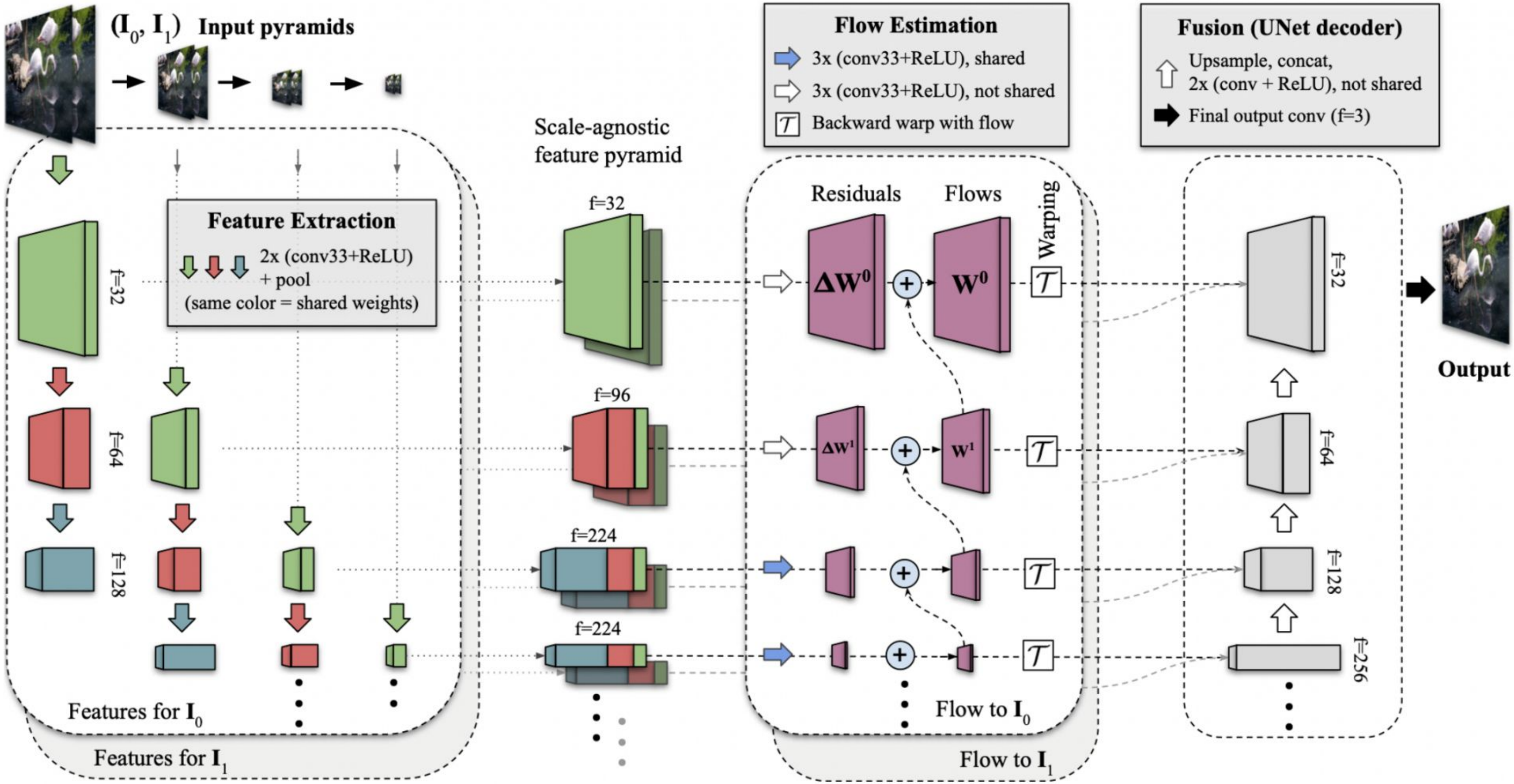
ECS 171: Machine Learning

Summer 2023

Edwin Solares
easolares@ucdavis.edu

Neural Networks

FILM Architecture Overview

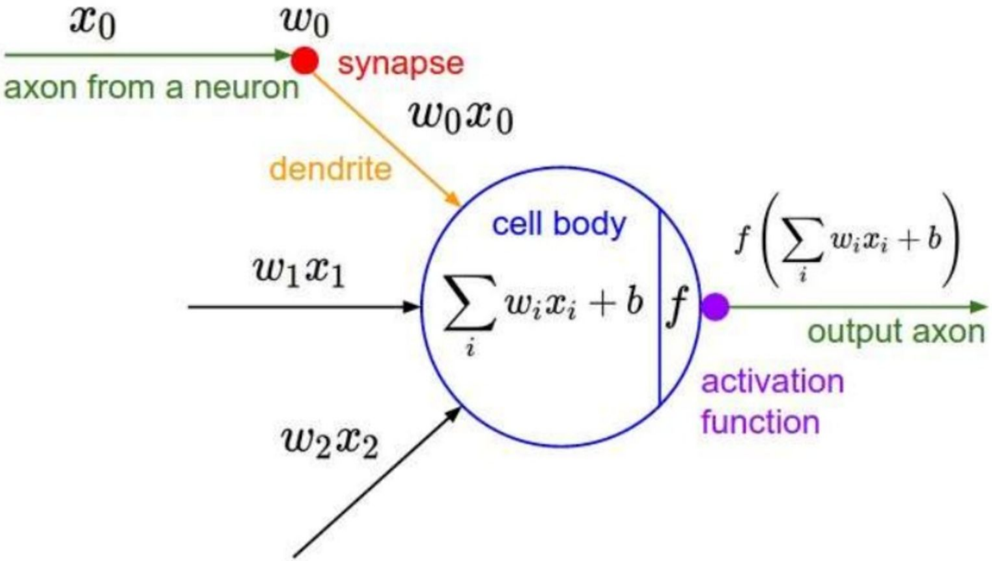


DNN and Drone

Open Source Computer Vision and Drones

Artificial Neuron History

2nd Generation Neuron



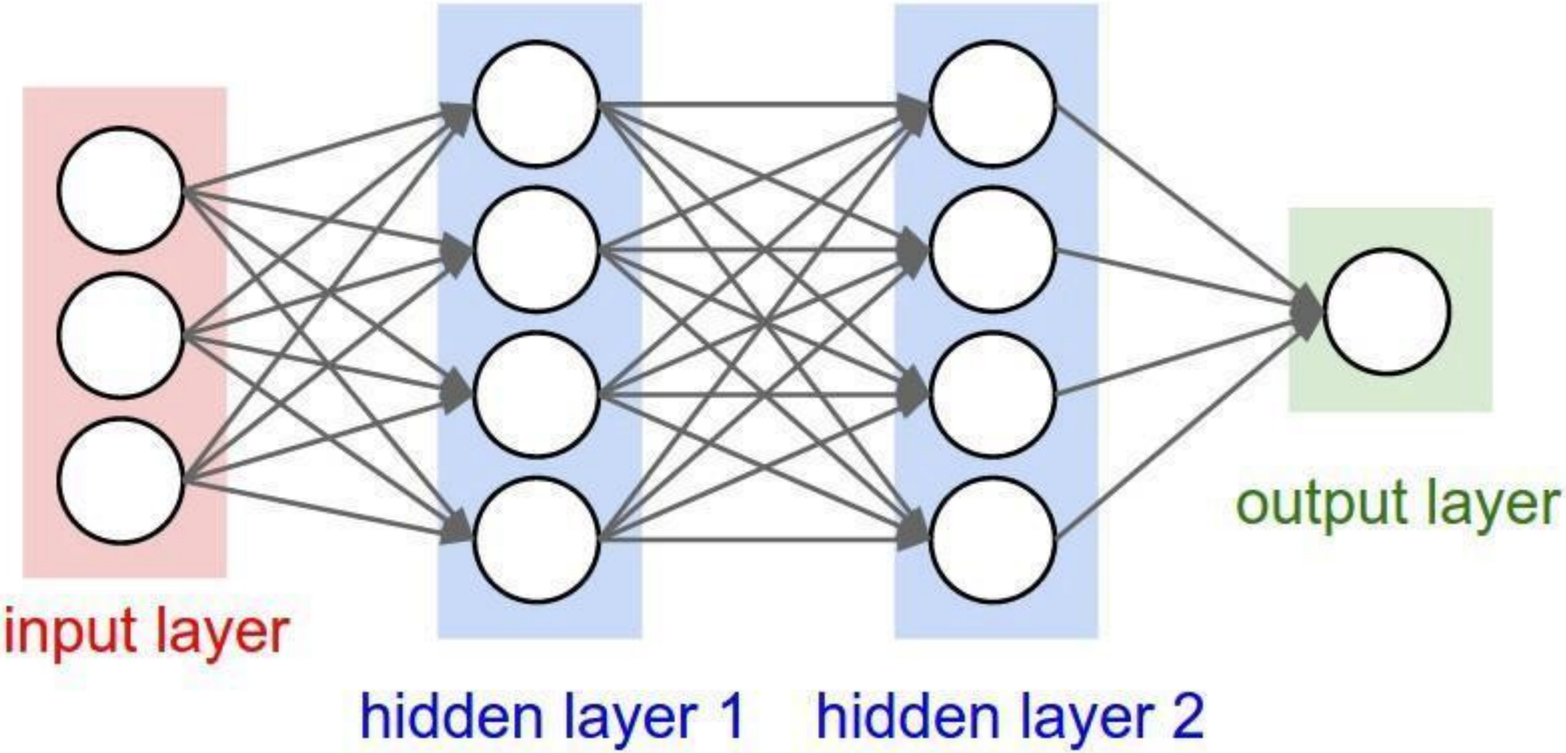
https://www.slideshare.net/hitechpro/introduction-to-spiking-neural-networksfrom-a-computational-neuroscience-perspective/30

Courtesy of Nvidia
Website

Simple Neural Net: 4 Nodes per Layer

