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

0.8  0.3  0.9

$p_1$  $p_2$  $p_3$
Multiclass Classification

Want exactly one class:

\[
\begin{align*}
\neg x_1 \neg x_2 \neg x_3 \\
\neg x_1 x_2 \neg x_3 \\
\neg x_1 \neg x_2 x_3
\end{align*}
\]
Multiclass Classification

Want exactly one class:

\[
\begin{align*}
\neg x_1 \neg x_2 \neg x_3 \\
\lor \\
\neg x_1 x_2 \neg x_3 \\
\lor \\
\neg x_1 \neg x_2 x_3
\end{align*}
\]

No information gained!
Why is mixing so difficult?

Deep Learning

• Continuous
• Smooth
• Differentiable

Logic

\[ P \lor L \]
\[ A \Rightarrow P \]
\[ K \Rightarrow (P \lor L) \]

• Discrete
• Symbolic
• Strong semantics
Multiclass Classification

Want exactly one class:

\[
\begin{align*}
  x_1 \neg x_2 \neg x_3 \\
  \vee \\
  \neg x_1 x_2 \neg x_3 \\
  \vee \\
  \neg x_1 \neg x_2 x_3
\end{align*}
\]

**Probability** constraint is satisfied
Use a \textit{probabilistic} interpretation!
Multiclass Classification

Want exactly one class:

\[
x_1 \lor x_2 \lor x_3 \lor \neg x_1 x_2 \lor x_3 \lor \neg x_1 \neg x_2 x_3
\]

**Probability** constraint is satisfied

\[
x_1(1-x_2)(1-x_3) + (1-x_1)x_2(1-x_3) + (1-x_1)(1-x_2)x_3 = 0.188
\]
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

\[
L(\alpha, p) = L(\alpha, p) = - \log(\cdots)
\]
Supervised Learning

- Predict shortest paths
- Add semantic loss representing paths

<table>
<thead>
<tr>
<th>Test accuracy %</th>
<th>Coherent</th>
<th>Incoherent</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-layer MLP</td>
<td>5.62</td>
<td><strong>85.91</strong></td>
<td>6.99</td>
</tr>
<tr>
<td>Semantic loss</td>
<td><strong>28.51</strong></td>
<td>83.14</td>
<td><strong>69.89</strong></td>
</tr>
</tbody>
</table>

- 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.

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