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
  [Minksy 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, λ)
  - $r(X, Y) \in [0,1]$
  - soft logic
    - e.g., $A \,\&\, B \coloneqq \max\{A + B - 1, 0\}$
    - takes values $\in [0,1]$
  - λ: confidence level of the rule

Rule knowledge distillation

- neural network $p_\theta(y|x)$

at iteration t:

true hard label

soft prediction of p_θ

$$\boldsymbol{\theta}^{(t+1)} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(\boldsymbol{y}_n, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n))$$

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:

true hard label soft prediction of p_θ

$$\boldsymbol{\theta}^{(t+1)} = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} \qquad \ell(\boldsymbol{y}_n, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n))$$

$$\ell(\boldsymbol{s}_n^{(t)}, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n)),$$

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)

at iteration t:

true hard label

soft prediction of p_θ

$$\boldsymbol{\theta}^{(t+1)} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{n=1}^{N}(1-\pi)\ell(\boldsymbol{y}_n, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n))$$

$$+ \pi\ell(\boldsymbol{s}_n^{(t)}, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n)),$$

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 λ

Teacher network construction

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

slack variable

$$\min_{q,\boldsymbol{\xi}\geq 0} \mathrm{KL}(q \| p_\theta(\boldsymbol{Y}|\boldsymbol{X})) + C \sum\nolimits_l \xi_l$$

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

$$l = 1, \ldots, L$$

rule constraints

Teacher network construction

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

slack variable

$$\min_{q,\boldsymbol{\xi}\geq 0} \mathrm{KL}(q \| p_\theta(\boldsymbol{Y}|\boldsymbol{X})) + C\sum\nolimits_l \xi_l$$
$$\text{s.t. } \lambda_l(1 - \mathbb{E}_q[r_l(\boldsymbol{X},\boldsymbol{Y})]) \leq \xi_l$$
$$l = 1, \ldots, L$$

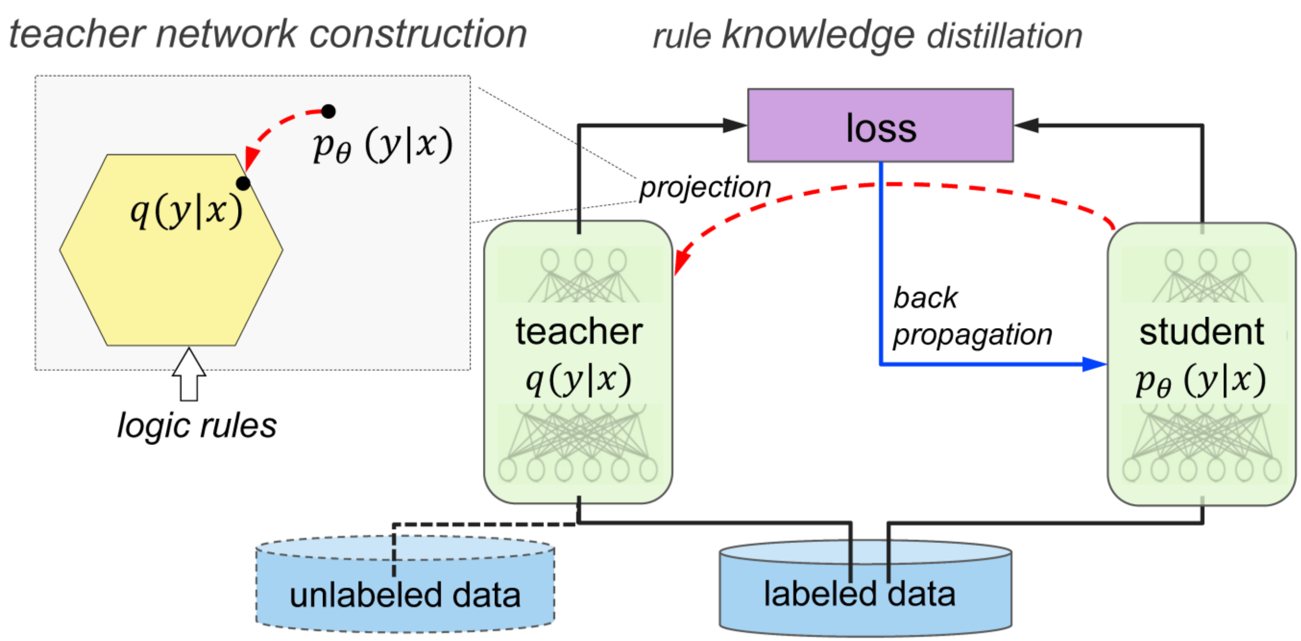
rule constraints

closed-form solution:

