E-mail spam classifier

Spam

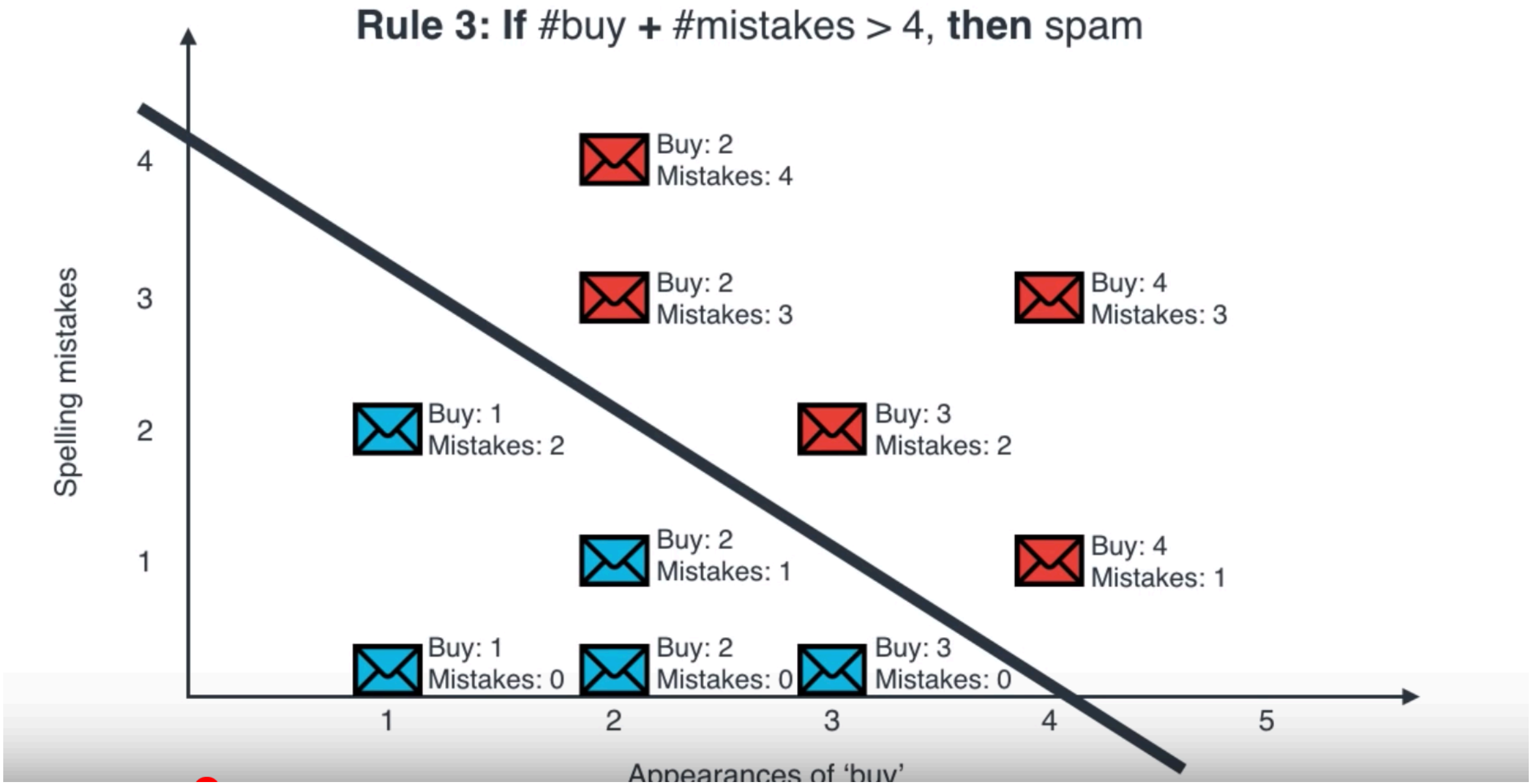
Buy, l0ts of money,
now, che@p buy
buy free mon3y

Non-spam (ham)

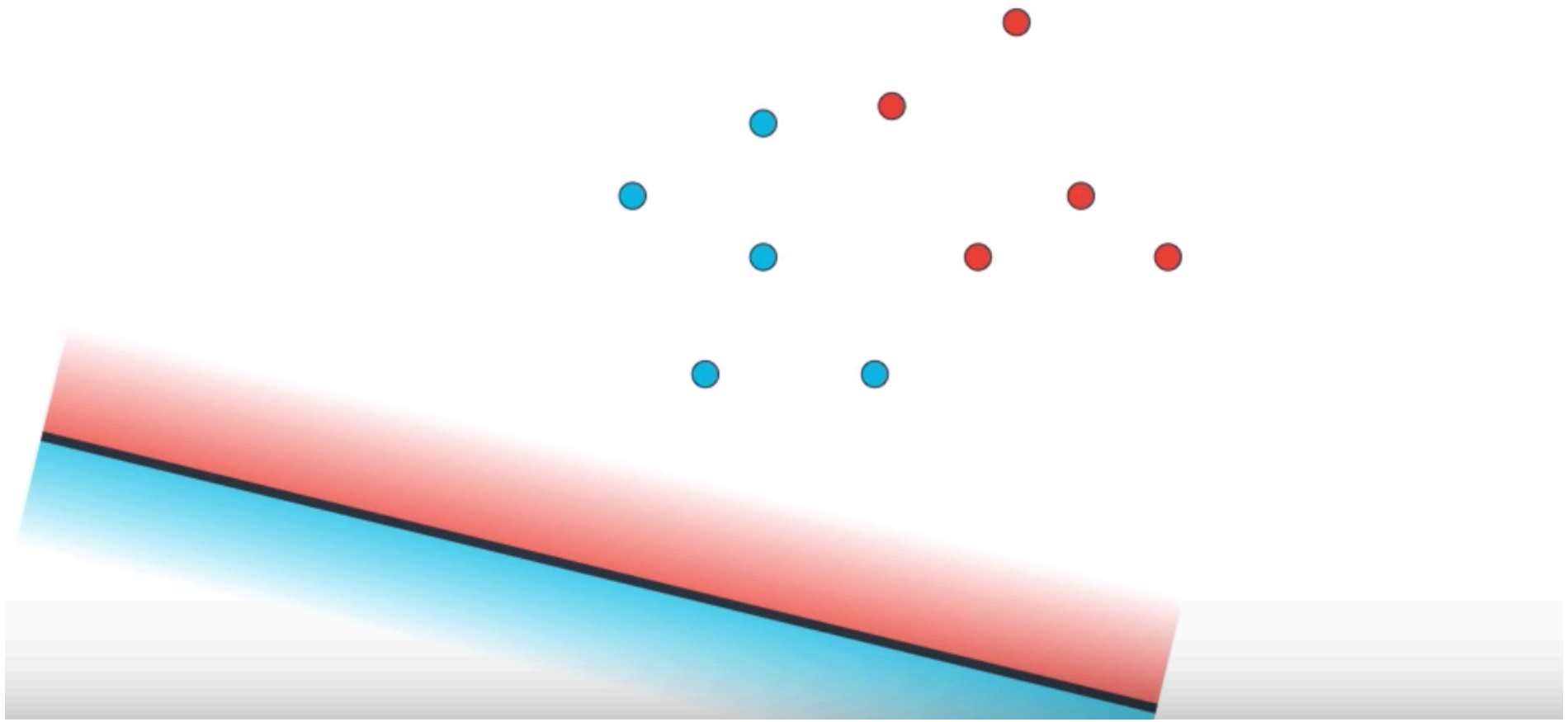
Hello grandson,
I made cookies.
Love, Grandma

‘buy’

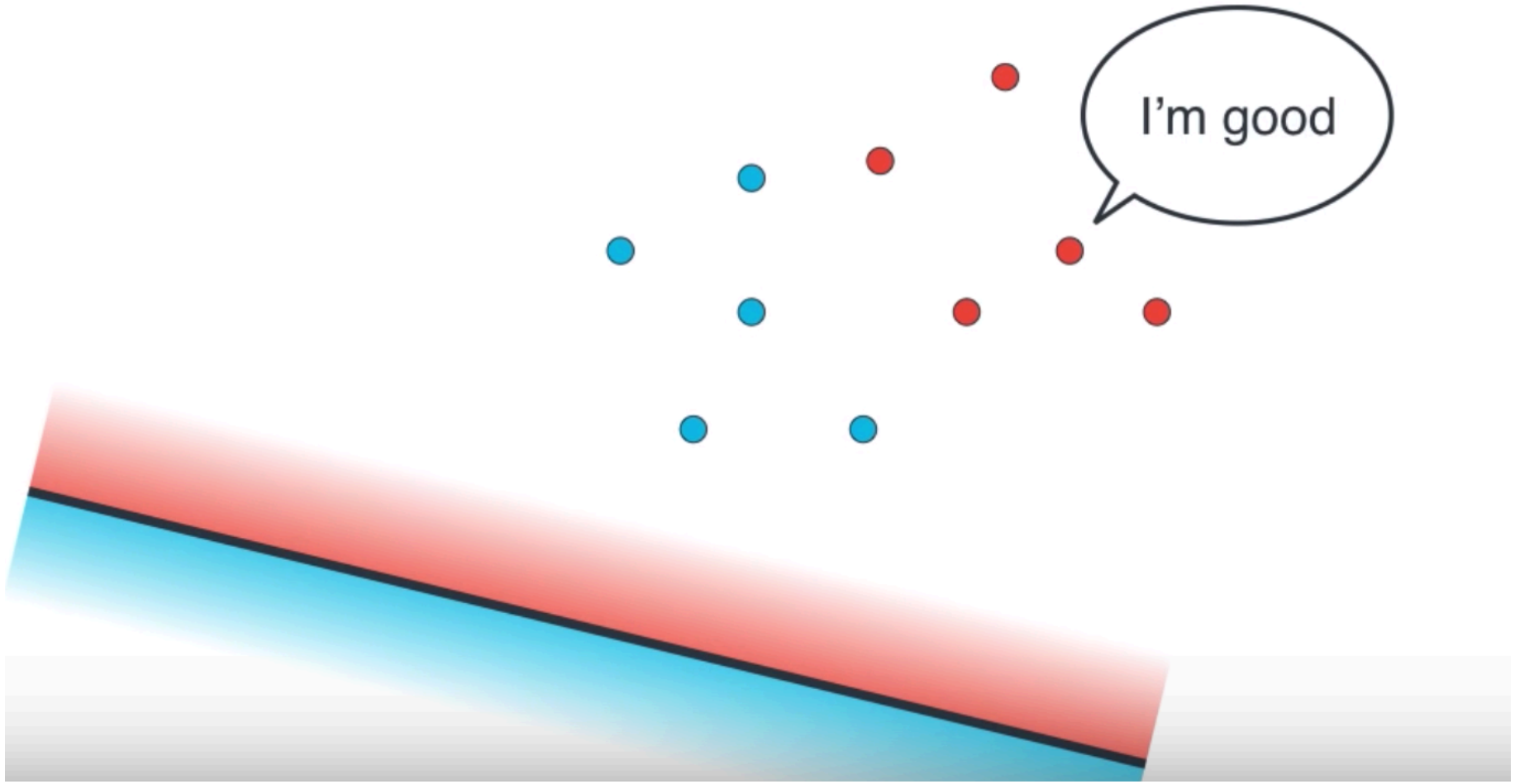
E-mails	Buy + Mistakes
 Buy: **4**, Mistakes: **3**	7
 Buy: **1**, Mistakes: **2**	3
 Buy: **0**, Mistakes: **2**	2
 Buy: **2**, Mistakes: **1**	3
 Buy: **4**, Mistakes: **1**	5
 Buy: **0**, Mistakes: **3**	3
 Buy: **2**, Mistakes: **3**	5
 Buy: **0**, Mistakes: **1**	1
 Buy: **2**, Mistakes: **4**	6
 Buy: **3**, Mistakes: **2**	5


Rule 3: If #buy + #mistakes > 4, then spam
Spelling mistakes
4
3
2
1
Buy: 2
Mistakes: 4
Buy: 2
Mistakes: 3
Buy: 4
Mistakes: 3
Buy: 1
Mistakes: 2
Buy: 3
Mistakes: 2
Buy: 2
Mistakes: 1
Buy: 4
Mistakes: 1
Buy: 1
Mistakes: 0
Buy: 2
Mistakes: 0
Buy: 3
Mistakes: 0
1
2
3
4
5
Appearances of 'buy'

