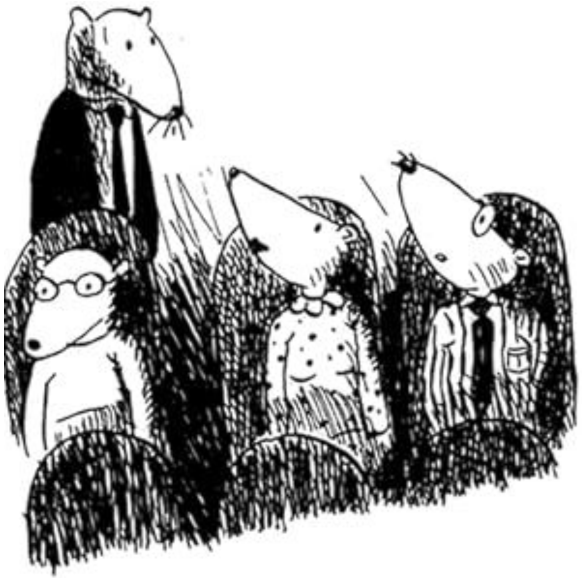
Loss of Plasticity in Deep Continual Learning

Ameli Alaeva

PLAN

- Intro
- Paper Review
  - Goals
  - Dataset & Methods
  - Experiments + Results
  - Continual Backpropagation
- Conclusion
- Q&A in the end



Deep Learning

(SGD, Backpropagation …)

<u>2 phases:</u>

- Training (weights of the network are updated)
- Evaluation / inference (weights are held constant)

NOW

Deep Learning

<u>New Data?</u>

→ Oh, I’m sorry, but *for better results* train from scratch :(

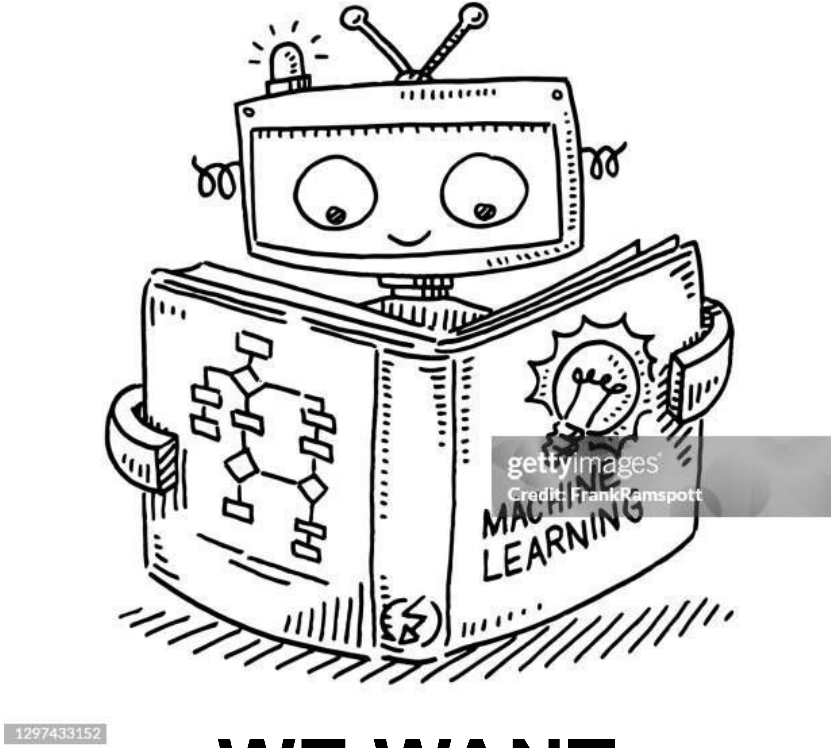
(old + new data together)

