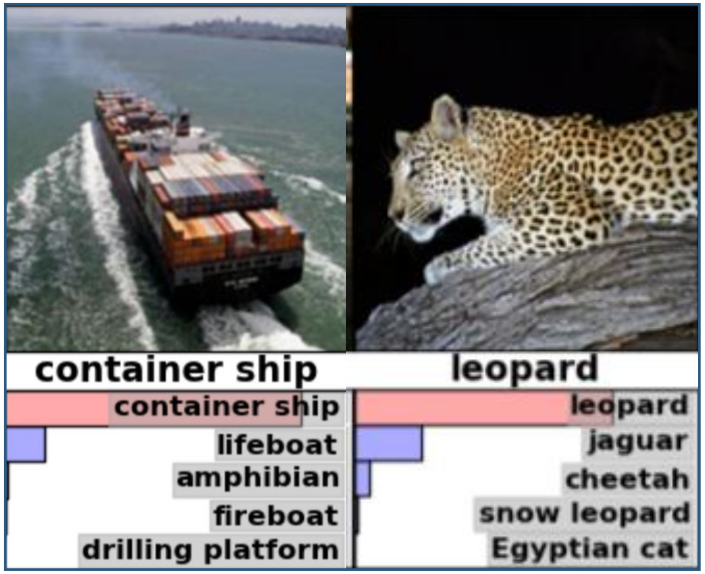
Parsimonious neural networks learn interpretable physical laws

Saaketh Desai[1], Alejandro Strachan[2]

1. Centre for Integrated Nanotechnologies, Sandia National Laboratories

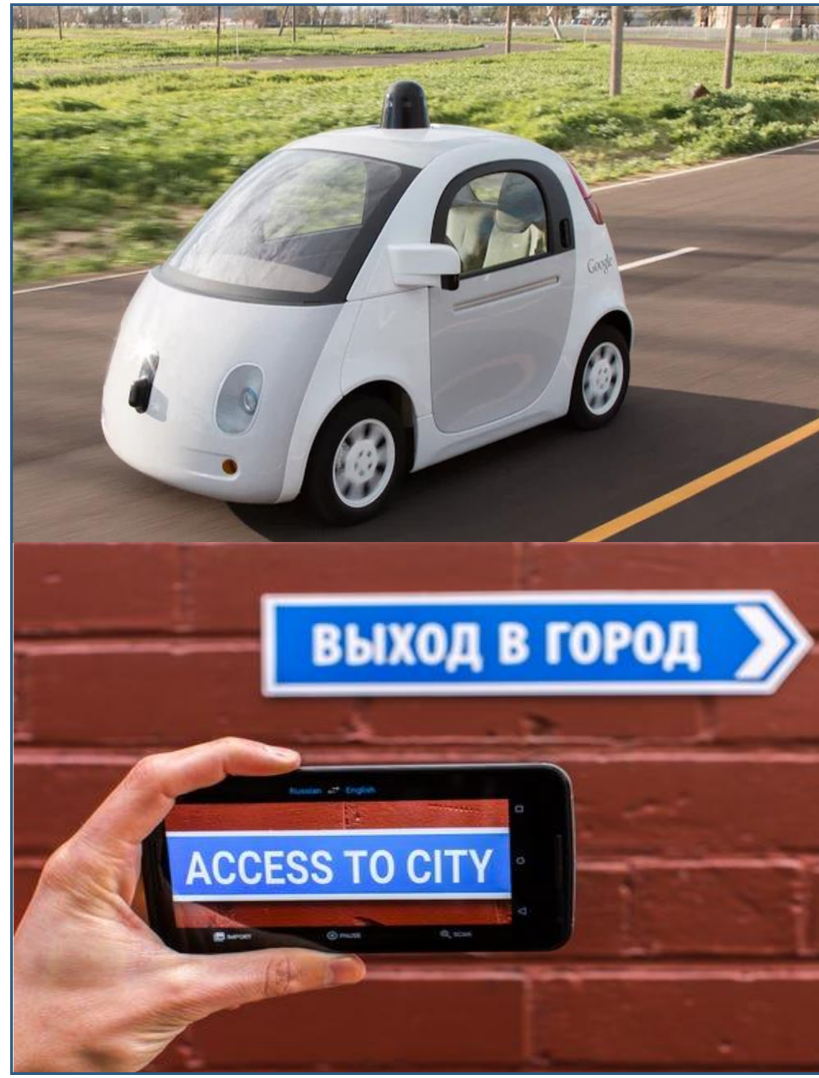
2. School of Materials Engineering, Purdue University

