Integrating Domain-Knowledge into Deep Learning

Russ Salakhutdinov

Machine Learning Department
Carnegie Mellon University
Canadian Institute for Advanced Research
Domain knowledge

- Two key ingredients of a Statistical Machine Learning system
  - Model architecture/class
  - Learning algorithms to learn from data

- How do we incorporate domain knowledge into either or both these ingredients?

- We can consider three classes of domain knowledge:
  - Relational
  - Logical
  - Scientific

Ravikumar, Salakhutdinov, 2019
Relational Knowledge

- Simple relations among entities
  - (father, Bob, Alice)

- Available via relational databases, or knowledge graphs

- Statistical Relational Models
  - Probabilistic Graphical Models (PGMs) to model relationships amongst entities
  - Probabilistic Relational Models (via Bayes Nets), Relational Dependency Networks

- Embeddings
  - Instead of distributional semantics, represent entities via vectors in some vector space
  - Learn these vector representations via predicting an entity given its “context”

- We show how to incorporate relational information in Deep Learning via knowledge graph propagation

Ravikumar, Salakhutdinov, 2019
Logical Knowledge

- Propositional and First Order Logic (FOL) based knowledge
  - In contrast to simpler tuple based relational knowledge
  - E.g. if object has a wing, and a beak, it is a bird

- Encode logical knowledge into Probabilistic Graphical Models
- Bayesian Networks from Horn clauses, Probabilistic Context Free Grammars, Markov Logic Networks

- We incorporate logical information (and more general constraints) into Deep Learning via distillation (student-teacher) framework

Ravikumar, Salakhutdinov, 2019
Scientific Knowledge

- Partial and Stochastic Differential Equations
  - Newton Laws of Motion
  - Navier-Stokes fluid dynamics equations
  - ...
- Conservation laws and principles, Invariances

- Learning PDEs from data
- Regularizing dynamical system (e.g. state space models) via PDEs

Ravikumar, Salakhutdinov, 2019
Her plain face broke into a huge smile when she saw Terry. “Terry!” she called out. She rushed to meet him and they embraced. “Hon, I want you to meet an old friend, Owen McKenna. Owen, please meet Emily.” She gave me a quick nod and turned back to X.
Her plain face broke into a huge smile when she saw Terry. “Terry!” she called out. She rushed to meet him and they embraced. “Hon, I want you to meet an old friend, Owen McKenna. Owen, please meet Emily.” She gave me a quick nod and turned back to the LAMBADA dataset, Paperno et al., 2016.
Her plain face broke into a huge smile when she saw Terry. “Terry!” she called out.
She rushed to meet him and they embraced.
“Hon, I want you to meet an old friend, Owen McKenna.
Owen, please meet Emily.”
She gave me a quick nod and turned back to X

\[ X = Terry \]
Her plain face broke into a huge smile when she saw Terry. “Terry!” she called out. She rushed to meet him and they embraced. “Hon, I want you to meet an old friend, Owen McKenna. Owen, please meet Emily.” She gave me a quick nod and turned back to X.

Incorporating Prior Knowledge

Core NLP
Freebase
WordNet

Coreference
Dependency Parses
Entity relations
Word relations

Deep Learning Model (e.g. RNN, Transformers)

Text Representation
Explicit Memory

Mary got the football
She went to the kitchen
She left the ball there

RNN
Coreference
Hyper/Hyponymy

Dhingra, Jin, Yang, et al, NAACL 2018
Explicit Memory

Memory as Acyclic Graph Encoding (MAGE) - RNN

Dhingra, Jin, Yang, et al, NAACL 2018
Open Domain Question Answering

- Finding answers to factual questions posed in Natural Language:

  **Who voiced Meg in Family Guy?**
  
  A. Lacey Chabert, Mila Kunis

  **Who first voiced Meg in Family Guy?**
  
  A. Lacey Chabert

Dhingra, Sun, et al., EMNLP 2018
Who first voiced Meg in Family Guy?

