DSC190: Machine Learning with Few Labels

Knowledge driven learning

Zhiting Hu

Lecture 12, November 2, 2021

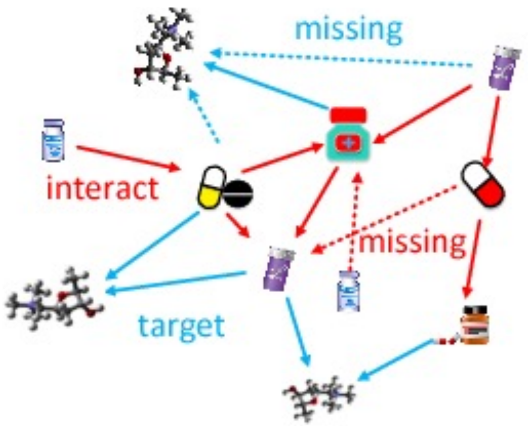
UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Outline

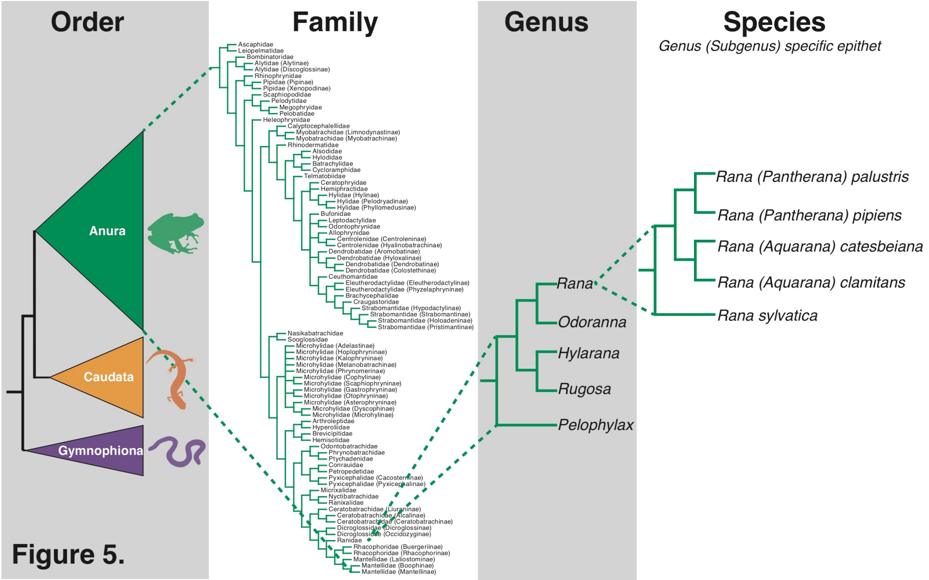
- Knowledge-driven learning (65mins)

- 1 Paper presentations (15 mins)
  - **Kejin Wu:** Variational Inference with Normalizing Flows

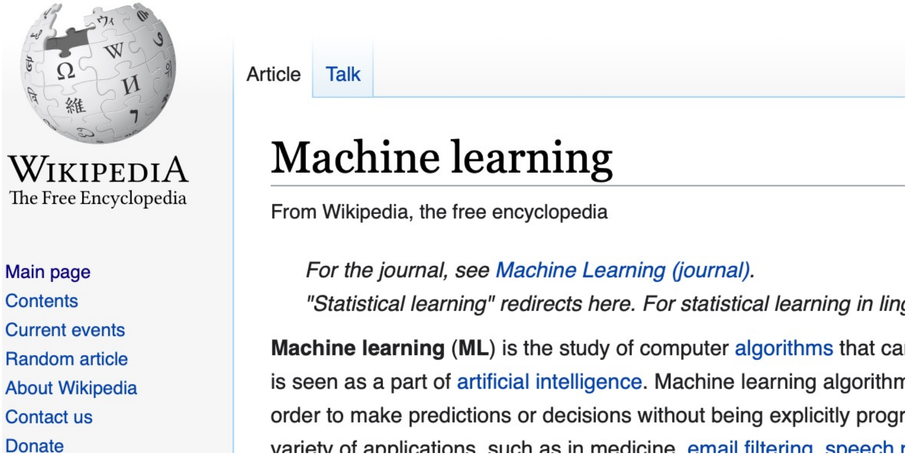
Structured knowledge



Knowledge graphs



Taxonomy



Encyclopaedia

Rules:

- *"Every part of speech sequence should have a verb"*
- *"Type-2 diabetes is 90% more common than type-1"*

Machine Learning, esp., deep learning

- Heavily rely on massive labeled data
- Uninterpretable
- Hard to encode human intention and domain knowledge

How Humans Learn

- Learn from **concrete** examples (as deep neural networks do)
- Learn from **abstract** knowledge (definitions, logic rules, etc) [Minksy 1980; Lake et al., 2015]

How Humans Learn

- Learn from **concrete** examples (as deep neural networks do)
- Learn from **abstract** knowledge (definitions, logic rules, etc) [Minksy 1980; Lake et al., 2015]

