DLRL Summer School 2019 Overview

Alex Fedorov

TReNDS Reading group
August, 16

dlrlsummerschool.ca

CIFAR amii

ABOUT ATTENDEES SPEAKERS SPONSORS FAQS SUMMER INSTITUTE CONTACT

Deep Learning and Reinforcement Learning Summer School 2019

July 24 – August 2, 2019
Edmonton, Alberta, Canada

DL Part

Continental Breakfast, 10:30	Continental Breakfast, 10:30	Continental Breakfast, 10:30	Continental Breakfast, 10:30
Deep Learning I: Hugo 11:00 – 12:15 1-430 CCIS, University of Alberta	Images: Angel Chang 11:00 – 12:15 1-430 CCIS, University of Alberta	Optimization in DL: Jimmy Ba 11:00 – 12:15 1-430 CCIS, University of Alberta	Hands-On Deep Learning II 11:00 – 12:15 1-430 CCIS, University of Alberta
Deep Learning II: Hugo 12:15 – 13:30 1-430 CCIS, University of Alberta	Hands-On Deep Learning I 12:15 – 13:30 1-430 CCIS, University of Alberta	NLP: Alona Fyshe 12:15 – 13:30 1-430 CCIS, University of Alberta	GANs: Ke Li 12:15 – 13:30 1-430 CCIS, University of Alberta
Lunch 13:30 – 15:30 PCL Lounge, CCIS, University of Alberta	Lunch 13:30 – 15:30 PCL Lounge, CCIS, University of Alberta	Lunch 13:30 – 15:30 PCL Lounge, CCIS, University of Alberta	Lunch 13:30 – 15:30 PCL Lounge, CCIS, University of Alberta
Summer Institute Talk 15:30 – 16:45 1-430 CCIS, University of Alberta	RNNs: Yoshua Bengio 15:30 – 16:45 1-430 CCIS, University of Alberta	Bayesian DL: Roger Grosse 15:30 – 16:45 1-430 CCIS, University of Alberta	Biological DL: Blake Richards 15:30 – 16:45 1-430 CCIS, University of Alberta
Coffee Break, 16:45	Coffee Break, 16:45	Coffee Break, 16:45	Coffee Break, 16:45
CNNs: Graham Taylor 17:15 – 18:30 1-430 CCIS, University of Alberta	Video: Greg Mori 17:15 – 18:30 1-430 CCIS, University of Alberta	Autoencoders/Unsupervised: 17:15 – 18:30 1-430 CCIS, University of Alberta	What's Next: Yoshua Bengio 17:15 – 18:30 1-430 CCIS, University of Alberta

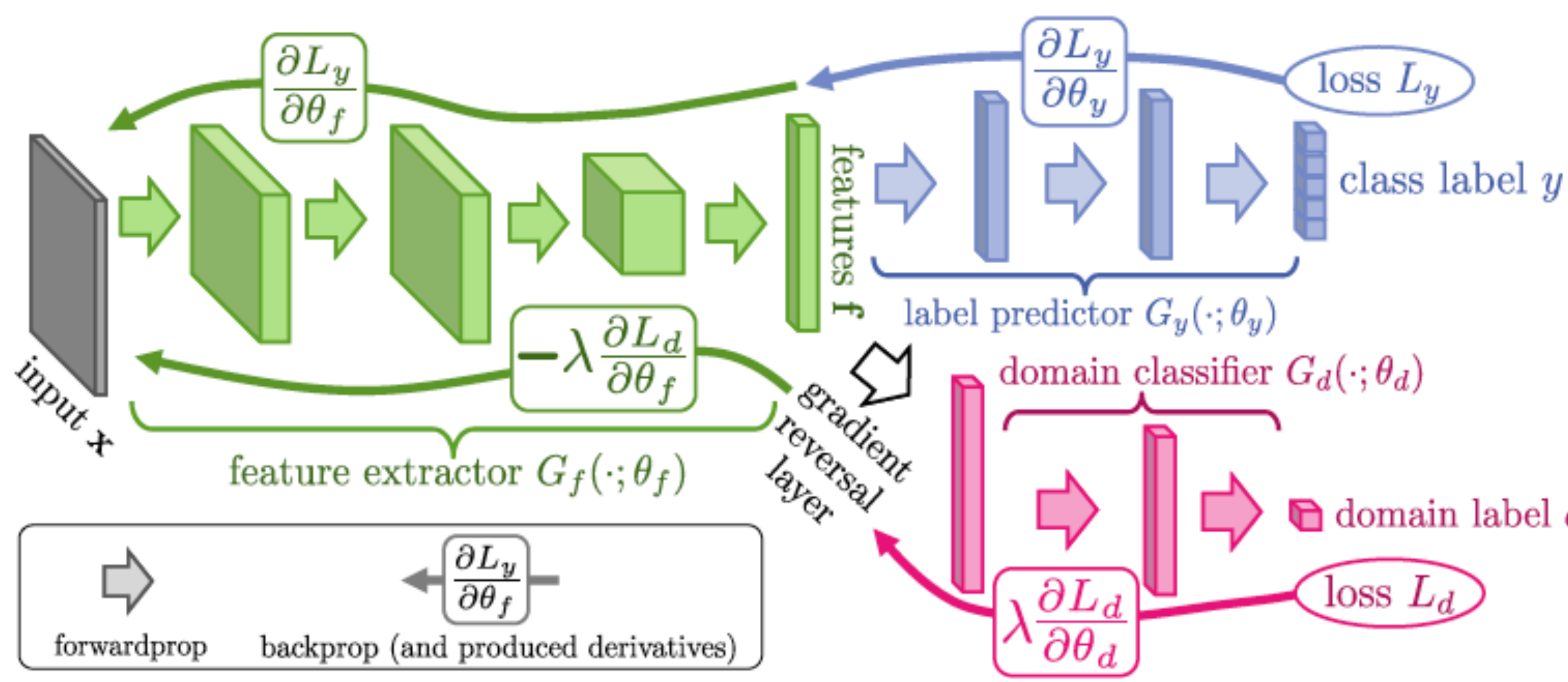
RL Part

Continental Breakfast, 10:30	**Continental Breakfast**, 10:30	**Continental Breakfast**, 10:30	**Continental Breakfast**, 10:30	**Continental Breakfast**, 10:30
TD/RL Intro: Adam White 11:00 – 12:15 1-430 CCIS, University of Alberta	**Hands-On Reinforcement** 11:00 – 12:15 1-430 CCIS, University of Alberta	**Human in the Loop: Matt** 11:00 – 12:15 1-430 CCIS, University of Alberta	**PG: Jan Peters** 11:00 – 12:15 1-430 CCIS, University of Alberta	**Doing RL on Robots: Jan** 11:00 – 12:15 1-430 CCIS, University of Alberta
Bandits: Csaba Szepesvari 12:15 – 13:30 1-430 CCIS, University of Alberta	**POMDPs: Pascal Poupart** 12:15 – 13:30 1-430 CCIS, University of Alberta	**Robust RL: Marek Petrik** 12:15 – 13:30 1-430 CCIS, University of Alberta	**DRL I: Matteo Hessel** 12:15 – 13:30 1-430 CCIS, University of Alberta	**Science with Robots: A.** 12:15 – 13:30 1-430 CCIS, University of Alberta
Lunch 13:30 – 15:30 PCL Lounge, CCIS, University of Alberta	**Lunch** 13:30 – 15:30 PCL Lounge, CCIS, University of Alberta	**Lunch** 13:30 – 14:45 PCL Lounge, CCIS, University of Alberta **Travel to Edmonton Co**, 14:45	**Lunch** 13:30 – 15:30 PCL Lounge, CCIS, University of Alberta	**Lunch** 13:30 – 15:30 PCL Lounge, CCIS, University of Alberta
Sample Efficient RL: Harm 15:30 – 16:45 1-430 CCIS, University of Alberta	**Model-Based RL: Martha** 15:30 – 16:45 1-430 CCIS, University of Alberta	**Multi-Agent: James Wright** 15:30 – 16:45 Hall D, Edmonton Convention Centre	**DRL II: Anna Harutyunyan** 15:30 – 16:45 1-430 CCIS, University of Alberta	**Applied RL on Robots: Patrick** 15:30 – 16:45 1-430 CCIS, University of Alberta
Coffee Break, 16:45	**Coffee Break**, 16:45	**Coffee Break**, 16:45	**Coffee Break**, 16:45	**Coffee Break**, 16:45
Off Policy: Doina Precup 17:15 – 18:30 1-430 CCIS, University of Alberta	**Optimization in RL: Dale** 17:15 – 18:30 1-430 CCIS, University of Alberta	**Multi-Agent: Michael Bowling** 17:15 – 18:30 Hall D, Edmonton Convention Centre	**Options (HRL): Andre Barreto** 17:15 – 18:30 1-430 CCIS, University of Alberta	**RL Research/Frontiers: Rich** 17:15 – 18:30 1-430 CCIS, University of Alberta

Based on Hugo Larochelle lectures

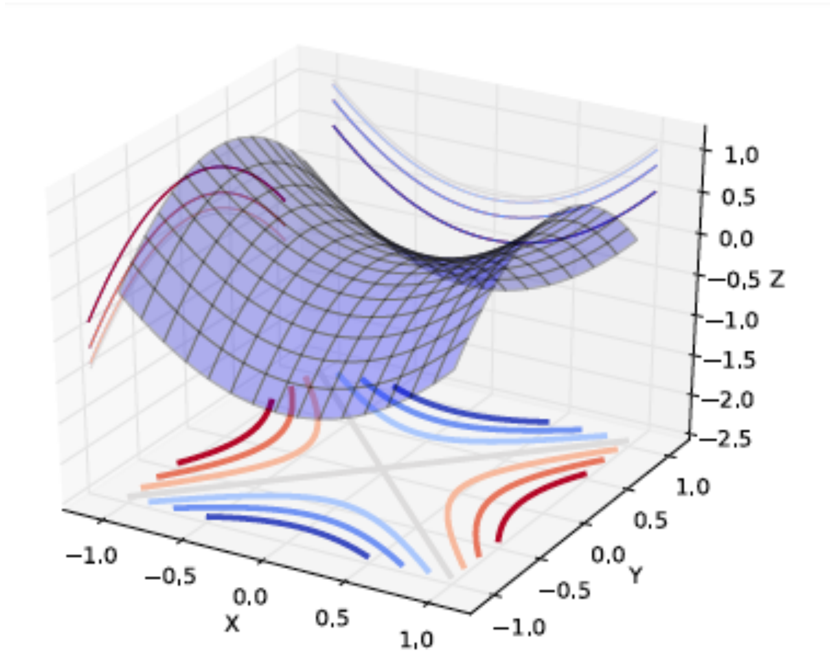
Domain Adaptation

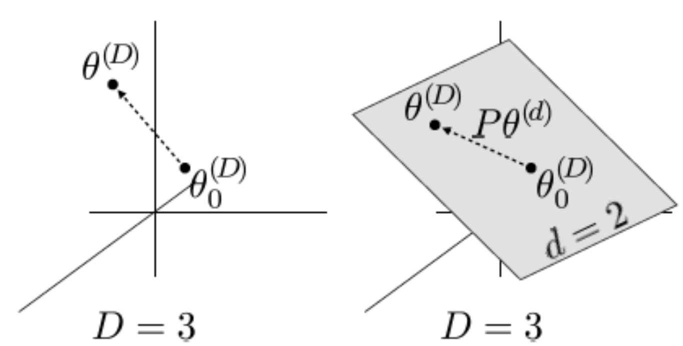
- Learning a predictor in the presence of a shift between training and test distributions.
    - Discriminativeness
    - Domain invariance
- Minimizing the loss of the label classifier and maximizing the loss of the domain classifier
- What can we do with that?
    - **Harmonization**
    - **Training on synthetic data and apply on the real data**
- DANN (Ganin et al. 2015)