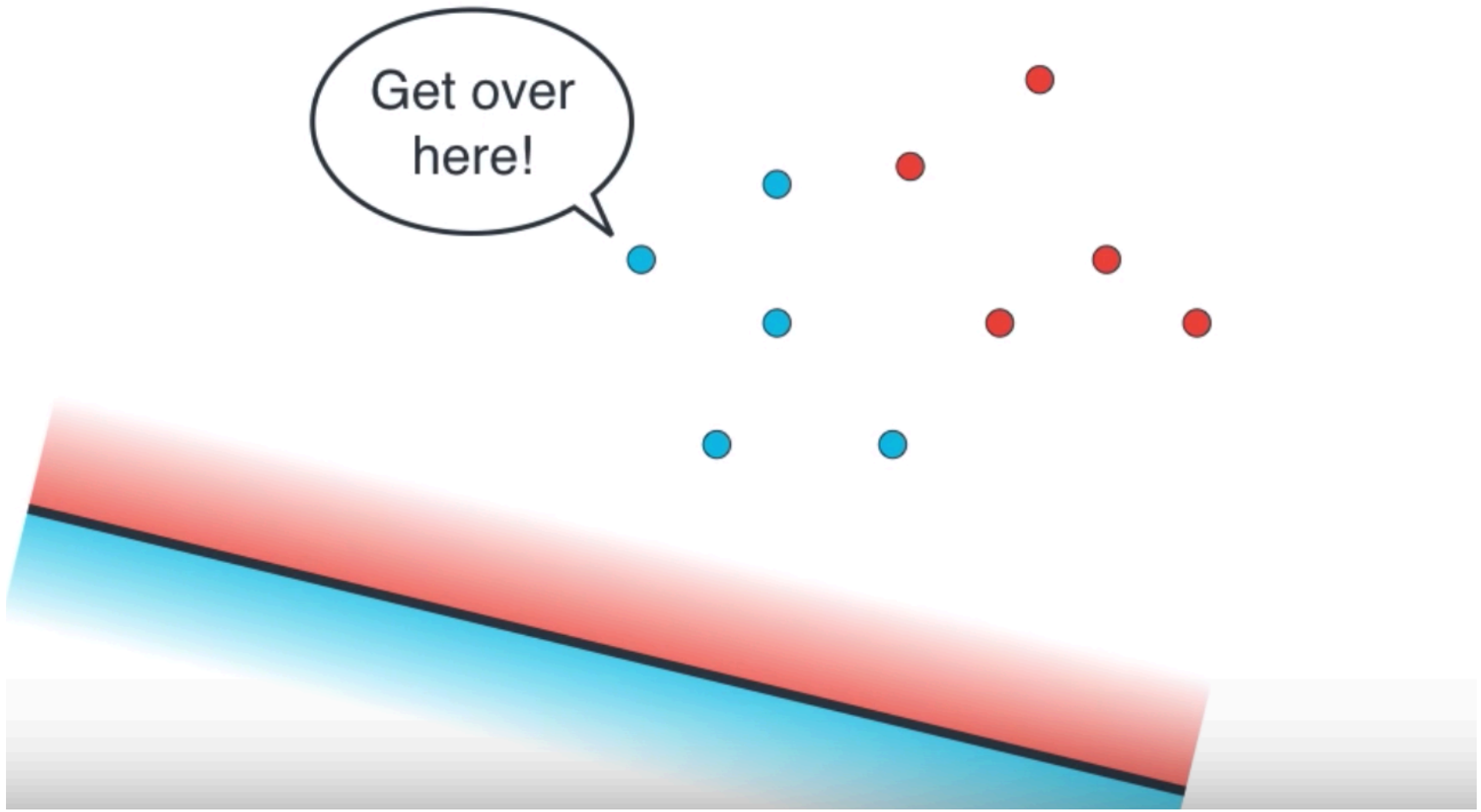
Perceptron Algorithm



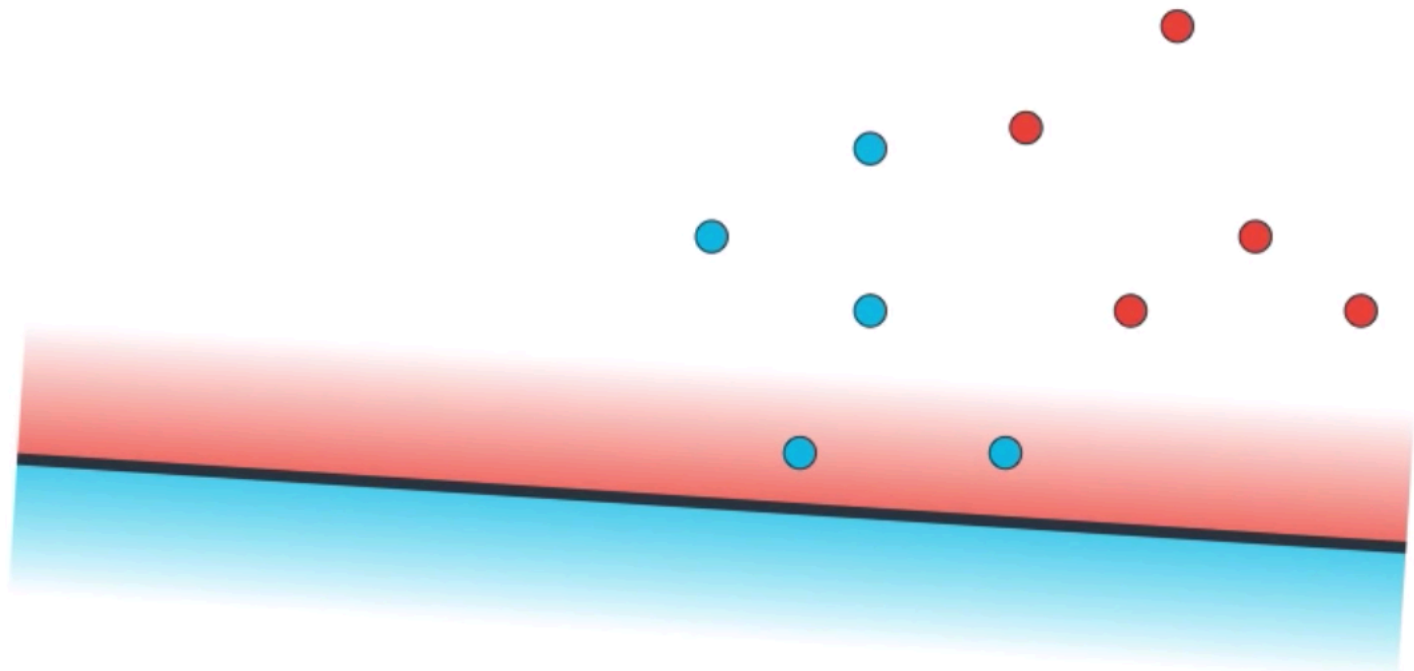
Perceptron Algorithm



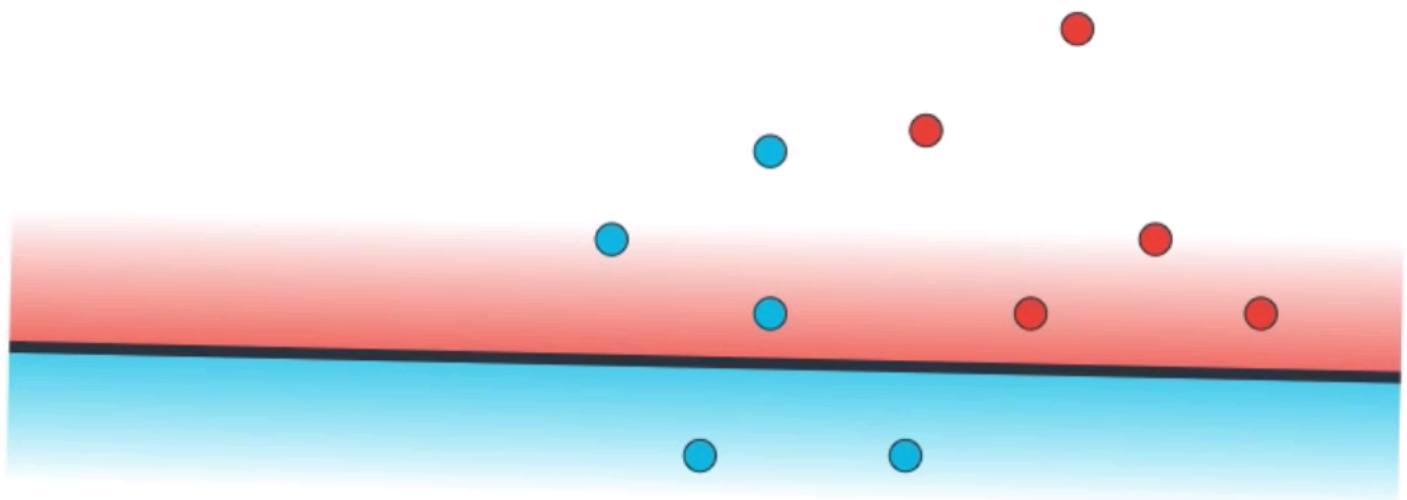
Perceptron Algorithm



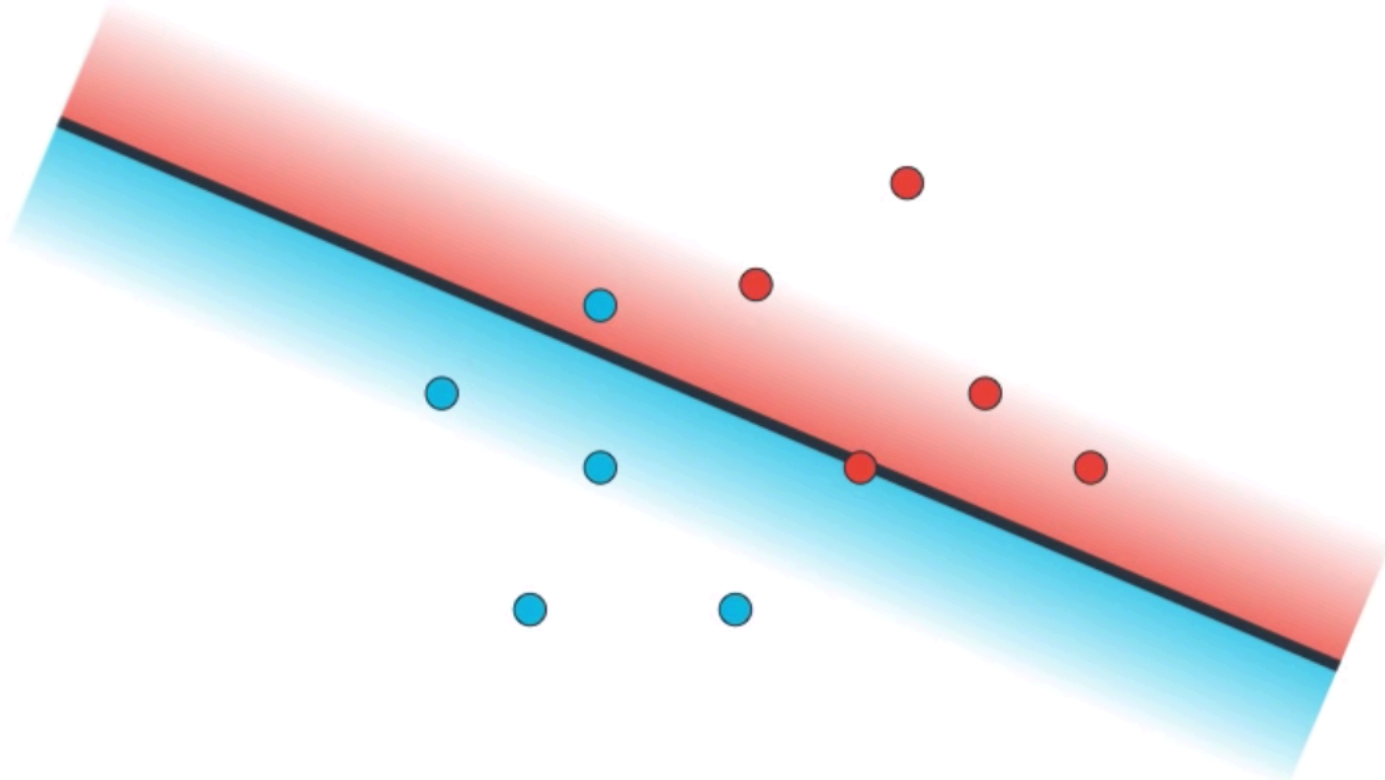
Perceptron Algorithm



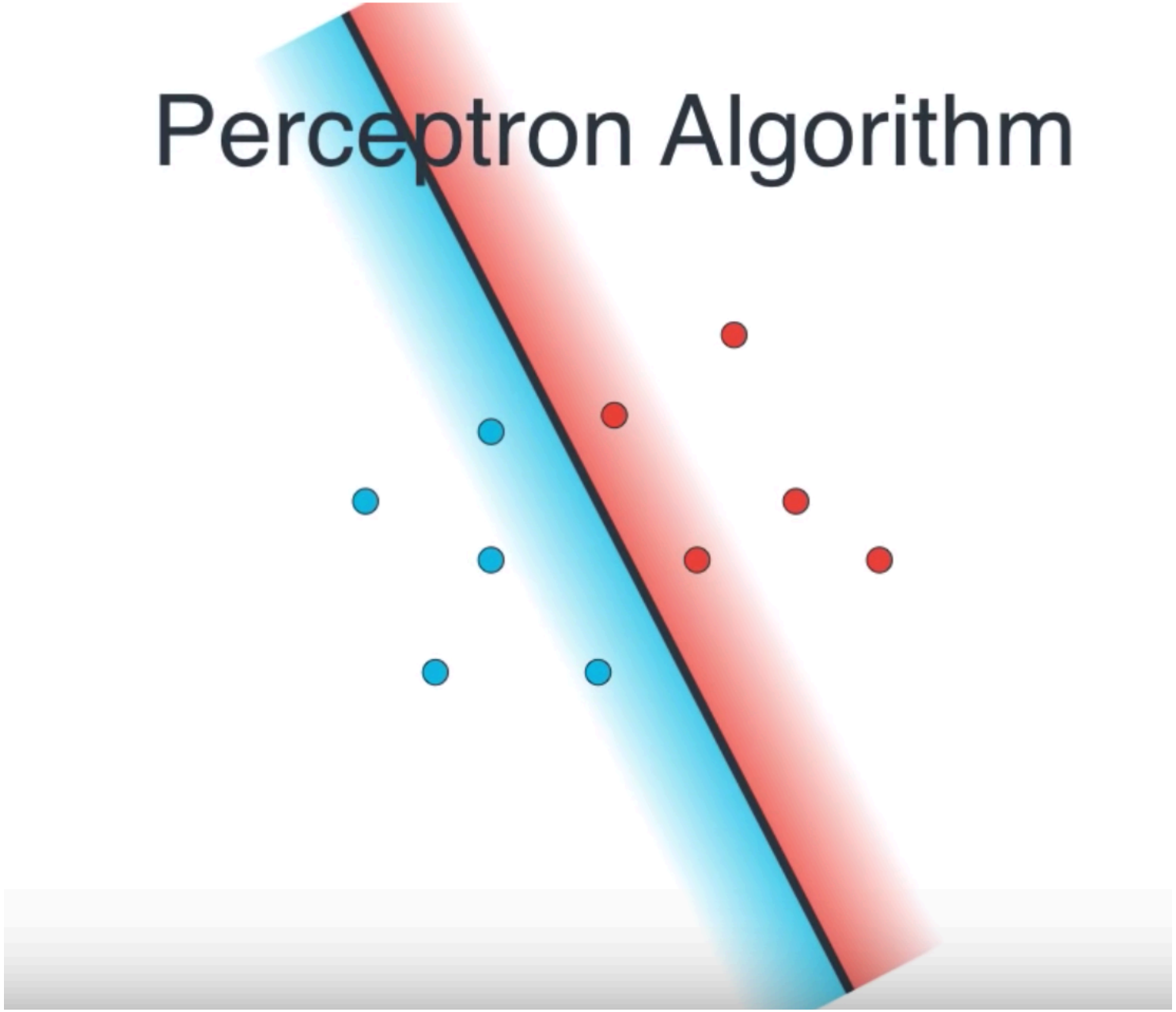
Perceptron Algorithm



Perceptron Algorithm

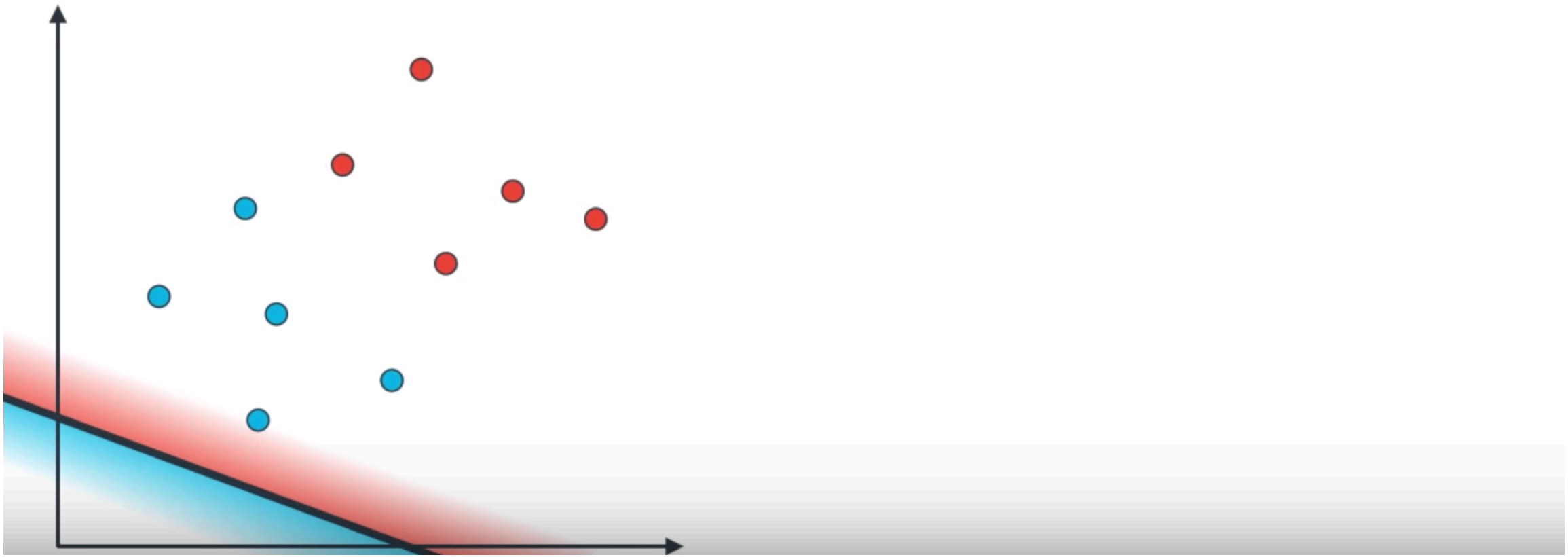


Perceptron Algorithm



Perceptron algorithm

Step 1: Start with a random line with blue and red sides.



Perceptron algorithm

Step 1: Start with a random line with blue and red sides.

Step 2: Pick a large number. (number of repetitions, or epochs)

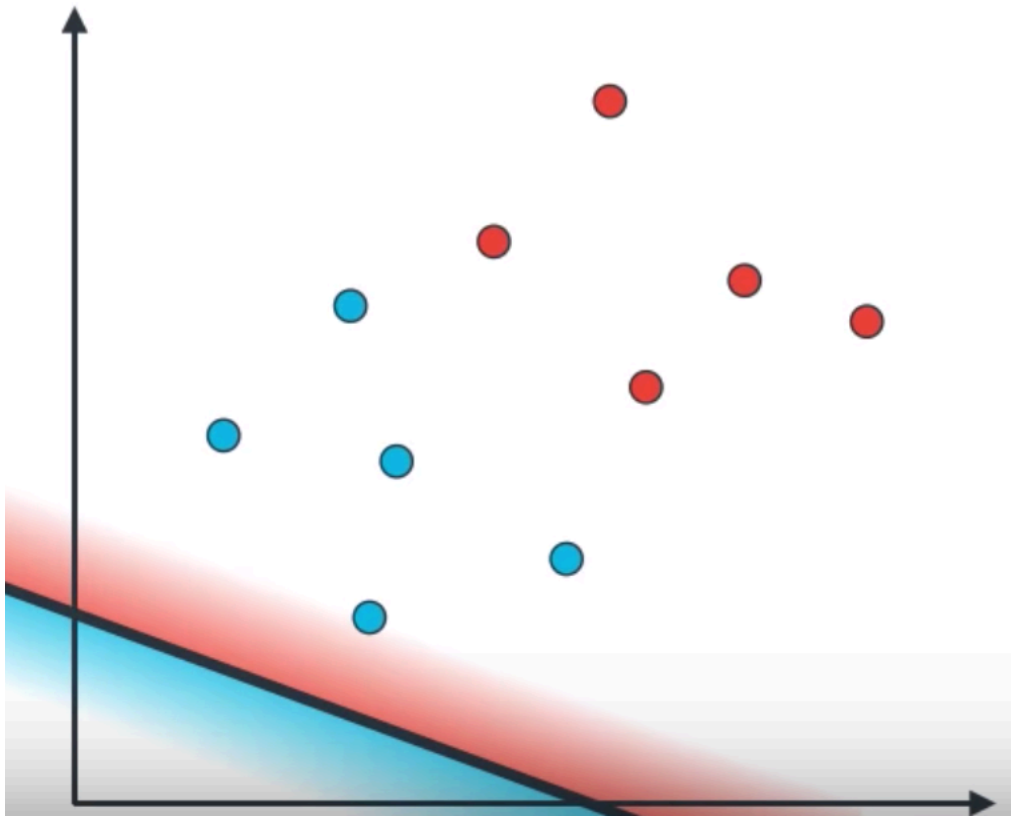
Perceptron algorithm

Step 1: Start with a random line with blue and red sides.

Step 2: Pick a large number. 1000 (number of repetitions, or epochs)

Step 3: (repeat 1000 times)

Perceptron algorithm



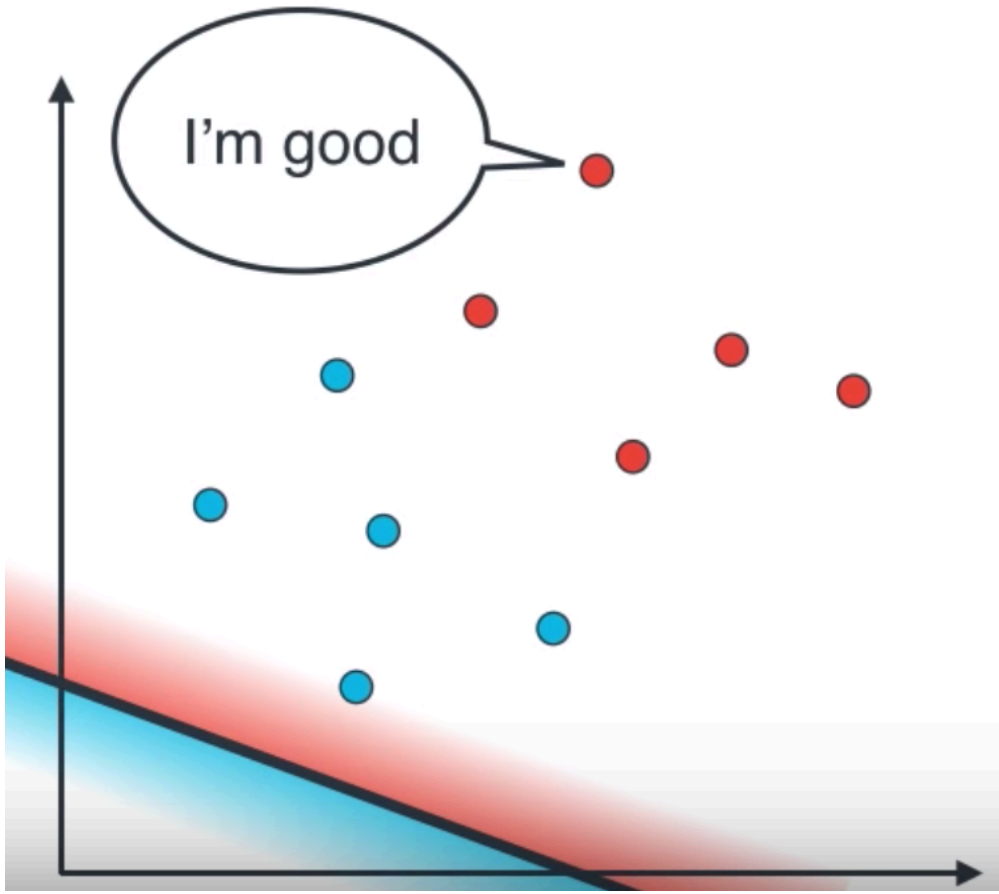
Step 1: Start with a random line with blue and red sides.

Step 2: Pick a large number. 1000 (number of repetitions, or epochs)

Step 3: (repeat 1000 times)

- Pick random point
- If point is correctly classified:

Perceptron algorithm



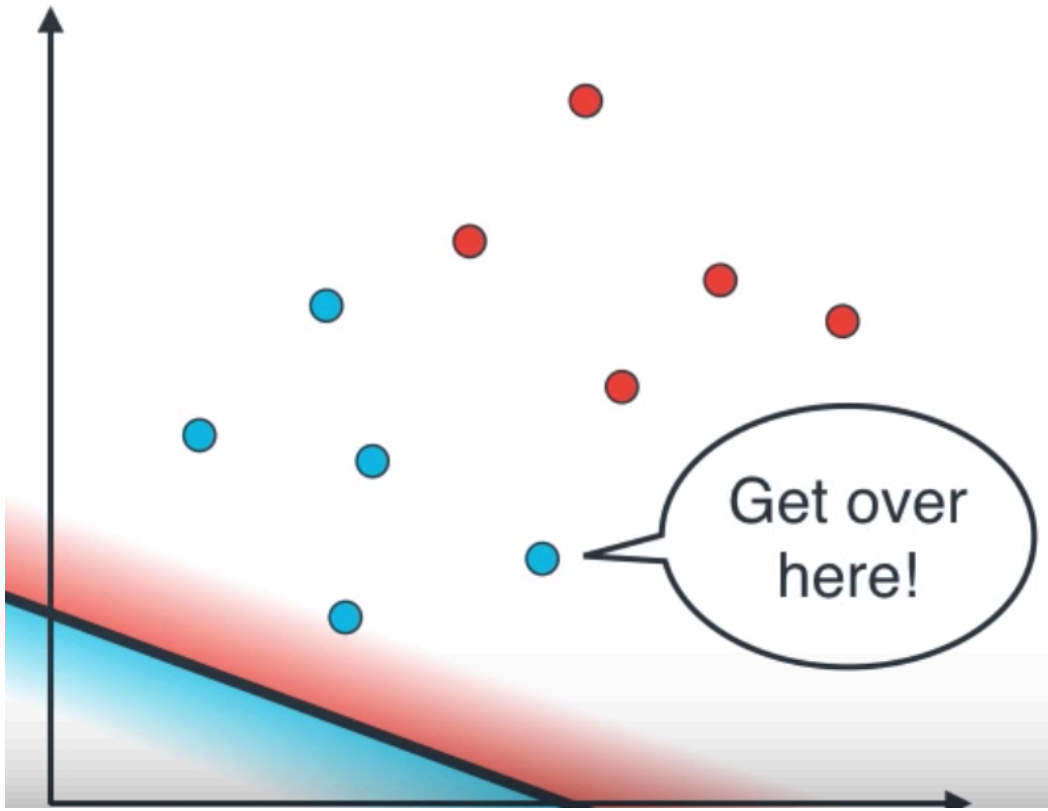
Step 1: Start with a random line with blue and red sides.

Step 2: Pick a large number. 1000 (number of repetitions, or epochs)

Step 3: (repeat 1000 times)

- Pick random point
- If point is correctly classified:
  - Do nothing

Perceptron algorithm



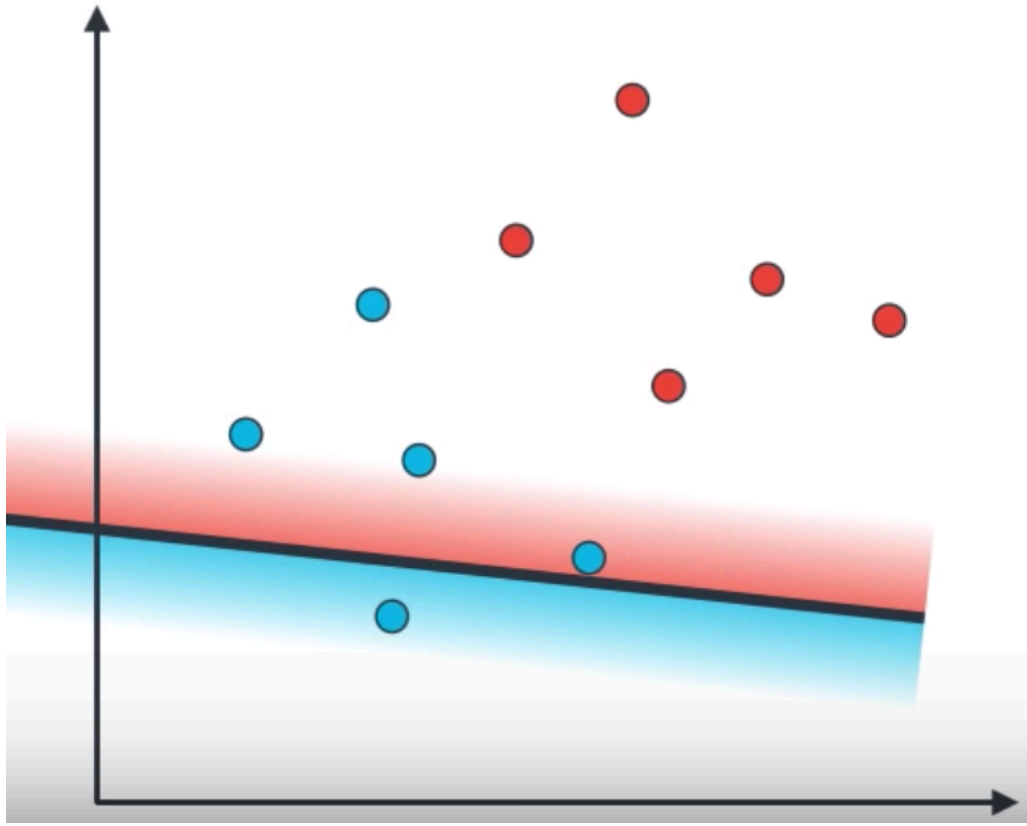
Step 1: Start with a random line with blue and red sides.

