PHY654

Machine learning (ML) in particle physics

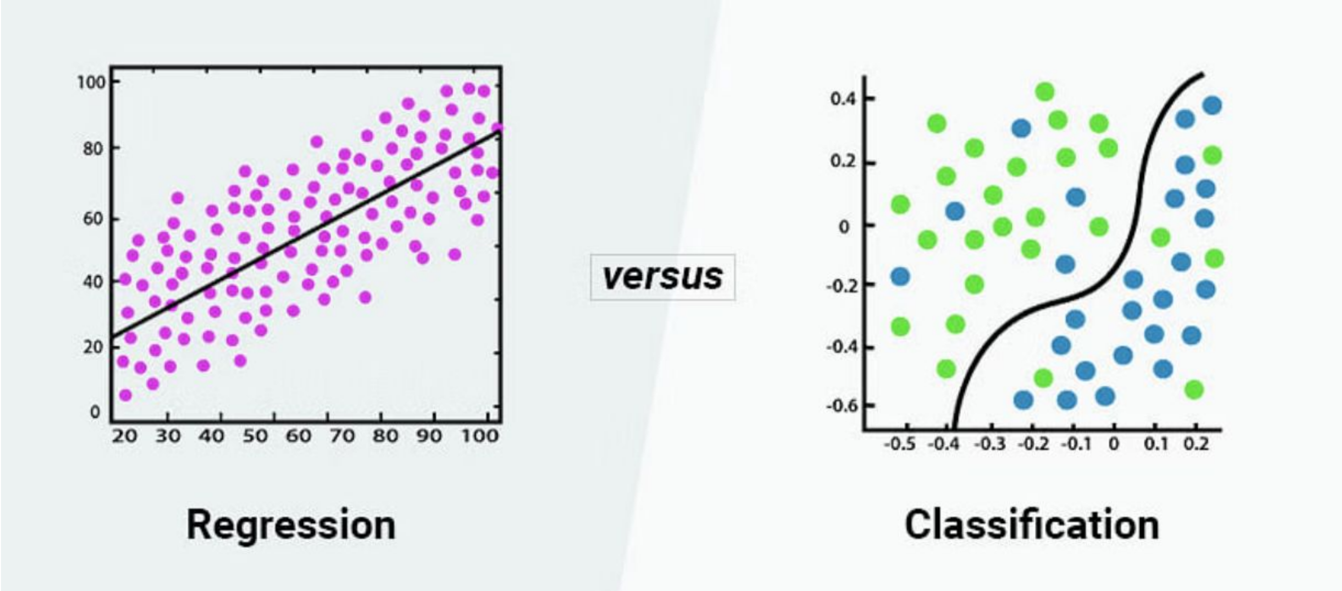
Swagata Mukherjee • IIT Kanpur
12th August 2024

Supervised vs unsupervised learning

- **Supervised** learning: Use labeled data for training.
- Task-driven
- Labeled data: examples with the correct answer (X_1, X_2 …. X_n and Y for every training example)
- The ML algorithms learns the relationship between inputs and outputs.
- The trained algorithm can then make predictions on new, unlabeled data.

- **Unsupervised** learning: The ML algorithm discover patterns and relationships in unlabeled data.
- Data-driven
- Example 1: Clustering algorithms → group similar data points based on inherent characteristics.
- Example 2 : Anomaly detection → find anomalies in data by looking for odd patterns.

Types of supervised learning



Classification example: Spam-detector in emails, Photon vs jet in a detector in HEP
Regression example: Housing price prediction, photon energy prediction

Logistic Regression

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

$$\begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \qquad x_0 = 1, y \in \{0, 1\}$$

Logistic Regression

It is an algorithm for classification problem.
In binary classification problem, the output y can take values 0 or 1

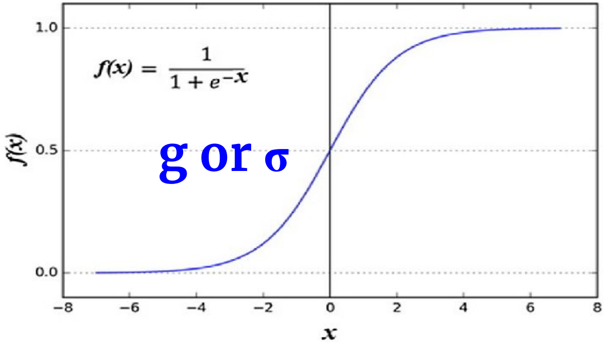
Logistic Regression: $0 \leq h_\theta(x) \leq 1$

For linear regression, the hypothesis was $h_\theta(x)=\theta^T X$

For logistic regression we will use $h_\theta(x)=g(\theta^T X)$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

Sigmoid Function



Where, ‘g’ is the Sigmoid Function (also called the Logistic Function)

$h_\theta(x)$ is the probability that the output is 1.

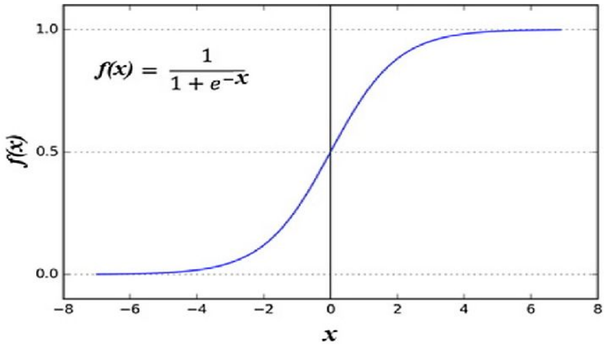
$$h_\theta(x) = P(y=1|x;\theta) = 1 - P(y=0|x;\theta)$$
$$P(y=0|x;\theta) + P(y=1|x;\theta) = 1$$

Logistic Regression

$$h_\theta(x) \geq 0.5 \rightarrow y = 1$$
$$h_\theta(x) < 0.5 \rightarrow y = 0$$

In classification problem, we want discrete output (0 or 1)

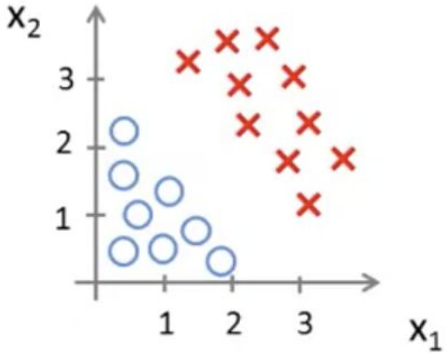
$$h_\theta(x) = g(\theta^T x) \geq 0.5$$
$$when\ \theta^T x \geq 0$$



$$\theta^T x \geq 0 \Rightarrow y = 1$$
$$\theta^T x < 0 \Rightarrow y = 0$$

$$z = 0, e^0 = 1 \Rightarrow g(z) = 1/2$$
$$z \rightarrow \infty, e^{-\infty} \rightarrow 0 \Rightarrow g(z) = 1$$
$$z \rightarrow -\infty, e^{\infty} \rightarrow \infty \Rightarrow g(z) = 0$$

Linear decision boundary

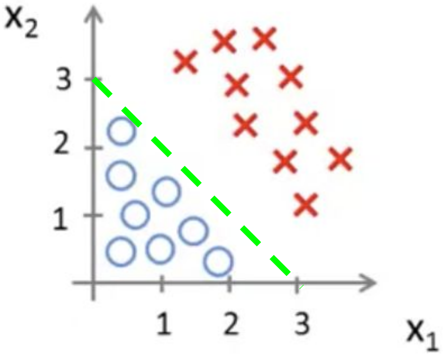


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

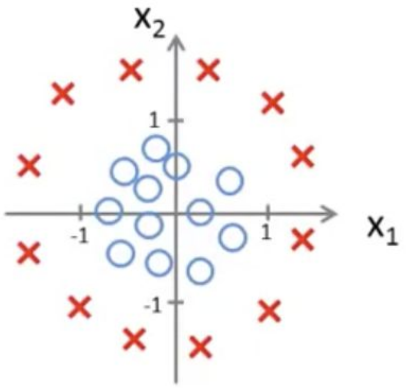
Let’s say, gradient descent finds parameters to be [-3, 1, 1]

Predict ‘y=1’ when:

$$-3 + x_1 + x_2 \geq 0$$



Non-linear decision boundary



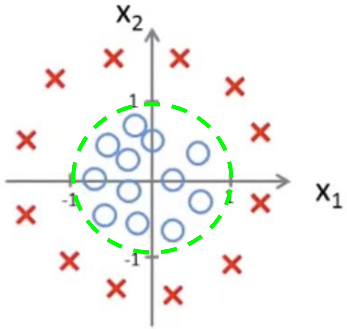
$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

make 2 new quadratic features →

Let's say, gradient descent finds parameters to be [-1, 0, 0, 1, 1]

Predict 'y=1' when:

$$-1 + x_1^2 + x_2^2 \geq 0$$



Cost function for logistic regression

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

Also called "loss"

Linear regression cost function was this

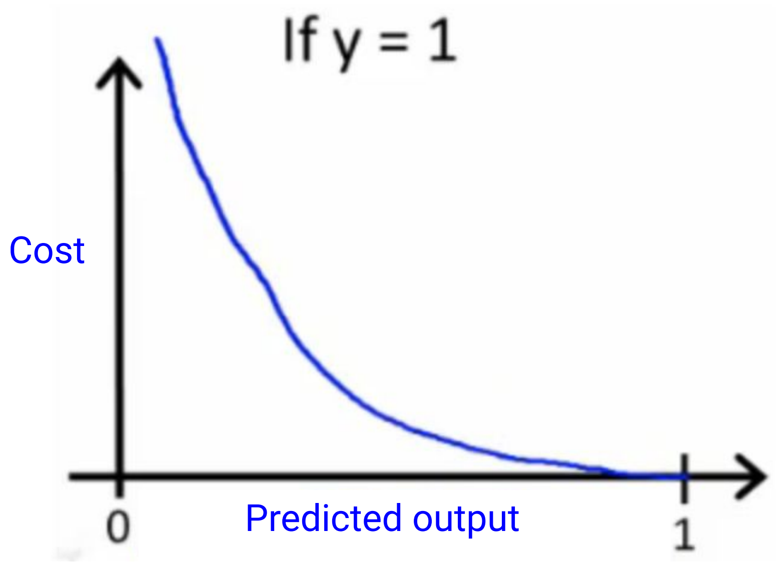
$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

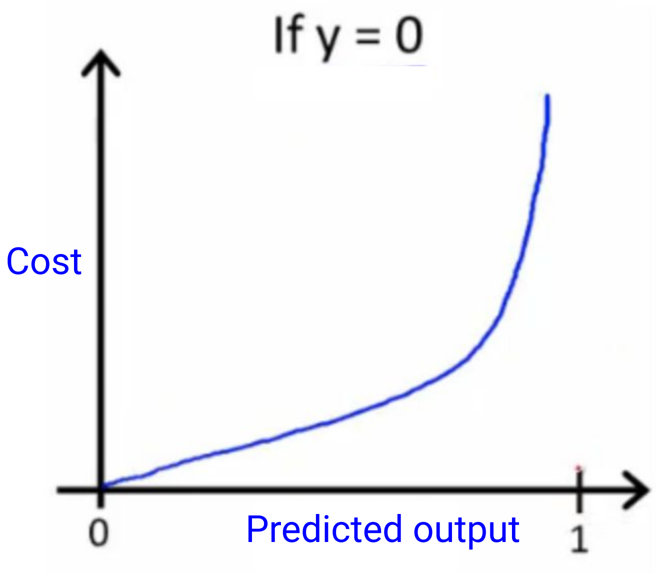
Linear regression cost function does not work for Logistic regression (many local optima, non-convex function)