<u>But LLMs are large…</u>

→ Well, too bad :(

NOW

Deep ***Continual*** Learning



WE WANT

Deep ***Continual*** Learning

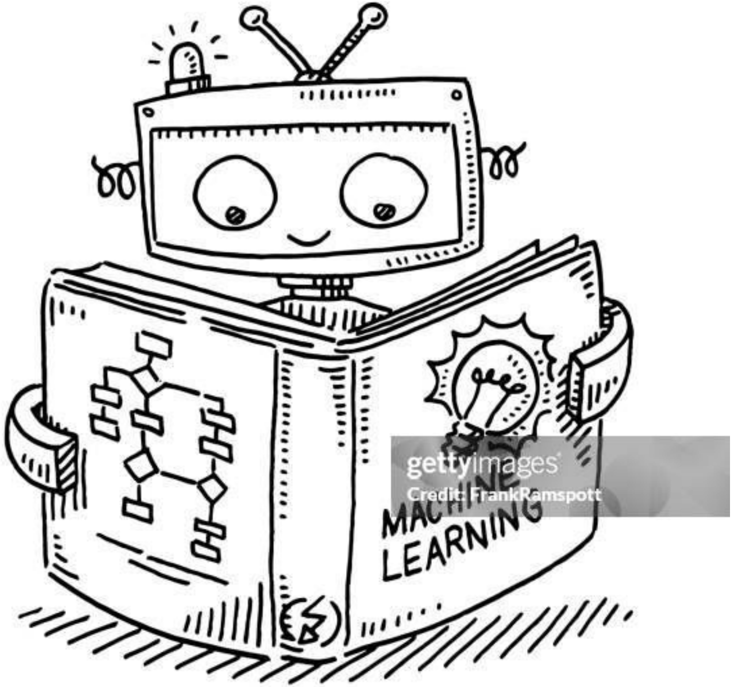
Trained Model (Inference) →

→ ***New*** Data / Task →

→ Train Model *Again* (***update weights***) –

→ ***Great Performance Old + New*** →

→ New Data / Task → …



WE WANT

Deep ***Continual*** Learning

Trained Model (Inference) →

→ ***New*** Data / Task →

→ Train Model *Again* (***update weights***) →

→ ~~***Great Pe*** ***Old + New***~~ →

→ New Data / Task → …

Bad Performance

WE <u>HAVE</u>

WE HAVE

Loss of plasticity

Article | Open access | Published: 21 August 2024

Loss of plasticity in deep continual learning

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood & Richard S. Sutton

Nature **632**, 768–774 (2024) | Cite this article

Goals of the Research

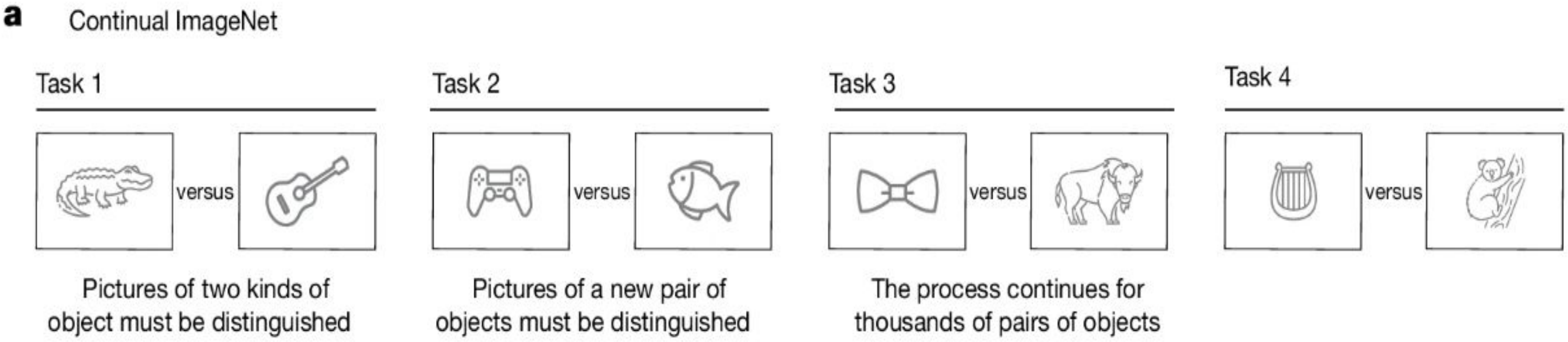
- Demonstrate loss of plasticity in standard deep-learning systems
- Propose algorithm, that maintains plasticity indefinitely

