Artificial Neural Networks
人工神经网络

权小军教授
中山大学计算机学院

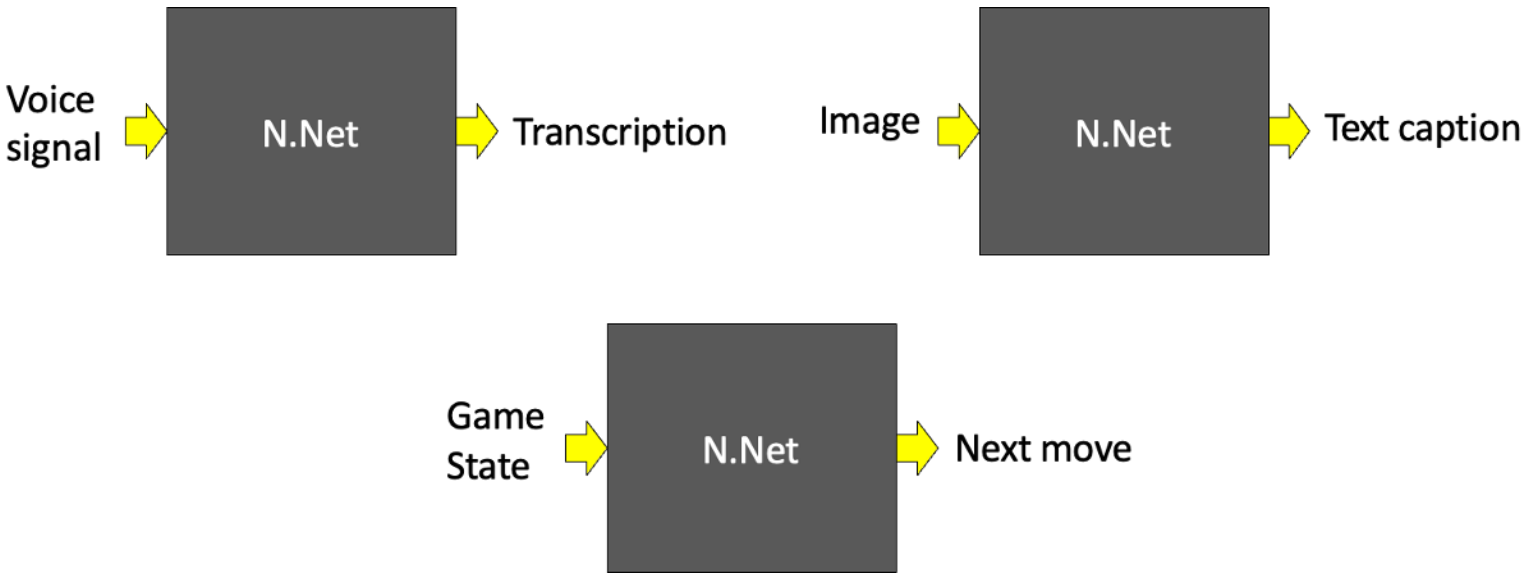
quanxj3@mail.sysu.edu.cn

2023 年 3 月 7 日

Lecture 2: Neural Networks Basics

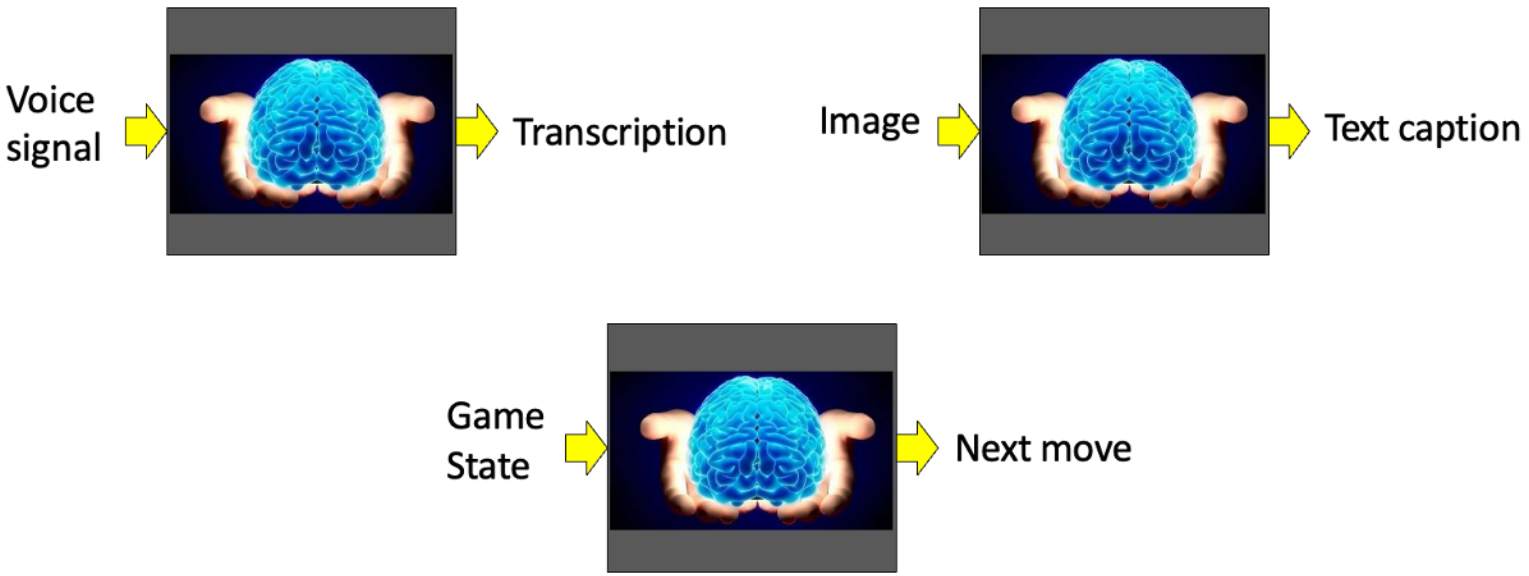
Lecture 2.1 Neural Network Units

What are neural networks?



- What are these boxes?
  - Functions that take an input and produce an output
  - What are these functions?

The human perspective

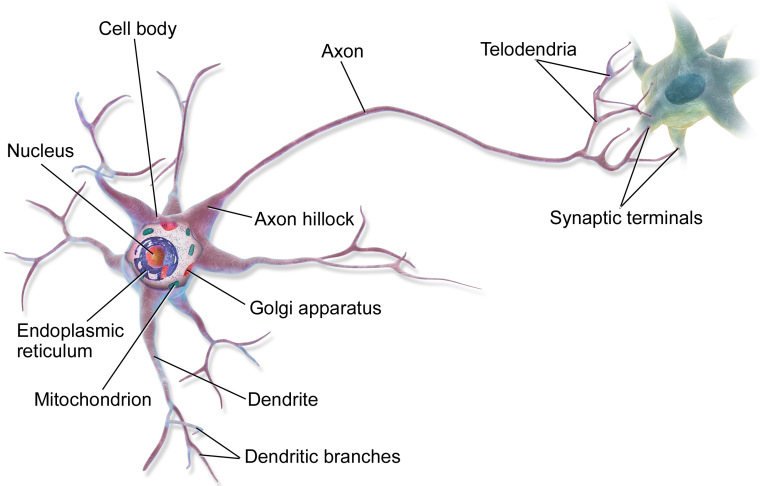


- In a human, those functions are computed by the brain...

