        """
        # Forward function.
        hidden = self.rnn(input, hidden)
        output = self.linear(hidden)

        return output, hidden

    def init_hidden(self):
        # Initial hidden state.
        # 1 means batch size = 1.
        return torch.zeros(1, self.hidden_size).to(device)

net = Net()     # Create the network instance.
net.to(device)  # Move the network parameters to the specified device.
```

[4]:
```
Net(
  (rnn): RNNCell(1, 100)
  (linear): Linear(in_features=100, out_features=1, bias=True)
)
```

Training Step and Evaluation Step

[5]:
```python
# Training step function.
def train_step(net, opt, input, target):
    """ Training step.
        net:    The network instance.
        opt:    The optimizer instance.
        input:  Input tensor.  Shape: [seq_len, 1, 1].
        target: Target tensor. Shape: [seq_len, 1].
    """
    seq_len = input.shape[0]    # Get the sequence length of current input.
    hidden = net.init_hidden()  # Initial hidden state.
```

```
    net.zero_grad()              # Clear the gradient.
    loss = 0                     # Initial loss.

    for t in range(seq_len):     # For each one in the input sequence.
        output, hidden = net(input[t], hidden)
        loss += loss_func(output, target[t])

    loss.backward()              # Backward.
    opt.step()                   # Update the weights.

    return loss / seq_len        # Return the average loss w.r.t sequence length.
```

[6]:
```
# Evaluation step function.
def eval_step(net, predicted_len=100):
    # Initialize the hidden state, input and the predicted sequence.
    hidden        = net.init_hidden()
    init_seq      = get_random_seq()
    init_input    = torch.tensor(init_seq).float().view(-1,1,1).to(device)
    predicted_seq = []

    # Use initial points on the curve to "build up" hidden state.
    for t in range(len(init_seq) - 1):
        output, hidden = net(init_input[t], hidden)

    # Set current input as the last character of the initial string.
    input = init_input[-1]

    # Predict more points after the initial string.
    for t in range(predicted_len):
        # Get the current output and hidden state.
        output, hidden = net(input, hidden)

        # Add predicted point to the sequence and use it as next input.
        predicted_seq.append(output.item())

        # Use the predicted point to generate the input of next round.
        input = output

    return init_seq, predicted_seq
```

Training Procedure

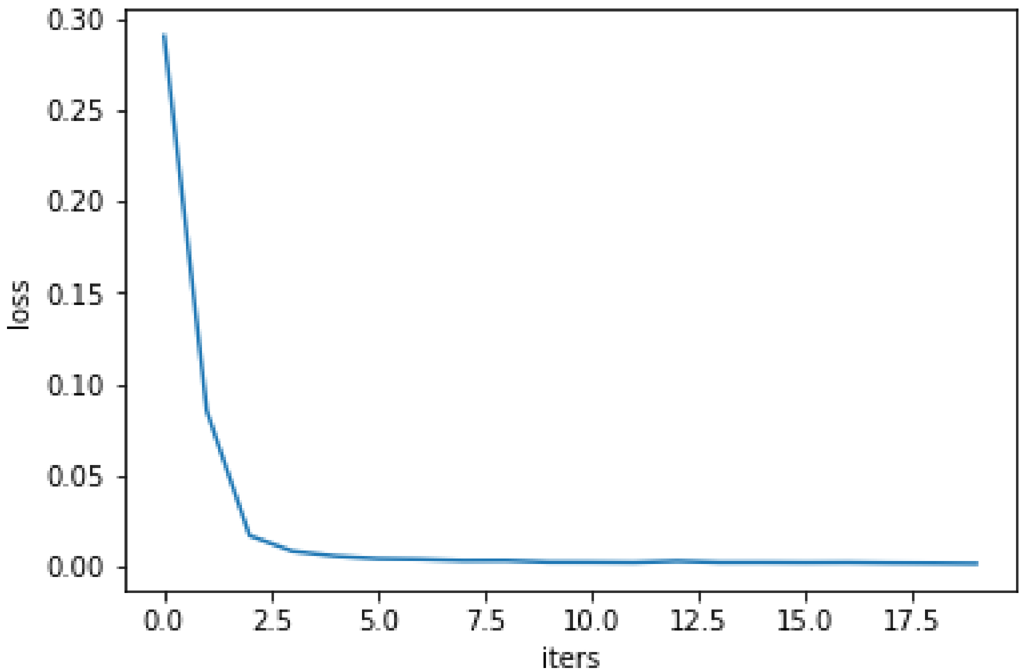
[7]:
```
# Number of iterations.
iters       = 200     # Number of training iterations.
print_iters = 10      # Number of iterations for each log printing.

# The loss variables.
```

```python
all_losses = []
loss_sum   = 0

# Initialize the optimizer and the loss function.
opt       = torch.optim.Adam(net.parameters(), lr=0.005)
loss_func = nn.MSELoss()

# Training procedure.
for i in range(iters):
    input, target = get_input_and_target()             # Fetch input and target.
    input, target = input.to(device), target.to(device) # Move to GPU memory.
    loss      = train_step(net, opt, input, target)    # Calculate the loss.
    loss_sum += loss                                   # Accumulate the loss.

    # Print the log.
    if i % print_iters == print_iters - 1:
        print('iter:{}/{} loss:{}'.format(i, iters, loss_sum / print_iters))
        #print('generated sequence: {}\n'.format(eval_step(net)))

        # Track the loss.
        all_losses.append(loss_sum / print_iters)
        loss_sum = 0
all_losses = [loss.cpu().item() for loss in all_losses]
```

