

Module 3.1: Sigmoid Neuron

The story ahead ...

- Enough about boolean functions!

The story ahead ...

- Enough about boolean functions!
- What about arbitrary functions of the form $y = f(x)$ where $x \in \mathbb{R}^n$ (instead of $\{0,1\}^n$) and $y \in \mathbb{R}$ (instead of $\{0,1\}$) ?

The story ahead ...

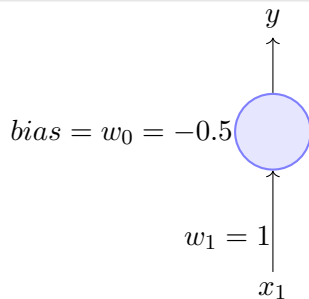
- Enough about boolean functions!
- What about arbitrary functions of the form $y = f(x)$ where $x \in \mathbb{R}^n$ (instead of $\{0,1\}^n$) and $y \in \mathbb{R}$ (instead of $\{0,1\}$) ?
- Can we have a network which can (approximately) represent such functions ?

The story ahead ...

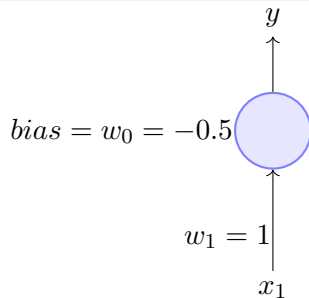
- Enough about boolean functions!
- What about arbitrary functions of the form $y = f(x)$ where $x \in \mathbb{R}^n$ (instead of $\{0, 1\}^n$) and $y \in \mathbb{R}$ (instead of $\{0, 1\}$) ?
- Can we have a network which can (approximately) represent such functions ?
- Before answering the above question we will have to first graduate from *perceptrons* to *sigmoidal neurons* ...

Recall

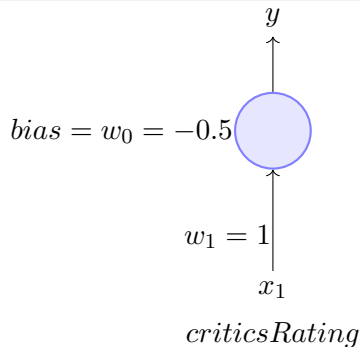
- A perceptron will fire if the weighted sum of its inputs is greater than the threshold ($-w_0$)



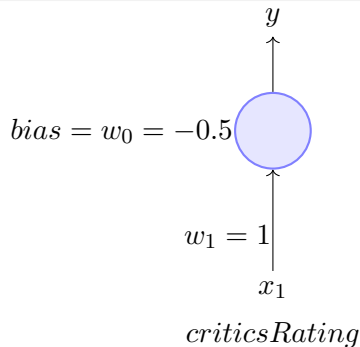
- The thresholding logic used by a perceptron is very harsh !



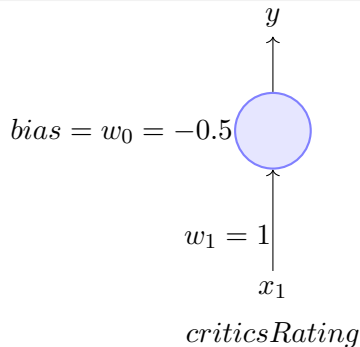
- The thresholding logic used by a perceptron is very harsh !
- For example, let us return to our problem of deciding whether we will like or dislike a movie



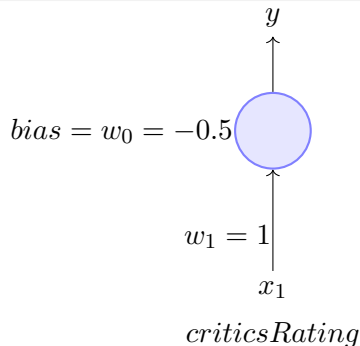
- The thresholding logic used by a perceptron is very harsh !
- For example, let us return to our problem of deciding whether we will like or dislike a movie
- Consider that we base our decision only on one input ($x_1 = \textit{criticsRating}$ which lies between 0 and 1)



