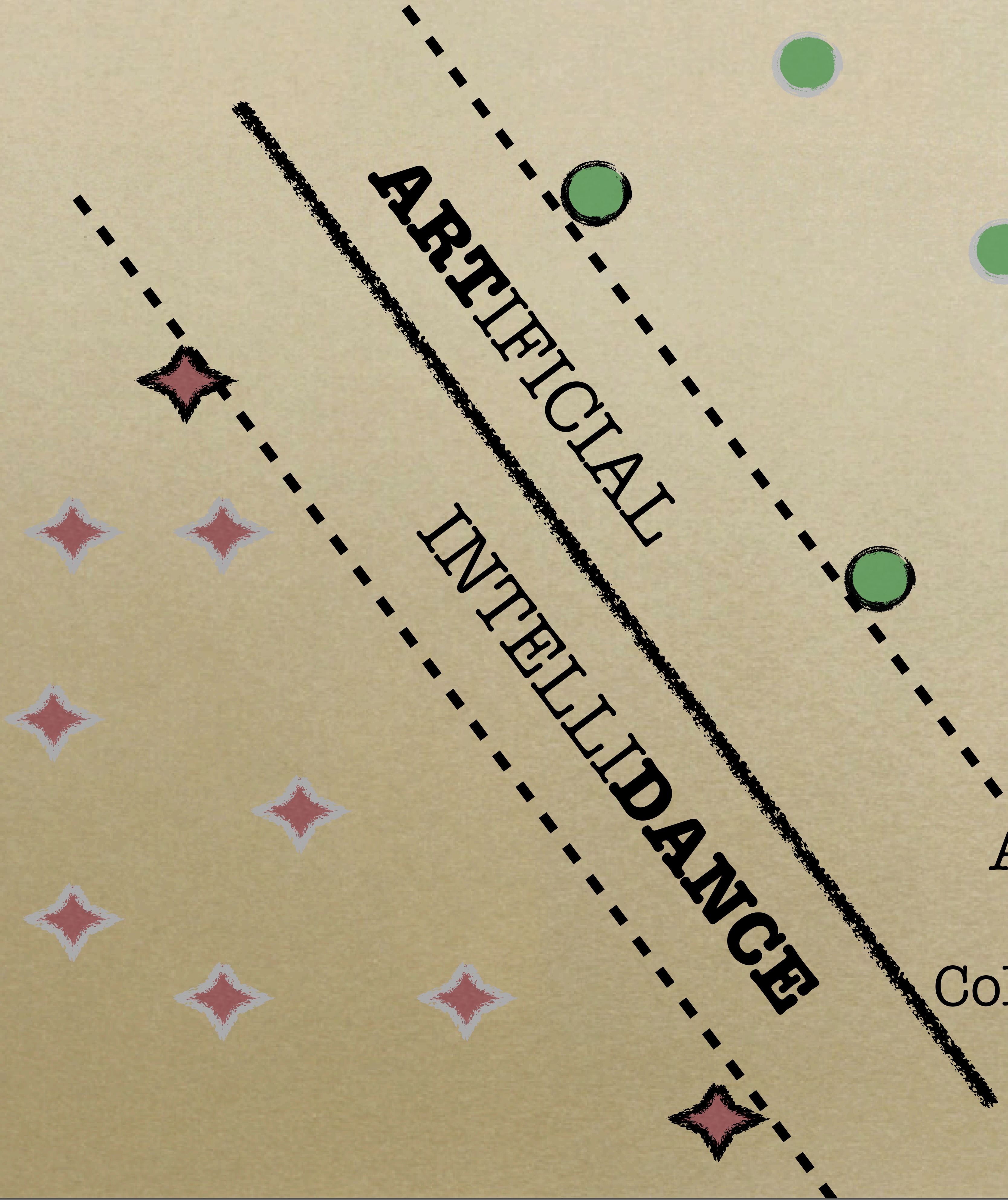


ARTIFICIAL INTELLI**DANCE**

Apoorv Agarwal
Caitlin Trainor
Columbia University
New York

What is it about?

What is it about?

It is a choreography that explains how

What is it about?

It is a choreography that explains how a machine learns to differentiate between two categories of objects.

What is it about?

It is a choreography that explains how a machine learns to differentiate between two categories of objects.

More technically:

What is it about?

It is a choreography that explains how a machine learns to differentiate between two categories of objects.

More technically:

- a perceptron

What is it about?

It is a choreography that explains how a machine learns to differentiate between two categories of objects.

More technically:

- a perceptron
- in it's dual form

What is it about?

It is a choreography that explains how a machine learns to differentiate between two categories of objects.

More technically:

- a perceptron
- in it's dual form
- uses convolution kernels

What is it about?

It is a choreography that explains how a machine learns to differentiate between two categories of objects.

More technically:

- a perceptron
- in it's dual form
- uses convolution kernels
- to learn to differentiate between two categories of objects.

What is it about?

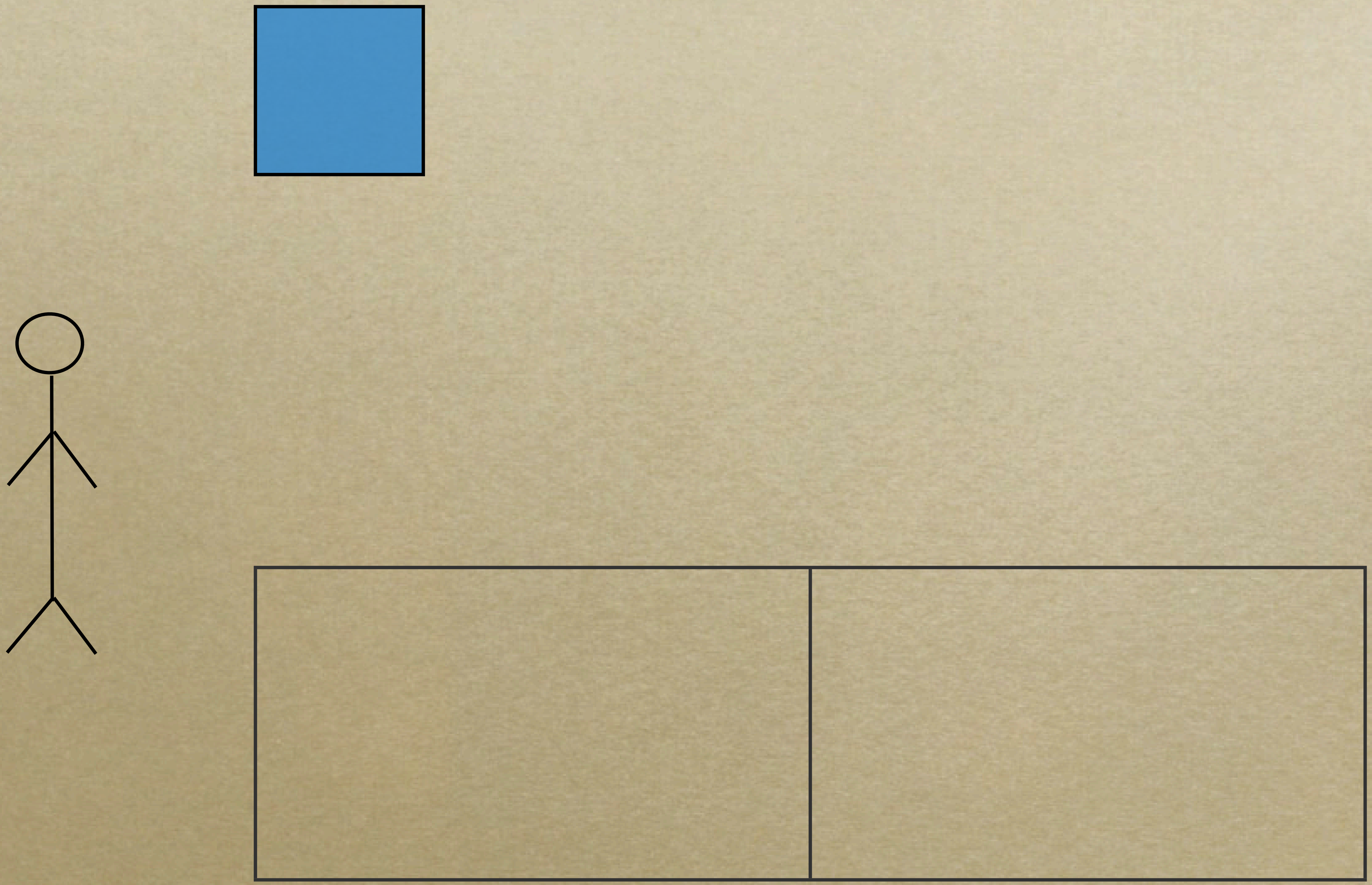
It is a choreography that explains how a machine learns to differentiate between two categories of objects.