Convincing Demonstration Strategy

- Systematic and extensive
- *Wide range* of standard deep-learning networks, learning algorithms and parameter settings
- For each of these, the experiments run long enough to expose *long-term plasticity loss*
- *Repeated* enough times to obtain *statistically significant* results

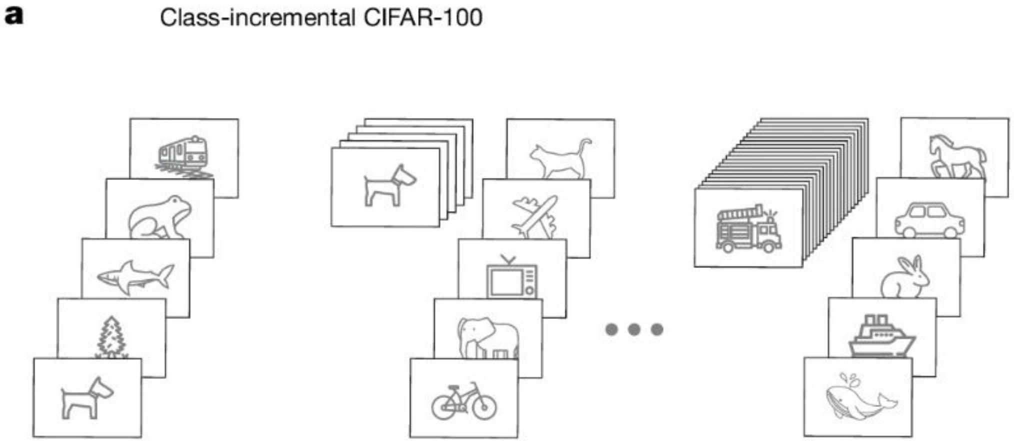
Datasets

<u>Continual ImageNet</u>



Datasets

Class-Incremental CIFAR-100

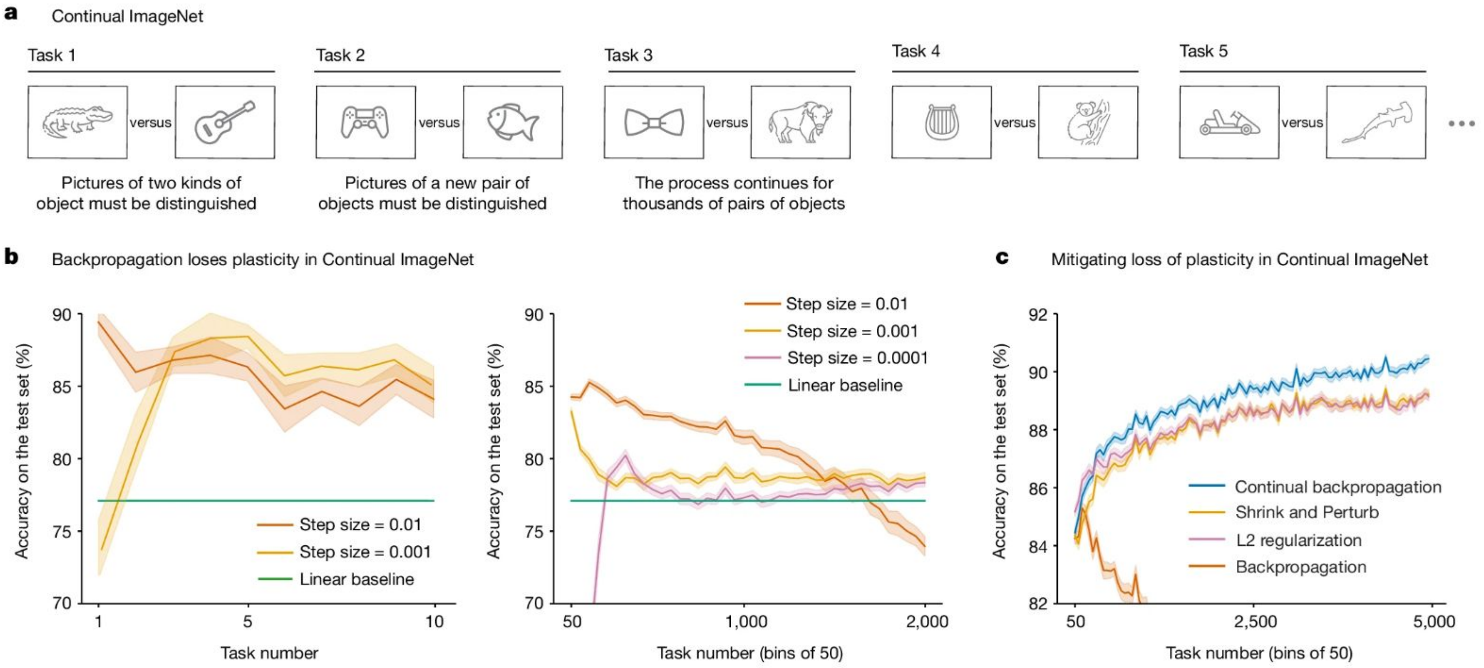


At the start, pictures from five classes have to be distinguished

Five more classes are added and pictures from all ten have to be distinguished