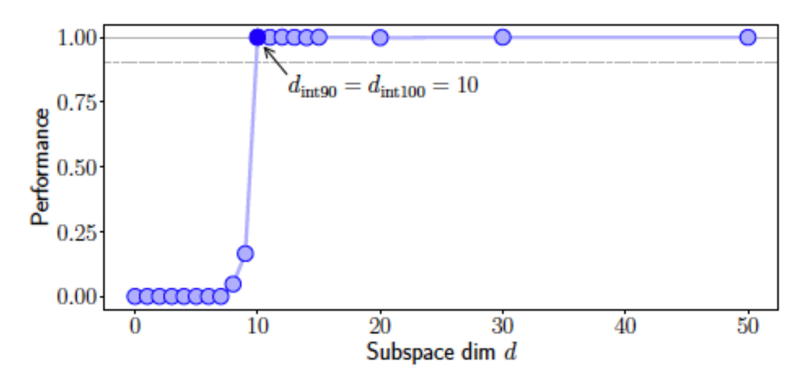


Strange behavior of NN

- **THEY ARE STRANGELY NON-CONVEX**
  - If dataset is created by labeling points using a N-hidden units neural network
    - training another N-hidden units network is likely to fail
    - but training a larger neural network is more likely to work! (saddle points seem to be a blessing)
  - Measuring the Intrinsic Dimension of Objective Landscapes Li, Farkhoor, Liu, Yosinski, ICLR 2018







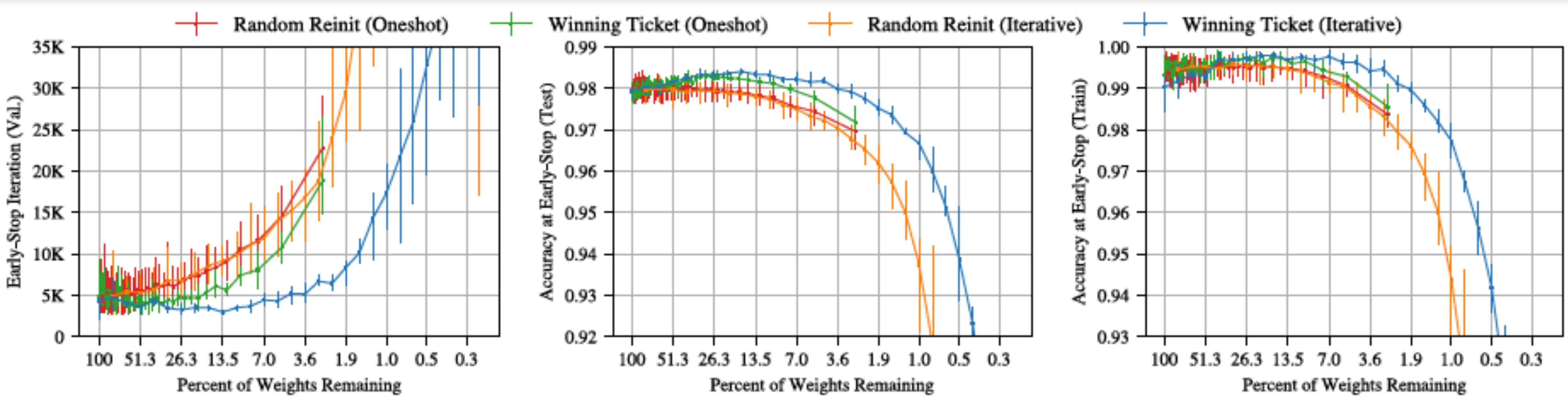
Dataset	MNIST		MNIST (Shuf Pixels)		MNIST (Shuf Labels)
Network Type	FC	LeNet	FC	LeNet	FC
Parameter Dim. D	199,210	44,426	199,210	44,426	959,610
Intrinsic Dim. d_{int90}	750	290	750	1,400	190,000

...	CIFAR-10		ImageNet	Inverted Pendulum	Humanoid	Atari Pong
...	FC	LeNet	SqueezeNet	FC	FC	ConvNet
...	656,810	62,006	1,248,424	562	166,673	1,005,974
...	9,000	2,900	$>$ 500k	4	700	6,000

Strange behavior of NN

Algorithm to Identify Winning Tickets (i.e. subnetworks):

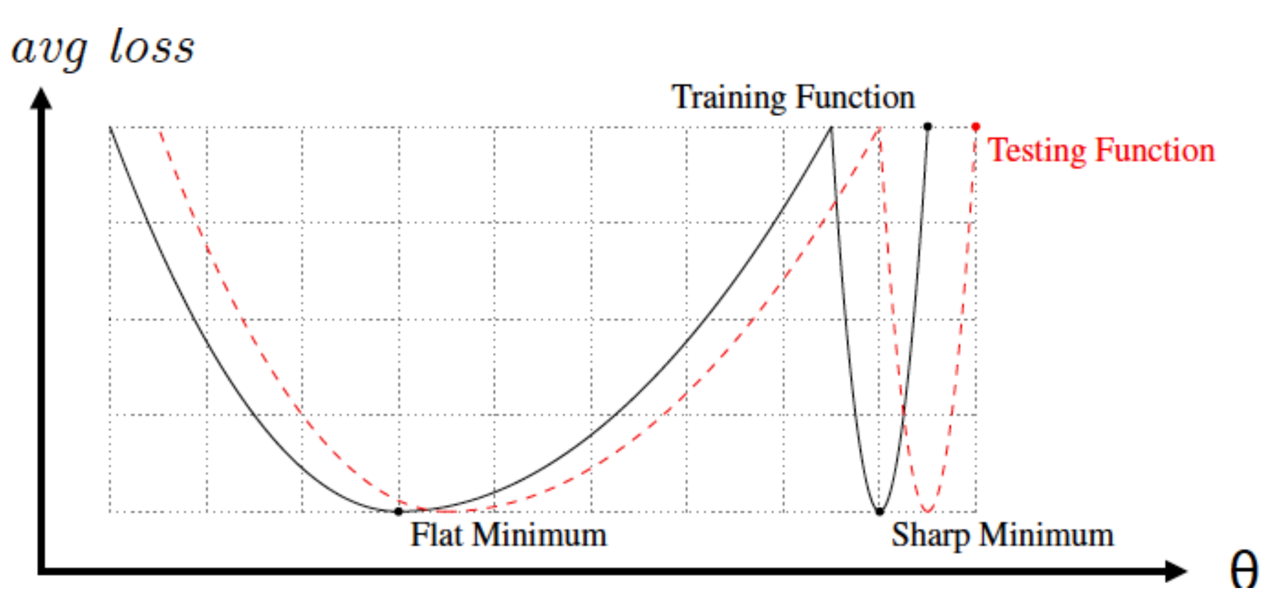
1. Randomly initialize a neural network $f(x; \theta_0)$ (where $\theta_0 \sim \mathcal{D}_\theta$).
2. Train the network for j iterations, arriving at parameters θ_j.
3. Prune $p\%$ of the parameters in θ_j, creating a mask m.
4. Reset the remaining parameters to their values in θ_0, creating the winning ticket $f(x; m \odot \theta_0)$.

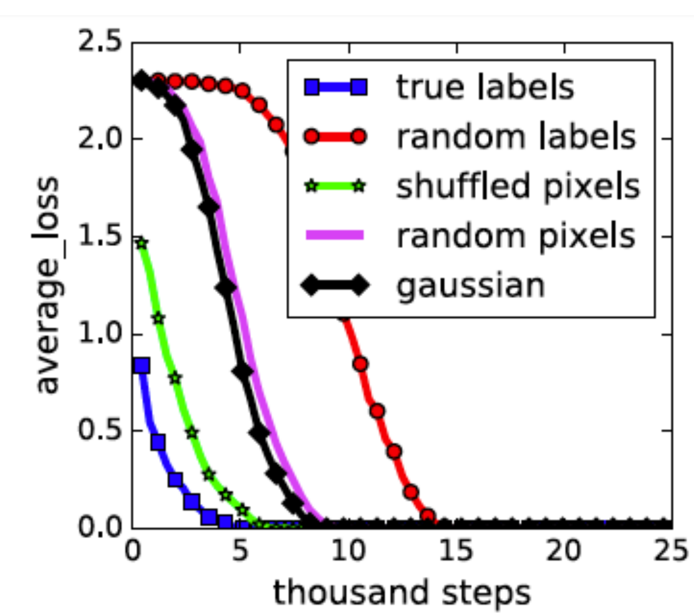


The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks Frankle, Carbin, ICLR 2019

Strange behavior of NN

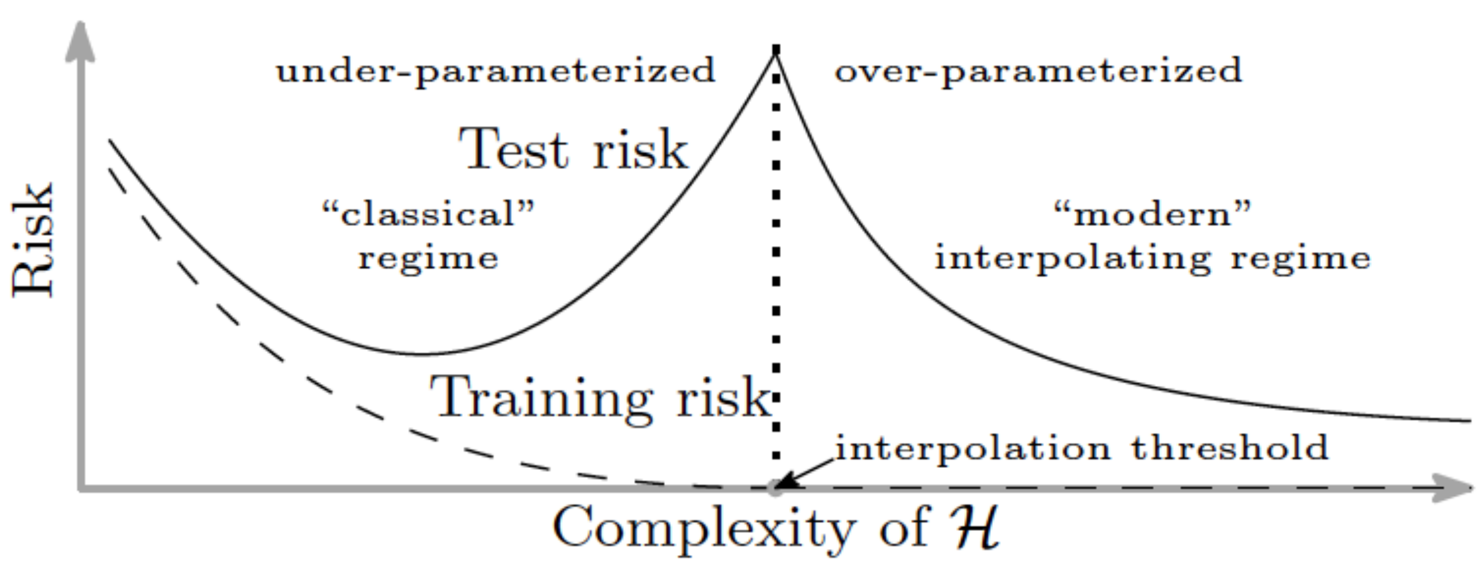
- **THEY WORK BEST WHEN BADLY TRAINED:**
  - Flat Minima Hochreiter, Schmidhuber, Neural Computation 1997
  - On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima Keskar, Mudigere, Nocedal, Smelyanskiy, Tang, ICLR 2017
    - found that using large batch sizes tends to find sharper minima and generalize worse
    - This means that we can’t talk about generalization without taking the training algorithm into account
- **THEY CAN EASILY MEMORIZE**: Understanding Deep Learning Requires Rethinking Generalization Zhang, Bengio, Hardt, Recth, Vinyals, ICLR 2017