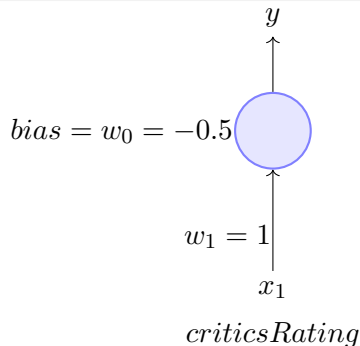
- The thresholding logic used by a perceptron is very harsh !
- For example, let us return to our problem of deciding whether we will like or dislike a movie
- Consider that we base our decision only on one input ($x_1 = criticsRating$ which lies between 0 and 1)
- If the threshold is 0.5 ($w_0 = -0.5$) and $w_1 = 1$ then what would be the decision for a movie with $criticsRating = 0.51$?



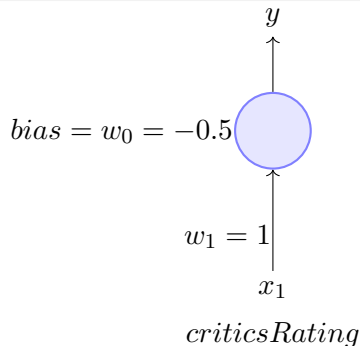
- The thresholding logic used by a perceptron is very harsh !
- For example, let us return to our problem of deciding whether we will like or dislike a movie
- Consider that we base our decision only on one input ($x_1 = criticsRating$ which lies between 0 and 1)
- If the threshold is 0.5 ($w_0 = -0.5$) and $w_1 = 1$ then what would be the decision for a movie with $criticsRating = 0.51$? (like)



- The thresholding logic used by a perceptron is very harsh !
- For example, let us return to our problem of deciding whether we will like or dislike a movie
- Consider that we base our decision only on one input ($x_1 = \textit{criticsRating}$ which lies between 0 and 1)
- If the threshold is 0.5 ($w_0 = -0.5$) and $w_1 = 1$ then what would be the decision for a movie with $\textit{criticsRating} = 0.51$? (like)
- What about a movie with $\textit{criticsRating} = 0.49$?

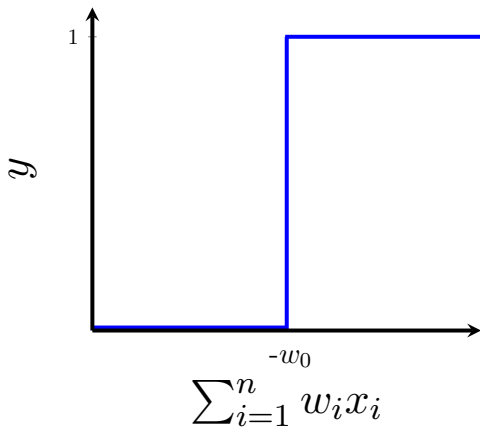


- The thresholding logic used by a perceptron is very harsh !
- For example, let us return to our problem of deciding whether we will like or dislike a movie
- Consider that we base our decision only on one input ($x_1 = criticsRating$ which lies between 0 and 1)
- If the threshold is 0.5 ($w_0 = -0.5$) and $w_1 = 1$ then what would be the decision for a movie with $criticsRating = 0.51$? (like)
- What about a movie with $criticsRating = 0.49$? (dislike)

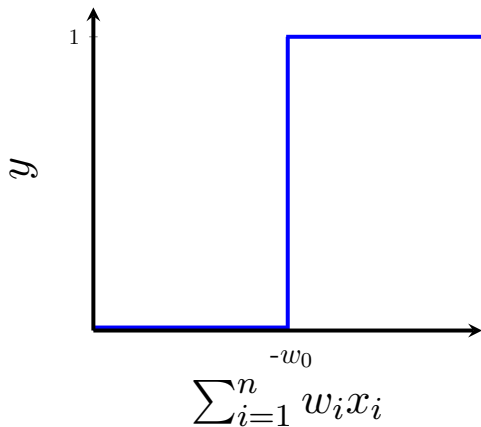


- The thresholding logic used by a perceptron is very harsh !
- For example, let us return to our problem of deciding whether we will like or dislike a movie
- Consider that we base our decision only on one input ($x_1 = \textit{criticsRating}$ which lies between 0 and 1)
- If the threshold is 0.5 ($w_0 = -0.5$) and $w_1 = 1$ then what would be the decision for a movie with $\textit{criticsRating} = 0.51$? (like)
- What about a movie with $\textit{criticsRating} = 0.49$? (dislike)
- It seems harsh that we would like a movie with rating 0.51 but not one with a rating of 0.49

