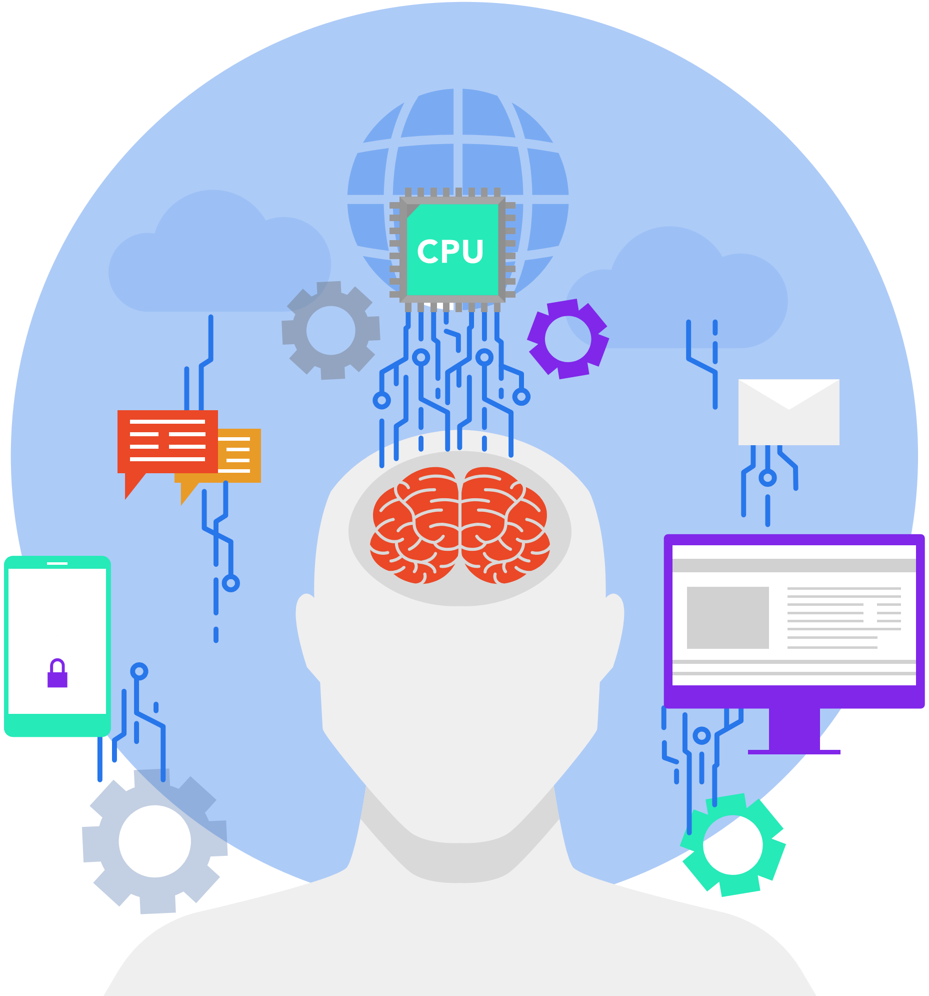
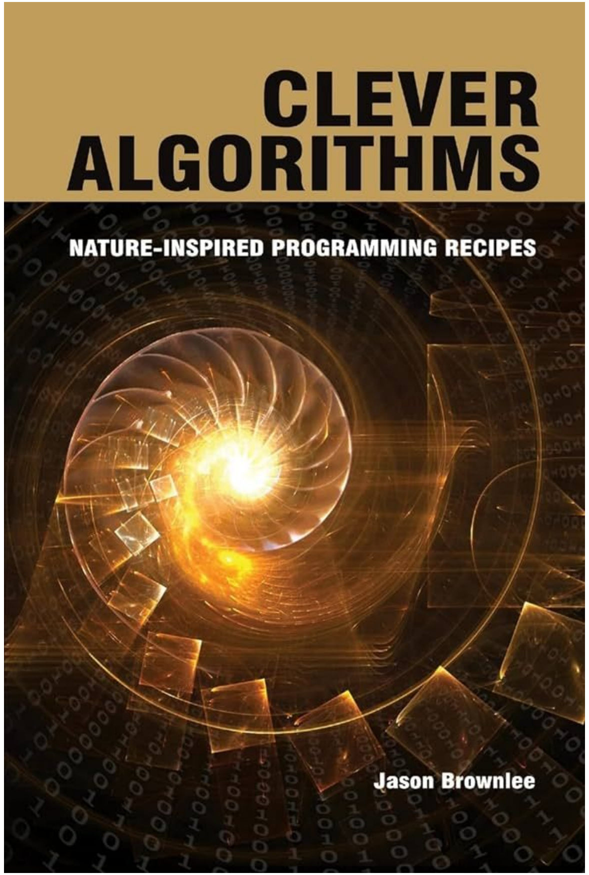
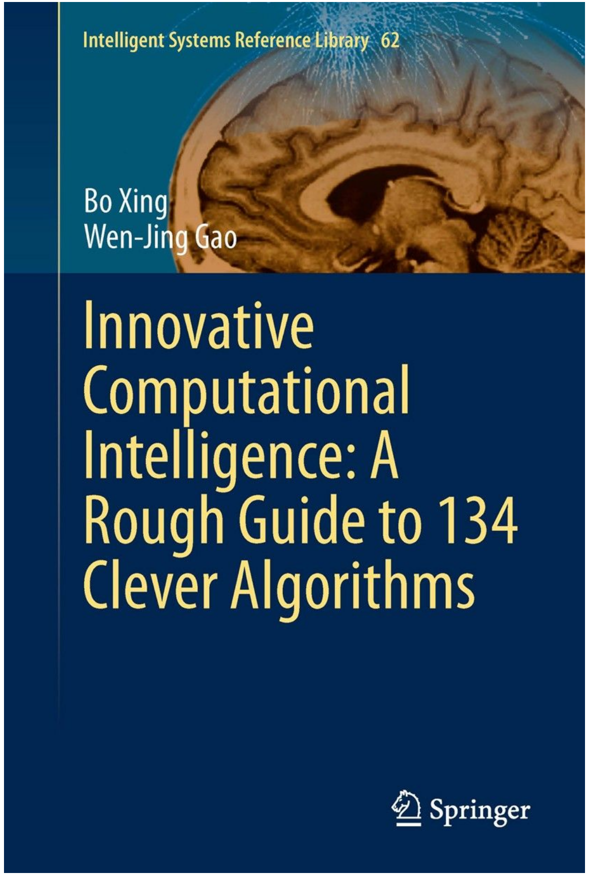
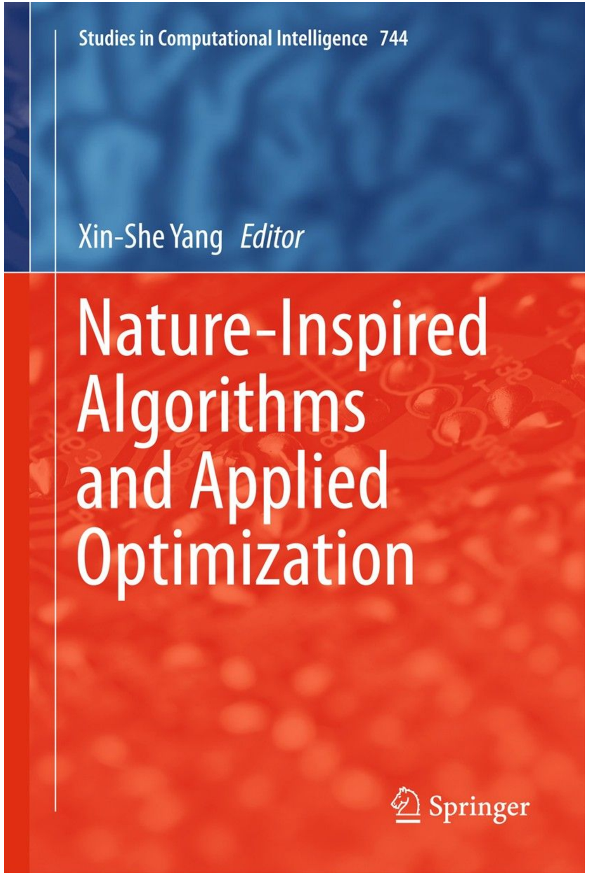