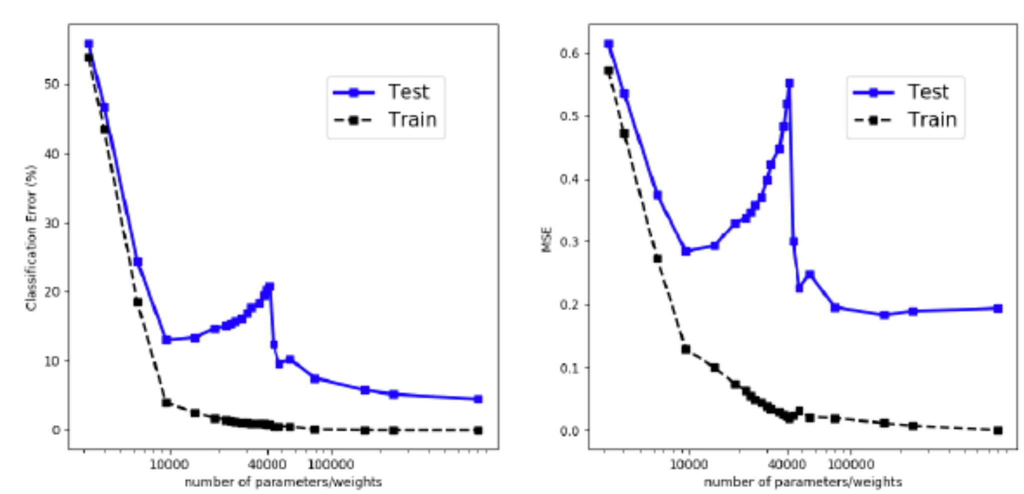




Strange behavior of NN

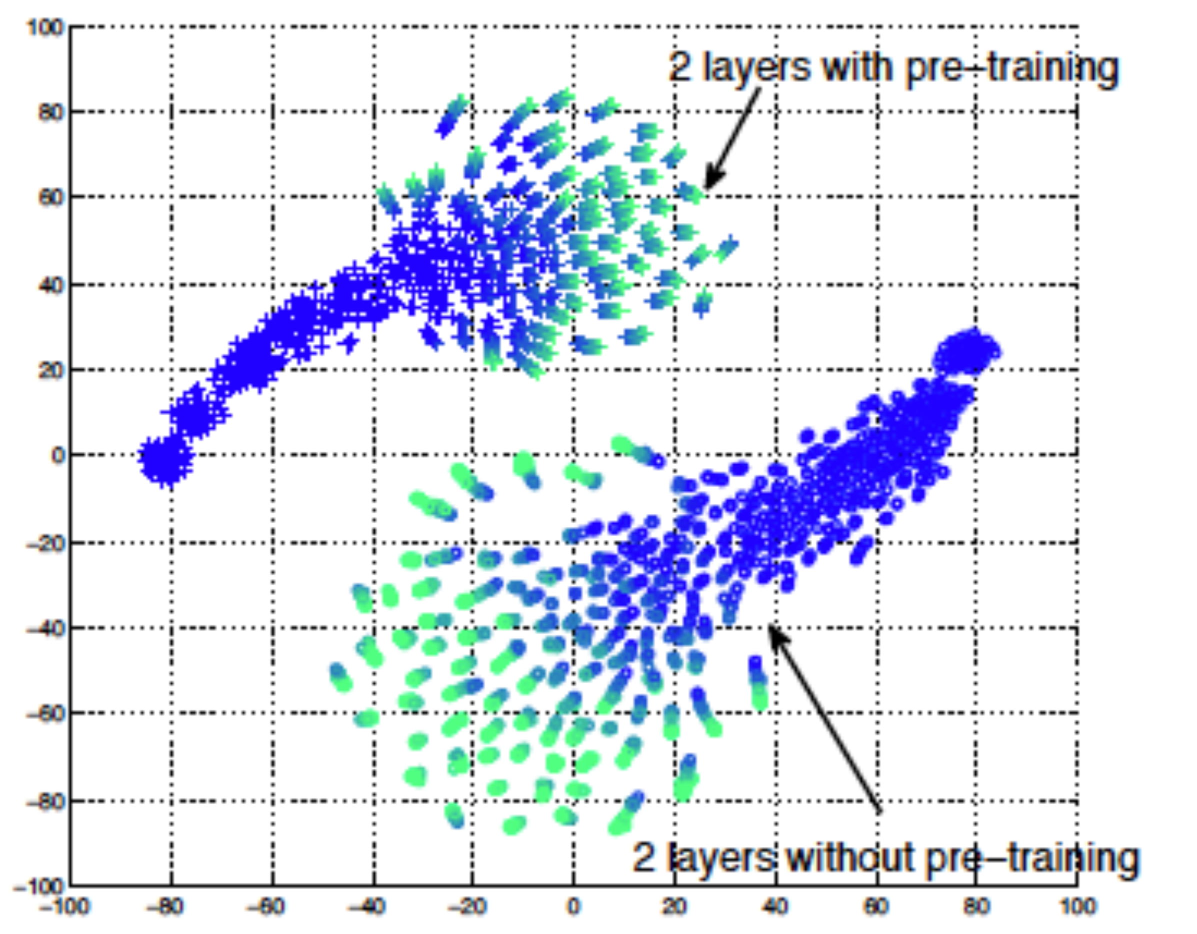
- **THEY UNDERFIT/OVERFIT STRANGELY**
  - Reconciling modern machine learning and the bias-variance trade-off Belkin et al. arXiv 2018

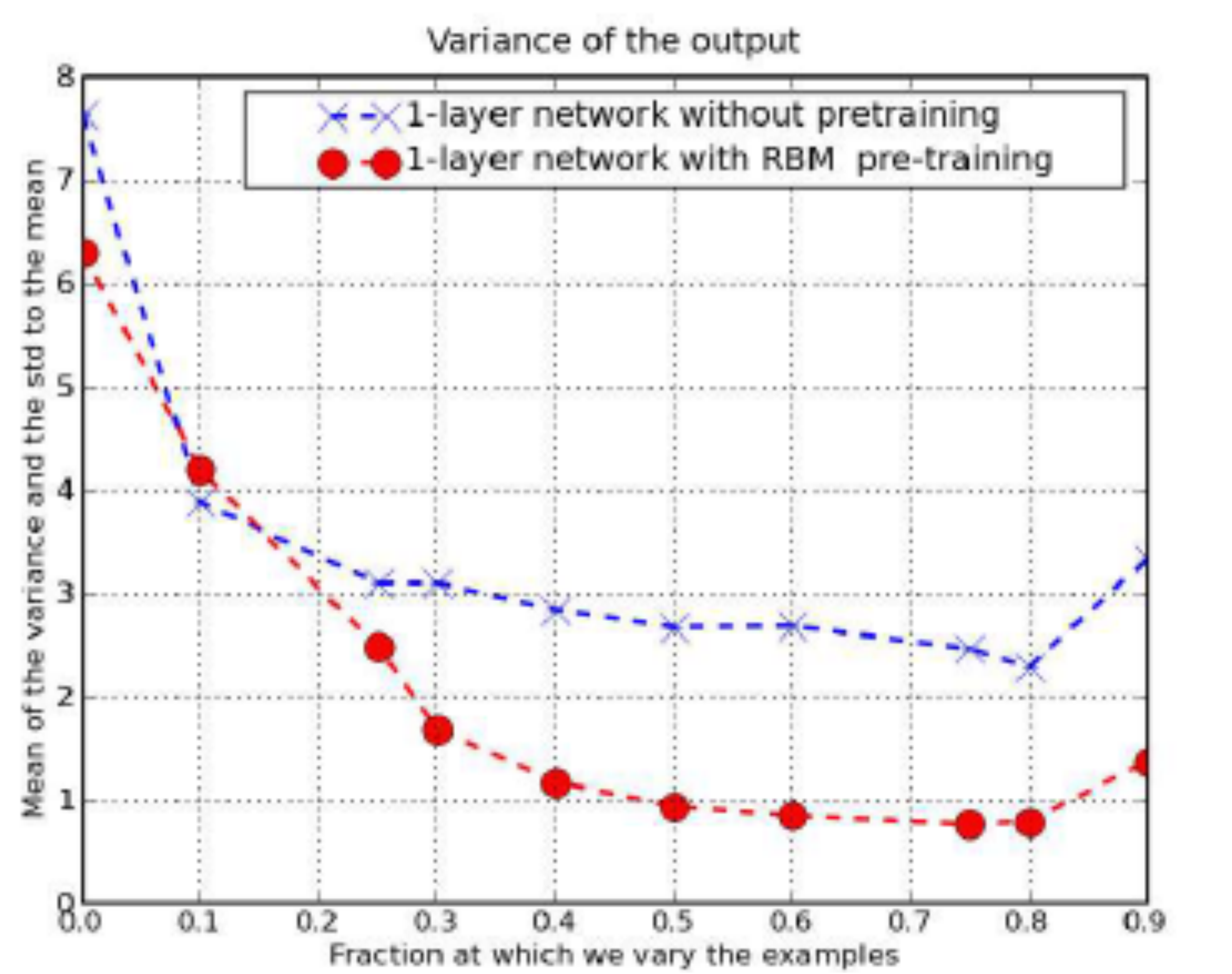




THEY ARE INFLUENCED BY INITIALIZATION AND FIRST EXAMPLES

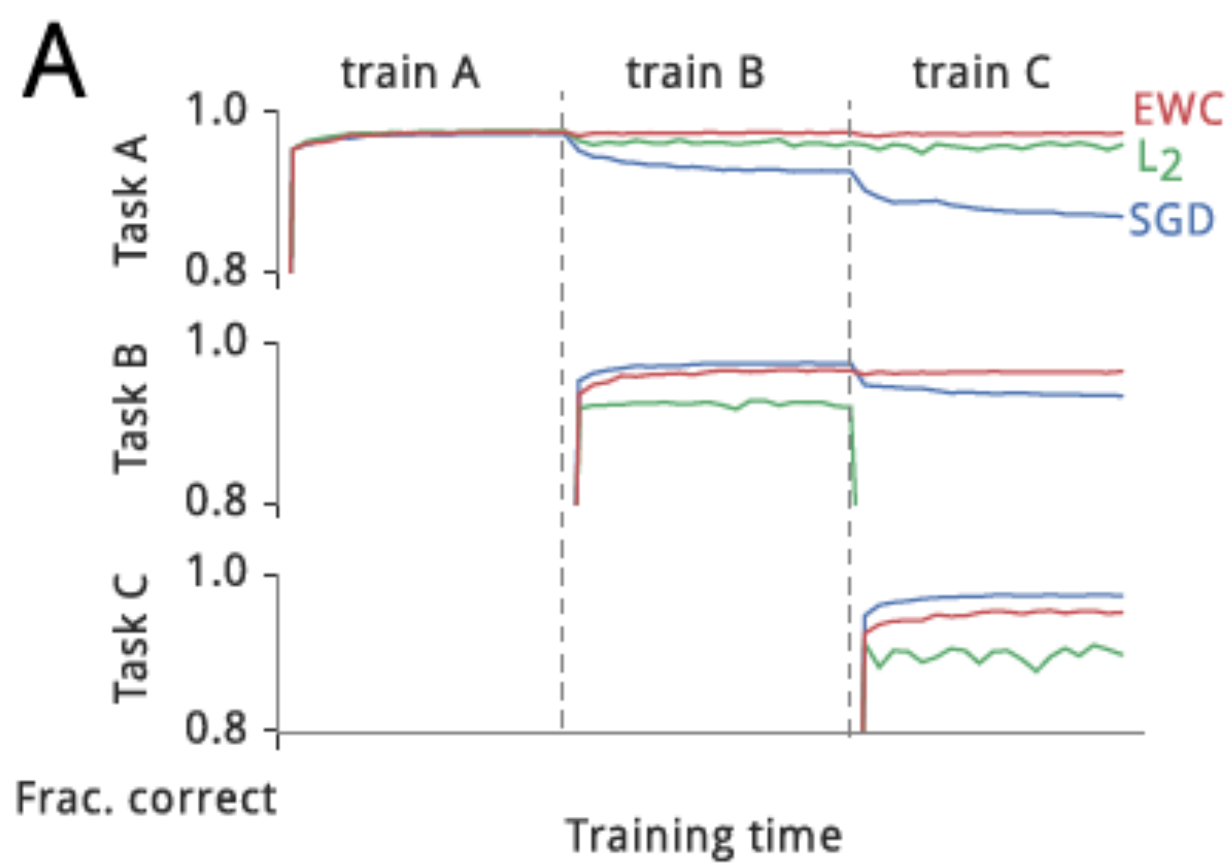
- Why Does Unsupervised Pre-Training Help Deep Learning
Erhan, Bengio, Courville, Manzagol, Vincent, JMLR 2010





