

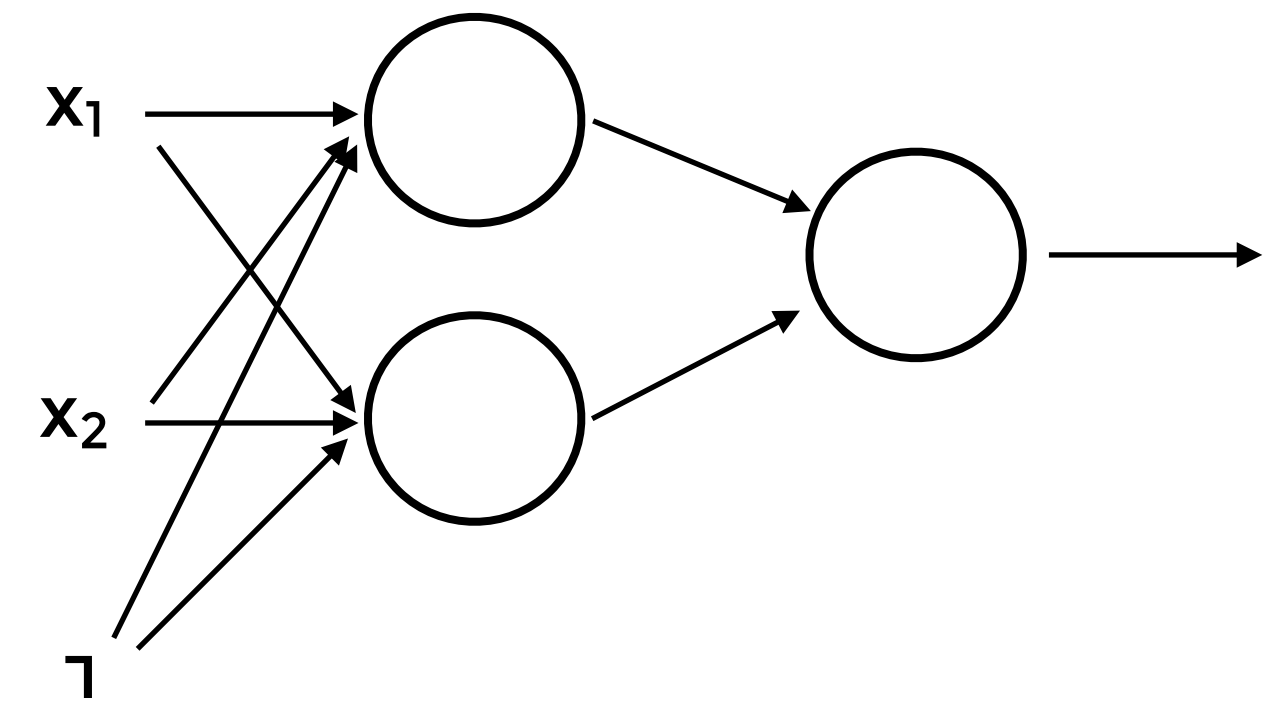
# Machine Learning I

## 60629A

**Summary**  
CNNs and RNNs  
— Week #5

# Neural Network models (architectures)

- **Feed-forward neural networks are standard**
  - **Input & Output: Fixed-length**
  - **Data is processed in parallel**



# Neural Network models (architectures)

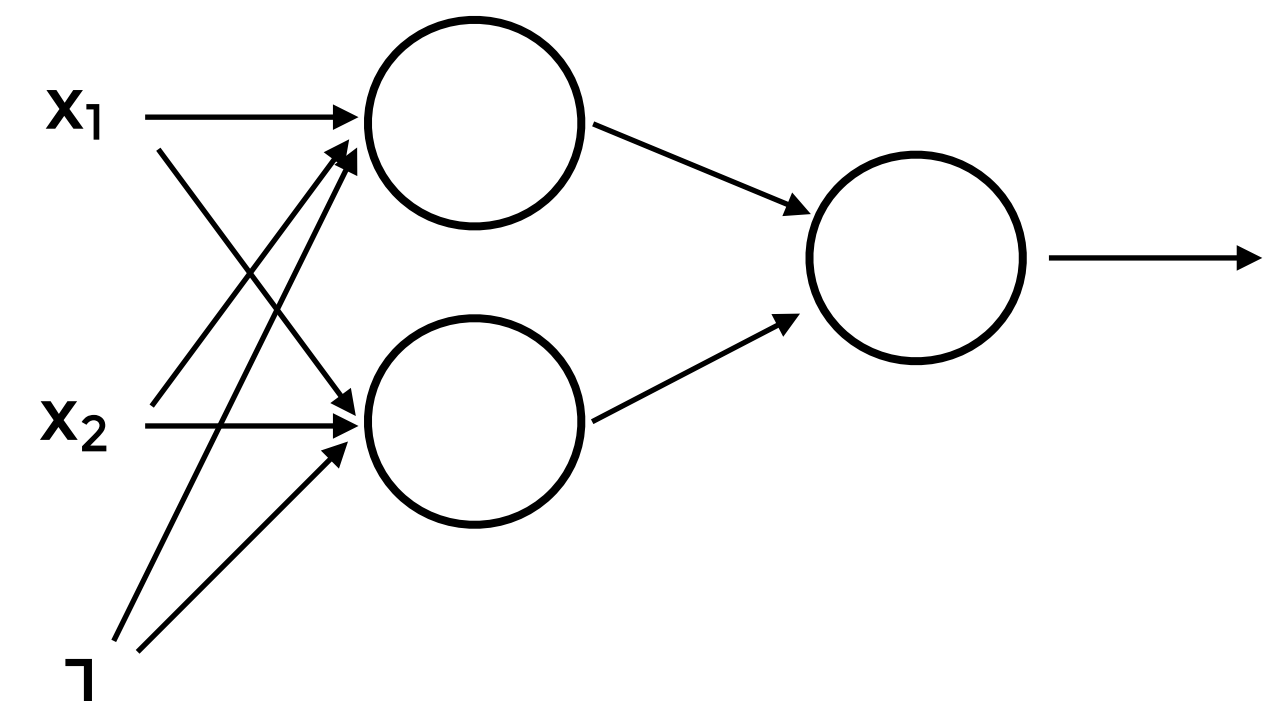
- **Feed-forward neural networks are standard**

- **Input & Output: Fixed-length**

- **Data is processed in parallel**

- **We can “specialize” neural networks**

- **Different data have different characteristics**



# Text Classification

```
From: bcash@crchh410.NoSubdomain.NoDomain (Brian Cash)
Subject: Re: free moral agency
Nntp-Posting-Host: crchh410
Organization: BNR, Inc.
Lines: 17

In article <735295730.25282@minster.york.ac.uk>, cjhs@minster.york.ac.uk writes:
|> : Are you saying that their was a physical Adam and Eve, and that all
|> : humans are direct decendents of only these two human beings.? Then who
|> : were Cain and Able's wives? Couldn't be their sisters, because A&E
|> : didn't have daughters. Were they non-humans?
|>
|> Genesis 5:4
|>
|> and the days of Adam after he begat Seth were eight hundred years, and
|> he begat sons and daughters:
|>
|> Felicitations -- Chris Ho-Stuart

Yeah, but these were not the wives. The wives came from Nod, apparently
a land being developed by another set of gods.

Brian /-|-\
```



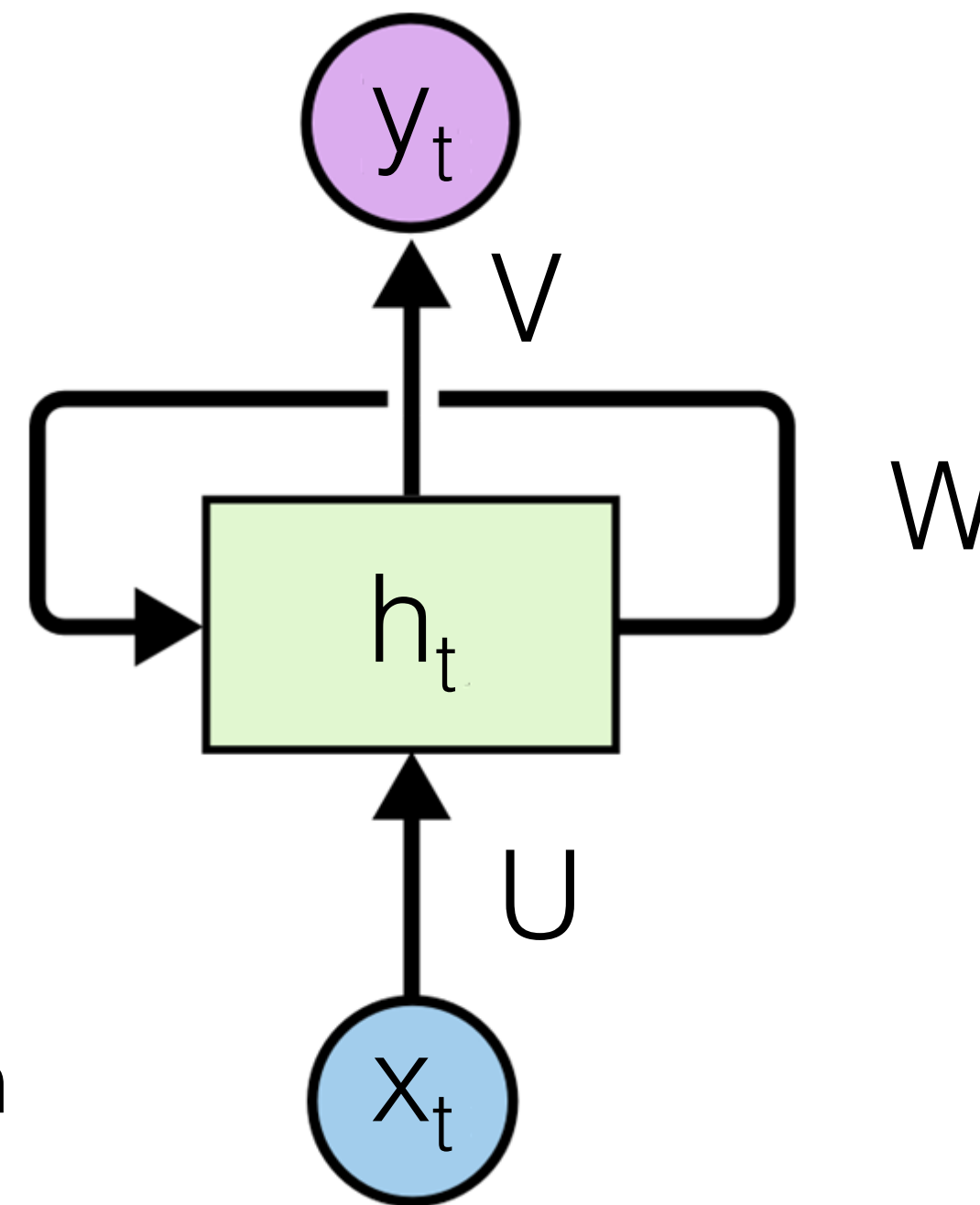
SPAM / HAM



# Basic idea (with parameters)

## Process through time (t)

- **U, V, W: parameters**
- **Shared through time**
- **Simplest parametrization**

