

02-deep-learning-tutorial

August 11, 2020

0.1 Building a dataset, training a Seq2Seq model, and testing it

The Virginia Tech Natural Motion Dataset contains .h5 files with unscripted human motion data collected in real-world environments as participants went about their day-to-day lives. This is a brief tutorial in using the dataset and then training and testing a neural network.

This tutorial illustrates how to use the shell (.sh) scripts to train a seq2seq model (particularly **train_seq2seq.sh** and **test_seq2seq.sh**). Similar shell scripts are also available for the Transformers (see **train_transformer.sh** and **test_transformer.sh**)

0.1.1 Building a dataset

We will first cover how to build a dataset with data from a few participants using the build-dataset.py file.

We are running the script from a Jupyter Notebook, but this can just as easily be run as a shell script (see build_dataset.sh).

In this case, we are drawing data from the h5-dataset folder located in the cloud. We are going to output the training.h5, validation.h5, and testing.h5 files to the folder data/set-2.

We will be using participants 1, 5, and 10 (P1, P5, P10, respectively) and extracting normOrientation and normAcceleration data on a few segments (norm* means data normalized relative to the pelvis). As output data we will be extracting normOrientation data for every segment.

In other words, our task is as follows: use orientation and acceleration from a set of sparse segments and try to train a model mapping that input data to orientations for every segment on the human body.

```
[1]: !mkdir -p /home/jackg7/VT-Natural-Motion-Processing/data/set-2
!python build-dataset.py --data-path "/groups/MotionPred/h5-dataset" \
--output-path "/home/jackg7/
→VT-Natural-Motion-Processing/data/set-2" \
--training "P1" \
--validation "P5" \
--testing "P10" \
--task-input "normOrientation normAcceleration" \
--input-label-request "T8 RightForeArm RightLowerLeg_
→LeftForeArm LeftLowerLeg" \
--task-output "normOrientation" \
```

```
--output-label-request "all"
```

```
2020-08-10 13:09:31 INFO      Writing X to the training file group...
2020-08-10 13:09:37 INFO      Writing X to the validation file group...
2020-08-10 13:09:40 INFO      Writing X to the testing file group...
2020-08-10 13:09:50 INFO      Writing Y to the training file group...
2020-08-10 13:09:57 INFO      Writing Y to the validation file group...
2020-08-10 13:09:58 INFO      Writing Y to the testing file group...
```

0.1.2 Training a seq2seq model

We can now train a seq2seq model to map the normOrientation and normAcceleration data from the sparse segments to the full-body normOrientation data.

We will be using a seq-length of 30 (at 240 Hz) downsample it by a factor of 6 (to 40 Hz). The resulting sequences will be of length 5 for the input and output. The in-out-ratio will then be used to reduce the output sequence length to 1.

The input sequence will be of shape (B, 5, 35) and output shape will be of shape (B, 1, 92). Orientations are stored as quaternions, so orientation value will be 4 in length. The number 35 comes from our use of 5 segment orientations and accelerations or $5 * 4 + 5 * 3 = 35$. The full-body has 23 segments and we're predicting orientation values for each one or $23 * 4 = 92$

We're training a seq2seq model with a hidden size of 512, a bidirectional encoder and dot product attention. The model will be trained for a single epoch.

