Semantic Probabilistic Layers for neuro-symbolic learning

Kareem Ahmed (he/him)

joint work with Stefano Teso, Kai-Wei Chang, Guy Van den Broeck and Antonio Vergari

Why?

"How can neural nets reason and learn with symbolic constraints reliably and efficiently?"

Why?

"How can neural nets reason and learn with symbolic constraints reliably and efficiently?"

integrate ***hard*** **(logical)** and ***soft*** constraints

Why?

"How can neural nets reason and learn with symbolic constraints reliably and efficiently?"

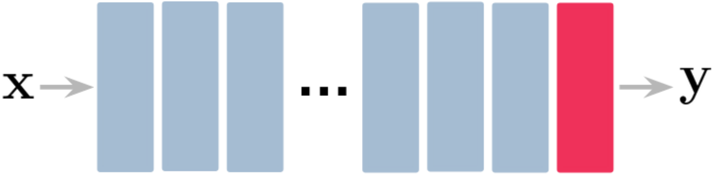
guarantee that predictions ***satisfy*** constraints

Why?

"How can neural nets reason and learn with symbolic constraints reliably and efficiently?"

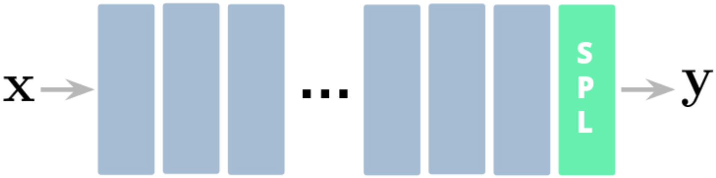
fast and ***exact*** gradients

Why?



make any neural network architecture...

Why?



...guarantee all predictions conform to constraints

When?

Ground Truth

e.g. predict shortest path in a map

When?

Ground Truth

given $\mathbf{x}$ *// e.g. a tile map*

NeSy structured output prediction (SOP) tasks

Vlastelica et al., "Differentiation of blackbox combinatorial solvers", ICLR, 2020

When?

Ground Truth

given $\mathbf{x}$ *// e.g. a tile map*

find $\mathbf{y}^* = \text{argmax}_{\mathbf{y}}\, p_\theta(\mathbf{y} \mid \mathbf{x})$ *// e.g. a configurations of edges in a grid*

NeSy structured output prediction (SOP) tasks

Vlastelica et al., "Differentiation of blackbox combinatorial solvers", ICLR, 2020

When?

Ground Truth

given $\mathbf{x}$ *// e.g. a tile map*

find $\mathbf{y}^* = \mathrm{argmax}_{\mathbf{y}}\, p_\theta(\mathbf{y} \mid \mathbf{x})$ *// e.g. a configurations of edges in a grid*

s.t. $\mathbf{y} \models \mathsf{K}$ *// e.g., that form a valid path*

NeSy structured output prediction (SOP) tasks

Vlastelica et al., "Differentiation of blackbox combinatorial solvers", ICLR, 2020

When?

Ground Truth

given $\mathbf{x}$ *// e.g. a tile map*

find $\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} p_\theta(\mathbf{y} \mid \mathbf{x})$ *// e.g. a configurations of edges in a grid*

s.t. $\mathbf{y} \models \mathsf{K}$ *// e.g., that form a valid path*

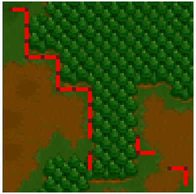
// for a 12×12 grid, 2^{144} states but only 10^{10} valid ones!

NeSy structured output prediction tasks

Vlastelica et al., "Differentiation of blackbox combinatorial solvers", ICLR, 2020

When?

Ground Truth

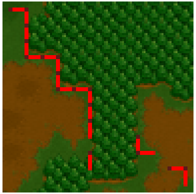


ResNet-18

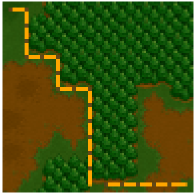
neural nets struggle to satisfy validity constraints!

Constraint losses

Ground Truth



ResNet-18

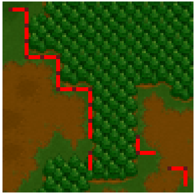


Semantic Loss

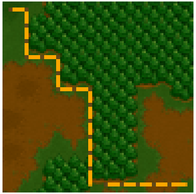
even losses cannot guarantee consistency at test time!

