Brain inspired Deep Learning Architectures

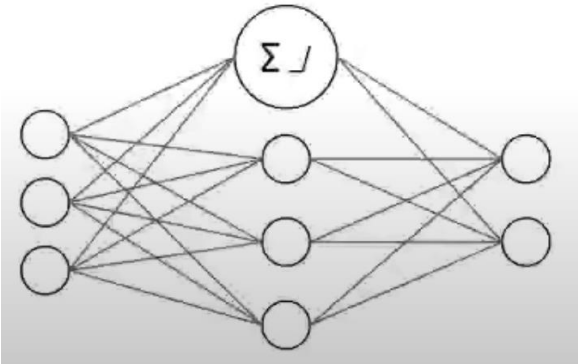
Alex Movila

FORTECH.

[IASI AI] Conventional artificial neural networks

- Inspired by the biological brain
- Benchmarked on tasks solved by the biological brain
- ...but compute in a fundamentally different way compared to the biological brain:

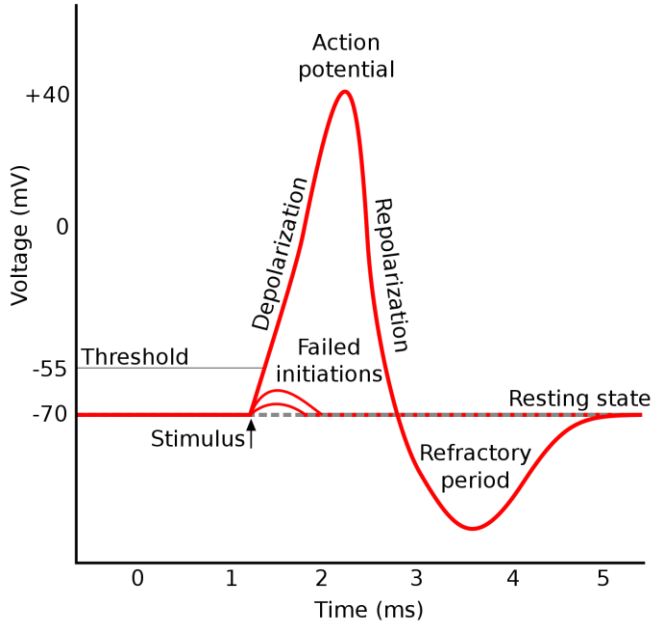
  - Synchronous processing
  - No true (continuous) **temporal dimension**

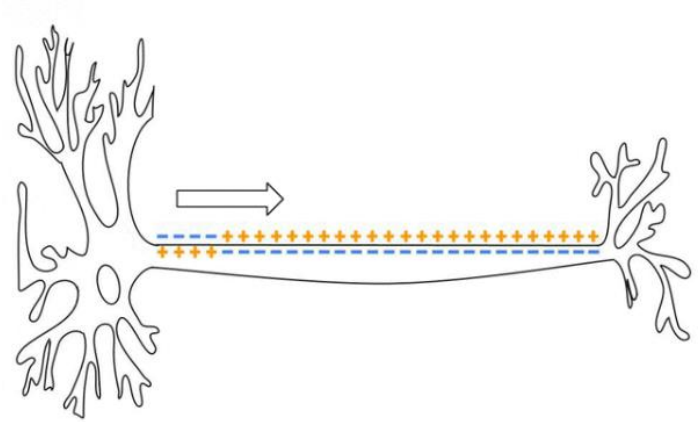


Iulia M. Comsa (Google Research) Talk

[IASI AI] Spiking Neural Networks

- Neurons communicate through action potentials (all-or-none principle)
- Asynchronous
- Can encode information in temporal patterns of activity
- Stateful (e.g. “predictive coding”)
- Energy-efficient





"All-or-none" principle = larger currents do not create larger action potentials

Wiki - Action potential , Action Potential in the Neuron

[IASI AI] Information coding in biological neurons

Rate coding

- cells with preferred stimulus features
- neurons fire with some probability proportional to the
- strength of the stimulus
- slow but reliable accumulation over spikes

Temporal coding

- information is encoded in the relative timing of spikes
- relative to other individual neurons or brain rhythms
- high temporal precision of spikes
- very fast information processing

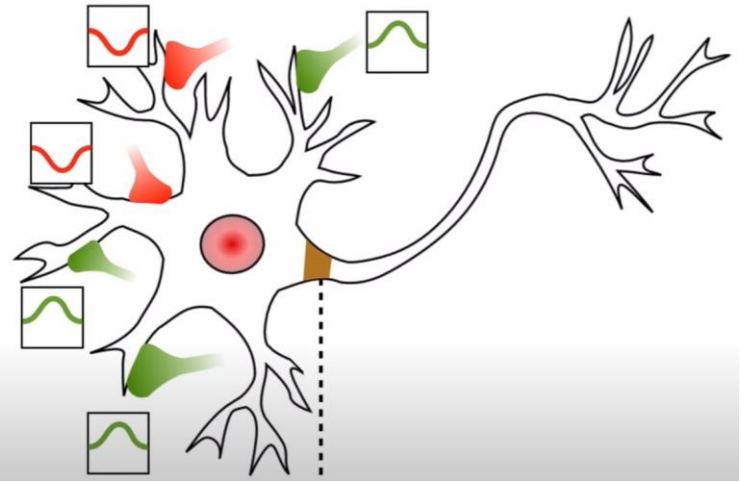
Information is carried by relative spike times (at least in visual part of the brain)

- retinal spikes are highly reproducible and convey more information through their timing than through their spike count (Berry et. al, 1997)
- retinal ganglion cells encode the spatial structure of an image in the relative timing of their first spikes (Gollish SrMeister, 2008)
- tactile afferents encode information about fingertip force and shape of the Surface in the relative timing of the first spikes (Johansson & Birznieks, 2004)

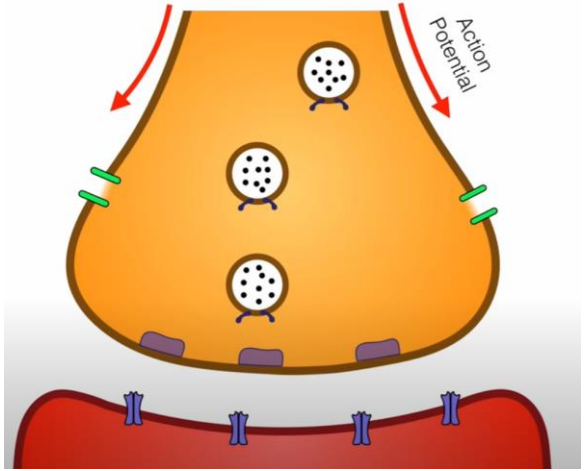
Iulia M. Comsa (Google Research) Talk, Blog, Code. Is coding a relevant metaphor for the brain?

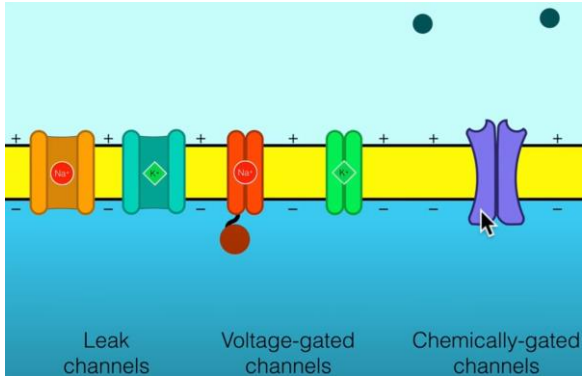
[IASI AI] Hebbian Learning - Neurons That Fire Together Wire Together

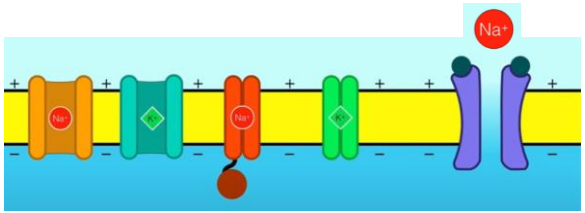
Hebb's Postulate:
"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."
(Donald Hebb, 1949)



Analog Digital

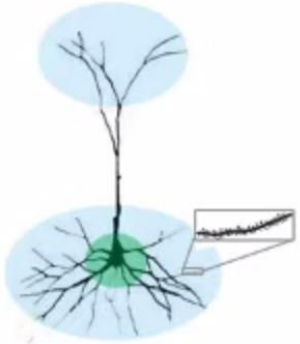






The Synapse

[IASI AI] DENDRITES DETECT SPARSE PATTERNS

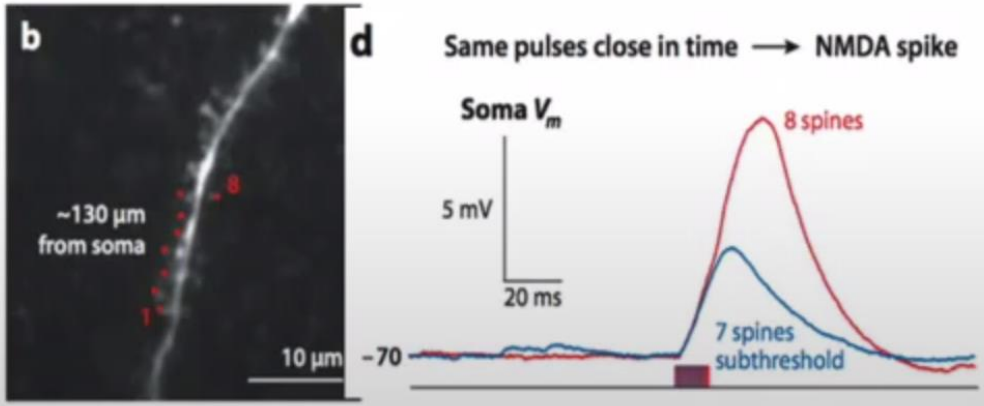


Pyramidal neuron
3K to 10K synapses

Dendrites split into dozens of independent computational segments

These segments activate with cluster of 10-20 active synapses

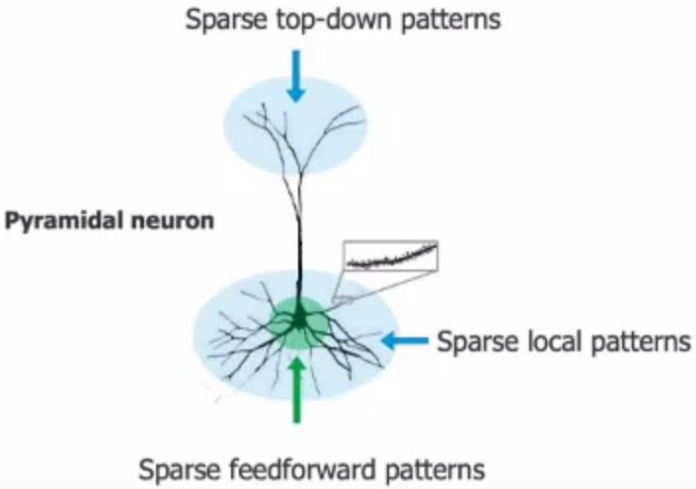
Neurons detect dozens of highly sparse patterns, in parallel



Major, Larkum and Schiller 2013

(Mel, 1992; Branco & Häusser, 2011; Schiller et al, 2000; Losonczy, 2006; Antic et al, 2010; Major et al, 2013; Spruston, 2008; Milojkovic et al, 2005, etc.)

[IASI AI] NEURONS UNDERGO SPARSE LEARNING IN DENDRITES



Learning localized to dendritic segments
"Branch specific plasticity"

If cell becomes active:

- If there was a dendritic spike, reinforce that segment
- If there were no dendritic spikes, grow connections by subsampling cells active in the past

If cell is not active:

- If there was a dendritic spike, weaken the segments

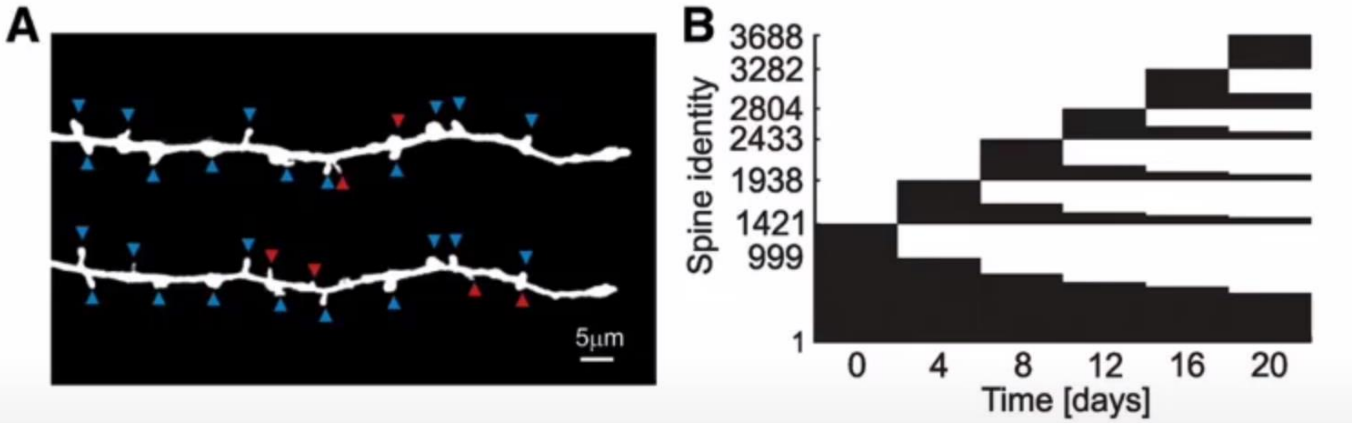
(Gordon et al., 2006; Losonczy et al., 2008; Yang et al., 2014; Cichon & Gang, 2015; El-Boustani et al., 2018; Weber et al., 2016; Sander et al., 2016; Holthoff et al., 2004)

[IASI AI]

HIGHLY DYNAMIC LEARNING AND CONNECTIVITY

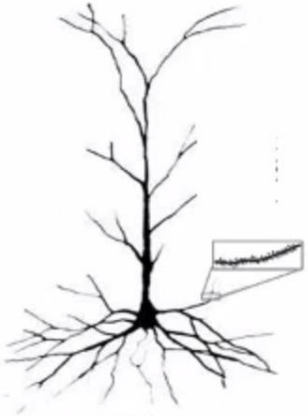
Learning involves growing and removing synapses

- Structural plasticity: network structure is dynamically altered during learning



"We observed substantial spine turnover, indicating that the architecture of the neuronal circuits in the auditory cortex is dynamic (Fig. 1B). Indeed, 31% ± 1% (SEM) of the spines in a given imaging session were not detected in the previous imaging session; and, similarly, 31 ± 1% (SEM) of the spines identified in an imaging session were no longer found in the next imaging session. (Loewenstein, et al., 2015)

[IASI AI] STABILITY OF SPARSE REPRESENTATIONS

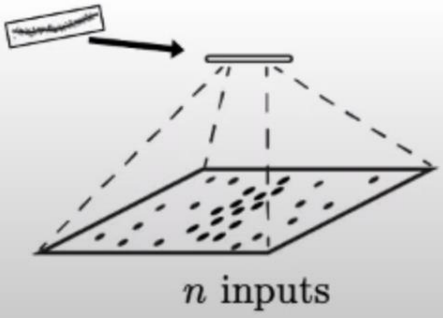


Pyramidal neuron
3K to 10K synapses

Thousands of neurons send input to any single neuron

On each neuron, 8-20 synapses on tiny segments of dendrites recognize patterns.

The connections are learned.



Sparse vector matching

$\boldsymbol{x}_i$ = connections on dendrite

$\boldsymbol{x}_j$ = input activity

$$P(\boldsymbol{x}_i \cdot \boldsymbol{x}_j \geq \theta)$$

[CLVision @ CVPR2020] Invited Talk: "Sparsity in the Neocortex and Implications..."

[IASI AI] STABILITY VS PLASTICITY

1. Sparsity in the neocortex

- Neural activations and connectivity are highly sparse
- Neurons detect dozens of independent sparse patterns
- Learning is sparse and incredibly dynamic

2. Sparse representations and catastrophic forgetting

- Sparse high dimensional representations are remarkably stable
- Local plasticity rules enable learning new patterns without interference

[CLVision @ CVPR2020] Invited Talk: "Sparsity in the Neocortex and Implications..."

[IASI AI] The Computational Power of Dendrites

- individual dendritic compartments could also perform a particular computation **“exclusive OR”** that mathematical theorists had previously categorized as **unsolvable by single-neuron**

- dendrites generated **local spikes**, had their **own nonlinear input-output curves** and had their **own activation thresholds**, distinct from those of the neuron as a whole

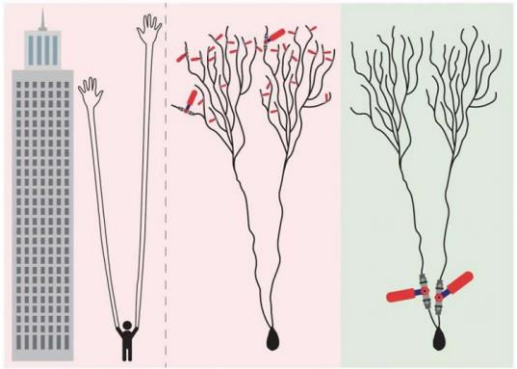
=>

- Much of the power of the processing that takes place in the cortex is actually subthreshold

- A single-neuron system can be more than just one integrative system. It can be **two layers, or even more**.

The newly discovered process of learning in the dendrites occurs at a much faster rate than in the old scenario suggesting that learning occurs solely in the synapses

Researchers suggest learning occurs in dendrites that are in closer proximity to neurons, as opposed to occurring solely in synapses.

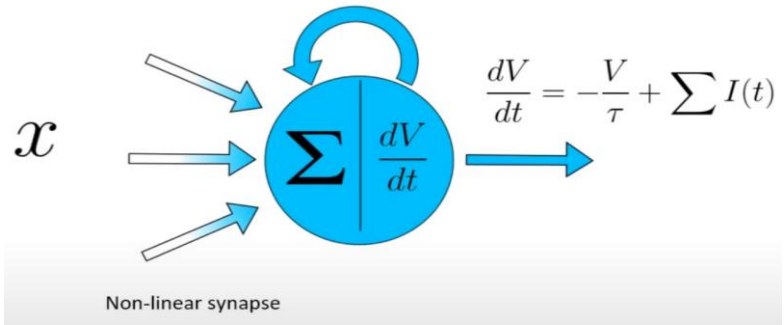


Hidden Computational Power Found in the Arms of Neurons,
The Brain Learns Completely Differently than We’ve Assumed Since the 20th Century

[IASI AI] Why study recurrent networks of spiking neurons?

