Improving Joint Training of Inference Networks and Structured Prediction Energy Networks

Lifu Tu[1] Yuanzhe Pang[2] Kevin Gimpel[1]

[1]Toyota Technological Institute at Chicago

[2]New York University

4th Workshop on Structured Prediction for NLP, 2020

Table of Contents

[Finkel et al, 2005]

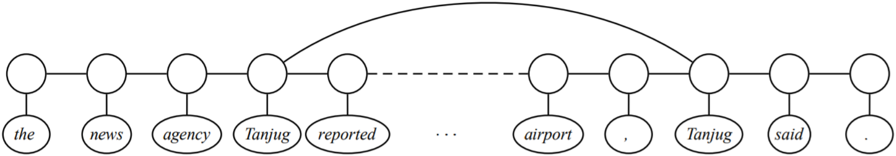


Figure: An example from CoNLL 2003 Named Entity Recognition

Enable label consistency

[Holtzman et al., 2020]

Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

Figure: Generated outputs from GPT-2 large language model.

Avoid repetition and incoherence

Motivation

Structured prediction:

- Capture Label dependency
- Avoid repetition in text generation
- ...

However,

- Hard to capture long dependency between structure outputs.
  - Energy-based models
- Inference for energy-based model is computational challenging!
  - Intractable
  - Exact inference/gradient descent inference is slow

Table of Contents

Energy Function and Inference Network

Definition

An energy function [LeCun et al., 2006; Belanger and McCallum, 2016] $E_\Theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ parametrized by Θ that uses a functional architecture to compute a scalar energy for an input/output pair.

Test-time inference

At test time, for a given input $\boldsymbol{x}$, prediction is done by choosing the output with the lowest energy.

$$\hat{\boldsymbol{y}} = \arg\min_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} E_\Theta(\boldsymbol{x}, \boldsymbol{y}).$$

Inference Networks [Tu and Gimpel, 2018]

A test-time inference network $\mathrm{A}_\Psi : \mathcal{X} \to \mathcal{Y}_R$ is parameterized by Ψ and trained with the goal that

$$\mathrm{A}_\Psi(\boldsymbol{x}) \approx \operatorname*{arg\,min}_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} E_\Theta(\boldsymbol{x}, \boldsymbol{y}).$$

Training Objectives

SPEN Loss

Belanger and McCallum (2016) use a structured hinge loss for training SPENs

$$\min_{\Theta} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} \left[\underbrace{\max_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} \left(\triangle(\boldsymbol{y}, \boldsymbol{y}_i) - E_{\Theta}(\boldsymbol{x}_i, \boldsymbol{y}) + E_{\Theta}(\boldsymbol{x}_i, \boldsymbol{y}_i) \right)}_{\text{cost-augmented inference}} \right]_{+}$$

Training Objectives

SPEN Loss

Belanger and McCallum (2016) use a structured hinge loss for training SPENs

$$\min_{\Theta} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} \left[\underbrace{\max_{\boldsymbol{y} \in \mathcal{Y}_R(\boldsymbol{x})} \left(\triangle(\boldsymbol{y}, \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i) \right)}_{\text{cost-augmented inference}} \right]_+$$

Our Objective [Tu and Gimpel, 2018]

We parametrize an inference network using Φ, alternately optimize Θ and Φ (like adversarial training):

$$\min_{\Theta} \max_{\Phi} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} [\triangle(\mathrm{F}_\Phi(\boldsymbol{x}_i), \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, \mathrm{F}_\Phi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)]_+$$

Pipeline

Recall: Θ is params for energy function, Φ for cost-augmented InfNet, Ψ for test-time InfNet.

Step 1: $\hat{\Theta}, \hat{\Phi} = \min_{\Theta} \max_{\Phi}$

$$\sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in \mathcal{D}} \left[\triangle(\mathrm{F}_\Phi(\boldsymbol{x}_i), \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, \mathrm{F}_\Phi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i) \right]_+$$

Update Φ to yield output with low energy and high cost.