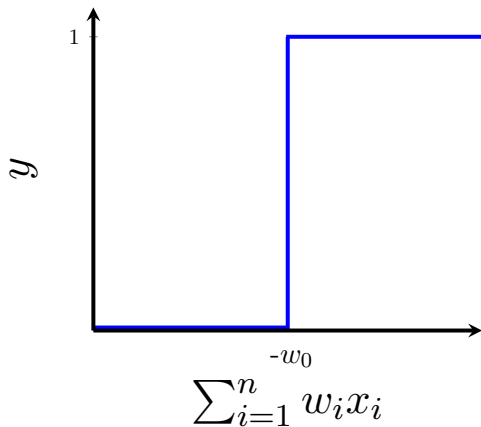
- This behavior is not a characteristic of the specific problem we chose or the specific weight and threshold that we chose



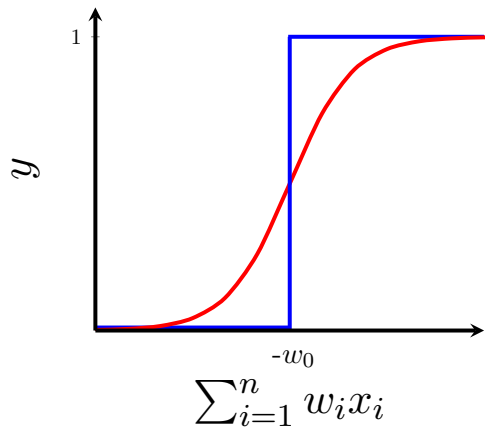
- This behavior is not a characteristic of the specific problem we chose or the specific weight and threshold that we chose
- It is a characteristic of the perceptron function itself which behaves like a step function



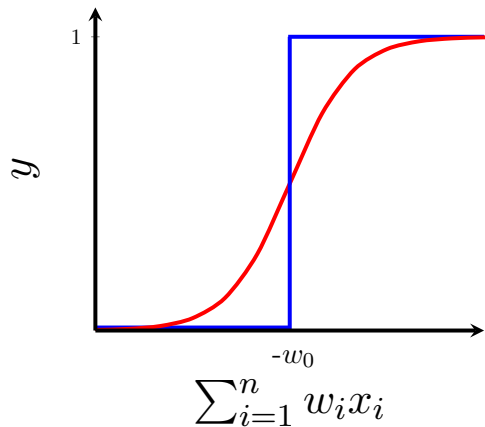
- This behavior is not a characteristic of the specific problem we chose or the specific weight and threshold that we chose
- It is a characteristic of the perceptron function itself which behaves like a step function
- There will always be this sudden change in the decision (from 0 to 1) when $\sum_{i=1}^n w_i x_i$ crosses the threshold ($-w_0$)



- This behavior is not a characteristic of the specific problem we chose or the specific weight and threshold that we chose
- It is a characteristic of the perceptron function itself which behaves like a step function
- There will always be this sudden change in the decision (from 0 to 1) when $\sum_{i=1}^n w_i x_i$ crosses the threshold ($-w_0$)
- For most real world applications we would expect a smoother decision function which gradually changes from 0 to 1

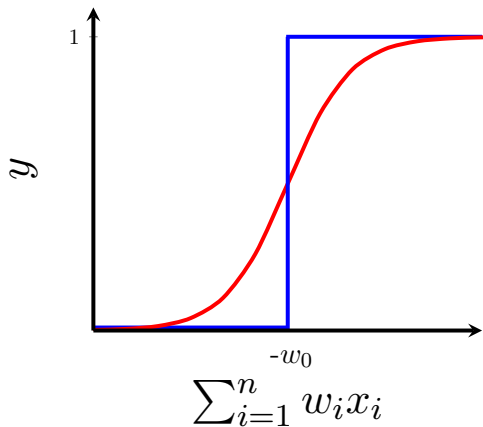


- Introducing sigmoid neurons where the output function is much smoother than the step function



