KonvergeAI

CCT 303 AI&CS UNIT 2 Supervised Learning

Linear Threshold Unit & Perceptron

Let us see this video and discuss

https://www.youtube.com/watch?v=-KLnurhX-Pg

LTU = A linear threshold unit is a simple artificial neuron whose output is its thresholded total net input. That is, an LTU with threshold T calculates the weighted sum of its inputs, and then outputs 0 if this sum is less than T, and 1 if the sum is greater than T. LTU's form the basis of perceptrons.

Let's understand this better with an example. Say you bike to work. You have two factors to make your decision to go to work: the weather must not be bad, and it must be a weekday. The weather's not that big a deal, but working on weekends is a big no-no. The inputs have to be binary, so let's propose the conditions as yes or no questions. Weather is fine? 1 for yes, 0 for no. Is it a weekday? 1 yes, 0 no.

Remember, I cannot tell the neural network these conditions; it has to learn them for itself. How will it know which information will be most important in making its decision? It does with something called **weights**. Remember when I said that weather's not a big deal, but the weekend is? Weights are just a numerical representation of these preferences. A higher weight means the neural network considers that input more important compared to other inputs.

For our example, let's purposely set suitable weights of 2 for weather and 6 for weekday. Now how do we calculate the output? We simply multiply the input with its respective weight, and sum up all the values we get for all the inputs. For example, if it's a nice, sunny (1) weekday (1), we would do the following calculation:

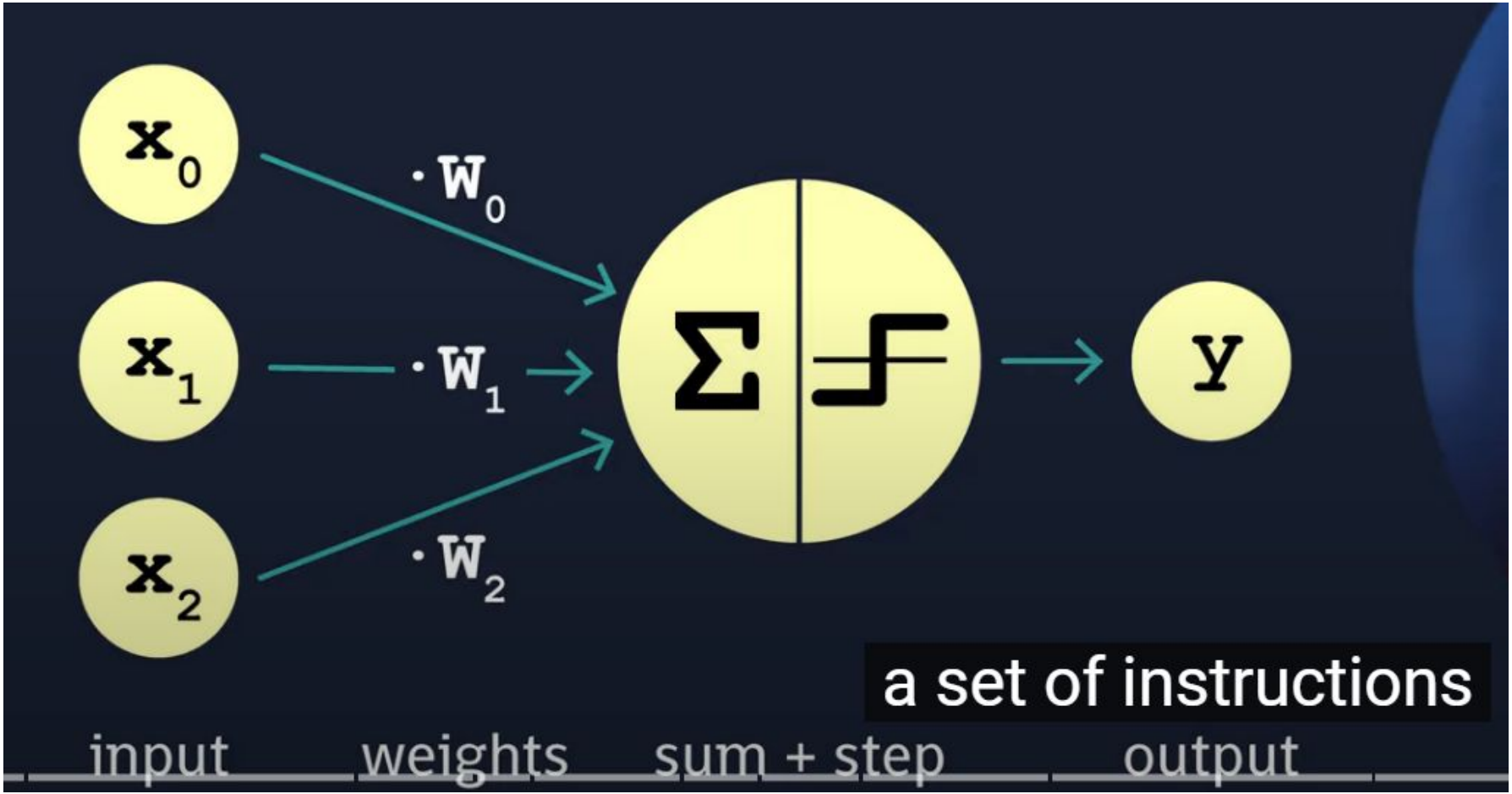
$$total = (nice..weather?) \times (weight_1) + (weekday?) \times (weight_2)$$

$$total = (1 \times 2) + (1 \times 6) = 8$$

This calculation is known as a **linear combination**. Now what does an 8 mean? We first need to define the **threshold value.** The neural network's output, 0 or 1 (stay home or go to work), is determined if the value of the linear combination is greater than the threshold value. Let's say the threshold value is 5, which means that if the calculation gives you a number less than 5, you can stay at home, but if it's equal to or more than 5, then you gotta go to work.