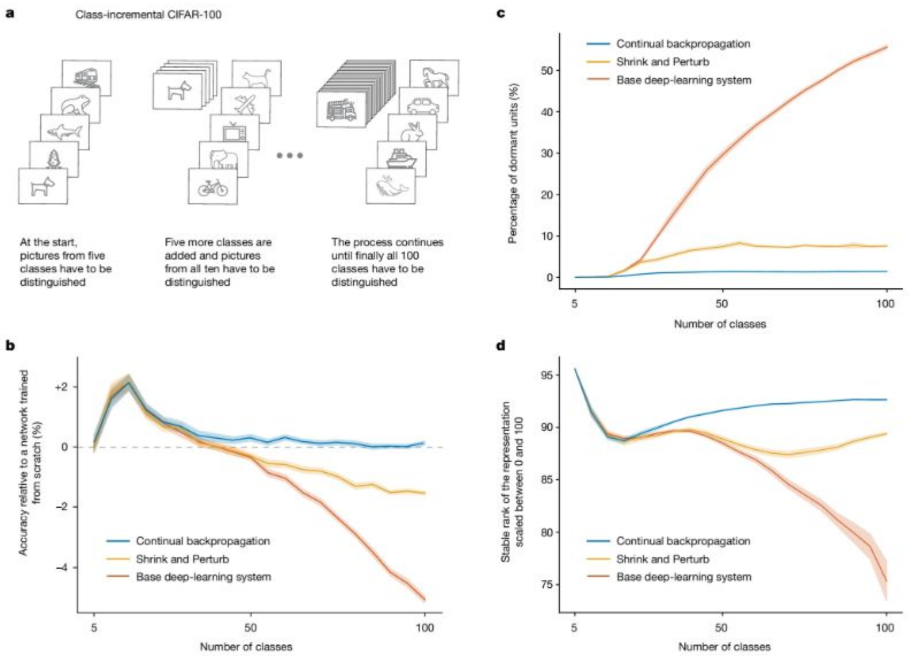
The process continues until finally all 100 classes have to be distinguished

Experiments + Results



a–**c**, In a sequence of binary classification tasks using ImageNet pictures (**a**), the conventional backpropagation algorithm loses plasticity at all step sizes (**b**), whereas the continual backpropagation, L2 regularization and Shrink and Perturb algorithms maintain plasticity, apparently indefinitely (**c**). All results are averaged over 30 runs; the solid lines represent the mean and the shaded regions correspond to ±1 standard error.

