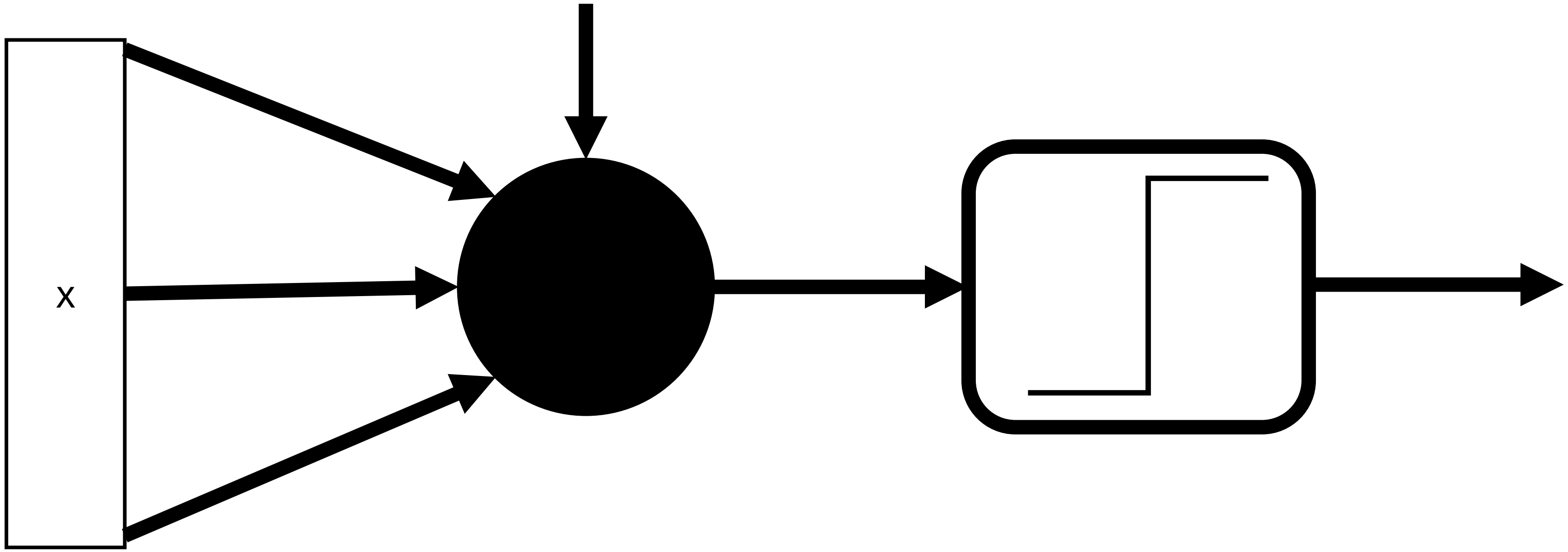
Multilayer Perceptrons

Rahul Shome

IML, S2 2024

Based on slides by UToronto, Purdue, TowardsDataScience

Single ‘Layer’ Perceptron

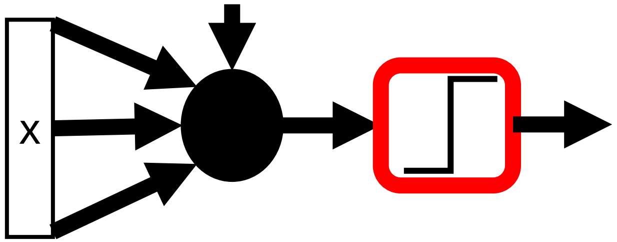


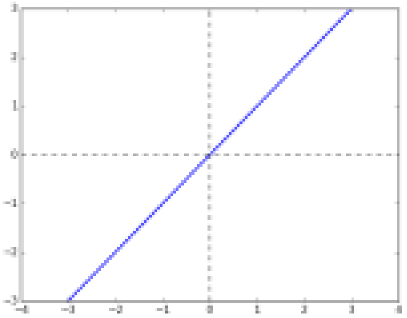
Input neurons $\boldsymbol{x}$

Weights $\boldsymbol{w}$

Predicted label $= \sigma(\boldsymbol{w}^T\boldsymbol{x} + w_0)$.