SPL

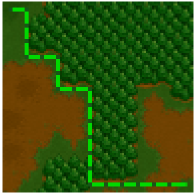
Ground Truth



ResNet-18



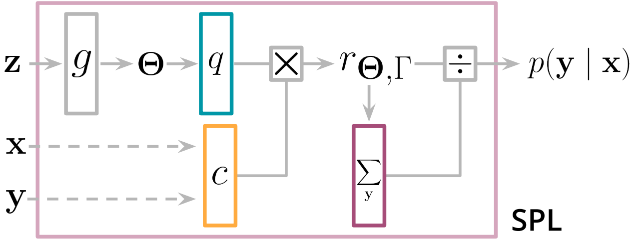
Semantic Loss



SPL (ours)

you can predict valid paths 100% of the time!

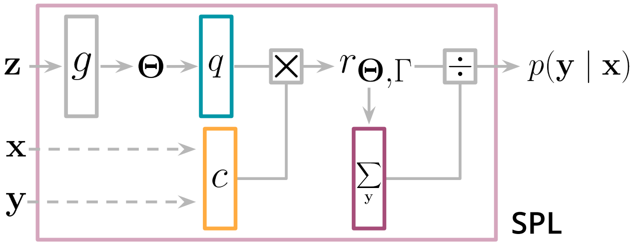
SPL



$$p(\mathbf{y} \mid \mathbf{x}) = q_{\boldsymbol{\Theta}}(\mathbf{y} \mid g(\mathbf{z}))$$

$q_{\boldsymbol{\Theta}}(\mathbf{y} \mid g(\mathbf{z}))$ is an expressive distribution over labels

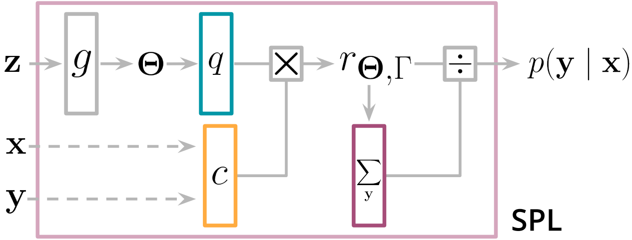
SPL



$$p(\mathbf{y} \mid \mathbf{x}) = q_{\Theta}(\mathbf{y} \mid g(\mathbf{z})) \cdot c_{\mathsf{K}}(\mathbf{x}, \mathbf{y})$$

$c_{\mathsf{K}}(\mathbf{x}, \mathbf{y})$ encodes the constraint $\mathbb{1}\{\mathbf{x}, \mathbf{y} \models \mathsf{K}\}$

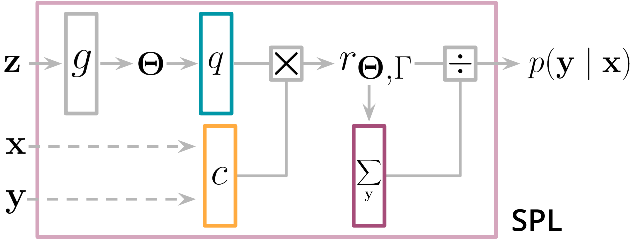
SPL



$$p(\mathbf{y} \mid \mathbf{x}) = q_{\Theta}(\mathbf{y} \mid g(\mathbf{z})) \cdot c_{\mathsf{K}}(\mathbf{x}, \mathbf{y})$$

a product of experts :(

SPL



$$p(\mathbf{y} \mid \mathbf{x}) = q_{\Theta}(\mathbf{y} \mid g(\mathbf{z})) \cdot c_{\mathsf{K}}(\mathbf{x}, \mathbf{y}) / \mathcal{Z}(\mathbf{x})$$

$$\mathcal{Z}(\mathbf{x}) = \sum_{\mathbf{y}} q_{\Theta}(\mathbf{y} \mid \mathbf{x}) \cdot c_{\mathsf{K}}(\mathbf{x}, \mathbf{y})$$

Goal

Can we design q and c
to be expressive models
yet yielding a tractable product?

Goal

Can we design q and c
to be expressive models
yet yielding a tractable product?

yes! as *circuits*!

