AI can learn from data. But can it learn to reason?

Guy Van den Broeck
Outline

1. The paradox of learning to reason from data
   deep learning

2. Learning with symbolic knowledge
   logical reasoning + deep learning
Outline

1. The paradox of learning to reason from data

2. Learning with symbolic knowledge
Can Language Models Perform Logical Reasoning?

Language Models achieve high performance on various “reasoning” benchmarks in NLP.

It is unclear whether they solve the tasks following the rules of logical deduction.

**Language Models:**

$input \rightarrow ? \rightarrow Carol \text{ is the grandmother of Justin}.$

**Logical Reasoning:**

$input \rightarrow Justin \text{ in Kristin’s son}; Carol \text{ is Kristin’s mother}; \rightarrow Carol \text{ is Justin’s mother’s mother}; if X \text{ is Y’s mother’s mother then X is Y’s grandmother} \rightarrow Carol \text{ is the grandmother of Justin}.$
Problem Setting: SimpleLogic

Rules: If witty, then diplomatic. If careless and condemned and attractive, then blushing. If dishonest and inquisitive and average, then shy. If average, then stormy. If popular, then blushing. If talented, then hurt. If popular and attractive, then thoughtless. If blushing and shy and stormy, then inquisitive. If adorable, then popular. If cooperative and wrong and stormy, then thoughtless. If popular, then sensible. If cooperative, then wrong. If shy and cooperative, then witty. If polite and shy and thoughtless, then talented. If polite, then condemned. If polite and wrong, then inquisitive. If dishonest and inquisitive, then talented. If blushing and dishonest, then careless. If inquisitive and dishonest, then troubled. If blushing and stormy, then shy. If diplomatic and talented, then careless. If wrong and beautiful, then popular. If ugly and shy and beautiful, then stormy. If shy and inquisitive and attractive, then diplomatic. If witty and beautiful and frightened, then adorable. If diplomatic and cooperative, then sensible. If thoughtless and inquisitive, then diplomatic. If careless and dishonest and troubled, then cooperative. If hurt and witty and troubled, then dishonest. If scared and diplomatic and troubled, then average. If ugly and wrong and careless, then average. If dishonest and scared, then polite. If talented, then dishonest. If condemned, then wrong. If wrong and troubled and blushing, then scared. If attractive and condemned, then frightened. If hurt and condemned and shy, then witty. If cooperative, then attractive. If careless, then polite. If adorable and wrong and careless, then diplomatic. Facts: Alice sensible Alice condemned Alice thoughtless Alice polite Alice scared Alice average
Query: Alice is shy?
Problem Setting: SimpleLogic

The easiest of reasoning problems:

1. **Propositional logic** fragment
   a. bounded vocabulary & number of rules
   b. bounded reasoning depth (≤ 6)
   c. finite space (≈ 10^360)

2. **No language variance**: templated language

3. **Self-contained**
   No prior knowledge

4. **Purely symbolic** predicates
   No shortcuts from word meaning

5. **Tractable** logic (definite clauses)
   Can always be solved efficiently

Facts:
- Alice is *fast*.
- Alice is *normal*.

Rules:
- If Alice is *fast* and *smart*, then Alice is *bad*.
- If Alice is *normal*, then Alice is *smart*.
- If Alice is *normal* and *happy*, then Alice is *sad*.

Query 1: Alice is *bad*. [Answer: True]
Query 2: Alice is *sad*. [Answer: False]

LMs: BERT, T5

True or False
Training a BERT model on SimpleLogic

(1) Randomly sample facts & rules.
Facts: B, C
Rules: A, B \rightarrow D, B \rightarrow E, B, C \rightarrow F.

(2) Compute the correct labels for all predicates given the facts and rules.

(2) Set B, C (randomly chosen among B, C, E, F) as facts and sample rules (randomly) consistent with the label assignments.

Test accuracy for different reasoning depths

<table>
<thead>
<tr>
<th></th>
<th>Test</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP</td>
<td></td>
<td>99.9</td>
<td>99.8</td>
<td>99.7</td>
<td>99.3</td>
<td>98.3</td>
<td>97.5</td>
<td>95.5</td>
</tr>
<tr>
<td>LP</td>
<td></td>
<td>100.0</td>
<td>100.0</td>
<td>99.9</td>
<td>99.9</td>
<td>99.7</td>
<td>99.7</td>
<td>99.0</td>
</tr>
</tbody>
</table>

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang and Guy Van den Broeck. On the Paradox of Learning to Reason from Data, 2022
Has BERT learned to reason from data?