- Introducing sigmoid neurons where the output function is much smoother than the step function
- Here is one form of the sigmoid function called the logistic function

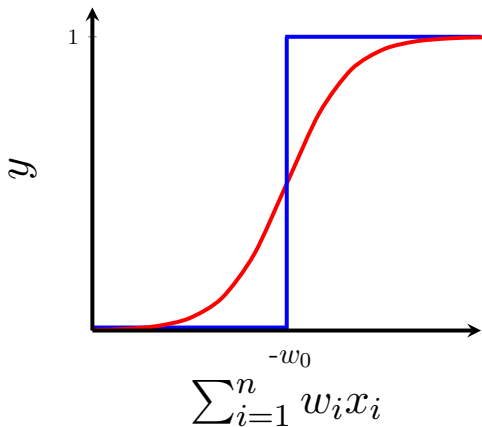
$$y = \frac{1}{1 + e^{-(w_0 + \sum_{i=1}^n w_i x_i)}}$$



- Introducing sigmoid neurons where the output function is much smoother than the step function
- Here is one form of the sigmoid function called the logistic function

$$y = \frac{1}{1 + e^{-(w_0 + \sum_{i=1}^n w_i x_i)}}$$

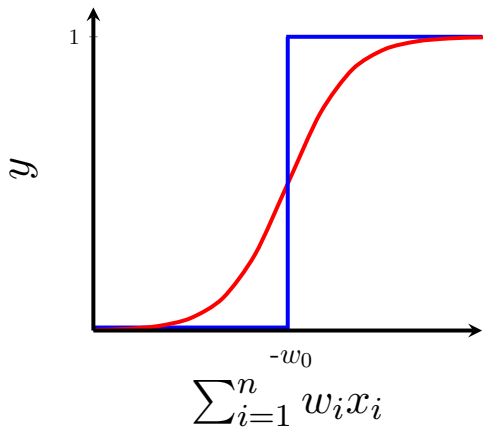
- We no longer see a sharp transition around the threshold $-w_0$



- Introducing sigmoid neurons where the output function is much smoother than the step function
- Here is one form of the sigmoid function called the logistic function

$$y = \frac{1}{1 + e^{-(w_0 + \sum_{i=1}^n w_i x_i)}}$$

- We no longer see a sharp transition around the threshold $-w_0$
- Also the output y is no longer binary but a real value between 0 and 1 which can be interpreted as a probability

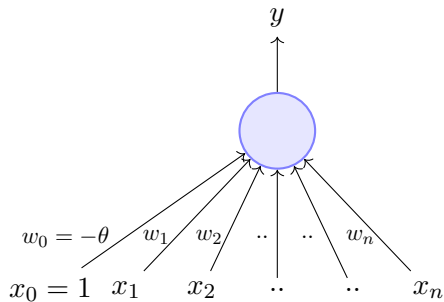


- Introducing sigmoid neurons where the output function is much smoother than the step function
- Here is one form of the sigmoid function called the logistic function

$$y = \frac{1}{1 + e^{-(w_0 + \sum_{i=1}^n w_i x_i)}}$$

- We no longer see a sharp transition around the threshold $-w_0$
- Also the output y is no longer binary but a real value between 0 and 1 which can be interpreted as a probability
- Instead of a like/dislike decision we get the probability of liking the movie