Neural networks and the brain

- In their basic form, neural networks mimic (模仿) the networked structure in the brain

The brain



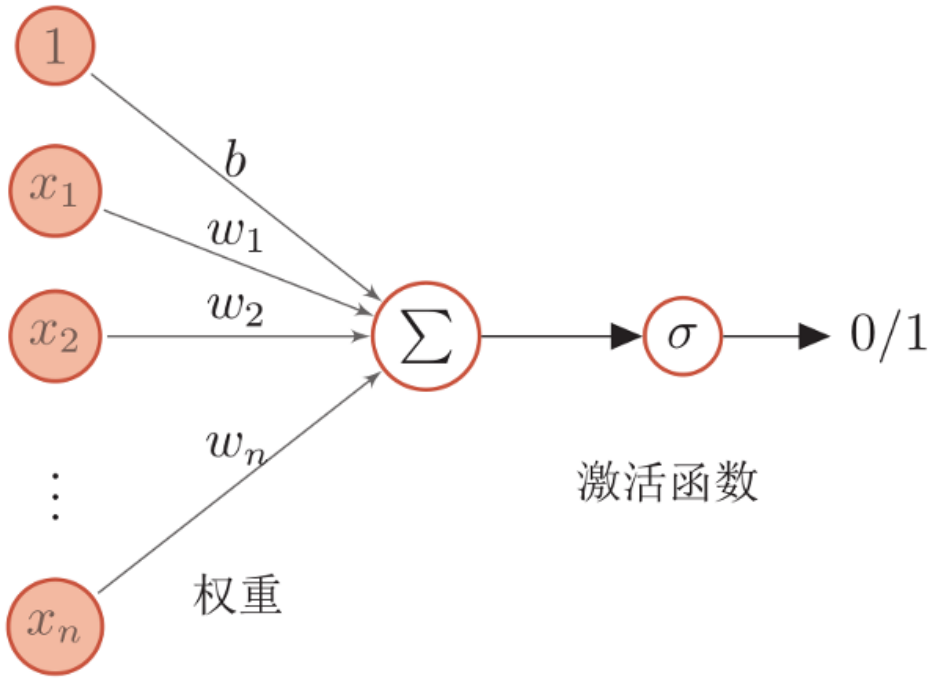
- The Brain is composed of networks of neurons

Perceptron (感知机)

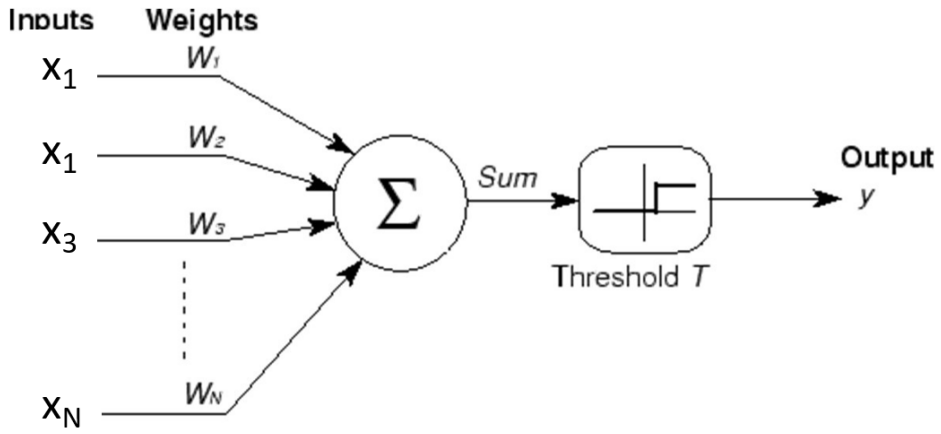
- Neural nets are composed of networks of computational models of neurons called perceptrons (感知机)

Perceptron (感知机)

- Neural nets are composed of networks of computational models of neurons called perceptrons (感知机)



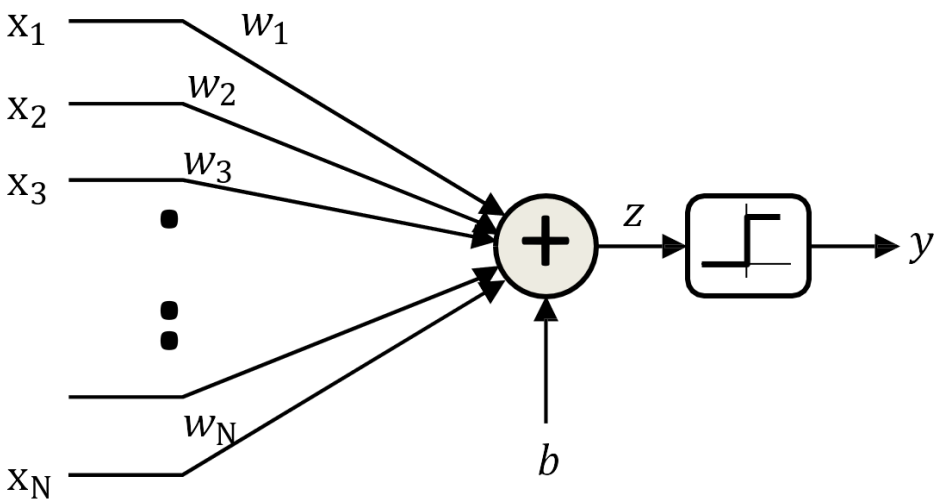
Perceptron (感知机)



$$y = \begin{cases} 1 & if \sum_i w_i x_i \geq T \\ 0 & else \end{cases}$$

- A threshold (阈值) unit
  - "Fires" if the weighted sum of inputs exceeds a threshold
  - Electrical engineers will call this a *threshold gate*
    - A basic unit of Boolean circuits (布尔电路)

