Backpropagation Neural Tree

Dr Varun Ojha

Department of Computer Science
University of Reading, UK
v.k.ojha@reading.ac.uk; vkojha@ieee.org
Github: https://github.com/vojha-code

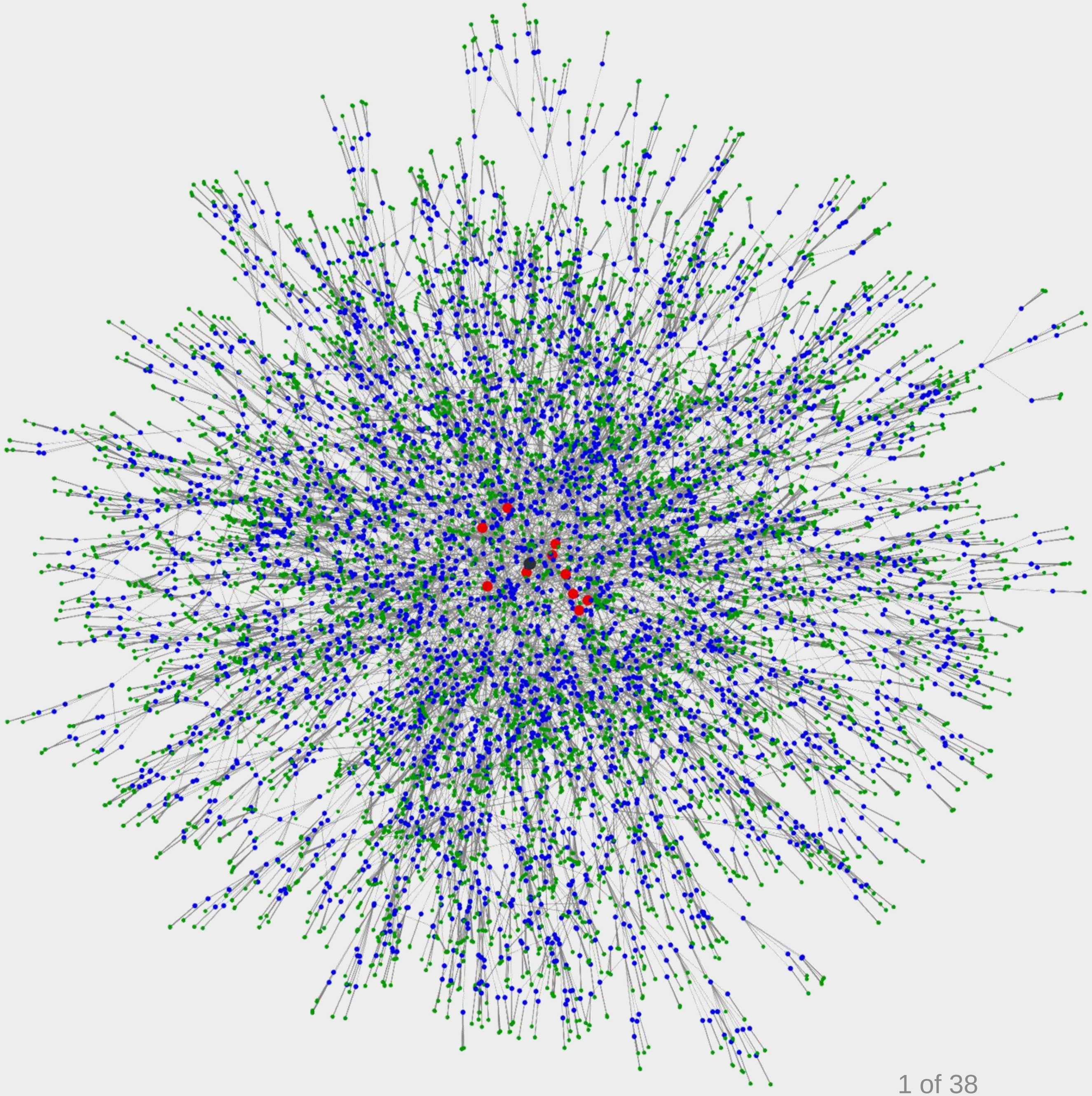
at

LOD2022

The 8th International Conference on
Machine Learning, Optimization, and Data Science

September 18 – 22, 2022

Siena – Tuscany, Italy



Intrinsic Intelligence of a child’s mind

Video Source:
https://www.youtube.com/watch?v=dEnDjyWHN4A
(Accessed on 21 Feb 2021)

Learning

Video Source:
https://www.youtube.com/watch?v=Ak7bPuR2rDw
(Accessed on 16 September 2022)

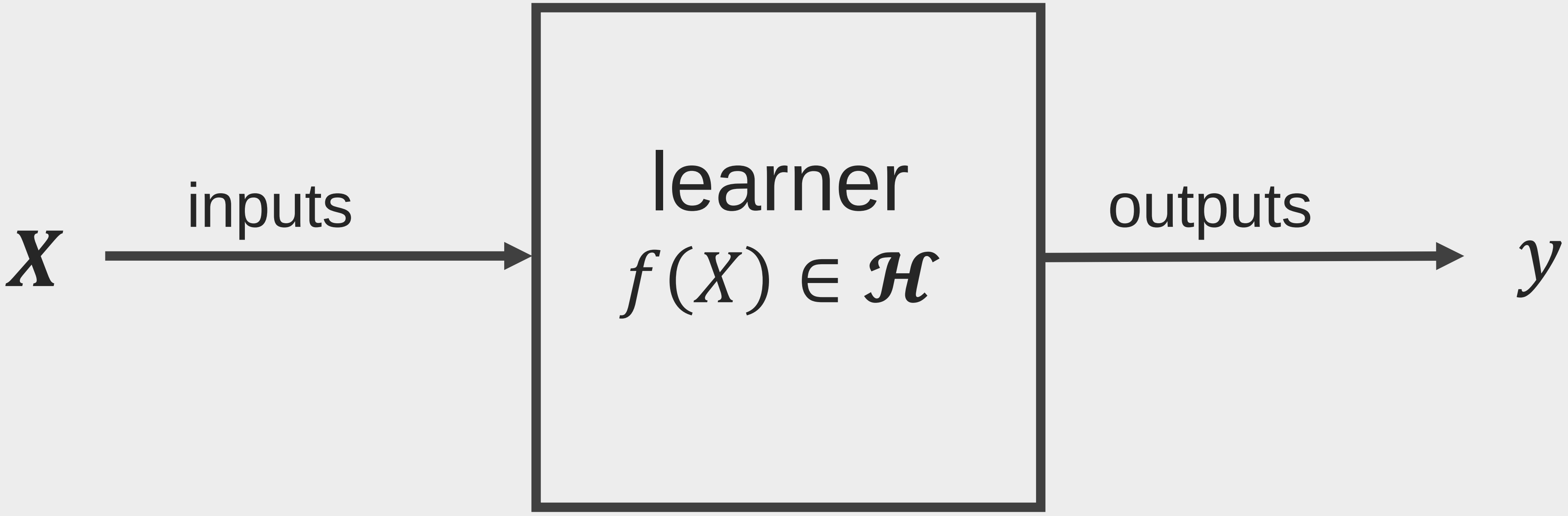
Content

- **Part 1: Supervised learning basics**
  - Learning process
  - Biological inspirations
- **Part 2: Neural Architectures**
  - Neural Networks
  - Neural Trees and Neural Computation
  - Neural Architecture Search
- **Part 3: Backpropagation Neural Tree**
  - Forward and Backward Pass Computation
  - Performance on regression and classification tasks
  - Solving a Image classification problem
- **Resources**

Part 1
Supervised Learning

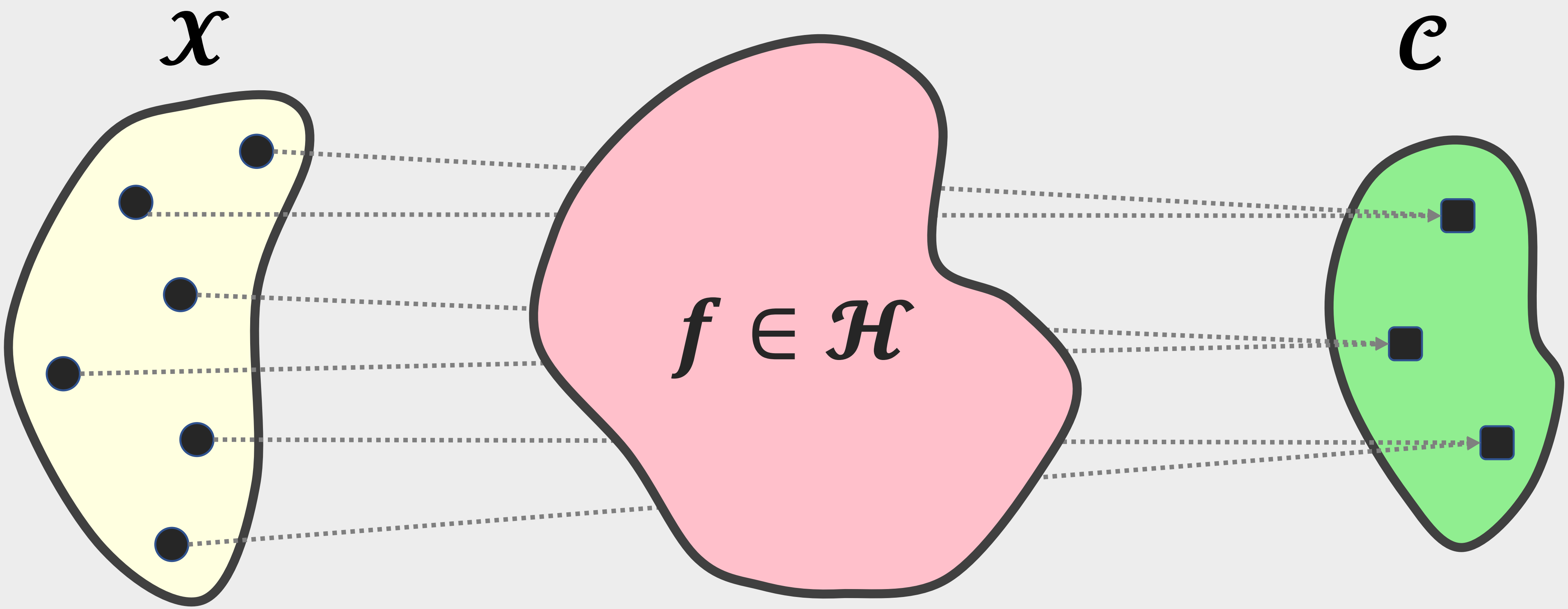
Learning $\boldsymbol{f}: \boldsymbol{X} \rightarrow \boldsymbol{y}$

Supervised learning approximates a function $g \sim f$ for mapping inputs X to outputs y



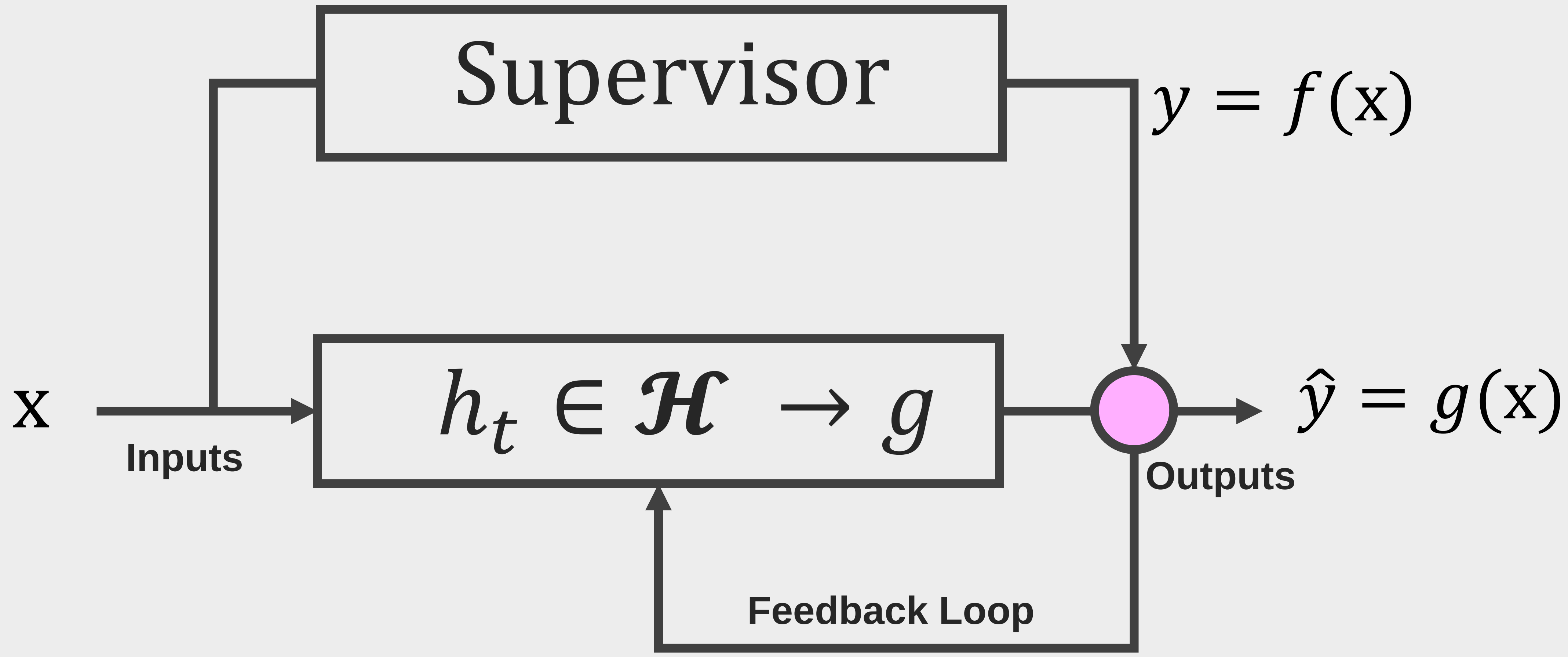
Learning $f: X \rightarrow y$

We need to find the unknown target function f that does the task of mapping



Input $\mathbf{X} \in$ Input space $\mathcal{X}$ hypothesis space $\mathcal{H}$ output $y \in$ concept space $\mathcal{C}$

How to produces a function $g: X \to y$



What Learning Needs

One given input-output data Learning needs the method(s) to

Represent

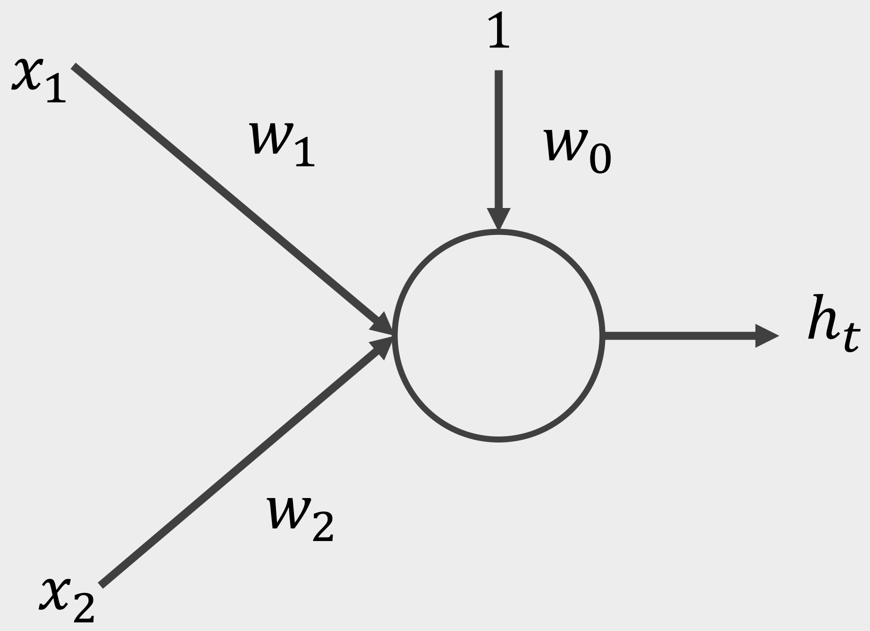
Evaluate

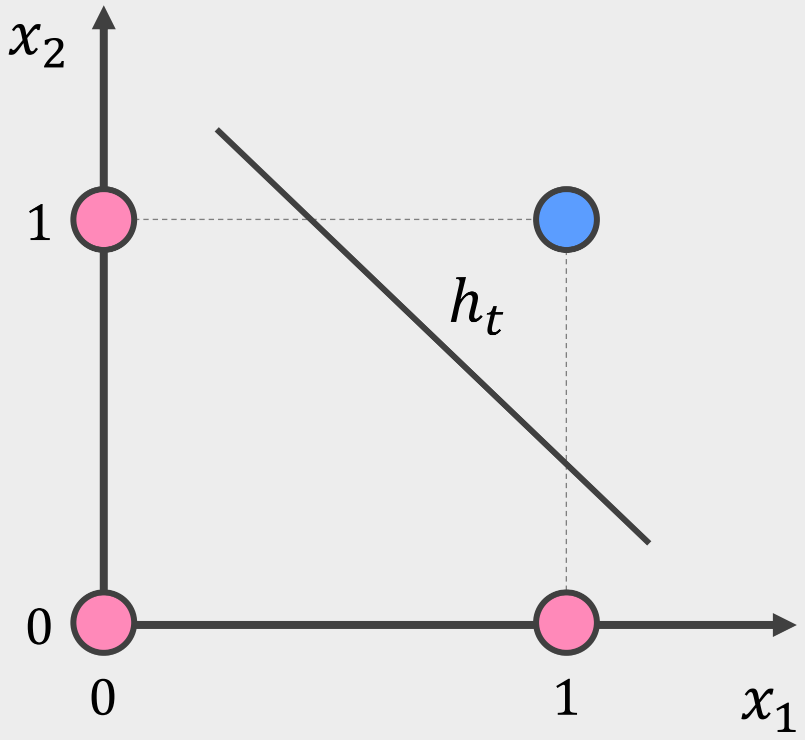
Optimize

a hypothesis h_t (e.g., a neural model)

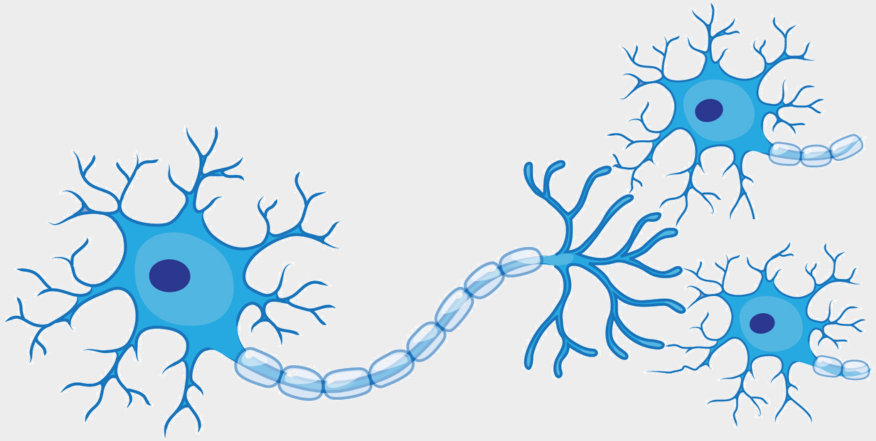
How to represent a hypothesis $h_t \in H$