Our loss function for training will be the L1Loss and our validation losses will be the L1Loss and the QuatDistance (cosine similarity) loss.

```
[2]: !mkdir -p /home/jackg7/VT-Natural-Motion-Processing/models/set-2
!python train-seq2seq.py --task conversion \
--data-path "/home/jackg7/VT-Natural-Motion-Processing/
→data/set-2" \
--model-file-path "/home/jackg7/
→VT-Natural-Motion-Processing/models/set-2/model.pt" \
--representation quaternions \
--batch-size=32 \
--seq-length=30 \
--downsample=6 \
--in-out-ratio=5 \
--stride=30 \
--learning-rate=0.001 \
--num-epochs=1 \
--hidden-size=512 \
--attention=dot \
--bidirectional
```

```
2020-08-10 13:10:06 INFO      task - conversion
2020-08-10 13:10:06 INFO      data_path - /home/jackg7/VT-Natural-Motion-
```

```

Processing/data/set-2
2020-08-10 13:10:06 INFO      model_file_path - /home/jackg7/VT-Natural-Motion-
Processing/models/set-2/model.pt
2020-08-10 13:10:06 INFO      representation - quaternions
2020-08-10 13:10:06 INFO      auxiliary_acc - False
2020-08-10 13:10:06 INFO      batch_size - 32
2020-08-10 13:10:06 INFO      learning_rate - 0.001
2020-08-10 13:10:06 INFO      seq_length - 30
2020-08-10 13:10:06 INFO      downsample - 6
2020-08-10 13:10:06 INFO      in_out_ratio - 5
2020-08-10 13:10:06 INFO      stride - 30
2020-08-10 13:10:06 INFO      num_epochs - 1
2020-08-10 13:10:06 INFO      hidden_size - 512
2020-08-10 13:10:06 INFO      dropout - 0.0
2020-08-10 13:10:06 INFO      bidirectional - True
2020-08-10 13:10:06 INFO      attention - dot
2020-08-10 13:10:06 INFO      Starting seq2seq model training...
2020-08-10 13:10:06 INFO      Retrieving training data for sequences 125 ms long
and downsampling to 40.0 Hz...
2020-08-10 13:10:09 INFO      Data for training have shapes (X, y):
torch.Size([259570, 35]), torch.Size([51914, 92])
2020-08-10 13:10:09 INFO      Reshaped training shapes (X, y): torch.Size([51914,
5, 35]), torch.Size([51914, 1, 92])
2020-08-10 13:10:09 INFO      Number of training samples: 51914
2020-08-10 13:10:09 INFO      Retrieving validation data for sequences 125 ms
long and downsampling to 40.0 Hz...
2020-08-10 13:10:09 INFO      Data for validation have shapes (X, y):
torch.Size([90880, 35]), torch.Size([18176, 92])
2020-08-10 13:10:09 INFO      Reshaped validation shapes (X, y):
torch.Size([18176, 5, 35]), torch.Size([18176, 1, 92])
2020-08-10 13:10:09 INFO      Number of validation samples: 18176
2020-08-10 13:10:09 INFO      Encoder for training: EncoderRNN(
  (gru): GRU(35, 512, bidirectional=True)
  (dropout): Dropout(p=0.0, inplace=False)
  (fc): Linear(in_features=1024, out_features=512, bias=True)
)
2020-08-10 13:10:09 INFO      Decoder for training: AttnDecoderRNN(
  (attention): Attention()
  (rnn): GRU(1116, 512)
  (out): Linear(in_features=1628, out_features=92, bias=True)
)
2020-08-10 13:10:09 INFO      Number of parameters: 4864876
2020-08-10 13:10:09 INFO      Optimizers for training: AdamW (
Parameter Group 0
  amsgrad: False
  betas: (0.9, 0.999)
  eps: 1e-08
  initial_lr: 0.001

```

```

lr: 0.001
weight_decay: 0.05
)
2020-08-10 13:10:09 INFO      Criterion for training: L1Loss()
2020-08-10 13:10:09 INFO      Epoch 1 / 1
2020-08-10 13:10:09 INFO      Total time elapsed: 0.2845275402069092 - Batch
Number: 0 / 1622 - Training loss: 0.5867175199891831
2020-08-10 13:10:28 INFO      Total time elapsed: 19.075539588928223 - Batch
Number: 162 / 1622 - Training loss: 0.06888134323321701
2020-08-10 13:10:46 INFO      Total time elapsed: 36.41547679901123 - Batch
Number: 324 / 1622 - Training loss: 0.06516135748519093
2020-08-10 13:11:06 INFO      Total time elapsed: 56.93833088874817 - Batch
Number: 486 / 1622 - Training loss: 0.06124250432828066
2020-08-10 13:11:27 INFO      Total time elapsed: 77.28701090812683 - Batch
Number: 648 / 1622 - Training loss: 0.06000007542016456
2020-08-10 13:11:49 INFO      Total time elapsed: 99.8214840888977 - Batch
Number: 810 / 1622 - Training loss: 0.06179196632301481
2020-08-10 13:12:13 INFO      Total time elapsed: 123.77892279624939 - Batch
Number: 972 / 1622 - Training loss: 0.055964607362772964
2020-08-10 13:12:34 INFO      Total time elapsed: 144.65489411354065 - Batch
Number: 1134 / 1622 - Training loss: 0.05693873948561733
2020-08-10 13:12:58 INFO      Total time elapsed: 167.9723401069641 - Batch
Number: 1296 / 1622 - Training loss: 0.05655428921935604
2020-08-10 13:13:21 INFO      Total time elapsed: 191.32336831092834 - Batch
Number: 1458 / 1622 - Training loss: 0.05438031324405435
2020-08-10 13:13:44 INFO      Total time elapsed: 214.2629885673523 - Batch
Number: 1620 / 1622 - Training loss: 0.05223539072065468
2020-08-10 13:14:03 INFO      Training Loss: 0.06224823312900671 - Val Loss:
0.10371989231654884, 22.61474124110037
2020-08-10 13:14:03 INFO      Saving model to /home/jackg7/VT-Natural-Motion-
Processing/models/set-2/model.pt
2020-08-10 13:14:04 INFO      Completed Training...
2020-08-10 13:14:04 INFO