A better figure

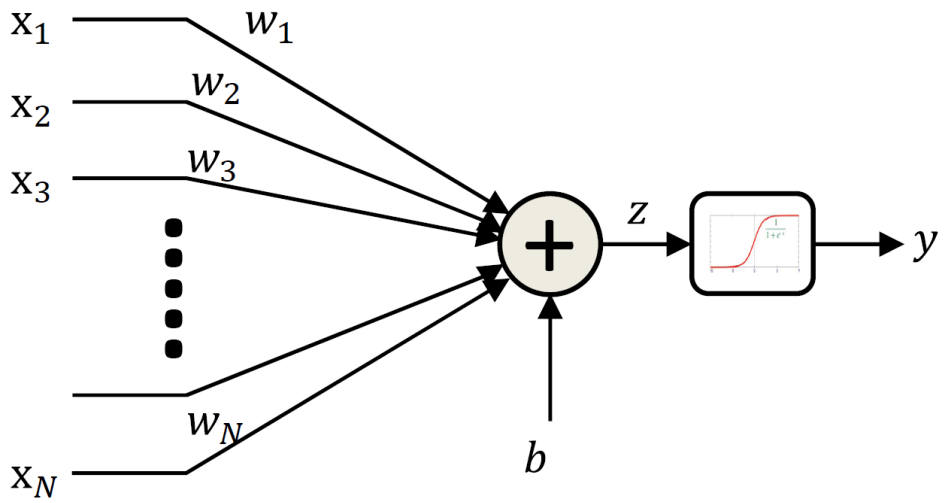


$$z = \sum_i w_i x_i + b$$

$$y = \begin{cases} 1 & if\ z \geq 0 \\ 0 & else \end{cases}$$

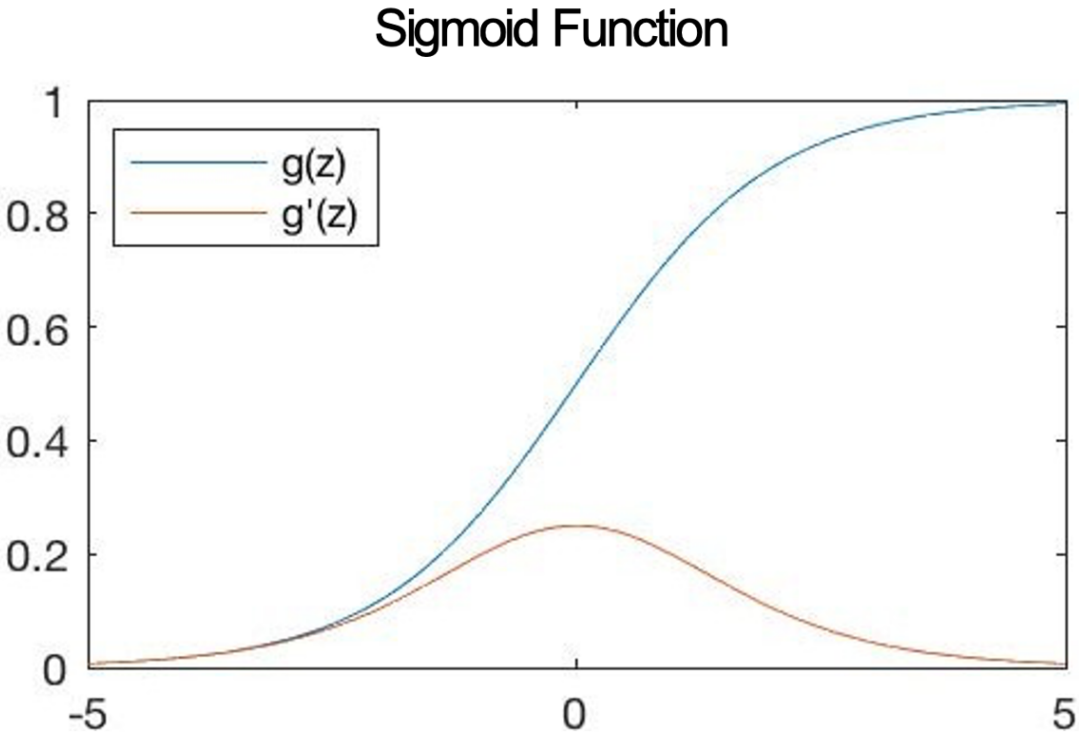
- A threshold unit
  - "Fires" if the output of the function is positive
    - The bias b is the negative of threshold T in the previous slide

The "soft" perceptron (logistic)



- A squashing (挤压) function instead of a threshold
  - The sigmoid "activation" replaces the threshold
    - An activation (激活): The function that acts on the weighted combination of inputs (and bias)

Other "activations"



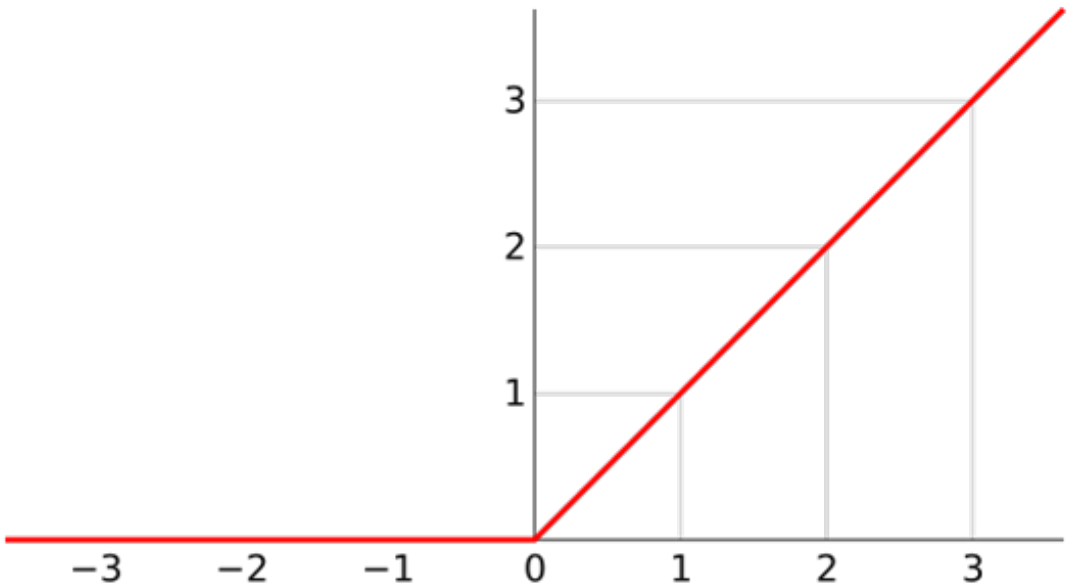
$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = g(z)(1 - g(z))$$

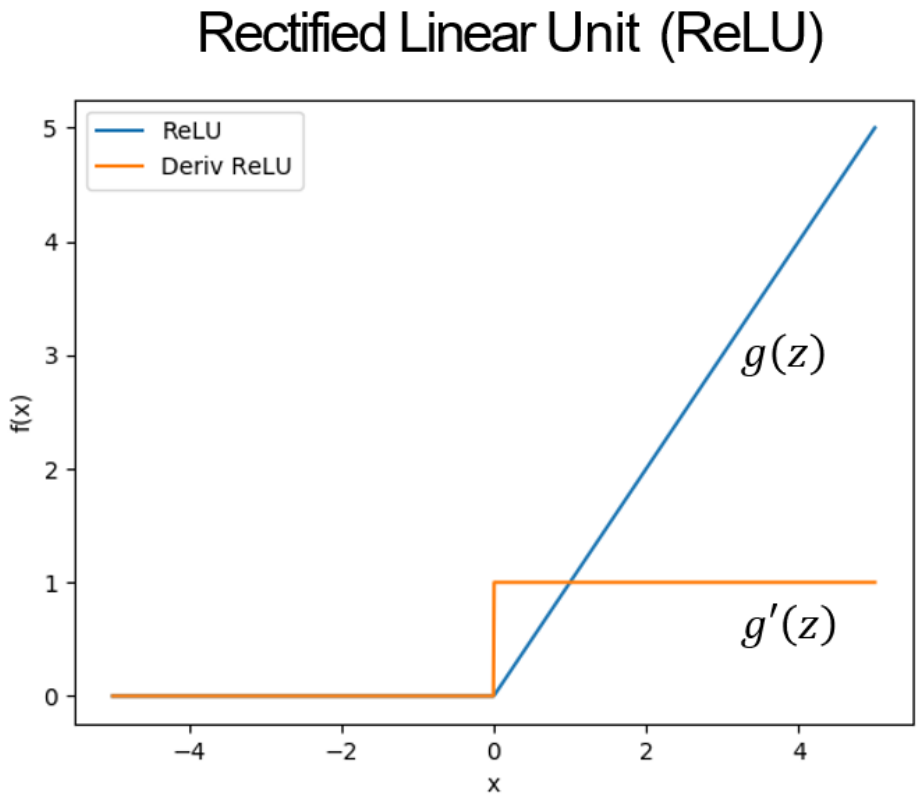
Other “activations”