For logistic regression, we will use this:

$$\begin{aligned} \mathrm{Cost}(h_\theta(x), y) &= -\log(h_\theta(x)) && \text{if y} = 1 \\ \mathrm{Cost}(h_\theta(x), y) &= -\log(1 - h_\theta(x)) && \text{if y} = 0 \end{aligned}$$

Penalize for large mistake





$$\mathrm{Cost}(h_\theta(x), y) = 0 \text{ if } h_\theta(x) = y$$
$$\mathrm{Cost}(h_\theta(x), y) \to \infty \text{ if } y = 0 \text{ and } h_\theta(x) \to 1$$
$$\mathrm{Cost}(h_\theta(x), y) \to \infty \text{ if } y = 1 \text{ and } h_\theta(x) \to 0$$

Cost function for logistic regression

$$\begin{aligned} \text{Cost}(h_\theta(x), y) &= -\log(h_\theta(x)) && \text{if y = 1} \\ \text{Cost}(h_\theta(x), y) &= -\log(1 - h_\theta(x)) && \text{if y = 0} \end{aligned}$$

Written in a compressed form

$$\text{Cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))$$

The full cost-function is:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

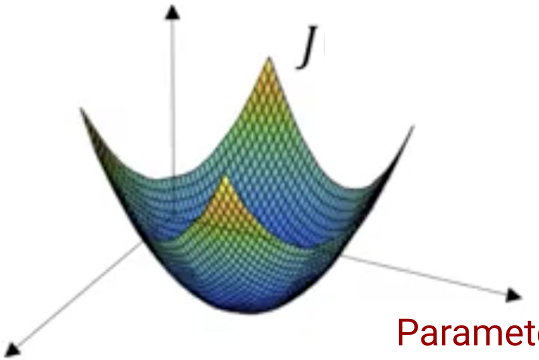
While implementing logistic regression in code, try to **avoid for loop** as much as possible, and try to do a **vectorized implementation.**

Gradient descent in logistic regression

Iterative process. Repeat the following until cost function is minimized.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

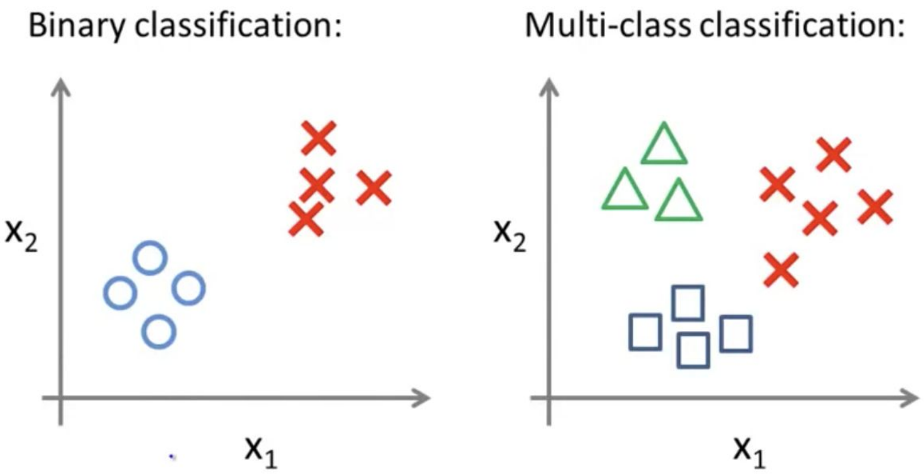
=> $$\theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$



<u>Note on vectorisation</u>:
It might not be possible to get rid of all **for loops** in all situations. For example, a for loop for **number of iterations** will still be needed.

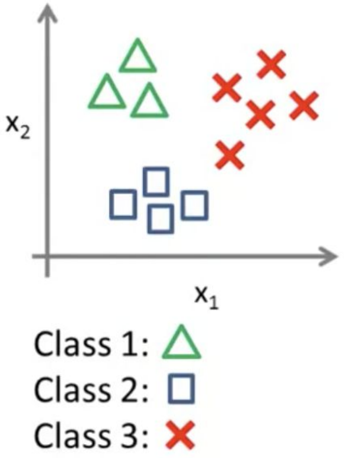
Multi-class classification problem

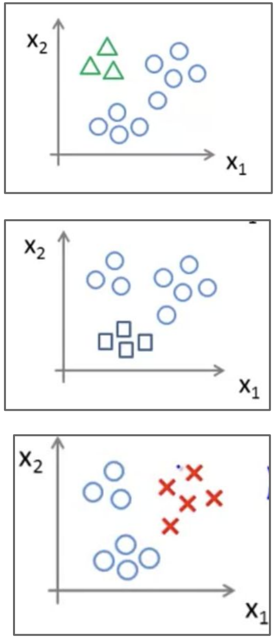
Instead of y = {0,1} we may have y = {0,1, 2, 3…}



Multi-class classification problem

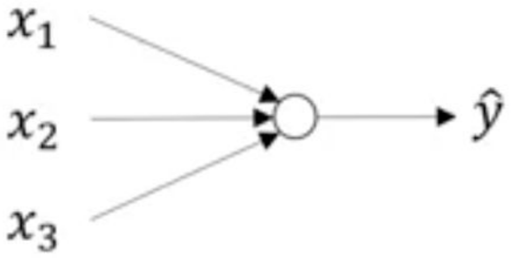
One-vs-all method



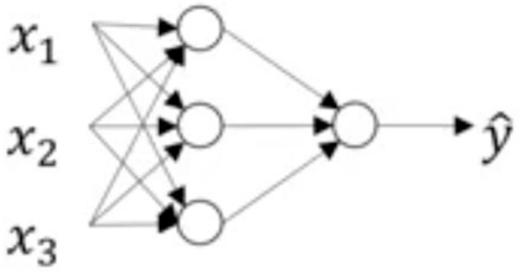


$$h_\theta^{(0)}(x) = P(y = 0|x;\theta)$$
$$h_\theta^{(1)}(x) = P(y = 1|x;\theta)$$
$$\dots$$
$$h_\theta^{(n)}(x) = P(y = n|x;\theta)$$
$$\text{prediction} = \max_i(h_\theta^{(i)}(x))$$

From Logistic regression to Neural Network (NN)



Logistic Regression (simplest possible neural network)



Neural Network

Logistic regression in NN terminologies

Sigmoid function is an activation function. Other activation functions are possible.

$\Theta_o \rightarrow b$

$\Theta_j \rightarrow w$

Parameters → weights

ŷ is also called “a”

In this terminology, the old equations can be rewritten

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

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

Logistic regression in NN terminologies

Gradient Descent

First, initialize w, b, then

$$
\begin{aligned}
&\text{Repeat } \{ \\
&w := w - \alpha \frac{\partial J(w,b)}{\partial w} \\
&b := b - \alpha \frac{\partial J(w,b)}{\partial b} \\
&\}
\end{aligned}
$$

We want to find w and b that

minimize the cost function J(w,b)

Logistic regression in NN terminologies

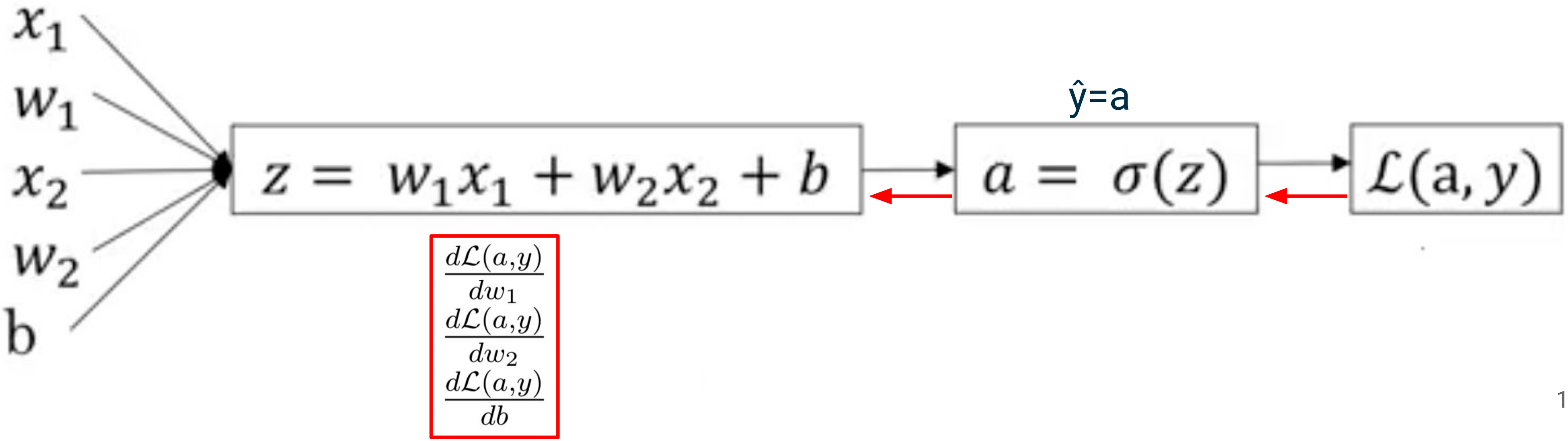
Forward propagation: compute the loss

Backward propagation: compute derivative

Computation graph

For one training example

Consider only 2 features, x_1 and x_2.

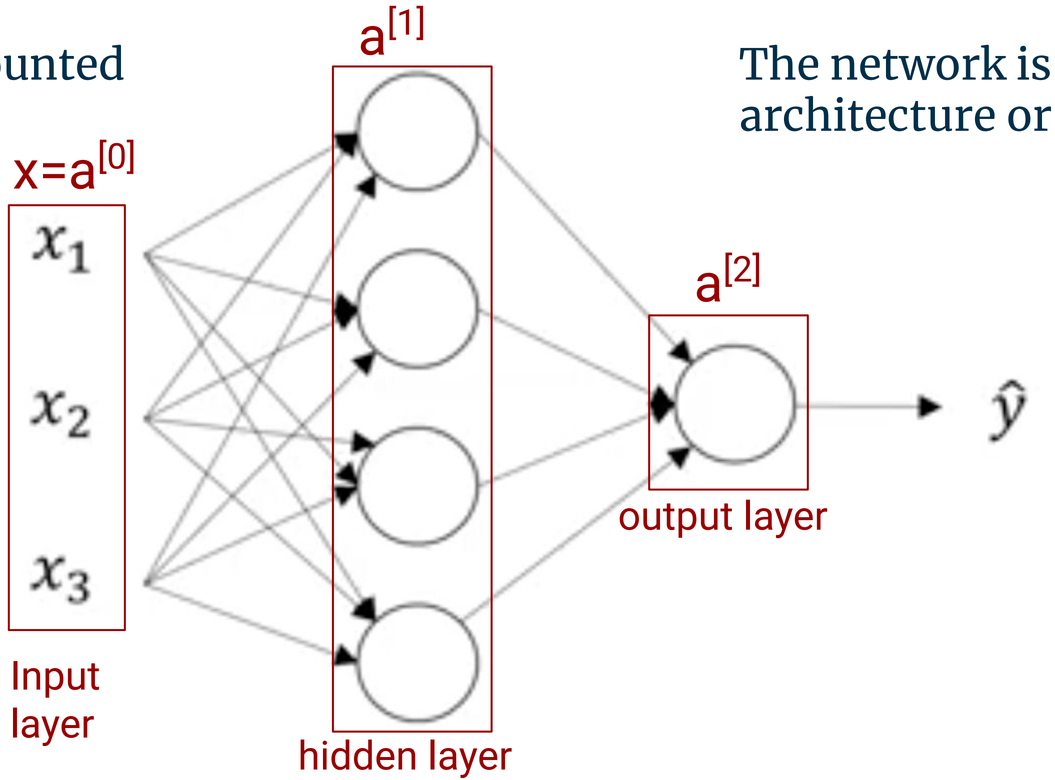


Layers in a Neural network (NN)

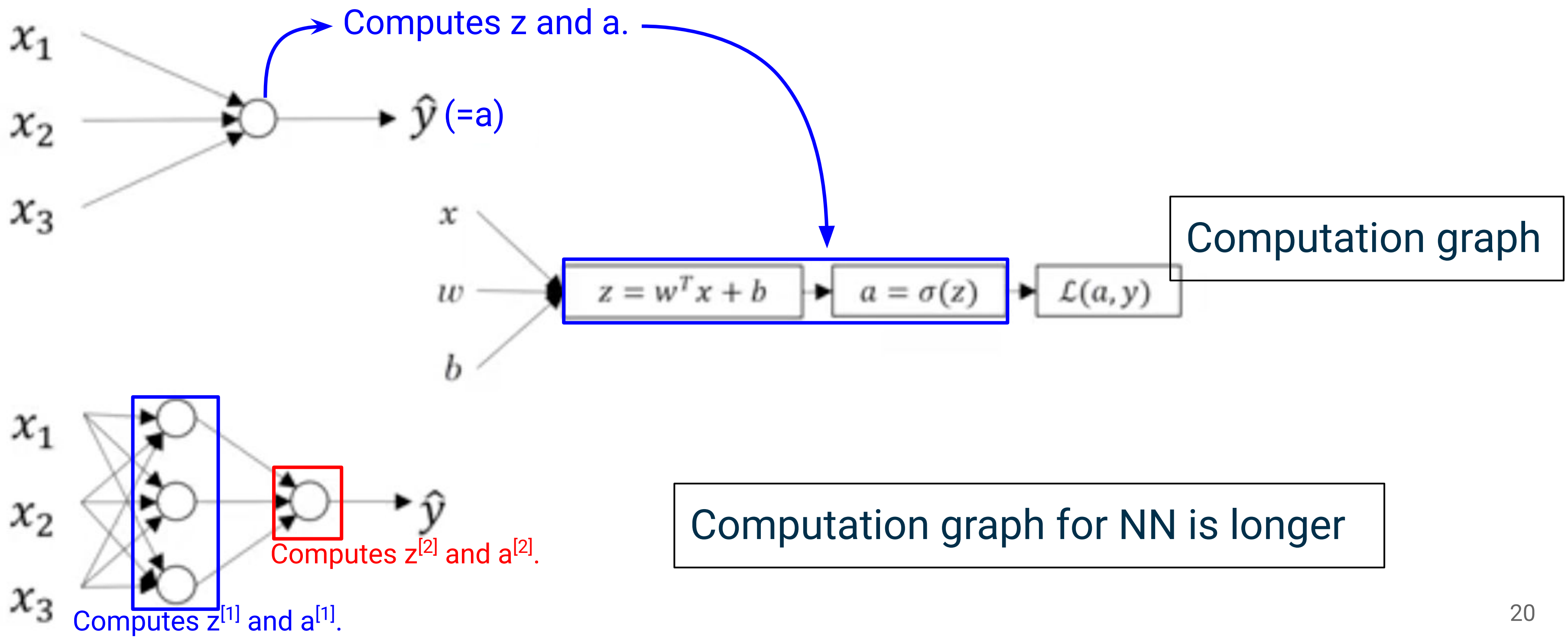
This is a 2-layer NN

Input layer not counted

The network is also called architecture or model.

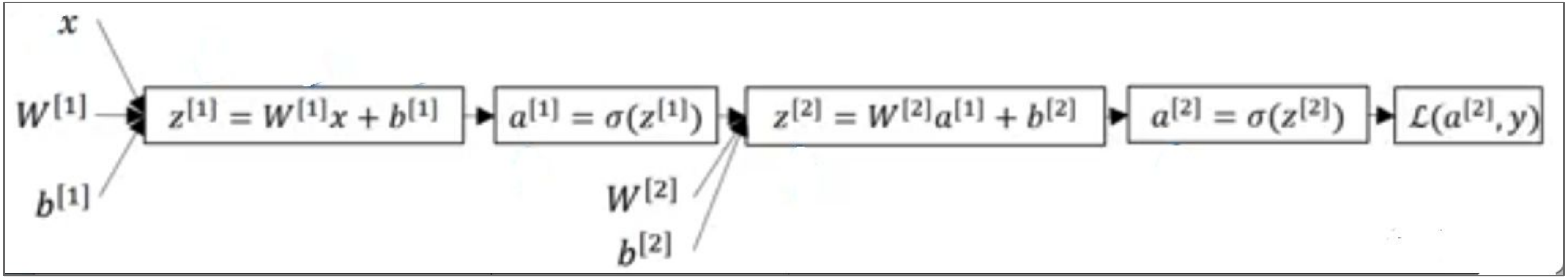


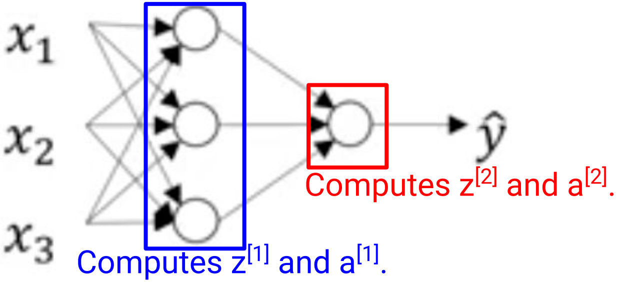
Logistic regression & NN: what is common?



