Efficient Deep Learning w. Humans in the Loop

Zachary Chase Lipton

email: zlipton@cmu.edu  shenanigans: @zacharylipton

collaborators: Anima Anandkumar, Yanyao Shen, Hyokun Yun, Ashish Khetan, Xiujun Li, Lihong Li, Jianfeng Gao, Li Deng, Peiyun Hu, Aditya Siddhant, David Lowell, Byron Wallace
One model to rule them all?

- With **large datasets**, massive computation, DL achieves high **accuracy**
- Success stories
  - Vision
  - Speech
  - NLP
  - RecSys
CNNs
RNNs

POS tagging

machine translation

medical diagnosis

product review generation

Poured from 12oz bottle into half-liter Pilsner Urquell branded pilsner glass. Appearance: Pours a cloudy golden-orange color with a small, quickly dissipating white head that leaves a bit of lace behind. Smell: Smells HEAVILY of citrus. By heavily, I mean that this smells like kitchen cleaner with added wheat. Taste: Tastes heavily of citrus—lemon, lime, and orange with a hint of wheat at the end. Mouthfeel: Thin, with a bit too much carbonation. Refreshing. Drinkability: If I wanted lemonade, then I would have bought that.

(a) Real Review

Poured from a 12oz bottle into a 16oz Samuel Adams Perfect Pint glass. Appearance: Very pale golden color with a thin, white head that leaves little lacing. Smell: Very mild and inoffensive aromas of citrus. Taste: Starts with the same tastes of the citrus and fruit flavors of orange and lemon and the orange taste is all there. There is a little bit of wheat that is pretty weak, but it is sort of harsh (in a good way) and ends with a slightly bitter aftertaste. Mouthfeel: Light body with a little alcohol burn. Finish is slightly dry with some lingering spice. Drinkability: A decent beer, but not great. I don’t think I would rate this anytime soon as it says that there are other Belgian beers out there, but this is a good choice for a warm day when it’s always available in the North Coast Brewing Company party.

(b) Synthetic review
Still, Most Big Problems are Unsolved

- DL requires **BIG DATA**, often prohibitively expensive to collect
- Supervised models make **predictions** but we want to take **actions**
- Deep reinforcement learning brittle, **even higher sample complexity**
- Supervised learning **doesn’t know why** a label applies
- In general, these models break under **distribution shift**
- Modeling causal mechanisms sounds right, but we **lack tools**
Outline

• Data-Efficient Deep Learning
  • Efficient exploration with BBQ Nets (AAAI 2018)
  • Deep Active Learning for Named Entity Recognition (ICLR 2018)
  • Learning From Noisy, Singly Labeled Data (ICLR 2018)
    https://arxiv.org/abs/1712.04577

• Plausible interactive learning
  • Active Learning with Partial Feedback https://arxiv.org/abs/1802.07427
  • Active Learning w/o the Crystal Ball (under review)
  • How transferable are the datasets collected by active learners? (in prep)
Efficient Exploration for Dialogue Policy Learning w. BBQ-Networks

Zachary C. Lipton, Xiujun Li, Lihong Li, Jiangeng Gao, Faisal Ahmed, Li Deng

Chatbots

InfoBot

Task Completion

Chit-Chat
Typical dialogue system architecture

- Language Understanding
- State Tracker
- Dialog Policy
- Natural Language Generation
Dialogue-Act Representations

• **Semantic representation of dialog utterances:**
  Agent: greeting()
  User:  request(ticket, numberofpeople=2)
  Agent: request(city)
  User:  inform(city=Seattle)
  Agent: request(genre)
  User:  inform(action)
  Agent: inform(multiplechoice={...})
  User:  inform(moviename=Our Kind of Traitor)
  Agent: inform(taskcomplete, theater=Cinemark LnCln Sq)
  User:  thanks()

• **Mapping from-to NLP handled by LU and NLG components**
Deep Reinforcement Learning

Agent  Environment  Actions  Rewards
Deep Q-Networks

For problems with many states and actions, must approximate Q function
DQNs are awesome at games ....

... but take squillions of interactions to train.
RL for Dialogue Policy Learning
RL agents balance exploitation vs exploration

- Standard DRL exploration strategies naive ($\epsilon$-greedy)
- DRL has high sample complexity
- Failures in dialogue have economic costs
  - Company reputation
  - Labor costs
- Primary rewards are sparse
- Want to learn policies efficiently
Thompson Sampling

• Alternative to ($\epsilon$-greedy)
• Choose each action according to probability that it’s the best
• Requires estimating uncertainty

• Conundrum:
  • Neural networks get best predictions, want to use them for estimating $Q$
  • OTOH, other approaches better-established for estimating uncertainty

• Solution:
  • Extract uncertainty estimates from neural networks
BBQ-Networks

• **At train time:**
  1. Sample weights from $q(w)$
  2. Make forward pass
  3. Generate TD target using MAP estimate from target network
  4. Update parameters with reparameterization trick

• **Action time**
  1. Sample weights from $q(w)$
  2. Choose best action according to that forward pass
Bayes-by-Backprop gives useful uncertainty estimates

Figure from *Weight Uncertainty* (Blundell et al. 2015)
Results for static (left) & domain extension (right)
Warm-starting
Deep Active Learning for Named Entity Recognition

Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, Anima Anandkumar

Active Learning

Image credit: Settles, 2010
Named Entity Recognition

around the left eye. <test>CT of the brain</test> showed no <problem>acute changes</problem>, <problem>left periorbital soft tissue swelling</problem>. <test>CT of the maxillofacial area</test> showed no <problem>facial bone fracture</problem>. <test>Echocardiogram</test> showed normal left ventricular function, <test>ejection fraction</test> estimated greater than 65%. She was set up with a skilled nursing facility, which took several days to arrange, where she was to be given <treatment>daily physical therapy</treatment> and <treatment>rehabilitation</treatment> until appropriate.
Modeling - Encoders