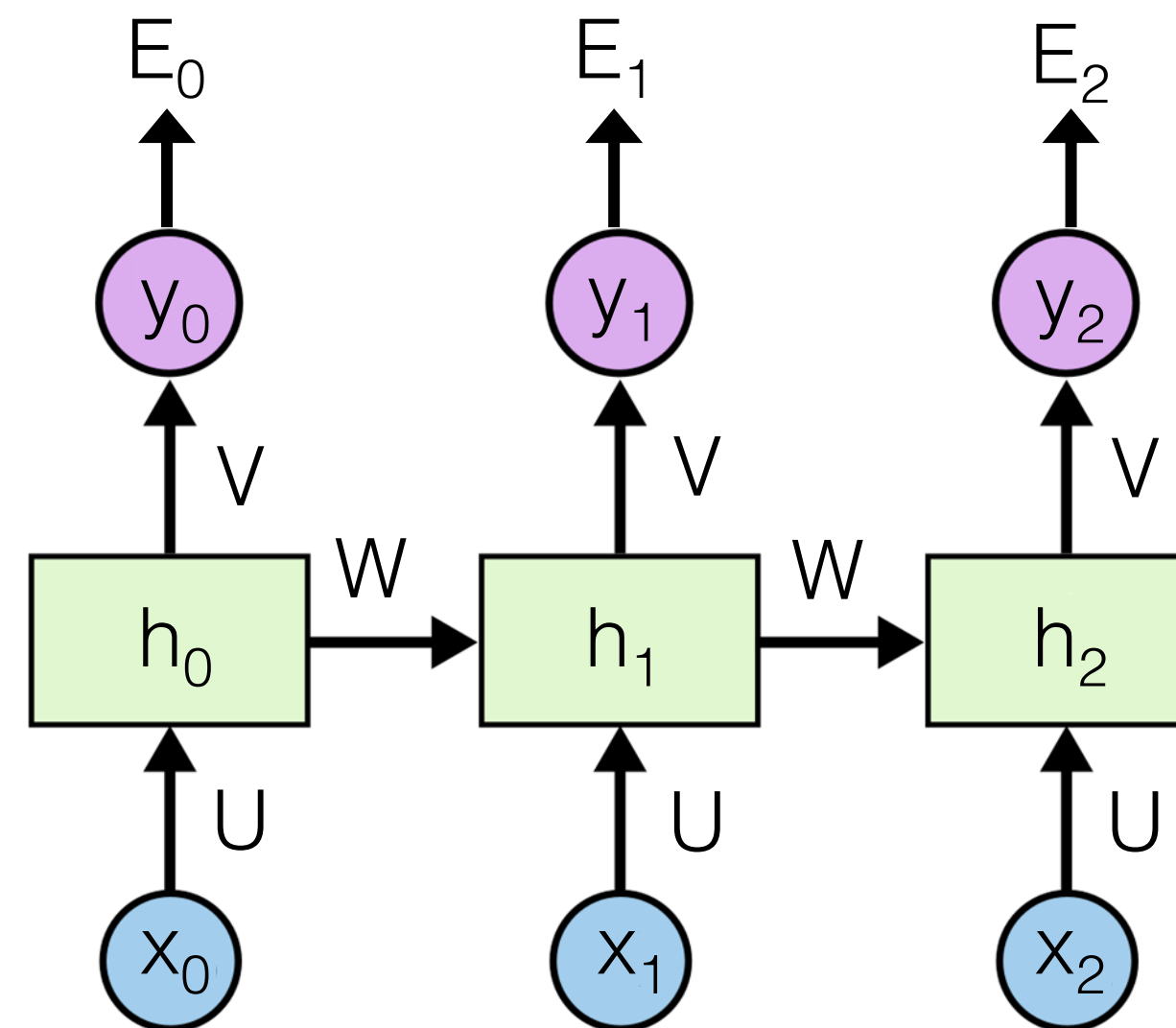


$$h_t = \tanh(Ux_t + Wh_{t-1})$$

$$y_t = f(Vh_t)$$

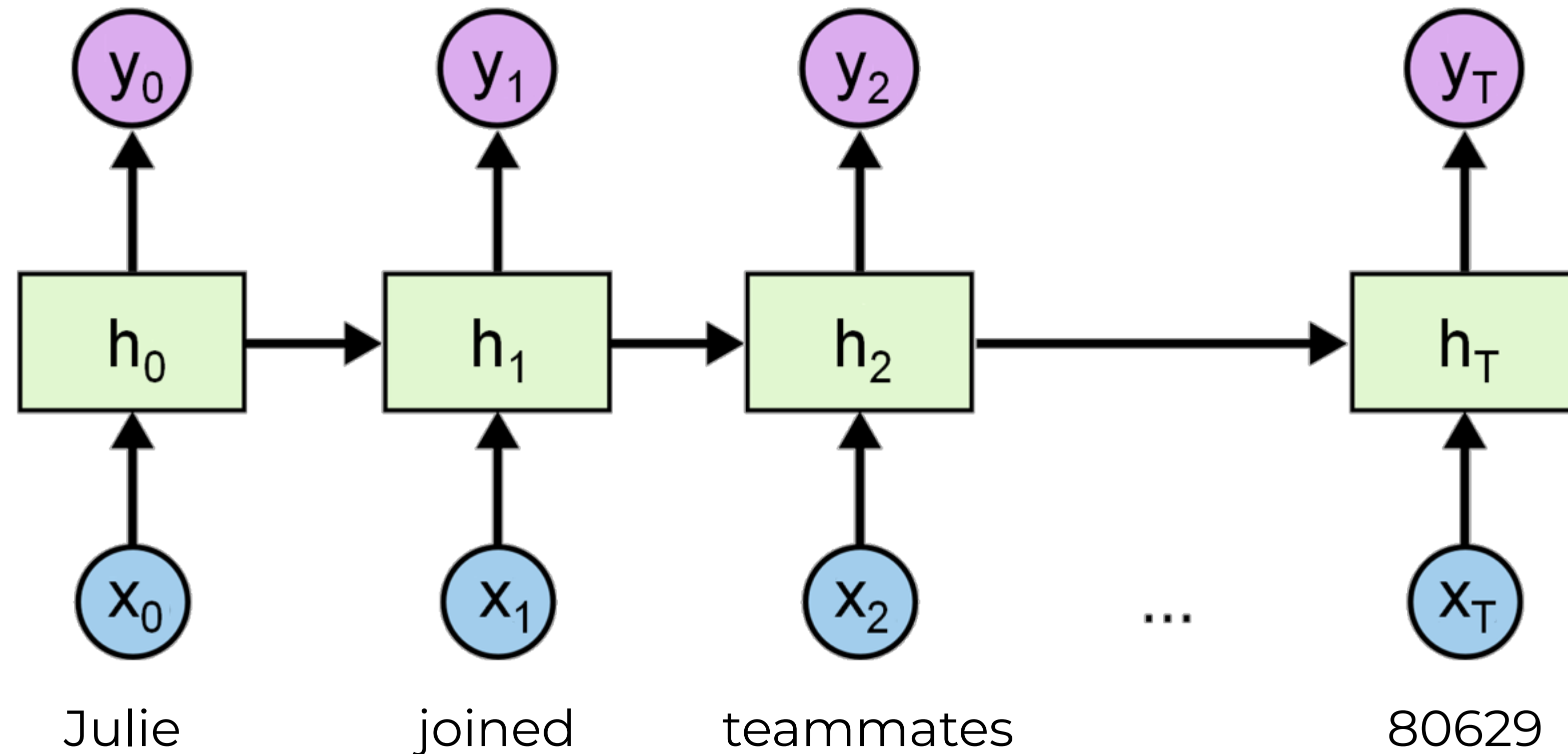
# Training RNNs

- **Gradient descent from the loss**  $E = \sum_t (y_t - \hat{y}_t)^2$
- **Following the structure the gradient is back propagated through time**



# Limitations

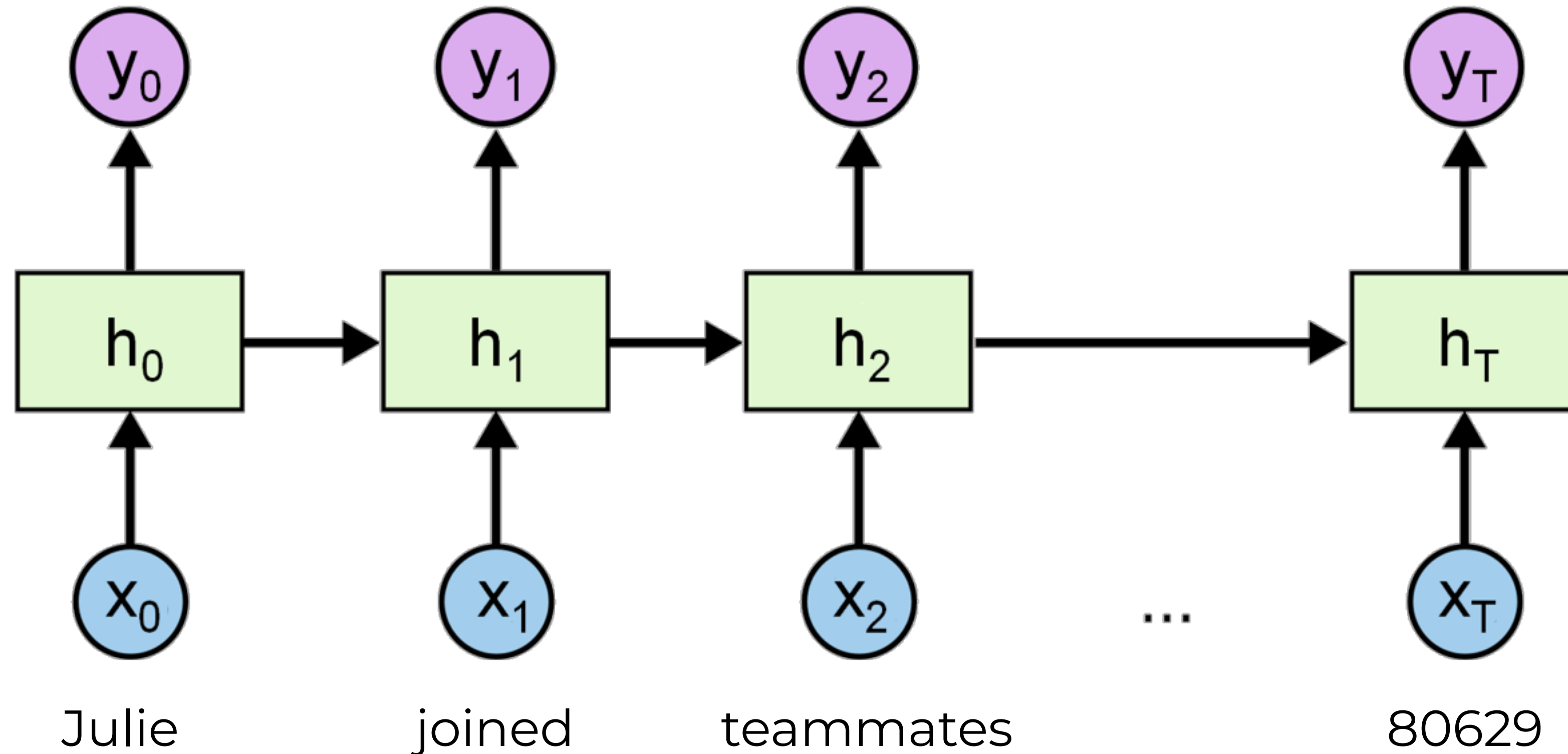
- Long-term dependencies are difficult to learn



# Limitations

- Long-term dependencies are difficult to learn

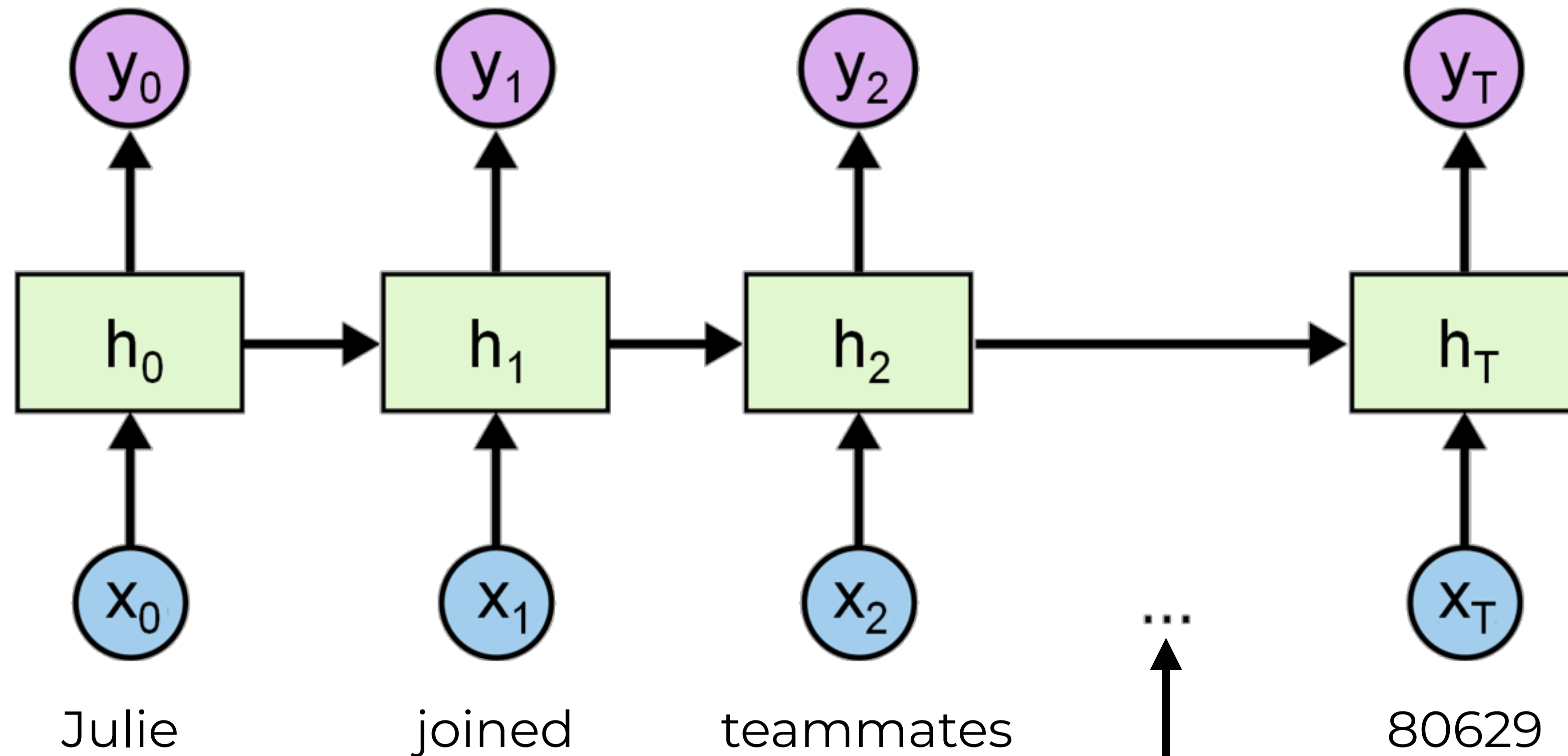
Q: Who joined the class?