Machine learning models and their applications



Krizhevsky et al., *Advances in neural information processing systems,* (2012)

Taken from wired.com

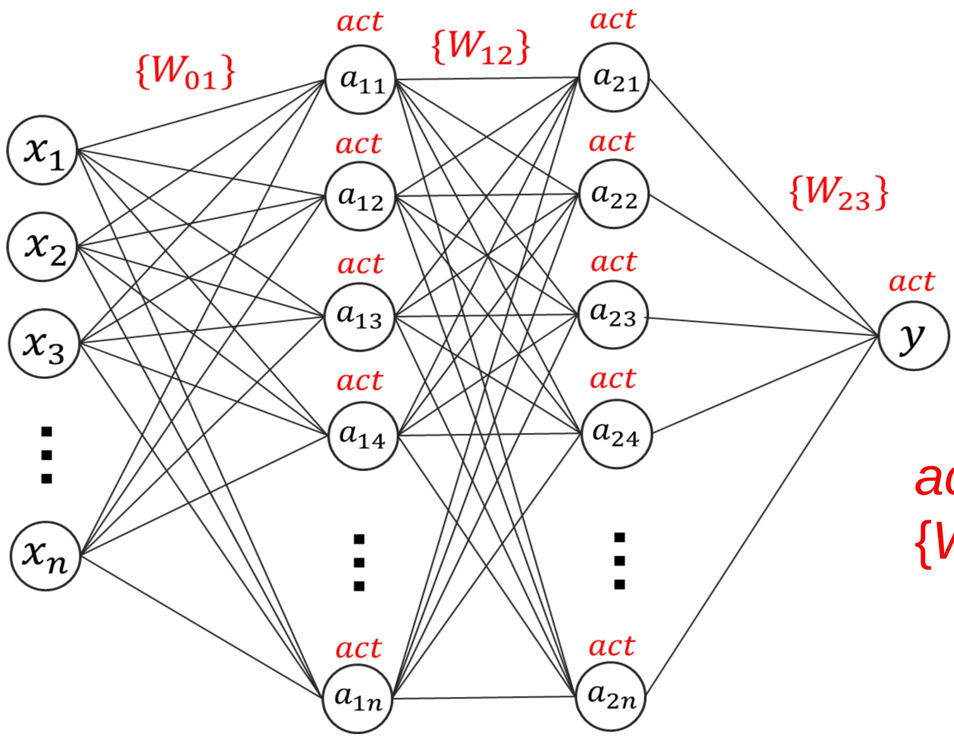


Taken from businessinsider.com

Machine learned models excel when data is plentiful

Encoding neural networks for genetic algorithms

Couple neural networks with genetic algorithms to balance interpretability and accuracy



Individual

[1,0,1,2,…, 2]

