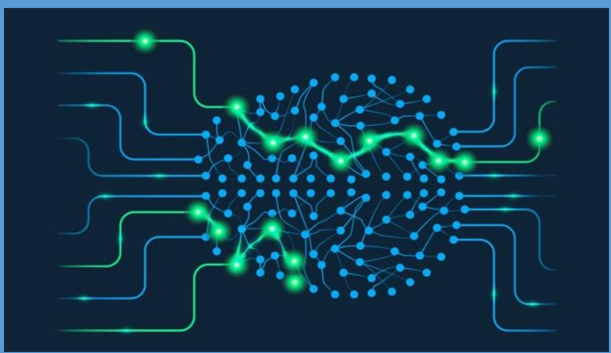
EMS702P Statistical Thinking and Applied Machine Learning

Week 8.2 – Neural Networks

Yunpeng Zhu



Neural Networks

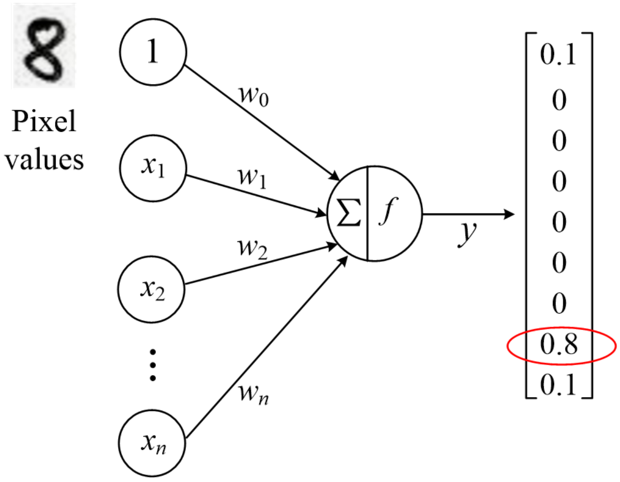
Edition: v1.1

Table of Contents

1 Multi-Layer perceptron

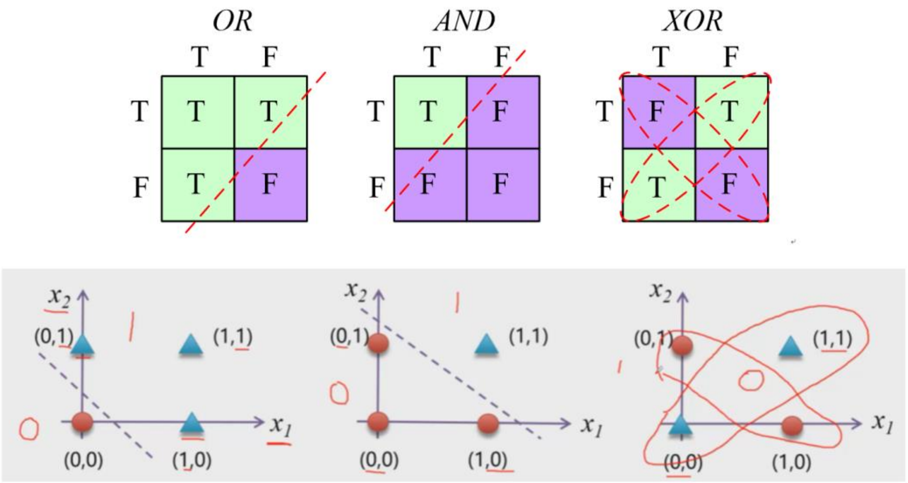
1.1 Multi-layer perceptron mechanism

We have learned perceptron, which can be used to separate two classes of data. Why don't we stop here?



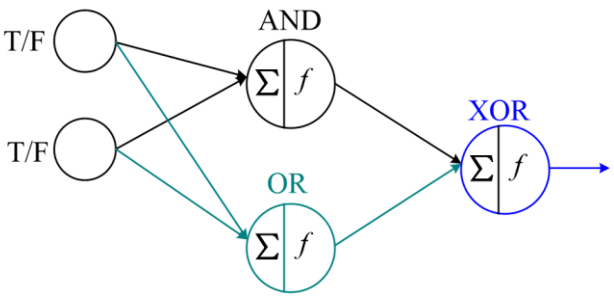
We mentioned that a perceptron can only resolve **Linearly Separable** problems. It cannot solve the eXclusive OR (XOR) problem.

XOR: False if two inputs are the same, otherwise True.

