Machine Learning Techniques for Neuroscience

Tutorial for Cog. Comp. Neuroscience Summer School 2023

Li Kevin Wenliang

Google DeepMind

August 5, 2023

Machine learning in/for/from/and neurocience

Today's overview

1. Modern machine learning techniques
2. Applications of machine learning for neuroscience
3. Neuroscience inspirations for machine learning (on very high level)

Modern machine learning godview

An (almost) universal description for machine learning:

$$\min_{f\in\mathcal{M}} \mathcal{L}_{\text{tr}}(f,\mathcal{D}_{\text{tr}}) \quad \text{so that} \quad \mathcal{L}_{\text{eval}}(f,\mathcal{D}_{\text{eval}}) \text{ is small,} \quad \text{where } \mathcal{D}_{\text{tr}},\mathcal{D}_{\text{eval}} \sim \mathcal{S}$$

- f: a model or a function
- $\mathcal{M}$: the class of model
- $\mathcal{S}$: task paradigm
- $\mathcal{D}_{\text{tr}}$ and $\mathcal{D}_{\text{eval}}$: training and evaluation datasets
- $\mathcal{L}_{\text{tr}}$: training objective
- $\mathcal{L}_{\text{eval}}$: is final evaluation criterion

Categorisation of different approaches:

By goal f and data $\mathcal{D}_{\text{tr}}$

- Supervised
  $f:\mathcal{X}\to\mathcal{Y}$, $\mathcal{D}=\{x_i,y_i\}$
- **Unsupervised / self-supervised**
  $f:\mathcal{X}\to\mathcal{Z}$, $\mathcal{D}=\{x_i\}$
- Reinforcement
  $f:\mathcal{X}\to\mathcal{A}$,
  $\mathcal{D}_{\text{tr}}$ collected from f

By model space, $\mathcal{M}$

- Parametric models: polynomials, splines, radial basis
- Nonparametric models: k-NN, decision tree, kernel methods,
- **Neural networks: CNN, RNN, GNN transformers...**

By task paradigm $\mathcal{S}$

- Multiple objectives
- Transfer / causal learning
- Online / continual / active learning
- Meta-learning

Related fields: mathematics, optimization, engineering, statistics, domain knowledges

Supervised learning

Recall image classification

Dataset $\mathcal{D}_{\text{tr}} := \{x_i, y_i\}_1^N$ where $x_i \in \mathcal{X} := \mathbb{R}_+^{w\times h\times c}$ is a vector of image pixels, $y_i \in \mathcal{Y} := \mathbb{1}_K$

$$\Longrightarrow \underbrace{\begin{bmatrix} 213 \\ 255 \\ \dots \\ 22 \end{bmatrix} \Big\} w \times h \times c}_{x_1}$$

$$\Longrightarrow \underbrace{\begin{bmatrix} 12 \\ 25 \\ \dots \\ 9 \end{bmatrix} \Big\} w \times h \times c}_{x_2}$$

"cat" $\Longrightarrow \underbrace{\begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \end{bmatrix} \Big\} K}_{y_1}$

"dog" $\Longrightarrow \underbrace{\begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \end{bmatrix} \Big\} K}_{y_2}$

$f : \mathbb{R}_+^{w\times h\times c} \to \Delta_K \qquad \mathcal{L}_{\text{tr}}(f, \mathcal{D}_{\text{tr}}) := \frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log f(x_i) \qquad \log(\cdot)$ is elementwise.

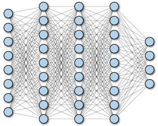
Supervised learning with neural networks

Supervised learning can solve the following problems

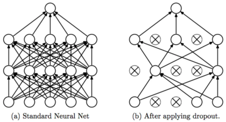
	image cls.	speech recog.	translation	gait recog.	image seg.	scene parsing
$\mathcal{X}$	$\mathbb{R}_+^{w\times h\times c}$	$\mathbb{R}^t$	$\mathbb{1}_K^L$	$\mathbb{R}^{T\times n\times 3}$	$\mathbb{R}_+^{w\times h\times c}$	$\mathbb{R}_+^{w\times h\times c}$
$\mathcal{Y}$	Δ_K	Δ_V^τ	Δ_V^τ	Δ_K	$\Delta_K^{w\times h\times c}$	$\{\Delta_K, \mathbb{N}^4\}_{m=1}^M$

- Machine supervised learning is a trivial problem to some. But is it?
- Most deep learning techniques and tricks are discovered through supervised learning
- Becoming a test bed for benchmarking theory and techniques (e.g. tricks)

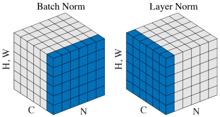
Key (overlapping) ingredients in machine learning