act: {linear, squared, tanh, relu, …}

$\{W_{ij}\}$: {0, 1, ½, 2, …, trainable}

linear: 0
relu: 1
tanh: 2

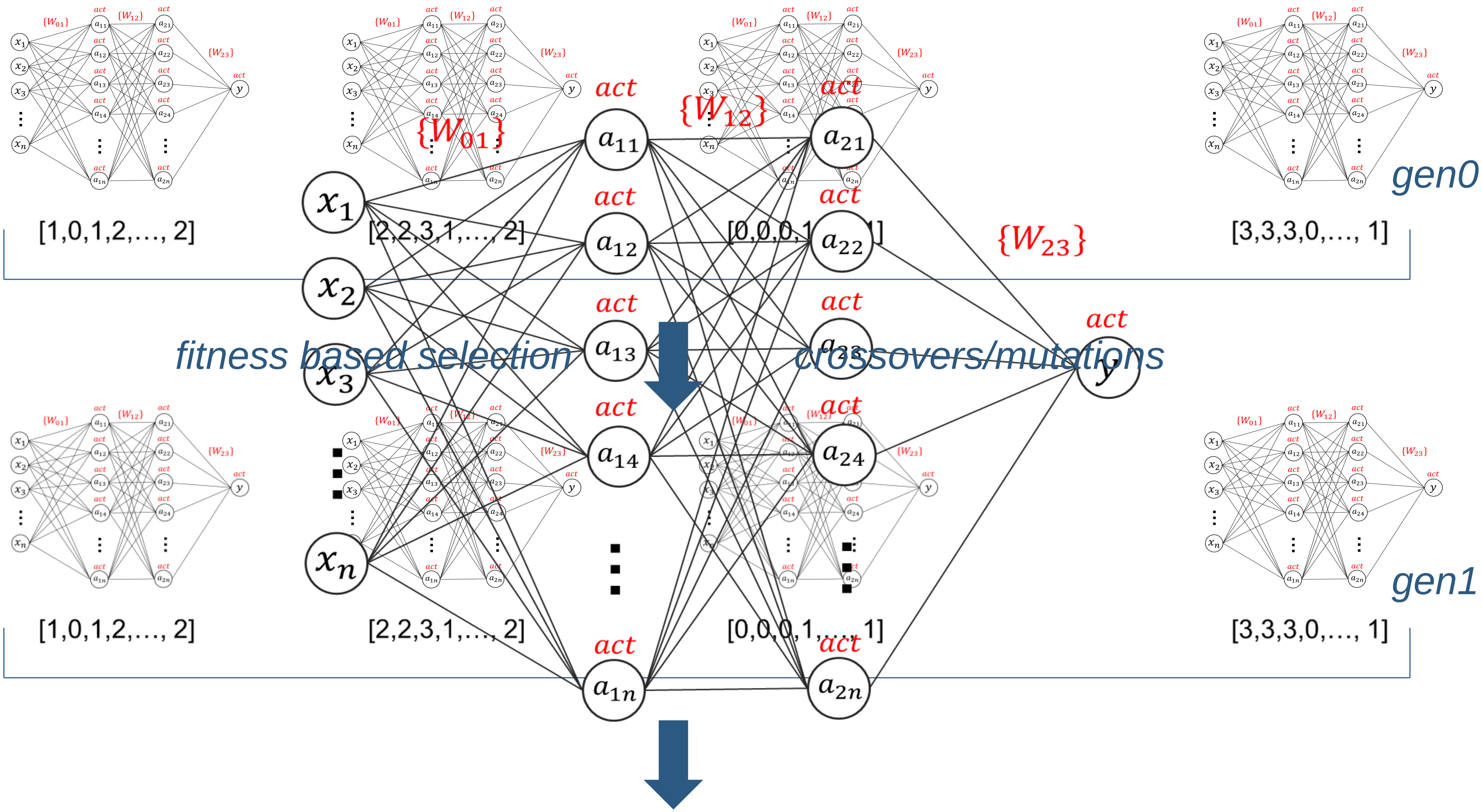
0: 0
1, ½ , 2: 1
trainable: 2

$$F = f_1(E_{test}) + p\left(\Sigma_{i=1}^{n_a} w_i^2 + \Sigma_{j=1}^{n_w} f_2(w_j)\right)$$

fitness — error on data — parsimony coefficient — simple activations — weight penalty

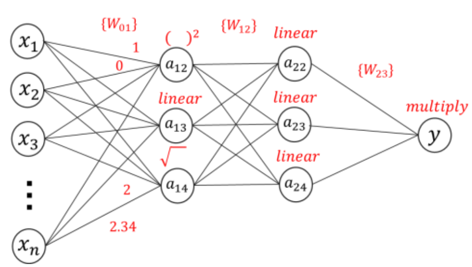
Keras

How to train a PNN?