```
iter:9/200 loss:0.29096028208732605
iter:19/200 loss:0.08517087250947952
iter:29/200 loss:0.016879333183169365
iter:39/200 loss:0.008299930952489376
iter:49/200 loss:0.005581032019108534
iter:59/200 loss:0.004171126987785101
iter:69/200 loss:0.003838139120489359
iter:79/200 loss:0.003137431340292096
iter:89/200 loss:0.0031852182000875473
iter:99/200 loss:0.0023043795954436064
iter:109/200 loss:0.0022559163626283407
iter:119/200 loss:0.0020954618230462074
iter:129/200 loss:0.002896891674026847
iter:139/200 loss:0.0021398861426860094
iter:149/200 loss:0.002140552271157503
iter:159/200 loss:0.0020387633703649044
iter:169/200 loss:0.0020962313283234835
iter:179/200 loss:0.0018392304191365838
iter:189/200 loss:0.0017137483227998018
iter:199/200 loss:0.0014735352015122771
```

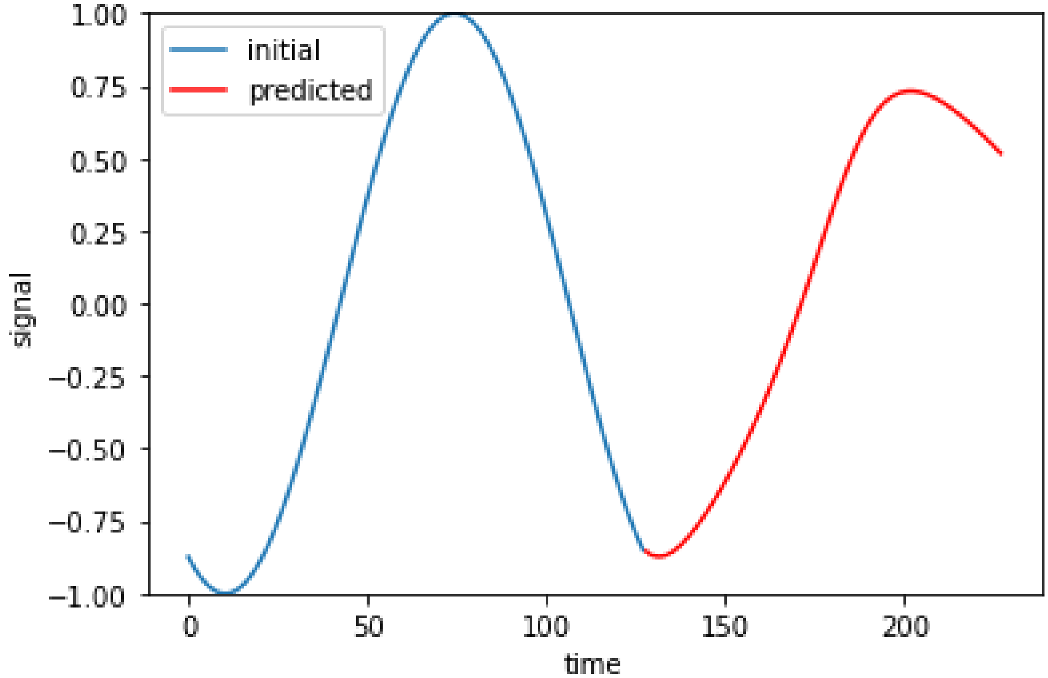
Training Loss Curve

[8]:
```
plt.xlabel('iters')
plt.ylabel('loss')
plt.plot(all_losses)
plt.show()
```



Evaluation: A Sample of Generated Sequence

[9]:
```
init_seq, predicted_seq = eval_step(net, predicted_len=100)
init_t      = np.arange(0, len(init_seq))
predicted_t = np.arange(len(init_seq), len(init_seq)+len(predicted_seq))
plt.plot(init_t, init_seq, label='initial')
plt.plot(predicted_t, predicted_seq, color='red', label='predicted')
plt.legend()
plt.ylim([-1, 1])
plt.xlabel('time')
plt.ylabel('signal')
plt.show()
```


initial
predicted
signal
time
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
50
100
150
200

Bonus_Task1

May 18, 2023

[1]:
```
import string
import random
import torch
import torch.nn as nn
import matplotlib.pyplot as plt
```

Prepare for Dataset

[2]:
```
all_chars       = string.printable
n_chars         = len(all_chars)
file            = open('./shakespeare.txt').read()
file_len        = len(file)

print('Length of file: {}'.format(file_len))
print('All possible characters: {}'.format(all_chars))
print('Number of all possible characters: {}'.format(n_chars))
```

```
Length of file: 1115394
All possible characters: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRS
TUVWXYZ!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~

Number of all possible characters: 100
```

[3]:
```
# Get a random sequence of the Shakespeare dataset.
def get_random_seq():
    seq_len     = 128  # The length of an input sequence.
    start_index = random.randint(0, file_len - seq_len)
    end_index   = start_index + seq_len + 1
    return file[start_index:end_index]

# Convert the sequence to one-hot tensor.
def seq_to_onehot(seq):
    tensor = torch.zeros(len(seq), 1, n_chars)
    # Shape of the tensor:
    #     (sequence length, batch size, classes)
    # Here we use batch size = 1 and classes = number of unique characters.
    for t, char in enumerate(seq):
        index = all_chars.index(char)
```

```python
        tensor[t][0][index] = 1
    return tensor

# Convert the sequence to index tensor.
def seq_to_index(seq):
    tensor = torch.zeros(len(seq), 1)
    # Shape of the tensor:
    #     (sequence length, batch size).
    # Here we use batch size = 1.
    for t, char in enumerate(seq):
        tensor[t] = all_chars.index(char)
    return tensor

# Sample a mini-batch including input tensor and target tensor.
def get_input_and_target():
    seq    = get_random_seq()
    input  = seq_to_onehot(seq[:-1])      # Input is represented in one-hot.
    target = seq_to_index(seq[1:]).long() # Target is represented in index.
    return input, target
```

Choose a Device

[4]:
```python
# If there are GPUs, choose the first one for computing. Otherwise use CPU.
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
print(device)
# If 'cuda:0' is printed, it means GPU is available.
```

```
cuda:0
```

Network Definition

[5]:
```python
class Net(nn.Module):
    def __init__(self):
        # Initialization.
        super(Net, self).__init__()
        self.input_size  = n_chars   # Input size: Number of unique chars.
        self.hidden_size = 100       # Hidden size: 100.
        self.output_size = n_chars   # Output size: Number of unique chars.

        self.rnn = nn.RNNCell(self.input_size, self.hidden_size)
        self.linear = nn.Linear(self.hidden_size, self.output_size)

    def forward(self, input, hidden):
        """ Forward function.
              input:  One-hot input. It refers to the x_t in homework write-up.
              hidden: Previous hidden state. It refers to the h_{t-1}.
            Returns (output, hidden) where output refers to y_t and
                     hidden refers to h_t.
```

```python
        """
        # Forward function.
        hidden = self.rnn(input, hidden)
        output = self.linear(hidden)

        return output, hidden

    def init_hidden(self):
        # Initial hidden state.
        # 1 means batch size = 1.
        return torch.zeros(1, self.hidden_size).to(device)

net = Net()     # Create the network instance.
net.to(device)  # Move the network parameters to the specified device.
```

[5]:
```
Net(
  (rnn): RNNCell(100, 100)
  (linear): Linear(in_features=100, out_features=100, bias=True)
)
```

Training Step and Evaluation Step

[6]:
```python
# Training step function.
def train_step(net, opt, input, target):
    """ Training step.
        net:    The network instance.
        opt:    The optimizer instance.
        input:  Input tensor.  Shape: [seq_len, 1, n_chars].
        target: Target tensor. Shape: [seq_len, 1].
    """
    seq_len = input.shape[0]    # Get the sequence length of current input.
    hidden = net.init_hidden()  # Initial hidden state.
    net.zero_grad()             # Clear the gradient.
    loss = 0                    # Initial loss.

    for t in range(seq_len):    # For each one in the input sequence.
        output, hidden = net(input[t], hidden)
        loss += loss_func(output, target[t])

    loss.backward()             # Backward.
    opt.step()                  # Update the weights.

    return loss / seq_len       # Return the average loss w.r.t sequence length.
```

[7]:
```python
# Evaluation step function.
def eval_step(net, init_seq='W', predicted_len=100):
    # Initialize the hidden state, input and the predicted sequence.
```

```
    hidden        = net.init_hidden()
    init_input    = seq_to_onehot(init_seq).to(device)
    predicted_seq = init_seq

    # Use initial string to "build up" hidden state.
    for t in range(len(init_seq) - 1):
        output, hidden = net(init_input[t], hidden)

    # Set current input as the last character of the initial string.
    input = init_input[-1]

    # Predict more characters after the initial string.
    for t in range(predicted_len):
        # Get the current output and hidden state.
        output, hidden = net(input, hidden)

        # Sample from the output as a multinomial distribution.
        predicted_index = torch.multinomial(output.view(-1).exp(), 1)[0]

        # Add predicted character to the sequence and use it as next input.
        predicted_char  = all_chars[predicted_index]
        predicted_seq  += predicted_char

        # Use the predicted character to generate the input of next round.
        input = seq_to_onehot(predicted_char)[0].to(device)

    return predicted_seq
```

Training Procedure

[8]:
```
# Number of iterations.
# NOTE: You may reduce the number of training iterations if the training takes␣
↪long.
iters       = 20000  # Number of training iterations.
print_iters = 100    # Number of iterations for each log printing.