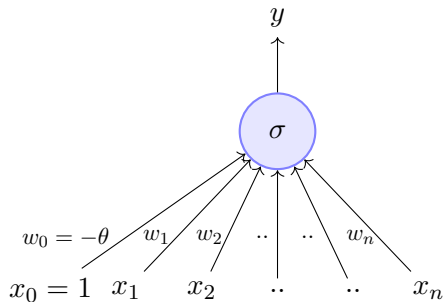
Perceptron



$$y = 1 \quad \text{if} \quad \sum_{i=0}^n w_i * x_i \geq 0$$

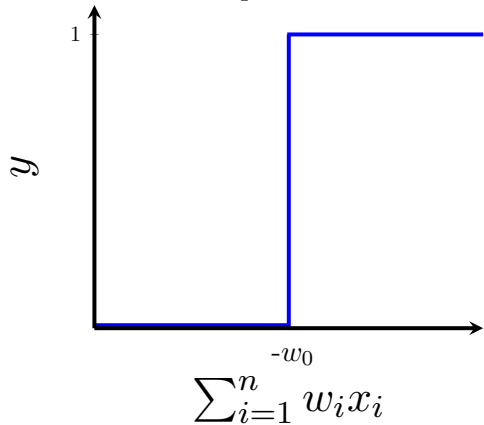
$$= 0 \quad \text{if} \quad \sum_{i=0}^n w_i * x_i < 0$$

Sigmoid (logistic) Neuron



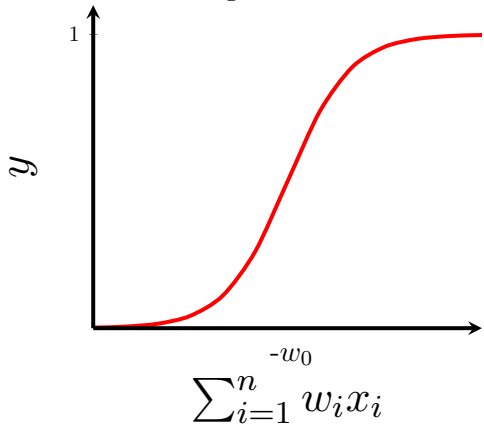
$$y = \frac{1}{1 + e^{-(\sum_{i=0}^n w_i x_i)}}$$

Perceptron



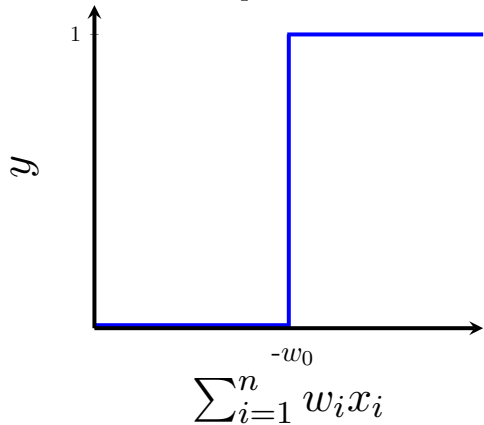
Not smooth, not continuous (at w_0), **not differentiable**

Sigmoid Neuron



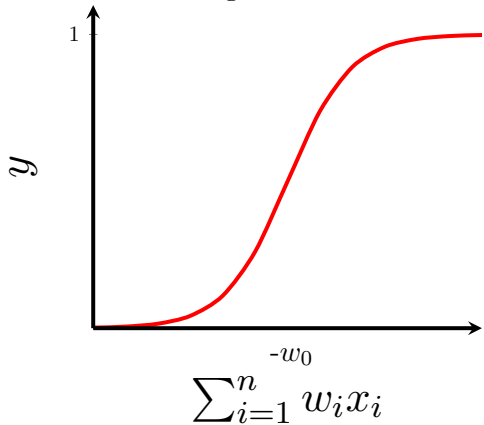
Smooth,
continuous, **differentiable**

Perceptron



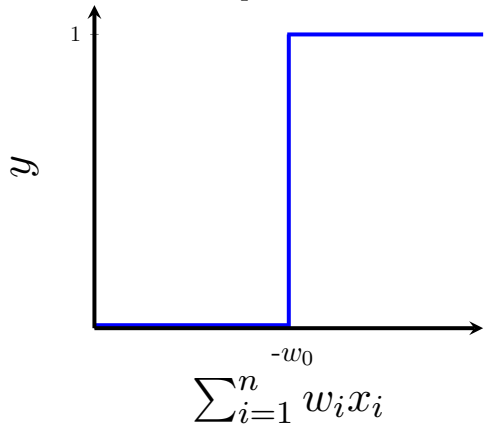
Not smooth, not continuous (at w_0), **not differentiable**