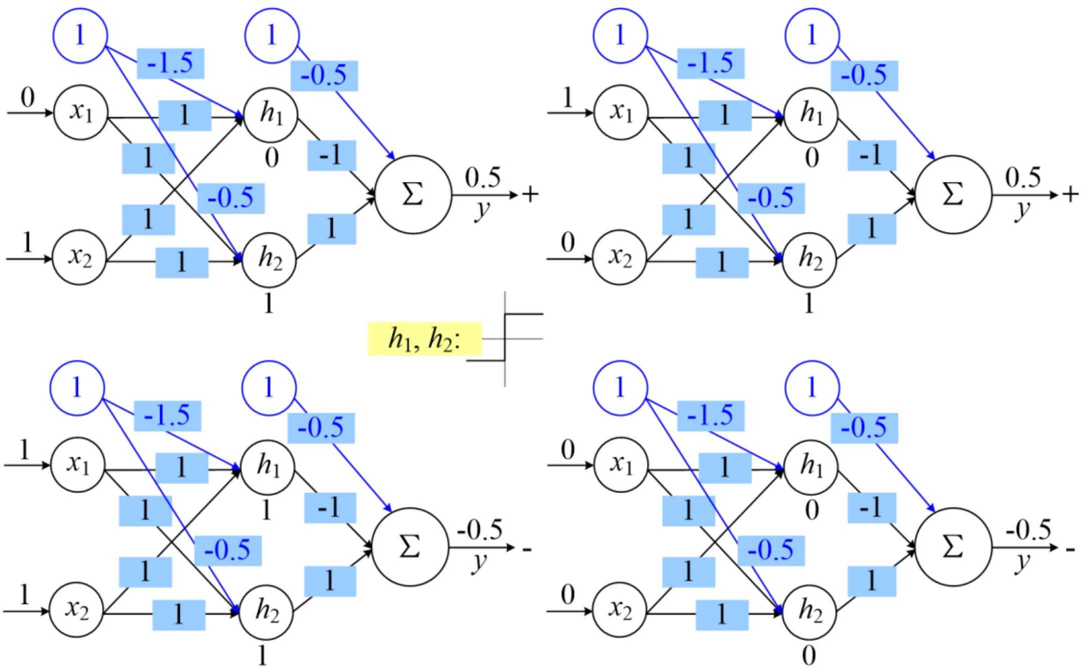


XOR problem is not linearly separable. The problem can be resolved by using multiple perceptrons, which is known as the Multi-Layer Perceptron.

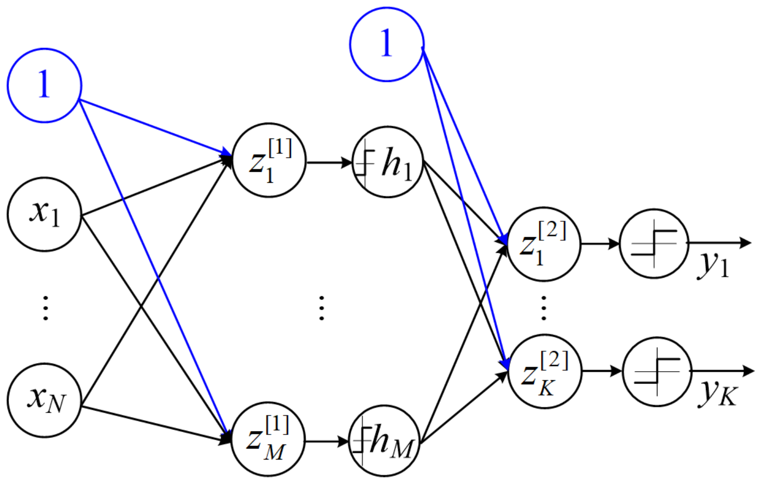
Considering one perceptron solve AND problem, and another solve OR problem. An additional perceptron for their output can be used to solve XOR problem.



For example:



A general structure of the Multi-Layer Perceptron is shown below



where

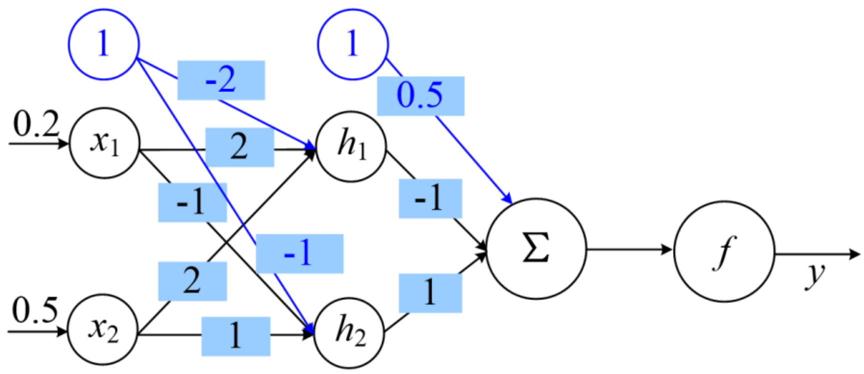
$$
\begin{cases}
\begin{cases}
z_m^{[1]} = w_{0,m}^{[1]} + w_{1,m}^{[1]} x_1 + \cdots + w_{N,m}^{[1]} x_N = w_{0,m}^{[1]} + \sum_{n=1}^{N} w_{n,m}^{[1]} x_n \\
h_m = f(z_m^{[1]})
\end{cases}, m = 1, \ldots, M \\
\begin{cases}
z_k^{[2]} = w_{0,k}^{[2]} + w_{1,k}^{[2]} h_1 + \cdots + w_{M,k}^{[2]} h_M = w_{0,k}^{[2]} + \sum_{m=1}^{M} w_{m,k}^{[2]} h_m \\
y_k = f(z_k^{[2]})
\end{cases}, k = 1, \ldots, K
\end{cases}
$$