- 线性整流函数 (Rectified Linear Unit, ReLU)

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



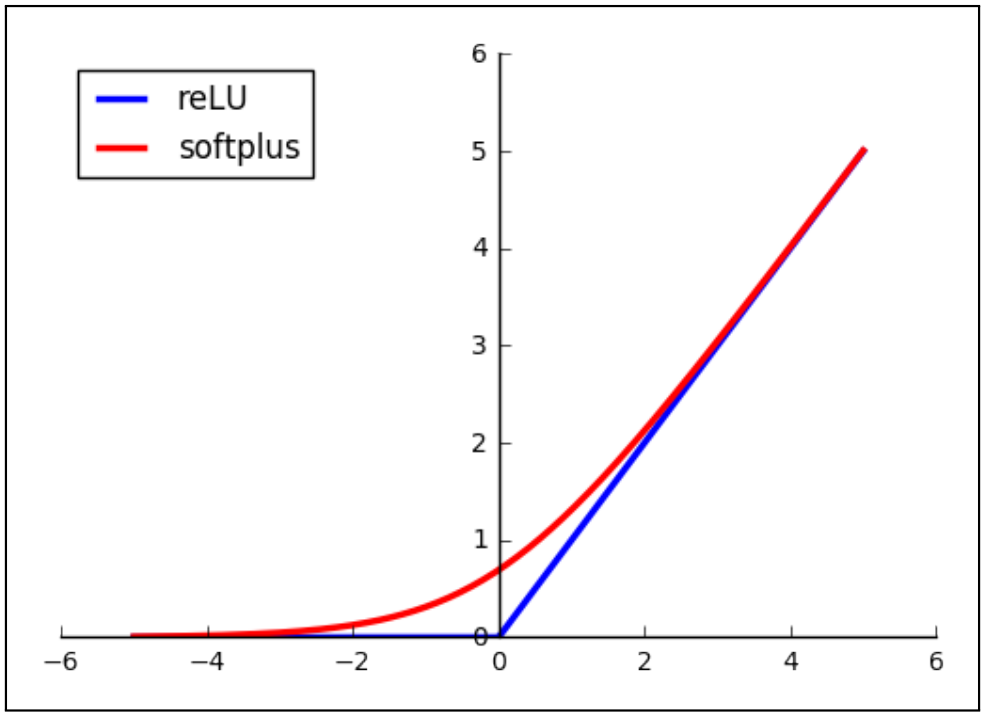
Derivative of ReLU



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

$$g'(z) = \begin{cases} 1, & z > 0 \\ 0, & otherwise \end{cases}$$

Softplus: a smooth approximation to ReLU



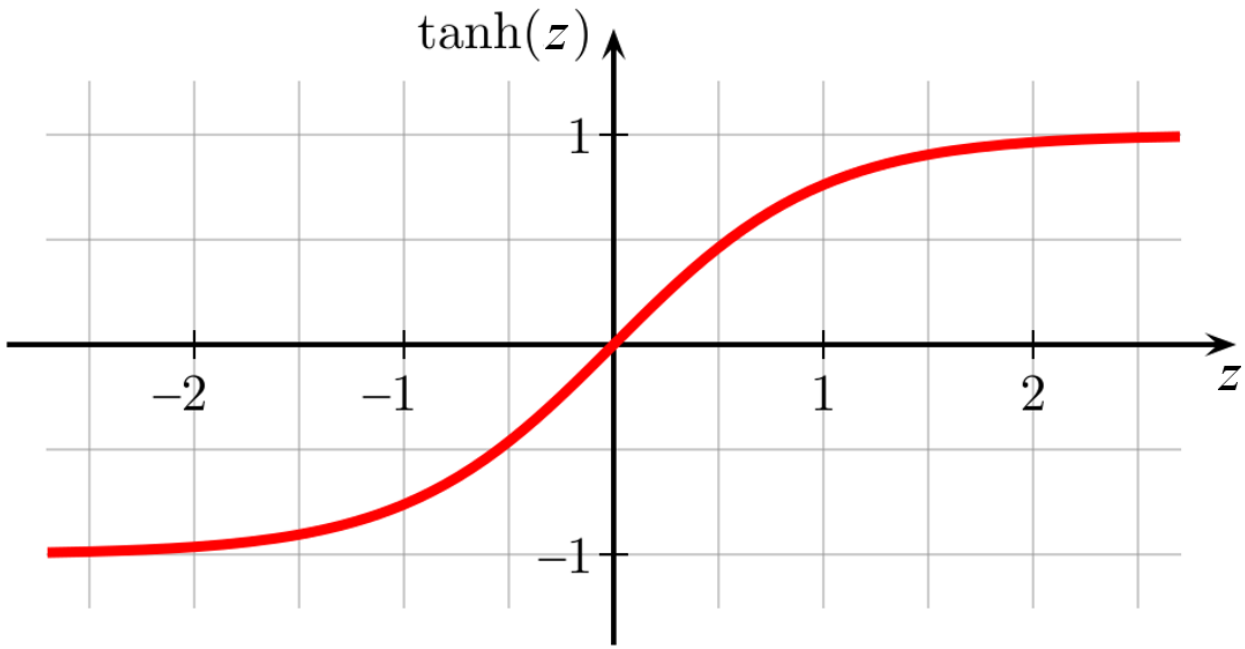
$$g(z) = \ln(1 + e^z)$$

$$g'(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$

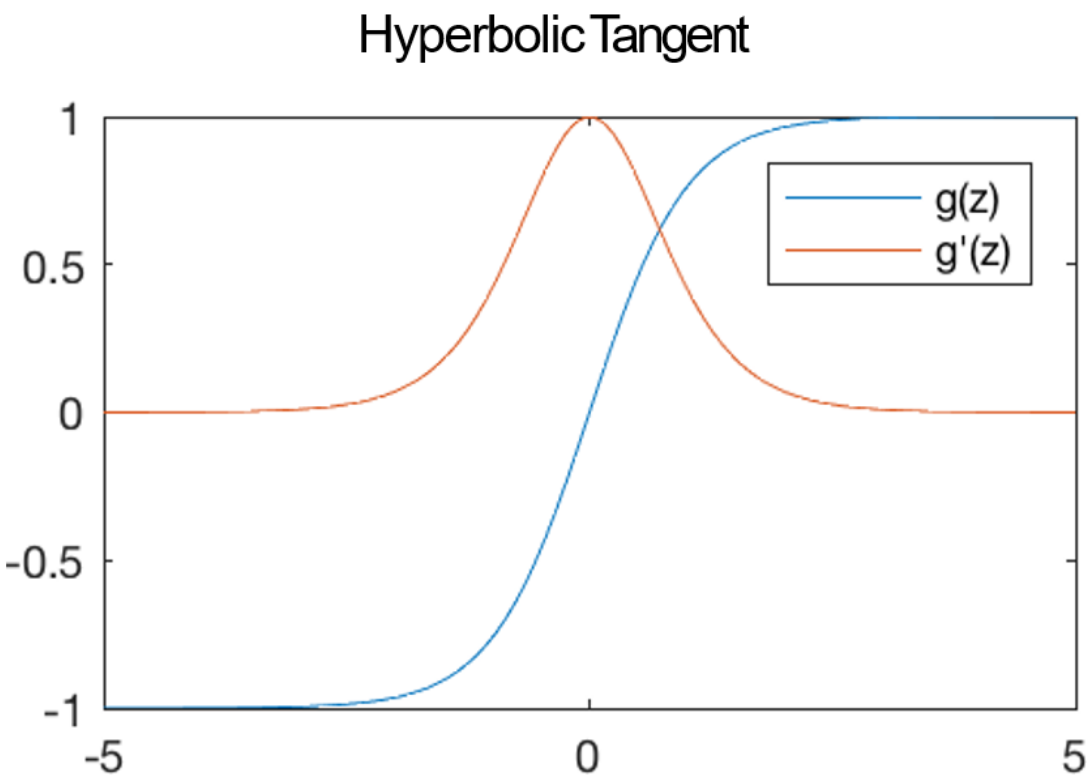
Other “activations”

- tanh (Hyperbolic function，双曲函数):

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$



Derivative of tanh



$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$g'(z) = 1 - g(z)^2$$

Other “activations”

为什么要引入激活函数?