YET THEY FORGET WHAT THEY LEARNED

- Overcoming Catastrophic Forgetting in Neural Networks Kirkpatrick et al. PNAS 2017



We have to understands NN theory better

RNNs from Yoshua Bengio talk

RNN Tricks

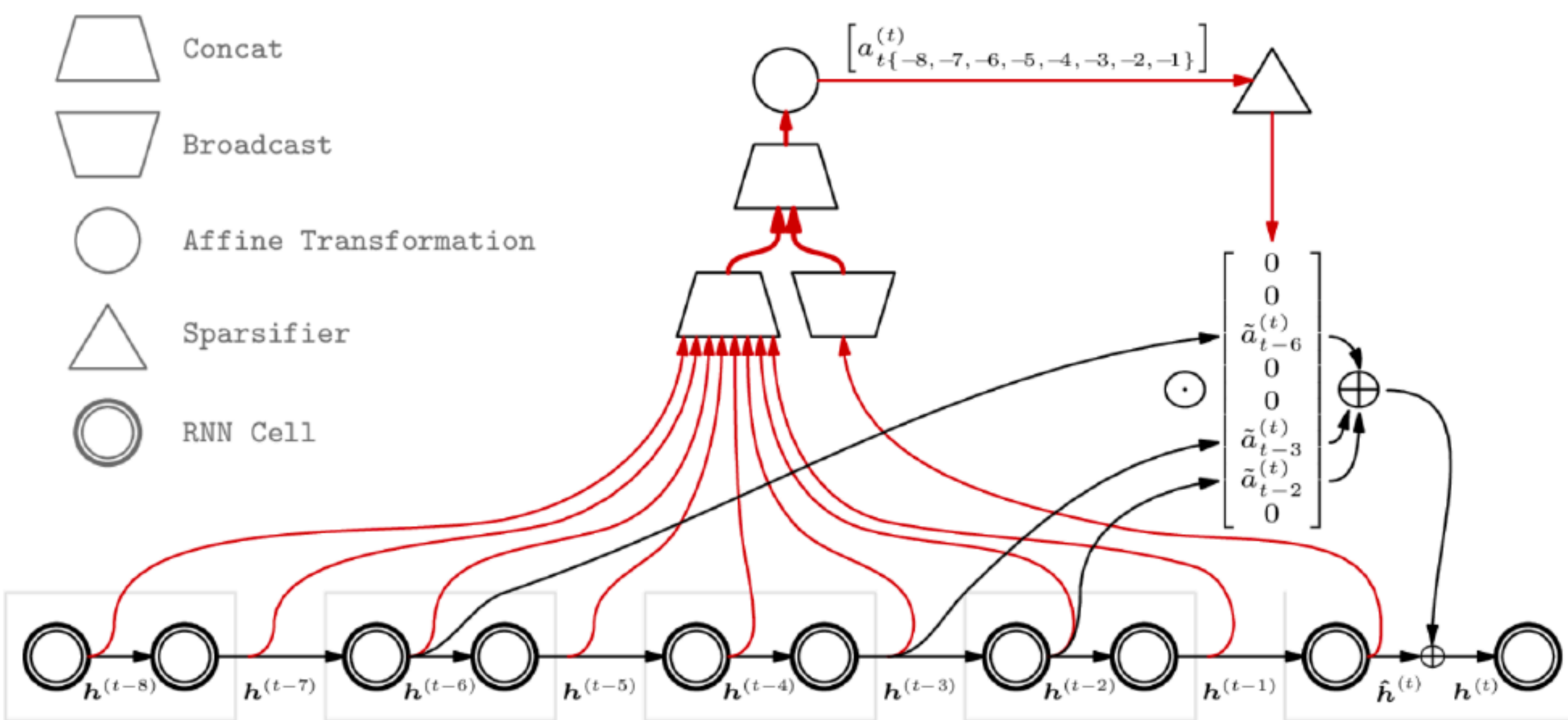
- Clipping gradients (avoid exploding gradients)
- Skip connections & leaky integration (propagate further)
- Multiple time scales / hierarchy (propagate further)
- Momentum (cheap 2_{nd} order)
- Initialization (start in right ballpark avoids exploding/vanishing)
- Sparse Gradients (symmetry breaking)
- Gradient propagation regularizer (avoid vanishing gradient)
- Gated self-loops (LSTM & GRU, reduces vanishing gradient)

Pascanu, Mikolov, Bengio, ICML 2013; Bengio, Boulanger & Pascanu, ICASSP 2013)

Using an Associative Memory to Bridge Large Time Spans and Avoid BPTT

Self-Attentive Backtracking, Ke et al NeurIPS'2018

- Associate past and present events using a predictor, which acts like a trainable attentive skip connection between associated events
- Sparse attention to select few such events



May be a way for brains to avoid implausible BPTT

Thinking, Fast and Slow

From Wikipedia, the free encyclopedia

Thinking, Fast and Slow is a best-selling[1] book published in 2011 by Nobel Memorial Prize in Economic Sciences laureate Daniel Kahneman. It was the 2012 winner of the National Academies Communication Award for best creative work that helps the public understanding of topics in behavioral science, engineering and medicine.[2]

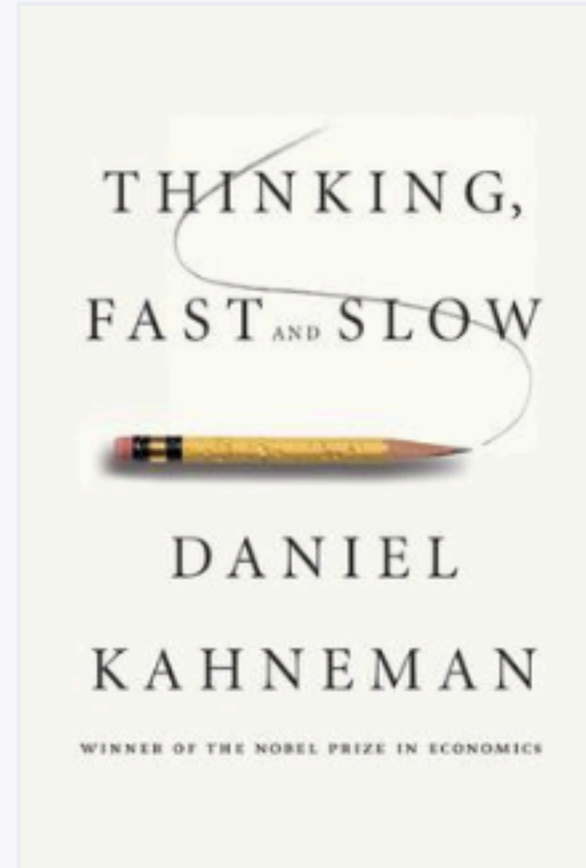
The book summarizes research that Kahneman conducted over decades, often in collaboration with Amos Tversky.[3][4] It covers all three phases of his career: his early days working on cognitive biases, his work on prospect theory, and his later work on happiness.[*not verified in body*]

The central thesis is a dichotomy between two modes of thought: "System 1" is fast, instinctive and emotional; "System 2" is slower, more deliberative, and more logical. The book delineates cognitive biases associated with each type of thinking, starting with Kahneman's own research on loss aversion. From framing choices to people's tendency to replace a difficult question with one which is easy to answer, the book highlights several decades of academic research to suggest that people place too much confidence in human judgement.[*not verified in body*]

The book also shares many insights from Kahneman's work with the Israel Defense Forces and with the various departments and collaborators that have contributed to his growth as a thinker and researcher.

Contents [hide]

Thinking, Fast and Slow



Hardcover edition

Author	Daniel Kahneman
Country	United States
Language	English language
Subject	Psychology
Genre	Non-fiction
Publisher	Farrar, Straus and Giroux
Publication date	2011
Media type	Print (hardcover, paperback)
Pages	499 pages
ISBN	978-0374275631
OCLC	706020998

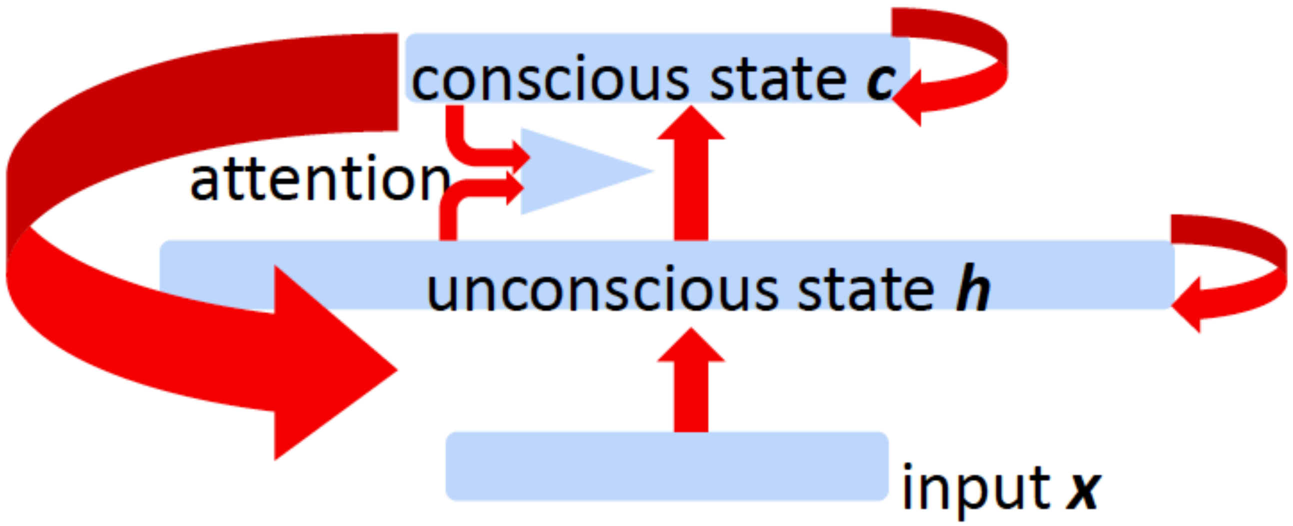
Integrating System 1 and System 2

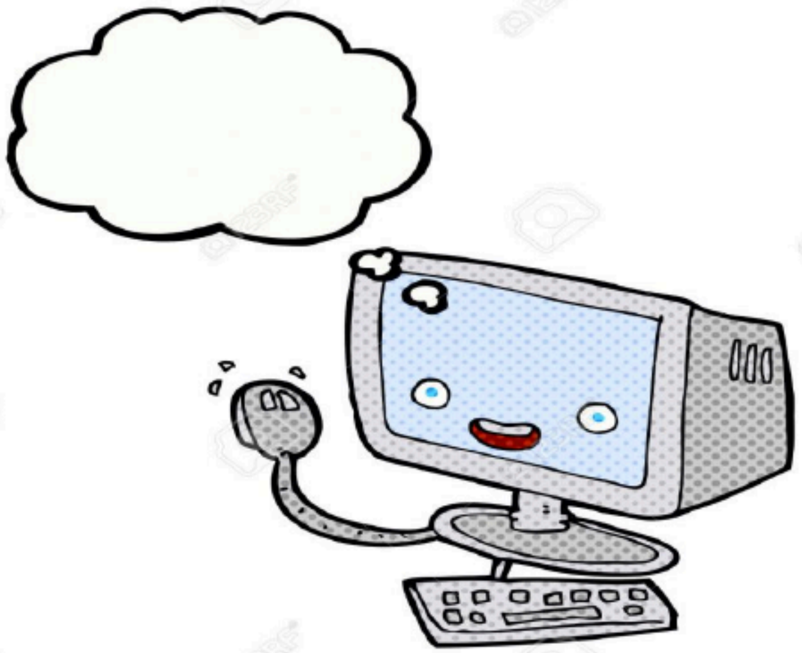
- System 2 model is very coarse and imperfect
- System 2 abstract concepts need to be grounded via system 1
- System 2 thinking allows counterfactual reasoning, i.e., imagining scenarios which did not and will not happen, as an exercise (e.g. for credit assignment, if I had done that…), allows generalization far from the training data, imagine dangerous scenarios without having to take the actual risks
- System 2 is too slow and inefficient, compile to system 1 into habits and intuitive behavior

