NIPS 2016

Ni Lao

2017/1/31

NIPS 2016

Largest ML conference

Barcelona, Spain

6000 attendees

Dec 5 tutorials, posters

Dec 6,7 presentations, posters

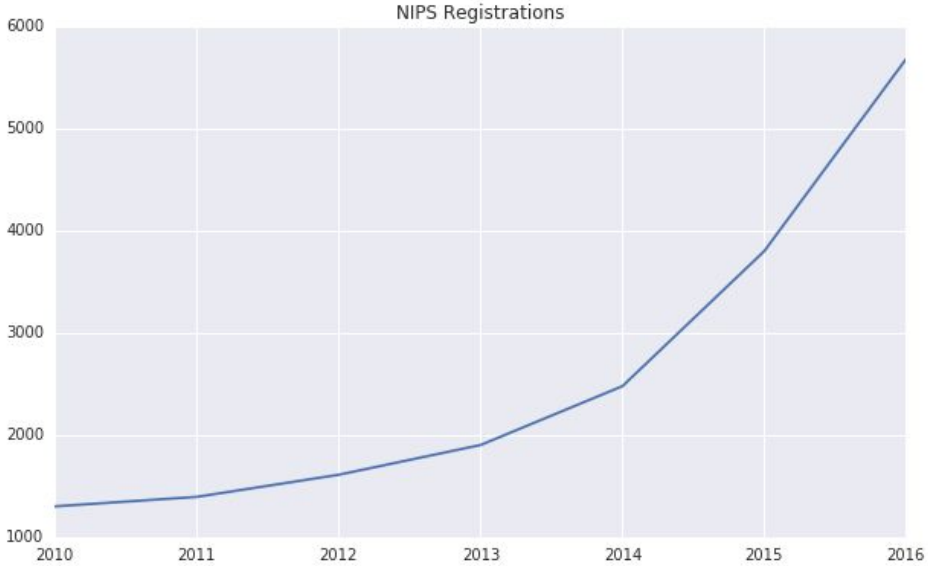
Dec 8 presentations, symposiums

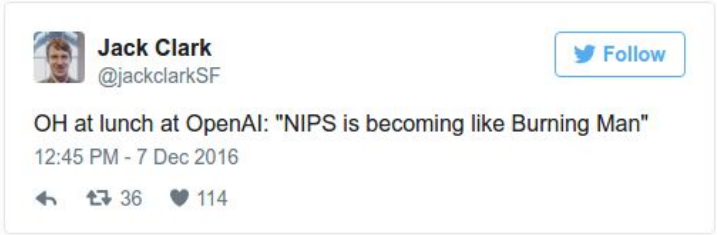
Dec 9,10 workshops

Credit http://blog.evjang.com/2017/01/nips2016.html

AI is getting popular

- A lot of industry presence
  - Facebook, Microsoft, Amazon, NVIDIA, most of Google Brain and most of DeepMind
  - Automotive, financial, e-commerce, and all kind of companies looking to grow their ML groups
  - Startup founders and CEOs of AI companies walking around





Credit http://blog.evjang.com/2017/01/nips2016.html

The venue for big announcements

- (2015) **Google** gave its introduction/tutorial on TensorFlow, released its best model on ImageNet
- (2015) **OpenAI** announced its existence
- **OpenAI** released their Universe platform for developing a reinforcement learning agent
- **DeepMind** open-sourced their platform for experimenting with RL agent
- **Uber** announced they had acqui-hired Geometric Intelligence in order to expedite their own AI efforts
- **Apple** announced that it will publish it’s research and make source-code available

Plan

- Keynote
- Generative Adversarial Nets (GAN)
- Deep RL
- Meta-Learning

Keynote

-- Yann Lecun

LeCun's Cake

- learning needs to be mostly unsupervized
- for good data representations

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning (cake)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- **(Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)**

He Really Meant End-to-End Training

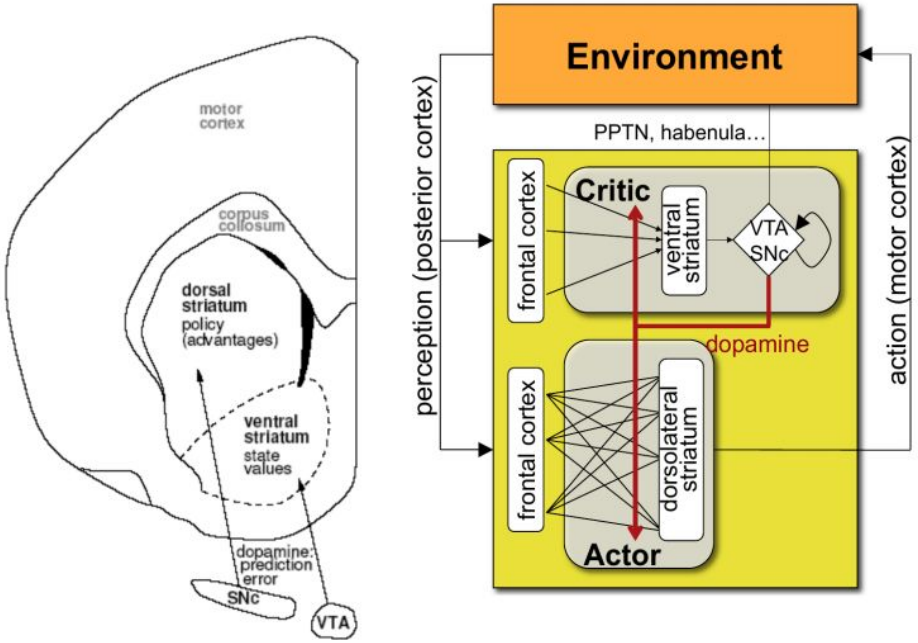
Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor

Mainstream Modern Pattern Recognition: Unsupervised mid-level features

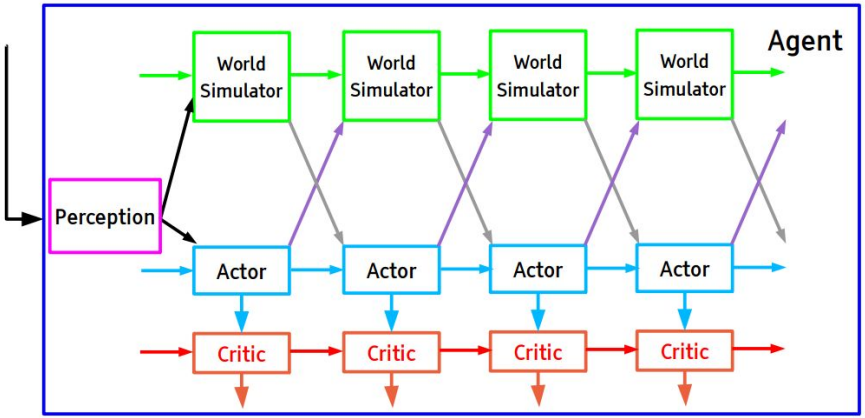
Deep Learning: Representations are hierarchical and trained

Learning Agents

- Inspiration from human brain



- The essence of intelligence is the ability to predict



Generative Adversarial Nets (GAN)

-- a tutorial by Ian Goodfellow

“the biggest breakthrough in Machine Learning in the last 1-2 decades.”
-- Yann Lecun

Why study generative models?

