PadhAI: Deep Neural Networks

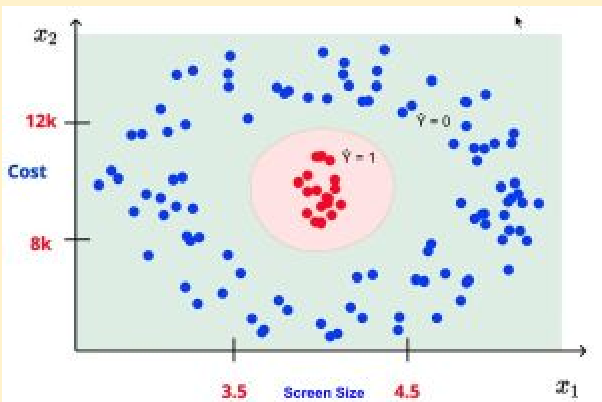
One Fourth Labs

Model

A Simple Deep Neural Network

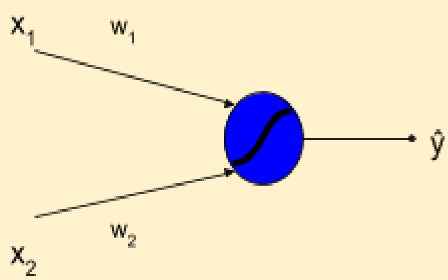
How to build complex functions using Deep Neural Networks

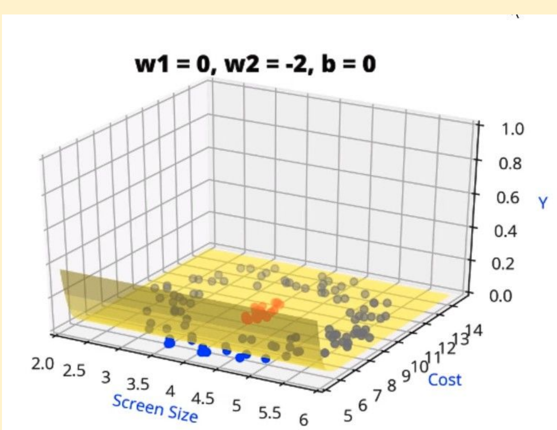
1. Consider the previously used example of mobile phone like/dislike predictor with the variables Screen-size and Cost. It has a complex decision boundary as shown here

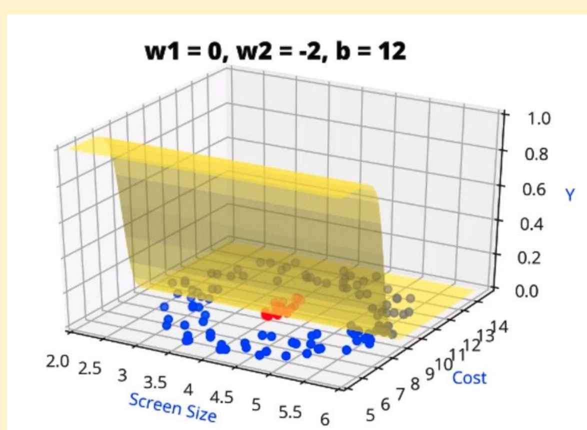


2. With a single sigmoid neuron, it is impossible to obtain this shape, regardless of how we vary the parameters w & b, as the sigmoid neuron can only produce a shape ranging from s-shaped to flat. The formula is $\hat{y} = f(x_1, x_2)$ or $\hat{y} = \frac{1}{1 + e^{-(w_1 * x_1 + w_2 * x_2 + b)}}$

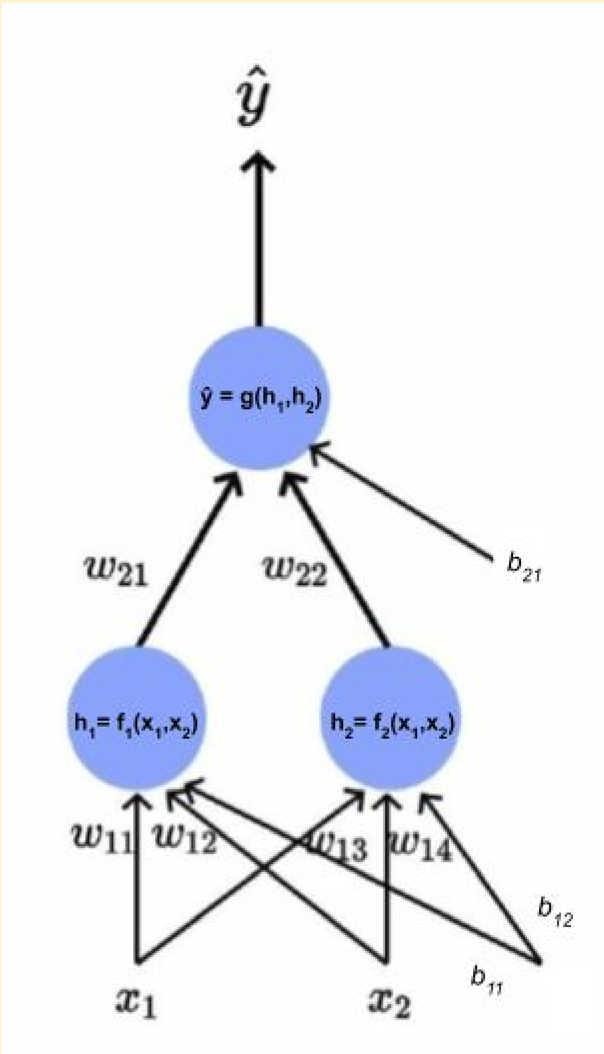
Sigmoid decision boundary, can range from s-shaped to flat, based on w and b values