You have just seen how weights are influential in determining the output. In this example, I set the weights to particular numbers that make the example work, but in reality, we set the weights to random values, and then the network adjusts those weights based on the output errors it made using the previous weights. This is called **training** the neural network.

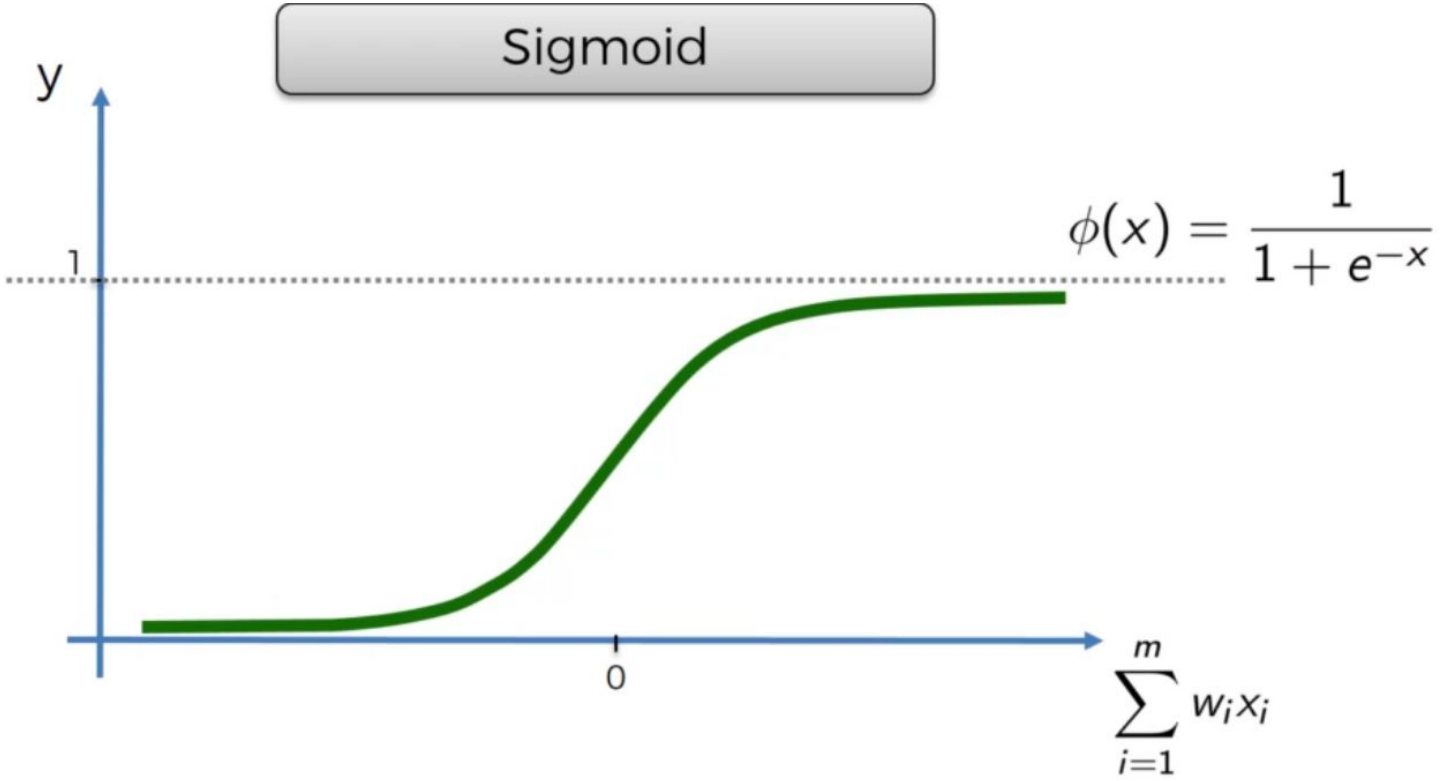
MultiLayer Networks & Backpropagation


x_0
x_1
x_2
$\cdot W_0$
$\cdot W_1$
$\cdot W_2$
Σ
y
a set of instructions
input
weights
sum + step
output