DISTRIBUTED EVOLUTIONARY ALGORITHMS IN PYTHON

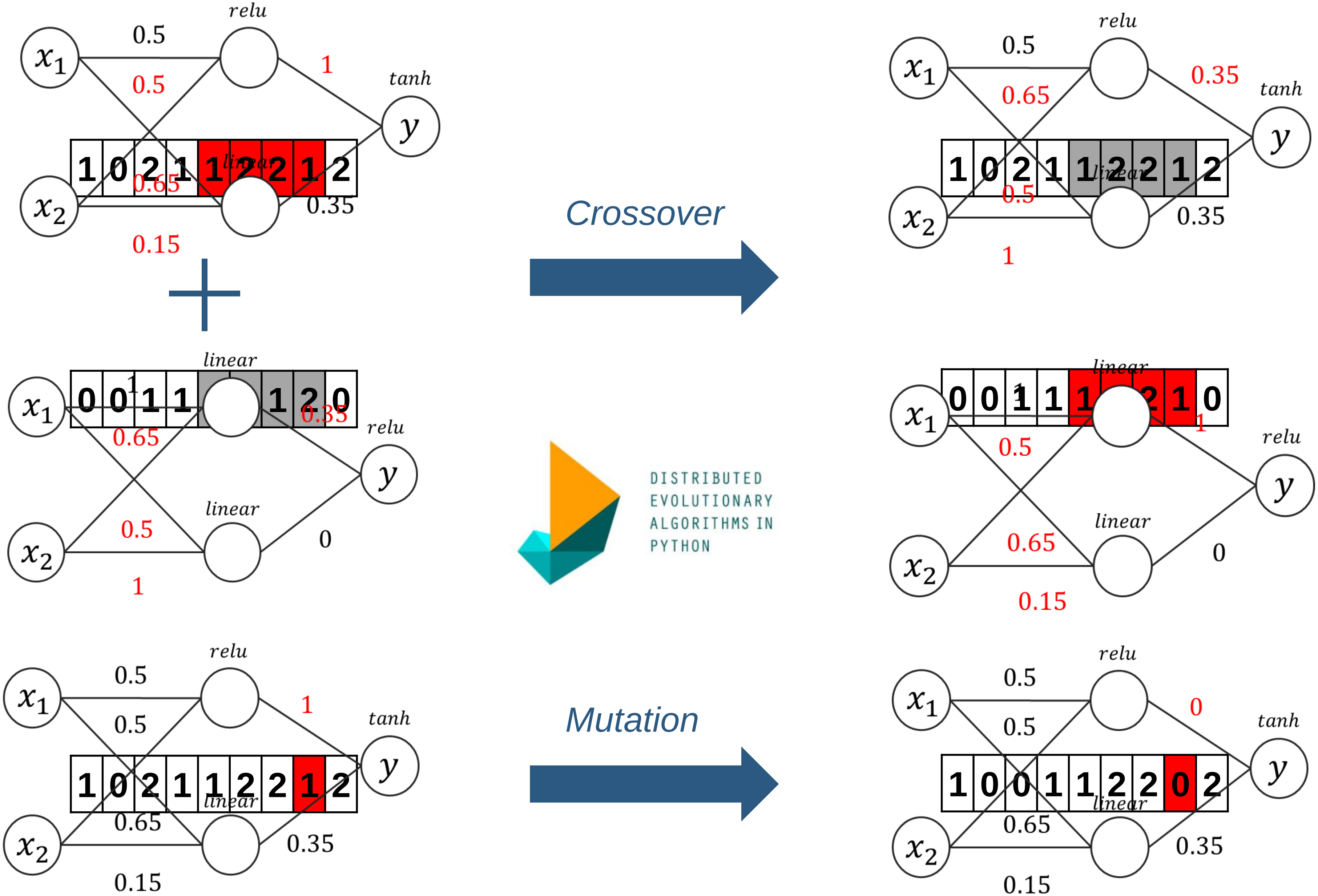
Fittest individual



Interpretable equation

genN

Genetic operations on neural networks



Parsimonious neural networks – melting point

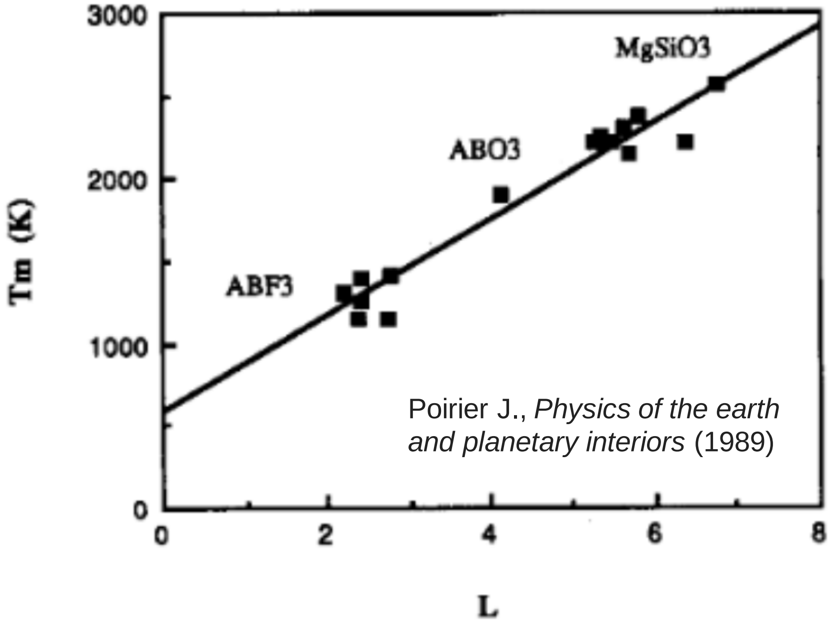
Material	Volume	Density	Bulk modulus	Shear Modulus	Melting temp
BaZrO3	30.700078	5.983317	143.0	88.0	2813.15
Nd2O3	139.781930	6.395583	124.0	51.0	2543.15
BaO2	17.382291	5.391927	67.0	35.0	723.15

Can we predict the melting temp based on fundamental inputs?

$$T_m^{lind} = \left(\frac{4\pi^2}{9h^2}\right) f^2\, a^2\, m\, T_D^2$$

fitting constant (f), interatomic spacing (a), mean atomic mass (m), Debye temperature (T_D)