Brains employ recurrent spiking neural networks (RSNNs) for computation
Why did nature go for recurrent networks?
Here are some obvious advantages:

- Selective integration of evidence over time / temporal processing
- capabilities
- Iterative inference (refining initial beliefs)
- Arbitrary depth with limited resource

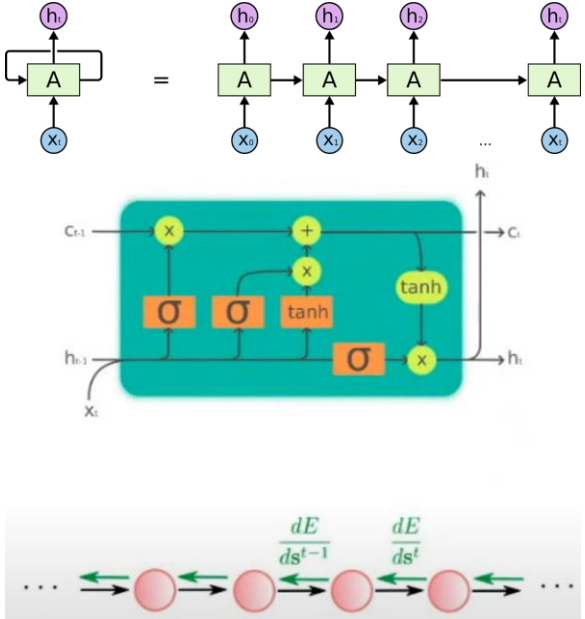


Brain-inspired Continuous-time Neural Networks

Going in circles is the way forward: the role of recurrence in visual inference, E-Prop Talk

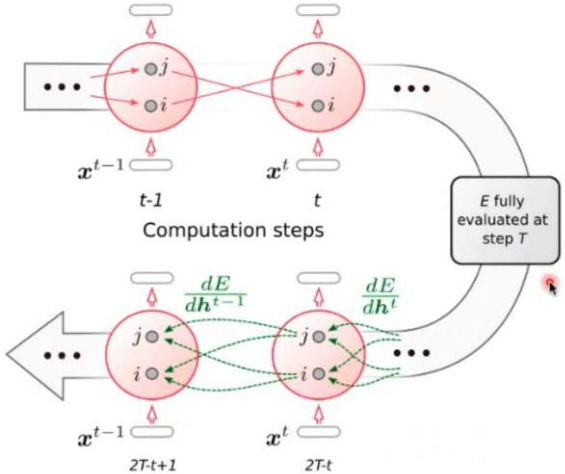
[IASI AI] Backpropagation Through Time (BPTT)

BPTT a success in ML but highly implausible in the brain



Long Short-Term Memory (LSTM) networks for computing (Hochreiter and Schmidhuber, 1997)

- Trained using Backpropagation Through Time (BPTT) for learning



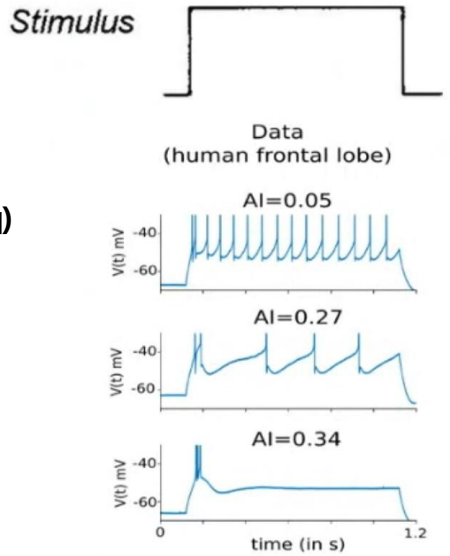
- BPTT unrolls T time steps of the computation of an RNN into a virtual „unrolled" feedforward network of depth T.
- Each time timestep corresponds to a copy of the RNN.
- Neurons (from the copy that represents t) send their output to neurons in the copy of the RNN corresponding to the next timestep (t+1)
- For an RSNN the resulting depth T is typically very large, e.g. T = 2000 for 1 ms time steps, and 2 s computing time.

[IASI AI] LSNN = RSNN + neuronal adaptation = LSTM performance (+ E-Prop)

Experimental data provides evidence of adaptive responses in pyramidal cells in both human and mouse neocortex (Allen Institute, 2018)

- **Spike frequency adaptation (SFA)**
These slower internal processes provide further memory to RSNNs and helps gradient-based learning (Bellec et al., 2018). It was demonstrated that LSNNs are on par with LSTMs on tasks with difficult temporal credit assignment

E-Prop alg – makes possible neuromorphic chips for training (online alg, no separate memory req)
Neurons and synapses maintain traces of recent activity, which are known to induce synaptic plasticity if closely followed by a top-down learning signal. These traces are commonly called eligibility traces.

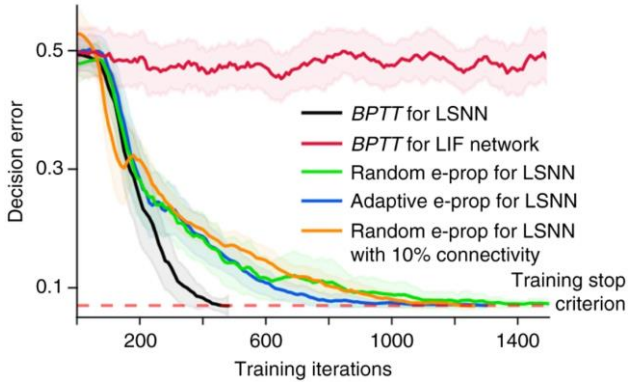


Spike frequency adaptation

$$W_{ji} \leftarrow W_{ji} - \eta\, L_j^t\, e_{ji}^t$$

Top-down learning signal
(specific for the postsynaptic neuron)

Eligibility trace (specific for synapse and independent of the loss function *E*)

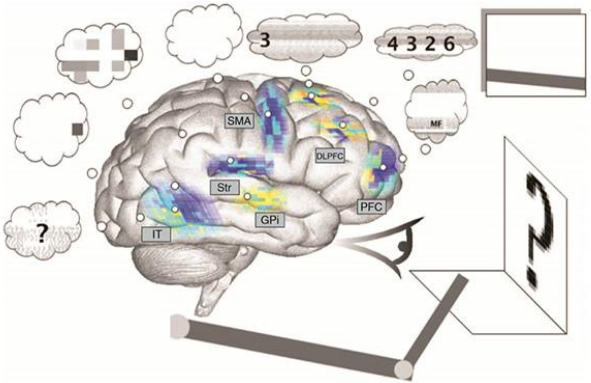


E-Prop Talk, Paper, OpenReview
New learning algorithm should significantly expand the possible applications of AI
Long short-term memory and learning-to-learn in networks of spiking neurons

[IASI AI] Spiking Neural Networks for More Efficient AI Algorithms

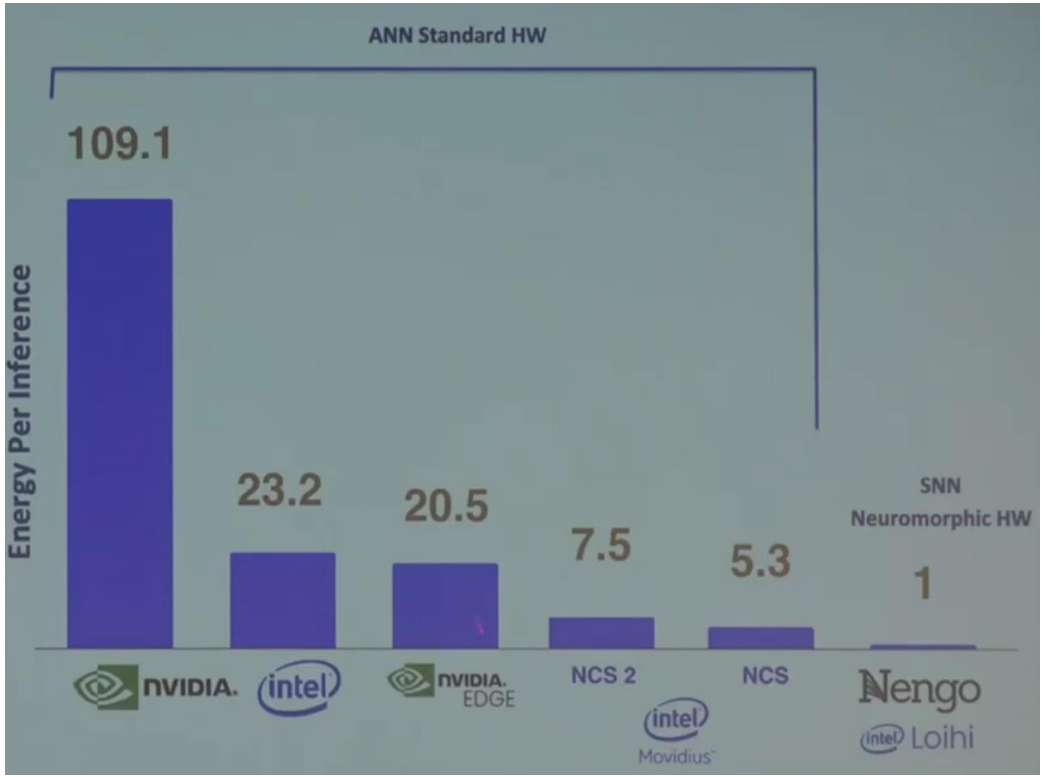
ANN Accuracy = 92.7%
SNN Accuracy = 93.8%

Nengo: Large-scale brain modelling in Python



World's largest brain model

- 6.6 million neurons
- 20 billion connections
- 12 tasks

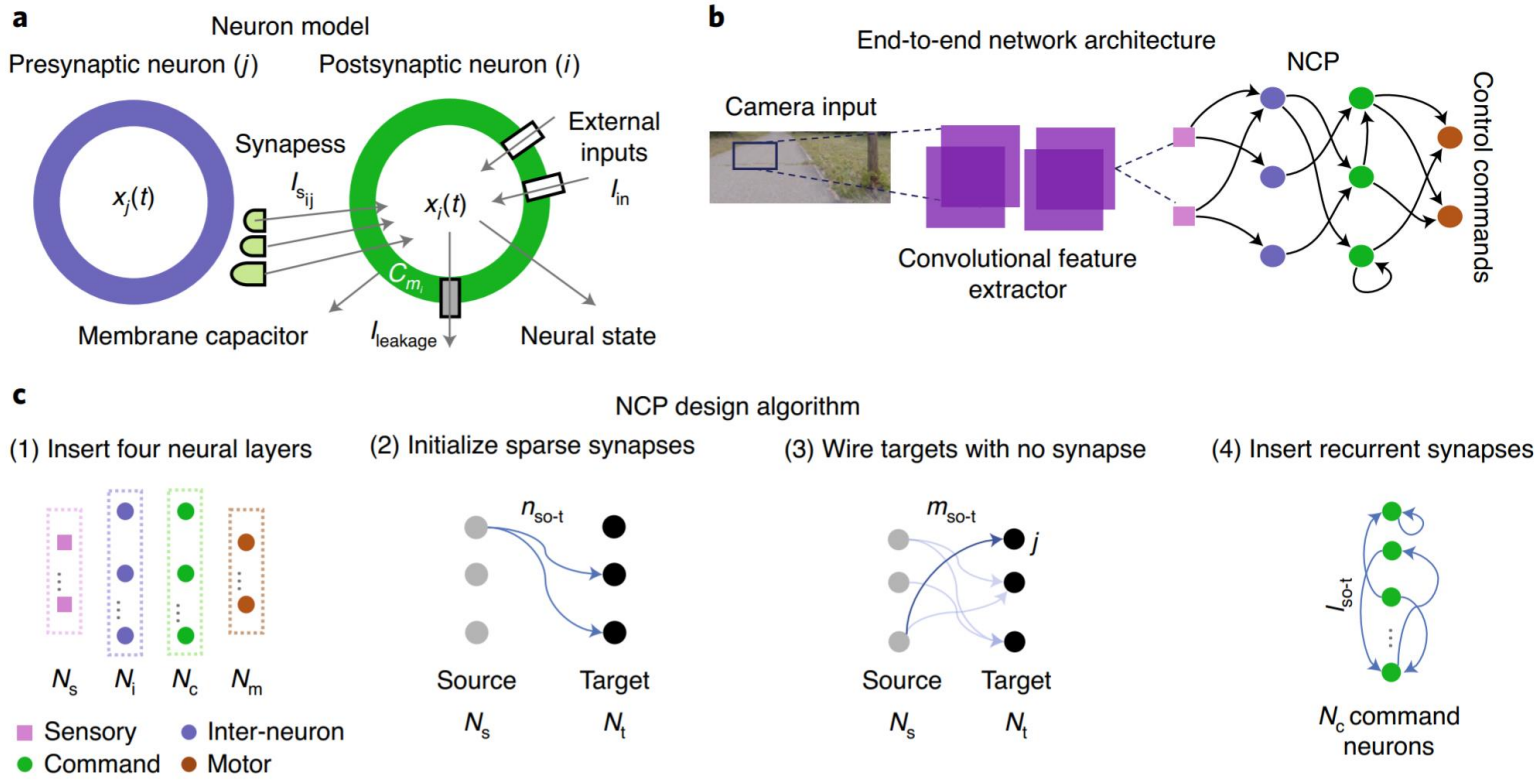


Spaun, the most realistic artificial human brain yet
Nengo PPT, Coming from TensorFlow to NengoDL

Spiking Neural Networks for More Efficient AI Algorithms,

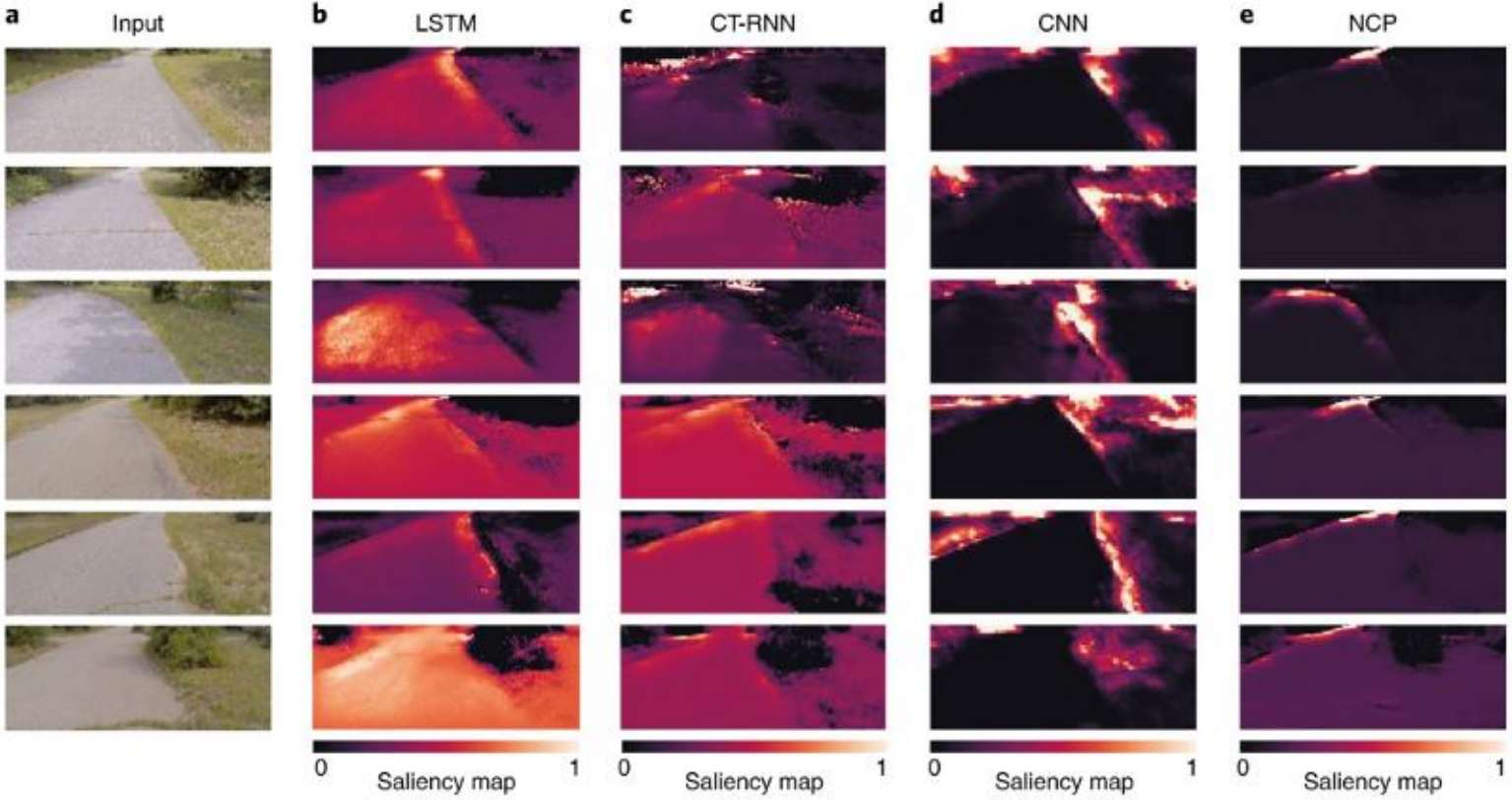
[IASI AI] Self-Driving car with 19 worm brain-inspired neurons (Neural circuit policies)

We discover that a single algorithm with **19 control neurons**, connecting 32 encapsulated input features to outputs by 253 synapses, learns to map high-dimensional inputs into steering commands. This system shows **superior generalizability, interpretability and robustness** compared with orders-of-magnitude larger black-box learning systems.



