

```
In [1]: import sys
from datetime import datetime

import torch
from torch.utils.data import DataLoader

try:
    from google.colab import drive

    IN_COLAB = True

    drive.mount('/content/drive')
    path = "/content/drive/My Drive/Colab Notebooks/"

    # for python imports from google drive
    sys.path.append(path)
except:
    IN_COLAB = False
    path = "./"

from hmlstm import HMLSTMNetwork, utils as hmlstm_utils
from utils.datasets import CharactersText as TextDataset
from utils.trainer import Trainer
```

```
In [2]: use_cuda = torch.cuda.is_available()
# use_cuda = False

if use_cuda:
    Trainer.print_cuda_info()
    torch.cuda.empty_cache()

device = torch.device('cuda' if use_cuda else 'cpu')

project_name = "trump"

seq_length = 100

path_data = path + "projects/" + project_name + "/data/"
train_dataset = TextDataset(path_data + "train.txt", seq_length)
val_dataset = TextDataset(path_data + "val.txt", seq_length)

batch_size = int((len(train_dataset) / 3) + 1)
train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True, pin_memory=use_cuda)
val_loader = DataLoader(val_dataset, batch_size=len(val_dataset), shuffle=True, pin_memory=use_cuda)

cuda device: 0 / name: GeForce GTX 1080 Ti / cuda-capability: (6, 1) / memory: 10.91522216796875 GB
```

```
In [3]: input_size = 1
embedding_size_input = 128