Step 2: Pick a large number. 1000 (number of repetitions, or epochs)

Step 3: (repeat 1000 times)

- Pick random point
- If point is correctly classified:
  - Do nothing
- If point is incorrectly classified
  - Move line towards point

Perceptron algorithm



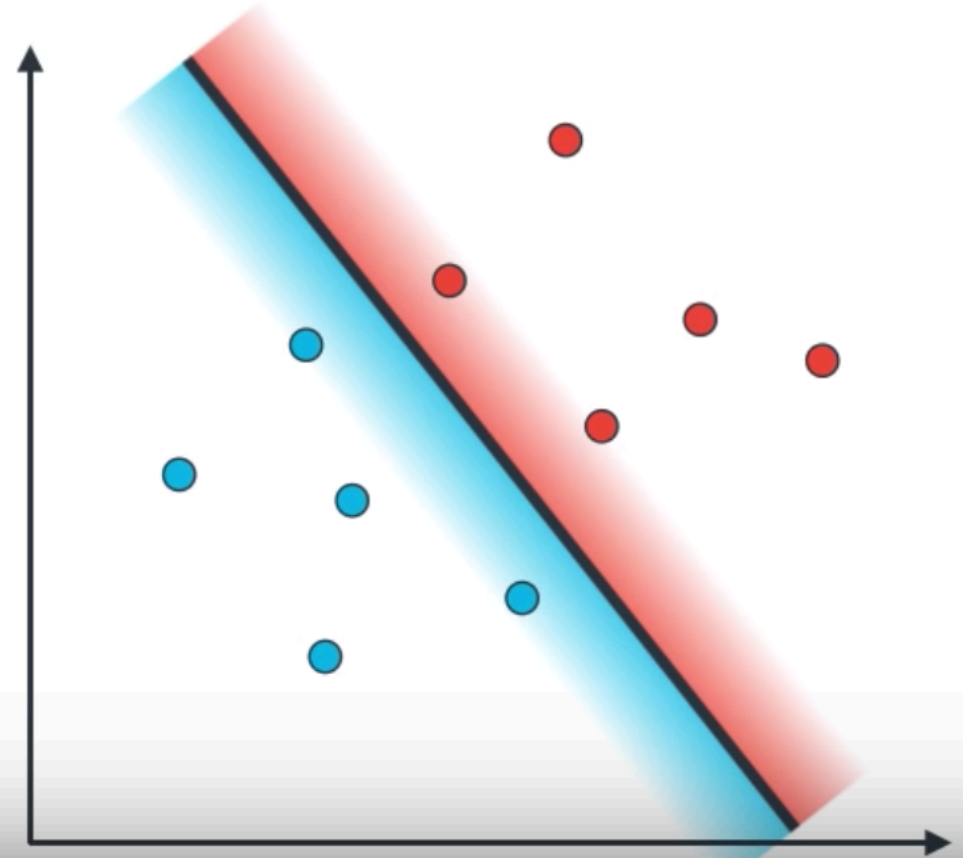
Step 1: Start with a random line with blue and red sides.

Step 2: Pick a large number. 1000 (number of repetitions, or epochs)

Step 3: (repeat 1000 times)

- Pick random point
- If point is correctly classified:
  - Do nothing
- If point is incorrectly classified
  - Move line towards point

Perceptron algorithm



Step 1: Start with a random line with blue and red sides.

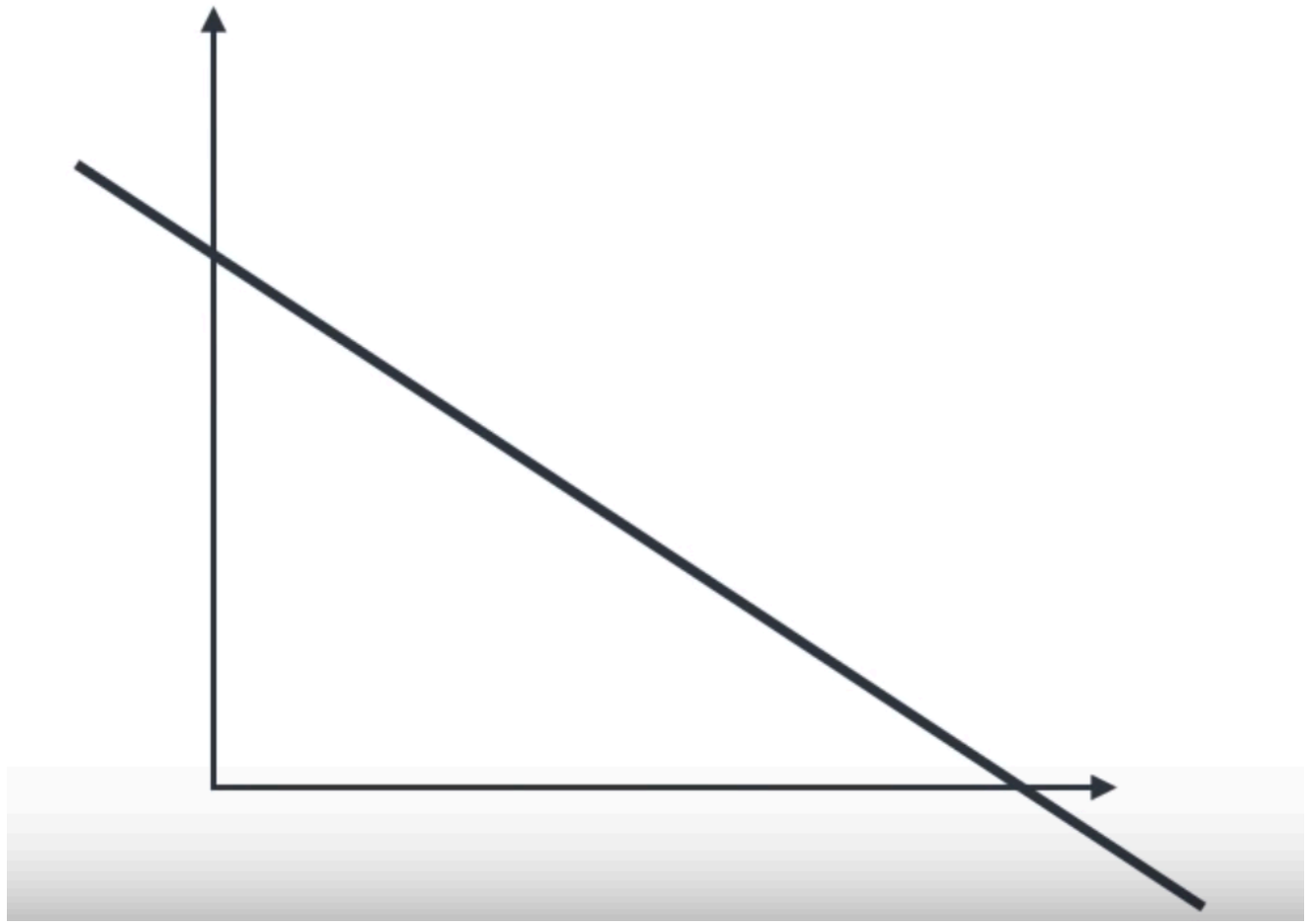
Step 2: Pick a large number. 1000 (number of repetitions, or epochs)

Step 3: (repeat 1000 times)

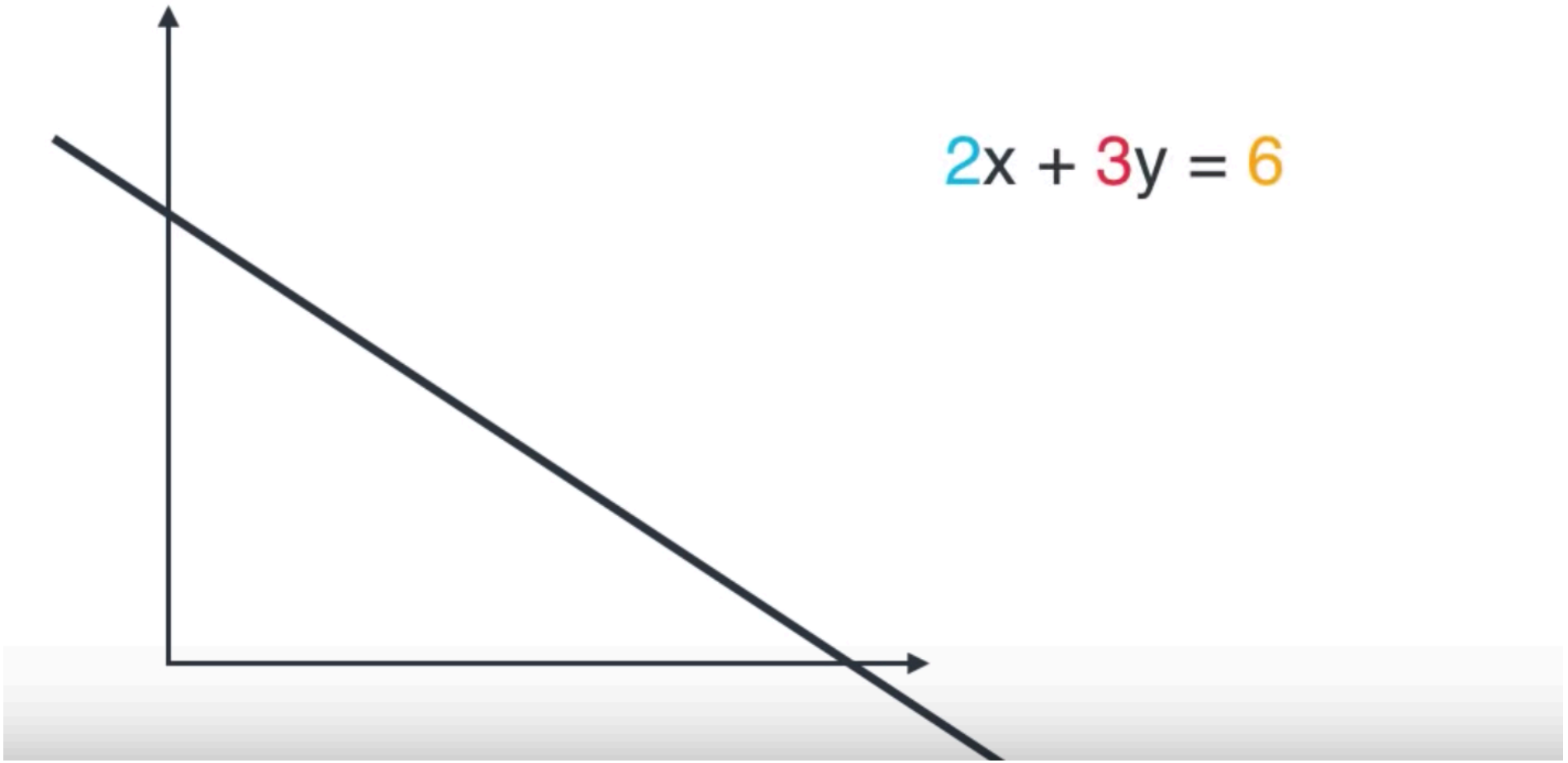
- Pick random point
- If point is correctly classified:
  - Do nothing
- If point is incorrectly classified
  - Move line towards point

Step 4: Enjoy your line that separates the data!