A New Brain-inspired Intelligent System Drives a Car Using Only 19 Control Neurons! (Daniela Rus, Radu Grosu), TEDxCluj, Demo

[IASI AI] Neural circuit policies – Results: Robust to Noise, Fast and Very Sparse



Saliency maps show where each network is learned to attend while driving

A New Brain-inspired Intelligent System Drives a Car Using Only 19 Control Neurons! (Daniela Rus, Radu Grosu), TEDxCluj, Demo

[IASI AI] From ResNets to Neural ODEs

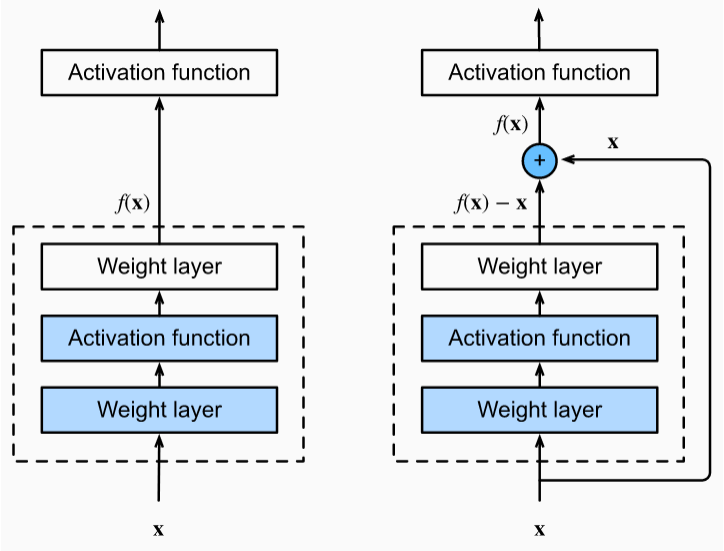
ResNet:

We can derive a continous version:

x^t+1 = x^t + F(x^t, W^t) =>
x^t+1 - x^t = F(x^t, W^t) =>

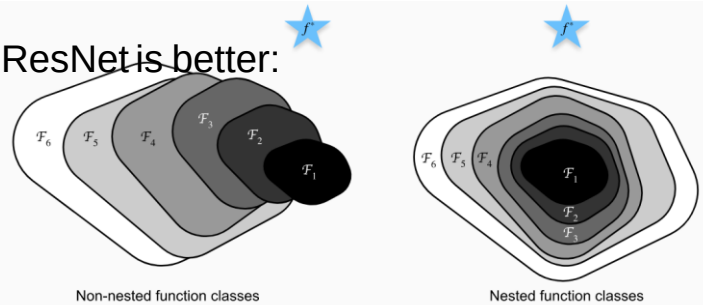
Deriv x(t) / dt = F(x(t), W(t))

= 1 continuous time layer with weights evolving in time (= infinite number of discrete layers)



A regular block (left) and a residual block (right).

Why ResNet is better:



For non-nested function classes, a larger (indicated by area) function class does not guarantee to get closer to the "truth" function (f^*). This does not happen in nested function classes.

ResNets, dl.ai Course, New deep learning models require fewer neurons

From ResNets to Neural ODEs

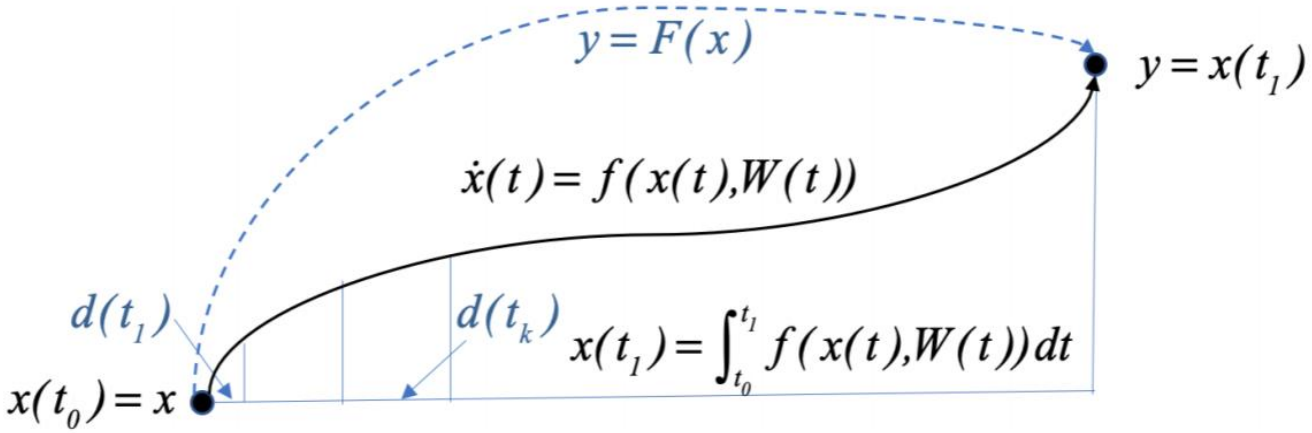
Neural ODE (CT version of ResNet):

$$d\mathbf{x}(t)/dt = f(\mathbf{x}(t), \mathbf{I}(t), t, \theta)$$

$\mathbf{x}(t) \in \mathbb{R}^D$ = hidden state at time t

$\mathbf{I}(t) \in \mathbb{R}^M$ = inputs

How to train? A: gradient descent through numerical ODE solver:



ResNets, NeuralODEs and CT-RNNs are Particular Neural Regulatory Networks, The Overlooked Side of Neural ODEs

[IASI AI] Liquid Time-constant NNs – more expressive than Neural ODE or CT-LSTM

CT-RNN:
- more stable, can reach equilibrium – implement a leaky term with a time constant

$$\frac{d\mathbf{x}(t)}{dt} = -\frac{\mathbf{x}(t)}{\tau} + f(\mathbf{x}(t), \mathbf{I}(t), t, \theta)$$

LTC (inspired from non-spiking neuron):

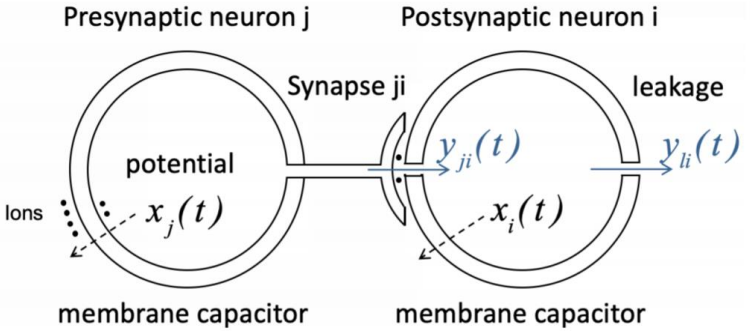
$$d\mathbf{x}(t)/dt = -\mathbf{x}(t)/\tau + f(\mathbf{x}(t), \mathbf{I}(t), t, \theta)(A - \mathbf{x}(t))$$

Let's rewrite:

$$\frac{d\mathbf{x}(t)}{dt} = -\left[\frac{1}{\tau} + f(\mathbf{x}(t), \mathbf{I}(t), t, \theta)\right]\mathbf{x}(t) + f(\mathbf{x}(t), \mathbf{I}(t), t, \theta)A.$$

More expressive – we have an input-dependent varying time-constant

$$\tau_{sys} = \frac{\tau}{1+\tau f(\mathbf{x}(t),\mathbf{I}(t),t,\theta)}$$



The electric representation of a nonspiking neuron.

$$\dot{x}_i(t) = w_{li}\,(E_{li} - x_i(t)) + \sum_{j=1}^{n} y_{ji}(t)$$

$$y_{ji}(t) = w_{ji}\,\sigma(x_j(t), \mu_j)\,(E_{ji} - x_i(t))$$

E_{ji} =synaptic potential

Paper , The Overlooked Side of Neural ODEs,

[IASI AI] Liquid Time-constant NNs – more expressive than Neural ODE or CT-LSTM

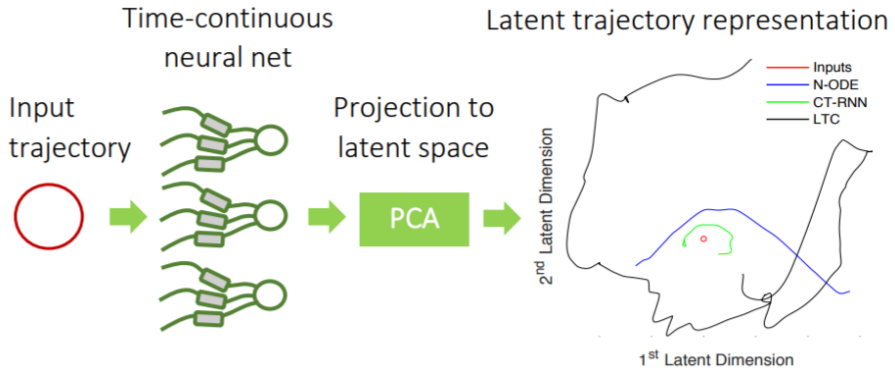


Figure 1: The complexity of the computed states of CT networks grows exponentially. The *trajectory length* [46] as a measure of expressivity of deep nets captures the complexity, and demonstrates flexible pattern explorations by LTCs.

Raghu et. al. ICML 2017 introduced novel measures of expressivity of deep neural networks unified by the notion of trajectory length.

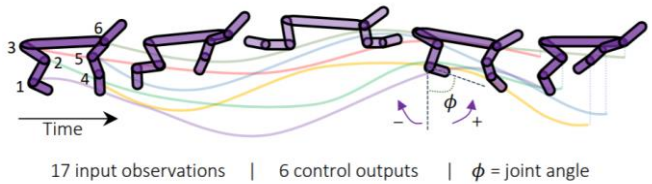


Figure 6: Half-cheetah physics simulation

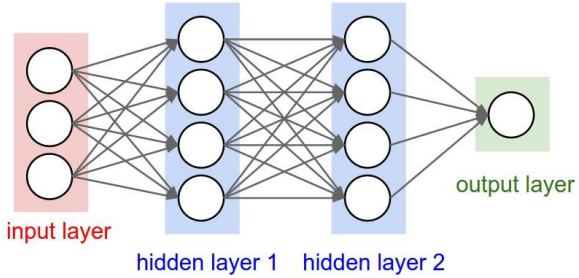
Table 6: Sequence modeling. Half-Cheetah dynamics n=5

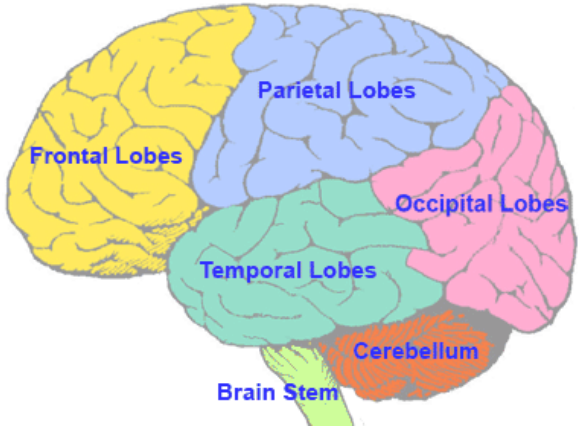
Algorithm	MSE
LSTM	2.500± 0.140
CT-RNN	2.838± 0.112
Neural ODE	3.805 ± 0.313
CT-GRU	3.014± 0.134
LTC (ours)	**2.308± 0.015**

[IASI AI] What can we borrow from the brain

Low Level:

- Local learning rules (Hebbian Learning)
- Global learning signals (Neurotransmitters like dopamine)
- Feedback connections (Top-down attention)
- Sparsity
- Plasticity (Dynamic connections & parameters & architecure)
- Modularity
- Specialization
- Recurrent connections (gives State)
- Lateral Connections
- Inhibitory / Excitatory Connections
- Time Continuous Processing (take temporal dimension into account)
- Asynchronous percessing
- Energy Efficient Neuromorphic hardware (allows embodied AI)





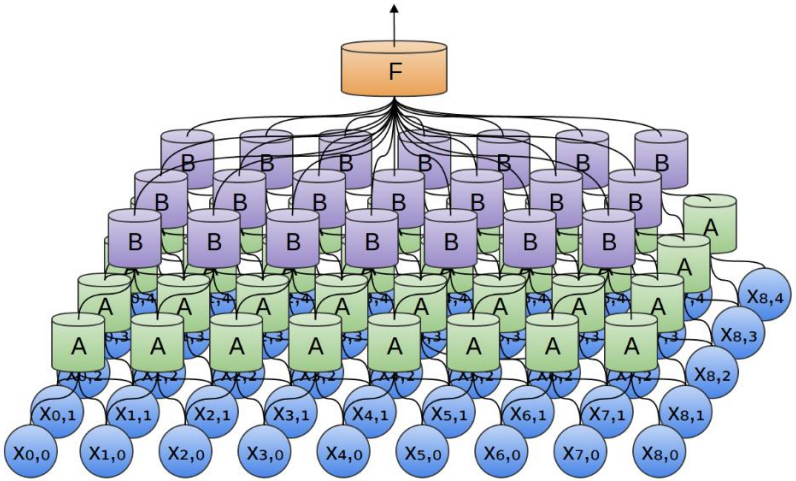
High Level:

- Reasoning
- Causality
- Continual / Multi-Modal / Bayesian/ Active/ Unsupervised/ Reinforcement / Supervised learning
- Sparse Learning / Experience Replay
- Conciousness

Recurrence in biological and artificial neural networks

[IASI AI] Convolutional NNs - Neuroscience Inspired

- Some individual neurons in the brain are activated or fired only in the presence of edges of a particular orientation like vertical or horizontal edges
- Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field.
- The receptive fields of different neurons partially overlap such that they cover the entire visual field. ()



A 2D Convolutional Neural Network

Hubel and Wiesel presented light stimuli to a cat while recording from neurons in the cat's visual cortex. The popping sounds you hear are the cells firing in response to the light. Increases in the number or the speed of the popping indicates that the cell strongly reacts to the current stimulus.

Visual Cell Recording

Conv Nets: A Modular Perspective

[IASI AI] Local Response Normalization in AlexNET = Lateral Inhibition

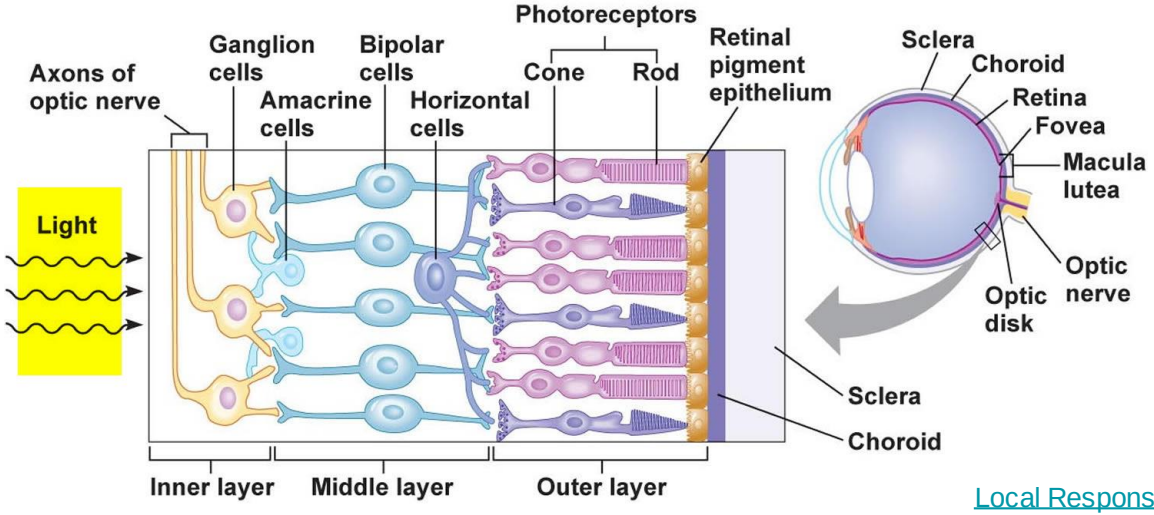
= a form of lateral inhibition inspired by the type found in real neurons, creating competition for big activities amongst neuron outputs computed using different kernels.

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

$a^i_{x,y}$ = the activity of a neuron computed by applying kernel i at position (x, y) and then applying the ReLU nonlinearity

where the sum runs over n "adjacent" kernel maps at the same spatial position, and N is the total number of kernels in the layer

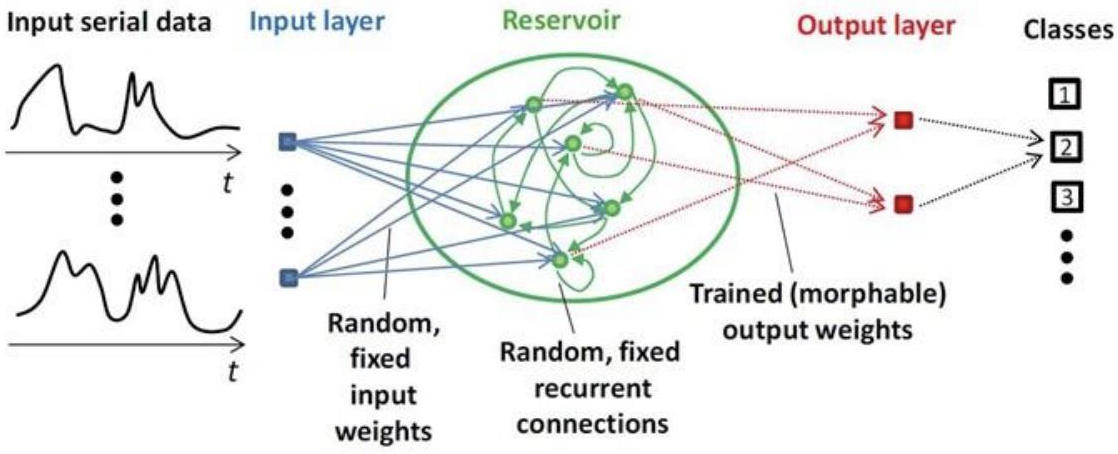
Results: CIFAR-10 dataset: a four-layer CNN achieved a 13% test error rate without normalization and 11% with normalization



Local Response Normalization for Deep Learning Explained , AlexNET paper

[IASI AI] Reservoir Computing – Echo State Networks – Very Fast Training

For certain species of animals – newborns can learn to walk in hours – how? Is it related to huge number of synapses?



To recap:
- The network nodes each have distinct dynamical behavior
- Time delays of signal may occur along the network links
- The network hidden part has recurrent connections
- The input and internal weights are fixed and randomly chosen
- Only the output weights are adjusted during the training.