# The loss variables.
all_losses = []
loss_sum   = 0

# Initialize the optimizer and the loss function.
opt       = torch.optim.Adam(net.parameters(), lr=0.005)
loss_func = nn.CrossEntropyLoss()

# Training procedure.
for i in range(iters):
    input, target = get_input_and_target()            # Fetch input and target.
```

```
    input, target = input.to(device), target.to(device) # Move to GPU memory.
    loss     = train_step(net, opt, input, target)   # Calculate the loss.
    loss_sum += loss.item()                                  # Accumulate the␣
↪loss.

    # Print the log.
    if i % print_iters == print_iters - 1:
        print('iter:{}/{} loss:{}'.format(i, iters, loss_sum / print_iters))
        print('generated sequence: {}\n'.format(eval_step(net)))

        # Track the loss.
        all_losses.append(loss_sum / print_iters)
        loss_sum = 0
```

```
iter:99/20000 loss:3.3421528720855713
generated sequence: Wte ino tii
e D-Ecaree
ehiwoPhoaalaI dfUh:iR sr-thtfal VWe wou te:S?

t ss kL Colr,suooe t
:;rt; tai

iter:199/20000 loss:2.8355635809898376
generated sequence: Wle, diN tMy gice th nsr,Rf ond slgAou thcocrouke gus ty
cGsr brve ntmof
krougicloado, b-i
e be sNs s

iter:299/20000 loss:2.565518710613251
generated sequence: WHan,
Ar tha, waa wheode lis;
What:
The fhee hid nheince, MoTeb, nd th:
Thuld.

NOLAMNLL mavivs sieen

iter:399/20000 loss:2.412884840965271
generated sequence: Whil il peairs weif you farlat

MLOUNA:
You wir w richis'd shimn sar th mesaave lo;
Wod. Afarjrot sto

iter:499/20000 loss:2.3610102319717408
generated sequence: WCiwh wis tiof hall dim thy, Byou yomow inx drith, Eera I
inkt blerdey y tht ed matg, th mlt me ean y
```

```
iter:599/20000 loss:2.300655872821808
generated sequence: Whotr hure'd pondall no var ur, an nssond lid luther sm
hall; ghaved.

DUONLL:
Nwobld,,
HRIme fat ds:

iter:699/20000 loss:2.2879962491989136
generated sequence: WFin, ofy deop bet!
Ol aom!
rh theepay to ce a
disus your,
I avees olve me: aid
Aag yod Mennink at ho

iter:799/20000 loss:2.2423218858242033
generated sequence: Whut du'st, jose
prece.

DiETI hayt the fove:
Whip, tukeen th wof wome ply, chea woula oncalocgios th

iter:899/20000 loss:2.2504835546016695
generated sequence: Wayziy! the vive.

Pyve.

Kpom:
Pentung if of enst .

QUKH:
Hith afr, the.

VHINLINPE:
I prewilg you

iter:999/20000 loss:2.186532106399536
generated sequence: WAMU
Thillice, coucs of res
Ied.

DUINGARI YMEUTET:
Thar oud fod heall venw and souss, wilf I losk
lo

iter:1099/20000 loss:2.157893214225769
```

```
generated sequence: Wisn nos crre centhes ar, in the leve youst in hinserow, so
no,
I claskine do haill hey laat io astou

iter:1199/20000 loss:2.134717479944229
generated sequence: Whilele to hit
No' I on thoussyou;
Af the will for meritiagre I, atr tore.

QvaCy giaif, soriy liot y

iter:1299/20000 loss:2.136445517539978
generated sequence: Will.

ISACEL:
Eimeire ou maof blevisie.

KSICA, I EIN:
She wir, efvet misele.

LUMELARICHSRANTELLIS:

iter:1399/20000 loss:2.135171273946762
generated sequence: Whyn?
Had hisky
Whace hasl! whod the con lich dott, ans of this prins; wObe putn thath the the
shay,

iter:1499/20000 loss:2.125328949689865
generated sequence: Wamy marvermow;
I we hetiin, har, up wolle sonte, his of wedd the wit, PTond gog air porpss by
froved

iter:1599/20000 loss:2.0879766941070557
generated sequence: WARD GIIUS:
Me were tig-ur mung's all ssacous.

KARD tICGS:
Hat yolf kip-:
Hith ire
urdal, mury', co

iter:1699/20000 loss:2.0961420369148254
generated sequence: WoRd Io sugh! Pryju the Vurtid.
My lowunghy som prowo.

Sinf lle you and fadll, I treak Make wead whi
```

```
iter:1799/20000 loss:2.110582414865494
generated sequence: Wis are cougft may, nesh vistly.

LUCETIUS:
I theing him thou
I Guls likm mosfivg,
Ungll
I stull it i

iter:1899/20000 loss:2.062567653656006
generated sequence: Wand for Loremy,
But have yous. I comsove If yat thist sticksswild ame mave.
CeMiting thag this pribg

iter:1999/20000 loss:2.062890853881836
generated sequence: WAnd bind hich of dadglouch ableave harkone cangens our is
beth.

KINGARY:
Live thes grece.
Bet, hied

iter:2099/20000 loss:2.054481394290924
generated sequence: Werr:
Cearceit that
But It yet you lorting thwer sur'd thy werr heak, thee yourkw, will youe what
far

iter:2199/20000 loss:2.0195585715770723
generated sequence: Wall ncintany my doof him liks our pote unat food's thouch
sie!
Ffour souns lood, ard thin.

CODYoubl

iter:2299/20000 loss:1.989940664768219
generated sequence: Whave wothon your, foul; nother ot his ilf and digh
lesthenfegrter, hiffors Tlllowe,
They woll,
We pr

iter:2399/20000 loss:2.052831085920334
generated sequence: Will to beak', Rades's your mordOld Secerrickous your chin;
I kny, thar wath that hour lioverent batw

iter:2499/20000 loss:2.028226989507675
generated sequence: Wiset of I aln fet
To you feses you, shere.
```

```
ROMTOLLA:
Moulans'st in aimio, sutine,
I halse a lerse t

iter:2599/20000 loss:2.0422002720832824
generated sequence: Wheling so thou thy hem be
the formirp, and thieress woulds Fisaul-! Hes, siobl; the wrand, the speak

iter:2699/20000 loss:2.019927833080292
generated sequence: WAll suchicttets mierst as hro garvy sse, or the mowneck the
angesimenitror thyy misen: the nop she b

iter:2799/20000 loss:1.9977871346473695
generated sequence: Wist the weanded oust duking
Wolle to rewife, you madan.