- Realistic generation tasks
- Semi-supervised learning

“What I cannot create,

I do not understand.”

—Richard Feynman

Human: *what is moral ?*
Machine: *what empowered humanity , what intellectual the essence is .*
Human: *what is immoral ?*
Machine: *the fact that you have a child .*
Human: *what is morality ?*
Machine: *what is altruism ?*
Human: *what is the definition of altruism ?*
Machine: *if you don ’t believe in god , then you don ’t know .*
Human: *ok ... so what is the definition of morality ?*
Machine: *well , the truth is , you ’re not a believer in god almighty .*
Human: *tell me the definition of morality , i am quite upset now !*
Machine: *i ’m not ashamed of being a philosopher !*

[Vinyals, Le, 2015]

[Ledig+, 2016]

Super-Resolution

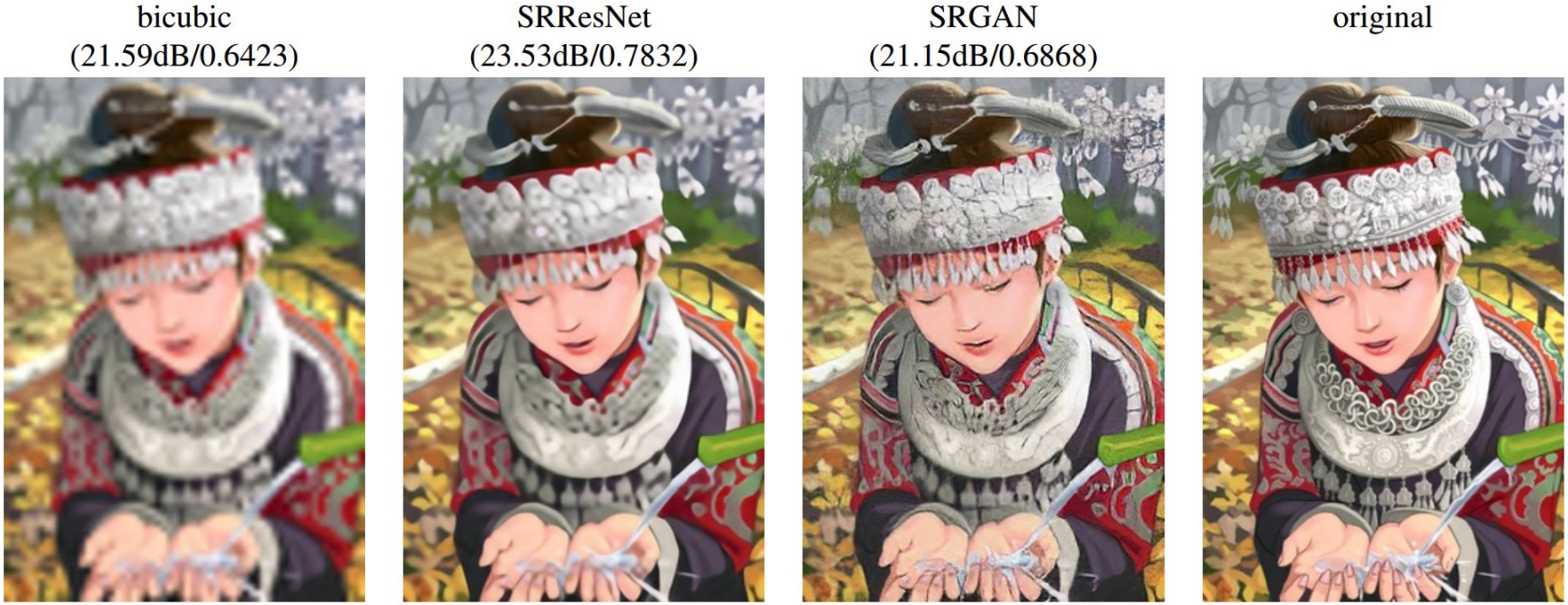


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

Art

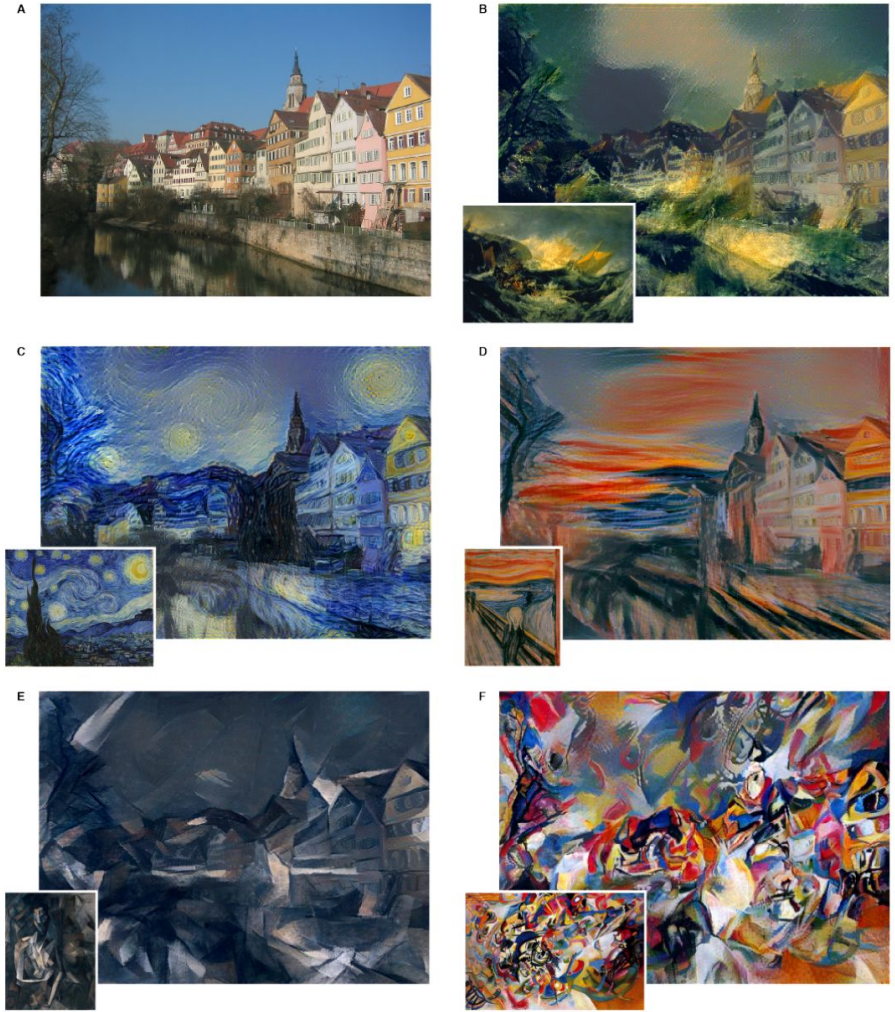


Figure 2: Images that combine the content of a photograph with the style of several well-known artworks. The images were created by finding an image that simultaneously matches the content representation of the photograph and the style representation of the artwork (see Methods).