Semantic Parsing

Knowledge Base as a Knowledge Source

KB

Query Graph

12/26/1999

Mila Kunis

character

appear_in

1/31/1999

Lacey Chabert

character

Family Guy

cast

series

cast

series

write

argmin

character

Meg Griffin

Family Guy

cast

actor

x

y

Lacey Chabert

01: from

actor

character

appear_in
Unstructured Text as a Knowledge Source

Step 1 (Information Retrieval):
Retrieve passages relevant to the Question using shallow methods.

Step 2 (Reading Comprehension):
Perform deep reading of passages to extract answers.

Q: How many of Warsaw’s inhabitants spoke Polish in 1933?

WIKIPEDIA
The Free Encyclopedia

Document Retriever

Document Reader

833,500
Meg Griffin is a character from the animated television series Family Guy. Originally voiced by Lacey Chabert during the first season, she has been voiced by Mila Kunis since season 2.
Given a graph $G = (\mathcal{V}, \mathcal{E})$ and a natural language question $q = (w_1, \ldots, w_T)$ learn a function $y_v = f(v) \forall v \in \mathcal{V}$, s.t. $y_v \in \{0, 1\}$ and $y_v = 1$ if and only if $v$ is an answer for $q$.

$$P(y_v = 1|G, q) = \frac{\exp h_q^T h_v}{\sum_{v'} \exp h_q^T h_{v'}}$$

$h_q$ -- Question Representation from an LSTM

$h_v$ -- Node Representation from a Graph Convolution Network

Dhingra, Sun, et al., EMNLP 2018
Graph Convolution Network

For each $v$

- Initialize $h_v^{(0)}$

$$h_v^{(t)} = f(W_1 h_v^{(t-1)} + W_2 \sum_{v' \in N(v)} \alpha_{v'} h_{v'}^{(t-1)})$$

- Repeat for $t = 1, \ldots, T$

Kipf et al., 2016
Relational Graph Convolution Network

Graphs with edge types

\[ h_v^{(t)} = f \left( \sum_r W_1 h_v^{(t-1)} + W_2^r \sum_{v' \in N_r(v)} \alpha_{v',h_v^{(t-1)}} \right) \]

Schlichtkrull et al. 2017
Graph Propagation / Graph Convolution

**Entities**

$$h_e^{(0)} = L(e) \in \mathbb{R}^p$$

**Documents**

$$h_d^{(0)} = \text{LSTM}(d_{w_1}, \ldots, d_{w_T}) \in \mathbb{R}^{T \times p}$$

Meg Griffin is a character from the animated television series *Family Guy*
Meg Griffin is a character from the animated television series *Family Guy*

\[
h_{d}^{(t)} = \text{LSTM}(h_{d1}^{(t-1)} || e_{w1}^{(t-1)}, \ldots, h_{dT}^{(t-1)} || e_{wT}^{(t-1)})
\]

Meg Griffin is a character from the animated television series *Family Guy*

\[
h_{e}^{(t)} = f(W_{1}h_{e}^{(t-1)} + \sum_{r} \sum_{v' \in N_{r}(v)} W_{2}^{r}h_{v'}^{(t-1)} + W_{3} \sum_{d:e \in d} h_{d}^{(t-1)})
\]

- Relational information via KB propagation

Dhingra, Sun, et al., EMNLP 2018
Domain knowledge

- We consider three classes of domain knowledge:
  - Relational
  - Logical (constraints)
  - Scientific
Incorporating Constraints

- Consider a statistical model \( x \sim p_\theta(x) \)
- Consider a constraint function, \( f_\phi(x) \in \mathbb{R} \) parameterized by \( \phi \)
  - Higher \( f_\phi(x) \) value, better \( x \) w.r.t the knowledge