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

Method summary

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

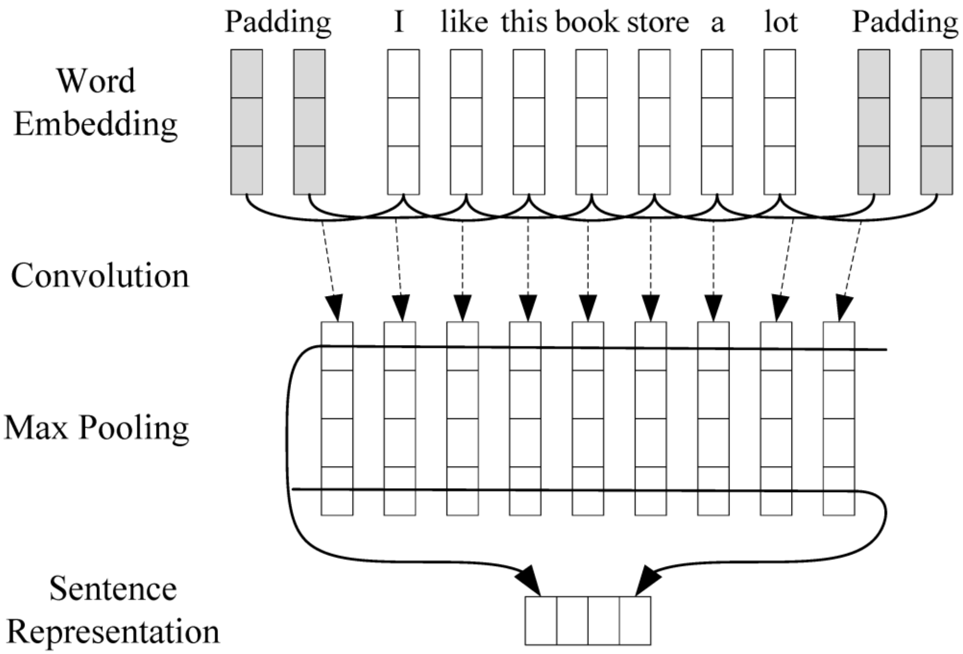


Method summary

- 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*:

    => sentiment of *B* dominates

$$\text{has-`A-but-B'-structure}(S) \Rightarrow$$
$$(\mathbf{1}(y=+) \Rightarrow \boldsymbol{\sigma}_\theta(B)_+ \;\wedge\; \boldsymbol{\sigma}_\theta(B)_+ \Rightarrow \mathbf{1}(y=+))$$

Results

- accuracy (%)

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

Comparisons to other rule integration methods

- SST2 dataset

	Model	Accuracy (%)
1	CNN (Kim, 2014)	87.2
2	-but-clause	87.3
3	-ℓ_2-reg	87.5
4	-project	87.9
5	-opt-project	88.3
6	-pipeline	87.9
7	-Rule-p	88.8
8	-Rule-q	**89.3**

Data size, semi-supervision

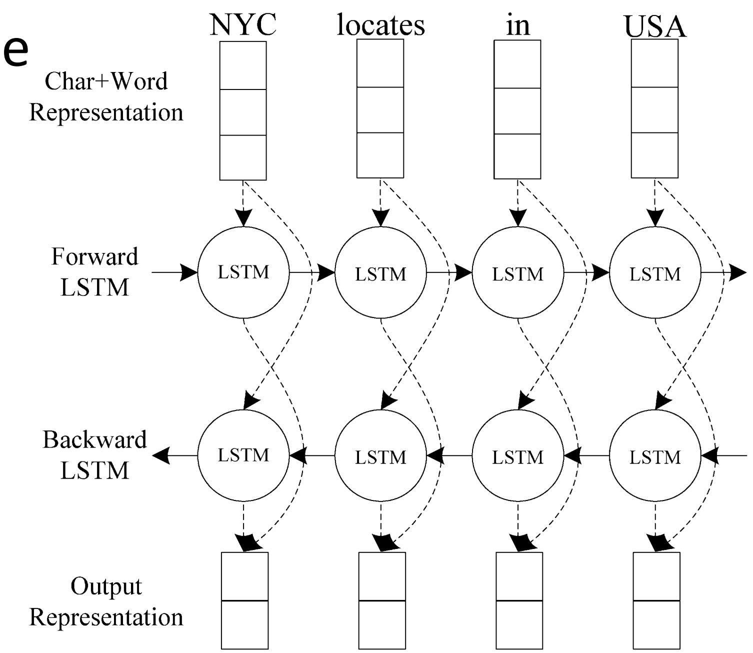
- SST2 dataset

	Data size	5%	10%	30%	100%
1	CNN	79.9	81.6	83.6	87.2
2	-Rule-p	81.5	83.2	84.5	88.8
3	-Rule-q	82.5	83.9	85.6	**89.3**
4	-semi-PR	81.5	83.1	84.6	–
5	-semi-Rule-p	81.7	83.3	84.7	–
6	-semi-Rule-q	**82.7**	**84.2**	**85.7**	–