为什么要引入激活函数?

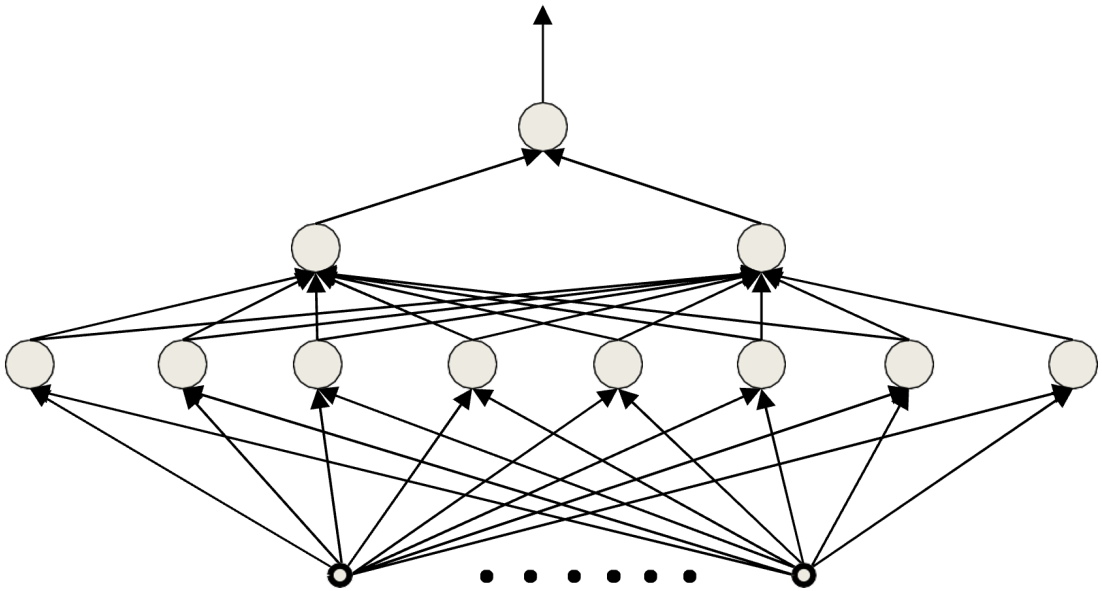
- 神经网络的激活函数（activation function）像一群空间魔法师，扭曲翻转特征空间，在其中寻找线性的边界
- 通用近似定理 (to be mentioned later)

图: From *Zhihu*

Lecture 2.2 Multi-layer Perceptron

多层感知机

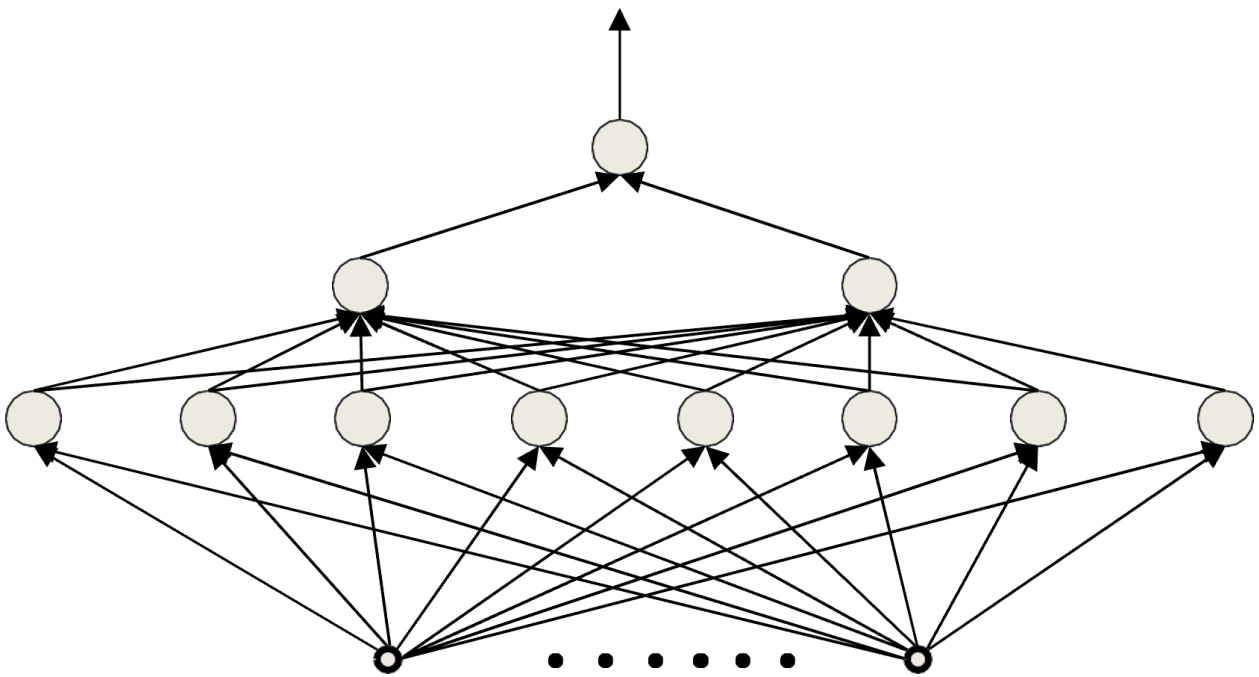
The *multi-layer* perceptron 多层感知机



- A network of perceptrons
  - Perceptrons "feed" other perceptrons
- The layer between input layer and output layer is called hidden layer (隐层或隐含层), and neurons in both hidden layer and output layer have activations
- Will give the "formal" definition of a *layer* later

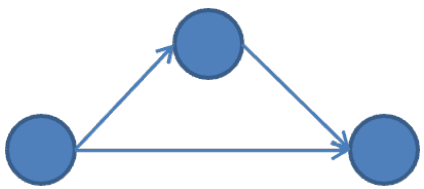
Defining "depth"

What is a "deep" network?