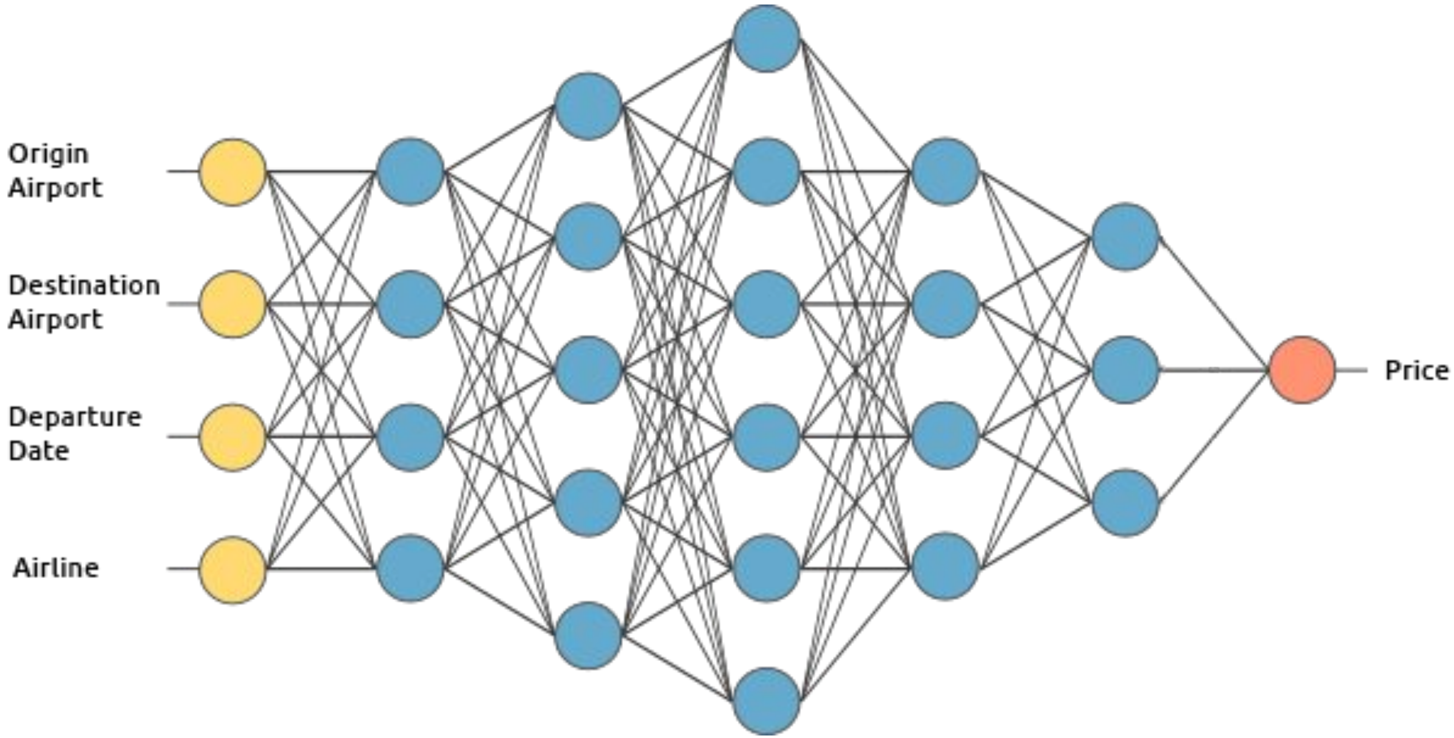
$x_1 {}^{*} w_1 + w_{0,1}$

$x_2 {}^{*} w_2 + w_{0,2}$



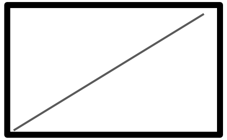
Courtesy of Nvidia Website

Deep Neural Net: Several Nodes per Layer

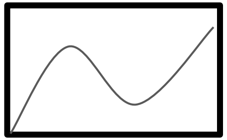


Activation Functions

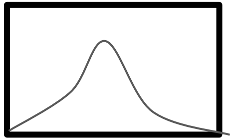
Linear



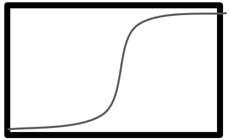
Polynomial



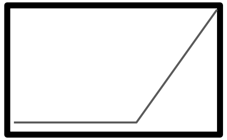
Gaussian



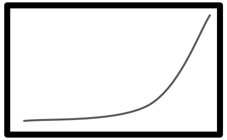
Sigmoid/Logistic



ReLU (Rectified Linear Unit)