```

0.1.3 Testing our model

We can now test our model and output a histogram of performance over the testing data. The model parameters must be the same to properly read in the model.

```

[3]: !mkdir -p /home/jackg7/VT-Natural-Motion-Processing/images
!python test-seq2seq.py --task conversion \
--data-path-parent /home/jackg7/
→VT-Natural-Motion-Processing/data \
--figure-file-path /home/jackg7/
→VT-Natural-Motion-Processing/images/seq2seq-test.pdf \
--figure-title "Seq2Seq" \

```

```

--model-dir /home/jackg7/VT-Natural-Motion-Processing/
→models/set-2 \
--representation quaternions \
--batch-size=512 \
--seq-length=30 \
--downsample=6 \
--in-out-ratio=5 \
--stride=30 \
--hidden-size=512 \
--attention=dot \
--bidirectional

```

```

2020-08-10 13:14:06 INFO      task - conversion
2020-08-10 13:14:06 INFO      data_path_parent - /home/jackg7/VT-Natural-Motion-
Processing/data
2020-08-10 13:14:06 INFO      figure_file_path - /home/jackg7/VT-Natural-Motion-
Processing/images/seq2seq-test.pdf
2020-08-10 13:14:06 INFO      figure_title - Seq2Seq
2020-08-10 13:14:06 INFO      include_legend - False
2020-08-10 13:14:06 INFO      model_dir - /home/jackg7/VT-Natural-Motion-
Processing/models/set-2
2020-08-10 13:14:06 INFO      representation - quaternions
2020-08-10 13:14:06 INFO      batch_size - 512
2020-08-10 13:14:06 INFO      seq_length - 30
2020-08-10 13:14:06 INFO      downsample - 6
2020-08-10 13:14:06 INFO      in_out_ratio - 5
2020-08-10 13:14:06 INFO      stride - 30
2020-08-10 13:14:06 INFO      hidden_size - 512
2020-08-10 13:14:06 INFO      dropout - 0.0
2020-08-10 13:14:06 INFO      bidirectional - True
2020-08-10 13:14:06 INFO      attention - dot
2020-08-10 13:14:06 INFO      Starting seq2seq model testing...
2020-08-10 13:14:06 INFO      Retrieving testing data for sequences 125 ms long
and downsampling to 40.0 Hz...
2020-08-10 13:14:09 INFO      Data for testing have shapes (X, y):
torch.Size([452760, 35]), torch.Size([90552, 92])
2020-08-10 13:14:09 INFO      Reshaped testing shapes (X, y): torch.Size([90552,
5, 35]), torch.Size([90552, 1, 92])
2020-08-10 13:14:09 INFO      Number of testing samples: 90552
2020-08-10 13:14:37 INFO      Inference Loss for set-2: 14.272361435695645
2020-08-10 13:14:37 INFO      Completed Testing...
2020-08-10 13:14:37 INFO

```

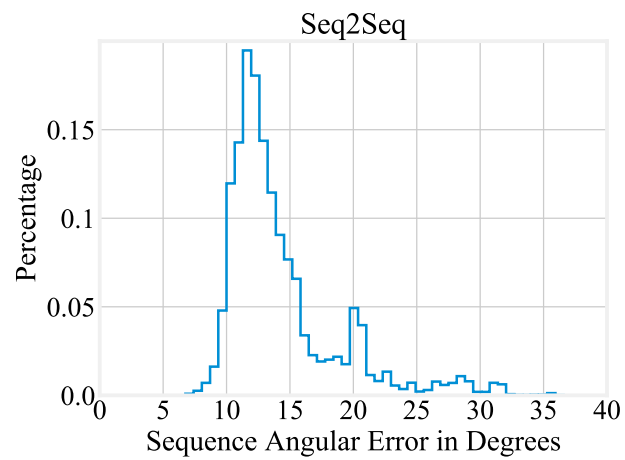
We can now visualize the performance of the seq2seq model on the test data.

```

[4]: from IPython.display import IFrame
IFrame("../images/seq2seq-test.pdf", width=600, height=300)

```

[4]: <IPython.lib.display.IFrame at 0x7f48ec1c3e50>



[]: