Non-linear Hypothesis

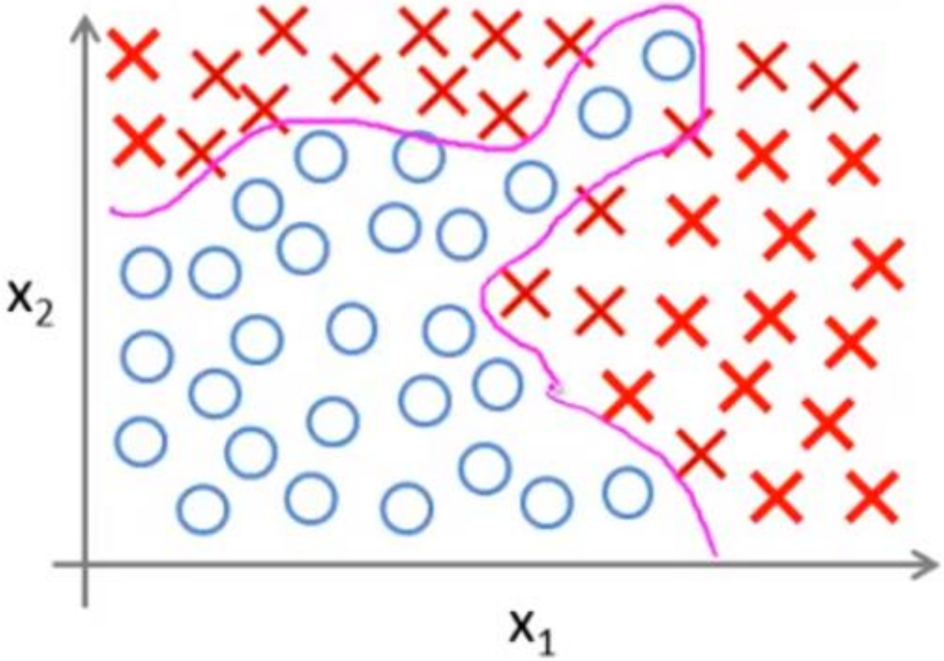
Motivations

Neural Networks: Representation

Introduction

- Neutral networks is actually a pretty old idea, but had fallen out of favor for a while.
- But today, it is the state of the art technique for many different machine learning problems.
- So why do we need yet another learning algorithm? We already have linear regression and we have logistic regression, so why do we need neural networks?

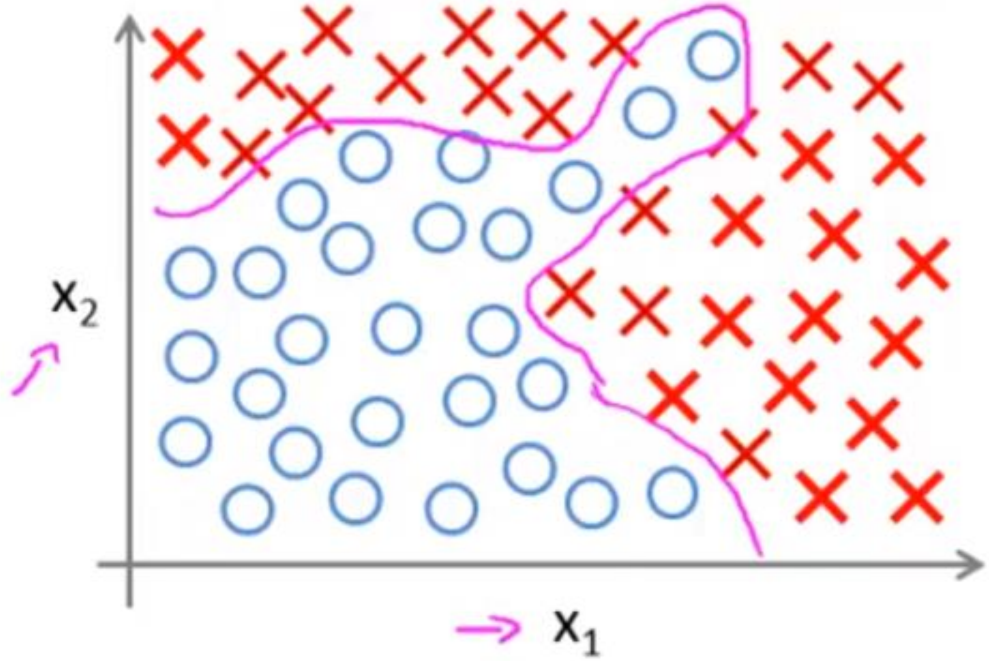
Non-linear Classification



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \ldots)$$

We have only two features here. What happens if we have much more features?

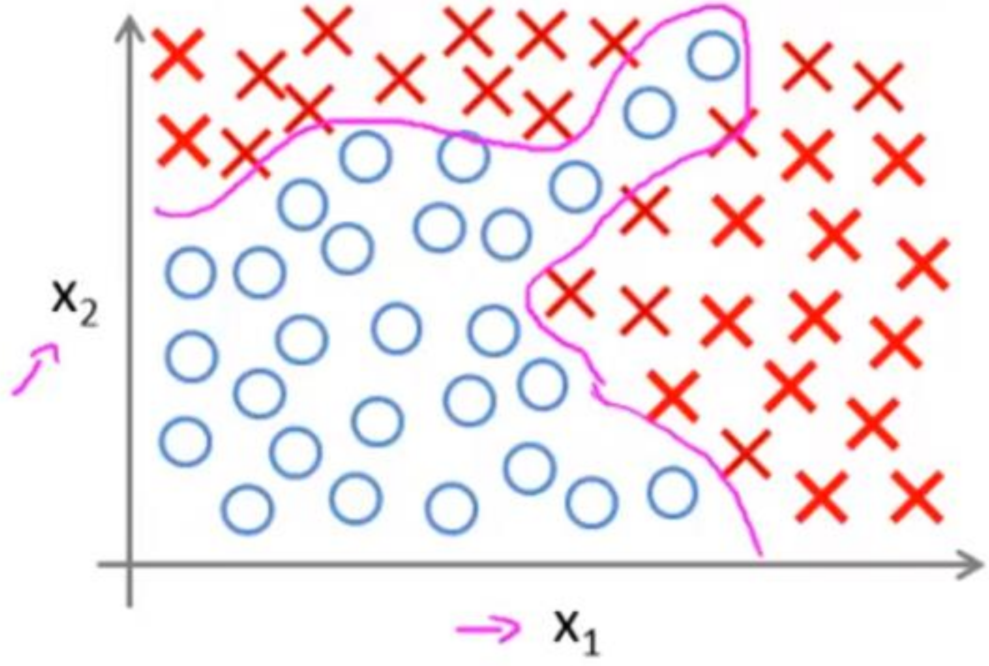
Non-linear Classification



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \dots)$$

$x_1 =$ size
$x_2 =$ # bedrooms
$x_3 =$ # floors
$x_4 =$ age
$\dots$
x_{100}

Non-linear Classification



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \ldots)$$

$x_1^2, x_1x_2, x_1x_3, x_1x_4 \ldots x_1x_{100}$
$x_2^2, x_2x_3 \ldots$
≈ 5000 feature $O(n^2)$

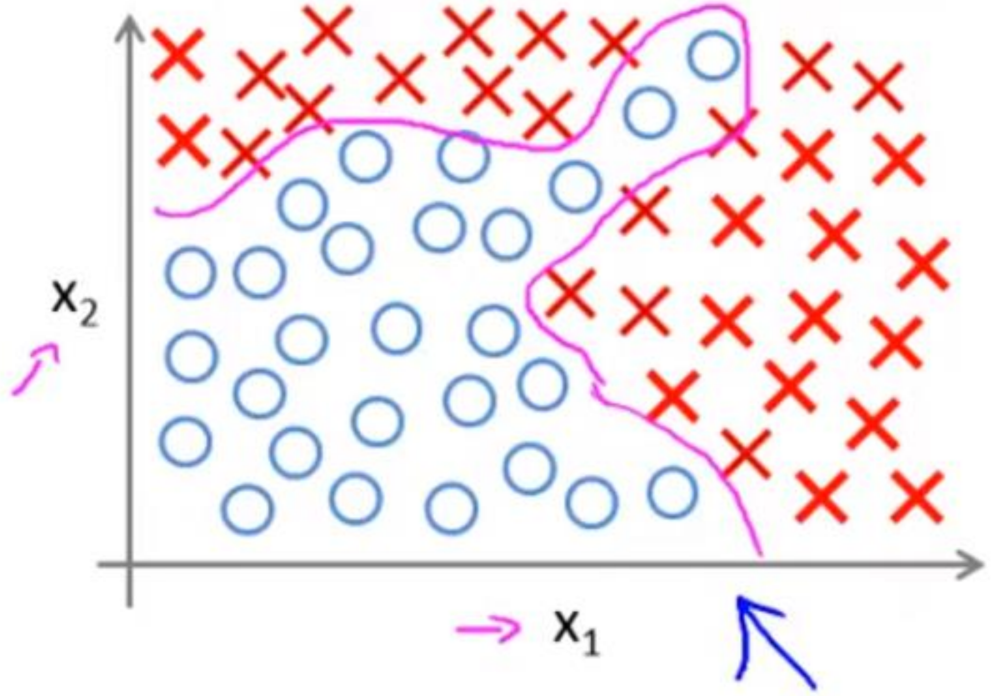
x_1 = size
x_2 = # bedrooms
x_3 = # floors
x_4 = age
...
x_{100}

$n = 100$

One way is to reduce the number of order of interactions.

2:49 / 9:36 Andrew Ng

Non-linear Classification



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \dots)$$

$x_1 =$ size

$x_2 =$ # bedrooms

$x_3 =$ # floors

$x_4 =$ age

$\dots$

x_{100}

$n = 100$

→ $x_1^2, x_1x_2, x_1x_3, x_1x_4 \dots x_1x_{100}$

$x_2^2, x_2x_3 \dots$

≈ 5000 feature $\quad O(n^2)$

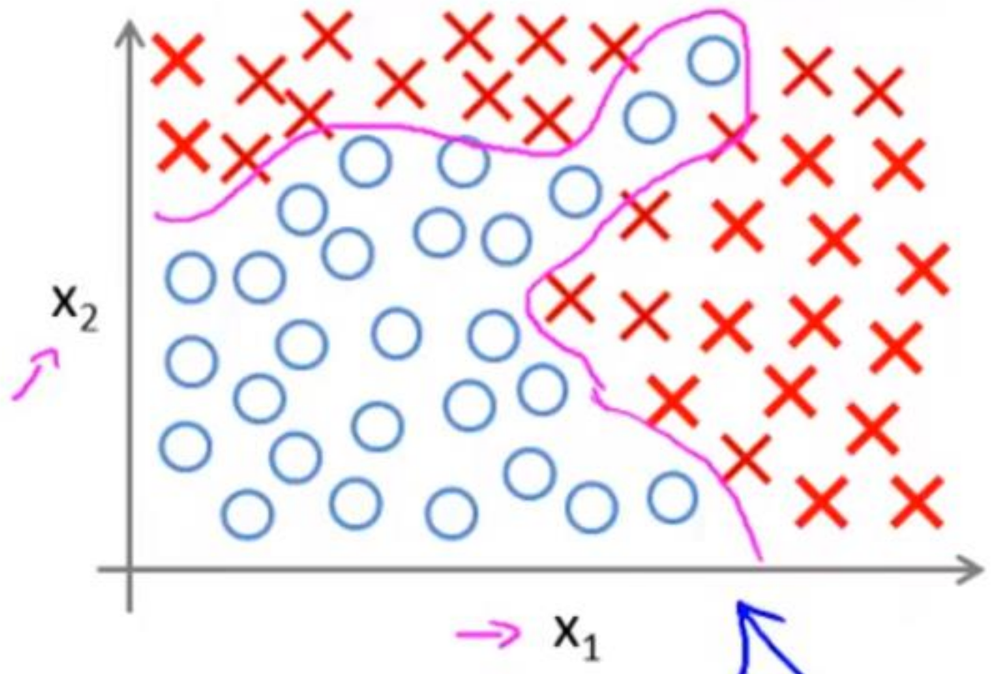
→ $x_1^2, x_2^2, x_3^2, \dots, x_{100}^2 \quad \approx \frac{n^2}{2}$

10

You cannot cover complex shapes like above…

Andrew Ng

Non-linear Classification



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2$$
$$+\theta_3 x_1 x_2 + \theta_4 x_1^2 x_2$$
$$+\theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \ldots)$$

$\rightarrow x_1^2, x_1x_2, x_1x_3, x_1x_4 \ldots x_1x_{100}$

$x_2^2, x_2x_3 \ldots$

$\approx$ 5000 feature $\quad O(n^2)$

$\approx \frac{n^2}{2}$

x_1 = size

x_2 = # bedrooms

x_3 = # floors

x_4 = age

...

x_{100}

$n=100$

$\rightarrow x_1^2, x_2^2, x_3^2, \ldots, x_{100}^2$

$\rightarrow x_1x_2x_3, x_1^2x_2, x_{10}x_{11}x_{17}, \ldots$

$O(n^3)$

170,000

What is this?

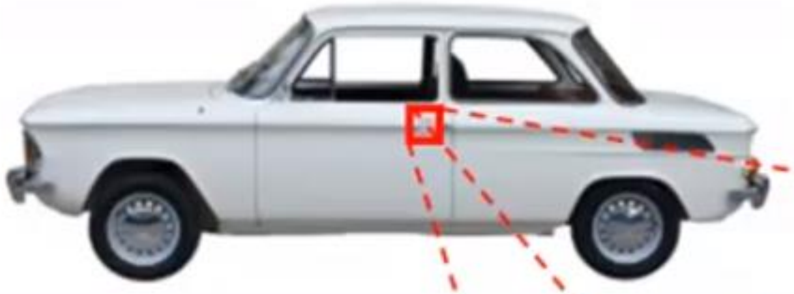
You see this:

What is this?

You see this:

What is this?

You see this:

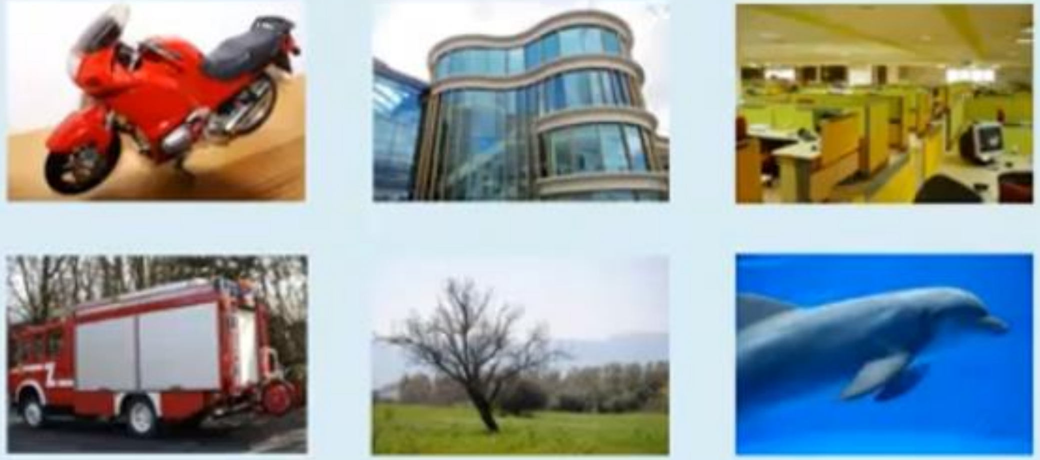


But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Computer Vision: Car detection

Cars



Not a car

Testing:

What is this?

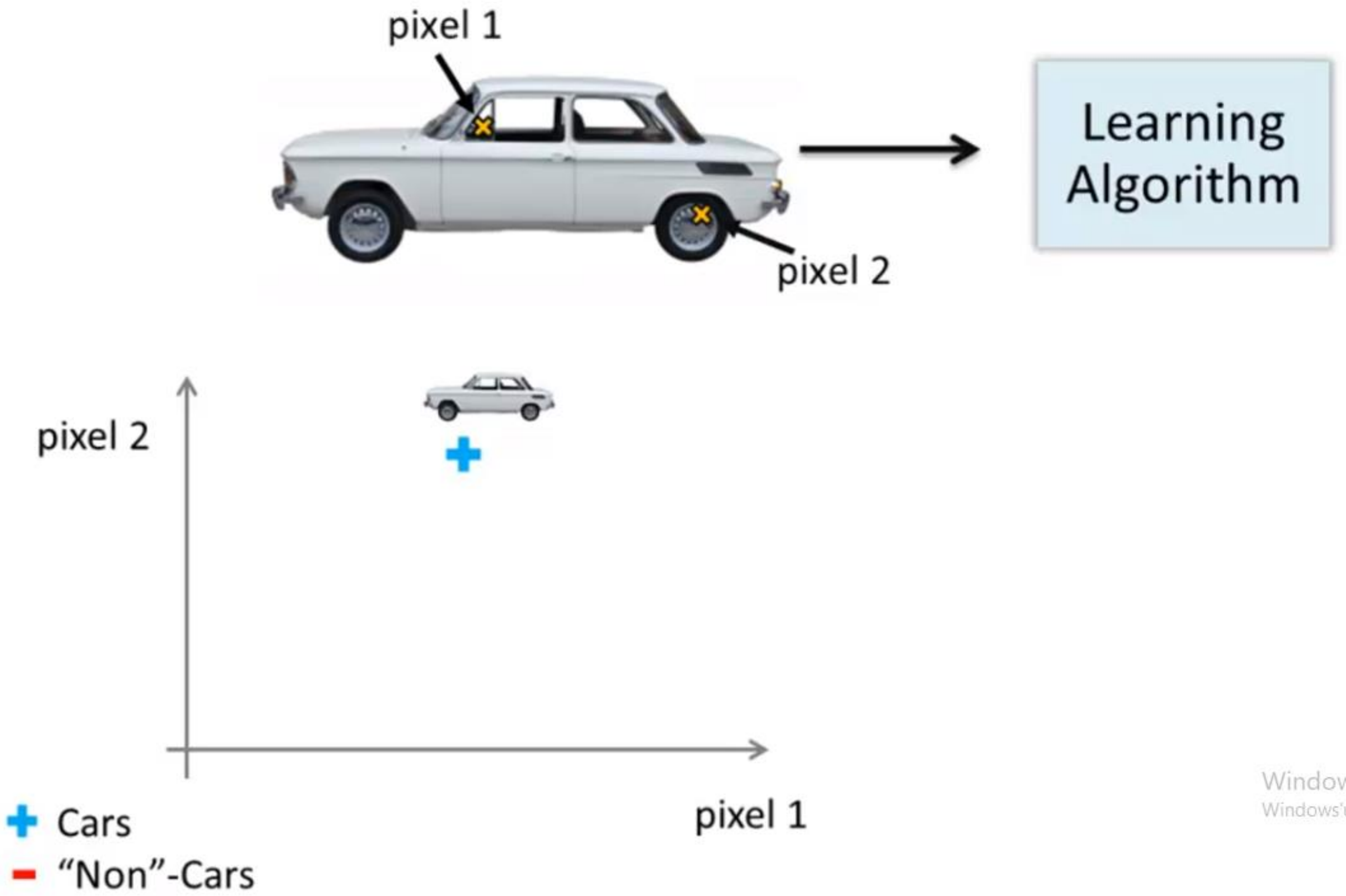
Andrew Ng

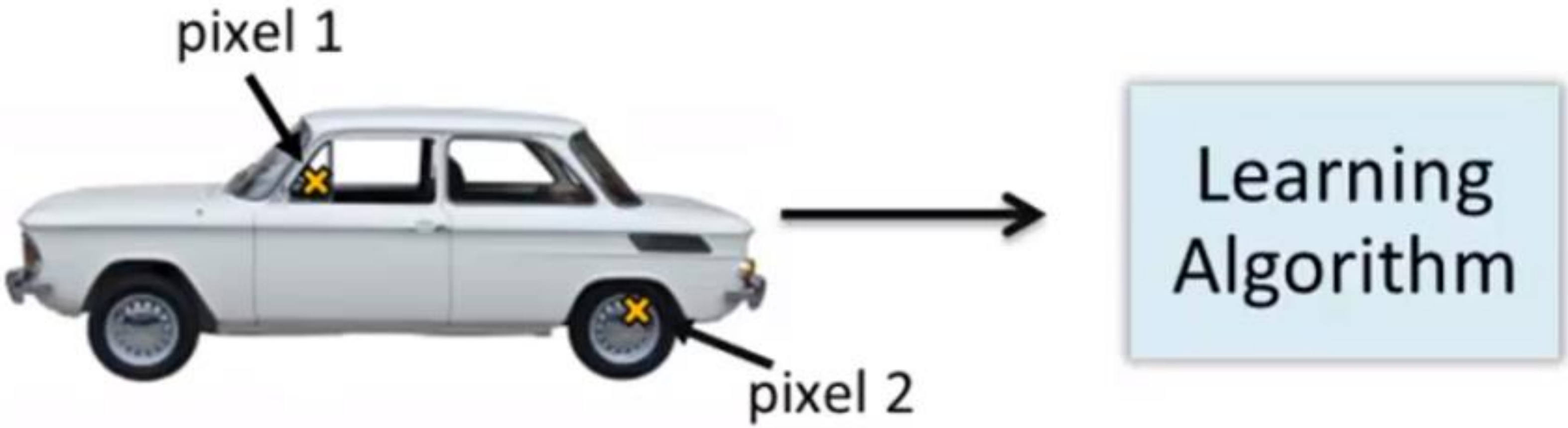
→ Learning Algorithm

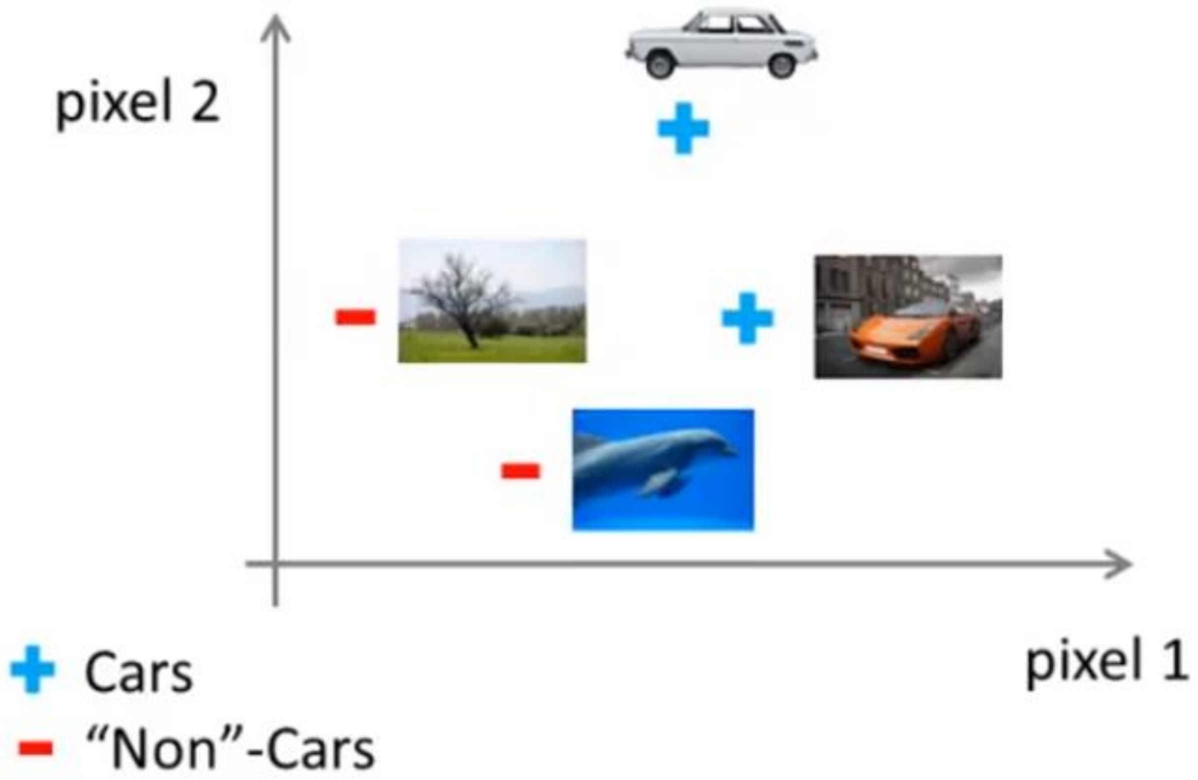
pixel 2

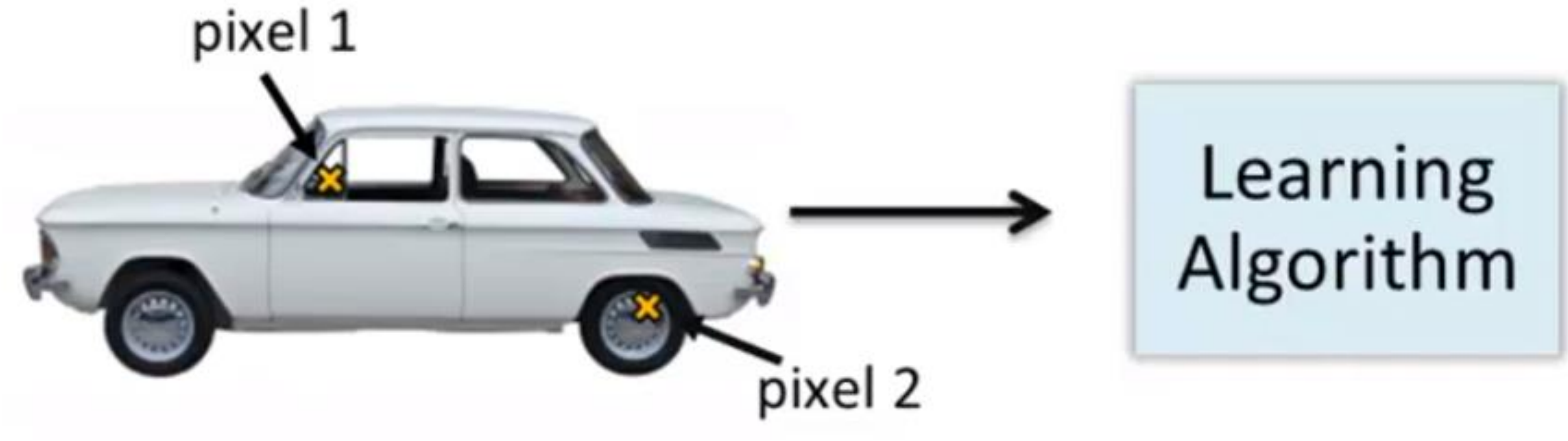
pixel 1

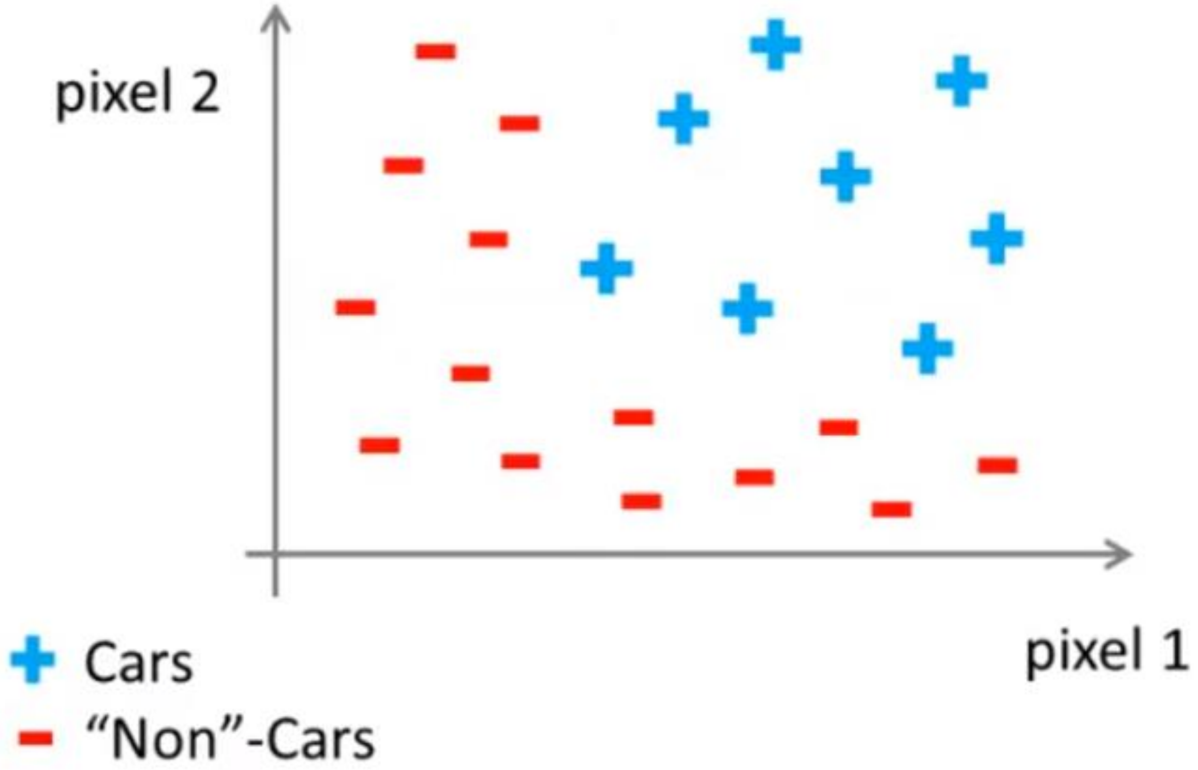
+ Cars
− “Non”-Cars

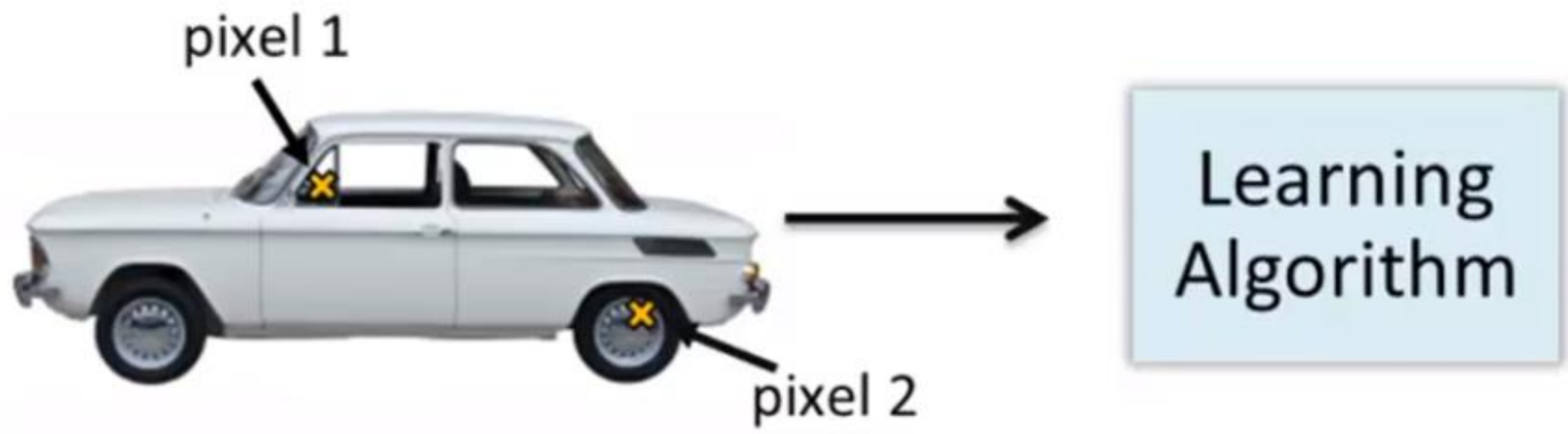

pixel 1
pixel 2
Learning Algorithm
pixel 2
pixel 1
Cars
"Non"-Cars

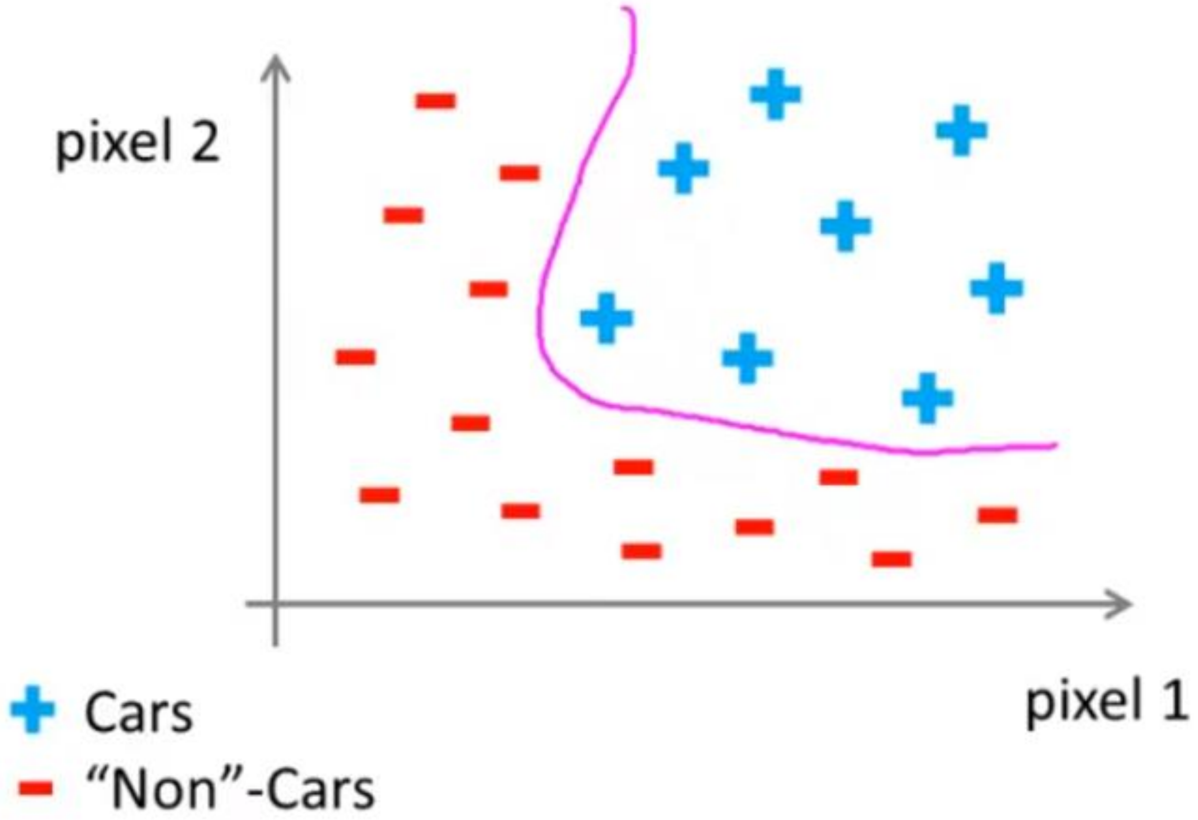

pixel 1
pixel 2
Learning Algorithm


pixel 2
pixel 1
Cars
“Non”-Cars

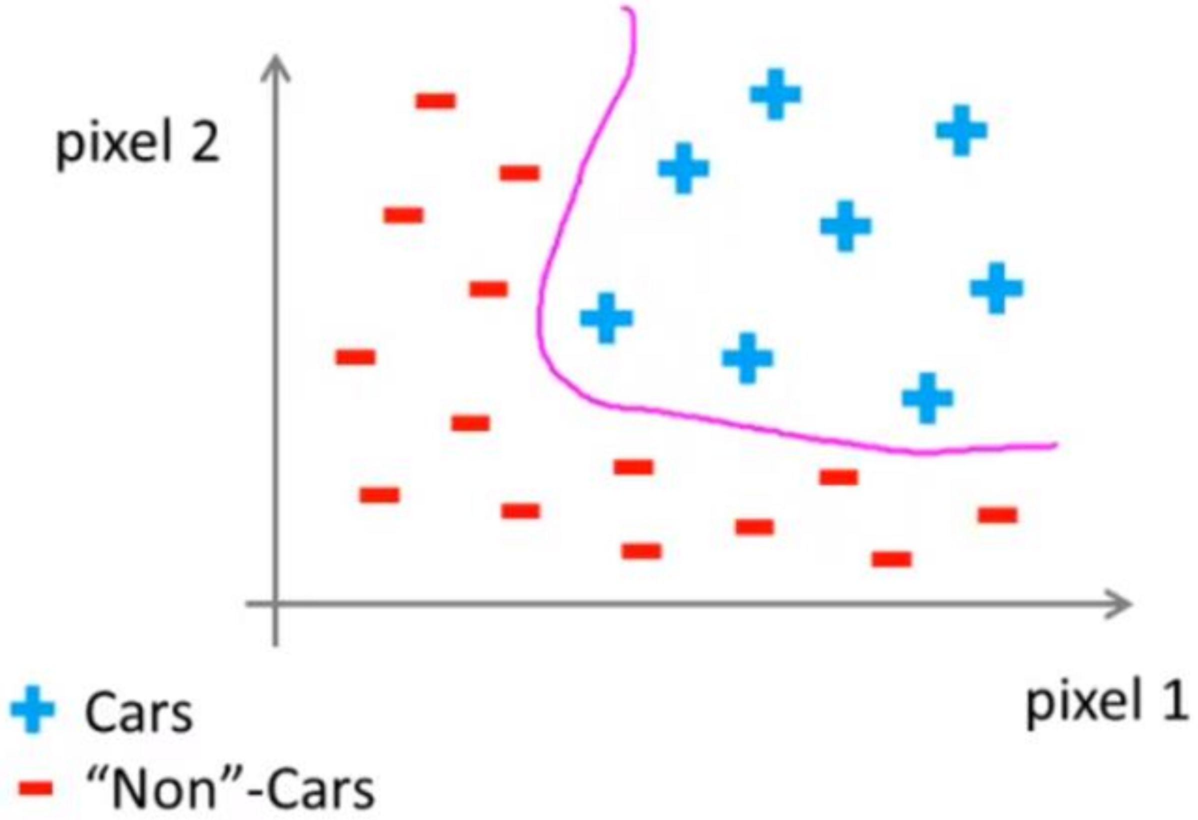

pixel 1
pixel 2
Learning Algorithm


pixel 2
pixel 1
Cars