Probabilistic Circuits (PCs)

A grammar for tractable computational graphs

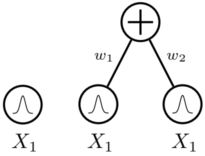
I. *A simple tractable function is a circuit*

Probabilistic Circuits (PCs)

A grammar for tractable computational graphs

I. *A simple tractable function is a circuit*

II. *A weighted combination of circuits is a circuit*



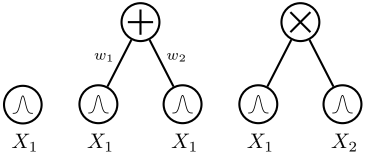
Probabilistic Circuits (PCs)

A grammar for tractable computational graphs

I. *A simple tractable function is a circuit*

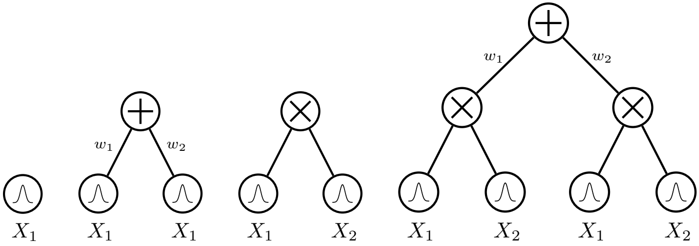
II. *A weighted combination of circuits is a circuit*

III. *A product of circuits is a circuit*



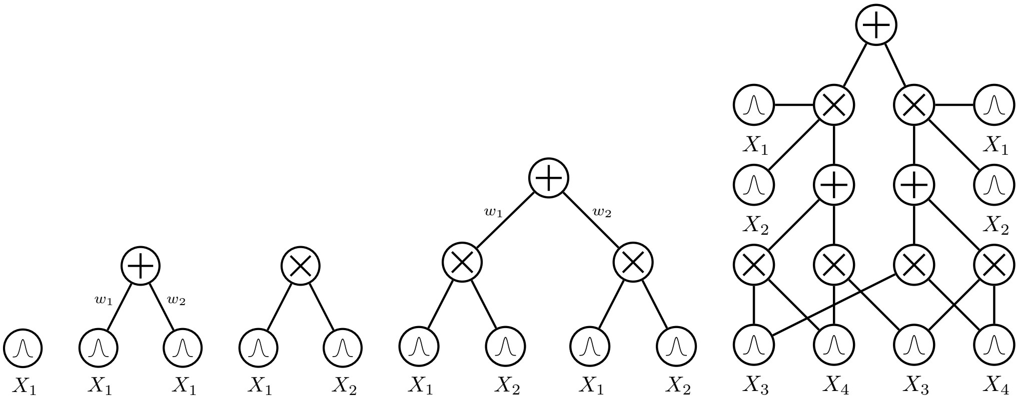
Probabilistic Circuits (PCs)

A grammar for tractable computational graphs



Probabilistic Circuits (PCs)

A grammar for tractable computational graphs



...why PCs?

1. A grammar for tractable models

One formalism to represent many models. *#HMMs #Trees #XGBoost*, ...

2. Expressiveness

Competitive with intractable models, VAEs, Flow...*#hierachical #mixtures #polynomials*

...why PCs?

1. A grammar for tractable models

One formalism to represent many models. *#HMMs #Trees #XGBoost*, ...

2. Expressiveness

Competitive with intractable models, VAEs, Flow...*#hierachical #mixtures #polynomials*

3. Tractability == Structural Properties!!!

Exact computations of reasoning tasks are certified by guaranteeing certain structural properties. *#marginals #expectations #MAP*, *#product* ...

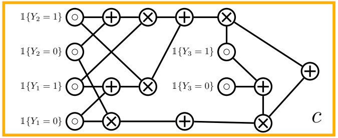
SPL recipe

$$\mathsf{K}: (Y_1 = 1 \implies Y_3 = 1)$$
$$\land \quad (Y_2 = 1 \implies Y_3 = 1)$$

1 Take your logical constraint

SPL recipe

$$K : (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \quad (Y_2 = 1 \implies Y_3 = 1)$$

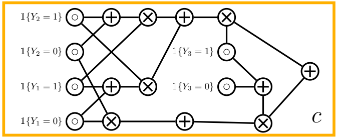


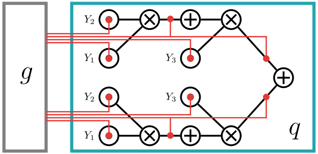
1 Take your logical constraint

2 Compile it into a constraint circuit

SPL recipe

$$
\begin{aligned}
\mathsf{K}: \; & (Y_1 = 1 \implies Y_3 = 1) \\
\land \; & (Y_2 = 1 \implies Y_3 = 1)
\end{aligned}
$$





