

Machine Learning for Large-Scale Data Analysis and Decision Making (MATH80629A) Winter 2022

Week #6 - Summary

Announcement

- **Study plan:** due **February 27, 2022**
- **Homework 1:** due **March 2, 2022**

Today

- **Fourth Quiz** on Gradescope!
- Summary of Recurrent Neural networks and Convolutional neural networks
- Q&A
- Hands-on session



QUIZ TIME

A colorful, three-dimensional sign with a glowing effect. The letters are outlined in black and filled with a gradient of colors: teal, yellow, red, orange, and green. Small yellow lights are visible along the edges of the letters.

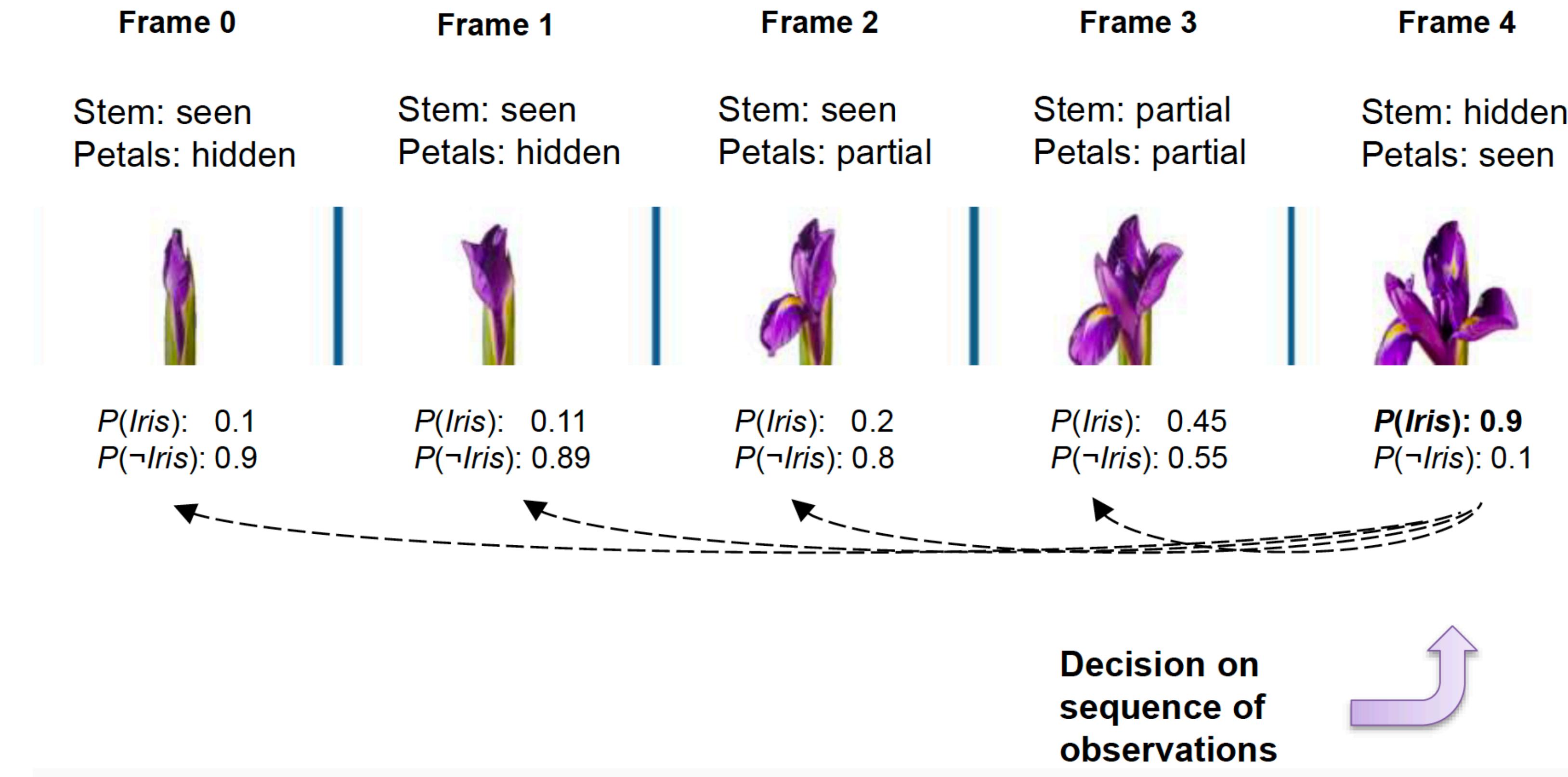
Quiz 3

Login to your Gradescope account

RNNs

Temporal dependencies

Analyzing temporal dependencies → Improved decisions



Sequential Data

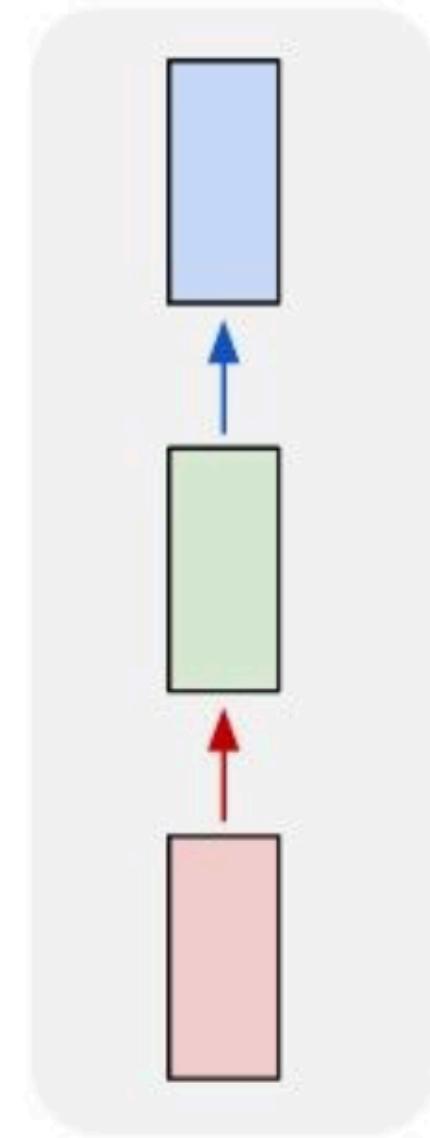
- Sometimes the sequence of data matters.
 - Text generation
 - Stock price prediction
- **For example: The clouds are in the ?**
 - **sky**

Sequential Data

- Sometimes the sequence of data matters.
 - Text generation
 - Stock price prediction
- **For example: The clouds are in the ?**
 - **sky**
- Simple solution: Neural networks?
 - Fixed input/output size
 - Fixed number of steps