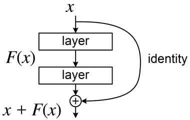
width vs depth



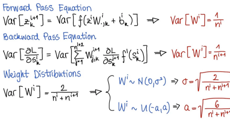
regularisation



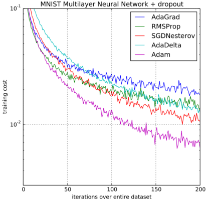
normalisation



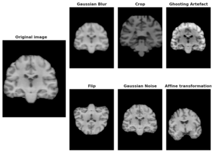
architecture



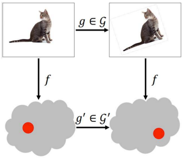
initialisation



optimization

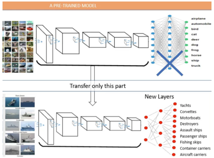


data augmentation

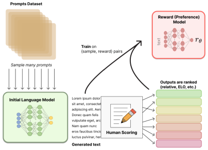


$f(g(\mathbf{x})) = g'(f(\mathbf{x}))$

equivariance



finetuning

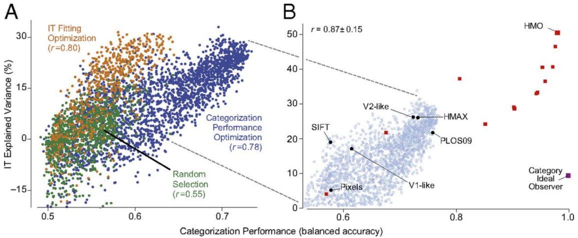


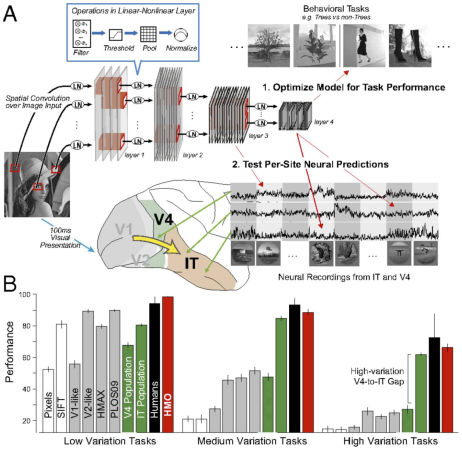
finetuning

A lot remains to be discovered, explained and improved...

Applications to neuroscience: models of vision

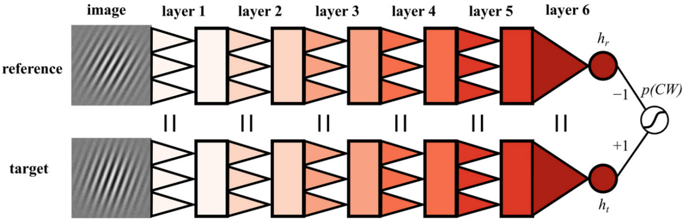
Supervised deep models show similarities to primate visual ventral stream (Yamins et al., 2014)

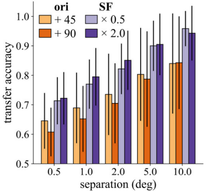


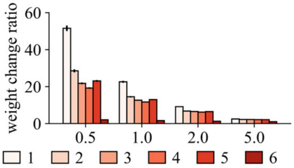


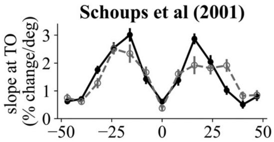
Applications to visual perceptual learning

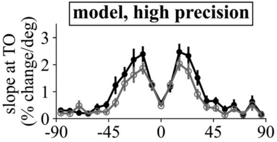
Supervised training replicates findings in visual plasticity on different analysis levels (Wenliang & Seitz, 2018)











Unsupervised learning

Goal: discover *useful* representation of complex data for downstream tasks
Quantifiable metrics $\mathcal{L}_{\text{eval}}$: outlier detection, generative quality, compression, transfer tasks, etc.

	clustering	**dim. reduction**	**manifold**	**representation**	**generation**
$\mathcal{X}$	$\mathbb{R}^n$	$\mathbb{R}^n$	$\mathbb{R}^n$	$\mathbb{R}^n$	$\mathbb{R}^n$
$\mathcal{Z}$	$\mathbb{1}_m$ or Δ_m	$\mathbb{R}^m$, $m < n$	$\mathbb{S}^m$, trees, etc.	$\mathbb{R}^m$	$\mathbb{R}^m$
$\mathcal{L}_{\text{tr}}$	distances density	reconstruction	reconstruction + prior	density + coarse labels	distributional metrics, denoising
$\mathcal{L}_{\text{eval}}$	visualisation, classification, outlier detection	reconstruction denoising	interpolation homology generation	classification generation	sample quality inpainting interpolation