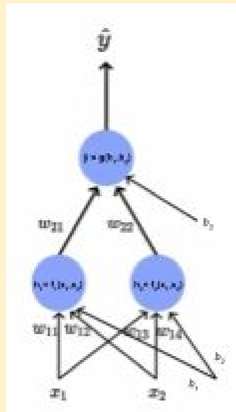
3. Now, let us consider a Deep Neural Network for the same mobile phone like/dislike predictor

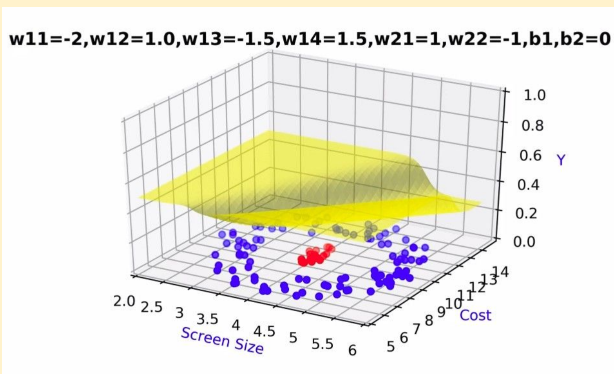


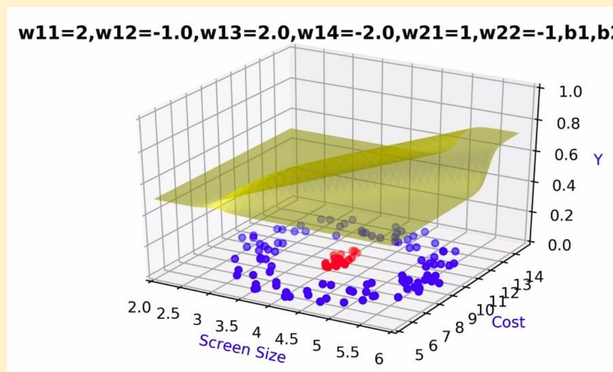
4. Breaking down the model:
   a. x_1 = Screen-Size, x_2 = Cost
   b. First Neuron $h_1 = f_1(x_1,x_2)$ or $h_1 = \frac{1}{1+e^{-(w_{11}*x_1 + w_{12}*x_2 + b_1)}}$
      i. Here, w_{11} and w_{12} are the weights of x_1 and x_2 corresponding to the first neuron h_1
      ii. b_{11} is the corresponding bias
   c. Second Neuron $h_2 = f_2(x_1,x_2)$ or $h_1 = \frac{1}{1+e^{-(w_{13}*x_1 + w_{14}*x_2 + b_2)}}$
      i. Here, w_{13} and w_{14} are the weights of x_1 and x_2 corresponding to the second neuron h_2
      ii. b_{12} is the corresponding bias
   d. Output Neuron $\hat{y} = g(h_1,h_2)$ or $\hat{y} = \frac{1}{1+e^{-(w_{21}*(\frac{1}{1+e^{-(w_{11}*x_1 + w_{12}*x_2 + b)}}) + w_{22}*(\frac{1}{1+e^{-(w_{13}*x_1 + w_{14}*x_2 + b)}}) + b_3)}}$
      i. Here, w_{21} and w_{22} are the weights of h_1 and h_2 corresponding to the output neuron $\hat{y}$
      ii. b_{21} is the corresponding bias
   e. From this configuration, we have 9 parameters (w_{11}, w_{12}, w_{13}, w_{14}, w_{21}, w_{22}, b_1, b_2, b_3), which allow for a much more complex decision boundary than a single sigmoid neuron with 3 parameters

5. The output would look something like this

   Deep Neural Network Decision Boundary, more complex than a single sigmoid neuron.

6. 





   * This simple neural network already allows for a much better decision boundary than with a single sigmoid neuron

7. The next step would be figuring out how to choose the best configuration of the DNN for our task, this is called **Hyperparameter Tuning.**
8. For now, we can rest easy knowing that by the **Universal Approximation Theorem** we will be able to approximate any kind of function with our Neural Network