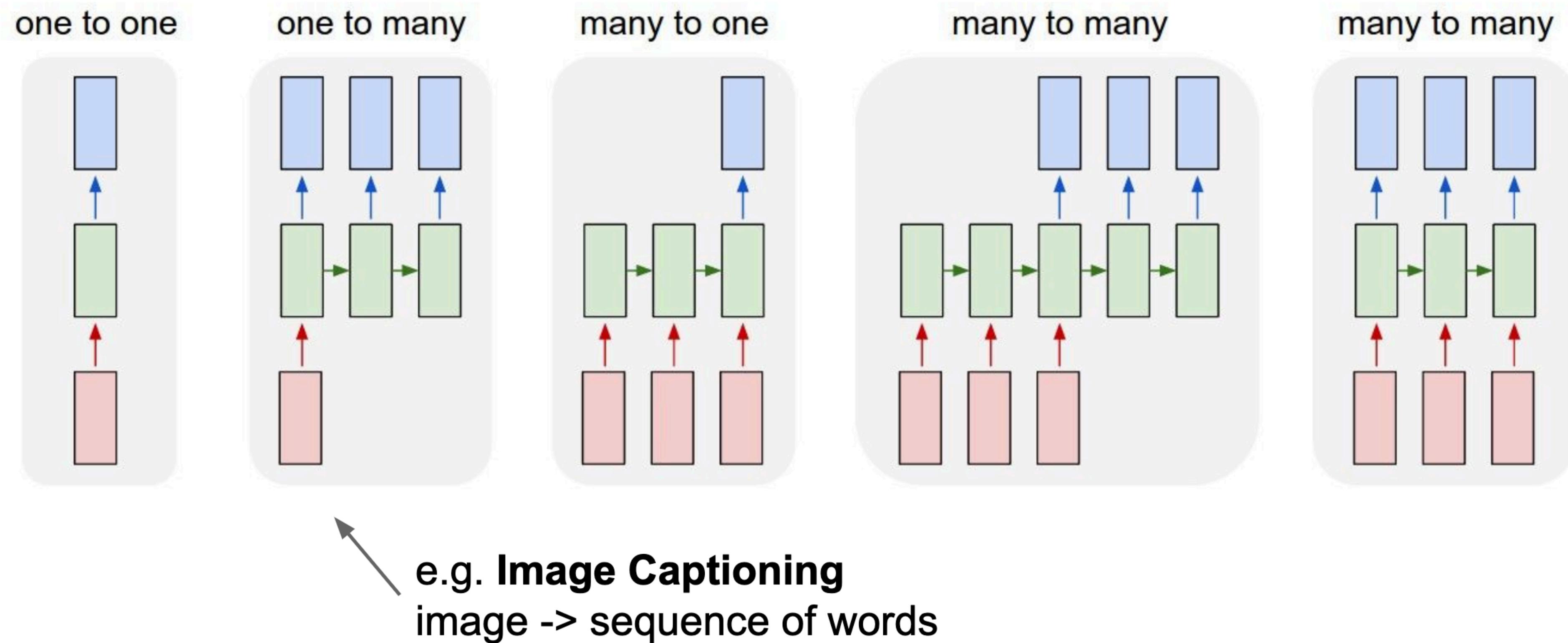
Neural Networks

one to one

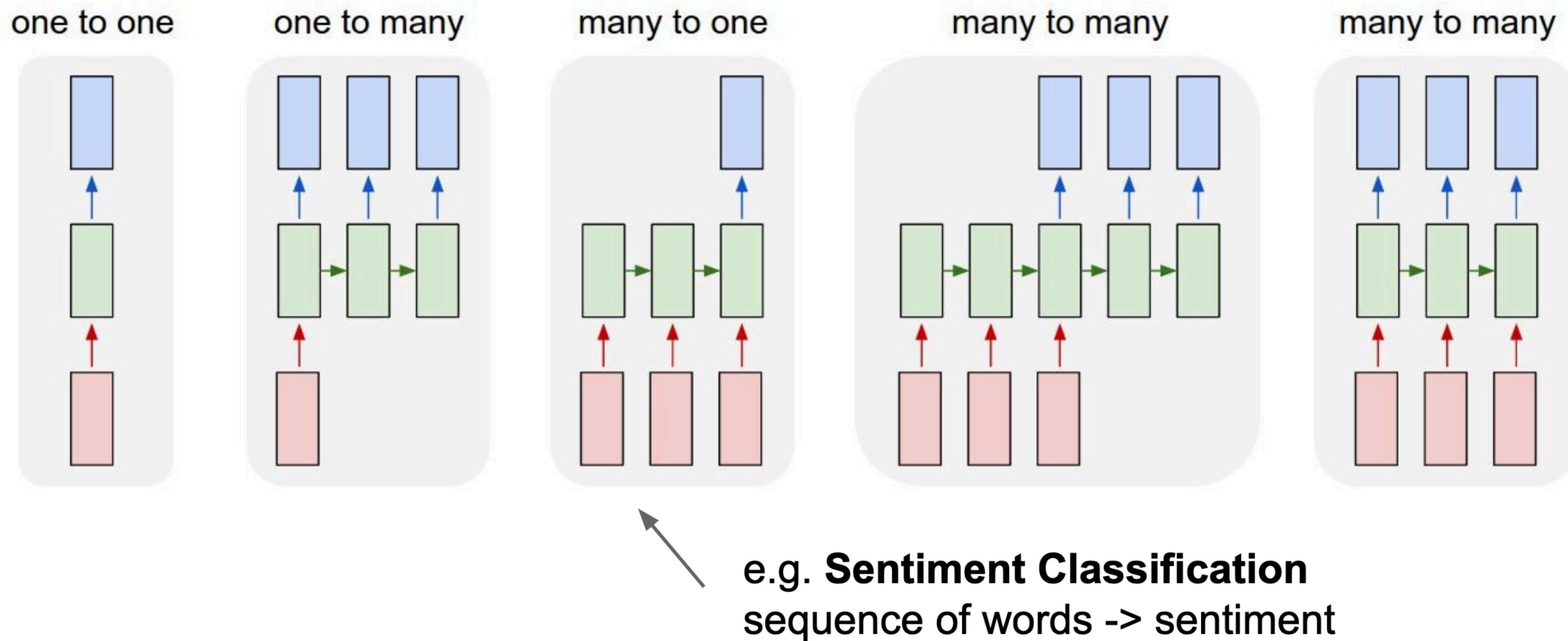


Vanilla Neural Networks

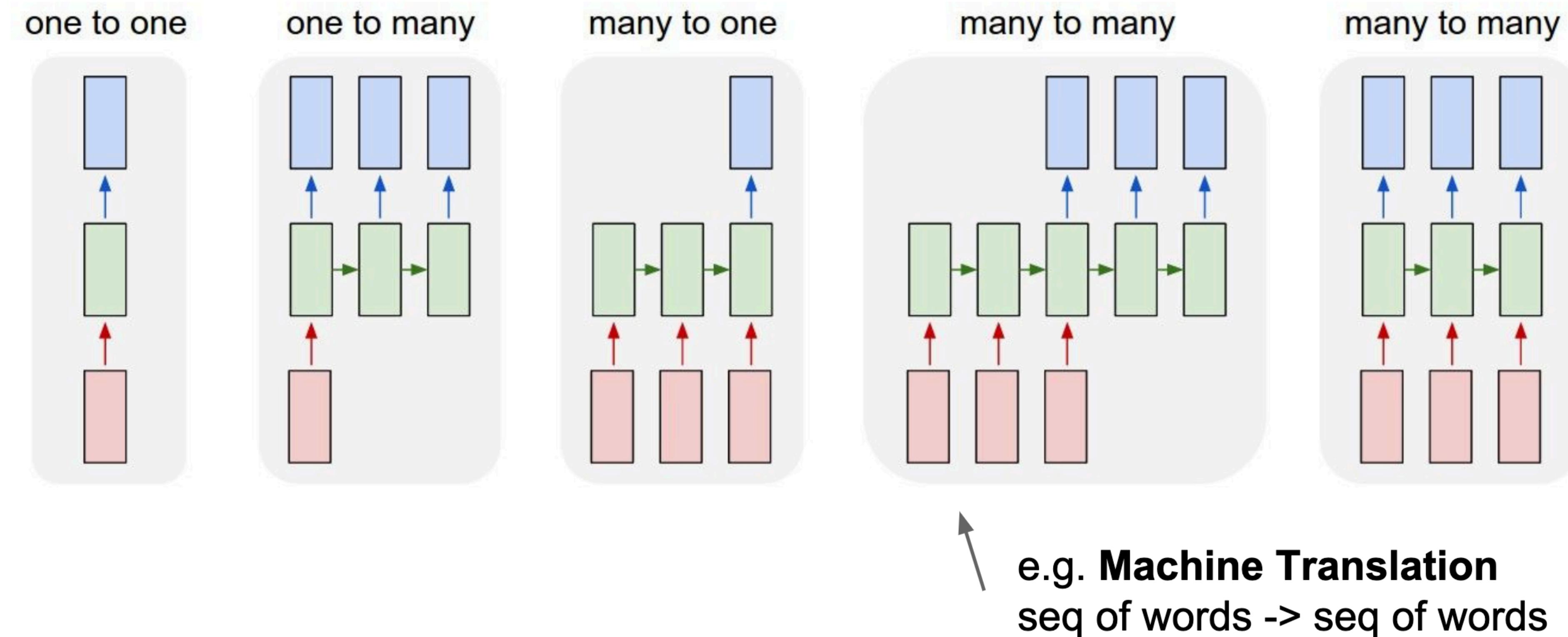
Recurrent Neural Networks



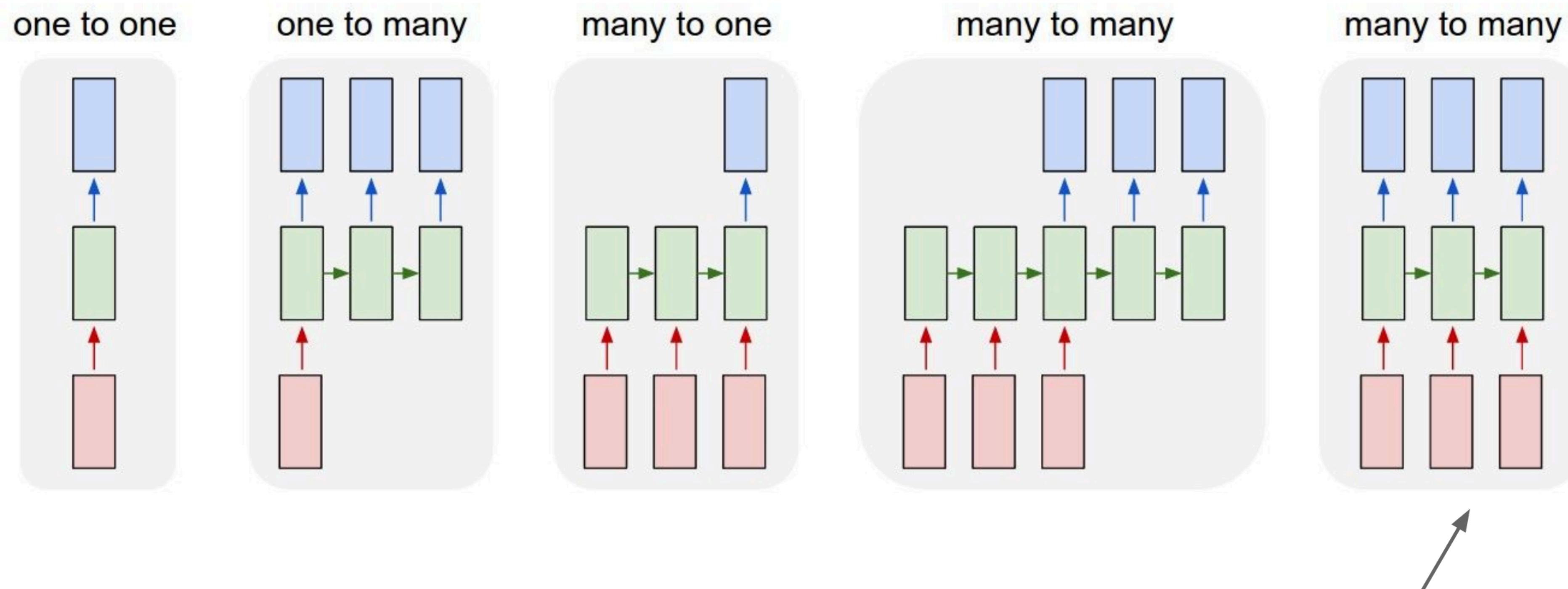
Recurrent Neural Networks



Recurrent Neural Networks



Recurrent Neural Networks

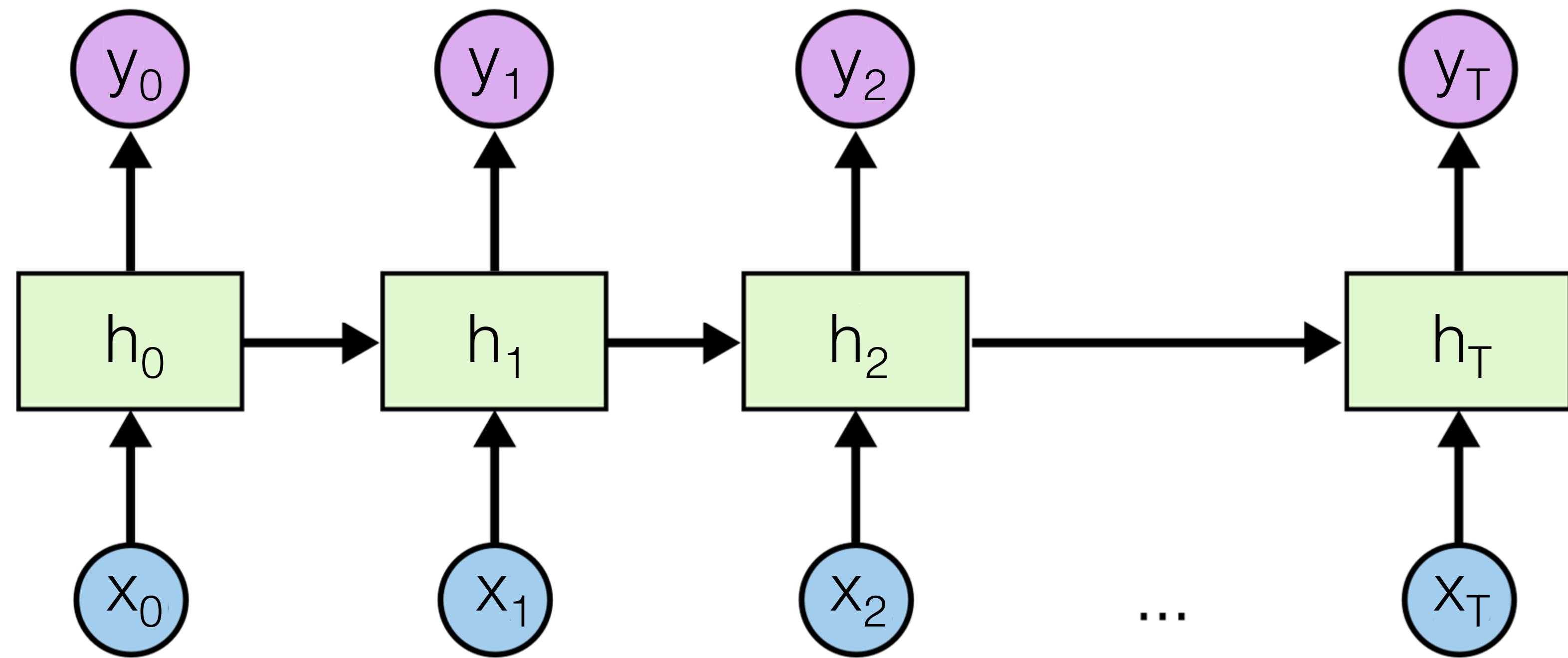


e.g. **Video classification on frame level**