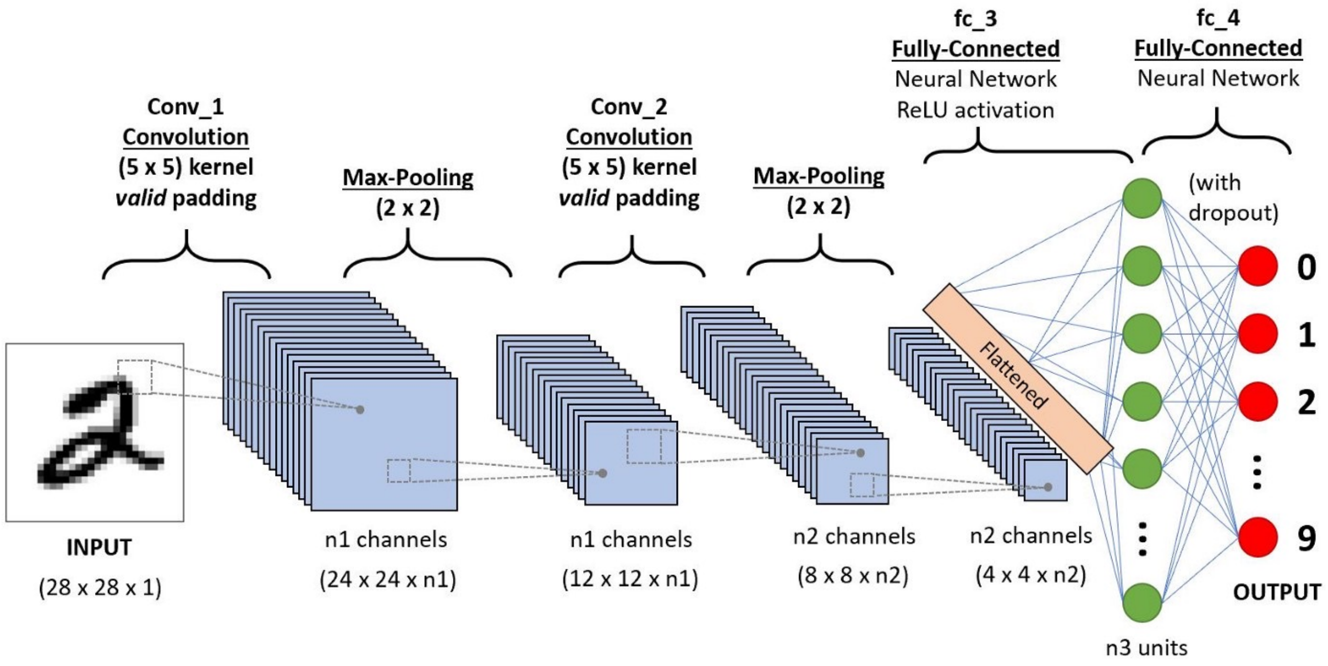
Past tense of verb

Examples:				*Rule:*
add	→	added	***V.S.***	regular verbs –d/-ed
accept	→	accepted		
ignore	→	ignored		
end	→	ended		
block	→	blocked		
love	→	loved		
…				

https://www.technologyreview.com/s/544606/can-this-man-make-aimore-human

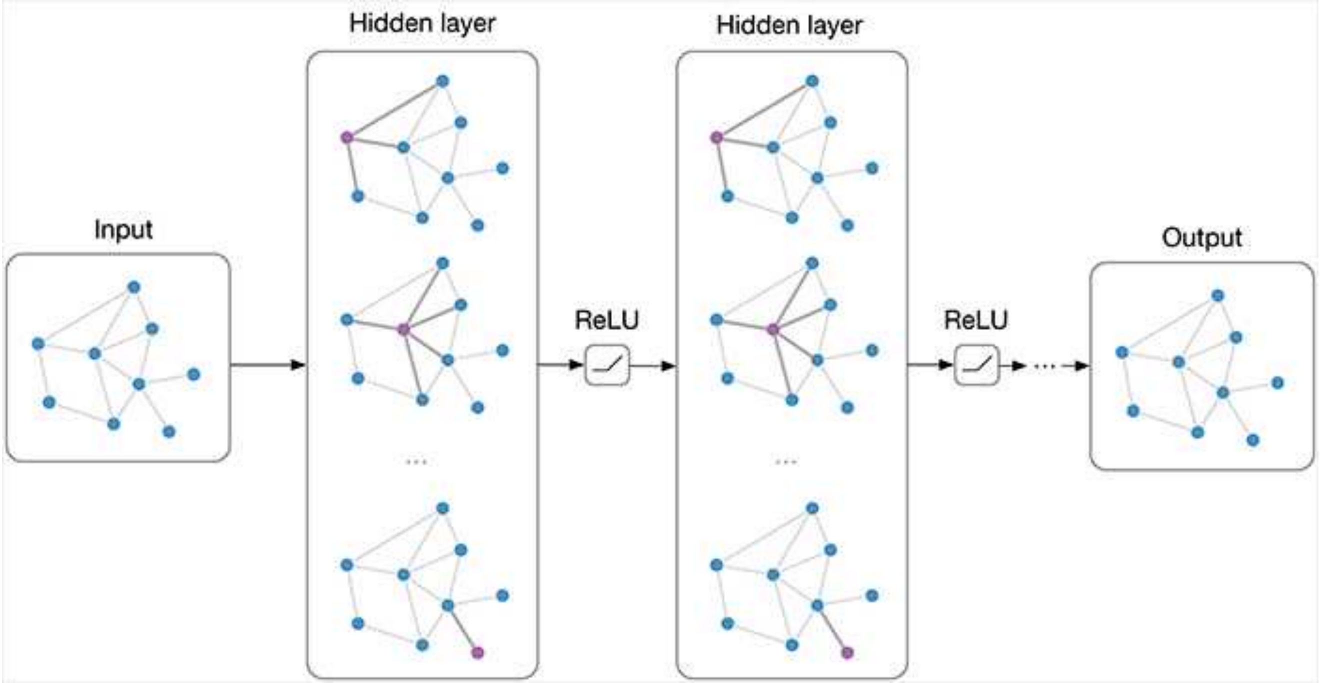
Integrating structured knowledge with machine learning

- Two general ways:
  - **Bake structured knowledge into specifically-designed neural architecture** ("inductive bias")
    - E.g., Convolutional networks (ConvNets): translation-invariance of image



Integrating structured knowledge with machine learning

- Two general ways:
  - **Bake structured knowledge into specifically-designed neural architecture** ("inductive bias")
    - E.g., Convolutional networks (ConvNets): translation-invariance of image
    - E.g., graph neural networks



Integrating structured knowledge with machine learning

- Two general ways:
  - **Bake structured knowledge into specifically-designed neural architecture** ("inductive bias")
    - E.g., Convolutional networks (ConvNets): translation-invariance of image
    - E.g., graph neural networks

The Bitter Lesson

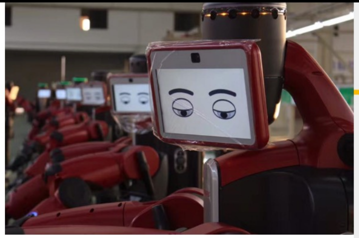
Rich Sutton

March 13, 2019

The biggest lesson that can be read from 70 years of AI research is that gene
The ultimate reason for this is Moore's law, or rather its generalization of co
as if the computation available to the agent were constant (in which case lev
slightly longer time than a typical research project, massively more comput
term, researchers seek to leverage their human knowledge of the domain, bu
run counter to each other, but in practice they tend to. Time spent on one is
or the other. And the human-knowledge approach tends to complicate meth
computation. There were many examples of AI researchers' belated learnin

http://www.incompleteideas.net/IncIdeas/BitterLesson.html

RODNEY BROOKS *Robots, AI, and other stuff* BLOG MIT ROBUST.AI



Search ...

POST: A BETTER LESSON

MARCH 19, 2019 — REVIEWS

A Better Lesson

rodneybrooks.com/a-better-lesson/

Just last week Rich Sutton published a very short blog post titled The Bitter Lesson. I'm going to try to keep this review shorter than his post. Sutton is well known for his long and sustained contributions to reinforcement learning.