The Consciousness Prior

Bengio 2017, arXiv:1709.08568

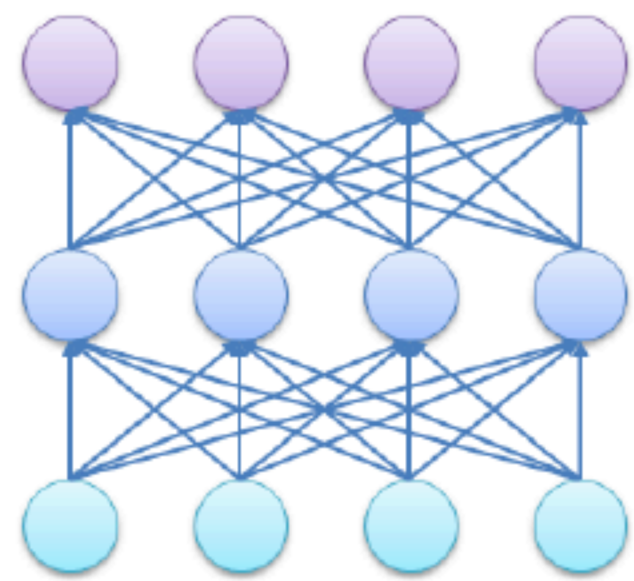
- 2 levels of representation:
  - High-dimensional abstract representation space (all known concepts and factors) ***h***
  - Low-dimensional conscious thought ***c***,





How to Discover Good Disentangled Representations

- How to discover abstractions?
- What is a good representation? ***(Bengio et al 2013)***
- Need clues (= priors) to help **disentangle** the underlying factors (**not necessarily statistically independent**), such as
  - Spatial & temporal scales
  - Approximate marginal independence
  - Simple dependencies between factors
    - *Consciousness prior*
  - Causal / mechanism independence
    - *Controllable factors*



Mila

Acting to Guide Representation Learning & Disentangling

(E. Bengio et al, 2017; V. Thomas et al, 2017)

- **Some factors (e.g. objects) correspond to 'independently controllable' aspects of the world**
  - Corresponds to maximizing mutual information between intentions (goal-conditioned policies) and changes in the state (trajectories), conditioned on the current state.
- *Can only be discovered by acting in the world*
  - *Control linked to notion of objects & agents*
  - *Causal but agent-specific & subjective: affordances*

Mila

Meta-Learning / Learning to learn

- Generalize the idea of hyper-parameter optimization
  - Inner loop optimization (normal training), a fn of meta-params

$$\theta_t(\omega) = \mathrm{approxmin}_\theta C(\theta, \omega, \mathcal{D}^t_{train})$$

  - Outer loop optimization (meta-training), optimize meta-params

$$\omega = \mathrm{approxmin}_\omega \sum_t L(\theta_t(\omega), \omega, \mathcal{D}^t_{test})$$

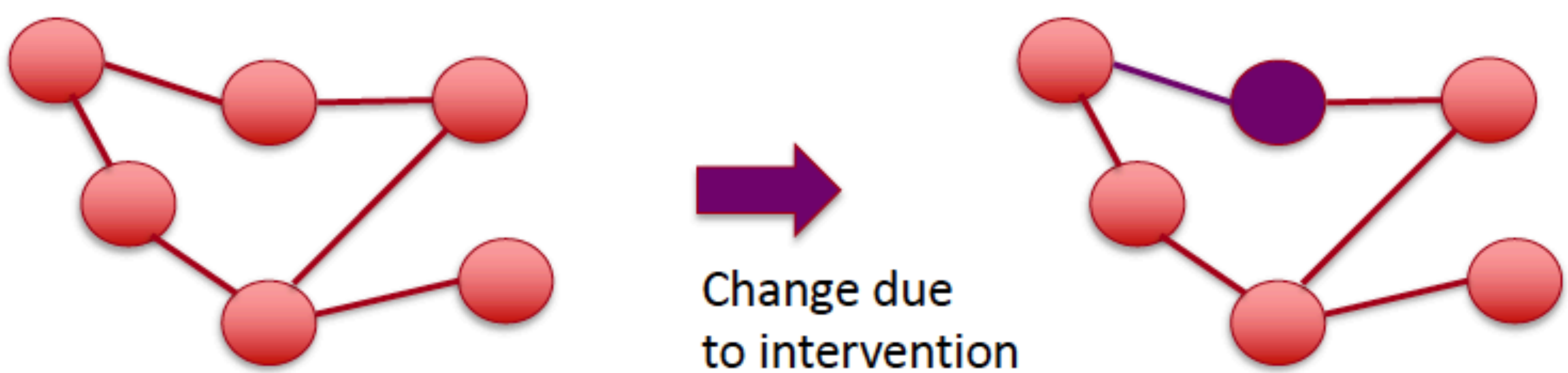
- Meta-parameters can be the learning rule itself (Bengio & Bengio 1991; Schmidhuber 1992), learn to optimize
- Meta-learn an objective or reward function, or a shared encoder
- Meta-learning can be used to learn to generalize or transfer
- Can backprop through θ_t , use RL, evolution, or other tricks

Missing from Current ML: Understanding & Generalization Beyond the Training Distribution

- Learning theory only deals with generalization within the same distribution
- Models learn but do not generalize well (or have high sample complexity when adapting) to modified distributions, non-stationarities, etc.
- Poor reuse, poor modularization of knowledge
- Meta-learning is an end-to-end learning approach to improving transfer learning and fast adaptation to changes in distribution

Separating Knowledge in Small Re-Usable Pieces

- Pieces which can be re-used combinatorially
- Pieces which are stable vs nonstationary, subject to interventions



Wrong Knowledge Factorization Leads to Poor Transfer

- With the wrong factorization P(B) P(A|B), a change in ground truth P(A) influences both modules, all the parameters
  - poor transfer: all the parameters must be adapted
- This is the normal situation with standard neural nets: every parameter participates to every relationship between all the variables
  - this causes ***catastrophic forgetting, poor transfer, difficulties with continual learning or domain adaptation, etc***

Mila

Causality not Captured by Joint Distribution

- Knowing the full joint distribution generally does not tell us which variable is a cause, which is an effect, and in general the question of causality may be meaningless in the wrong space
- Agents need to know about causal structure in order to properly infer how the joint distribution would change under interventions (theirs or of other agents)
- Understanding causal structure allows counterfactual reasoning and making sense of changes in distribution due to agents

Deep Learning Objective: discover high-level representation capturing cause and effect variables

- What are the right representations?
  - Causal variables explaining the data
- How to discover them? (learn the mythical encoder)
- How to discover their causal relationship, the causal graph?

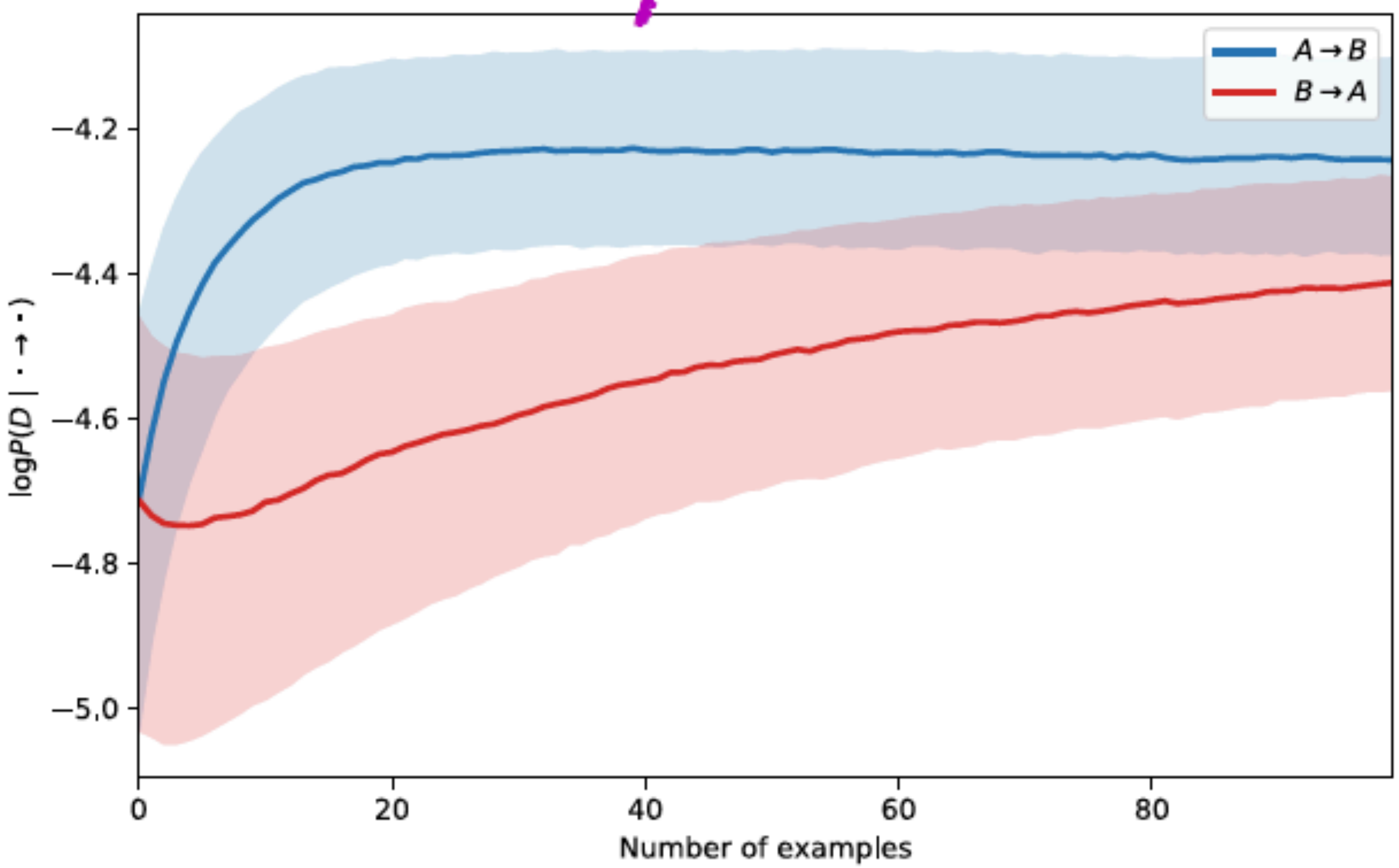
Mila

Small Change ➔
Small Sample Complexity

Few parameters need to change ➔ small L2 change ➔ *few examples needed to recover from the change*

Under the right parametrization ➔ fast adaptation to interventions

Empirical Confirmation: Correct Causal Structure Leads to Faster Adaptation



A→B is the correct causal structure: faster online adaptation to modified distribution = lower NLL regret

Mila

Turning a Hindrance into a Useful Signal

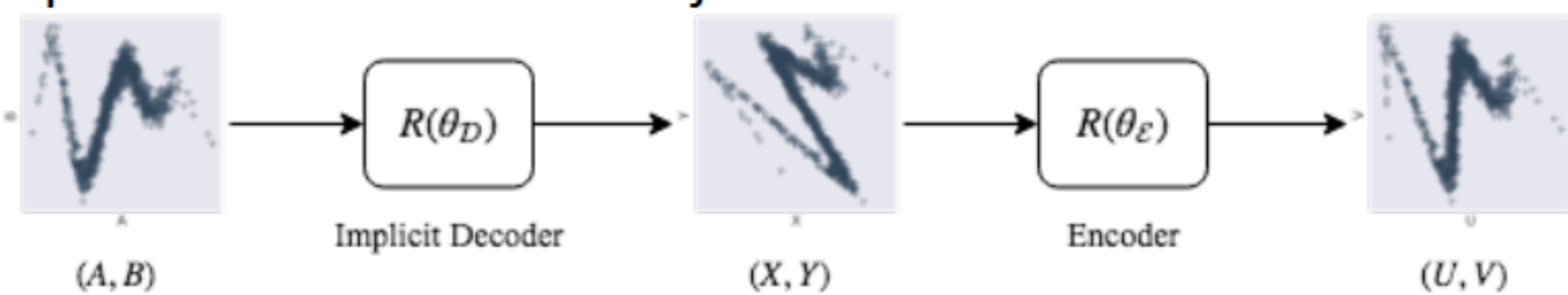
ArXiv paper, Bengio et al 2019: *A Meta-Transfer Objective for Learning to Disentangle Causal Mechanisms*

- Changes in distribution (nonstationarities in agent learning, transfer scenarios, etc) are seen as a bug in ML, a challenge
- Turn them into a feature, an asset, to help discover causal structure, or more generally to help **factorize knowledge:**
- **Tune knowledge factorization (e.g. causal structure) to maximize fast transfer**
- *"Nature does not shuffle environments, we shouldn't"*
  L. Bottou

Mila

Disentangling the Causes

- Realistic settings: causal variables are not directly observed
- Need to learn an encoder which maps raw data to causal space
- Consider both the encoder parameters and the causal graph structural parameters as meta-parameters trained together wrt proposed meta-transfer objective



Mila

Experiments successful in 2-D with simple linear mappings, Bengio et al 2019.