Deep learning methods for unsupervised learning

We briefly review the objectives and intuitions of the following approaches

1. Variational autoencoders (VAE)
2. Generative adversarial networks
3. Constractive pre-training

Latent variable model

Definition

Given dataset $\mathcal{D} := \{x_i\}_{i=1}^N$, a latent variable models (LVM) posits that each data point $x_i \in \mathcal{X}$ is generated from a latent variable $z_i \in \mathcal{Z}$ through a model parametrised by θ

$$z_i \xrightarrow{\theta} x_i$$

Example

Linear model: data generated by a linear mapping $G \in \mathbb{R}^{d\times k}$, where $k < d$

$$x_i = Gz_i + \epsilon_i$$

Interpretation of latent variable models:

- z_i is **specific** to each data instance x_i
- θ captures **overall** patterns for the whole dataset
- alternatively, z_i is a **local** parameter for x_i, and θ is a **global** variable for $\mathcal{D}$.

Generative latent variable model

To let the z_i be controllabe/interpretable, we place a prior $p_\theta(z)$

Example

Prior $p(z)$ can be

$\mathcal{N}(0,1)$	Laplace	uniform circular	Bernoulli	hyperbolic	Markov chain
common choice	sparsity priors	rotation-symmetry	discrete	hierarchical	time-series

Likewise, we can specify a flexible and learnable mapping $G : \mathcal{Z} \rightarrow \mathcal{X}$

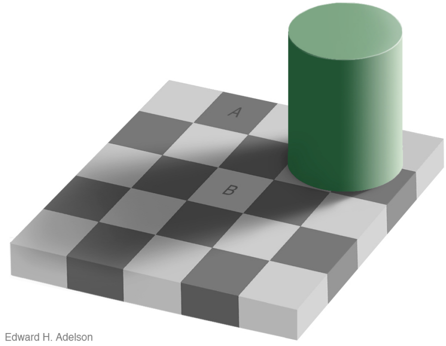
Example

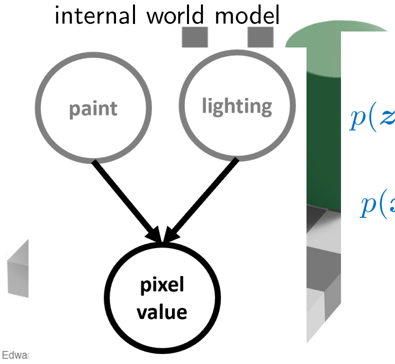
The likelihood $p(x|z)$ can be

$x = Az + \epsilon$	$x = G_\theta(z) + \epsilon$	$z_0 \rightarrow h_1, z_1 \rightarrow \cdots \rightarrow x$	$z, y \rightarrow x$
linear + noise	nonlinear + noise	hierarchical	conditional

The joint distribution $p_\theta(x, y) = p_\theta(z)p_\theta(x|z)$ induces a posterior $p(z|x)$ through Bayes rule.

Generative model: applications to cognitive science





generative, internal

$p(z)$ $\boldsymbol{z}$

$p(\boldsymbol{x}|\boldsymbol{z})$ $\downarrow$

$\boldsymbol{x}$

Inferential, recognition

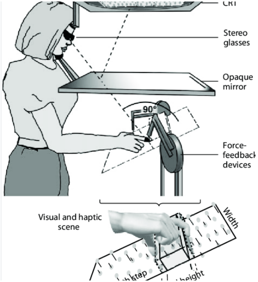
$q(\boldsymbol{z}|\boldsymbol{x})$

$\uparrow$

$\boldsymbol{x}$

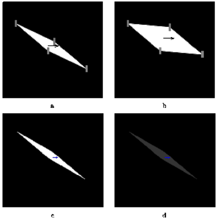
Generative model: applications to perception

cue combination



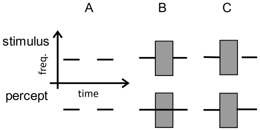
Ernst & Bank, 2002

motion illusion

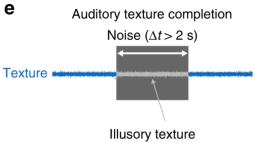


Weiss et al, 2005

continuity illusions

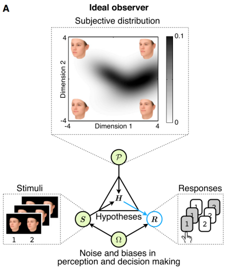


Green & Swets, 1966



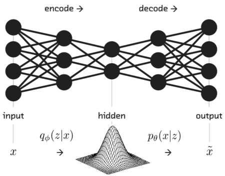
McWalter & McDermott, 2019

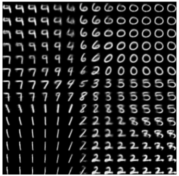
visual prior

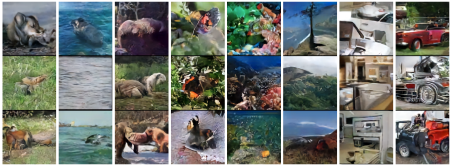


Houlsby, et al, 2013

The variational autoencoder (VAE) and other variants







The variational autoencoder trains the likelihood $p_\theta(x|z)$ and an encoder $q(z|x)$ jointly

