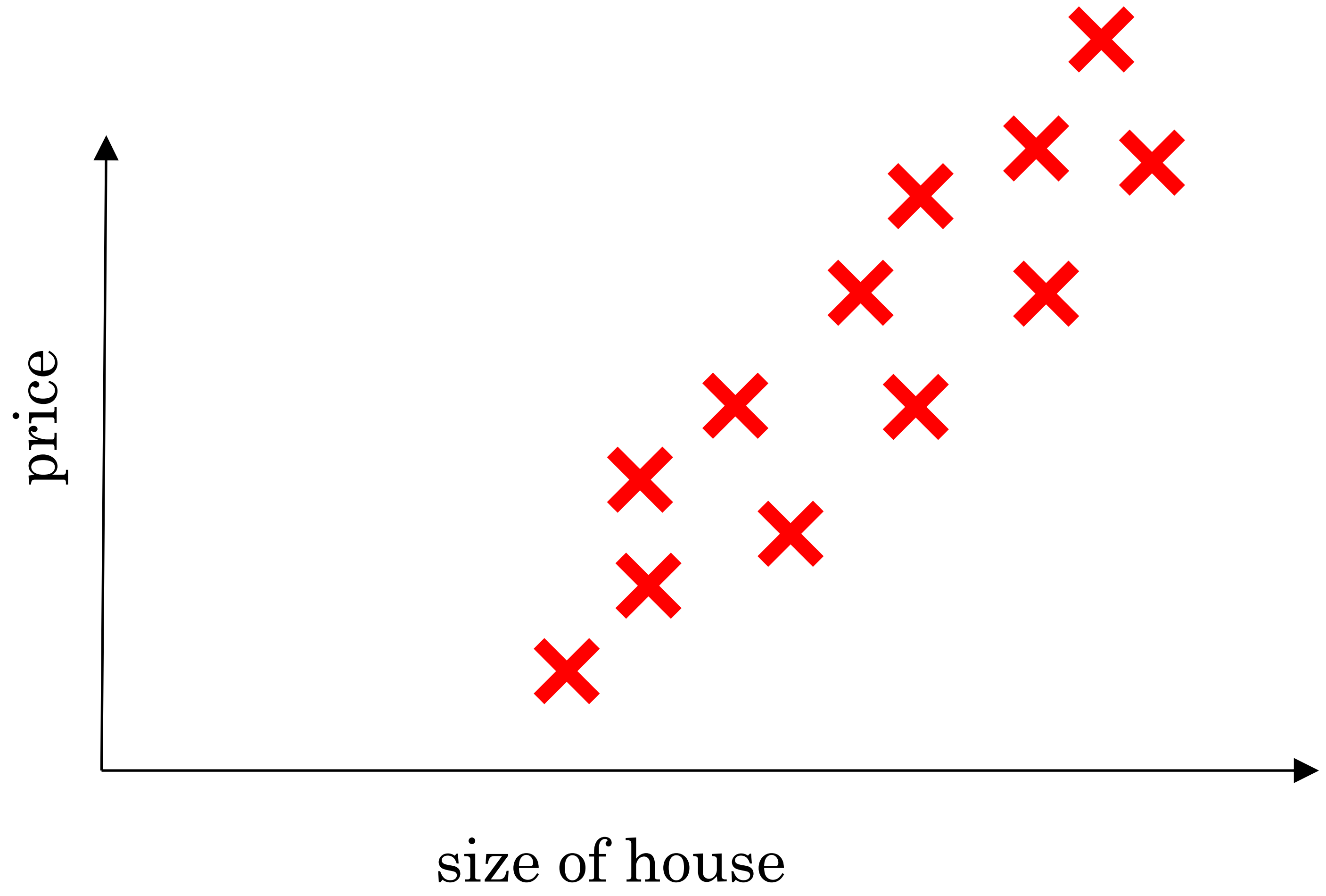
LOGISTIC REGRESSION AS A NEURAL NETWORK

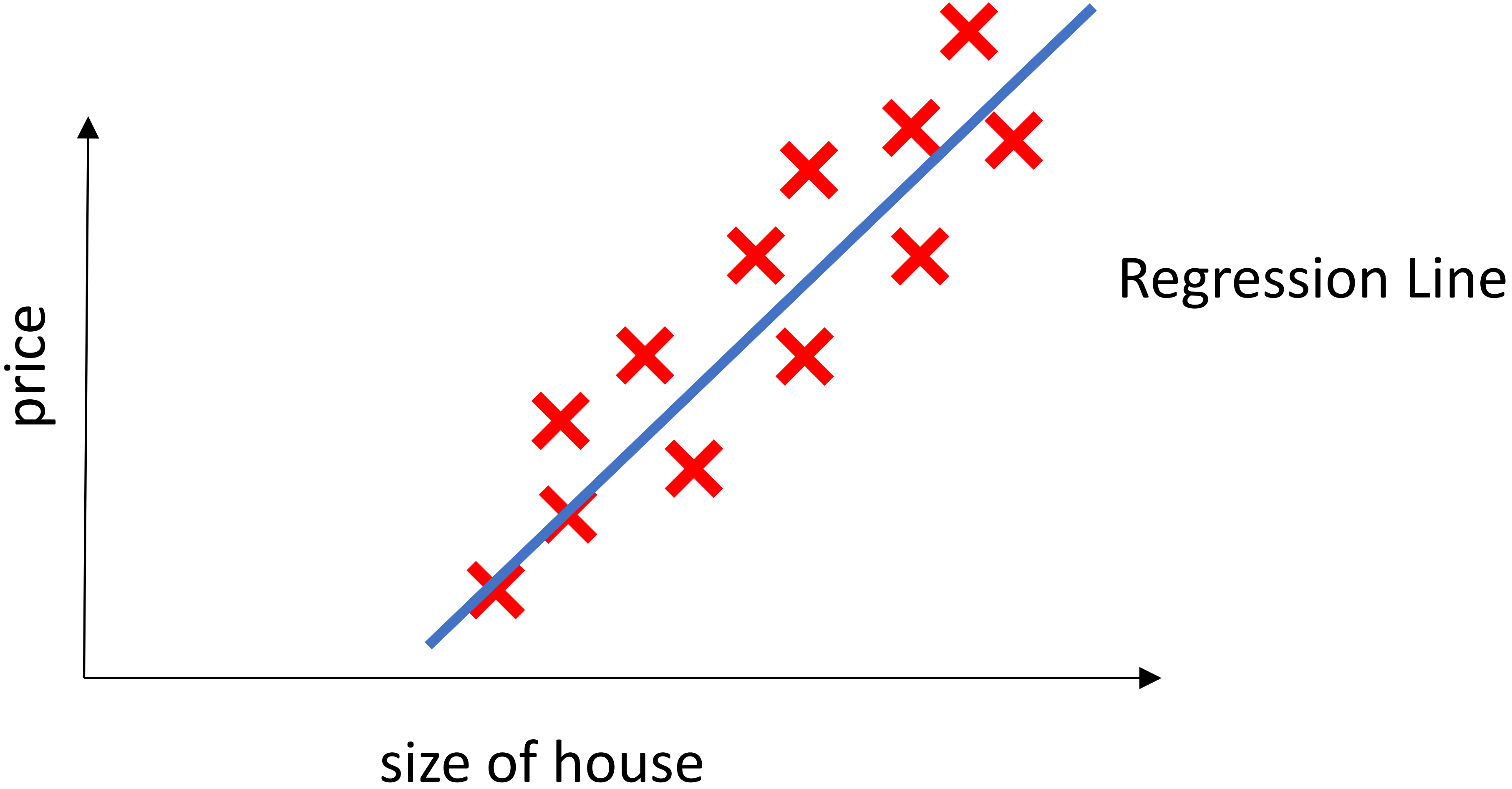
Filippo Cavallari

filippo.cavallari@southwales.ac.uk