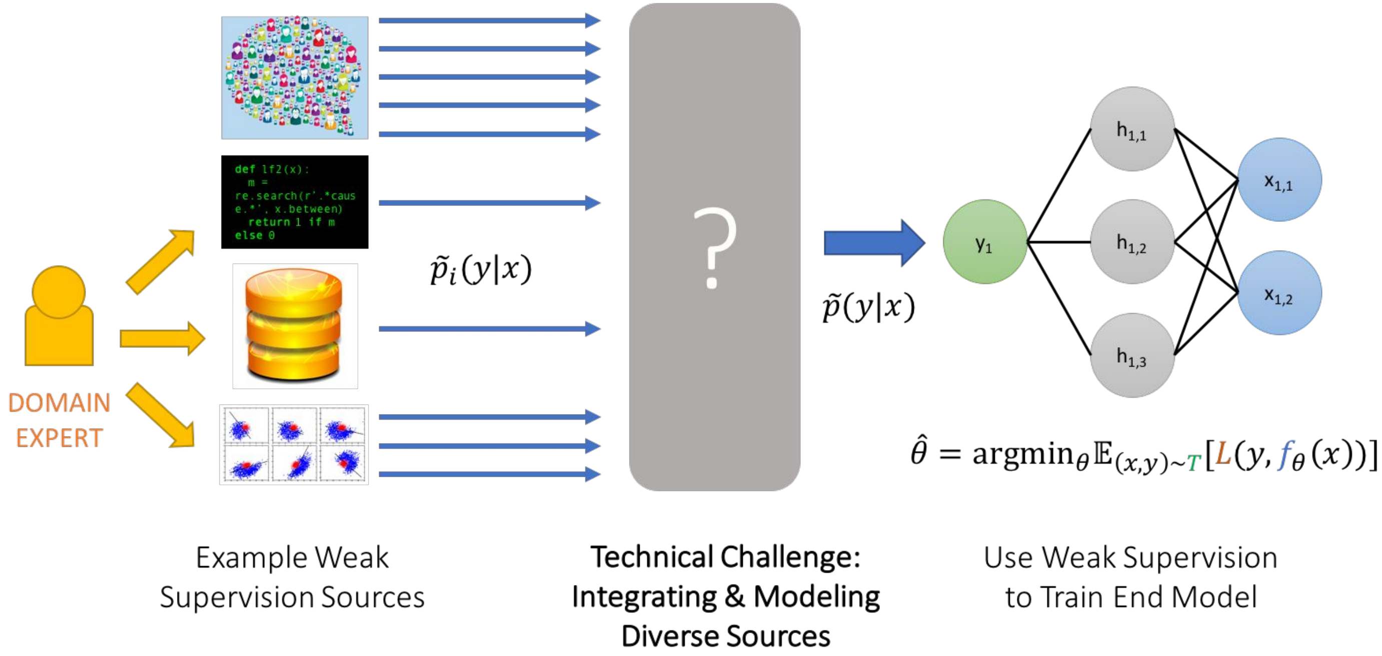
https://rodneybrooks.com/a-better-lesson/

Integrating structured knowledge with machine learning

- Two general ways:
  - **Bake structured knowledge into specifically-designed neural architecture** ("inductive bias")
    - E.g., Convolutional networks (ConvNets): translation-invariance of image
    - E.g., graph neural networks
  - **Integrating knowledge through learning:**
    - Loss and/or constraints defined by the structured knowledge
    - Model-agnostic
    - E.g., Weak supervision
    - E.g., Posterior regularization
    - E.g., Integer linear programming (ILP)

Recap: Weakly supervised learning

- Converts knowledge into weak-supervision labels
- Learn with supervised learning methods



Source: A. Ratner et. al https://dawn.cs.stanford.edu/2017/07/16/weak-supervision/

Recap: Posterior regularization

- Knowledge as constraints

$$\min_{q,\boldsymbol{\xi}} \quad -\mathrm{H}\left(q(\boldsymbol{z})\right) - \mathbb{E}_{q(\boldsymbol{z})}\left[\log p(\boldsymbol{x}^*|\boldsymbol{z})\pi(\boldsymbol{z})\right] + \sum_i \xi_i$$

$$s.t. \;\; \mathbb{E}_q\left[T(\boldsymbol{x}^*, \boldsymbol{z})\right] \leq \boldsymbol{\xi}$$

$$\boldsymbol{\xi} \geq 0,$$

Knowledge as constraints

- Consider a statistical model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$
  - Conditional model, $p_\theta(\boldsymbol{x}|\ inputs)$
  - Generative model, e.g., $\boldsymbol{x}$ is an image
  - Discriminative model, e.g., x is a sentence label