# Limitations

- Long-term dependencies are difficult to learn

Q: Who joined the class?



(and colleagues for the start of a new ML class)

Training.  
Given previous words,  
predict the next word

Target

guardian

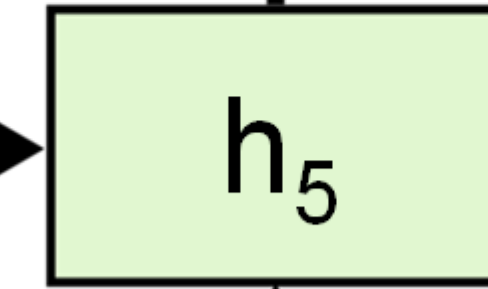
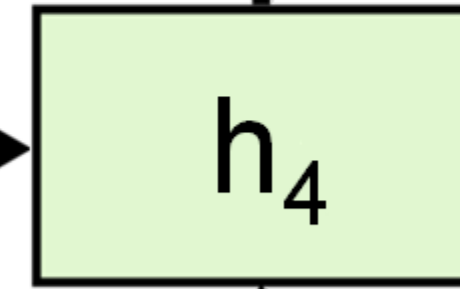
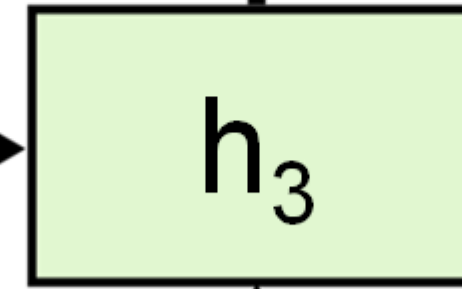
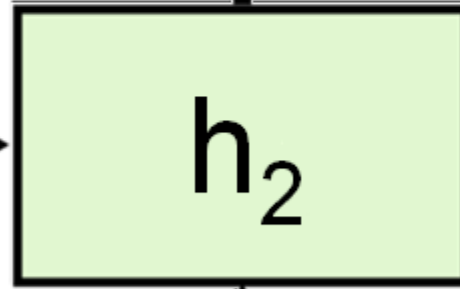
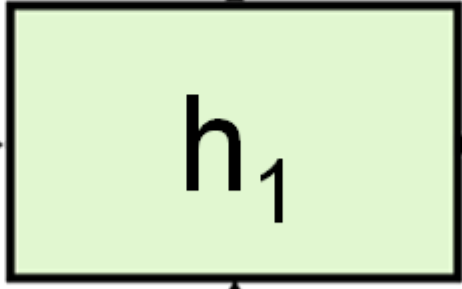
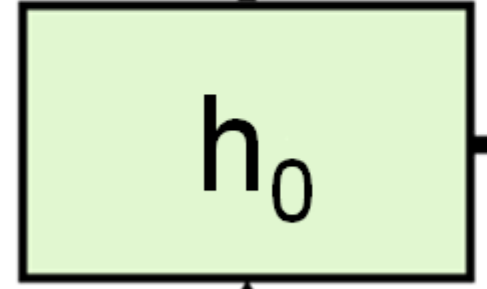
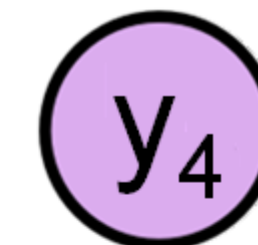
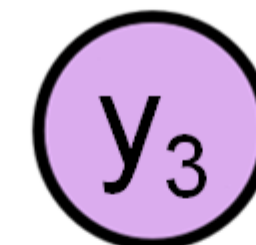
of

the

land

of

an



Input

The

guardian

of

the

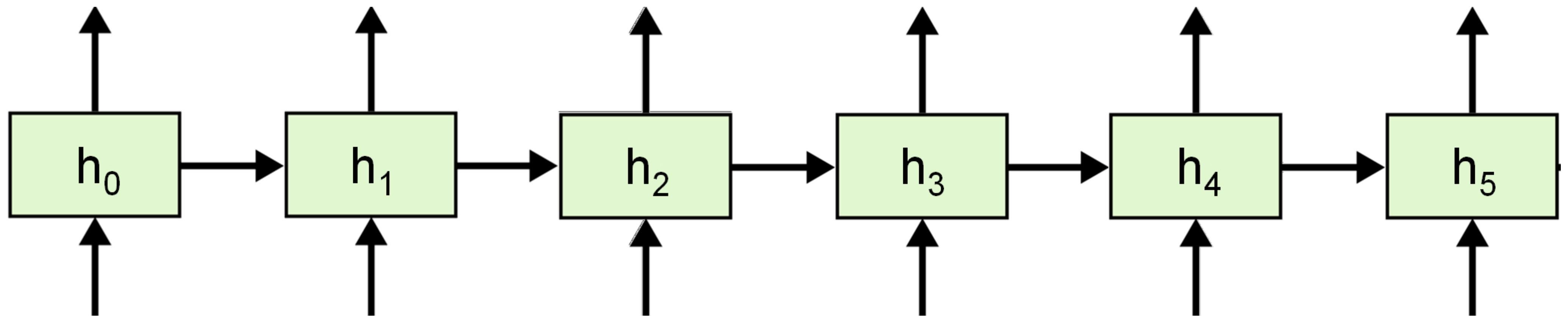
land

of

Test.

1. Predict One word at a time.
2. Feed the prediction back to the model

Prediction

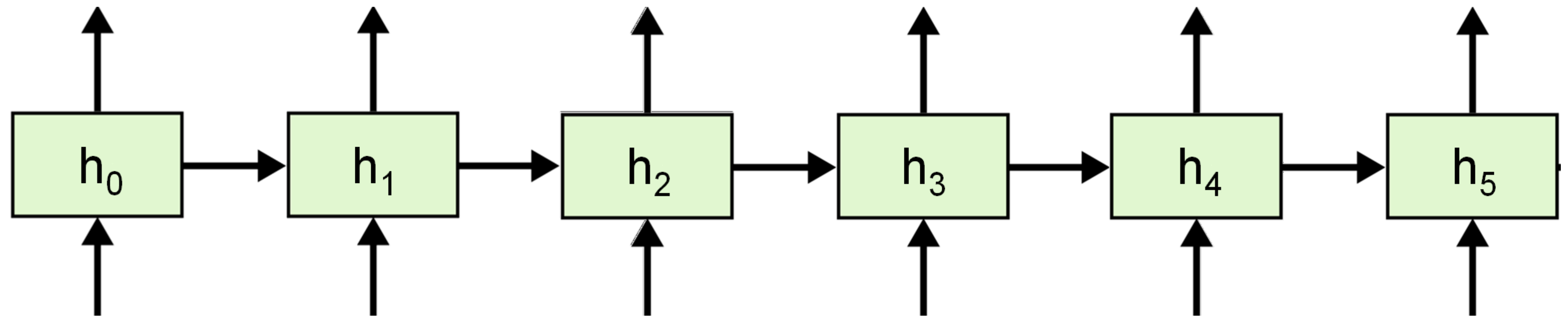


Input

Test.

1. Predict One word at a time.
2. Feed the prediction back to the model

Prediction



Input

The

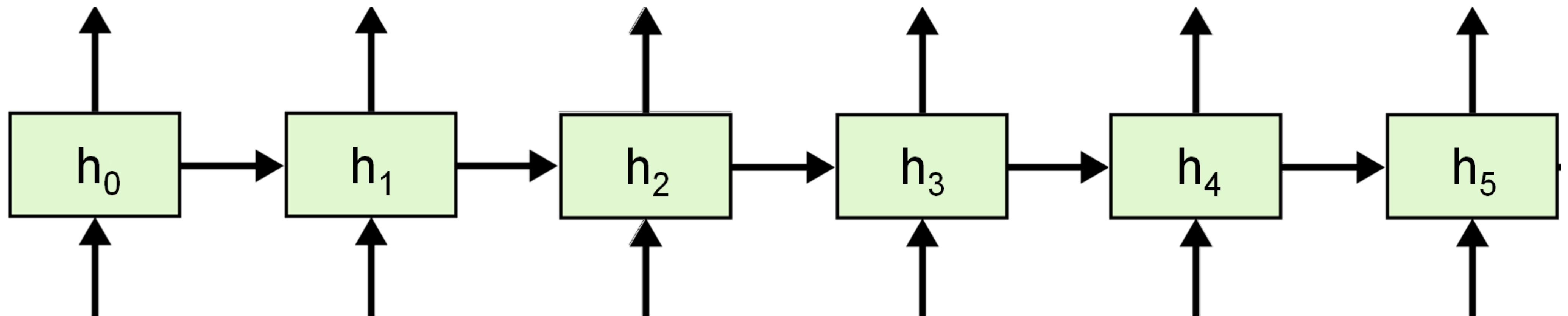


Test.

1. Predict One word at a time.
2. Feed the prediction back to the model

Prediction

woman



Input

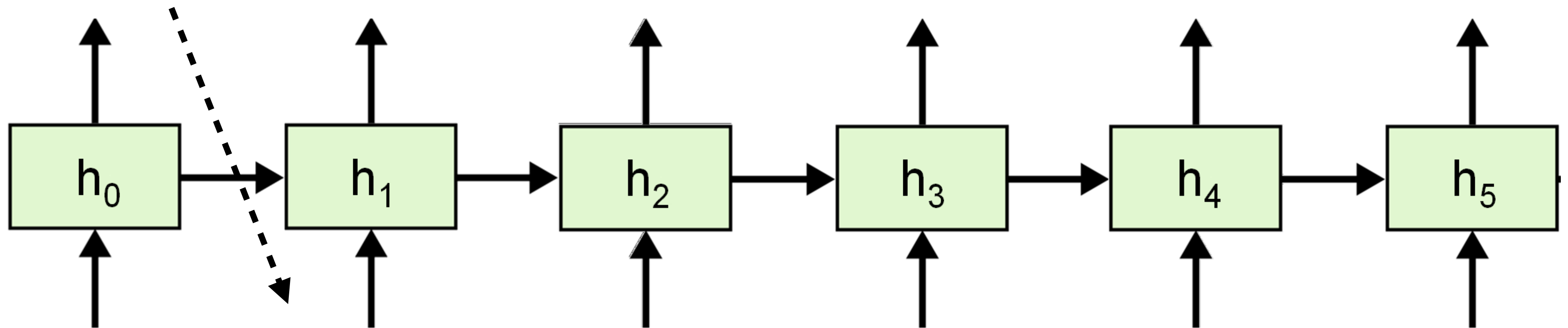
The

Test.

1. Predict One word at a time.
2. Feed the prediction back to the model

Prediction

woman



Input

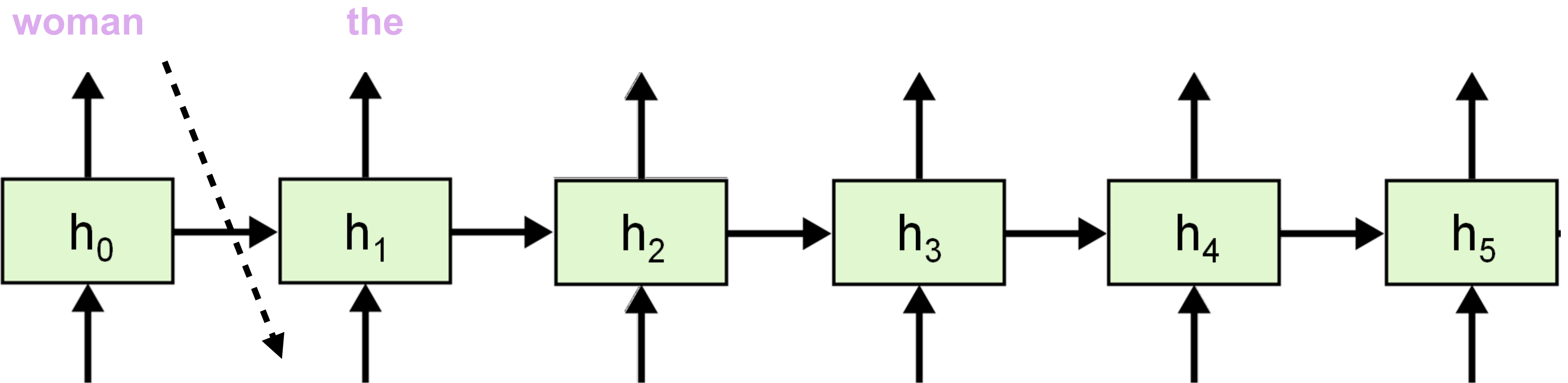
The

woman

Test.