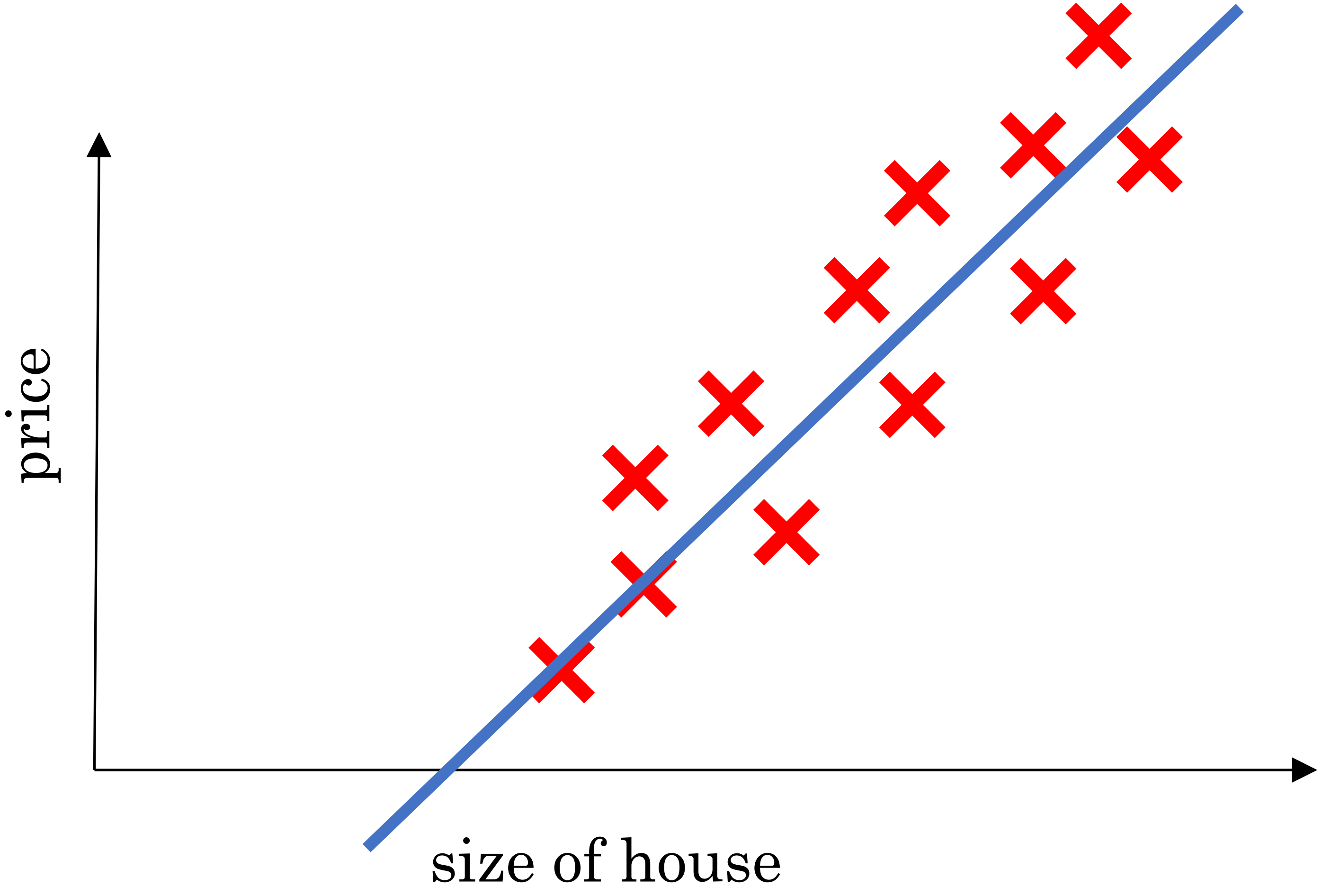
HOUSE PRICE PREDICTION



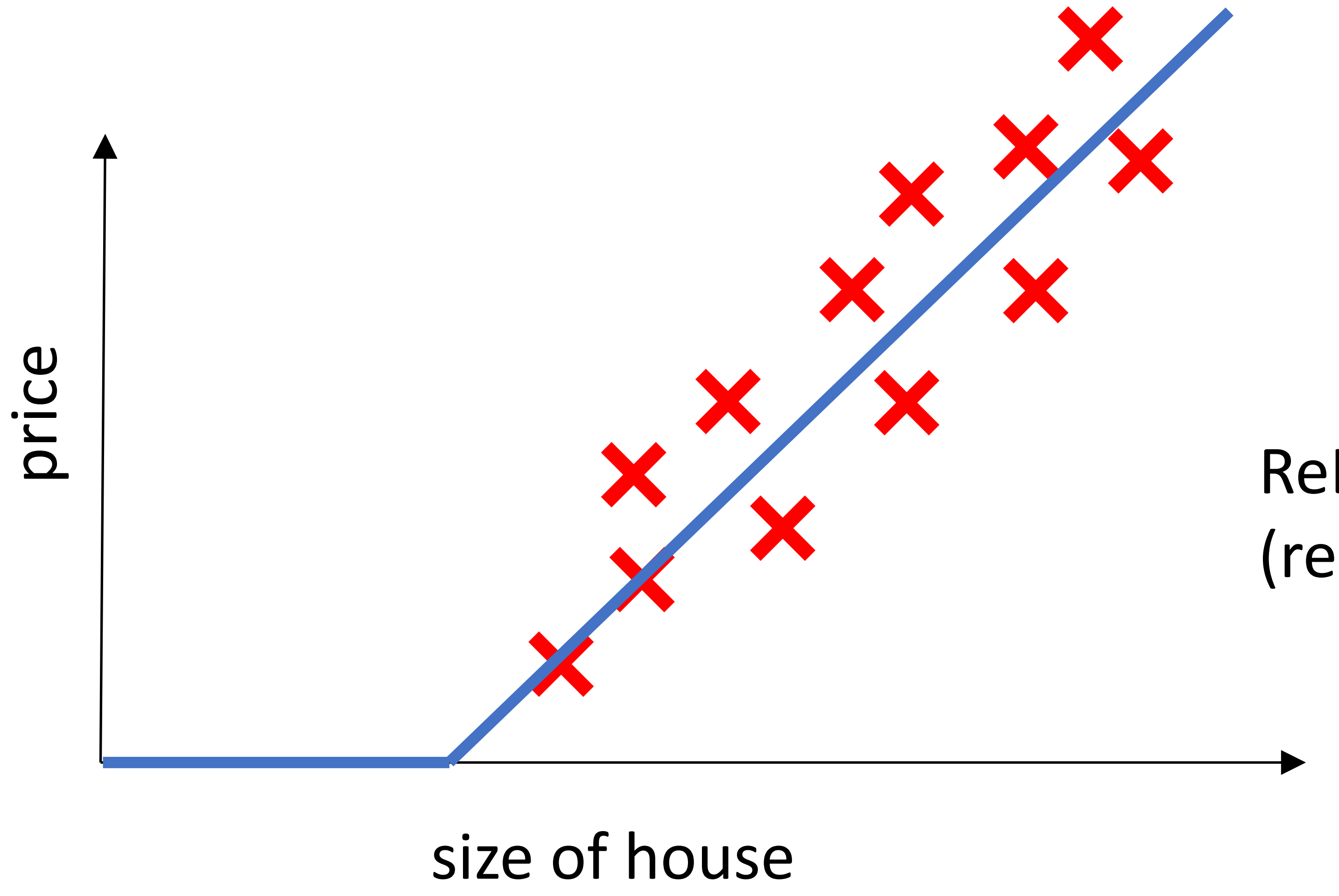
HOUSE PRICE PREDICTION



HOUSE PRICE PREDICTION

