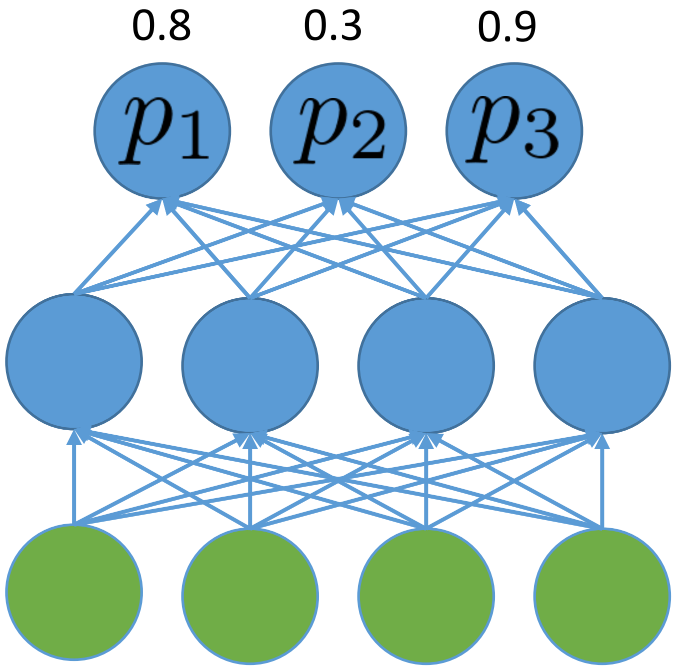
A Semantic Loss Function for Deep Learning with Symbolic Knowledge

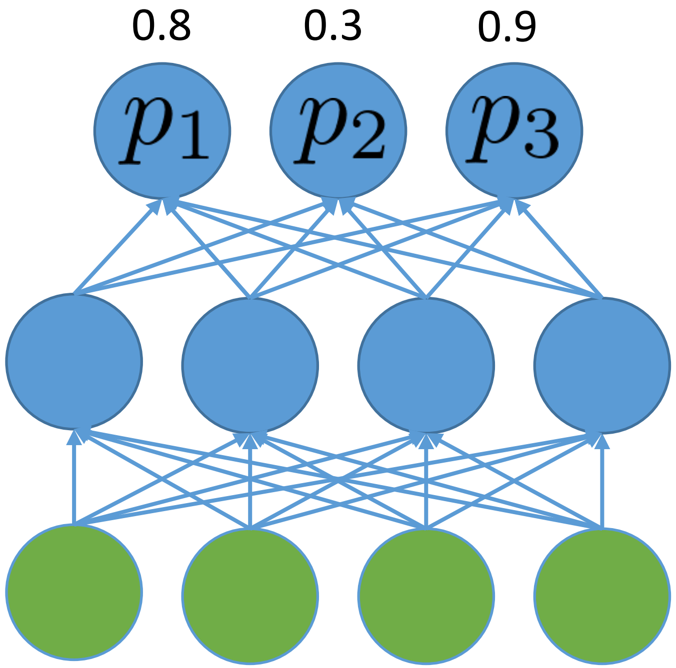
Jingyi Xu, Zilu Zhang, **Tal Friedman**, Yitao Liang,
Guy Van den Broeck