hidden_sizes = [100, 150, 100, 75]

embedding_size_output = 96
linear_sizes = [128]
output_size = len(TextDataset.VOCABULARY)

model = HMLSTMNetwork(input_size, embedding_size_input, hidden_sizes, embedding_size_output, linear_sizes, output_size)
trainer = Trainer(model, train_loader, val_loader, device)
```

```
In [4]: epoch = 200

trainer.set_device(device)

fn_save = datetime.now().strftime("%Y%m%d_%H%M%S") + "_" + str(trainer.epoch) + ".pt"
path_states = path + "projects/" + project_name + "/states/"

#trainer.train(epoch=epoch, lr=0.01, validate=True)
#trainer.save_state(path_states, fn_save)
```

```

In [5]: embedding_size_input = 128
hidden_sizes = [256, 128, 64]

embedding_size_output = 256
linear_sizes = [128]

model = HMLSTMNetwork(input_size, embedding_size_input, hidden_sizes, embedding_size_output, linear_sizes, output_size)
trainer = Trainer(model, train_loader, val_loader, device)

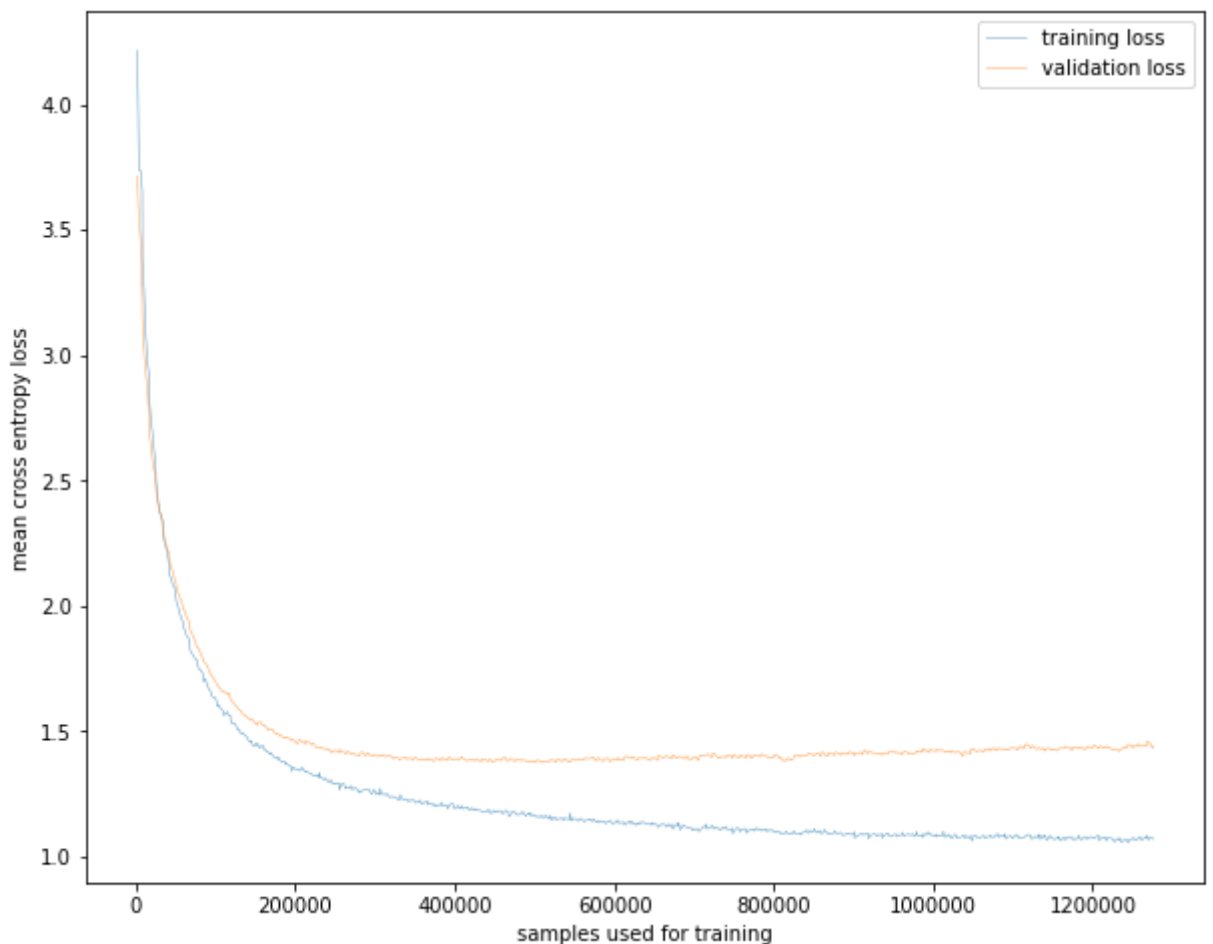
fn_load = "20200222_210342_trump_150.pt"

trainer.load_state(path_states + fn_load)
trainer.plot_loss()

trainer.set_device("cpu")
text = "So when I call for moratoriums and I call for"
sampled_text, h, z = trainer.sample_text(text, length=100, k=3, online=True)

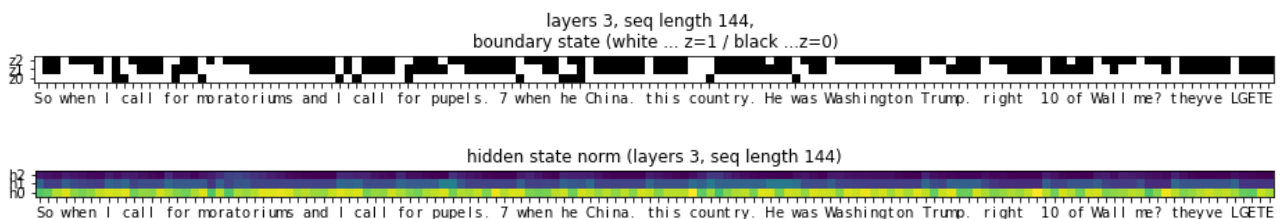
hmlstm_utils.plot_z(z, list(sampled_text))
hmlstm_utils.plot_h(h, hidden_sizes, list(sampled_text))