More ~~technically:~~

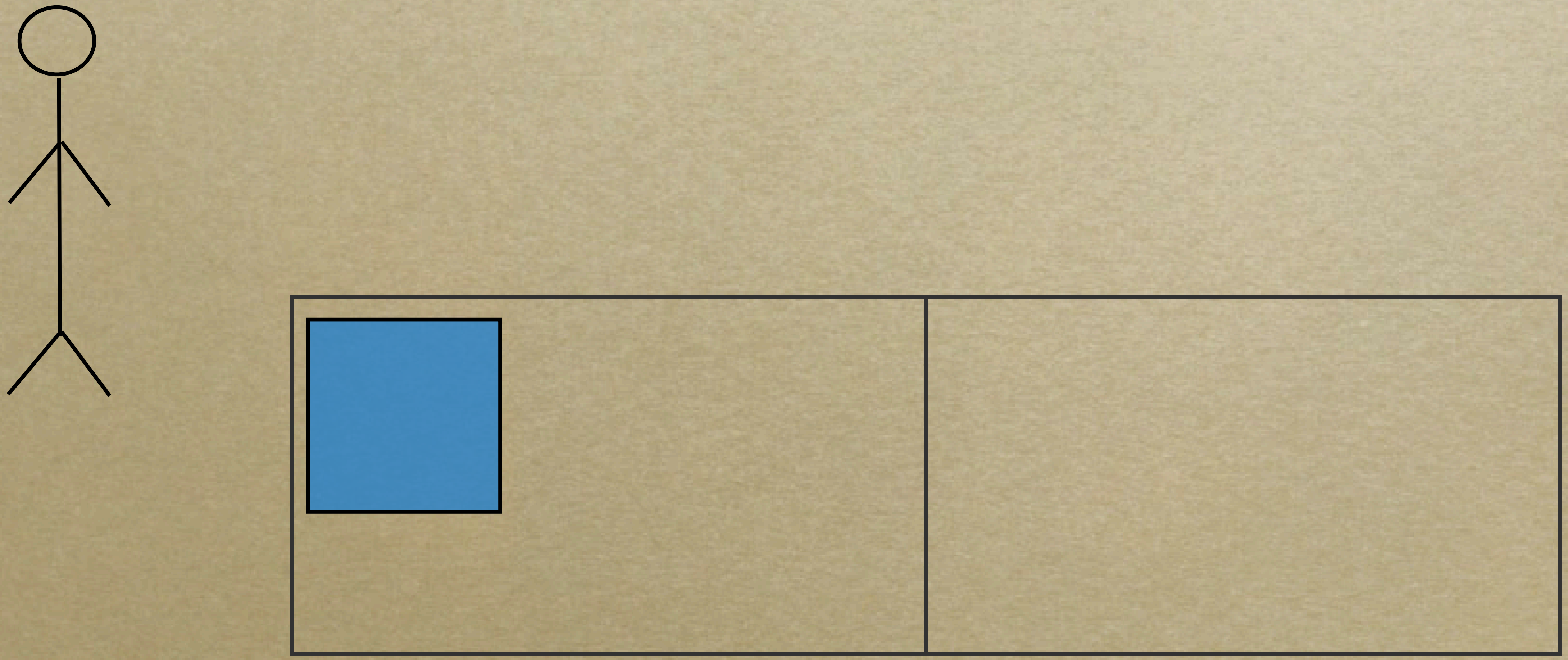
- ~~a perceptron~~
- ~~in it's dual form~~
- ~~uses convolution kernels~~
- ~~to learn to differentiate between two categories of objects.~~

How do humans learn?

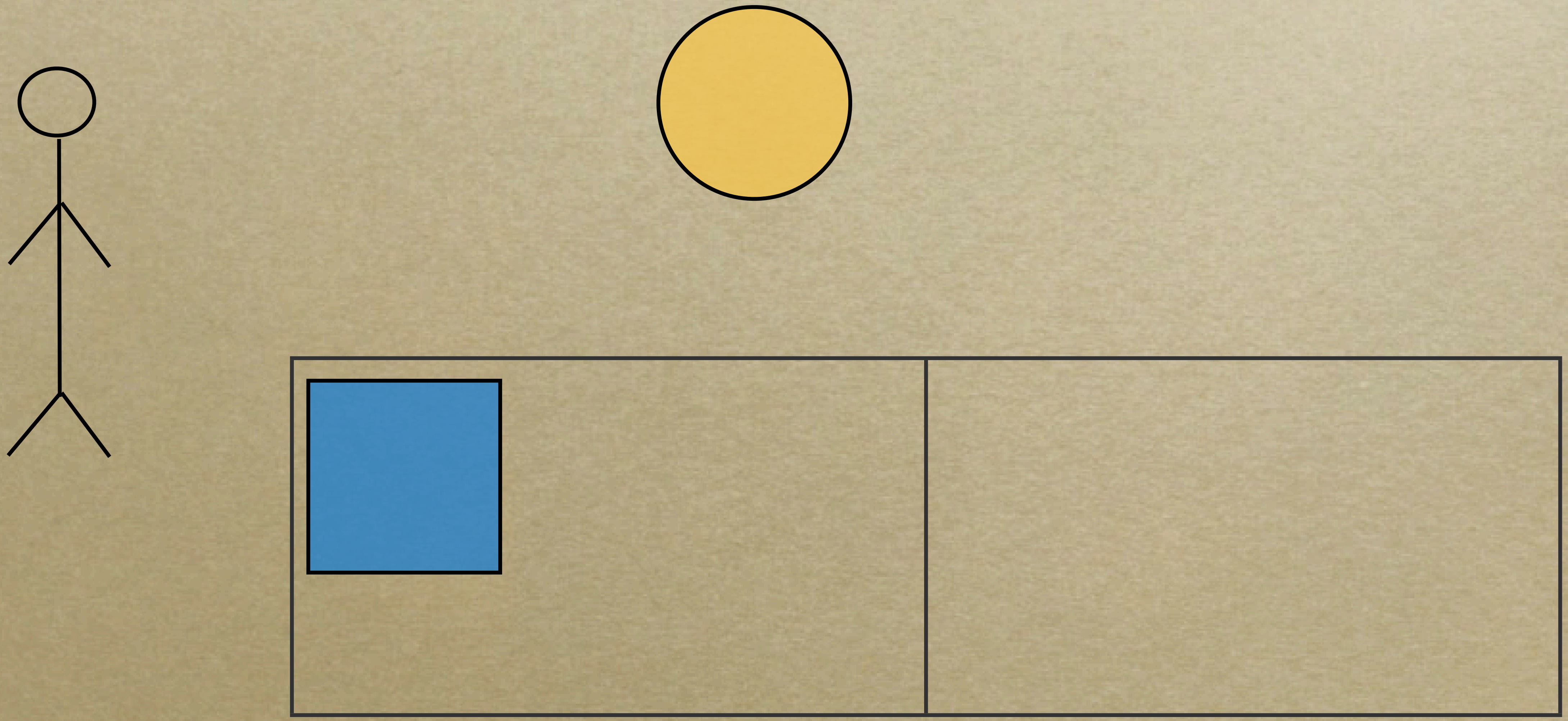
How do humans learn?



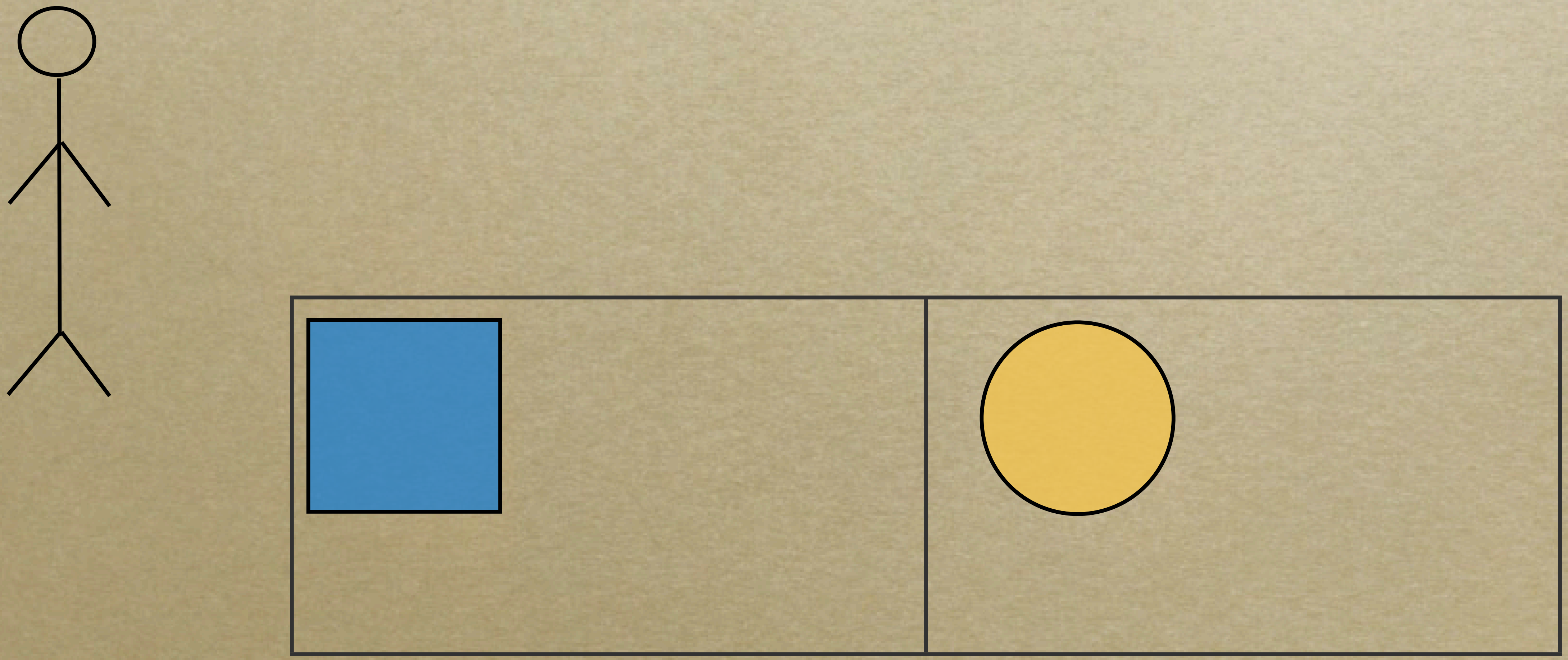
How do humans learn?