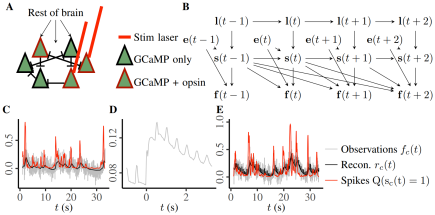
$$\mathcal{L}_{\text{tr}}(\theta, q; x) := \underbrace{\mathbb{E}_{z\sim q(z|x)}\left[\log p_\theta(x|z)\right]}_{\text{expected recon. loss}} - \underbrace{\mathbb{D}\left[q(z|x)||p(z)\right]}_{\text{prior constraint}},$$

where $\mathbb{D}$ is come distributional distances.

- deterministic $q(z|x)$ and zero $\mathbb{D} \implies$ conventional nonlinear autoencoder
- Gaussian $p(z)$, $p_\theta(x|z)$ and $q(z|x)$, $\mathbb{D} = \mathbb{KL} \implies$ VAE $\mathcal{L}_{\text{tr}}(\theta; x) \leq \log p_\theta(x)$ (Kingma & Welling, 2014; Rezende et al. 2014)
- Gaussian $p(z)$, deterministic $p_\theta(x|z)$ and $q(z|x)$, $\mathbb{D}$ is $\mathcal{W}_2 \implies$ Wasserstein AE (Tolstikhin et al. 2017)
- $\mathbb{D} = \beta\mathbb{KL} \implies$ beta-VAE (Higgins et al. 2017)
- discrete $q(z|x)$ and vector-quantization loss $\mathbb{D} \implies$ VQ-VAE (Oord et al., 2018)
- Separate network $q(z|x)$ trained by sample from $p(z, x) \implies$ Helmholtz machine and wake-sleep algorithm (Dayan et al., 1994, Hinton et al., 1995)
- Implicit $q(z|x)$ by nonlinear moments $\Rightarrow$ biologically plausible training (Vertex & Sahani 2018, Wenliang & Sahani 2019)

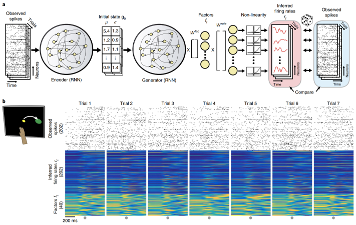
VAE: applications to neural data analysis

all-optic interogation



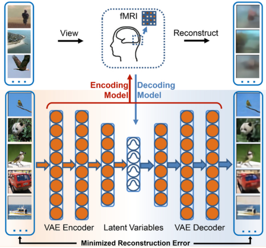
Aitchison et al., 2017

LFADS



Pandarinath et al., 2018

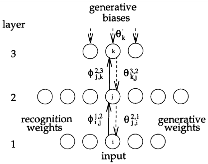
image decoding



Han et al., 2019

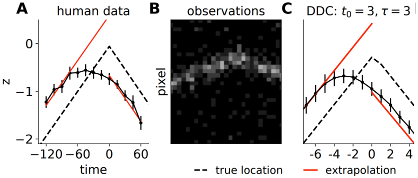
Wake-sleep algorithms

precursor of VAE



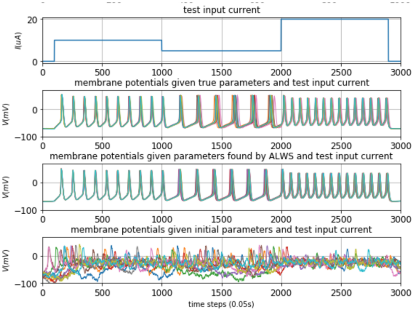
Dayan et al., 1994

dynamic postdiction



Wenliang & Sahani., 2019

training HH models with kernel



Wenliang et al., 2020

Implicit models

Definitions

Implicit generative model defines a prior $p(z)$ and a deterministic mapping $G_\theta : \mathcal{Z} \to \mathcal{X}$.

The only randomness is in the prior: a latent z maps directly to x, no additional noise.

Example

Differential eqns: Wilson-Cowan, Hodgkin-Huxley models and attractor models.

Technicality: the generative distribution may be supported on a lower-dimensional subspace. The likelihood of $p_\theta(x)$ may be ill-defined for a given data point x.