Pose-conditional Human Image Generation

source image \( \rightarrow \) target pose \( \rightarrow \) target

Generative Model \( p_\theta(x) \) \( \rightarrow \) generated image

DeepFashion, Liu et.al., CVPR 2016
Incorporating Constraints

- Consider a statistical model $x \sim p_\theta(x)$
- Consider a constraint function, $f_\phi(x) \in \mathbb{R}$ parameterized by $\phi$
  - Higher $f_\phi(x)$ value, better $x$ w.r.t the knowledge

DeepFashion, Liu et al., CVPR 2016

Hu, Yang, et al., NeurIPS 2018
Learning with Constraints

- Consider a statistical model $\mathbf{x} \sim p_\theta(\mathbf{x})$
- Consider a constraint function, $f_\phi(\mathbf{x}) \in \mathbb{R}$ parameterized by $\phi$
  - Higher $f_\phi(\mathbf{x})$ value, better $\mathbf{x}$ w.r.t the knowledge

Pose-conditional Human Image Generation

DeepFashion, Liu et al., CVPR 2016

Hu, Yang, et al., NeurIPS 2018
Learning with Constraints

- Consider a statistical model $\mathbf{x} \sim p_\theta(\mathbf{x})$
- Consider a constraint function, $f_\phi(\mathbf{x}) \in \mathbb{R}$ parameterized by $\phi$
  - Higher $f_\phi(\mathbf{x})$ value, better $\mathbf{x}$ w.r.t the knowledge

- Sentiment prediction:
  - This was a terrific movie, but the director could have done better

- Logical Rules:
  - Sentence $S$ with structure $A$-but-$B$: $\implies$ sentiment of $B$ dominates
Learning with Constraints

- Consider a statistical model \( x \sim p_\theta(x) \)
- Consider a constraint function, \( f_\phi(x) \in \mathbb{R} \) parameterized by \( \phi \)
  - Higher \( f_\phi(x) \) value, better \( x \) w.r.t the knowledge

- One way to impose the constraint is to maximize: \( \mathbb{E}_{p_\theta}[f_\phi(x)] \)
- Objective:

\[
\min_\theta \left( L(\theta) - \alpha \mathbb{E}_{p_\theta}[f_\phi(x)] \right)
\]

Regular objective (e.g. cross-entropy loss, etc.)

Regularization: imposing constraints – difficult to compute
Posterior Regularization (Ganchev et al., 2010)

- Consider a statistical model $\mathbf{x} \sim p_\theta(\mathbf{x})$
- Consider a constraint function, $f_\phi(\mathbf{x}) \in \mathbb{R}$ parameterized by $\phi$

\[
\min_\theta \left( \mathcal{L}(\theta) - \alpha \mathbb{E}_{p_\theta}[f_\phi(\mathbf{x})] \right)
\]

\[
\mathcal{L}(\theta, q) = \text{KL}(q(\mathbf{x}) \| p_\theta(\mathbf{x})) - \lambda \mathbb{E}_q[f_\phi(\mathbf{x})]
\]

- Introduce variational distribution $q$, which is encouraged to stay close to $p$

- Objective:

\[
\min_{\theta, q} \left( \mathcal{L}(\theta) + \alpha \mathcal{L}(\theta, q) \right)
\]
Posterior Regularization (Ganchev et al., 2010)

$$\min_{\theta, q} \left( \mathcal{L}(\theta) + \alpha \mathcal{L}(\theta, q) \right)$$

$$\mathcal{L}(\theta, q) = \text{KL}(q(x) || p_\theta(x)) - \lambda \mathbb{E}_q[f_\phi(x)]$$

- Optimal solution for q:

$$q^*(x) = p_\theta(x) \exp\left( \lambda f_\phi(x) \right) / \mathcal{Z}$$

Higher value -- higher probability under q – “soft constraint”

- How do we fit our model parameters $\theta$?
Logical Rule Formulation (Zhiting Hu et al., 2016)

- Consider a supervised learning: $p_\theta(y|x)$, e.g. deep neural network.
- Input-Target space $(X,Y)$.
- First-order logic rules: $(r, \lambda)$
  - $r(X,Y) \in [0,1]$, could be soft.
  - $\lambda$ is the confidence level of the rule.