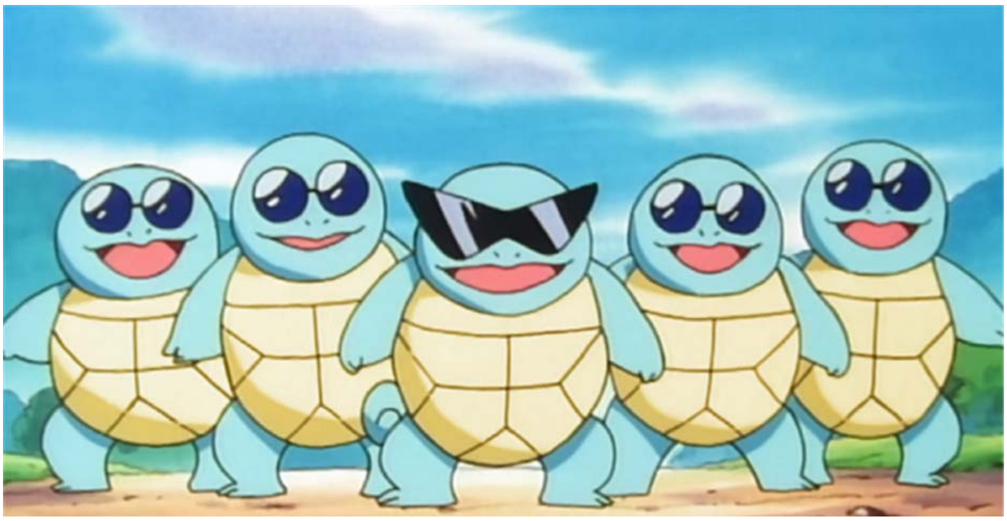
with

- $w_{nm}^{[1]}$ representing the weights of the first connections from the nth node of the input layer to the mth node of the middle layer, while
- $w_{mk}^{[2]}$ representing the weights of the second connections from the mth node of the middle layer to the kth node of the output layer.
- $f(.)$ is a step function

Quiz 1.1:

Calculate the output of the multiple layer perceptron:

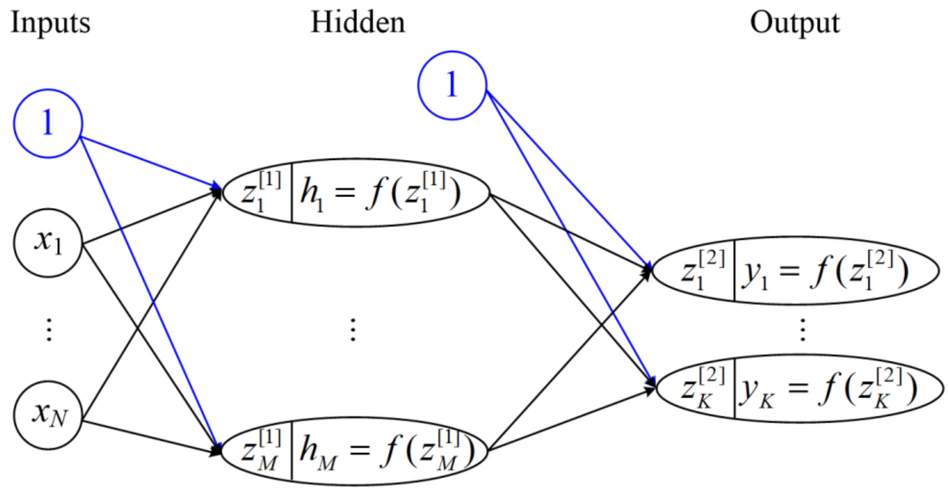




2 Fully connected feed-forward artificial neural network

2.1 Structure of the feed-forward artificial neural network

In a Multi-Layer Perceptron each layer's outputs are activated by an activation function. The new network is a typical structure of neural network.