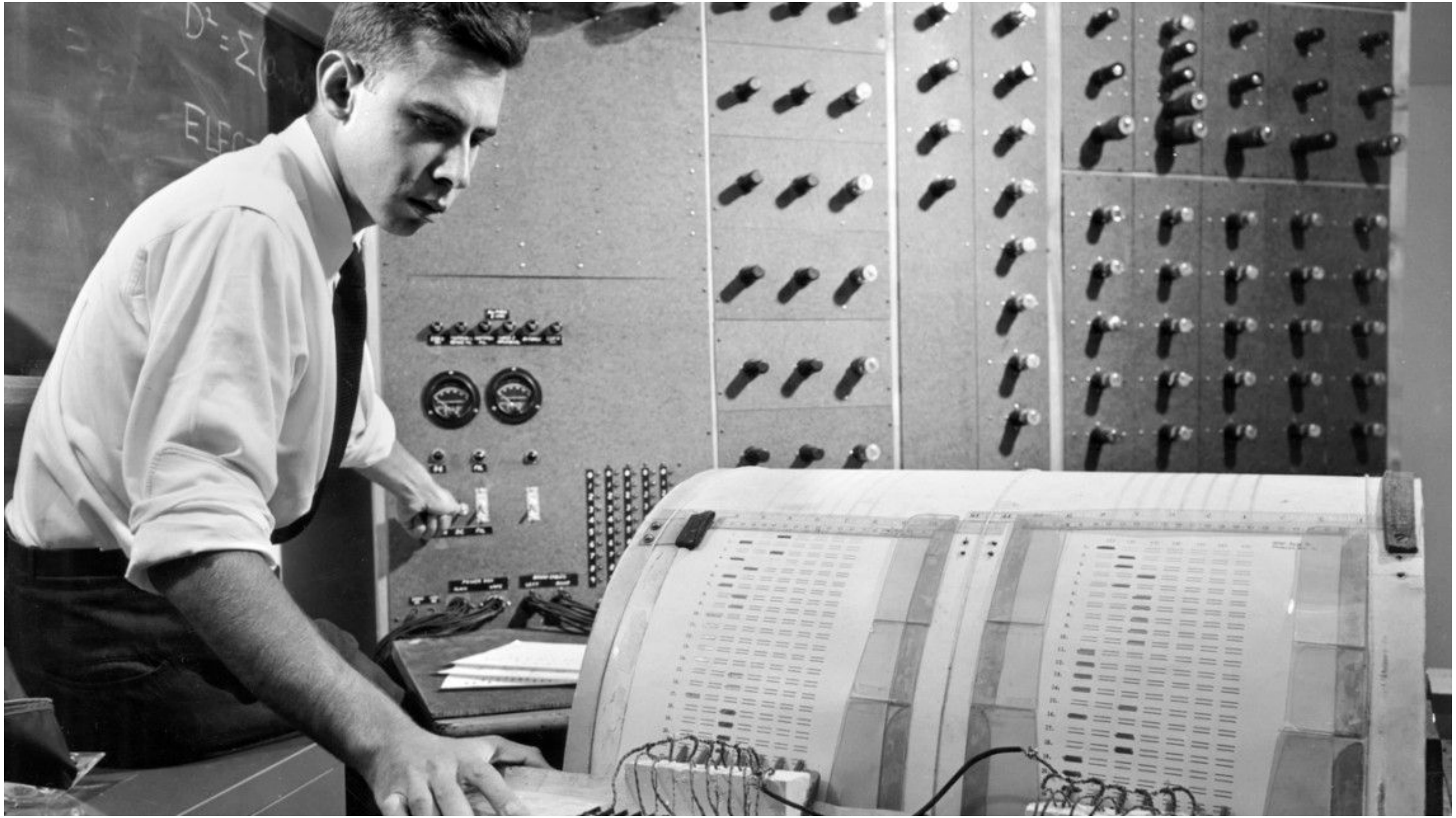

# Nature Inspired Deep Learning

Alexander Kovalenko

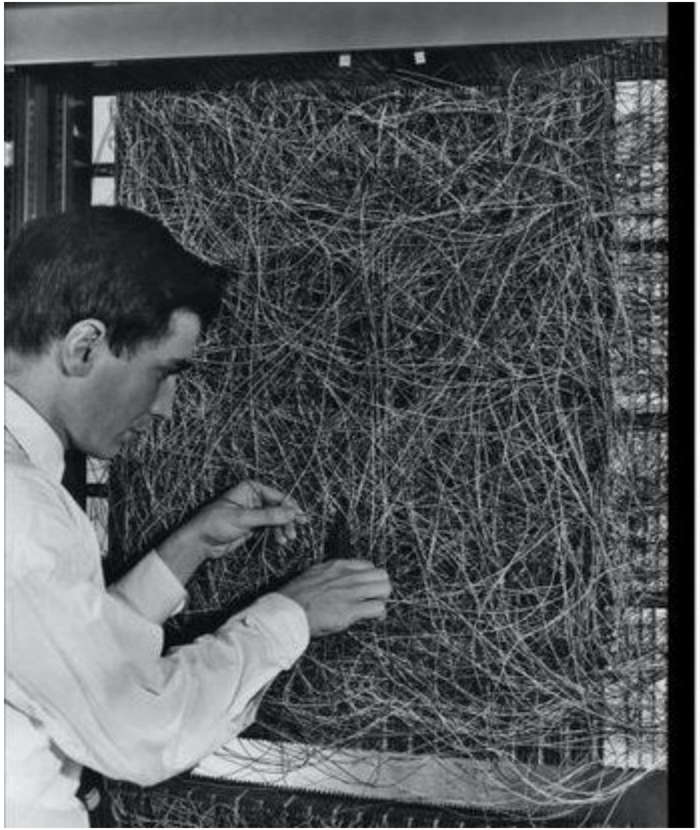

CLEVER ALGORITHMS
NATURE-INSPIRED PROGRAMMING RECIPES
Jason Brownlee


Intelligent Systems Reference Library 62
Bo Xing
Wen-Jing Gao
Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms
Springer


Studies in Computational Intelligence 744
Xin-She Yang Editor
Nature-Inspired Algorithms and Applied Optimization
Springer



# Frank Rosenblatt with his Mark I



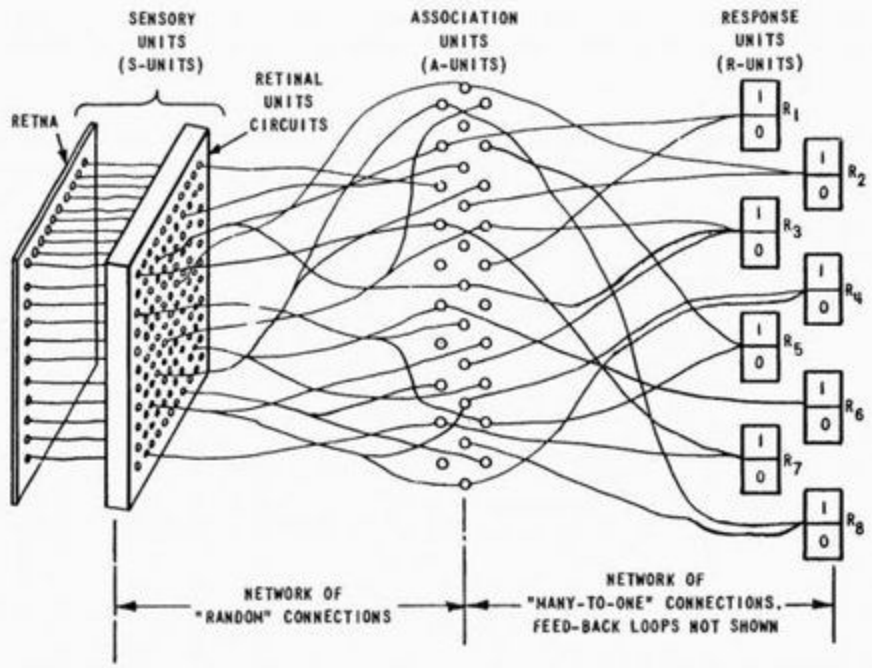
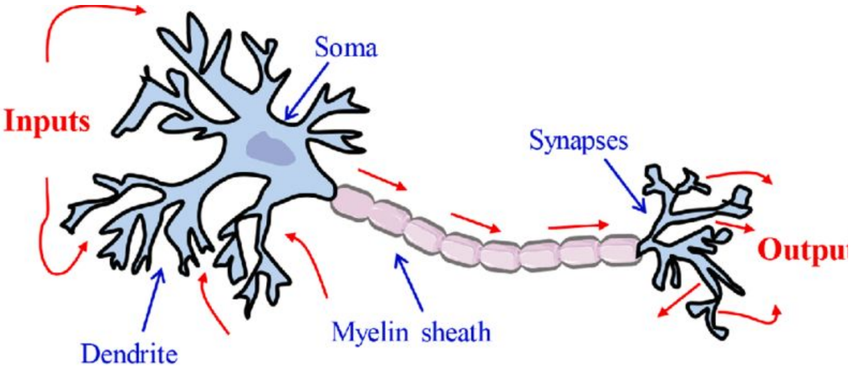


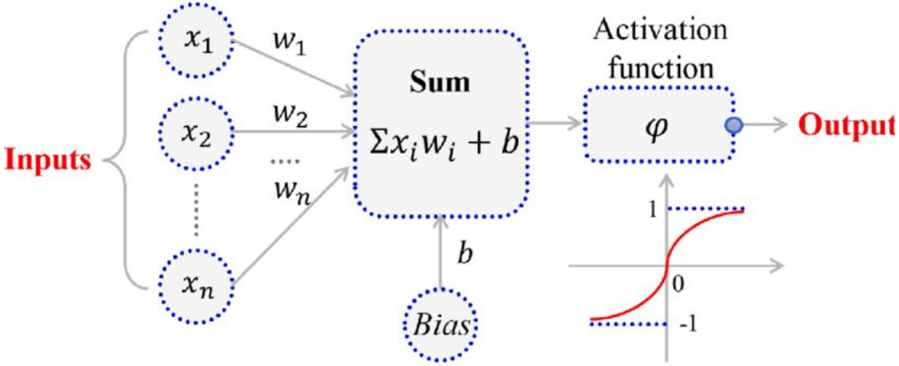

Figure I ORGANIZATION OF THE MARK I PERCEPTRON

# Biological vs. Artificial Neuron

Frank Rosenblatt




(a) Biological neuron




(b) Artificial neuron

Santiago Ramón y Cajal

https://sci-hub.se/10.1073/pnas.56.5.1423

# Attempt of Transfer Learning in Biology😱

*BEHAVIORAL ASSAY PROCEDURES FOR TRANSFER OF LEARNED BEHAVIOR BY BRAIN EXTRACTS,**

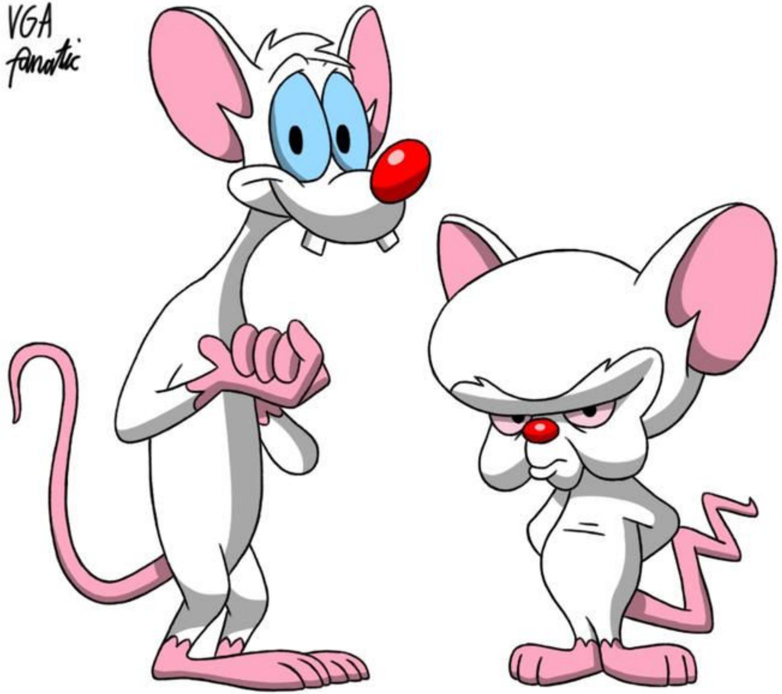
BY FRANK ROSENBLATT AND RODMAN G. MILLER

SECTION OF NEUROBIOLOGY AND BEHAVIOR, DIVISION OF BIOLOGICAL SCIENCES, CORNELL UNIVERSITY

*Communicated by F. A. Long, August 31, 1966*

In previous papers[1, 2] we have reviewed work by our own group and others suggesting that learned behavior may be transferred from trained to untrained rats by means of brain extracts. Subsequent experiments in other laboratories, however, many of which have recently been reviewed by Byrne *et al.*,[3] have revealed an unexpected degree of difficulty in producing the transfer effect reliably. In this and a succeeding paper we shall summarize a further series of ten experiments aimed chiefly at the development of a sensitive and reliable behavioral assay technique. While a certain amount of chemical experimentation has been done in the course of these studies, the improvement of chemical extraction procedures will be dealt with as a separate topic in subsequent reports.

In our previous experiments, we have relied on statistical control techniques to correct for the variability of activity levels of individual rats used as recipients of brain extract. While the conclusions based on these techniques still appear to be valid, it is clearly desirable to find a method which is less sensitive to individual and group biases which might be introduced by such factors as age, health, diet, toxicity of the extracts, cage conditions, or other variables which might influence the activity level of the animals. The present series of experiments includes a number of designs in which activity measures were employed as the criterion of transfer; after surveying our experience with these methods, however, it



# ANN vs. BNN

## Similarities:

**Basic Structure Inspiration:** Both types of neurons have a similar conceptual structure, consisting of inputs, a processing mechanism, and outputs.

- **Inputs:** In biological neurons, these are dendrites that receive signals. In artificial neurons, inputs are represented by data fed into the model.
- **Processing Center:** The soma (cell body) of a biological neuron and the summation function in an artificial neuron both serve to integrate incoming signals.
- **Output:** Axons in biological neurons transmit the signal to other neurons, while in artificial neurons, the output value is passed on to the next layer or as the final output of the network.

**Activation:** Both biological and artificial neurons use a form of activation function to decide whether to pass information further. In biological neurons, this is an all-or-none response (action potential), whereas artificial neurons typically use mathematical functions like sigmoid, ReLU, or tanh to decide the output.