Word embedding

Sentence encoding

\[ w_{i}^{\text{full}} := (w_{i}^{\text{char}}, w_{i}^{\text{emb}}) \]
Tag Decoder

• Each tag conditioned on
  1. Current sentence representation
  2. Previous decoder state
  3. Previous decoder output

• Greedy decoding
  1. For OntoNotes NER wide beam gives little advantage
  2. Faster, necessary for active learning
Active learning heuristics

Normalized maximum log probability

$$\max_{y_1, \ldots, y_n} \frac{1}{n} \sum_{i=1}^{n} \log \mathbb{P}[y_i \mid y_1, \ldots, y_{n-1}, \{x_{ij}\}]$$

Bayesian active learning by disagreement (BALD)

$$f_i = 1 - \frac{\max_y | \{ m : \text{argmax}_{y'} \mathbb{P}^m [y_i = y'] = y \} |}{M}$$
Results — 25% samples, 99% performance
Problems!

• Active learning sounds **great on paper**
• But...
  1. Paints a **cartoonish picture** of annotation
  2. Hindsight is 20/20, **but not our foresight**
  3. In reality, can’t run 4 strategies & **retrospectively** pronounce a winner
  4. Can’t use full set of labels to pick architecture, hyperparameters
  5. Supervised learner **can mess up** – active learner **must be right** 1st time
Learning from Noisy Singly-Labeled Data

Ashish Khetan, Zachary C. Lipton, Anima Anandkumar

https://arxiv.org/abs/1712.04577
Classical Crowdsourcing Setup

• Redundant labeling to overcome noise
• Task: aggregate intelligently
• Naive baseline: majority vote
• Can do better with EM
• Classic algos ignore features
• Given 1 label/ex. all workers perfect!
Expectation Maximization (EM)

- Initialize expected ground truth labels by majority voting
- Repeat:
  - Estimate confusion matrix: MLE given predicted ground truth labels
  - Estimate ground-truth labels: MLE given predicted confusion matrices
Bootstrapping EM

• **Insight:** Learned model agrees with workers more when they are right

• Learning algorithm:
  1. Aggregate labels by weighted majority (no-op if singly labeled)
  2. Train model on noisy labels
  3. Use predictions to estimate worker confusion matrices
  4. Given the estimated confusion matrices, retrain model with *probability-weighted* loss function
CIFAR10 Results – Varying Redundancy

hammer-spammer workers

class-wise hammer-spammer workers
Fixed Annotation Budget (CIFAR10)
Label Once and Move On!

Probability-weighted EM: Optimal redundancy 3/5
Probability-weighted bootstrapped EM: Optimal redundancy 1
Fixed annotation budget (ImageNet): label once and move on!
Active Learning with Partial Feedback

Peiyun Hu, Zachary C. Lipton, Anima Anandkumar, Deva Ramanan

What labeling really looks like

• Crowd-workers **don’t hand us the answer** (e.g. out of 1000 classes)
• Real platforms **more primitive**
• ImageNet compiled with just **binary questions:**
• **Does image X belong to class y?**
• Cleverly used Google Image Search to filter candidates

Figure 7: **Left:** Is there a Burmese cat in the images? Six randomly sampled users have different answers. **Right:** The confidence score table for “Cat” and “Burmese cat”. More votes are needed to reach the same degree of confidence for “Burmese cat” images.
Opening the black annotation black box

• Given only binary questions, how can we most efficiently:
  1. Learn accurate classifiers?
  2. Label our datasets?

• Key ideas
  1. Exploit hierarchies among classes
  2. Actively select (example, class pairs)
Active learning with partial feedback

• **Actively** select questions (example, class) pairs

• Class can be **composite** e.g. dog, or **atomic**, e.g. pitbull

• Annotator gives **binary feedback**

• Update our model with **both the fully and partially-labeled data**

• We aim to:
  1. Produce the **best classifier on fixed budget** (# of questions)
  2. **Annotate the full dataset** with fewest # queries
Three sampling strategies

• **Expected information gain (EIG)**
  • Generalizes maximum entropy acquisition function

• **Expected decrease in potential classes (EDC)**
  • Choose questions expected to eliminate the most classes

• **Expected number of remaining classes (ERC)**
  • Choose question expected to yield most exact label

• Note, EDC and ERC rank examples differently, but rank questions the same given an example
Quantitative results
Qualitative analysis
Active Learning without the Crystal Ball
(work with Aditya Siddhant)

• Simulated active learning shows results on 1 problem, 1-2 datasets, with 1 model
• Peeks at data for hyperparameters of inherited architectures
• We look across settings to see: does consistent story emerge?
  • Surprisingly, BALD performs best across wide range of NLP problems
  • Both Dropout & Bayes-by-Backprop work
How Transferable are Active Sets Across Learners?
(w David Lowell & Byron Wallace)

• Datasets tend to have a longer shelf-life than models
• When model goes stale, will active set transfer to new models?
• Answer is dubious
• Sometimes outperforms, but often underperforms i.i.d. data
Thanks!

• **Stay in touch**
  Interested in this work? Let’s talk!

• **Contact**
  [zlipton@cmu.edu](mailto:zlipton@cmu.edu)

• **Papers**

• **Acknowledgments**
  Yanyao Shen Hyokun Yun, Ashish Khetan, Anima Anandkumar, Xiujun Li, Lihong Li, Jianfeng Gao, Li Deng, Peiyun Hu, Aditya Siddhant, David Lowell, Byron Wallace