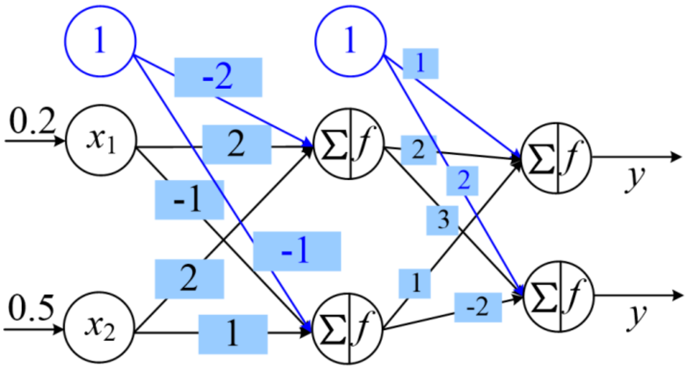
In a neural network, the first layer is the **input layer**, the middle layer is the **hidden layer**, and the final layer is the **output layer**. The network in the plot may be described as a **2-layer network**, because it is the number of layers of adaptive weights that is important for determining the network properties.

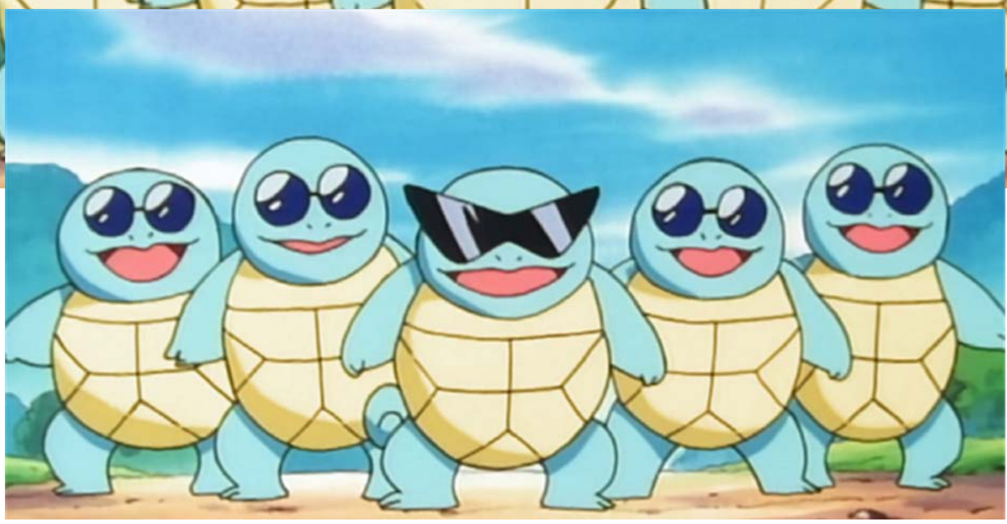
$$
\begin{cases}
\begin{cases}
z_m^{[1]} = w_{0,m}^{[1]} + w_{1,m}^{[1]} x_1 + \cdots + w_{N,m}^{[1]} x_N = w_{0,m}^{[1]} + \sum_{n=1}^{N} w_{n,m}^{[1]} x_n \\
h_m = f(z_m^{[1]})
\end{cases} , m = 1, \ldots, M \\
\begin{cases}
z_k^{[2]} = w_{0,k}^{[2]} + w_{1,k}^{[2]} h_1 + \cdots + w_{M,k}^{[2]} h_M = w_{0,k}^{[2]} + \sum_{m=1}^{M} w_{m,k}^{[2]} h_m \\
y_k = f(z_k^{[2]})
\end{cases} , k = 1, \ldots, K
\end{cases}
$$

where $f(.)$ is an activation function

It can be seen that a Multi-Layer Perceptron is a special case of the neural network.

Quiz 2.1: the activation functions are Sigmoid functions. Calculate the outputs:

