Harnessing Deep NNs with Logic Rules

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, Eric Xing
School of Computer Science
Carnegie Mellon University
Deep NNs
Deep NNs

• heavily rely on massive labeled data
• uninterpretable
• hard to encode human intention/domain knowledge
How humans learn

• learn from *concrete* examples (as DNNs do)
• learn from *general* knowledge and rich experiences
  [Minsky 1980; Lake et al., 2015]
  • the past tense of verbs\(^1\):
    • regular verbs –d/-ed

\(^1\) [https://www.technologyreview.com/s/544606/can-this-man-make-aimore-human]
DNNs + knowledge
DNNs + knowledge

• logic rule
  • a flexible declarative language
  • express structured knowledge
DNNs + knowledge

• logic rule
  • a flexible declarative language
  • express structured knowledge

• DNNs + logic rules
Related work

- **neural-symbolic system** [Garcez et al., 2012]
  - specialized NNs from a rule set to execute reasoning

- **learning interpretable hidden layer** [Kulkarni et al., 2011; Karaletsos et al., 2016]
  - specialized types of knowledge (e.g., similarity tuples)

- **posterior regularization on latent variable models** [Ganchev et al., 2010; Liang et al., 2009; Zhu et al., 2014]
  - not directly applicable to NNs
  - or poor performance

- **structure compilation/knowledge distillation** [Liang et al., 2008; Hinton et al., 2015; Bucilu et al., 2006]
  - pipelined method with CRF/NN ensembles
This work

• enhances *general* types of NNs
• *with general* types of knowledge expressed as logic rules
This work

• enhances *general* types of NNs
• *with general* types of knowledge expressed as logic rules

• *iterative rule knowledge distillation*
  • transfers rule knowledge into NNs
  • generality
    • CNN for sentiment classification
    • RNN for named entity recognition
Rule formulation

• input-target space: \((X, Y)\)
• first-order logic (FOL) rules: \((r, \lambda)\)
  • \(r(X, Y) \in [0,1]\)
  • soft logic
    • e.g., \(A \& B := \max\{A + B - 1, 0\}\)
    • takes values \(\in [0,1]\)
  • \(\lambda\): confidence level of the rule
Rule knowledge distillation

• neural network $p_{\theta}(y|x)$

at iteration $t$:

$$\theta^{(t+1)} = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, \sigma_\theta(x_n))$$

true hard label

soft prediction of $p_\theta$
Rule knowledge distillation

- neural network $p_{\theta}(y|x)$
- train to imitate the outputs of a rule-regularized teacher network (i.e. distillation)

at iteration $t$:

$\theta^{(t+1)} = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, \sigma_{\theta}(x_n))$

true hard label

soft prediction of $p_{\theta}$

soft prediction of the teacher network
Rule knowledge distillation

- neural network $p_\theta(y|x)$
- train to imitate the outputs of a rule-regularized teacher network (i.e. distillation)

Method

\[ \theta(t+1) = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} (1 - \pi) \ell(y_n, \sigma_\theta(x_n)) \]

at iteration $t$:

- true hard label
- soft prediction of $p_\theta$
- balancing parameter
- soft prediction of the teacher network
Teacher network construction

• teacher network: \( q(Y | X) \)
  • comes out of \( p \)
  • fits the logic rules: \( E_q[r(X, Y)] = 1 \), with confidence \( \lambda \)
Teacher network construction

• teacher network: $q(Y|X)$
  • comes out of $p$
  • fits the logic rules: $E_q[r(X,Y)] = 1$, with confidence $\lambda$

$$\min_{q, \xi \geq 0} \text{KL}(q||p_\theta(Y|X)) + C \sum_l \xi_l$$

s.t. $\lambda_l (1 - \mathbb{E}_q[r_l(X,Y)]) \leq \xi_l$

$l = 1, \ldots, L$
Teacher network construction

- teacher network: $q(Y|X)$
  - comes out of $p$
  - fits the logic rules: $E_q[r(X,Y)] = 1$, with confidence $\lambda$

$$\min_{q, \xi \geq 0} \text{KL}(q||p_\theta(Y|X)) + C \sum_l \xi_l$$

s.t. $\lambda_l (1 - \mathbb{E}_q[r_l(X,Y)]) \leq \xi_l$

$l = 1, \ldots, L$

closed-form solution:

$$q^*(Y|X) \propto p_\theta(Y|X) \exp \left\{ - \sum_l C\lambda_l (1 - r_l(X,Y)) \right\}$$
Method summary