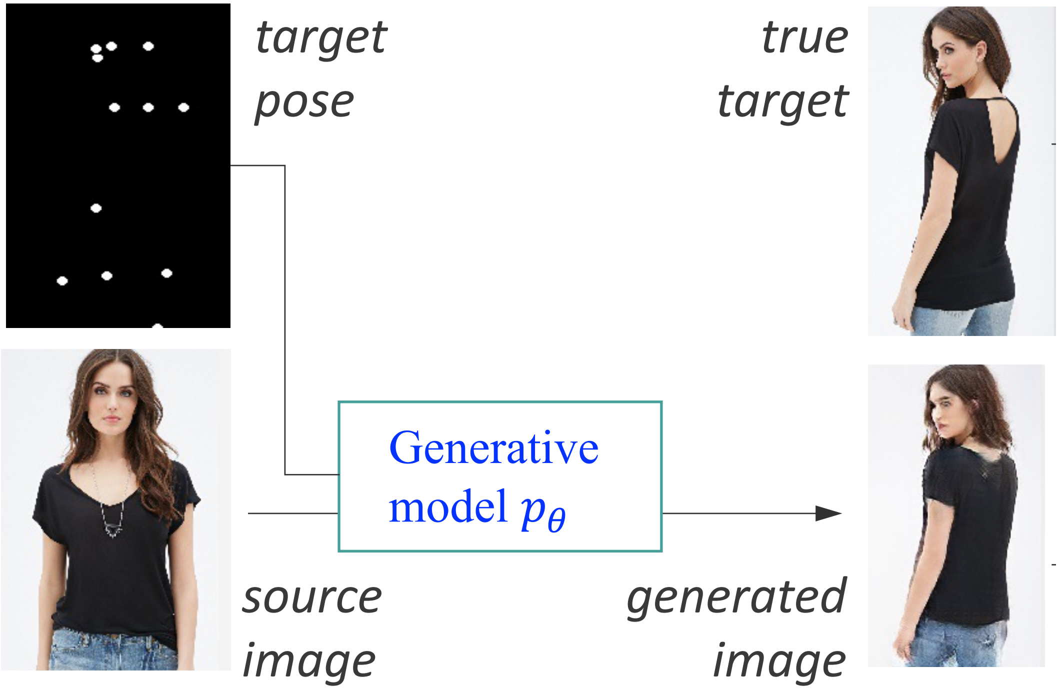
Knowledge as constraints

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

Knowledge as constraints

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

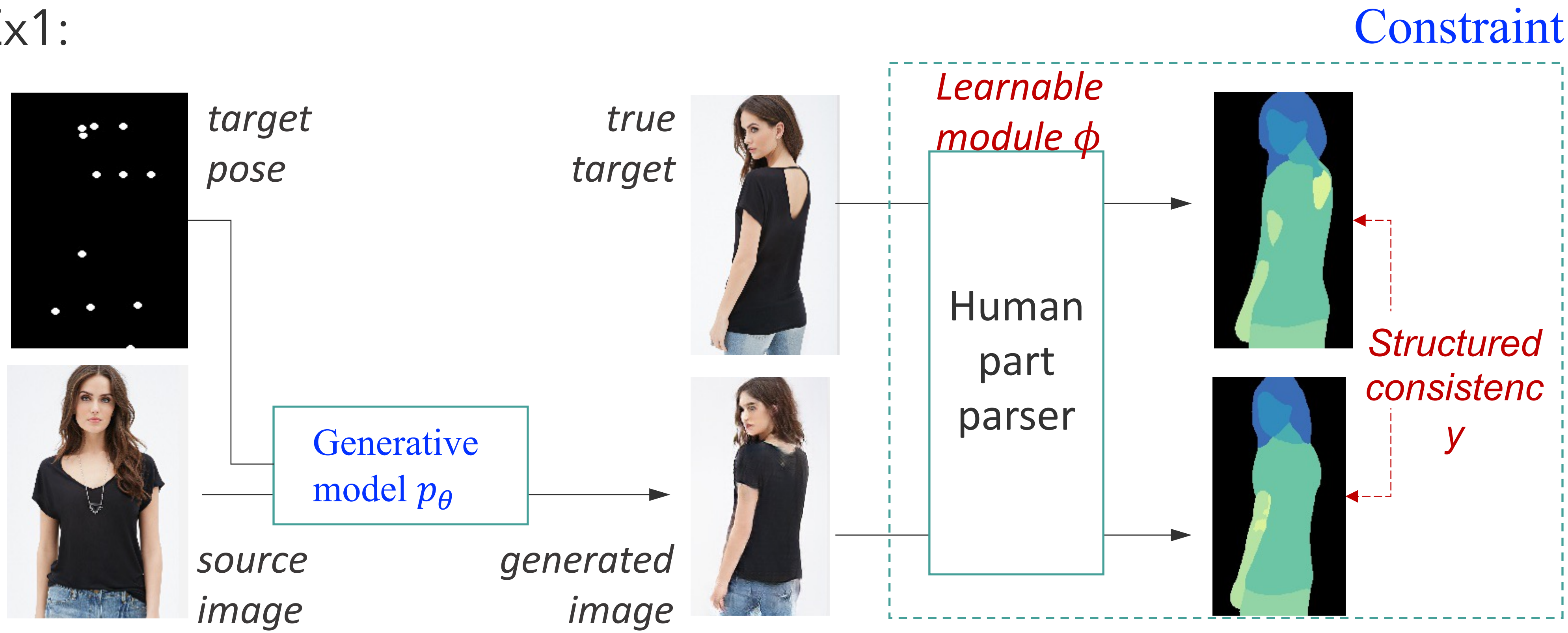
Ex1:



Knowledge as constraints

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

Ex1:



Knowledge as constraints

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

Ex2:

- Sentiment classification
  - "This was a terrific movie, but the director could have done better"

- Logical Rules:
  - Sentence S with structure A-but-B => sentiment of *B* dominates

Knowledge as constraints

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

- One way to impose the constraint is to maximize: $\mathbb{E}_{p_\theta}[f(\boldsymbol{x})]$

Knowledge as constraints

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

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

$$\min_\theta \ \mathcal{L}(\boldsymbol{\theta}) - \alpha \, \mathbb{E}_{p_\theta}[f(\boldsymbol{x})]$$

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

Regularization: imposing constraints (difficult to compute)

Knowledge as constraints

- Consider a statistical model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$
- Consider a constraint function $f(\boldsymbol{x}) \in \mathbb{R}$

$$\min_\theta \; \mathcal{L}(\boldsymbol{\theta}) - \alpha\, \mathbb{E}_{p_\theta}[f(\boldsymbol{x})]$$

Knowledge as constraints

- Consider a statistical model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$
- Consider a constraint function $f(\boldsymbol{x}) \in \mathbb{R}$

$$\min_\theta \ \mathcal{L}(\boldsymbol{\theta}) - \alpha \boxed{\mathbb{E}_{p_\theta}[f(\boldsymbol{x})]}$$

$$\mathcal{L}(\boldsymbol{\theta}, q) = \overbrace{\mathrm{KL}(q(\boldsymbol{x}) || \, p_\theta(\boldsymbol{x})) - \lambda \, \mathbb{E}_q[f(\boldsymbol{x})]}$$

Slide courtesy: Ruslan Salakhutdinov

Knowledge as constraints

- Consider a statistical model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$
- Consider a constraint function $f(\boldsymbol{x}) \in \mathbb{R}$

$$\min_\theta \; \mathcal{L}(\boldsymbol{\theta}) - \alpha \, \underbrace{\boxed{\mathbb{E}_{p_\theta}[f(\boldsymbol{x})]}}_{}$$

$$\mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{x})) - \lambda \, \mathbb{E}_q[f(\boldsymbol{x})]$$

Posterior Regularization
[Ganchev et al., 2010]

- Introduce variational distribution q
    - Impose constraint on q
    - Encourage q to stay close to p

Knowledge as constraints

- Consider a statistical model $\boldsymbol{x} \sim p_\theta(\boldsymbol{x})$
- Consider a constraint function $f(\boldsymbol{x}) \in \mathbb{R}$

$$\min_\theta \; \mathcal{L}(\boldsymbol{\theta}) - \alpha \, \underbrace{\boxed{\mathbb{E}_{p_\theta}[f(\boldsymbol{x})]}}_{}$$

$$\mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{x})) - \lambda \, \mathbb{E}_q[f(\boldsymbol{x})]$$

Posterior Regularization
[Ganchev et al., 2010]

- Introduce variational distribution q
  - Impose constraint on q
  - Encourage q to stay close to p
- Objective

$$\min_{\theta,q} \; \mathcal{L}(\boldsymbol{\theta}) + \alpha \, \mathcal{L}(\boldsymbol{\theta}, q)$$

Knowledge as constraints

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

$$\mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x}) || \, p_{\theta}(\boldsymbol{x})) - \lambda\, \mathbb{E}_q[f(\boldsymbol{x})]$$

