Learning and Predicting Animations from Crowdsourcing

Student: Anil Poriya | arp6763@rit.edu
Advisor: Dr. Christopher Homan | cmh@cs.rit.edu

Motivation

- Can we learn complex, collaborative behaviors by observing crowdsourced examples?
- Apparition: wizard-of-Oz crowdsourcing platform for rapid UI prototyping.
- Engaging remote workers to collaborate and complete tasks is repetitive and expensive.

Challenge

- **Goal**: develop a long short term memory (LSTM) recurrent neural network that can predict UI animations from crowdsourced examples.
- **Key challenge**: predict stopping point.

Data

<table>
<thead>
<tr>
<th>Geometric Features</th>
<th>Object Shape Features</th>
<th>Movement Type Features</th>
<th>Number of Object Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>line</td>
<td>move</td>
<td>20 features to indicate number of elements in the animation</td>
</tr>
<tr>
<td>y</td>
<td>circle</td>
<td>matrix</td>
<td></td>
</tr>
<tr>
<td>width</td>
<td>triangle</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>height</td>
<td>square</td>
<td>restore</td>
<td></td>
</tr>
<tr>
<td>rotationAngle</td>
<td>ellipse</td>
<td>translate</td>
<td></td>
</tr>
<tr>
<td>time</td>
<td>pentagon</td>
<td>transpose</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hexagon</td>
<td>end</td>
<td></td>
</tr>
<tr>
<td></td>
<td>octagon</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rect</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rombus</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>path</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Workflow

- Extract animation from Apparition
- Upload animation to Apparition
- Upload synthetic animation data to MongoDB
- Feed extracted data to learning model
- PAM: Predicted Animation Movement
- Encoder-Decoder Model

Results

- **Graph 1**: Input sequence length (SL) vs. Step numbers for "end"
- **Graph 2**: Test Data 100 epochs
- **Graph 3**: Multiple Train Single Test 50 epochs
- **Graph 4**: Multiple Train Single Test 100 epochs

**Graph 1**

- **x-axis**: Input sequence length (SL)
- **y-axis**: Step numbers for "end"

**Graph 2**

- **x-axis**: SL
- **y-axis**: Step numbers for "end"

**Graph 3**

- **x-axis**: Test Data 100 epochs
- **y-axis**: Step numbers for "end"

**Graph 4**

- **x-axis**: Test Data 100 epochs
- **y-axis**: Step numbers for "end"

Experiments

- Generate synthetic training data (40 animation steps/item)
- Test on an animation prefix
- **Var**: x and y coordinates, input sequence lengths, number of epochs

Conclusion

- Successfully built an encoder-decoder model in Keras that could learn and predict the "end" feature for animations having around 50 steps.
- Predicted animation movements differ considerably from the training animations.