A line separating data can be considered a hypothesis

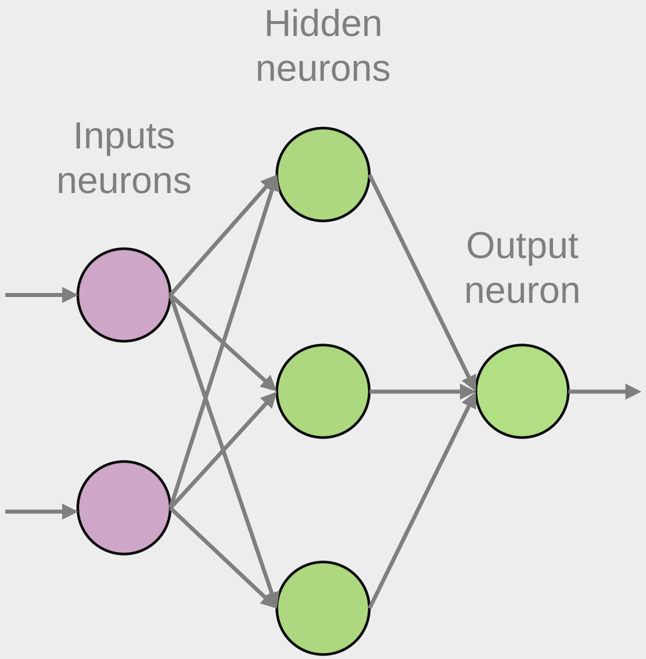




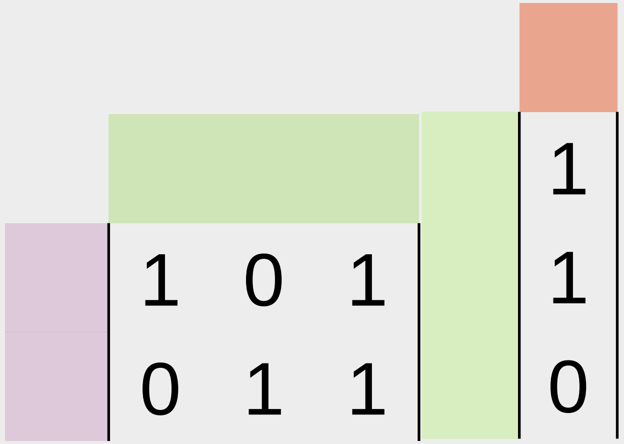
Learning Systems: Neural Networks



1 Biological networks of neurons in human brains

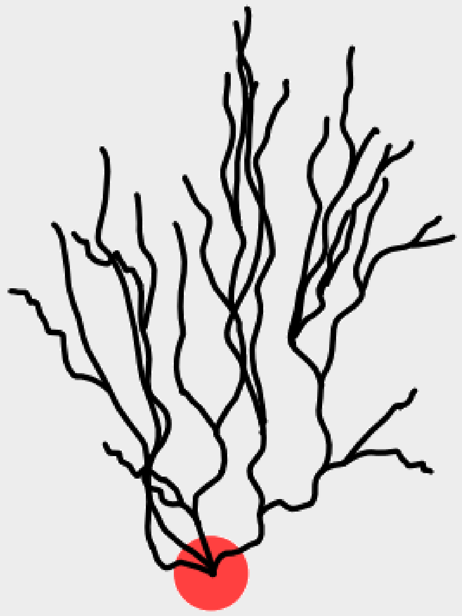


2 AI representation of biological neural networks

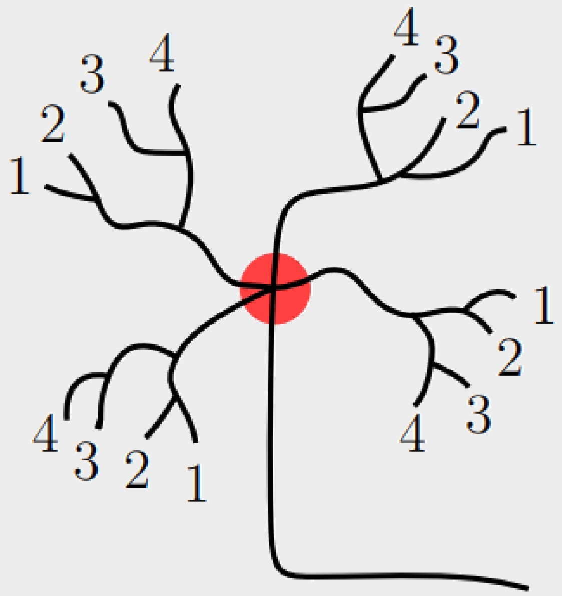


3 Mathematical representation of the neural networks

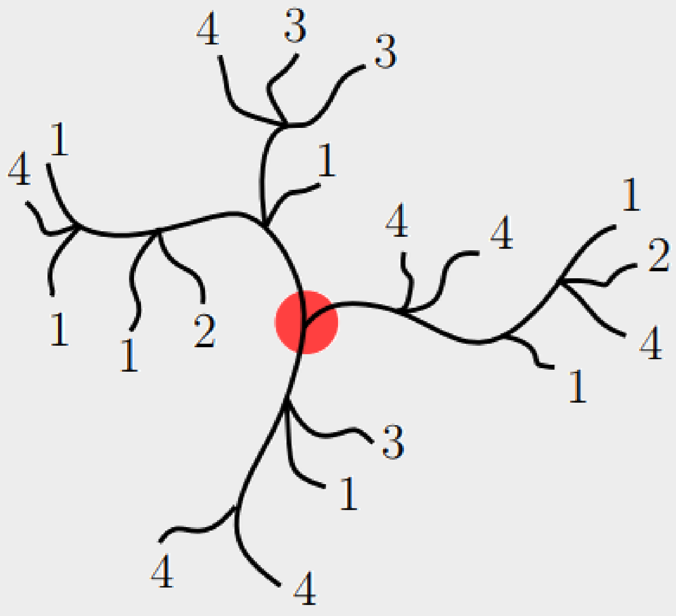
Plausible Biological Inspiration



Travis et al. (2005)



Jones and Kording (2021)



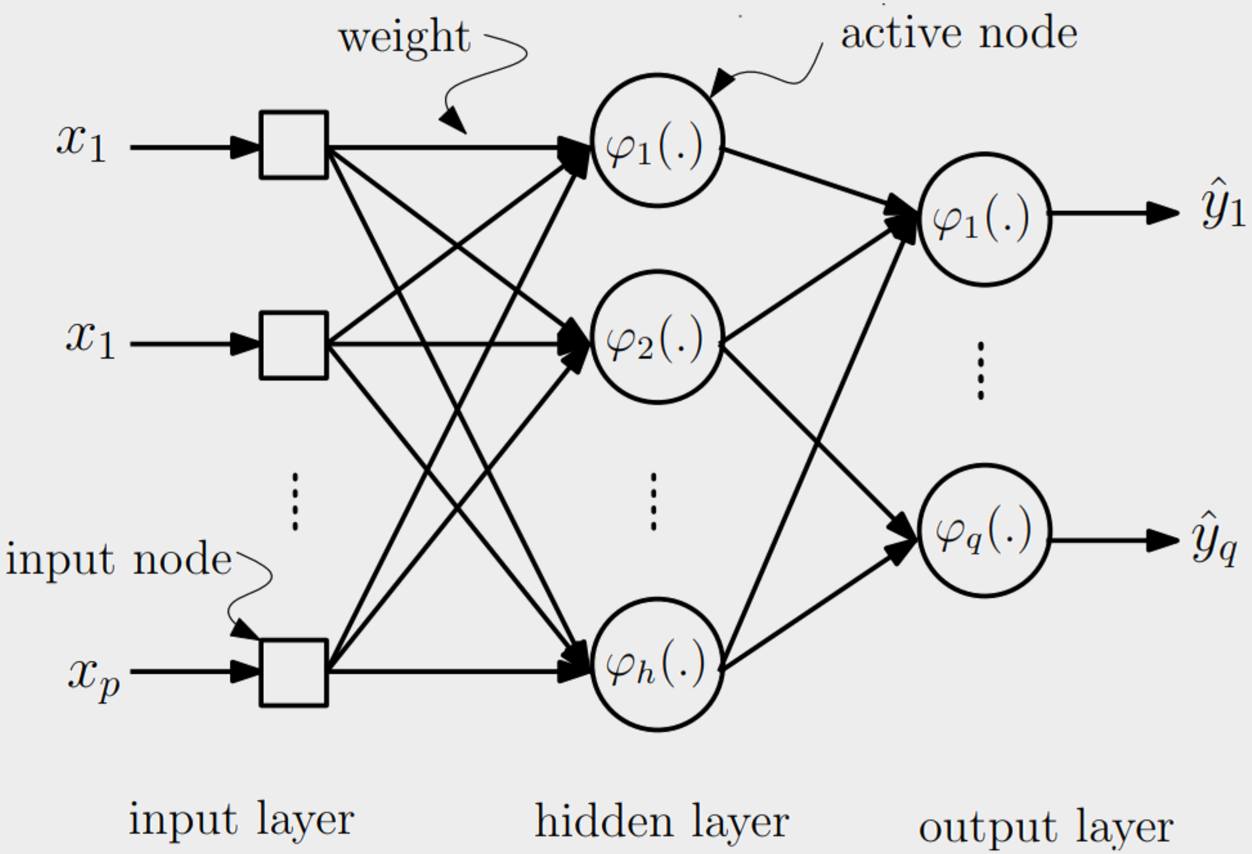
Ojha and Nicosia (2022)

Part 2
Neural Architectures

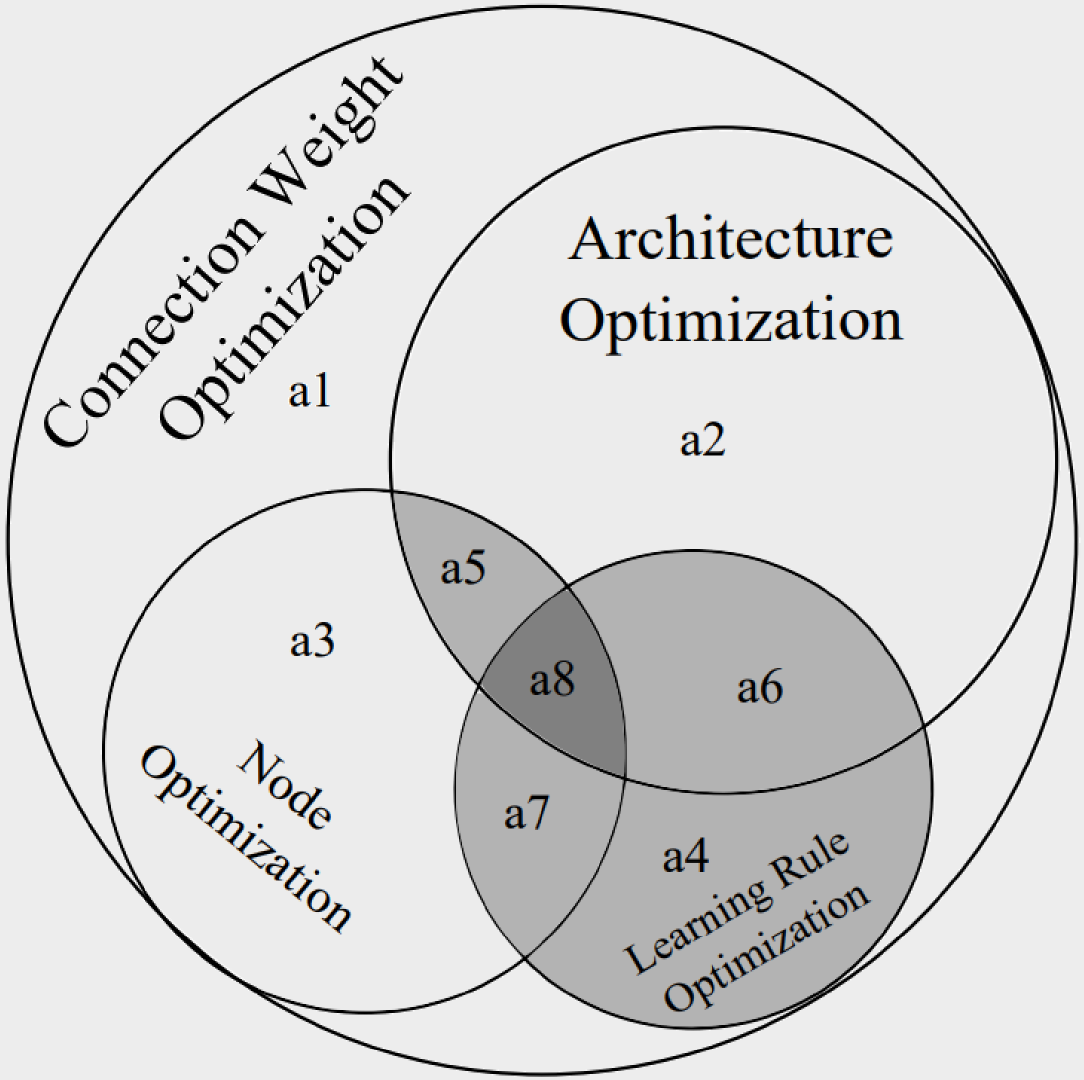
Neural Networks

NN components:

- Inputs
- Weights
- Architecture
- Activation functions
- Learning algorithms

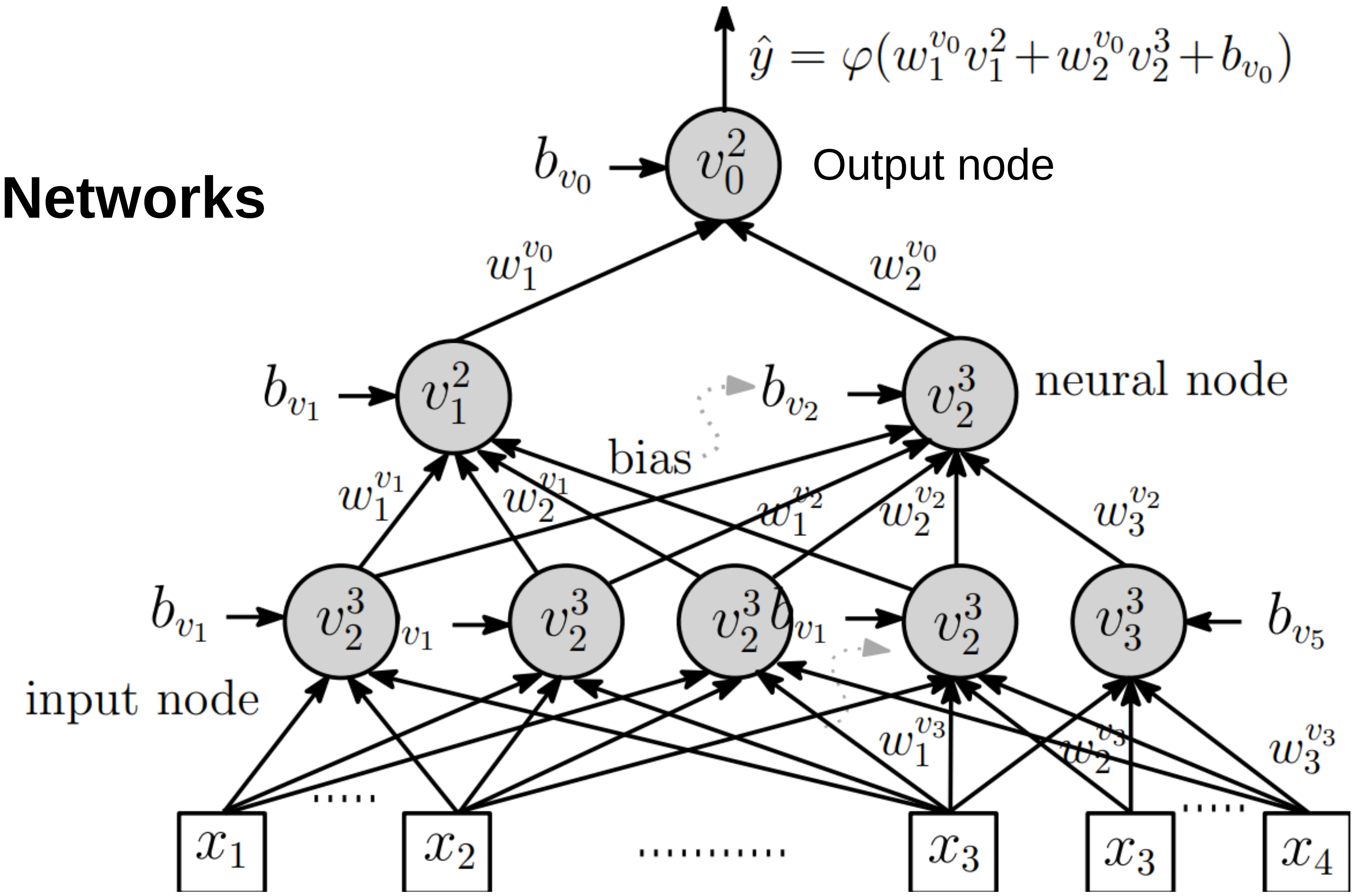


What could be optimized?