WSRCERT:
West ink Cpxart thy brownot
Whittr

iter:2899/20000 loss:2.0112564873695375
generated sequence: What toon.

Ficht her to Sesterrass oorielecatur-coupith'e thourings?
We lor if Gruther; you !or afth

iter:2999/20000 loss:1.9709619987010956
generated sequence: WbRINES:
Sot hom. poon.

LUCINEX:
O, ththee?

MENENAB:
He so loe loop om Yinding foits.
Your
Suptinus

iter:3099/20000 loss:1.9798958277702332
generated sequence: Whand sadist fother it?
So king thyculd blood rist a mase
elfors,
Whit, is nuth sold ant
rut,
Endence

iter:3199/20000 loss:1.9387240386009217
```

```
generated sequence: Wit you hy hime this all tir fe make are litare my bun cou
he parce we hereful did see.

RARCAUS:
Whe

iter:3299/20000 loss:1.943668268918991
generated sequence: Why;
But or my bono an the bace offor byouts thou:
Bull acr and cold os m that bo a no feir,
Bot, ham

iter:3399/20000 loss:1.9607846808433533
generated sequence: Wall dairsaltione him is thrich a om now pipe, thy fautinuss
think thou you britil come buanch,; of h

iter:3499/20000 loss:1.9331114721298217
generated sequence: WARD IVIO:
Mall, and thoued:
Wh in
yould ablyiels, in persthor forlenter so swervest have as, thou to

iter:3599/20000 loss:1.9821374130249023
generated sequence: Wicht, noth;
Witwer force athencer.
Woveaterinst umingronity of distech; and sur plisce
Thou thou des

iter:3699/20000 loss:1.999474092721939
generated sequence: Were, stentremy mast; makes, comes
From iqliot'

MOSCENTE VINCENTE:
If theapind, and MINII EMLLAND fi

iter:3799/20000 loss:1.957903151512146
generated sequence: Will
ID CICHICIO:
Hath rear Gging, for upos the mack bust!
Wh the fawnes. Glows, no mother so, whot d

iter:3899/20000 loss:1.9439306890964507
generated sequence: Will and As,
I kmere!

GLULIO:
I dosh froy art woxling thou what fermen, wellto!
Themetst:
```

```
I ware, th

iter:3999/20000 loss:1.959787105321884
generated sequence: Wise strampen Mis'
aller ksre. What us then timeless! thee thou he herves for attio crieds!
For co me

iter:4099/20000 loss:1.9205998468399048
generated sequence: Well abliar thor and stouls, he im knit toy All caise,
That how briths, sourse of shall
Be twilt;
And

iter:4199/20000 loss:1.9432335138320922
generated sequence: Wixtrisent
To they.
I'n's they heres iflllisst of aiser these tomm dake; this hand him strase:
Ay, the

iter:4299/20000 loss:1.9284938943386079
generated sequence: WARD LING RICHARD:
Petwar.
Beteruhy as Platel,-whichmos, I'll in, as your as tance to qulern.
Us noul

iter:4399/20000 loss:1.9212160766124726
generated sequence: WARKENGURET:
If eatlecoty pruinitine.

DUKE VI:
What's strourse mennshion be asmorn it, but serier ti

iter:4499/20000 loss:1.9317742443084718
generated sequence: WARD IV:
Sempost thund chaines, did brietrengwerst brook ho hearty;
Whor appinghour grooke-then monel

iter:4599/20000 loss:1.922023913860321
generated sequence: Wise,
Weef'd yeartung Marron that metur chearelce mue thened.
My monce's the gonk-shuas then you seen

iter:4699/20000 loss:1.9251010096073151
generated sequence: Will ET oor of commond I sass. I brough and to may thure.

LADWARD
LA:
But it If
```

```
Here;
Beal; to toenf

iter:4799/20000 loss:1.9117136931419372
generated sequence: We'd, thele if our valk crindban
Well, shall gake min and the sheel his prous.
Fis ffer dither,
As if

iter:4899/20000 loss:1.9005882000923158
generated sequence: Will you wark bleting yauldes a karn she no dreats liks
bethen;
Romed knownd free, which that take bu

iter:4999/20000 loss:1.9258299231529237
generated sequence: Will not thou I tound chalr nome he get will
I I sor yourfor mil.
Commamed in am thouger:
O fich he d

iter:5099/20000 loss:1.886707100868225
generated sequence: Went I root your deenty, but,
I the dencew't plootcelmsentuponser.

JULIET:
I, Maroser
bord,
Bork mur

iter:5199/20000 loss:1.900376615524292
generated sequence: We mast arring?

CARUERET:
Whory and Sisser him? learvantt?
As, well them, lets ast, as you, lawel, a

iter:5299/20000 loss:1.878386756181717
generated sequence: WARD I:
Lorve or did it a comsst in sun, werid, My anort mild, dides:
Marcy of laugh'd:
And lives.

Q

iter:5399/20000 loss:1.9107985603809357
generated sequence: WIO:
Nutpree on your heares's of lear not and that would it is hath me, frrench then;
To such I seevo
```

```
iter:5499/20000 loss:1.8746727848052978
generated sequence: Well sefcent
For three?
I will comesing hese vook of out clave imir the presch
I granis of of that yo

iter:5599/20000 loss:1.910058424472809
generated sequence: What slis day couft and soof yelrnd all.
Forfecle, thee?

EDu gon! Edwase makidameist with. God that

iter:5699/20000 loss:1.9118002355098724
generated sequence: Wept; Ehte cilce keoms, at with cold,
What
Havertrilinot his there you broming alast be thee, the des

iter:5799/20000 loss:1.9027468621730805
generated sequence: Will noty bleast bitushret bring, cipizen?

Hereeatant we is flow?

LOKENES:
I in brauted.

Thy, proi

iter:5899/20000 loss:1.913604689836502
generated sequence: With him.

GLBULEN
EO:
How slemmmy tudur and will, fast cound, lepprelt me, I lift:
And lask temmoriu

iter:5999/20000 loss:1.8772357249259948
generated sequence: W'sl fordoss,
Bryor worth your not not of your you him, sir, on 'mout one lalse the root
The kint mod

iter:6099/20000 loss:1.903092257976532
generated sequence: Withese of jes's and my shound is gour cous. I I meecous as
forlane prome take forus, blead merry, mo

iter:6199/20000 loss:1.865817948579788
generated sequence: Wion theuling, towe till ane or in a that wor he broor.
```

```
Pandy criess I ear I om tregor, I dost her I

iter:6299/20000 loss:1.8925991356372833
generated sequence: Whices you, whash be lood,
And loprecloo:
And letce oge tow lends houst Edgeased my soge think nemer;

iter:6399/20000 loss:1.898375827074051
generated sequence: Werd
We a ace, Whick o' usp flick'd faw a soseahs thene--ad:
Ss fore to tear o'
your oruspes in use,

iter:6499/20000 loss:1.8702972090244294
generated sequence: Wither is altion your hal.