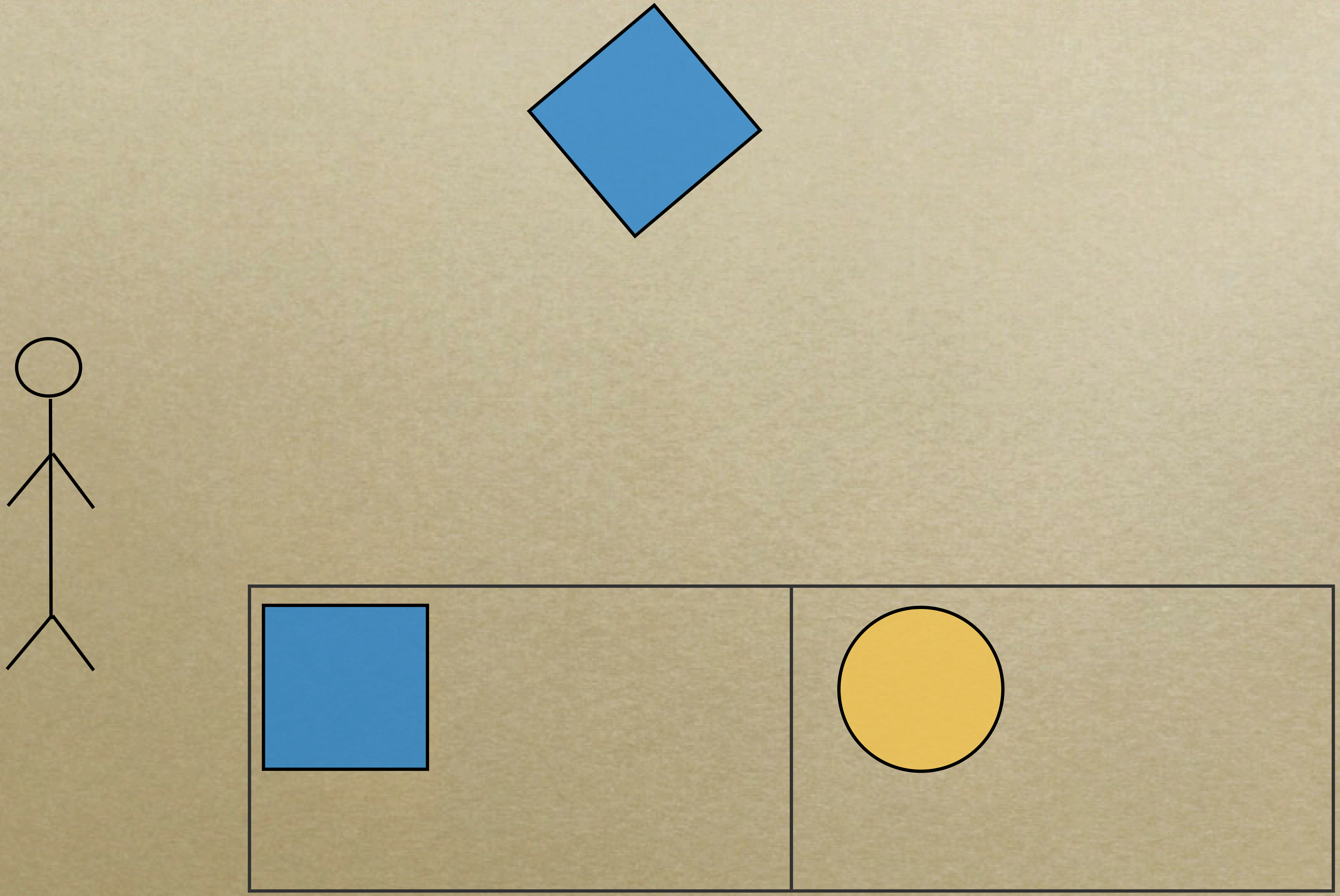
How do humans learn?



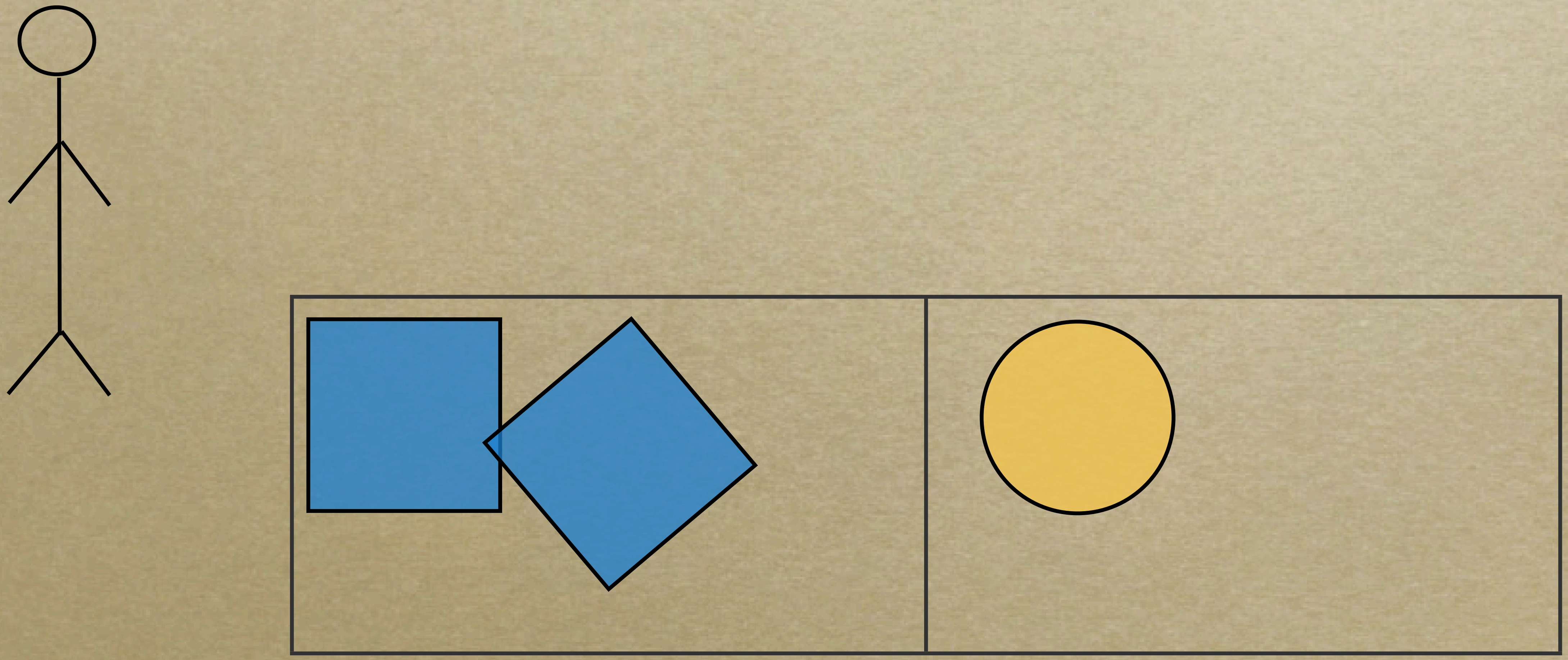
How do humans learn?



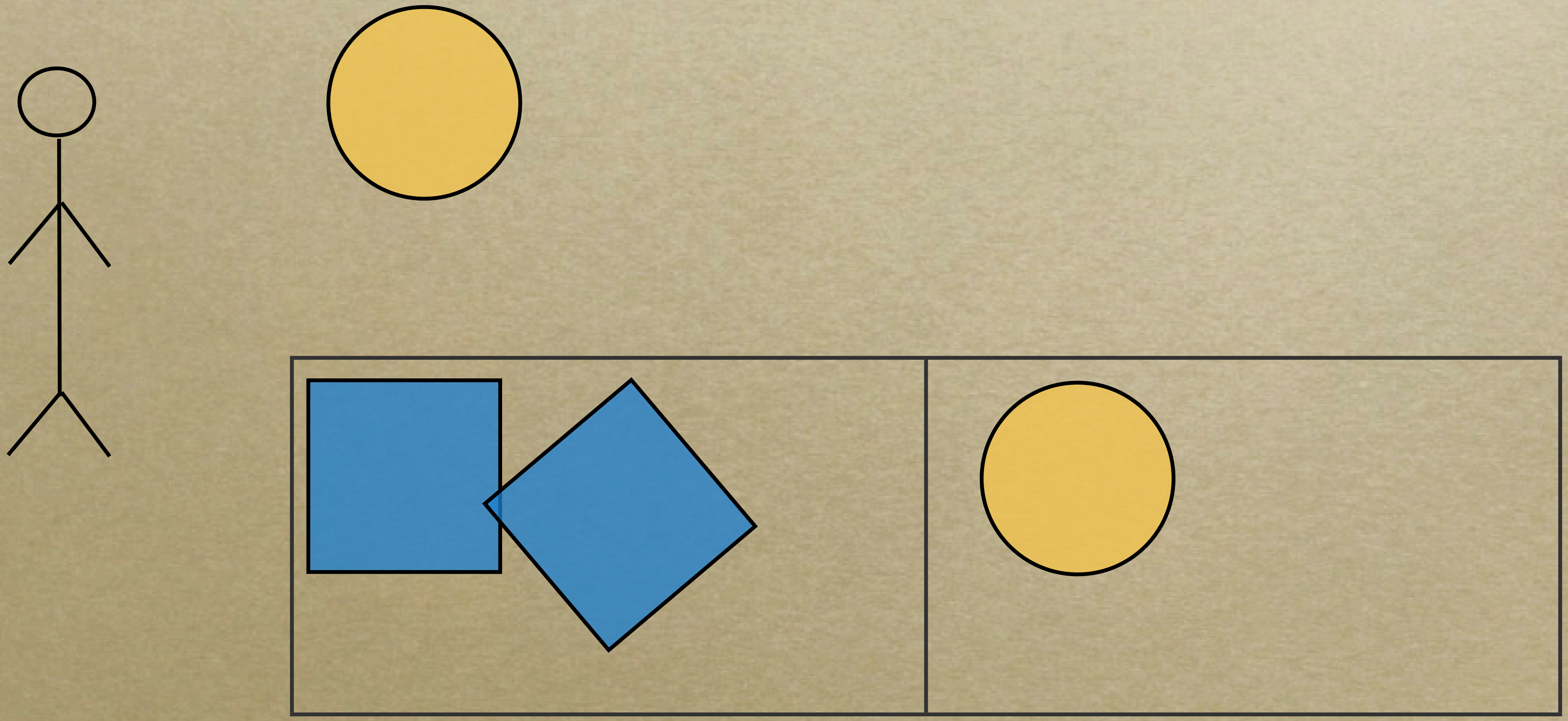
How do humans learn?



How do humans learn?



How do humans learn?



Convolution Kernels

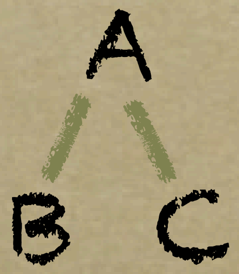
Abstract Objects

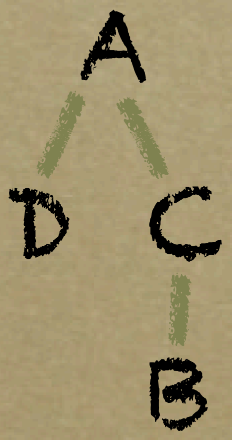
$K: X \times X \to R$

Convolution Kernels

Abstract Objects

K: XxX -> R

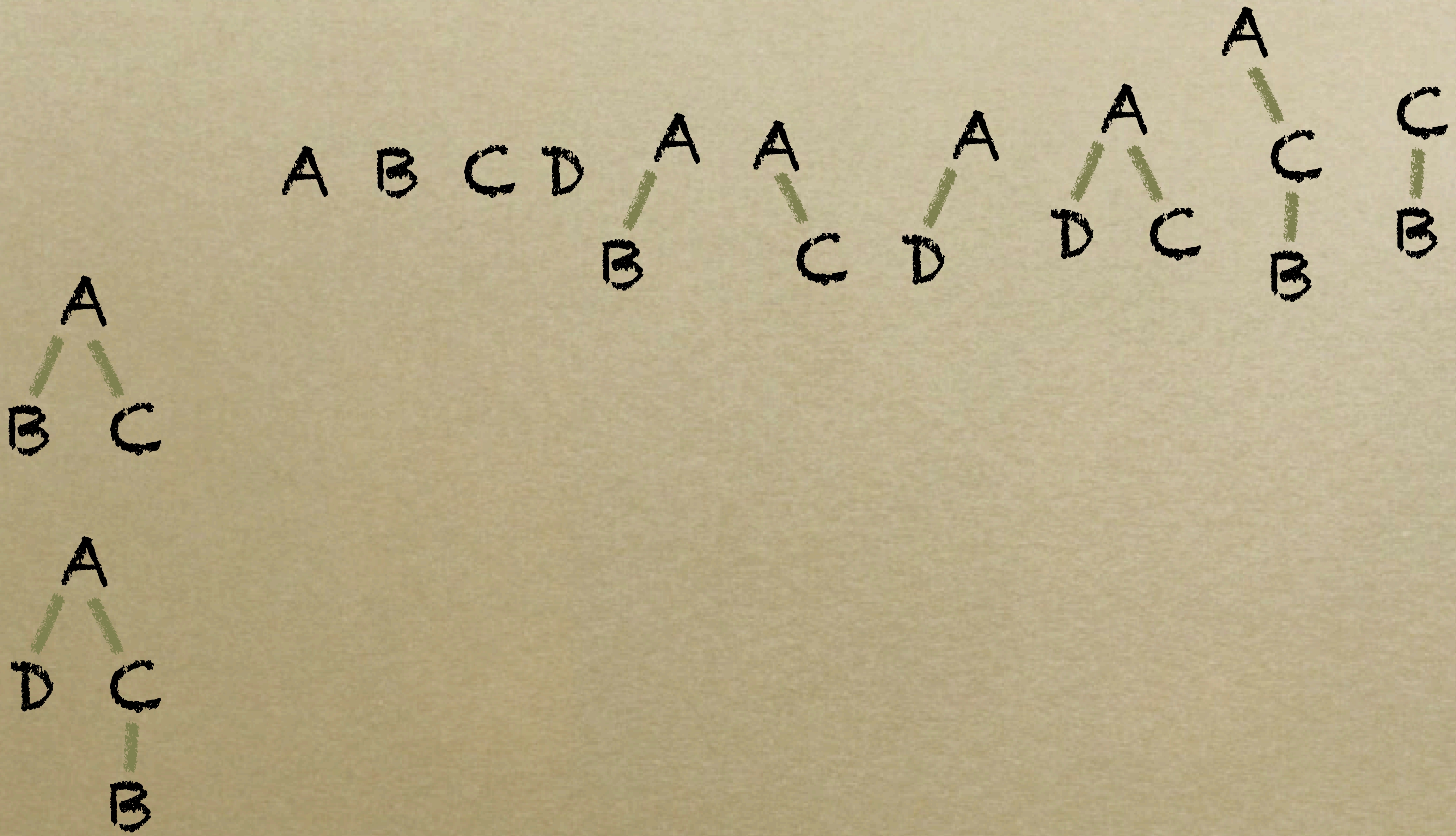




Convolution Kernels

Abstract Objects

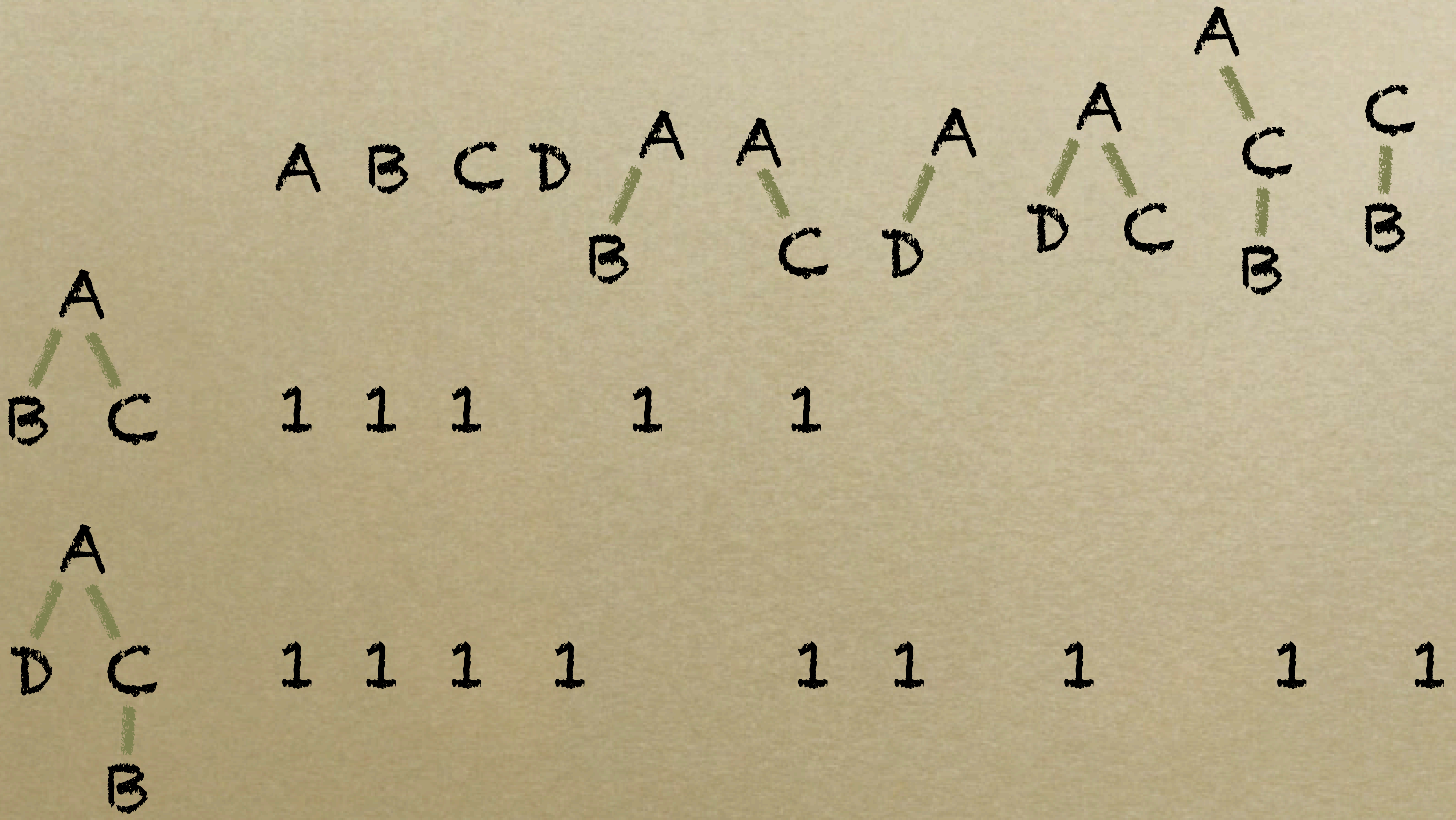
K: XxX -> R



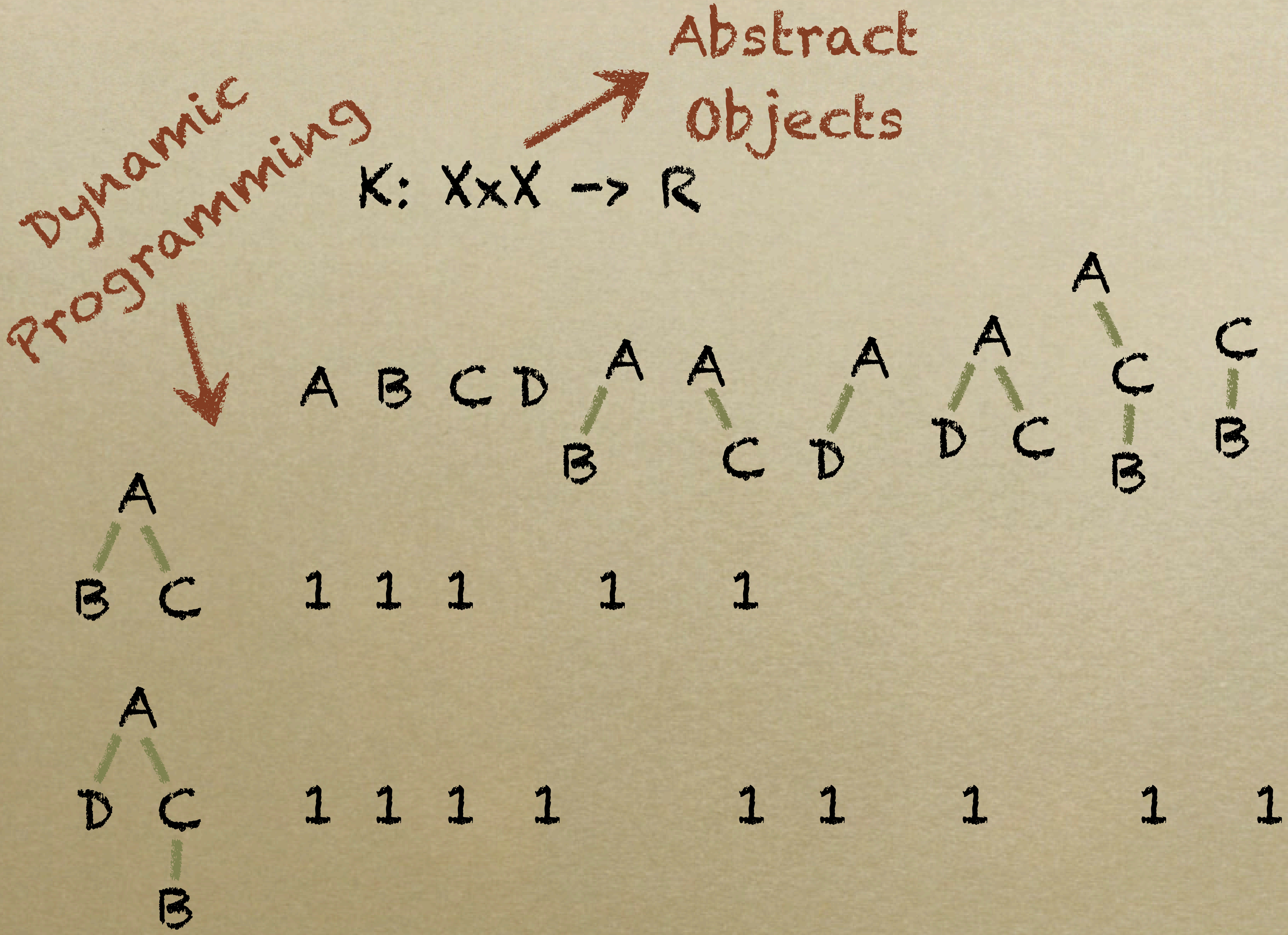
Convolution Kernels

Abstract Objects

K: XxX -> R



Convolution Kernels

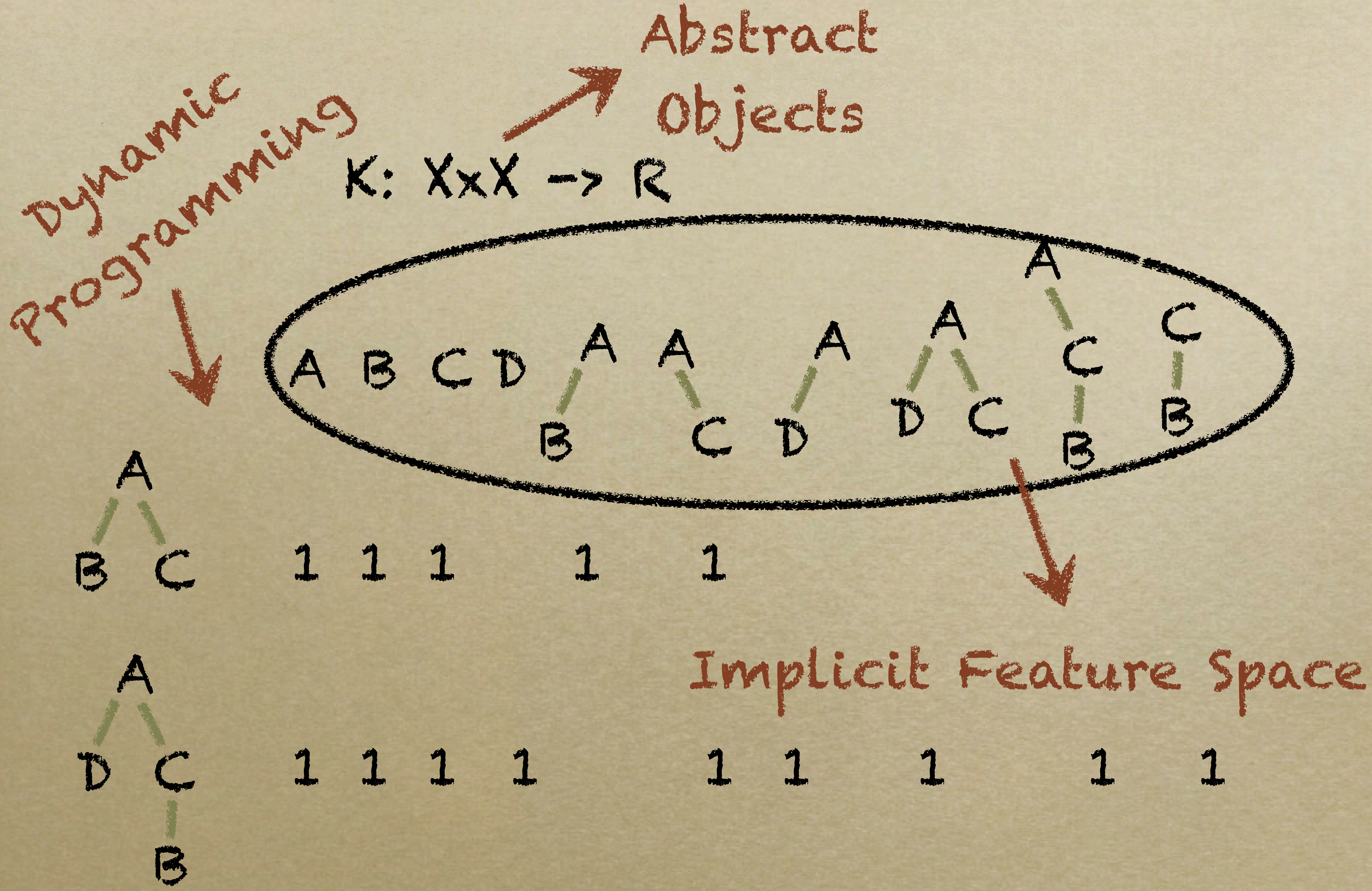


Abstract Objects

Dynamic Programming

K: XxX -> R

Convolution Kernels



Abstract Objects

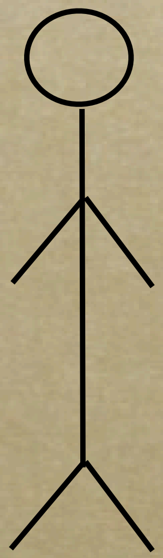
Dynamic Programming

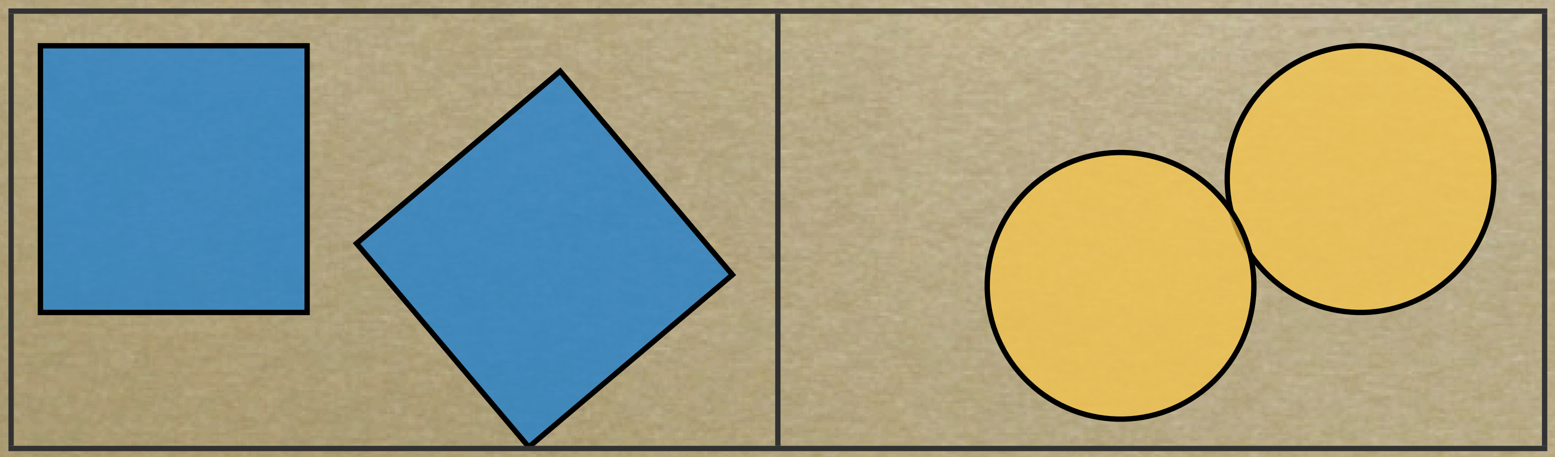
K: XxX -> R

Implicit Feature Space

The choreography

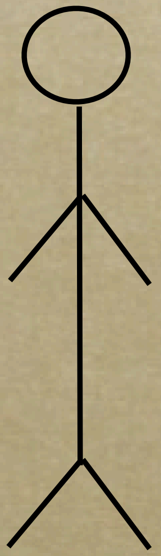
Two questions?

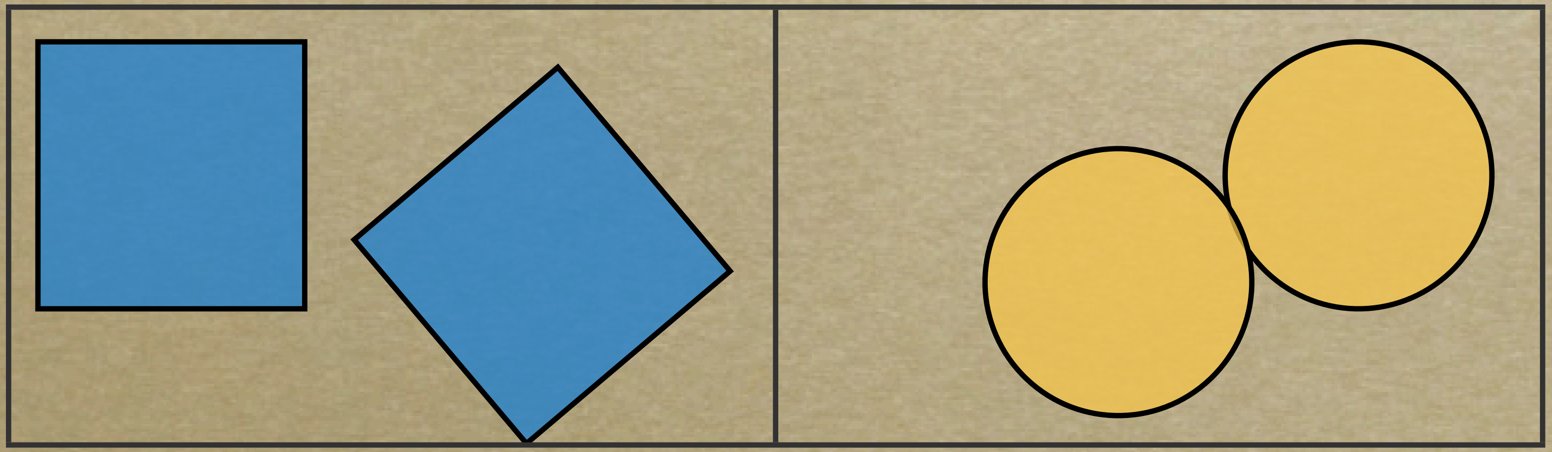




Two questions?

1. What are the classes/categories?

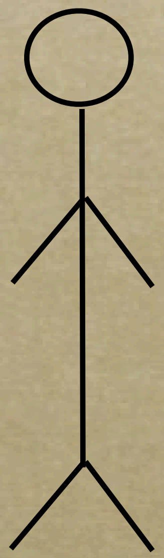


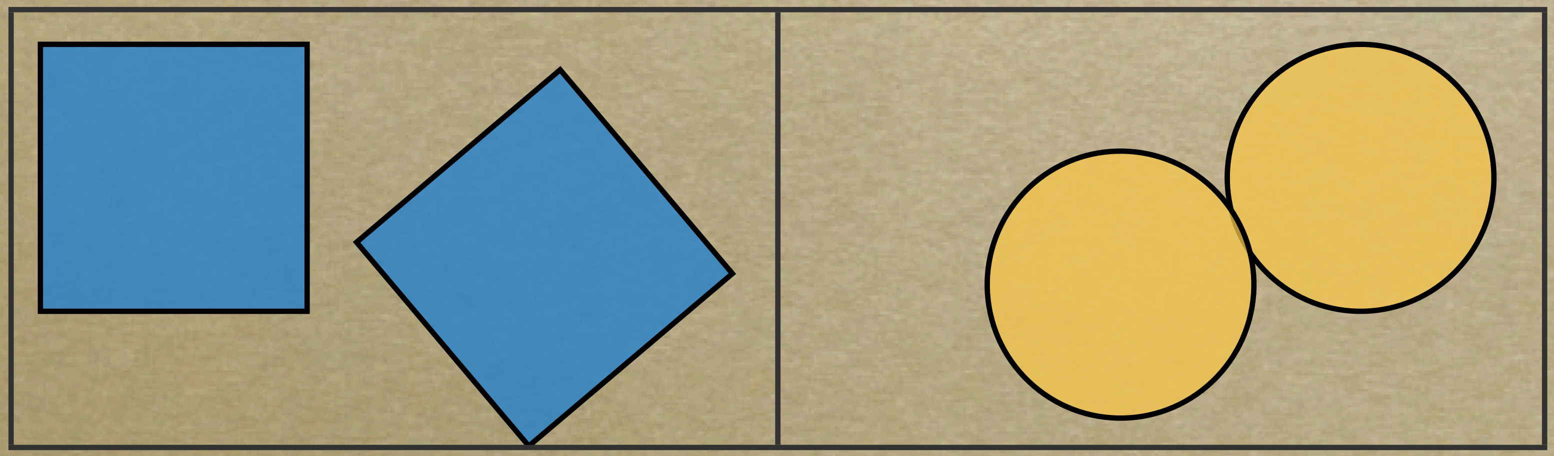


Two questions?

1. What are the classes/categories?

2. How is the machine calculating similarities?



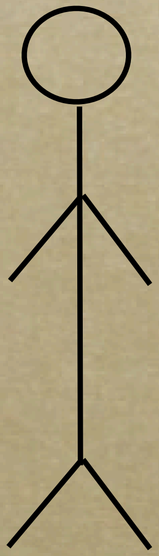


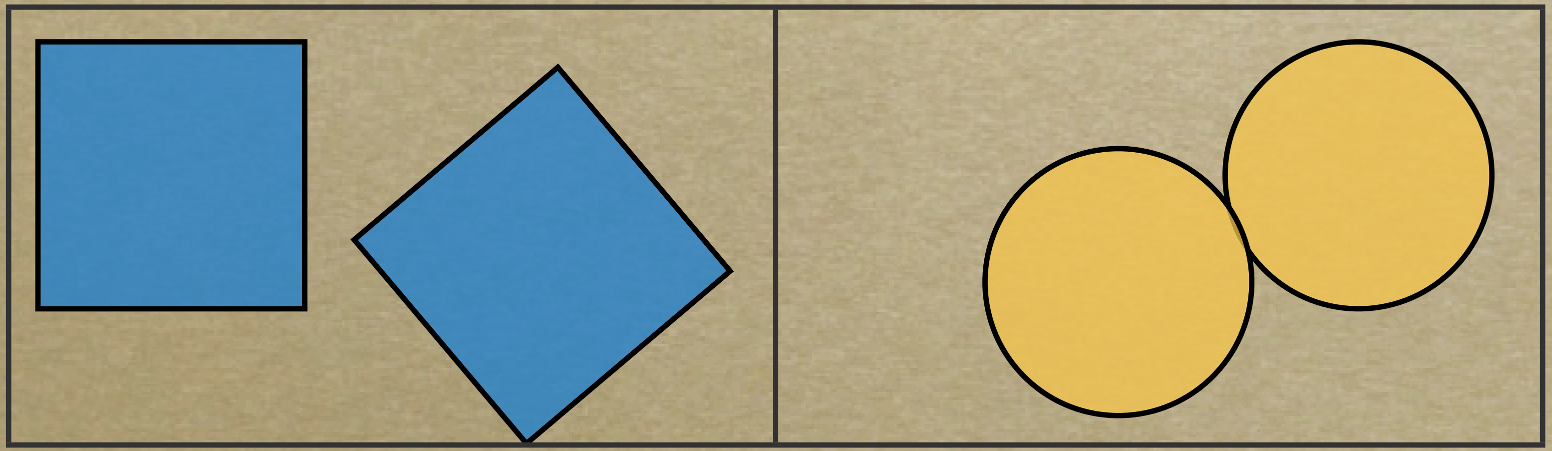
Two questions?