**Adaptability:** Both types of neurons have mechanisms to adapt based on feedback. Biological neurons adapt through changes in synaptic strength, while artificial neurons adjust through weight updates during training (e.g., backpropagation*).

# ANN vs. BNN

## Differences:

### Operation Mechanism:

- Biological Neurons: They process and transmit information through electrochemical signals. Neurotransmitters and action potentials play crucial roles in the communication between neurons.
- Artificial Neurons: They operate using mathematical functions where the input values are weighted, summed, and then passed through a non-linear function to produce output.

### Complexity:

- Biological Neurons: They are incredibly complex, with capabilities for growth, self-repair, and conducting various biochemical processes within a single cell.
- Artificial Neurons: They are relatively simplistic, consisting of inputs, weights, biases, and a straightforward mathematical activation function.

# ANN vs. BNN

## Differences:

### Speed:

- Biological Neurons: They operate slower, with signal transmissions occurring in milliseconds.
- Artificial Neurons: They can process inputs and produce outputs almost instantaneously depending on the computational power available.

### Energy Efficiency:

- Biological Neurons: Highly energy-efficient, the human brain consumes about 20 Watts of power.
- Artificial Neurons: Less energy-efficient, modern computing systems, especially those running large-scale neural networks, can require substantial amounts of electrical power.

# ANN vs. BNN

## Differences:

### Learning and Plasticity:

- Biological Neurons: They exhibit a high degree of plasticity; synaptic connections can strengthen or weaken over time, influenced by factors like neurogenesis and other dynamic biological processes.
- Artificial Neurons: While learning mechanisms like weight updates are inspired by synaptic plasticity, artificial models do not inherently change structure or form new connections without predefined algorithms.

### Scalability:

- Biological Neurons: Naturally scalable, the human brain contains approximately 86 billion neurons forming trillions of synaptic connections.
- Artificial Neurons: Scalability depends on computational resources and technological advances in hardware. Increasing the number of artificial neurons and connections can lead to exponentially higher computational requirements.

# ANN vs. BNN

## Differences:

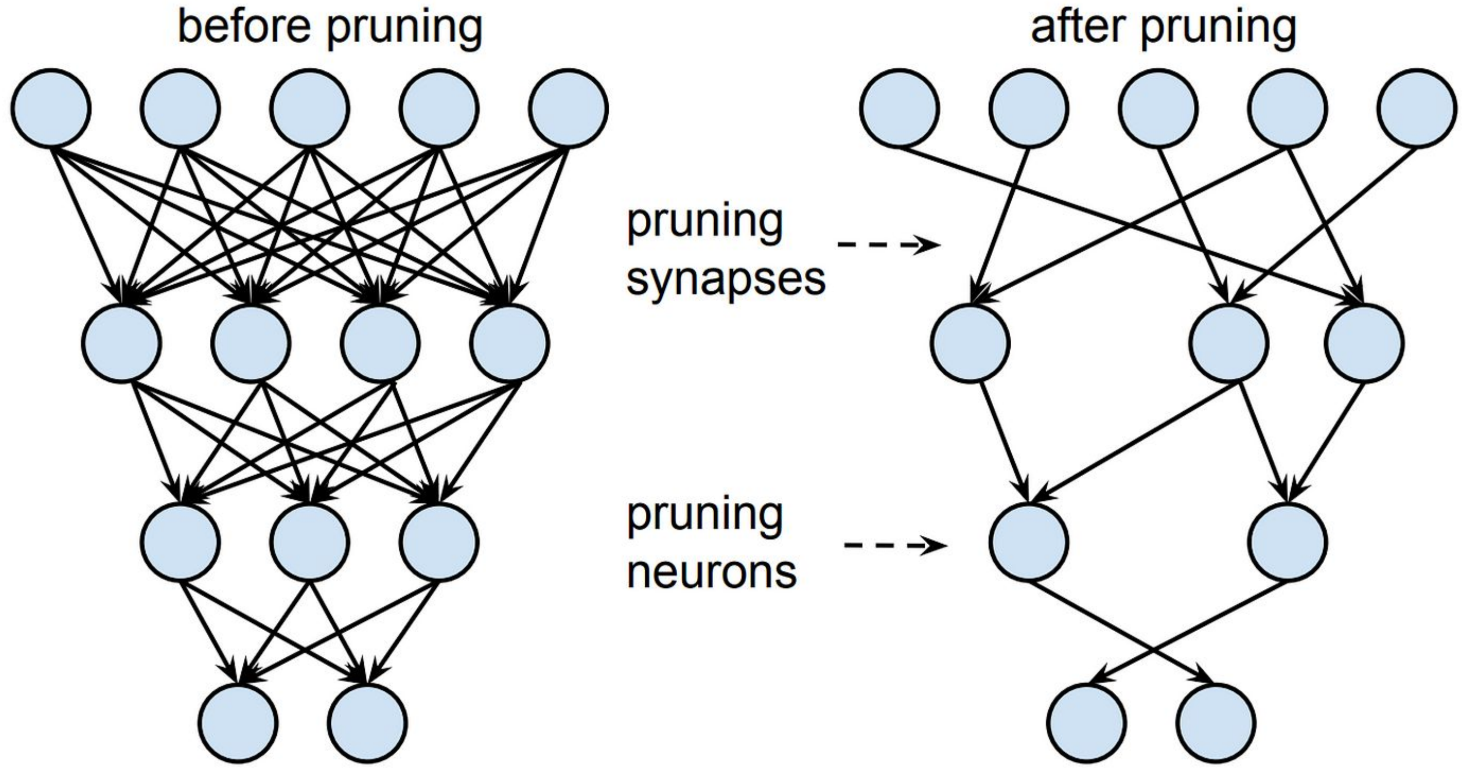
### Density:

- Biological Neurons: They sparsely connected. Out of approximately 86 billions of neurons each neuron has 1000 to 10000 connections. Rewiring of the connections is evident (doi: 10.1016/j.neubiorev.2018.03.001)
- Artificial Neurons: Mainly ANN layers are densely connected, rewiring is not a common thing.

# Why Shall We Adapt Nature Patterns in ANN?

- Because it is cool!
- The Nature did stuff much longer than us;
- ANNs are badly overparameterized;
- BNNs are much more energy efficient than ANNs;
- ANNs need much more examples to learn than ANNs;
- BNNs are ~~more~~ adaptable to new tasks;
- Unlike ANNs, BNNs can extrapolate (debatable);