Sigmoid Neuron



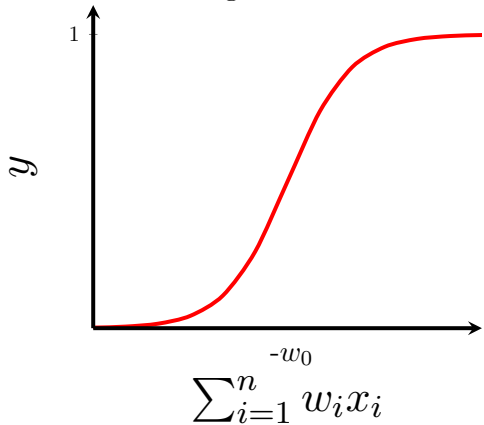
Smooth,
continuous, **differentiable**

Perceptron



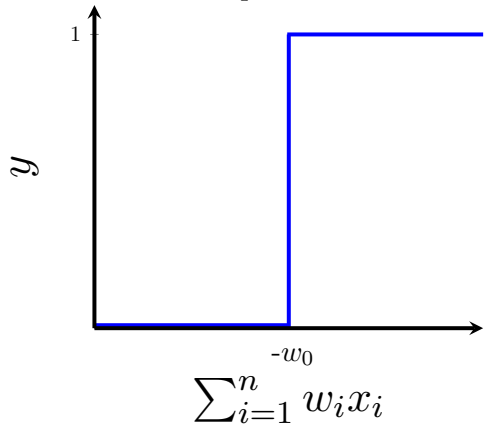
Not smooth, not continuous (at w_0), **not differentiable**

Sigmoid Neuron



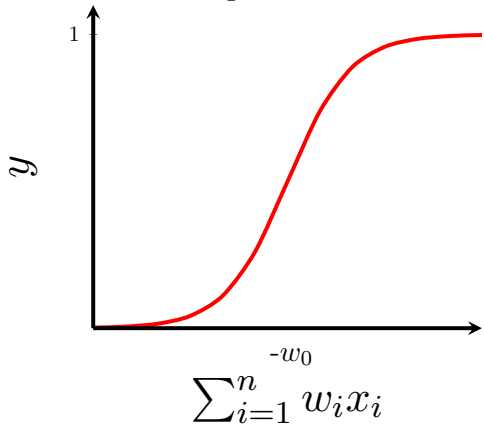
Smooth,
continuous, **differentiable**

Perceptron



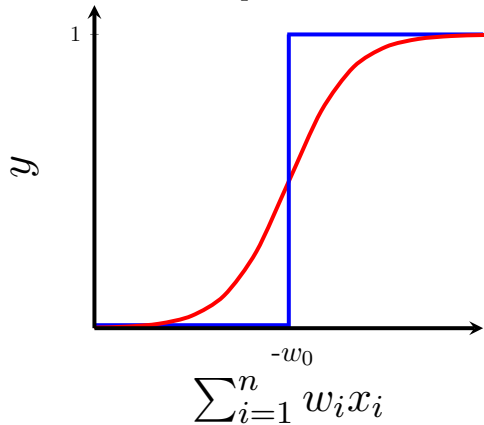
Not smooth, not continuous (at w_0), **not differentiable**

Sigmoid Neuron



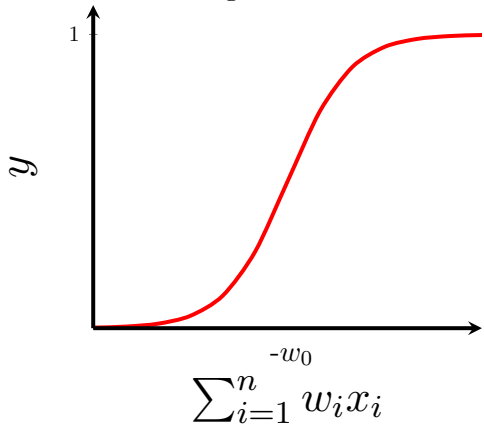
Smooth,
continuous, **differentiable**

Perceptron



Not smooth, not continuous (at w_0), **not differentiable**

Sigmoid Neuron



Smooth,
continuous, **differentiable**