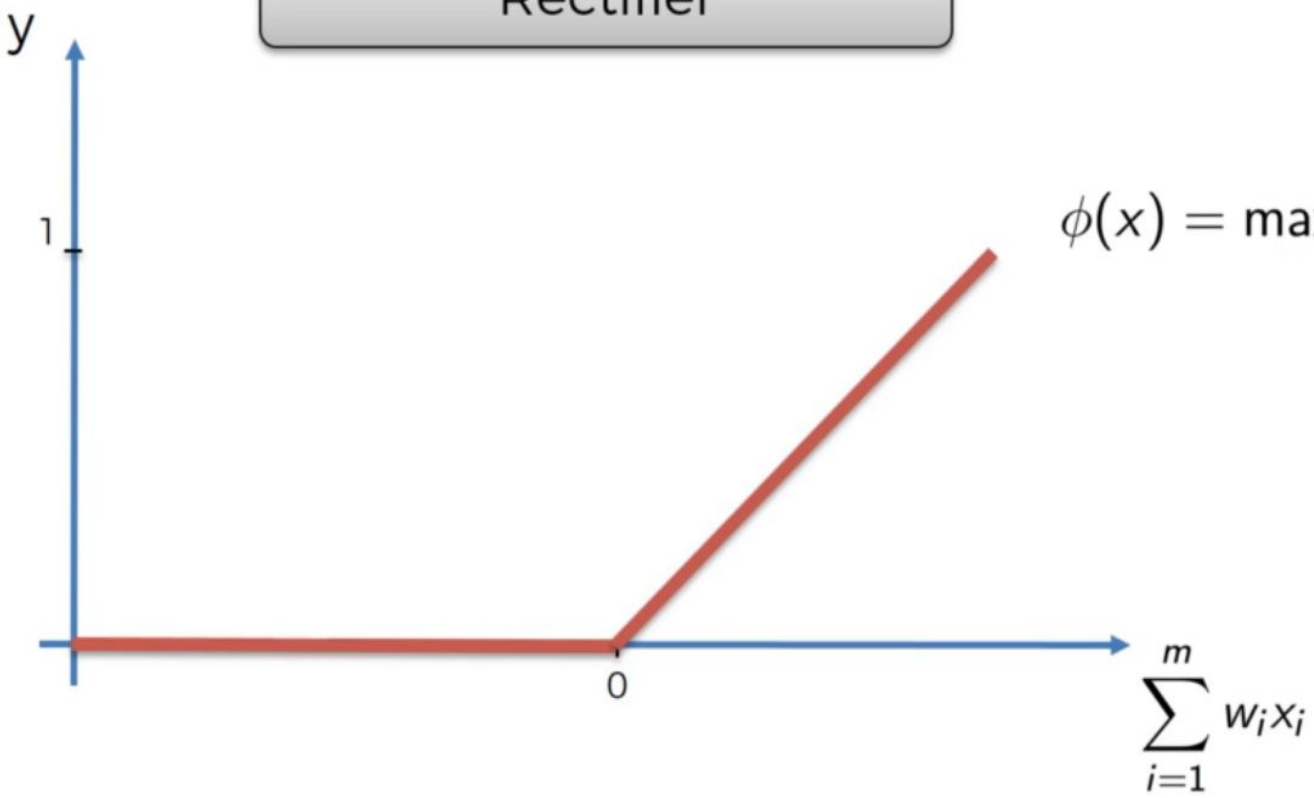
Perceptron Components

Threshold