- at each iteration
  - construct a teacher network through posterior constraints
  - train the NN to emulate the predictions of the teacher
Method summary

• at *test* time, can use either the distilled network $p$, or the teacher network $q$
• both improve over the base NN significantly
• $q$ generally performs better than $p$
• $p$ is more light-weight
  • no explicit rule expression
  • e.g., rule assessment is expensive/unavailable at test time
Sentiment classification

- sentence -> positive/negative
- base network: CNN [Kim, 2014]
Rule knowledge

• identify contrastive sense
  • capture the dominant sentiment

• conjunction word ``but’’
  • sentence $S$ with structure $A$-but-$B$:
    $\Rightarrow$ sentiment of $B$ dominates

\[
\text{has-‘A-but-B’-structure}(S) \Rightarrow \\
(1(y = +) \Rightarrow \sigma_\theta(B)_+ \land \sigma_\theta(B)_+ \Rightarrow 1(y = +))
\]
## Results

- accuracy (%)

<table>
<thead>
<tr>
<th>Model</th>
<th>SST2</th>
<th>MR</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CNN (Kim, 2014)</td>
<td>87.2</td>
<td>81.3±0.1</td>
<td>84.3±0.2</td>
</tr>
<tr>
<td>2 CNN-Rule-(p)</td>
<td>88.8</td>
<td>81.6±0.1</td>
<td>85.0±0.3</td>
</tr>
<tr>
<td>3 CNN-Rule-(q)</td>
<td>89.3</td>
<td>81.7±0.1</td>
<td>85.3±0.3</td>
</tr>
<tr>
<td>4 MGNC-CNN (Zhang et al., 2016)</td>
<td>88.4</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>5 MVCNN (Yin and Schutze, 2015)</td>
<td>89.4</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>6 CNN-multichannel (Kim, 2014)</td>
<td>88.1</td>
<td>81.1</td>
<td>85.0</td>
</tr>
<tr>
<td>7 Paragraph-Vec (Le and Mikolov, 2014)</td>
<td>87.8</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>8 CRF-PR (Yang and Cardie, 2014)</td>
<td>–</td>
<td>–</td>
<td>82.7</td>
</tr>
<tr>
<td>9 RNTN (Socher et al., 2013)</td>
<td>85.4</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>10 G-Dropout (Wang and Manning, 2013)</td>
<td>–</td>
<td>79.0</td>
<td>82.1</td>
</tr>
</tbody>
</table>
Comparisons to other rule integration methods

• SST2 dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN (Kim, 2014)</td>
<td>87.2</td>
</tr>
<tr>
<td>-but-clause</td>
<td>87.3</td>
</tr>
<tr>
<td>-$\ell_2$-reg</td>
<td>87.5</td>
</tr>
<tr>
<td>-project</td>
<td>87.9</td>
</tr>
<tr>
<td>-opt-project</td>
<td>88.3</td>
</tr>
<tr>
<td>-pipeline</td>
<td>87.9</td>
</tr>
<tr>
<td>-Rule-(p)</td>
<td>88.8</td>
</tr>
<tr>
<td>-Rule-(q)</td>
<td><strong>89.3</strong></td>
</tr>
</tbody>
</table>
Data size, semi-supervision