"There (is) order and even great beauty in what looks like total chaos. If we look closely enough at the randomness around us, patterns will start to emerge." — Aaron Sorkin

Predicting Stock Prices with Echo State Networks

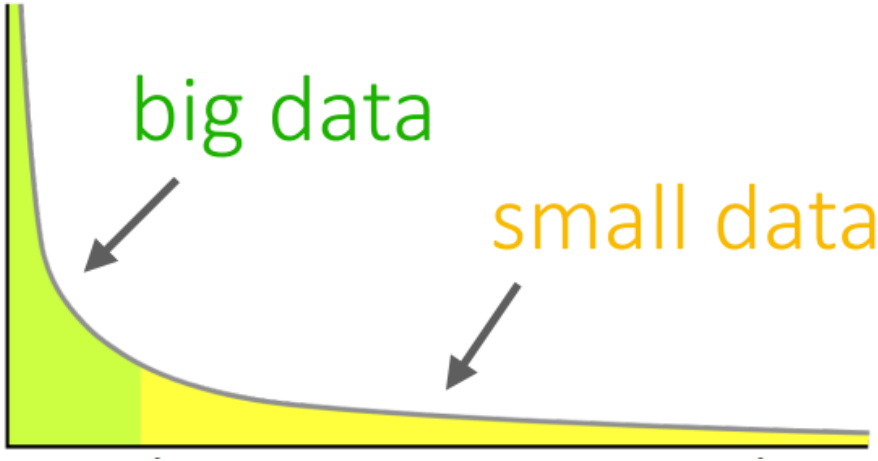
Reservoir computing is a framework for computation derived from recurrent neural network theory that maps input signals into higher dimensional computational spaces through the dynamics of a fixed, non-linear system called a reservoir.

[IASI AI] What if you want a general-purpose AI system in the real world?

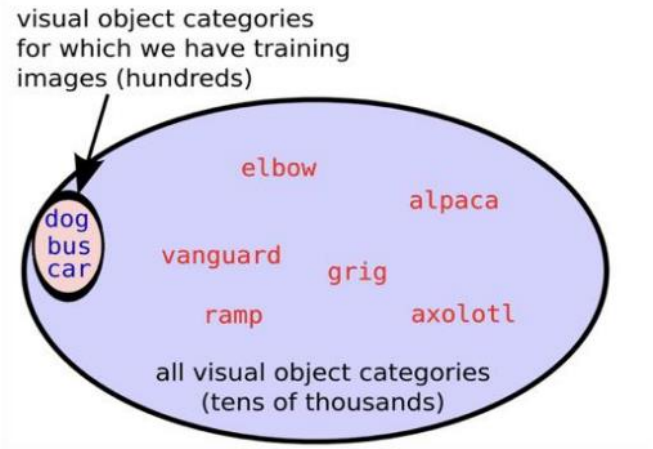
- Need to continuously adapt and learn on the job.
- Learning each thing from scratch won't cut it.

- What if your data has a long tail?

No of data points



Objects encountered
Interactions with people
Words heard
Driving scenarios



We never have enough training data to classify all the categories!

[IASI AI] Solution – Use prior related experience

Humans have prior experience.

A rough analogy can be made to evolution: a slow and expensive meta-learning process, which has resulted in life-forms that at birth already have priors that facilitate rapid learning and inductive leaps.

So we need more data-driven priors.

How to inject more data:

- transfer learning
- domain adaptation
- unsupervised learning
- learning to learn (a new task) (=meta-learning)

> The idea: In order to generalize to a new environment, you have to practice generalizing to a new environment. It's so simple when you think about it. Children do it all the time. When they move from one room to another room, the environment is not static, it keeps changing. Children train themselves to be good at adaptation.
>
> Yoshua Bengio, Revered Architect of AI, Has Some Ideas About What to Build Next

Meta-learning = similar with transfer learning + fine-tuning for a new task
The difference is that **meta-learning adaptation is done with very few data examples**

How to evaluate a meta-learning algorithm

5-way, 1-shot image classification (MiniImagenet)

Given 1 example of 5 classes:

Classify new examples

Meta-learning defined

$\mathcal{D}$

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$$

Can we incorporate additional data?
Yes. From prior similar tasks:

$\mathcal{D}_1$

$\mathcal{D}_2$

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$$

$$\mathcal{D}_i = \{(x_1^i, y_1^i), \ldots, (x_k^i, y_k^i)\}$$

Train together with D + D meta-train -> not convenient to keep meta-train data

Better use meta-train data to train and distill initial network parameters that make the network very adaptable:

$$meta\text{-}parameters\ \theta:$$

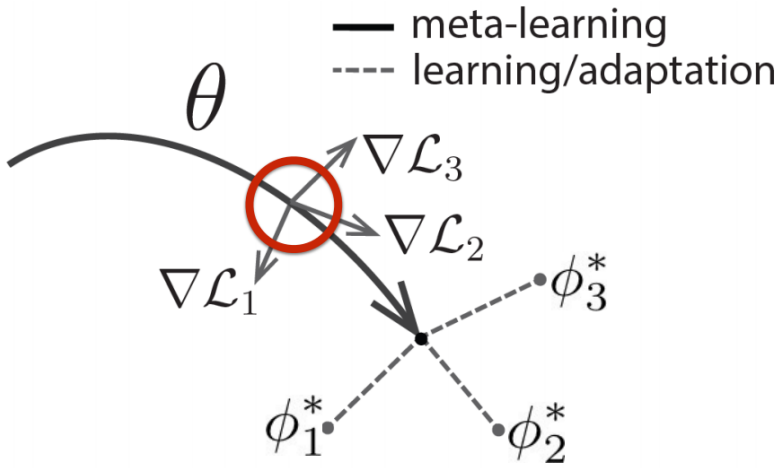
Model-Agnostic Meta-Learning (MAML)

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}(\theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$$

Bring us close to optimal params for each task:

θ parameter vector being meta-learned

ϕ_i^* optimal parameter vector for task i



Model-Agnostic Meta-Learning

Optimization-Based Inference via MAML

General algorithm for Model-Agnostic Meta-Learning

1. Sample task $\mathcal{T}_i$ *(or mini batch of tasks)*
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$
3. ~~Compute $\phi_i \leftarrow f_\theta(\mathcal{D}_i^{\text{tr}})$~~ Optimize $\phi_i \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})$
4. Update θ using $\nabla_\theta \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{test}})$

Unfortunately -> brings up second-order derivatives (more on this later – can be mitigated)

Idea: Combine methods - Learn initialization via MAML but replace gradient update to that initialization with learned network (black-box adaptation)

[IASI AI] Meta-Learning with Implicit Gradients (Implicit MAML)

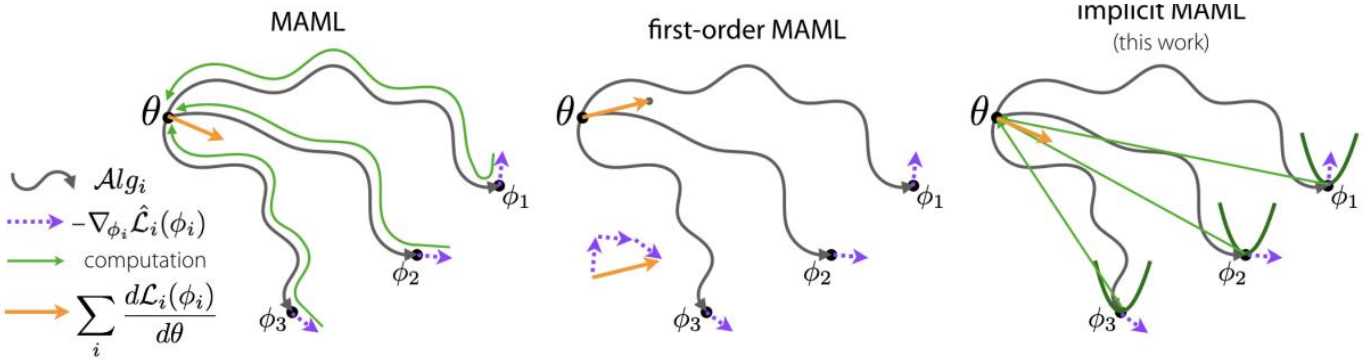


Figure 1: To compute the meta-gradient $\sum_i \frac{d\mathcal{L}_i(\phi_i)}{d\theta}$, the MAML algorithm differentiates through the optimization path, as shown in green, while first-order MAML computes the meta-gradient by approximating $\frac{d\phi_i}{d\theta}$ as $\boldsymbol{I}$. Our implicit MAML approach derives an analytic expression for the exact meta-gradient without differentiating through the optimization path by estimating local curvature.

Algorithm	Compute	Memory	Error
MAML (GD + full back-prop)	$\kappa \log\left(\frac{D}{\delta}\right)$	$\text{Mem}(\nabla\hat{\mathcal{L}}_i) \cdot \kappa \log\left(\frac{D}{\delta}\right)$	0
MAML (Nesterov's AGD + full back-prop)	$\sqrt{\kappa} \log\left(\frac{D}{\delta}\right)$	$\text{Mem}(\nabla\hat{\mathcal{L}}_i) \cdot \sqrt{\kappa} \log\left(\frac{D}{\delta}\right)$	0
Truncated back-prop [53] (GD)	$\kappa \log\left(\frac{D}{\delta}\right)$	$\text{Mem}(\nabla\hat{\mathcal{L}}_i) \cdot \kappa \log\left(\frac{1}{\epsilon}\right)$	ϵ
Implicit MAML (this work)	$\sqrt{\kappa} \log\left(\frac{D}{\delta}\right)$	$\text{Mem}(\nabla\hat{\mathcal{L}}_i)$	δ

Table 3: Mini-ImageNet 5-way-1-shot accuracy

Algorithm	5-way 1-shot
MAML	48.70 ± 1.84 %
first-order MAML	48.07 ± 1.75 %
Reptile	49.97 ± 0.32 %
iMAML GD (ours)	48.96 ± 1.84 %
iMAML HF (ours)	49.30 ± 1.88 %

Meta-Learning with Implicit Gradients , iMAML: Meta-Learning with Implicit Gradients (Paper Explained)

[IASI AI] Predictive Coding – can replace Backprop (+ parallel / – 100x slower)

Central to the theory is the idea that **the core function of the brain is to minimize prediction errors between what is expected to happen and what actually happens**. Predictive coding views the brain as composed of multiple hierarchical layers which predict the activities of the layers below. **Unpredicted activity is registered as prediction error which is then transmitted upwards for a higher layer to process.** Over time, synaptic connections are adjusted so that the system improves at minimizing prediction error.

2015:
The animal can refine its guess for the food size by combining the sensory stimulus with the prior knowledge on how large the food items usually are, that it had learnt from experience. = Bayesian variational inference

A tutorial on the free-energy framework for modelling perception and learning

2017:
For unsupervised learning the backpropagation algorithm can be closely approximated in a model that uses a simple local Hebbian plasticity rule.

An Approximation of the Error Backpropagation Algorithm in a Predictive Coding Network with Local Hebbian Synaptic Plasticity (Code)

2019:
An architecture of a predictive coding network contains error nodes that are each associated with corresponding value nodes.
…
However, the one-to-one connectivity of error nodes to their corresponding value nodes is inconsistent with diffused patterns of neuronal connectivity in the cortex.

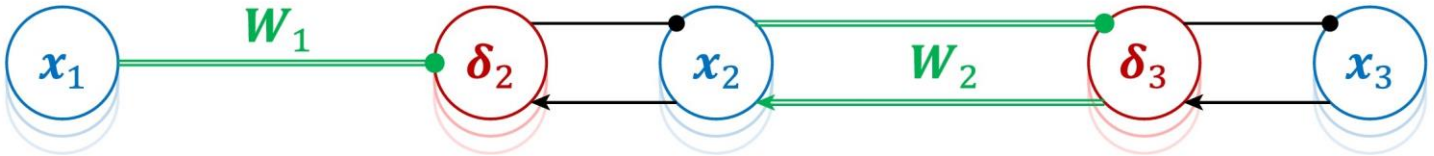
Theories of Error Back-Propagation in the Brain

2020:
Here we present a generalized form of predictive coding applied to arbitrary computation graphs.
….
This gives the predictive coding CNN approximately a **100x computational overhead** compared to backprop.

Predictive Coding Approximates Backprop along Arbitrary Computation Graphs (Code), Optical Illusions: When Your Brain Can't Believe Your Eyes

[IASI AI] Predictive Coding aproximates BackProp

(A)

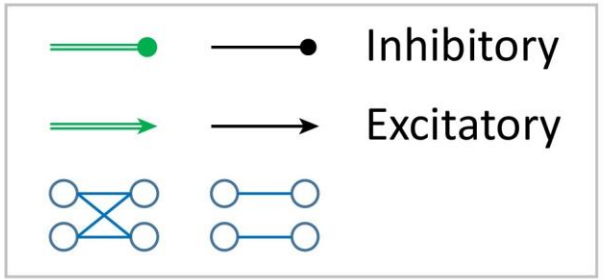


(B)

Dynamics:

$$\delta_l = x_l - W_{l-1} x_{l-1} \qquad (3.1)$$

$$\dot{x}_l = -\delta_l + W_l^T \delta_{l+1} \qquad (3.2)$$



(C)

During prediction $\delta_l \to 0$ so from 3.1 $x_l = W_{l-1} x_{l-1}$

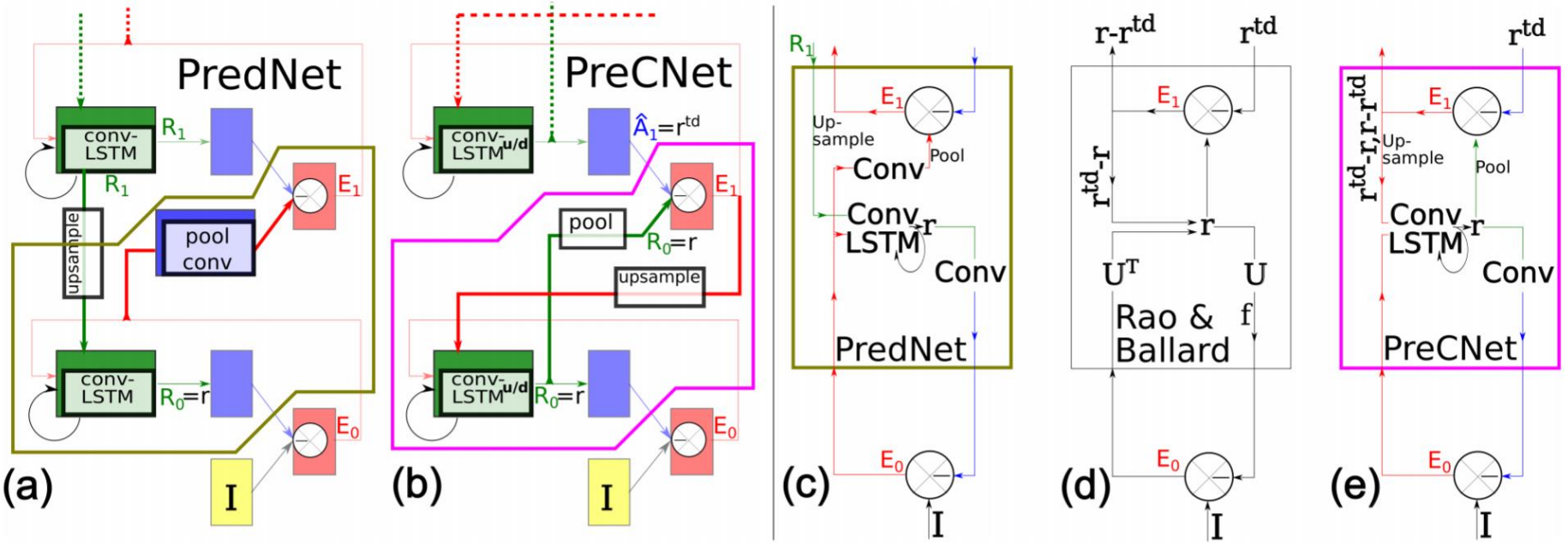
(D)

During learning $\dot{x}_l \to 0$ so from 3.2 $\delta_l = W_l^T \delta_{l+1}$

Trends in Cognitive Sciences

Theories of Error Back-Propagation in the Brain

[IASI AI] PreCNet: Next Frame Video Prediction Based on Predictive Coding



1) **Prediction phase**. The information flow goes iteratively from a higher to a lower module. At the end of this phase (at Module 0), the prediction of the incoming input image $\hat{A}_0$ is outputted.
2) **Correction phase**. In this phase, the information flow goes iteratively up. The error between the prediction and actual input is propagated upward.

Multiple frame video prediction evaluation schema. After inputting 10 frames, the predicted frames are inputted instead of the actual frames. The prediction errors are therefore zeros. The predicted frames are compared—using MSE, PSNR, SSIM—with the actual inputs.

PreCNet: Next Frame Video Prediction Based on Predictive Coding,
There is a new wave of deep neural networks coming

[IASI AI] Bayesian Inference

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Bayes's Theorem

Conjunction rule:

p(A and B) = p(A) p(B|A)

p(B and A) = p(B) p(A|B)

Equate the right hand sides

p(B) p(A|B) = p(A) p(B|A)

"Given the test result, what is the probability that I actually have this disease?"

$$P(\text{Disease}|+) = \frac{P(+|\text{Disease})P(\text{Disease})}{P(+)}$$

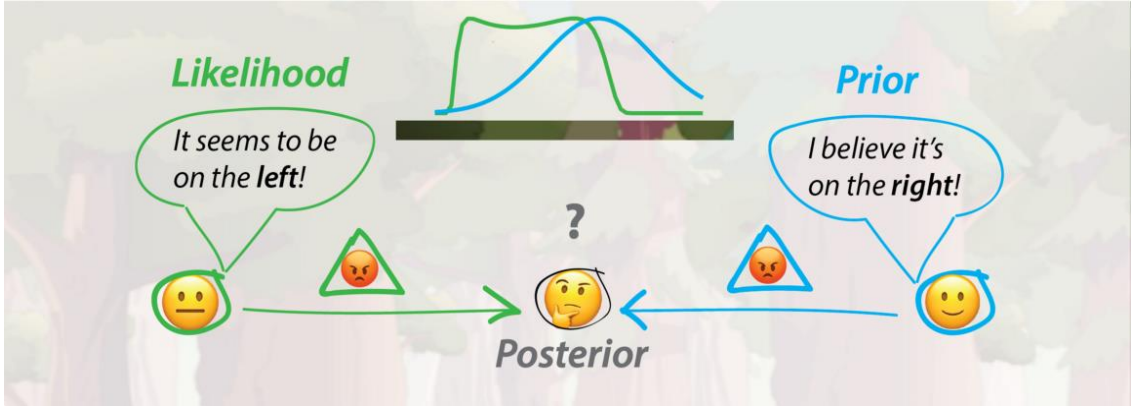
P(Disease) = *prior* probability of the disease P(Disease). Think of this as the incidence of the disease in the general population

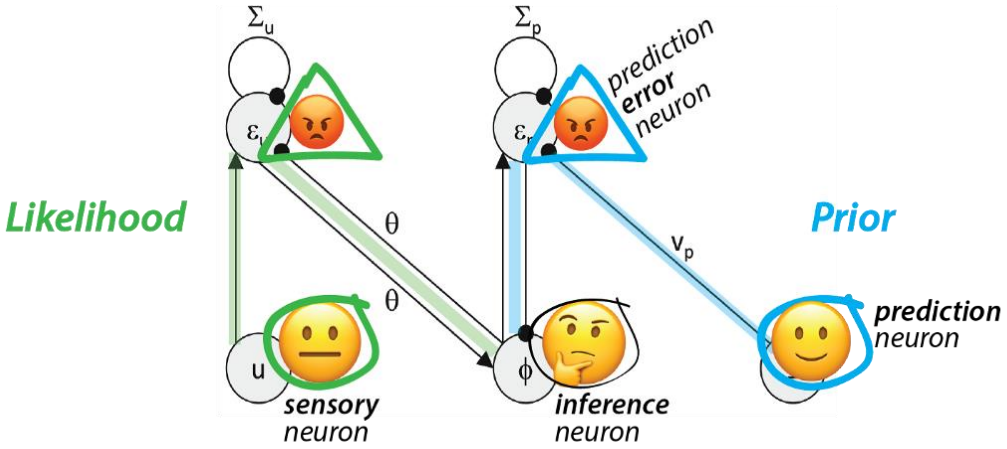
P(+|Disease) = test accuracy: How often does the test correctly report a negative result for a healthy patient, and how often does it report a positive result for someone with the disease?

