Overview

This project attempts motion-based handwriting recognition to explore an alternative to a vision-based approach, for various use cases where there is no convenient surface to write on (e.g. VR).

For this project, we build the collection hardware with Arduino and motion sensor, collect our own original dataset, then use various techniques to process data and perform classification.

The inputs of all of our models are the yaw, pitch roll data for each time stamp, and the outputs are the predictions of the corresponding letter for each writing event.

Data

We have collected 10,000+ data sequences from 20+ subjects, and preprocessed the gyroscope and accelerometer data.

Feature Interpolation & Re-sampling

Raw sequence of variable length

<table>
<thead>
<tr>
<th>t=0</th>
<th>t=14</th>
<th>t=27</th>
<th>t=43</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>80</td>
<td>70</td>
<td>90</td>
</tr>
</tbody>
</table>

Feature vector of fixed length

| 180 | 86.7 | 82.1 | 73.8 |

Since the sensor samples at a variable rate, and people write in different speed, we interpolate each sequence based on time delta, to a fixed length for models that require a fixed-sized input.

Denosing Autoencoder

We explore autoencoder to remove noise from the raw data. After efficiently learned the latent representation of the data, the model is used to preprocess each input for the classifiers before training.

Data Augmentation

Due to the difficulty to obtain a large dataset, we augment our dataset by: 1. stretching by a scalar for all timestamps; 2. adding a Gaussian noise to each timestamp; 3. rotating by a small quaternion, to generate more training samples for our models.

Neural Networks

We implement two neural network based models for this task.

- **CNN**: We flatten the data to a fixed length input for CNN model. We consider each feature component as an individual channel, and construct layers of 1-D convolution, with activation and pooling to construct a network.
- **RNN with LSTM**: RNN has the inherent structure for sequential data. We use a 5 layer LSTM RNN to process the inputs and a fully-connected layer applied on the last LSTM output for classification.

We use cross-entropy loss to train both models.

Conclusion and Discussion

- **RNN** models achieve the best performance among all the algorithms we have experimented with.
- Data augmentation strategy is able to enhance the performance of all deep models.
- Denosing autoencoder is effective for handle partially destructed inputs.
- All models suffer from overfitting problem due to the lack of generality of the data set.

Future Work

- Collect more data from different subjects and include more diverse demographic groups.
- Incorporate unused attributes collected.
- Develop a framework for handwriting word prediction based on the current models.