1. Predict One word at a time.
2. Feed the prediction back to the model

Prediction



Input

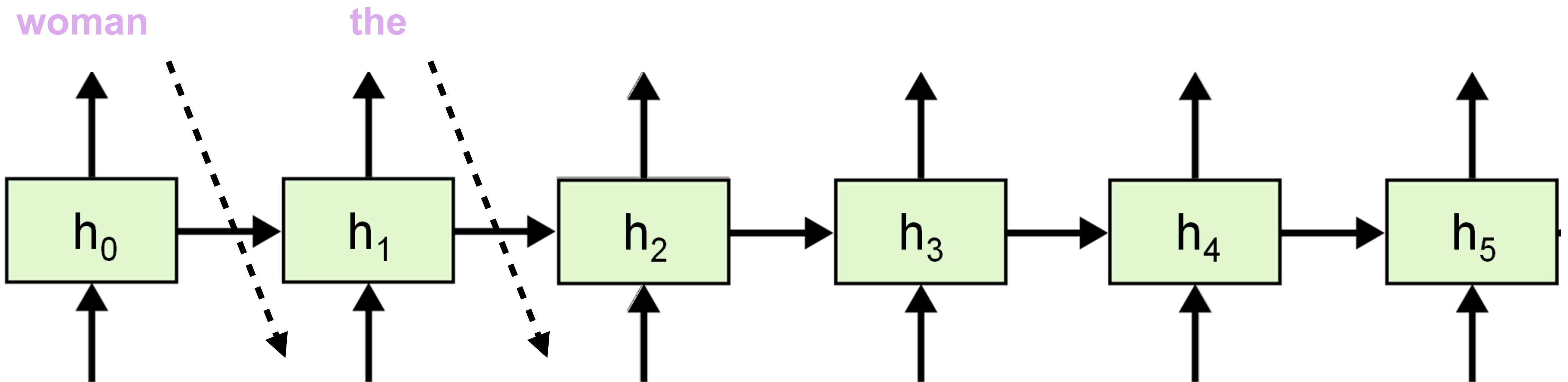
The

woman

Test.

1. Predict One word at a time.
2. Feed the prediction back to the model

Prediction



Input

The

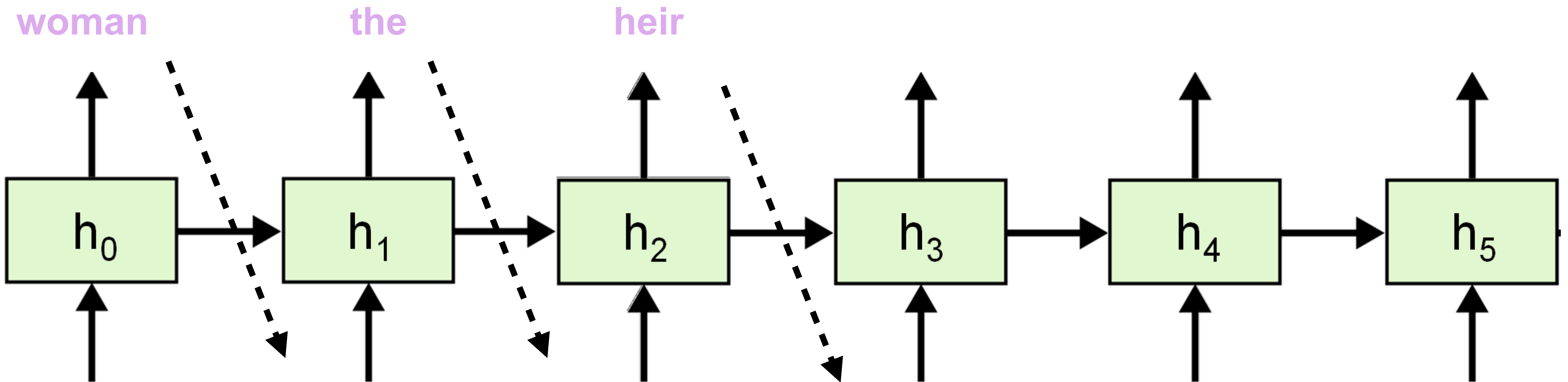
woman

the

Test.

1. Predict One word at a time.
2. Feed the prediction back to the model

Prediction



Input

The

woman

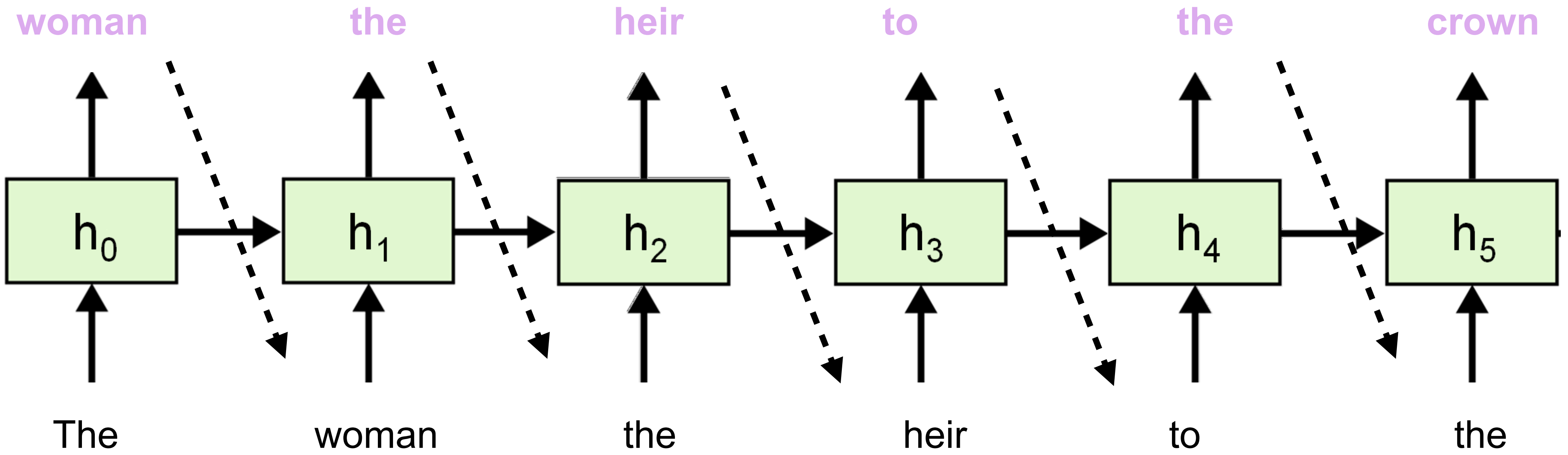
the

heir

Test.

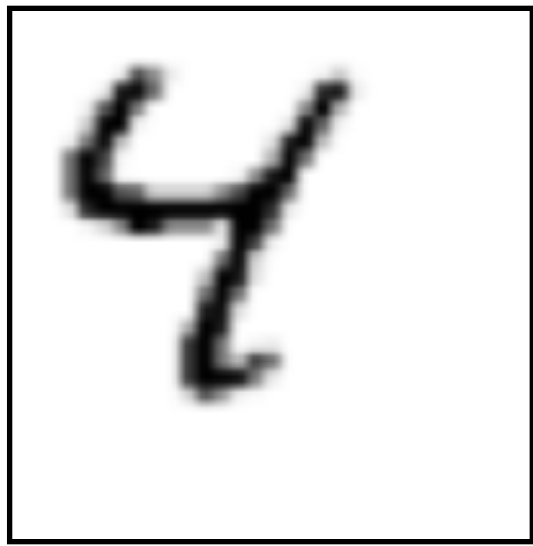
1. Predict One word at a time.
2. Feed the prediction back to the model

Prediction

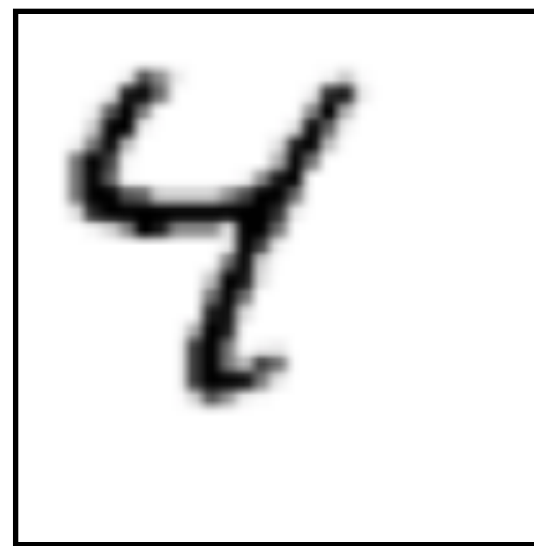


Input

Image



Image



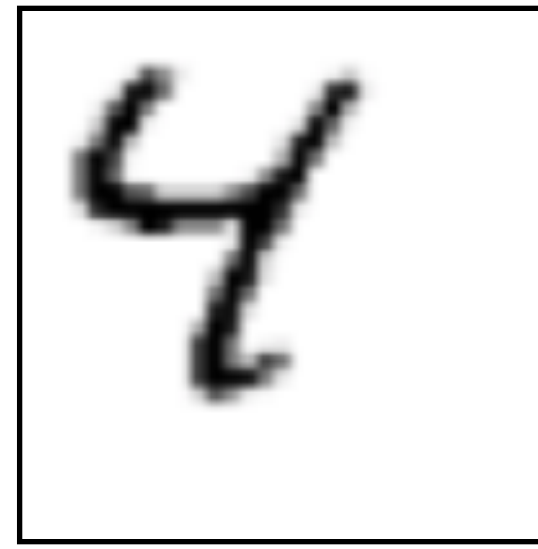
Filter (Kernel)



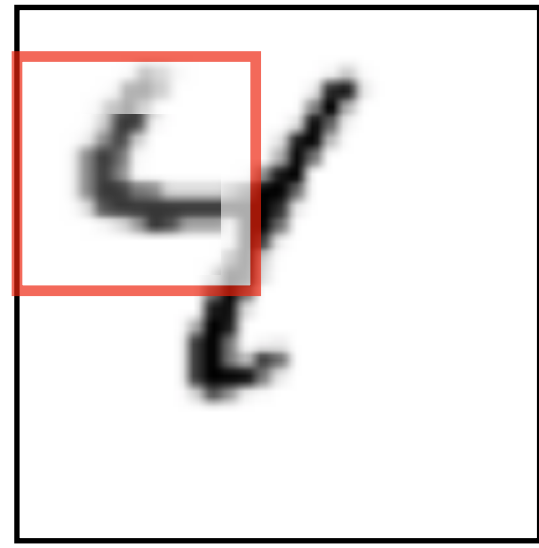
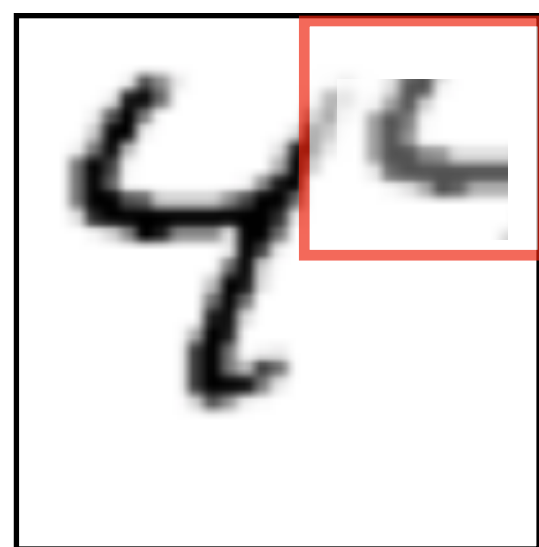
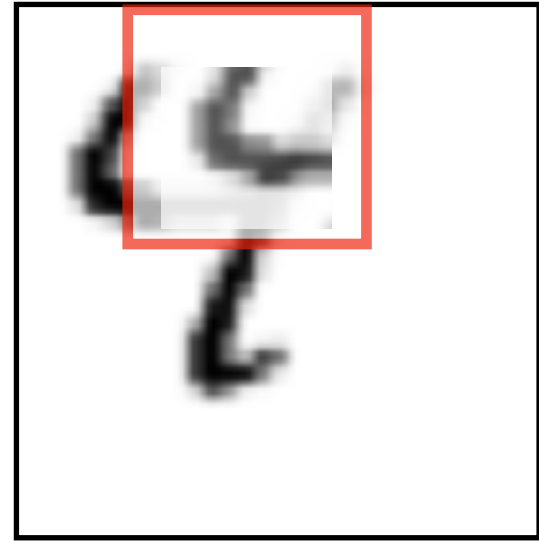
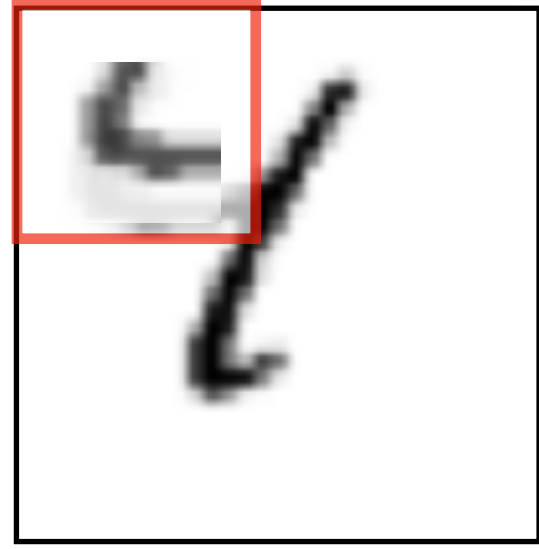