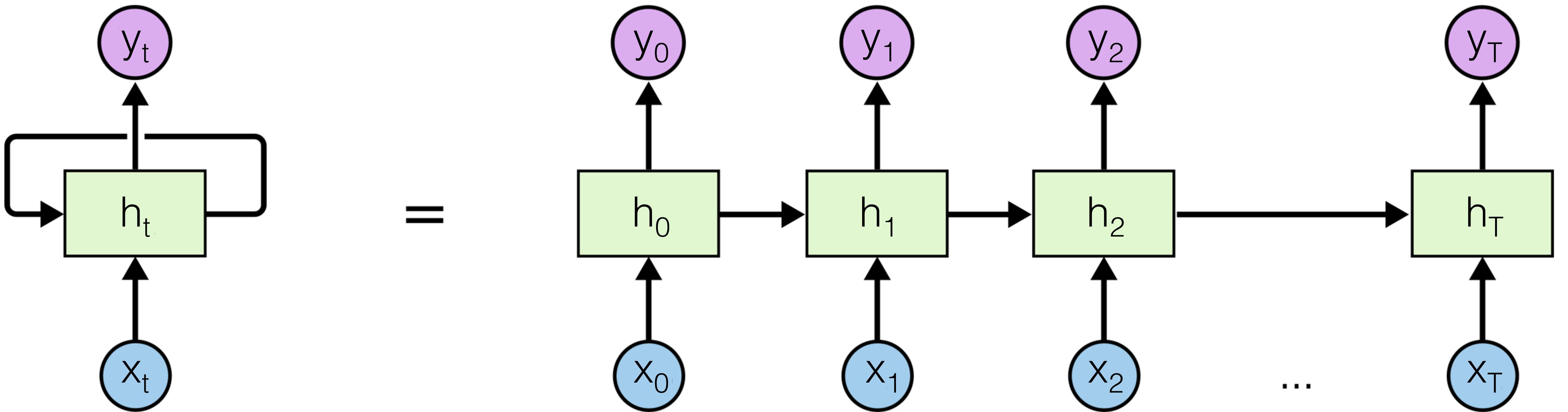
Basic idea

A many-to-many task

- x : inputs
- y : outputs
- h : hidden layers
- subscripts index time



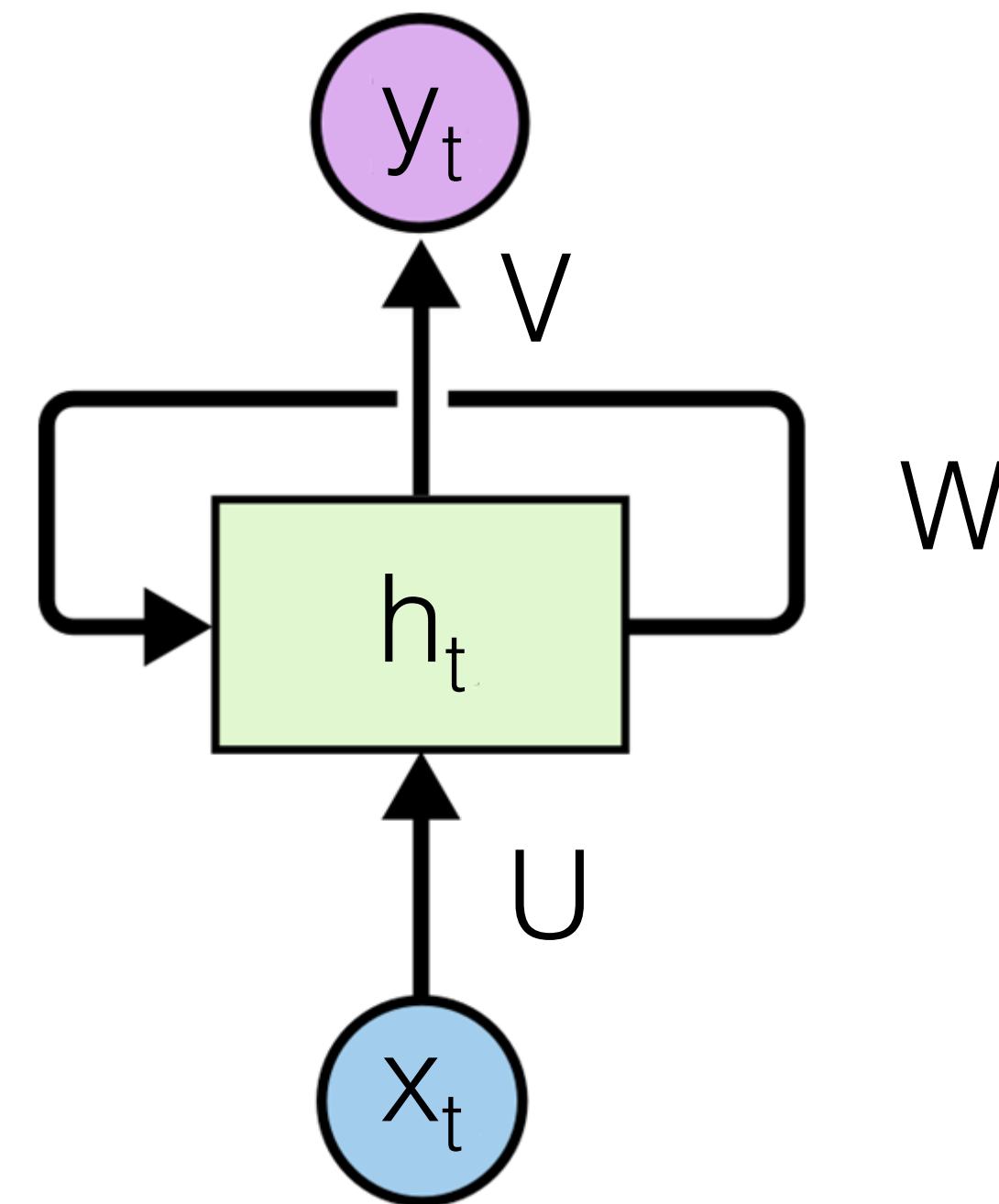
Unroll through time



Basic idea (with parameters)

Process through time (t)

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



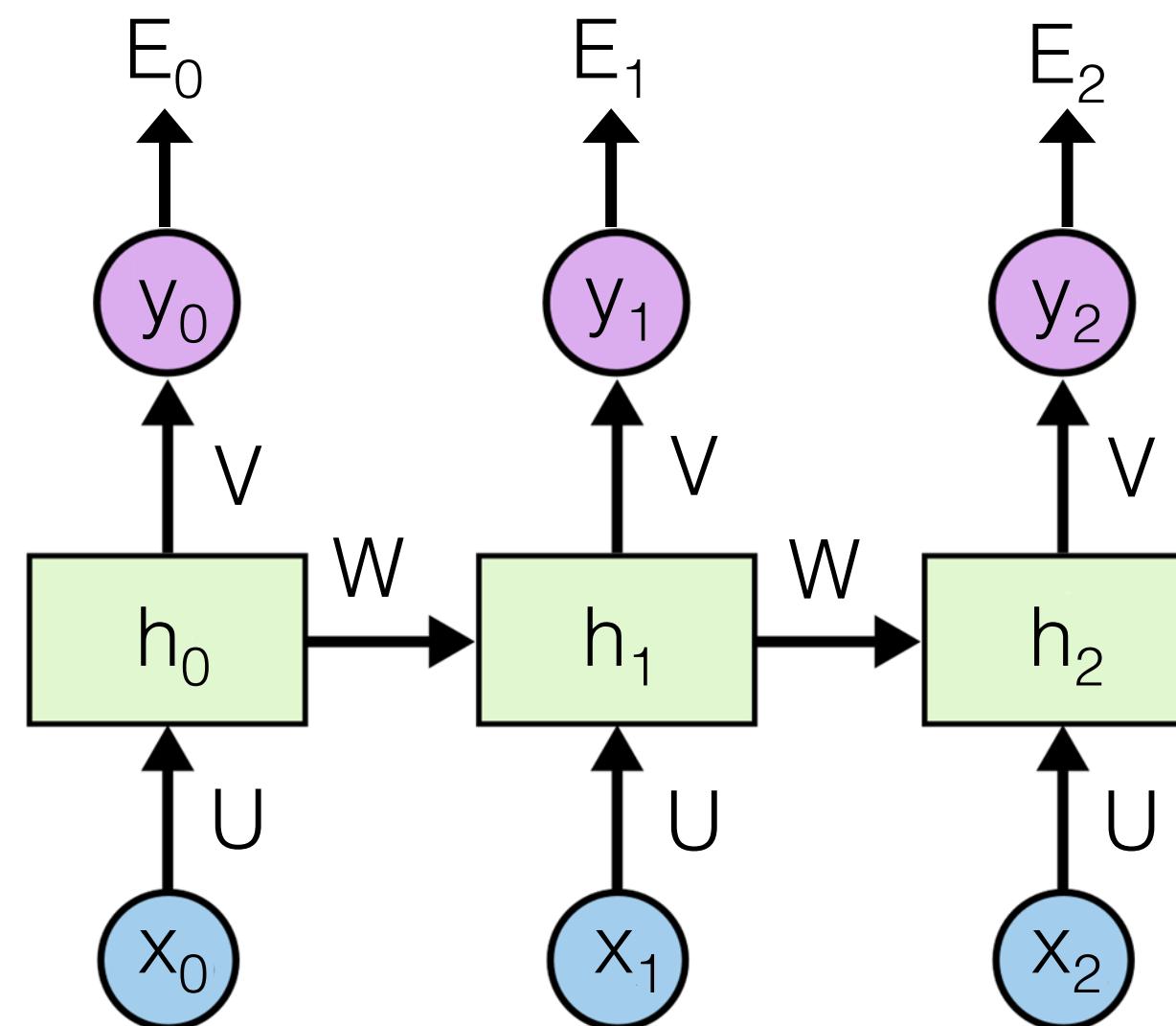
$$h_t = \tanh(Ux_t + Wh_{t-1})$$

$$y_t = f(Vh_t)$$

How does gradient flow in RNN?