Optimising distributional distances

Fitting a generative distribution requires a **distributional distance**

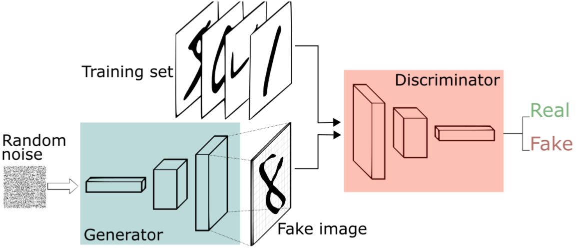
- Maximising the log-likelihood is equivalent to minimising the Kullback-Leibler divergence

$$\mathbb{KL}\left[q \| p\right] = \int q(x) \log \frac{q(x)}{p(x)} \mathrm{d}x = \int q(x) \log q(x) \mathrm{d}x - \int q(x) \log p(x) \mathrm{d}x$$

- The first version of GAN (Goodfellow, 2014) optimises the Jensen-Shannon divergence

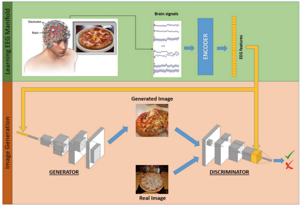
$$\mathbb{JS}\left[q \| p\right] = \frac{1}{2}\mathbb{KL}\left[q \| \frac{1}{2}(p+q)\right] + \frac{1}{2}\mathbb{KL}\left[p \| \frac{1}{2}(p+q)\right]$$

- Later GANs optimises other objectives: MMD-GAN, Cramer-GAN, optimal transport GAN, Wasserstein GAN, f-divergence GAN, etc.

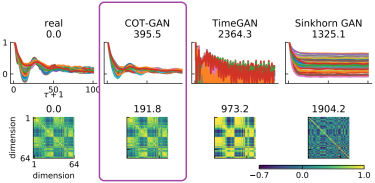


GAN for neuroscience

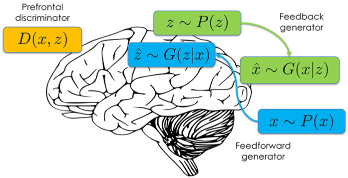
GANs have not made much applications in neuroscience...



Palazzo et al., 2017



Xu, Wenliang et al., 2020

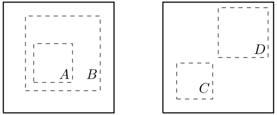


Gershman 2019

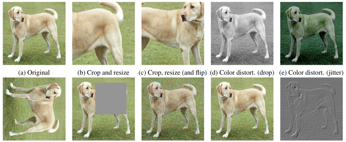
Contrastive self-supervised learning

Can we just learn representation without generating the data?

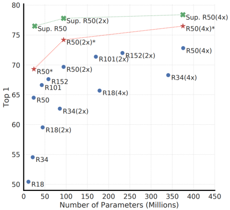
Contrastive learning (SimCLR, Chen et al., 2019) obtains features invariant to all irrelevant transformations of data.



(a) Global and local views. (b) Adjacent views.



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)
(f) Rotate {90°, 180°, 270°} (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering



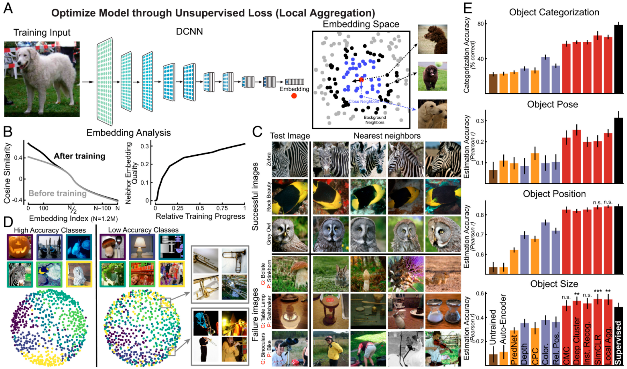
- Sample transformations $t, t' \sim \mathcal{T}$
- For each $x \in \mathcal{D}$, obtain two transformed images $x_i = t(x)$ and $x_j = t'(x)$
- then transform through a DNN to obtain representations $z_i = h(x_i)$ and $z_j = h(x_j)$
- For m data points, compute *similarity* $s_{ij} := \rho(z_i, z_j)$ from one image x, also similarities from different images s_{ik}
- Minimise the contrastive loss $\mathcal{L}_{\text{tr}}(x_i) := \frac{1}{2m}\sum_{i=1}^{m} \ell(x_i, x_j) + \ell(x_j, x_i)$ where

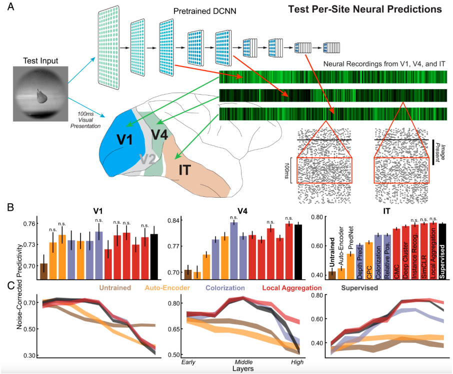
$$\ell(x_i, x_j) = -\log \frac{\exp\left(s_{ij}/\tau\right)}{\sum_{k \neq j} \exp\left(s_{ik}/\tau\right)}$$

- Test on other losses $\mathcal{L}_{\text{eval}}$, such as classification

Self-supervised learning: application to neuroscience

Self-supervised models can transfer to other tasks and predict neural activities (Zhuang et al., 2021)





Problems: self-supervised learning usually requires **HUGE** dataset and compute power.