Pass the filter over the image  
(Convolutions)

Image

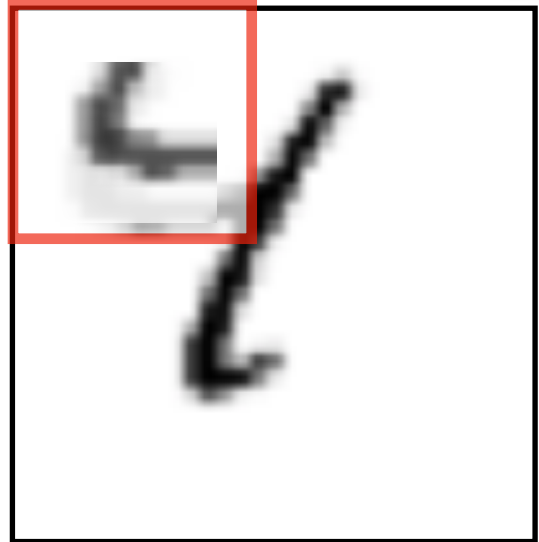
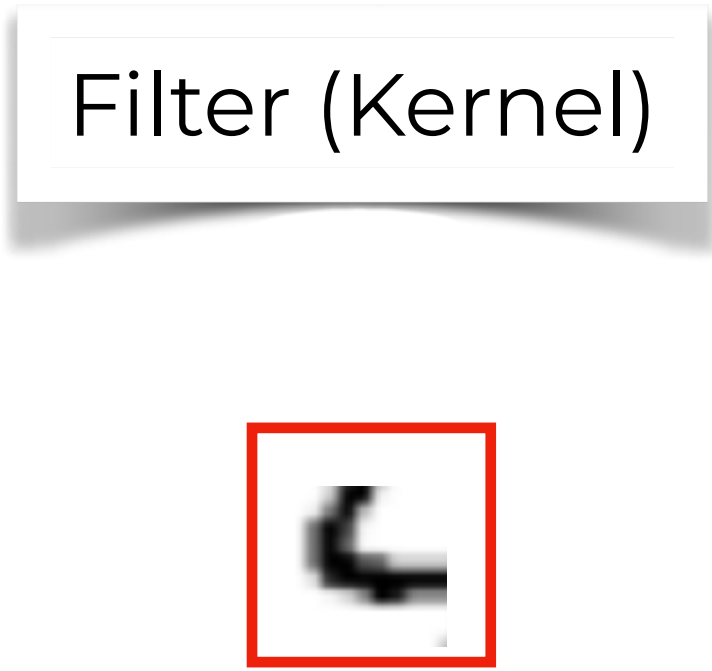
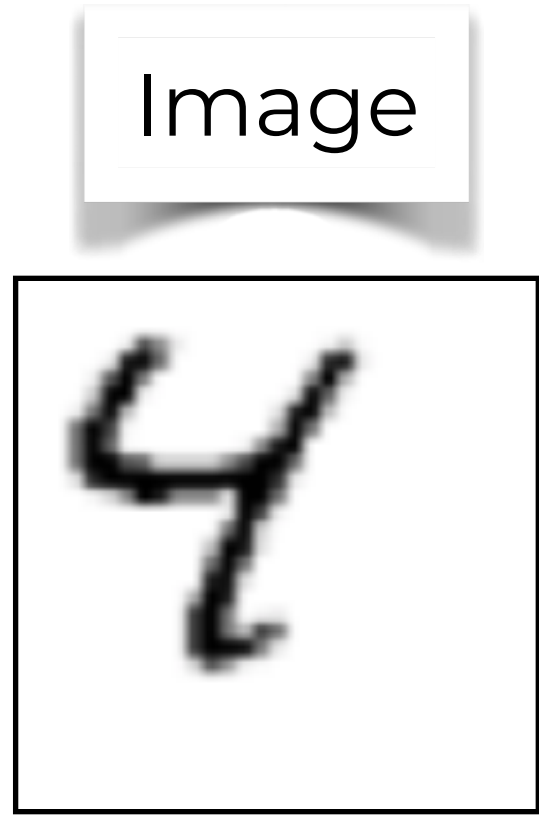


Filter (Kernel)

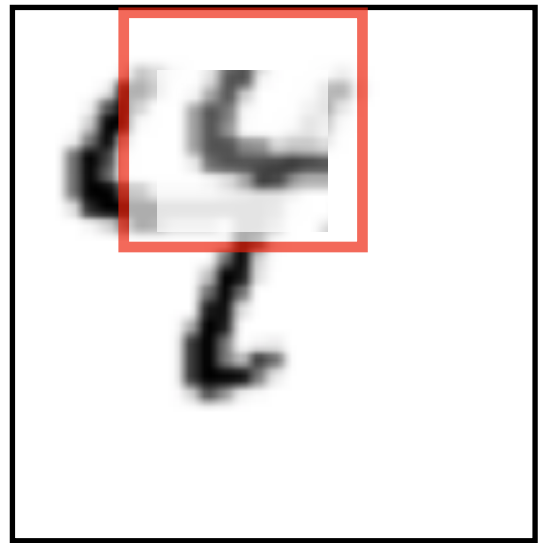


...

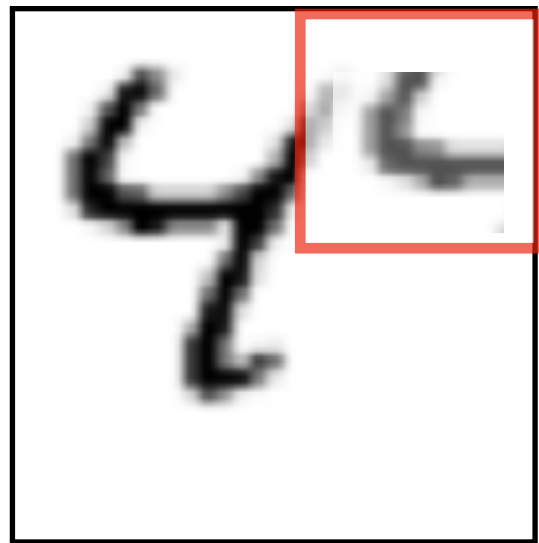
Pass the filter over the image  
(Convolutions)



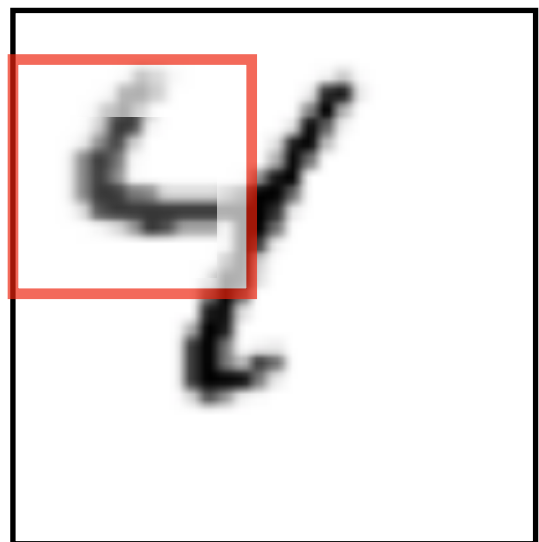
No detection  
Output: Low



No detection  
Output: Low



No detection  
Output: Low

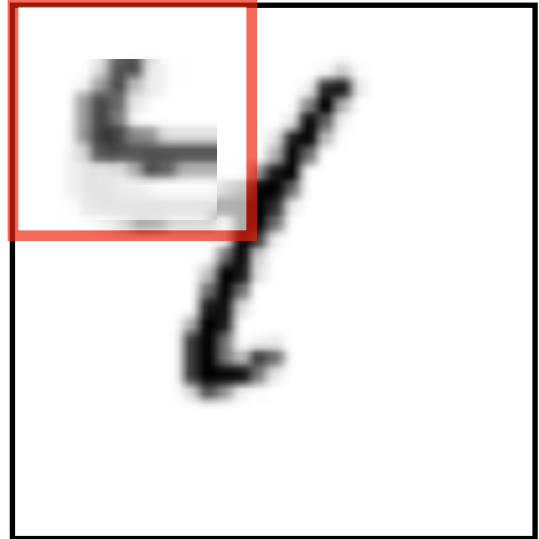
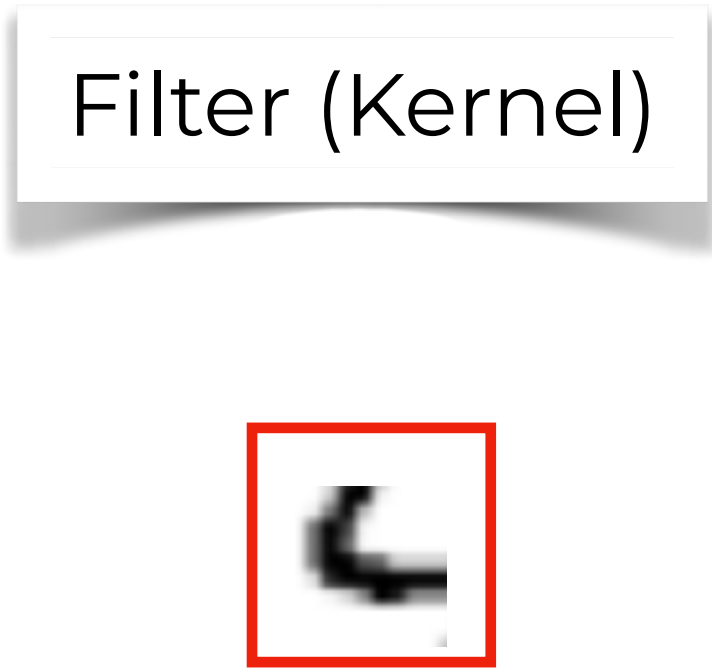
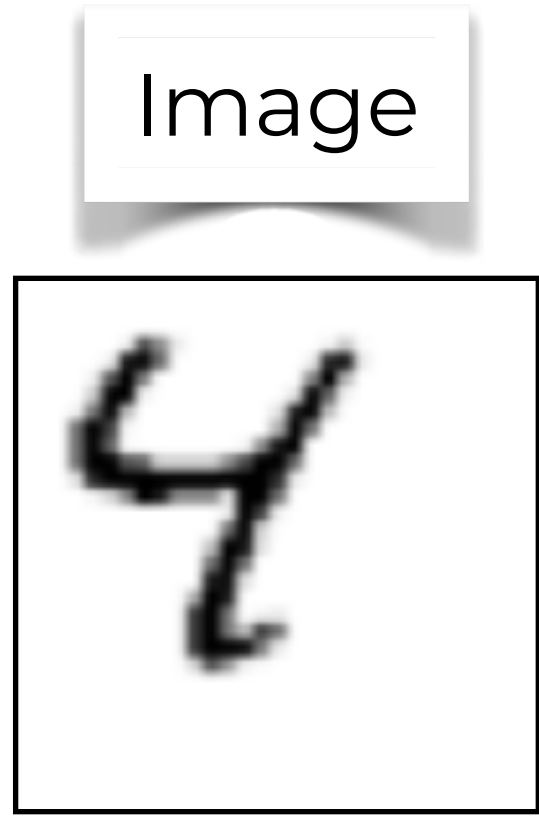


**Detection**  
**Output: High**

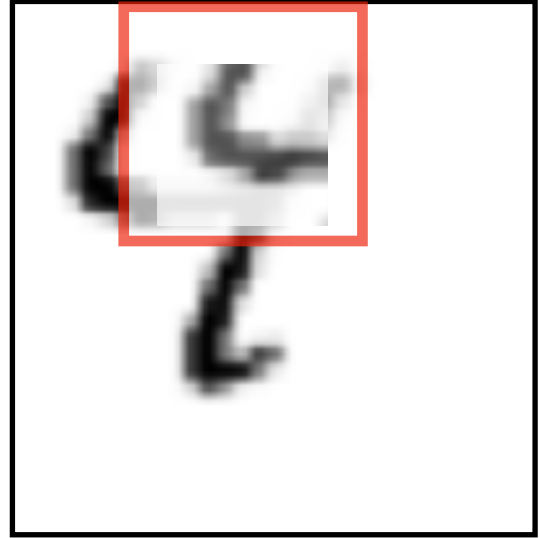
...