***Goal**: Constrain neural network outputs using logic*

Multiclass Classification

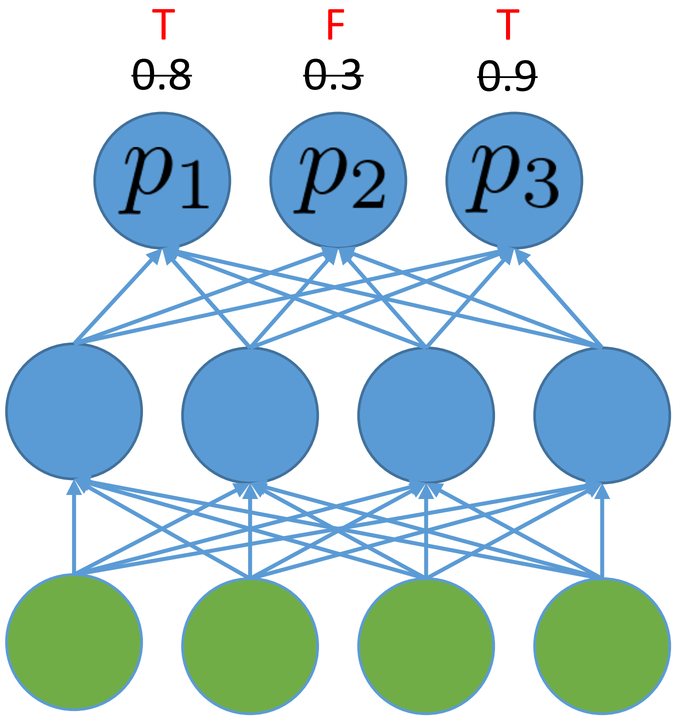


Multiclass Classification



Want exactly one class:
$$\begin{cases} \boldsymbol{x_1 \neg x_2 \neg x_3} \\ \vee \\ \boldsymbol{\neg x_1 x_2 \neg x_3} \\ \vee \\ \boldsymbol{\neg x_1 \neg x_2 x_3} \end{cases}$$

Multiclass Classification

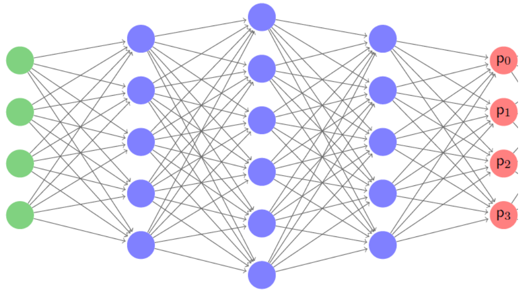


Want exactly one class: $\begin{cases} \boldsymbol{x_1 \neg x_2 \neg x_3} \\ \vee \\ \boldsymbol{\neg x_1 x_2 \neg x_3} \\ \vee \\ \boldsymbol{\neg x_1 \neg x_2 x_3} \end{cases}$

No information gained!

Why is mixing so difficult?