Training RNNs

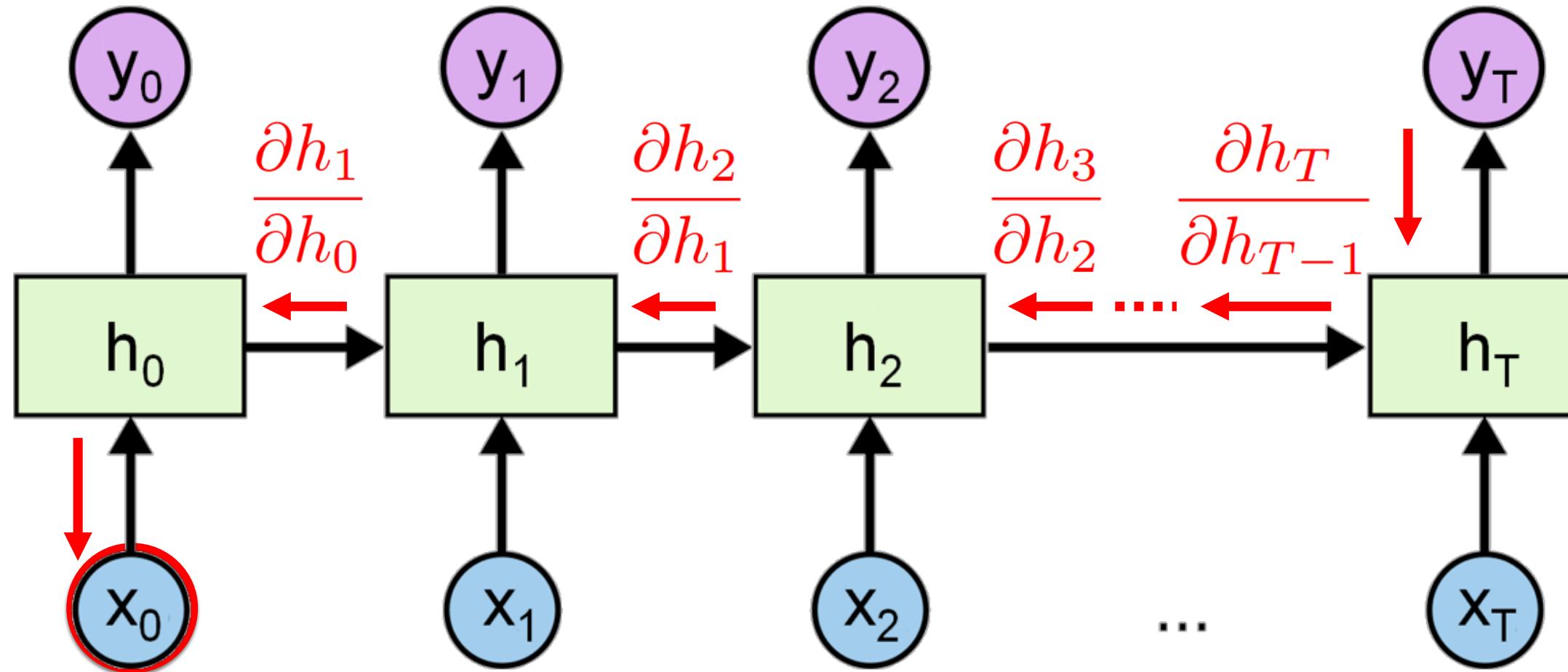
- Gradient descent from the loss $E = \sum_t (y_t - \hat{y}_t)^2$
- Following the structure the gradient is back propagated through time



RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994

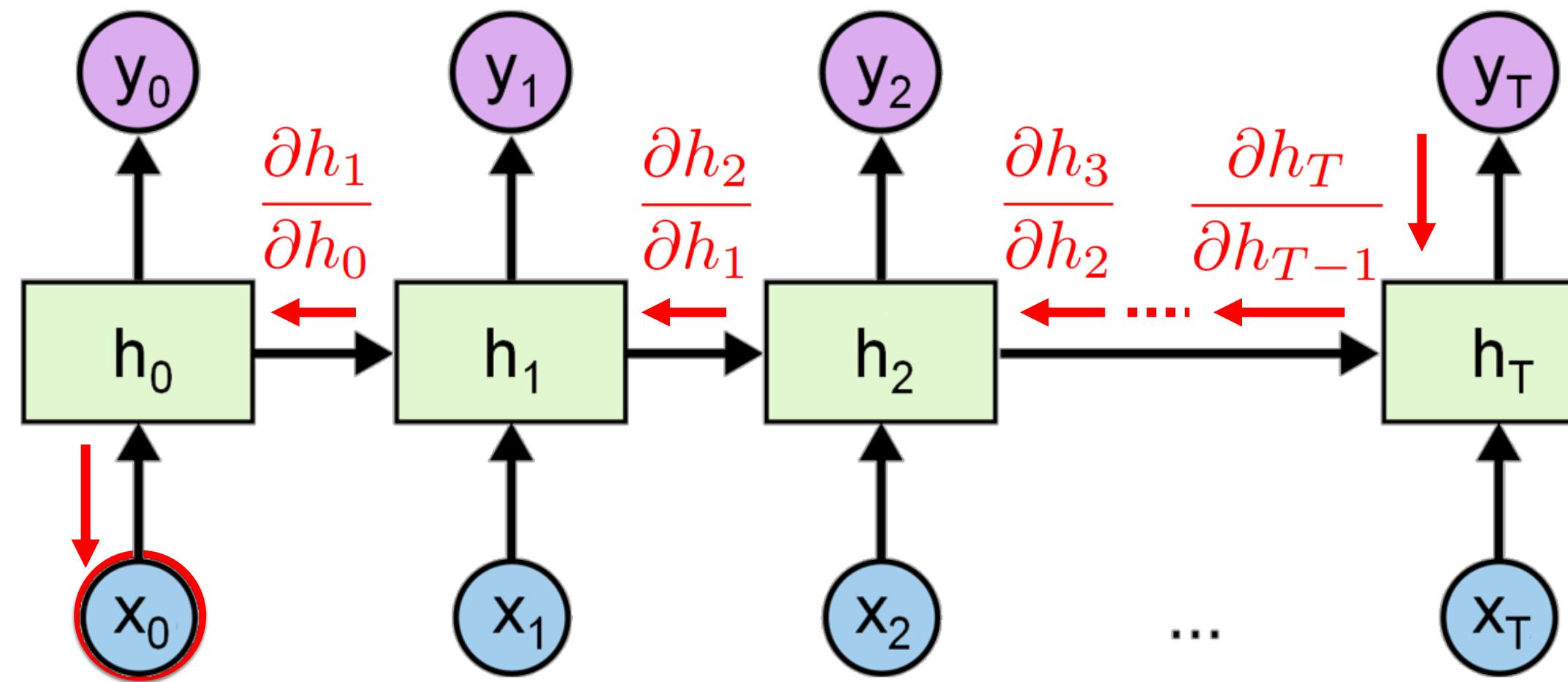
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



Largest singular value > 1 :
Exploding gradients

Largest singular value < 1 :
Vanishing gradients

RNN Gradient Flow



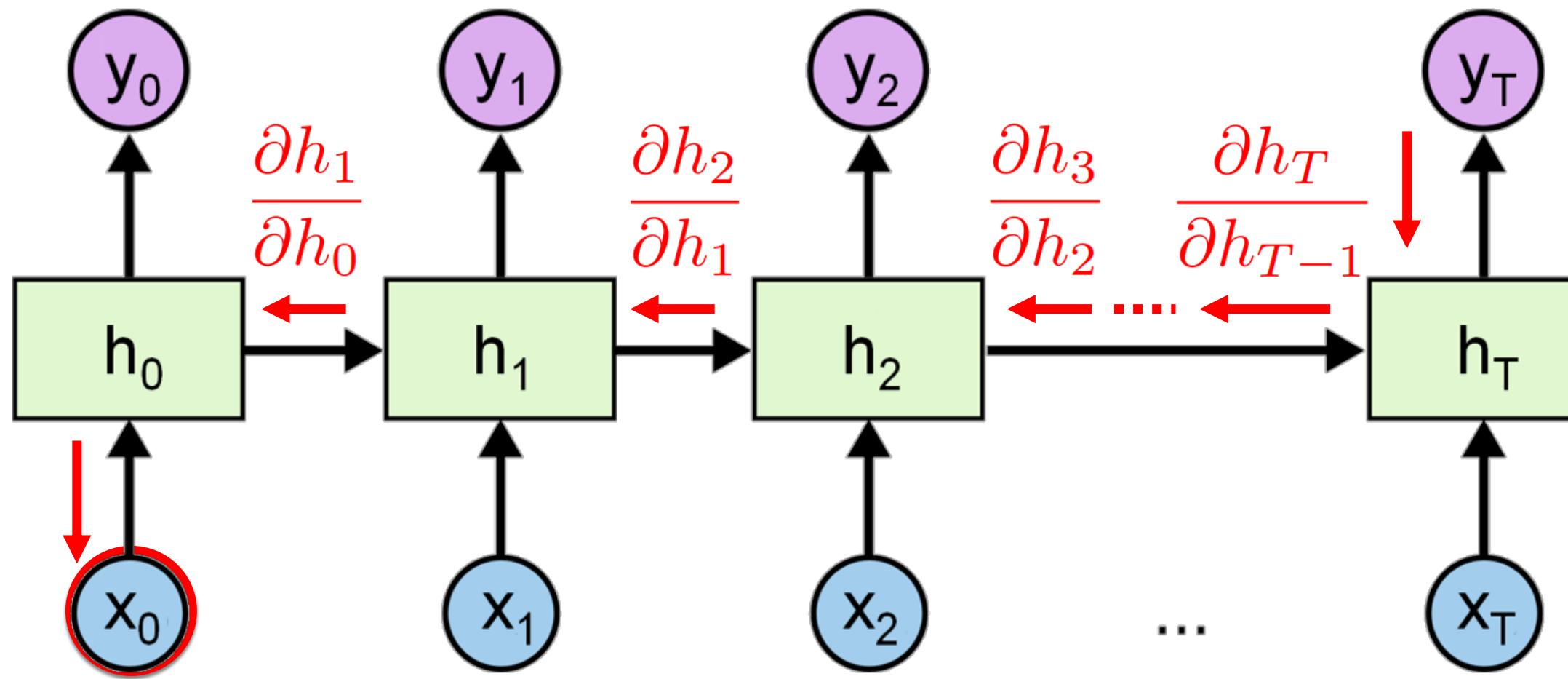
Largest singular value > 1 :
Exploding gradients