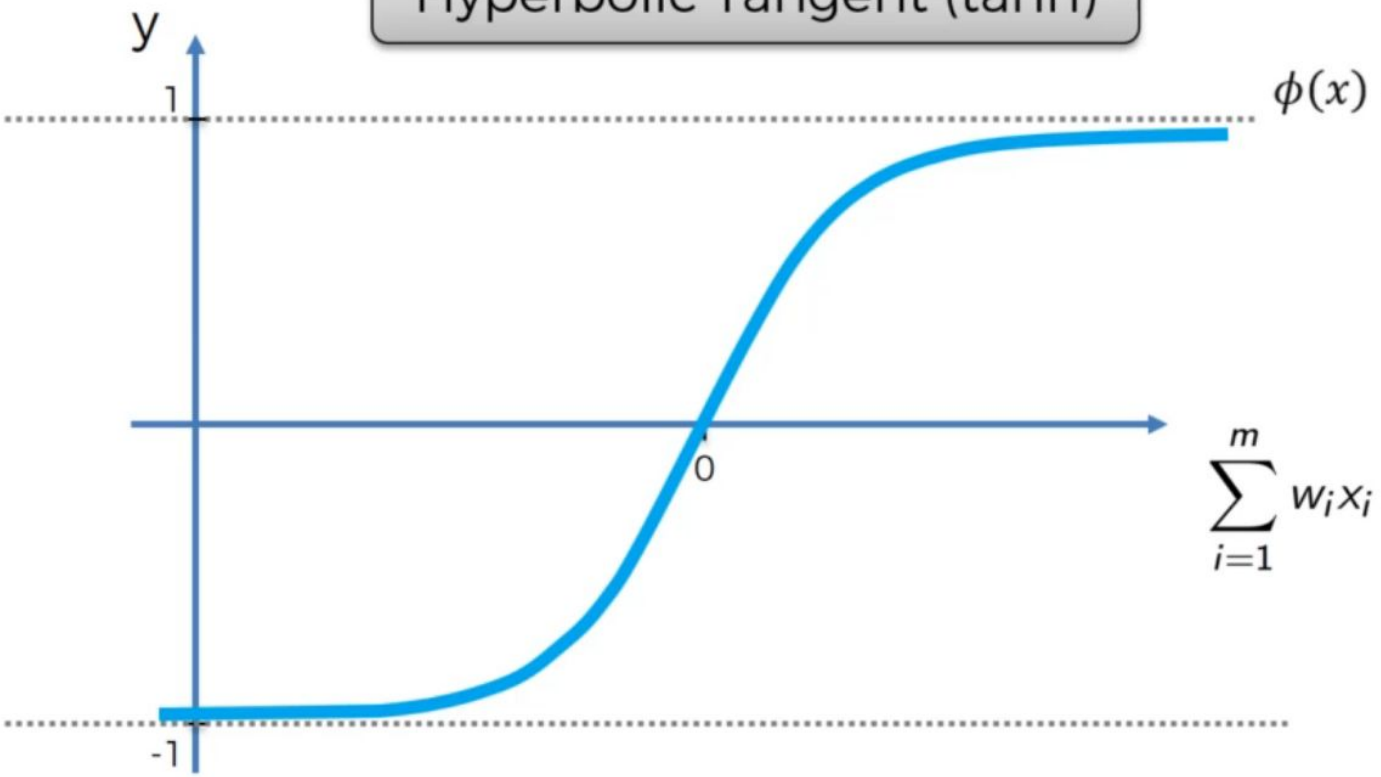
Activation Functions



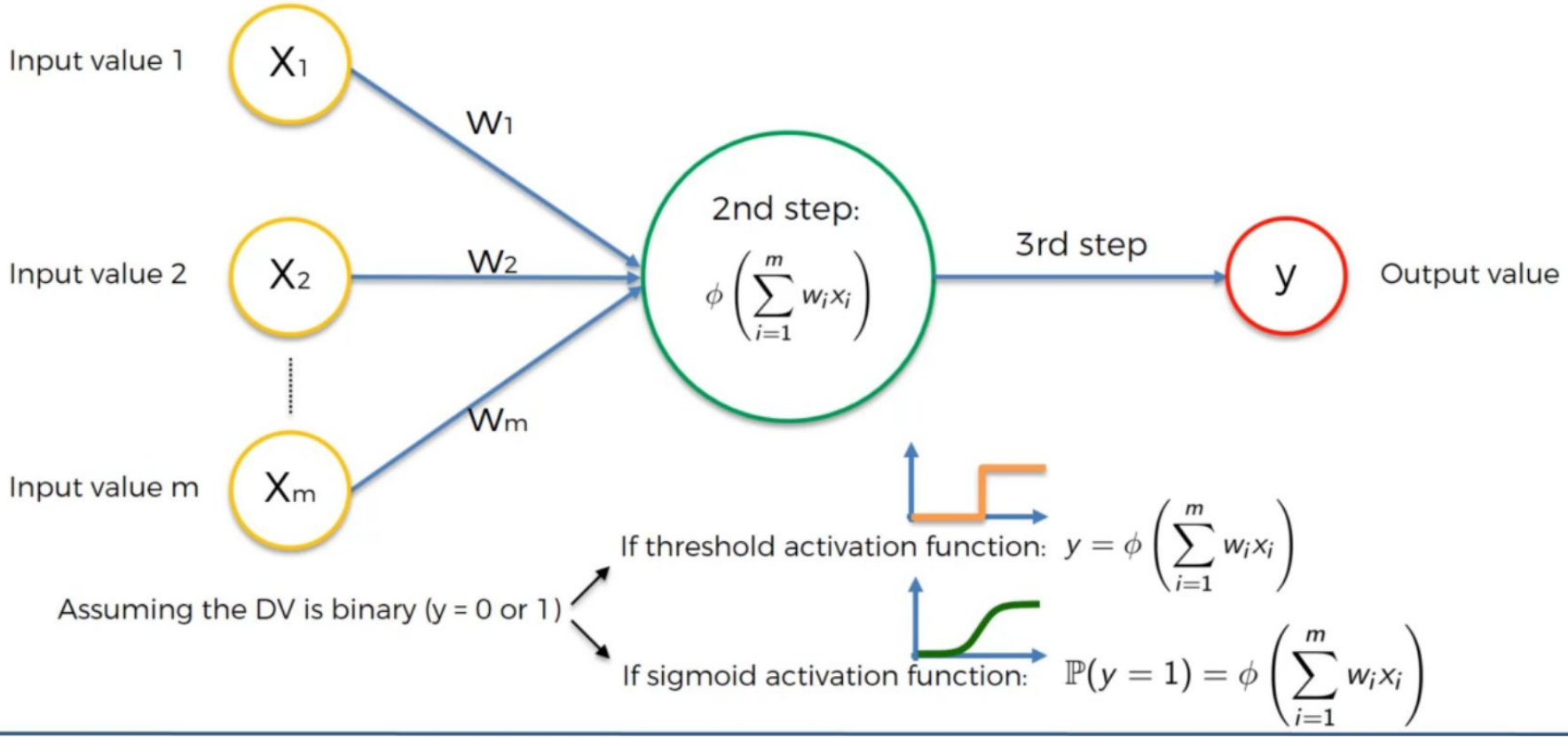