Computation graph for NN is longer

NN computation graph





Activation function

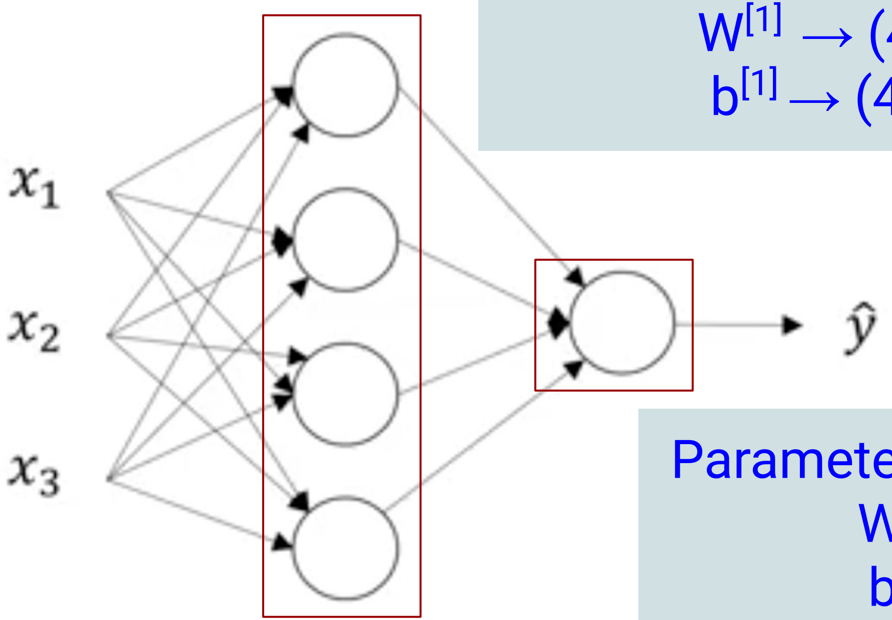
Using non-linear activation function is an essential part for NN.

NN learns interesting features from the given features by using the non-linearity of the activation.

In the absence of non-linear activation function, the power of NN is lost. This can be checked by using identity activation.

How many parameters are there?

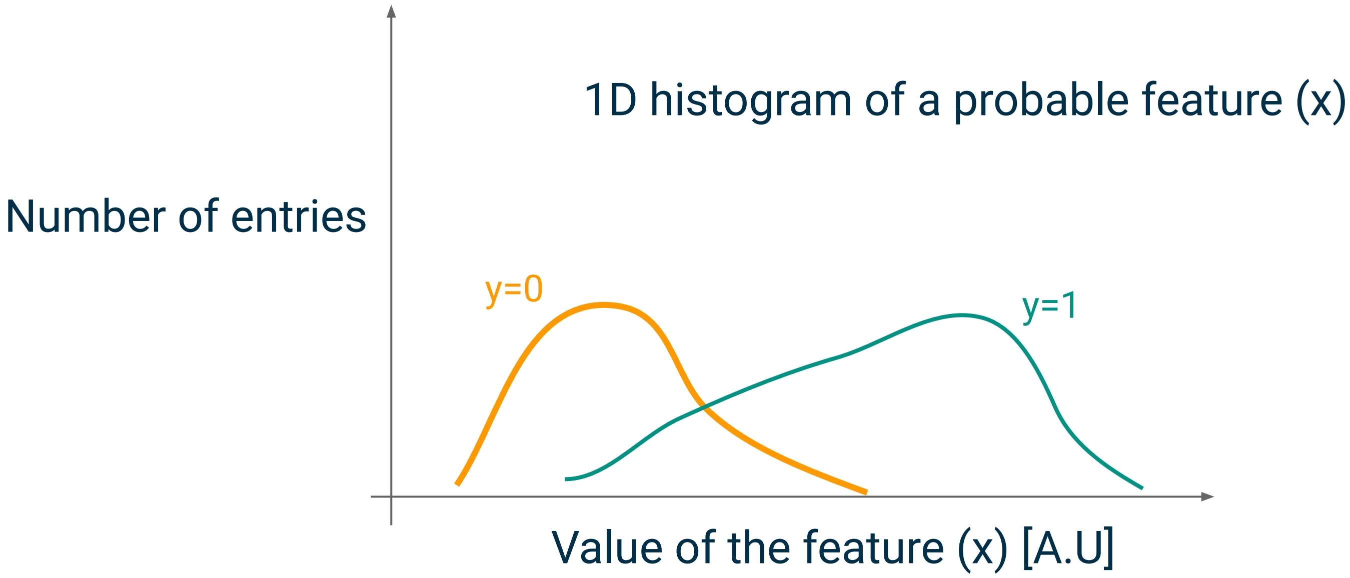
Parameters associated with layer 1
$W^{[1]} \rightarrow$ (4x3) matrix
$b^{[1]} \rightarrow$ (4x1) matrix



Parameters associated with layer 2
$W^{[2]} \rightarrow$ (1x4) matrix
$b^{[2]} \rightarrow$ (1x1) matrix