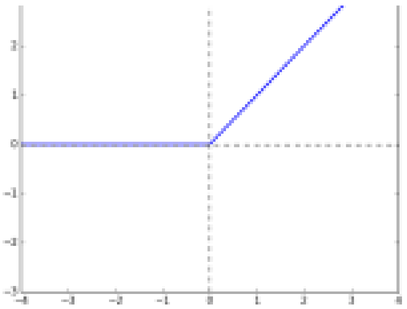
Activation Functions





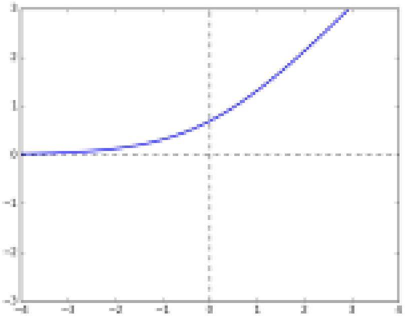
Linear

$$y = z$$



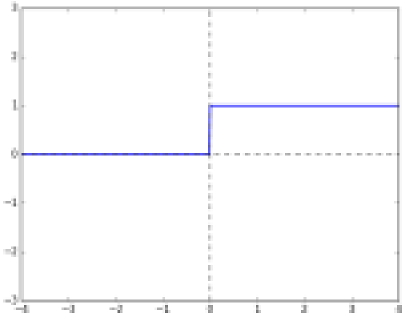
Rectified Linear Unit (ReLU)