- EM-style procedure for solving the problem

Knowledge as constraints

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

$$\mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x}) || \, p_\theta(\boldsymbol{x})) - \lambda\, \mathbb{E}_q[f(\boldsymbol{x})]$$

- EM-style procedure for solving the problem
  - E-step

$$q^*(\boldsymbol{x}) = p_\theta(\boldsymbol{x}) \exp\{\, \lambda f(\boldsymbol{x}) \,\}/Z$$

Knowledge as constraints

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

$$\mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{x})) - \lambda\,\mathbb{E}_q[f(\boldsymbol{x})]$$

- EM-style procedure for solving the problem
  - E-step

$$q^*(\boldsymbol{x}) = p_\theta(\boldsymbol{x}) \exp\{\,\lambda f(\boldsymbol{x})\,\}/Z$$

Higher value -- higher probability under q – "soft constraint"

Knowledge as constraints

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

$$\mathcal{L}(\boldsymbol{\theta}, q) = \mathrm{KL}(q(\boldsymbol{x})\,||\,p_\theta(\boldsymbol{x})) - \lambda\,\mathbb{E}_q[f(\boldsymbol{x})]$$

- EM-style procedure for solving the problem
  - E-step

$$q^*(\boldsymbol{x}) = p_\theta(\boldsymbol{x})\exp\{\,\lambda f(\boldsymbol{x})\,\}/Z$$

  - M-step

Higher value -- higher probability under q – "soft constraint"

$$\min_\theta \mathcal{L}(\boldsymbol{\theta}) - \mathbb{E}_{q^*}[\log p_\theta(\boldsymbol{x})]$$

Logical Rule Constraints

- Consider a supervised learning: $p_\theta(y|\boldsymbol{x})$
- Input-Target space (X, Y)

[Hu et al., 2016]

Logical Rule Constraints

- Consider a supervised learning: $p_\theta(y|\boldsymbol{x})$
- Input-Target space (X, Y)
- First-order logic rules: (r, λ)

[Hu et al., 2016]

Logical Rule Constraints

- Consider a supervised learning: $p_\theta(y|\boldsymbol{x})$
- Input-Target space (X, Y)
- First-order logic rules: (r, λ)
  - $r(X, Y) \in [0, 1]$, could be soft
  - λ is the confidence level of the rule

[Hu et al., 2016]

Logical Rule Constraints

- Consider a supervised learning: $p_\theta(y|\boldsymbol{x})$
- Input-Target space (X, Y)
- First-order logic rules: (r, λ)
  - $r(X, Y) \in [0, 1]$, could be soft
  - λ is the confidence level of the rule

- Given l rules:
  - E-step:

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

  - M-step:

$$\min_\theta \mathcal{L}(\boldsymbol{\theta}) - \mathbb{E}_{q^*}[\log p_\theta(y|\boldsymbol{x})]$$

[Hu et al., 2016]

Logical Rule Constraints

- Consider a supervised learning: $p_\theta(y|\boldsymbol{x})$
- Input-Target space (X, Y)
- First-order logic rules: (r, λ)
  - $r(X, Y) \in [0, 1]$, could be soft
  - λ is the confidence level of the rule

- Given l rules:
  - E-step:

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

  - M-step:

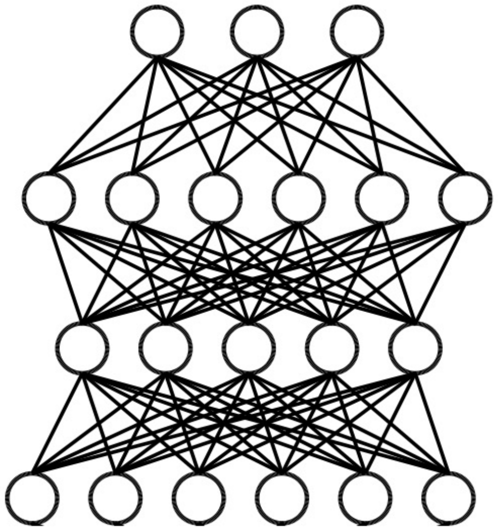
$$\min_\theta \mathcal{L}(\boldsymbol{\theta}) - \mathbb{E}_{q^*}[\log p_\theta(y|\boldsymbol{x})]$$

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

[Hu et al., 2016]

Knowledge Distillation

$p_\theta(y|\boldsymbol{x})$

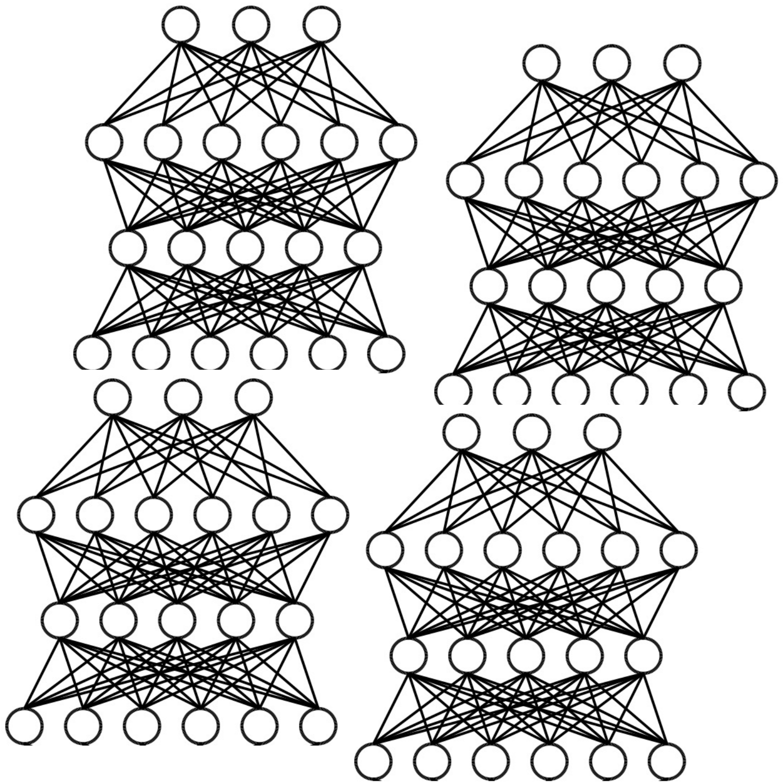


Student

[Hinton et al., 2015; Bucilu et al., 2006]