Meta-learning and Causality

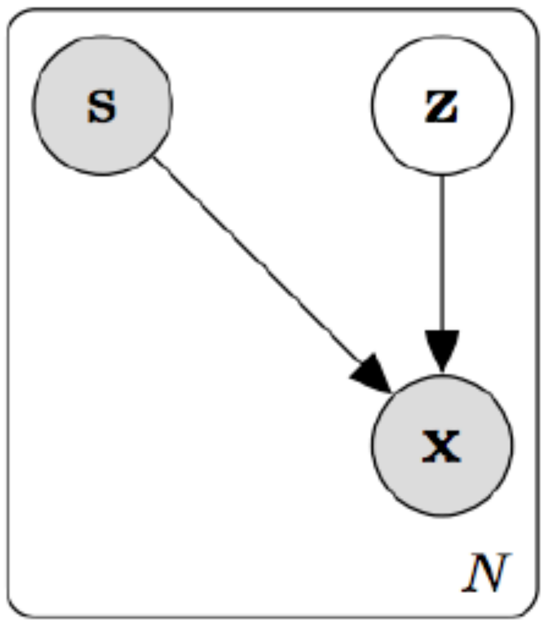
Unsupervised Learning
Jörn-Henrik Jacobsen

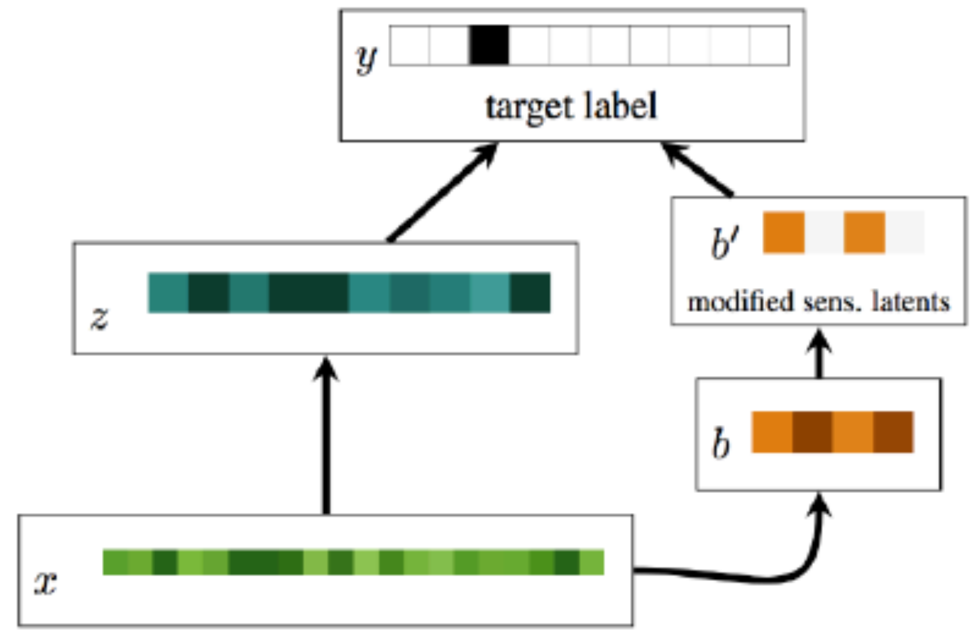
Variational AutoEncoder Applications

VQ-VAE extension, adds discrete encoder and learned prior into mix.

["Generating Diverse High-Fidelity Images with VQ-VAE-2"; Razavi et al., 2019]

Variational AutoEncoder Applications





Promising results in Fair representation learning.

One major goal: Learn representations invariant to changes of "sensitive attributes" in inputs. Can easily be phrased as constrained on latent space!

["Variational Fair Autoencoder"; Louizos et al., 2016,
"Flexibly Fair Representation Learning by Disentanglement"; Creager et al., 2019]

Autoregressive Models

Bigger models are always better	$p(x_1)$
Bigger models are always better	$p(x_2 \mid x_1)$
Bigger models are always better	$p(x_3 \mid x_2, x_1)$
Bigger models are always better	$p(x_4 \mid x_3, x_2, x_1)$
Bigger models are always better	$p(x_5 \mid x_4, x_3, x_2, x_1)$

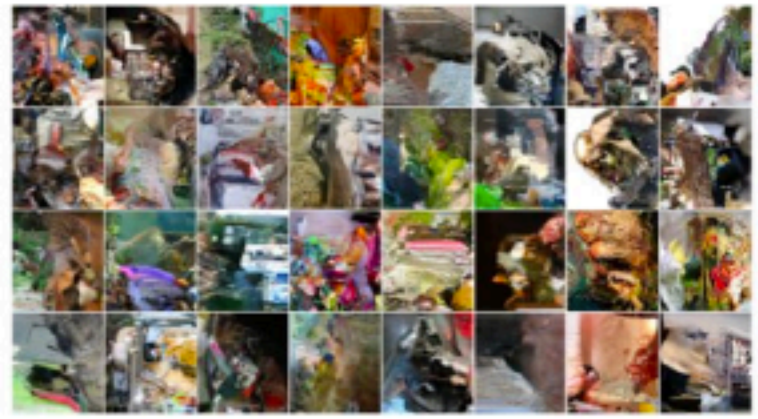
NLL Objective:

$$L(\theta, D) = \sum_{x \sim P_{data}} \sum_{i=1}^{N} -\log p_\theta(x_i \mid x_{<i})$$

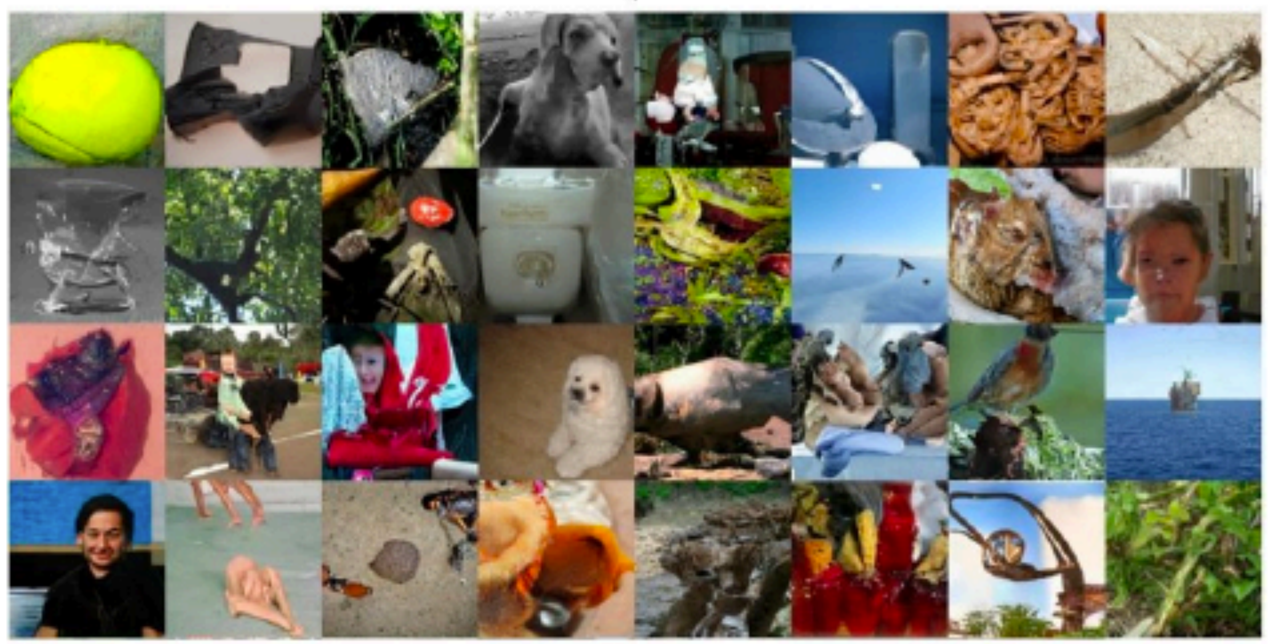
In practice, probability is discrete softmax output, similar to training classifier.

Conditioning is done via network state, e.g. RNNs, LSTMs, masked convolutions or Transformers.

Autoregressive Models



PixelRNN (2016) ImageNet 64 x 64

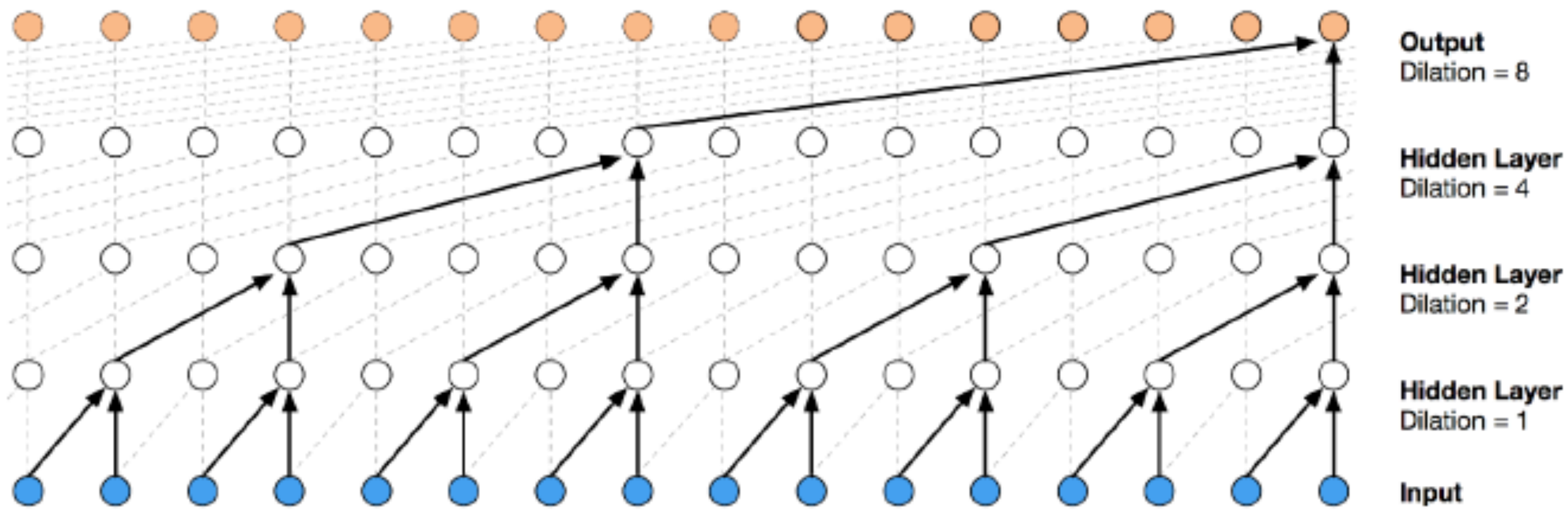


SPN (2018) ImageNet 128 x 128

Main difference between PixelRNN and SPN: dimension ordering, suitable for data domain (+ improved models)!