$$y = \max(0, z)$$



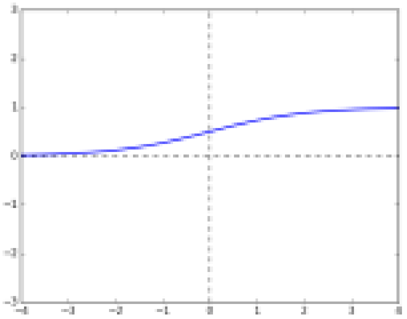
Soft ReLU

$$y = \log 1 + e^z$$



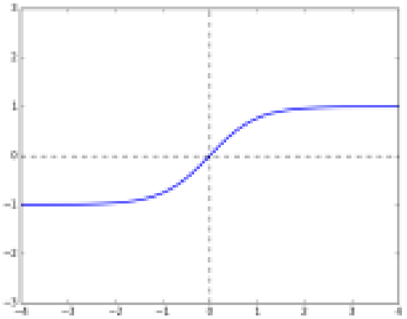
Hard Threshold

$$y = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$



Logistic

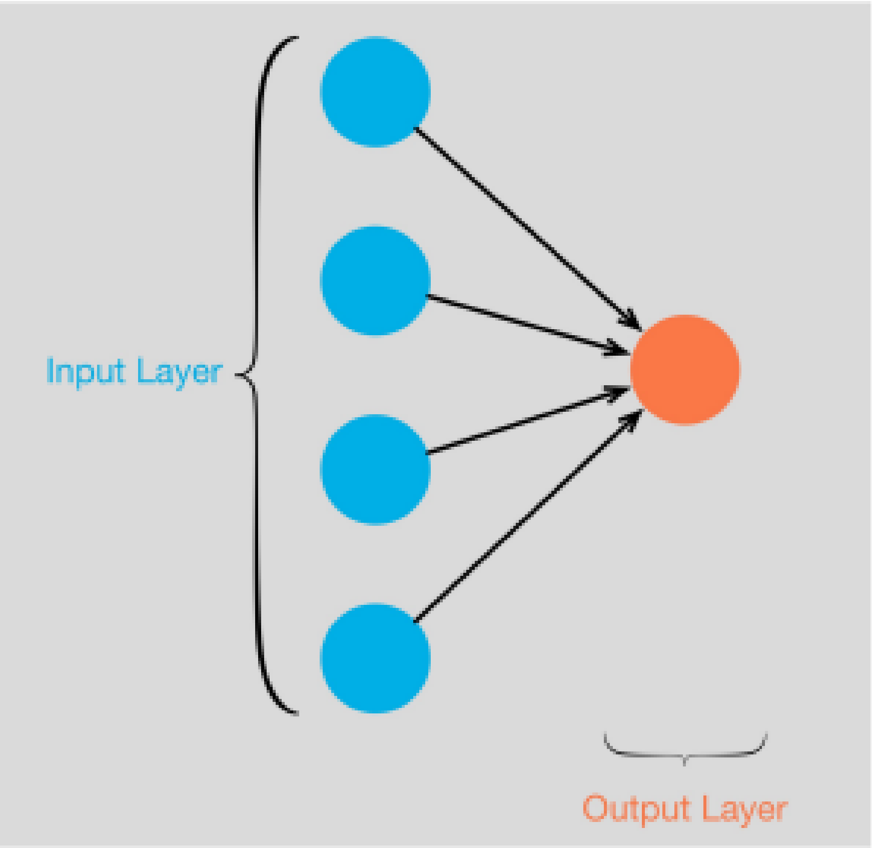
$$y = \frac{1}{1 + e^{-z}}$$

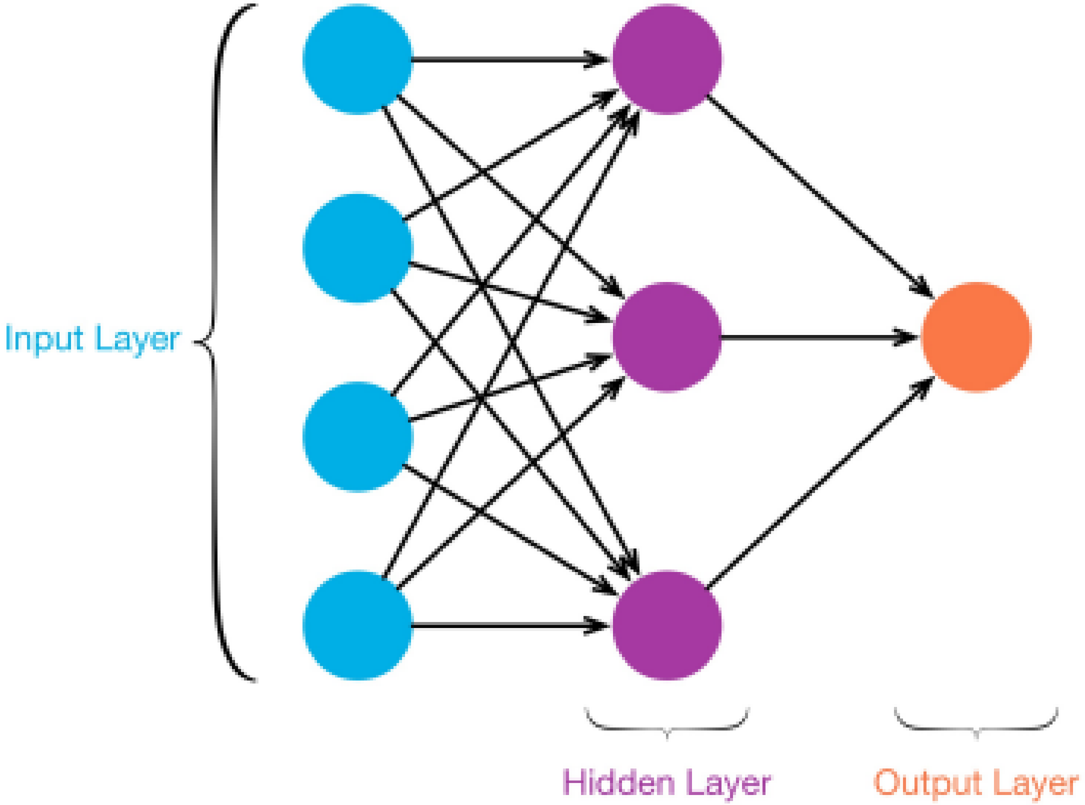


Hyperbolic Tangent (tanh)