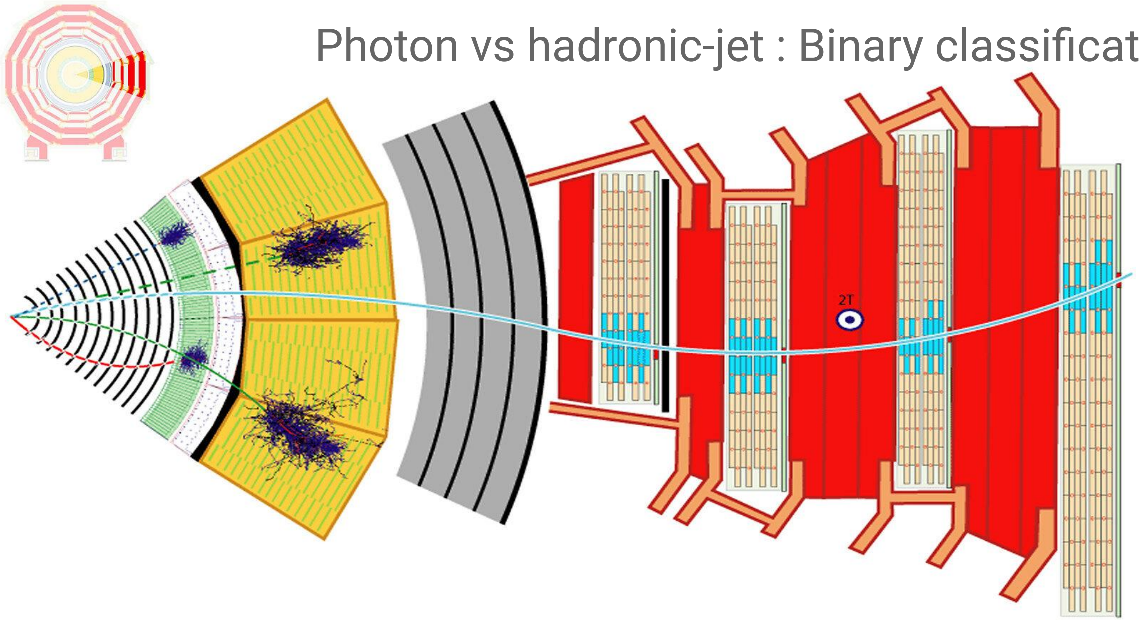
Identify good features (x)

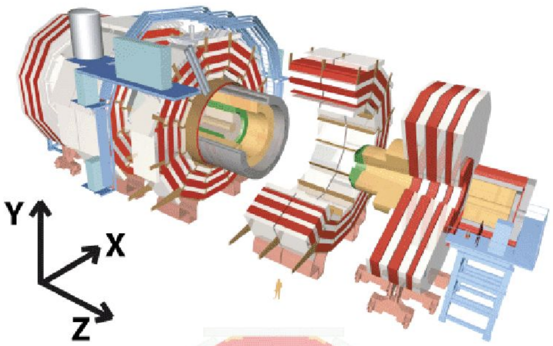
A good feature will have some discrimination power

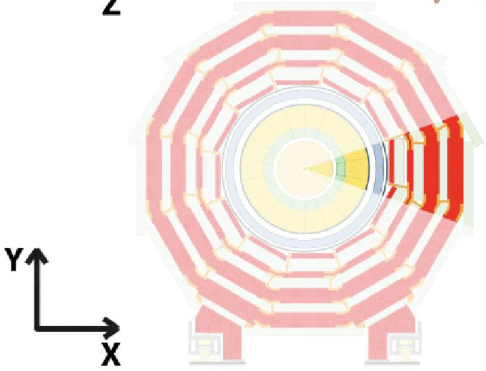


Example use-case in physics experiments

Photon vs hadronic-jet : Binary classification



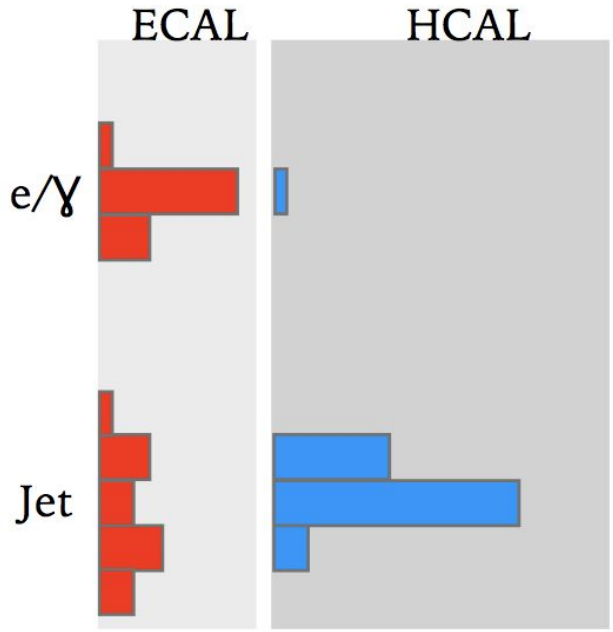




Photon is signal (y=1)
Hadronic-jet is background (y=0)