Pass the filter over the image  
(Convolutions)

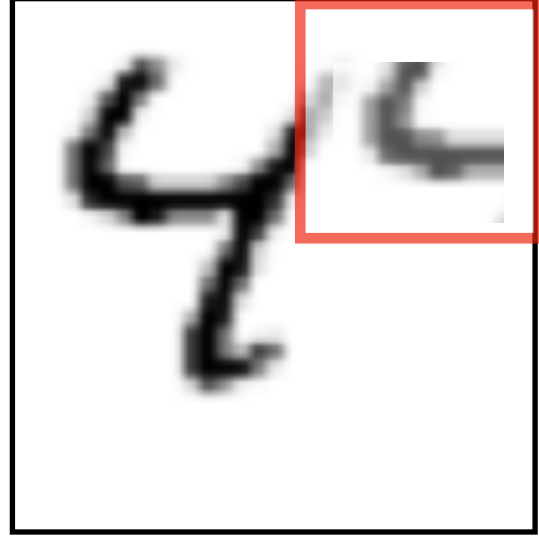
Aggregate the output of  
the filter  
(Pooling)



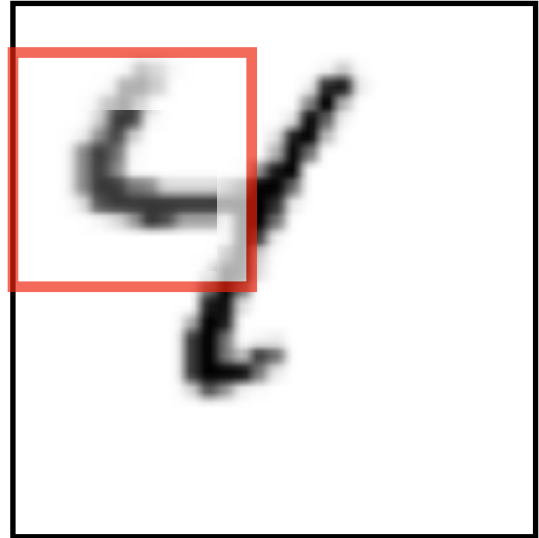
No detection  
Output: Low



No detection  
Output: Low



No detection  
Output: Low

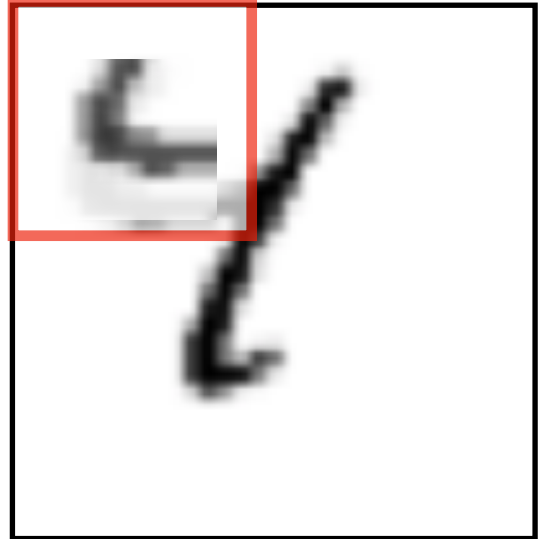
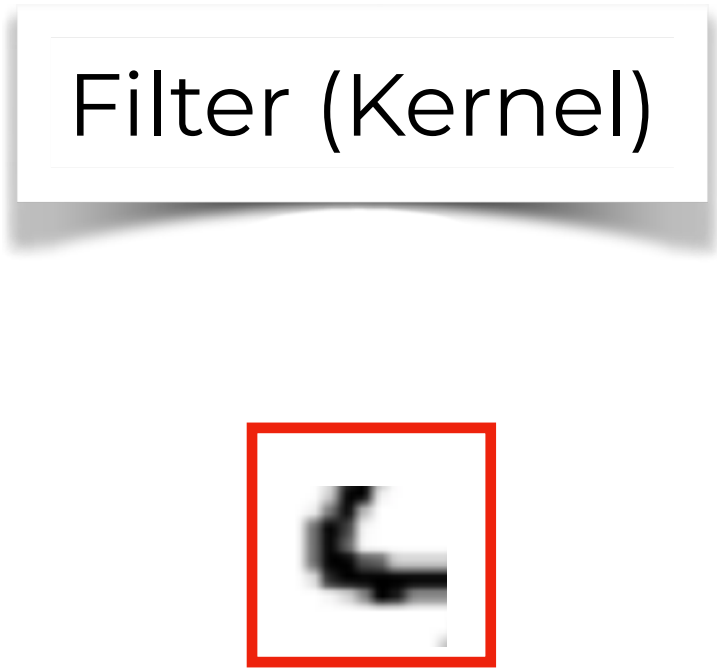
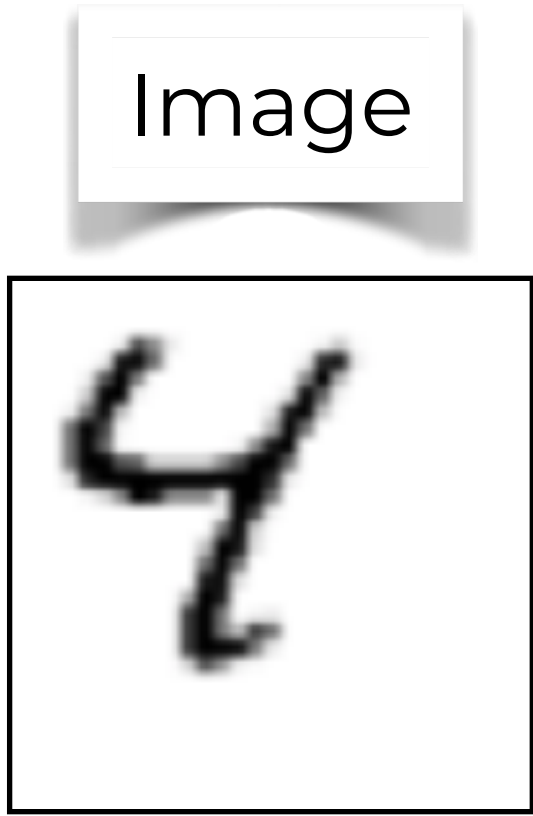


**Detection**  
**Output: High**

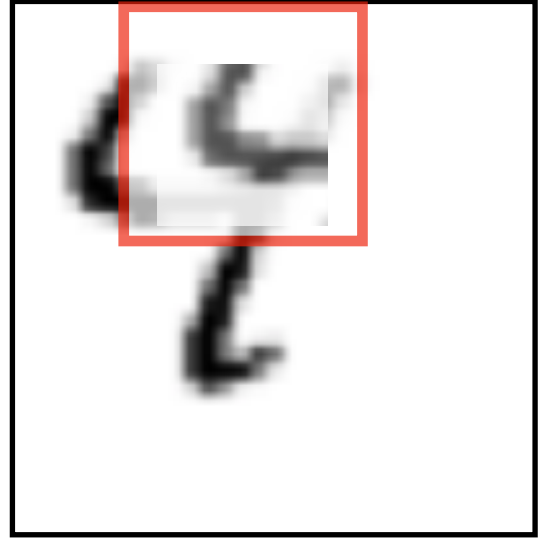
**One of the filters**  
**detected an object**  
**part**  
**Output: High**

Pass the filter over the image  
(Convolutions)

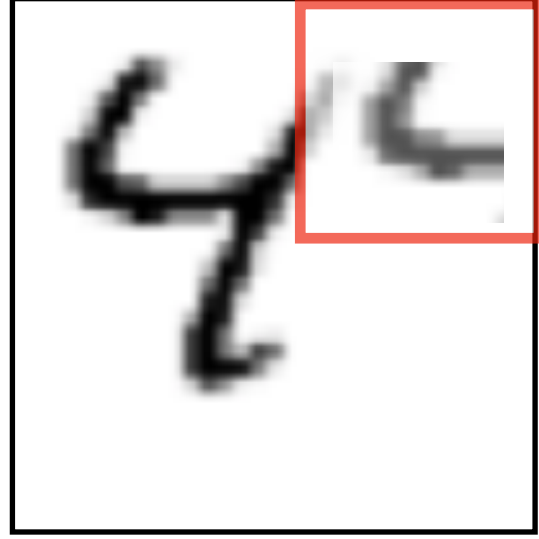
Aggregate the output of  
the filter  
(Pooling)



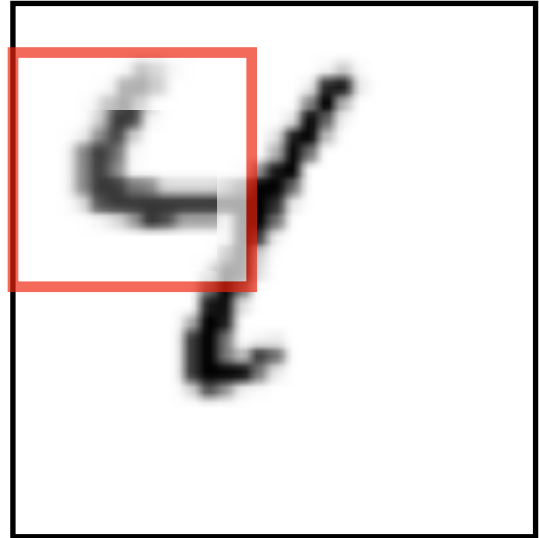
No detection  
Output: Low



No detection  
Output: Low



No detection  
Output: Low



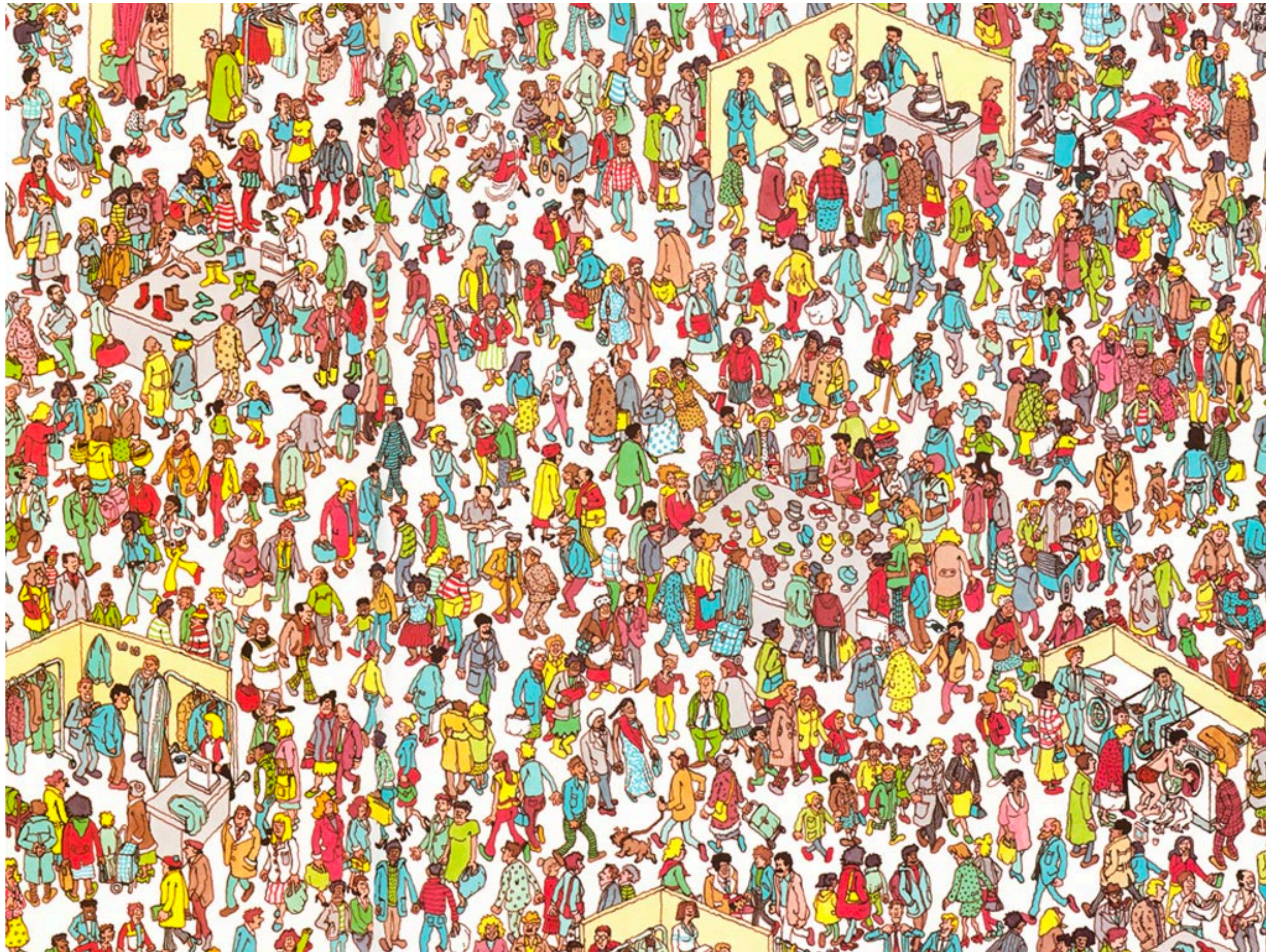
**Detection**  
**Output: High**

...