- SST2 dataset

<table>
<thead>
<tr>
<th>Data size</th>
<th>5%</th>
<th>10%</th>
<th>30%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CNN</td>
<td>79.9</td>
<td>81.6</td>
<td>83.6</td>
<td>87.2</td>
</tr>
<tr>
<td>2 -Rule-ρ</td>
<td>81.5</td>
<td>83.2</td>
<td>84.5</td>
<td>88.8</td>
</tr>
<tr>
<td>3 -Rule-𝑞</td>
<td>82.5</td>
<td>83.9</td>
<td>85.6</td>
<td>89.3</td>
</tr>
<tr>
<td>4 -semi-PR</td>
<td>81.5</td>
<td>83.1</td>
<td>84.6</td>
<td>–</td>
</tr>
<tr>
<td>5 -semi-Rule-ρ</td>
<td>81.7</td>
<td>83.3</td>
<td>84.7</td>
<td>–</td>
</tr>
<tr>
<td>6 -semi-Rule-𝑞</td>
<td>82.7</td>
<td>84.2</td>
<td>85.7</td>
<td>–</td>
</tr>
</tbody>
</table>
Named entity recognition (NER)

- to locate and classify words into entity categories
  - Persons/Organizations/Locations/...
- assigns to each word a named entity tag:
  - B-PER: beginning of a person name
  - I-ORG: inside an organization name
- base NN: bidirectional LSTM RNN

[Chiu and Nichols, 2015]
Rule knowledge

• constraints on successive labels for a valid tag sequence
  • e.g., I-ORG cannot follow B-PER

• listing structure
  • “1. Juventus, 2. Barcelona, 3. ...”
  • “Juventus” is an organization, so “Barcelona” must be an organization, rather than a location
## Results

- **F1 score on CoNLL-2003 dataset**

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  BLSTM</td>
<td>89.55</td>
</tr>
<tr>
<td>2  BLSTM-Rule-trans</td>
<td>$p$: 89.80, $q$: 91.11</td>
</tr>
<tr>
<td>3  BLSTM-Rules</td>
<td>$p$: 89.93, $q$: <strong>91.18</strong></td>
</tr>
<tr>
<td>4  NN-lex (Collobert et al., 2011)</td>
<td>89.59</td>
</tr>
<tr>
<td>5  S-LSTM (Lample et al., 2016)</td>
<td>90.33</td>
</tr>
<tr>
<td>6  BLSTM-lex (Chiu and Nichols, 2015)</td>
<td>90.77</td>
</tr>
<tr>
<td>7  BLSTM-CRF$_1$ (Lample et al., 2016)</td>
<td>90.94</td>
</tr>
<tr>
<td>8  Joint-NER-EL (Luo et al., 2015)</td>
<td>91.20</td>
</tr>
<tr>
<td>9  BLSTM-CRF$_2$ (Ma and Hovy, 2016)</td>
<td><strong>91.21</strong></td>
</tr>
</tbody>
</table>
Conclusions

• iterative rule knowledge distillation
  • combines FOL rules with DNNs

• general applicability
  • CNNs/RNNs
  • knowledge expressed in FOL
  • tasks: sentiment analysis/NER
Future work

• human knowledge
  • abstract, fuzzy, built on high-level concepts
  • e.g., a *dog* has four *legs*
Future work

• human knowledge
  • abstract, fuzzy, built on high-level concepts
  • e.g., a dog has four legs

• DNN
  • end-to-end

→ dog
Future work

• human knowledge
  • abstract, fuzzy, built on high-level concepts
  • e.g., a dog has four legs

• DNN
  • end-to-end
Future work

• human knowledge
  • abstract, fuzzy, built on high-level concepts
  • e.g., a dog has four legs
• DNN
  • end-to-end

• learn modules for complete knowledge representation
  \[ r_\phi(X, Y) \]
Future work

• human knowledge
  • abstract, fuzzy, built on high-level concepts
  • e.g., a dog has four legs

• DNN
  • end-to-end

• learn modules for complete knowledge representation
  $r_{\phi}(X,Y)$

• learn knowledge confidence $\lambda$
References


[Ganchev et al., 2010] Kuzman Ganchev, Joao Grac,a, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. JMLR


[Zhu et al., 2014] Jun Zhu, Ning Chen, and Eric P Xing. 2014. Bayesian inference with posterior regularization and applications to infinite latent SVMs. JMLR