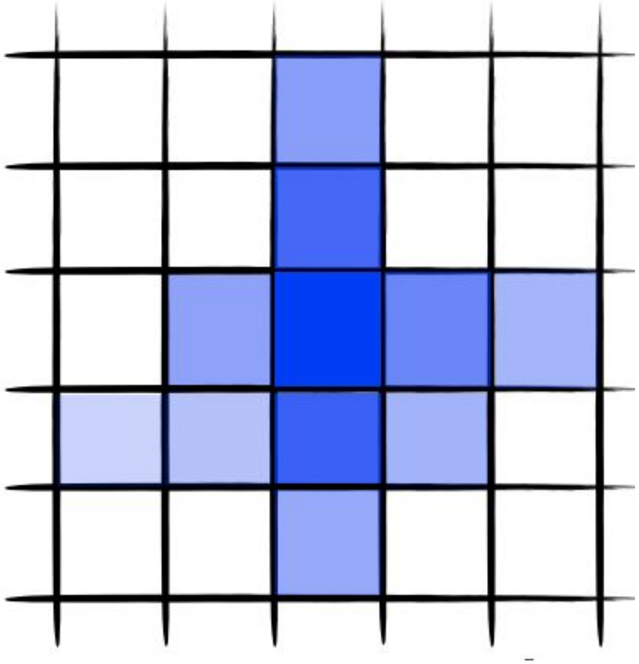
Feature 1 : H/E

Photon vs hadronic-jet : Binary classification



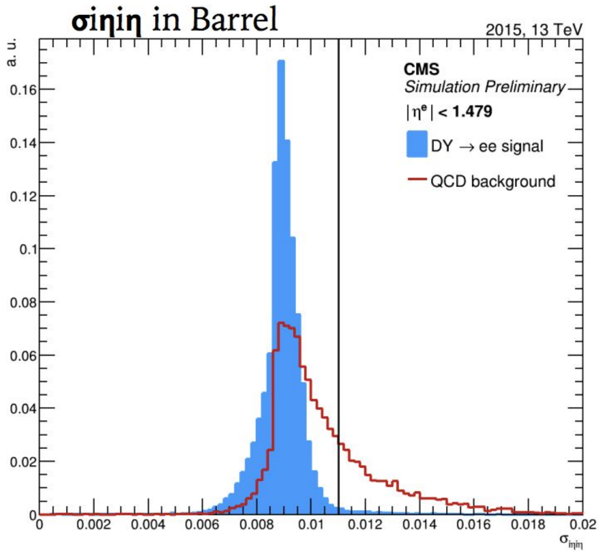
H/E variable is a good discriminator variable.

Feature 2 : Showershape in ECAL



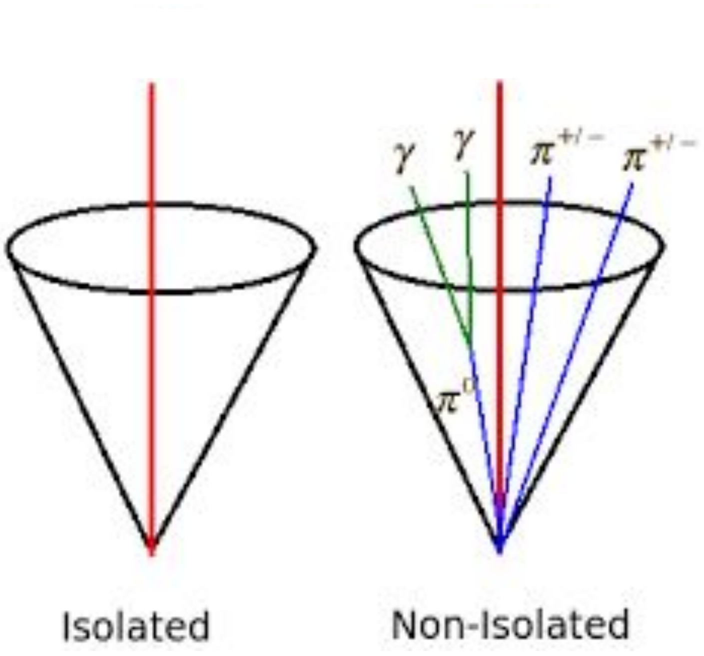
Photon vs hadronic-jet : Binary classification

Showershape is also a good discriminator between signal and background



Feature 3 : Isolation sum

Photon vs hadronic-jet : Binary classification



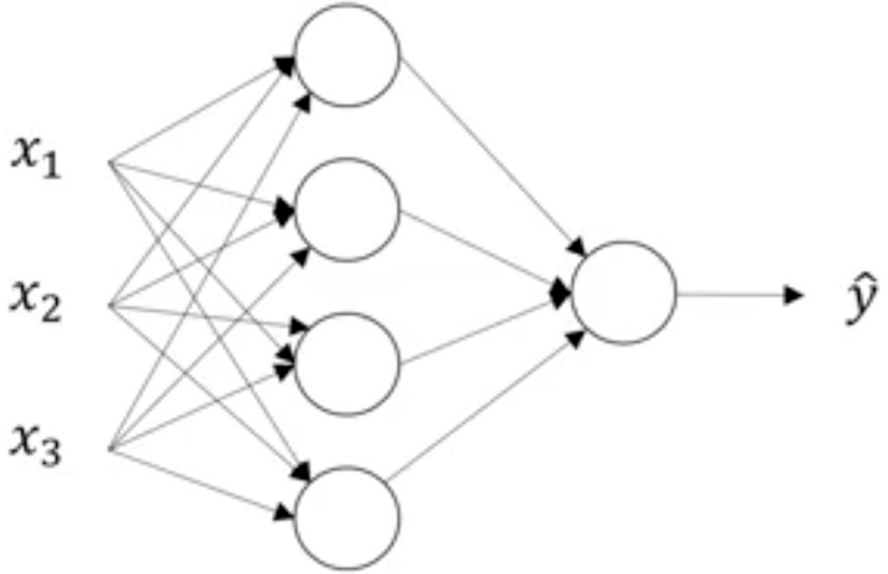
Isolation sum is also a good discriminator between signal and background

Example use-case in physics experiments

H/E

Showershape
in ECAL

Isolation sum



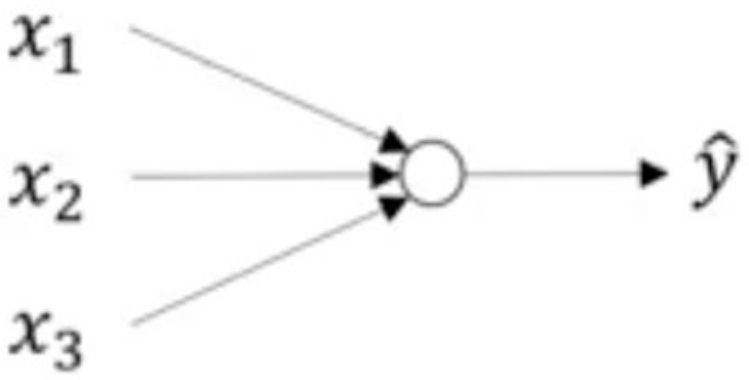
Photon vs hadronic-jet?

Example use-case in physics experiments

H/E

Showershape
in ECAL

Isolation sum



Photon vs hadronic-jet?

back up slides