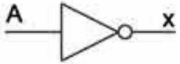


SoftMax



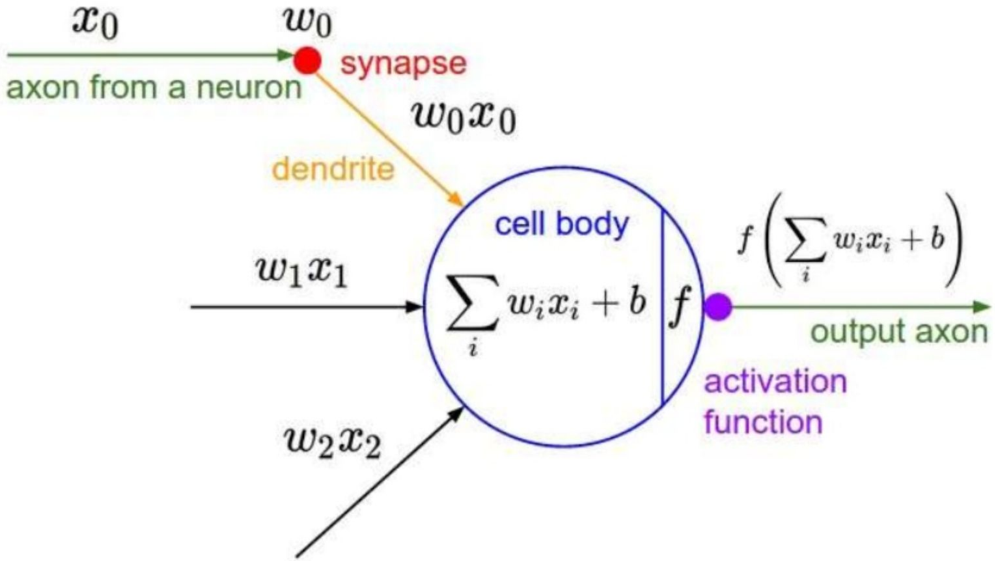
https://cs231n.github.io/neural-networks-1/
https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html#elu

Logic Gates

Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A \oplus B$	$\overline{A \oplus B}$
Symbol							
Truth Table	A | X 0 | **1** 1 | **0**	B A | X 0 0 | **0** 0 1 | **0** 1 0 | **0** 1 1 | **1**	B A | X 0 0 | **1** 0 1 | **1** 1 0 | **1** 1 1 | **0**	B A | X 0 0 | **0** 0 1 | **1** 1 0 | **1** 1 1 | **1**	B A | X 0 0 | **1** 0 1 | **0** 1 0 | **0** 1 1 | **0**	B A | X 0 0 | **0** 0 1 | **1** 1 0 | **1** 1 1 | **0**	B A | X 0 0 | **1** 0 1 | **0** 1 0 | **0** 1 1 | **1**

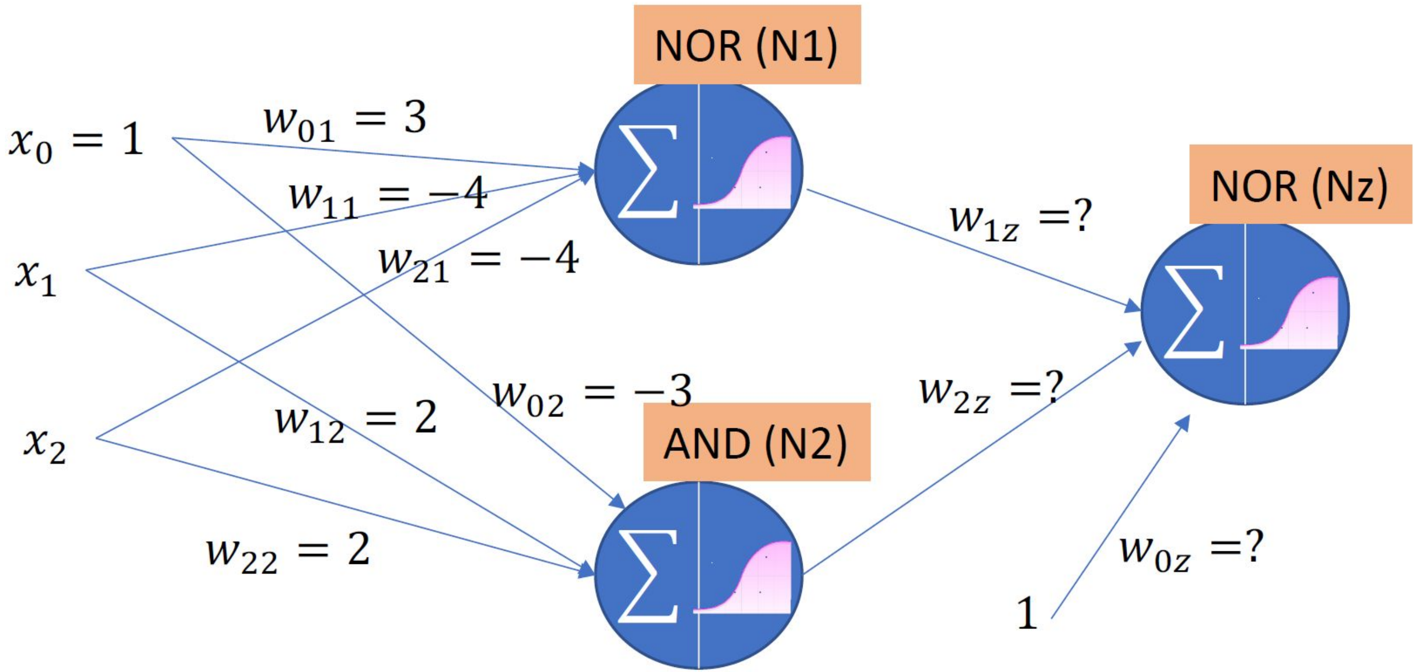
Artificial Neuron History

2nd Generation Neuron



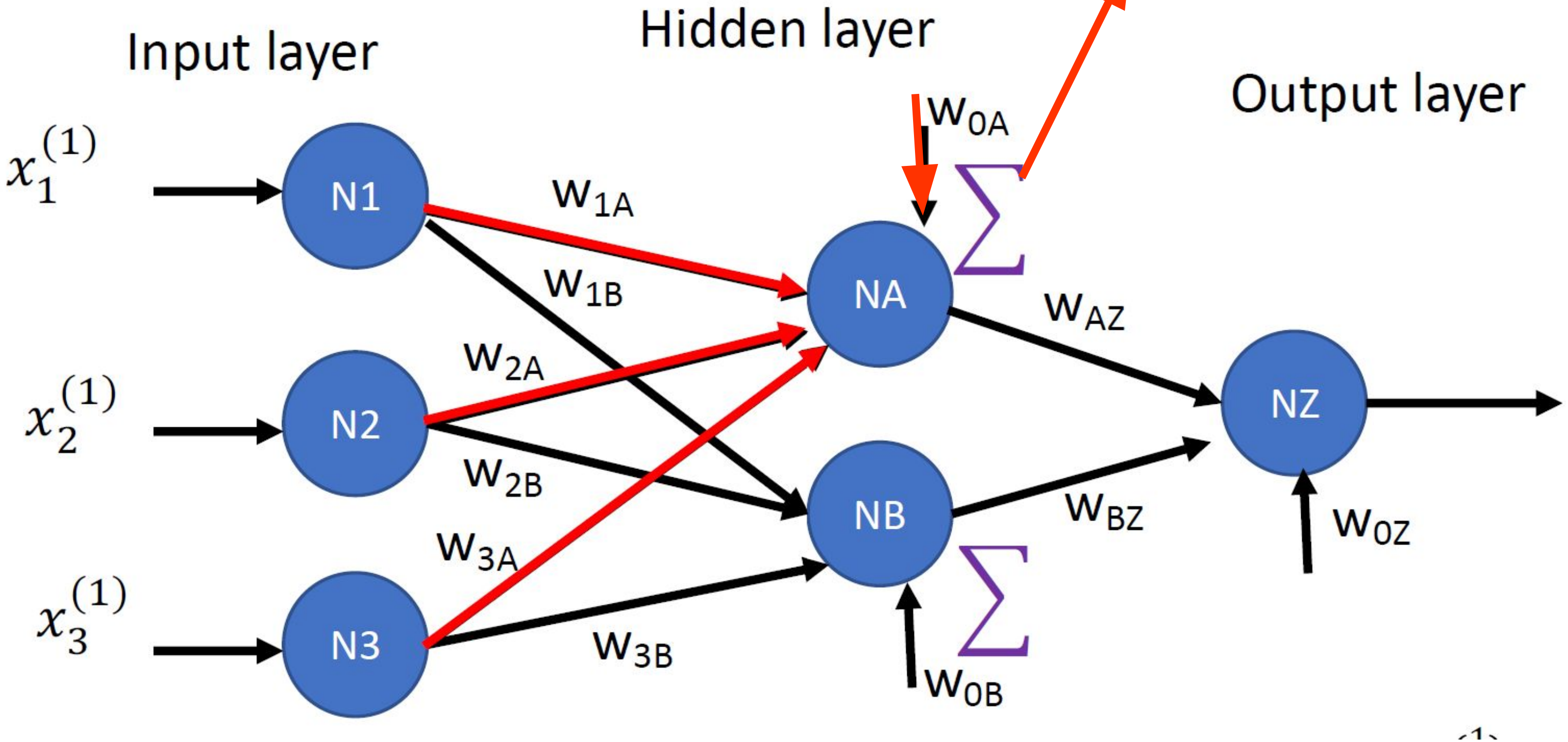
https://www.slideshare.net/hitechpro/introduction-to-spiking-neural-networksfrom-a-computational-neuroscience-perspective/30