Rectifier

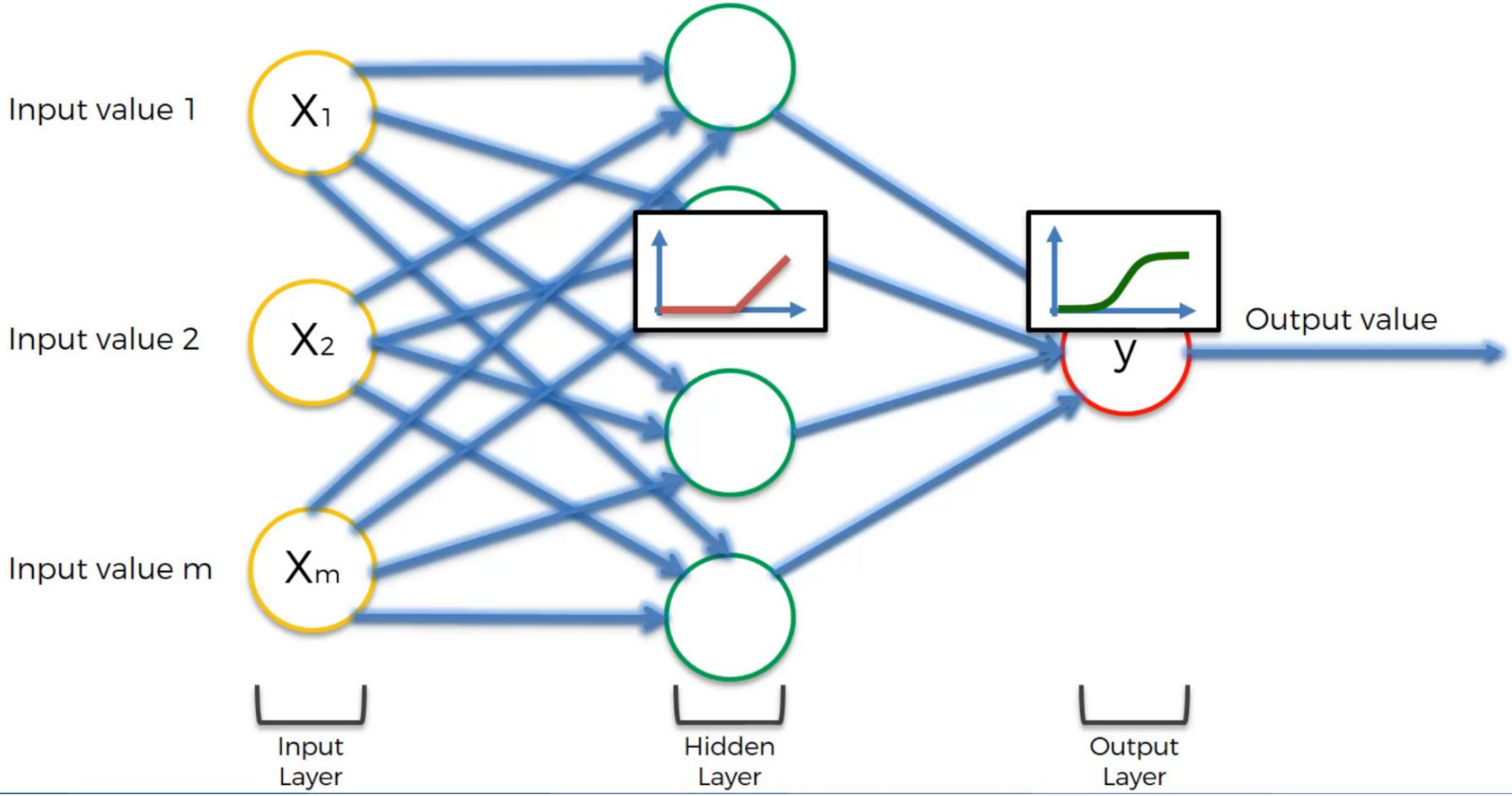


Hyperbolic Tangent (tanh)