$$y = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

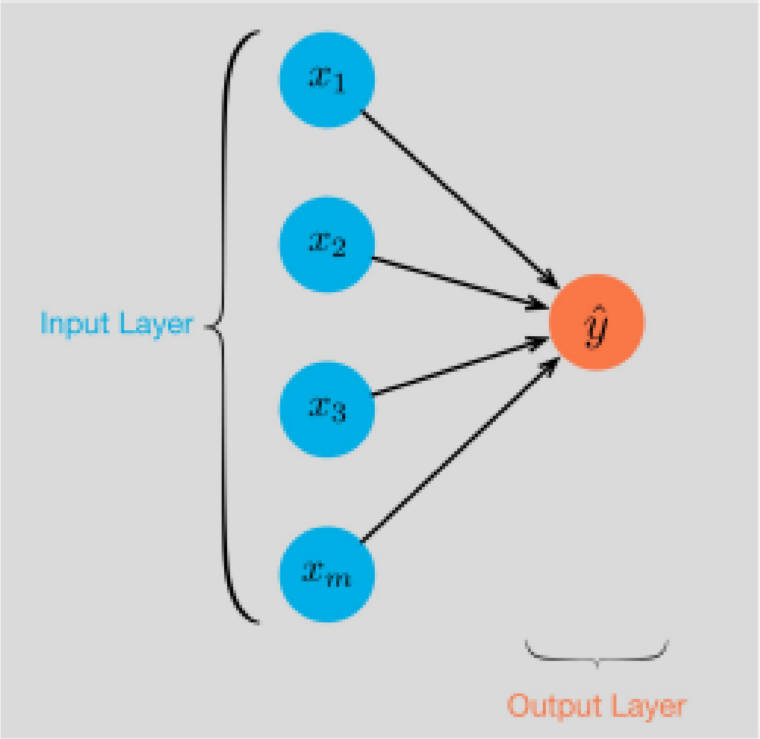
Multi-Layer Network

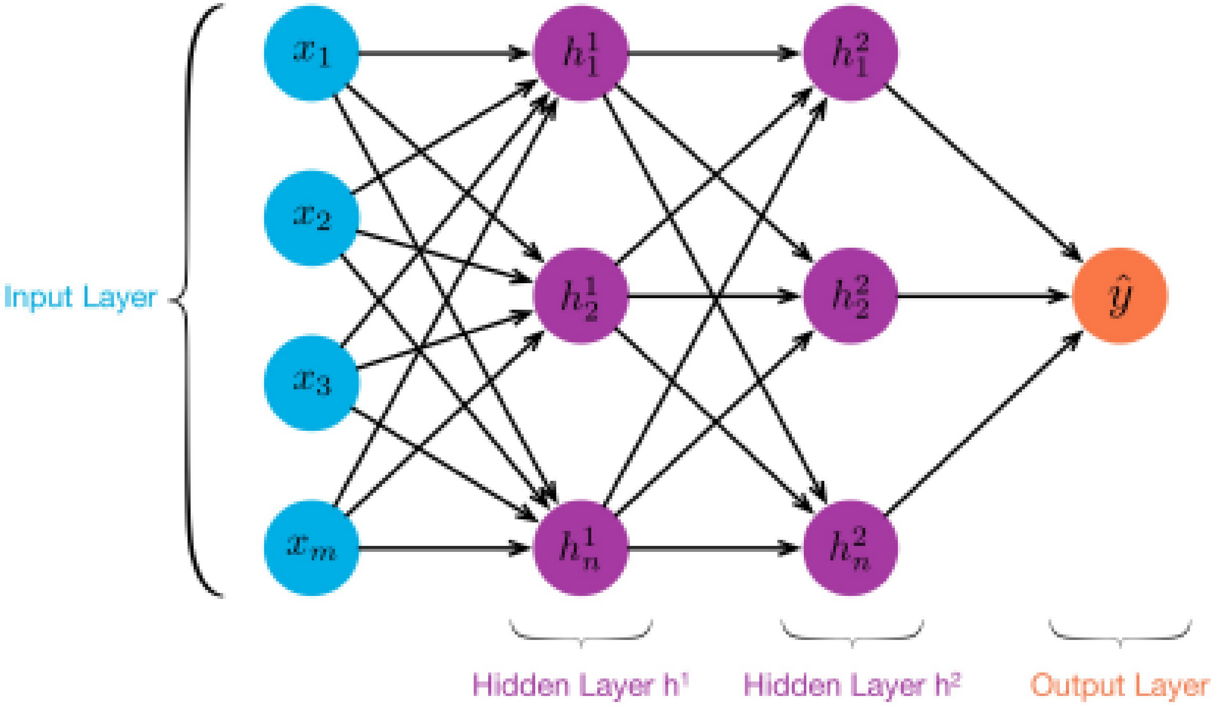




- “Hidden” layers

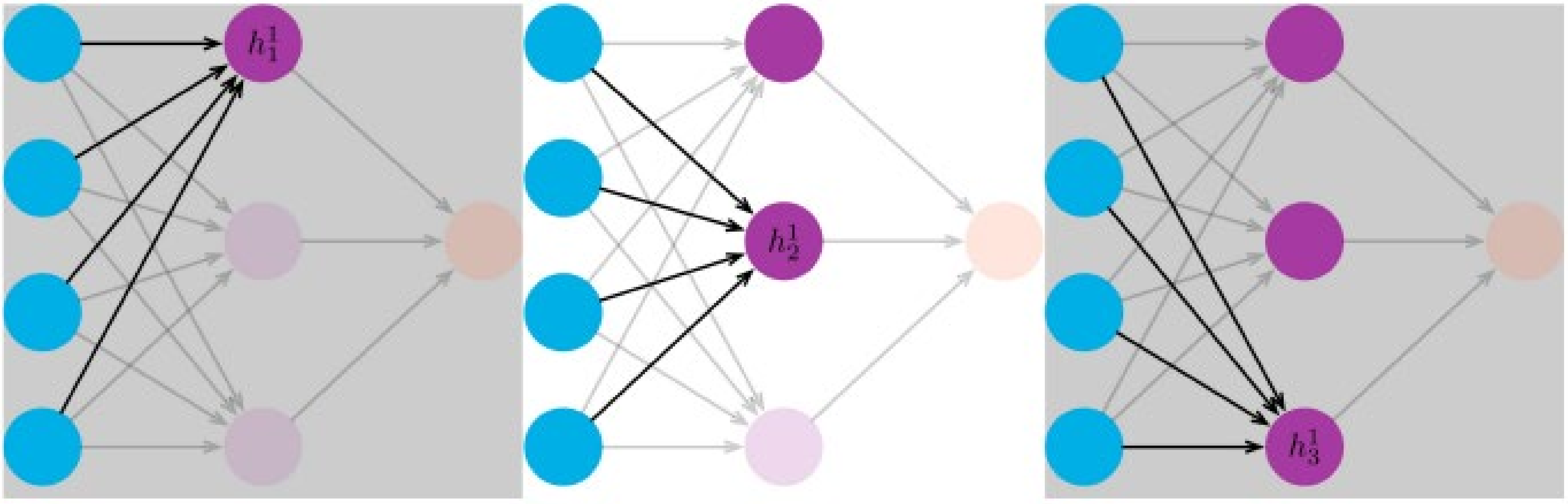