["Generating high fidelity images with subscale pixel networks and multidimensional upscaling"; Menick et al., 2018]

Autoregressive Models



WaveNet uses dilated causal convolutions to model audio autoregressively, large receptive field important due to high temporal resolution of raw audio!

Autoregressive models have been applied to many more modalities like language, tabular data …

[“WaveNet: A Generative Model for Raw Audio”; Van den Oord et al., 2016]

Unifying Discriminative and Generative Models

Autoregressive models have very different structure than the state-of-the-art in discriminative tasks -> ***i.e. ResNet-like architectures***

Why does this matter?

If model does not work well in discriminative setting, probably won't provide gain when pre-trained unsupervised

There is a family of generative models which is quite similar to the best supervised models: Normalizing Flows!

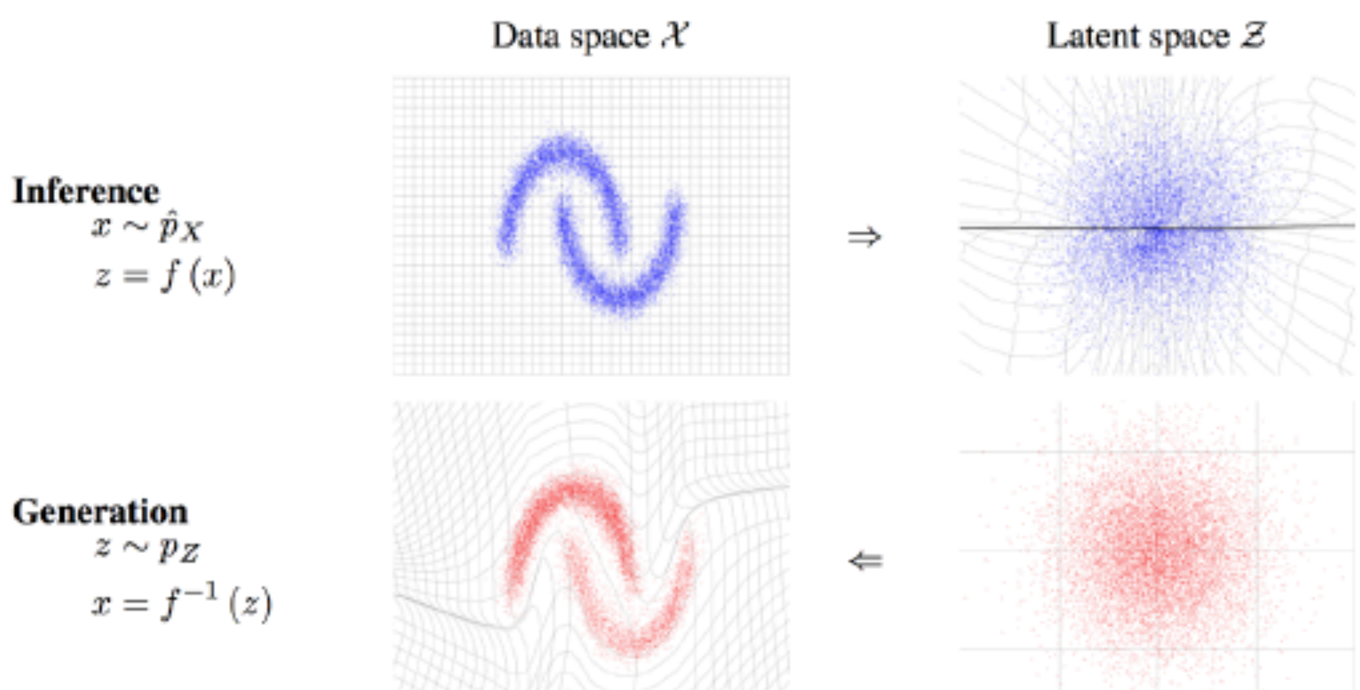
Simple and elegant way to train with exact maximum likelihood via change-of-variable formula:

$$\log P_X(x) = \log P_Z(z) + \log |\det J_F(x)|$$

<u>Normalizing Flow idea:</u> specify *simple base distribution in output space* and compute *complex data-likelihood in input space via change-of-variable formula.*

Problems:

1) We need expressive bijective deep networks
2) Log determinant of Jacobian of this network needs to be easy to compute
3) Sampling from model requires tractable inverse

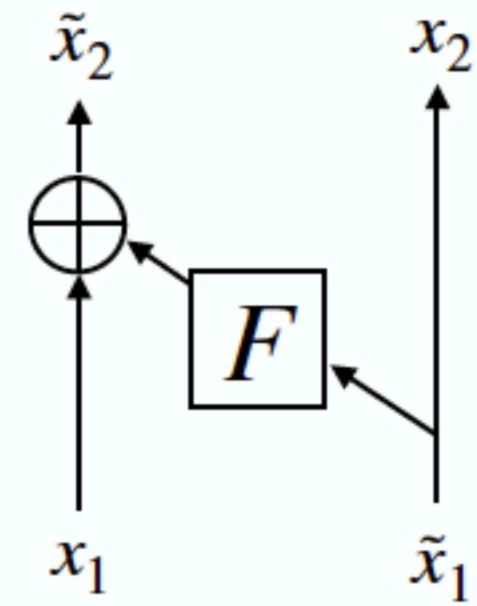


Nonlinear Independent Component Analysis

NICE builds invertible network as composition of volume conserving invertible blocks.

Recall ResNet block:

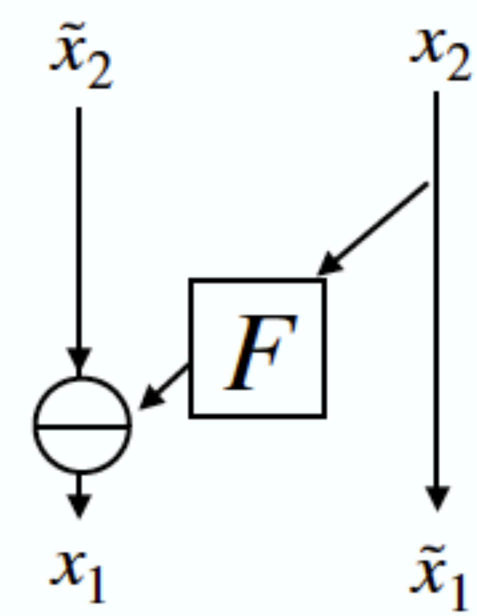
$$x_2 = x_1 + F_\theta(x_1)$$



“Coupling Block” differs in dimension splitting:

$$x_2 = \tilde{x}_1$$

$$\tilde{x}_2 = x_1 + F_\theta(\tilde{x}_1)$$



Inverse is simple and efficient:

$$\tilde{x}_1 = x_2$$

$$x_1 = \tilde{x}_2 - F_\theta(\tilde{x}_1)$$

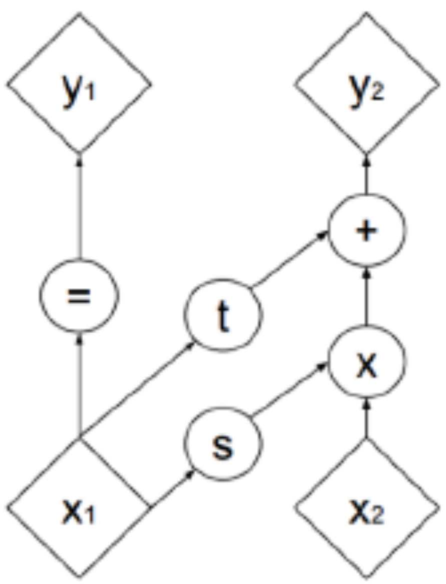
Jacobian is triangular with ones on diagonal, NICE is volume conserving:

$$\log|det J_F(x)| = 0$$

[“Nonlinear Independent Component Analysis”; Dinh et al., 2014]

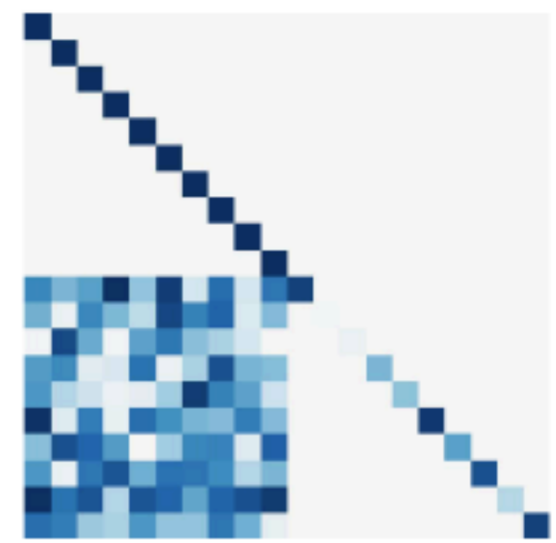
Real-NVP: Non-volume Conserving NICE

Main idea is same as NICE, but additive coupling block is extended to affine transformation and multi-scale architecture is introduced.



$$\begin{cases} y_{1:d} & = x_{1:d} \\ y_{d+1:D} & = x_{d+1:D} \odot \exp\big(s(x_{1:d})\big) + t(x_{1:d}) \end{cases}$$

$$\Leftrightarrow \begin{cases} x_{1:d} & = y_{1:d} \\ x_{d+1:D} & = \big(y_{d+1:D} - t(y_{1:d})\big) \odot \exp\big(-s(y_{1:d})\big) \end{cases}$$



Just like NICE, model is trained by minimizing negative data log-likelihood, where log det term is sum over log of diagonal entries:

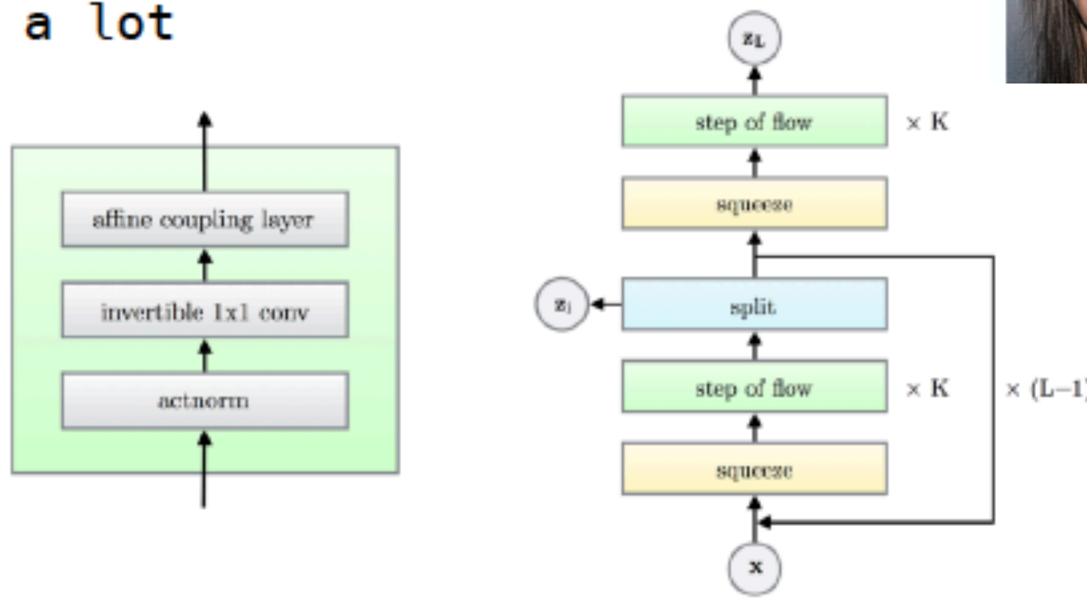
$$L(\theta, D) = \sum_{x \sim P_{data}} - \log p_Z(f_\theta(x)) - \log |det J_f(x)|$$

["Density estimation using Real NVP"; Dinh et al., 2016]

GLOW

Glow improved upon Real-NVP in multiple ways:

1) Learn dimension splitting with 1x1 convolutions
2) Removes batch dependency due to Batchnorm
3) Scales models up a lot



(a) One step of our flow. (b) Multi-scale architecture (Dinh et al., 2016).