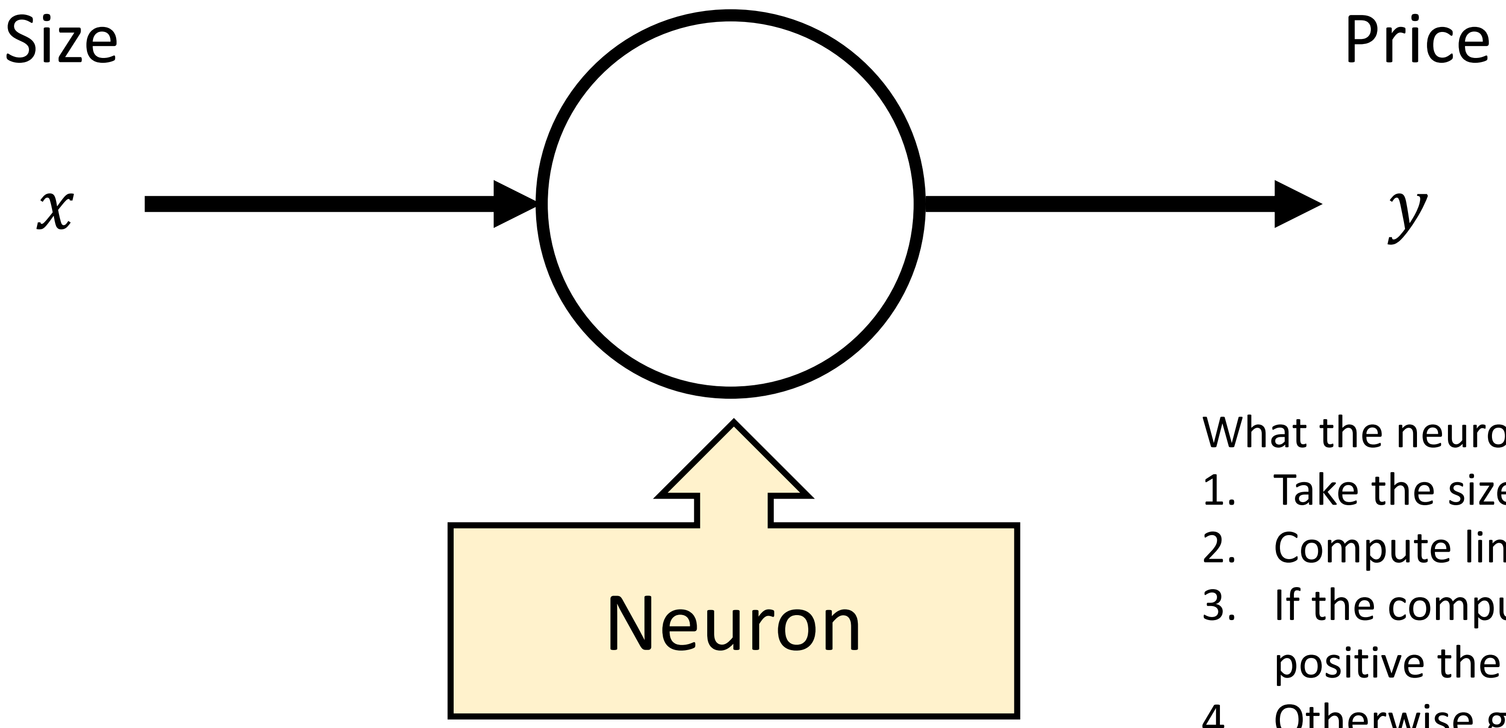


HOUSE PRICE PREDICTION



ReLU function
(rectified linear unit)

ONE SINGLE NEURON



What the neuron does?

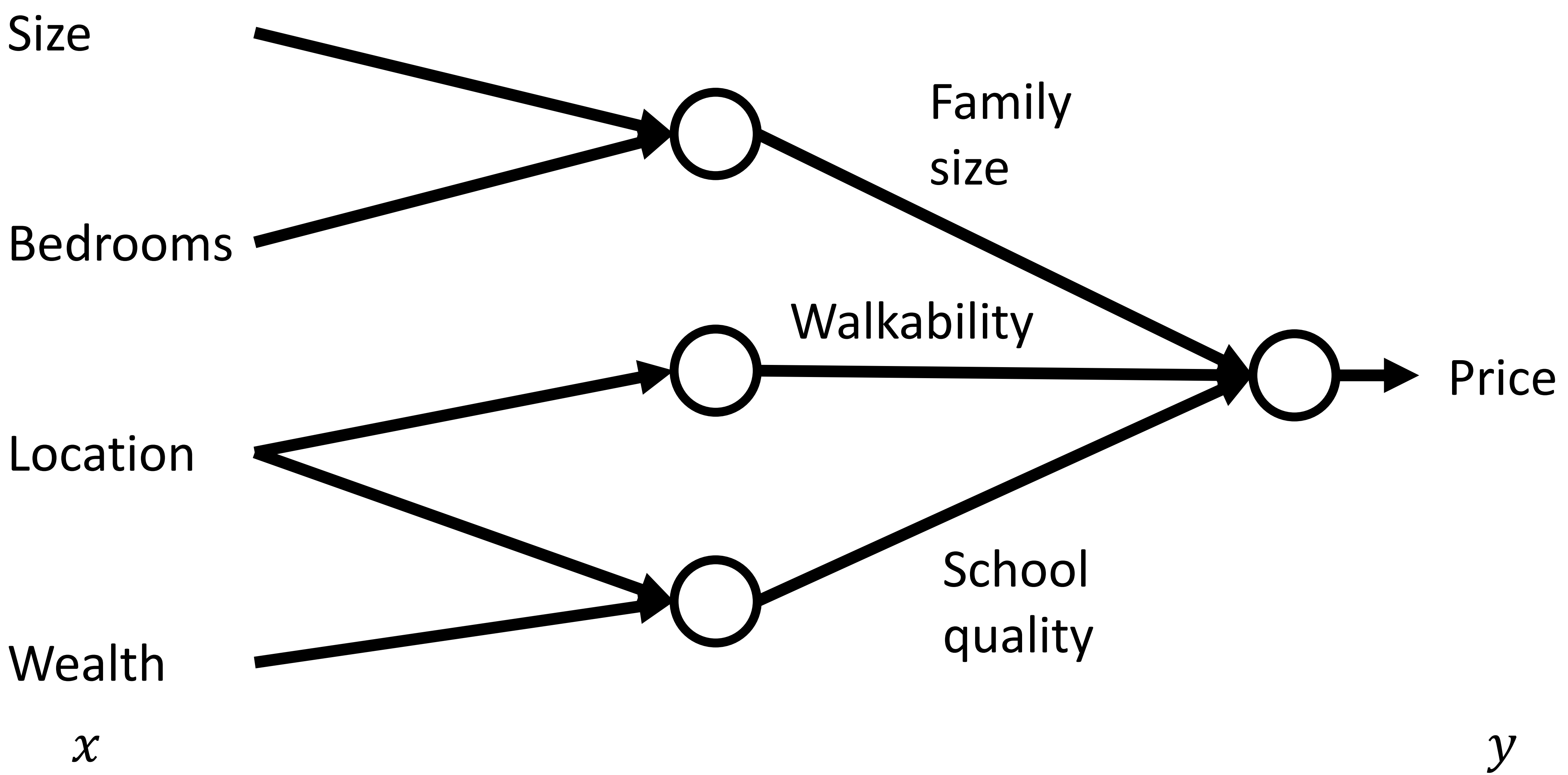
1. Take the size as input
2. Compute linear function
3. If the computed value is positive the gives it as output
4. Otherwise gives 0 (zero) as output