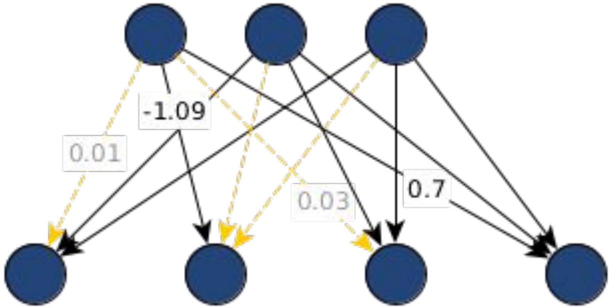
# Sparse Neural Networks



# Sparse Neural Networks

**Sparsifying strategies:**

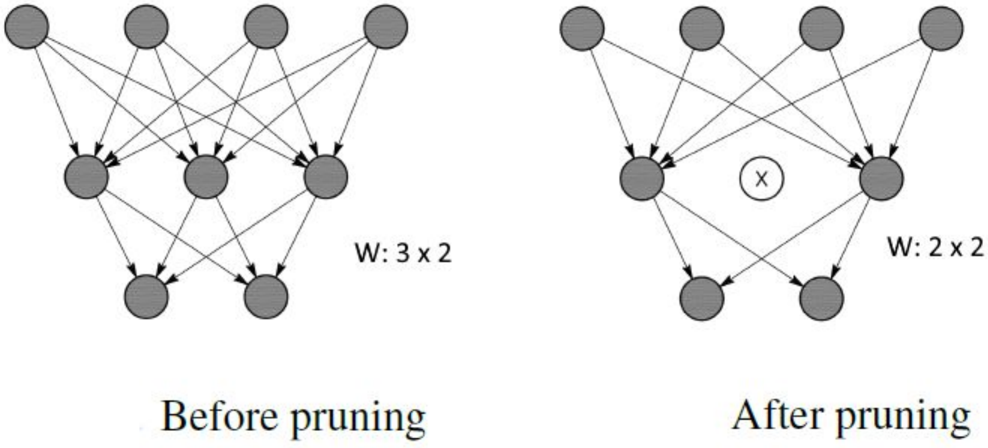
- Weight pruning



# Sparse Neural Networks

**Sparsifying strategies:**

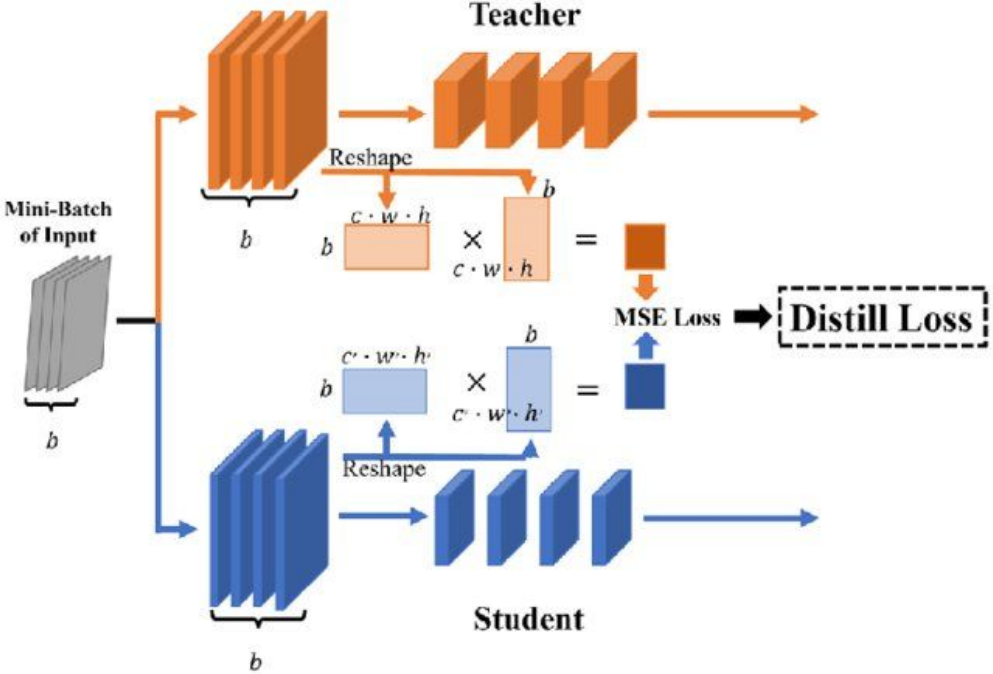
- Neuron Pruning



# Sparse Neural Networks

**Sparsifying strategies:**

- Pruning by Distillation

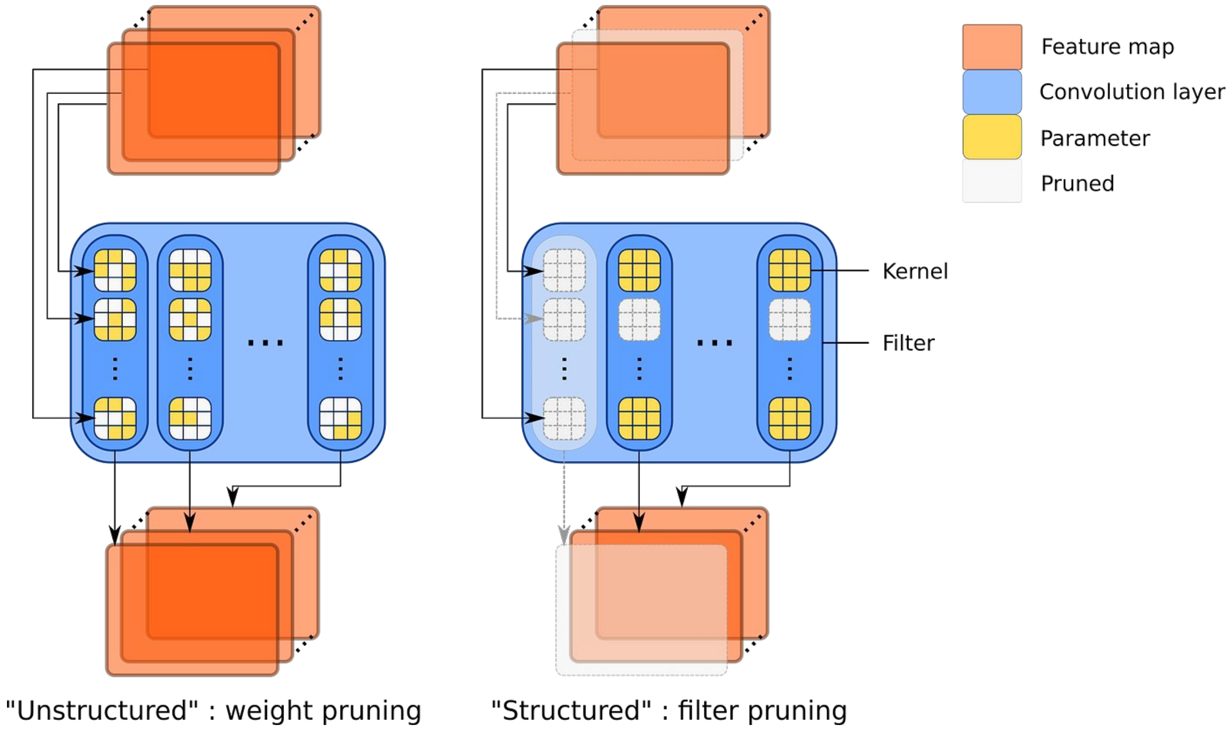


https://arxiv.org/abs/2303.00566

# Sparse Neural Networks

## Sparsifying strategies:

- Structured Pruning

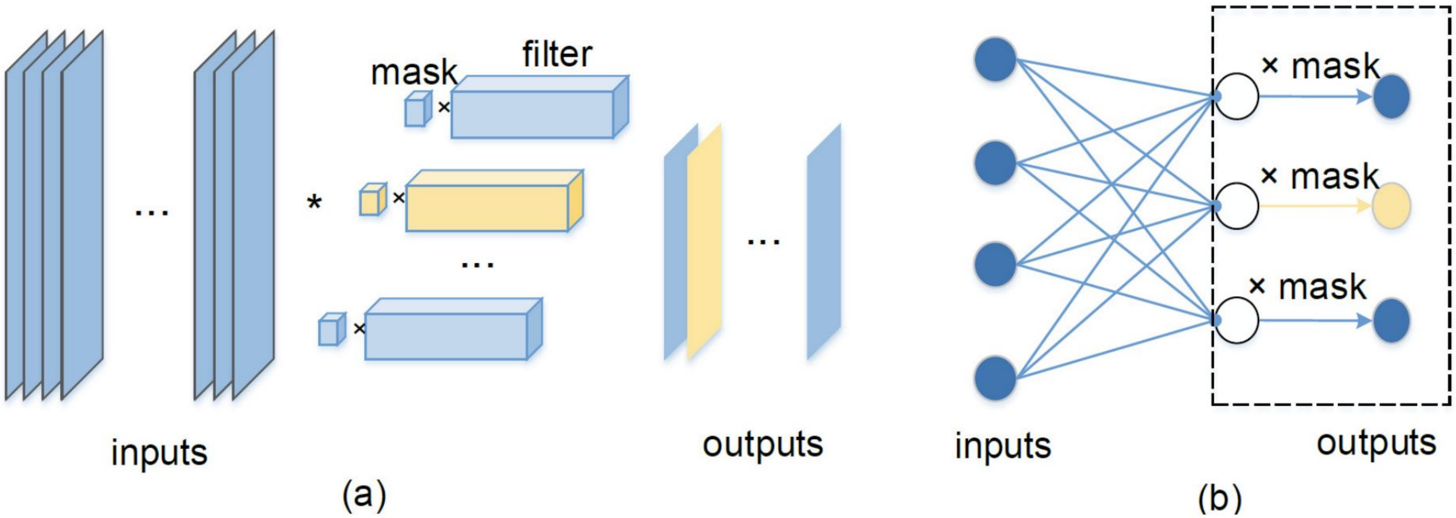


https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8787574

# Sparse Neural Networks

**Sparsifying strategies:**

- L1 regularization

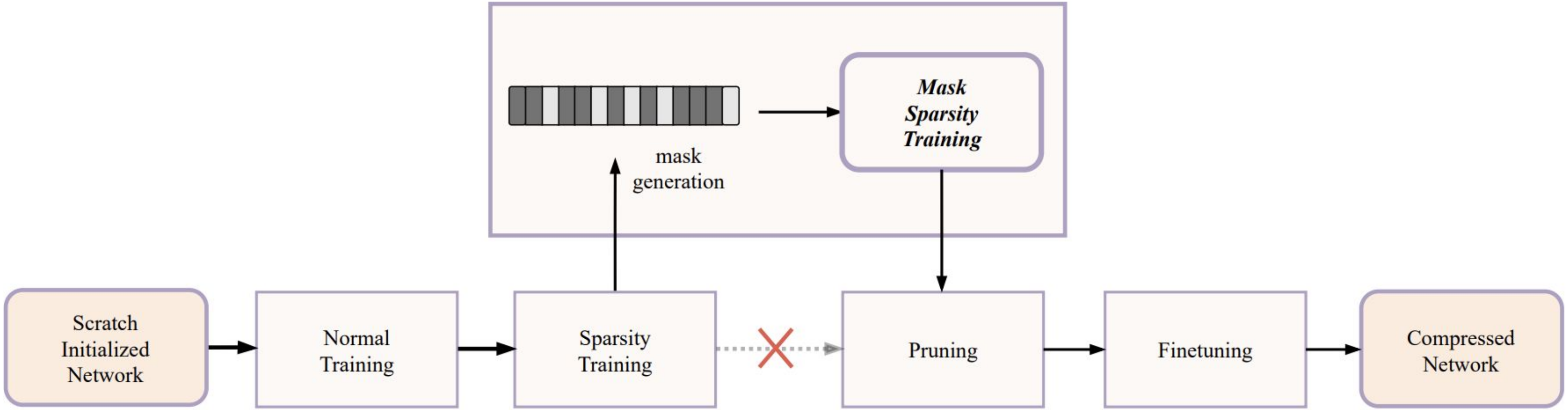


https://arxiv.org/abs/2201.06776

# Sparse Neural Networks

**Sparsifying strategies:**

- Sparse Regularizers

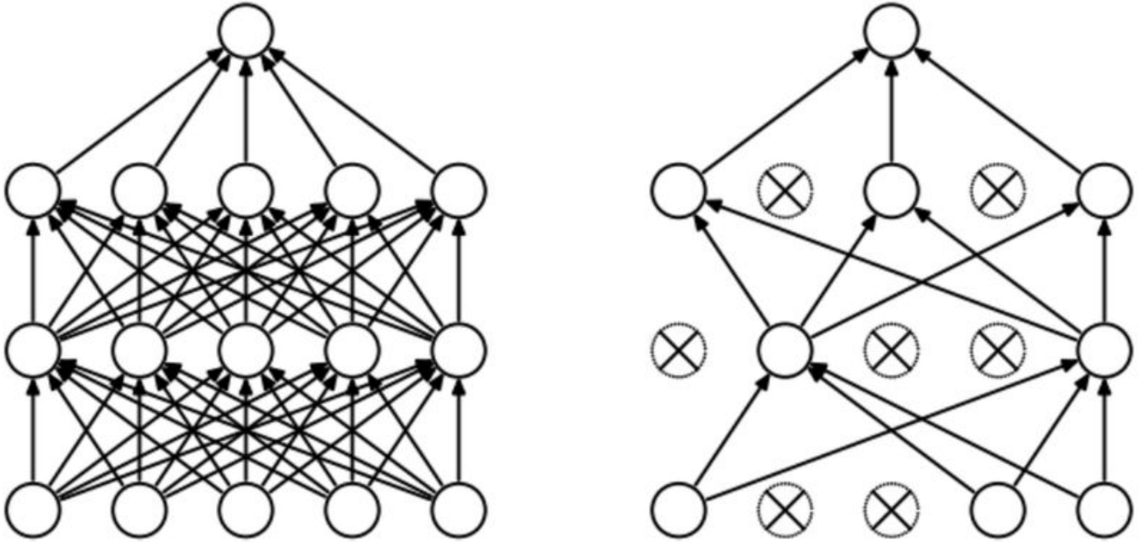


https://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf

# Sparse Neural Networks

**Sparsifying strategies:**

- Dropout(!)



(a) Standard Neural Net

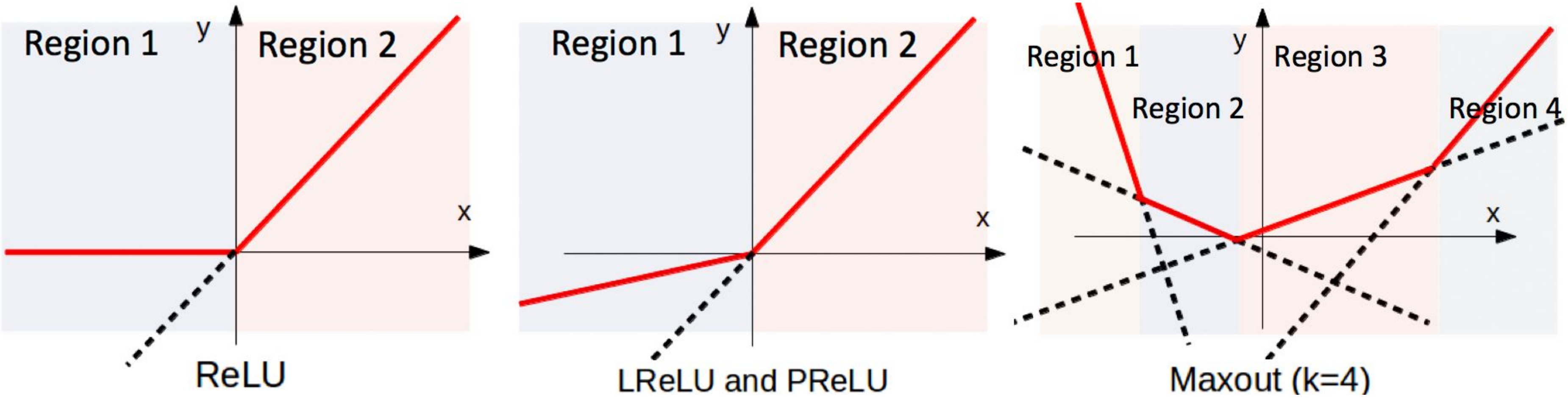
(b) After applying dropout.