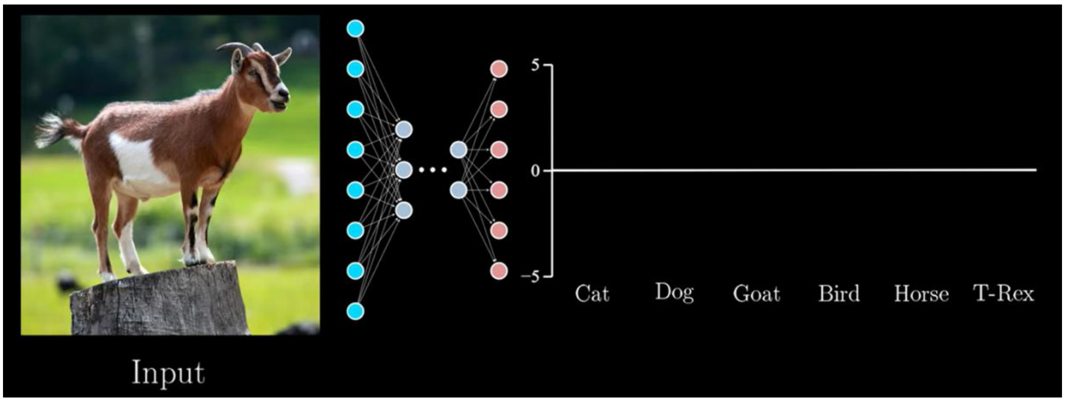


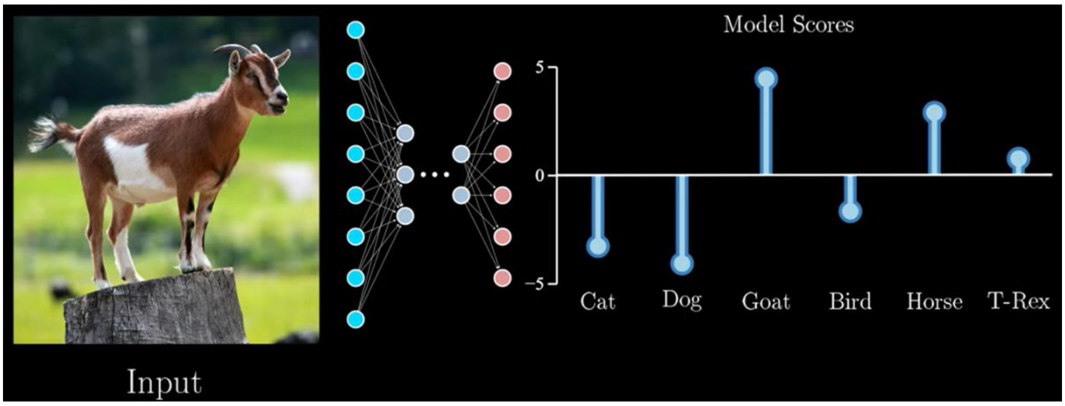


2.2 Neural network based classification and regression

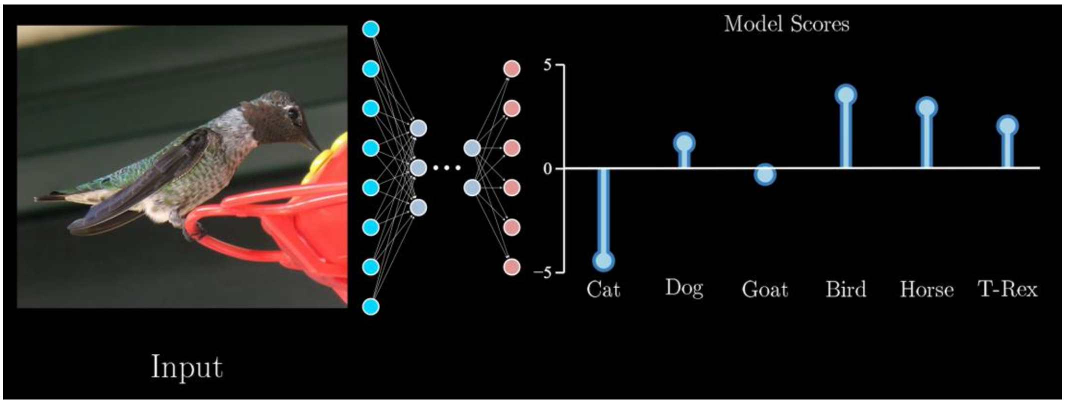
(1) Classification

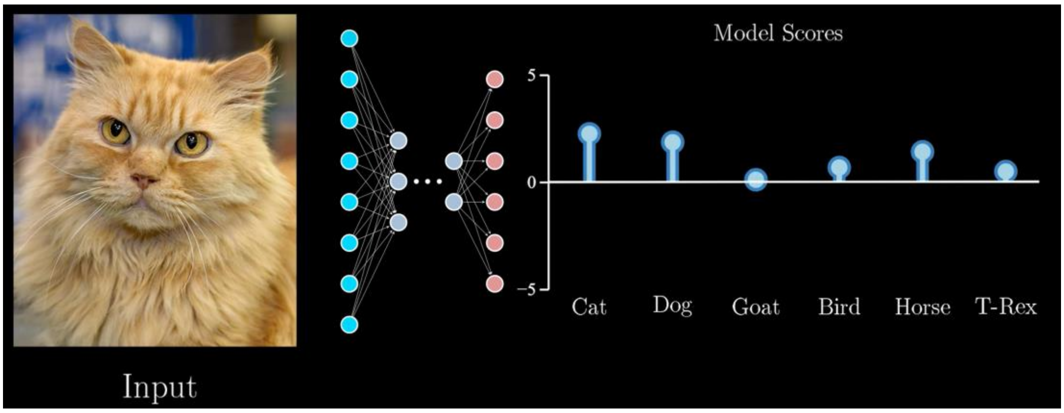
As we have learned in the linear classification, multiple linear functions can also be used to solve a K-class classification problem based on one-vs-the rest approach. However, many classification problems are not **linearly separable**. Neural network provide a nonlinear classifier to solve this problem as illustrated in the Multi-Layer Perceptron.