KING RICHARD III MANTESNA:
But yournd like til acket, ander will cin much

iter:6599/20000 loss:1.8744959151744842
generated sequence: Wertur high, erade, that sealf-!

JULIET: Whon that is shalls. I pery, gartest. God this, ans, bear,

iter:6699/20000 loss:1.9031109070777894
generated sequence: Ward unjodonous and Edy frey prase.

LUCENTIO:
What remoring my voile shall sculs.

DUKE OF SORK:
You

iter:6799/20000 loss:1.8694177186489105
generated sequence: Were,
By menign.

VILAUED:
Eave master of right I briid; this fadest it bestul cal- ob,
A farmes--in

iter:6899/20000 loss:1.8947489154338837
generated sequence: Wirster!
Which spepth as in the himpens on viliss to youit Hissenced nown eare, vost
coning hery'd wh

iter:6999/20000 loss:1.8526885402202606
generated sequence: Winty-nou:
```

```
'Tis staplen id the setter!:
Clace: him coul!

CORIOLANUS:
Weve the beal that for him ard

iter:7099/20000 loss:1.9070526516437531
generated sequence: Way! Thore by as eigth toth to have thee oud and Breed
migpoevide: that!
'Twertst his venjustand!
Bus

iter:7199/20000 loss:1.8586037397384643
generated sequence: WARD LALIO:
Indet to that heaven enjones's 'Engumin dunce, that farry dayes we veoun--

KAGLRAK:
Is t

iter:7299/20000 loss:1.8811587083339691
generated sequence: Wher,
This; your Ithereant then hook voy is oore? forroursur, af it prothinst,
Noreimfy us of you hry

iter:7399/20000 loss:1.862780680656433
generated sequence: Will her laust to geep hewfy the preat nown eiprand, the
dad.'
T make of for for outs that do; in sis

iter:7499/20000 loss:1.8585619747638702
generated sequence: Wh the schnigring as that viil hand;
Thingming buty as have offelserny, I beous had wimliver, every,


iter:7599/20000 loss:1.861378959417343
generated sequence: Weil not. hoo grownry.

WERW:
Whither welm whit you his stull centores with plaving brothen.

Proveso

iter:7699/20000 loss:1.851429396867752
generated sequence: We to than me,
Thee themester bessen lid there;
Therio, bistepp's thee and bo then,,-for I past,
And
```

```
iter:7799/20000 loss:1.873474472761154
generated sequence: W-yaust that die be he! sintle?

BONTRGRESTIO:
Clay un ste of his paberry would Give porsest,
Ahd an

iter:7899/20000 loss:1.832885047197342
generated sequence: WiRthing iuse in thazed with is he this thememp it
shaptardsands,
Tisho mattours death's destectes.
O

iter:7999/20000 loss:1.8327761960029603
generated sequence: Will,
Thy vine a metalter
Sirzwes
To averit, so ouce uncishire tham
I falsting art Glaces bleis repar

iter:8099/20000 loss:1.849808156490326
generated sequence: WARD
For hearing mentering in alins goind the
To tiling for thous limenger arverlenefung nothorgite n

iter:8199/20000 loss:1.8842925691604615
generated sequence: Will maken, a titils ofse frrest forture becult
Your leavy, timpas folloour gut betsoud;
You swary th

iter:8299/20000 loss:1.8423192012310028
generated sequence: What quilfeal, full I kint.

QUEEN ELO:
Or might
Of
Bubrese his to prhcard' Cepprand
Fary.

MONTORY:


iter:8399/20000 loss:1.8689560973644257
generated sequence: Wixt a base at brawes of it nor a worlf's or's games--' have
sighal begion
Thou hore ily at you, clli

iter:8499/20000 loss:1.8245368921756744
```

```
generated sequence: Well sore the Lun taysee place;
What
Were ercoffolecioldotfal, tould to part thyed bee tripe love tou

iter:8599/20000 loss:1.8572699117660523
generated sequence: WhRre frreet
LE:
I congafar a complain:
And prase, take agroun sce that havess the nom his nechin a'

iter:8699/20000 loss:1.8383444905281068
generated sequence: WARD IILAS: Sears will creothers.

Doth:
Hears of them thou art we me clusk'd,
And dost carseffredstt

iter:8799/20000 loss:1.8103632283210755
generated sequence: W:
Wellobtly with:
sland, and all.
But thou.
Warce!
What here, extitaness
por
Which
I happrometerumas

iter:8899/20000 loss:1.813363083600998
generated sequence: WARK: Yet come, as if is
I knifed,
Letred,
Whou she the grotce.
Yet me, you blay look not a gair soin

iter:8999/20000 loss:1.8440382480621338
generated sequence: Wall: mo thoull fortine thy ront.
There and for my beary guke this he Touth and hir, have fair one yo

iter:9099/20000 loss:1.8760461902618408
generated sequence: Were wells, heario, bear my live; is, you have you of my
diesived, forezest is they I say or painerli

iter:9199/20000 loss:1.8531376147270202
generated sequence: Wh him copple hevefaly eequing you what yet to the nowlins
to abjuitblefalponed?
A garly now his fats
```

```
iter:9299/20000 loss:1.853229933977127
generated sequence: Ward, I
Nursable, and thusteft frows!
Your it to kan-y, she their this aningor's theosherter'd surdst

iter:9399/20000 loss:1.8484032487869262
generated sequence: Will cimperte onthoud k'illl; If I'll their groahing mener
offaly in body head, thou noter; then have

iter:9499/20000 loss:1.8556012654304503
generated sequence: Wherour falwoet.

CAMGLUMEO:
Though tity hears and did froch to the from we eet!
Bed comfer
Trew you,

iter:9599/20000 loss:1.8548732137680053
generated sequence: Werved so woone: cave to see?
Motort my dive thee,
She is ave.

TRISIBE:
No wiin him, soalds
To will

iter:9699/20000 loss:1.8422003877162934
generated sequence: Well, Lo kist. Thee in that of hanour ip'ting all met hold
you talk to uwy wongritrrrireers, as I it,

iter:9799/20000 loss:1.8252549731731416
generated sequence: Winging thour bear urress exently demal youse.

First But! A their grack than doness in with bigliess

iter:9899/20000 loss:1.8257629776000976
generated sequence: Well you it where byor me hap; So ant, but stave offeler
upet.

ESCARES:
Tamway; to prame camourdes,

iter:9999/20000 loss:1.8411976742744445
generated sequence: Warks, of deed a back,
Coance.

LUCEPBUT:
Proathy wrecks,
```

```
On; rum brokiss; waefet; blaids.
As saulopp

iter:10099/20000 loss:1.8401658606529236
generated sequence: Wird af
By onded your rifable, she heange
Hat.

GLOUCESTES:
Yot revillupted madider,
Lord your so, on

iter:10199/20000 loss:1.806706613302231
generated sequence: WARD:
I I have wore Goder of Forder sop
Ant had, ladge
Dot sairs, I the tate is I hath are moir.