How to move a line

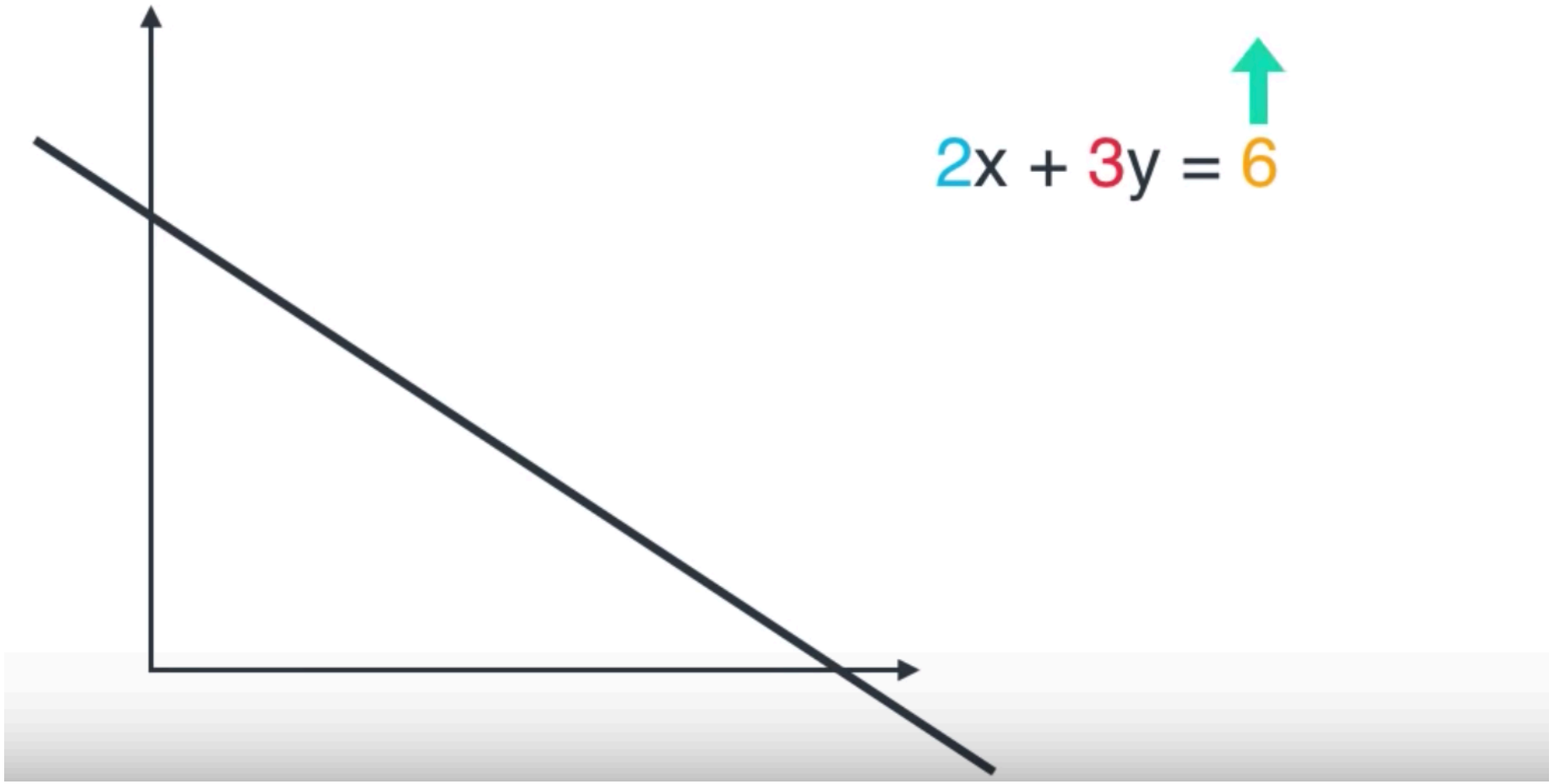


Rotating and translating

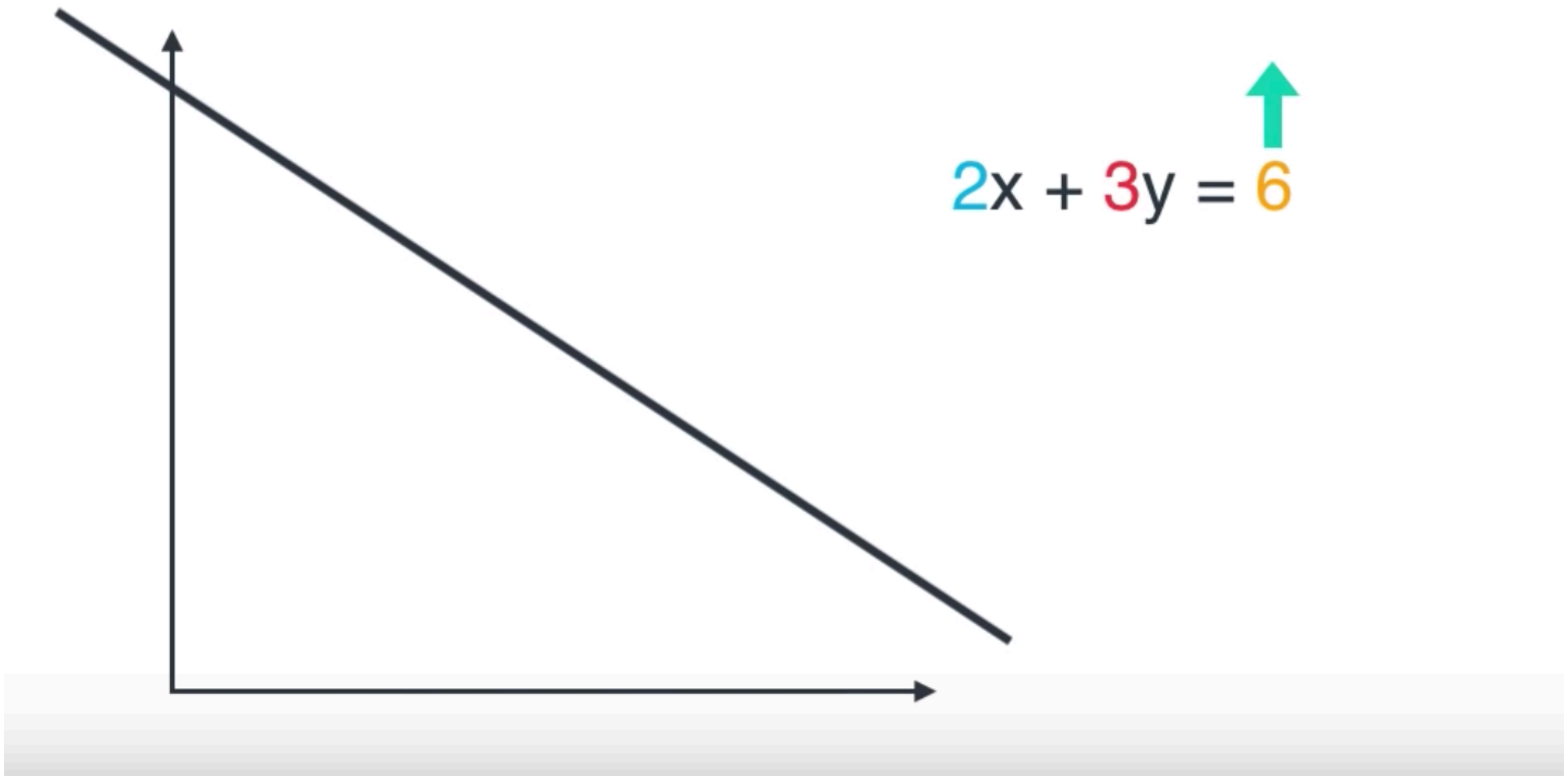


$$2x + 3y = 6$$

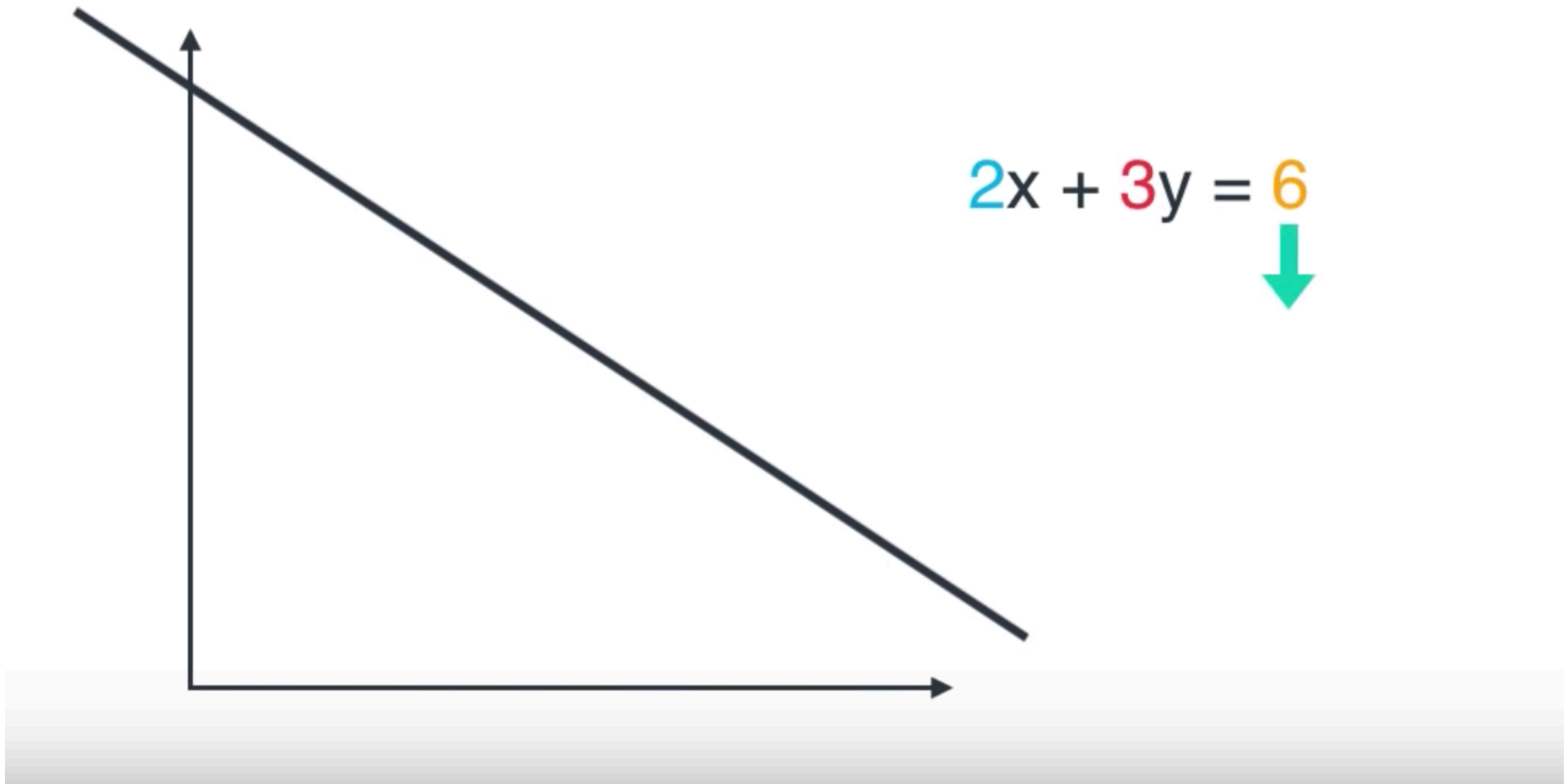
Rotating and translating



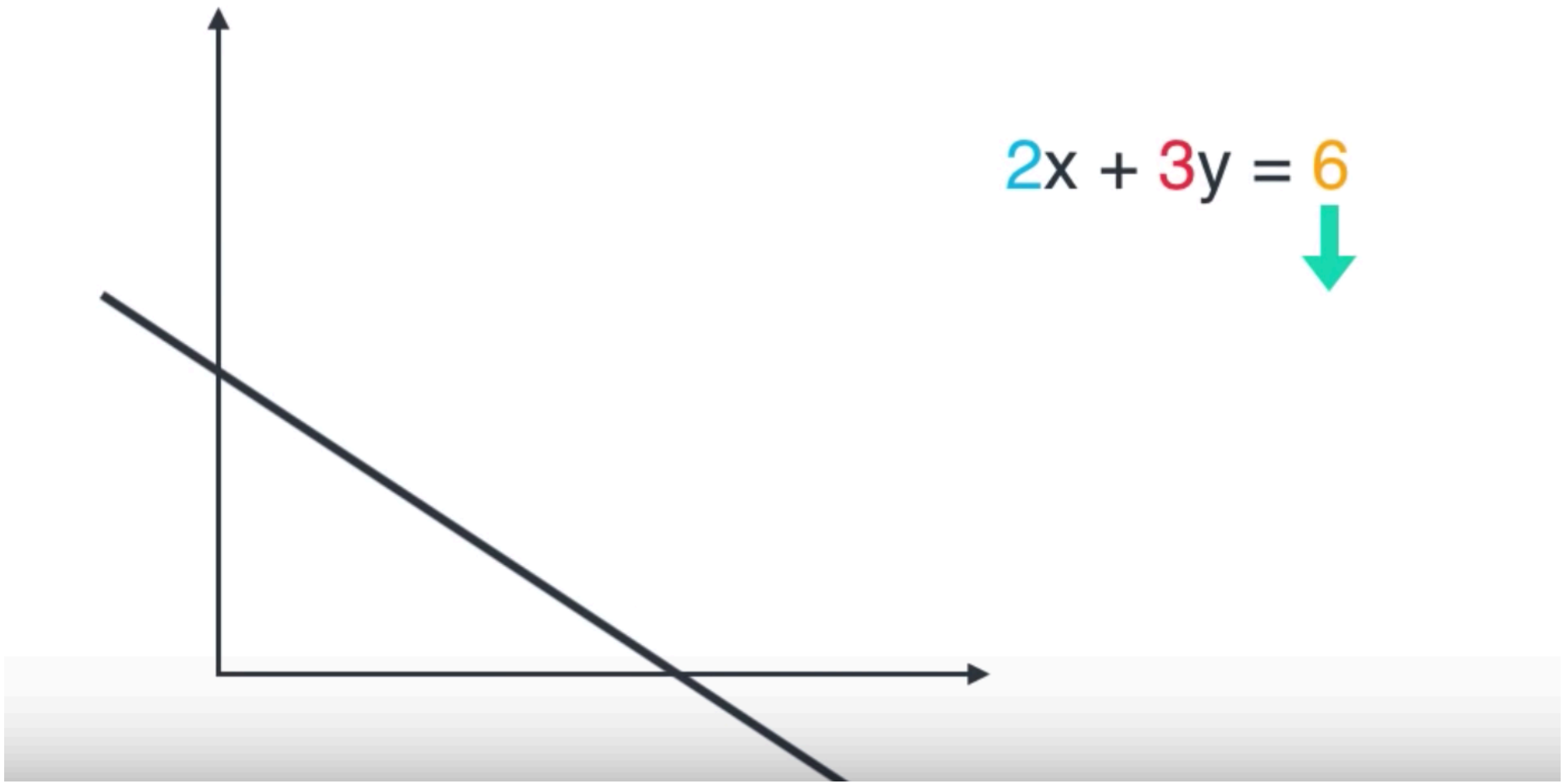
Rotating and translating



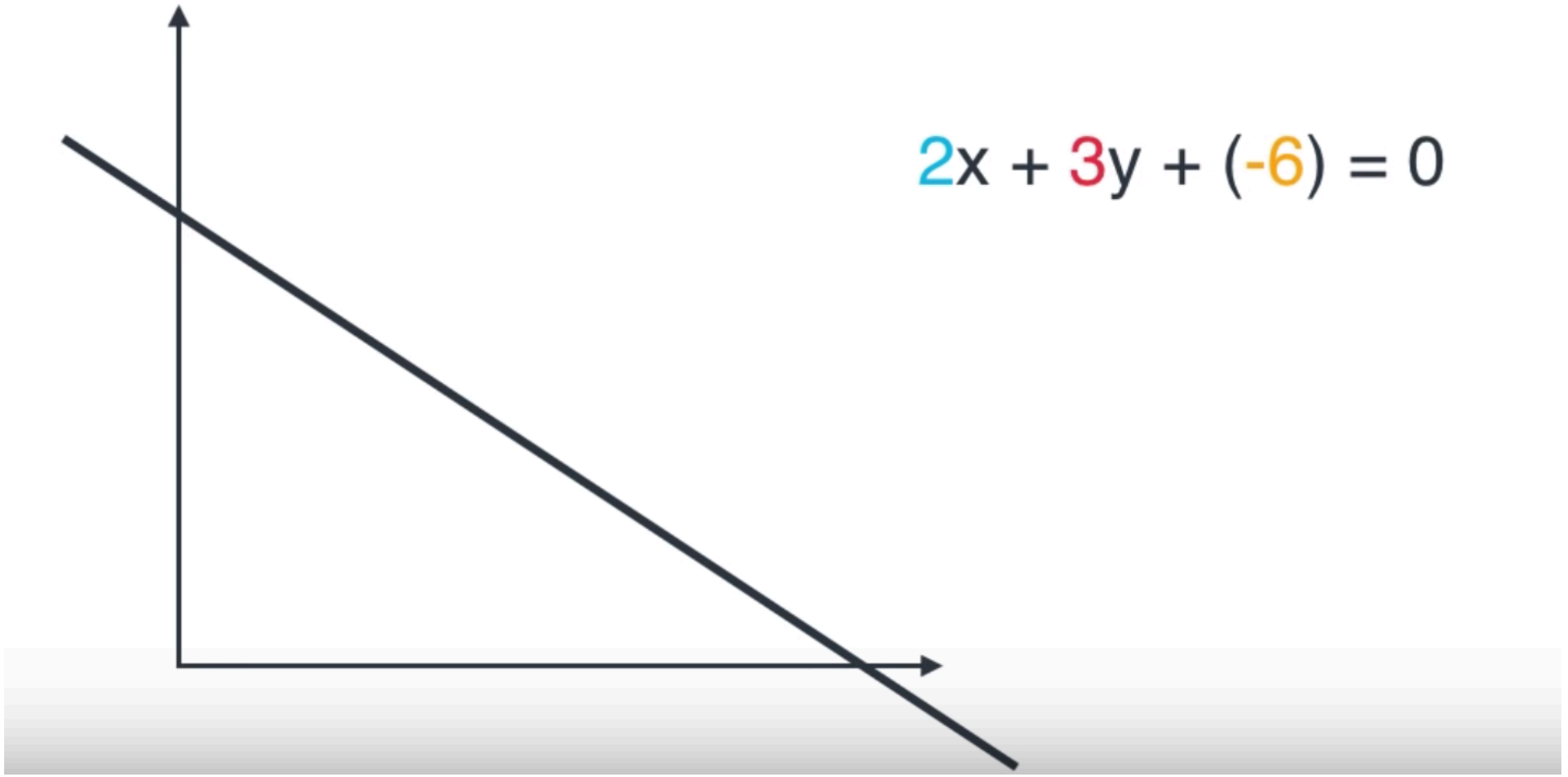
Rotating and translating



Rotating and translating

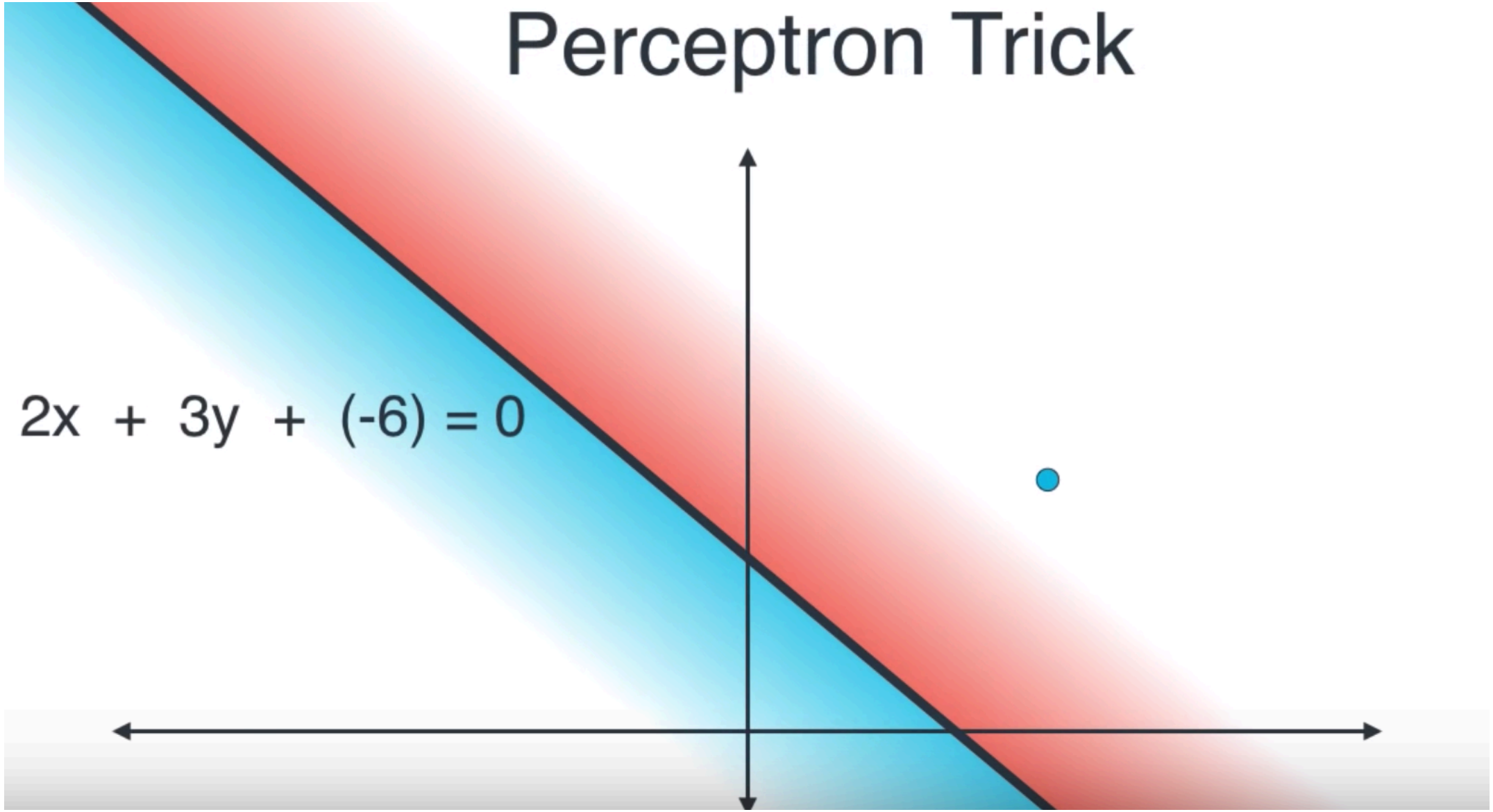


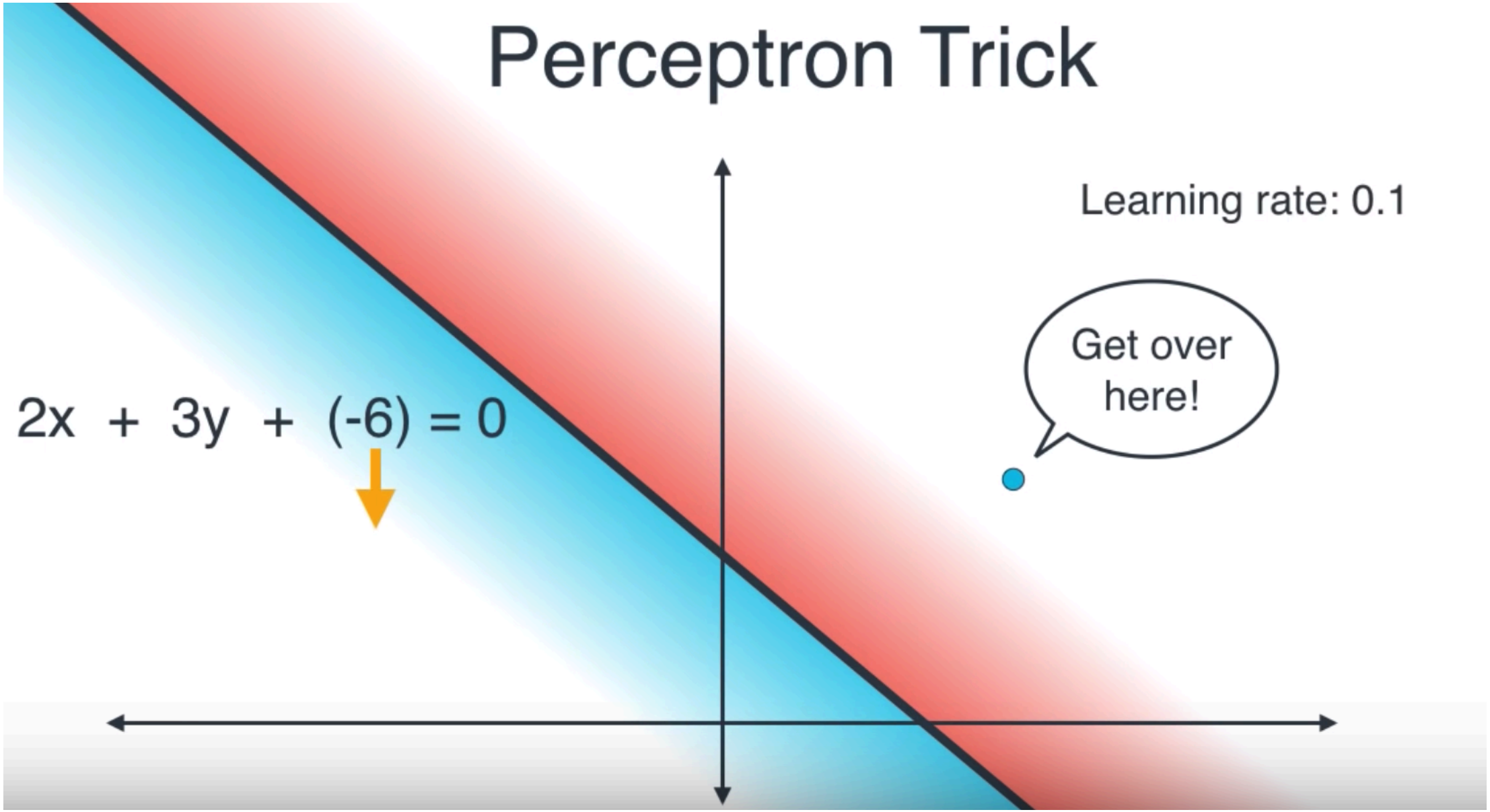
Rotating and translating

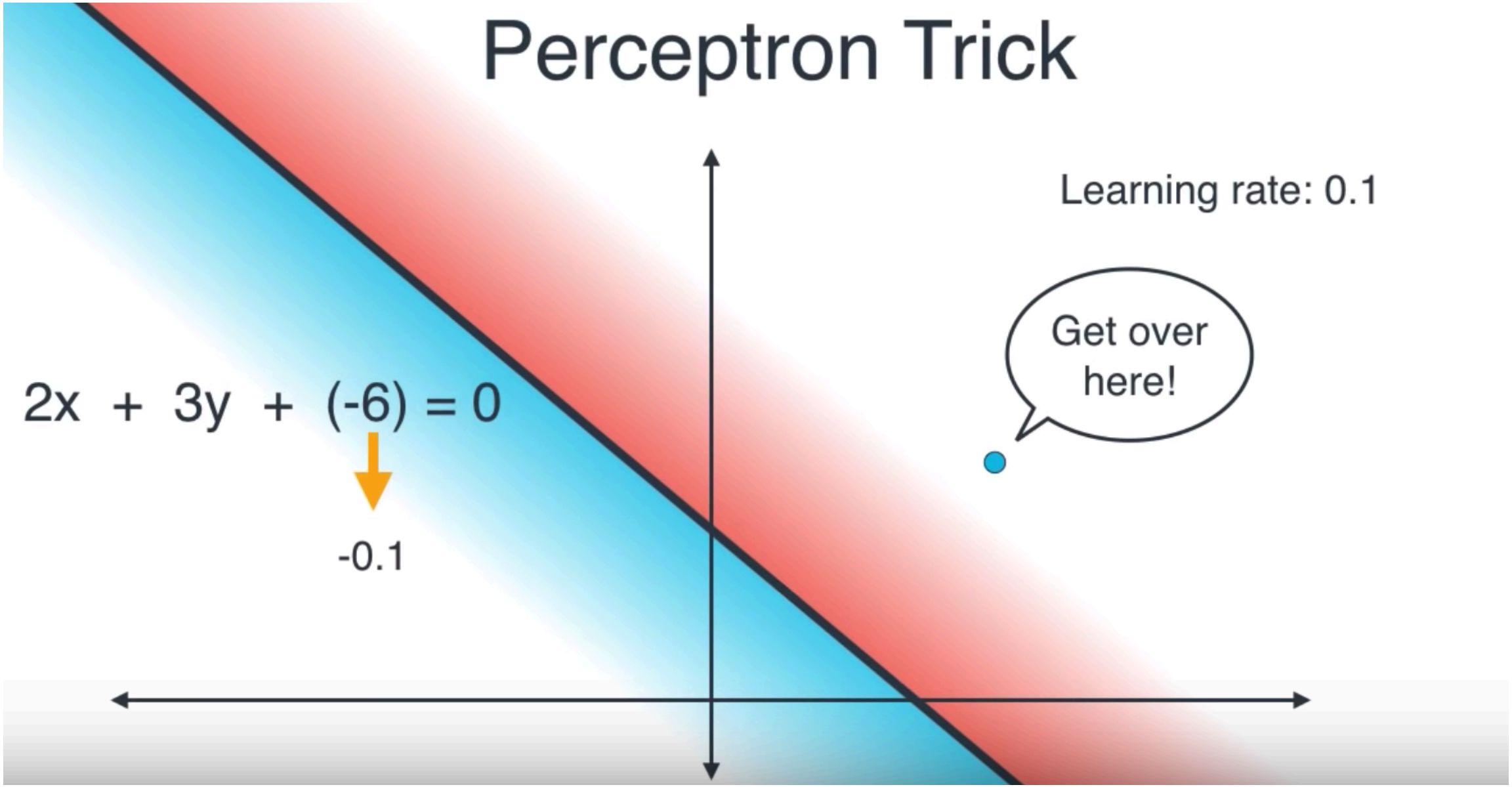


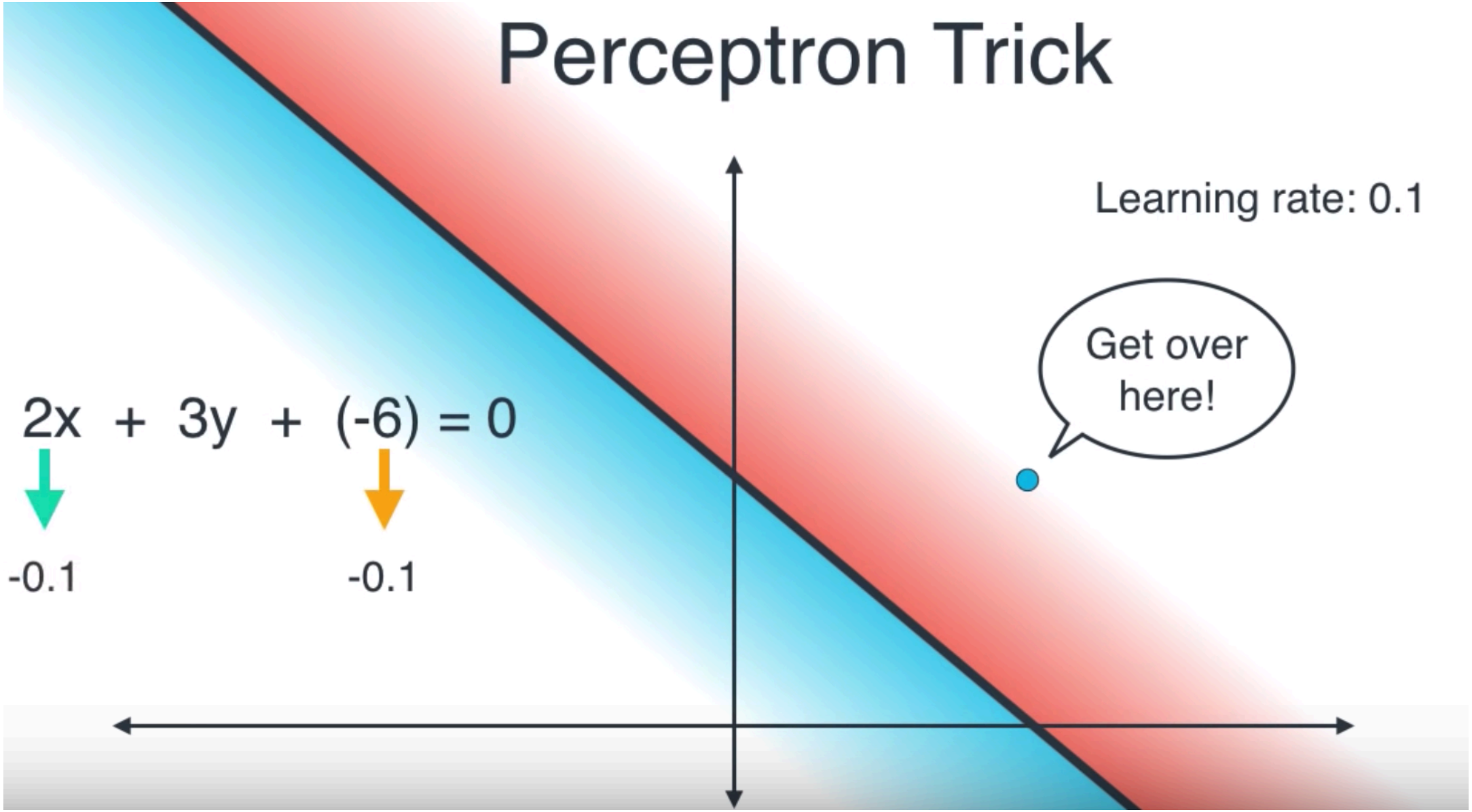
$$2x + 3y + (-6) = 0$$

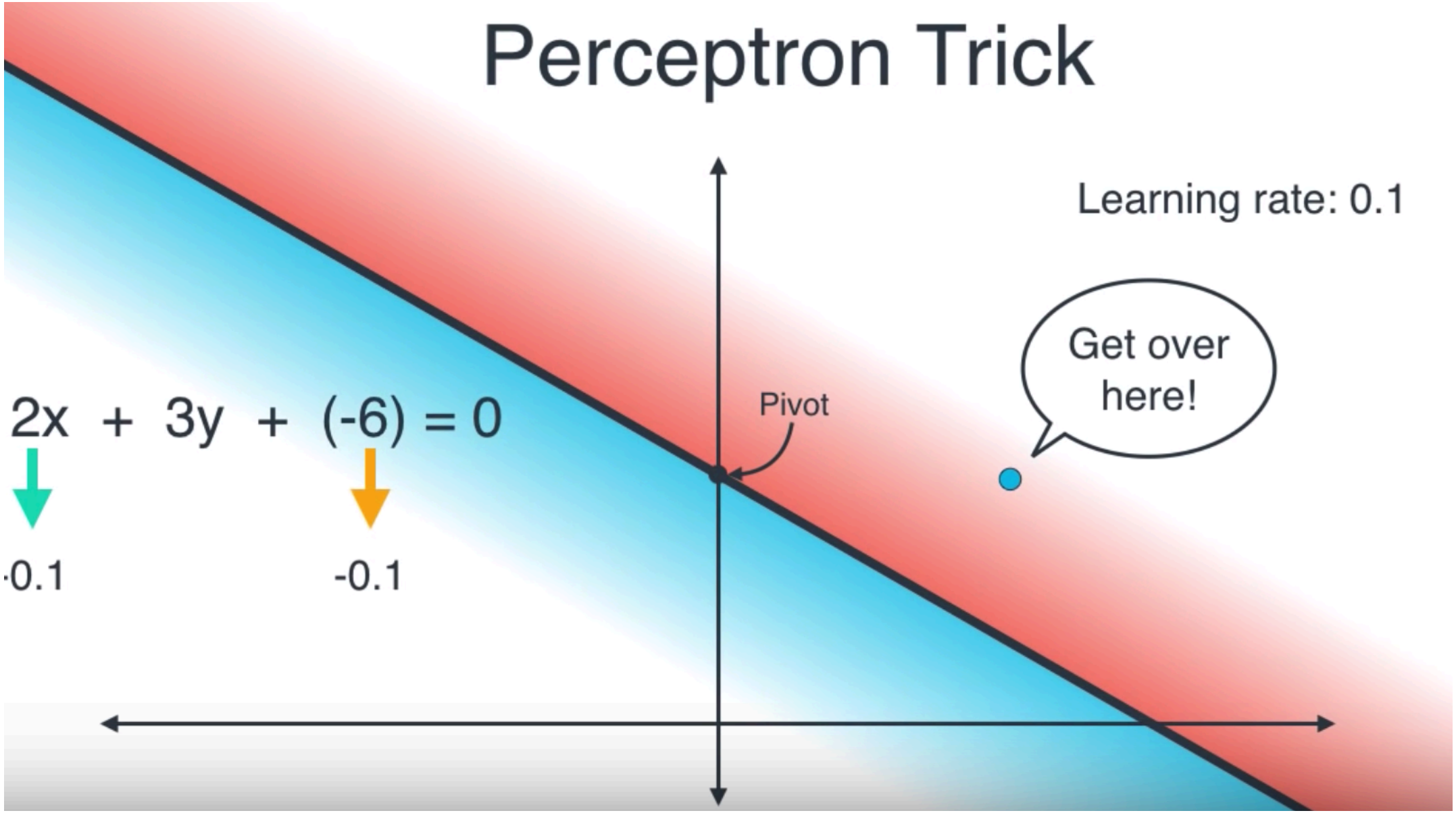
Perceptron Trick

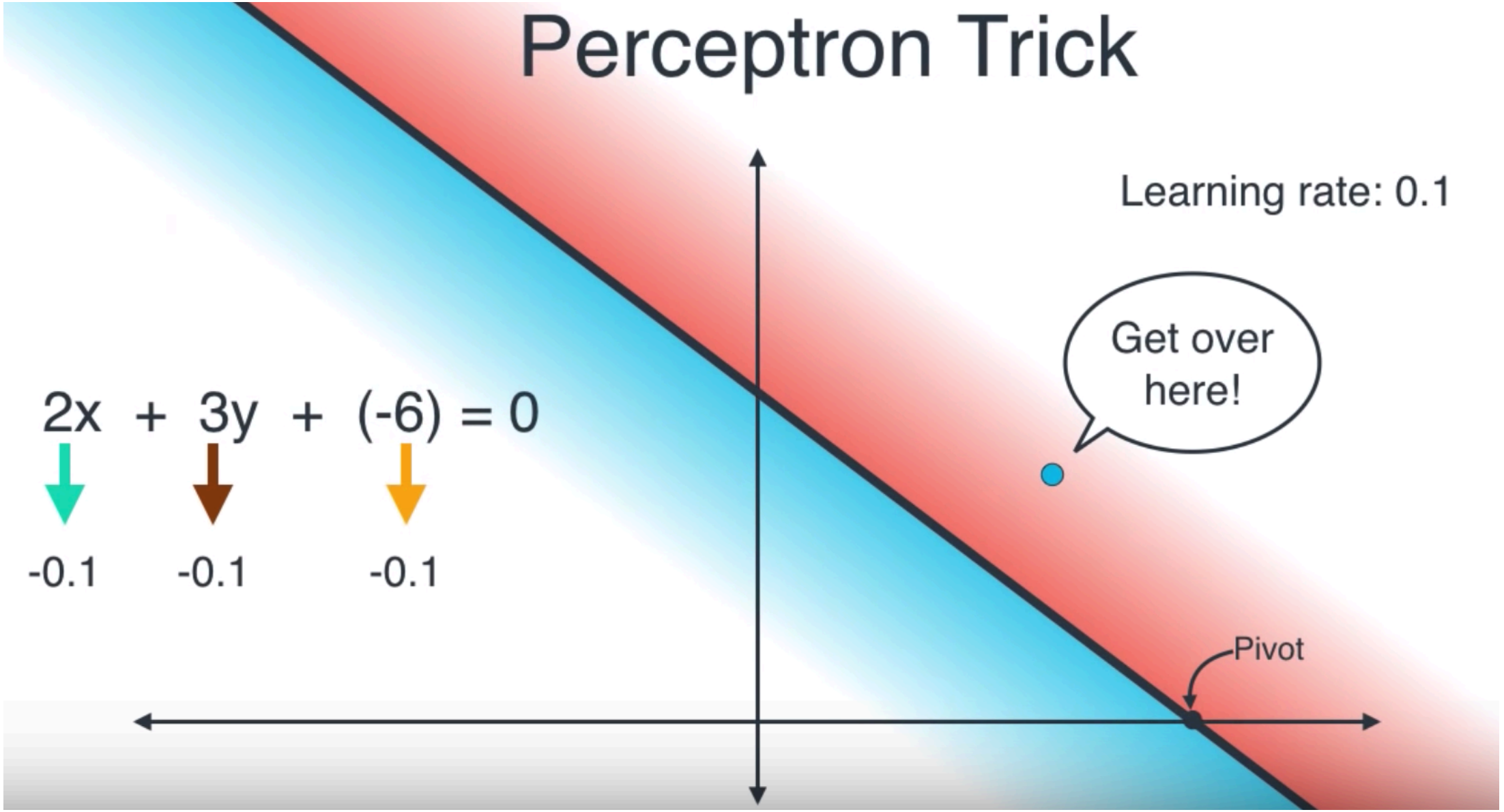



Perceptron Trick
Learning rate: 0.1
Get over here!
2x + 3y + (-6) = 0

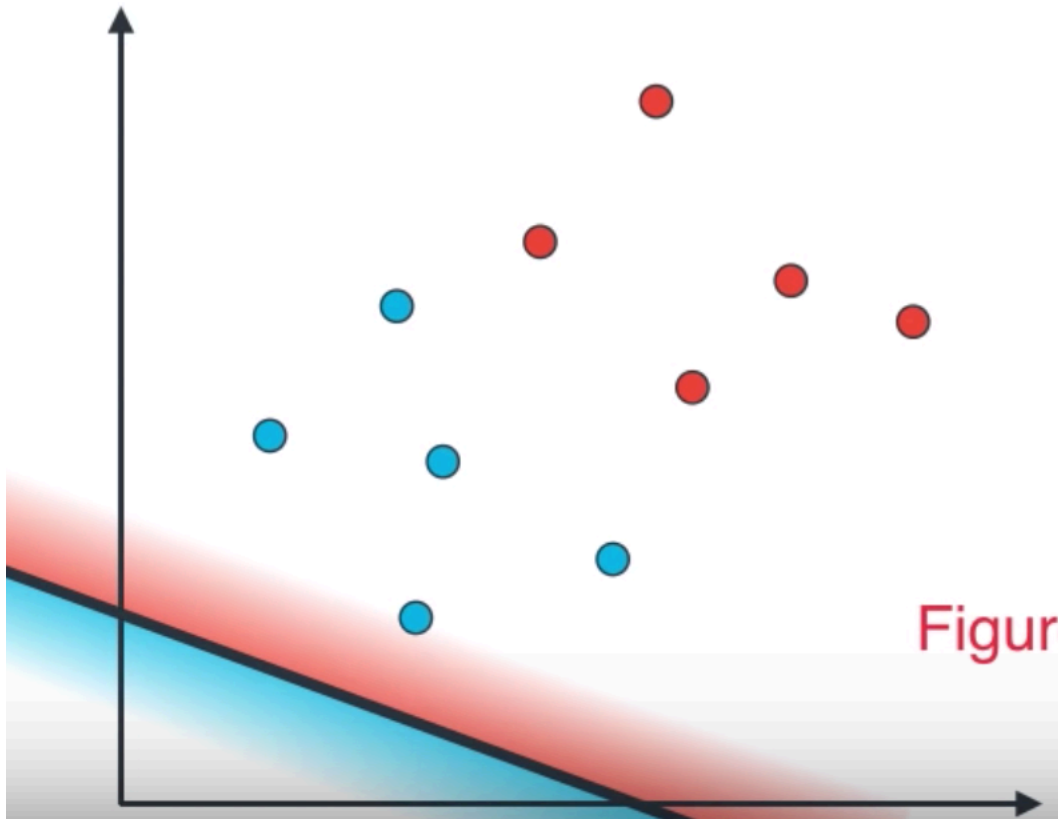

Perceptron Trick
Learning rate: 0.1
Get over here!
2x + 3y + (-6) = 0