Multi-Layer Network





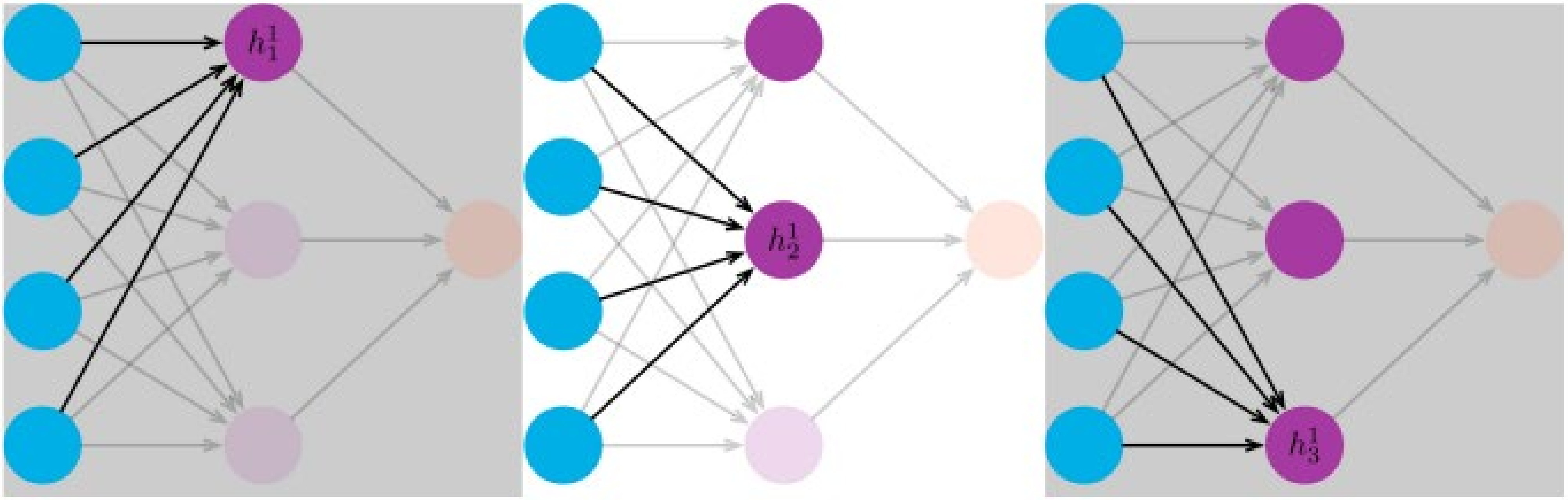
- “Hidden” layers need a set of weights

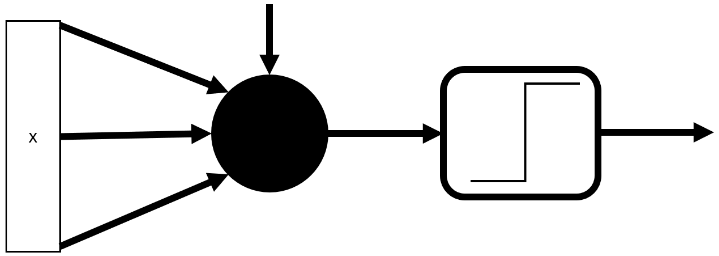
Multi-Layer Network



- Each neuron is the output of a perceptron

Does it look like a perceptron?