HOUSE PRICE PREDICTION

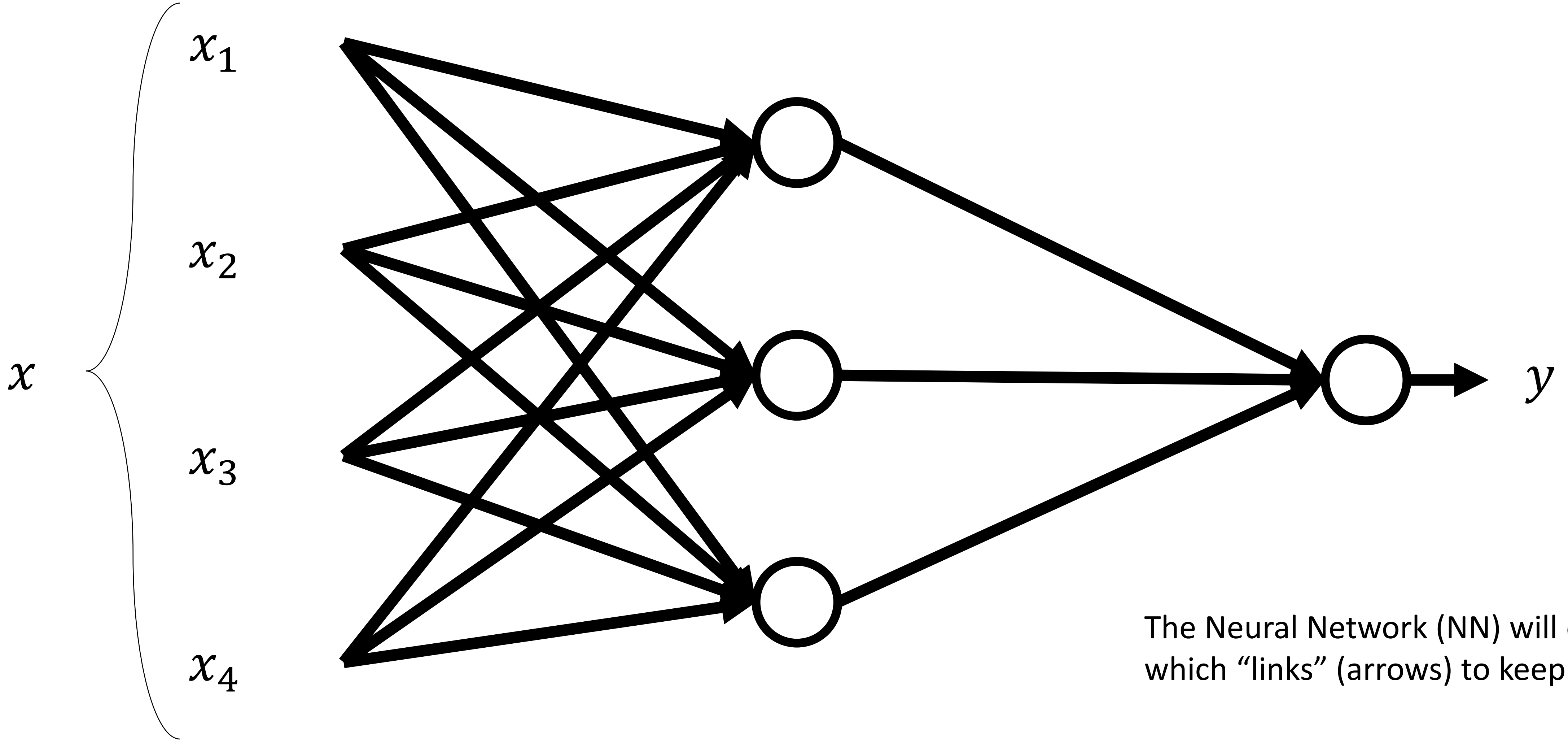
What if we have other features? For example:

- Size
- Bedrooms
- Location
- Wealth

HOUSE PRICE PREDICTION



IN GENERAL



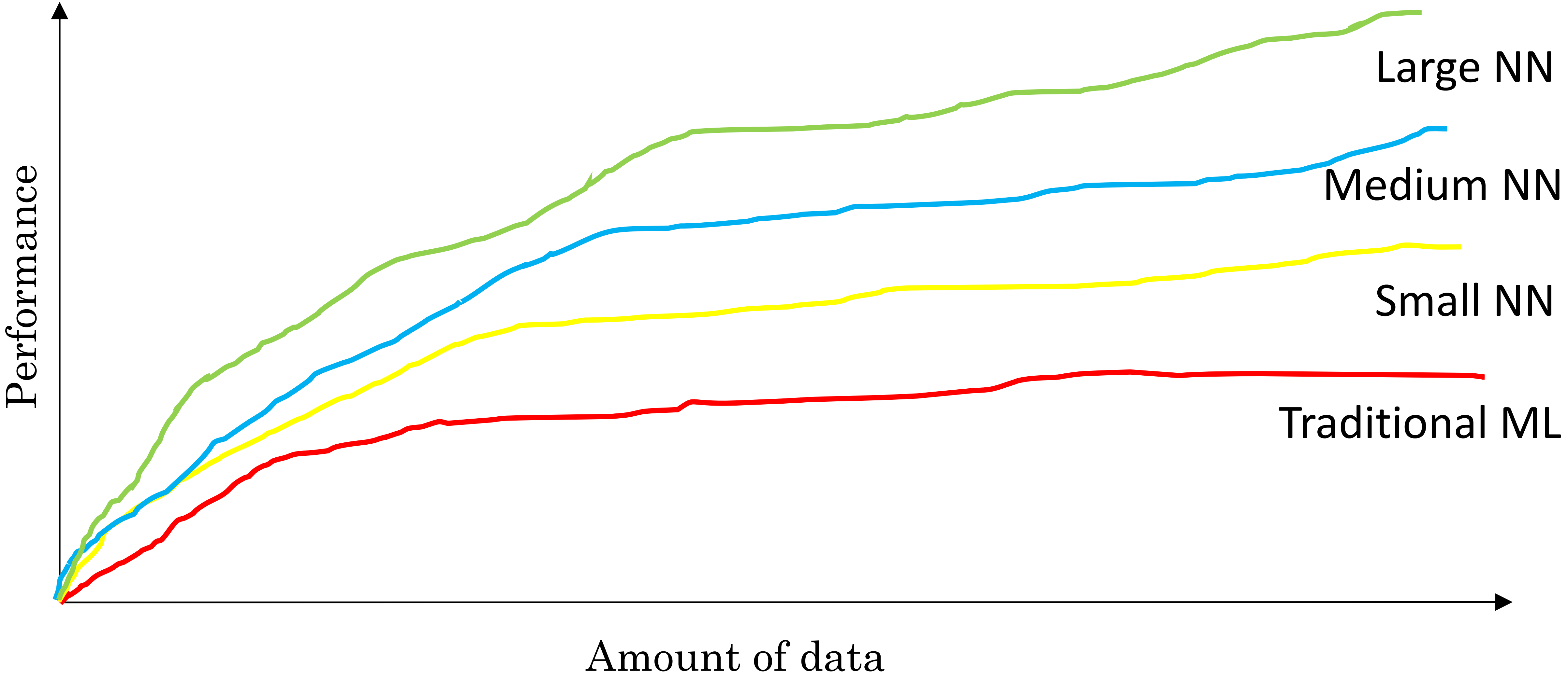
The Neural Network (NN) will decide which “links” (arrows) to keep