Neural Net with Weights & Logic Gates



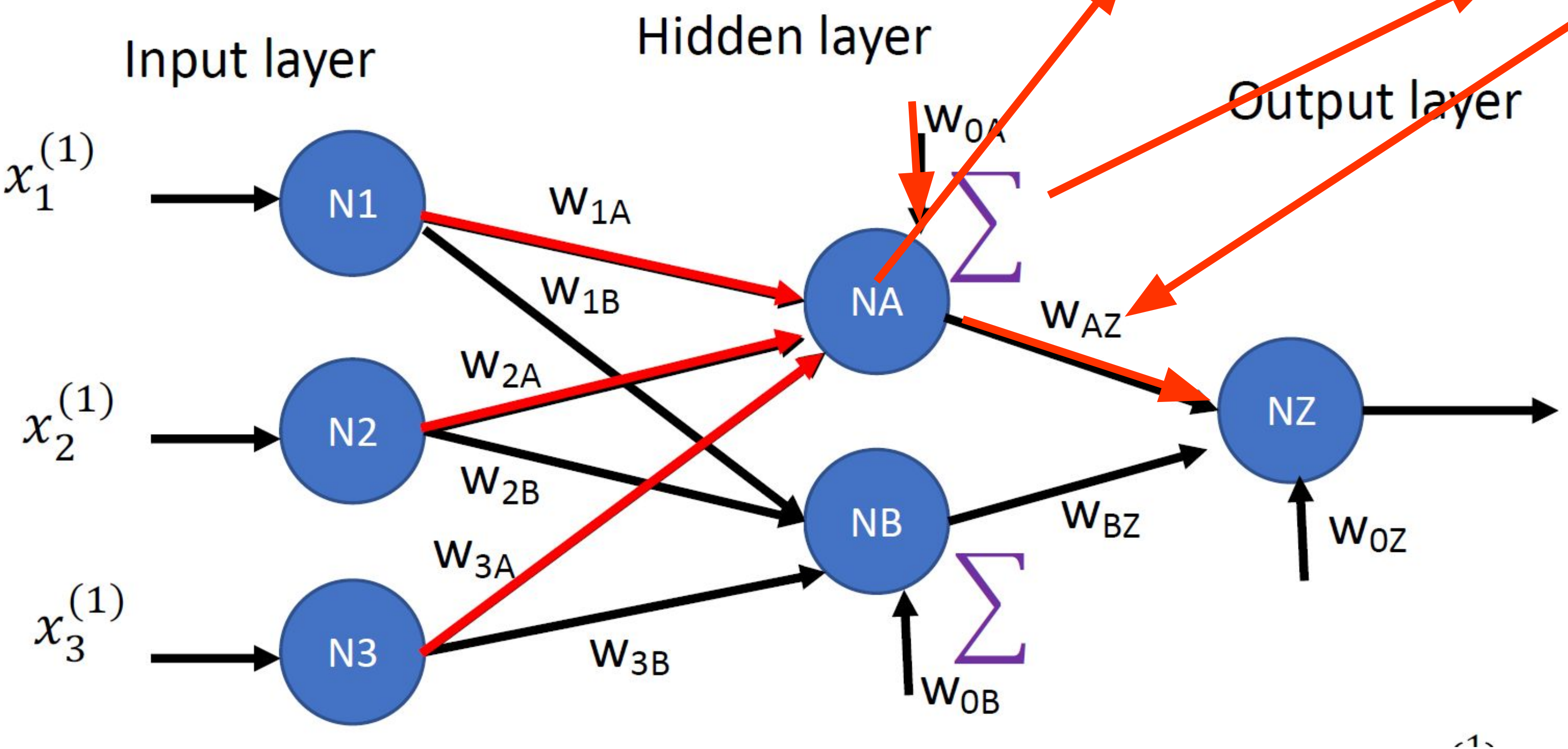
Neural Net with Weights

$$w_{0A} + x_1 w_{1A} + x_2 w_{2A} + x_3 w_{3A} = 1.32$$

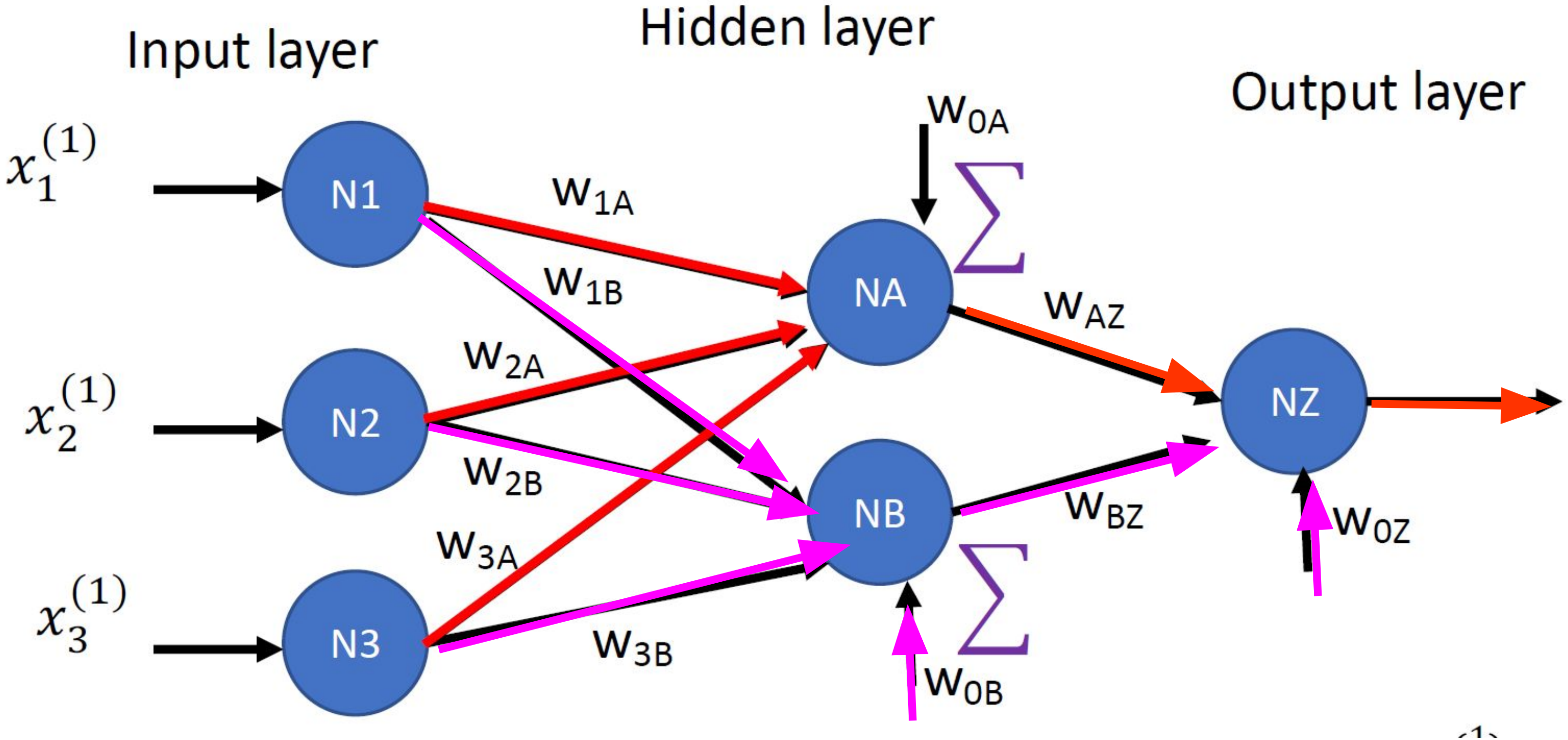