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

	Model	F1
1	BLSTM	89.55
2	BLSTM-Rule-trans	p: 89.80, q: 91.11
3	BLSTM-Rules	p: 89.93, q: **91.18**
4	NN-lex (Collobert et al., 2011)	89.59
5	S-LSTM (Lample et al., 2016)	90.33
6	BLSTM-lex (Chiu and Nichols, 2015)	90.77
7	BLSTM-CRF_1 (Lample et al., 2016)	90.94
8	Joint-NER-EL (Luo et al., 2015)	91.20
9	BLSTM-CRF_2 (Ma and Hovy, 2016)	**91.21**

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*

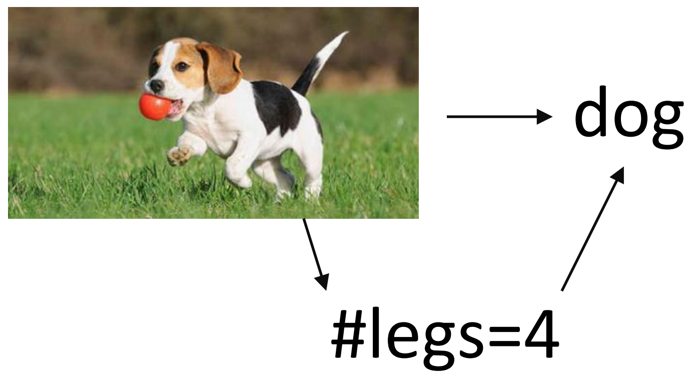
Future work

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

$\longrightarrow$ dog

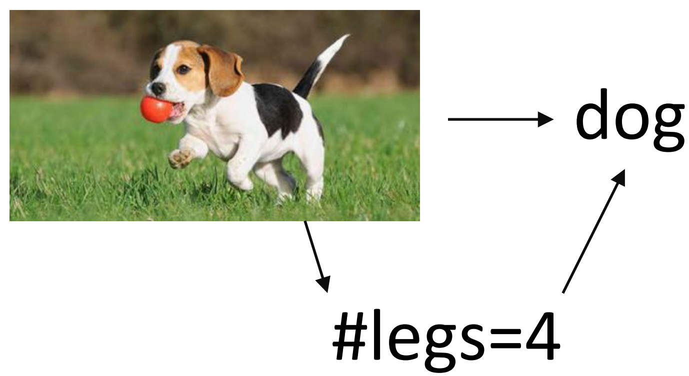
Future work

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



Future work

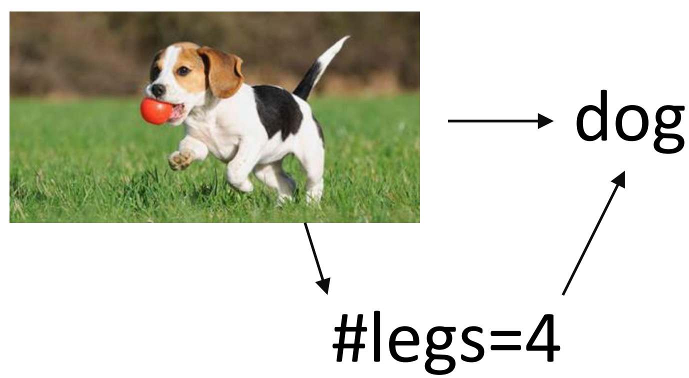
- human knowledge
  - abstract, fuzzy, built on high-level concepts
  - e.g., a *dog* has four *legs*
- DNN
  - end-to-end



- 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 λ

References

[Minksy, 1980] Marvin Minksy. 1980. Learning meaning. Technical Report AI Lab Memo.

[Lake et al., 2015] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. Science.

[Garcez et al., 2012] Artur S d'Avila Garcez, Krysia Broda, and Dov M Gabbay. 2012. Neural-symbolic learning systems: foundations and applications. Springer Science & Business Media

[Kulkarni et al., 2011] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. 2015. Deep convolutional inverse graphics network. NIPS.

[Karaletsos et al., 2016] Theofanis Karaletsos, Serge Belongie, Cornell Tech, and Gunnar R¨atsch. 2016. Bayesian representation learning with oracle constraints. ICLR

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

[Liang et al., 2009] Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning from measurements in exponential families. ICML.

[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

[Liang et al., 2008] Percy Liang, Hal Daum´e III, and Dan Klein. 2008. Structure compilation: trading structure for features. ICML

[Kim, 2014] Yoon Kim. 2014. Convolutional neural networks for sentence classification. EMNLP

[Chiu and Nichols, 2015] Jason PC Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional LSTM-CNNs. arXiv