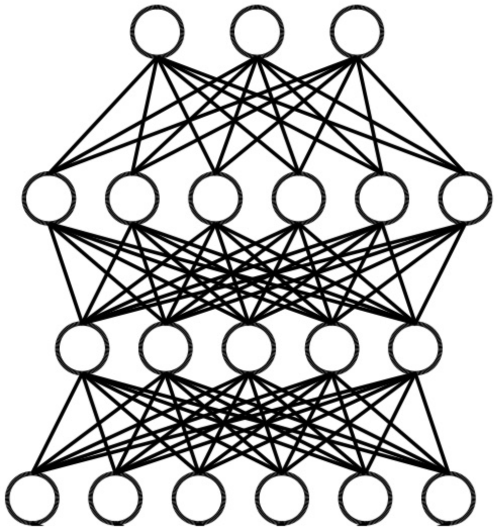
Knowledge Distillation

$q(y|\boldsymbol{x})$



Teacher
(Ensemble)

$p_\theta(y|\boldsymbol{x})$



Student

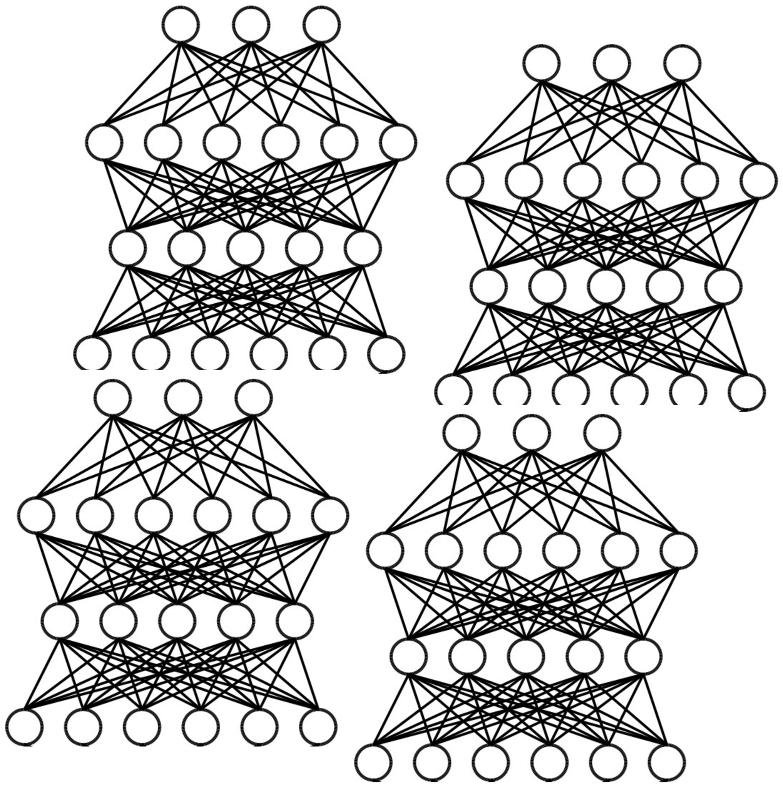
[Hinton et al., 2015; Bucilu et al., 2006]

Knowledge Distillation

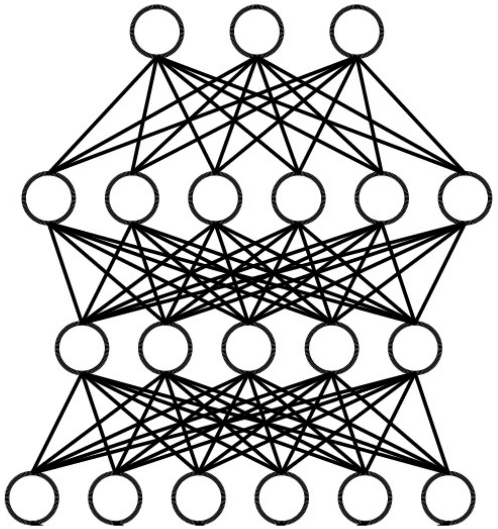
$q(y|\boldsymbol{x})$

Match soft predictions of the teacher network and student network

$p_\theta(y|\boldsymbol{x})$



Teacher (Ensemble)



Student

[Hinton et al., 2015; Bucilu et al., 2006]

Rule Knowledge Distillation

$$\min_\theta \mathcal{L}(\boldsymbol{\theta}) - \mathbb{E}_{q^*}[\log p_\theta(y|\boldsymbol{x})]$$

- Neural network $p_\theta(y|\boldsymbol{x})$
- Train to imitate the outputs of the rule-regularized teacher network

[Hu et al., 2016]

Rule Knowledge Distillation

$$\min_\theta \boxed{\mathcal{L}(\boldsymbol{\theta})} - \mathbb{E}_{q^*}[\log p_\theta(y|\boldsymbol{x})]$$

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

true hard label

soft prediction of $p_\theta(y|x)$

$$\boldsymbol{\theta}^{(t+1)} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(\boldsymbol{y}_n, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n))$$

[Hu et al., 2016]

Rule Knowledge Distillation

$$\min_\theta \mathcal{L}(\boldsymbol{\theta}) - \mathbb{E}_{q^*}[\log p_\theta(y|\boldsymbol{x})]$$

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

true hard label

soft prediction of $p_\theta(y|x)$

$$\boldsymbol{\theta}^{(t+1)} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \ell(\boldsymbol{y}_n, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n))$$

$$\ell(\boldsymbol{s}_n^{(t)}, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n)),$$

soft prediction of the teacher network

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

[Hu et al., 2016]

Rule Knowledge Distillation

$$\min_\theta \boxed{\mathcal{L}(\boldsymbol{\theta})} - \boxed{\mathbb{E}_{q^*}[\log p_\theta(y|\boldsymbol{x})]}$$

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

true hard label

soft prediction of $p_\theta(y|x)$

$$\boldsymbol{\theta}^{(t+1)} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{n=1}^{N}(1-\pi)\ell(\boldsymbol{y}_n, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n))$$

$$+\pi\ell(\boldsymbol{s}_n^{(t)}, \boldsymbol{\sigma}_\theta(\boldsymbol{x}_n)),$$