Neural Net with Weights

$$f(net_A) = \frac{1}{1+e^{-(1.32)}} = 0.7892$$



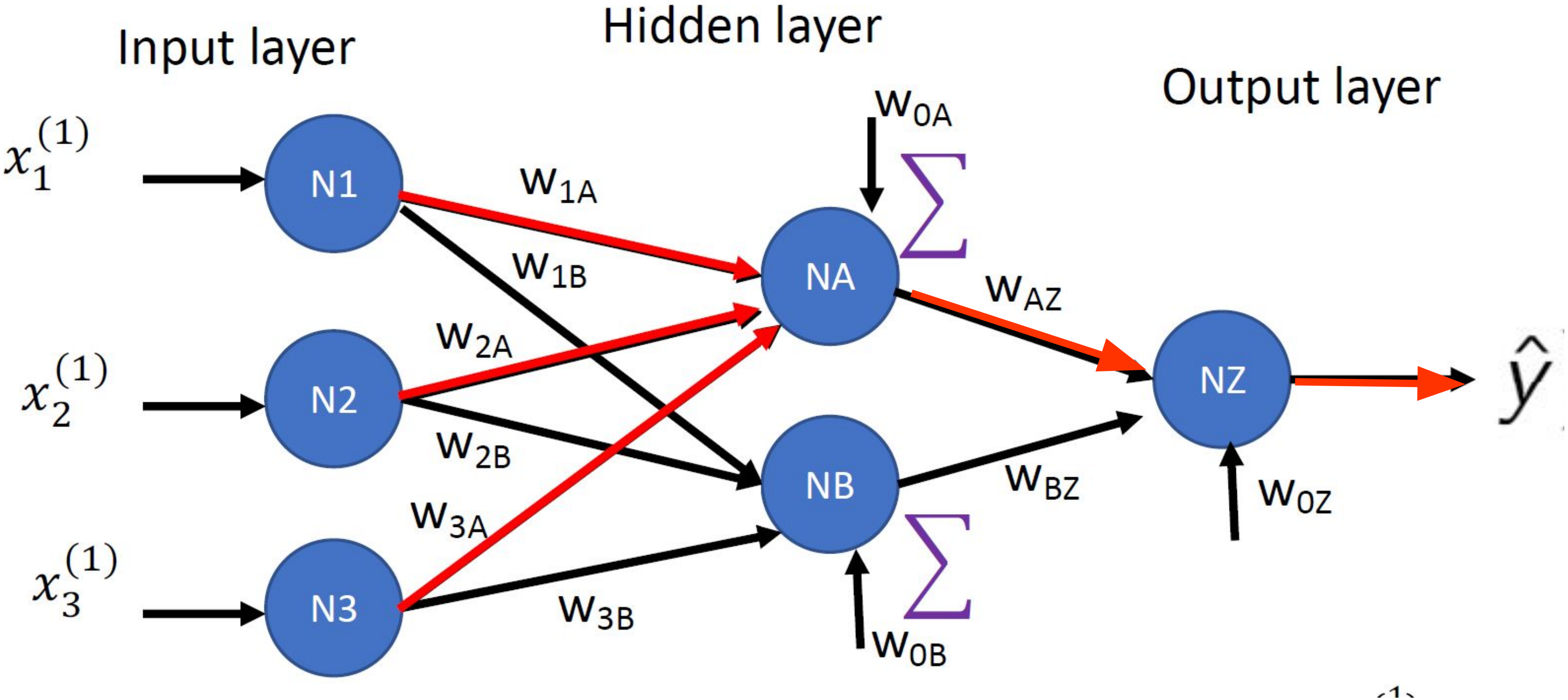
Neural Net with Weights



Neural Net with Weights

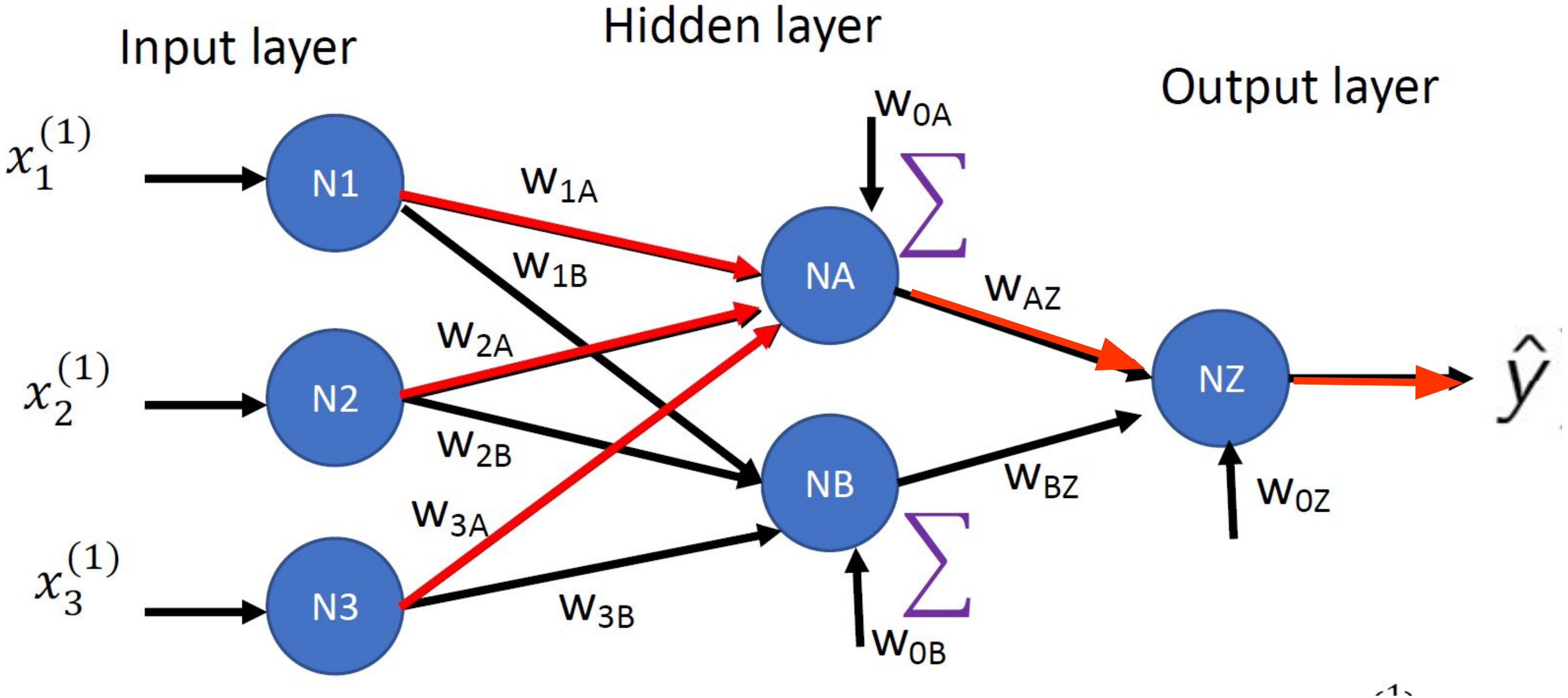
Loss Function!!!