[Gatys, Ecker, Bethge, 2015]

[Isola+, 2016]

Graphics

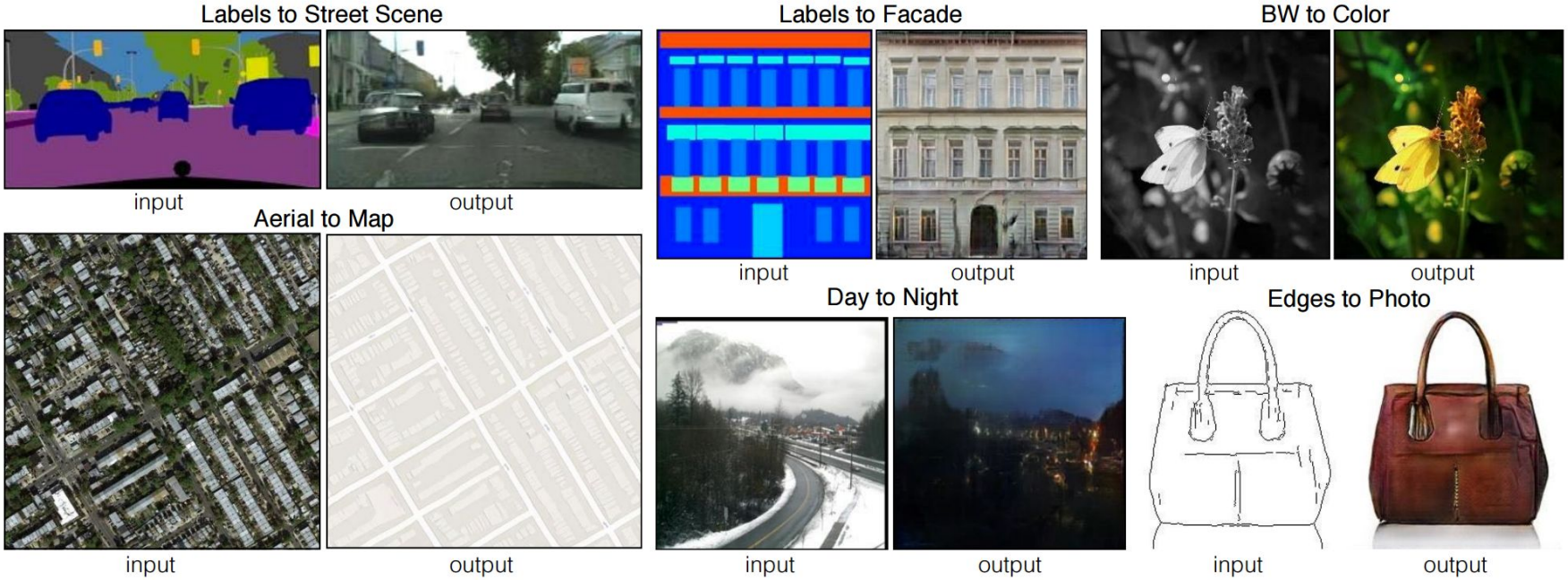
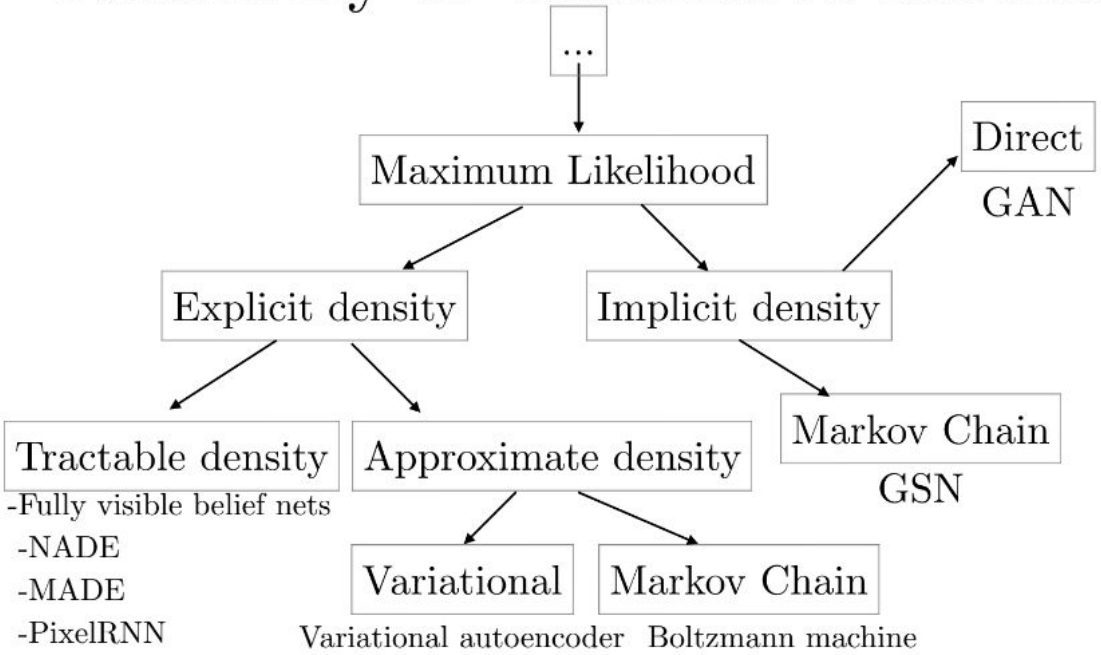
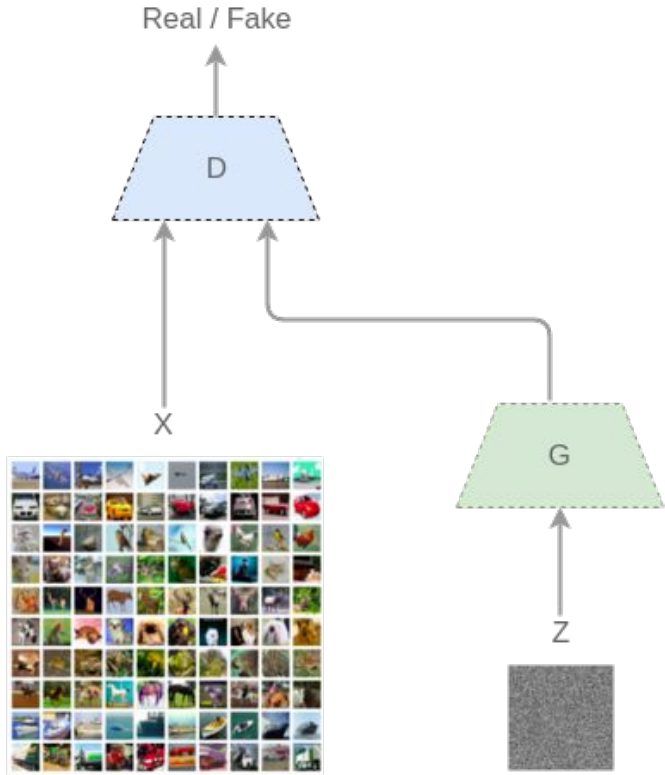


Figure 1: Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image. These problems are often treated with application-specific algorithms, even though the setting is always the same: map pixels to pixels. Conditional adversarial nets are a general-purpose solution that appears to work well on a wide variety of these problems. Here we show results of the method on several. In each case we use the same architecture and objective, and simply train on different data.