soft prediction of the teacher network

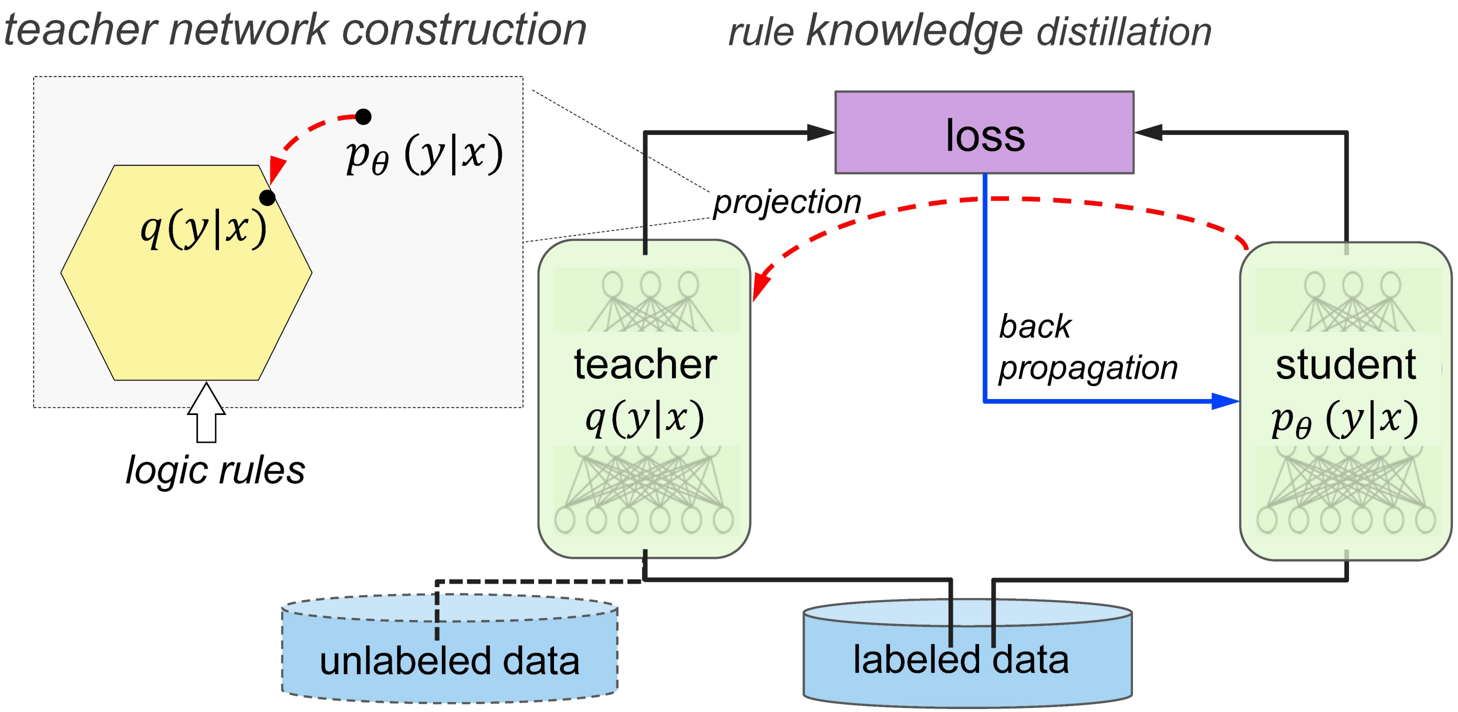
balancing parameter

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

[Hu et al., 2016]

Rule Knowledge Distillation

- Neural network $p_\theta(y|\boldsymbol{x})$
- At each iteration
  - Construct a teacher network with "soft constraint"
  - Train DNN to emulate the teacher network



Learning Rules / Constraints

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

- Learn the confidence value λ_l for each logical rule [Hu et al., 2016b]

Learning Rules / Constraints

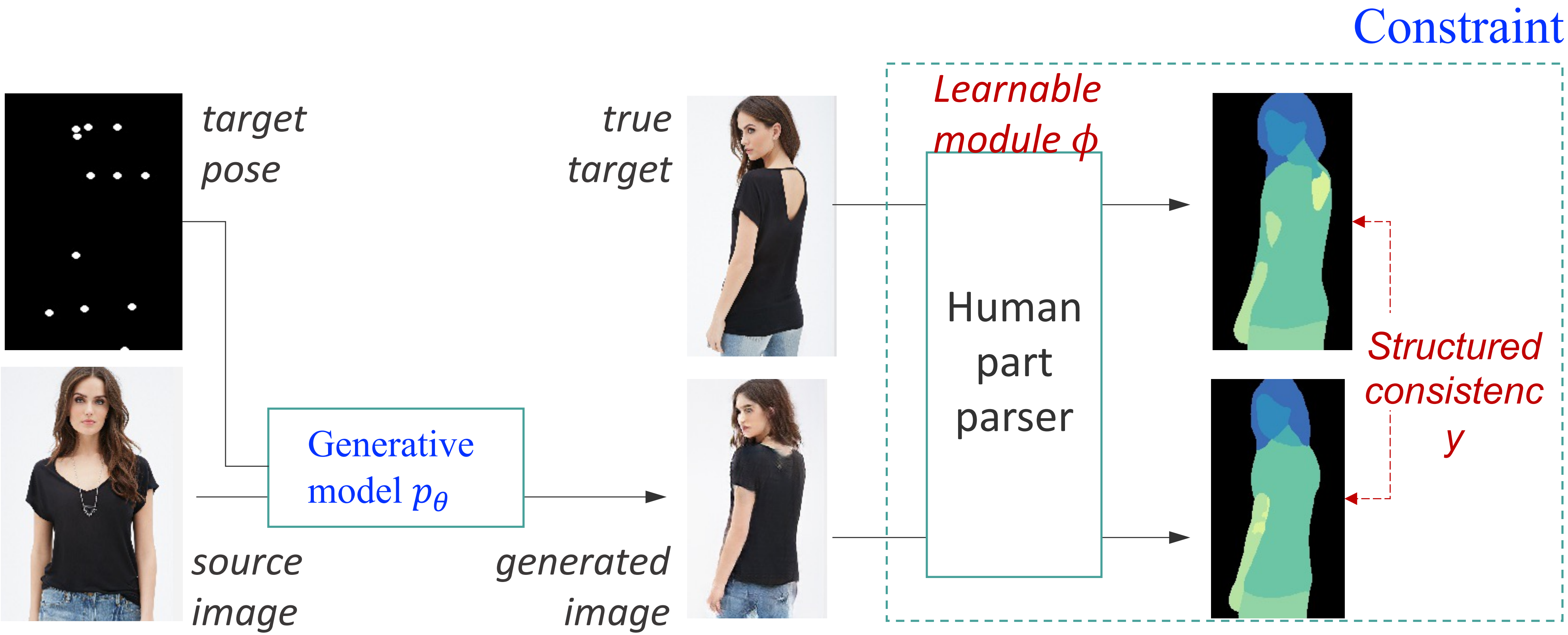
$$q^*(y|\boldsymbol{x}) = p_\theta(y|\boldsymbol{x}) \exp\left\{\sum_l \lambda_l r_l(y,\boldsymbol{x})\right\}/Z$$

- Learn the confidence value λ_l for each logical rule [Hu et al., 2016b]

$$q^*(\boldsymbol{x}) = p_\theta(\boldsymbol{x}) \exp\{\,\lambda f_\phi(\boldsymbol{x})\,\}/Z$$