$$\sum_{i=1}^{m}(y_i - \hat{y})^2$$



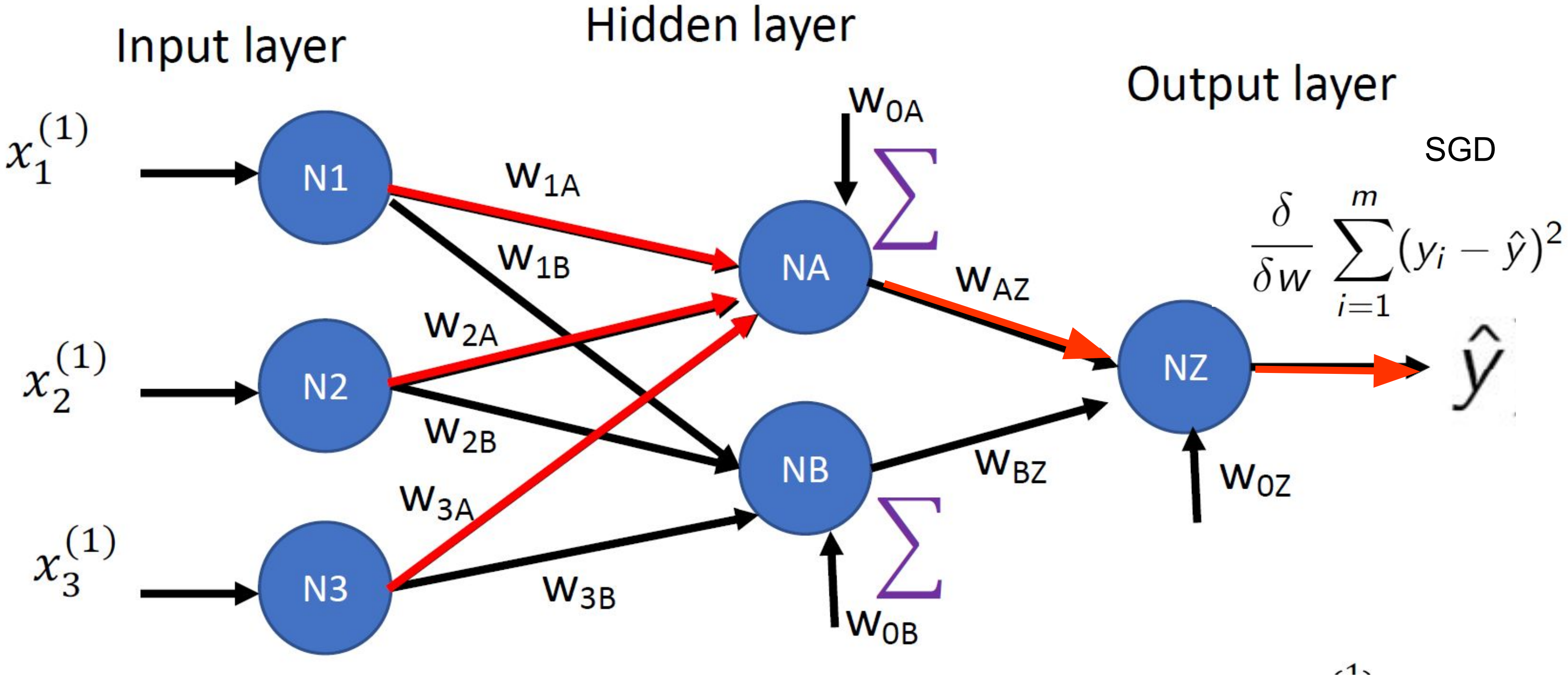
Neural Net with Weights & GD

$$\frac{\delta}{\delta w}\sum_{i=1}^{m}(y_i - \hat{y})^2$$



Neural Net with Weights & SGD

SGD using a randomly selected observation and perform GD on it



Gradient Descent Methods

$$J(w_j)_t := J(w_j)_{t-1} - \alpha(\frac{1}{m}\sum_{i=1}^{m}(y_i - x_i w) + \frac{1}{m}\sum_{i=1}^{m} x_i(y_i - x_i w))$$

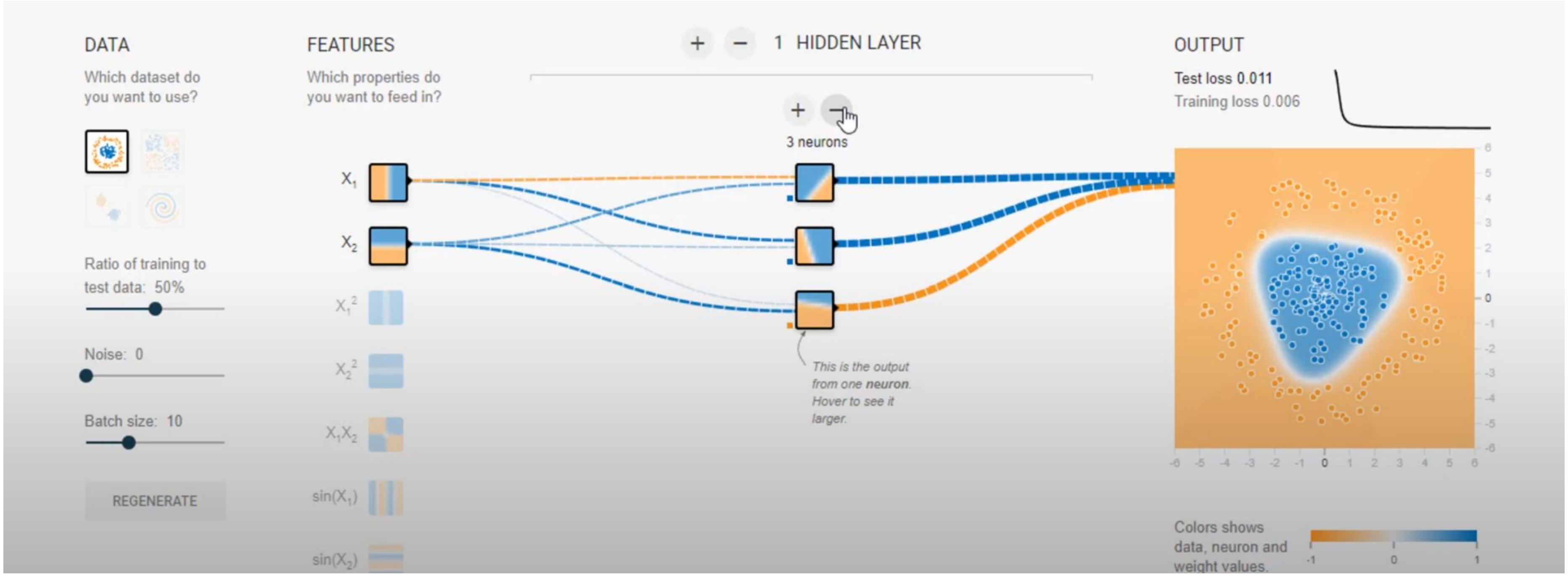
Batch GD → Performing GD on all observations