Taxonomy of Generative Models



(Goodfellow 2016)



Minimax Game

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

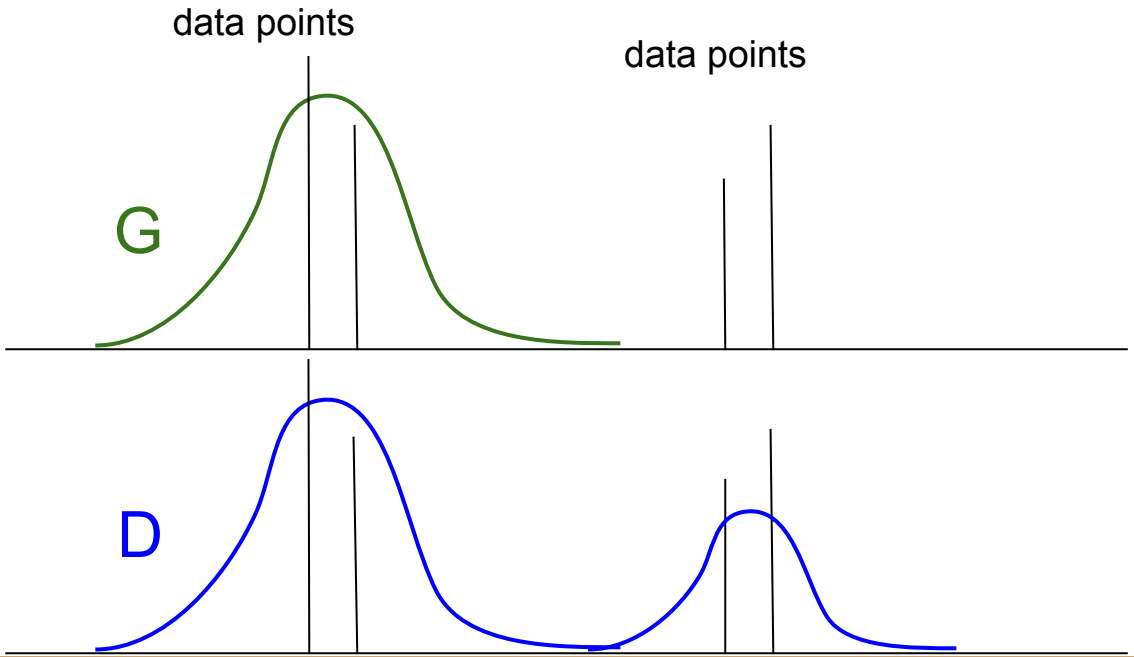
-Equilibrium is a saddle point of the discriminator loss

-Resembles Jensen-Shannon divergence

-Generator minimizes the log-probability of the discriminator being correct

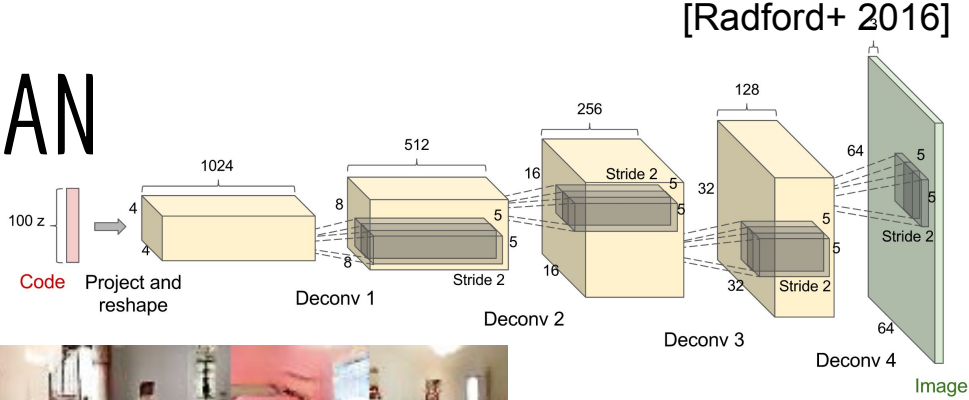
GAN as a way of regularization

- Less incentive to fit individual data points



[Radford+ 2016]

Deep Convolutional (DC) GAN



[Radford+ 2016]

Deep Convolutional (DC) GAN

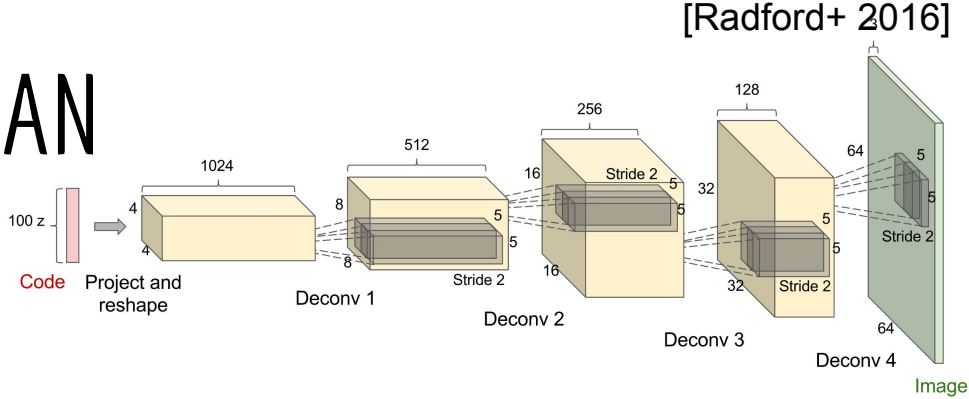


Table 2: SVHN classification with 1000 labels

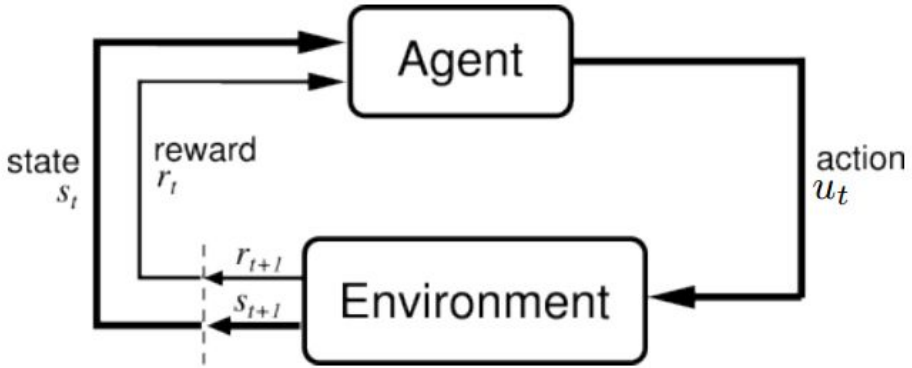
Model	error rate
KNN	77.93%
TSVM	66.55%
M1+KNN	65.63%
M1+TSVM	54.33%
M1+M2	36.02%
SWWAE without dropout	27.83%
SWWAE with dropout	23.56%
DCGAN (ours) + L2-SVM	22.48%
Supervised CNN with the same architecture	28.87% (validation)