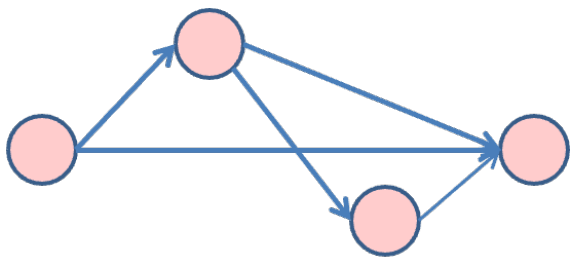


Deep Structures

- In directed graph with a source node (源节点) and a sink node (汇节点), "depth" is the length of the longest path from source to sink
  - A "source" node in a directed graph is a node that has only outgoing edges (只有出度)
  - A "sink" node is a node that has only incoming edges (入度)



Depth = 2



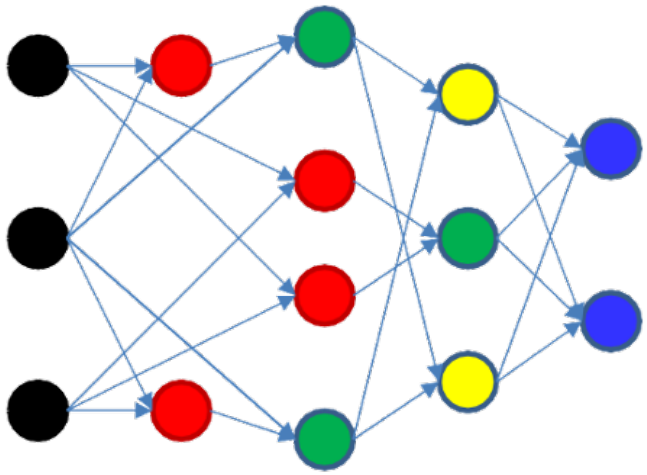
Right: Depth = 3

Deep Structures

- *Deep* structure
  - *The input is the "source",*
  - *The output nodes are "sinks"*

- Depth: 除了输入层以外的层数
- "Deep": 深度大于 2 的神经网络

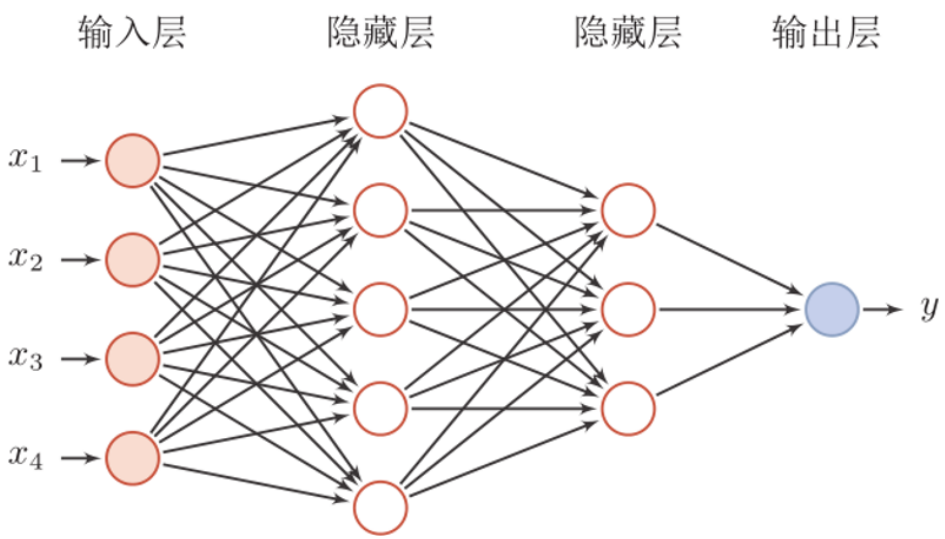
What is a layer?



- A "layer" is the set of neurons that are all at the same depth with respect to the input
  - *Depth* of a layer: the depth of neurons in the layer w.r.t. input

Feedforward Neural Networks (前馈神经网络)

- Feedforward Neural Networks （前馈神经网络、全连接神经网络、多层感知机）
  - 各神经元分别属于不同的层，层内无连接
  - 相邻两层之间的神经元全部两两连接
  - 信号从输入层向输出层单向传播，可用有向无环图表示



Universal Approximation Theorem 通用近似定理

在人工神经网络的数学理论中，通用近似定理，也称万能逼近定理（universal approximation theorem）指出，对于具有线性输出层和至少一个使用“挤压”性质的激活函数（例如sigmoid）的隐藏层组成的前馈神经网络，它能以任意的精度来近似任何从一个定义在实数空间中的有界闭集函数。

Universal Approximation Theorem

这一定理表明，只要给予了适当的参数，我们便可以通过简单的神经网络架构去拟合一些现实中非常有趣、复杂的函数。这一拟合能力也是神经网络架构能够完成现实世界中复杂任务的原因。尽管如此，此定理并没有涉及到这些参数的算法可学性 (algorithmic learnablity)。