SGD → Calculating GD & performing step on a randomly selected observation

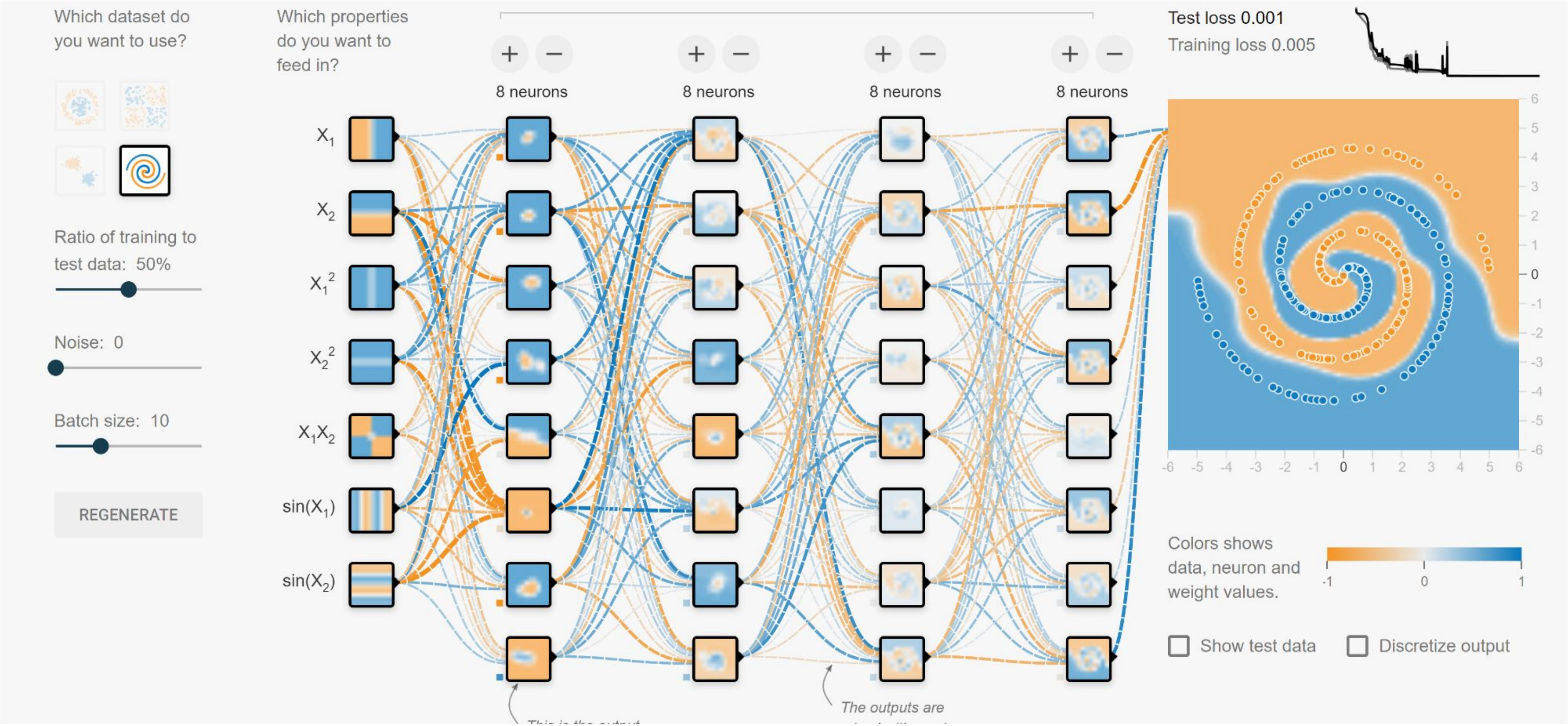
mini-Batch GD → SGD but with several data points (a subset of observations)

Neural Networks can use either, but computation is expensive so SGD is often used