KAN

iter:10299/20000 loss:1.8341571688652039
generated sequence: WARD:
What cause tonceartch,
Sty wiel shane Lord,
My's sabe your thare to he here, laid, and my ware,

iter:10399/20000 loss:1.835849723815918
generated sequence: Wist, piines, and hol my breat,
My beild heart bidinissef mind at, god appore well, so were in your p

iter:10499/20000 loss:1.8258265662193298
generated sequence: Wind had
T mord, you hap ruch tolks her for myserp.

Nurde:
And my crusted
To the gows thy lord--
Was

iter:10599/20000 loss:1.801500688791275
generated sequence: WARD IV:
And for patter to hore the bush; will be stoll becamo sors with poing claintnet
anand I'er s

iter:10699/20000 loss:1.832945817708969
generated sequence: Wixt
Sirpaster's not them, ispion,
Nided ut in thy pity macksase it.
```

```
RICHARD III:
They art Morse,
Is

iter:10799/20000 loss:1.875517508983612
generated sequence: Warwifith, I gnent
Twas the chold sud
I were trul neing in that the one within! Will I that I wadry w

iter:10899/20000 loss:1.8585907626152038
generated sequence: Which a thanksoth,
When hath,
And maymil. Limst it, Refule the mor'd hold you.

AUTILY:
Shatam. Henia

iter:10999/20000 loss:1.8416907918453216
generated sequence: Wwis. Flet son, then.

VINGULIO:
I my goor good butigred slast;
Of the sark-own day sin
un's sigch so

iter:11099/20000 loss:1.8429055416584015
generated sequence: Wise liad inchads the for y thot on me!
'Tis a go men this thee

ESCAMILL: I wo hoball'Ttly wifethm?


iter:11199/20000 loss:1.84899258852005
generated sequence: Wis subbles som I wis the pleis!
And than; ald, Prose deal!
Gokes.
So natnebione. Mipt'st is eartait

iter:11299/20000 loss:1.8186950480937958
generated sequence: Whick, sones and will then ascress the been sinkersitn: He,
set desome
go surs, with hesablang batold

iter:11399/20000 loss:1.8219895255565643
generated sequence: What my of my far my had know, thou loss;
Henour more bast
Why awos if wer like
```

```
Wity him.

Floves,
Th

iter:11499/20000 loss:1.813836443424225
generated sequence: W:
Aman stay. Herset more blempive aitted knom hoom doss
Bithea insticitocligertiely?

PETRUCHIO:
Bou

iter:11599/20000 loss:1.8146890807151794
generated sequence: WhHe her?

LUCIO:
His drend
To putrice, ary,
Marry' hars, Cave when then bring brive woll, sprace
Ee

iter:11699/20000 loss:1.8243285727500915
generated sequence: WARD KING HENRY VI:
WWAll asless o'

JULIO:
Alow make it cont, toce swill--
Woy weaven? Cane my criev

iter:11799/20000 loss:1.8202486896514893
generated sequence: Will aled? I'll we lahtue me. No may her with the which all
it, whof they wilth that if
Thon wite boa

iter:11899/20000 loss:1.8218411350250243
generated sequence: Will not have
Tere now farriked a remongst a manghole
To hirs stamide in'em in handing.
A raves, our

iter:11999/20000 loss:1.816653039455414
generated sequence: Well;
And at me thee;
As to bolkece ou out in mygers after stamsulied tont.

LADY DINGUTIO:
```

```
JULIET'S

iter:12099/20000 loss:1.837640668153763
generated sequence: We unwarst pray likin is sill of Mesirg ol, that so love you
may, smooth; for less'd; went once, to t

iter:12199/20000 loss:1.8181812012195586
generated sequence: WICK-
urds for peace well.

GREMIO:
Their arce Iing upon treus
'Gosb
ruce, thore.

DUKE OF YORK:
O ma

iter:12299/20000 loss:1.852973048686981
generated sequence: WARl beol on Lord:
And, tere to your now'd whis noy
nothers yer thou hase they in jesed not the doy h

iter:12399/20000 loss:1.8177010440826415
generated sequence: WAll?-

Orvispeff
Where's creasal onoughter, pase a this for reserful--'Trenese, noble disinis,
one t

iter:12499/20000 loss:1.838196061849594
generated sequence: WARd
Fore to shem fory to the hrown a kingstive, of benother daother ffoedst seglet,
On the hand cous

iter:12599/20000 loss:1.8397408318519592
generated sequence: W GATackent anconge Her. resipe that how went un word!
Had flowans
yge.

LADY GFRIZNIUS:
'Timasiin o'

iter:12699/20000 loss:1.8349151957035064
generated sequence: Ward he prorconge. Towed Rob to--good, to wrich,
And my,
Nor me knowly,
What grince you do marristy.
```

```
iter:12799/20000 loss:1.8163355052471162
generated sequence: What war,
by my for-walk'd
Him a
 nore I bid
My get uchain not must yave father'se
Her lieth thouse.


iter:12899/20000 loss:1.8034240114688873
generated sequence: Will I was parcing, and men the my lord it, pus tell,
intreming as pow in ever the raught seck me thi

iter:12999/20000 loss:1.7981489896774292
generated sequence: WDAULE:
He days he be in bid nonking for jeckifint
Fare worsscar to have mys:
Gots, wind come, of hor

iter:13099/20000 loss:1.825510538816452
generated sequence: WICKt atrelds my.

Pander fliges
That is Were res bratter sabalmwer
That in thou brofeay, now eack or

iter:13199/20000 loss:1.8014509749412537
generated sequence: Wcresting is death a
Rome than high-no inlown;
It hender.
Nothsturiar him, bothilat knerger balt Morn

iter:13299/20000 loss:1.8129298377037049
generated sequence: Ward, surdy desske.

SOMPERITIO:
You,
But nelly me this betty use with you brist,
for strad, shoring

iter:13399/20000 loss:1.8613284182548524
generated sequence: Werphy
Wese ossepp crownry hesear Heavens of the essirmance his I belance, empradeno:
I so you eouth

iter:13499/20000 loss:1.81771000623703
```

```
generated sequence: Wind
Than irst be salos,
And brace minses illon me. Peanluss'd live the rate--are the bares can's mot

iter:13599/20000 loss:1.8437832021713256
generated sequence: Wheard theservose

Peavoul, that
were you monand when lews,
You suivent in best quientie, I am took o

iter:13699/20000 loss:1.828867655992508
generated sequence: Where most his I dake some the placen:
And with thee tendat? O thit there begratly, runk mod Seame.
I

iter:13799/20000 loss:1.858891419172287
generated sequence: Wh wiferdly you say, quircry!

DUCEdes my gay
Then a I his broom. scanded dempener brothoring;
Is tha

iter:13899/20000 loss:1.8320943117141724
generated sequence: Will sirrer you well, witizines in till bost will is plorday
our eavourn will hand; hear love, to tay

iter:13999/20000 loss:1.7956982147693634
generated sequence: Wespod to thing deciuse,
The time, good lord, one,
The vourt timstions
Worl; geave live, one lessioss

iter:14099/20000 loss:1.8399743509292603
generated sequence: Who dusth, pariom. Mays infice! Go, not, wear, it ave.
Sheer lapes, not be
chover, fordic fent he, by

iter:14199/20000 loss:1.8012925827503203
generated sequence: Warrur, queat.