Deep RL

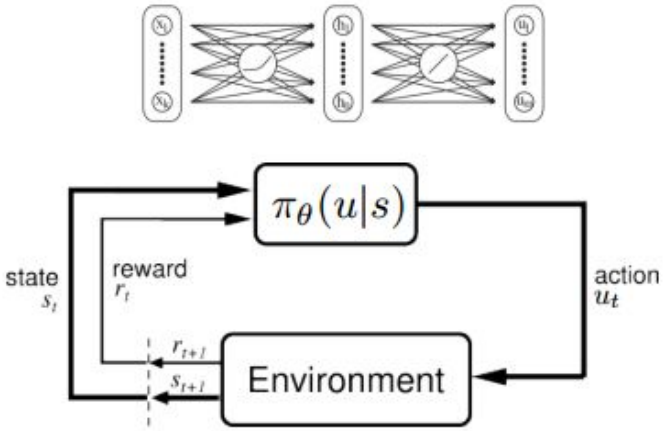
-- tutorials by Pieter Abbeel and John Schulman

Reinforcement Learning

- Any ML problem can be formulated as a RL problem



Policy Optimization



- Consider control policy parameterized by parameter vector θ

$$\max_\theta \quad \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

- Often stochastic policy class (smooths out the problem):

$\pi_\theta(u|s)$: probability of action u in state s

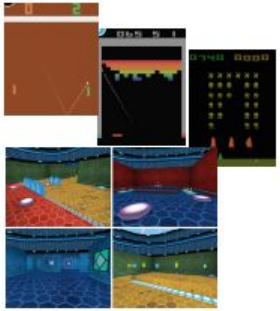
A relatively new field with recent successes

Kohl and Stone, 2004

Ng et al, 2004

Tedrake et al, 2005

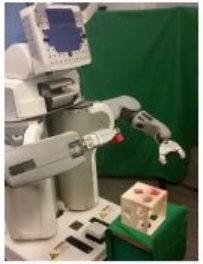
Kober and Peters, 2009



Mnih et al, 2015
(A3C)

Silver et al, 2014
(DPG)
Lillicrap et al, 2015
(DDPG)

Schulman et al,
2016 (TRPO + GAE)



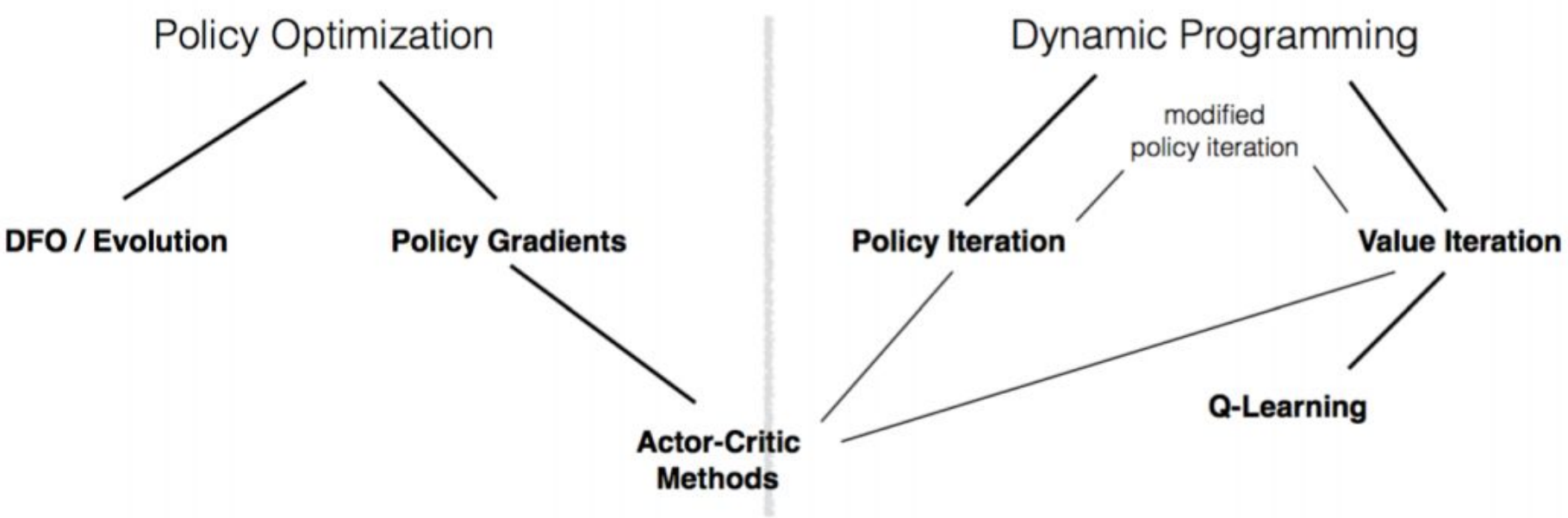
Levine*, Finn*, et
al, 2016
(GPS)

Silver*, Huang*, et
al, 2016
(AlphaGo**)

The RL landscape

- Simple

- Stable
- Data efficient



[Szita and Lorincz 2006]

Cross-Entropy Method

$$\max_\theta U(\theta) = \max_\theta \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

CEM:

for iter i = 1, 2, ...

for population member e = 1, 2, ...

sample $\theta^{(e)} \sim P_{\mu^{(i)}}(\theta)$

execute roll-outs under $\pi_{\theta^{(e)}}$

store $(\theta^{(e)}, U(e))$

endfor

$\mu^{(i+1)} = \arg\max_\mu \sum_{\bar{e}} \log P_\mu(\theta^{(\bar{e})})$

where $\bar{e}$ indexes over top p %

endfor

- Can work surprisingly well
- not data efficient

Table 1: Average Tetris Scores of Various Algorithms.

Method	Mean Score	Reference
Nonreinforcement learning		
Hand-coded	631,167	Dellacherie (Fahey, 2003)
Genetic algorithm	586,103	(Böhm et al., 2004)
Reinforcement learning		
Relational reinforcement learning+kernel-based regression	≈50	Ramon and Driessens (2004)
Policy iteration	3183	Bertsekas and Tsitsiklis (1996)
Least squares policy iteration	<3000	Lagoudakis, Parr, and Littman (2002)
Linear programming + Bootstrap	4274	Farias and van Roy (2006)
Natural policy gradient	≈6800	Kakade (2001)
CE+RL	21,252	
CE+RL, constant noise	72,705	
CE+RL, decreasing noise	348,895	

Likelihood Ratio Policy Gradient

[Aleksandrov, Sysoyev, & Shemeneva, 1968]
[Rubinstein, 1969]
[Glynn, 1986]
[Reinforce, Williams 1992]
[GPOMDP, Baxter & Bartlett, 2001]

We let τ denote a state-action sequence $s_0, u_0, \ldots, s_H, u_H$.