1. Easiest of reasoning problems (no variance, self-contained, purely symbolic, tractable)
2. RP/LP data covers the whole problem space
3. The learned model has almost 100% test accuracy
4. There exist BERT parameters that compute the ground-truth reasoning function:

   **Theorem 1:** For a BERT model with $n$ layers and 12 attention heads, by construction, there exists a set of parameters such that the model can correctly solve any reasoning problem in SimpleLogic that requires at most $n - 2$ steps of reasoning.

Surely, under these conditions, BERT has learned the ground-truth reasoning function!
The Paradox of Learning to Reason from Data

1. If BERT has learned to reason, it should not exhibit such generalization failure.

2. If BERT has not learned to reason, it is baffling how it achieves near-perfect in-distribution test accuracy.

The BERT model trained on one distribution fails to generalize to the other distribution within the same problem space.

<table>
<thead>
<tr>
<th>Train</th>
<th>Test</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP</td>
<td>RP</td>
<td>99.9</td>
<td>99.8</td>
<td>99.7</td>
<td>99.3</td>
<td>98.3</td>
<td>97.5</td>
<td>95.5</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>99.8</td>
<td>99.8</td>
<td>99.3</td>
<td>96.0</td>
<td>90.4</td>
<td>75.0</td>
<td>57.3</td>
</tr>
<tr>
<td>LP</td>
<td>RP</td>
<td>97.3</td>
<td>66.9</td>
<td>53.0</td>
<td>54.2</td>
<td>59.5</td>
<td>65.6</td>
<td>69.2</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>100.0</td>
<td>100.0</td>
<td>99.9</td>
<td>99.9</td>
<td>99.7</td>
<td>99.7</td>
<td>99.0</td>
</tr>
</tbody>
</table>
Why? Statistical Features

Monotonicity of entailment:
Any rules can be freely added to the hypothesis of any proven fact.

The more rules given, the more likely a predicate will be proved.

Pr(label = True | Rule # = x) should increase (roughly) monotonically with x
BERT leverages statistical features to make predictions

RP_b downsamples from RP such that Pr(label = True | rule# = x) = 0.5 for all x

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP</td>
<td>99.9</td>
<td>99.8</td>
<td>99.7</td>
<td>99.3</td>
<td>98.3</td>
<td>97.5</td>
<td>95.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP_b</td>
<td>99.0</td>
<td>99.3</td>
<td>98.5</td>
<td>97.5</td>
<td>96.7</td>
<td>93.5</td>
<td>88.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Accuracy drop from RP to RP_b indicates that the model is using rule# as a statistical feature to make predictions.

2. Though removing one statistical feature from training data can help with model generalization, there are potentially countless statistical features and it is computationally infeasible to jointly remove them.
First Conclusion

Experiments unveil the fundamental difference between

1. learning to reason, and
2. learning to achieve high performance on benchmarks using statistical features.

Be careful deploying AI in applications where this difference matters.
Outline

1. The paradox of learning to reason from data
   deep learning

2. Learning with symbolic knowledge
   logical reasoning + deep learning
The AI Dilemma of 2022

**Deep learning** approaches the problem of designing intelligent machines by postulating a large number of very simple information processing elements, arranged in a [.] network, and certain processes for facilitating or inhibiting their activity.

**Knowledge representation and reasoning** take a much more macroscopic approach [ ]. They believe that intelligent performance by a machine is an end difficult enough to achieve without “starting from scratch”, and so they build into their systems as much complexity of information processing as they are able to understand and communicate to a computer.
Neural cybernetics approaches the problem of designing intelligent machines by postulating a large number of very simple information processing elements, arranged in a network, and certain processes for facilitating or inhibiting their activity.

Cognitive model builders take a much more macroscopic approach. They believe that intelligent performance by a machine is an end difficult enough to achieve without “starting from scratch”, and so they build into their systems as much complexity of information processing as they are able to understand and communicate to a computer.
The AI Dilemma

Pure Reasoning

Integrate reasoning into modern deep learning algorithms

Pure Learning
Knowledge in Vision, Robotics, NLP, Activity Recognition

People appear at most once in a frame

Rigid objects don’t overlap

At least one verb in each sentence.
If X and Y are married, then they are people.

Cut the orange before squeezing the orange

Predict Loan Amount

Neural Network Model: Increasing income can decrease the approved loan amount

Monotonicity (Prior Knowledge): Increasing income should increase the approved loan amount

WARCRAFT SHORTEST PATH

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

[Differentiation of Blackbox Combinatorial Solvers, Marin Vlastelica, Anselm Paulus, Vit Musil, Georg Martius, Michal Rolínek, 2019]
Knowledge vs. Data

• Where did the world knowledge go?
  – Python scripts
    • Decode/encode/search cleverly
    • Fix inconsistent beliefs
  – Rule-based decision systems
  – Dataset design
  – “a big hack” (with author’s permission)

• In some sense we went backwards
  Less principled, scientific, and intellectually satisfying ways of incorporating knowledge
pylon

A PyTorch Framework for Learning with Constraints

Kareem Ahmed    Tao Li    Thy Ton    Quan Guo,
Kai-Wei Chang    Parisa Kordjamshidi    Vivek Srikumar
Guy Van den Broeck    Sameer Singh

http://pylon-lib.github.io
Declarative Knowledge of the Output

How is the output structured? Are all possible outputs valid?