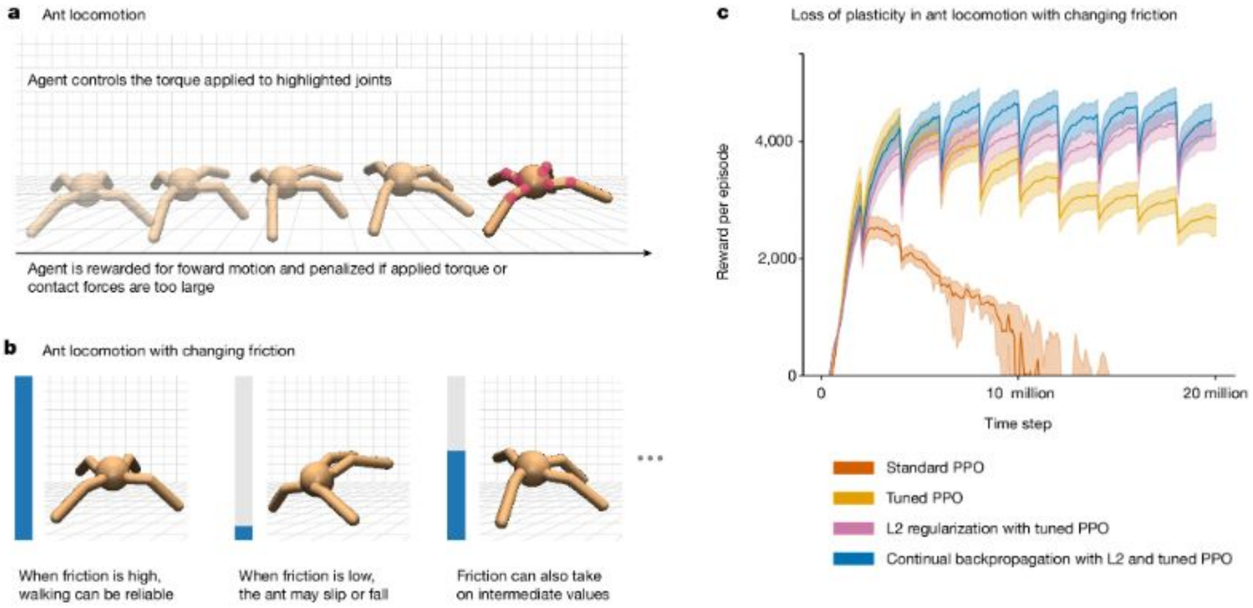
Experiments + Results



a, An incrementally growing image-classification problem. **b**, Initially, accuracy is improved by incremental training compared with a network trained from scratch, but after 40 classes, accuracy degrades substantially in a base deep-learning system, less so for a Shrink and Perturb learning system and not at all for a learning system based on continual backpropagation. **c**, The number of network units that are active less than 1% of the time increases rapidly for the base deep-learning system, but less so for Shrink and Perturb and continual backpropagation systems. **d**, A low stable rank means that the units of a network do not provide much diversity; the base deep-learning system loses much more diversity than the Shrink and Perturb and continual backpropagation systems. All results are averaged over 30 runs; the solid lines represent the mean and the shaded regions correspond to ±1 standard error.