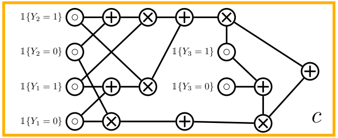
1 Take your logical constraint

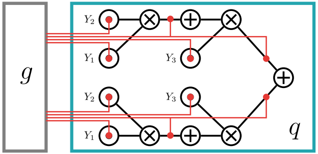
2 Compile it into a constraint circuit

3 Multiply it by a circuit distribution

SPL recipe

$$\mathsf{K} : (Y_1 = 1 \implies Y_3 = 1) \\ \land \quad (Y_2 = 1 \implies Y_3 = 1)$$





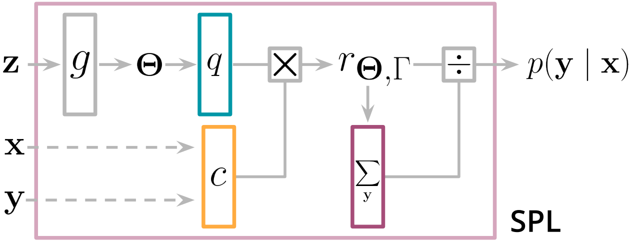
1 Take your logical constraint

2 Compile it into a constraint circuit

3 Multiply it by a circuit distribution

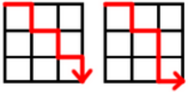
4 ***train end-to-end by sgd!***

Experiments



how good are SPLs?

Experiments



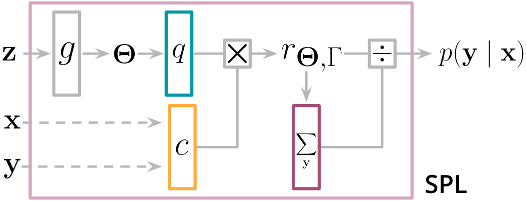
Architecture	Simple Path			Preference Learning		
	Exact	Hamming	Consistent	Exact	Hamming	Consistent
MLP+FIL	5.6	85.9	7.0	1.0	**75.8**	2.7
MLP+$\mathcal{L}_{\text{SL}}$	28.5	83.1	75.2	15.0	72.4	69.8
MLP+NeSyEnt	30.1	83.0	91.6	18.2	71.5	96.0
MLP+SPL	**37.6**	**88.5**	**100.0**	**20.8**	72.4	**100.0**

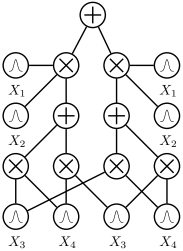
Experiments

Architecture	Exact	Hamming	Consistent
ResNet-18+FIL	55.0	**97.7**	56.9
ResNet-18+$\mathcal{L}_{\text{SL}}$	59.4	**97.7**	61.2
ResNet-18+SPL	**78.2**	96.3	**100.0**

Experiments

Dataset	Exact Match	
	HMCNN	MLP+SPL
CellCycle	3.05 ± 0.11	$\mathbf{3.79 \pm 0.18}$
Derisi	1.39 ± 0.47	$\mathbf{2.28 \pm 0.23}$
Eisen	5.40 ± 0.15	$\mathbf{6.18 \pm 0.33}$
Expr	4.20 ± 0.21	$\mathbf{5.54 \pm 0.36}$
Gasch1	3.48 ± 0.96	$\mathbf{4.65 \pm 0.30}$
Gasch2	3.11 ± 0.08	$\mathbf{3.95 \pm 0.28}$
Seq	5.24 ± 0.27	$\mathbf{7.98 \pm 0.28}$
Spo	$\mathbf{1.97 \pm 0.06}$	1.92 ± 0.11
Diatoms	48.21 ± 0.57	$\mathbf{58.71 \pm 0.68}$
Enron	5.97 ± 0.56	$\mathbf{8.18 \pm 0.68}$
ImCLEF07a	79.75 ± 0.38	$\mathbf{86.08 \pm 0.45}$
ImCLEF07d	76.47 ± 0.35	$\mathbf{81.06 \pm 0.68}$





Check out our code at
github.com/KareemYousrii/SPL
and come to our poster to learn more!