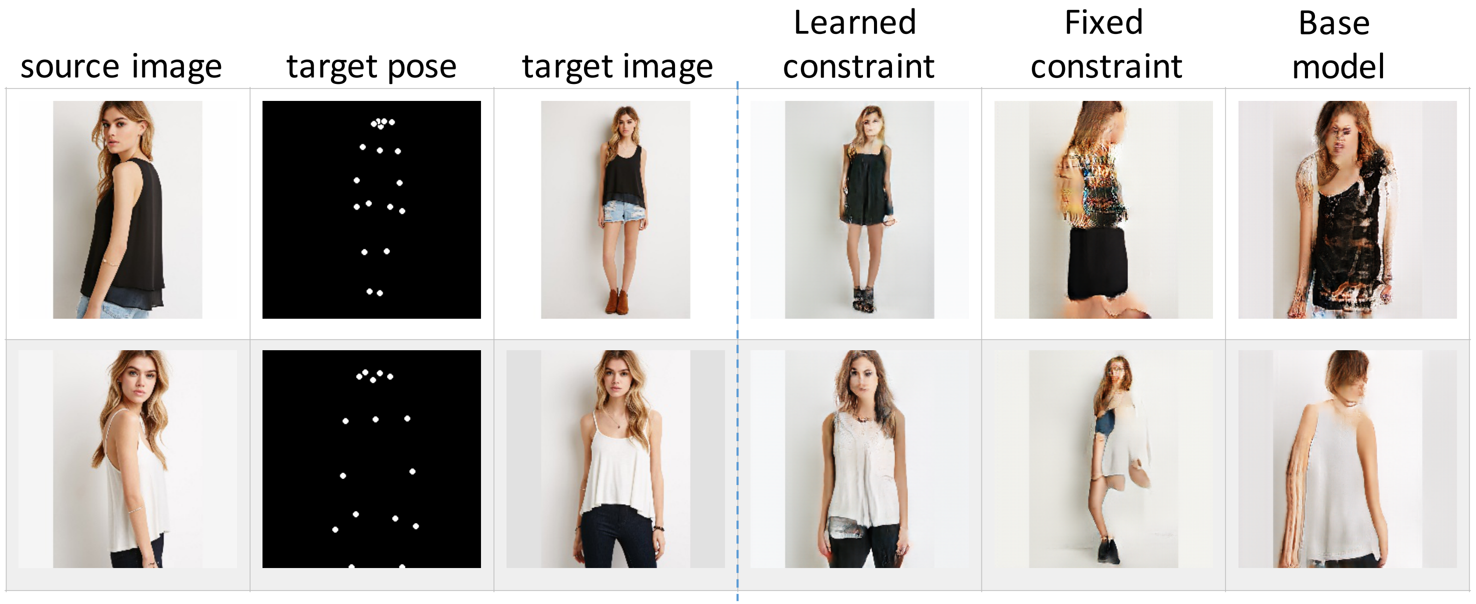
- More generally, optimize parameters of the constraint $f_\phi(\boldsymbol{x})$ [Hu et al., 2018]
  - Treat $f_\phi(\boldsymbol{x})$ as an extrinsic reward function
  - Use MaxEnt Inverse Reinforcement Learning to learn the "reward"

Pose-conditional Human Image Generation



[Hu et al., 2018]

Pose-conditional Human Image Generation



Samples generated by the models

	Method	SSIM	Human
1	Ma et al. [38]	0.614	—
2	Pumarola et al. [44]	0.747	—
3	Ma et al. [37]	0.762	—
4	Base model	0.676	0.03
5	With fixed constraint	0.679	0.12
6	With learned constraint	**0.727**	**0.77**

Quantitative and Human Evaluation

[Hu et al., 2018]

Knowledge as constraints: Integer Linear Programming (ILP)

- An integer linear program (ILP) is an optimization problem of the form

$$\max_{\boldsymbol{z}} \quad \boldsymbol{a} \cdot \boldsymbol{z} \quad \text{subj. to} \quad \text{linear constraints on } \boldsymbol{z}$$

$$\boldsymbol{z} \in \mathbb{Z}^n \text{ (integers)}$$

  - For a fixed vector a

- Well-engineered solvers exist
  - e.g, Gurobi
  - Useful for prototyping
  - But general not as efficient as dynamic programming

Ex: Casting sequence labeling as an ILP

- Sequence labeling, e.g., named entity recognition
  - names of people, organizations, locations

Ex: "Brendan Iribe, a co-founder of Oculus VR and a prominent University of Maryland donor, is leaving Facebook four years after it purchased his company."

- BIO labeling scheme for NER

x = [Brendan, Iribe, ",", a, co-founder, of, Oculus, VR, and, a, prominent, University, of, Maryland, donor, ",", is, leaving, Facebook, four, years, after, it, purchased, his, company, "."]

y = [B-PER, I-PER, O, O, O, O, B-ORG, I-ORG, O, O, O,B-ORG, I-ORG, I-ORG, O, O, O,B-ORG, O, O, O, O, O, O, O, O]

Ex: Casting sequence labeling as an ILP

Many NLP tasks can be framed as sequence labeling

x = [Brendan, Iribe, “,”, a, co-founder, of, Oculus, VR, and, a, prominent, University, of, Maryland, donor, “,”, is, leaving, Facebook, four, years, after, it, purchased, his, company, “.”]

y = [B-PER, I-PER, O, O, O, O, B-ORG, I-ORG, O, O, O,B-ORG, I-ORG, I-ORG, O, O, O,B-ORG, O, O, O, O, O, O, O, O]

“BIO” labeling scheme for named entity recognition

Ex: Casting sequence labeling as an ILP

- Step 1: Define variables z as binary indicator variables which encode an output sequence y

$$z_{l,k',k} = \mathbf{1}[\text{label } l \text{ is } k \text{ and label } l-1 \text{ is } k']$$

- Step 2: Construct the linear objective function

$$a_{l,k',k} = \boldsymbol{w} \cdot \phi_l(\boldsymbol{x}, \langle \ldots, k', k \rangle)$$

Ex: Casting sequence labeling as an ILP

- Step 3: Define constraints to ensure a well-formed solution
  - Z's should be binary: for all l, k', k

$$z_{l,k',k} \in \{0,1\}$$

  - For a given position l, there is exactly one active z

$$\sum_k \sum_{k'} z_{l,k',k} = 1 \text{ for all } l$$

  - The z's are internally consistent

$$\sum_{k'} z_{l,k',k} = \sum_{k''} z_{l+1,k,k''} \text{ for all } l, k$$

Key Takeaways

- Two general ways of integrating structured knowledge with ML:
  - Model architecture (inductive bias)
  - Integrating knowledge through learning (loss, constraints)
    - Weak supervision
    - Posterior regularization
    - Integer linear programming (ILP)\
    - Others ..

Questions?