Largest singular value < 1 :
Vanishing gradients

Gradient clipping: Scale
gradient if its norm is too big

```
grad_norm = np.sum(grad * grad)
if grad_norm > threshold:
    grad *= (threshold / grad_norm)
```

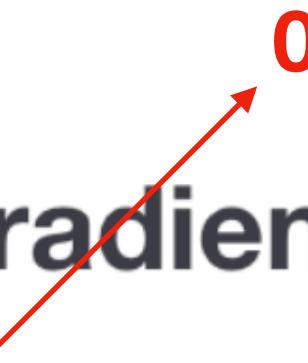
RNN Gradient Flow



Largest singular value > 1:
Exploding gradients

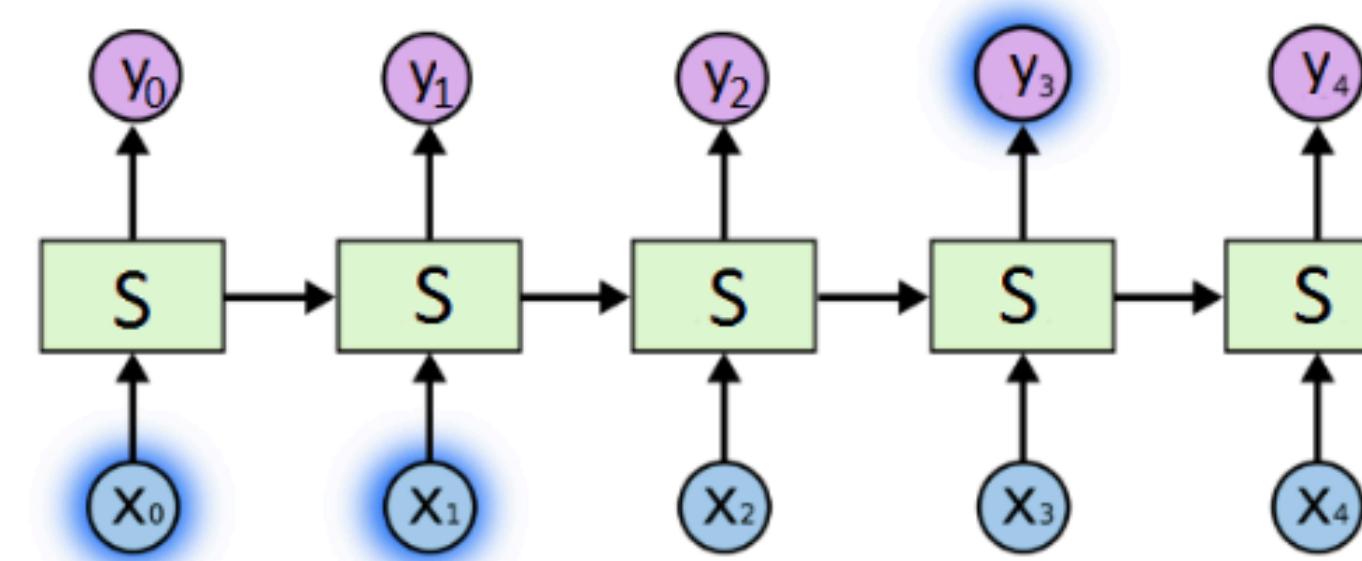
Largest singular value < 1:
Vanishing gradients

new weight = weight - learning rate * gradient

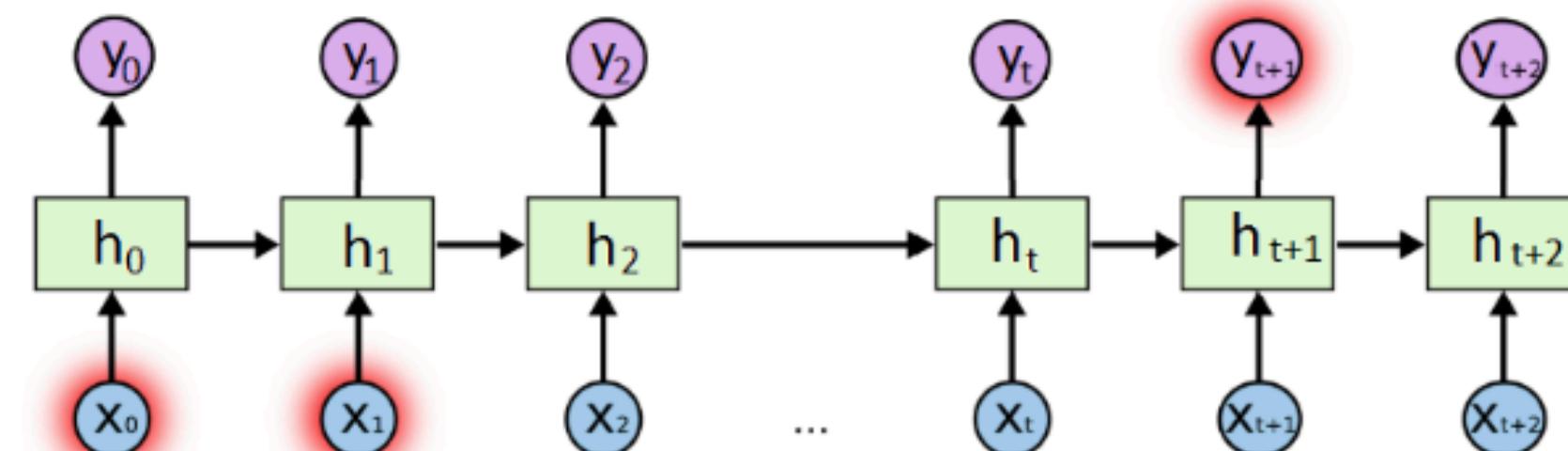


The Problem of Long-term Dependencies

- RNNs connect previous information to present task:
 - may be enough for predicting the next word for "the clouds are in the **sky**"



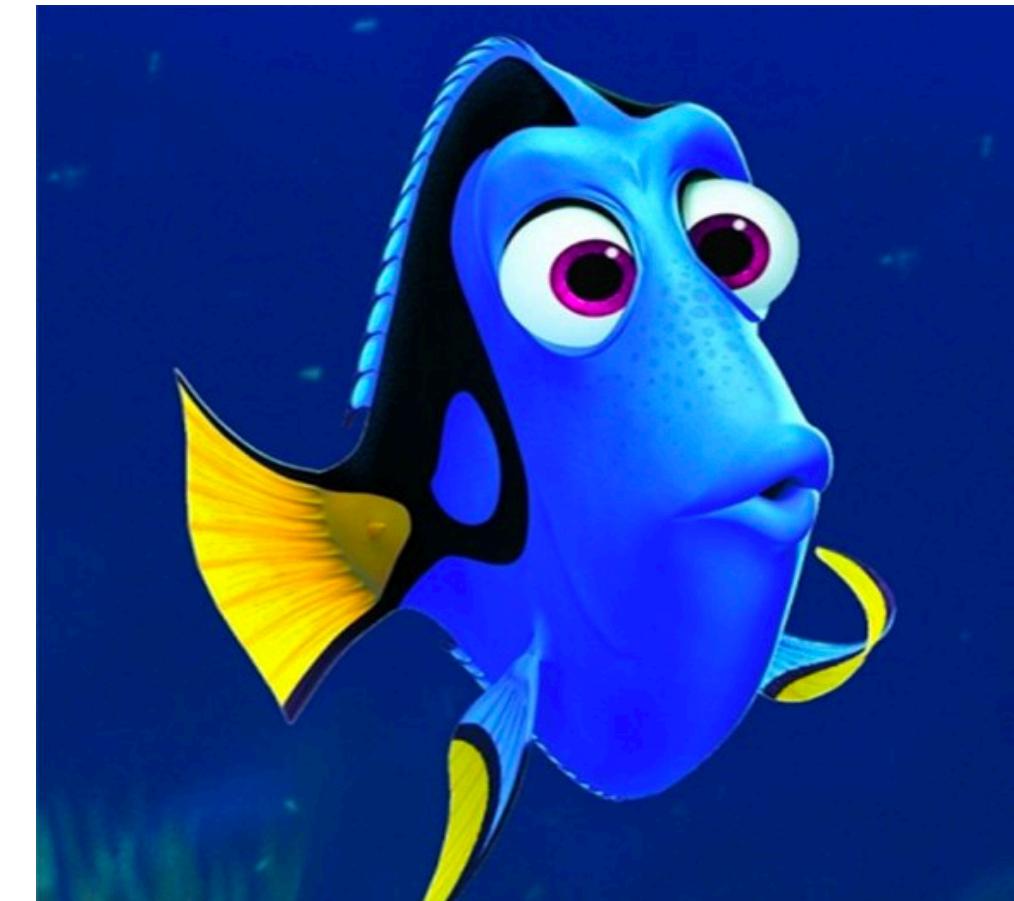
- may not be enough when more context is needed