- Optimizing the expected utility $U(\theta) = \mathrm{E}[\sum_{t=0}^{H} R(s_t, u_t); \pi_\theta] = \sum_{\tau} P(\tau; \theta) R(\tau)$

- Is almost the same as MLE except for a weight P(t;θ)R(t)
- Valid even if R and sample space are discrete!!

- Unstable, need good model initialization and ways to reduce gradient variances

$$
\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_{\tau} P(\tau; \theta) R(\tau) \\
&= \sum_{\tau} \nabla_\theta P(\tau; \theta) R(\tau) \\
&= \sum_{\tau} \frac{P(\tau; \theta)}{P(\tau; \theta)} \nabla_\theta P(\tau; \theta) R(\tau) \\
&= \sum_{\tau} P(\tau; \theta) \frac{\nabla_\theta P(\tau; \theta)}{P(\tau; \theta)} R(\tau) \\
&= \sum_{\tau} P(\tau; \theta) \nabla_\theta \log P(\tau; \theta) R(\tau)
\end{aligned}
$$

The Step Size Problem

Why are step sizes a big deal in RL?

- Supervised learning
  - Step too far → next updates will fix it
- Reinforcement learning
  - Step too far → bad policy
  - Next batch: collected under bad policy
  - Can't recover, collapse in performance!

- Bad stability

[Kakade and Langford 2002]
[Schulman+ 2015]

Surrogate Objective

- Collect data with an old policy (for stability)
- Reweight examples by importance sampling
  - The probability ratio between the new policy and the old policy

$$L(\pi) = \mathbb{E}_{\pi_{\text{old}}}\left[\frac{\pi(a \mid s)}{\pi_{\text{old}}(a \mid s)} A^{\pi_{\text{old}}}(s, a)\right]$$

$$\nabla_\theta L(\pi_\theta)\big|_{\theta_{\text{old}}} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta_{\text{old}}} \quad \text{(policy gradient)}$$

[Mnih+ 2015, 2016]
[Chen+ 2016]

Experience Replay

- Keep a set of (hard to find, or human generated) good examples
- Repeatedly use them for training (together with recent bad examples)
  - E.g., 1M replay buffer for DQN Artari training
  - E.g., Neural symbolic machines keep track of the best program for each query

The Delayed Reward Problem

- With policy gradient methods, we are confounding the effect of multiple actions:

$$\hat{A}_t = r_t + r_{t+1} + r_{t+2} + \cdots - b(s_t)$$

  mixes effect of $a_t, a_{t+1}, a_{t+2}, \ldots$
- SNR of $\hat{A}_t$ scales roughly as $1/T$
  - Only a_t contributes to *signal* $A^\pi(s_t, a_t)$, but $a_{t+1}, a_{t+2}, \ldots$ contribute to noise.

- Bad data efficiency

Bootstrapping

- use the value function to estimate future rewards
  - Subtracting out baselines, we get advantage estimators

$$\hat{A}_t^{(1)} = r_t + \gamma V(s_{t+1}) - V(s_t)$$
$$\hat{A}_t^{(2)} = r_t + r_{t+1} + \gamma^2 V(s_{t+2}) - V(s_t)$$
$$\dots$$
$$\hat{A}_t^{(\infty)} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots - V(s_t)$$

  - $\hat{A}_t^{(1)}$ has low variance but high bias, $\hat{A}_t^{(\infty)}$ has high variance but low bias.
  - Using intermediate k (say, 20) gives an intermediate amount of bias and variance

[Mnih+ 2015, 2016]

Advantage Actor-Critic

- Minimize reward loss and value function error at the same time

for iteration=$1, 2, \ldots$ **do**
Agent acts for T timesteps (e.g., $T = 20$),
For each timestep t, compute

$$\hat{R}_t = r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V(s_t)$$
$$\hat{A}_t = \hat{R}_t - V(s_t)$$

$\hat{R}_t$ is target value function, in regression problem
$\hat{A}_t$ is estimated advantage function
Compute loss gradient $g = \nabla_\theta \sum_{t=1}^{T} \left[-\log \pi_\theta(a_t \mid s_t) \hat{A}_t + c(V(s) - \hat{R}_t)^2 \right]$
g is plugged into a stochastic gradient descent variant, e.g., Adam.
end for

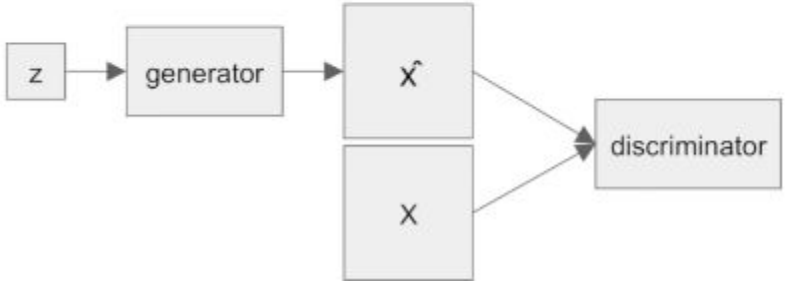
Meta-Learning

[Radford+ 2016]

Meta-Generative Models

- Two models works better than one

Variational Autoencoders (VAE)
Kingma and Welling [1312.6114]

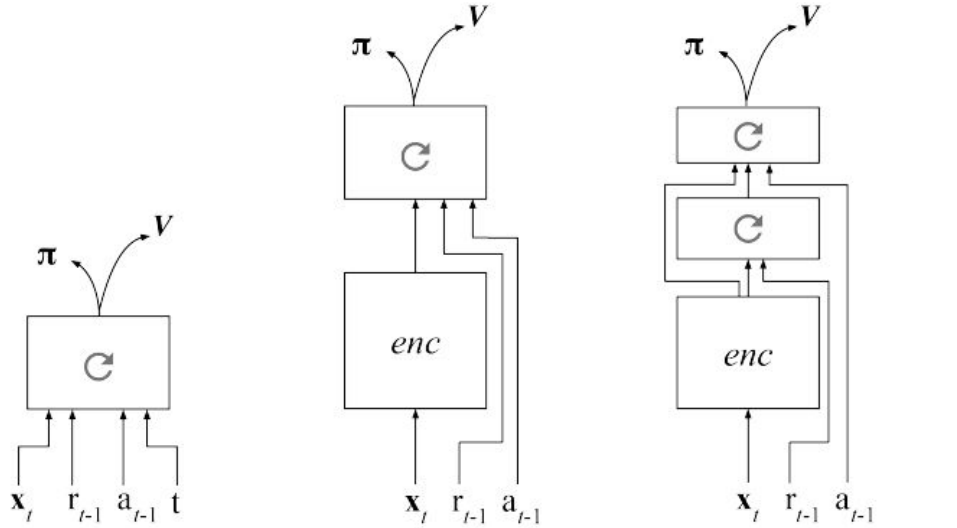


Generative Adversarial Networks (GAN)
Goodfellow et [1406.2661]