How are the outputs related to each other?

Learning this from data is inefficient. Much easier to express this declaratively.
Library that extends PyTorch to allow injection of declarative knowledge

- **Easy to Express Knowledge**: users write arbitrary constraints on the output
- **Integrates with PyTorch**: minimal change to existing code
- **Efficient Training**: compiles into loss that can be efficiently optimized
  - Exact semantic loss (see later)
  - Monte-carlo estimate of loss
  - T-norm approximation
  - *your solver?*
PyTorch Code

```python
for i in range(train_iters):
    ...
    py = model(x)
    ...
    loss = CrossEntropy(py,...)
```

1 Specify knowledge as a predicate

```python
def check(y):
    ...
    return isValid
```
PyTorch Code

```python
for i in range(train_iters):
    ...
    py = model(x)
    ...
    loss = CrossEntropy(py,...)
    loss += constraint_loss(check)(py)
```

1. Specify knowledge as a predicate
   ```python
def check(y):
    ...
    return is_Valid
   ```

2. Add as loss to training
   ```python
   loss += constraint_loss(check)
   ```
PyTorch Code

```python
for i in range(train_iters):
    ...
    py = model(x)
    ...
    loss = CrossEntropy(py,...)
    loss += constraint_loss(check)(py)
```

1. Specify knowledge as a predicate
   ```python
def check(y):
    ...
    return isValid
```

2. Add as loss to training
   ```python
   loss += constraint_loss(check)
   ```

3. `pylon` derives the gradients (solves a combinatorial problem)
without constraint

with constraint

without constraint

with constraint
## Warcraft min-cost simple-path prediction results

<table>
<thead>
<tr>
<th></th>
<th>Test accuracy</th>
<th>Coherent</th>
<th>Incoherent</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-18</td>
<td>44.8</td>
<td>97.7</td>
<td>56.9</td>
<td></td>
</tr>
</tbody>
</table>

*Is prediction the shortest path?*

This is the real task!

*Are individual edge predictions correct?*

*Is output a path?*
without constraint  

with constraint  

without constraint  

with constraint  

without constraint  

with constraint
## Warcraft min-cost simple-path prediction results

<table>
<thead>
<tr>
<th>Test accuracy %</th>
<th>Coherent</th>
<th>Incoherent</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-18</td>
<td>44.8</td>
<td>97.7</td>
<td>56.9</td>
</tr>
<tr>
<td>+ Semantic loss</td>
<td><strong>50.9</strong></td>
<td><strong>97.7</strong></td>
<td><strong>67.4</strong></td>
</tr>
</tbody>
</table>
a) A network uncertain over both valid & invalid predictions

\[ L^s(\alpha, p) \propto -\log \sum_{x} \prod_{i:x \models X_i} p_i \prod_{i:x \models \neg X_i} (1 - p_i) \]

Probability of satisfying constraint \( \alpha \) after sampling from neural net output layer \( p \)

In general: \#P-hard ☹

Do this probabilistic-logical reasoning during learning in a computation graph

Neuro-Symbolic Learning

p(y|x)

\[ p(y|x) \]

\[ y \]

\[ m(\alpha) \]

c) A network allocating most of its mass to models of constraint
\[ \alpha: A \land B \Rightarrow C \]

\[ -\log(p) \]

**Semantic Loss**

**Probability**
a) A network uncertain over both valid & invalid predictions

b) A network allocating most of its mass to an invalid prediction.

c) A network allocating most of its mass to models of constraint

d) A network allocating most of mass to one model of formula

\[
-\mathbb{E}_{P(Y|x, \alpha)} \left[ \log P(Y|x, \alpha) \right]
\]
Warcraft min-cost simple-path prediction results

<table>
<thead>
<tr>
<th>Test accuracy</th>
<th>Coherent</th>
<th>Incoherent</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-18</td>
<td>44.8</td>
<td>97.7</td>
<td>56.9</td>
</tr>
<tr>
<td>Semantic loss</td>
<td>50.9</td>
<td>97.7</td>
<td>67.4</td>
</tr>
<tr>
<td>+ Full Entropy</td>
<td>51.5</td>
<td>97.6</td>
<td>67.7</td>
</tr>
<tr>
<td>+ NeSy Entropy</td>
<td><strong>55.0</strong></td>
<td><strong>97.9</strong></td>
<td><strong>69.8</strong></td>
</tr>
</tbody>
</table>
# Joint entity-relation extraction in natural language processing