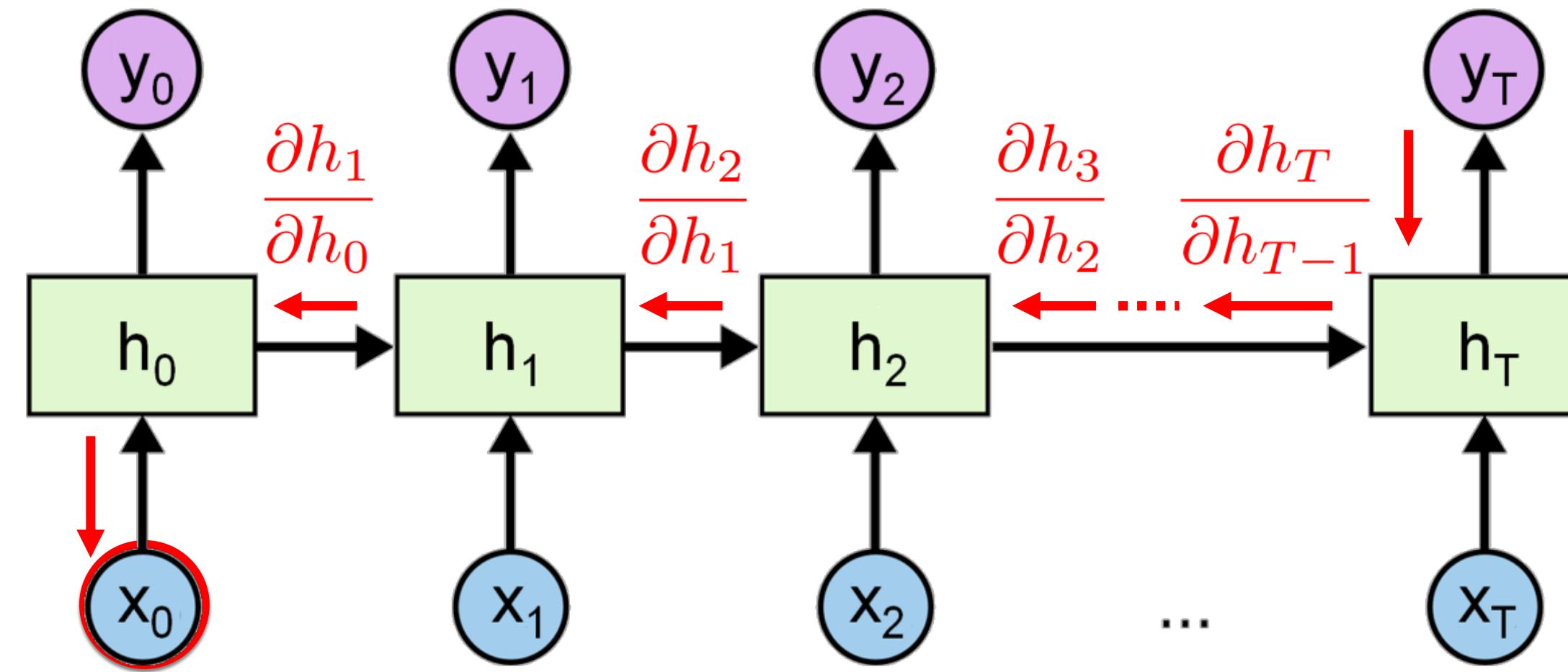
Short-Term memory

- RNNs suffer from what is known as short-term memory!

I was born in France, but I have been working in South Africa
working for ... (another 200 words) ... Therefore my mother tongue
is:



RNN Gradient Flow

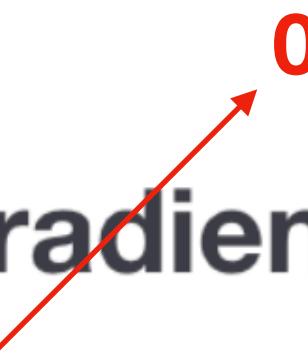


Largest singular value > 1 :
Exploding gradients

Largest singular value < 1 :
Vanishing gradients

→ Change RNN architecture

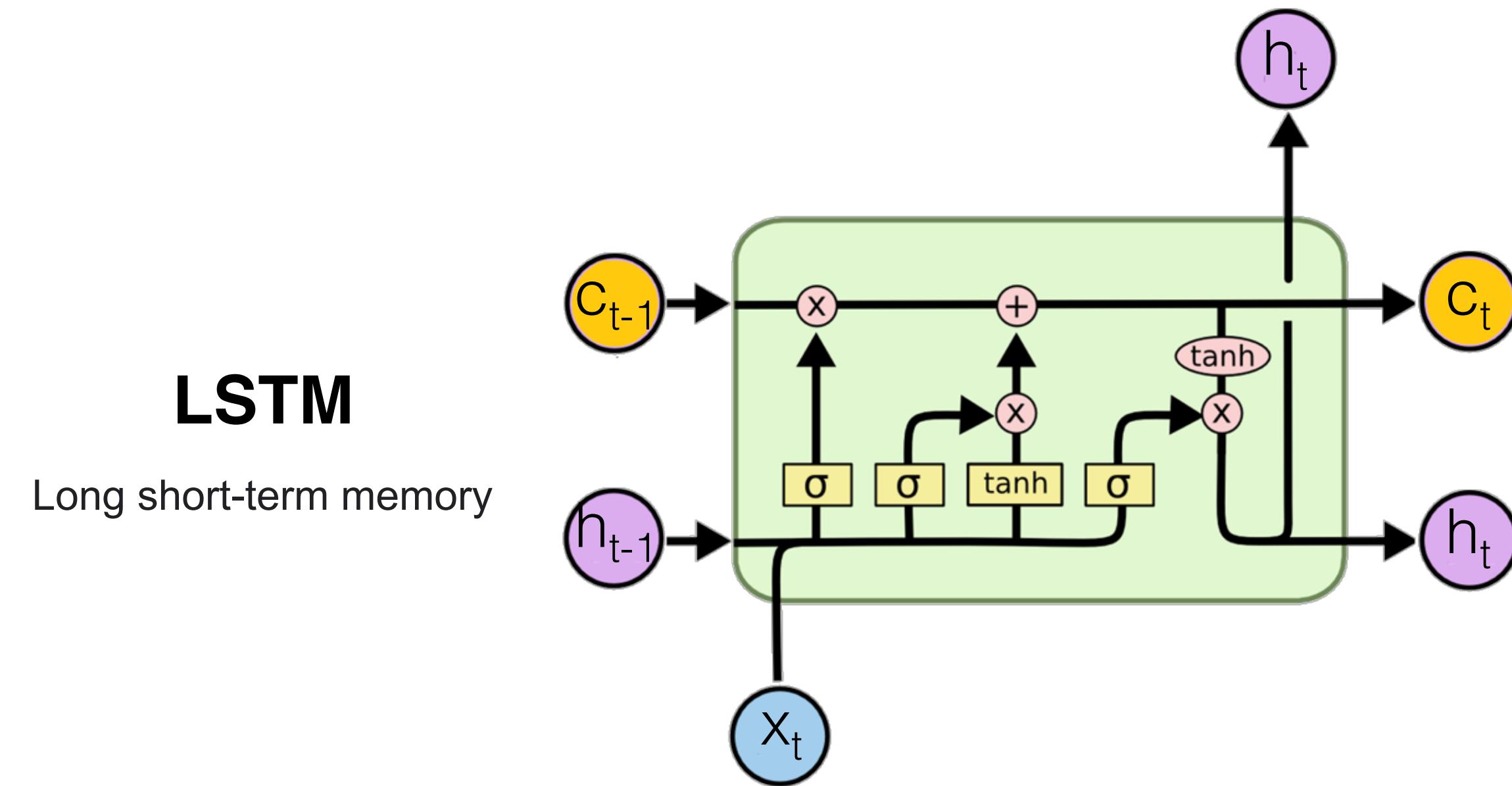
new weight = weight - learning rate * gradient



Gradient Vanishing

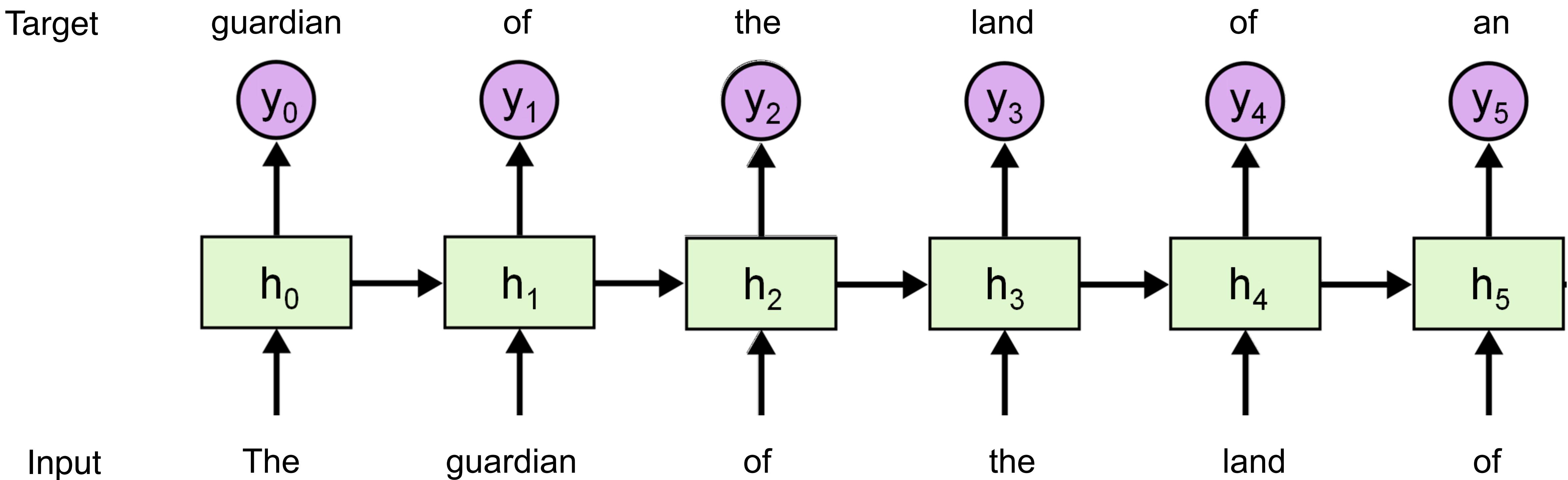
- No simple solution
- Change the “memory cells” of the RNN

$$\mathbf{h}_t = \tanh(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1})$$
$$\mathbf{y}_t = \mathbf{f}(\mathbf{V}\mathbf{h}_t)$$