Experiments + Results

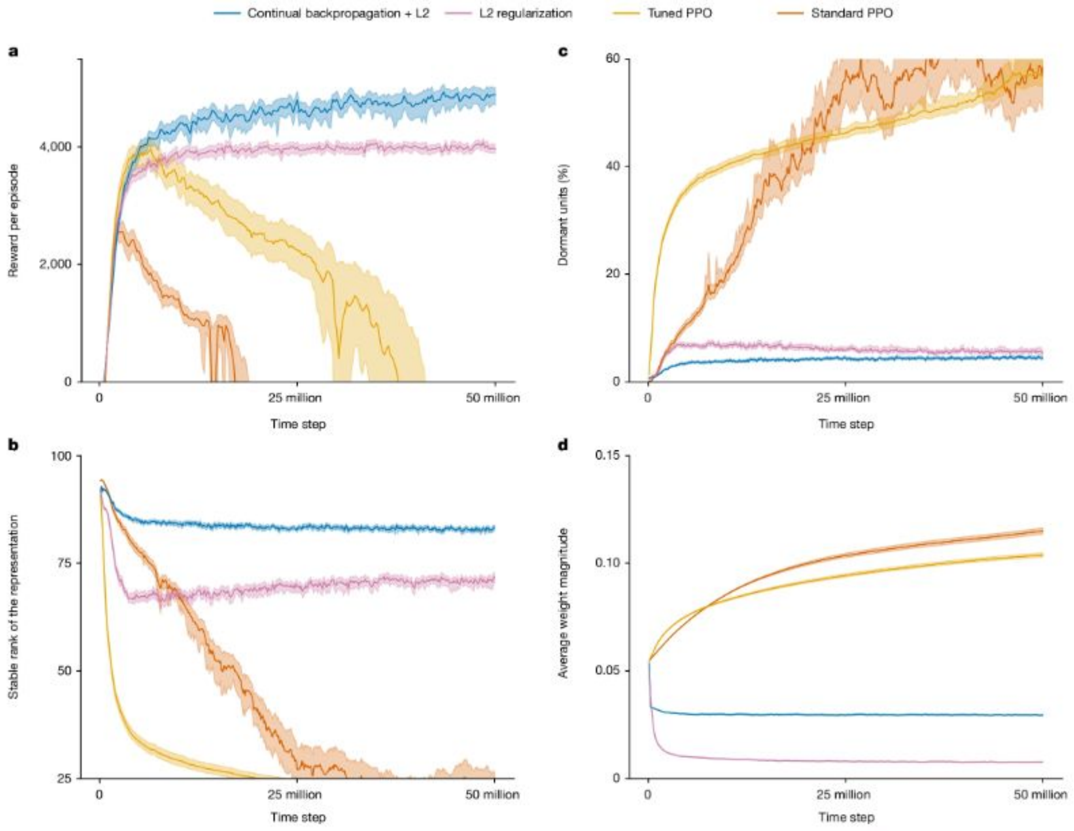
Reinforcement-Learning



a, The reinforcement-learning agent controls torques at the eight joints of the simulated ant (red circles) to maximize forward motion and minimize penalties. **b**, Here we use a version of the ant problem in which the friction on contact with the ground is abruptly changed every 2 million time steps. **c**, The standard PPO learning algorithm fails catastrophically on the non-stationary ant problem. If the optimizer of PPO (Adam) is tuned in a custom way, then the failure is less severe, but adding continual backpropagation or L2 regularization is necessary to perform well indefinitely. These results are averaged over 100 runs; the solid lines represent the mean and the shaded regions represent the 95% bootstrapped confidence interval.

Experiments + Results

Reinforcement-Learning



a, The four reinforcement-learning algorithms performed similarly on this and the non-stationary problem (compare with Fig. 3c). **b**,**c**, A closer look inside the networks reveals a similar pattern as in supervised learning (compare with Fig. 2c,d). **d**, The absolute values of the weights of the networks increased steadily under standard and tuned PPO, whereas they decreased and stayed small under L2 regularization with or without continual backpropagation. These results are averaged over 30 runs; the solid lines represent the mean and the shaded regions represent the 95% bootstrapped confidence interval.