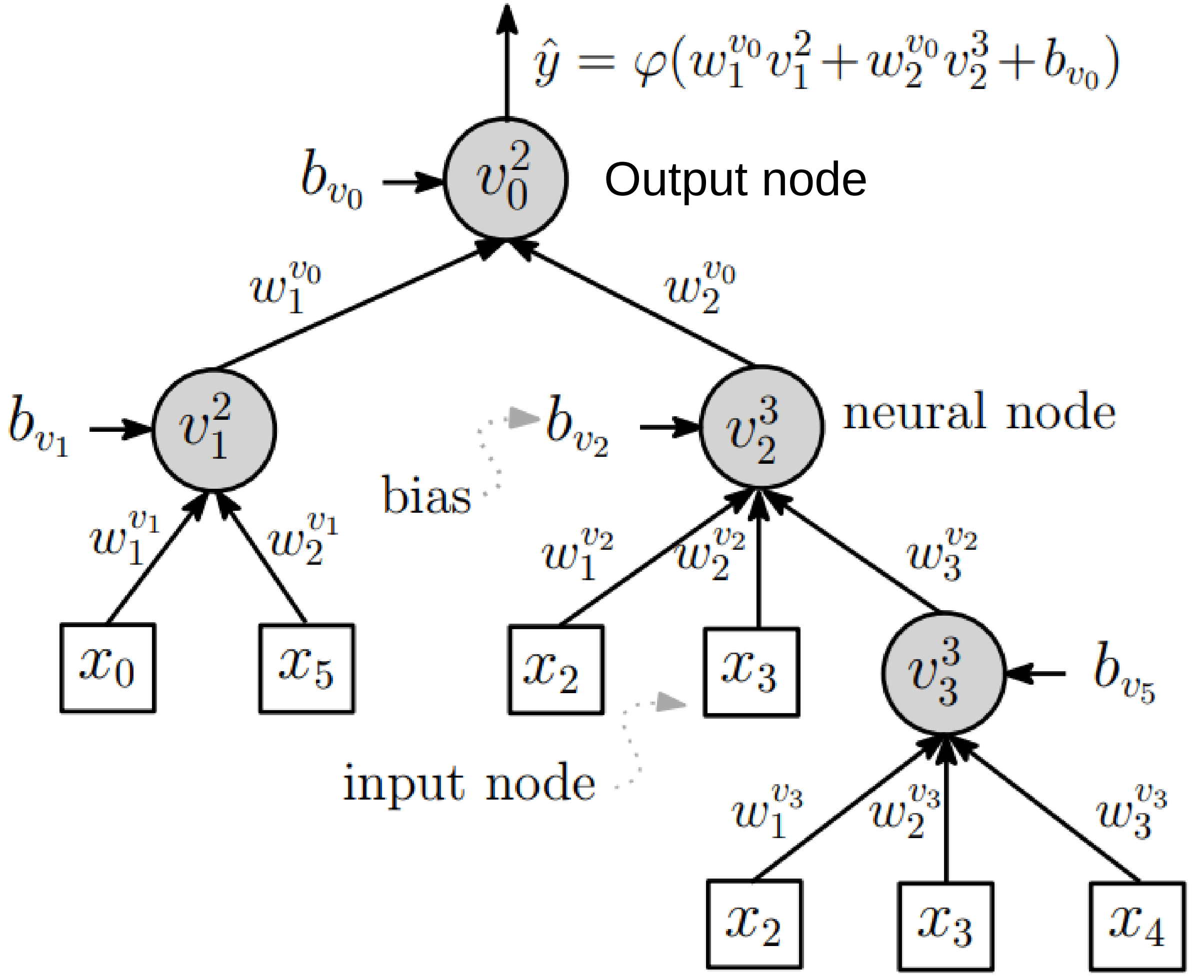
Neural Architecture

A Feedforward Neural Networks

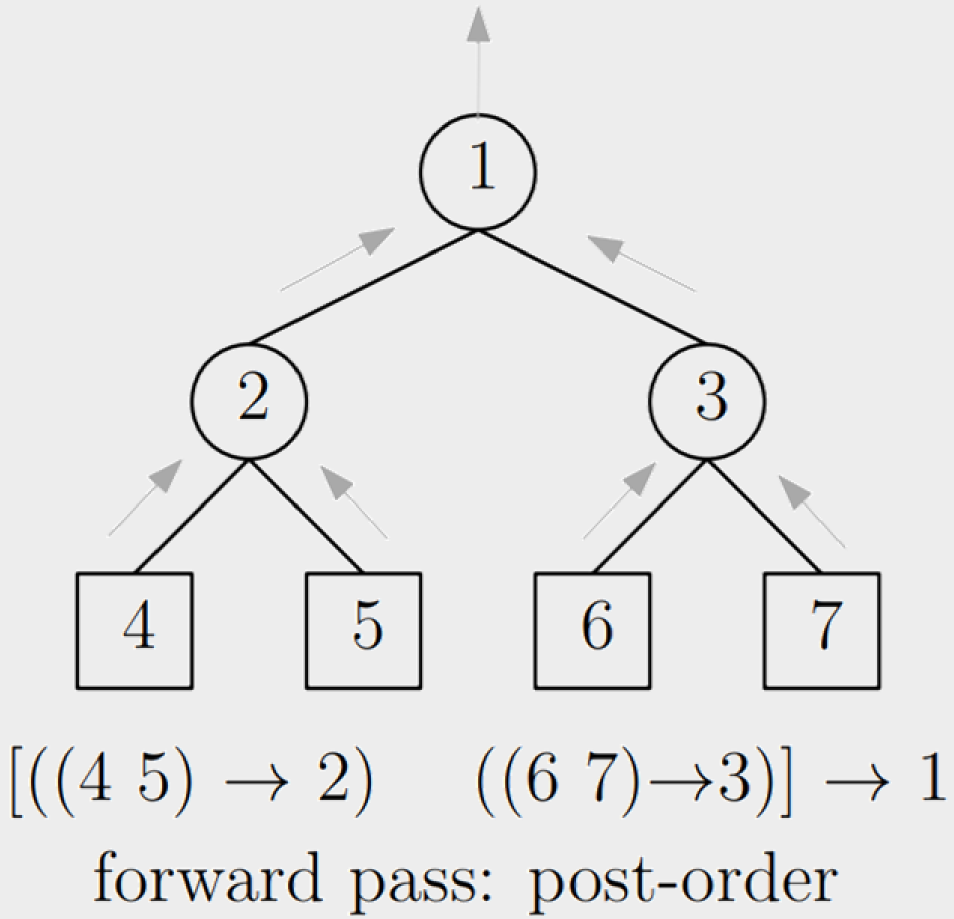


Neural Architecture

A Feedforward Neural Tree

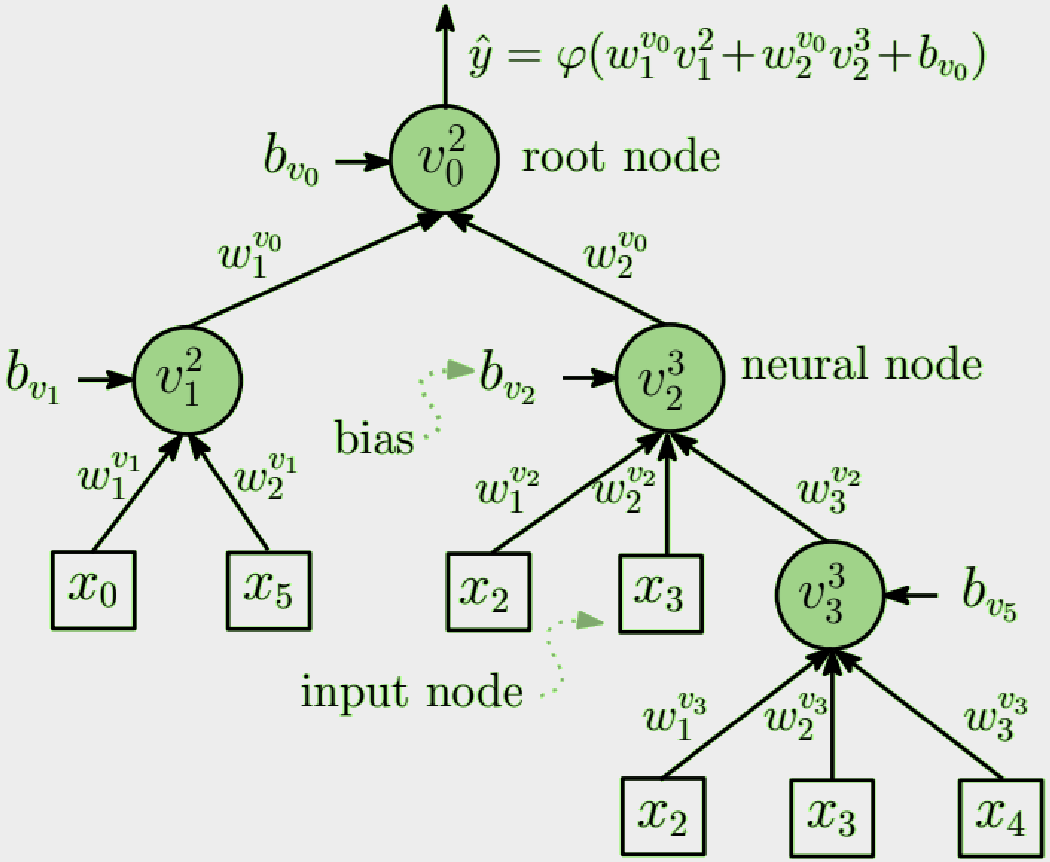


Neural Computation



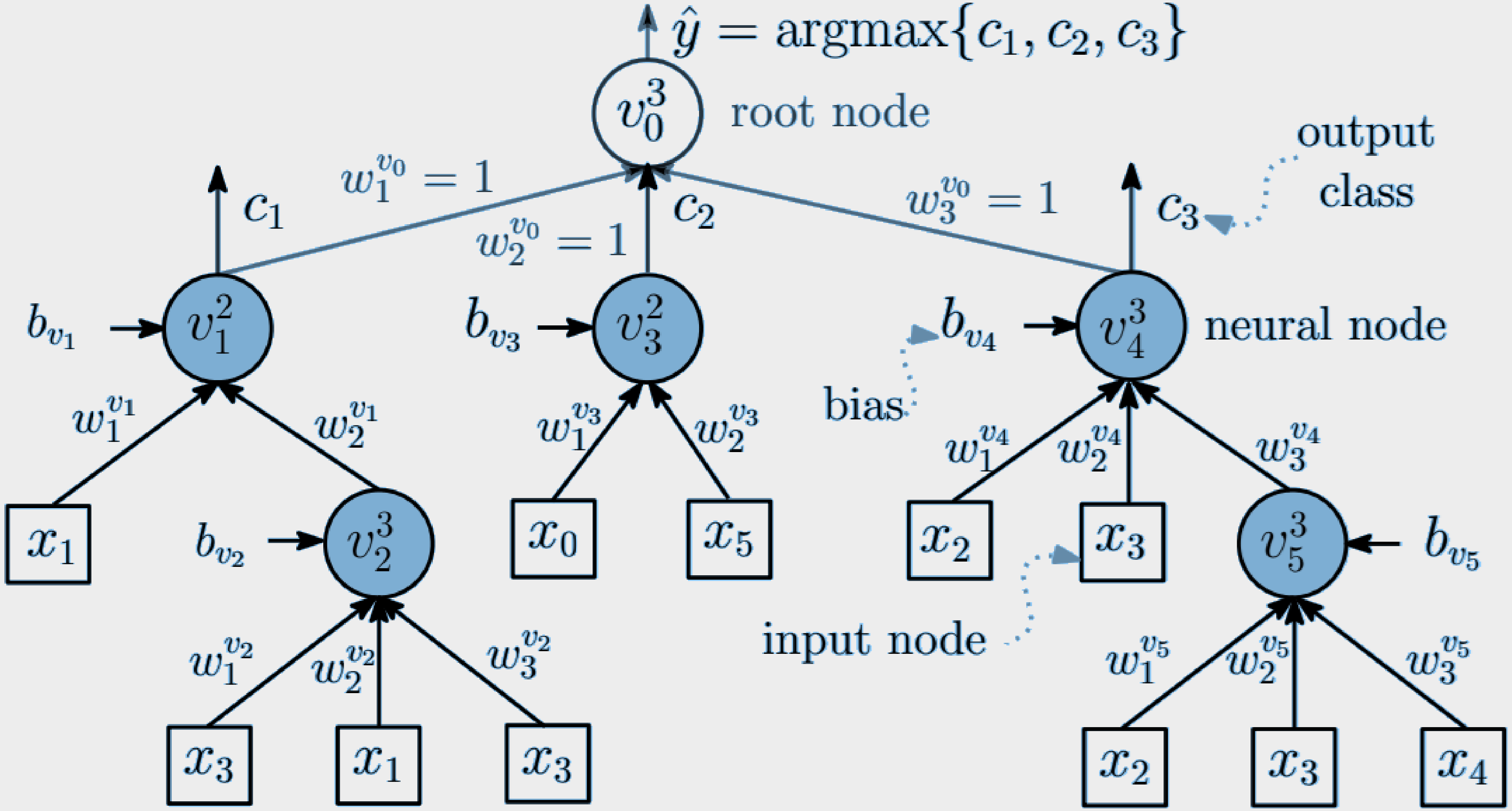
Types of Neural Tree

Regression Tree



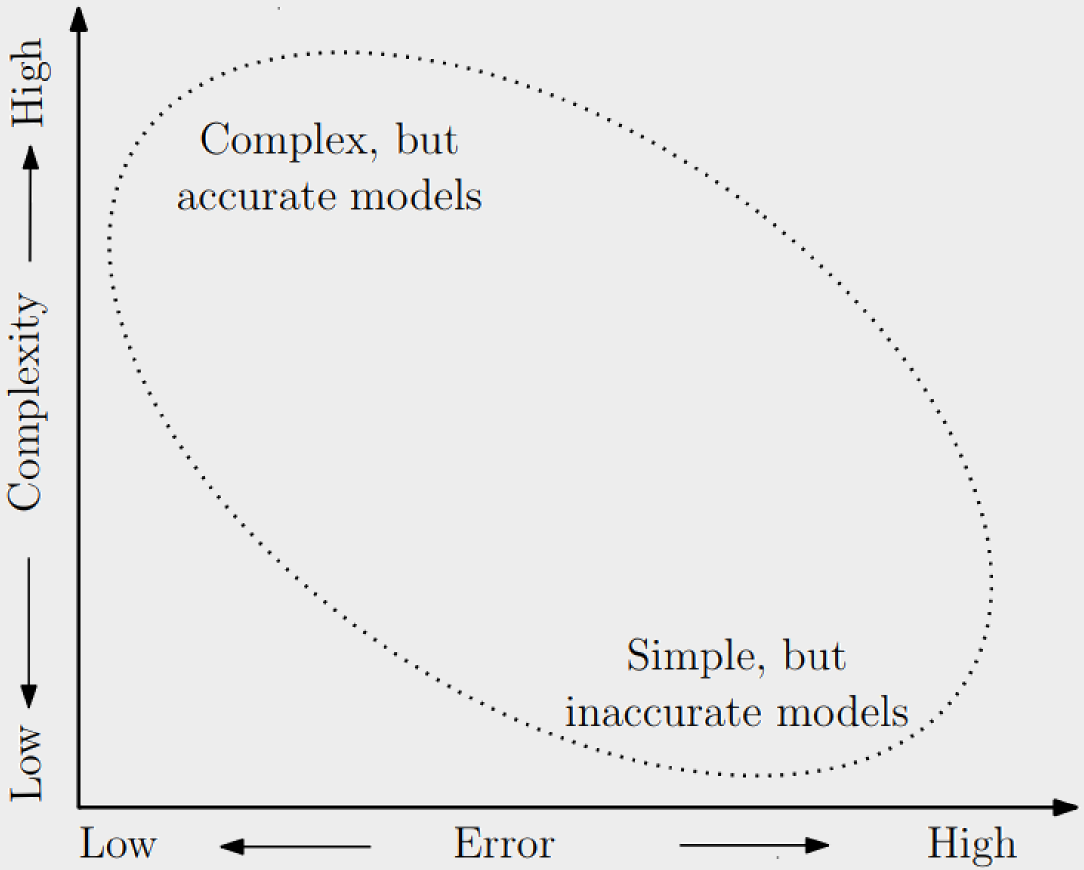
Types of Neural Tree

Classification Tree



Neural Architecture Search

Trade-offs

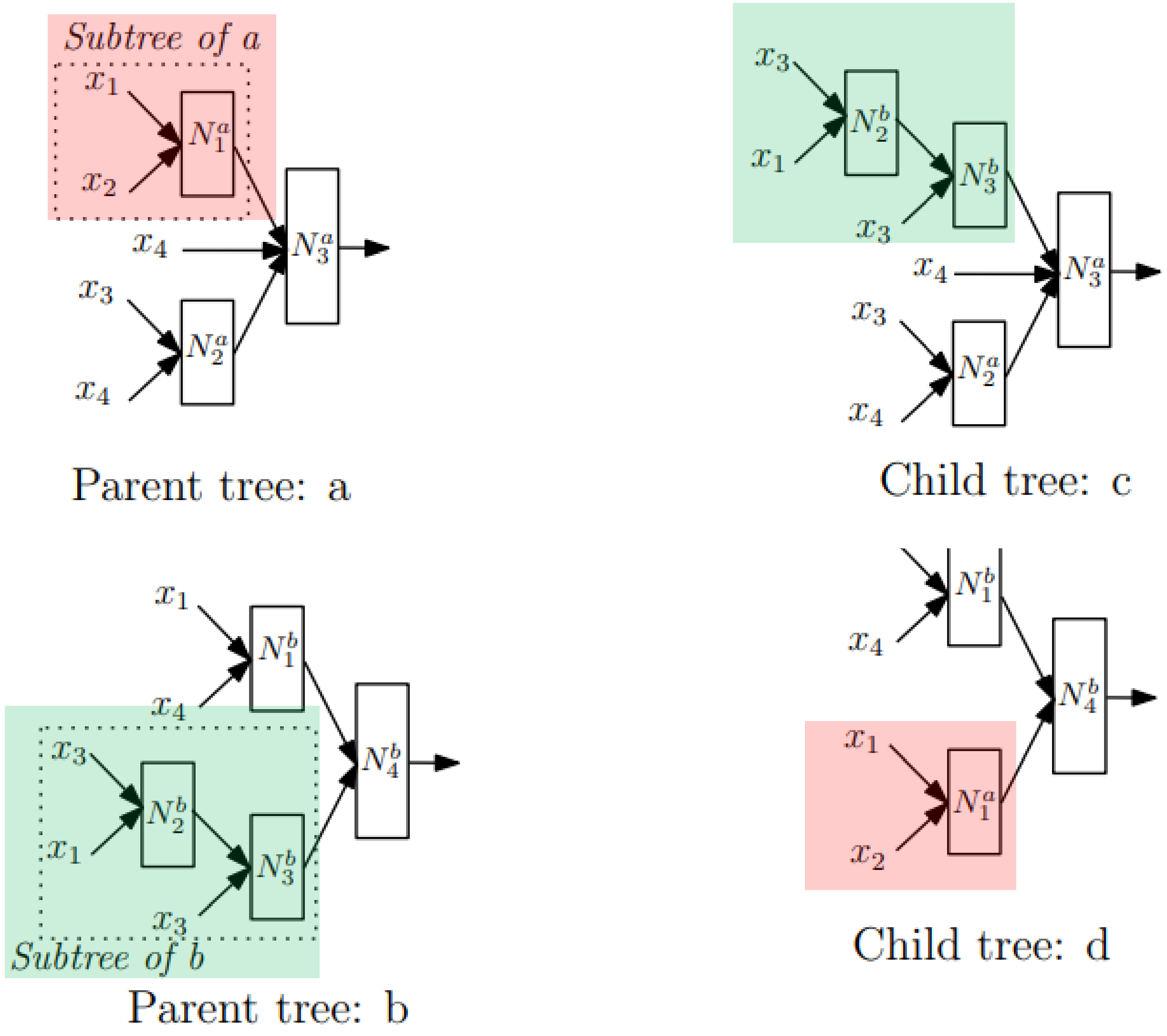


Neural Architecture Search

Trade-offs

Multiobjective
Genetic Programming
Crossover

Ojha et al (2017), *IEEE Trans. Fuzzy Systems*

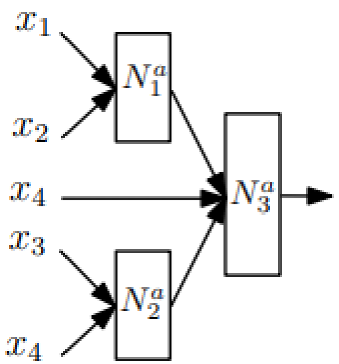


Neural Architecture Search

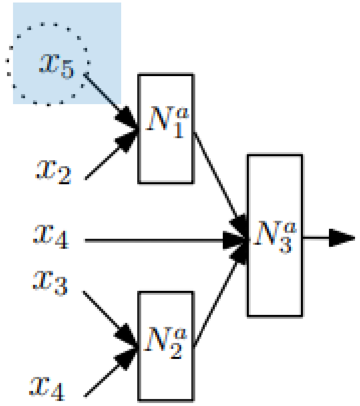
Trade-offs

Multiobjective Genetic Programming
Mutation

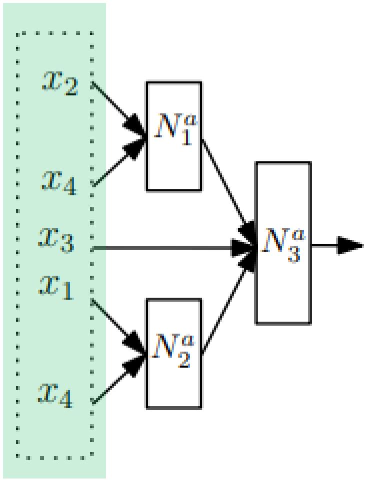
Ojha et al (2017), *IEEE Trans. Fuzzy Systems*