TYPES OF DATA

Structured data (such as tables of numbers)

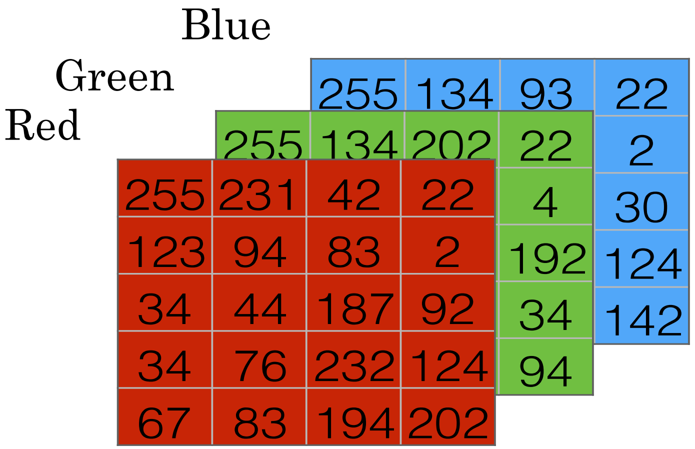
Unstructured data (such as images, audio, text)

SCALE DRIVES DEEP LEARNING PROGRESS



BINARY CLASSIFICATION

1 (cat) or 0 (non cat)



$$x = \begin{bmatrix} 255 \\ 231 \\ \cdots \\ 255 \\ 134 \\ \cdots \\ 255 \\ 134 \\ \cdots \end{bmatrix}$$

$$n = (5 \times 4) \times 3 = 60$$

NOTATION

A **single training example** is a pair (x, y) where $x \in \mathbb{R}^n$ and y $\in \{0,1\}$

The **training set** has m training examples

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(i)}, y^{(i)}), \dots, (x^{(m)}, y^{(m)})\}$$

NOTATION

We define the input matrix $X \in \mathbb{R}^{n \times m}$

$$X = \underbrace{\begin{bmatrix} | & | & \cdots & | & \cdots & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(i)} & \cdots & x^{(m)} \\ | & | & \cdots & | & \cdots & | \end{bmatrix}}_{m} \left.\vphantom{\begin{bmatrix} | \\ x \\ | \end{bmatrix}}\right\} n$$

NOTATION

We also define the output vector $Y \in \mathbb{R}^{1 \times m}$

$$\underbrace{Y = \begin{bmatrix} y^{(1)} & y^{(2)} & \dots & y^{(i)} & \dots & y^{(m)} \end{bmatrix}}_{m} \Big\} \, 1$$

NOTATION AND BASIC DERIVATIVES

Given $w, x \in \mathbb{R}^n$ and $b \in \mathbb{R}$, that is

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

we write

$$w^T x + b = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + b = \sum_{i=1}^{n} w_i x_i + b = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

Observe that

$$\frac{\partial}{\partial w_k}(w^T x + b) = x_k \quad \text{and} \quad \frac{\partial}{\partial b}(w^T x + b) = 1$$

LOGISTIC REGRESSION

What we want?

Given an input $x \in \mathbb{R}^n$ we want $\hat{y} = P(y = 1|x)$. Hence $0 \leq \hat{y} \leq 1$.

What are the **parameters** of our model?

$w \in \mathbb{R}^n$ and $b \in \mathbb{R}$

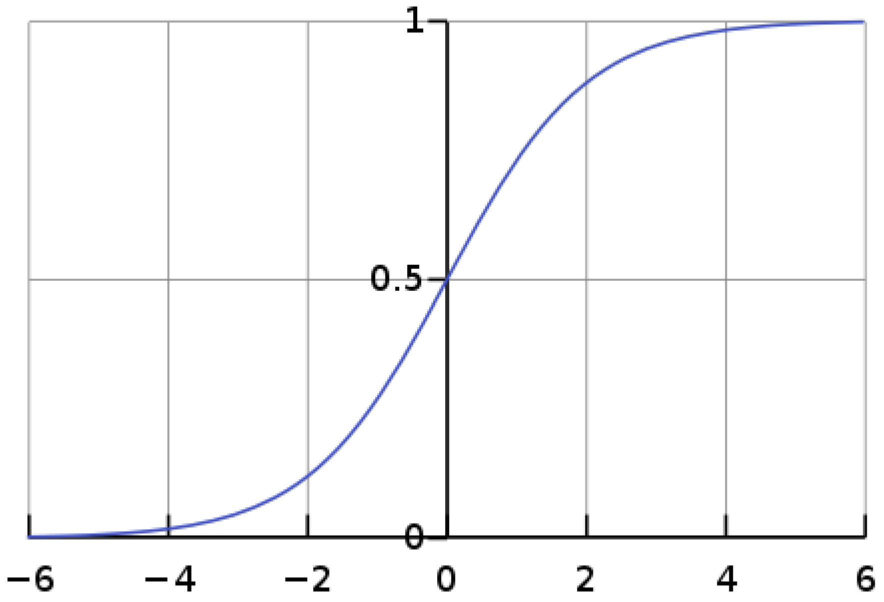
How we build the output of our model?

$\hat{y} = \sigma(w^T x + b)$ where $\sigma(z)$ is the sigmoid function

WHAT IS THE SIGMOID FUNCTION?