-0.1


Perceptron Trick
Learning rate: 0.1
Get over here!
2x + 3y + (-6) = 0
-0.1
-0.1


Perceptron Trick
Learning rate: 0.1
Get over here!
Pivot
2x + 3y + (-6) = 0
-0.1
-0.1


Perceptron Trick
Learning rate: 0.1
Get over here!
2x + 3y + (-6) = 0
-0.1
-0.1
-0.1
Pivot

Algorithm



Step 1: Start with a random line of equation $ax + by + c = 0$

Step 2: Pick a large number. 1000 (number of repetitions, or epochs)

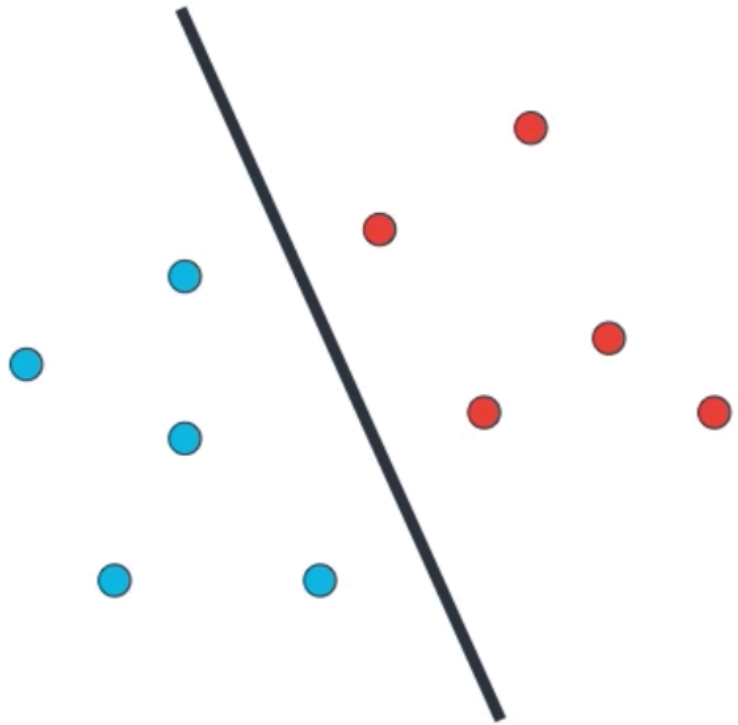
Step 3: Pick a small number. 0.01 (learning rate)

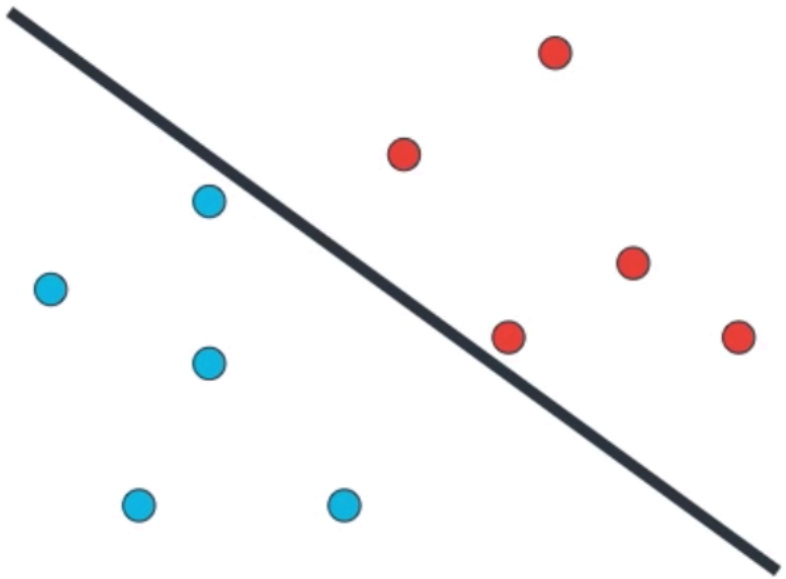
Step 4: (repeat 1000 times)

- Pick random point
- If point is correctly classified
  - Do nothing
- If point is incorrectly classified
  - **Add ± 0.01 to a**
  - **Add ± 0.01 to b**
  - **Add ± 0.01 to c**

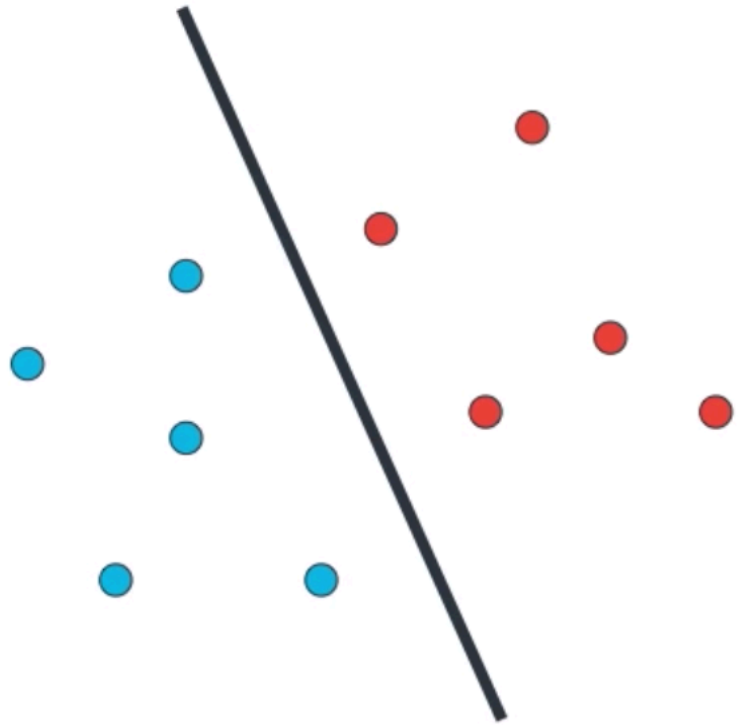
Figure out!

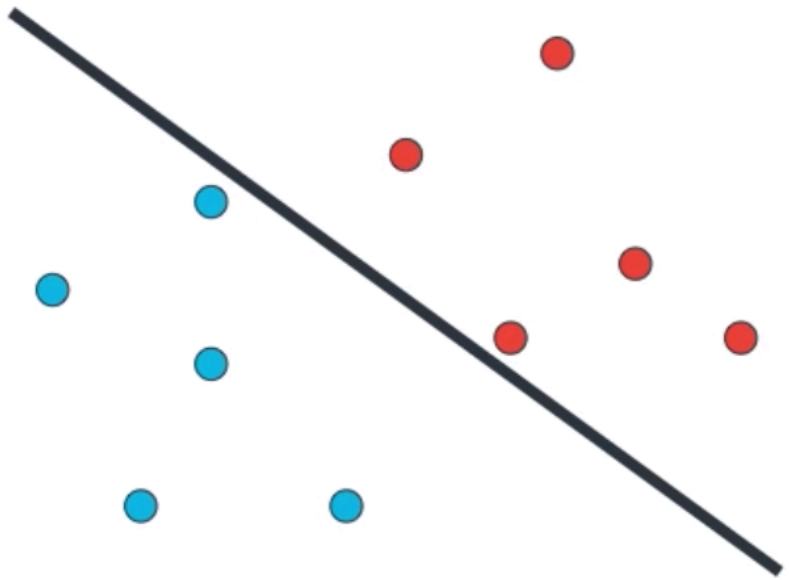
Which line is better?