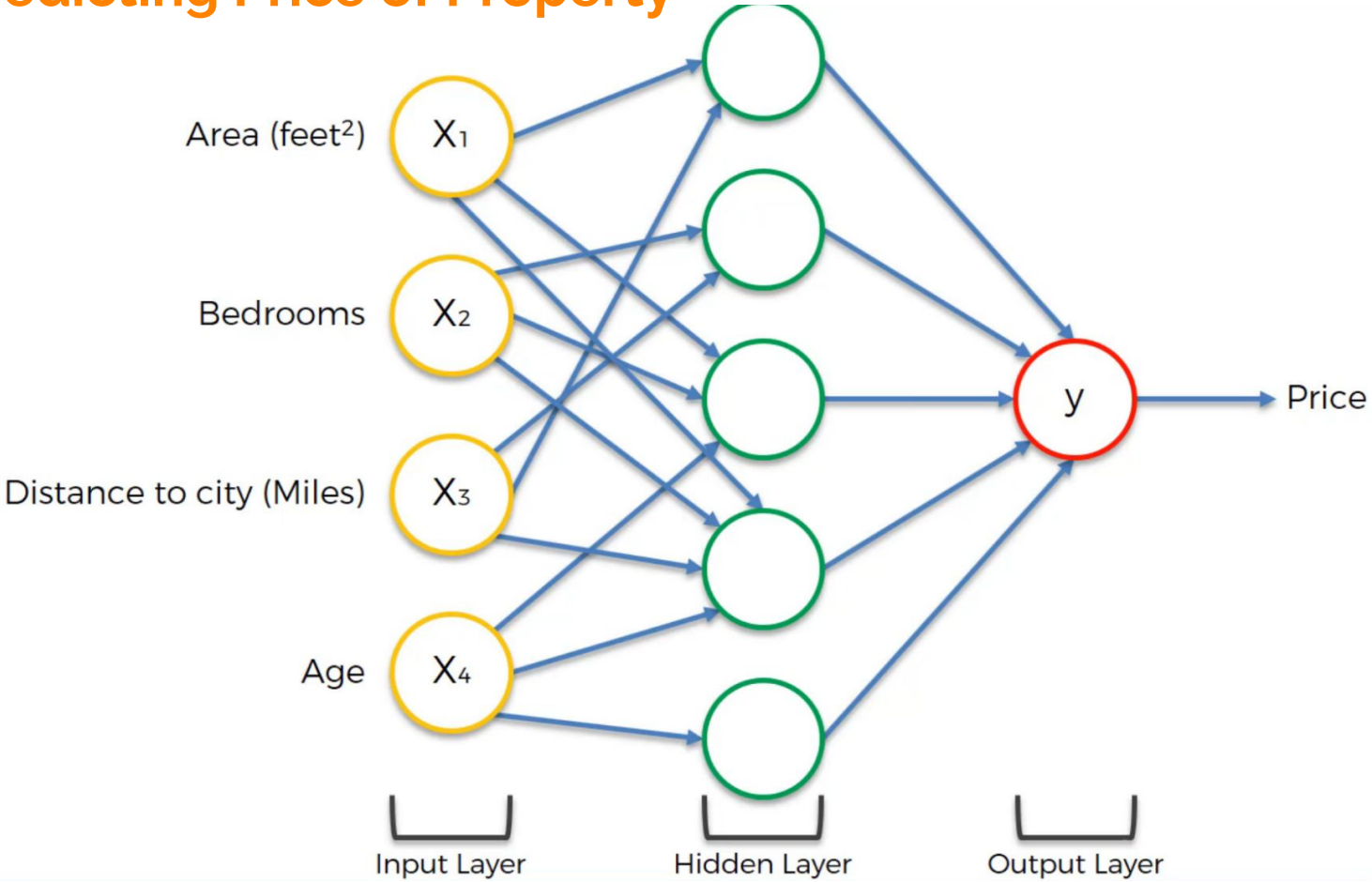


$$\phi(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

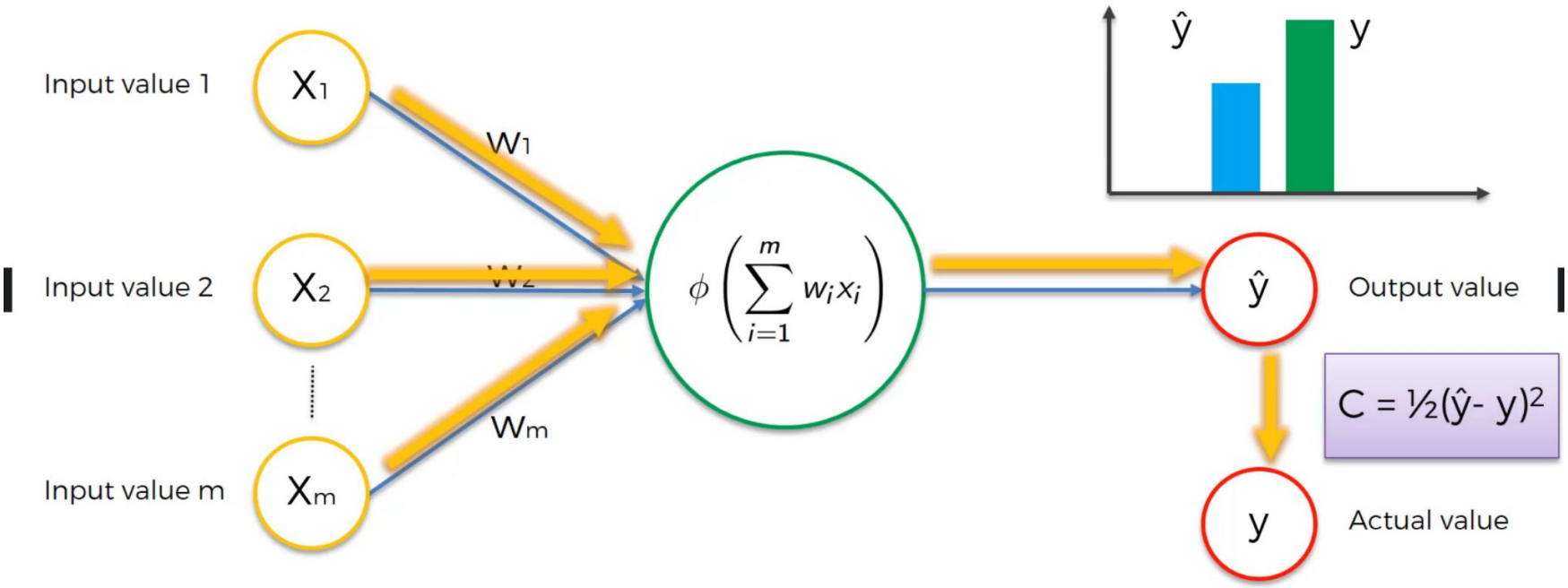

Input value 1
X1
W1
Input value 2
X2
W2
Input value m
Xm
Wm
2nd step:
$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$
3rd step
y
Output value
Assuming the DV is binary (y = 0 or 1)
If threshold activation function: $y = \phi\left(\sum_{i=1}^{m} w_i x_i\right)$
If sigmoid activation function: $\mathbb{P}(y = 1) = \phi\left(\sum_{i=1}^{m} w_i x_i\right)$


Input value 1
X1
Input value 2
X2
Input value m
Xm
y
Output value
Input Layer
Hidden Layer
Output Layer