```



we are sampling on cpu

So when I call for moratoriums and I call for pupels. 7 when he China. this country. He was Washington Trump. right 10 of Wall me? theyve LGETER



```

In [6]: embedding_size_input = 100
hidden_sizes = [75, 100, 100, 125]

embedding_size_output = 150
linear_sizes = [125, 75]

model = HMLSTMNetwork(input_size, embedding_size_input, hidden_sizes, embedding_size_output, linear_sizes, output_size)
trainer = Trainer(model, train_loader, val_loader, device)

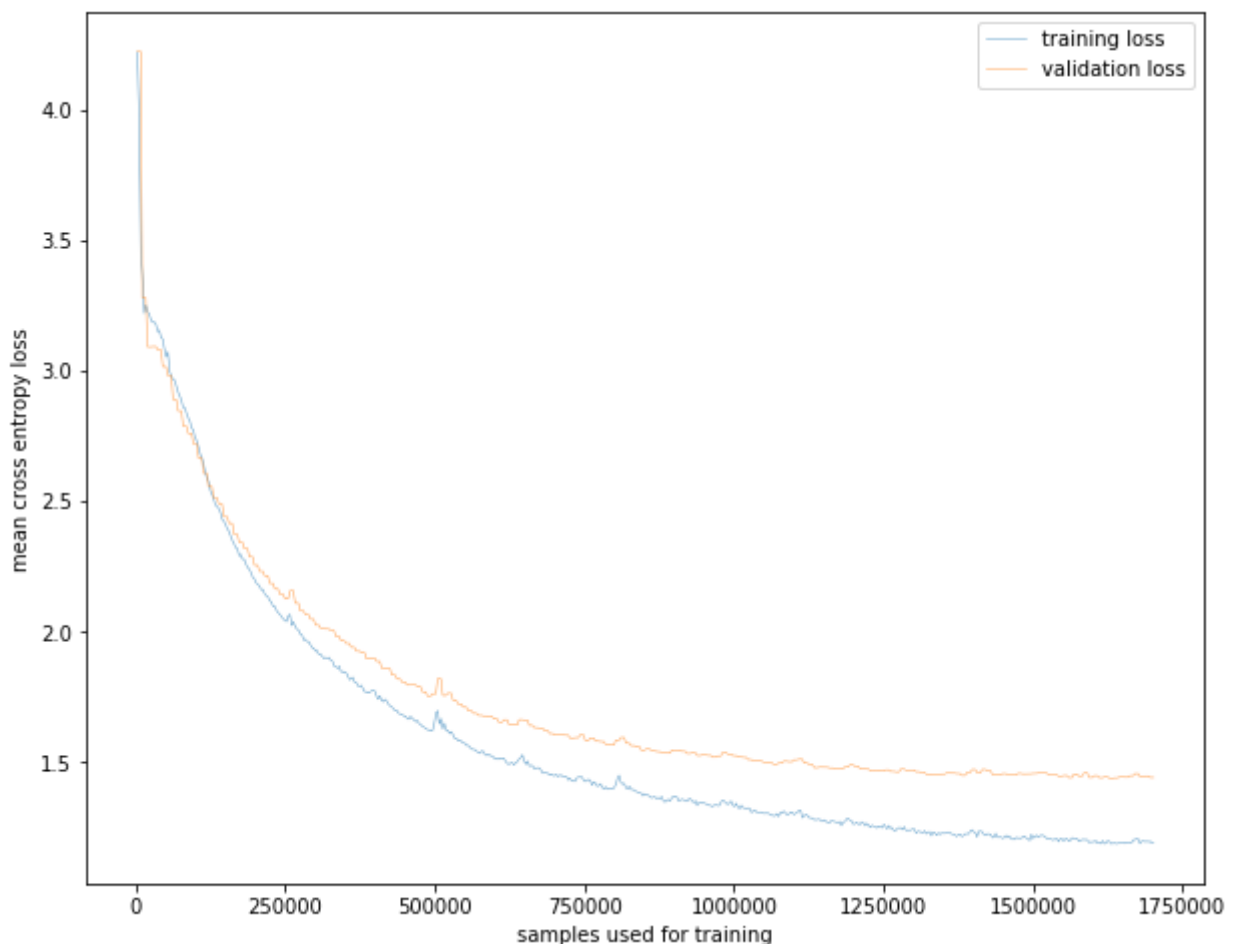
fn_load = "20200223_173634_0.pt"

trainer.load_state(path_states + fn_load)
trainer.plot_loss()

trainer.set_device("cpu")
text = "So when I call for moratoriums and I call for"
sampled_text, h, z = trainer.sample_text(text, length=100, k=3, online=True)

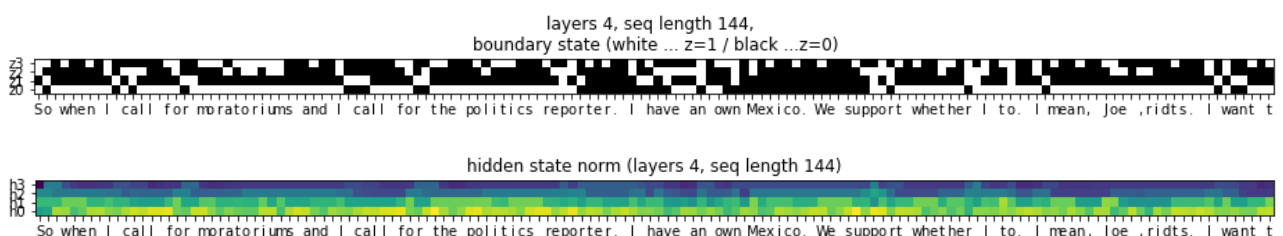
hmlstm_utils.plot_z(z, list(sampled_text))
hmlstm_utils.plot_h(h, hidden_sizes, list(sampled_text))