{Ballet, Modern}

1. What are the classes/categories?

2. How is the machine calculating similarities?





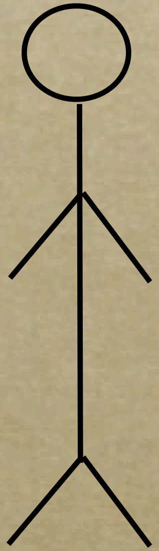
Two questions?

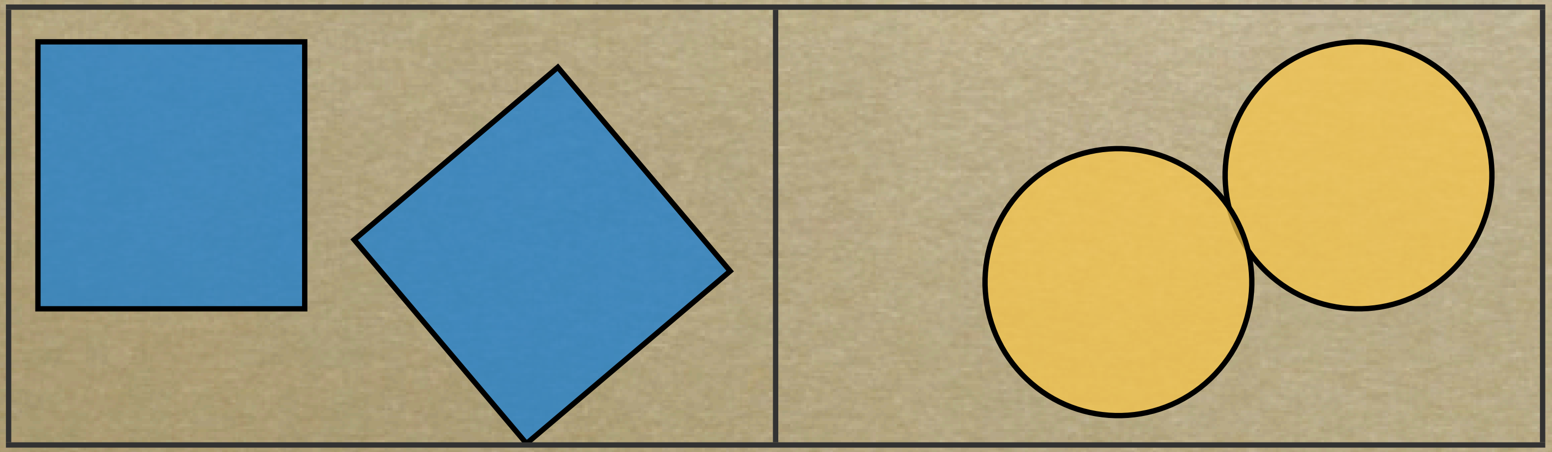
1. What are the classes/categories?