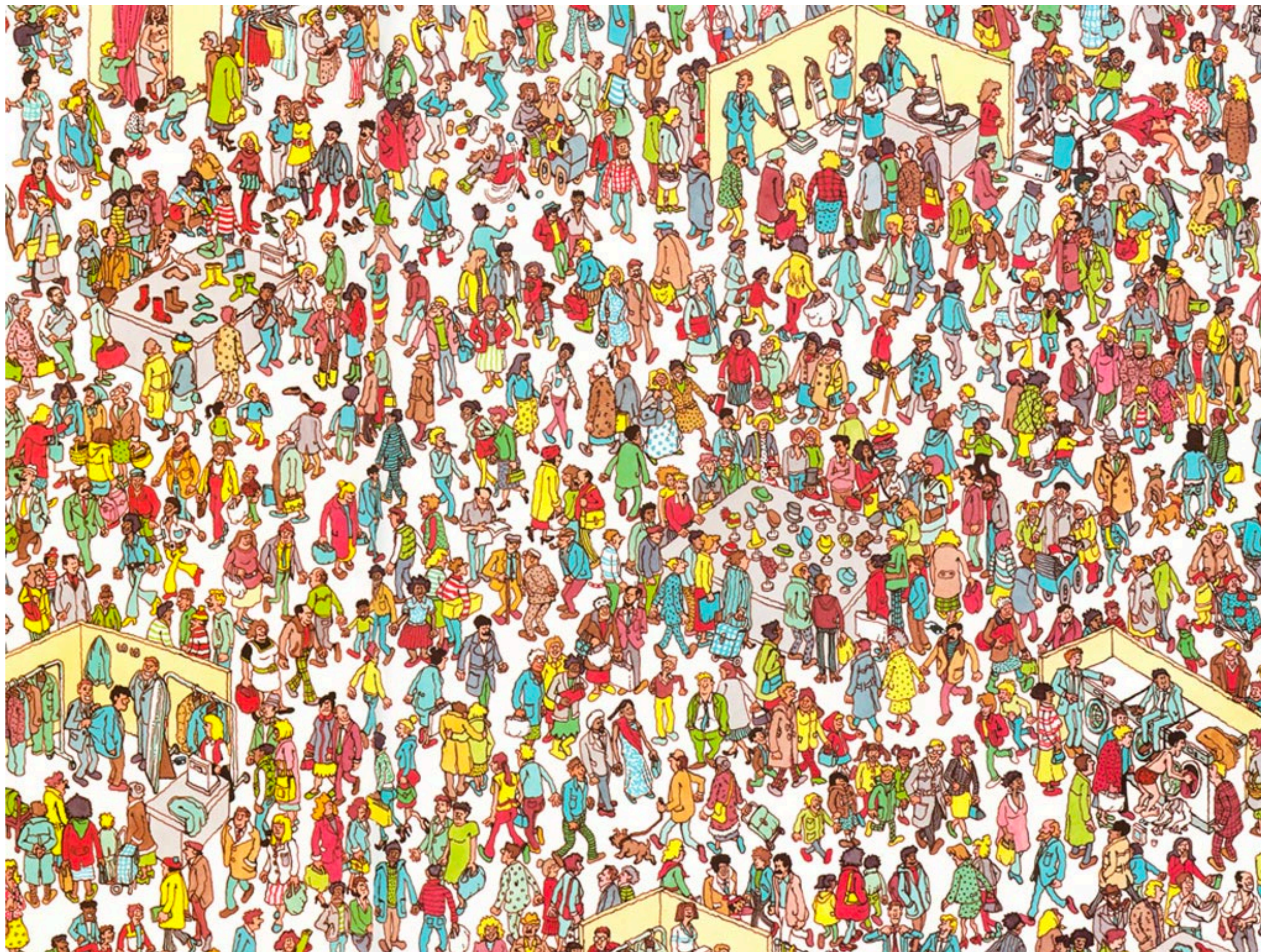
**One of the filters  
detected an object  
part  
Output: High**

To do well you need to:  
1) learn the filters;  
2) use many filters.



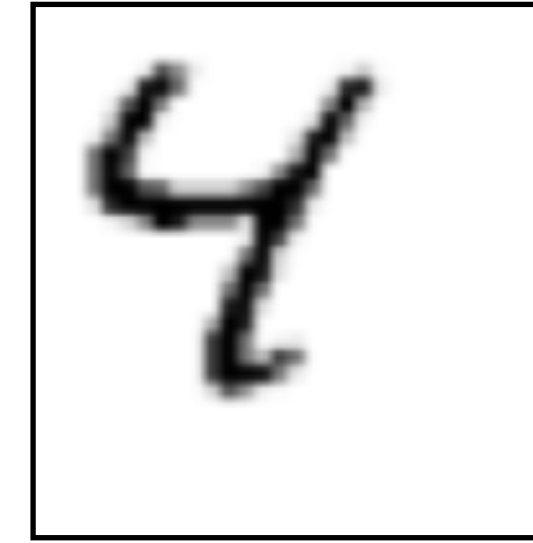








# Convolutions (to the rescue)



For pixel  $(i,j)$ :

$$S(i,j) = (K * I)(i,j) = \sum_m^{k. \text{ width}} \sum_n^{k. \text{ height}} I(i+m, j+n) K(m,n)$$

Kernel  $\nearrow$   $\nwarrow$  Input



- Dot product between “the kernel and the region”

# Convolutions (to the rescue)

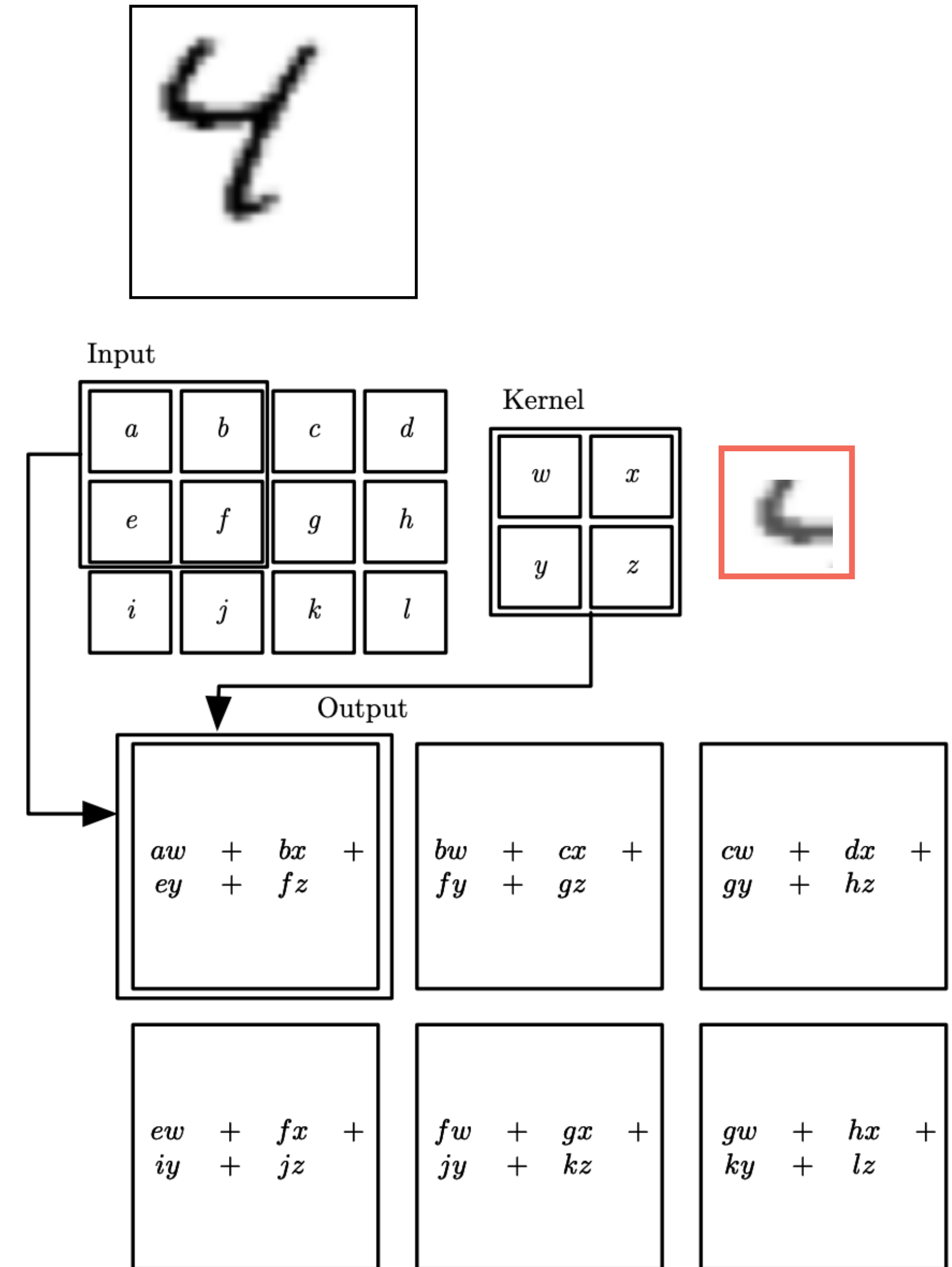
For pixel (i,j):

$$S(i,j) = (K * I)(i,j) = \sum_m^{k. \text{ width}} \sum_n^{k. \text{ height}} I(i+m, j+n) K(m,n)$$

Kernel

Input

- Dot product between “the kernel and the region”



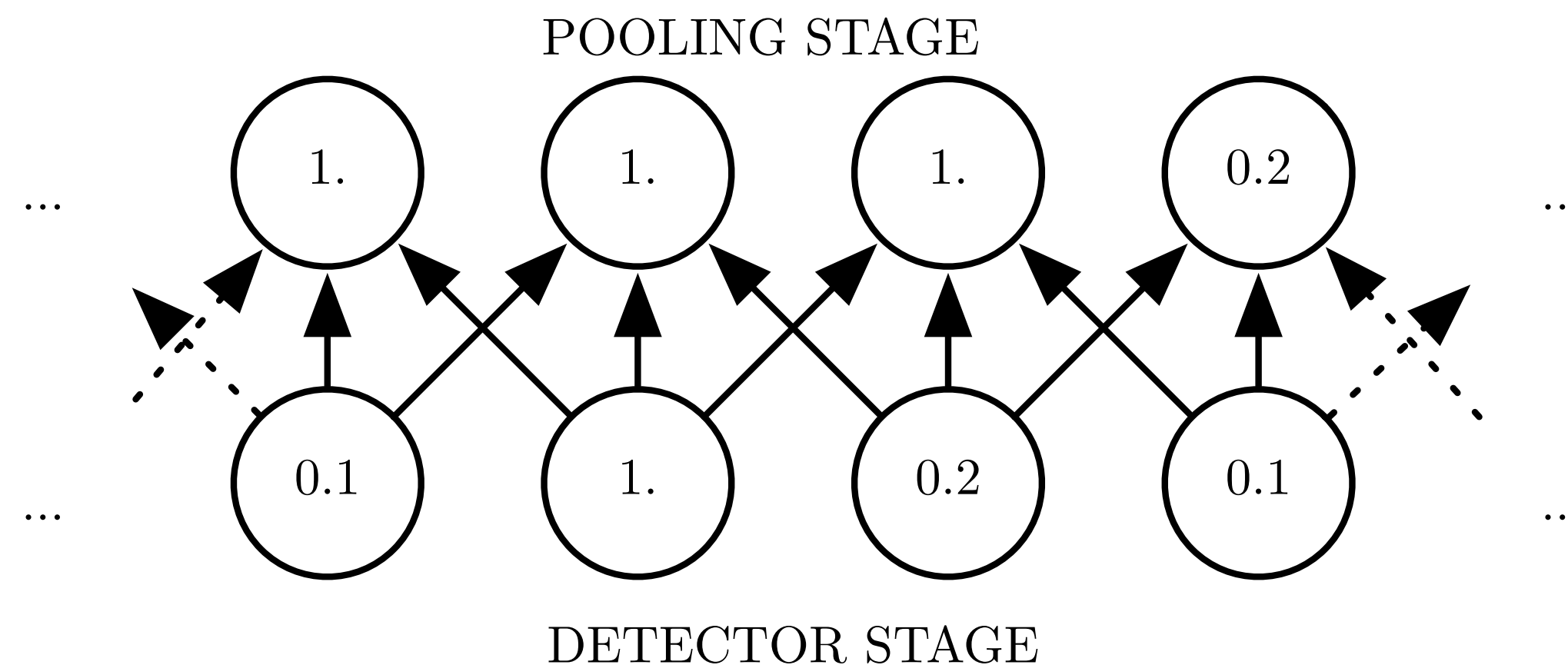


# Pooling

- **Make the representation invariant to small translations in the input**
- **“Pool” the value of neighbour units**
- **E.g., max-pooling takes the max from its input.**