- Within PR framework given $l$ rules

\[
q^*(y|x) = p_\theta(y|x) \exp \left( \sum_l \lambda_l r_l(y,x) \right) / Z
\]

- How to train a neural network: Knowledge Distillation [Hinton et al., 2015; Bucilu et al., 2006].
Knowledge Distillation

Match soft predictions of the teacher network and student network

Knowledge Distillation [Hinton et al., 2015; Bucilu et al., 2006].
Rule Knowledge Distillation

- Deep neural network $p_\theta(y|x)$
- Train to imitate the outputs of the rule-regularized teacher network
- At iteration $t$:

$$\theta^{(t+1)} = \arg\min_\theta \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, \sigma_\theta(x)) + \alpha \ell(s_n^{(t)}, \sigma_\theta(x))$$

where $q^*(y|x) = p_\theta(y|x) \exp\left(\sum_{l} \lambda_l r_l(y, x)\right)/Z$

Zhiting Hu et al., ACL 2016
Rule Knowledge Distillation (Zhiting Hu et al., 2016)

- Deep neural network $p_\theta(y|x)$

Zhiting Hu et al., ACL 2016
Rule Knowledge Distillation (Zhiting Hu et al., 2016)

- Deep neural network $p_\theta(y|x)$
Rule Knowledge Distillation (Zhiting Hu et al., 2016)

- Deep neural network $p_\theta(y|x)$
- At each iteration:
  - Construct a teacher network $q(y|x)$ with “soft constraints”
  - Train DNN to emulate the teacher network

Zhiting Hu et al., ACL 2016
Rule Knowledge Distillation (Zhiting Hu et al., 2016)

- Deep neural network $p_\theta(y|x)$
- At each iteration:
  - Construct a teacher network $q(y|x)$ with “soft constraints”
  - Train DNN to emulate the teacher network

- Sentiment classification,
- Named entity recognition

Zhiting Hu et al., ACL 2016
Learning Rules / Constraints

\[ q^*(y|x) = p\theta(y|x) \exp \left( \sum_l \lambda_l r_l(y, x) \right) / Z \]

- We can also learn the "confidence" values \( \lambda_l \) for logical rules

- More generally, we can optimize parameters of the constraint function \( f_\phi(x) \)

\[ q^*(x) = p\theta(x) \exp \left( \lambda f_\phi(x) \right) / Z \]

- Treat \( f_\phi(x) \) as the reward function to be learned within the MaxEnt Inverse Reinforcement Learning
Pose-conditional Human Image Generation

Table 2: Results of image generation using Structural Similarity (SSIM) [52] between generated and true images, and human survey where the full model yields better generations than the base models (Rows 5-6) on 77% test cases. See the text for more results and discussion.

<table>
<thead>
<tr>
<th>Method</th>
<th>SSIM</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Ma et al. [38]</td>
<td>0.614</td>
<td>—</td>
</tr>
<tr>
<td>2 Pumarola et al. [44]</td>
<td>0.747</td>
<td>—</td>
</tr>
<tr>
<td>3 Ma et al. [37]</td>
<td>0.762</td>
<td>—</td>
</tr>
<tr>
<td>4 Base model</td>
<td>0.676</td>
<td>0.03</td>
</tr>
<tr>
<td>5 With fixed constraint</td>
<td>0.679</td>
<td>0.12</td>
</tr>
<tr>
<td>6 With learned constraint</td>
<td>0.727</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Samples generated by the models. Enforcing learned human part constraint generates correct poses and better preserves human body structure.
Template-guided Sentence Generation

- **Task:** Given a template, generate a complete sentence following the template
- **Constraint:** force to match between infilling content of the generated sentence with the true content