```



we are sampling on cpu

So when I call for moratoriums and I call for the politics reporter. I have an own Mexico. We support whether I to. I mean, Joe ,ridts. I want th



```
In [7]: embedding_size_input = 150
hidden_sizes = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

embedding_size_output = 150
linear_sizes = [100]

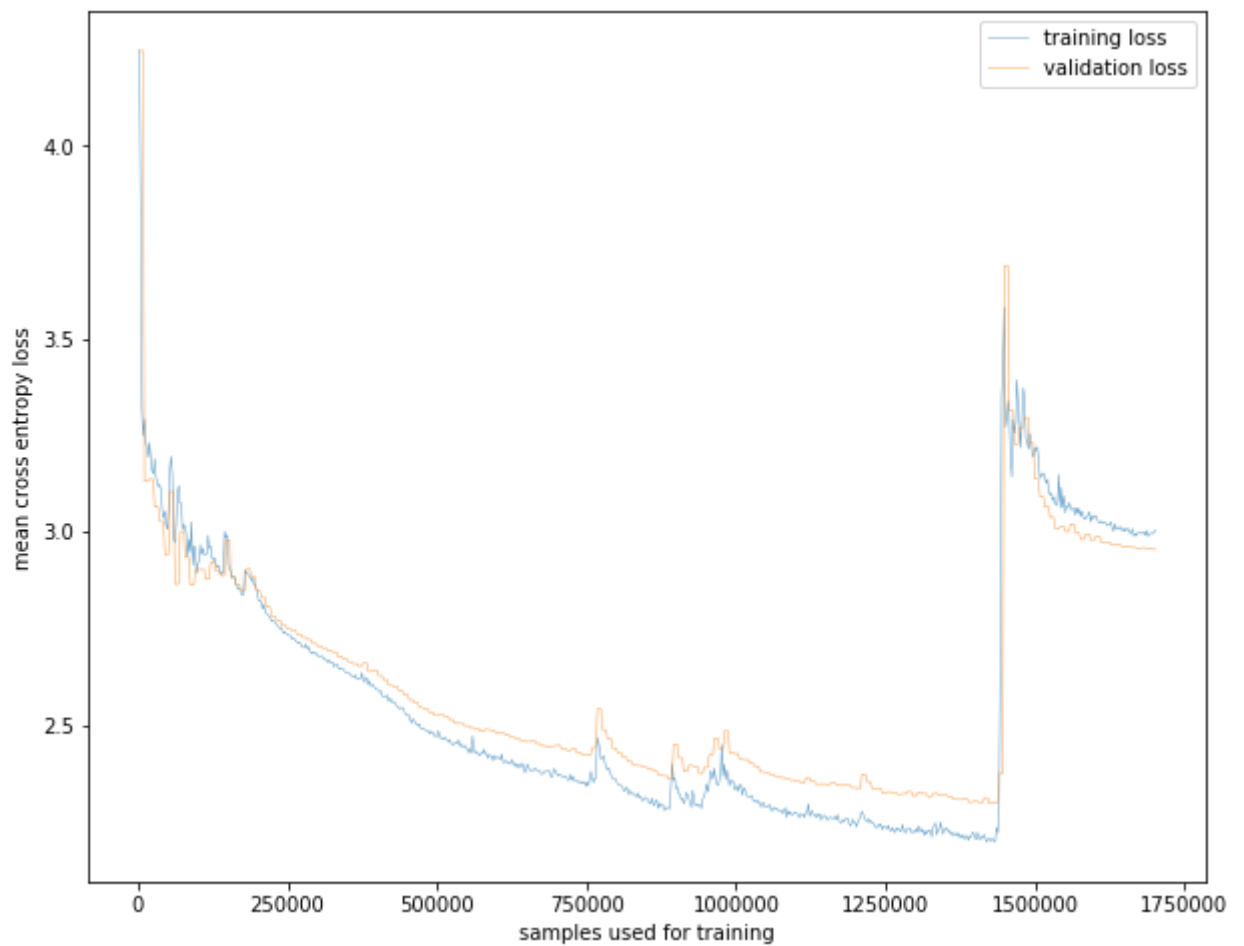
model = HMLSTMNetwork(input_size, embedding_size_input, hidden_sizes, embedding_size_output, linear_sizes, output_size)