Lindemann law developed in 1910



Poirier J., *Physics of the earth and planetary interiors* (1989)

Can PNNs learn improved descriptions of melting from data?

Dimensional analysis on inputs

$$\theta_0 = \frac{\hbar v_m}{k_b a} \qquad \theta_1 = \frac{\hbar^2}{m a^2 k_b} \qquad \theta_2 = \frac{a^3 G}{k_b} \qquad \theta_3 = \frac{a^3 K}{k_b}$$

Temperature units

$$v_s = \sqrt{\frac{G}{\rho}} \qquad v_p = \sqrt{\frac{K + \frac{4}{3}G}{\rho}} \qquad v_m = \left[\frac{3}{\left(\frac{1}{v_p}\right)^3 + 2\left(\frac{1}{v_s}\right)^3}\right]^{\frac{1}{3}}$$

<u>Dimensionless inputs</u>

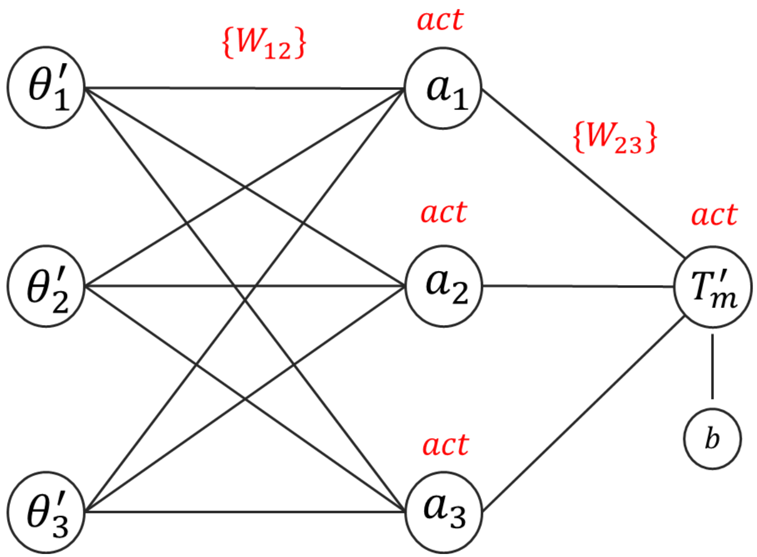
$$\theta_1' = \frac{\hbar}{m a v_m} = \frac{\theta_1}{\theta_0}$$