"Non"-Cars


pixel 1
pixel 2
Learning
Algorithm


pixel 2
pixel 1
Cars
“Non”-Cars

Learning Algorithm

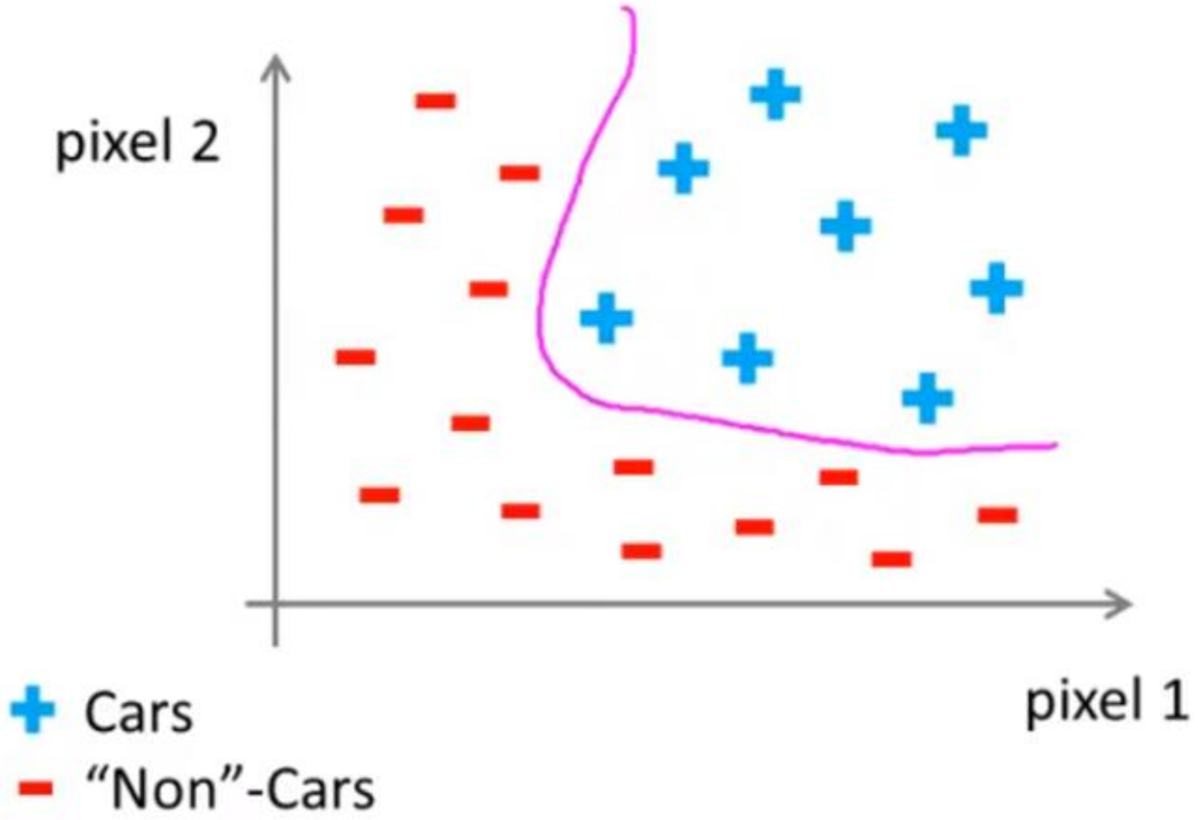


50 x 50 pixel images → 2500 pixels

$n = 2500$ (7500 if RGB)