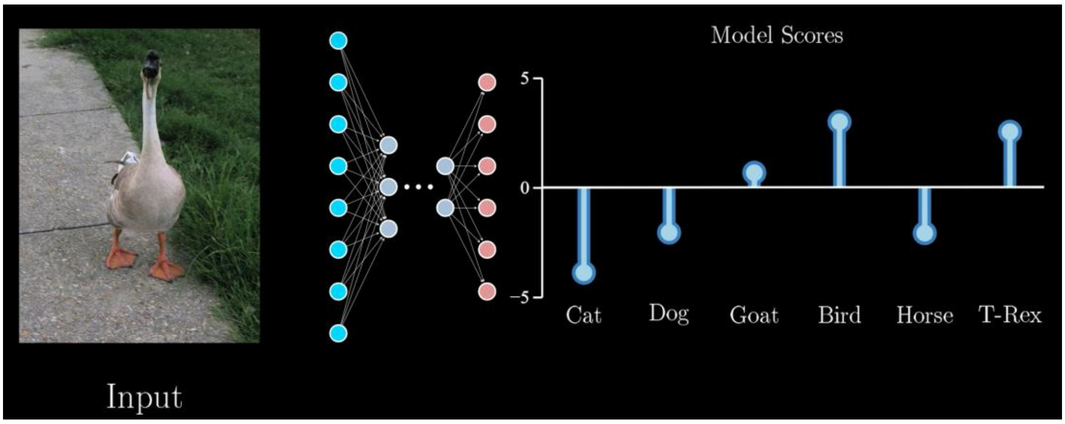


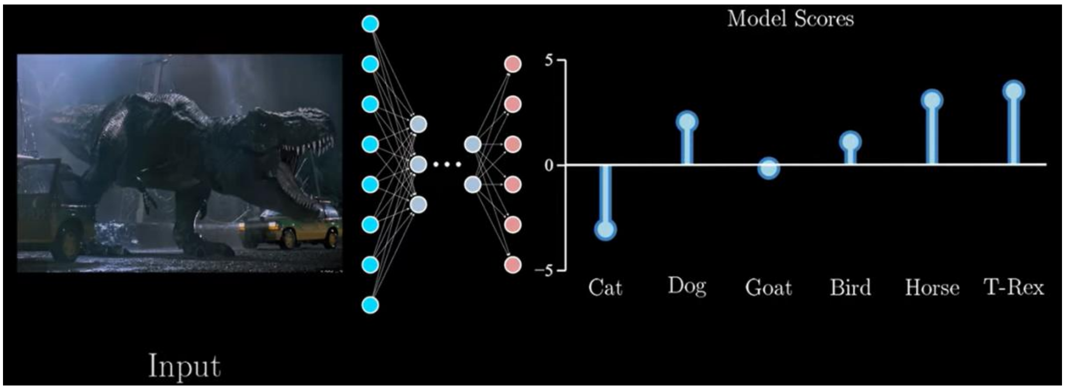


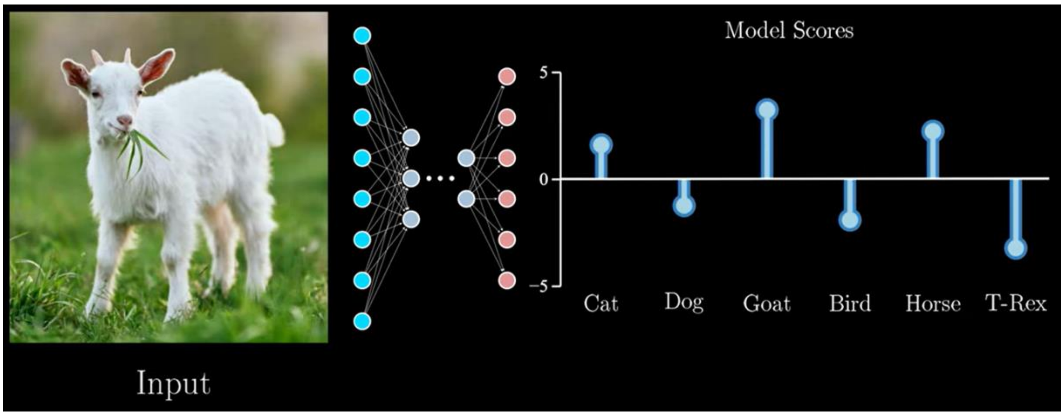
For different inputs:





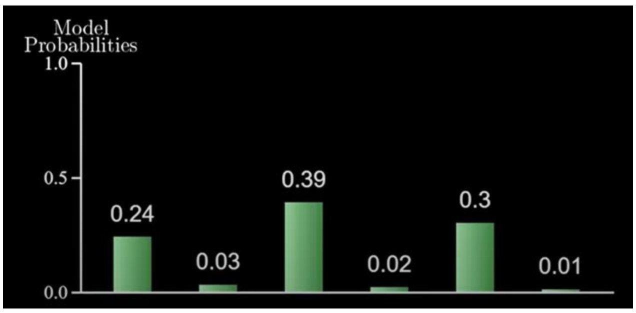


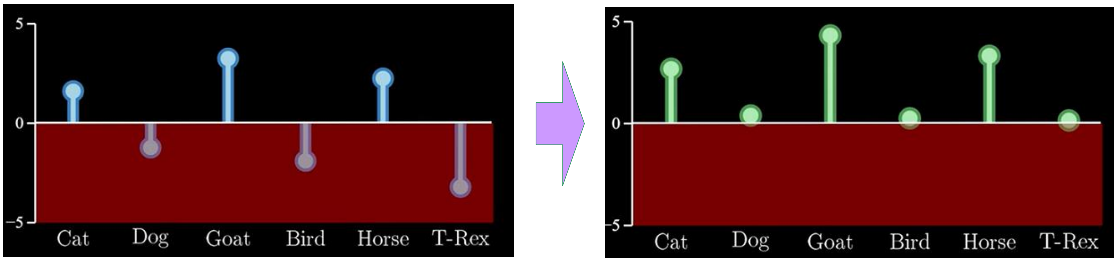




This is **incomplete**. Very often, these scores need to be mapped to **probabilities**.

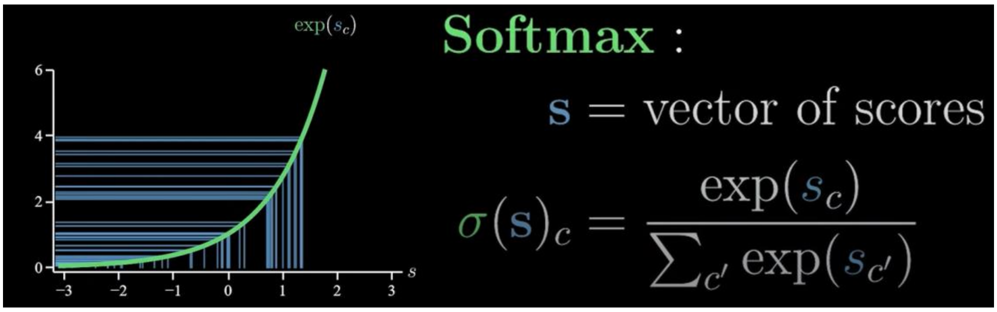
- Sum of the probabilities is 1.
- The mapping function is differentiable.





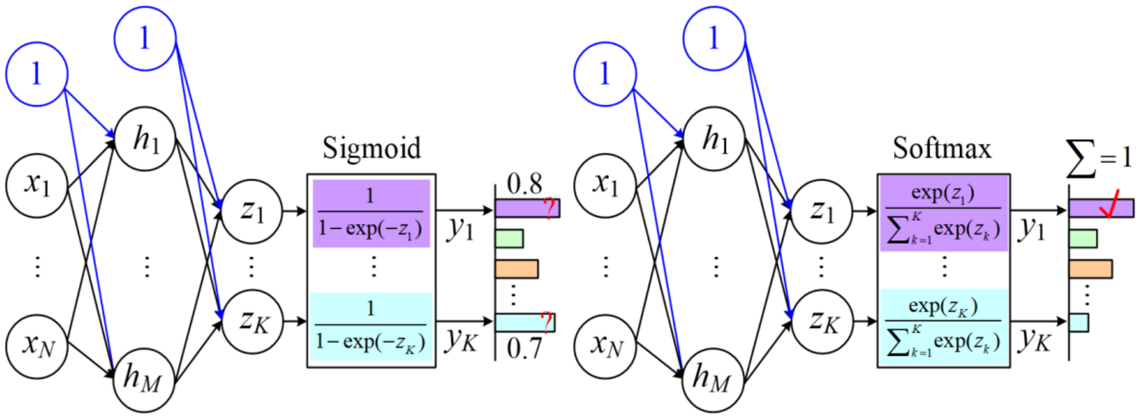
In the neural network based classification, the activation functions of the output layer are often defined as a Softmax functions:

$$f(z_k) = \frac{\exp(z_k)}{\sum_{k=1}^{K} \exp(z_k)}$$

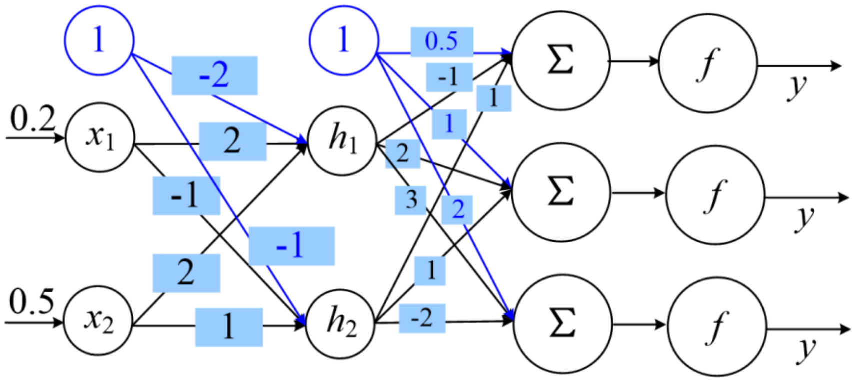


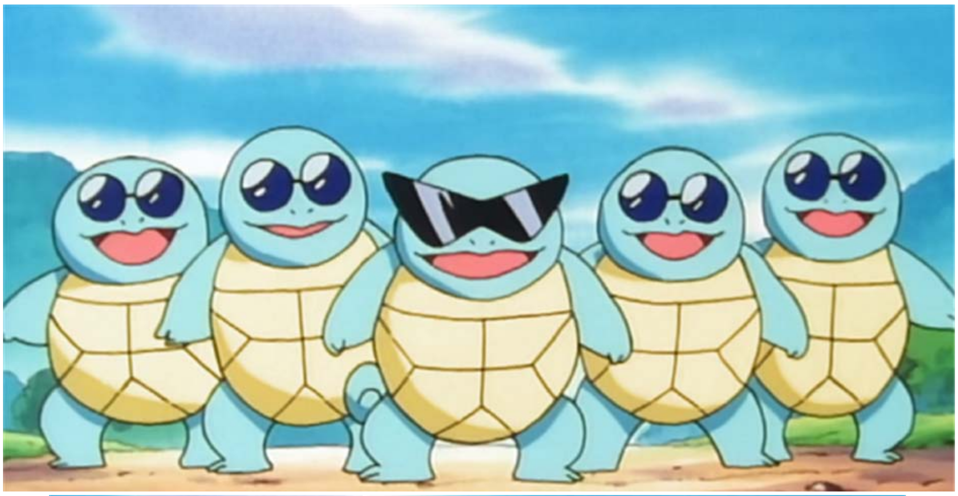
Compared with simply using a Sigmoid function, Softmax function can provide a single classification result for each input set. Therefore, Softmax function is often used to solve multi-class classification problems (i.e. cat/dog), while Sigmoid function is often used to solve multi-label classification problems (i.e. 4 legs/tail/fur) [4].

For a 2-class classification problem, the Softmax function is the same as the Sigmoid function. A Softmax function has the value from 0 to 1, representing the probability of each class from all classes.



Quiz 2.2: f is the softmax function, the activation function are step functions. Calculate the outputs:

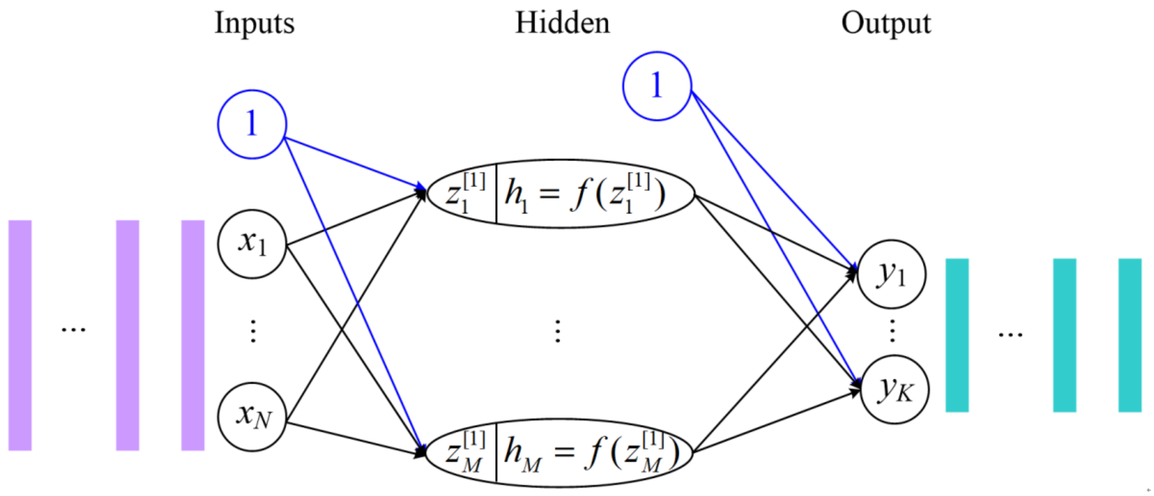




(2) Regression

The introduction of neural networks is often based on classification problems. How to use a neural network to solve regression problems?

From the aspect of network structures, neural network for regression doesn't use activation function in the output layer.



Python: FFNN_221.py

FFNN_Sine_181.py