Step 2: $\hat{\Psi} = \arg\min_\Psi E_\Theta(\boldsymbol{x}, \mathrm{A}_\Psi(\boldsymbol{x}))$ where A_Ψ is initialized by trained F_Φ.

Table of Contents

New Objective

Objective

$$\hat{\Theta}, \hat{\Phi} = \min_{\Theta} \max_{\Phi} \sum_{i} \underbrace{\left[\triangle(\mathrm{F}_{\Phi}(\boldsymbol{x}), \boldsymbol{y}_i) - E_{\Theta}(\boldsymbol{x}_i, \mathrm{F}_{\Phi}(\boldsymbol{x})) + E_{\Theta}(\boldsymbol{x}_i, \boldsymbol{y}_i)\right]_{+}}_{\text{margin-rescaled loss}}$$

Cost-augmented inference: $\mathrm{F}_{\Phi} \approx \arg\min_{\boldsymbol{y}'} (E_{\Theta}(\boldsymbol{x}, \boldsymbol{y}') - \triangle(\boldsymbol{y}', \boldsymbol{y}))$,

Test-time inference: $\mathrm{A}_{\Psi} \approx \arg\min_{\boldsymbol{y}'} E_{\Theta}(\boldsymbol{x}, \boldsymbol{y}')$.

New Objective

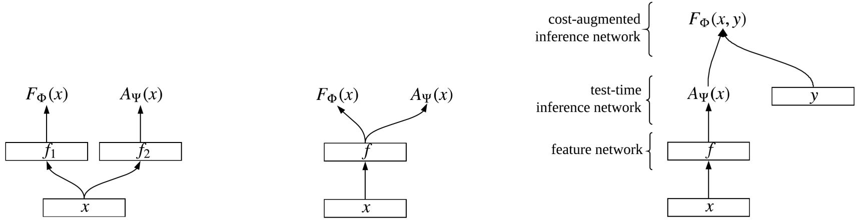
$$\hat{\Theta}, \hat{\Phi}, \hat{\Psi} = \min_{\Theta} \max_{\Phi,\Psi} \sum_i \underbrace{[\triangle(\mathrm{F}_\Phi(\boldsymbol{x}), \boldsymbol{y}_i) - E_\Theta(\boldsymbol{x}_i, \mathrm{F}_\Phi(\boldsymbol{x})) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)]_+}_{\text{margin-rescaled loss}}$$

$$+ \lambda \underbrace{[-E_\Theta(\boldsymbol{x}_i, \mathrm{A}_\Psi(\boldsymbol{x}_i)) + E_\Theta(\boldsymbol{x}_i, \boldsymbol{y}_i)]_+}_{\text{perceptron loss}}$$

Cost-augmented inference: $\mathrm{F}_\Phi \approx \arg\min_{\boldsymbol{y}'} (E_\Theta(\boldsymbol{x}, \boldsymbol{y}') - \triangle(\boldsymbol{y}', \boldsymbol{y}))$,

Test-time inference: $\mathrm{A}_\Psi \approx \arg\min_{\boldsymbol{y}'} E_\Theta(\boldsymbol{x}, \boldsymbol{y}')$.

Joint parametrization for cost-augmented (F_Φ) and test-time (A_Ψ) inference networks



(a) separated networks

(b) shared feature networks

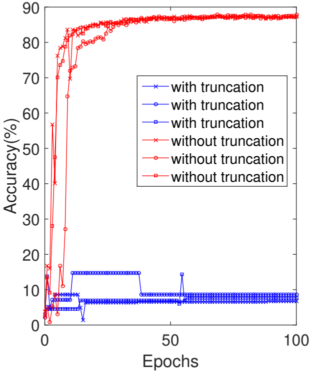
(c) stacked networks with y as extra input to F_Φ

Cost-augmented inference: $\mathrm{F}_\Phi \approx \arg\min_{\boldsymbol{y}'}(E_\Theta(\boldsymbol{x}, \boldsymbol{y}') - \triangle(\boldsymbol{y}', \boldsymbol{y}))$,

Test-time inference: $\mathrm{A}_\Psi \approx \arg\min_{\boldsymbol{y}'} E_\Theta(\boldsymbol{x}, \boldsymbol{y}')$.