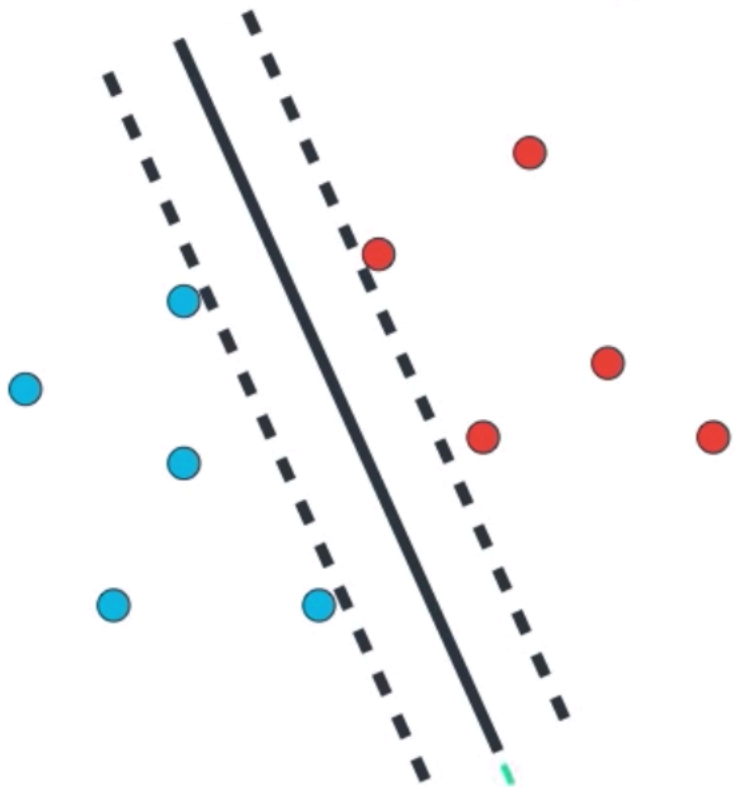


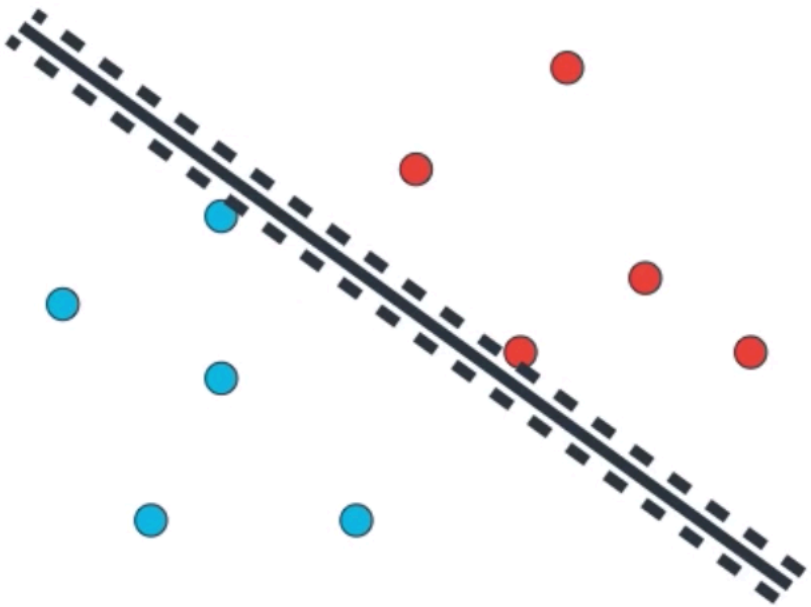
Which line is better?



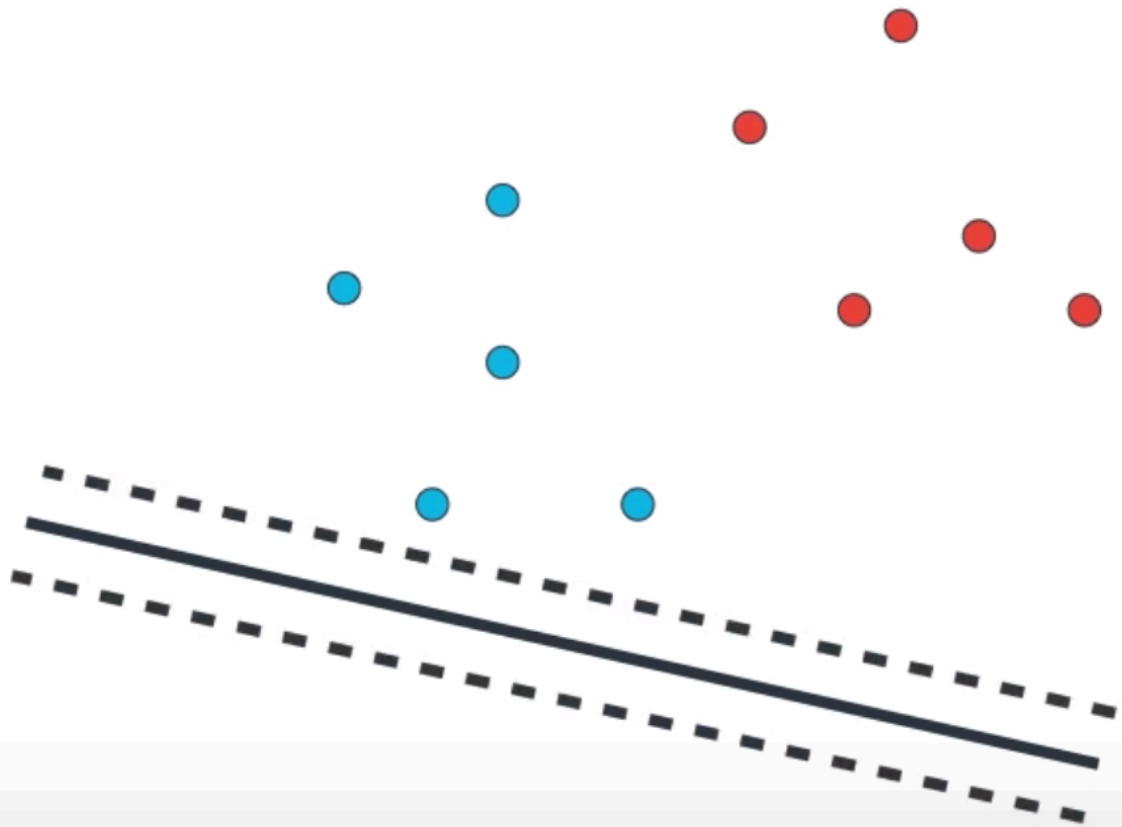


Which line is better?

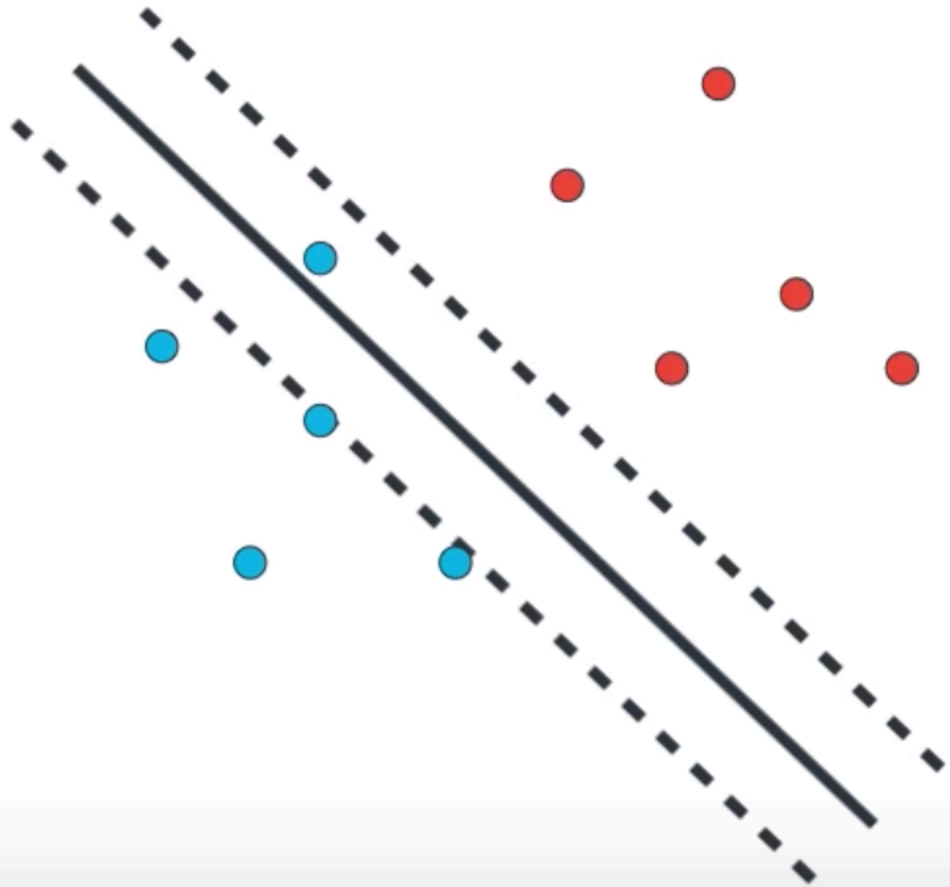




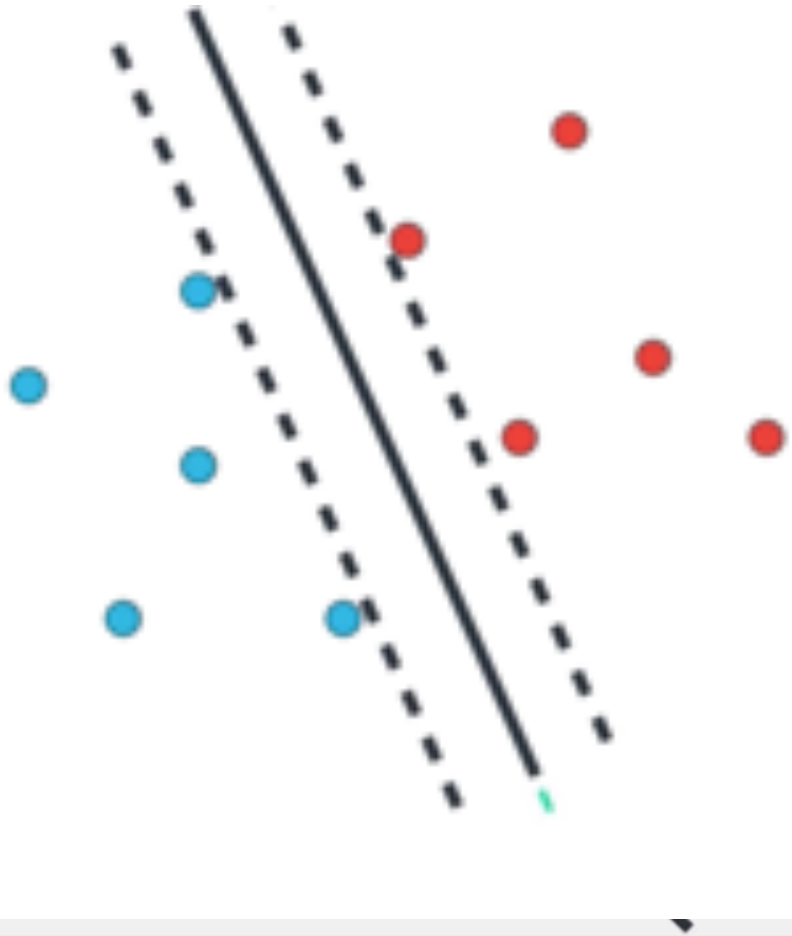
Split data - separate lines



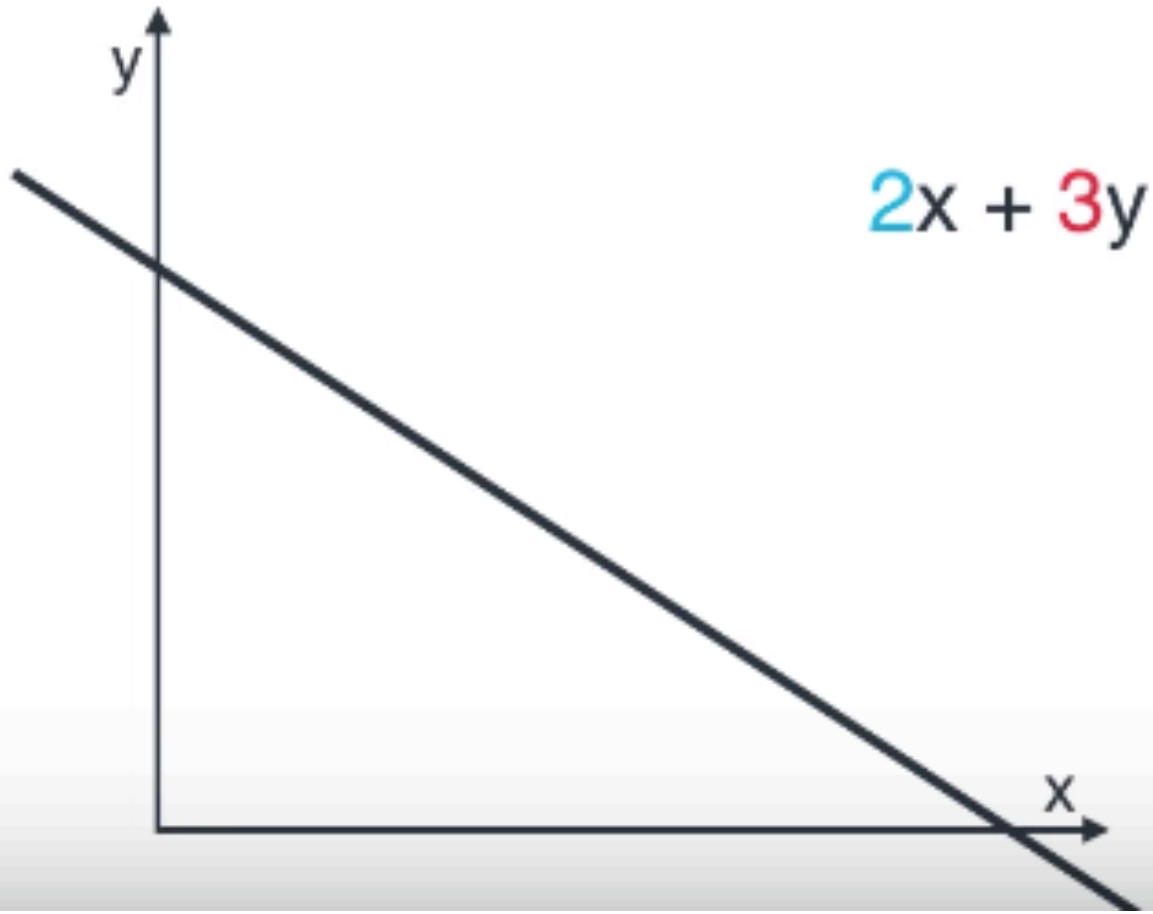
Split data - separate lines



Split data - separate lines

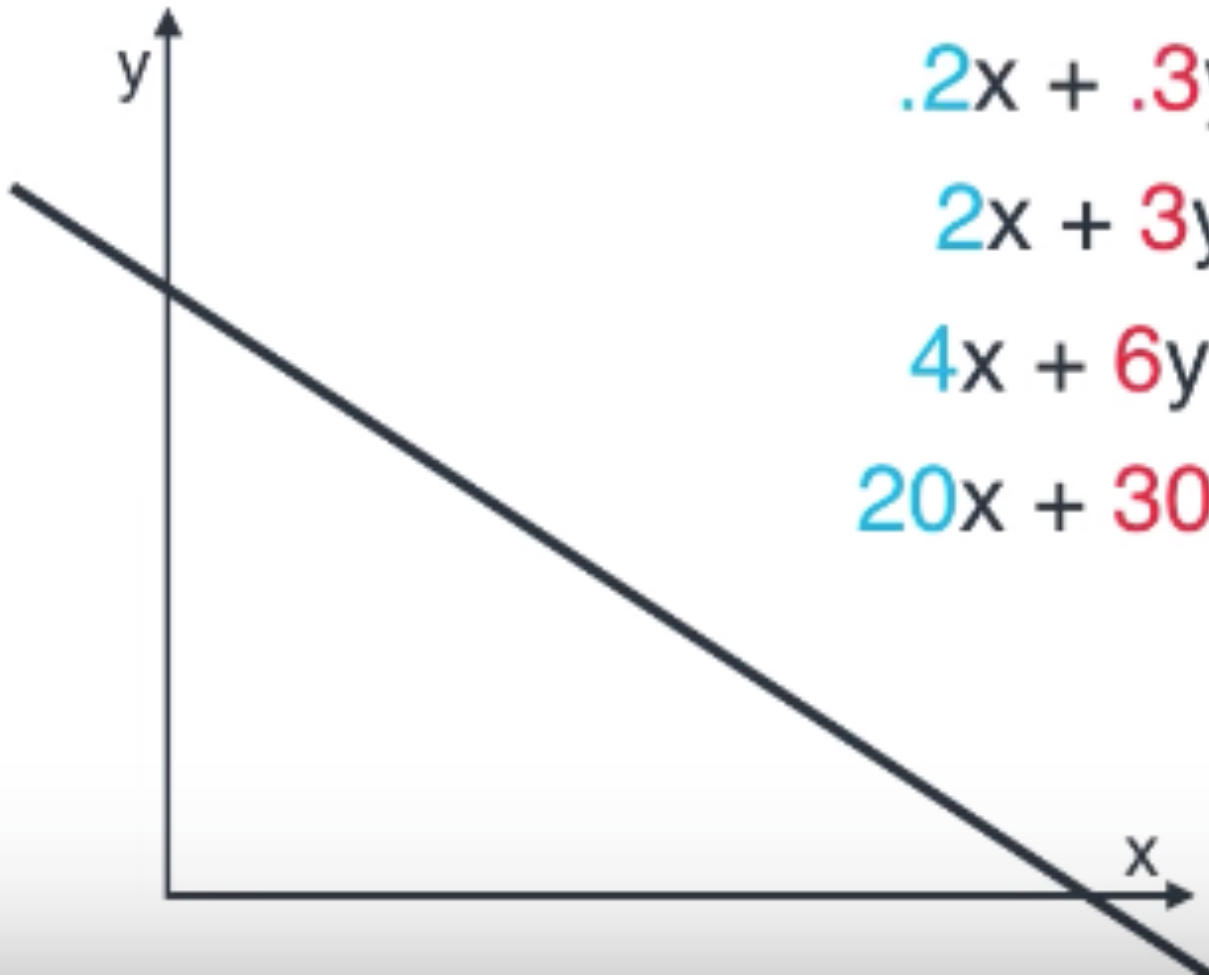


How to separate lines?



$$2x + 3y + (-6) = 0$$

How to separate lines?