<table>
<thead>
<tr>
<th></th>
<th>#</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>25</th>
<th>50</th>
<th>75</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Baseline</strong></td>
<td><strong>Self-training</strong></td>
<td><strong>Product t-norm</strong></td>
<td><strong>Semantic Loss</strong></td>
<td><strong>+ Full Entropy</strong></td>
<td><strong>+ NeSy Entropy</strong></td>
<td><strong>ACE05</strong></td>
<td><strong>SciERC</strong></td>
</tr>
<tr>
<td></td>
<td>4.92 ± 1.12</td>
<td>7.24 ± 1.75</td>
<td>13.66 ± 0.18</td>
<td>15.07 ± 1.79</td>
<td>21.65 ± 3.41</td>
<td>28.96 ± 0.98</td>
<td>33.02 ± 1.17</td>
<td>12.00 ± 3.81</td>
</tr>
<tr>
<td></td>
<td>7.72 ± 1.21</td>
<td>12.83 ± 2.97</td>
<td>16.22 ± 3.08</td>
<td>17.55 ± 1.41</td>
<td>27.00 ± 3.66</td>
<td>32.90 ± 1.71</td>
<td>37.15 ± 1.42</td>
<td>15.78 ± 1.90</td>
</tr>
<tr>
<td></td>
<td>8.89 ± 5.09</td>
<td>14.52 ± 2.13</td>
<td>19.22 ± 5.81</td>
<td>21.80 ± 7.67</td>
<td>30.15 ± 1.01</td>
<td>34.12 ± 2.75</td>
<td>37.35 ± 2.53</td>
<td>20.92 ± 1.60</td>
</tr>
</tbody>
</table>

Semantic Probabilistic Layers

- How to give a 100% guarantee that Boolean constraints will be satisfied?
- Bake the constraint into the neural network as a special layer

- Secret sauce is again tractable circuits – computation graphs for reasoning

Table 3: Warcraft shortest path prediction results

<table>
<thead>
<tr>
<th>ARCHITECTURE</th>
<th>EXACT MATCH</th>
<th>HAMMING SCORE</th>
<th>CONSISTENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-18+FIL</td>
<td>55.0</td>
<td>97.7</td>
<td>56.9</td>
</tr>
<tr>
<td>ResNet-18+\mathcal{L}_{SL}</td>
<td>59.4</td>
<td>97.7</td>
<td>61.2</td>
</tr>
<tr>
<td>ResNet-18+SPL</td>
<td>75.1</td>
<td>97.6</td>
<td>100.0</td>
</tr>
<tr>
<td>Overparam. SDD</td>
<td>78.2</td>
<td>96.3</td>
<td>100.0</td>
</tr>
</tbody>
</table>
Hierarchical Multi-Label Classification

“if the image is classified as a dog, it must also be classified as an animal”

“if the image is classified as an animal, it must be classified as either cat or dog”

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Exact Match</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HMCNN</td>
</tr>
<tr>
<td>CELL_CYCLE</td>
<td>3.05 ± 0.11</td>
</tr>
<tr>
<td>DERISI</td>
<td>1.39 ± 0.47</td>
</tr>
<tr>
<td>EISEN</td>
<td>5.40 ± 0.15</td>
</tr>
<tr>
<td>EXPR</td>
<td>4.20 ± 0.21</td>
</tr>
<tr>
<td>GASCH1</td>
<td>3.48 ± 0.96</td>
</tr>
<tr>
<td>GASCHE2</td>
<td>3.11 ± 0.08</td>
</tr>
<tr>
<td>SEQ</td>
<td>5.24 ± 0.27</td>
</tr>
<tr>
<td>SPO</td>
<td>1.97 ± 0.06</td>
</tr>
<tr>
<td>DIATOMS</td>
<td>48.21 ± 0.57</td>
</tr>
<tr>
<td>ENRON</td>
<td>5.97 ± 0.56</td>
</tr>
<tr>
<td>IMCLEF07A</td>
<td>79.75 ± 0.38</td>
</tr>
<tr>
<td>IMCLEF07D</td>
<td>76.47 ± 0.35</td>
</tr>
</tbody>
</table>
Outline

1. The paradox of learning to reason from data
   deep learning

2. Learning with symbolic knowledge
   logical (and probabilistic) reasoning + deep learning
The AI Dilemma

Integrate reasoning into modern deep learning algorithms
• Knowledge is (hidden) everywhere in ML
• A little bit of reasoning goes a long way!
Thanks

This was the work of many wonderful students/postdocs/collaborators!

Honghua  Kareem

References: http://starai.cs.ucla.edu/publications/