trainer = Trainer(model, train_loader, val_loader, device)

fn_load = "20200223_181542_0.pt"

trainer.load_state(path_states + fn_load)
trainer.plot_loss()

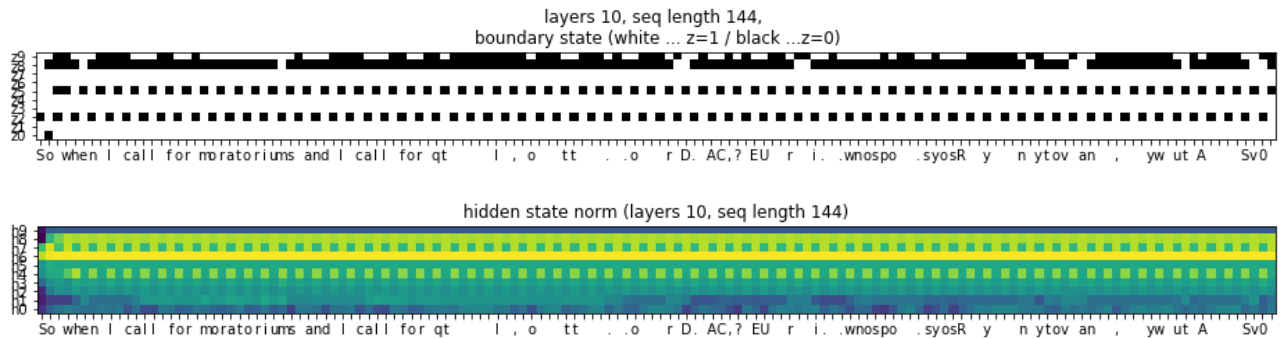
trainer.set_device("cpu")
text = "So when I call for moratoriums and I call for"
sampled_text, h, z = trainer.sample_text(text, length=100, k=3, online=True)

hmlstm_utils.plot_z(z, list(sampled_text))
hmlstm_utils.plot_h(h, hidden_sizes, list(sampled_text))
```