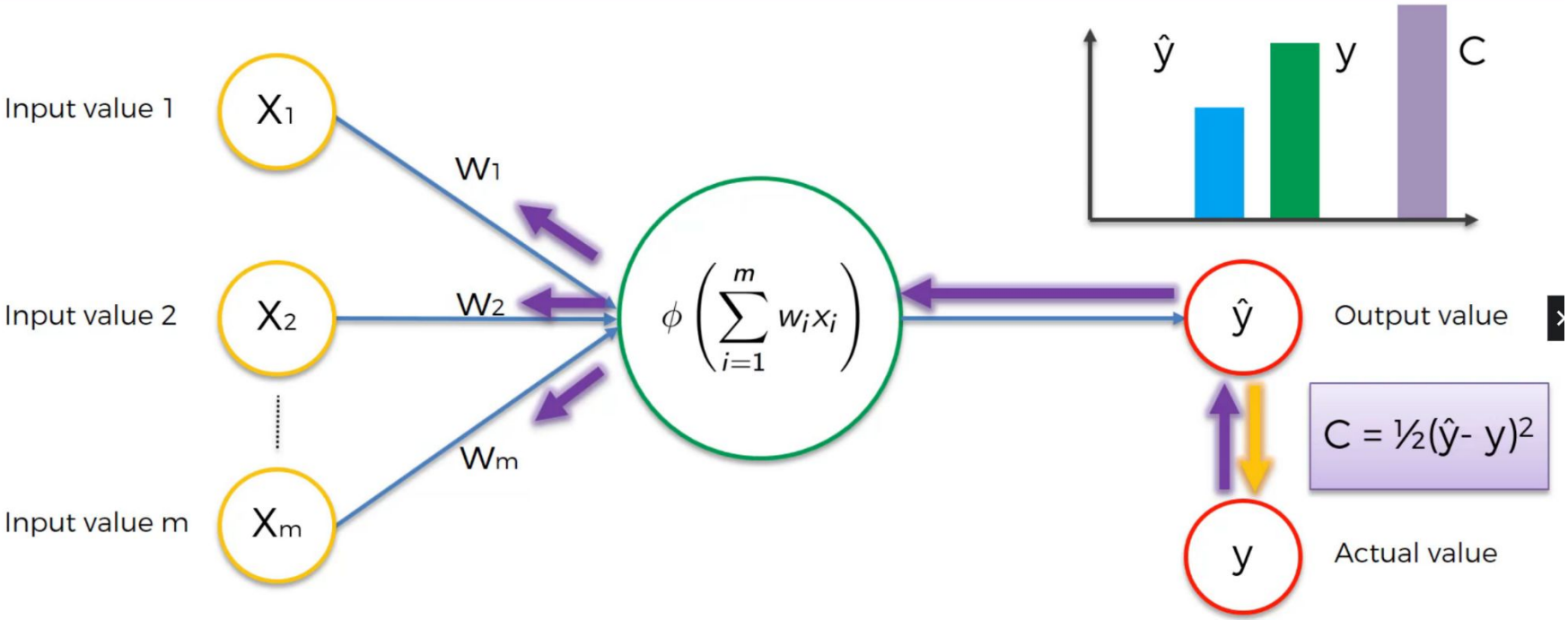
Predicting Price of Property



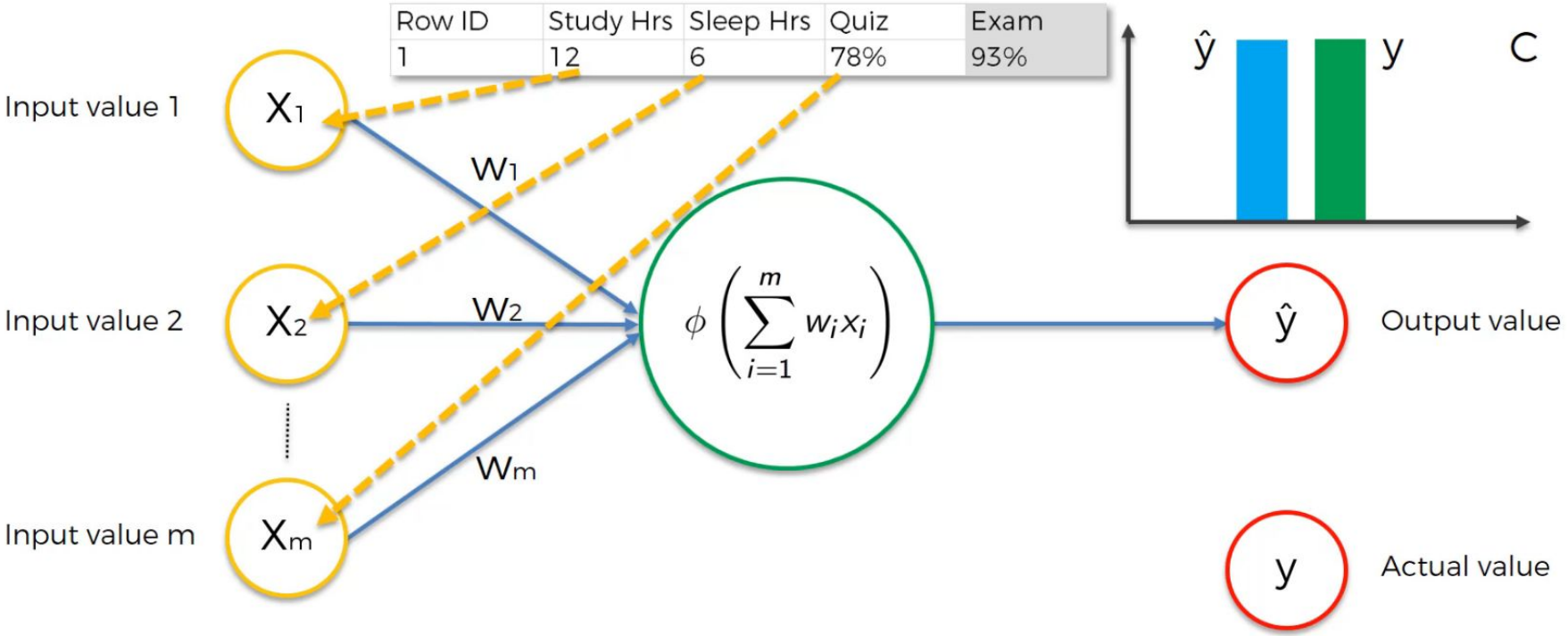
How do Neural Networks learn?