Augment and train, not much thinking

Supervised learning can solve the following problems

	image cls.	speech recog.	translation	gait recog.	image seg.	scene parsing
$\mathcal{X}$	$\mathbb{R}_+^{w\times h\times c}$	$\mathbb{R}^t$	$\mathbb{1}_K^L$	$\mathbb{R}^{T\times n\times 3}$	$\mathbb{R}_+^{w\times h\times c}$	$\mathbb{R}_+^{w\times h\times c}$
$\mathcal{Y}$	Δ_K	Δ_V^τ	Δ_V^τ	Δ_K	$\Delta_K^{w\times h\times c}$	$\{\Delta_K, \mathbb{N}^4\}_{m=1}^M$

- Modify these to be self-supervised learning.
- Are there more principled methods to introduce augmentation?
- Can we enumerate all possible augmentations?

Deep reinforcement learning

Definition

A Markov decision process (MDP) is given by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, P_{\mathcal{X}}, P_{\mathcal{R}}, \gamma)$, consisting an environment with transition dynamics $P_{\mathcal{X}}(s'|s, a)$ and reward distribution $P_{\mathcal{R}}(r|s, a)$ for $s, s' \in \mathcal{S}$, $a \in \mathcal{A}$ and $r \in \mathcal{R}$, discounting factor $\gamma > 0$.

Broadly categorised into three approaches

- Valued-based
  - model-free/model-based
  - offline RL (similar to supervised learning)
  - distributional RL
- Actor-critic
- Policy-based
  - REINFORCE
  - Deterministic policy gradient

Valued-based RL

Goal: estimate the value function $Q^\pi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ **given a policy** π

For each transition $s' \sim P_{\mathcal{X}}(\cdot|s,a)$ and reward $r \sim P_{\mathcal{R}}(\cdot|s,a)$

- Simple Q-learning in a tabular environment:

$$Q^\pi(s,a) \leftarrow Q^\pi(s,a) + \alpha \left[r + \gamma \max_{a^*} Q^\pi(s',a^*) - Q^\pi(s,a) \right]$$

- Deep Q Network (DQN, Mnih et al., 2015) constructs a neural network $Q_\theta(s,a)$

$$\theta \overset{\text{sgd}}{\leftarrow} \frac{\partial}{\partial \theta} \left(r + \gamma \max_{a^*} Q_{\text{sg}(\theta)}(s',a^*) - Q_\theta(s,a) \right)^2$$

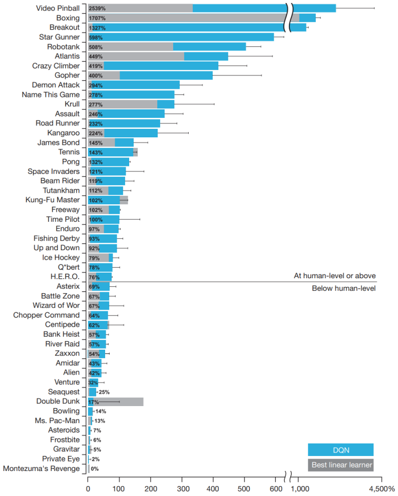
  where sg is stop-gradient operator (".detach()" in PyTorch).

The Q-values are used to derive a policy: ϵ-greedy, softmax, etc.

Important tricks to **make training data more i.i.d.**:

- **replay buffer**: the transitions are accumulated into a replay buffer (biologically inspired?)
- **offline RL**: maintain a behavioural network and a target network, occasionaly copy

Results on Atari

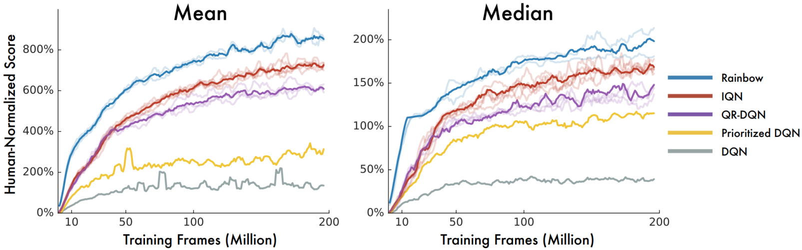


Distributional RL

Goal: estimate the return distribution $\eta^\pi : \mathcal{X} \times \mathcal{A} \to \mathcal{P}_\mathbb{R}$ given a policy
Instead of finding $Q(s,a) := \mathbb{E}_\pi[G(s,a)]$ for $G(s,a) := \sum_{t=1}^\infty \gamma^t R_t$, dist. RL estimates the distribution

$$\eta^\pi(s,a) := \text{distribution}(G(s,a))$$

- Distributional versions of Bellman update (Bellemare, Dabney & Rowland, 2023)
- Requires a form of distributional representation (e.g. histogram, quantiles)
- Biological evidence of dopamine neurons signaling (Dabney et al., 2020)



The field is exploding...