The sigmoid function we use for logistic regression is the logistic function

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



If z is large positive $\sigma(z) \approx 1$

If z is large negative $\sigma(z) \approx 0$

DERIVATIVE OF THE SIGMOID FUNCTION

$$\sigma'(z) = \frac{e^{-z}}{(1+e^{-z})^2}$$

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

$$= \frac{1-1-e^{-z}}{(1+e^{-z})^2} = \frac{1+e^{-z}}{(1+e^{-z})^2} - \frac{1}{(1+e^{-z})^2} = \sigma(z) - \sigma^2(z)$$

$$= \sigma(z)[1-\sigma(z)]$$

LOGISTIC REGRESSION

$\hat{y}^{(i)} = \sigma\left(w^T x^{(i)} + b\right)$ where $\sigma\left(z^{(i)}\right) = \frac{1}{1+e^{-z^{(i)}}}$ and $z^{(i)} = w^T x^{(i)} + b$

Given the training set

$$\left\{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \dots, \left(x^{(i)}, y^{(i)}\right), \dots, \left(x^{(m)}, y^{(m)}\right)\right\}$$

we want

$$\hat{y}^{(i)} = y^{(i)}$$

LOSS FUNCTION

How do we measure the error? A possible choice is to calculate, for each observation, the **loss function** (**cross entropy loss**)

$$\mathcal{L}(\hat{y}, y) = -\big(y \log \hat{y} + (1 - y)\log(1 - \hat{y})\big)$$

Why?

If $y = 1$ then $\mathcal{L}(\hat{y}, y) = -\log \hat{y}$ hence we want $\hat{y}$ large

If $y = 0$ then $\mathcal{L}(\hat{y}, y) = -\log(1 - \hat{y})$ hence we want $\hat{y}$ small

LOGISTIC REGRESSION – COST FUNCTION

To evaluate the overall error we calculate the **cost function**, the average of the loss functions evaluated on all observations

$$E(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\hat{y}^{(i)} + \left(1 - y^{(i)}\right)\log\left(1 - \hat{y}^{(i)}\right)\right]$$

RECAP

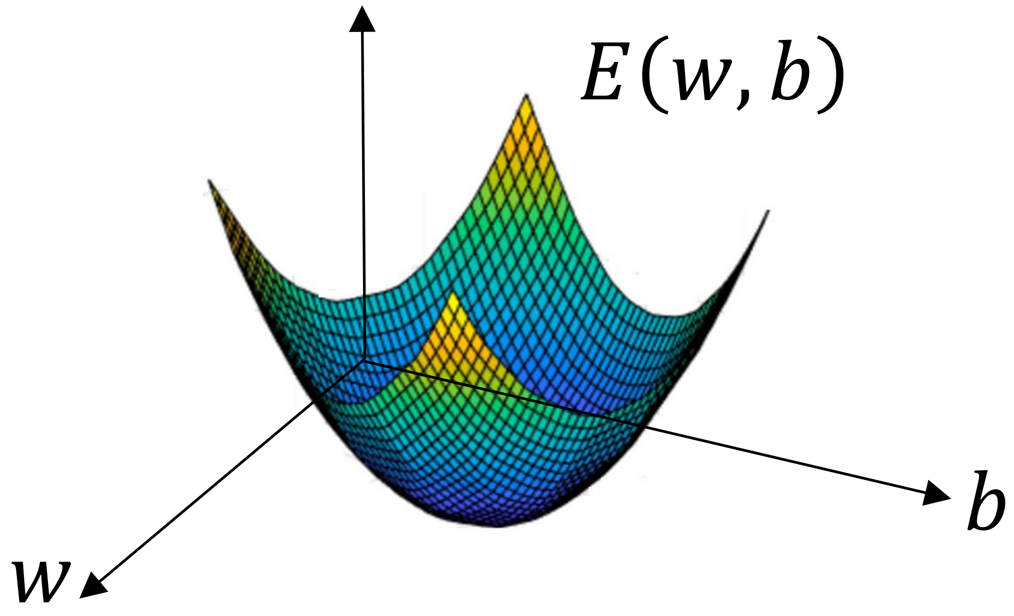
$\hat{y} = \sigma(w^T x + b)$ where $\sigma(z) = \frac{1}{1+e^{-z}}$

$$E(w, b) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

We want to find w, b that minimise $\mathrm{E}(w, b)$ the cost function

GRADIENT DESCENT

We want to find w, b that minimise $\mathrm{E}(w, b)$ the cost function



It turns out the $E(w, b)$ is a convex function, hence admits one global minimum

GRADIENT DESCENT – ONE VARIABLE

How minimise the cost function?

The algorithm:

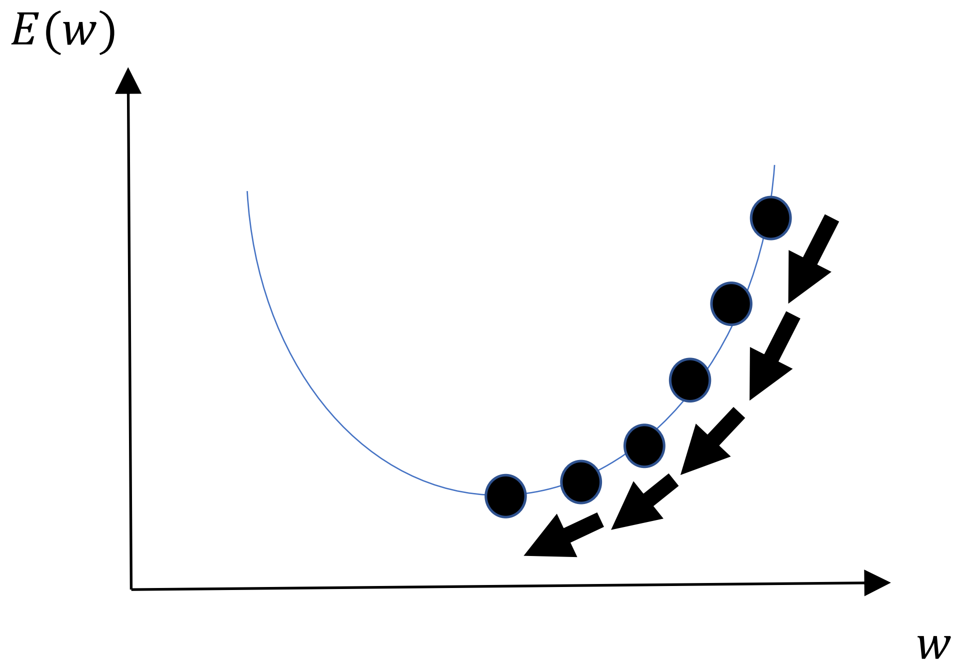
Repeat {

$$w = w - \alpha \frac{dE(w)}{dw}$$

}

Where α is the **learning rate** or **step size**

$\frac{dE(w)}{dw}$ is the slope of the tangent line



GRADIENT DESCENT – MORE VARIABLES

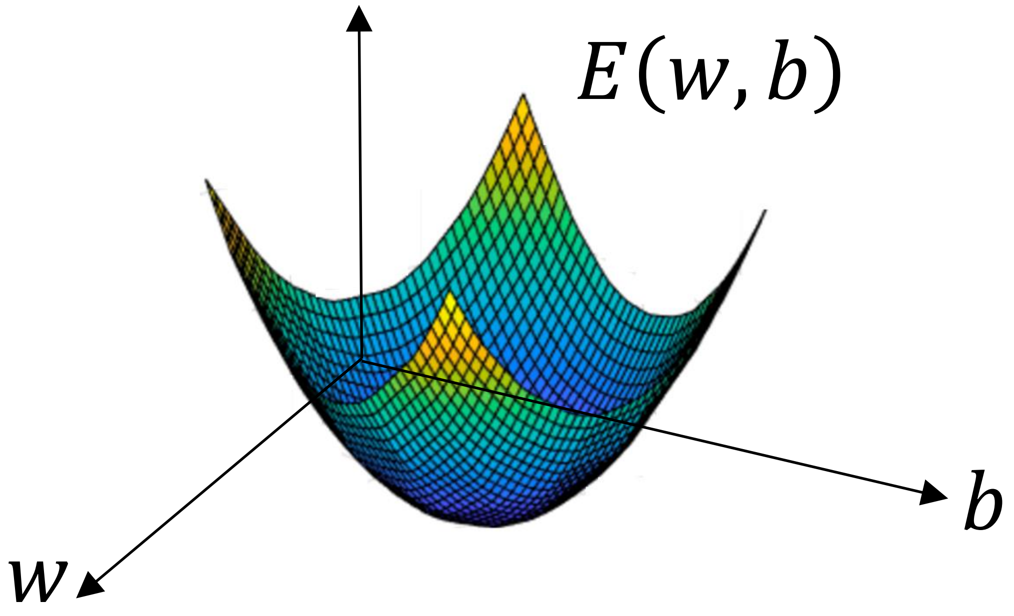
The algorithm:

Repeat {

$$w = w - \alpha \frac{\partial E(w,b)}{\partial w}$$

$$b = b - \alpha \frac{\partial E(w,b)}{\partial b}$$

}



GRADIENT DESCENT

We have (using the **chain rule**)

$$\frac{\boldsymbol{\partial}}{\boldsymbol{\partial w_k}}\boldsymbol{\mathcal{L}}(\hat{\boldsymbol{y}},\boldsymbol{y}) = \frac{\partial}{\partial w_k}\{-[y\log\hat{y} + (1-y)\log(1-\hat{y})]\} = -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\frac{\partial}{\partial w_k}\hat{y}$$

$$= -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\hat{y}(1-\hat{y})\frac{\partial}{\partial w_k}(w^T x + b) = -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\hat{y}(1-\hat{y})x_k$$

$$\frac{\boldsymbol{\partial}}{\boldsymbol{\partial b}}\boldsymbol{\mathcal{L}}(\hat{\boldsymbol{y}},\boldsymbol{y}) = \frac{\partial}{\partial b}\{-[y\log\hat{y} + (1-y)\log(1-\hat{y})]\} = -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\frac{\partial}{\partial b}\hat{y}$$

$$= -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\hat{y}(1-\hat{y})\frac{\partial}{\partial b}(w^T x + b) = -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\hat{y}(1-\hat{y})$$

GRADIENT DESCENT

Hence

$$\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{w_k}} = -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\hat{y}(1-\hat{y})x_k = -[y(1-\hat{y}) - (1-y)\hat{y}]x_k$$

$$= -(y - y\hat{y} - \hat{y} + y\hat{y})x_k = (\hat{y} - y)x_k$$

$$\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{b}} = -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\hat{y}(1-\hat{y}) = -[y(1-\hat{y}) - (1-y)\hat{y}]$$

$$= -(y - y\hat{y} - \hat{y} + y\hat{y}) = (\hat{y} - y)$$

GRADIENT DESCENT

In the case of the logistic regression we have

$$\frac{\partial \mathcal{L}}{\partial w_k} = (\hat{y} - y)x_k$$

$$\frac{\partial \mathcal{L}}{\partial b} = (\hat{y} - y)$$

GRADIENT DESCENT

Hence, if we call $\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) = \mathcal{L}^{(i)}$ then we can write

$$\frac{\partial \mathcal{L}^{(i)}}{\partial w_k} = \left(\hat{y}^{(i)} - y^{(i)}\right) x_k^{(i)}$$

$$\frac{\partial \mathcal{L}^{(i)}}{\partial b} = \hat{y}^{(i)} - y^{(i)}$$

GRADIENT DESCENT

Finally recalling that $E(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}^{(i)}$

and since the "derivative of the sum is the sum of the derivatives"

$$\frac{\partial E}{\partial w_k} = \frac{1}{m}\sum_{i=1}^{m} \frac{\partial \mathcal{L}^{(i)}}{\partial w_k} = \frac{1}{m}\sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right) x_k^{(i)}$$

$$\frac{\partial E}{\partial b} = \frac{1}{m}\sum_{i=1}^{m} \frac{\partial \mathcal{L}^{(i)}}{\partial b} = \frac{1}{m}\sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right)$$