https://arxiv.org/pdf/1302.4389v4.pdf

# Sparse Neural Networks

**Sparsifying strategies:**

- ReLU(!)
- Maxout Activation

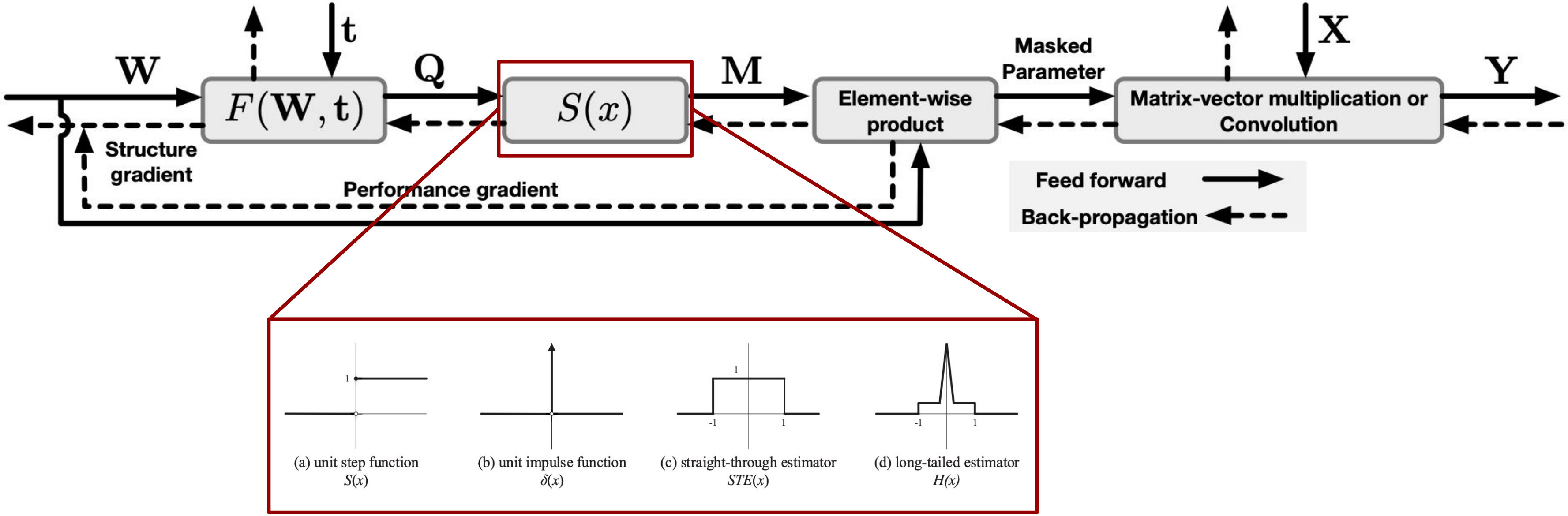


https://arxiv.org/pdf/2005.06870.pdf

# Sparse Neural Networks

## Sparsifying strategies:

- Dynamic Sparsity



https://arxiv.org/pdf/2005.06870.pdf

# Sparse Neural Networks

**Sparsifying strategies:**

- Dynamic Sparsity

| Architecture | Dense Baseline (%) | Model Remaining Percentage (%) | Sparse Accuracy (%) | Difference |
|---|---|---|---|---|
| Lenet-300-100 | $98.16 \pm 0.06$ | $2.48 \pm 0.21$ | $97.69 \pm 0.14$ | -0.47 |
| Lenet-5-Caffe | $99.18 \pm 0.05$ | $1.64 \pm 0.13$ | $99.11 \pm 0.07$ | -0.07 |
| LSTM-a | $98.64 \pm 0.12$ | $1.93 \pm 0.03$ | $98.70 \pm 0.06$ | **+0.06** |
| LSTM-b | $98.87 \pm 0.07$ | $0.98 \pm 0.04$ | $98.89 \pm 0.11$ | **+0.02** |

Table 1: The pruning results on MNIST for various architectures

https://arxiv.org/pdf/2005.06870.pdf

# Sparse Neural Networks

**Sparsifying strategies:**

- Dynamic Sparsity

| Architecture | Method | Dense baseline | Model Remaining Percentage (%) | Sparse Accuracy | Difference |
|---|---|---|---|---|---|
| VGG-16 | Sparse Momentum | $93.51 \pm 0.05$ | 10 | $93.36 \pm 0.04$ | -0.15 |
| | | | 5 | $93.00 \pm 0.07$ | -0.51 |
| | DST (Ours) | $93.75 \pm 0.21$ | $8.82 \pm 0.34$ | **93.93** $\pm 0.05$ | **+0.18** |
| | | | $3.76 \pm 0.53$ | **93.02** $\pm 0.37$ | -0.73 |
| WideResNet-16-8 | Sparse Momentum | $95.43 \pm 0.02$ | 10 | $94.87 \pm 0.04$ | -0.56 |
| | | | 5 | $94.38 \pm 0.05$ | -1.05 |
| | DSR | $95.21 \pm 0.05$ | 10 | $94.93 \pm 0.04$ | -0.28 |
| | | | 5 | $94.68 \pm 0.05$ | -0.53 |
| | DST (Ours) | $95.18 \pm 0.06$ | $9.86 \pm 0.22$ | **95.05** $\pm 0.08$ | **-0.13** |
| | | | $4.64 \pm 0.15$ | **94.73** $\pm 0.11$ | **-0.45** |

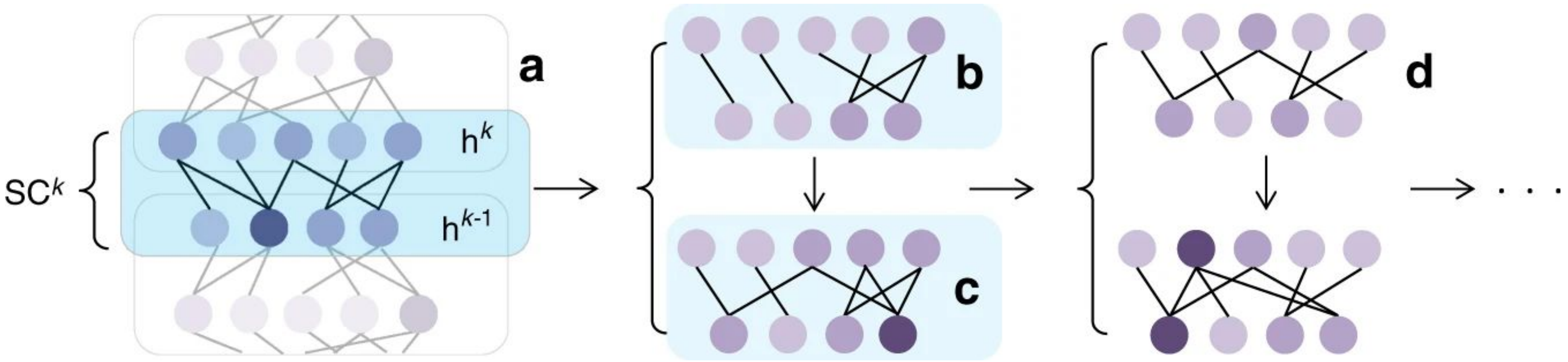
Table 2: Comparison with other sparse training methods on CIFAR-10.

https://www.nature.com/articles/s41467-018-04316-3

# Sparse Neural Networks

**Sparsifying strategies:**

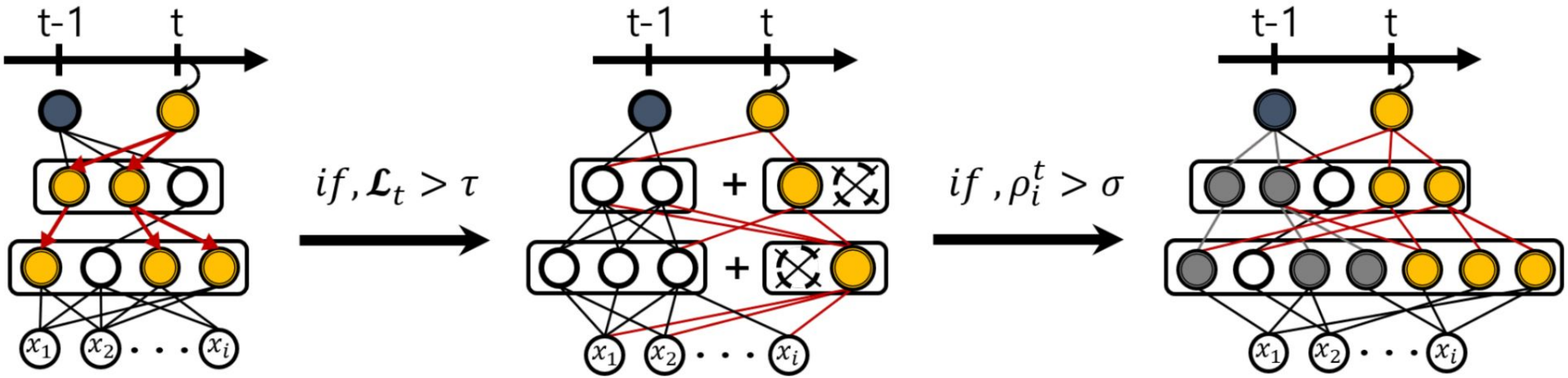
- Sparse Evolutionary Training



https://arxiv.org/pdf/1708.01547.pdf

# Dynamically Expandable Neural Networks

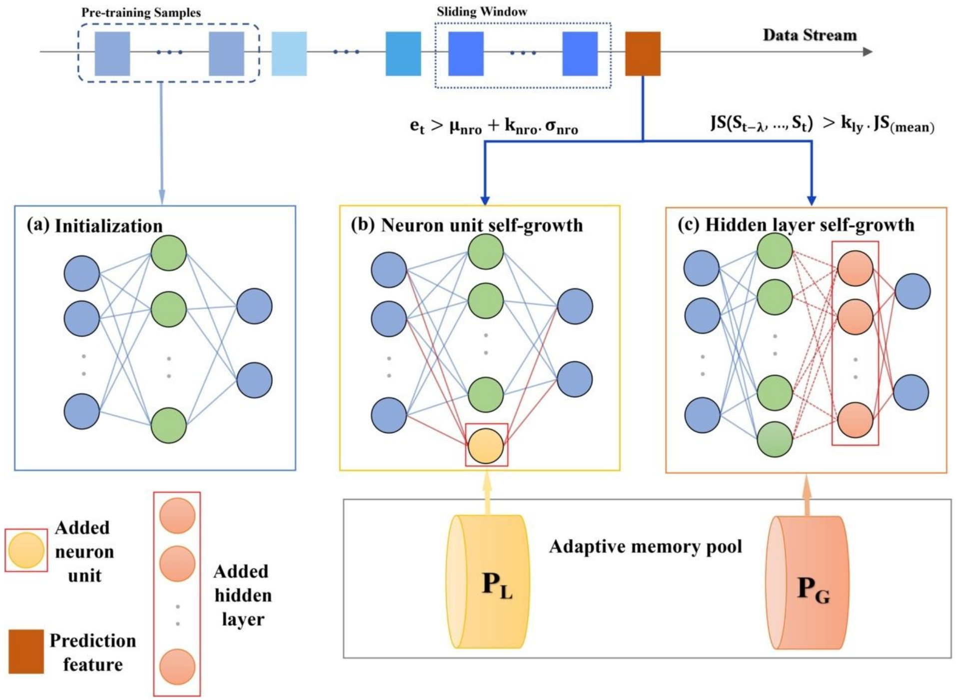
Lifelong Learning with Dynamically Expandable Networks