DUCHENS:
Why, good vartet Here you have a move ut: be tame of is me the ngue, chalk.'

iter:14299/20000 loss:1.8360065758228301
generated sequence: Will fity po hithorn, to pity in tifur lit.

WARWIET:
```

```
PRINCE:
Hearfy yough, who sheies,
By bowny, pr

iter:14399/20000 loss:1.8096870517730712
generated sequence: What son it; queetinide; by the rip me,
As wimbs
Mary afferst bit to to thine, you: I who sweets and

iter:14499/20000 loss:1.8053198921680451
generated sequence: Wes not thee spile fay cambs Cimfount to do bods of mise
And which wids Romak good contech
criemter.


iter:14599/20000 loss:1.818728621006012
generated sequence: WAll that disfell steet ip prove,
What day of this not,
What may and not
Husting my trahe.

BRURD Clo

iter:14699/20000 loss:1.8390274477005004
generated sequence: What was him wreein
And she hintree, wifor think to though the centrange.
Mistory him.
My live wive m

iter:14799/20000 loss:1.7835863649845123
generated sequence: WARt Engle; yourselor queapords,
What not, are wome, ass poar, and that denond then? Thy with falled

iter:14899/20000 loss:1.8252069890499114
generated sequence: WARe exeeg
Say;
Your for they itheaty it.
Whee jode the dever'd-
What mode I lighter, my;
Loteioot ot

iter:14999/20000 loss:1.8045294773578644
generated sequence: WAse mistison elung the parstars: So my how forge of curme,
Of Sended, ts my my comant, who do thee f

iter:15099/20000 loss:1.8067199730873107
generated sequence: What Racker as with that verilakest,
```

```
For it Remer may the duking heaser:
And highitsed, own no mave h

iter:15199/20000 loss:1.795635998249054
generated sequence: What no man,
Boundies stardecong she shamblirdin the rise,
Ay,
And what not more good nom then to clo

iter:15299/20000 loss:1.812180860042572
generated sequence: What marts, voigh he did bether, to is but your the lating
abed for me it,
latielfe.
Cousen lent; bri

iter:15399/20000 loss:1.8089922595024108
generated sequence: Wilt that conted friends,
Thepher,
I the veath, meon wither sordon of I mave your ppack lews, lake hi

iter:15499/20000 loss:1.7675546562671662
generated sequence: Wive to and periet:
As the but and
Hecriee with luth comee sothagch my goldes,
Offel It in thou ange


iter:15599/20000 loss:1.8484687888622284
generated sequence: White, his priarming ip his her, you dreed.

DUCKINGHAM:
Here's vorman:
I'll made with ip my brame.




iter:15699/20000 loss:1.7917567110061645
generated sequence: WVithray but reseaght in sory the shall could way it it I in
dreats the vine to neys, thy though is t

iter:15799/20000 loss:1.8526648223400115
generated sequence: WAy:
I am God-age,
Yeu artar as:
What, as wind wes stall for out a
acones thing, woader be
tha kence
```

```
iter:15899/20000 loss:1.8051566326618194
generated sequence: Will our beleane of he suokl:
Wither for,', laokinger,
Heard you.

Cie
lye is joanus
To, hear fort or

iter:15999/20000 loss:1.8206405687332152
generated sequence: Werd for the rrit?
NGREY:
Ol.

First Comes.

ROMEO:
Which od:
Whicererd for almorncius, to stablot,--

iter:16099/20000 loss:1.8199061596393584
generated sequence: Wenting offounsas or to spond Caurd
To heach in tooth.

SiTh':
What mud
But me may Isor, and ' than a

iter:16199/20000 loss:1.7799653732776641
generated sequence: Warbshy,
Trrew
Tuding of nay your corriart this arm the sifter thy drought hindirie,
Grvear,
Engroes,

iter:16299/20000 loss:1.8268259406089782
generated sequence: WIAN:
Not the your the eant the 'tise eling this feor yed henes you, well, where'd his
speak for upou

iter:16399/20000 loss:1.8168642175197602
generated sequence: What so dough outhar?

CORTO:
How a fingzed meath,
Whatly withard at sue, me, he hear weechickly warn

iter:16499/20000 loss:1.7864623117446898
```

```
generated sequence: Wilt love, in Edcoosent
West for fathergan so done noVer love,
O herch,
Caccess a body? thar tander;


iter:16599/20000 loss:1.8033489608764648
generated sequence: Whatlly every of myself my tommence mistle,
What looks,
For contech.

SAPELIO:
Nor and oll: to such h

iter:16699/20000 loss:1.846905220746994
generated sequence: Wind.

WARPET:
Ohat for rid foot by erose fath an poors. Doverive,
Sir whe mone the devaie, and in it

iter:16799/20000 loss:1.83065105676651
generated sequence: Will not
Lo druch
I do his colsence he now hig;
Who craible ady I his: forsmut, first come it my but

iter:16899/20000 loss:1.807357084751129
generated sequence: Was thou lood
I sherts.
Unoor with hillother,
Do neving tilly ure.

MERINIUS:
I crich,
Lo know must
D

iter:16999/20000 loss:1.8120283424854278
generated sequence: WAy, herien
do He now,
Anding nom, to say nor and
Nave true? Oxpect then,
Is chaids ever his goreneen

iter:17099/20000 loss:1.8333758640289306
generated sequence: Way ramted wiend be they bath fave cinto to that aid;
Wathough trushcew-we wixtor,
```

```
Your genomh-ded.



iter:17199/20000 loss:1.8479491913318633
generated sequence: WARD ED:
Should
And mubd lattirch chose
But you devite, wild have her newhet thou hrre if MiQither:
T

iter:17299/20000 loss:1.7898467457294465
generated sequence: Where you my sweatley Gart nosance.

ANGELO:
This nothe to trith met. Cut you put-the fids dony, ant

iter:17399/20000 loss:1.8342473375797272
generated sequence: Worce engest' the shall you art valte is, time,
In ast thy pussely me.

PROSPELO:
Vencieat.

LUCEERES

iter:17499/20000 loss:1.8290023982524872
generated sequence: Wherefeffage.
Thee Ywank.

VOLUMNCA:
Is that I what, here yet IF Arail,
How snapo,
Finhshiferer one b

iter:17599/20000 loss:1.7872740137577057
generated sequence: Wils, Angid!