Deep Learning



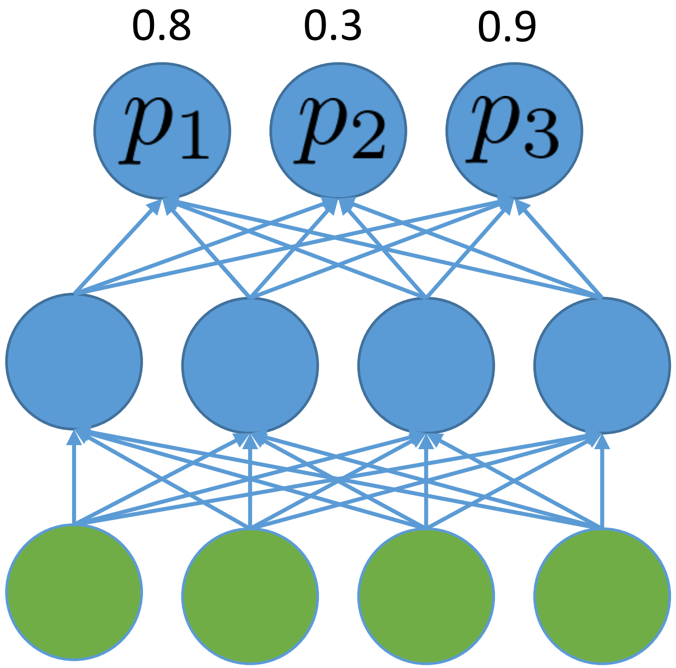
- Continuous
- Smooth
- Differentiable

Logic

$$P \vee L$$
$$A \Rightarrow P$$
$$K \Rightarrow (P \vee L)$$

- Discrete
- Symbolic
- Strong semantics

Multiclass Classification

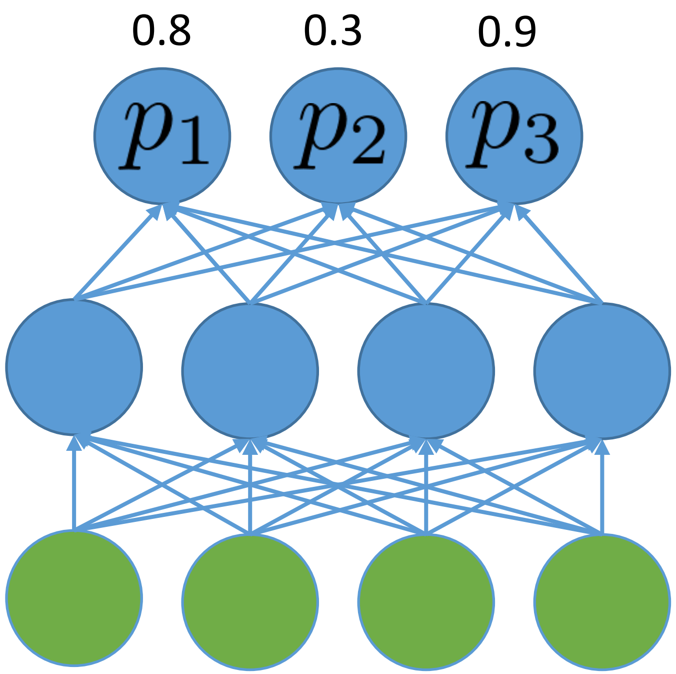


Want exactly one class: $\begin{cases} \boldsymbol{x_1 \neg x_2 \neg x_3} \\ \lor \\ \boldsymbol{\neg x_1 x_2 \neg x_3} \\ \lor \\ \boldsymbol{\neg x_1 \neg x_2 x_3} \end{cases}$

Probability constraint is satisfied

*Use a **probabilistic** interpretation!*

Multiclass Classification



Want exactly one class: $\begin{cases} x_1 \neg x_2 \neg x_3 \\ \vee \\ \neg x_1 x_2 \neg x_3 \\ \vee \\ \neg x_1 \neg x_2 x_3 \end{cases}$

Probability constraint is satisfied

$$\begin{gathered} \boldsymbol{x_1(1 - x_2)(1 - x_3)} \\ \boldsymbol{+(1 - x_1)x_2(1 - x_3)} \\ \boldsymbol{+(1 - x_1)(1 - x_2)x_3} \\ \boldsymbol{= 0.188} \end{gathered}$$

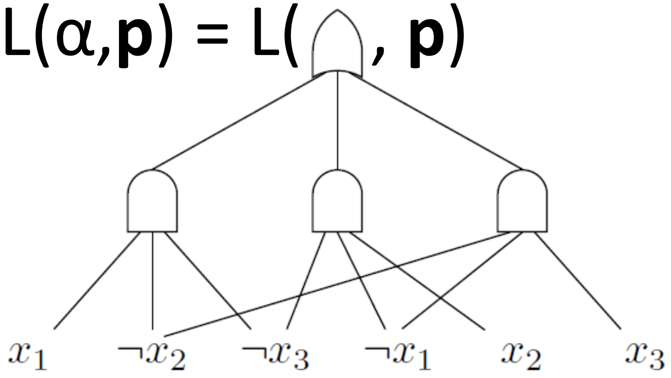
Semantic Loss

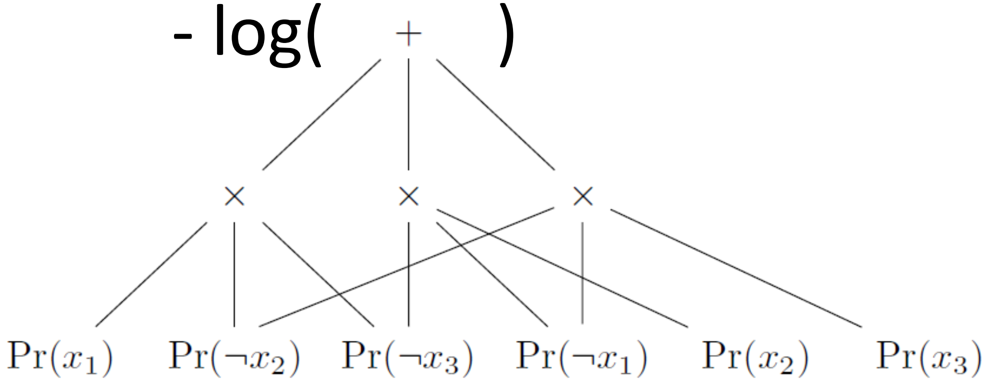
- Continuous, smooth, easily differentiable function
- Represents how close outputs are to satisfying the constraint
- Axiomatically respects semantics of logic, maintains precise meaning – independent of syntax

How do we compute semantic loss?