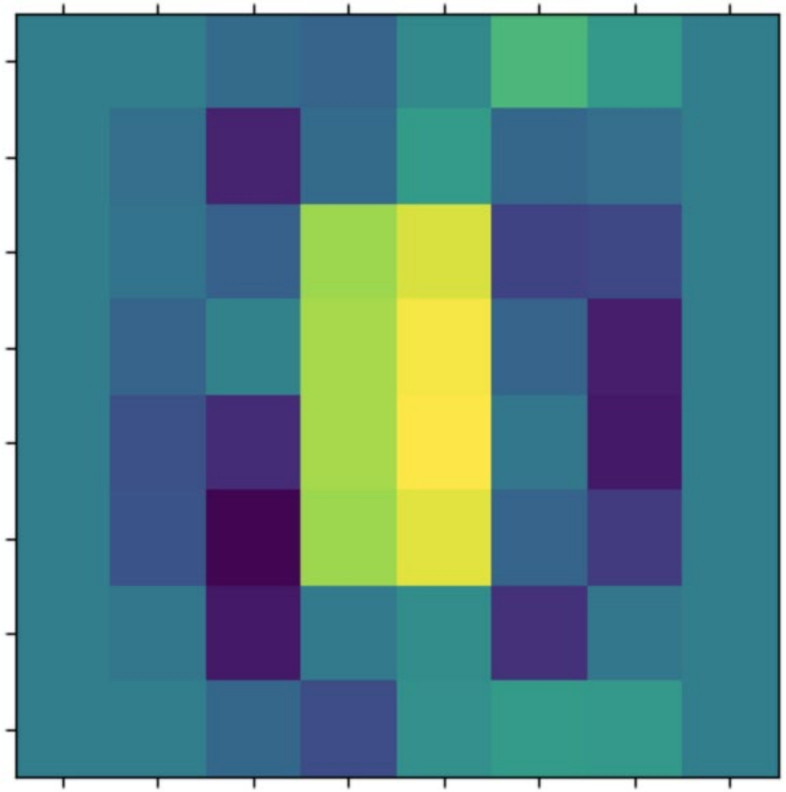


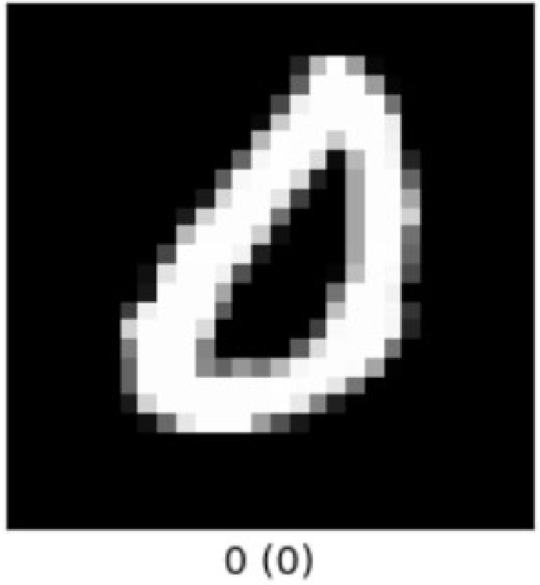
Weights Get Combined

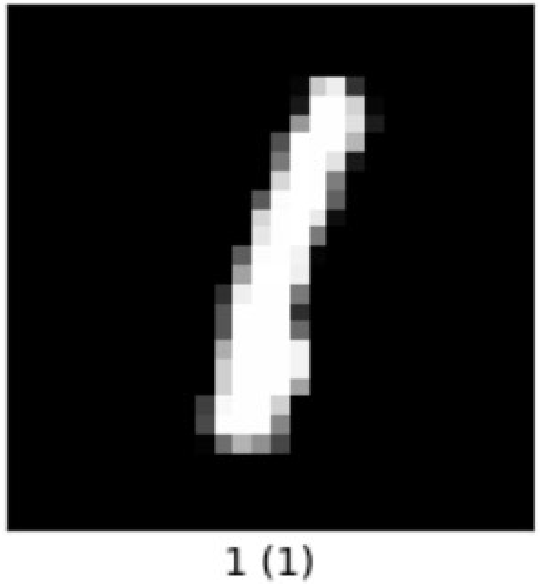
$$\begin{bmatrix} h_1^1 \\ h_2^1 \\ \cdots \\ h_n^1 \end{bmatrix} = \begin{bmatrix} w_{11}^1 & w_{12}^1 & \cdots & w_{1n}^1 \\ w_{21}^1 & w_{22}^1 & \cdots & w_{2n}^1 \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1}^1 & w_{m2}^1 & \cdots & w_{mn}^1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Interpreting Weights