Language Modelling using RNNs

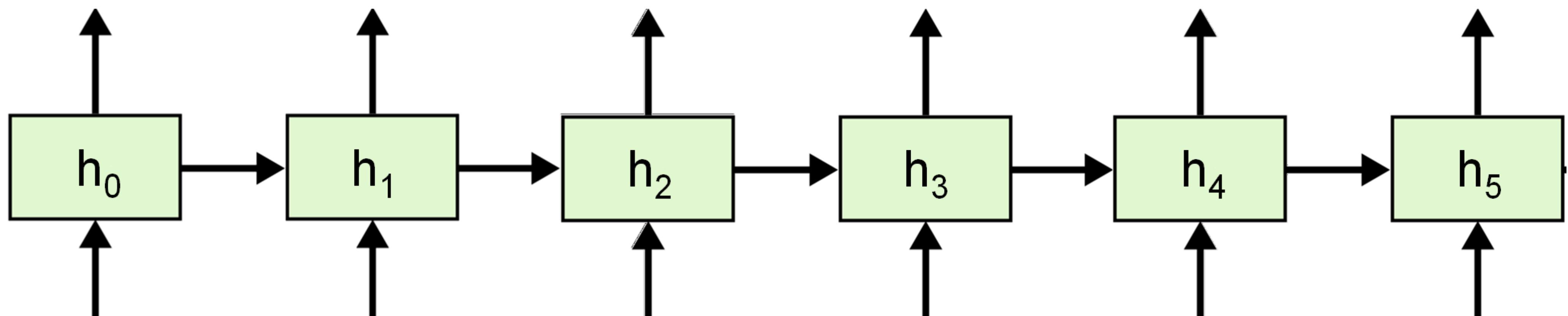
Training.
Given previous words,
predict the next word



Test.

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

Prediction

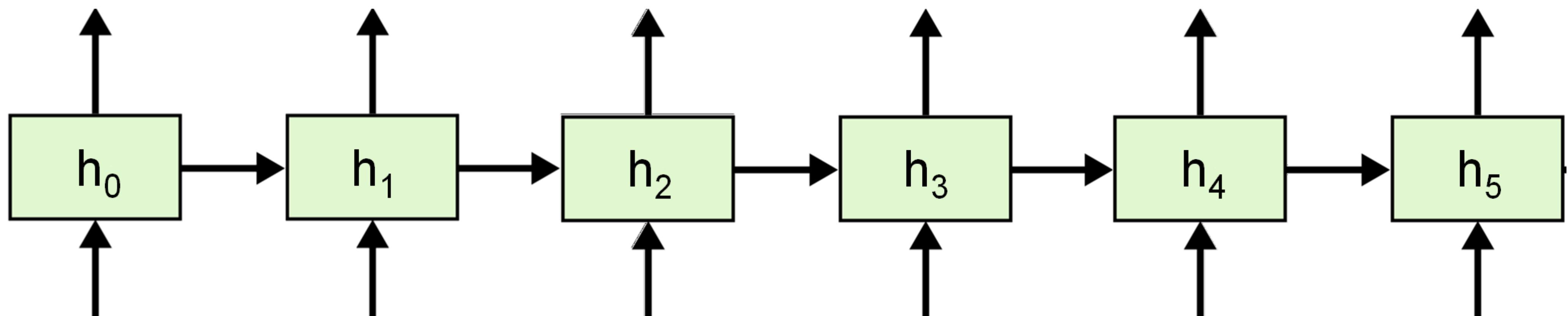


Input

Test.

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

Prediction



Input

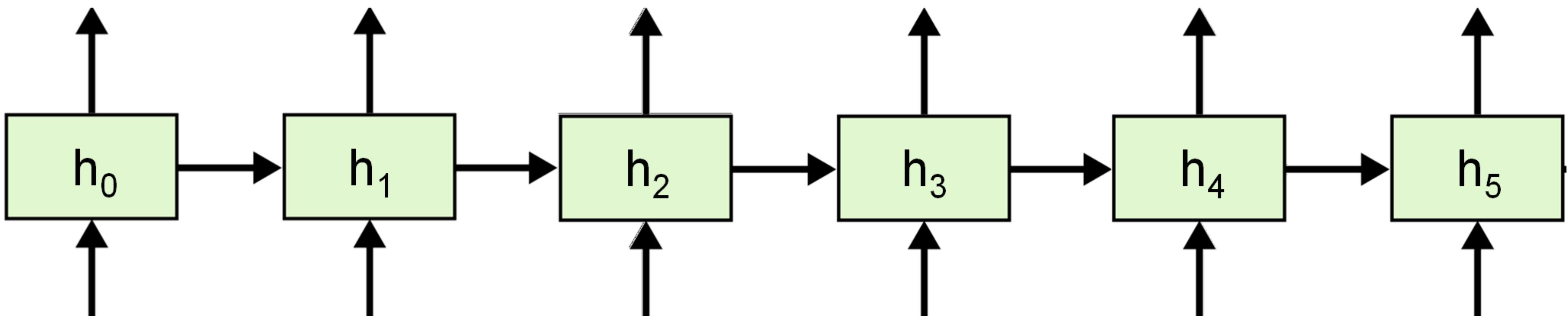
The

Test.

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

Prediction

woman

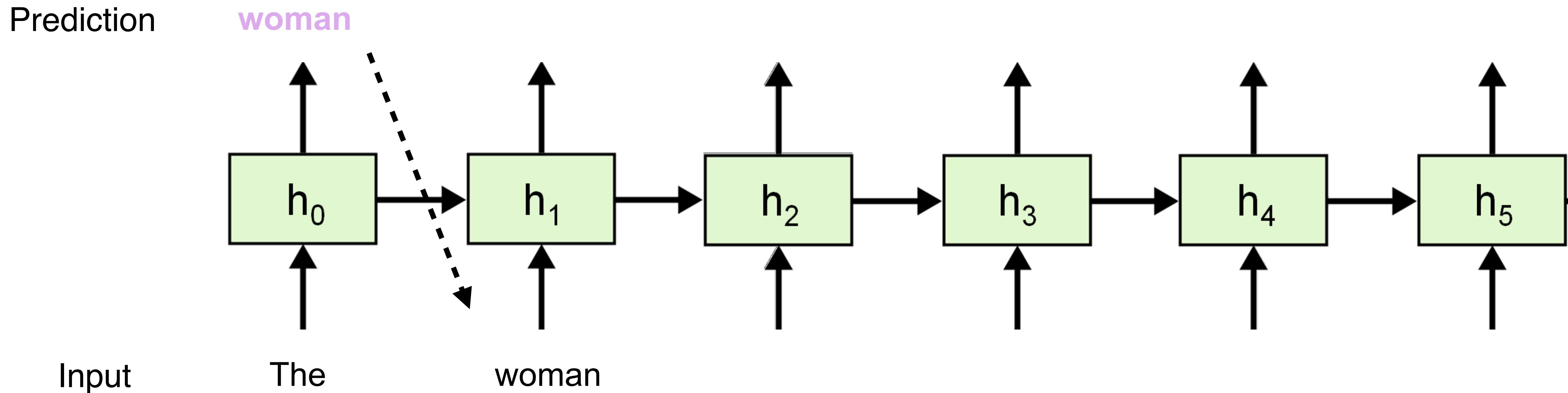


Input

The

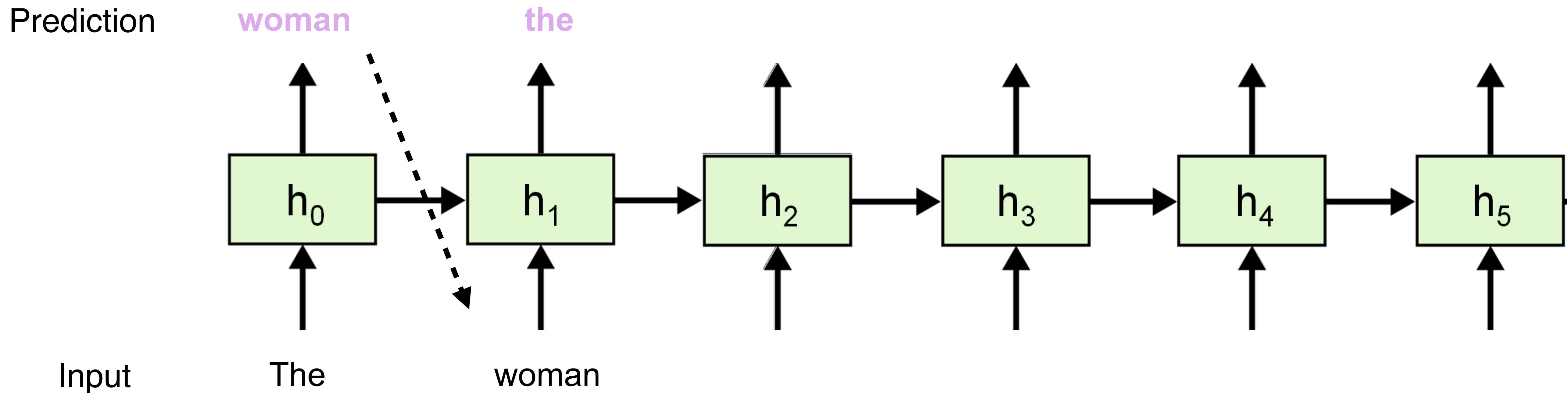
Test.

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



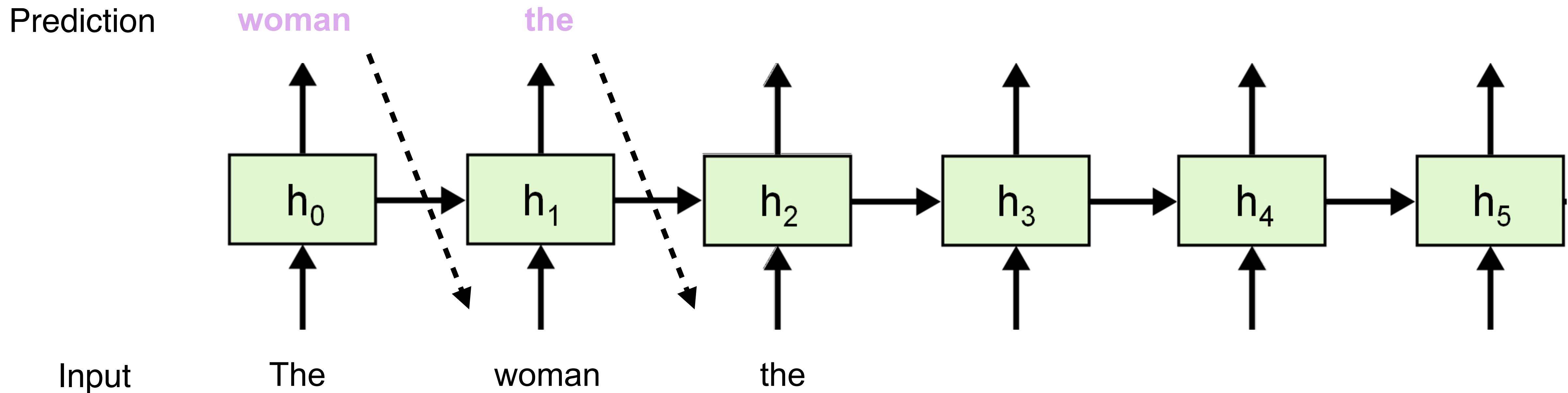
Test.