Logical Circuits

- In general: #P-hard
- Linear in size of circuit

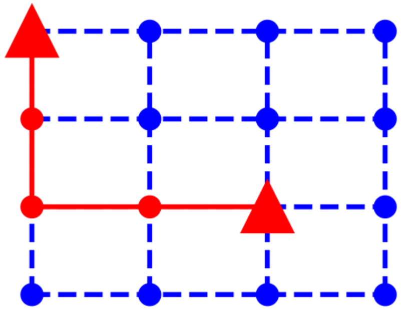
$L(\alpha,\mathbf{p}) = L($ 

$, \mathbf{p})$ $=$ $-\log($ 

$)$

Supervised Learning

- Predict shortest paths
- Add semantic loss representing paths



Test accuracy %	Coherent	Incoherent	Constraint
5-layer MLP	5.62	**85.91**	6.99
Semantic loss	**28.51**	83.14	**69.89**

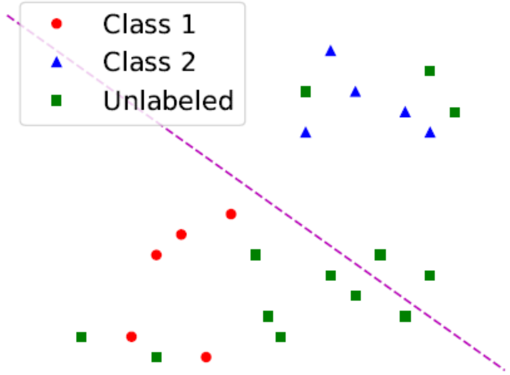
Is output the true shortest path?

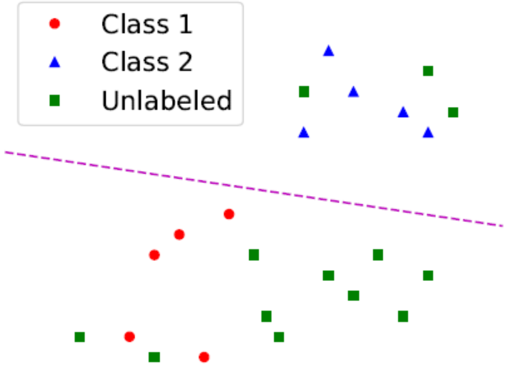
Does output have true edges?

Is output a path?

Semi-Supervised Learning

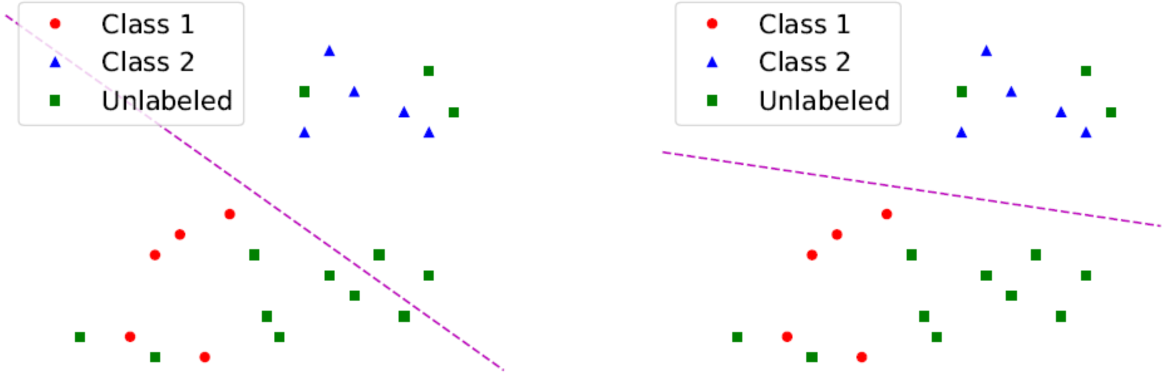
- Unlabeled data must have some label





Semi-Supervised Learning

- Unlabeled data must have some label



- Exactly-one constraint increases confidence

Table 2: FASHION. Test accuracy comparison between MLP with semantic loss and ladder nets.

Accuracy % with # of used labels	100	500	1000	ALL
Ladder Net (Rasmus et al., 2015)	81.46 (±0.64)	85.18 (±0.27)	86.48 (± 0.15)	**90.46**
Baseline: MLP, Gaussian Noise	69.45 (±2.03)	78.12 (±1.41)	80.94 (±0.84)	89.87
MLP with Semantic Loss	**86.74** (±0.71)	**89.49** (±0.24)	**89.67** (±0.09)	89.81

Main Takeaway

- Deep learning and logic **can** be combined by using a probabilistic approach
- Maintain precise meaning while fitting into the deep learning framework

Thanks!