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$

Learning Algorithm

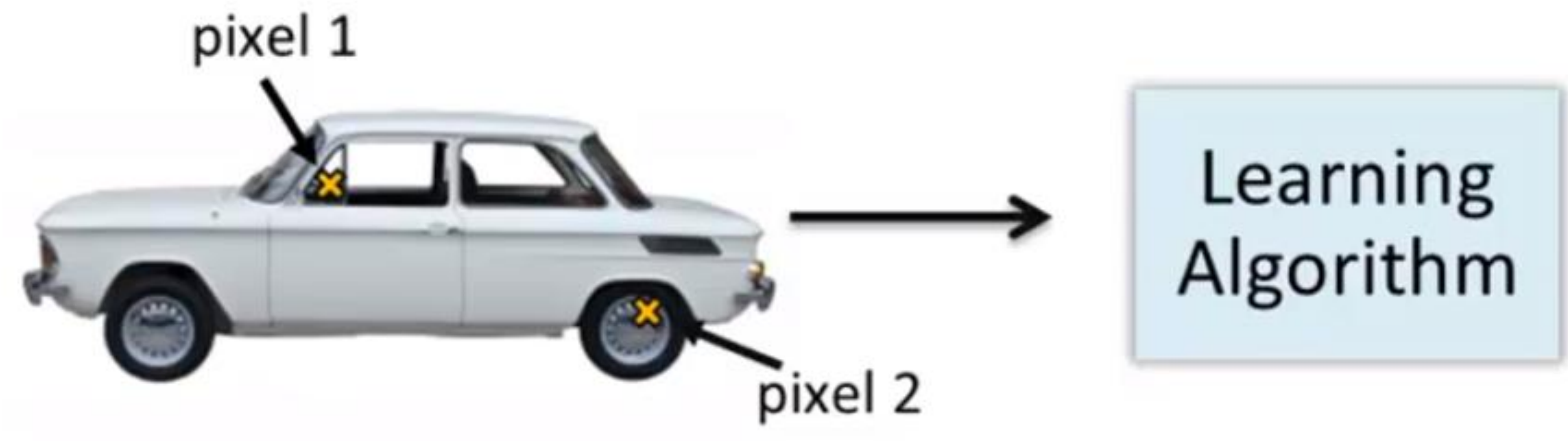


50 x 50 pixel images → 2500 pixels

$\boxed{n = 2500}$ (7500 if RGB)

0-255

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$



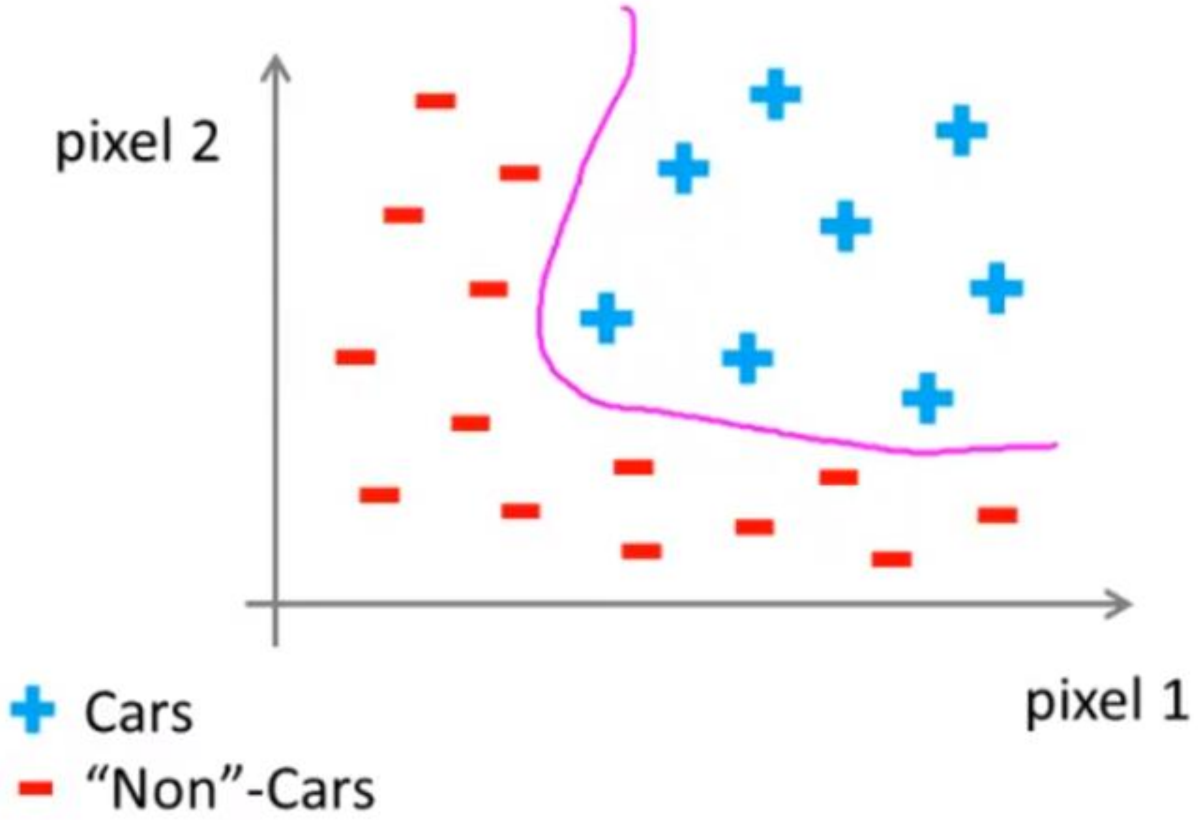
50 x 50 pixel images → 2500 pixels

$\boxed{n = 2500}$ (7500 if RGB)

0-255

$$\rightarrow x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$

Quadratic features ($x_i \times x_j$): ≈3 million features



Andrew Ng

Exercise

- Suppose you are learning to recognize cars from 100×100 pixel images (grayscale, not RGB). Let the features be pixel intensity values. If you train logistic regression including all the quadratic terms ($x_i x_j$)as features, about how many features will you have?
  - 5.000
  - 100.000
  - 50 M
  - 5 B