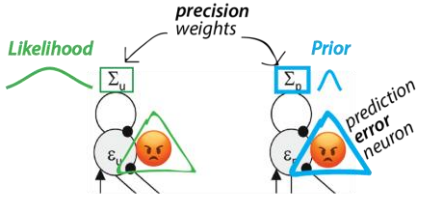
P(+) = the overall probability of a positive result

Bayesian Inference ,
Bayesian Statistics Made Simple | Scipy 2019 Tutorial

[IASI AI] Bayesian inference via Predictive Coding



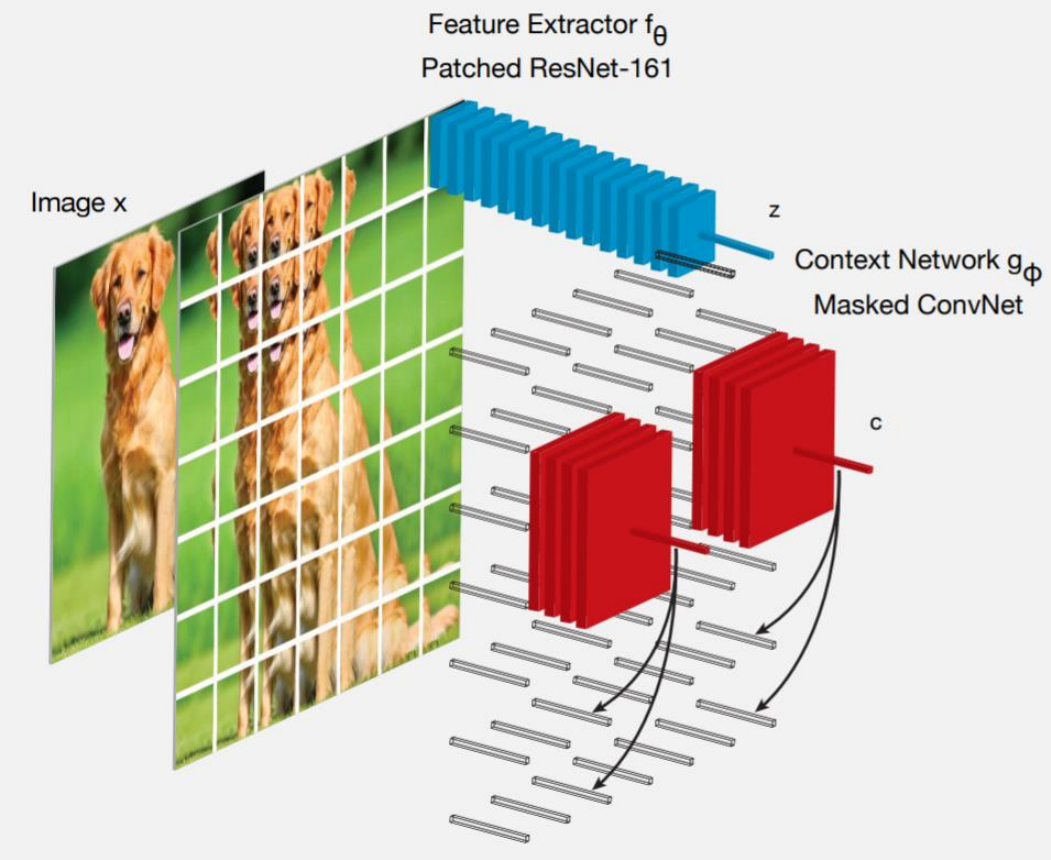




Intuitions on predictive coding and the free energy principle

A new kind of deep neural networks

[IASI AI] Data-Efficient Image Recognition with Contrastive Predictive Coding



Contrastive Predictive Coding as formulated in (van den Oord et al., 2018) learns representations by training neural networks to predict the representations of future observations from those of past ones. When applied to images, CPC operates by predicting the representations of patches below a certain position from those above it.

First, an image is divided into a grid of overlapping patches. Each patch is encoded independently from the rest with a feature extractor (blue) which terminates with a mean-pooling operation, yielding a single feature vector for that patch. Doing so for all patches yields a field of such feature vectors (wireframe vectors). Feature vectors above a certain level (in this case, the center of the image) are then aggregated with a context network (red), yielding a row of context vectors which are used to linearly predict features vectors below. NeurIPS 2019 Talk , Contrastive Predictive Coding v2 (CPC v2)

[IASI AI] Self-supervised learning with "SwAV" (Swapping Assignments between Views)

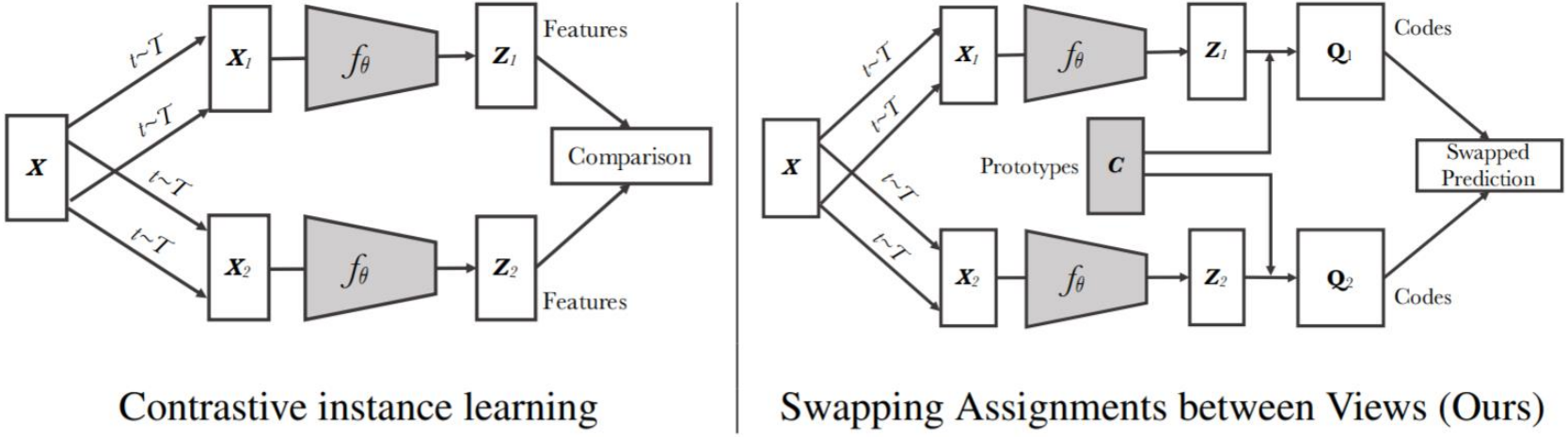


Figure 1: **Contrastive instance learning (left) *vs*. SwAV (right).** In contrastive learning methods applied to instance classification, the features from different transformations of the same images are compared directly to each other. In SwAV, we first obtain "codes" by assigning features to prototype vectors. We then solve a "swapped" prediction problem wherein the codes obtained from one data augmented view are predicted using the other view. Thus, SwAV does not directly compare image features. Prototype vectors are learned along with the ConvNet parameters by backpropragation.

[IASI AI] SwAV - Results

Only six hours and 15 minutes to achieve 72.1 percent top-1 accuracy with a standard ResNet-50 on ImageNet — outperforming the self-supervised method SimCLR trained for 40 hours.

Method	Arch.	Param.	Top1
Supervised	R50	24	76.5
Colorization [64]	R50	24	39.6
Jigsaw [45]	R50	24	45.7
NPID [57]	R50	24	54.0
BigBiGAN [15]	R50	24	56.6
LA [67]	R50	24	58.8
NPID++ [43]	R50	24	59.0
MoCo [24]	R50	24	60.6
SeLa [2]	R50	24	61.5
PIRL [43]	R50	24	63.6
CPC v2 [27]	R50	24	63.8
PCL [36]	R50	24	65.9
SimCLR [10]	R50	24	70.0
MoCov2 [11]	R50	24	71.1
SwAV	R50	24	**75.3**

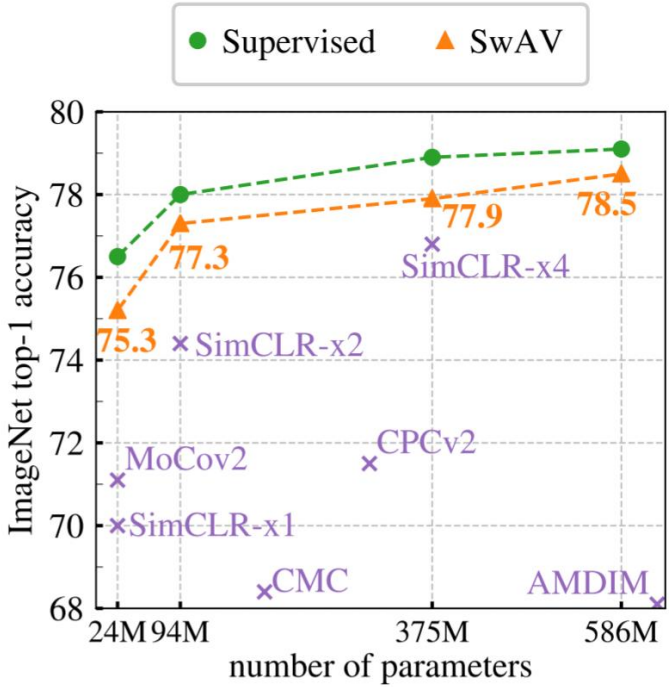
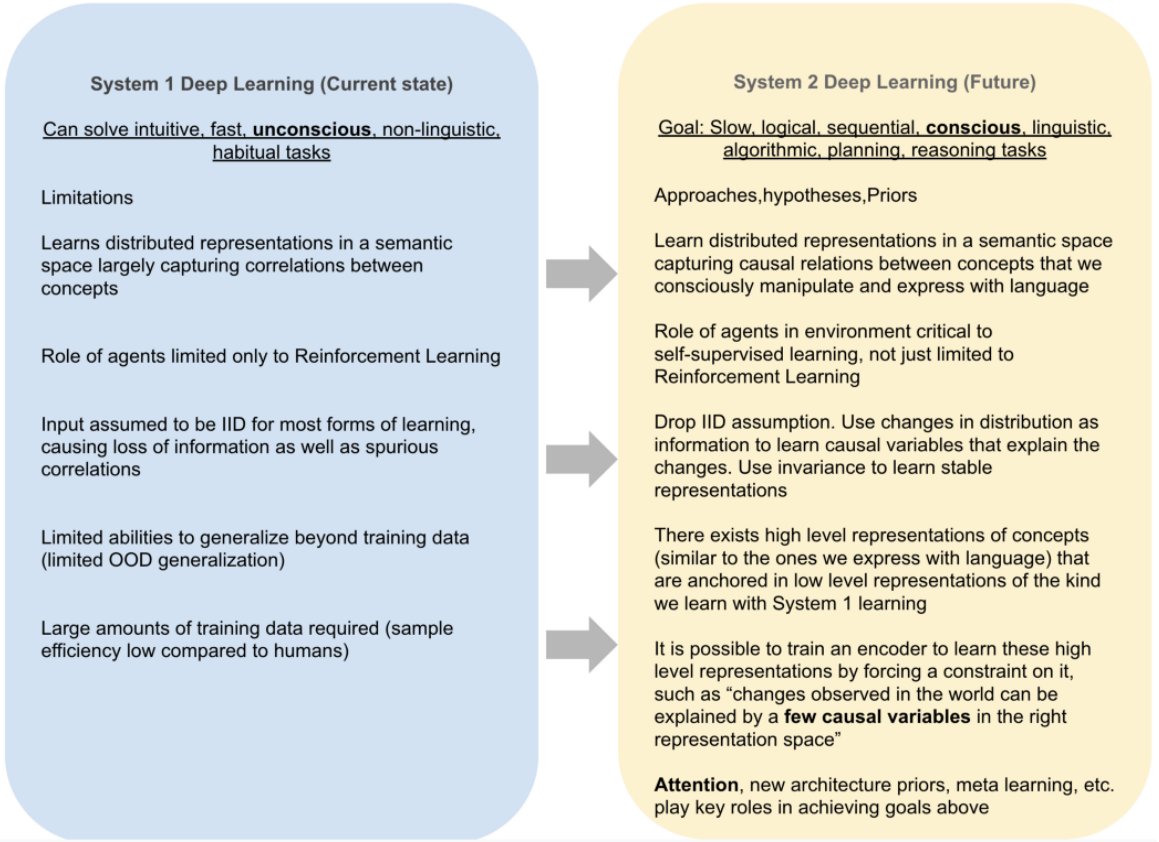


Figure 2: **Linear classification on ImageNet.** Top-1 accuracy for linear models trained on frozen features from different self-supervised methods. **(left)** Performance with a standard ResNet-50. **(right)** Performance as we multiply the width of a ResNet-50 by a factor $\times 2$, $\times 4$, and $\times 5$.

SwAV Article
Other SOTA method

[IASI AI] System 1 vs System 2



"System 1" is fast, instinctive and emotional; "System 2" is slower, more deliberative, and more logical.

Wiki

AAAI-20 Fireside Chat with Daniel Kahneman

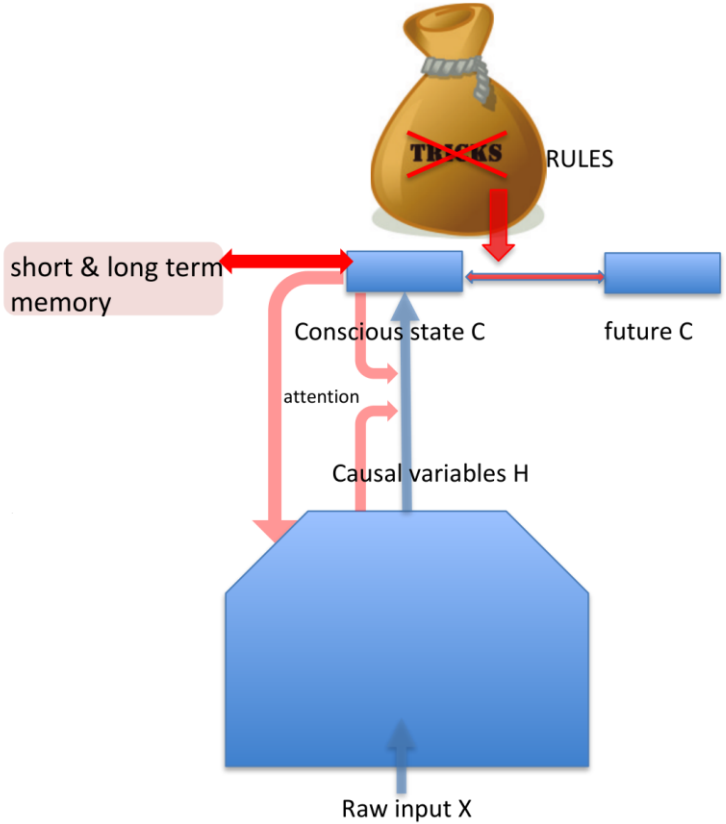
This figure is synthesized from recent talks by Yoshua Bengio (NeurIPS 2019 talk), Yann LeCun and Leon Bottou. Acronym IID in figure expands to Independent and Identically Distributed random variables; OOD expands to Out Of Distribution

Deep Learning beyond 2019

[IASI AI] Causal Consciousness Prior

– Encoder maps sensory data to space where a few sparse rules relate causal variables together, following the consciousness prior

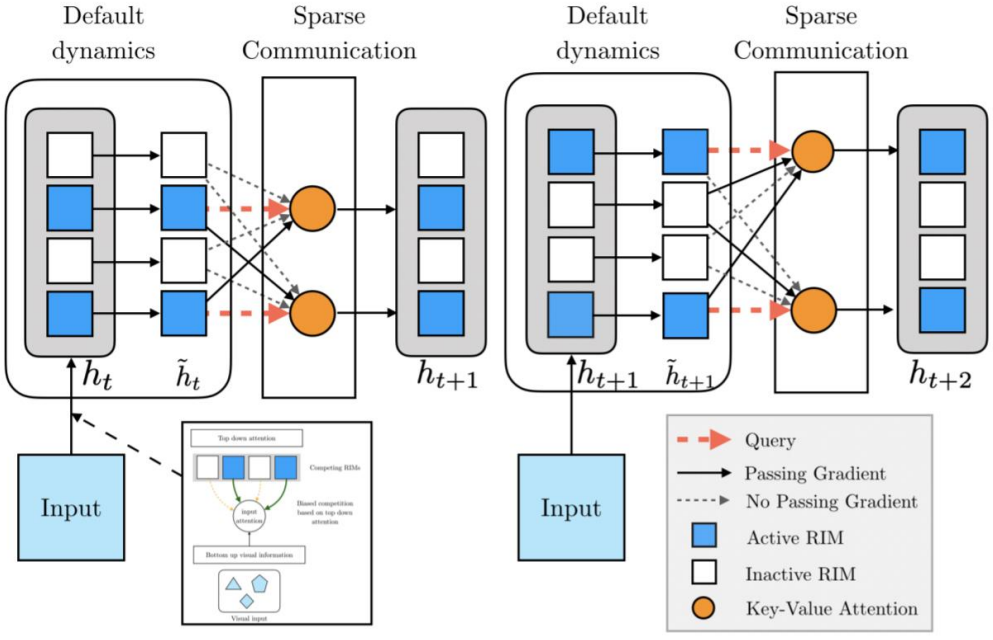
– Need to handle uncertainty in state: P(H|X)