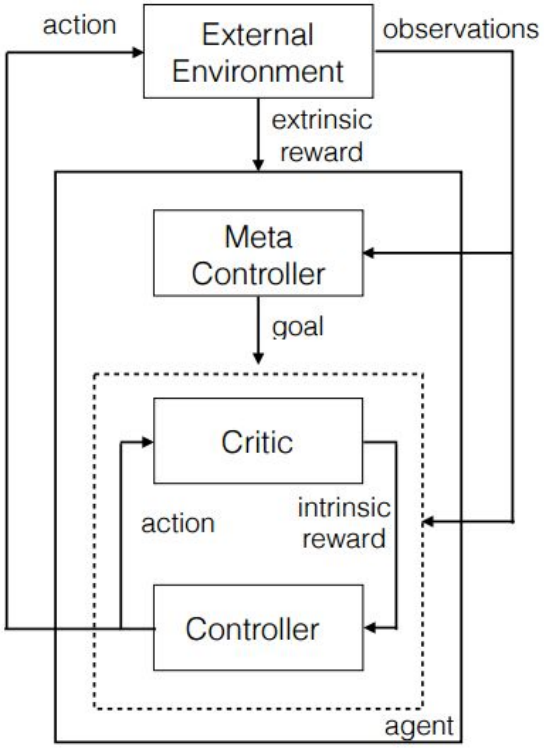
[Wang+ 2016]

Meta-RL Models

- Two learning systems:
  - one lower-level system that learns relatively quickly, and which is primarily responsible for adapting to each new task;
  - and a slower higher-level system that works across tasks to tune and improve the lower-level system



Meta-RL Models



[Kulkarni+ 2016]

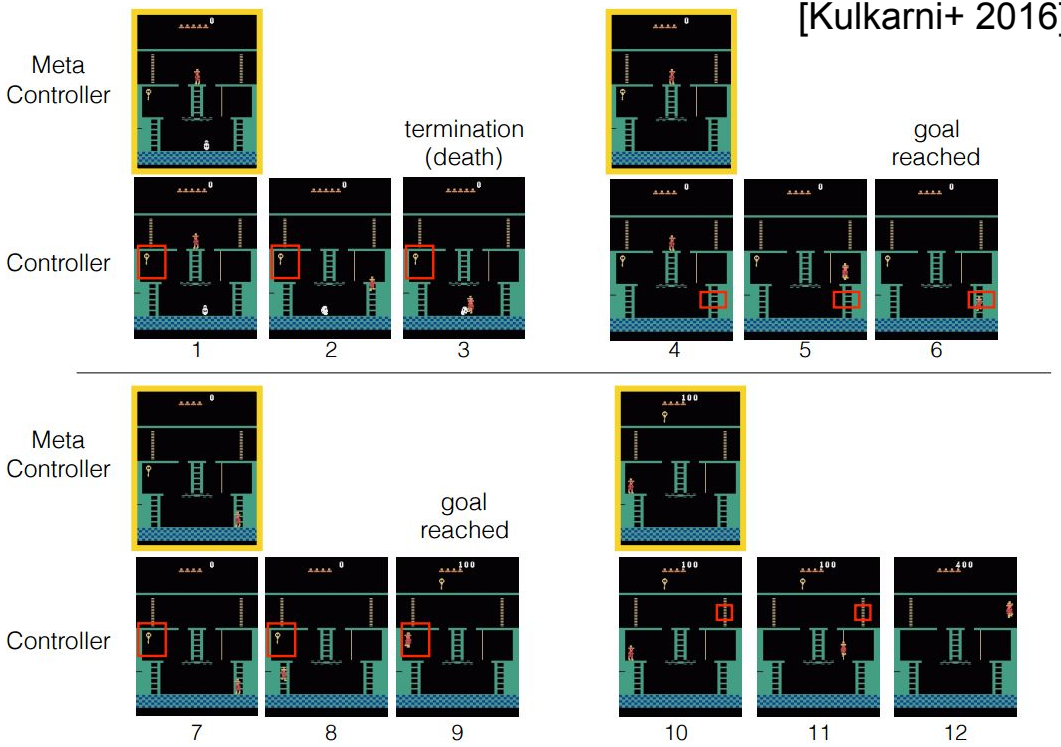


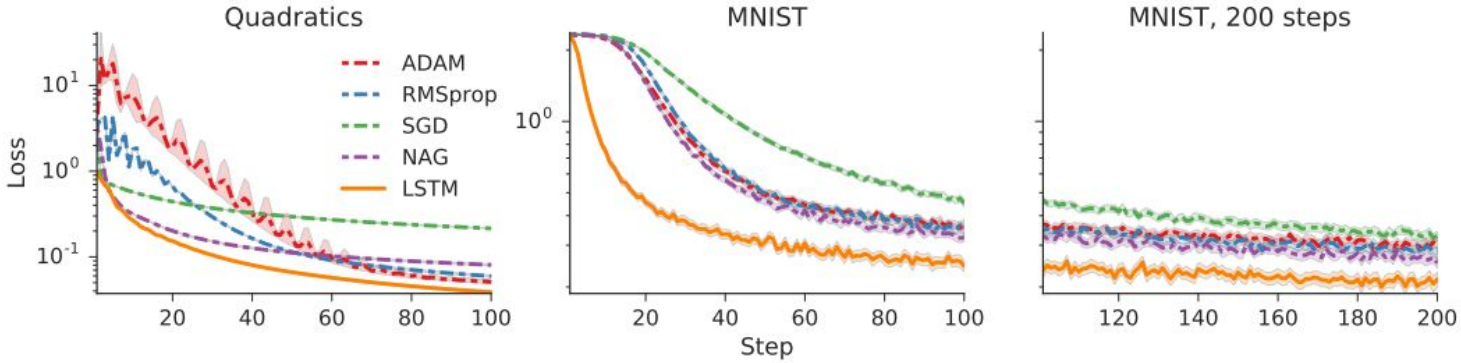
Figure 7: **Sample gameplay by our agent on Montezuma's Revenge:** The four quadrants are arranged in a temporally coherent manner (top-left, top-right, bottom-left and bottom-right). At the very beginning, the meta-controller chooses key as the goal (illustrated in *red*). The controller then tries to satisfy this goal by taking a series of low level actions (only a subset shown) but fails due to colliding with the skull (the episode terminates here). The meta-controller then chooses the bottom-right ladder as the next goal and the controller terminates after reaching it. Subsequently, the meta-controller chooses the key and the top-right door and the controller is able to successfully achieve both these goals.