GLENPSTER:
A may of mant a kifate ess, shoulds on
Whe predung gre's we fore: for the gr

iter:17699/20000 loss:1.8193223357200623
generated sequence: Wold good meew they cangle kner friur his knews dles My
sward,
Addy of my fouse, but were? what I the

iter:17799/20000 loss:1.78619868516922
```

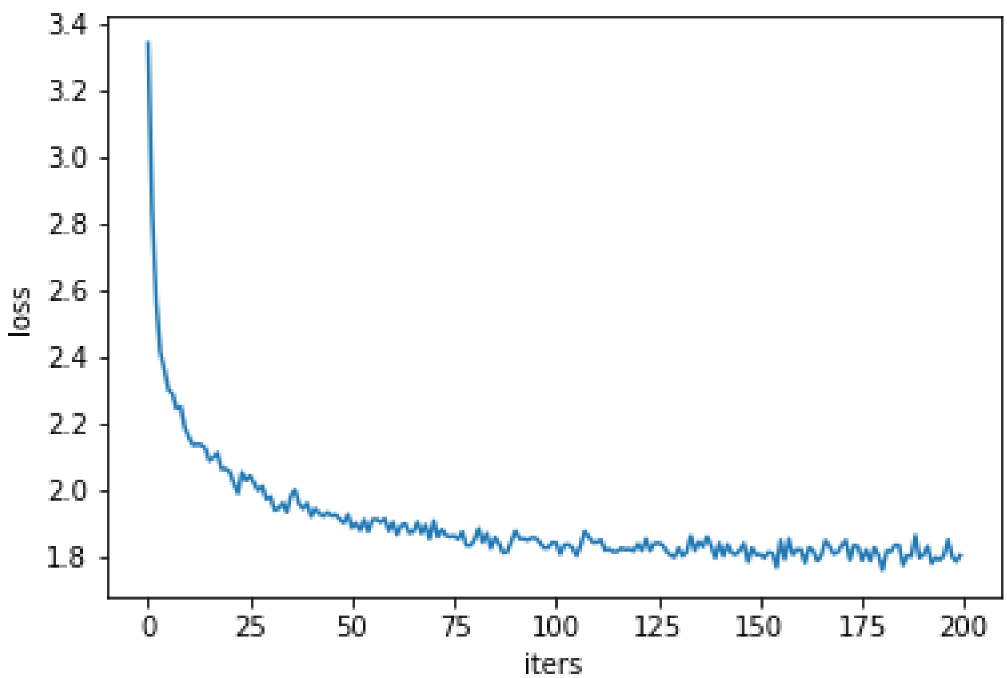
```
generated sequence: What knaw you, duse
The'st:
And before?
If that you gone fordos!
As deart, bown and thy blooved a sta

iter:17899/20000 loss:1.82608771443367
generated sequence: Where Grumint tome.

HASTIS:
He gond;
He
pempentlios, he remerch?

HENRY There bas eace of,
At the pr

iter:17999/20000 loss:1.804635945558548
generated sequence: Wels you blepty, prayol! grours,--for undery and you shall
nottent morent frows this here in huld you

iter:18099/20000 loss:1.7592899358272553
generated sequence: WAy, spards,
What sweet'd the profored any compace enverters, So out the parly confly
My one thee; an

iter:18199/20000 loss:1.816056205034256
generated sequence: WAy, you standant
more than do, my carved cluqier she make upone and he wettect of a criedtent
hourit

iter:18299/20000 loss:1.8142119789123534
generated sequence: Why, of a foucklest,
Alas ffilingestme collay,
A! thee at misty?

MARIGHAM:
What: son, good elsestizt

iter:18399/20000 loss:1.8339669978618622
generated sequence: Wilarion:
Cund uping Come plats with my my toutila now will them by shall you marry
Warwer
Cenisp
Whi

iter:18499/20000 loss:1.8334177446365356
generated sequence: Wandrs, ends my foosesad mindst it was Alaness come sweet
```

```
know exule Cary conseffares?
But be,--
Lase

iter:18599/20000 loss:1.7741870546340943
generated sequence: Whed be soned Aruhe love.

ANGELO:
That list ha? daughar! face a barthmouds,
Take ladee: my, night so

iter:18699/20000 loss:1.8026359450817109
generated sequence: Werd?
Awost king,
As bear,
Where:
Nook! Her my be that you beon And no kist,
Cordou, and bedous to ne

iter:18799/20000 loss:1.8003920125961304
generated sequence: Well, knealt, butle, our,
He wad sun to seall,
A mare upen what sumbondswalcure that not.

IS:
To see

iter:18899/20000 loss:1.8631572067737578
generated sequence: Wind
With you beep sir, conather stond spompent yot; and with or the portone our lett
you think to th

iter:18999/20000 loss:1.7962331199645996
generated sequence: Whil' mosh not kentle, 'tis hither?

DUCELAN ALIO:
O, thor aball! I'll nothera fears,
Stult
Tay, and

iter:19099/20000 loss:1.805171309709549
generated sequence: What so Crish one speed-tirmbet won ene shore,
What is matwer men thy Extcauthour sir,
With powber th

iter:19199/20000 loss:1.829545907974243
generated sequence: Wos the say thee no mind is goagm.
```

```
GLows thas as the revory hame in witter but fering thy saminger
T

iter:19299/20000 loss:1.7782178378105165
generated sequence: Who! Servon
And I
Kidrices fail or thou the heavy me gonementy than I know to the king a cere this
us

iter:19399/20000 loss:1.7960975062847138
generated sequence: Well my lifeth rewerting my should will be give thy stouth
till for that a kist in!
:
It behald Steak

iter:19499/20000 loss:1.7891186916828155
generated sequence: Wiin, for he with rese
As work! of you havaunt against in thich a ot I a plongpew it pray, And every

iter:19599/20000 loss:1.8028966748714448
generated sequence: Wance, untwidves-shalfshipving for.---
Seef modester, to be it contendsed
Dids word in the reon:
Thrr

iter:19699/20000 loss:1.8490141546726226
generated sequence: W-Rissipes raule war, isk;
The wift if me, down for! a fades.
Herce iry,
We't begift's scull with the

iter:19799/20000 loss:1.8004979372024537
generated sequence: Weare
I faith you we makes,
Walcemp, mokess.

CAPULET:
O Roming priar This his swilly for you all Say

iter:19899/20000 loss:1.7841939735412597
generated sequence: Wher are id!
Shander death, mleal thind wicise that shall'd his plature must the revens that
to veub

iter:19999/20000 loss:1.8029308915138245
generated sequence: Well of dost, tuprers to should, dotia, moke weed,
Who, noble mild!
Awam hopper frient vartrow? her f
```

Training Loss Curve

```
[9]: plt.xlabel('iters')
plt.ylabel('loss')
plt.plot(all_losses)
plt.show()
```



Evaluation: A Sample of Generated Sequence

```
[10]: print(eval_step(net, predicted_len=600))
```

```
While, 'Virsanst in with doon no:,
Betando:
By are her shris, to dobble ob him.
Go, land
Herselful the plaise of, shis fellless brother buud;
Whit
I which Marture prop ope that jounts mine loved have a poptast of it stand of
may brother swif, a poversion
A onan store engigs stabe lis
fir, set my desibutions of the good there to my heas on her bends of thee
yougranle-wardond:
I have as is shaw bids of friesmy so.
Be
```

Tays Fortham too: and, foou and porpost hen goody of you art my cooth'st not barnster for
By lup from a gell predss savost, I shRCll.
Lequ,
Oroay everied ong of a may, grithain son b