$$\theta_2' = \frac{a^4 G}{\hbar v_m} = \frac{\theta_2}{\theta_0}$$

$$\theta_3' = \frac{a^4 K}{\hbar v_m} = \frac{\theta_3}{\theta_0}$$

act: {linear, multiply, squared, tanh, …}

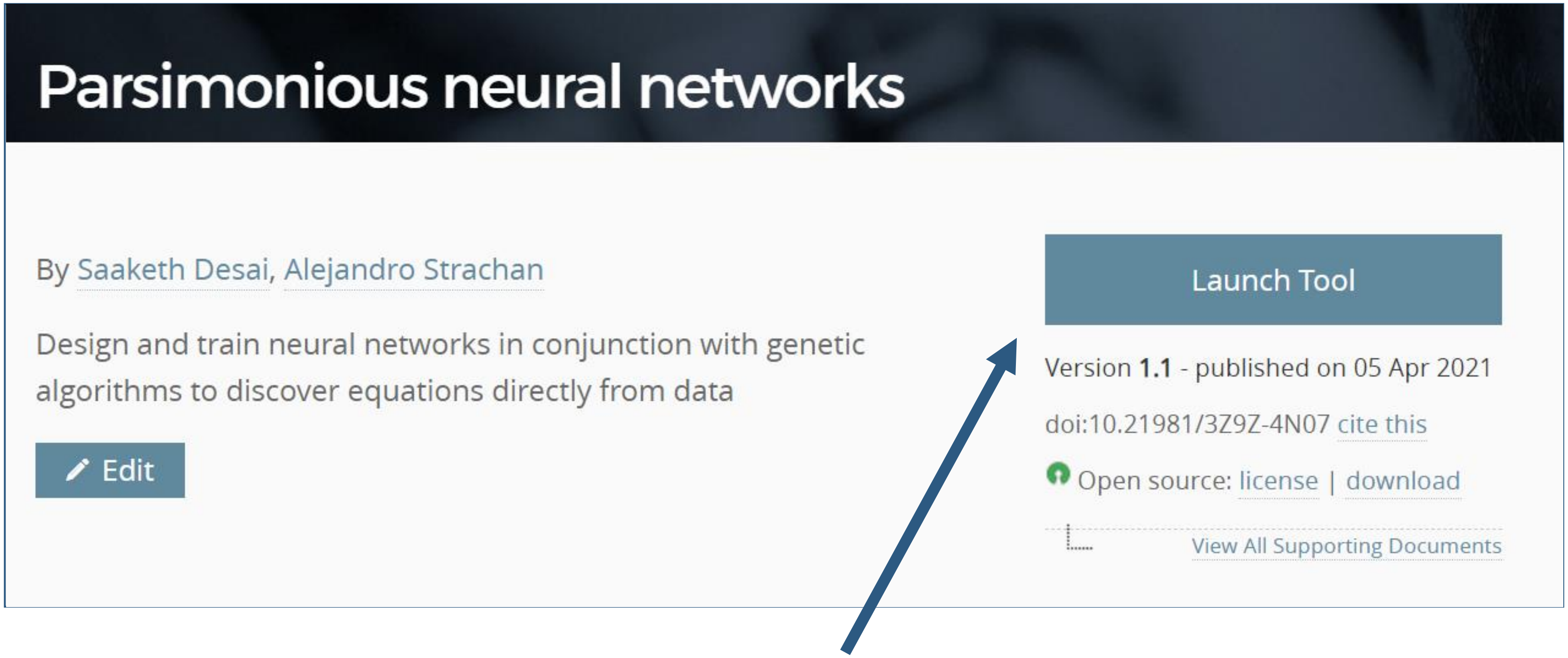
$\{W_{ij}\}$: {0, 1, …, trainable}



Launching the nanoHUB tool

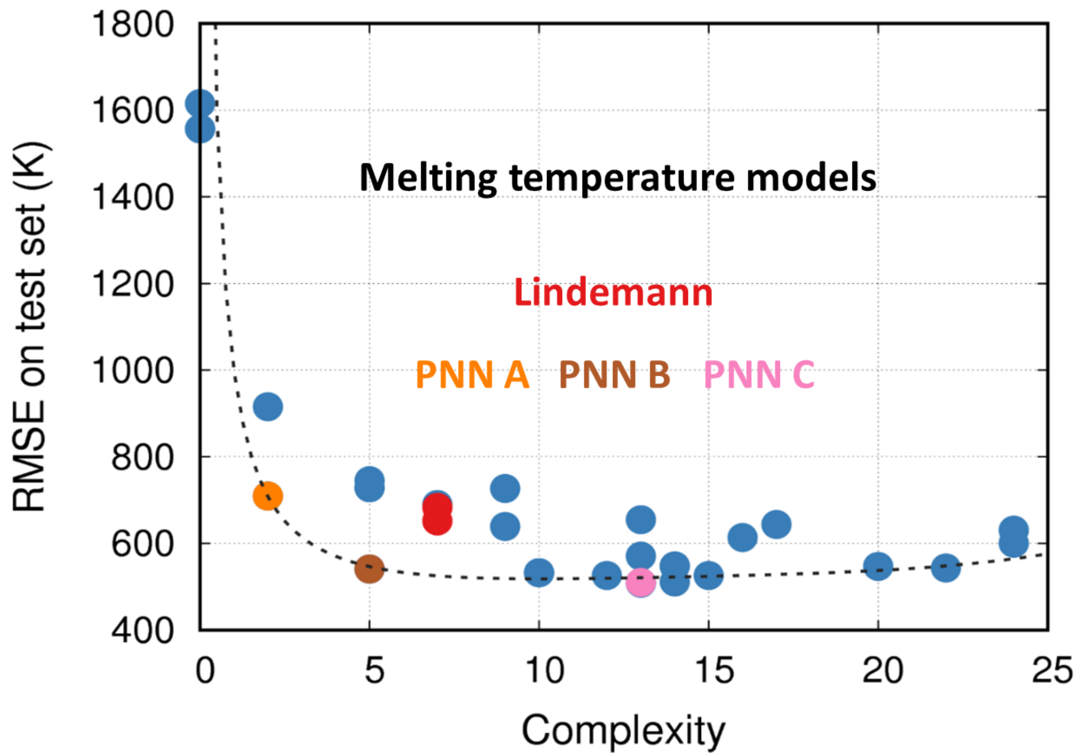
Parsimonious neural networks

From your browser go to link: https://nanohub.org/tools/pnndemo/



Click on Launch Tool to begin