https://www.sciencedirect.com/science/article/pii/S0020025523009969

# Dynamically Expandable Neural Networks

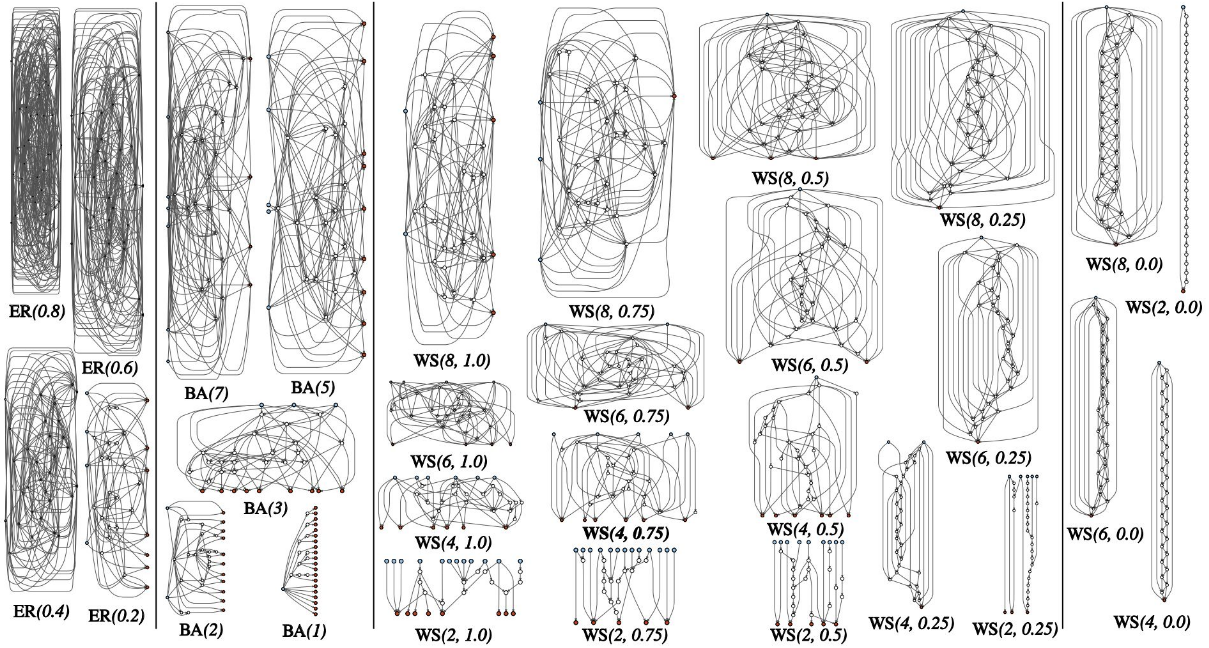
Dynamically evolving deep neural networks with continuous online learning



https://arxiv.org/pdf/1904.01569.pdf

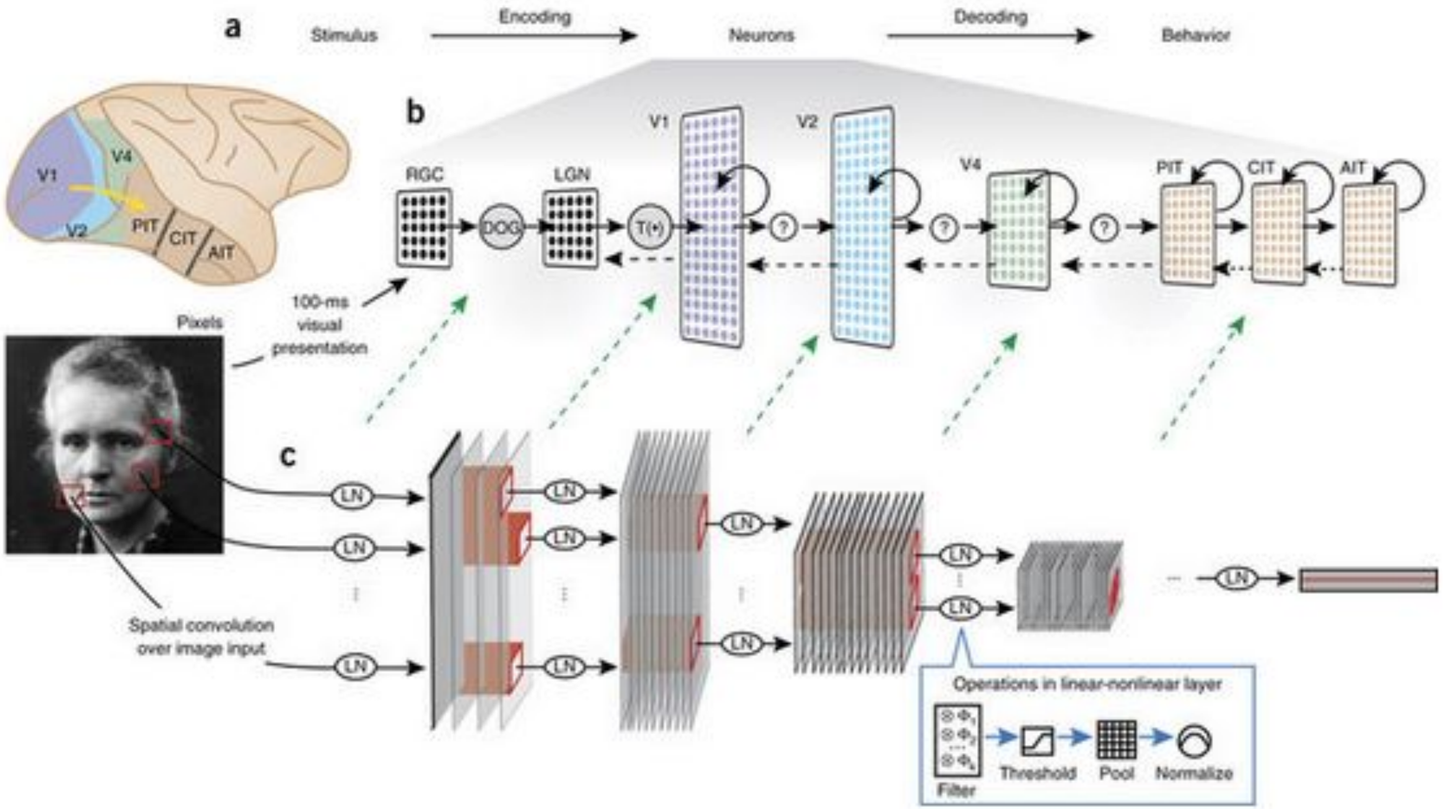
# Rewirable Neural Networks

Exploring Randomly Wired Neural Networks for Image Recognition



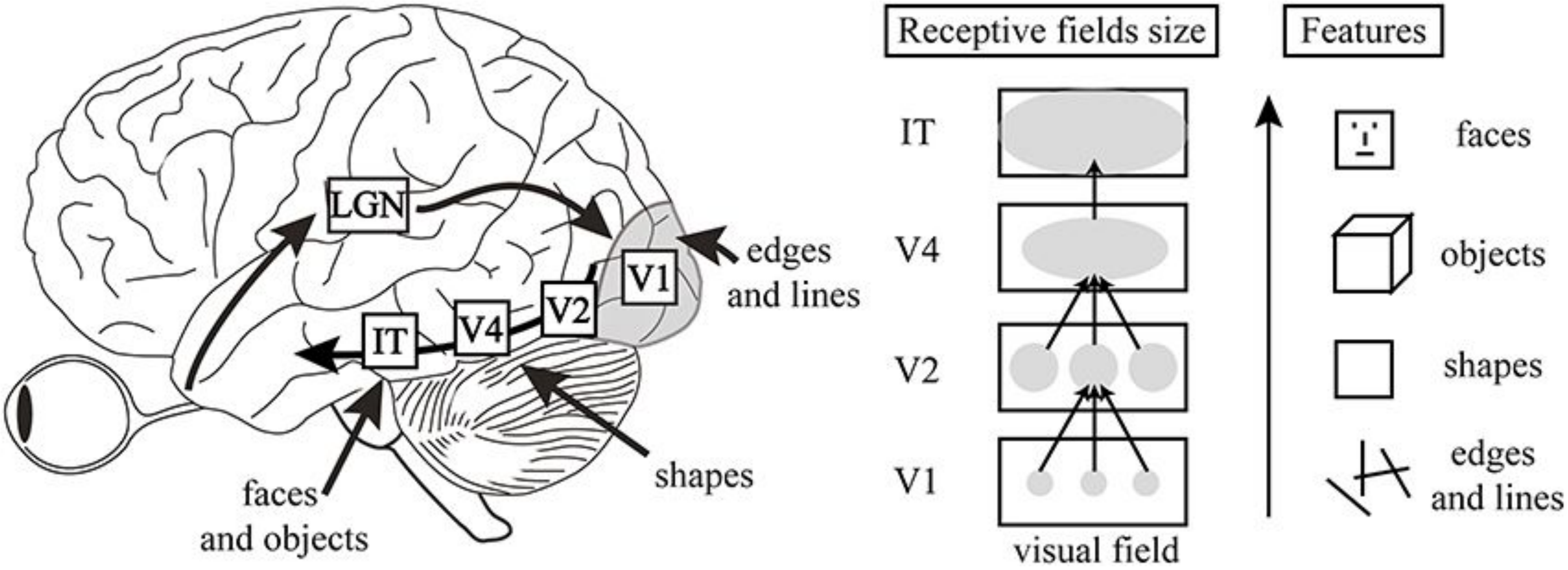
https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

