Dropout in Recurrent Neural Networks
A Theoretically Grounded Dropout Variant in RNNs using Variational Inference
Yarin Gal yg279@cam.ac.uk

**Existing dropout in RNNs**

Let’s use dropout then. But lots of research has claimed that that’s a bad idea:

- Pachitariu & Sahani, 2013
  - Noise added in the recurrent connections of an RNN leads to model instabilities
- Bayer et al., 2013
  - With dropout, the RNN’s dynamics change dramatically
- Pham et al., 2014
  - Dropout in recurrent layers disrupts the RNN’s ability to model sequences
- Bluche et al., 2015
  - Exploratory analysis of the performance of dropout before, inside, and after the RNN’s
- Moon et al., 2015
  - Drop elements in the LSTM’s cell using the same mask at every time step.

Many settled on using dropout for inputs and outputs alone.

**Why does it make sense?**

- **Input**: sequence of vectors \( x = \{x_1, \ldots, x_T\} \) with \( T \) time steps
- Let \( \omega = (\text{all model weight matrices}) \) and put prior \( p(\omega) \) (e.g. standard Gaussian)
- Define \( h_t = f_t^\omega (x_t, h_{t-1}) \)
  - Single recurrent unit transition. E.g. \( \tanh(W x_t + U h_{t-1} + b) \) (similarly for LSTM, GRU)
- Set \( F_t^\omega (x_t) = f_t^\omega (h_t) \)
  - Model output (e.g. affine transformation of last state, or function of all states)
- Lastly, define \( p(y|F_t^\omega (x_t)) \)
  - Model likelihood on random function output \( F_t^\omega (x_t) \). E.g. \( \mathcal{N}(y; \mu(F_t^\omega (x_t)), \sigma^2) \)

- **Variational interpretation of dropout** [Gal and Ghahramani, 2015]:
  Dropout objective minimises
  \[
  KL(q(\omega)||p(\omega|X, Y)) \propto -\int q(\omega) \log p(Y|X, \omega) d\omega + KL(q(\omega)||p(\omega))
  \]
  \[
  = -\sum_{t=1}^{T} \int q(\omega) \log p(y_t|F_t^\omega (x_t)) d\omega + KL(q(\omega)||p(\omega)).
  \]
  with \( q(\omega) \) factorising over weight columns \( w_{a, t} \), e.g. \( q(w_{a, t}) = \rho(\delta_{a, t} + (1-\rho)\delta_{a, t}) \)
- But
  \[
  \int q(\omega) \log p(y_t|F_t^\omega (x_t)) d\omega = \int q(\omega) \log p \left( y_t \bigg| \underbrace{f_t^\omega (x_t), \ldots, f_t^\omega (x_t)}_{\text{all units}} \right) d\omega,
  \]
  \[
  = \sum_{t=1}^{T} \log p \left( y_t \bigg| \underbrace{f_t^\omega (x_t), \ldots, f_t^\omega (x_t)}_{\text{all units}} \right) + KL(q(\omega)||p(\omega)).
  \]
  So using MC integration with \( q(\omega) 
  \]
- Using random mask to set weight columns to zero (dropping units), repeating the same mask at each time step for all weight matrices (including embedding layer)

**Results**

- **Penn Treebank language modelling**
<table>
<thead>
<tr>
<th></th>
<th>Medium LSTM</th>
<th>Medium WPS</th>
<th>Large LSTM</th>
<th>Large WPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>Test</td>
<td>Validation</td>
<td>Test</td>
<td>Validation</td>
</tr>
<tr>
<td>Non-regularized</td>
<td>Early stopping</td>
<td>121.1</td>
<td>121.7</td>
<td>128.5</td>
</tr>
<tr>
<td>Moon et al. [2015] +emb dropout</td>
<td>109.7</td>
<td>97.6</td>
<td>4.9K</td>
<td>122.9</td>
</tr>
<tr>
<td>Moon et al. [2015]</td>
<td>88.9</td>
<td>86.7</td>
<td>4.9K</td>
<td>88.8</td>
</tr>
<tr>
<td>Zaremba et al. [2014]</td>
<td>86.2</td>
<td>82.7</td>
<td>5.5K</td>
<td>82.2</td>
</tr>
<tr>
<td>Variational (tied weights)</td>
<td>81.8±0.2</td>
<td>79.7±0.1</td>
<td>1.7K</td>
<td>72.3±0.2</td>
</tr>
<tr>
<td>Variational (tied weights, MC)</td>
<td>79.0±0.1</td>
<td>–</td>
<td>–</td>
<td>74.1±0.0</td>
</tr>
<tr>
<td>Variational (untied weights)</td>
<td>81.9±0.2</td>
<td>79.7±0.1</td>
<td>2.1K</td>
<td>77.9±0.3</td>
</tr>
<tr>
<td>Variational (untied weights, MC)</td>
<td>78.6±0.1</td>
<td>–</td>
<td>–</td>
<td>73.4±0.0</td>
</tr>
</tbody>
</table>

Single model perplexity (on test and validation sets). Two LSTM sizes are compared using Zaremba, Sutskever, and Vinyals [2014]’s setup.

- **Sentiment analysis** (raw Cornell film reviews corpus, Pang and Lee [2005])

  LSTM test error
  GRU test error