Predicting

Given a dataset of $x, y$ coordinates of human written text on a digital surface, our model generates any given sentence in a handwriting style close to a human. We can choose between a specific style to be copied or picking a random style.

We built a model using LSTM with Mixture density networks and attention mechanism to achieve this.

Results

```
you know nothing.
now you doing
oh my god.

happy go lucky.
```

<table>
<thead>
<tr>
<th>Batch Size</th>
<th>#Minibatches</th>
<th>Dropout Rate</th>
<th>#Hidden Layers</th>
<th>Train Loss</th>
<th>Valid Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>15</td>
<td>0.98</td>
<td>3</td>
<td>-1.23</td>
<td>-2.36</td>
</tr>
<tr>
<td>128</td>
<td>48</td>
<td>0.95</td>
<td>3</td>
<td>-2.56</td>
<td>-2.34</td>
</tr>
<tr>
<td>128</td>
<td>48</td>
<td>0.95</td>
<td>2</td>
<td>-2.51</td>
<td>-2.28</td>
</tr>
</tbody>
</table>

Data and Data processing

We used IAM Online handwriting database which contains forms of handwritten English text acquired on a whiteboard. The collected data is stored in xml format. Apart from other data, each XML file contains a Strokeset, which contains a Stroke with its color and time information and a Point with $x, y$ and time value.

Sample writing:

```
most his family
```

Models

Loss equation:

$$ L(x) = \sum_{t=1}^{T} - \log \left( \frac{1}{|S|} \sum_{s \in S} \mathcal{N}(x_t | \mu_s, \sigma_s^2) \right) $$

Attention mechanism equations:

$$ a_t \propto \sum_{t=1}^{T} \exp(-E(x_t, u)) $$

$$ a_t = \sum_{t=1}^{T} a_t \cdot x_t $$

$$ x_t = \text{softmax}(a_t) $$

Features

- Picked collection of tstep points as lines.
- Removed lines with less than tstep points.
- Each point is composition of $x, y$ coordinate and end-of-stroke boolean value.
- Took offset of points from whiteboard corner coordinates.
- Scaled point values with scale factor.

Future

- Train the model to generate text from any language, even symbols.
- Tweaks the model to generate sensible next words.
- Improve the model to perform better for deep-in-time scenarios. Example: generating a very long sentence.
- Tackle the exploding gradient problem better.

Discussion

- RNN with LSTM can learn and generate complex long-range structures using next-step prediction.
- Exploding gradients is a prominent problem while training RNNs. Exponential learning rate decay and regularization are useful to tackle this.
- Interestingly, we observed the results didn't change much even after using 2 LSTM layers instead of 3, which reduced the number of variables greatly.
- The model shows potential to generate sensible next characters if it is forced to generate beyond the specified sentence.

References

2. Realistic handwriting with tensor flow by Sam Greydanus
3. Resources on Coursera and Youtube