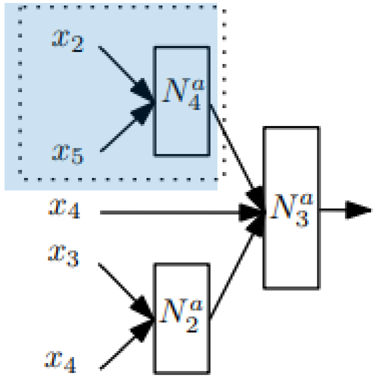
Parent tree



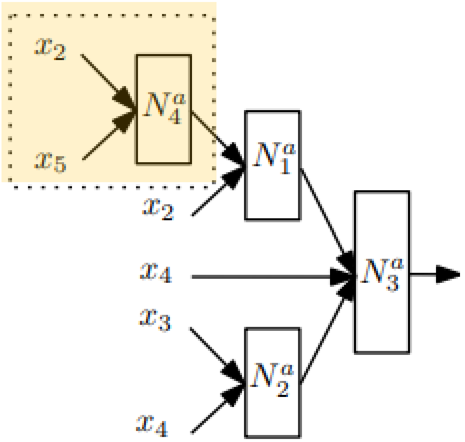
Single leaf mutation



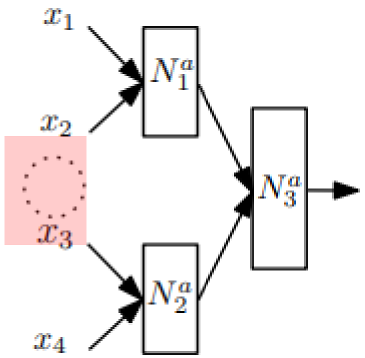
All leaves mutation



A subtree replacement



A subtree insertion

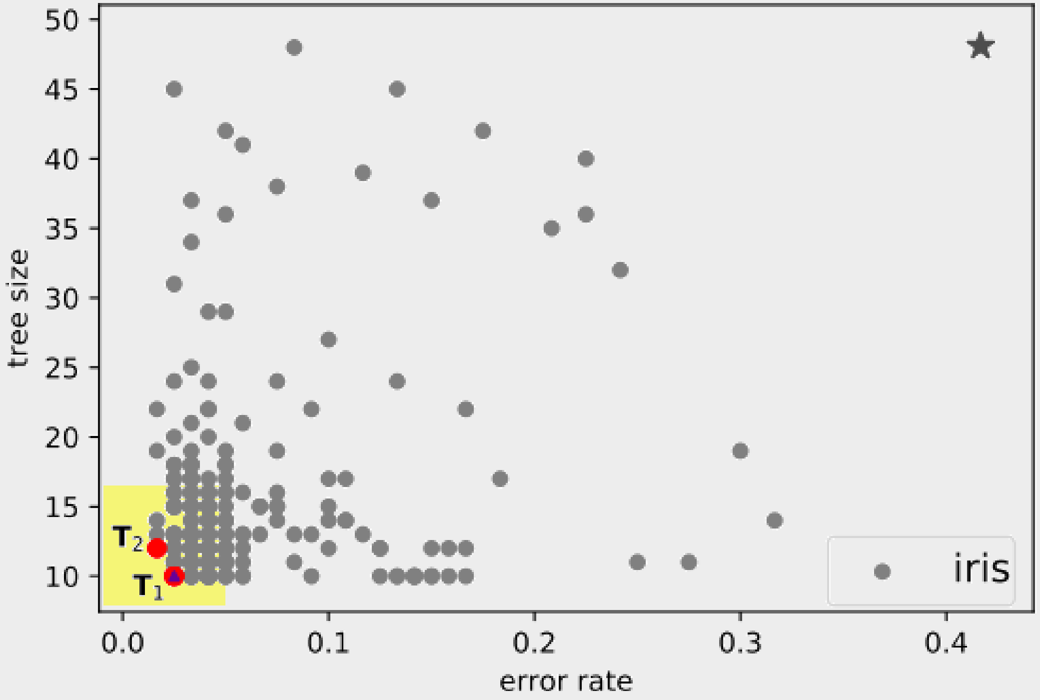


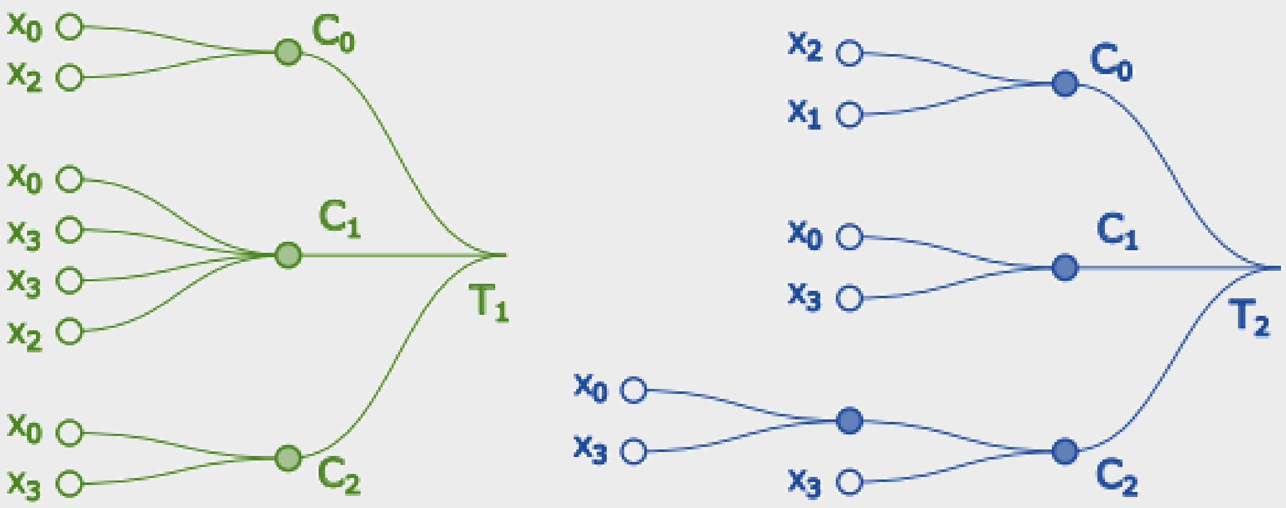
A subtree deletion

Architecture Search Trade-offs

Multiobjective Genetic Programming

Selection of trees using Hypervolume indicator from a Pareto Front

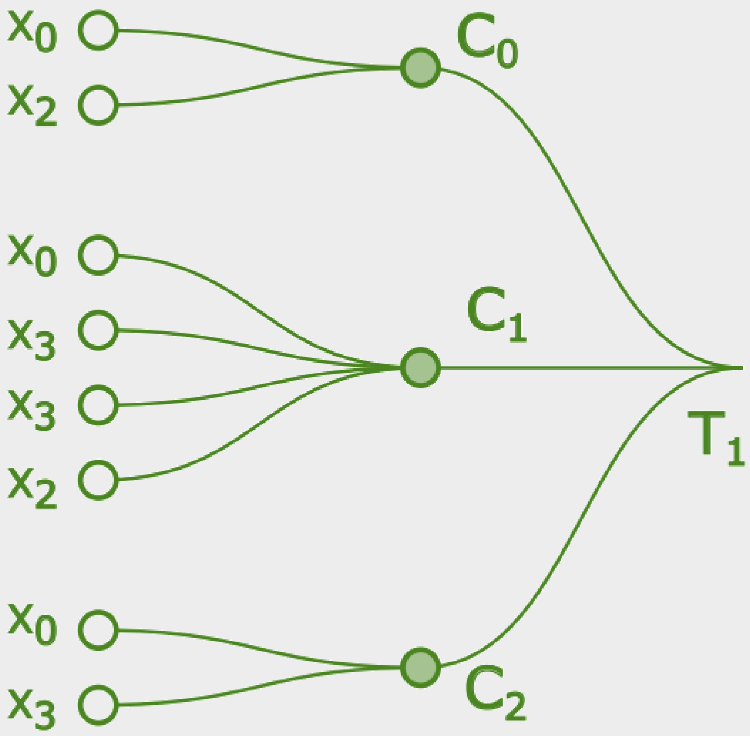


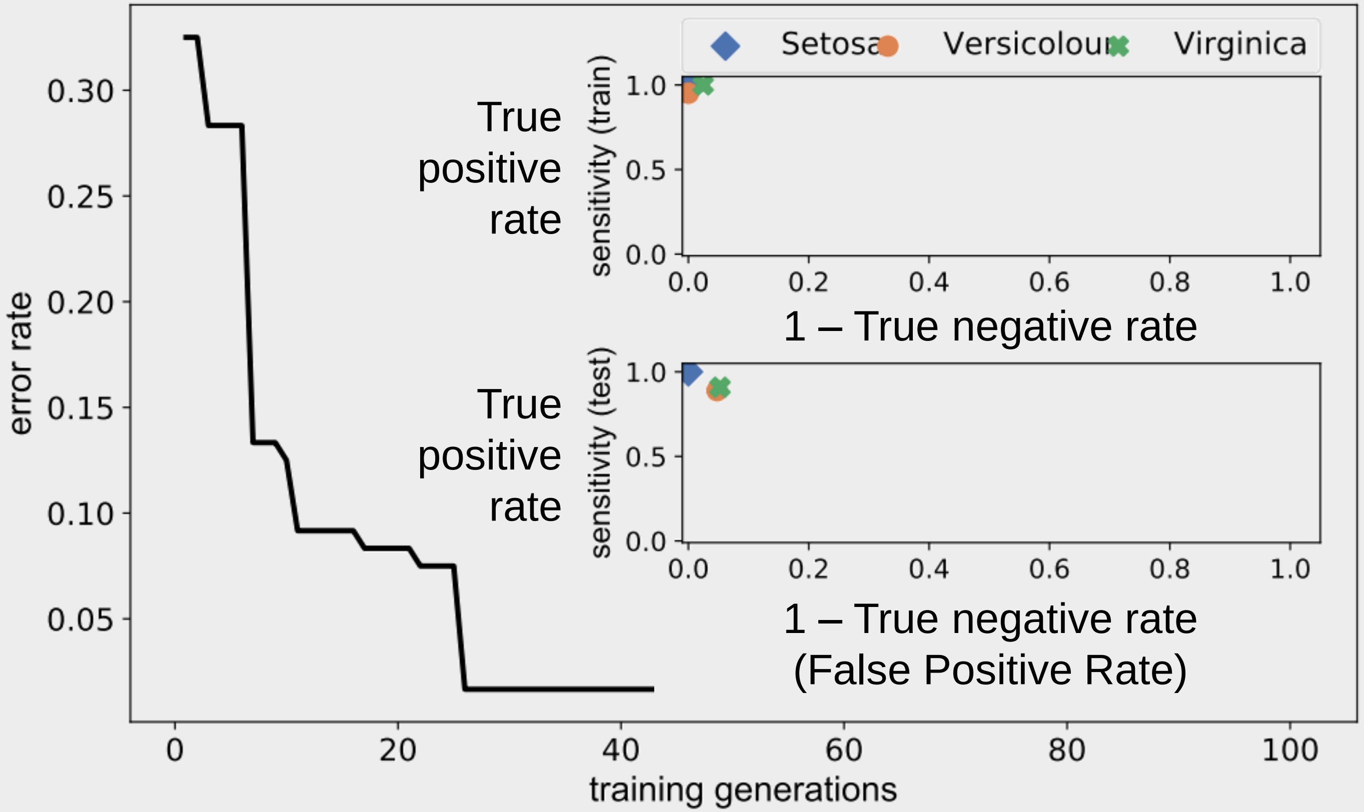


Ojha and Nicosia (2020), *CEC*

Learnability of Classes

Competition between classes



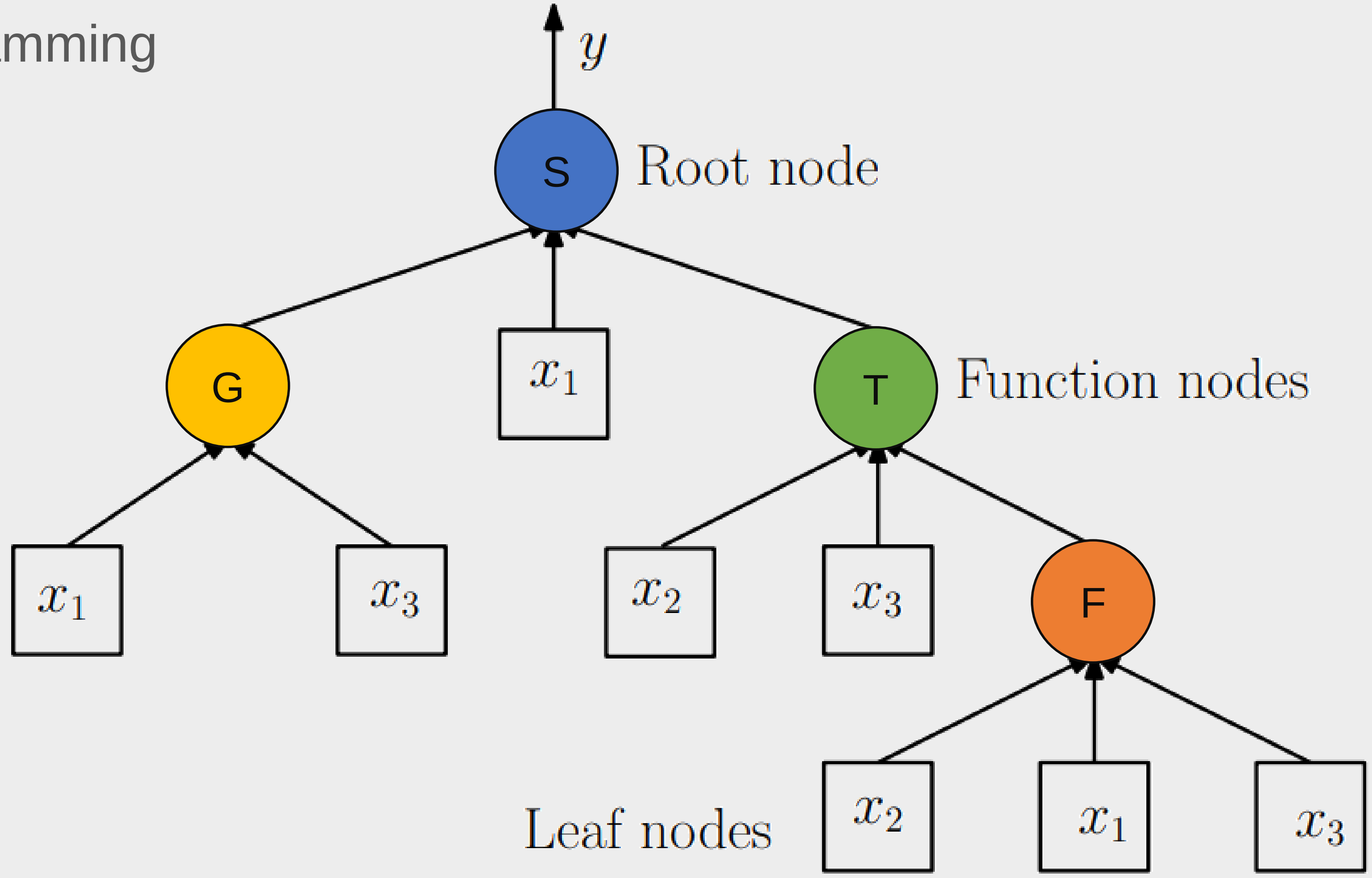


Heterogeneous Neural Tree

Multiobjective Genetic Programming

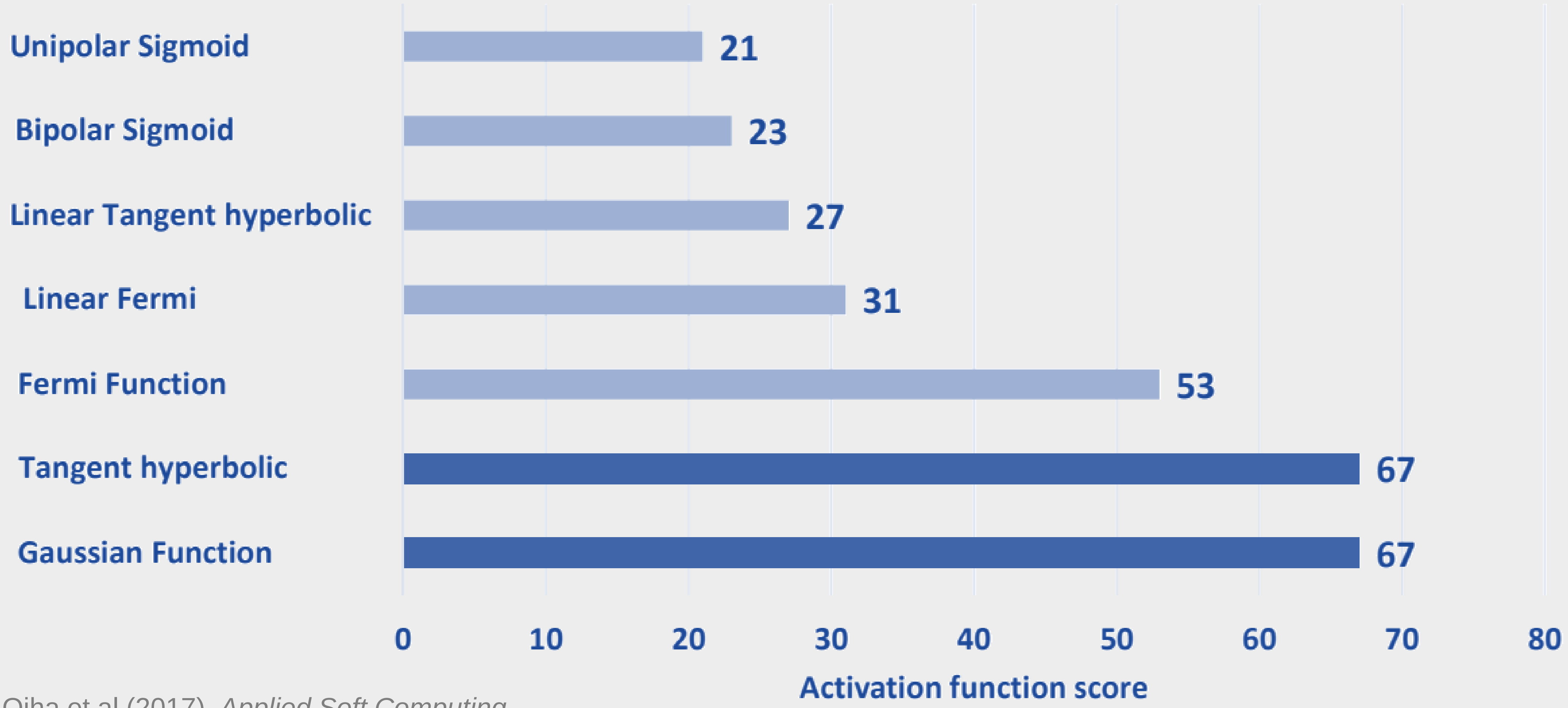
Activation Function Search

- S – Sigmoid
- G – Gaussian
- T – Tanh
- F – Fermi



Activation Function Performance