Classical learning paradigms are losing attention from research as industries begin to prevail.
Forefront of machine learning is addressing more challenging and diverse set of learning problems.

- Theory
- Meta-learning
- Approximating complex physical systems (differential equations)
- Learning from human feedback

The following slides are just a brief taste of how much is going on...

Categories ▾ English ▾

Google Scholar

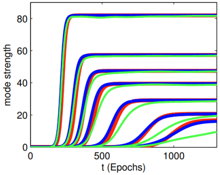
Top publications

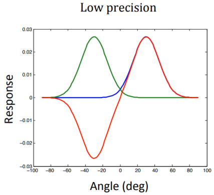
	Publication	h5-index	h5-median
1.	Nature	444	667
2.	The New England Journal of Medicine	432	780
3.	Science	401	614
4.	IEEE/CVF Conference on Computer Vision and Pattern Recognition	389	627
5.	The Lancet	354	635
6.	Advanced Materials	312	418
7.	Nature Communications	307	428
8.	Cell	300	505
9.	International Conference on Learning Representations	286	533
10.	Neural Information Processing Systems	278	436
11.	JAMA	267	425
12.	Chemical Reviews	265	444
13.	Proceedings of the National Academy of Sciences	256	364
14.	Angewandte Chemie	245	332
15.	Chemical Society Reviews	244	386
16.	Journal of the American Chemical Society	242	344
17.	IEEE/CVF International Conference on Computer Vision	239	415
18.	Nucleic Acids Research	238	550
19.	International Conference on Machine Learning	237	421
20.	Nature Medicine	235	389

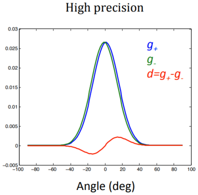
Theory: linear deep networks

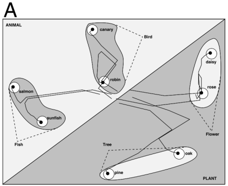
Linear deep networks $y = W_L W_{L-1} \cdots W_1 x$

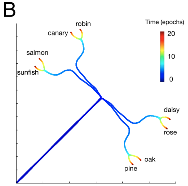
- no more representation power than a single layer $y = \left[\prod_{l=1}^{L} W_l\right]$
- show nonlinear dynamics
    - related to cognitive development of percpetual and semantic learning







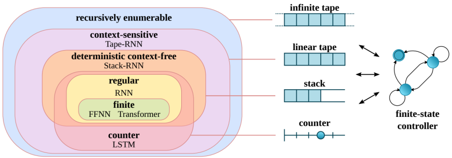




Theory: neural networks and the Chompsky hierarchy

Task: compare performance of different neural architectures on tasks of the Chompsky hierarchy (Delétang et al., 2022)

$$\min_{f \in \text{RNN class}} \mathcal{L}_{\text{tr}}(f; x_{1:100}, y_{1:100}) \qquad \text{test on } \mathcal{L}_{\text{tr}}(f; x_{1:500}, y_{1:500})$$



Level	Task	RNN	Stack-RNN	Tape-RNN	Transformer	LSTM
R	Even Pairs	**100.0**	**100.0**	**100.0**	**96.4**	**100.0**
	Modular Arithmetic (Simple)	**100.0**	**100.0**	**100.0**	24.2	**100.0**
	Parity Check†	**100.0**	**100.0**	**100.0**	52.0	**100.0**
	Cycle Navigation†	**100.0**	**100.0**	**100.0**	61.9	**100.0**
DCF	Stack Manipulation	56.0	**100.0**	**100.0**	57.5	59.1
	Reverse String	62.0	**100.0**	**100.0**	62.3	60.9
	Modular Arithmetic	41.3	**96.1**	**95.4**	32.5	59.2
	Solve Equation°	51.0	56.2	64.4	25.7	67.8
CS	Duplicate String	50.3	52.8	**100.0**	52.8	57.6
	Missing Duplicate	52.3	55.2	**100.0**	56.4	54.3
	Odds First	51.0	51.9	**100.0**	52.8	55.6
	Binary Addition	50.3	52.7	**100.0**	54.3	55.5
	Binary Multiplication×	50.0	52.7	58.5	52.2	53.1
	Compute Sqrt	54.3	56.5	57.8	52.4	57.5
	Bucket Sort†★	27.9	78.1	70.7	**91.9**	**99.3**