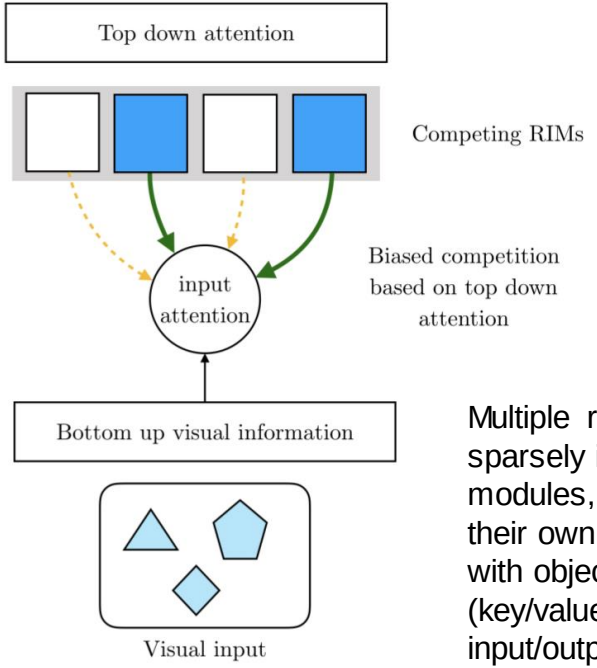


Scientists just proved these two brain networks are key to consciousness, Yoshua Bengio Talk, Attention is a core ingredient of 'conscious' AI

[IASI AI] Recurrent Independent Mechanisms (RIMs) - better OOD at test

A new recurrent architecture in which multiple groups of recurrent cells operate with nearly independent transition dynamics, communicate only sparingly through the bottleneck of attention, and compete with each other so they are updated only at time steps where they are most relevant. We show that this leads to specialization amongst the RIMs





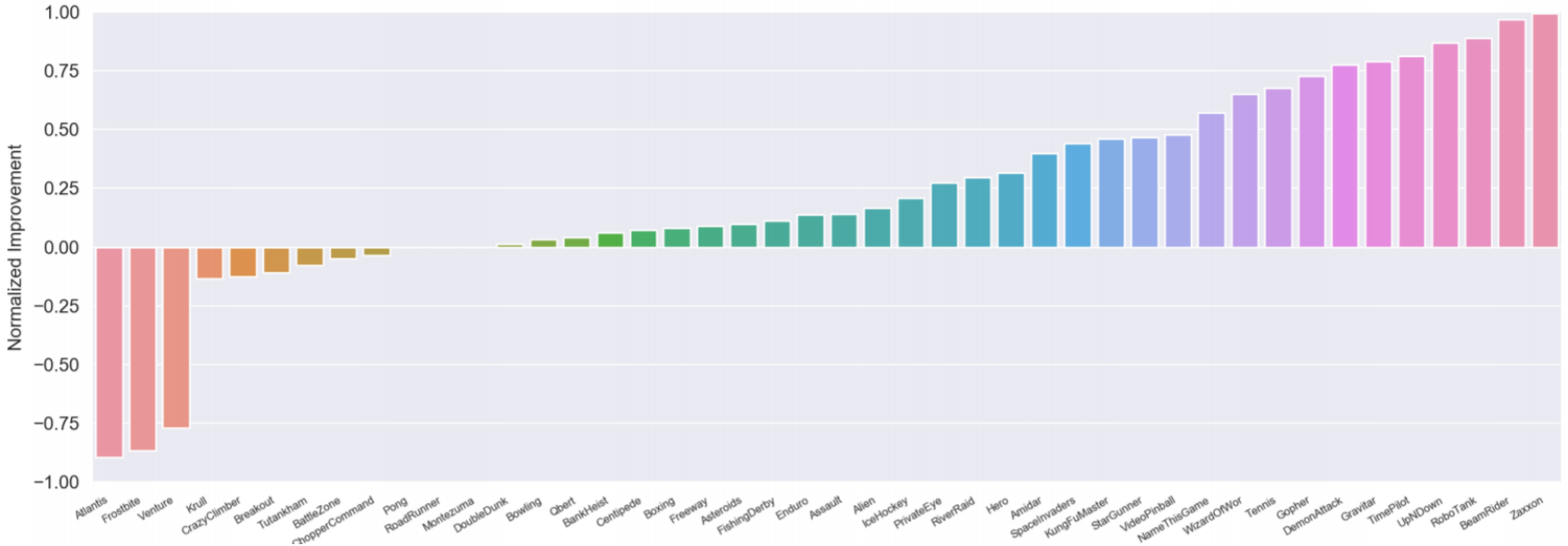
- selective activation of RIMs as a form of top-down modulation
- independent RIM dynamics
- communication between RIMs

"This allows an agent to adapt faster to changes in a distribution or … inference in order to discover reasons why the change happened," said Bengio.

Multiple recurrent sparsely interacting modules, each with their own dynamics, with object (key/value pairs) input/outputs selected by multi-head attention

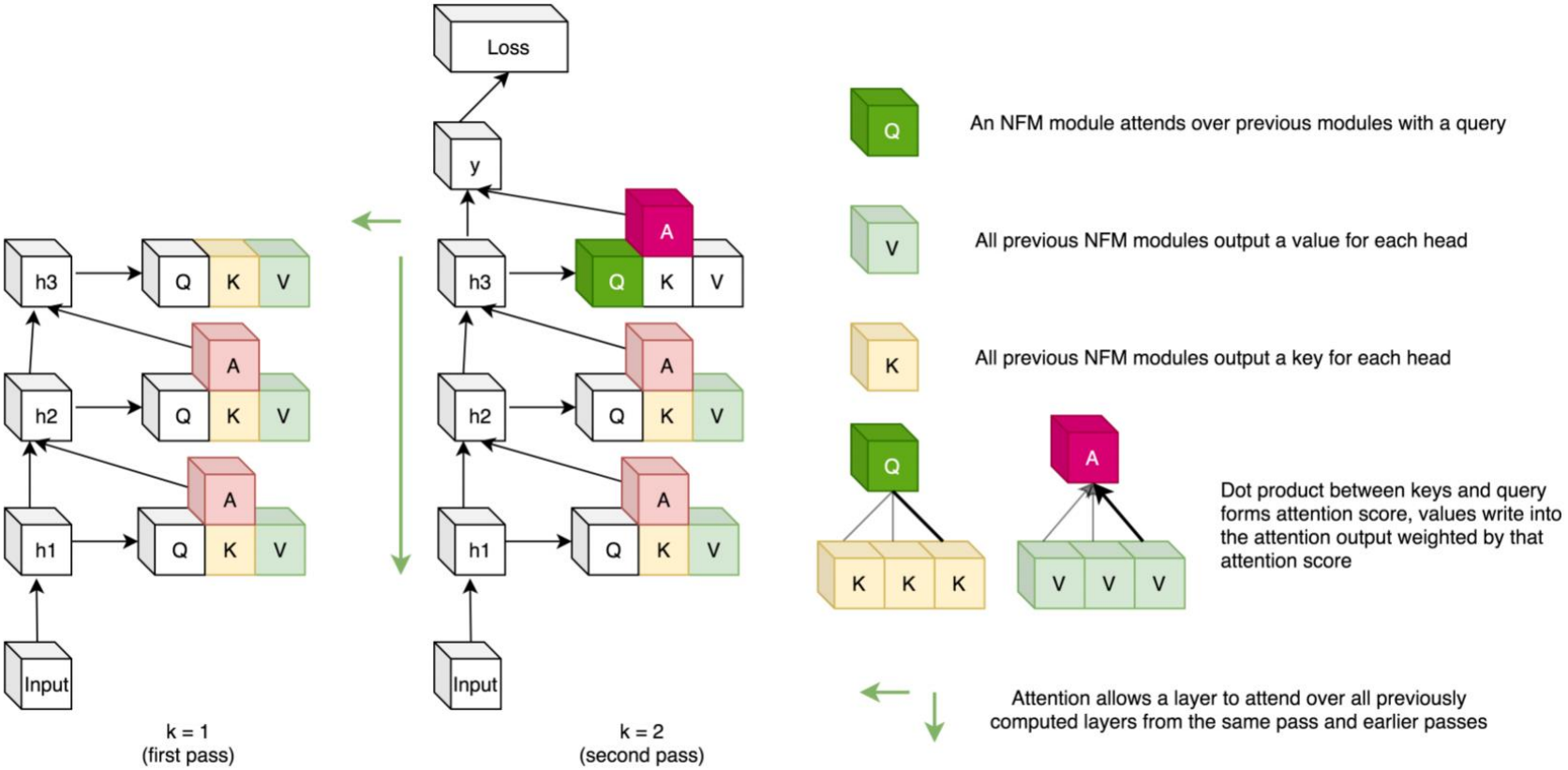
Paper + Code

[IASI AI] Recurrent Independent Mechanisms (RIMs) - Results



RIMs-PPO relative score improvement over LSTM-PPO baseline (Schulman et al., 2017) across all Atari games averaged over 3 trials per game. In both cases, PPO was used with the exact same settings, and the only change is the choice of recurrent architecture.

[IASI AI] Neural Function Modules (NFM) with Sparse Arguments



An illustration of NFM where a network is ran over the input twice (K = 2). A layer where NFM is applied sparsely attends over the set of previously computed layers, allowing better specialization as well as top-down feedback.

Paper

[IASI AI] Towards Continual Learning

Transfer Learning - re-use knowledge for related tasks on either the same or similar datasets
A classic example is learning to recognize cars and then applying the model to the task of recognizing trucks
One type of Transfer Learning is **Domain Adaptation -** learn on one domain, or data distribution, and then apply the model to and optimizing it for a related data distribution

Multi Domain Learning – un-related data distribution

Lifelong Learning - overlaps with Transfer Learning, but the emphasis is gathering general purpose knowledge that transfers across multiple consecutive tasks for an 'entire lifetime'

Curriculum Learning – to learn a specific task train + transfer learn on an easy version of that one task and making it subsequently harder and harder.

In **One-shot Learning**, the algorithm can learn from one or very few examples.

Instance Learning is one way of achieving that, constructing hypotheses from the training instances directly.
A related concept is **Multi-Modal Learning**, where a model is trained on different types of data for the same task. An example is learning to classify letters from the way they look with visual data, and the way they sound, with audio

Meta-learning
- learning to learn (Auto ML, NAS could be a way to implement)
- quick learner - a learner that can generalize from a small number of examples, optimize to quickly adapt to new similar tasks

[IASI AI] Towards Continual Learning

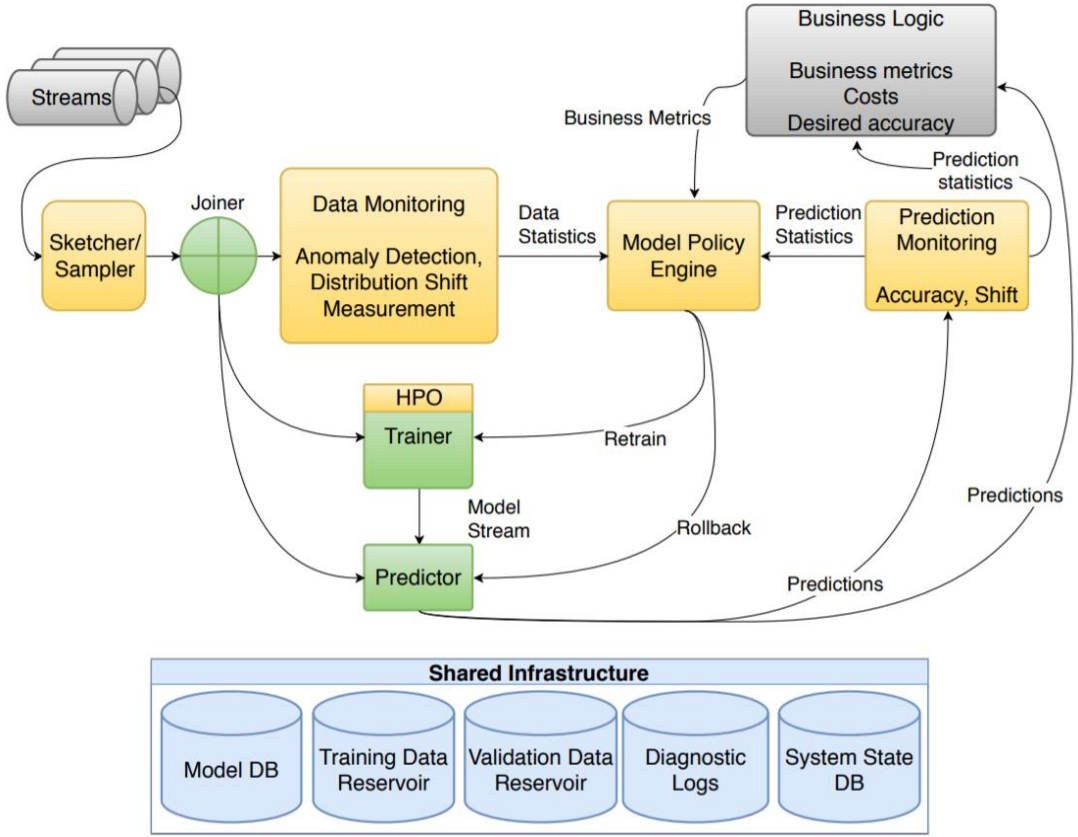
Online Learning - learn iteratively with new data, in contrast to learning from a pass of a whole dataset as commonly done in conventional supervised and unsupervised learning (**Batch Learning**).
- useful when the whole dataset does not fit into memory at once
- or new data is observed over time
- the underlying input data distribution is not static i.e. a non-stationary distribution (**Non-stationary Problems)**
- susceptible to 'forgetting'. That is, becoming less effective at modelling older data. The worst case is failing completely and suddenly, known as **Catastrophic Forgetting** or **Catastrophic Interference**

Incremental Learning (IL), as the name suggests, is about learning bit by bit, extending the model and improving performance over time. It explicitly handles the level of forgetting of past data. In this way, it is a type of online learning that avoids catastrophic forgetting

Now that we have some greater clarity around these terms, we recognize that they are all important features of what we consider to be **Continuous Learning** for a successful AGI agent.

Continual Learning - **(CL)** is the ability of a model to **learn** continually from a stream of data, building on what was learnt previously, hence exhibiting positive transfer, as well as being able to remember previously seen tasks

Continual Learning in Practice



Data flow in the Auto-Adaptive Machine Learning architecture

[IASI AI] Continual Learning desiderata

1. Avoid forgetting
   • Performance over previous tasks should not decrease

2. Fixed memory and compute
   • If not possible, grow sub-linearly with tasks

3. Enable forward transfer
   • Knowledge acquired over previous tasks should help learning future tasks

4. Enable backward transfer
   • While learning the current task, performance in previous tasks may also increase

5. Do not store examples
   • Or store as few as possible

Embracing Change: Continual Learning in Deep Neural Networks (Razvan Pascanu)

[IASI AI] Continual Learning Methods

MODEL GROWING

Increase the model capacity for every new task

PARAMETER ISOLATION

Explicitly identify important parameters for each task

REGULARIZATION

Penalize (some) parameter variations

KNOWLEDGE DISTILLATION

Use the model in a previous training state as a teacher

REHEARSAL

Store old inputs and replay them to the model.

[IASI AI] Continual Learning - Methods

Task-specific methods:

- to reduce interference between tasks use different parts of a neural network for different problems.
- let a neural network grow or recruit new resources when it encounters new tasks

Cons: require knowledge of the task identity at both training and test time - not suitable for class-incremental learning

Regularization-based methods:

- add a regularization loss to penalise changes to model parameters that are important for previous tasks

Cons: gradually reduce the model's capacity for learning new tasks

Replay methods:

- To preserve knowledge, replay methods periodically rehearse previously acquired information during training. Exact or experience replay store data from previous tasks and revisit them when training on a new task.

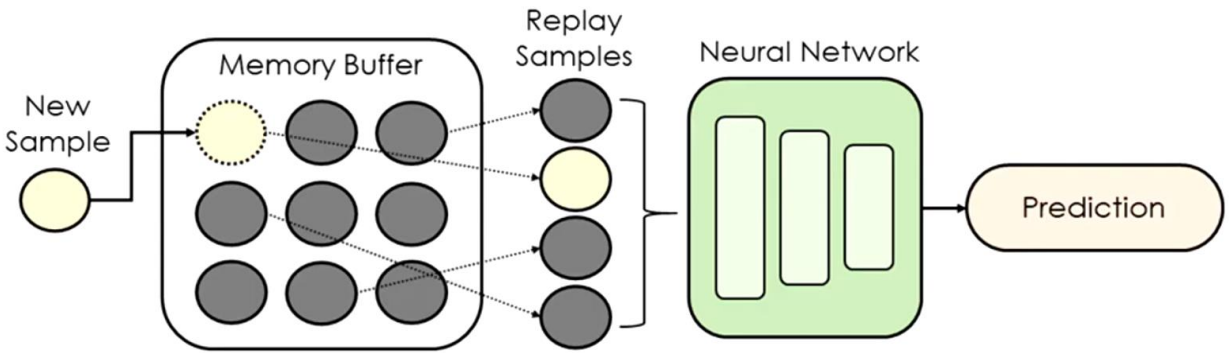
Cons:

- Storing data might not always be possible

Replay in Artificial Neural Networks

- **Replay (rehearsal)** is one of the most effective methods for mitigating forgetting in neural networks

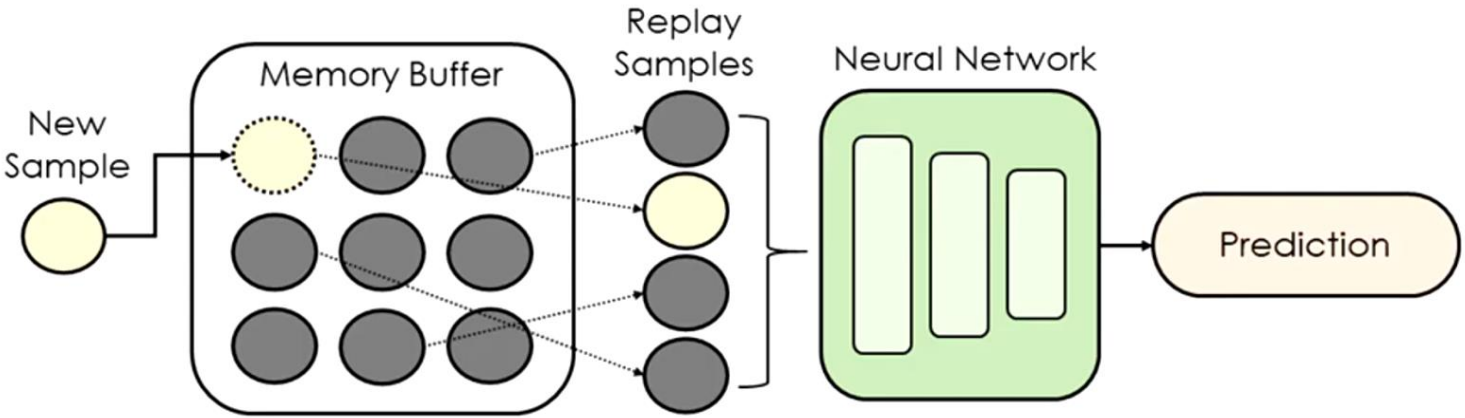
- Involves storing previous examples in a buffer and mixing new instances with old ones to fine-tune the network