we are sampling on cpu

So when I call for moratoriums and I call for qt I , o tt . .o r
D. AC,? EU r i. .wnospo .syosR y n ytoV an , yw ut A Sv0 a



```

In [8]: embedding_size_input = 128
hidden_sizes = [100, 150, 100, 75]

embedding_size_output = 96
linear_sizes = [128]

model = HMLSTMNetwork(input_size, embedding_size_input, hidden_sizes, embedding_size_output, linear_sizes, output_size)
trainer = Trainer(model, train_loader, val_loader, device)

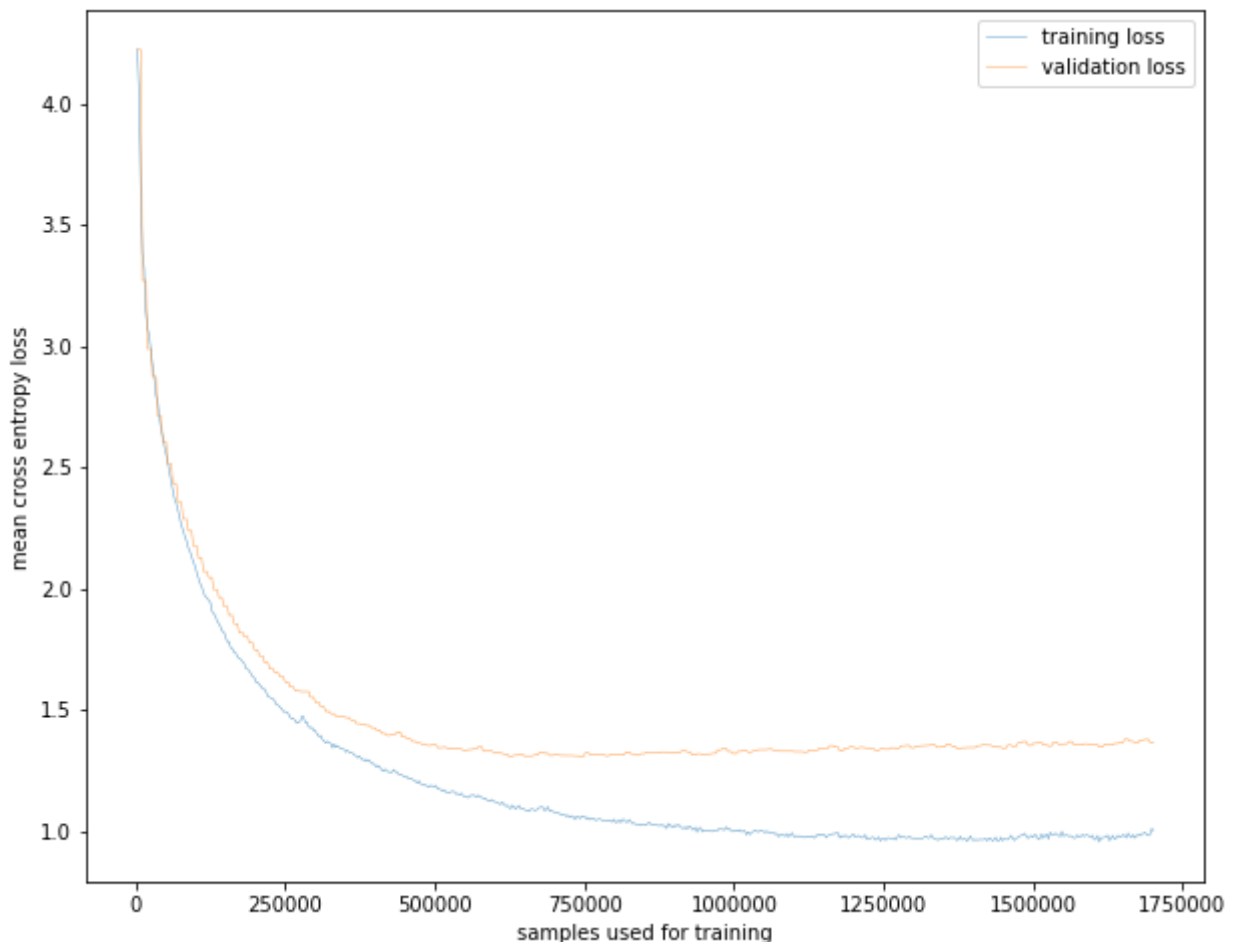
fn_load = "20200223_192134_0.pt"

trainer.load_state(path_states + fn_load)
trainer.plot_loss()

trainer.set_device("cpu")
text = "So when I call for moratoriums and I call for"
sampled_text, h, z = trainer.sample_text(text, length=100, k=3, online=True)

hmlstm_utils.plot_z(z, list(sampled_text))
hmlstm_utils.plot_h(h, hidden_sizes, list(sampled_text))

```



we are sampling on cpu

So when I call for moratoriums and I call for the Denicated people have no s
trent. We want to pression South Korea? This is Going your head, Hey,