Meta-learning

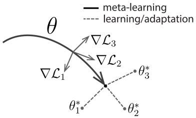
Goal: learning to learn, finding an learning algorithm from data

From a sequence of tasks/datasets $\mathcal{D}_{\text{tr}}^{(1)}, \cdots \mathcal{D}_{\text{tr}}^{(n)} \sim \mathcal{S}$

$$\min \mathcal{L}_{\text{tr}}(f, \mathcal{D}_{\text{tr}}^{(1)}, \cdots \mathcal{D}_{\text{tr}}^{(n-1)}) \quad \text{so that} \quad \mathcal{L}_{\text{eval}}(f, \mathcal{D}_{\text{tr}}^{(n)}) \text{ is small.}$$

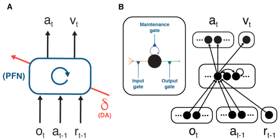
Weight-based: find f that can adapt

Finn et al., 2017



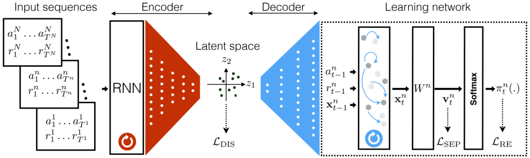
Memory/Activity-based: activity encodes task

Wang et al., 2018



Low-rank weights + memory

Dezfouli et al. 2019



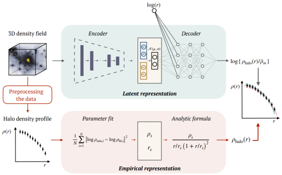
Learning complex dynamical systems

Traditional approach: simulate large-scale differential equations

The deep approach: throw in data (+tricks, inductive biases, etc.) and just train...

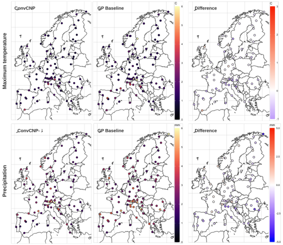
Predicting dark matter halo density

Lucie-Smith et al., 2022



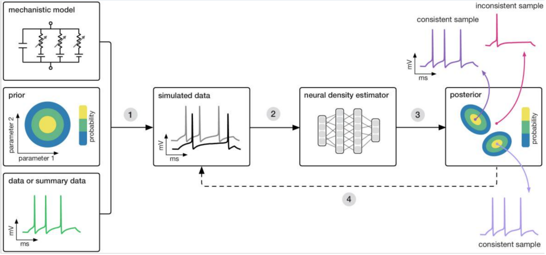
Weather forecasting

Vaughan et al., 2021



Estimating Hodgkin-Huxley model parameters

Gonçalves et al., 2020

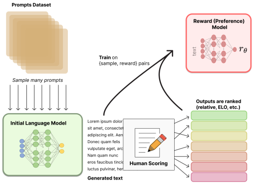


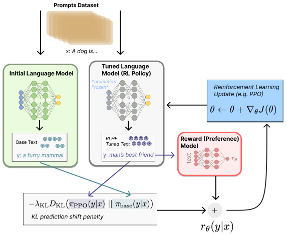
Learning from human preferences

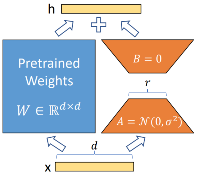
Large language models (LLMs) require a large amount of expert inputs

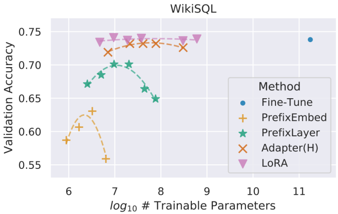
Different ways of improving a trained LLM

- prompt engineering / in-context learning
- self-improvement with external tools
- **weight finetuning**









Concluding remarks

$$\min_{f \in \mathcal{M}} \mathcal{L}_{\text{tr}}(f, \mathcal{D}_{\text{tr}}) \quad \text{so that} \quad \mathcal{L}_{\text{eval}}(f, \mathcal{D}_{\text{eval}}) \text{ is small,} \quad \text{where } \mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{eval}} \sim \mathcal{S}$$

Deep learning is the main workhorse for tech industry and aid for scientifc advances.

- Traditional boundaries between forms of learning are getting blurred
- Being smart is sometimes less important having interesting ideas (designing $\mathcal{L}_{\text{tr}}$ and $\mathcal{S}$)
  - Transforming learning problems into data engineering
  - Thinking about natural cognitive abilities is helpful for generating ideas
  - Unclear how implementation level knowledge directly and exclusively drive deep learning
  - More tricks to be discovered
  - Theory of learning is important but have not generated big leaps
  - Imagination is the only limit
- **If you want to do research, you must have a deep learning plan.**