Higher values are better



Ojha et al (2017), *Applied Soft Computing*

Part 3

Backpropagation

Neural Tree

Backpropagation Neural Tree

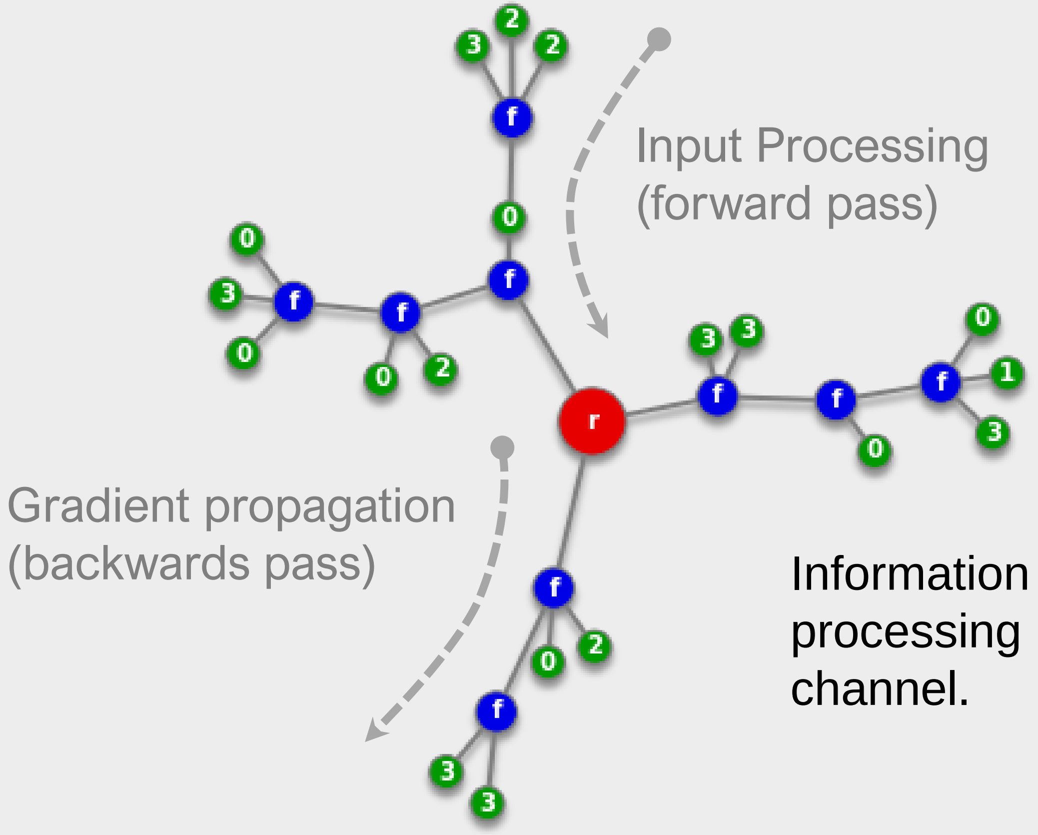
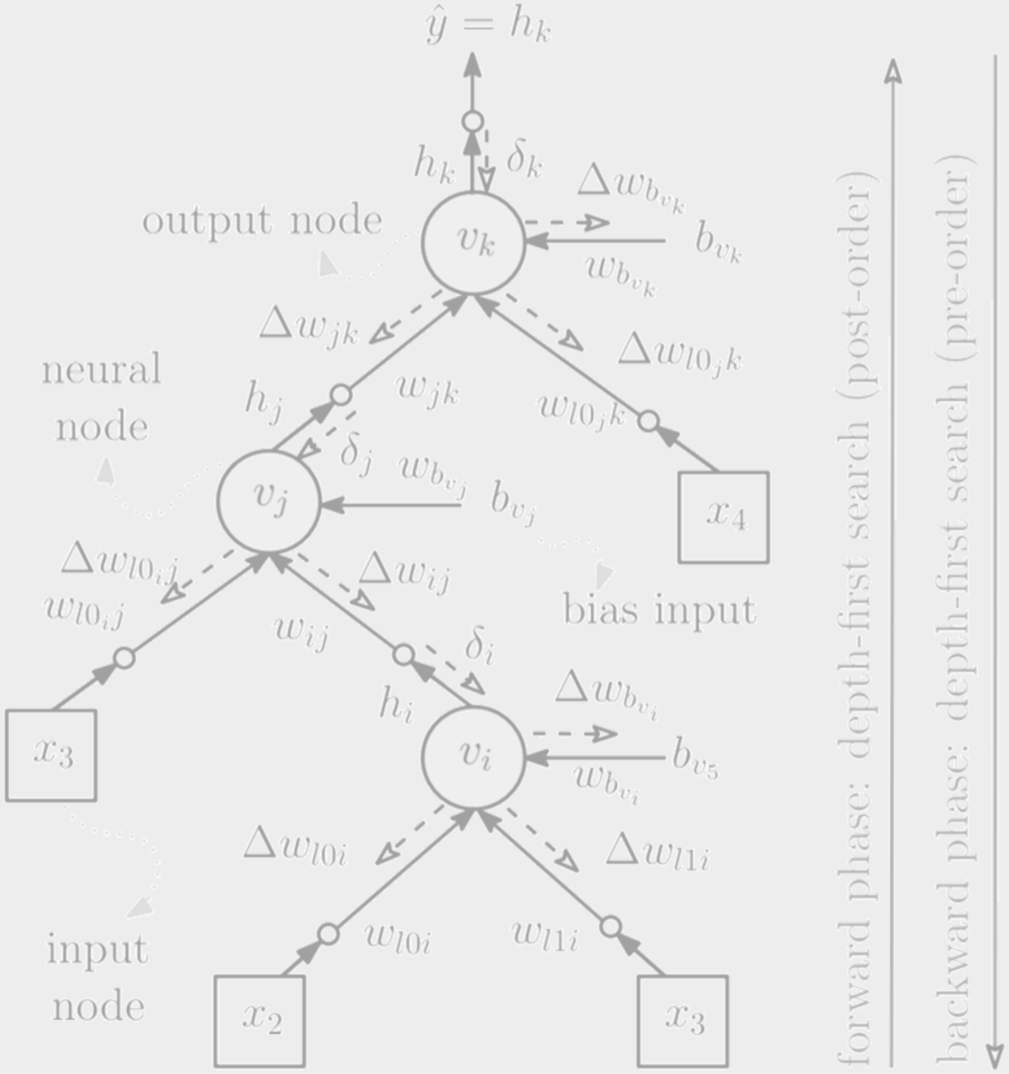
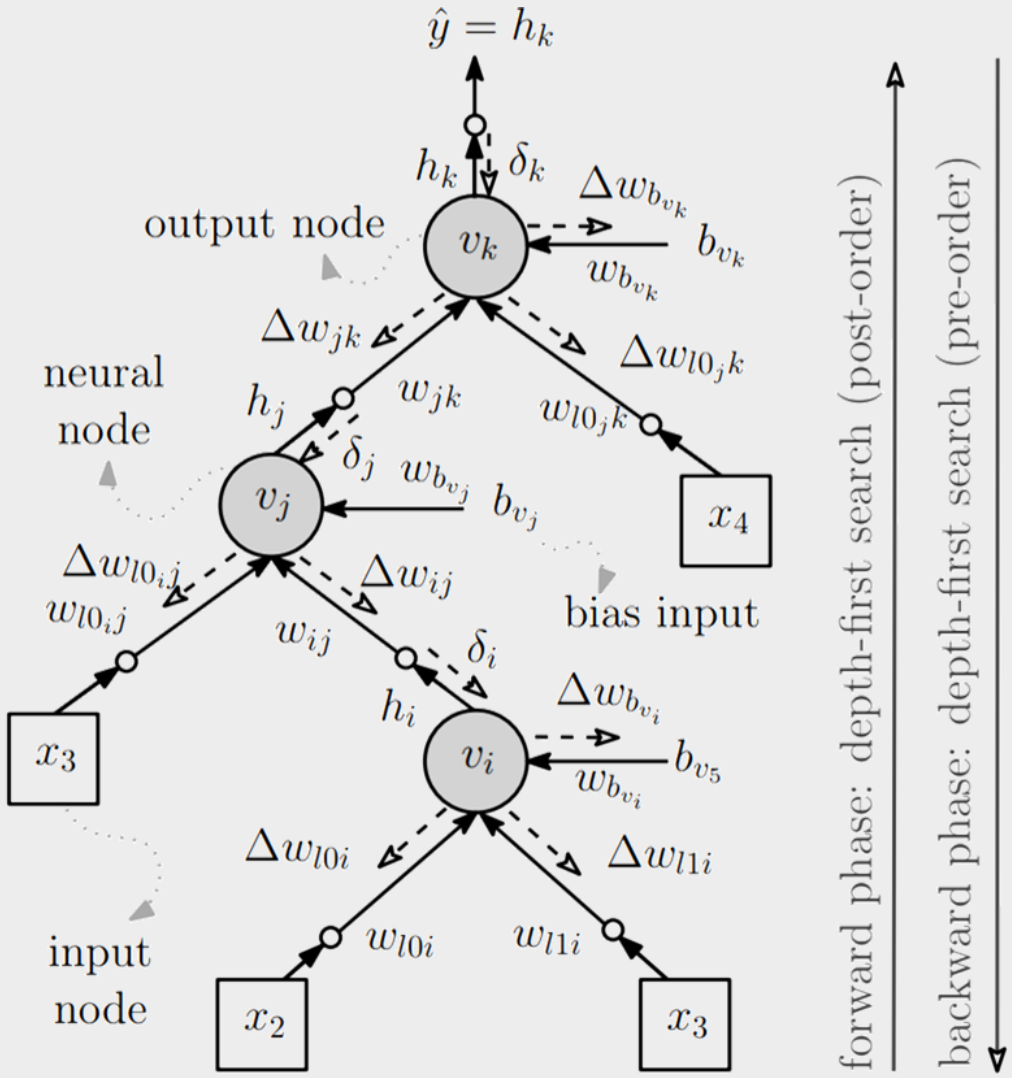
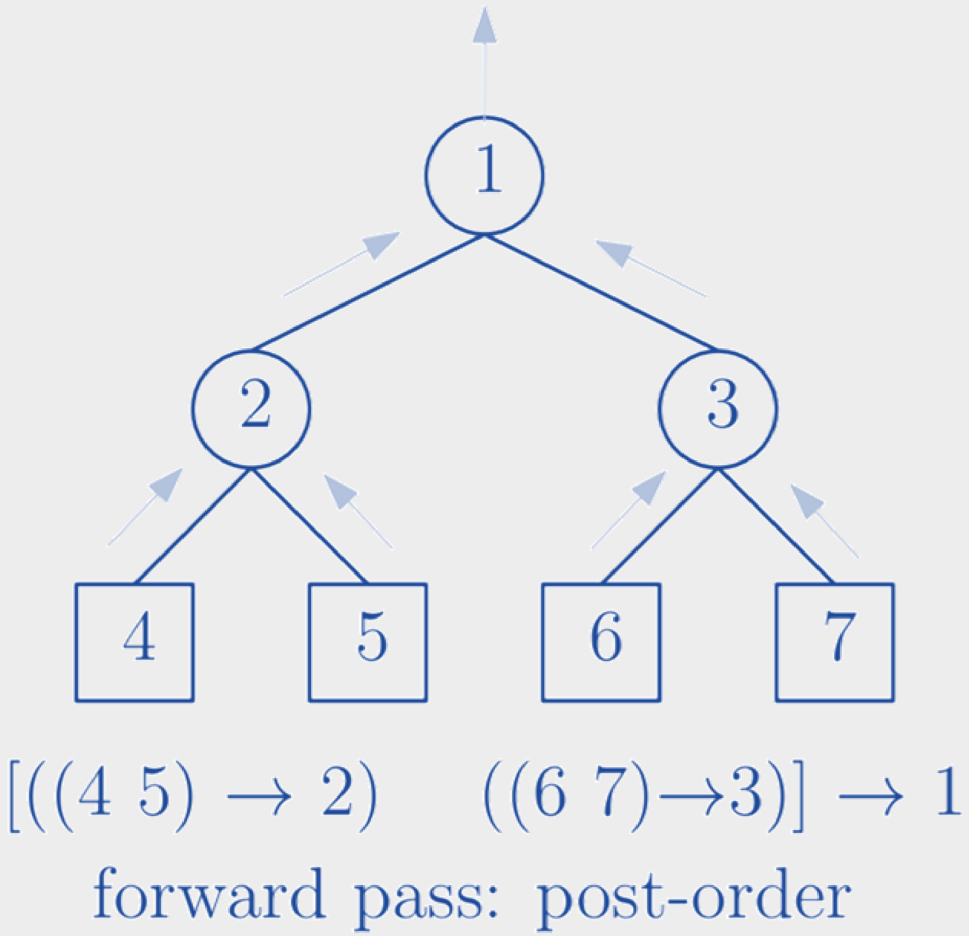


Fig A. Forward pass and gradient backpropagation

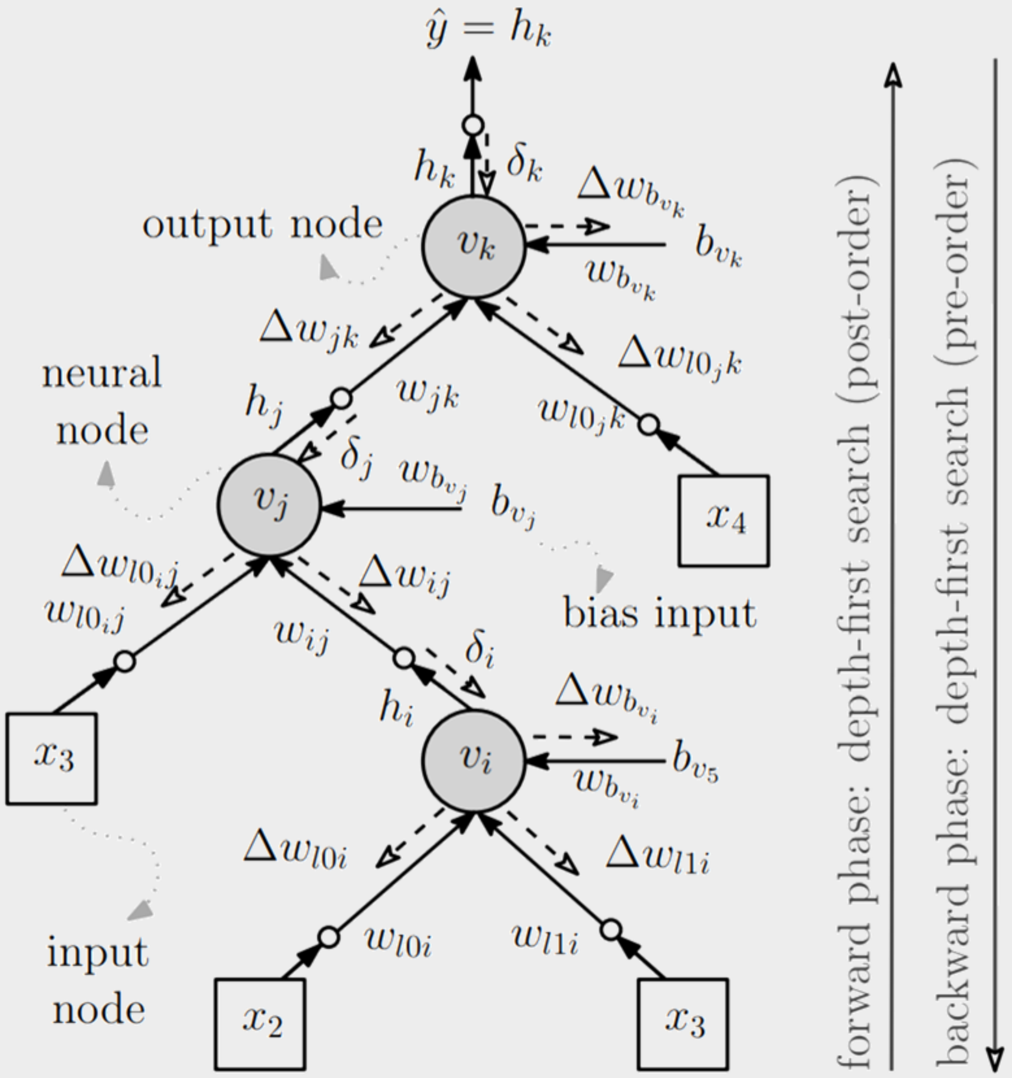


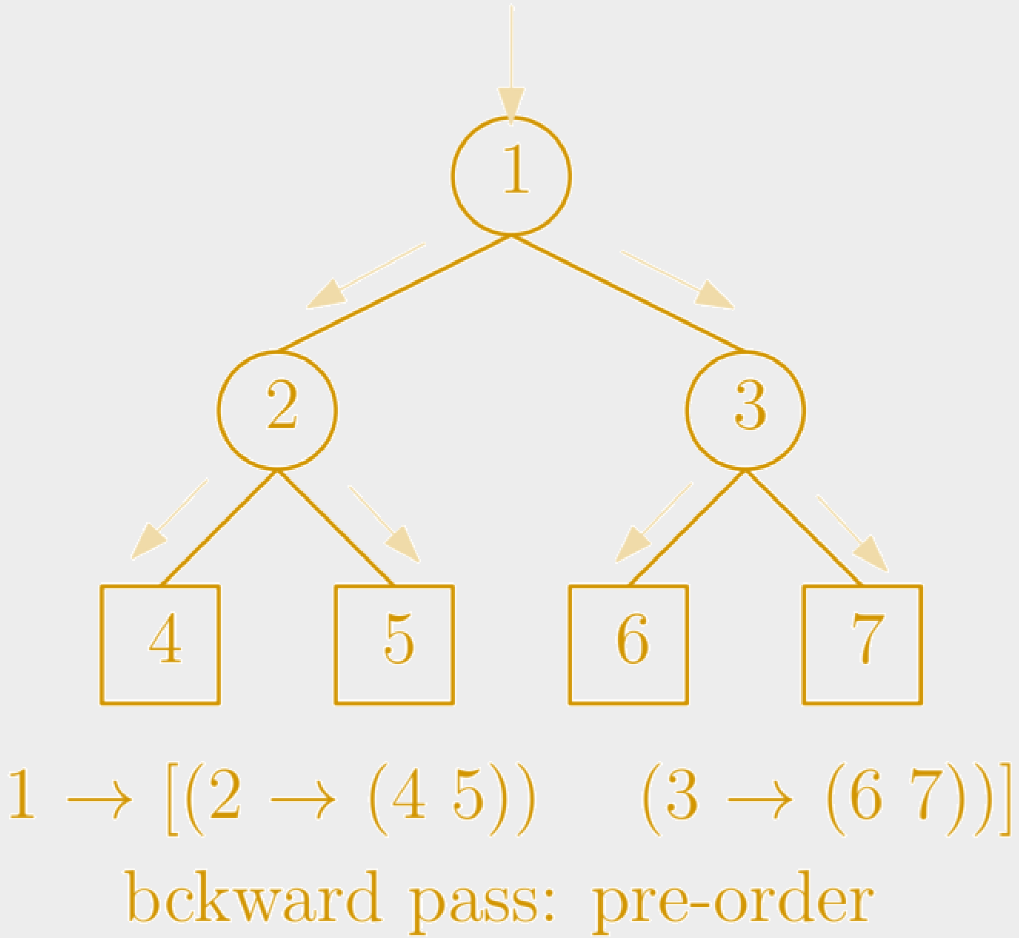
Backpropagation Neural Tree: Forward Pass



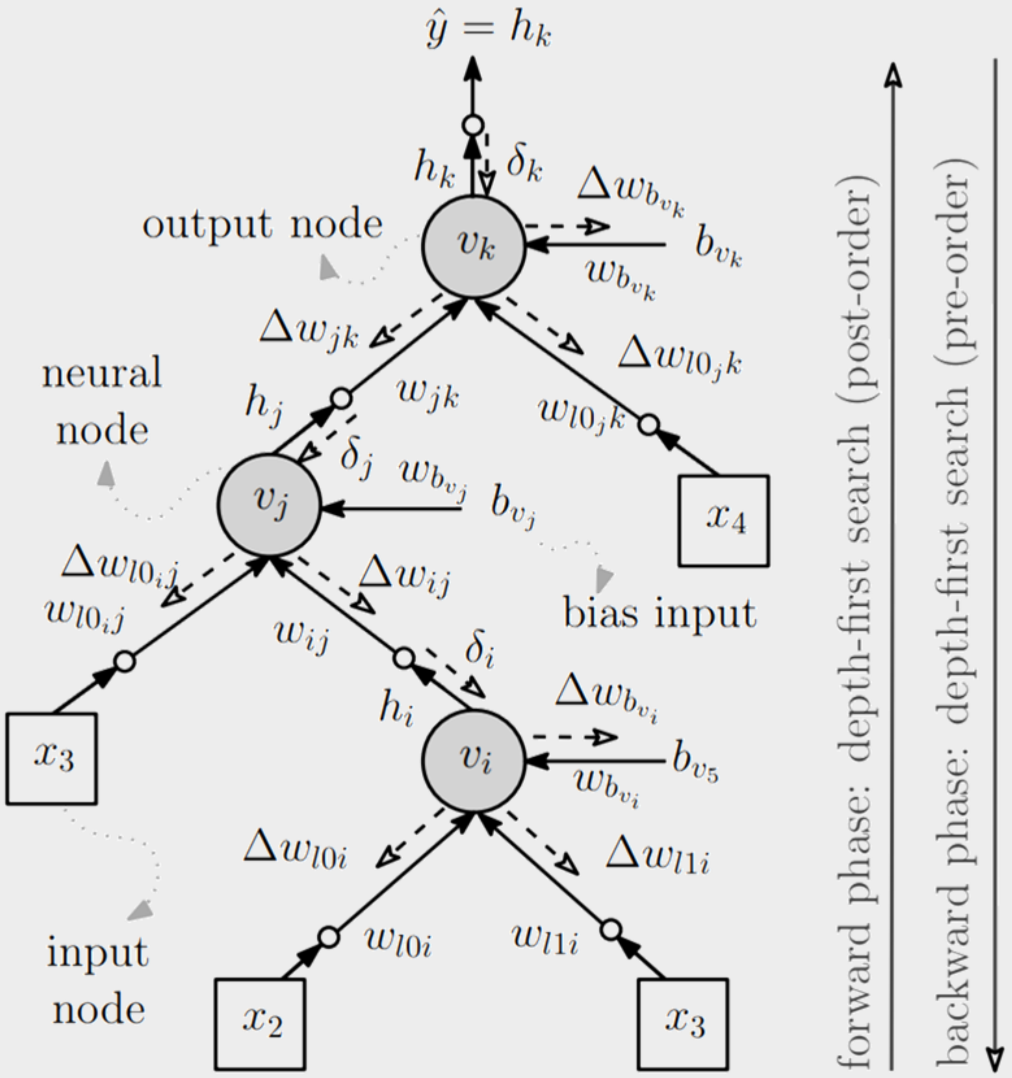


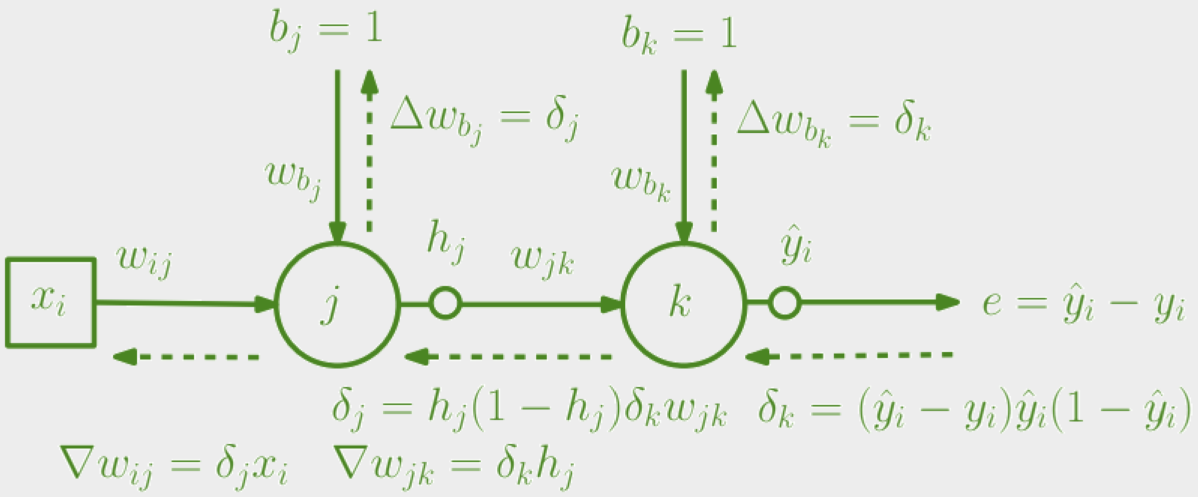
Backpropagation Neural Tree: Backward Pass





Backpropagation Neural Tree: Gradient Backpropagation

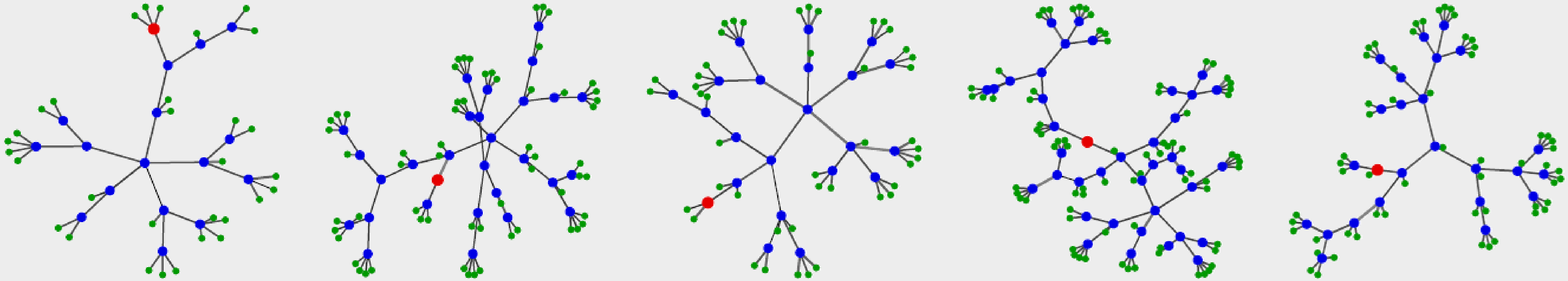




Backpropagation

Backpropagation Neural Tree: Performance on Regression

Regression results



(a) baseball (.85, 48) (b) dee (.89, 89) (c) diabetese (.63, 67) (d) friedman (.95, 116) (e) mpg6 (.9, 82)