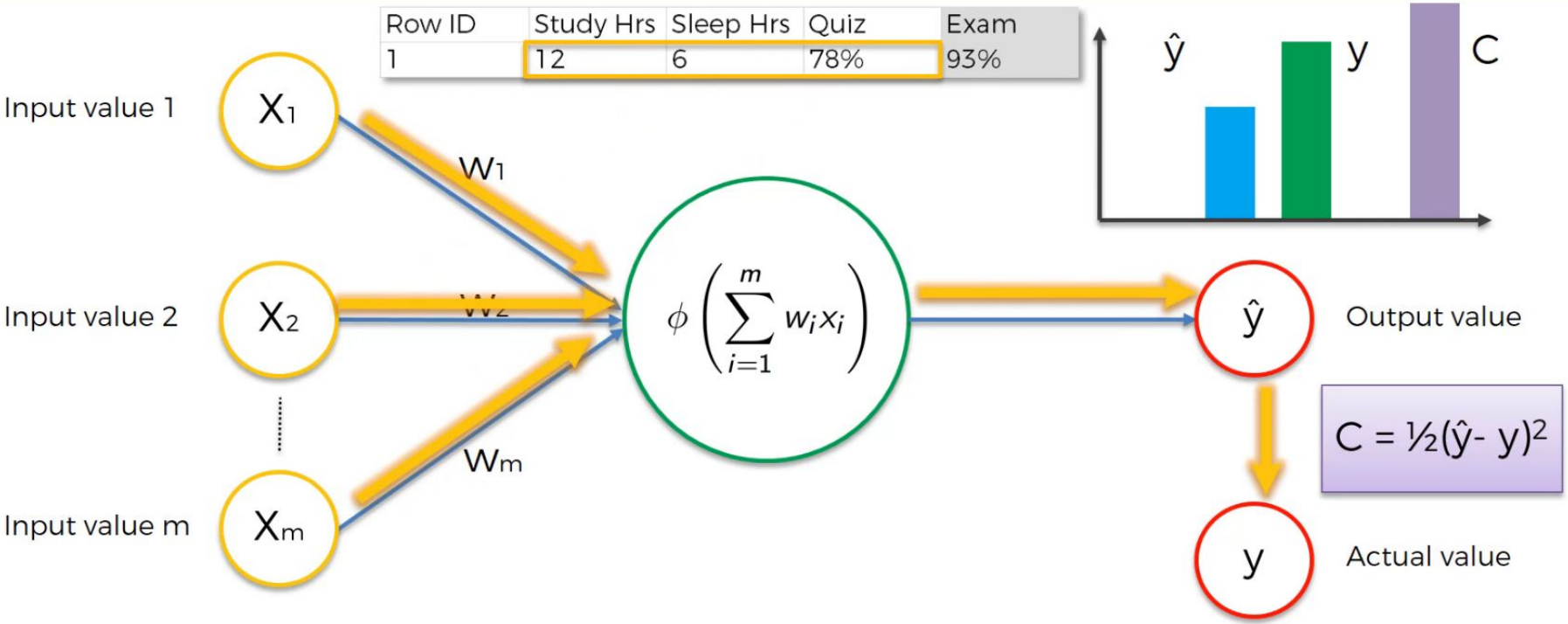
How do Neural Networks learn?



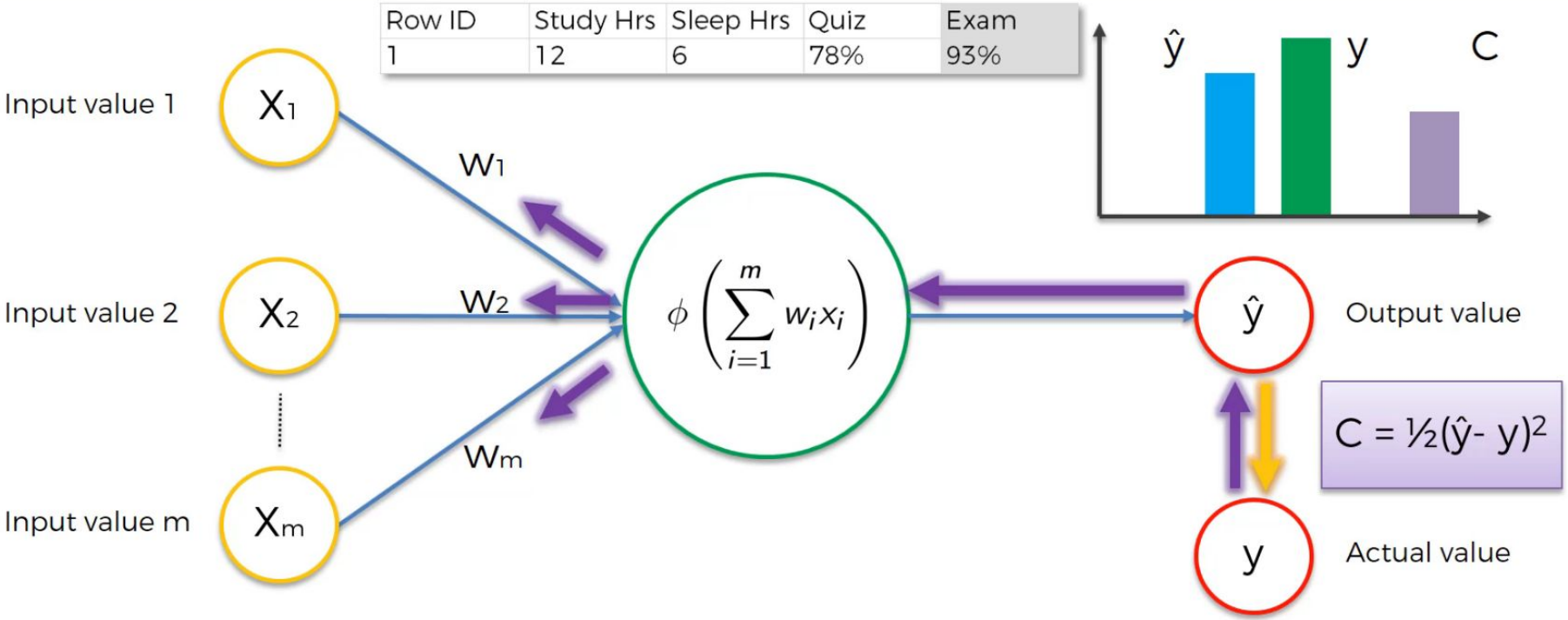
How do Neural Networks learn?



How do Neural Networks learn?



How do Neural Networks learn?



How do Neural Networks learn?