[IASI AI] Replay in Artificial Neural Networks

1. Hippocampal indexing theory* postulates that the hippocampus stores **compressed representations** of neocortical activity patterns, which are reactivated during consolidation
    a) Visual inputs are high in the visual processing hierarchy, e.g., not raw pixel representations

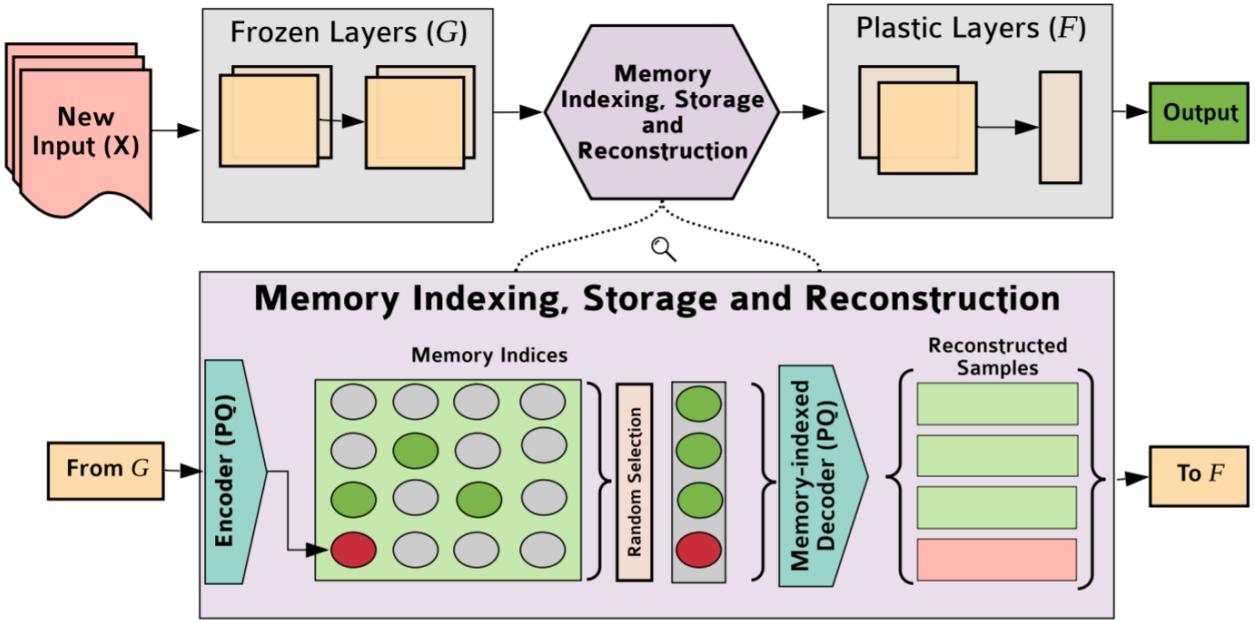
2. Animals perform immediate online streaming learning from non-iid experiences

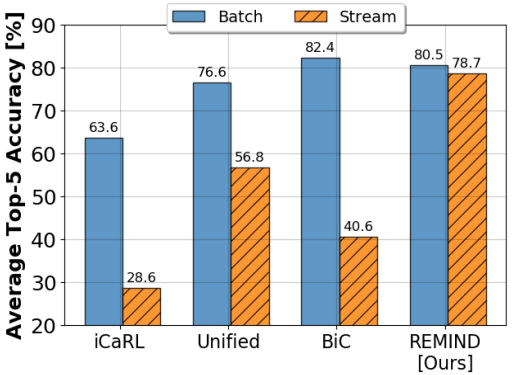


*Teyler, T. J., & Rudy, J. W. (2007). The hippocampal indexing theory and episodic memory: updating the index. Hippocampus.

[IASI AI] Online streaming learning with REMIND

REMIND Your Neural Network to Prevent Catastrophic Forgetting:



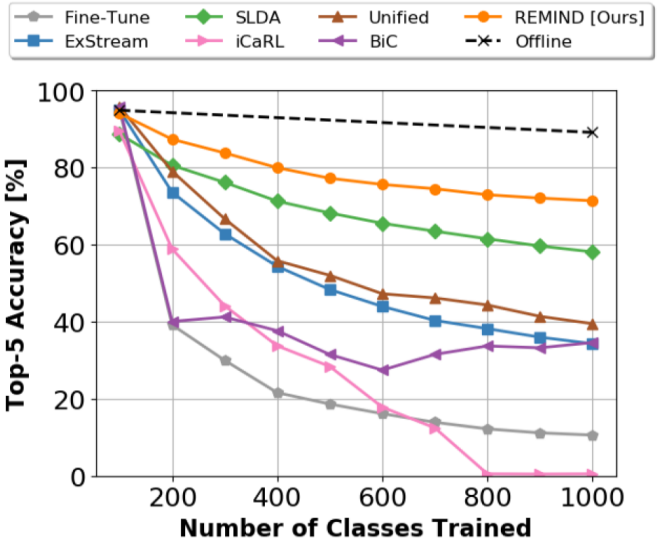


Average top-5 accuracy results for streaming and incremental batch versions of state-of-theart models on ImageNet.

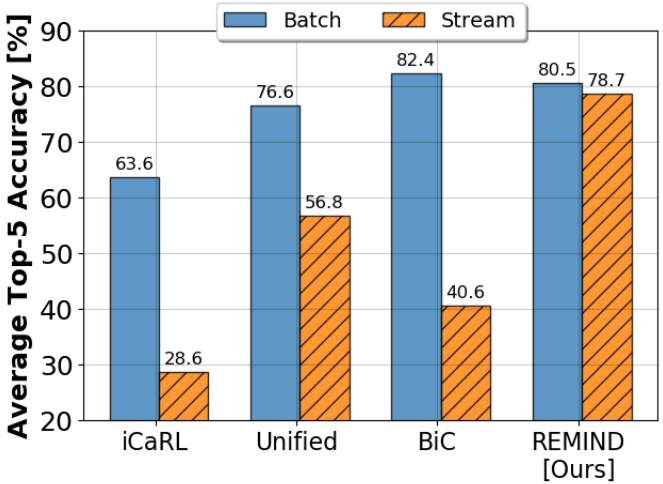
REMIND takes in an input image and passes it through frozen layers of the network (G) to obtain tensor representations (feature maps). It then quantizes the tensors via product quantization and stores the indices in memory for future replay. The decoder reconstructs tensors from the stored indices to train the plastic layers (F) of the network before a final prediction is made.

Paper + Code, Summary, CLVision Workshop @ CVPR 2020 Talk

[IASI AI] Online streaming learning with REMIND



Performance of streaming ImageNet models.

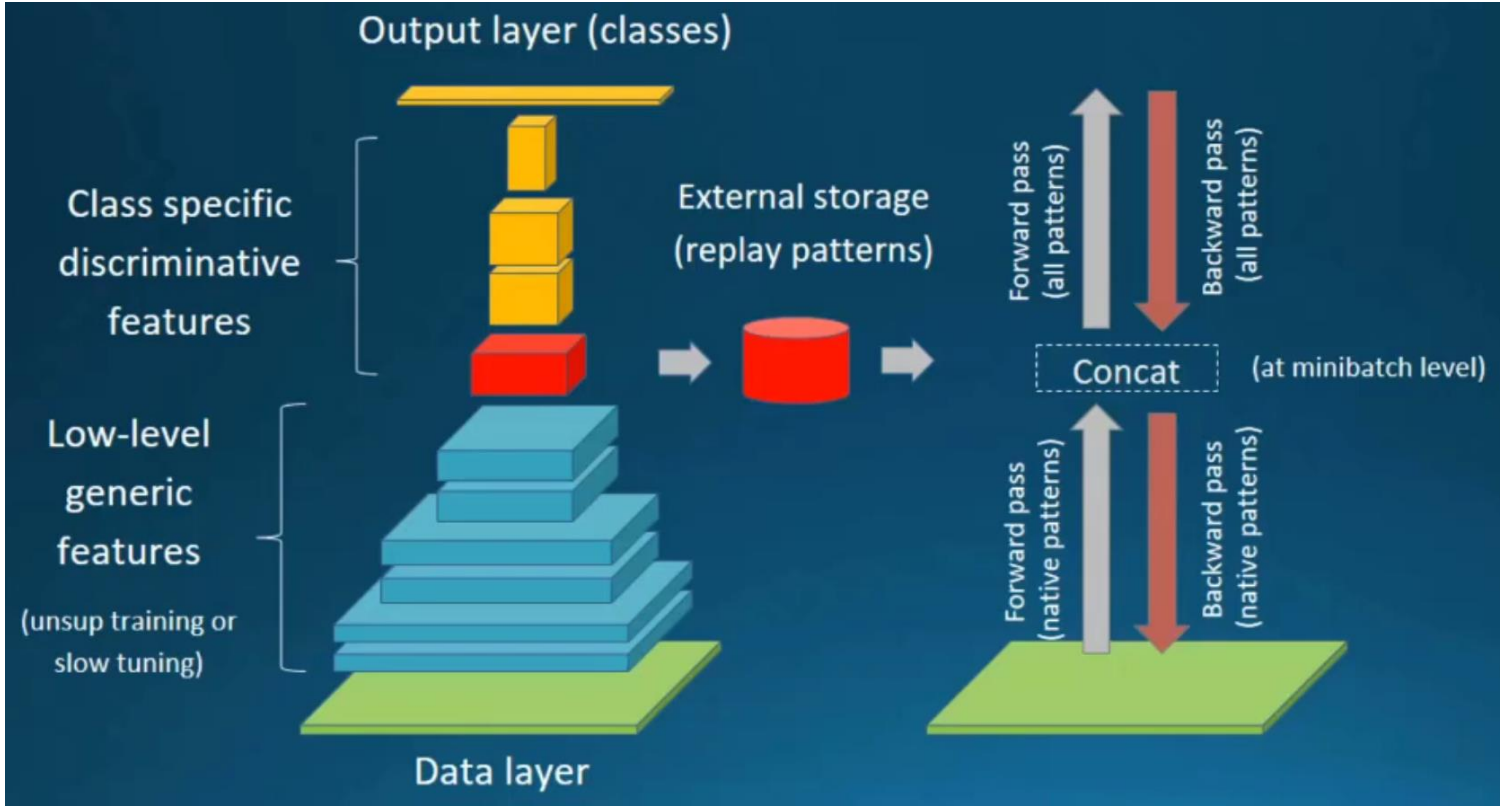


Average top-5 accuracy results for streaming and incremental batch versions of state-of-the art models on ImageNet.

- REMIND achieves state-of-the-art results compared to recent methods in CVPR-2019 (BiC, Unified).
- All models use same ResNet-18 CNN pre-trained with 100 ImageNet classes before continual learning.
- Streaming mode for incremental batch methods: Small batch size of 50 examples and only a single epoch.
- For ImageNet, iCaRL, Unified, BiC, and REMIND are given 1.5 GB of auxiliary memory. iCaRL, Unified, and BiO store raw images.

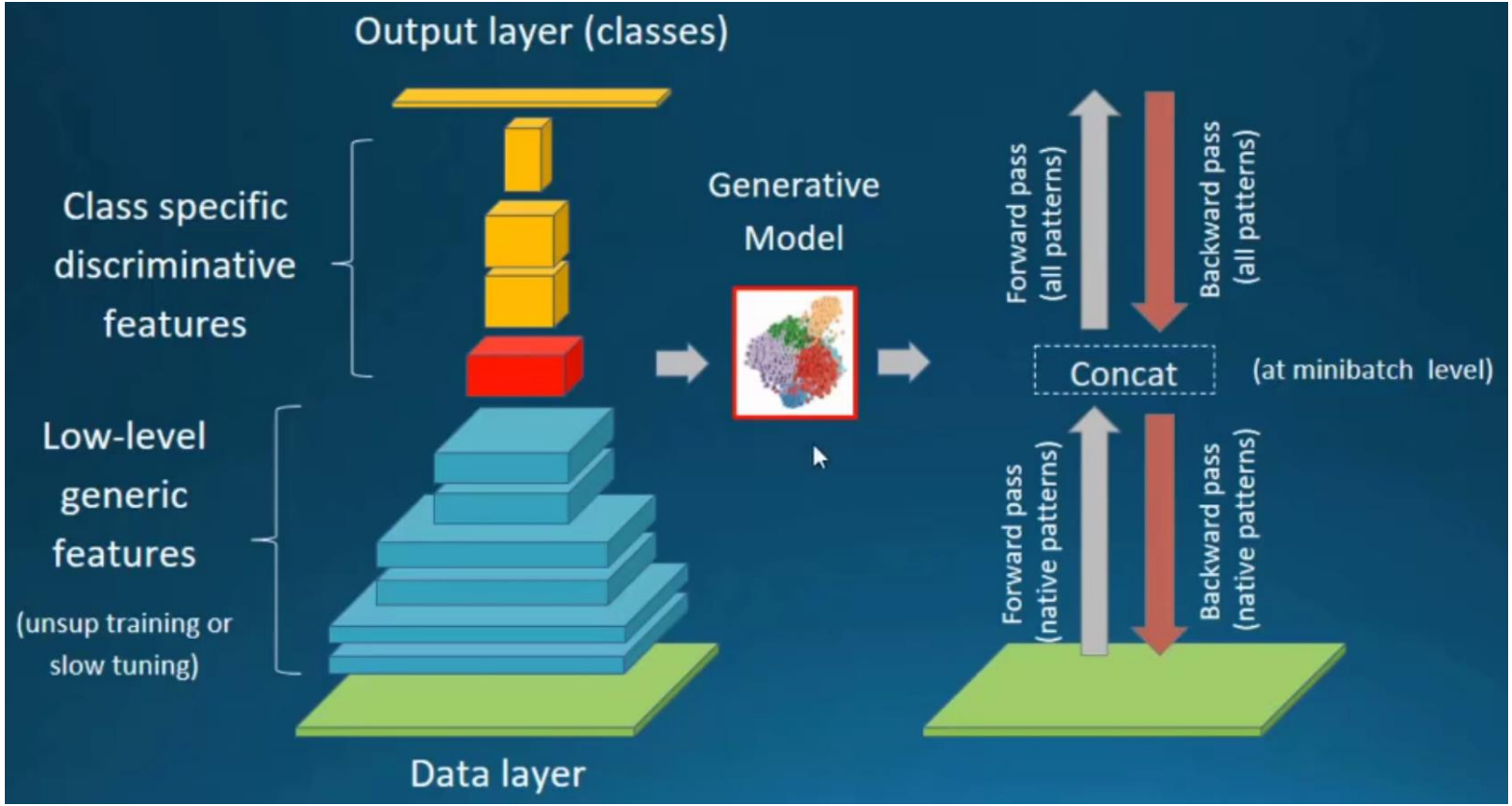
RODEO: Replay for Online Object Detection (Summary)

[IASI AI] AR-1 with Latent Replay



Latent Replay for Real-Time Continual Learing at the Edge (The CL CORe App) (Demo) (Paper)

[IASI AI] AR-1 with Latent Generative Replay (WIP)



([CLVision @ CVPR2020] Invited Talk: Real-Time CL from Short Videos)

[IASI AI] AR-1 with Latent Generative Replay (WIP) - Practical Issues

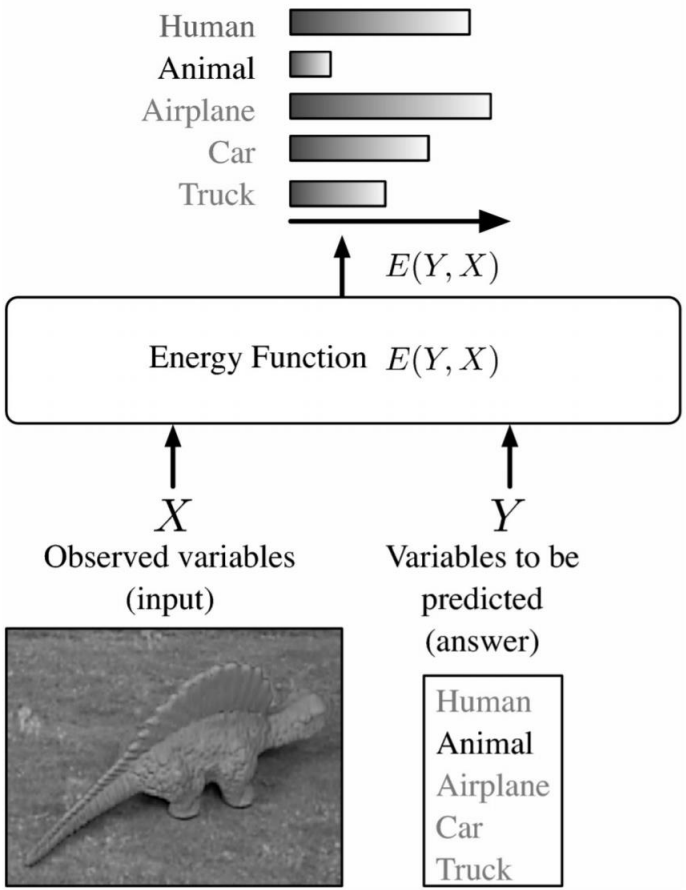
1. Training (continuously) a model to generate useful images does not scale to complex datasets
   - Ok for MNIST, CIFAR10, but not for CIFAR100, ImageNet, Core50
2. Generating features (instead of row data) was recently proved* to be a good alternative.
   - However, generation was performed at the semi-last (flat) level
3. We would like to train the generator online.
   - GAN training could be to slow

Challenge: generating complex activation volumes in near-real time.

*Xialei Liu, et al. ***Generative Feature Replay For Class-Incremental Learning***, arXiv:2004.09199, 2020.