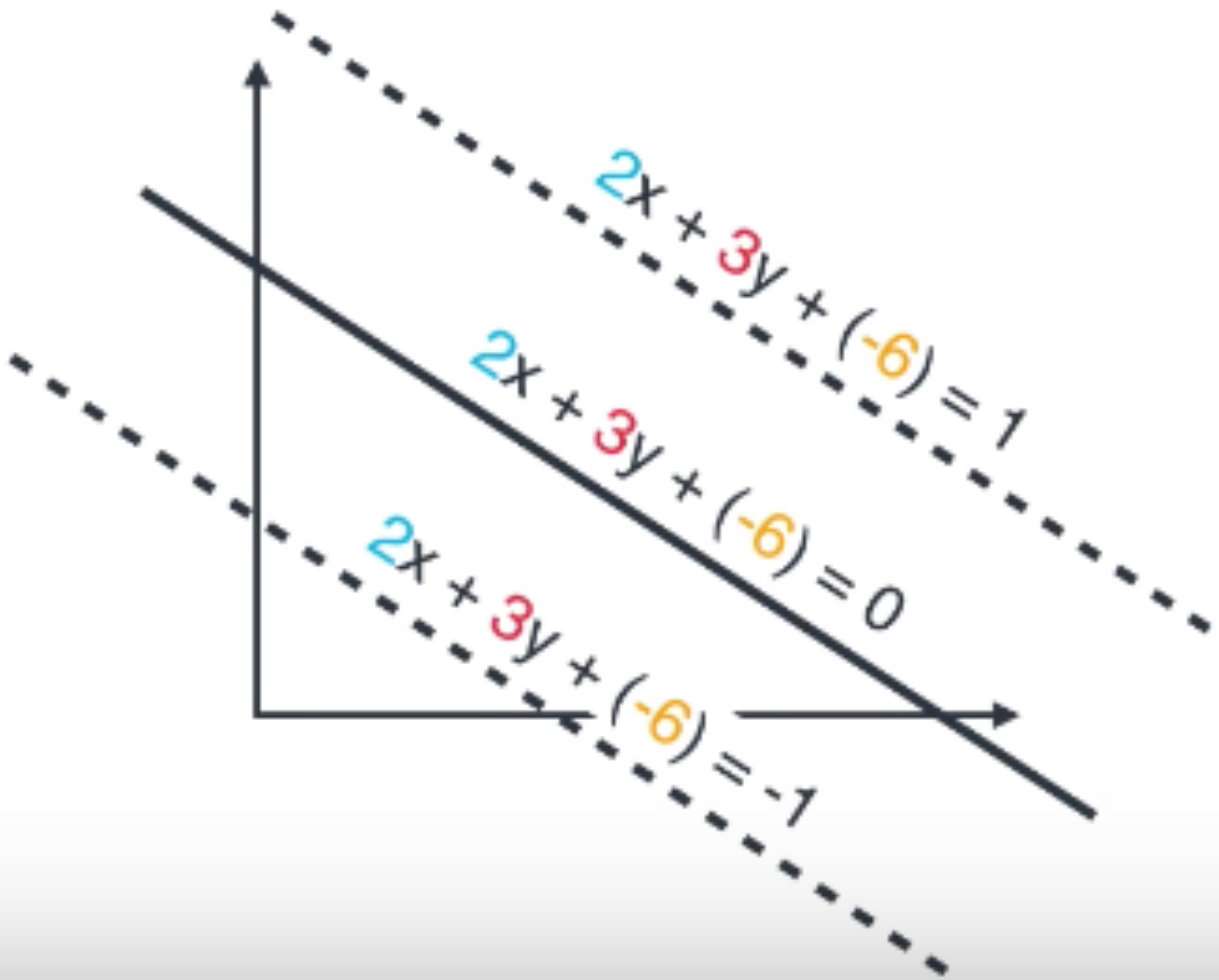


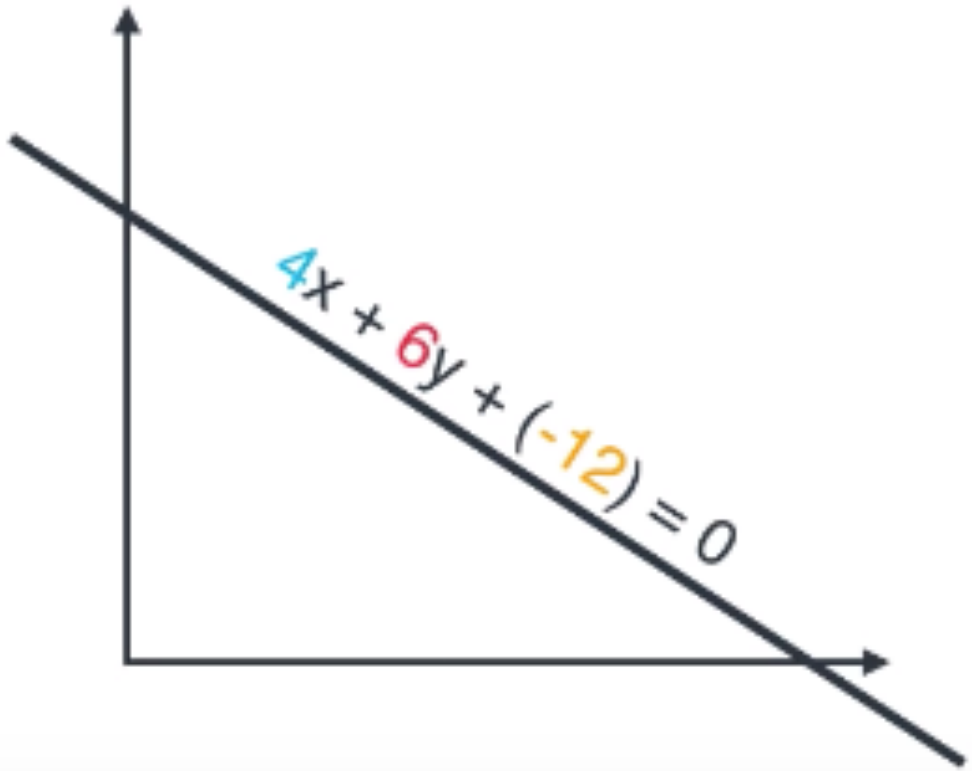
$$.2x + .3y + (-.6) = 0$$

$$2x + 3y + (-6) = 0$$

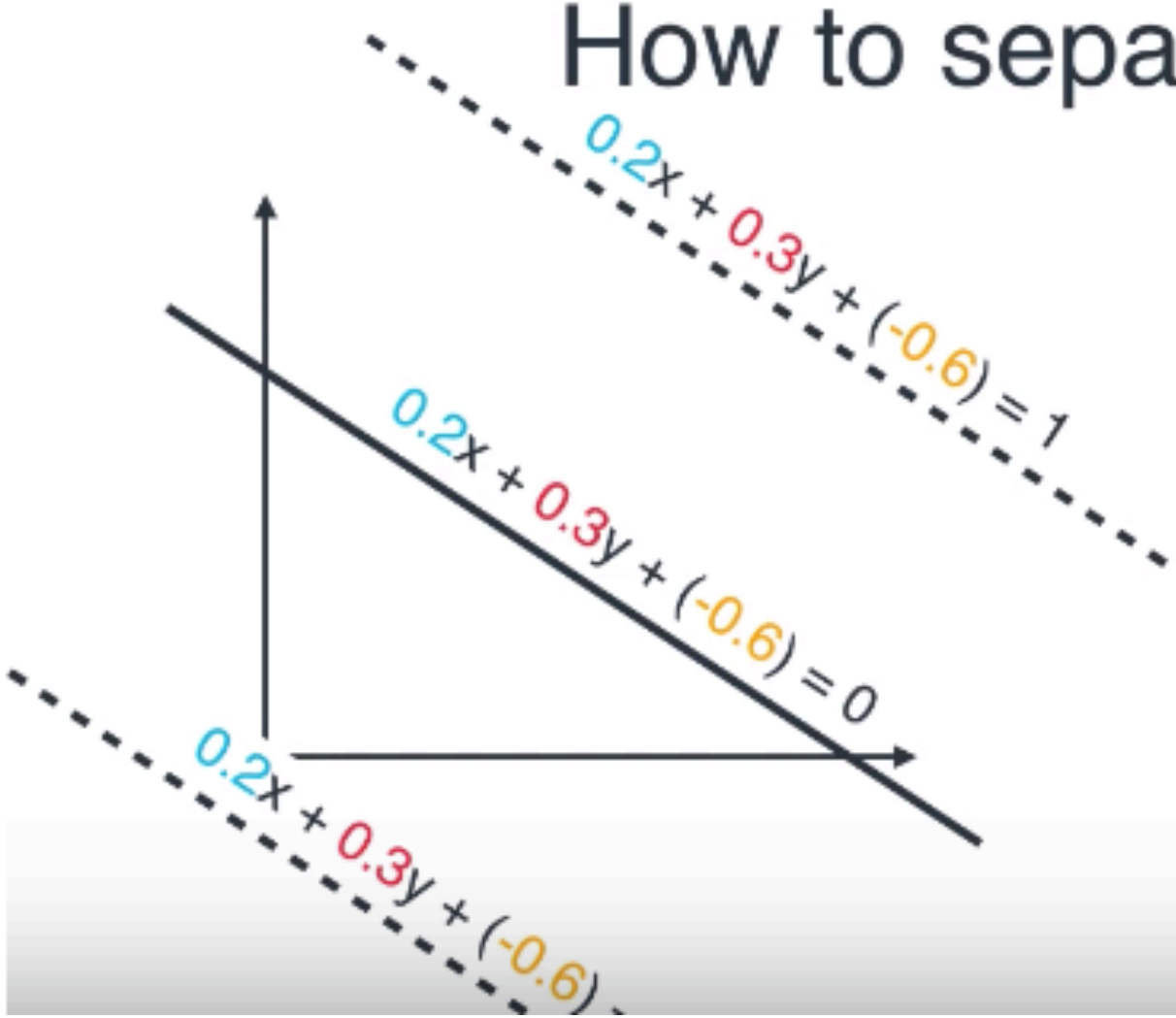
$$4x + 6y + (-12) = 0$$

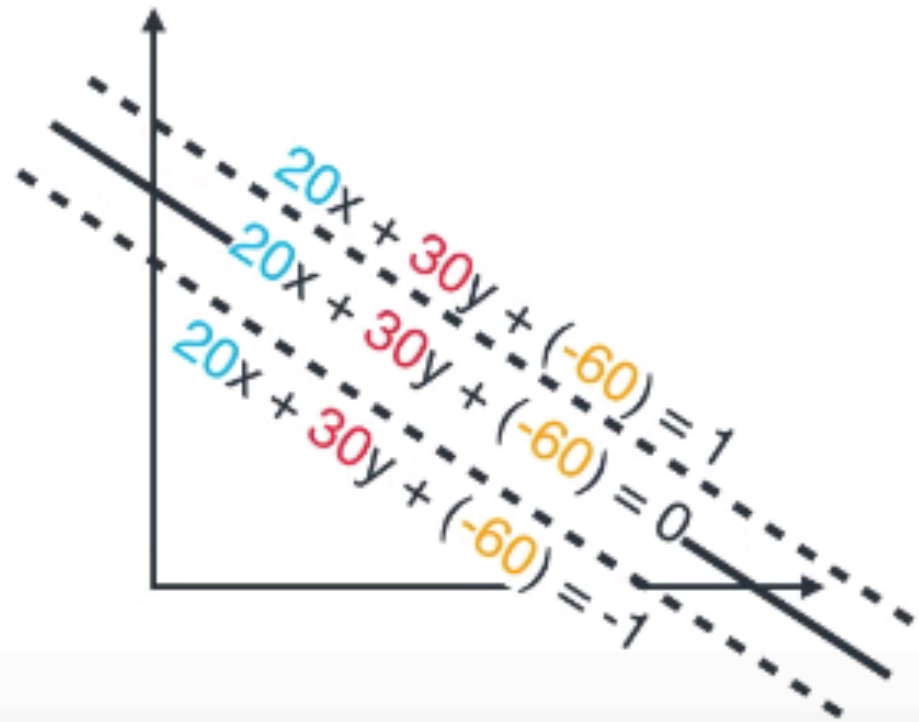
$$20x + 30y + (-60) = 0$$


2x + 3y + (-6) = 1
2x + 3y + (-6) = 0
2x + 3y + (-6) = -1

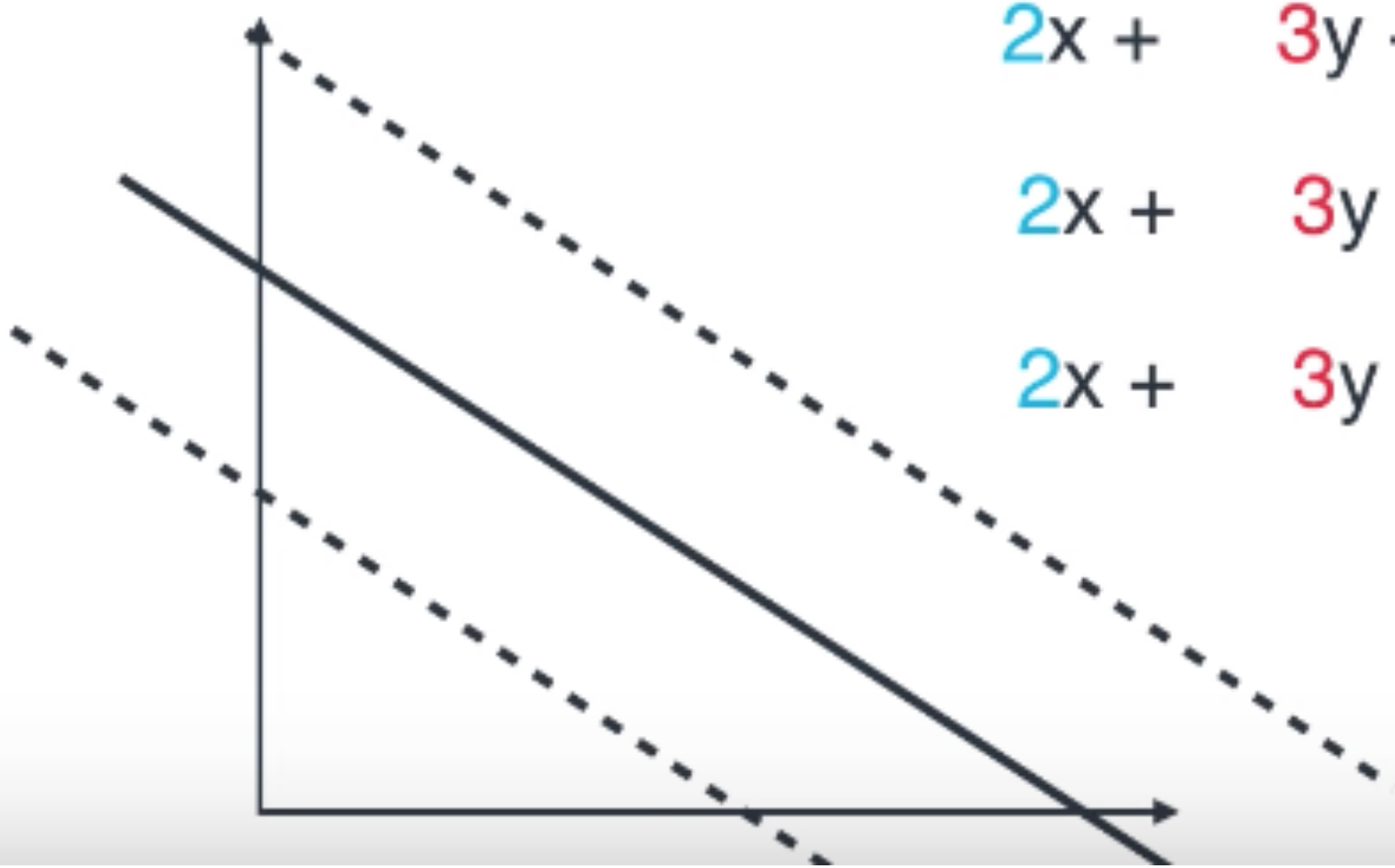

4x + 6y + (-12) = 0

How to separate lines?





Expanding rate



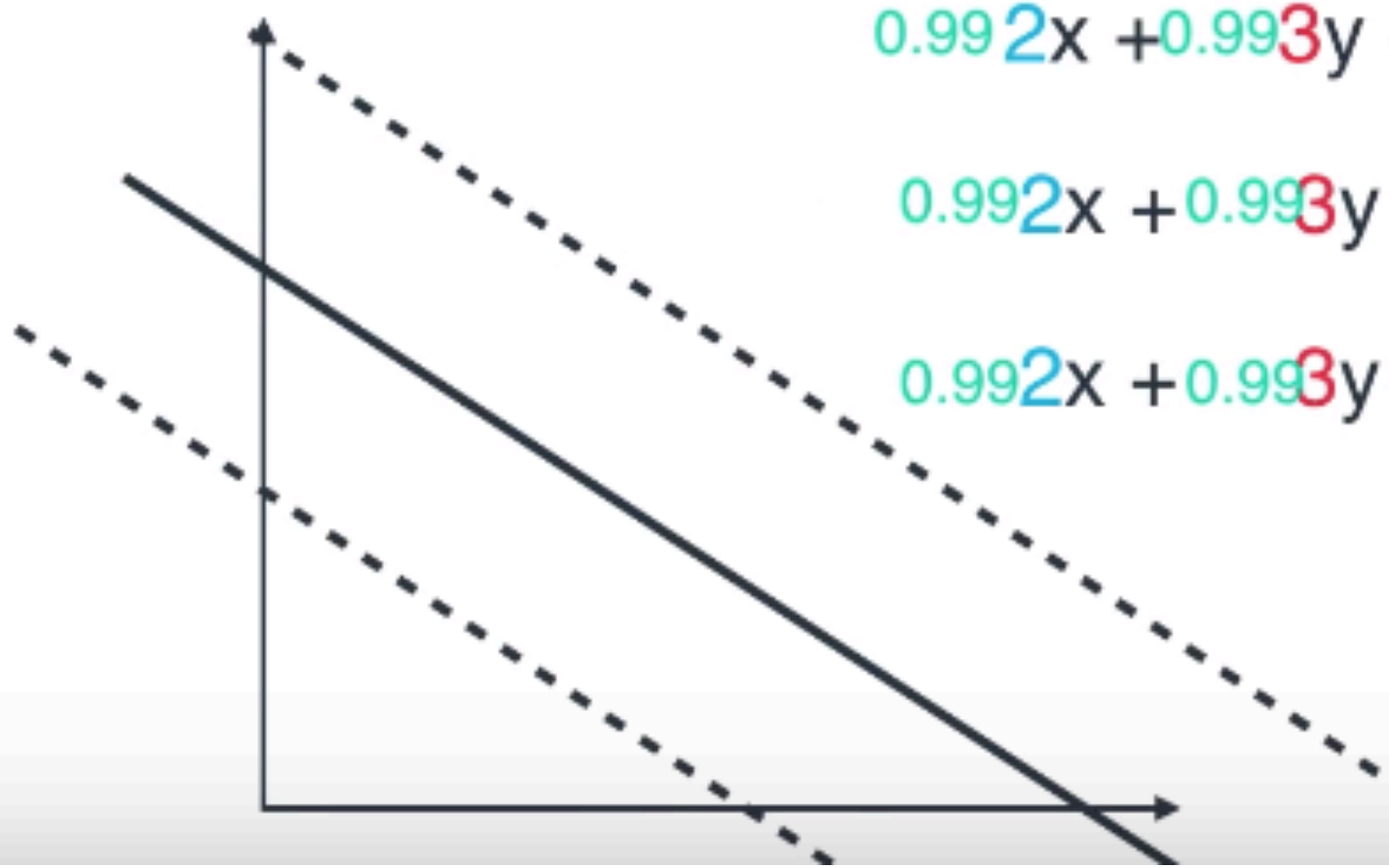
$2x + 3y + (-6) = -1$

$2x + 3y + (-6) = 0$

$2x + 3y + (-6) = 1$

Expanding rate

Expanding rate

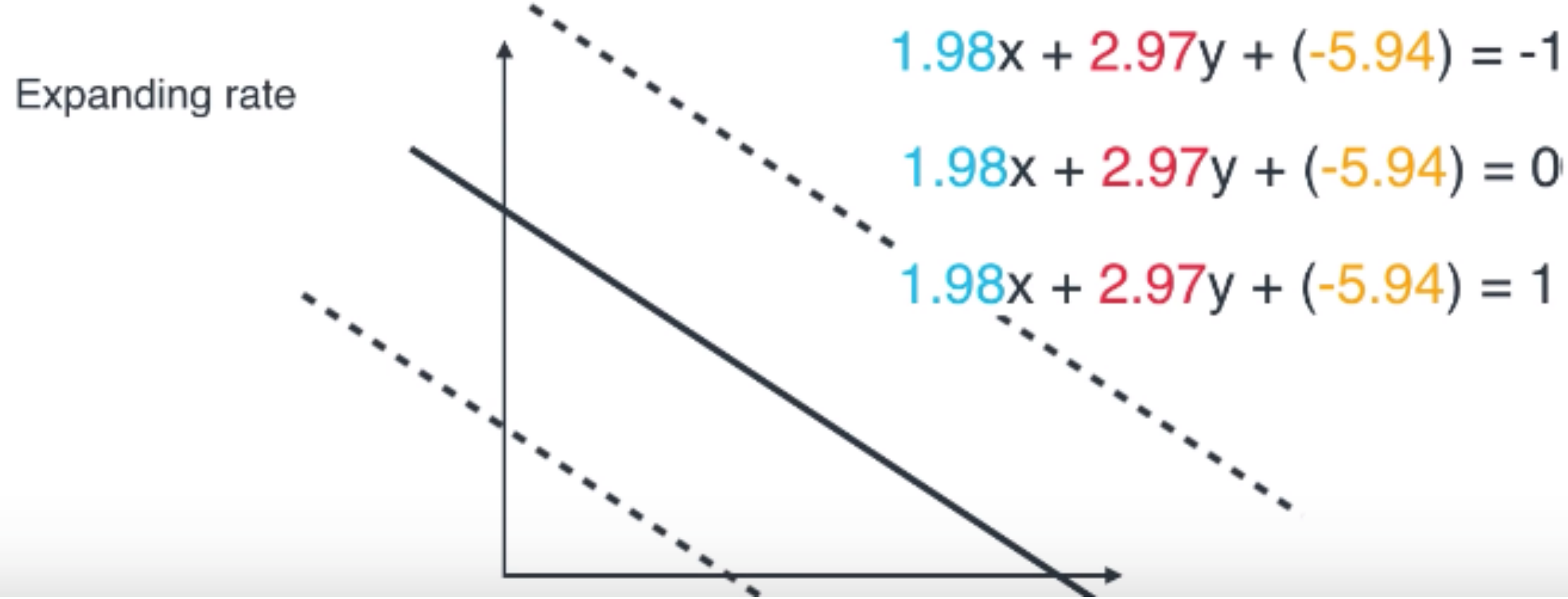


$$0.992x + 0.993y + 0.99(-6) = -1$$

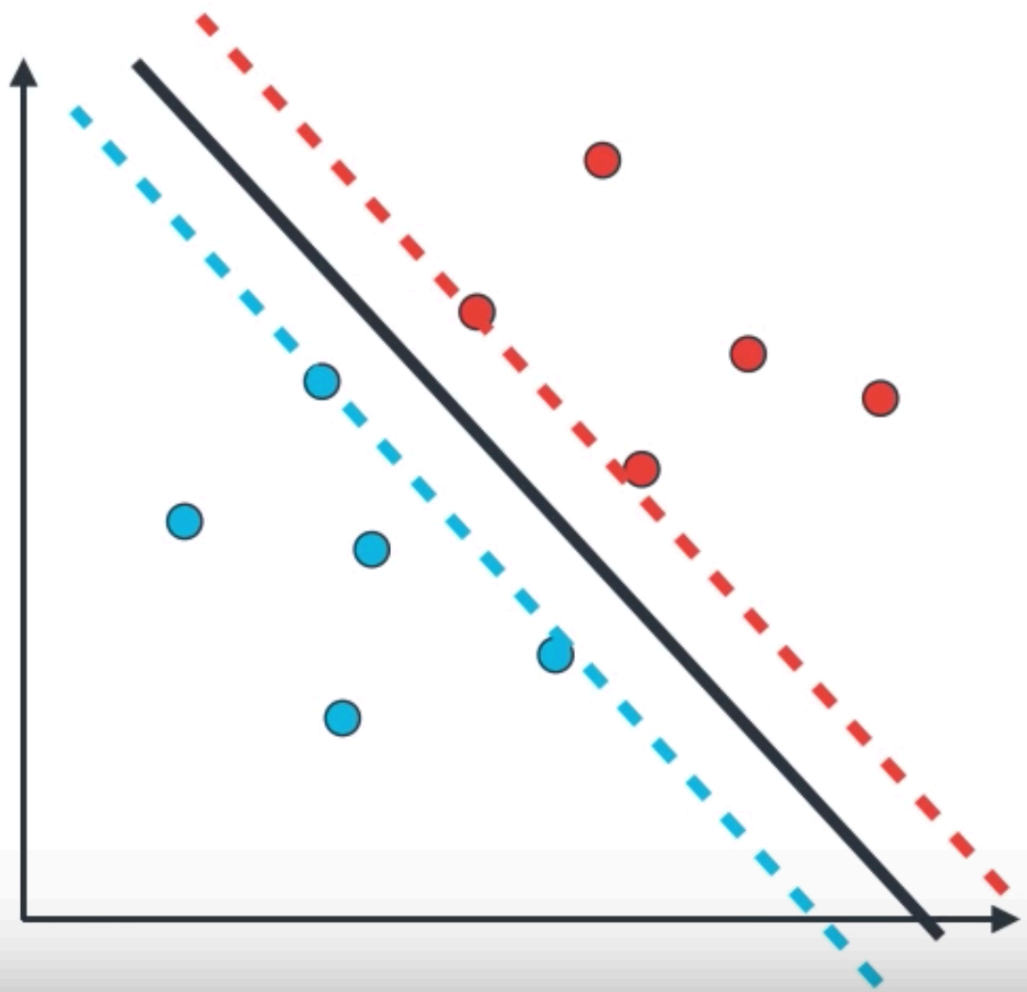
$$0.992x + 0.993y + 0.99(-6) = 0$$

$$0.992x + 0.993y + 0.99(-6) = 1$$

Expanding rate



SVM algorithm



Step 1: Start with a random line of equation $ax + by + c = 0$.
Draw parallel lines with equations:
- $ax + by + c = 1$, and
- $ax + by + c = -1$