{Ballet, Modern}

2. How is the machine calculating similarities?

Movement kernel



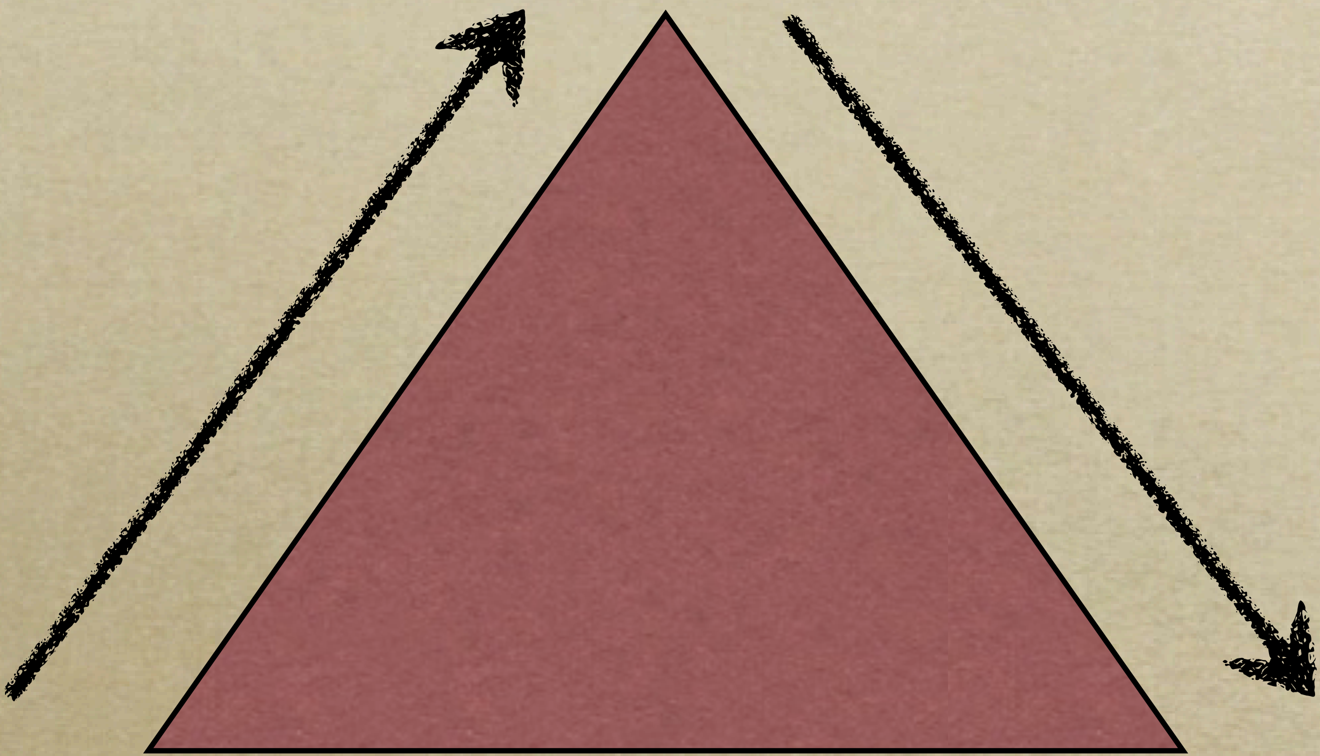


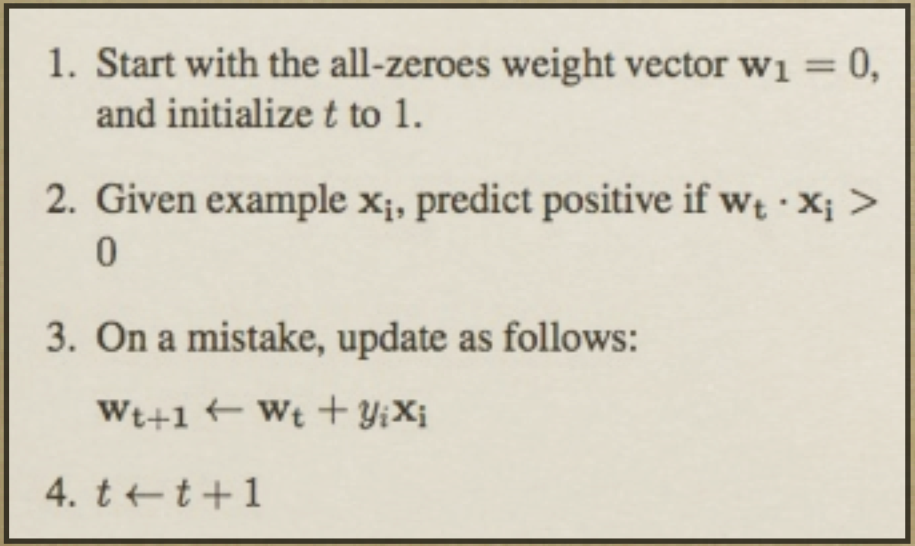
Hilbert Space

Hilbert Space

Conclusion

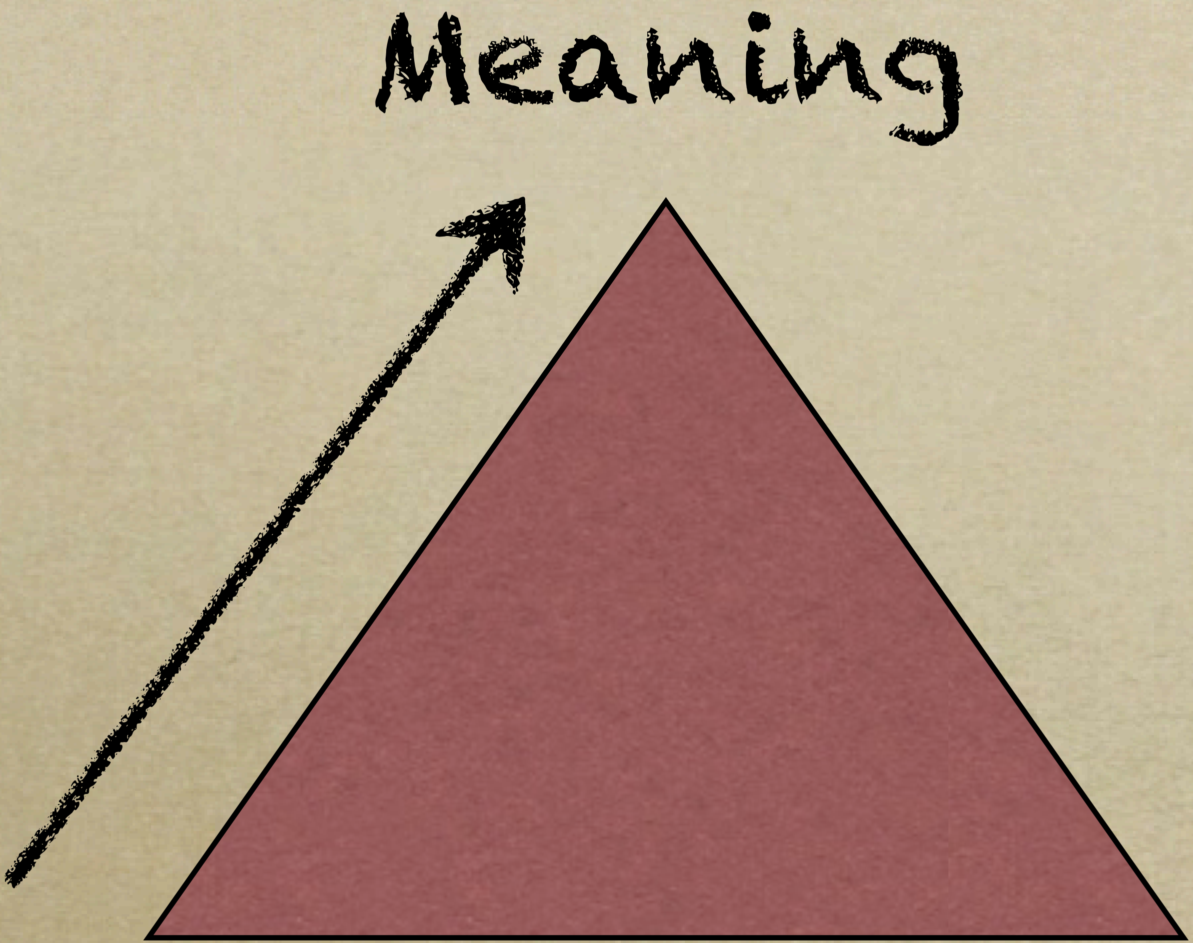
Meaning



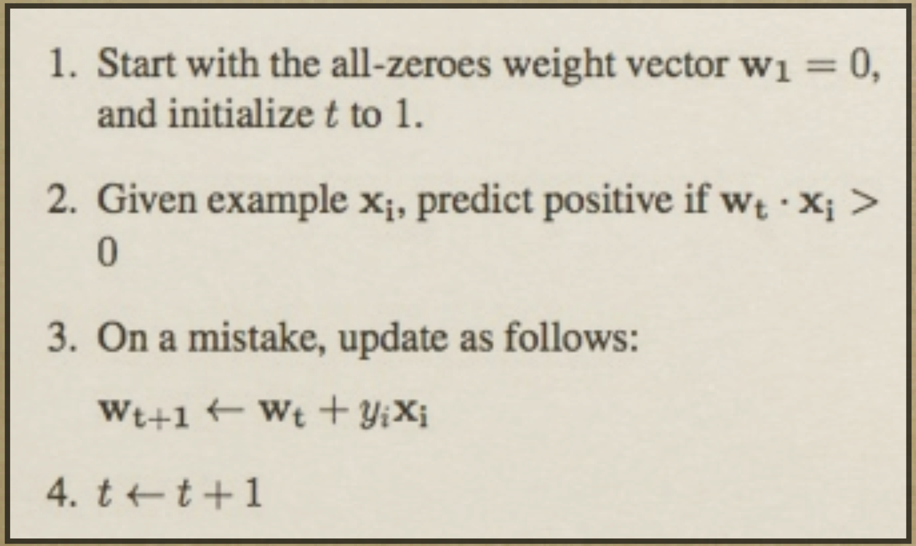


```
1. Start with the all-zeroes weight vector $\mathbf{w_1} = 0$,
   and initialize $t$ to 1.
2. Given example $\mathbf{x_i}$, predict positive if $\mathbf{w_t} \cdot \mathbf{x_i} >$
   $0$
3. On a mistake, update as follows:
   $\mathbf{w_{t+1}} \leftarrow \mathbf{w_t} + y_i\mathbf{x_i}$
4. $t \leftarrow t + 1$
```

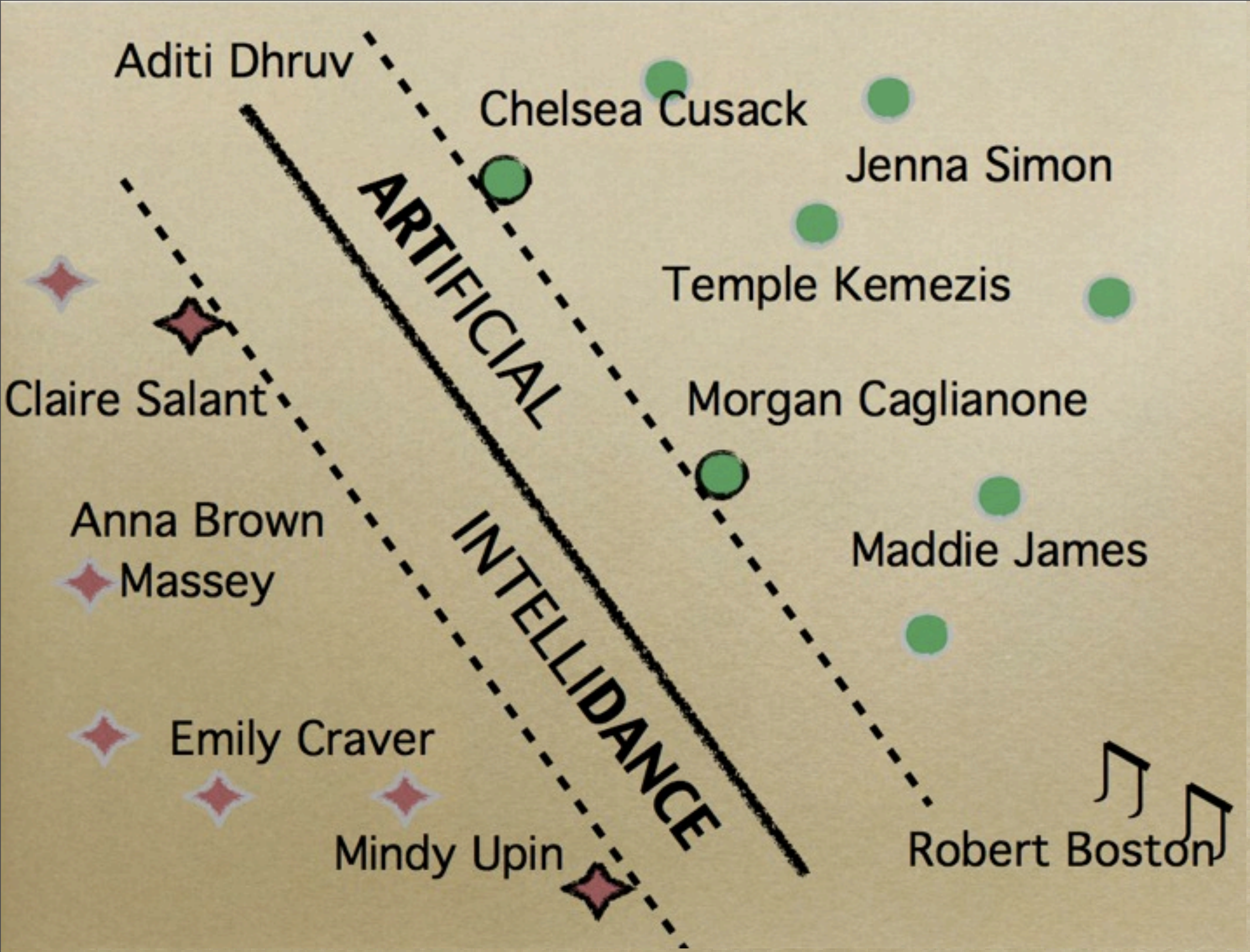
Conclusion



Meaning



```
1. Start with the all-zeroes weight vector $\mathbf{w_1} = 0$, and initialize $t$ to 1.
2. Given example $\mathbf{x_i}$, predict positive if $\mathbf{w_t} \cdot \mathbf{x_i} > 0$
3. On a mistake, update as follows:
   $\mathbf{w_{t+1}} \leftarrow \mathbf{w_t} + y_i \mathbf{x_i}$
4. $t \leftarrow t + 1$
```



Questions?

TEDxColumbiaEngineering: http://tinyurl.com/mte8wda

Like us on FB: https://www.facebook.com/ArtificialIntelliDance