Algorithm	Bas	Dee	Dia	Frd	Mpg	Avg Acc	Avg Weights
BNeuralT	0.665	0.837	0.492	0.776	0.867	**0.727**	**152**
MLP	0.721	0.829	0.49	0.943	0.874	**0.772**	**1041**

Ojha and Nicosia (2022), *Neural Networks*

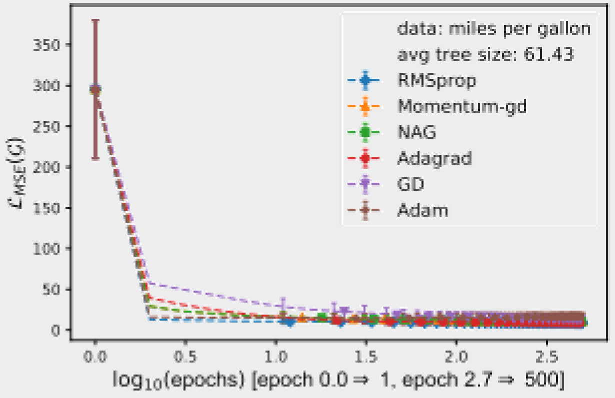
Backpropagation Neural Tree: Performance on Regression

Regression results

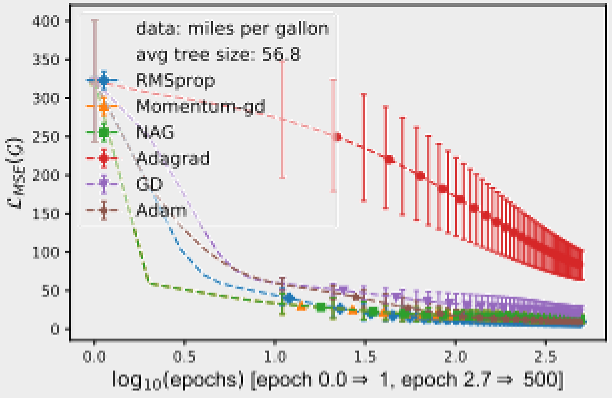
- BNeuralT used only 14.6% of MLP
- Accuracy differs only 5.8% lower than the best MLP result

Neural Tree vs Neural Networks

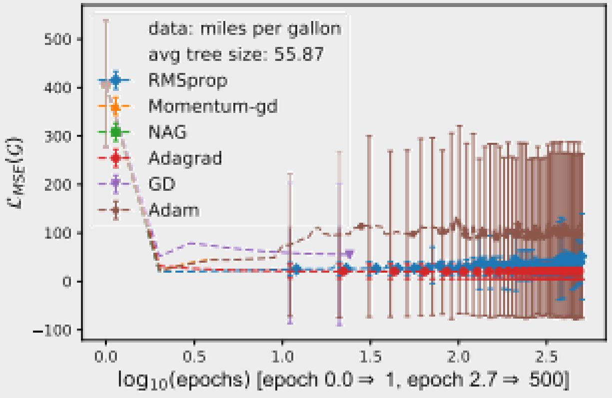
Regression Problems



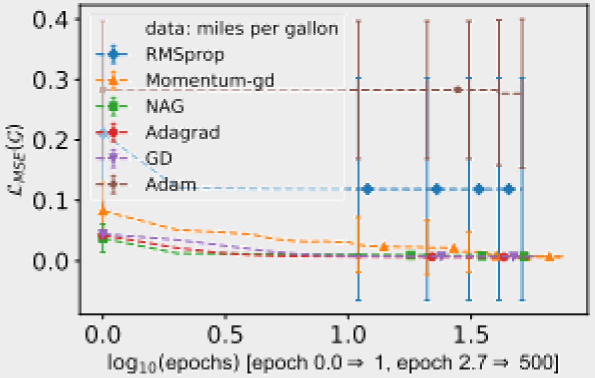
(g) BNeuralT: Sigmod, $\eta = 0.1$



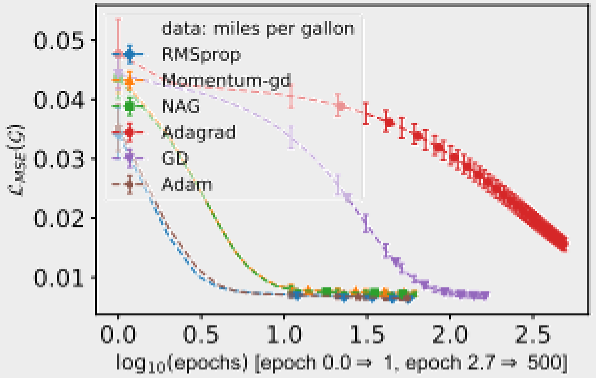
(h) BNeuralT: Sigmod, $\eta = default$



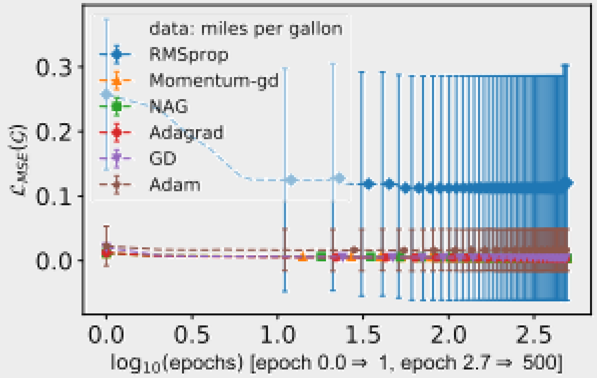
(i) BNeuralT: ReLU, $\eta = 0.1$



(j) MLP: Sigmod, $\eta = 0.1$



(k) MLP: Sigmod, $\eta = default$



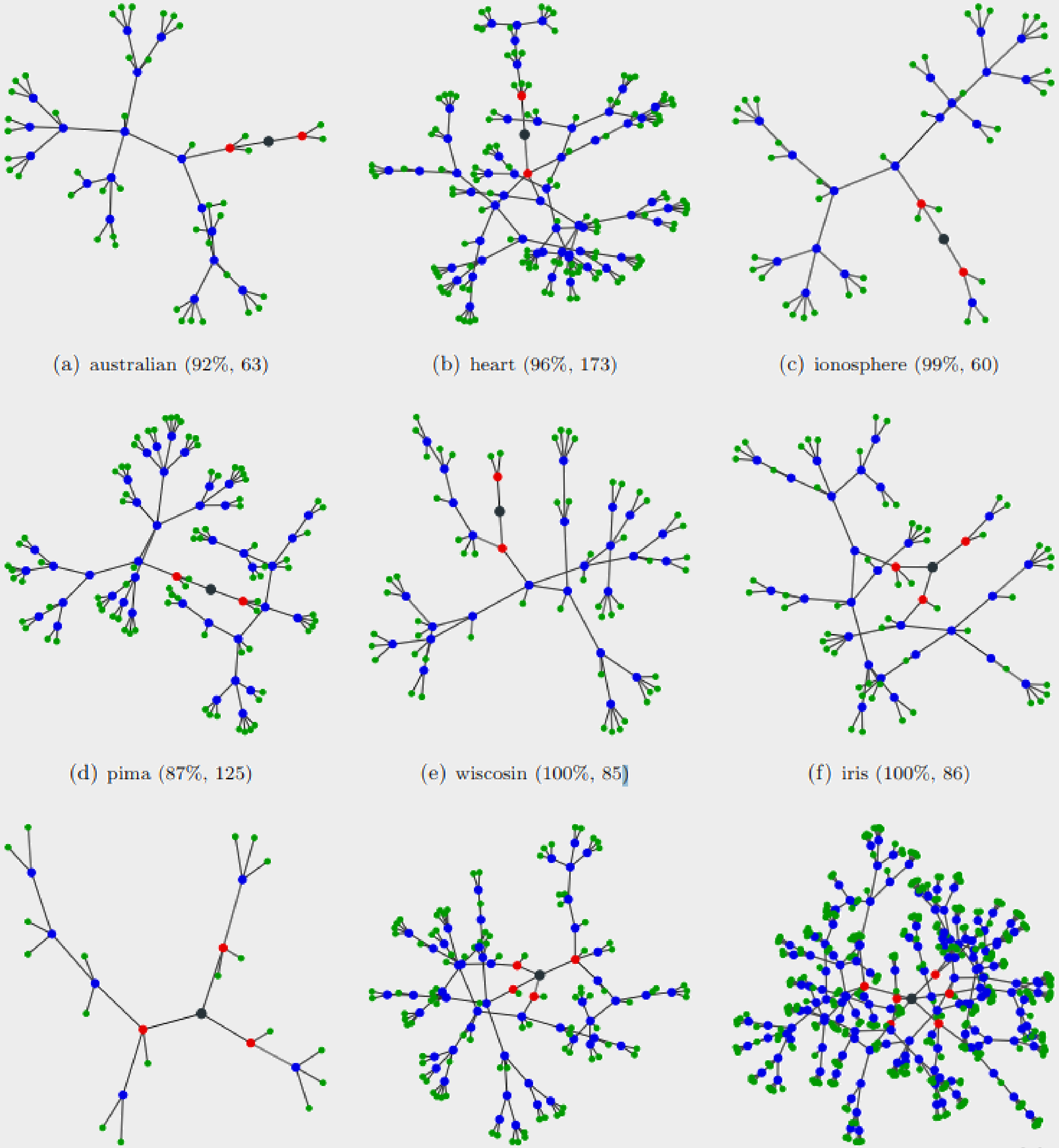
(l) MLP: ReLU, $\eta = 0.1$

Backpropagation Neural Tree: Performance on Classification

Classification results.

Data	BNeuralT	MLP
Aus	0.895	0.876
Hrt	0.897	0.833
Ion	0.952	0.882
Pma	0.822	0.774
Wis	0.986	0.984
Irs	0.992	0.972
Win	0.991	0.991
Vhl	0.75	0.826
Gls	0.732	0.635
Avg. Accuracy	**0.891**	**0.863**
Avg. Weights	**261**	1969

Ojha et al (2022), Neural Networks



(a) australian (92%, 63) (b) heart (96%, 173) (c) ionosphere (99%, 60)

(d) pima (87%, 125) (e) wiscosin (100%, 85) (f) iris (100%, 86)

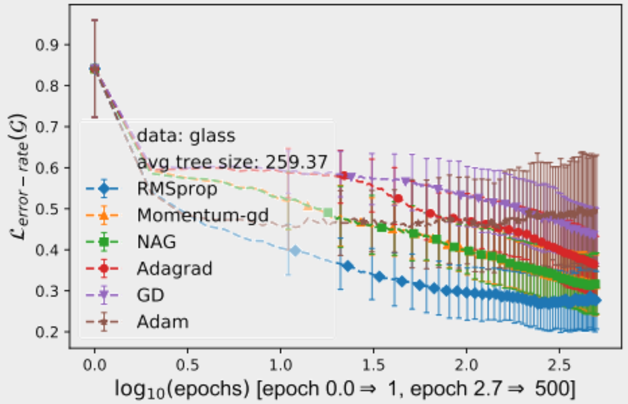
Backpropagation Neural Tree: Performance on Classifcation

Classification results

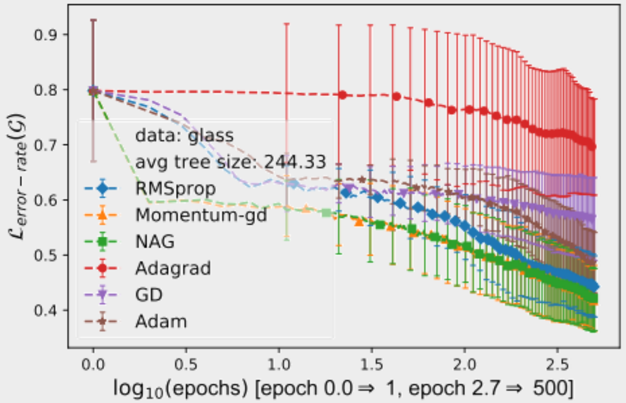
- BNeuralT used **only 13.25% parameters** of MLP
- Accuracy is **2.65% better than the best MLP** result