Neural Net with Weights

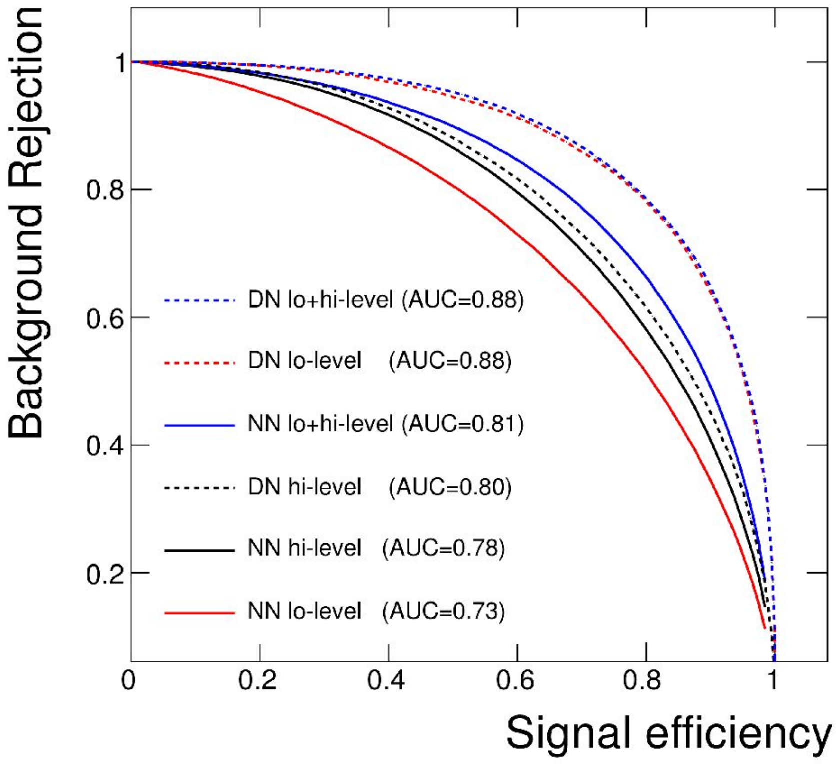


Neural Net with Weights



Higgs Boson Detection

Thanks to
Dr. Baldi



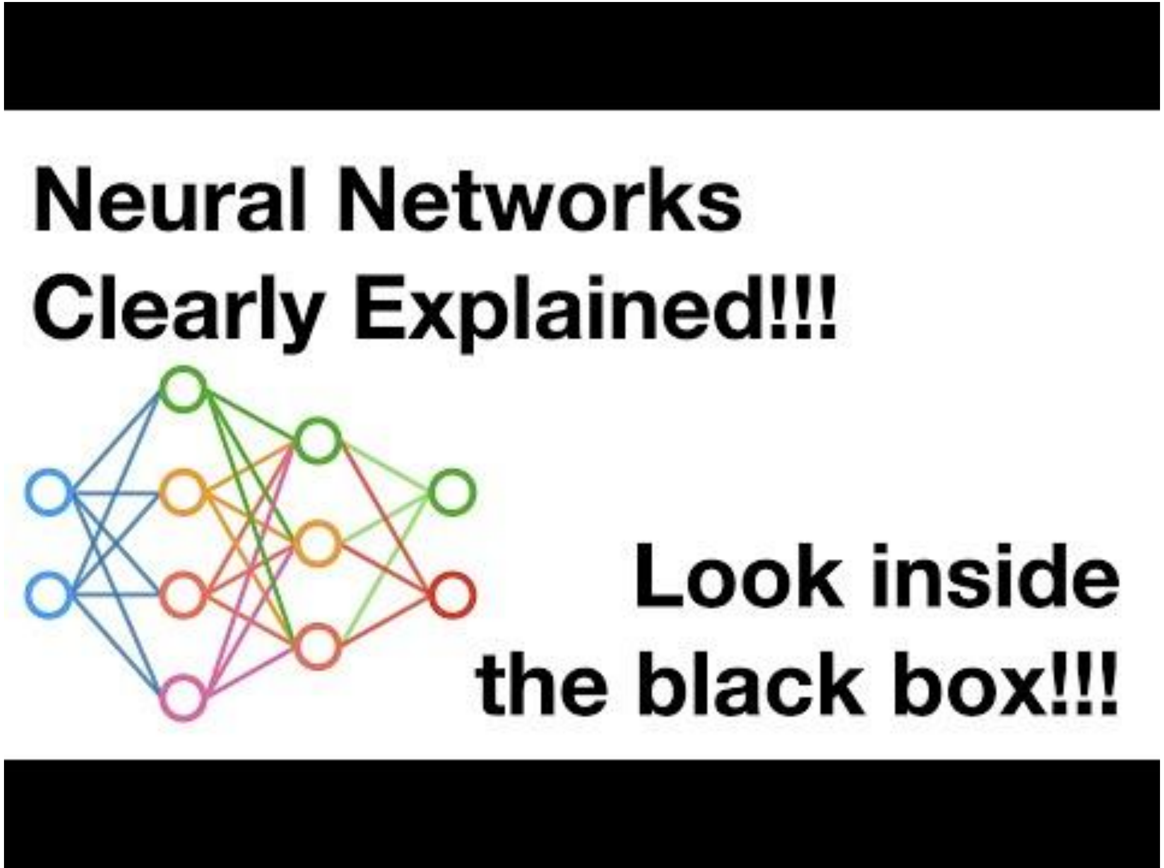
	AUC		
Technique	Low-level	High-level	Complete
BDT	0.73	0.78	0.81
NN	0.733 (0.007)	0.777 (0.001)	0.816 (0.004)
DN	0.880 (0.001)	0.800 ($<$ 0.001)	0.885 (0.002)

BDT= Boosted Decision Trees in TMVA package
NN = Shallow Neural Nets
DN = Deep Neural Nets
AUC = Mean Area Under the Curve

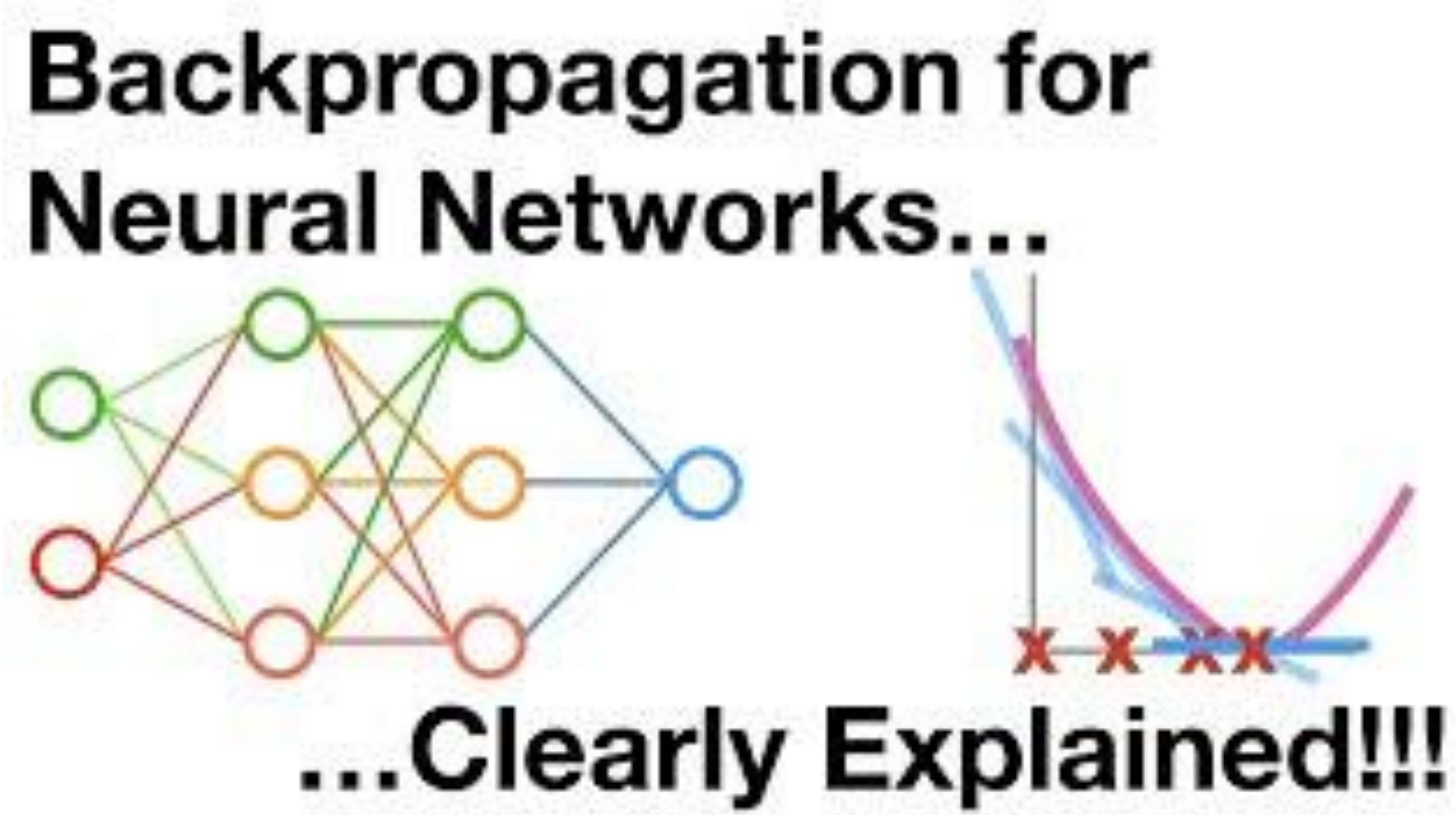
Deep neural network improves AUC by 8%

Nature Communications, July 2014

In Depth Relatable NN Example



In Depth Relatable Backpropogation Example



Tensorflow + Breast Cancer Data

Jupyter Notebooks Time!

https://colab.research.google.com