Discovering melting point laws



Melting temperature models

$$T_m^{PNN\ A} = 21.8671\ \theta_0$$

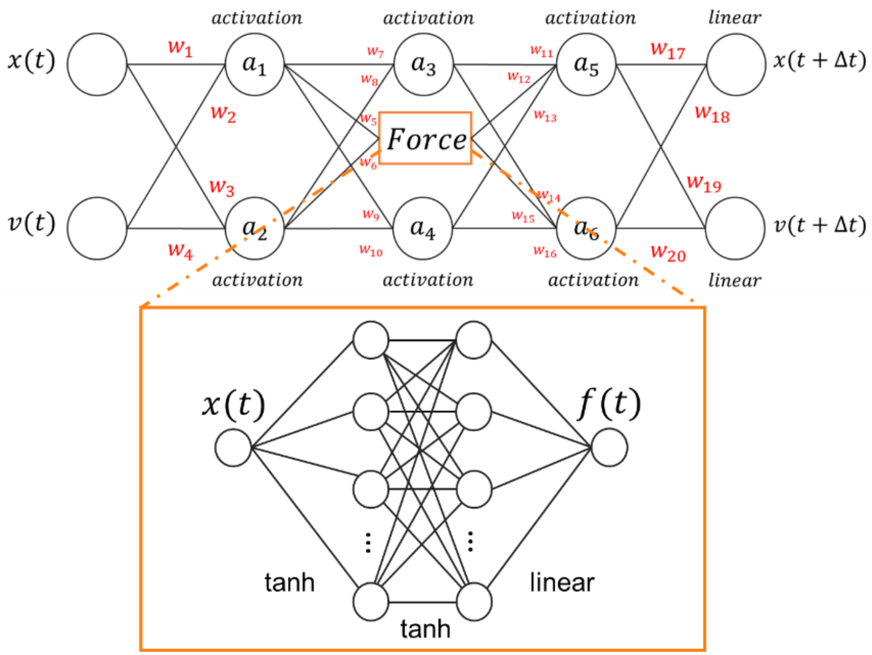
$$T_m^{lind} = \frac{k_b}{9\hbar^2} f^2 a^2 m T_D^2 = C\frac{\theta_0^2}{\theta_1}$$

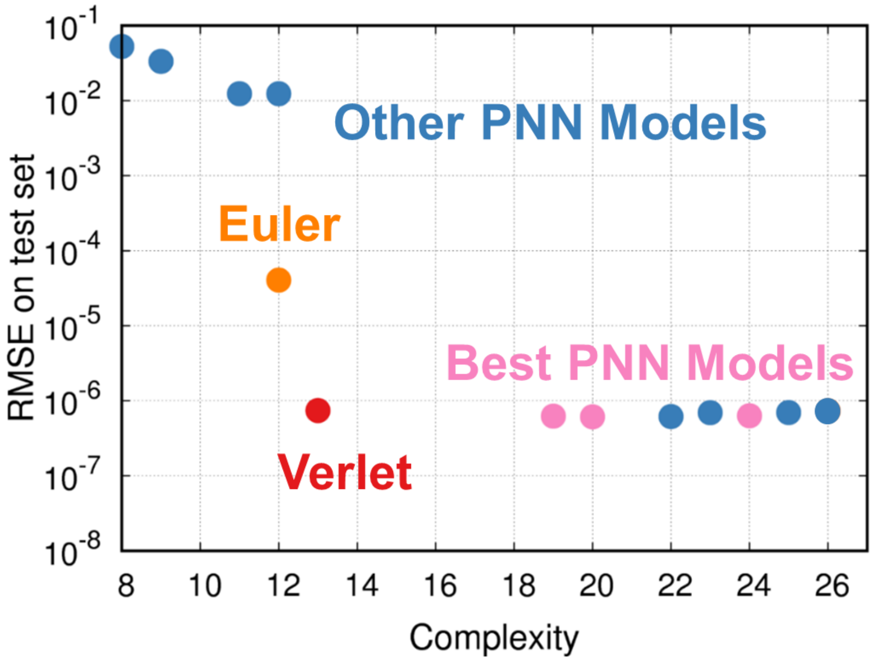
$$T_m^{PNN\ B} = 17.553\ \theta_0 + 0.00198\ \theta_2$$

$$T_m^{PNN\ C} = 11.903\ \theta_0 + 0.0005\ \theta_3 + 0.008\frac{\theta_0^2}{\theta_1}$$

Parsimonious neural networks learn non-linear interpretable laws

Discovering integration schemes from data





$$x(t+\Delta t) = x(t) + 1.0001\, v(t)\Delta t + 0.9997\, \frac{1}{2} f\left(x(t) + v(t)\frac{\Delta t}{2}\right)\frac{\Delta t^2}{m}$$

$$v(t+\Delta t) = v(t) + 0.9997 f\left(x(t) + v(t)\frac{\Delta t}{2}\right)\frac{\Delta t}{m}$$

Position Verlet integration scheme

Parsimonious neural networks learn underlying physics directly from data