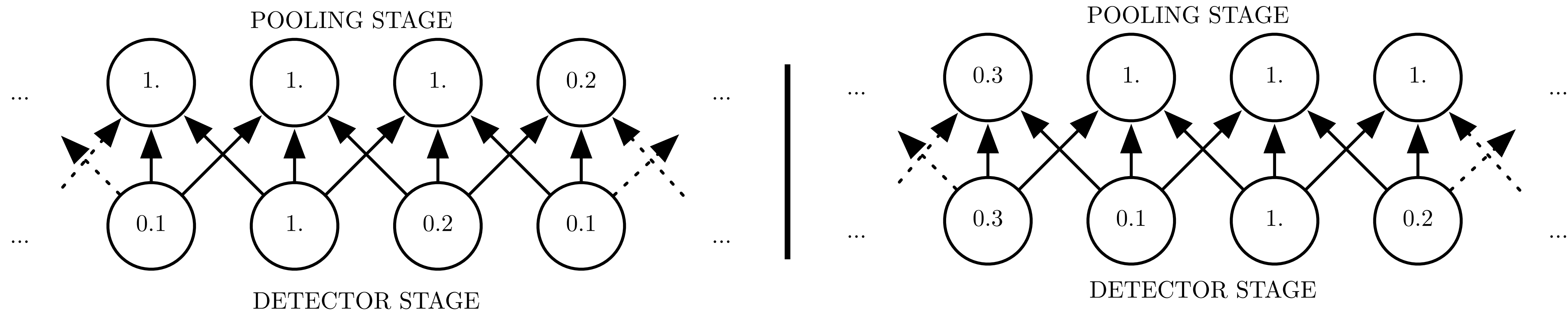
# Pooling

- **Make the representation invariant to small translations in the input**
- **“Pool” the value of neighbour units**
- **E.g., max-pooling takes the max from its input.**

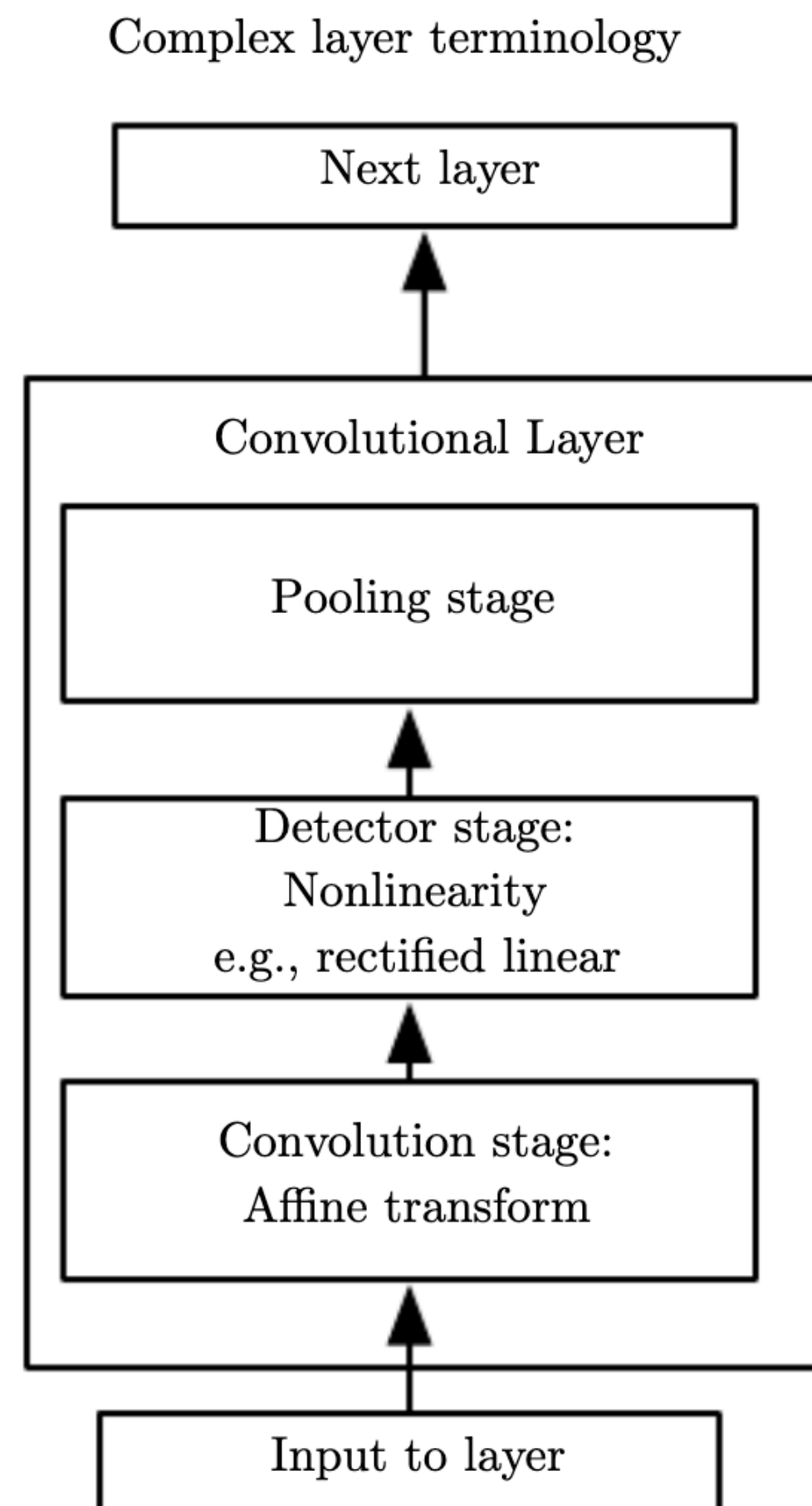


# Pooling

- **Make the representation invariant to small translations in the input**
- **“Pool” the value of neighbour units**
- **E.g., max-pooling takes the max from its input.**

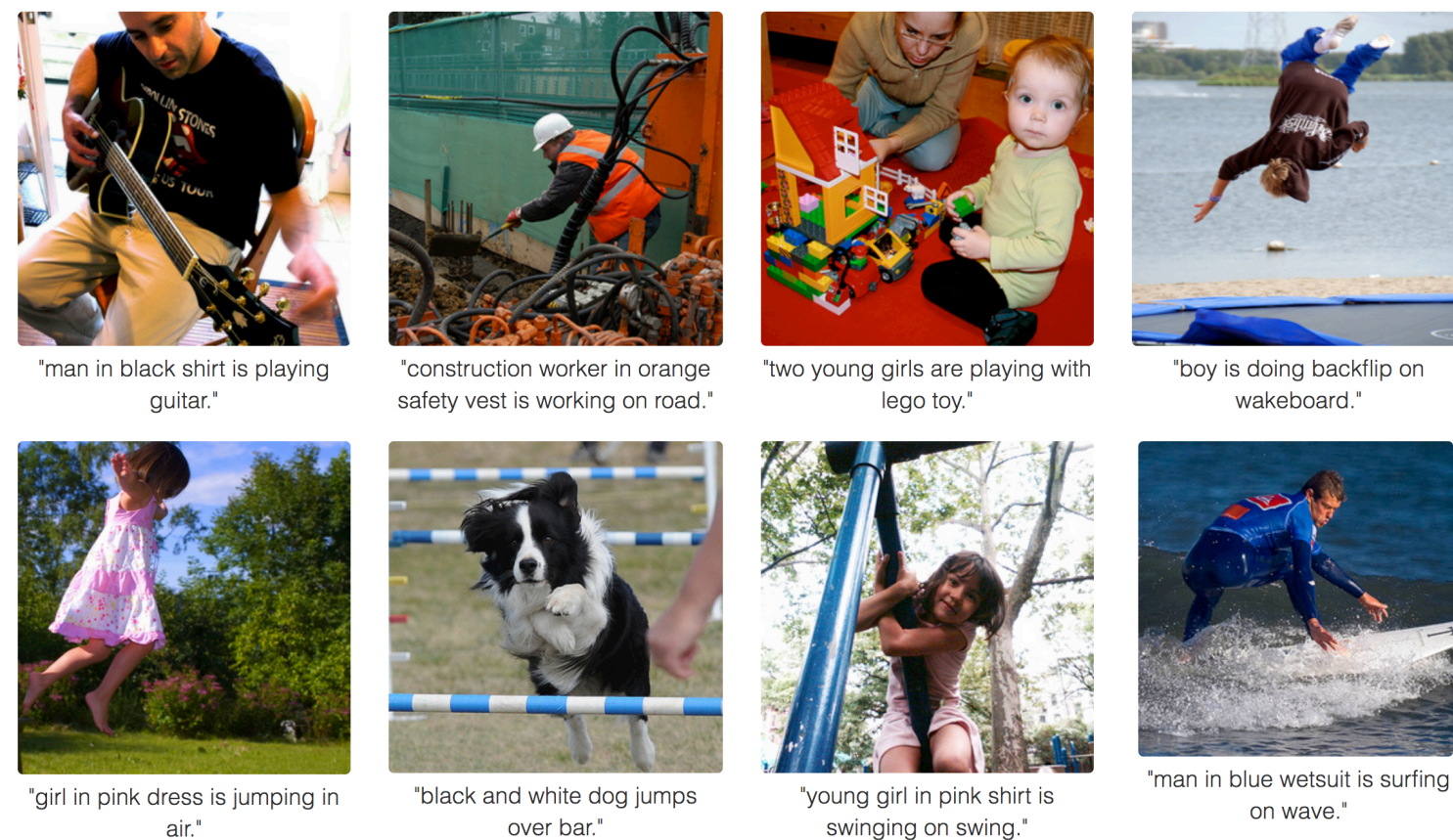


# Putting it all together

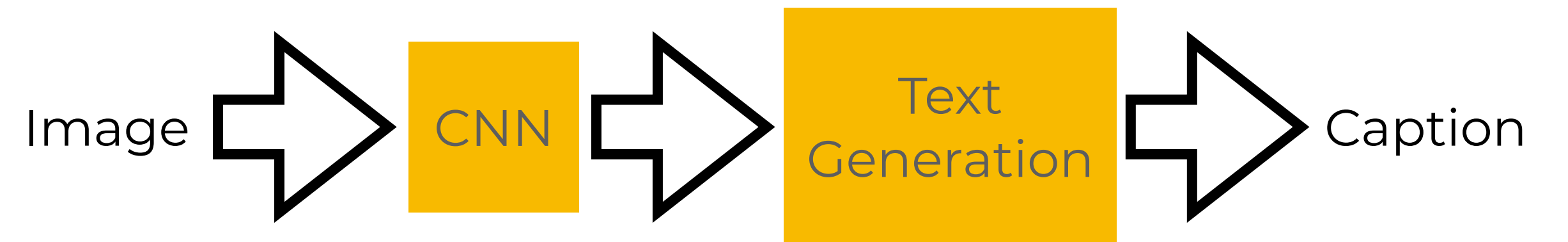


# CNNs can be used as modules inside larger networks

- **Image Captioning**



[<https://cs.stanford.edu/people/karpathy/cvpr2015.pdf>]



- **Visual Question Answering**

Is the umbrella upside down?



[<https://arxiv.org/pdf/1612.00837.pdf>]