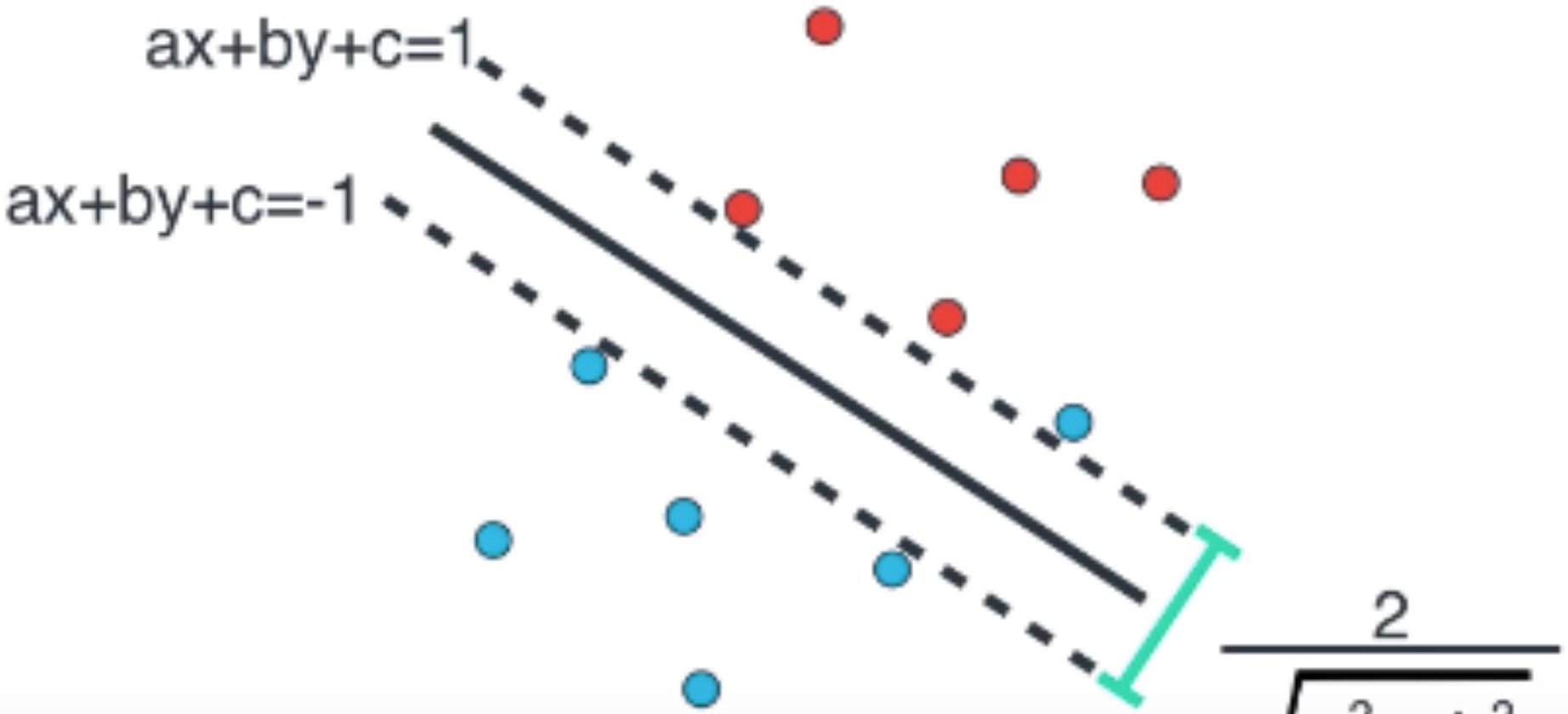
Step 2: Pick a large number. **1000** (number of repetitions, or epochs)

Step 3: Pick a learning rate. **0.01**

Step 4: Pick an expanding rate. **0.99**

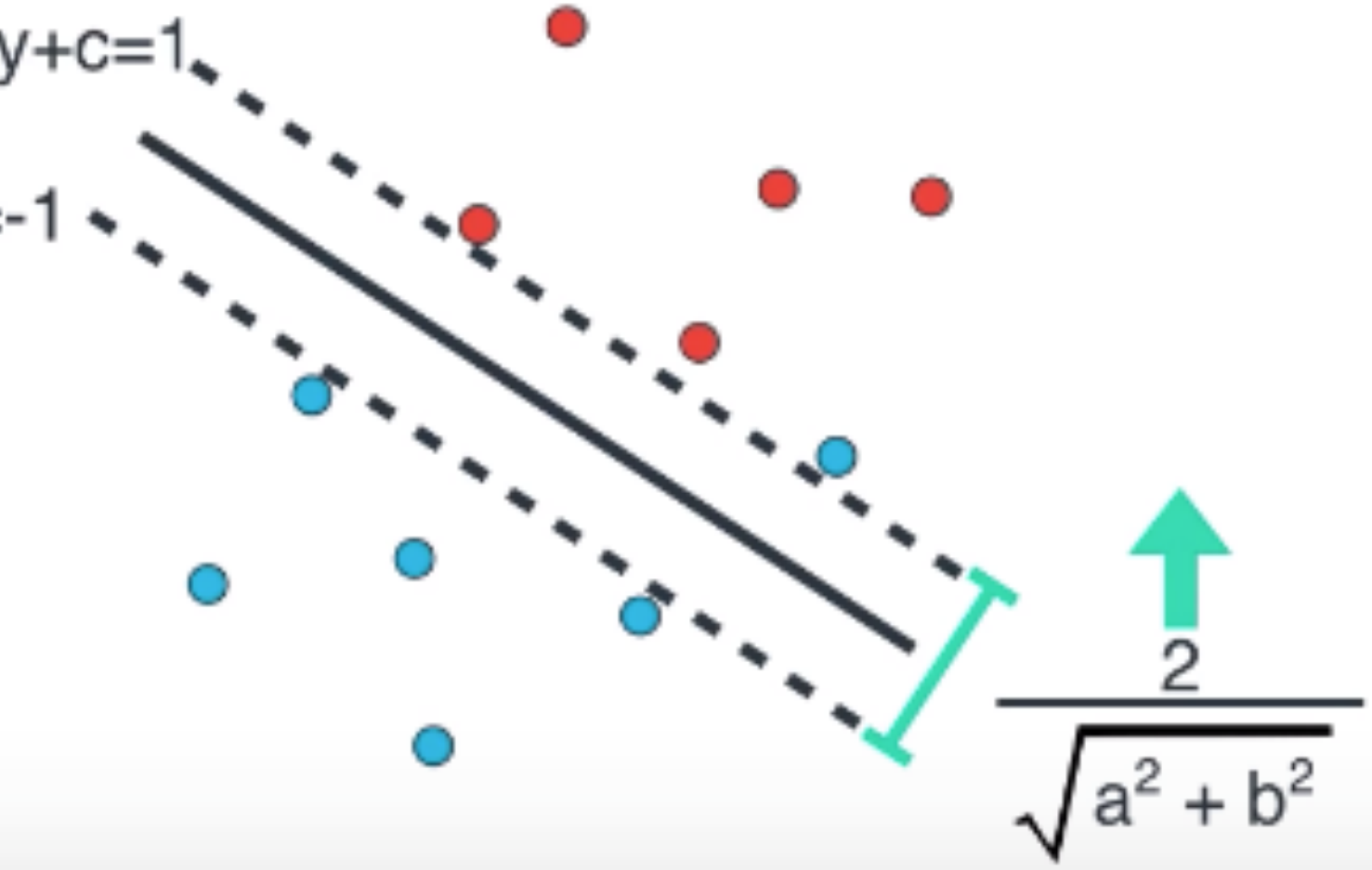
Step 5: (repeat **1000** times)
- Pick random point **(p,q)**
- If point is correctly classified
  - Do nothing
- If point is blue, and $ap+bq+c > 0$
  - Subtract 0.01**p** to a
  - Subtract 0.01**q** to b
  - Subtract 0.01 to c
- If point is, red and $ap+bq+c < 0$
  - Add 0.01**p** to a
  - Add 0.01**q** to b
  - Add 0.01 to c
- **Multiply a, b, c, by 0.99**

Margin Error



$$\frac{2}{\sqrt{a^2 + b^2}}$$

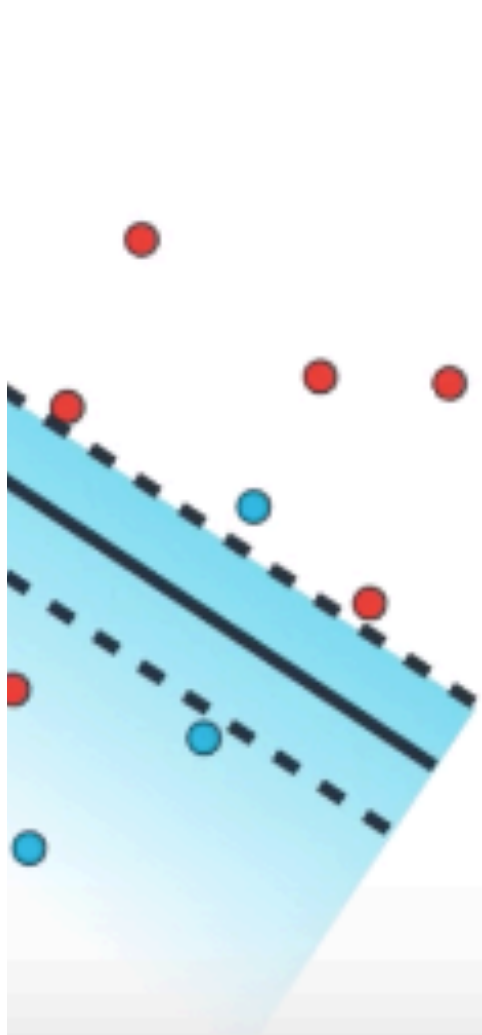
Margin Error

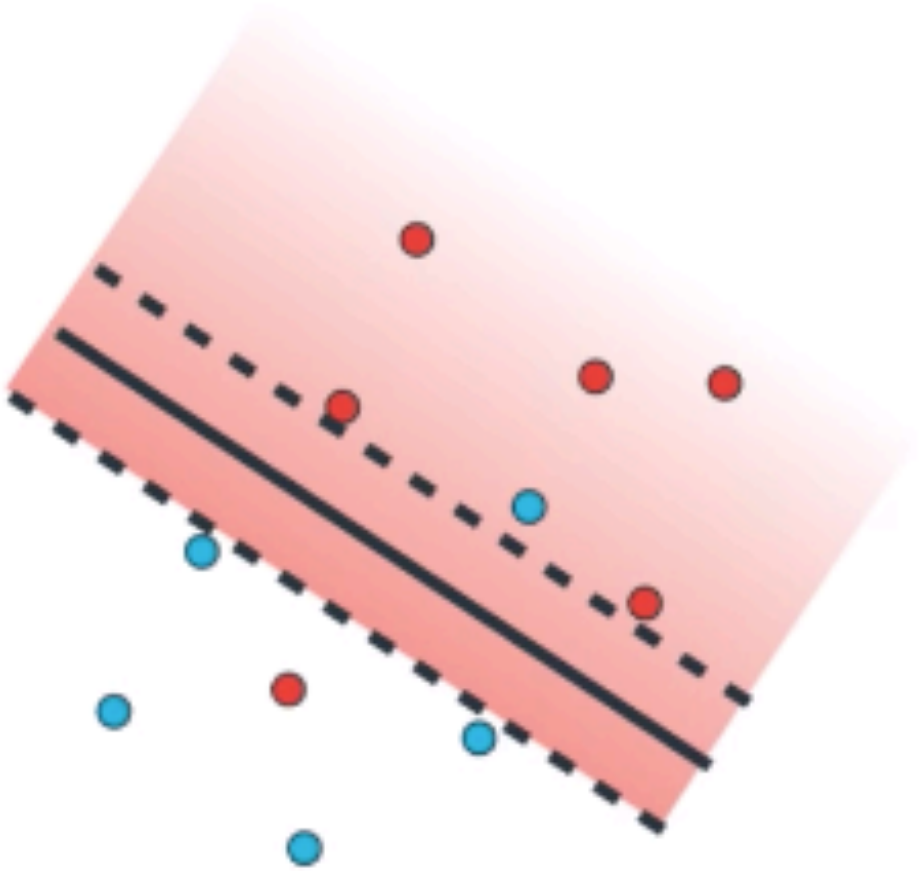


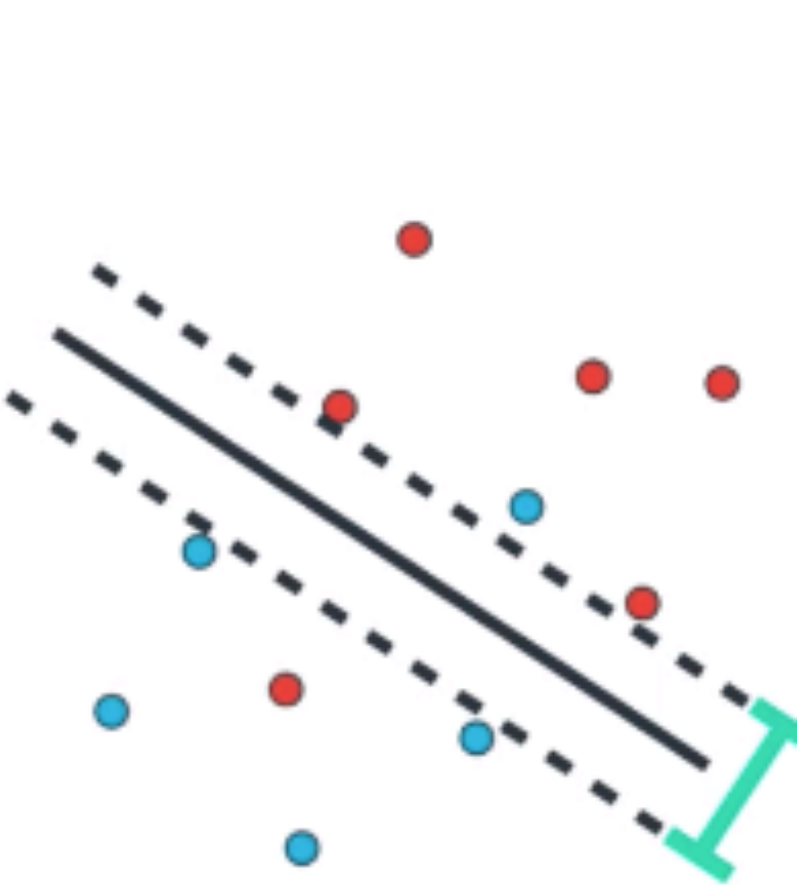
$$\frac{2}{\sqrt{a^2 + b^2}}$$

Margin error = $a^2 + b^2$

SVM Error





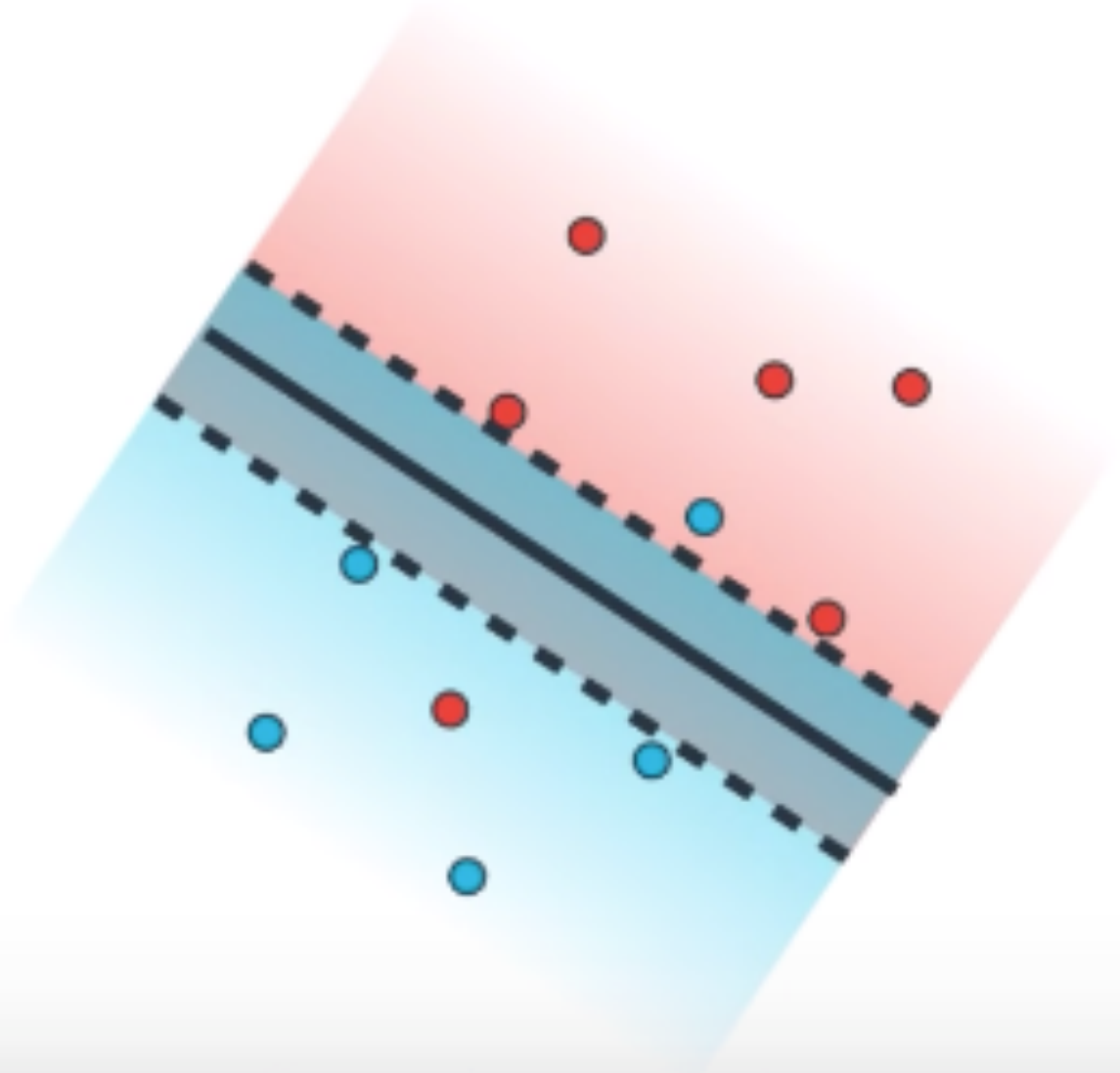


fication Error

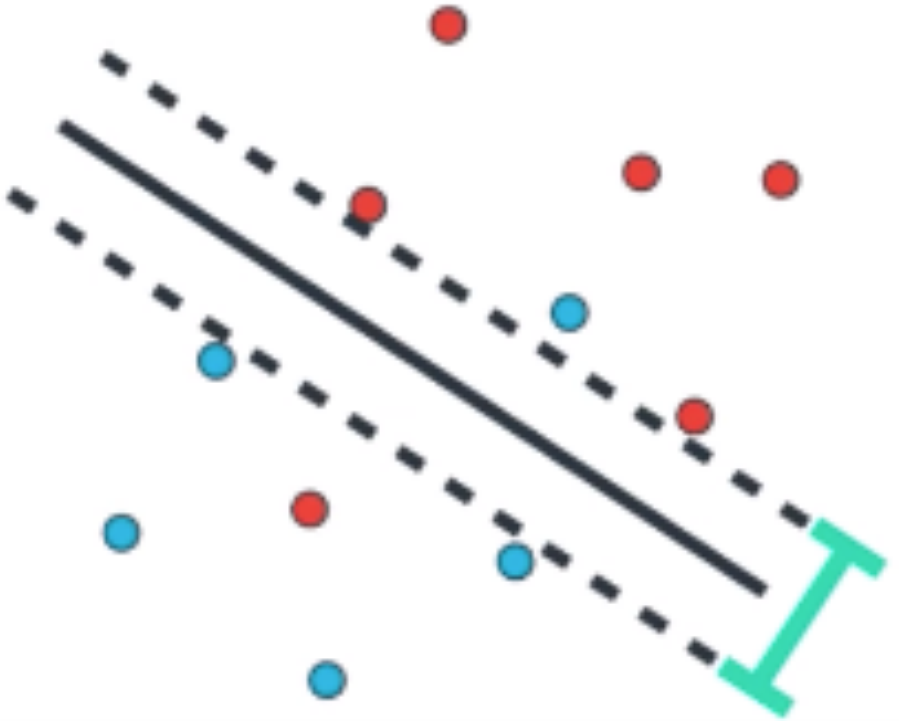
Red Classification Error

Margin Error

SVM Error



Classification Error



Margin Error

Gradient Descent

Same as the SVM trick!

Minimize using calculus (gradient descent)