# Convolutional Neural Networks



https://www.frontiersin.org/articles/10.3389/fncom.2014.00135/full

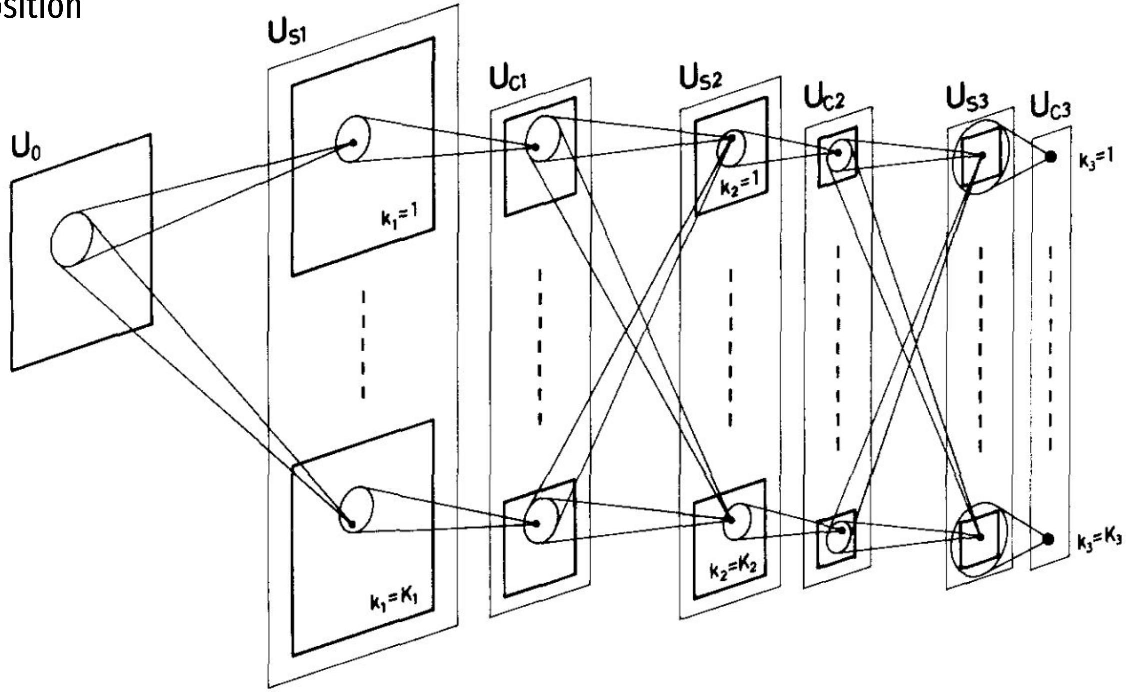
# Convolutional Neural Networks



https://www.rctn.org/bruno/public/papers/Fukushima1980.pdf

# Convolutional Neural Networks

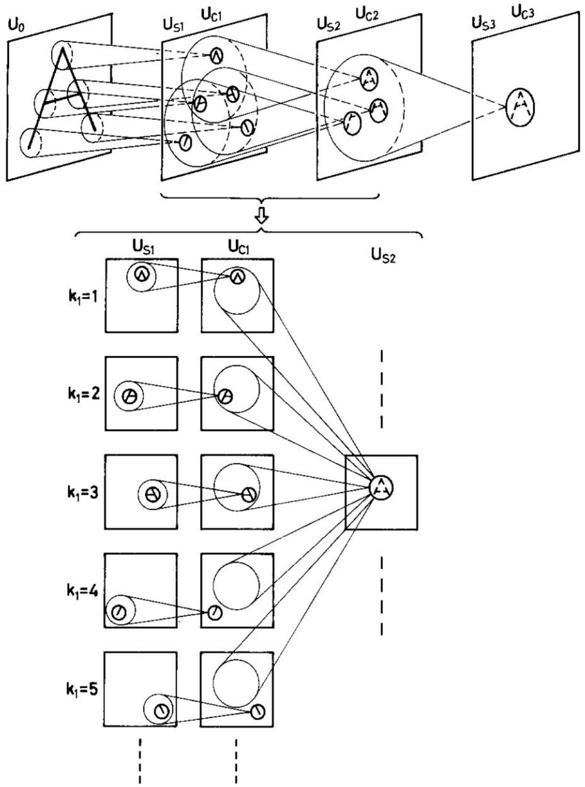
Neocognitron: A Self-organizing Neural Network Model
for a Mechanism of Pattern Recognition
Unaffected by Shift in Position



https://www.rctn.org/bruno/public/papers/Fukushima1980.pdf

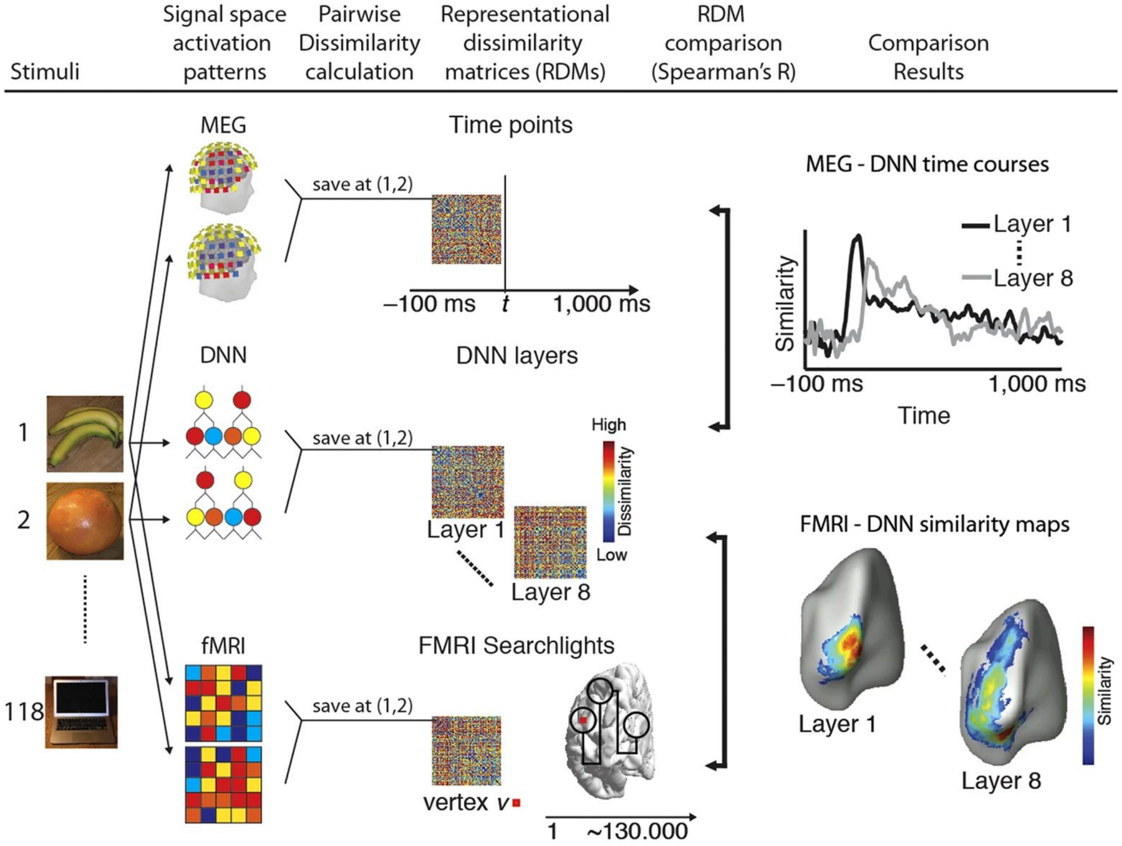
# Convolutional Neural Networks

Neocognitron: A Self-organizing Neural Network Model
for a Mechanism of Pattern Recognition
Unaffected by Shift in Position

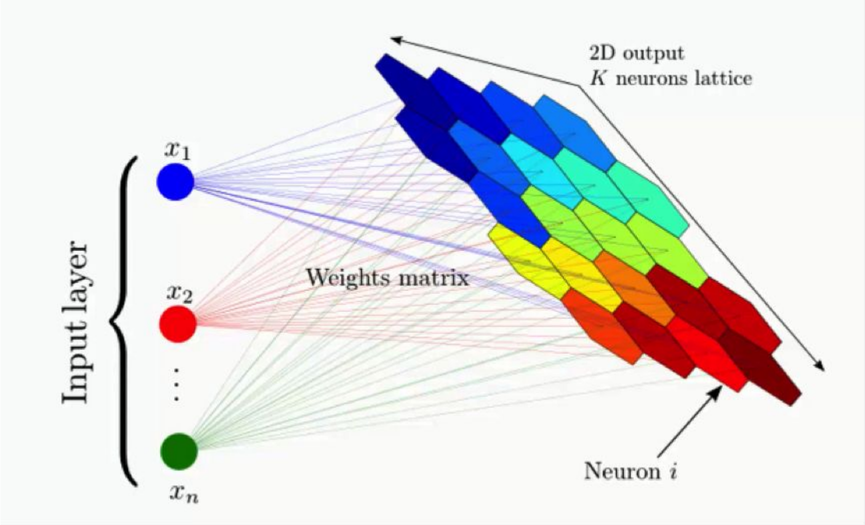


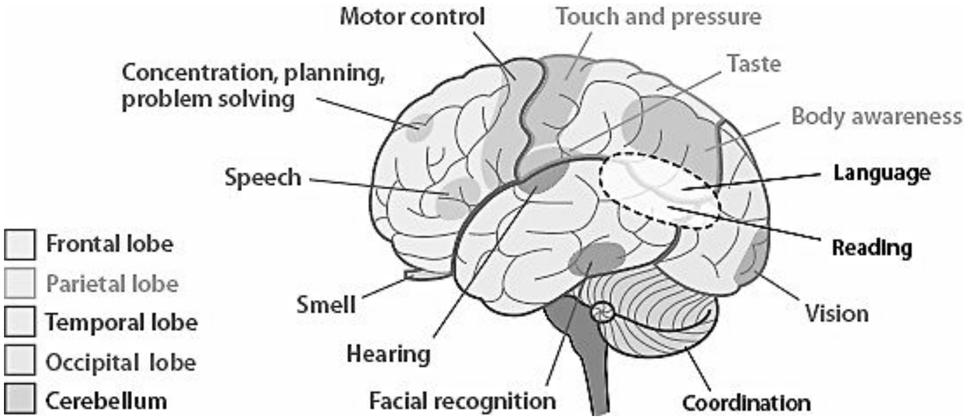
https://www.nature.com/articles/srep27755