Maintaining Plasticity

- Popular methods (Adam, Dropout, normalization) - **increased loss of plasticity**
- **L2 regularization** (weights stop becoming too large; moving them towards zero at each step) - **small weights** allow the **network to remain plastic**
- **Shrink and Perturb** is L2 regularization plus small random changes in weights at each step (injection of variability) - **remain plastic**
- **Continual Backpropagation**

Continual Backpropagation

- Initialization with small random weights before training
- Gradient descent at each training step

The initialization provides variability initially, but, with continued training, it tends to be lost, as well as plasticity along with it.

To maintain the variability, continual backpropagation, reinitializes a small number of units during training, typically fewer than one per step.

To prevent disruption of what the network has already learned, only the least-used units are considered for reinitialization.

Continual Backpropagation

Continual Backpropagation selectively reinitializes low-utility units in the network.

Utility measure (the contribution utility) is defined for each connection or weight and each unit.

Intuition behind: magnitude of the product of units' activation and outgoing weight - how valuable this connection is to its consumers.

If the contribution of a hidden unit to its consumer is small - hidden unit is not useful to its consumer

Contribution utility of a hidden unit as the sum of the utilities of all its outgoing connections.

In a feed-forward neural network, the contribution utility, ul[i], of the ith hidden unit in layer l at time t is updated as

$$\mathbf{u}_l[i] = \eta \times \mathbf{u}_l[i] + (1-\eta) \times |\mathbf{h}_{l,i,t}| \times \sum_{k=1}^{n_{l+1}} |\mathbf{w}_{l,i,k,t}|, \tag{1}$$

in which $\mathbf{h}_{l,i,t}$ is the output of the ith hidden unit in layer l at time t, $\mathbf{w}_{l,i,k,t}$ is the weight connecting the ith unit in layer l to the kth unit in layer $l+1$ at time t and n_{l+1} is the number of units in layer $l+1$.

Read More

- Article: Loss of plasticity in deep continual learning
- Github

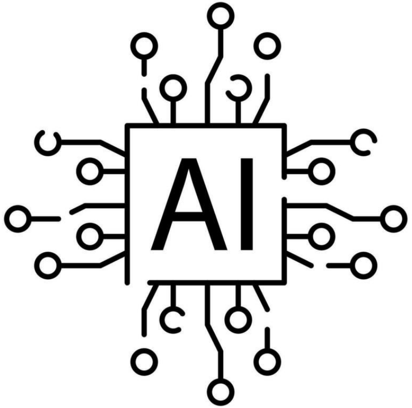
Conclusion

Goal: Deep Continual Learning

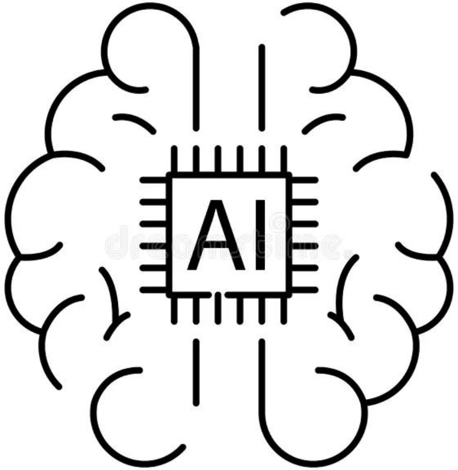
Problem: Loss of Plasticity

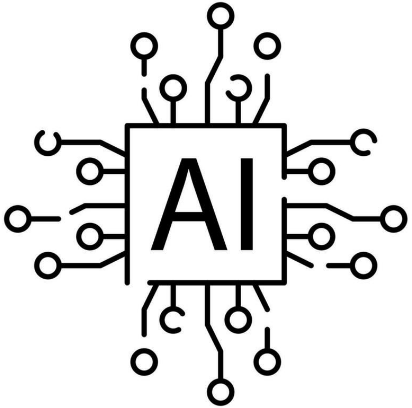
Possible Solution:

- L2 regularization
- Shrink and Perturb
- Continual Backpropagation



Thank you for attention!





Any questions?