Neural Tree vs Neural Networks

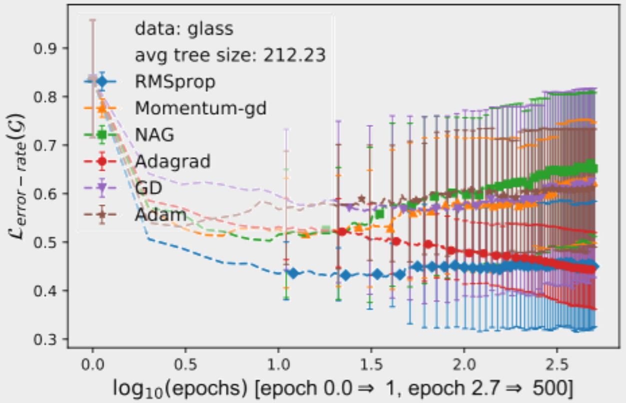
Classification Problems



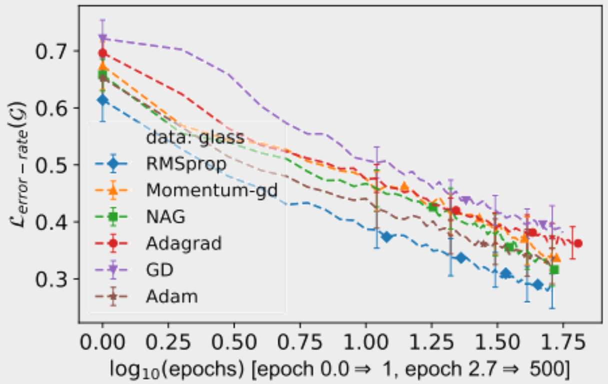
(a) BNeuralT: Sigmod, $\eta = 0.1$



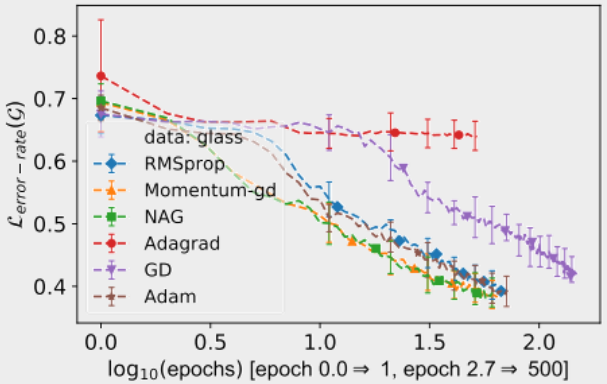
(b) BNeuralT: Sigmod, $\eta = default$



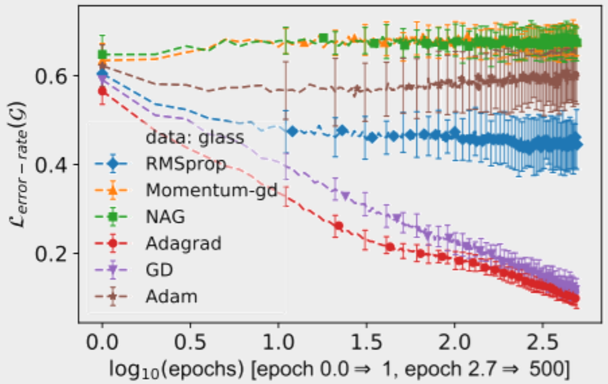
(c) BNeuralT: ReLU, $\eta = 0.1$



(d) MLP: Sigmod, $\eta = 0.1$

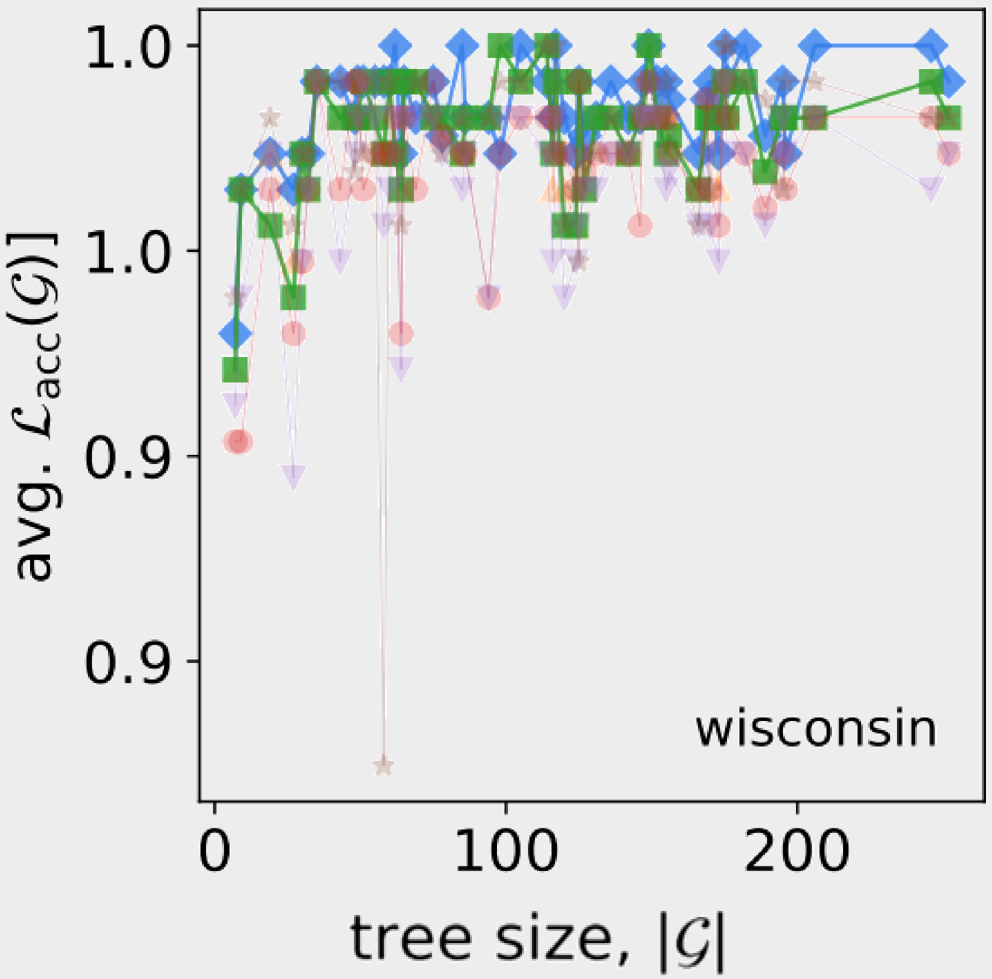


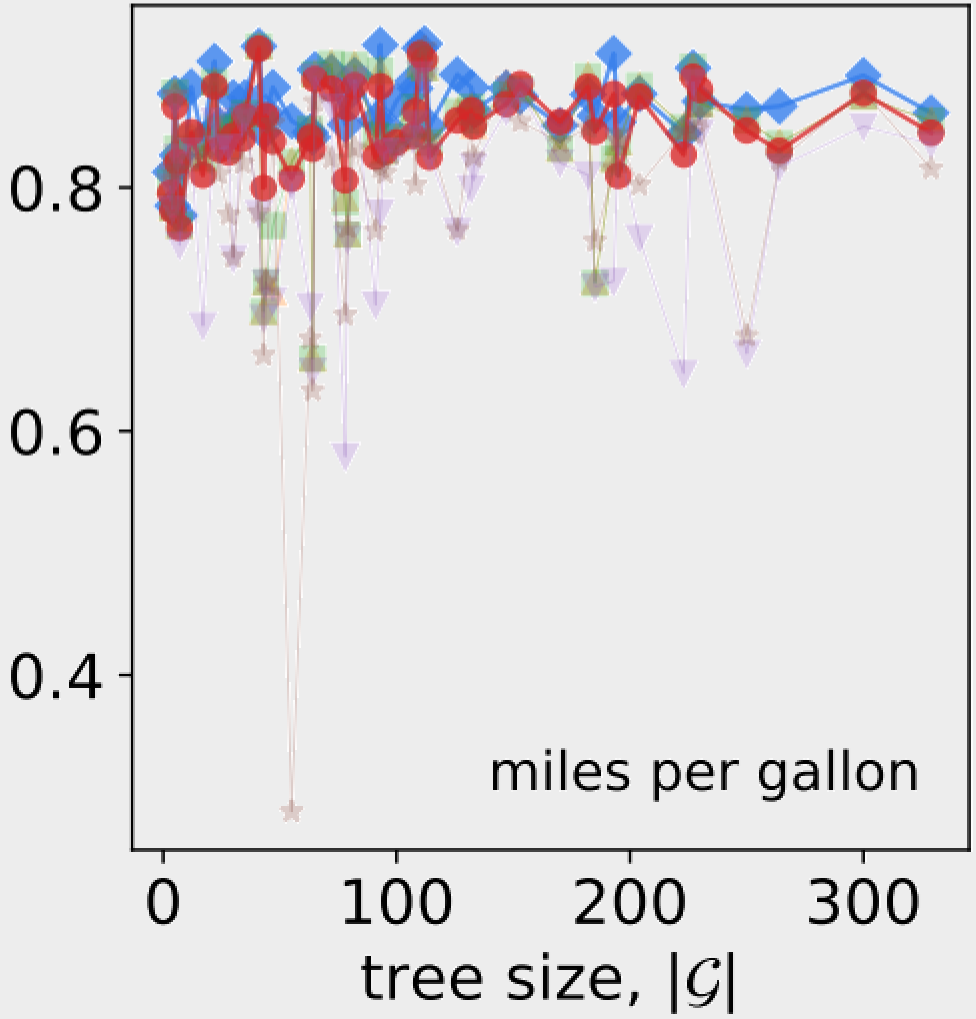
(e) MLP: Sigmod, $\eta = default$



(f) MLP: ReLU, $\eta = 0.1$

Architectural Stochasticity

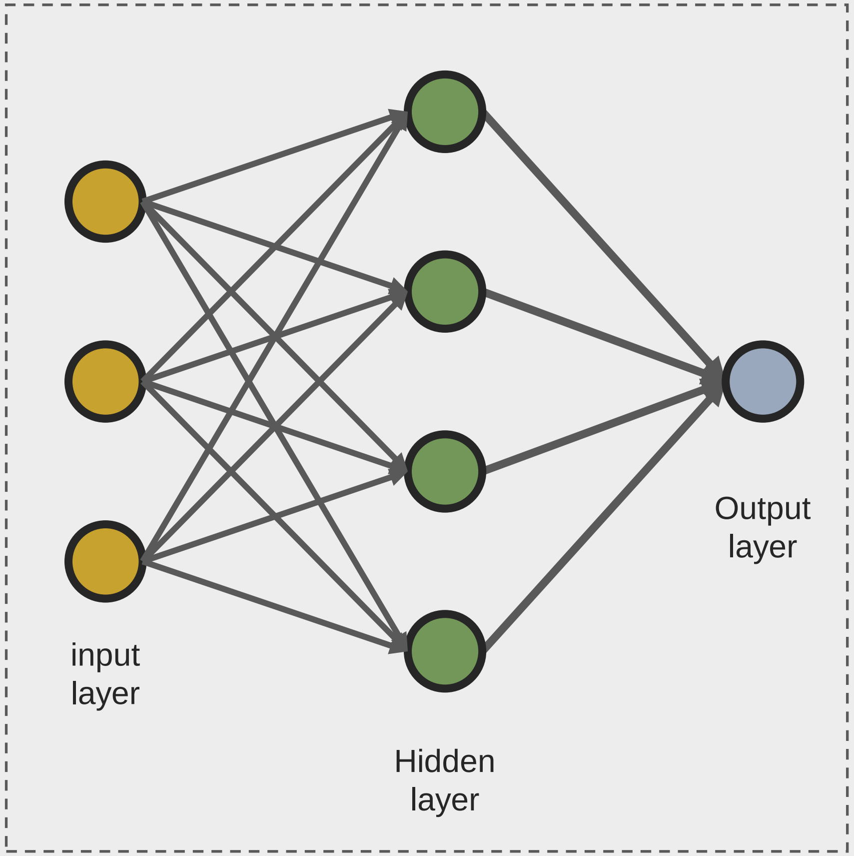




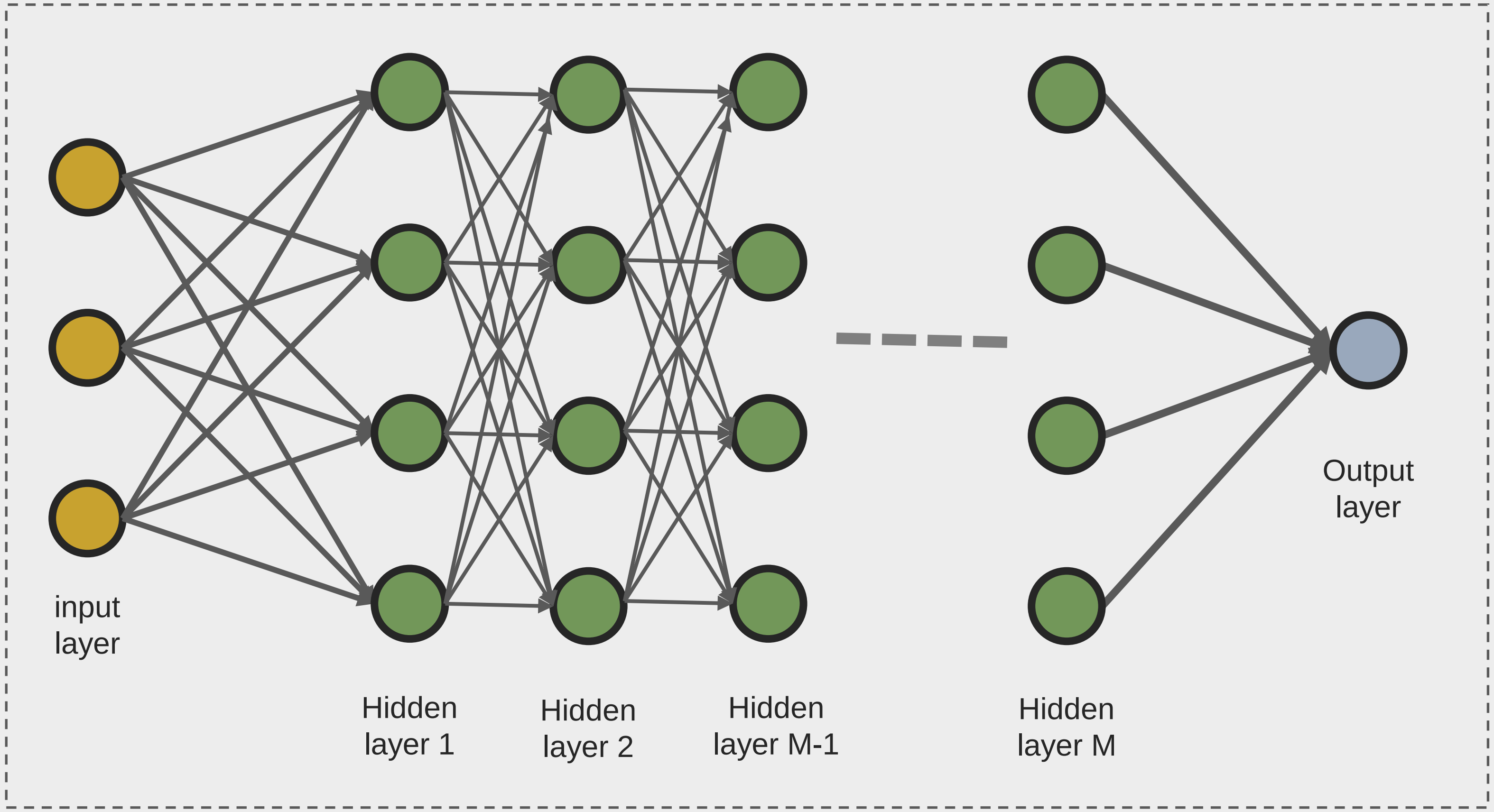
Ojha et al (2022), Neural Networks

Neural Network Architecture

A regular neural network architecture



A deep neural network architecture



SHALLOW LEARNING

DEEP LEARNING

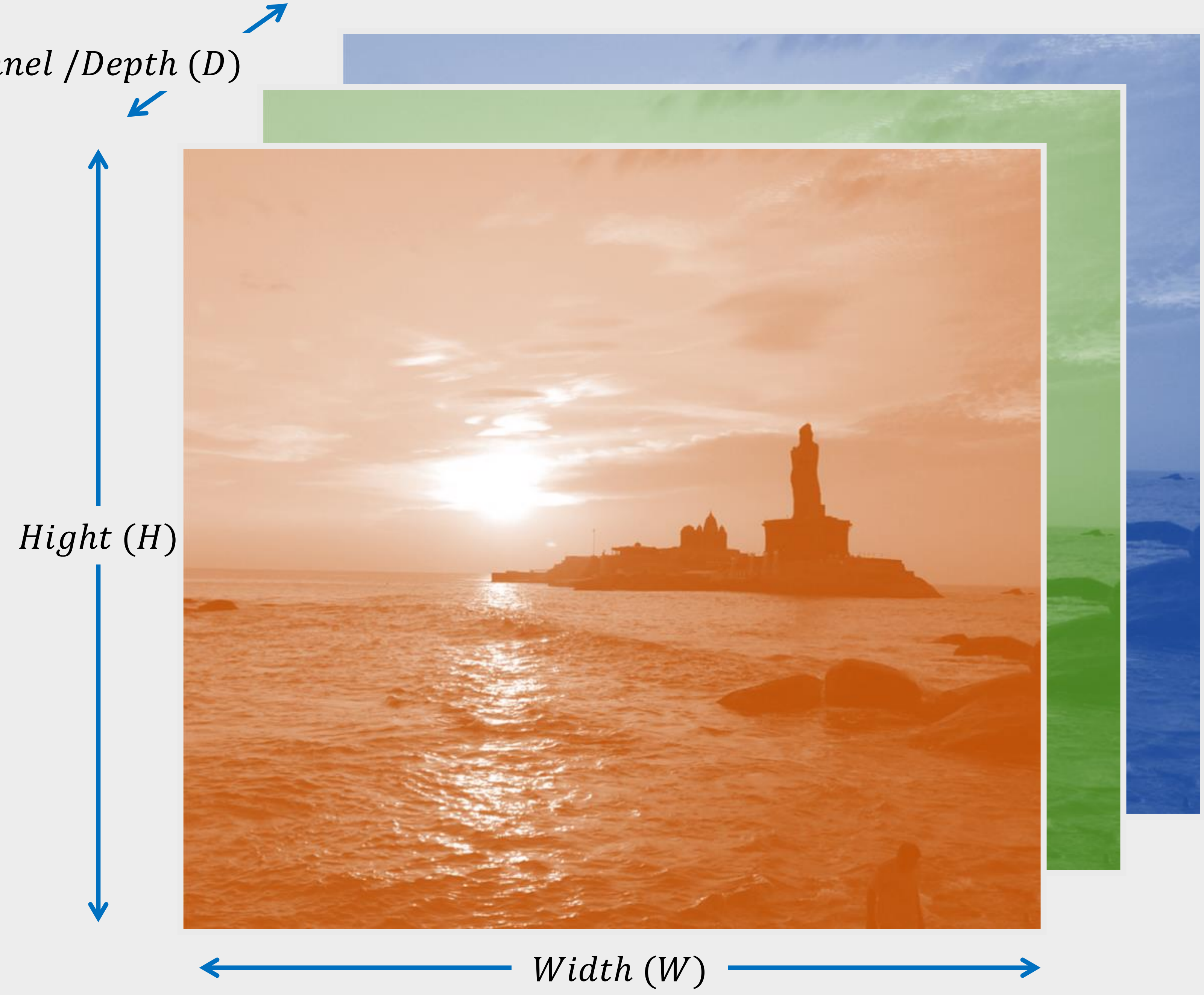
Data

Image: Colour

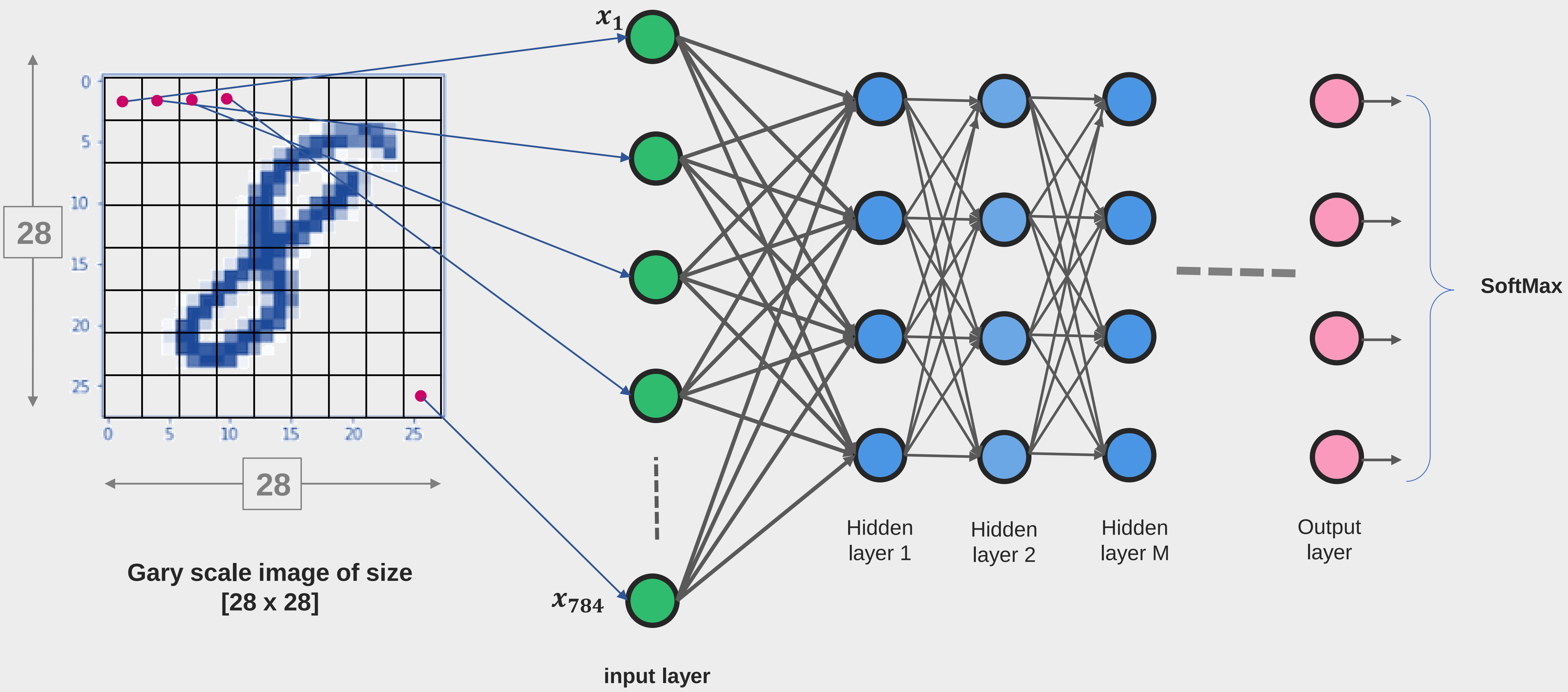
$$I_{RED} = \begin{bmatrix} p_{11} & \cdots & p_{1,W} \\ \vdots & \ddots & \vdots \\ p_{H,1} & \cdots & p_{H}, p_{W} \end{bmatrix}$$

$$I_{Green} = \begin{bmatrix} p_{11} & \cdots & p_{1,W} \\ \vdots & \ddots & \vdots \\ p_{H,1} & \cdots & p_{H}, p_{W} \end{bmatrix}$$

$$I_{Blue} = \begin{bmatrix} p_{11} & \cdots & p_{1,W} \\ \vdots & \ddots & \vdots \\ p_{H,1} & \cdots & p_{H}, p_{W} \end{bmatrix}$$



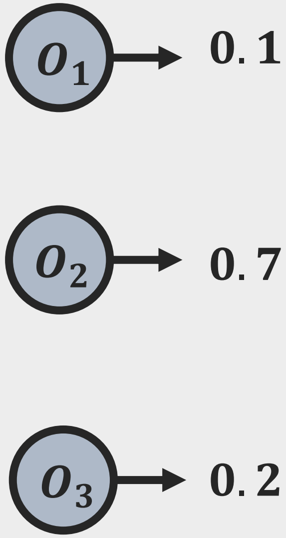
Deep Neural Networks



Gary scale image of size [28 x 28]

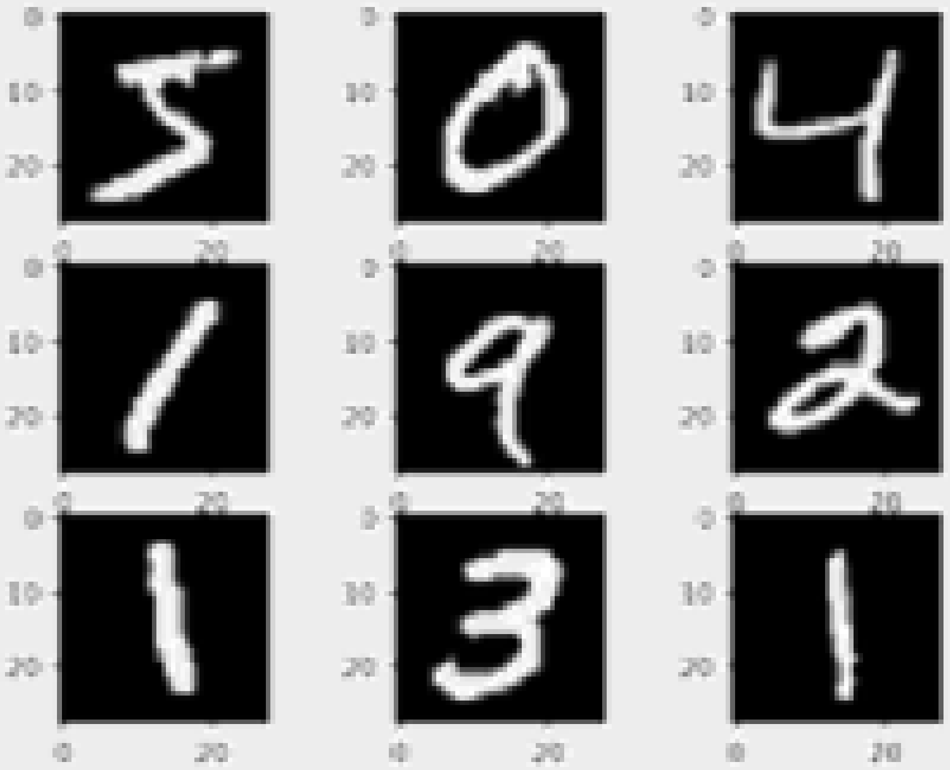
SoftMax Activation

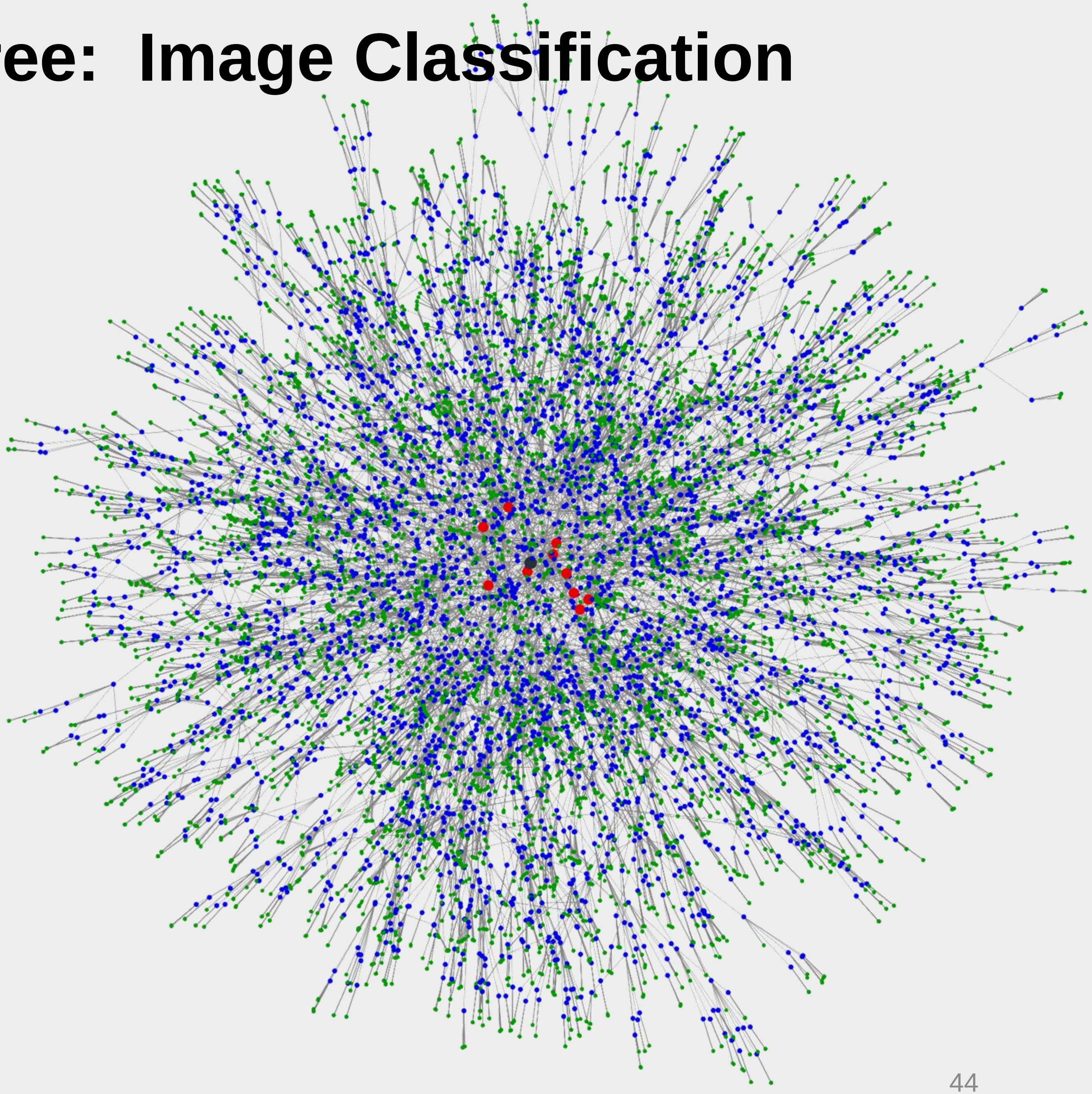
$$\varphi(x_i) = \frac{e^{x_i}}{\sum_i^k e^{x_j}} \text{ for } k \text{ units}$$