(CLVision @ CVPR2020] Invited Talk: Real-Time CL from Short Videos)

Energy Based Models for Decision Making



- **Model:** Measures the compatibility between an observed variable X and a variable to be predicted Y through an energy function E(Y,X).

$$Y^* = \text{argmin}_{Y \in \mathcal{Y}} E(Y, X).$$

- **Inference:** Search for the Y that minimizes the energy within a set $\mathcal{Y}$
- If the set has low cardinality, we can use exhaustive search.

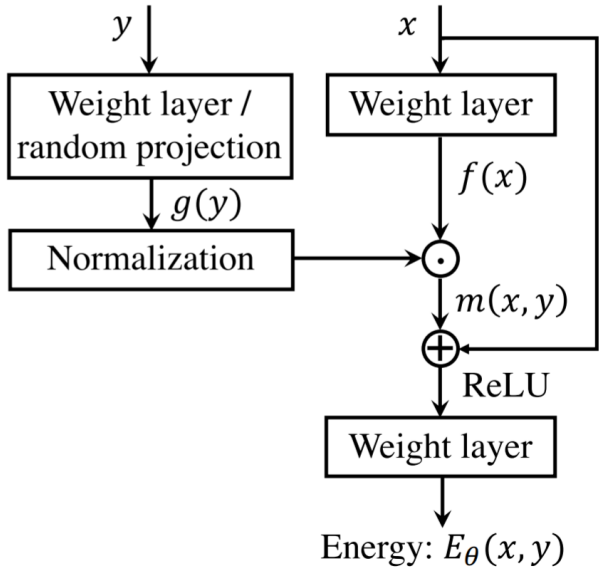
Slides (LeCun Tut 2006)

[IASI AI] Energy Based Models (EBM) for Continual Learning

We show that EBMs are naturally adaptable to a more general continual learning setting that the data distribution gradually changes without the notion of separate tasks.
Our EBMs reduce catastrophic forgetting without requiring knowledge of task-identity, without gradually restricting the model's learning capabilities and without using stored data

Why:

- energy score mitigates a critical problem of softmax confidence with arbitrarily high values for OOD examples
- no normalizing over all classes but instead sampling a single negative class from the current training batch = less interference with previous classes
- we can treat y as an attention filter or gate to select the most relevant information between x and y

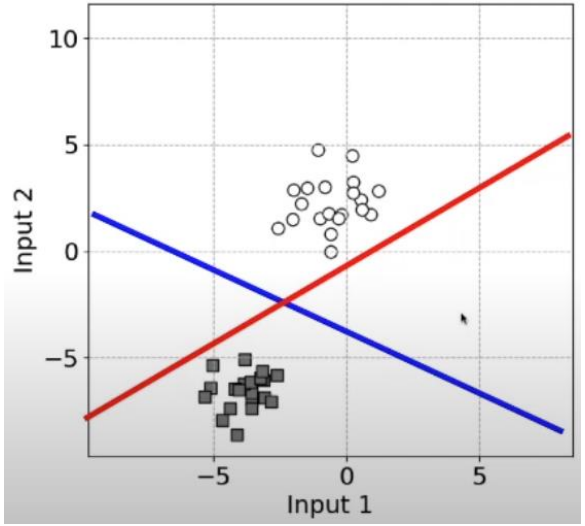


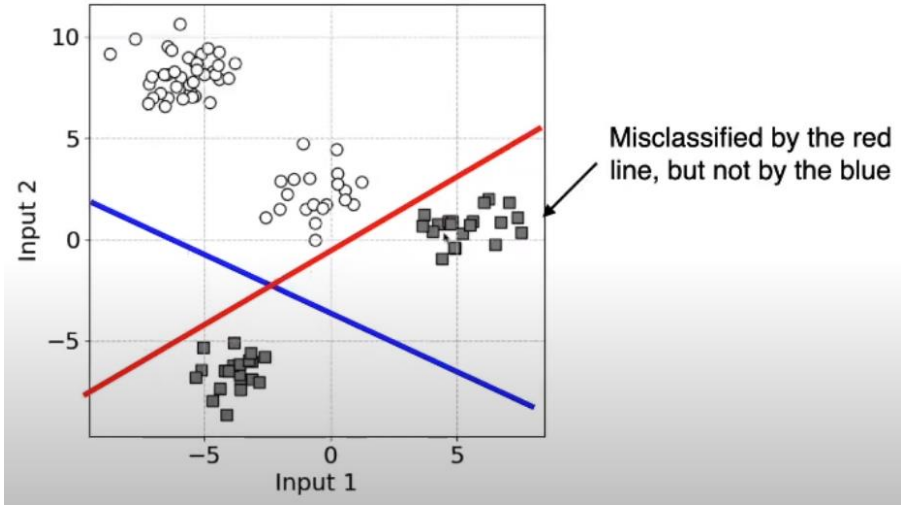
Energy-Based Models Lesson , EBM for Continual Learning , Energy-based Out-of-distribution Detection

[IASI AI]

Bayesian Learning

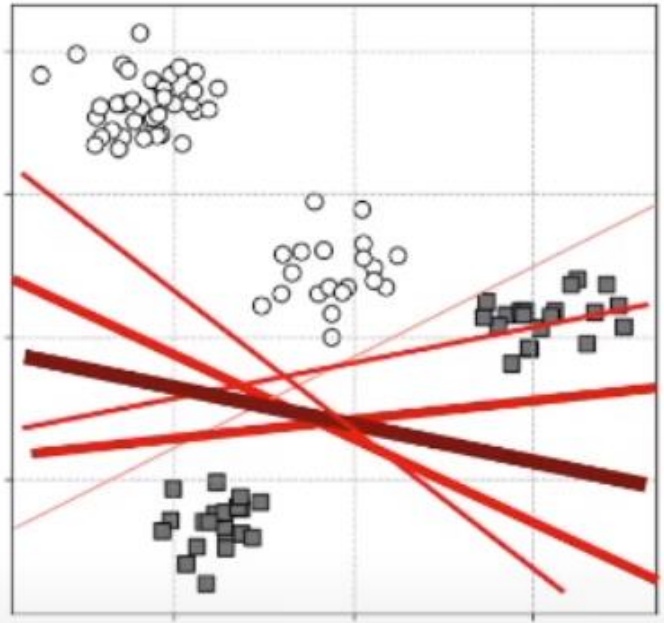
Application: Uncertainty estimation in deep learning
Application: Continual learning of deep networks





Deep Learning with Bayesian Principles

Bayesian Learning



(1) Keep your options open

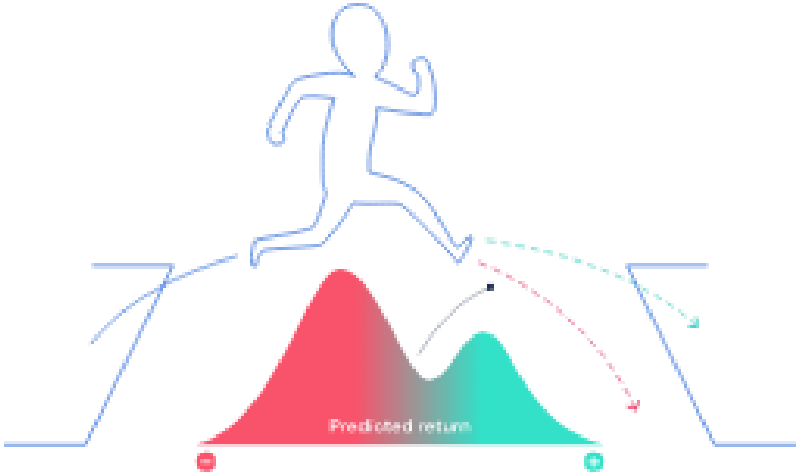
$$p(\theta|\mathcal{D}_1) = \frac{p(\mathcal{D}_1|\theta)p(\theta)}{\int p(\mathcal{D}_1|\theta)p(\theta)d\theta}$$

(2) Revise with new evidence

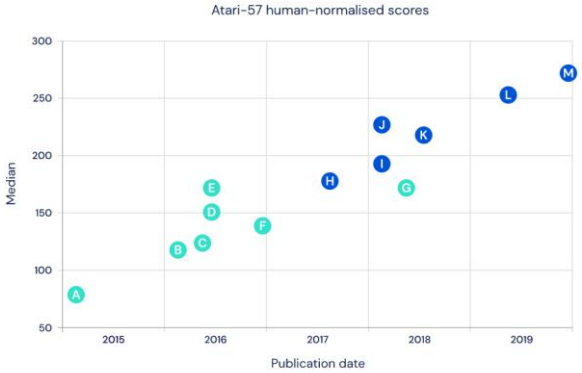
$$p(\theta|\mathcal{D}_2, \mathcal{D}_1) = \frac{p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)}{\int p(\mathcal{D}_2|\theta)p(\theta|\mathcal{D}_1)d\theta}$$

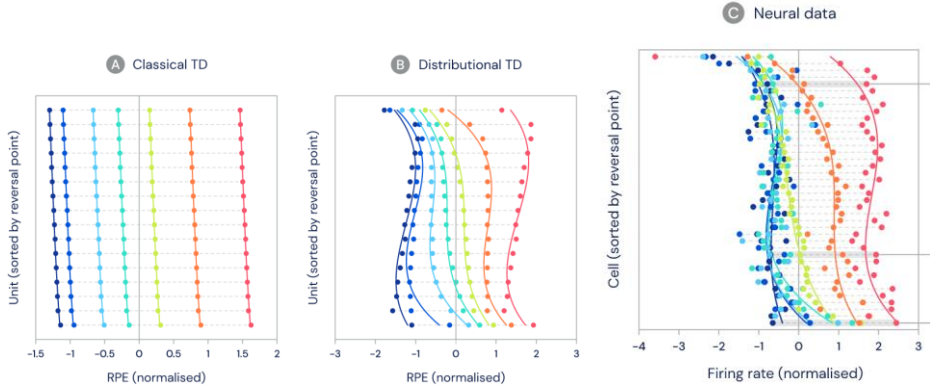
Bayesian Principles for Learning Machines

[IASI AI] Distributional Reinforcement Learning



When the future is uncertain, future reward can be represented as a probability distribution. some possible futures are good (teal), others are bad (red). distributional reinforcement learning can learn about this distribution over predicted rewards through a variant of the td algorithm.





Distributional reinforcement learning

[IASI AI] Common-Sense Reasoning via Grounded Language

Learn Grounded Language via Visual Question and Answering

- Supervised Learning for Image Captioning
  - Train on images paired with sentences that describe the images
  - Train on images paired with word tags mapped to specific objects in a image (New from MS!)

- Unsupervised Learning using "Vokenization" (New!) (find corresponding image vokens to word tokens)

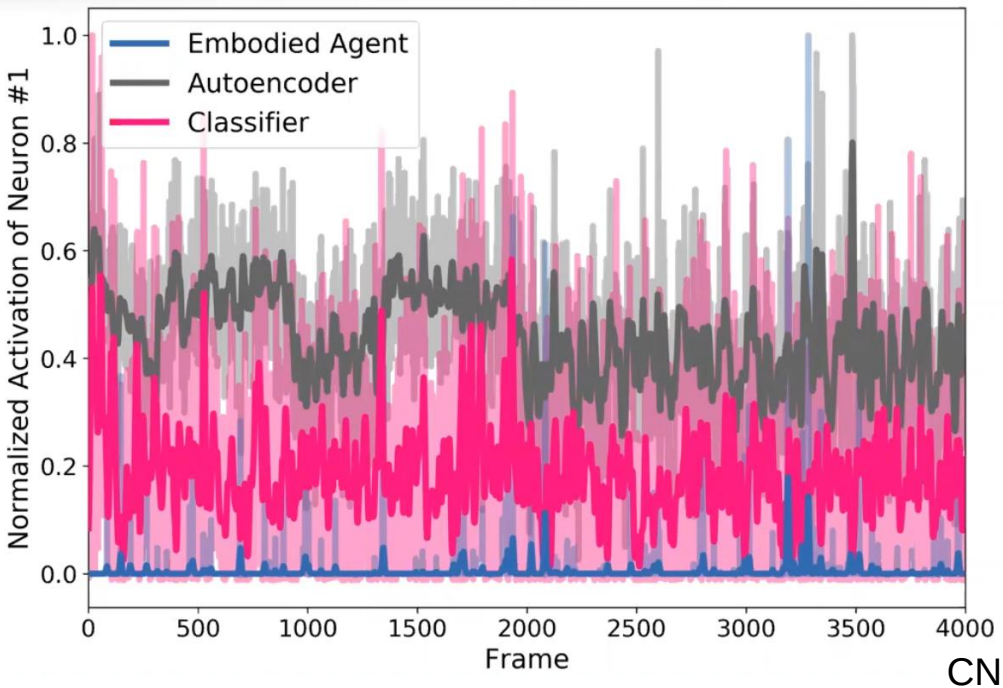
Results: "The algorithm only found vokens for roughly 40% of the tokens. But that's still 40% of a data set with nearly 3 billion words."

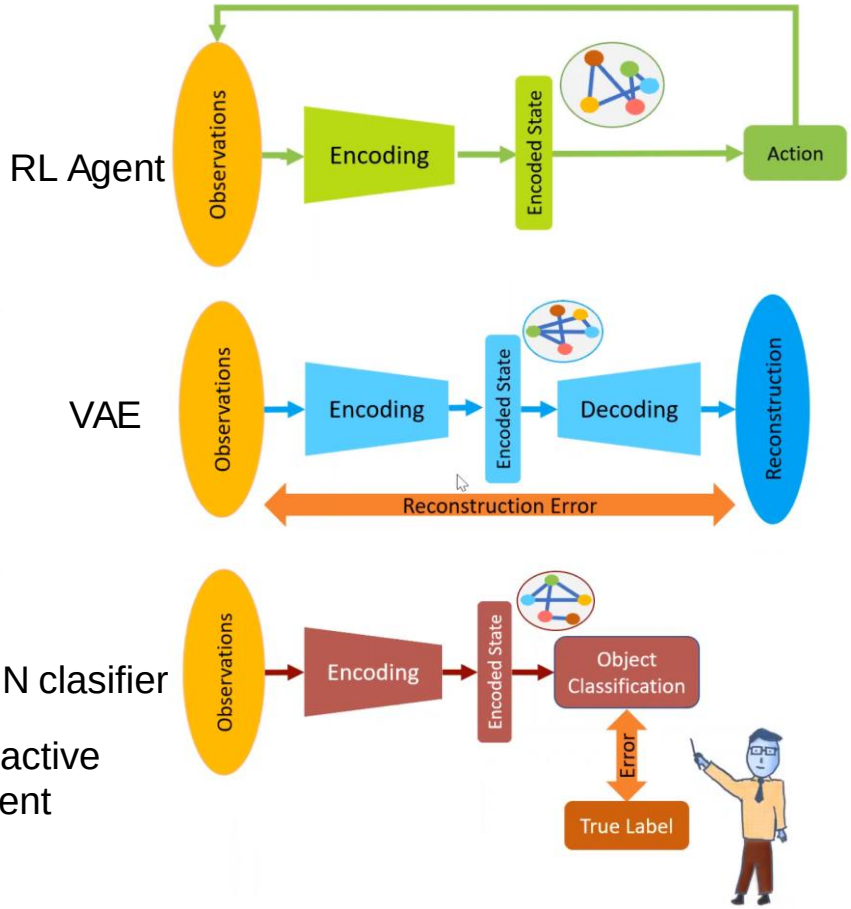
This could lead to the next big breakthrough in common sense AI
Microsoft AI breakthrough in automatic image captioning (Blog)

VIVO: Surpassing Human Performance in Novel Object Captioning with Visual Vocabulary Pre-Training
"This is done by pre-training a multi-layer Transformer model that learns to align image-level tags with their corresponding image region features"

Vokenization Explained!

[IASI AI] Sparse and Meaningful Representations Through Embodiment





While the autoencoder and the classifier neurons are mostly active all the time and modulate the strength of their activity, the agent learns a more spike-like activation pattern

Talk, Paper

[IASI AI] Neuromorphic Computing - Links

Neuromorphic computing finds new life in machine learning
Soft-grasping with an anthropomorphic robotic hand using spiking neurons (Article)
New learning algorithm should significantly expand the possible applications of AI
Brain-Inspired Robot Controller Uses Memristors for Fast, Efficient Movement

[IASI AI] Sparsity - Links

How Sparsity Adds Umph to AI Inference
Sparsity in Reservoir Computing Neural Networks
SBNet: Leveraging Activation Block Sparsity for Speeding up Convolutional Neural Networks
Is the future of Neural Networks Sparse? An Introduction
Google Introduces RigL Algorithm For Training Sparse Neural Networks
Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices
Gradient Flow in Sparse Neural Networks and How Lottery Tickets Win

[IASI AI] Neuroscience - Links

Human intelligence just got less mysterious, neuroscientist says
Single Neuron Coding of Identity in the Human Hippocampal Formation
Researchers discover 'spooky' similarity in how brains and computers see
Yoshua Bengio: Attention is a core ingredient of ‘conscious’ AI
Canonical Microcircuits for Predictive Coding
The Genius Neuroscientist Who Might Hold the Key to True AI

[IASI AI] Continual Learning - Links

Nick Cheney - Learning to Continually Learn

ContinualAI Organization

The new 'Learn to Grow' framework makes Artificial Intelligence systems a lifelong learner (Paper)

DeepMind's PathNet: A Modular Deep Learning Architecture for AGI

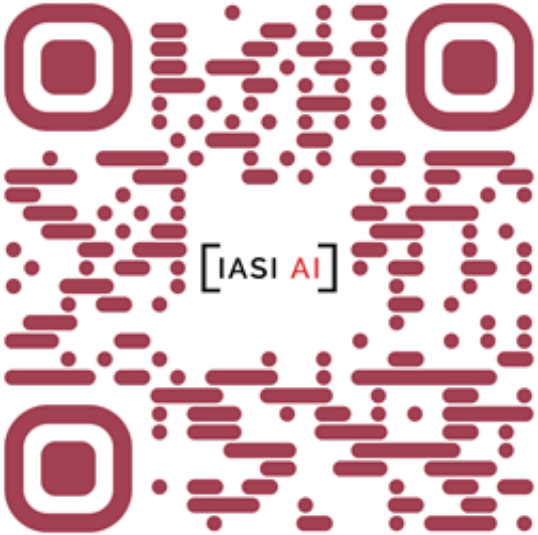
Deep Learning Architecture Search and the Adjacent Possible

Continual Lifelong Learning with Neural Networks: A Review

IBM's Quest to Solve the Continual Learning Problem and Build Neural Networks Without Amnesia

Framework improves 'continual learning' for artificial intelligence

Thank You!



meetup.com/IASI-AI/

facebook.com/AI.Iasi/

Community Partners