Adversarial examples and (Virtual) Adversarial training

- **Adversarial examples** exist in the state of the art models on image domain.
- Training models to be robust to the adversarial examples (Adversarial training) improves generalization performance (Goodfellow et al 2014, Miyato et al 2016).
- **Virtual Adversarial Training** (Miyato et al, 2016) can be applied to semi-supervised learning tasks and achieves good performances on image classification tasks.

Adversarial training on text

- Abundant unlabeled examples in text domain.
- Training on text can take a very long time (e.g. recurrent models).
- (Virtual) Adversarial training requires little tuning of hyperparameters.
- In our work, we applied adversarial and virtual adversarial training to semi-supervised text classification.
- Achieved state of the art performance.

Model

We use a simple LSTM model for text classification and define adversarial perturbations on its word embeddings, instead of the sequence of words.

- A sequence of T words: \( \{e(t)\} = 1, \ldots, T \)
- Label: \( y \)
- Embedding matrix: \( V \in \mathbb{R}^{K \times t} \)
- Normalized embeddings:

\[
\tilde{v}_n = \frac{v_n}{\sqrt{\sum_n (v_n^T v_n)}}
\]

Proposed (Virtual) adversarial loss on the text classification model

Just add the below losses to the neg. log-likelihood!

- **Adversarial loss**

\[
L_{adv}(\theta) = -\frac{1}{N} \sum_{n=1}^{N} \log p(y_n | s_n + r_{adv}, \theta)
\]

where

\[
r_{adv} = -\epsilon \frac{1}{|g|} g |g| \quad \text{where} \quad g = \nabla_s \log p(y | s, \theta)
\]

- **Virtual Adversarial loss**

\[
L_{v-adv}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{KL} \left[ p(\cdot | s_n, \tilde{v}_n) || \frac{1}{N} \sum_{n=1}^{N} p(\cdot | s_n' + r_{v-adv}, \tilde{v}_n') \right]
\]

where

\[
r_{v-adv} = \epsilon \frac{1}{|g|} g |g| \quad \text{where} \quad g = \nabla_s \log p(y | s, \theta)
\]

Experiments

- We first do pretraining following Dai and Le [2015] (recurrent language model).
  - This procedure is important to (virtual) adversarial training, because the perturbations on the initialized embeddings can be interpreted as the perturbations of “semantics”.
- **Dataset (4 semi-supervised and 1 supervised dataset):**

<table>
<thead>
<tr>
<th>Classes</th>
<th>Train</th>
<th>Test</th>
<th>Unlabled</th>
<th>( \text{Avg. T} )</th>
<th>Max ( T )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMDB</td>
<td>2</td>
<td>25,000</td>
<td>25,000</td>
<td>50,000</td>
<td>239, 2,506</td>
</tr>
<tr>
<td>Elec</td>
<td>2</td>
<td>24,792</td>
<td>24,897</td>
<td>197,025</td>
<td>110, 5,123</td>
</tr>
<tr>
<td>Rotten Tomatoes</td>
<td>2</td>
<td>9,596</td>
<td>10,664</td>
<td>7,911,684</td>
<td>20, 54</td>
</tr>
<tr>
<td>DBpedia</td>
<td>14</td>
<td>5,600,000</td>
<td>70,000</td>
<td>49</td>
<td>953</td>
</tr>
<tr>
<td>RCV1</td>
<td>55</td>
<td>15,564</td>
<td>49,858</td>
<td>668,640</td>
<td>153, 9,852</td>
</tr>
</tbody>
</table>

- We optimized dropout rate on embeddings and norm constraint \( \epsilon \) on adversarial and virtual adversarial training with each validation set.
- We used the method with only embedding dropout as the baseline.

Results on IMDB

- Virtual adversarial training achieved comparable performance with the state of the art semi-supervised method.

On other semi-supervised datasets (Elec, RCV1 and Rotten Tomatoes)

- Our proposed method achieved state of the art performance on both datasets.
  - However, The performance with virtual adversarial training was worse than the baseline on Rotten Tomatoes.
  - We speculate that Virtual adversarial loss on unlabeled examples would overwhelm the supervised loss, and this would cause "wrong label" propagation.

On supervised datasets (DBpedia)

- Achieved state of the art performance with virtual adversarial training.

Conclusion

- Adversarial and virtual adversarial training are good regularizers for text classification tasks and achieved state of the art performance.
- With tuning of the additional hyperparameter \( \epsilon \) we can improve over the baseline and achieve state of the art performance.

References

  - ICLR 2015.
  - ICLR 2016.
- Semi-supervised Sequence Learning, A. M. Dai and Q. V. Le.
  - NeurIPS 2015.