PROBABILITIES DISTRIBUTION OF ALL LABELS

Backpropagation Neural Tree: Image Classification



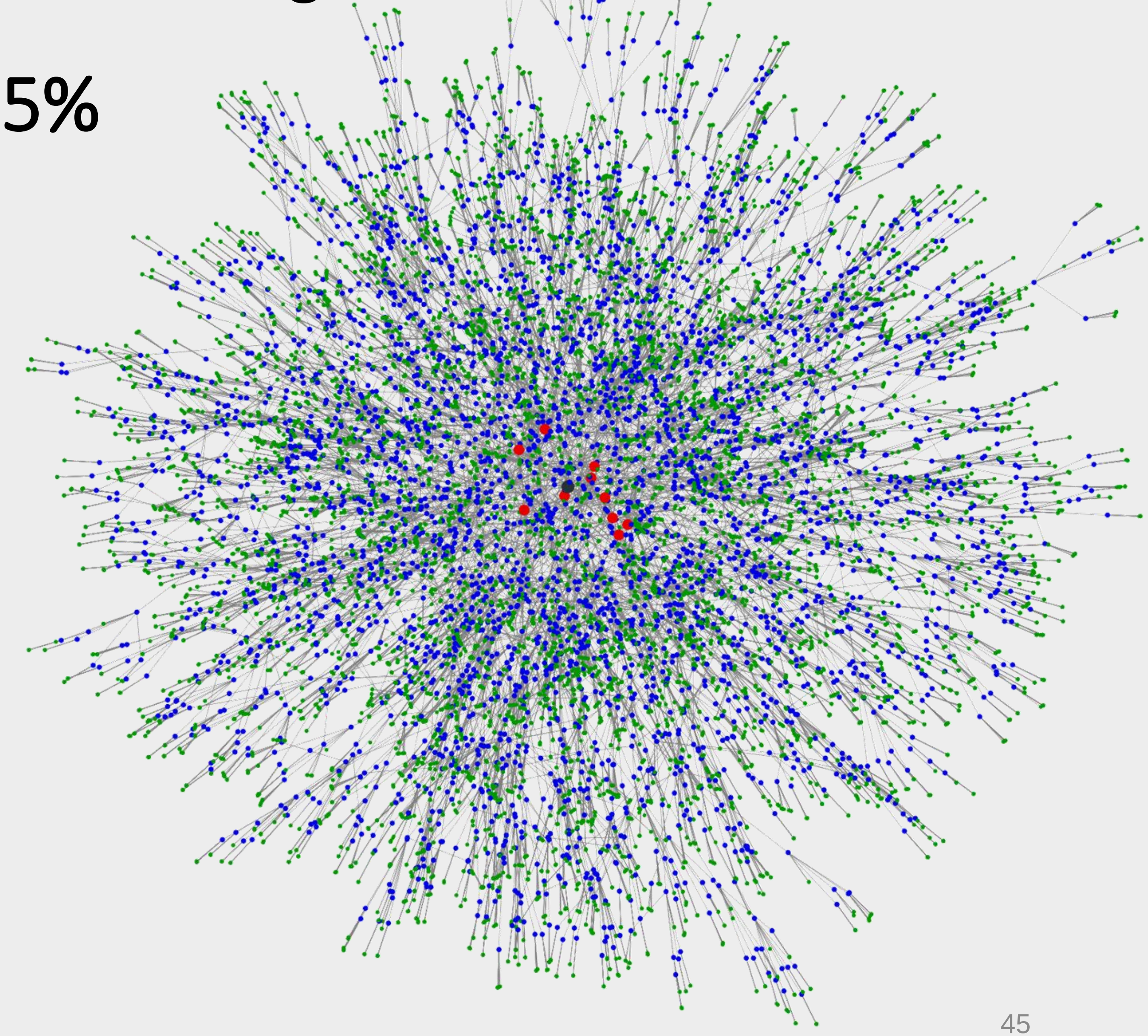


Ojha and Nicosia (2022), *Neural Networks*

Backpropagation Neural Tree: Image Classification

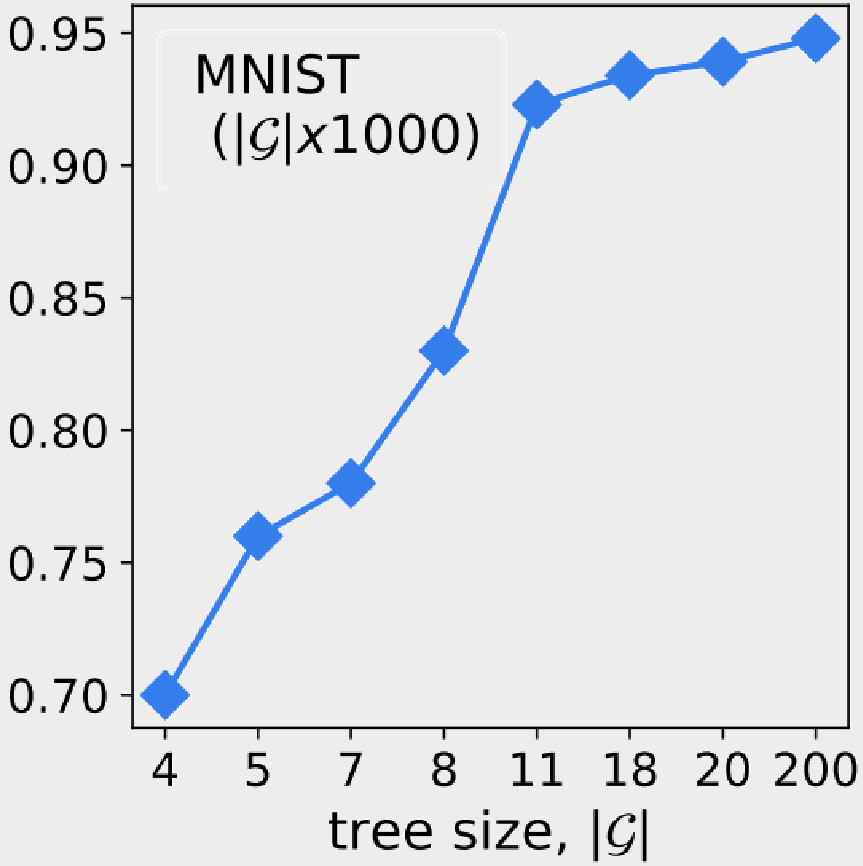
MNIST Model Accuracy ~95%

	Algorithms	Error(%)
BNeuralTs	BNeuralT-10K (pixels)	7.74
	BNeuralT-18K (pixels)	6.58
	BNeuralT-20K (pixels)	6.08
	BNeuralT-200K† (pixels)	**5.19**
Classification Trees	GUIDE (pixels, oblique split)	26.21
	OC1 (pixels, oblique split)	25.66
	GUIDE (pixels)	21.48
	CART-R (pixels)	11.97
	CART-P (pixels)	11.95
	C5.0 (pixels)	11.69
	TAO (pixels)	11.48
	TAO (pixels, oblique split)	5.26



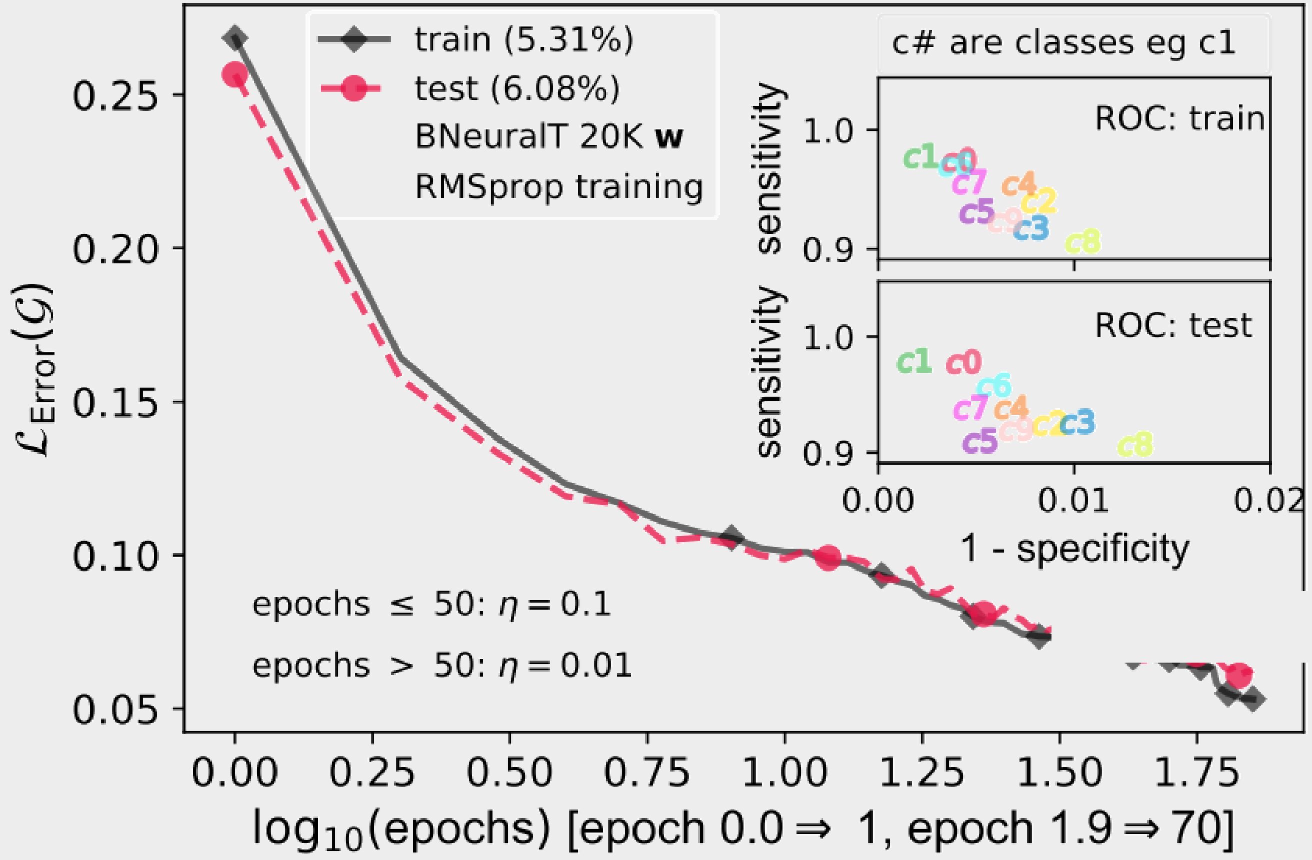
Model Size vs Accuracy

	Algorithms	Error(%)
BNeuralTs	BNeuralT-10K (pixels)	7.74
	BNeuralT-18K (pixels)	6.58
	BNeuralT-20K (pixels)	6.08
	BNeuralT-200K† (pixels)	**5.19**
Classification Trees	GUIDE (pixels, oblique split)	26.21
	OC1 (pixels, oblique split)	25.66
	GUIDE (pixels)	21.48
	CART-R (pixels)	11.97
	CART-P (pixels)	11.95
	C5.0 (pixels)	11.69
	TAO (pixels)	11.48
	TAO (pixels, oblique split)	5.26

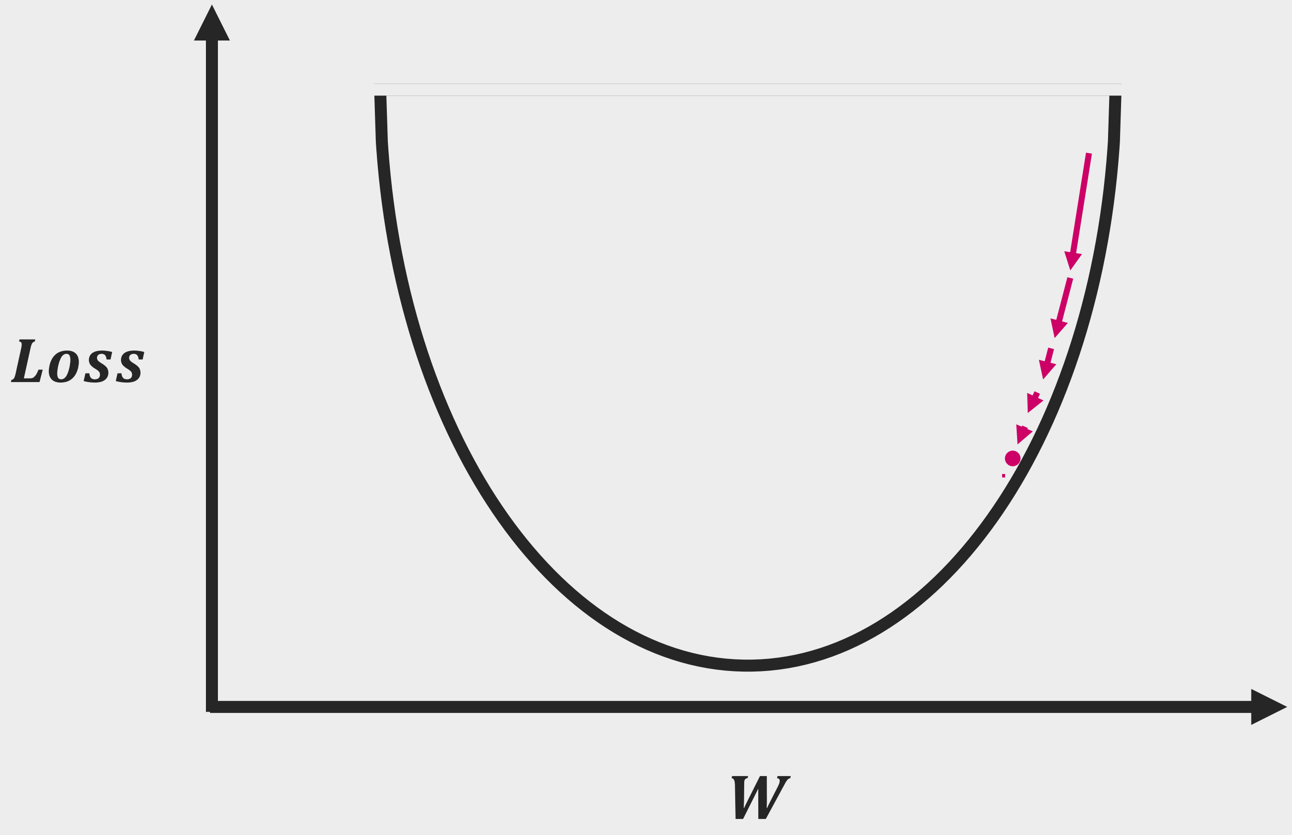


Learnability of different Classes

True positive rate Vs False Positive Rate (1 – True negative rate)

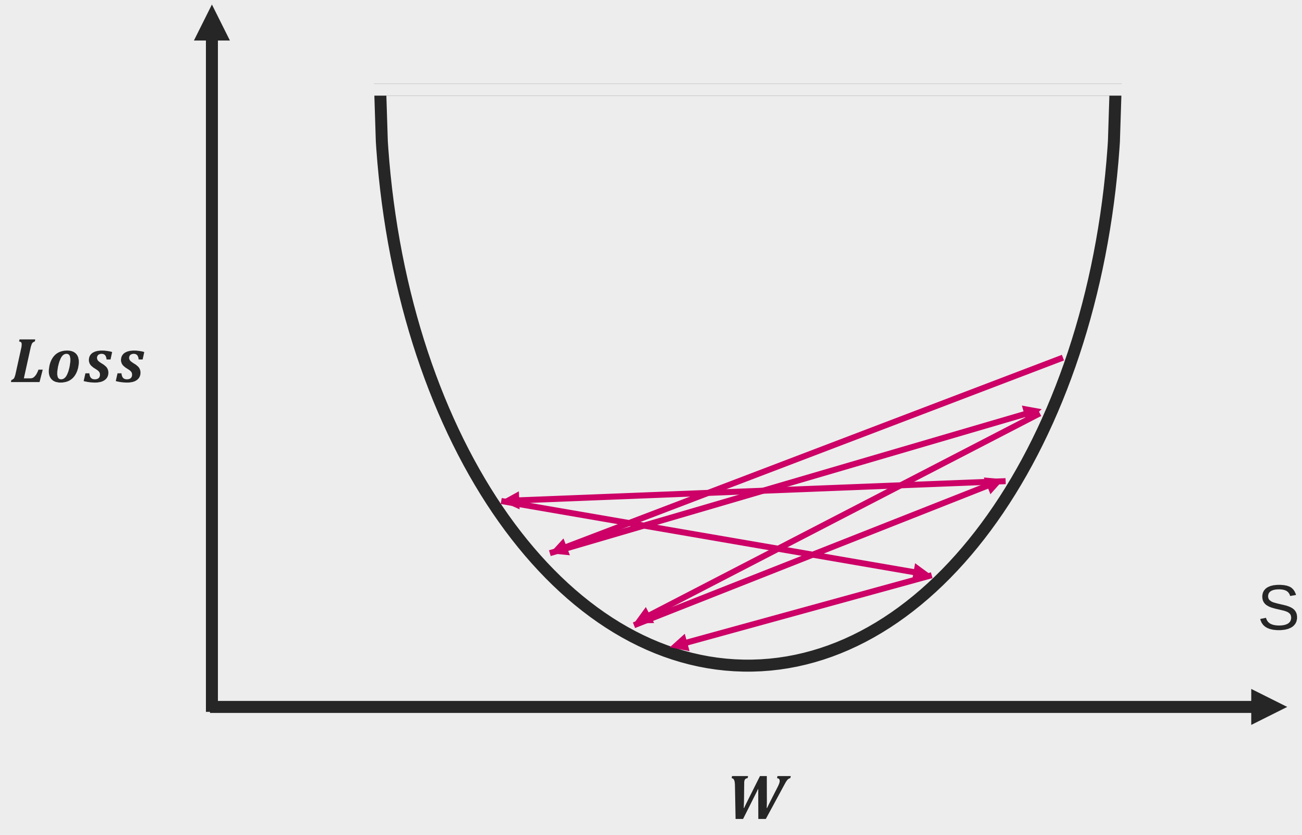


Neural Tree Learning Scheme: Slow learning rate



Convergence virtually stops because weights do not change much

Neural Tree Learning Scheme: Very fast learning rate



Highly fluctuating gradient descent

Weight abruptly changes.

Skips Global minima

Summary

stochastic gradient descent training of any a priori arbitrarily “thinned” network has the potential to solve machine learning tasks with an equivalent or better degree of accuracy than a fully connected symmetric and systematic neural network architecture.

References

- Ojha, V., & Nicosia, G. (2022). Backpropagation neural tree. Neural Networks, 149, 66-83. URL: https://arxiv.org/abs/2202.02248 Code: https://github.com/vojha-code/bneuralt

- Ojha, V., & Nicosia, G. (2020). Multi-objective optimisation of multi-output neural trees. In 2020 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE. URL: https://arxiv.org/abs/2010.04524 Code: https://github.com/vojha-code/multi-output-neural-tree

- Ojha, V. K., Abraham, A., & Snášel, V. (2017). Ensemble of heterogeneous flexible neural trees using multiobjective genetic programming. Applied Soft Computing, 52, 909-924.

- Ojha, V. K., Snášel, V., & Abraham, A. (2017). Multiobjective programming for type-2 hierarchical fuzzy inference trees. IEEE Transactions on Fuzzy Systems, 26(2), 915-936.

v.k.ojha@reading.ac.uk; vkojha@ieee.org

Github: https://github.com/vojha-code