Universal Approximation Theorem

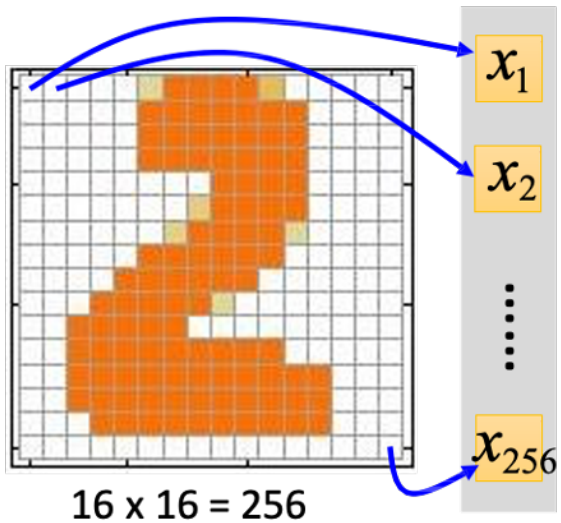
失败的可能原因：

1. 用于训练的优化算法可能找不到用于期望函数的参数值
2. 训练算法可能由于过拟合而选择了错误的函数

Lecture 2.3 Examples

An Example

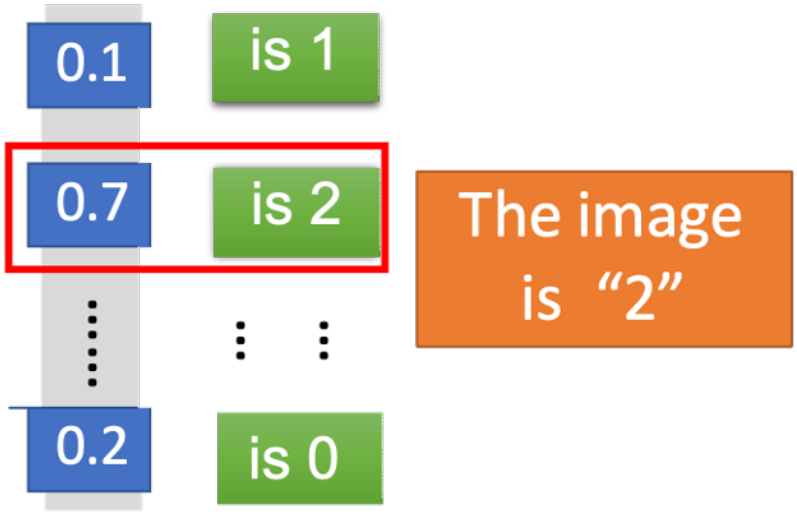
输入



Ink → 1

No ink → 0

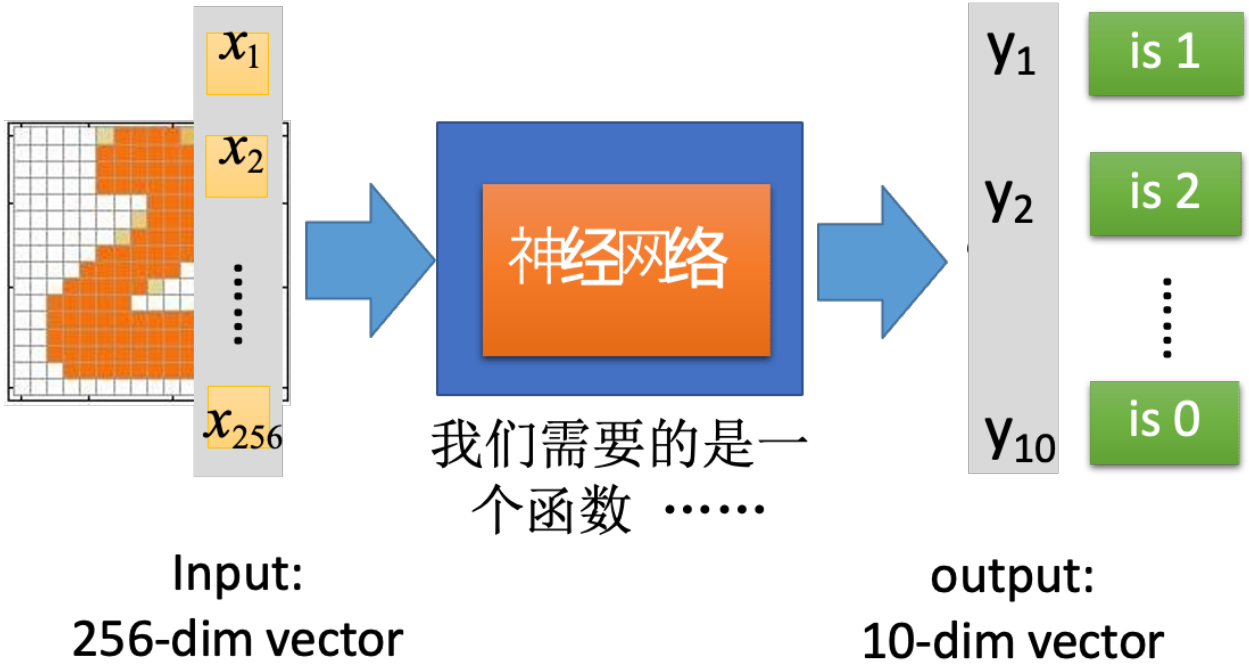
输出



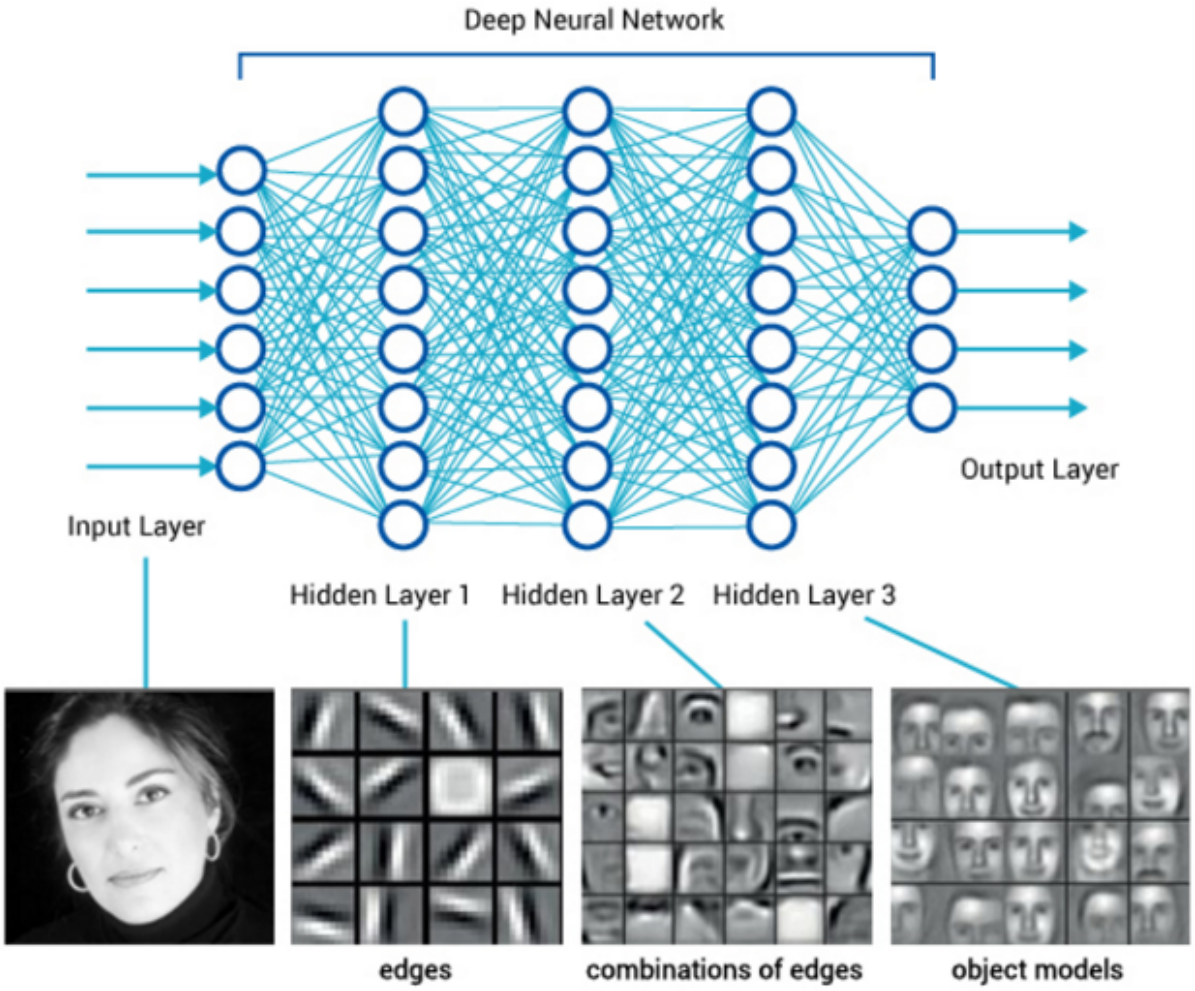
每一维表示对应数字的概率

An Example