Table 2: Best results in bits per dimension of our model compared to RealNVP.

Model	CIFAR-10	ImageNet 32x32	ImageNet 64x64	LSUN (bedroom)	LSUN (tower)	LSUN (church outdoor)
RealNVP	3.49	4.28	3.98	2.72	2.81	3.08
Glow	**3.35**	**4.09**	**3.81**	**2.38**	**2.46**	**2.67**

Results start to close gap to autoregressive models and also showed that stability as well as hand-designed dimension splitting are major bottlenecks in flow models.

[“Glow: Generative Flow with Invertible 1x1 Convolutions”; Kingma et al., 2018]

Taking it Further: Invertible Residual Networks

Can we guarantee invertibility in SOTA non-invertible networks?

-> i.e. without dimension splitting

Remarkable similarity between ResNets and Euler's method for ODE initial value problems:

$$x_{t+1} = x_t + g_{\theta_t}(x_t) \quad \text{ResNet Update}$$

$$x_{t+1} = x_t + hf_{\theta_t}(x_t) \quad \text{Euler Update}$$

Dynamics backward in time are given by implicit backward Euler discretization:

$$x_t = x_{t+1} - g_{\theta_t}(x_t)$$

$$x_t = x_{t+1} - hf_{\theta_t}(x_t)$$

["Invertible Residual Networks"; Berhmann et al., 2019]

Invertible Residual Networks

Compute inverse via fixed-point iteration:

Inverse:
$$x^0 = y$$
$$x^{i+1} := y - g(x^i)$$

Sufficient condition for invertibility is contraction:

$$Lip(g) < 1$$

Guarantees convergence of iteration due to Banach fixed point theorem!

[“Invertible Residual Networks”; Berhmann et al., 2019]

Invertible Residual Networks

Original:

Inverse | Unconstrained:

Inverse | Lip(g) < 1:

Original:

Inverse | Unconstrained:

Inverse | Lip(g) < 1:

MNIST even invertible without constraint!

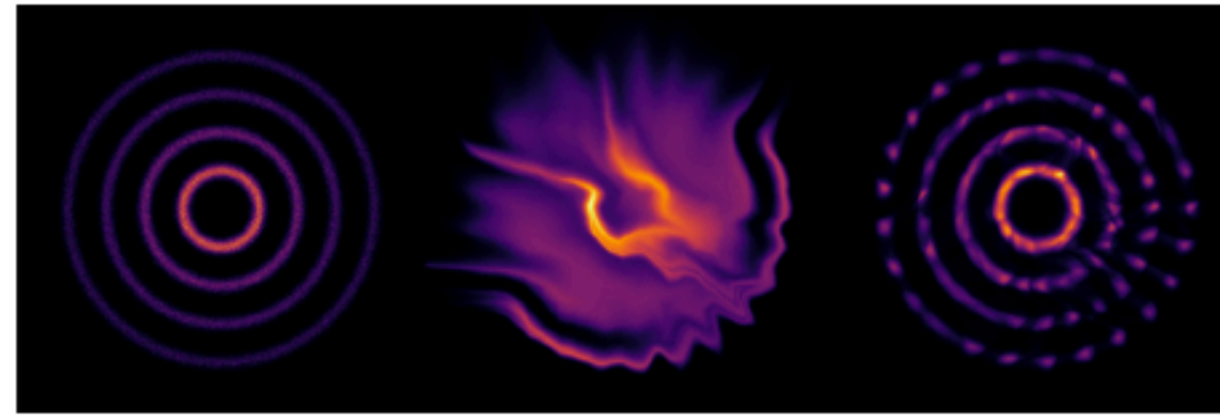
Performance not significantly affected in all cases

Affine Glow 1×1 Conv	Additive Glow Reverse	i-ResNet Glow-Style	i-ResNet 164
12.63	12.36	8.03	6.69

Classification accuracy Cifar-10

[“Invertible Residual Networks”; Berhmann et al., 2019]

Residual Flows / Invertible ResNets



Data Samples Glow i-ResNet

Model	MNIST	CIFAR-10	ImageNet 32×32	ImageNet 64×64
Real NVP (Dinh et al., 2017)	1.06	3.49	4.28	3.98
Glow (Kingma and Dhariwal, 2018)	1.05	3.35	4.09	3.81
FFJORD (Grathwohl et al., 2019)	0.99	3.40	—	—
Flow++ (Ho et al., 2019)	—	**3.29** (3.09)	— (3.86)	— (3.69)
i-ResNet (Behrmann et al., 2019)	1.05	3.45	—	—
Residual Flow (Ours)	**0.97**	**3.29**	**4.02**	**3.78**

Competitive or better than state-of-the-art flow models, while being based on standard ResNets!

Method of choice for all applications where competitive density estimation performance *and* strong discriminative downstream performance is important.

[“Residual Flows for Invertible Generative Modeling”; Chen et al., 2019]

Summary: Likelihood-based Density Modeling

<u>**3 Common Approaches**</u>

Variational Autoencoders:

Optimize tractable lower bound on data likelihood

$$\log p(x) \leq \mathbb{E}_{z \sim q}\left[\log p(x \mid z)\right] - D_{KL}(q(z \mid x) \mid\mid p(z))$$

Autoregressive Models:

Apply chain rule of probability and break up N-dimensional modeling problem into N 1-dimensional conditional densities

$$p(x) = p(x_1, x_2, \ldots, x_N) = \prod_{i=1}^{N} p(x_i \mid x_{i-1}, x_{i-2}, \ldots, x_1)$$

Normalizing Flows:

Design models with tractable Jacobian determinant and match transformed data to simple base distribution with change-of-variable formula

$$\log p(x) = \log p(z) + \log \lvert \det J_F(x) \rvert$$

Summary: Likelihood-based Density Modeling

Variational Autoencoders:

Pro: Very flexible, few constraints on encoder / decoder / prior
Contra: Posterior collapse, often lower NLL and worse samples than exact likelihood models

Autoregressive Models:

Pro: Simple yet powerful approach, very good performance in NLL, applicable to any data modality
Contra: Models very different from discriminative models, sampling is slow

Normalizing Flows:

Pro: Simple objective, SOTA ResNets can be turned into flows with one additional constraint, excellent performance on joint discriminative and generative tasks
Contra: Need to design invertible networks with tractable Jacobian determinant

Nonlinear ICA bases on invertible normalizing flows
or neural networks?

Materials

- Hands-on Deep Learning was based on
  - www.coursera.org/specializations/deep-learning
- Hugo Larochelle http://info.usherbrooke.ca/hlarochelle/neural_networks/content.html
- CS231n: Convolutional Neural Networks for Visual Recognition
  - cs231n.stanford.edu
- Hands-on Reinforcement Learning
  - www.coursera.org/specializations/reinforcement-learning
- Bandits book https://banditalgs.com/

Bandits on a single slide

- K "arms"
- Go for $n > 0$ rounds
- In any given round:
  - Choose an arm
  - Arm "generates" a "reward"
  - Observe generated reward
  - Repeat
- Goal: Maximize total reward

Drug 1	Drug 2	Drug 3
0	NA	NA
NA	1	NA
NA	NA	1
1	NA	NA
?	?	?

Medical interventions

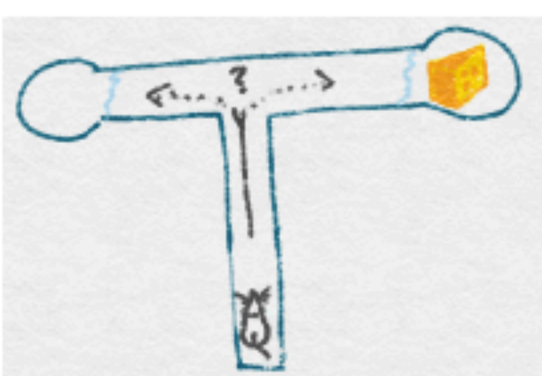


Image credit - Pundimon

Applications

Drug 1	Drug 2	Drug 3
0	NA	NA

1. A/B testing
2. Drug testing
3. Advert placement
4. Network routing (packets, planes, cars)
5. Tree search (MCTS)
6. Recommendation services (for example, news or movies)
7. Ranking (for example, search)
8. Educational games
9. Resource allocation (memory, bandwidth, manufacturing space)
10. Waiting problems (hard-disk shutdown, auto logout, waiting for a bus)
11. Dynamic pricing (for example, on Amazon)
12. A core component of RL

From Rich Sutton about Research and RL

The many dimensions of RL

Increasing in difficulty to the right →

- Problem dimensions
  - Prediction — control
  - Bandits — MDPs
  - Discounted — episodic — average reward
  - Fully observable — partially observable
  - Empirical results — convergence theory— rate theory

The frontier??

Human-in-the Loop ???

- Method dimensions
  - Function approximation: tabular — state aggregation — linear — nonlinear
  - Model-free — model-based
  - On-policy — off-policy (Gradient-TD — Emphatic-TD — Tree backup — Q(σ) — V-trace)
  - Bootstrapping: Monte Carlo — temporal difference learning
    - Unified treatment by n-step methods — eligibility traces
    - Trace type: Accumulating—replace—dutch—true online
  - Value-based — policy-based
  - State values — action values
  - Batch — online

- Options
- Distributional
- Double and triple methods
- Interest and emphasis

There are no authorities in science

- Don't be impressed by what you don't understand
- Don't try to impress others by what they don't understand
- You should be brave and ambitious…
  ...but also humble and transparent
- Humble before the great task — understanding the mind
  - nature is subtle but not devious
  - it is waiting to be discovered… if we can only see it

Your thoughts are, potentially, of great value

How can you train yourself to think carefully & productively?

The best way is to write for yourself

(and discuss with others)

They say it takes 10,000 hours
to become an expert at anything

This could well be true
for thinking about thinking

Are you willing to do the work?

It is not super difficult,
but you do have to show up, day after day

45 years of
my notebooks

A prose poem for your notebook

To write is to begin to think.

To write in a special place,

—a book such as this—

is to honor your thoughts

and to help them build,

one upon the other.

When you get stuck, persist

- In thinking on important questions, you will often reach an apparent dead end, with no where to go
- Here are some techniques for moving forward again:
  - Define your terms
  - Go multiple (What are some of the conceivable answers?)
  - Go meta (What would an answer look like? What properties would it have?)
  - Retreat (to a clearer question that you *can* make progress on)

What is intelligence?

- “Intelligence is the most powerful phenomenon in the universe”

  —Ray Kurzweil

- “Intelligence is the computational part of the ability to achieve goals in the world”

  —John McCarthy

- “Intelligence is in the eye of the beholder”

The predictive knowledge hypothesis

"Almost all knowledge of the world can be well thought of as statistics (predictions) about the agent's future data stream"

Exceptions:

- mathematical knowledge
- knowledge of what to do (policies)
- features
- memories of the past

The most important insight you will ever contribute

- Is probably something that you already know
- Is probably something that is obvious to you
  - *so obvious that you can't see it!*

Sometimes the obvious is the hardest to see. For example:

- The discovery of gravity, by Isaac Newton
- The discovery that people are animals, evolved from animals, by Charles Darwin
- The discovery of air/vacuum
- The discovery of reinforcement learning by Harry Klopf in the 1970s

Harry Klopf
1941–1997

Are there obvious things that we struggle to see now?

- No animal does supervised learning
- No mind generates images or videos
- Neural networks are not in any meaningful sense “neural”
- People are machines
- The purpose of life is pleasure (and pain)
- The world is much more complex than any mind that tries to understand it
  - therefore, a prior distribution on the world could never be reasonable
- Mind is computational, and computation is increasing exponentially
- Human input doesn’t scale; the only scalable methods are search and learning

Thank you