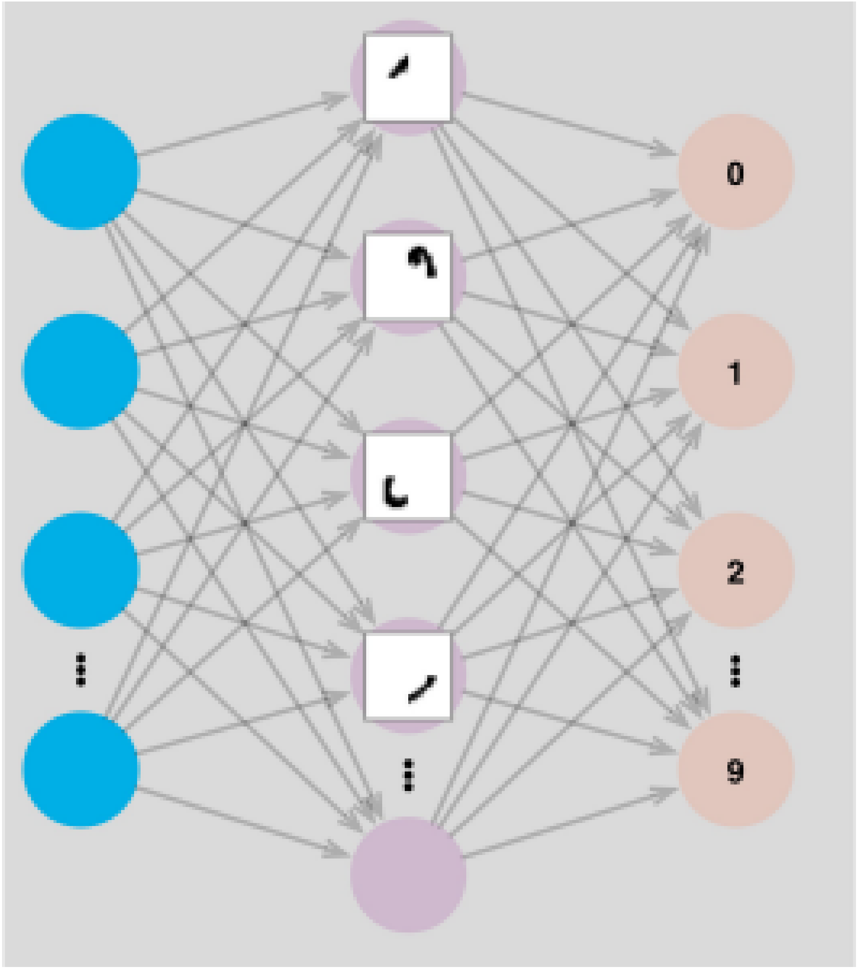
- Distinguishing 0s and 1s

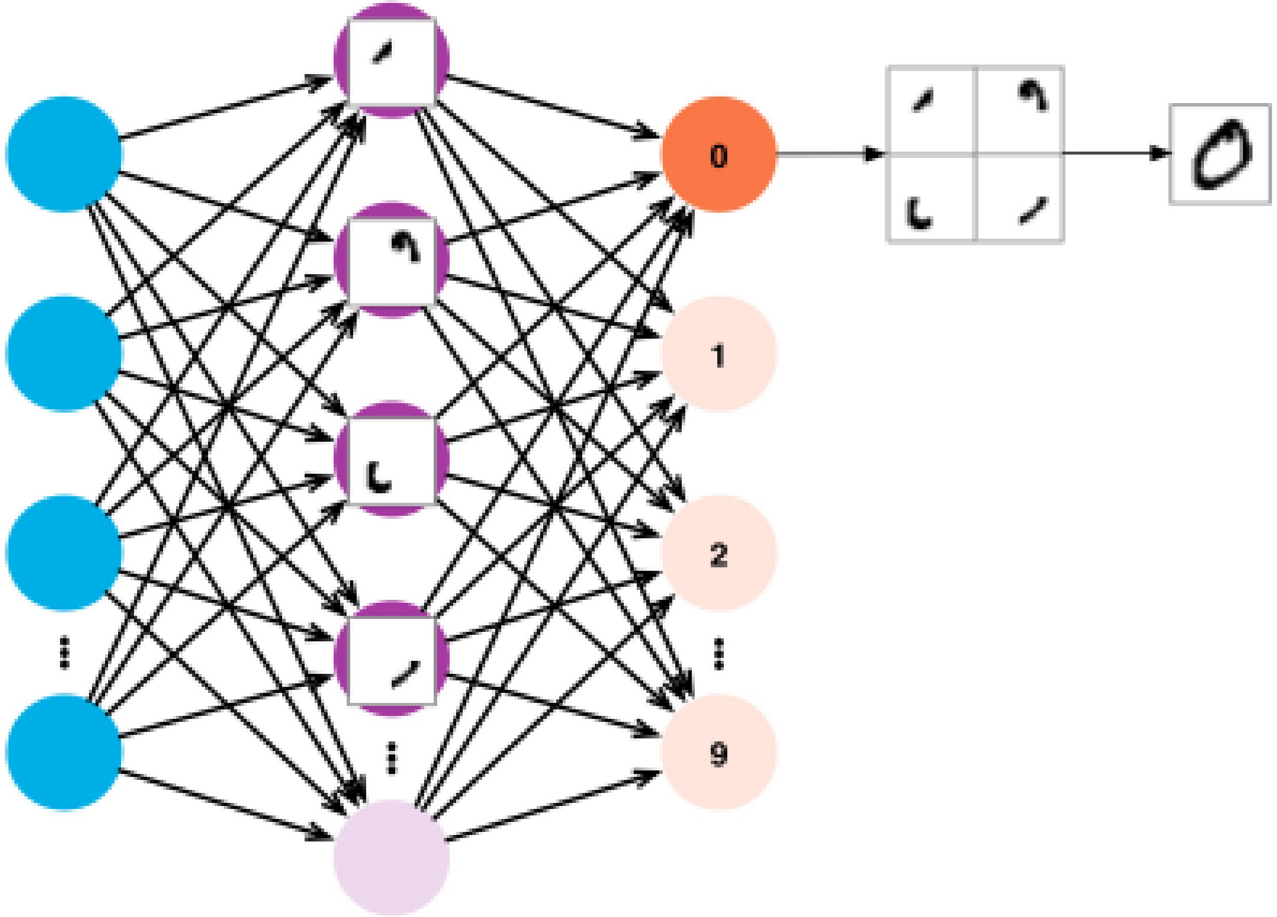




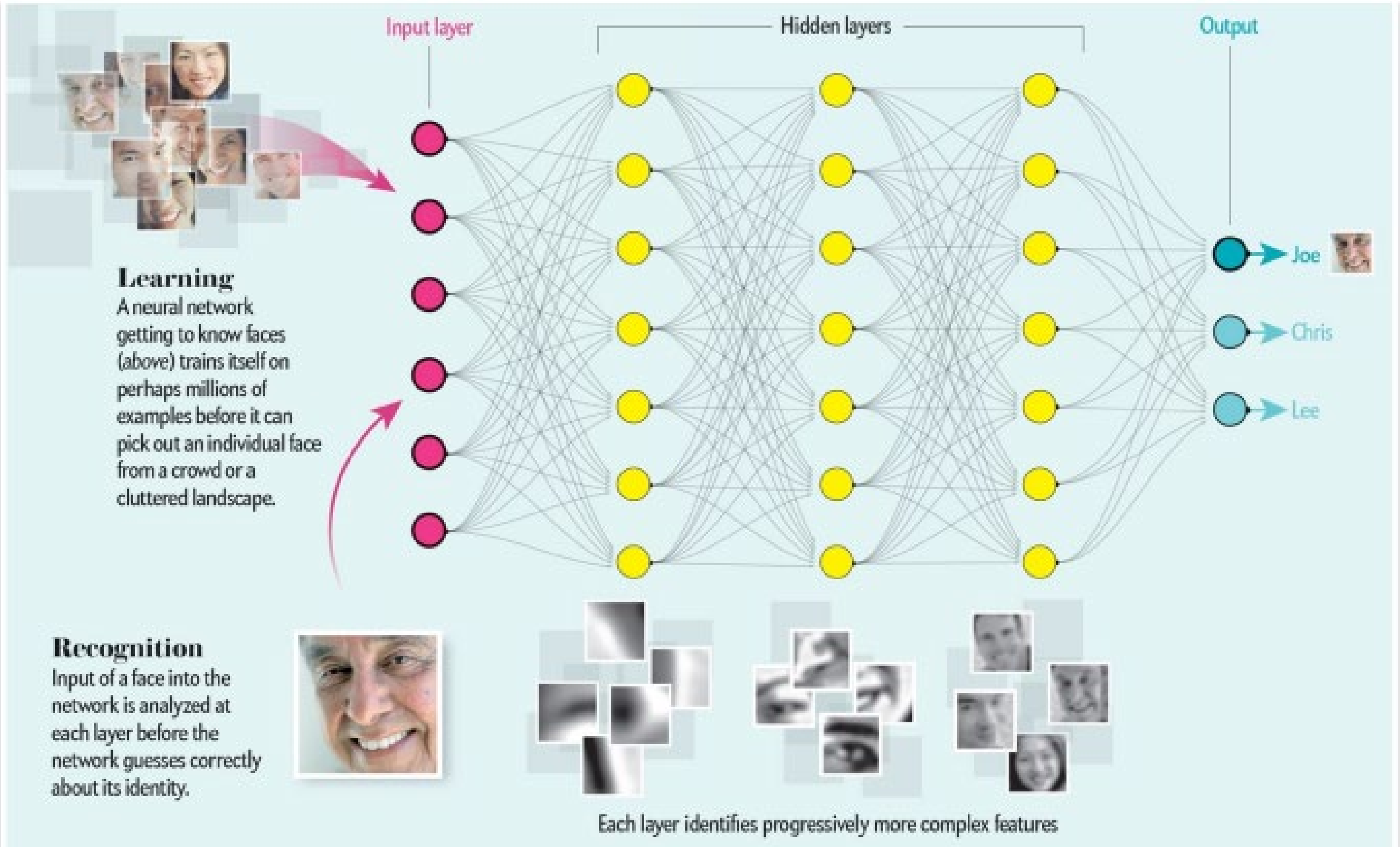


Weights and Hidden Layers





Learning Complex Features and Functions



Learning
A neural network getting to know faces (*above*) trains itself on perhaps millions of examples before it can pick out an individual face from a crowd or a cluttered landscape.

Recognition
Input of a face into the network is analyzed at each layer before the network guesses correctly about its identity.

Each layer identifies progressively more complex features

https://www.scientificamerican.com/article/springtime-for-ai-the-rise-of-deep-learning/

How Do You Optimise?

- Model Parameters?
- Loss?
- Gradient?