[Andrychowicz+ 2016]

Meta-Optimizer

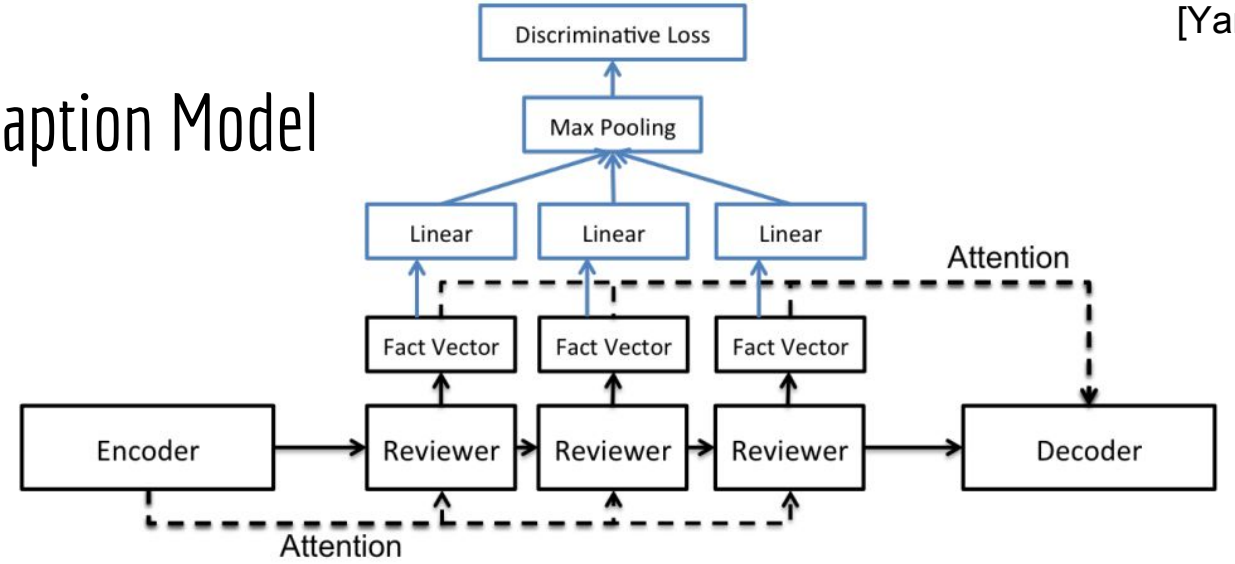
- Control NN parameter updates using LSTMs

$$\mathcal{L}(\phi) = \mathbb{E}_f \left[\sum_{t=1}^{T} w_t f(\theta_t) \right] \qquad \text{where} \qquad \theta_{t+1} = \theta_t + g_t \,,$$
$$\begin{bmatrix} g_t \\ h_{t+1} \end{bmatrix} = m(\nabla_t, h_t, \phi)$$



[Yang+ 2016]

Multiresolution Caption Model



A giraffe standing next to a tree in a zoo standing field tree giraffe trees near by grass giraffes their walking fence zoo enclosure tall

Multiresolution Dialogue Models

- Sorry I can't find their poster online

Table 5: Twitter Coarse Sequence Examples

Natural Language Tweets	Noun Representation
<first_speaker> at pinkberry with my pink princess enjoying a precious moment <url>	present_tenses pinkberry princess moment
<second_speaker>- they are adorable, alma still speaks about emma bif sis . hugs	present_tenses alma emma bif sis hugs
<first_speaker> <at> when you are spray painting, where are you doing it ? outside ? in your apartment ? where ?	present_tenses spray painting apartment
<second_speaker> <at> mostly spray painting outside but some little stuff in the bathroom .	present_tenses spray stuff bathroom

[Serban+ 2016]

Multiresolution Recurrent Neural Network (MrRNN):

Model dialogue as two stochastic processes:

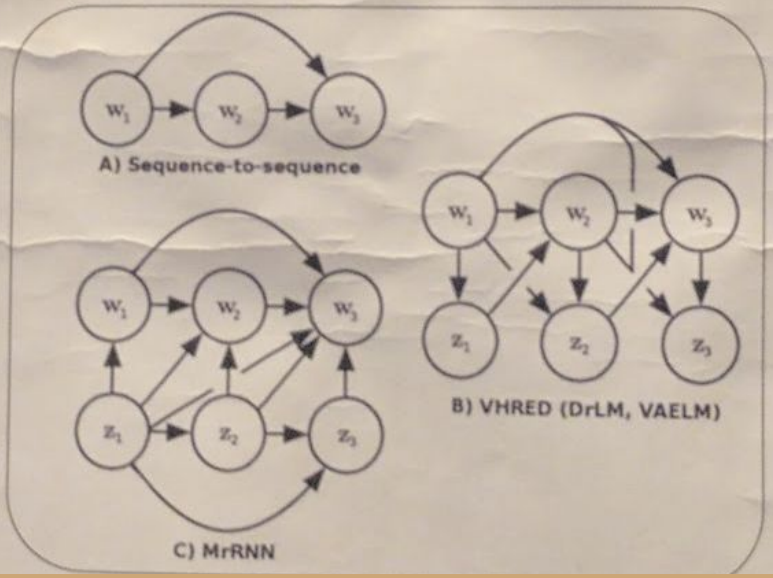
$$P_\theta(\mathbf{w}_1, \ldots, \mathbf{w}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N) = \prod_{n=1}^{N} P_\theta(\mathbf{z}_n|\mathbf{z}_1, \ldots, \mathbf{z}_{n-1}) P_\theta(\mathbf{w}_n|\mathbf{w}_1, \ldots, \mathbf{w}_{n-1}, \mathbf{z}_1, \ldots, \mathbf{z}_n)$$

Utterances *Coarse Sequences*

MrRNNs generalize better by having z's:

- **follow latent stochastic process** (z's are marginally independent w's)
- **use factorial representation** ($\mathbf{z}_n$ is a sequence of discrete tokens)
- **incorporate prior knowledge** (e.g. technical support representations)

Hierarchical generation process helps generate more meaningful and on-topic (goal-driven) responses.



[Serban+ 2016]

Multiresolution Dialogue Models

Table 6: Ubuntu Coarse Sequence Examples

Natural Language Dialogues	Activity-Entity Coarse Dialogues
if you can get a hold of the logs, there 's stuff from **unknown** about his inability to install amd64	future_tenses get_activity install_activity amd64_entity no_cmd
I'll check fabbione 's log, thanks sounds like he had the same problem I did ew, why ? ...	no_tenses check_activity no_cmd past_present_tenses none_activity no_cmd no_tenses none_activity no_cmd ...
upgrade lsb-base and acpid	no_tenses upgrade_activity lsb_entity acpid_entity no_cmd
i'm up to date	no_tenses none_activity no_cmd
what error do you get ?	present_tenses get_activity no_cmd
i don't find error :/ where do i search from ? acpid works, but i must launch it manually in a root sterm ...	present_tenses discover_activity no_cmd present_future_tenses work_activity acpid_entity root_entity no_cmd ...

Thanks

Reference

1. Overview http://beamandrew.github.io/deeplearning/2016/12/12/nips-2016.html
2. Overview http://blog.evjang.com/2017/01/nips2016.html
3. Keynote: https://t.co/LDzqac7na1
4. GAN: https://arxiv.org/abs/1701.00160
5. GAN: http://www.slideshare.net/indicods/deep-advancements-in-generative-modeling
6. RL: http://people.eecs.berkeley.edu/~pabbeel/nips-tutorial-policy-optimization-Schulman-Abbeel.pdf
7. RL: http://rll.berkeley.edu/deeprlcourse/docs/nuts-and-bolts.pdf