# Convolutional Neural Networks

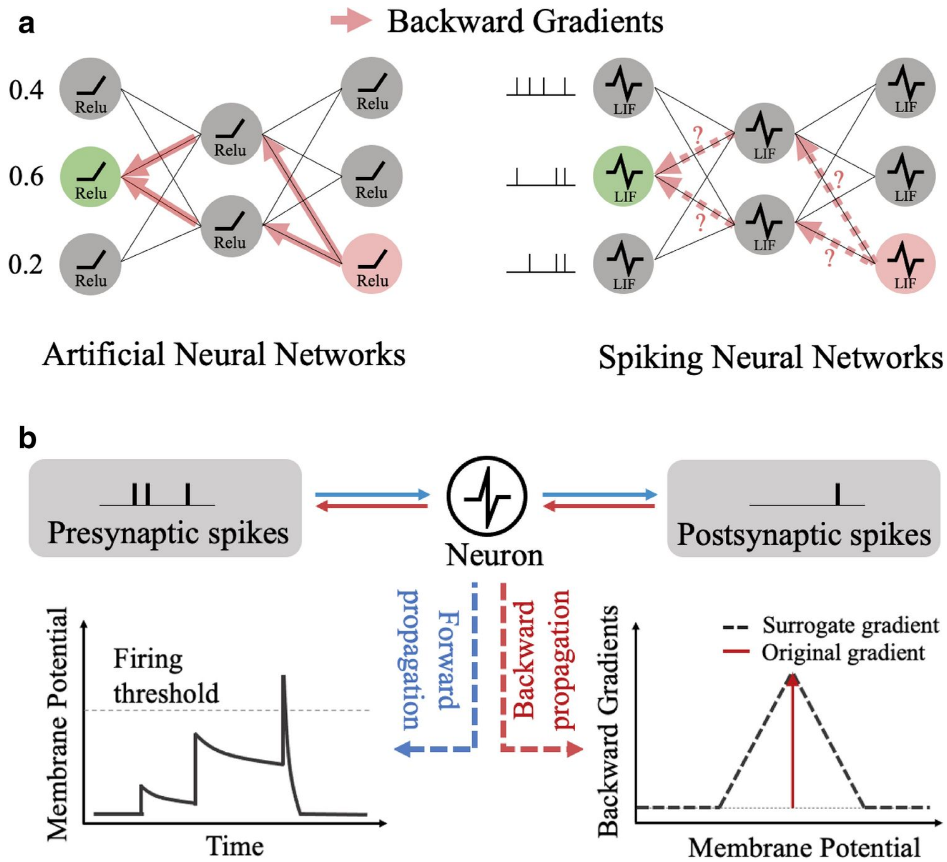


# Self-Organizing Maps



# Spiking Neural Networks



# Spiking Neural Networks

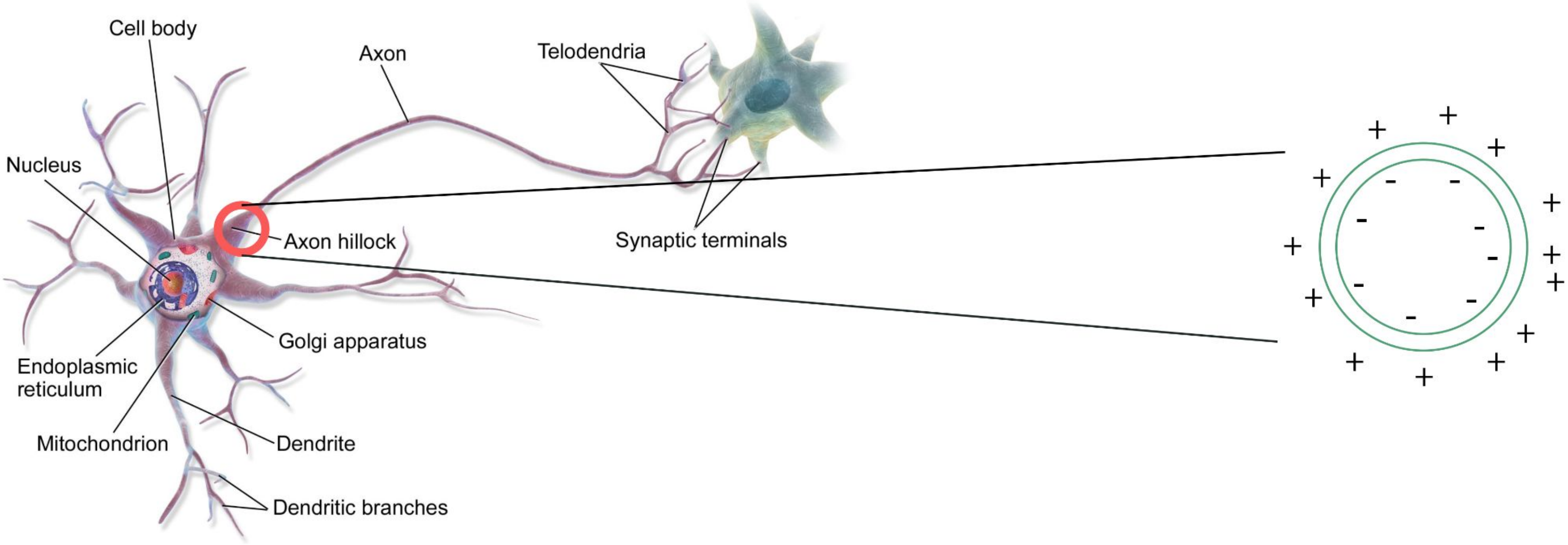
**Possible computational advantages of SNNs:**

- Fast decision making
- Robust to noise
- Robust to adversarial attacks
- More generalisable
- Low power (neuromorphic hardware)

**SNN may help to answer some of biological questions:**

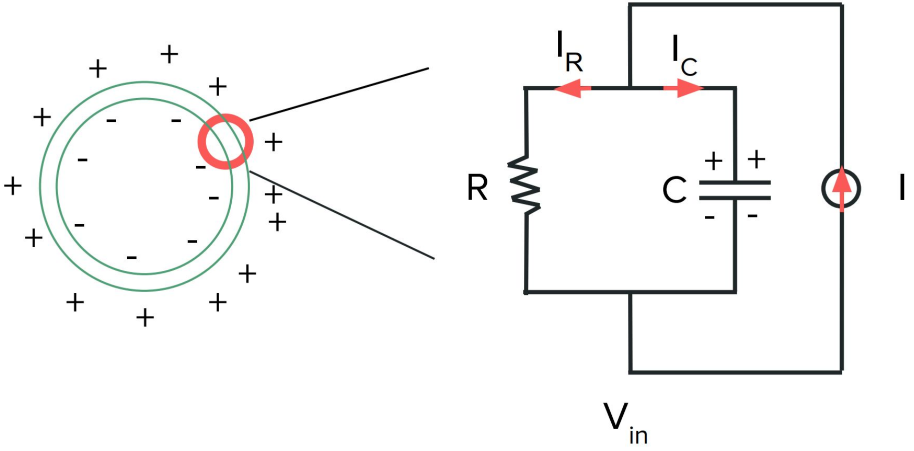
- What is the role of spikes (efficiency)?
- Local learning rules?
- Interaction with synapse/neuron dynamics?

# Spiking Neural Networks



# Spiking Neural Networks

$V = V_{out} - V_{in}$




$I_R + I_C = I$

$I_R = V/R$

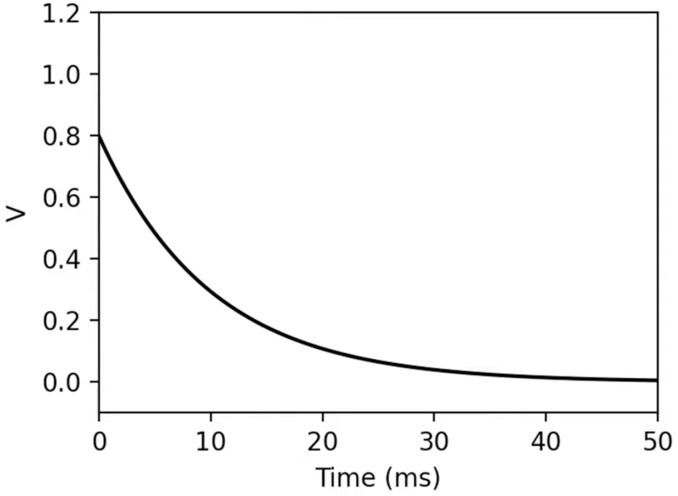
$I_C = C(dV/dt)$

$V/R + C(dV/dt) = I$

$$\boxed{dV/dt = -V/RC + I/C}$$

# Spiking Neural Networks

$$\tau \frac{\mathrm{d}V}{\mathrm{d}t} = -V$$

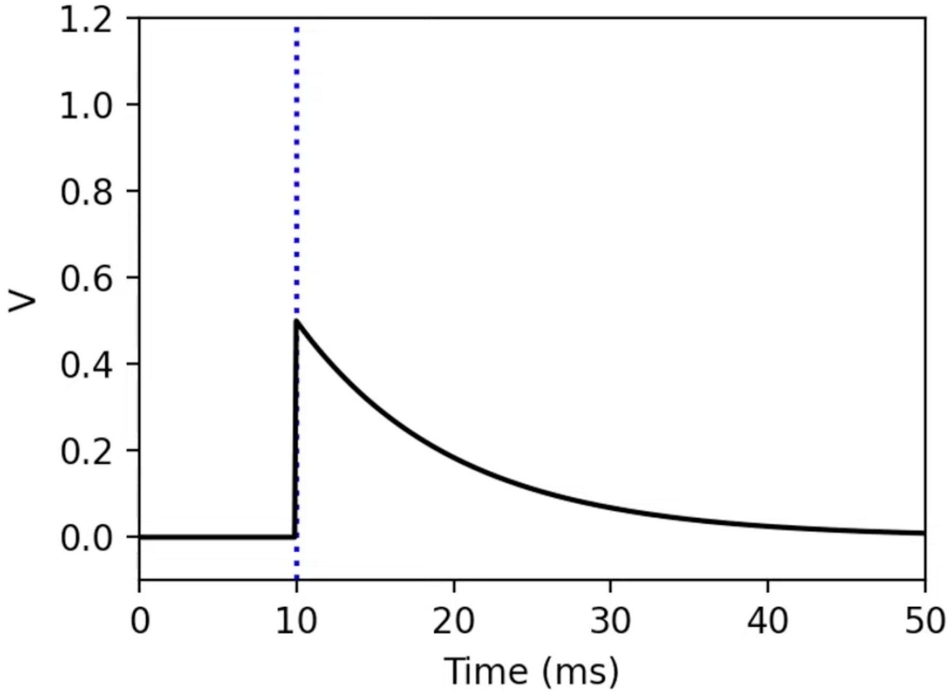


# Spiking Neural Networks

$$\tau \frac{\mathrm{d}V}{\mathrm{d}t} = -V$$

When a neuron receives a spike, V increases by synaptic weight w:

$$V \leftarrow V + w$$



# Spiking Neural Networks

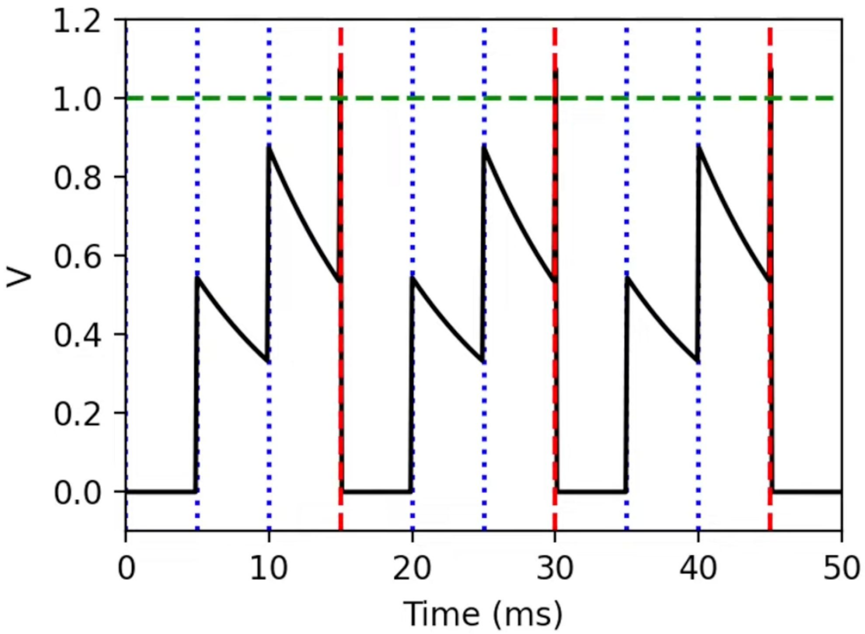
$$\tau \frac{\mathrm{d}V}{\mathrm{d}t} = -V$$

When a neuron receives a spike, V increases by synaptic weight w:

$$V \leftarrow V + w$$

When $V > V_t$ the neuron "fires a spike" and resets:

$$V \leftarrow 0$$



# How to Train SNNs?

- Spike-Timing-Dependent Plasticity (STDP) (doi: 10.1016/j.neuron.2012.08.001)
- Backpropagation Through Time for Spikes (BPTT-S) (doi: 10.3389/fnins.2023.1047008)
- Conversion from ANNs to SNNs (https://arxiv.org/abs/2205.10121)
- Reinforcement Learning (RL) (https://arxiv.org/abs/2005.05941)
- Unsupervised and Local Learning (https://arxiv.org/abs/2207.02727)
- Surrogate Gradient Learning (https://arxiv.org/pdf/1901.09948.pdf)

# How to Train SNNs?

- Spike-Timing-Dependent Plasticity (STDP) (doi: 10.1016/j.neuron.2012.08.001)
- Backpropagation Through Time for Spikes (BPTT-S) (doi: 10.3389/fnins.2023.1047008)
- Conversion from ANNs to SNNs (https://arxiv.org/abs/2205.10121)
- Reinforcement Learning (RL) (https://arxiv.org/abs/2005.05941)
- Unsupervised and Local Learning (https://arxiv.org/abs/2207.02727)
- **Surrogate Gradient Learning (https://arxiv.org/pdf/1901.09948.pdf)**