Removing Zero Truncation

For inference network objective: $\max_\Psi [h_\Psi]_+ \rightarrow \max_\Psi h_\Psi$

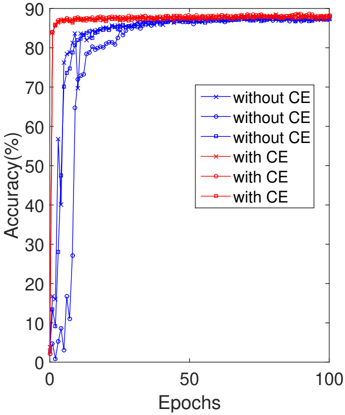


training trajectories without CE

Without truncation, the inference network can work well even without any stabilization terms.

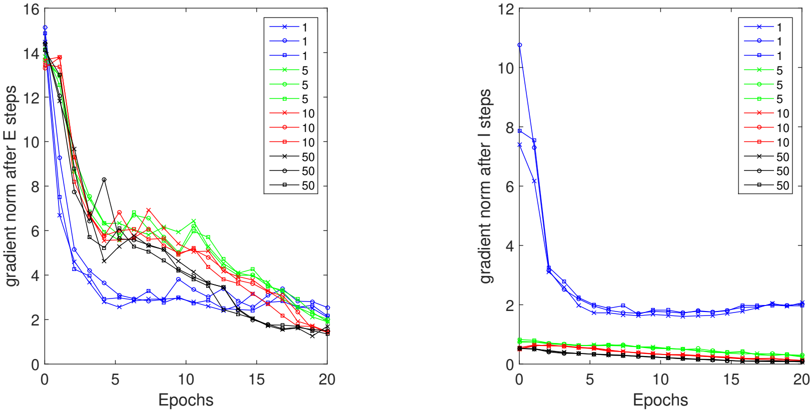
Local Cross Entropy Loss

Including $\sum_{t=1}^{|\boldsymbol{y}|} \mathrm{CE}(\boldsymbol{y}_t, \mathrm{A}(\boldsymbol{x})_t)$ or not when without truncation?



The local loss helps speed up convergence and improve accuracy

Multiple Inference Network Update Steps



Multiple steps of inner loop optimization help inference network maintain near its optimal solution for the given input.

Experimental Setup

- Energy function: BLSTM-CRF
- Inference network architecture: BLSTM
- Two sequence labeling tasks: Twitter POS tagging (POS) and CoNLL 2003 Named Entity Recognition (NER)

Comparison of Loss Functions

		POS	NER
baseline: margin-rescaled		89.3	85.2
	separated	89.4	85.0
new losses:	shared	85.6	85.6
	stacked	**89.8**	85.6

New losses outperform the baseline.

Comparing cost-augmented (F_Φ) and test-time (A_Ψ) inference networks

		POS	NER
		$A_\Psi - F_\Phi$	$A_\Psi - F_\Phi$
margin-rescaled		0.2	0
	separated	2.2	0.4
combined	shared	1.9	0.5
	stacked	**2.6**	**1.7**

Stacked parameterization shows largest difference between F_Φ and A_Ψ

Qualitative Analysis

test-time (A_Ψ)	cost-augmented (F_Φ)
common noun	proper noun
proper noun	common noun
common noun	adjective
proper noun	proper noun + possessive
adverb	adjective
preposition	adverb
adverb	preposition
verb	common noun
adjective	verb
common noun	verb

Table: Top 10 most frequent output differences between A_Ψ and F_Φ.

F_Φ tends to output tags that are highly confusable with those output by A_Ψ!

Table of Contents

Conclusions

- SPENs are powerful but learning and inference are hard (due to gradient descent for inference)
- Inference networks can make it easier and more efficient to use SPENs
- Separating inference networks for the two inference problems (cost-augmented and test-time inference) improves accuracy and leads to complementary functionality


Q&A