**template:**

```
“____ meant to ___
not to _______”
```

**true target:**

```
“It was meant to dazzle
not to make sense.”
```

**generated:**

```
“It was meant to dazzle
not to make it.”
```

**Constraint**

```
Learnable module \( \phi \)
```

Infilling content matching

Hu, Yang, et al., NeurIPS 2018
Template-guided Sentence Generation

<table>
<thead>
<tr>
<th>Model</th>
<th>Perplexity</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Base model</td>
<td>30.30</td>
<td>0.19</td>
</tr>
<tr>
<td>2 With binary D</td>
<td>30.01</td>
<td>0.20</td>
</tr>
<tr>
<td>3 With constraint updated in M-step (Eq.5)</td>
<td>31.27</td>
<td>0.15</td>
</tr>
<tr>
<td>4 With learned constraint</td>
<td>28.69</td>
<td>0.24</td>
</tr>
</tbody>
</table>

Samples by the full model are considered as of higher quality in 24% cases.

Two test examples, including the template, the sample by the base model, and the sample by the constrained model.

Template: I will give the movie ___

Base model: ___ is the acting.

With binary D: ___ is also very good.

With learned constraint: ___ acting ___ acting

Full model: I will give the movie 10 out of 10.

Hu, Yang, et al., NeurIPS 2018
Summary So Far

- **Limitations**: We considered very simple forms of domain knowledge: relational, logical, simple constraints
- **Human Knowledge**: abstract, fuzzy, build on high-level concepts
  - e.g. dogs have 4 legs

How do we encode this knowledge and how do we efficiently integrate this into deep learning models

![Figure 1: Example of how semantic knowledge about the world aids classification.](image)

Marino et al., CVPR 2017
MineRL
Towards Sample Efficient Reinforcement Learning

William H. Guss*, Brandon Houghton*, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso and Ruslan Salakhutdinov
The growing problem of sample inefficiency in RL

- The **number of environment samples** to train policies on domains of increasing complexity is growing exponentially

*AlexNet to AlphaGo Zero: A 300,000x Increase in Compute*

*Dario Amodei & Danny Hernandez* Open AI 2019.
The growing problem of sample inefficiency in RL

- The **number of environment samples** to train policies on domains of increasing complexity is growing exponentially.

- Training **complex policies** in real-world environments is quickly becoming **intractable**, without significant infrastructure.

Levine et. al. 2016
Demonstration as an Answer to Sample Inefficiency

- The number of samples required can be drastically reduced using expert demonstrations.

- No open, large-scale dataset of demonstrations across a variety of open/closed world tasks exists.
We have created one of the largest imitation learning datasets with over 60 million frames of recorded human player data across 6+ complex tasks in Minecraft.
MineRL: Why Minecraft?

- Open-world, infinite/procedurally generated
- Sparse & dense rewards
- Many innate task hierarchies and subgoals
- Encompasses many of problems we must solve as we approach the problem of general AI.

A glimpse into the Minecraft item hierarchy
MineRL: Dataset Details

- Consists of over 500+ hours of human demonstrations over 1000+ unique player sessions.

- Rich set of annotations including: subtask completion, rewards, player meta-data, gamestate.

- Rerenderable! We record game-state not just player-pixels

Plots of XY positions of players in several tasks (diversity & rich annotations)
MineRL: Hierarchality of Data

- Players complete sparsely rewarded tasks **following a specific task hierarchy/dependence graph.**

- Many ways to obtain an item, but data exhibits the existence of **canonical pathways.**
MineRL: Expert demonstrations help

- On the **Navigate** task, using the MineRL-v0 dataset helps **drastically reduce** the number of samples for standard algorithms.

- However, **better algorithms still need to be developed**, especially for the long-term, hierarchical tasks exhibited in Minecraft.
MineRL: NeurIPS 2019 Competition

- To foster research in this direction, we are hosting the **MineRL Competition on Sample Efficient Reinforcement Learning** at NeurIPS 2019!

- Competitors must learn to obtain a diamond in under 4-days of training.

https://www.youtube.com/watch?v=KFMuI4TfC7c
MineRL: Get started now!

http://minerl.io/
Thank you