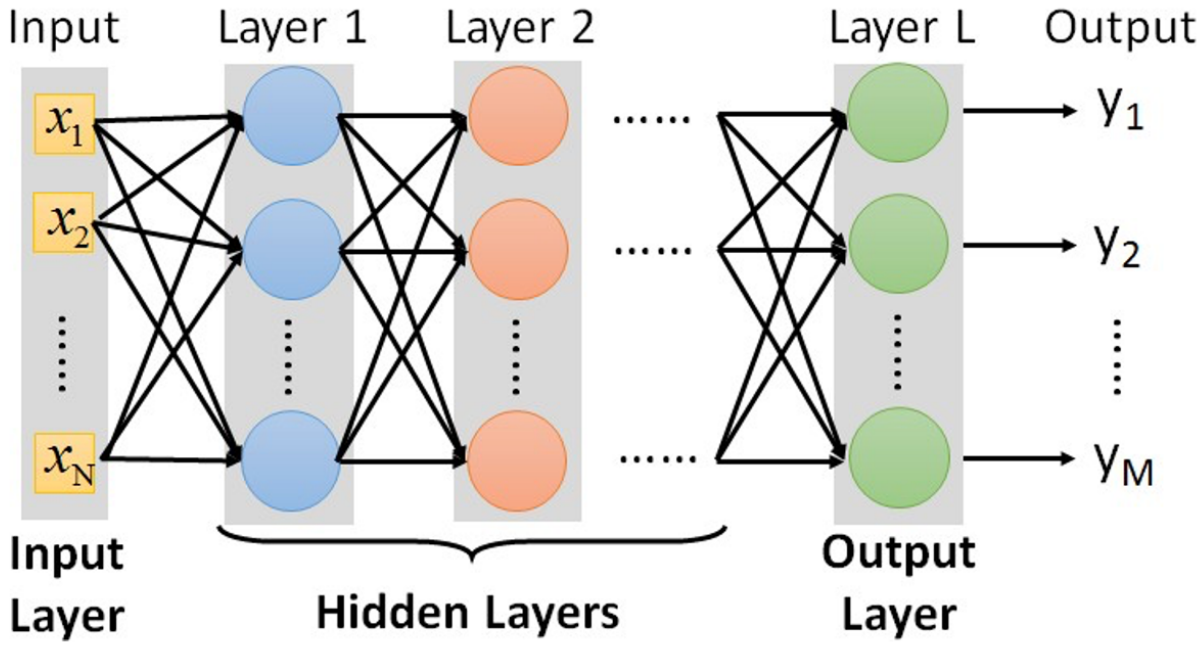
- 书写体数字识别



Another Example

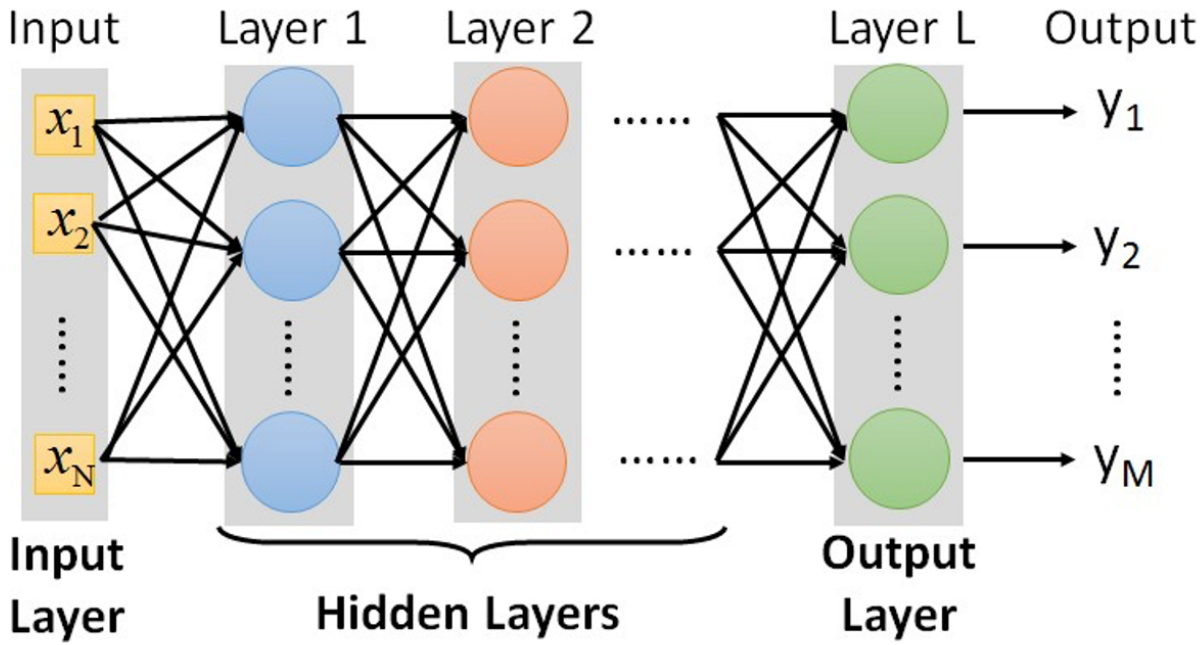


FAQ



- Q1: 神经网络需要多少层? 每层有多少个神经元?

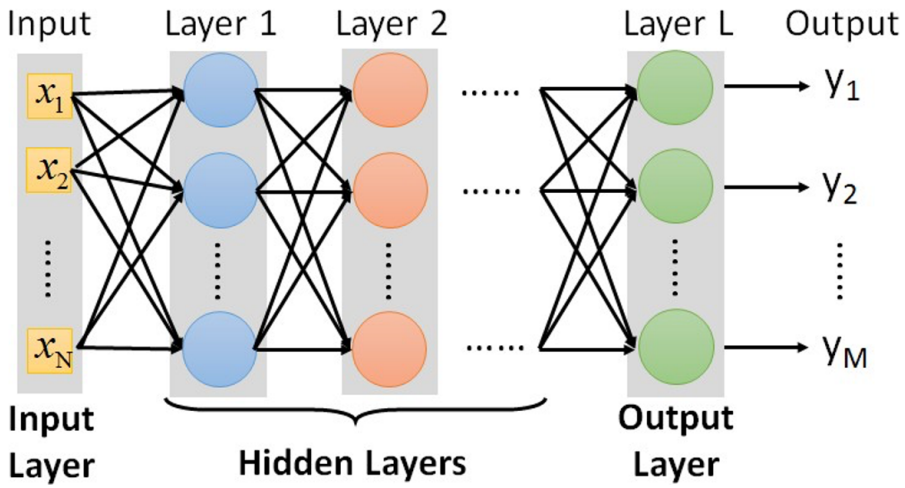
FAQ



- Q1: 神经网络需要多少层? 每层有多少个神经元?

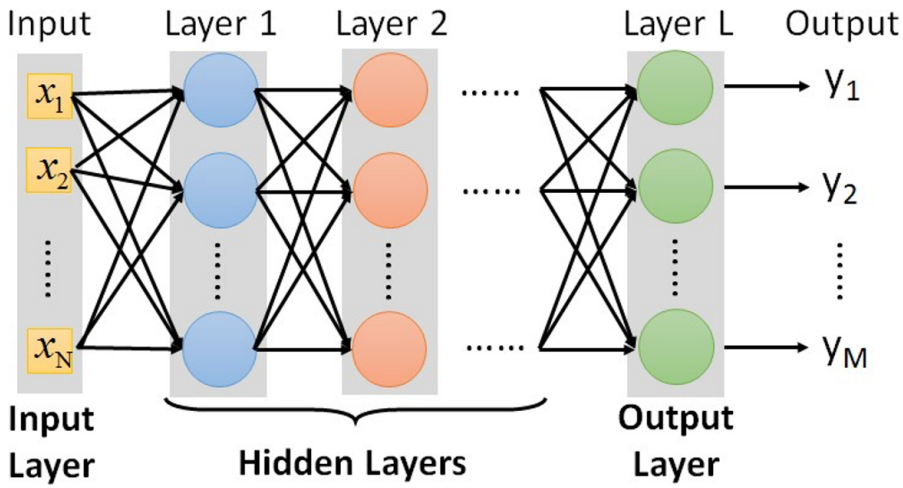
试验和错误 + 直觉

FAQ



- Q2: 网络结构是否可以自动确定?

FAQ



- Q2: 网络结构是否可以自动确定?
- A: AutoML 是一个新的研究方向，能够根据问题自动确定最优参数和网络结构

Thank you!