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



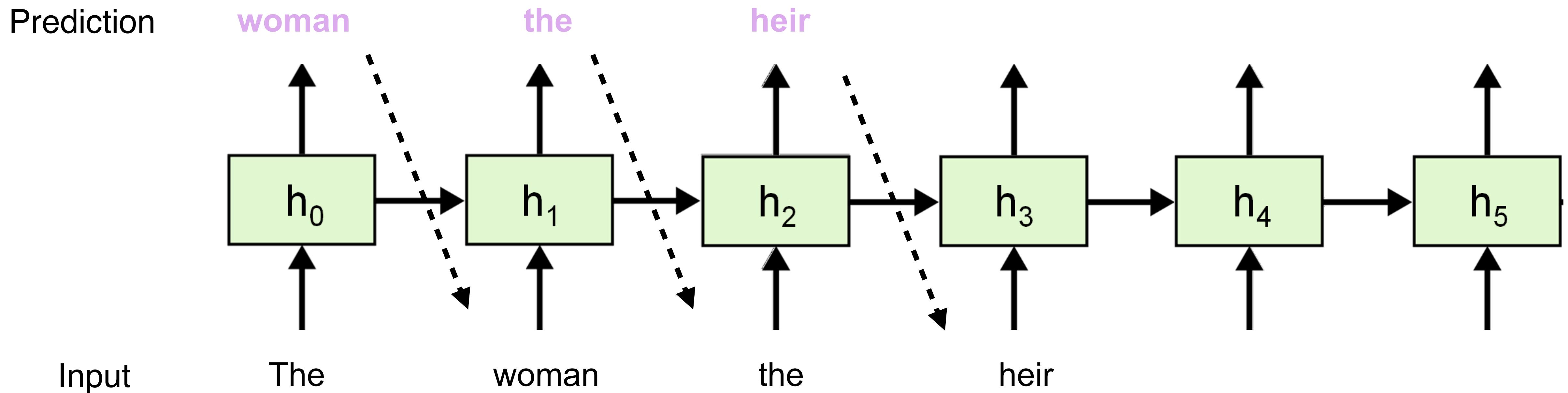
Test.

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



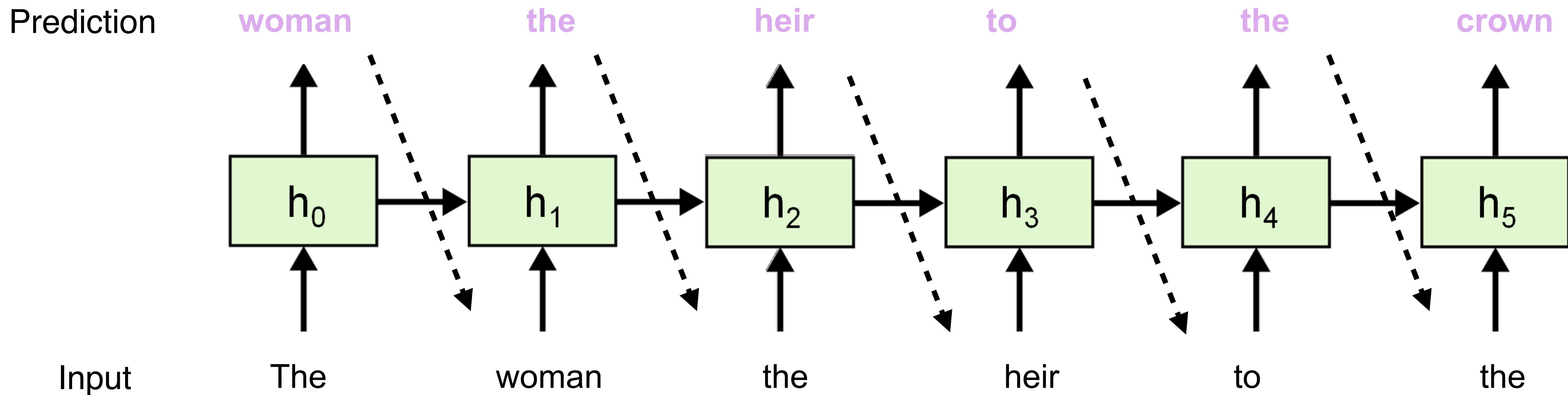
Test.

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



Test.

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



Fun example to try!

The screenshot shows the Shortly AI interface. On the left, there's a sidebar with three dropdown menus: 'Writing Stats', 'Article brief', and 'Output length'. The 'Output length' menu is open, showing a slider with three options: 'A little', 'Somewhere in between', and 'A lot'. A blue bar is positioned between 'A little' and 'A lot'. Below the slider is a large text input field containing the placeholder 'Your title here...'. At the bottom of the sidebar is a blue button labeled 'Write for me'. To the right of the sidebar is a large, empty text area where the generated content would be displayed. In the top right corner of this area, there are three small buttons: 'Saved' (purple), 'Share' (grey), and a trash can icon (grey). At the bottom right of the main text area is a small square icon.

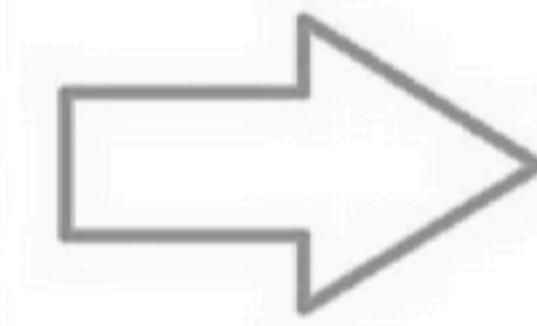


<https://shortlyai.com/>

CNNs



This is how I see

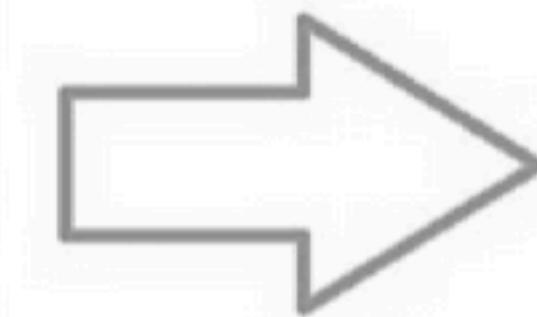


88	126	145	85	123	142	85	123	142	86	124
86	125	142	84	123	140	83	122	139	85	124
85	124	141	82	121	138	82	121	138	84	123
82	119	135	80	117	133	80	117	133	85	122
78	114	128	77	113	127	79	115	129	84	120
79	115	129	78	114	128	80	116	130	83	119
82	118	130	81	117	129	81	117	129	82	118
83	117	129	82	116	128	82	116	128	82	116
79	113	123	79	113	123	80	114	124	81	115
76	108	119	76	108	119	77	109	120	80	112
76	109	118	76	109	118	77	110	119	79	112

This is how my computer sees



This is how I see



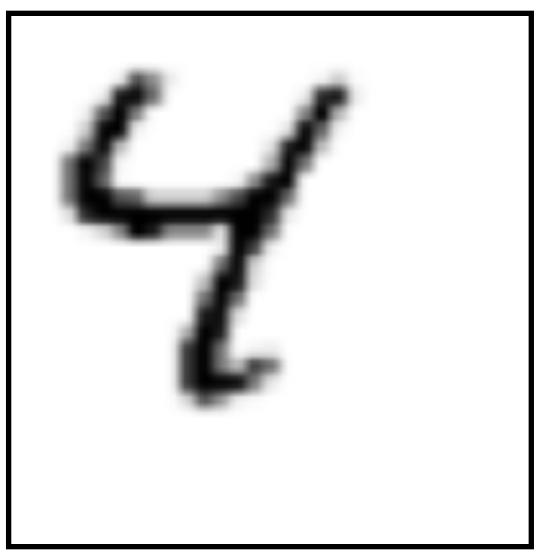
88	126	145	85	123	142	85	123	142	86	124
86	125	142	84	123	140	83	122	139	85	124
85	124	141	82	121	138	82	121	138	84	123
82	119	135	80	117	133	80	117	133	85	122
78	114	128	77	113	127	79	115	129	84	120
79	115	129	78	114	128	80	116	130	83	119
82	118	130	81	117	129	81	117	129	82	118
83	117	129	82	116	128	82	116	128	82	116
79	113	123	79	113	123	80	114	124	81	115
76	108	119	76	108	119	77	109	120	80	112
76	109	118	76	109	118	77	110	119	79	112

This is how my computer sees

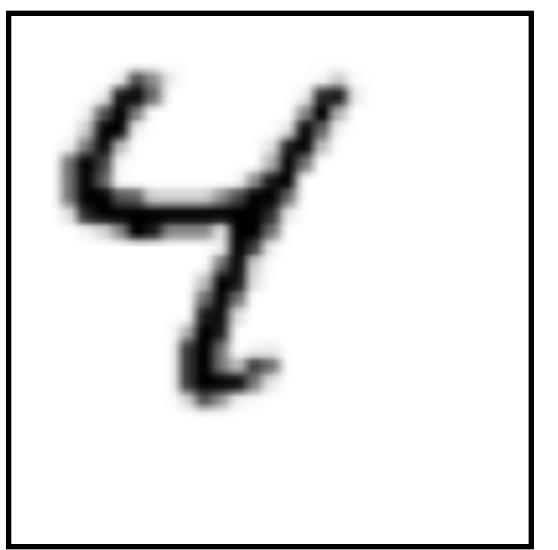
- A 100 x 100 pixel image has 10,000 dimensions
 - Often have a color dimension (data is a tensor)
 - Modern iPhone (12 MP): 4032 x 3024 pixels

Source: <https://medium.com/deep-math-machine-learning-ai/chapter-8-0-convolutional-neural-networks-for-deep-learning-364971e34ab2>

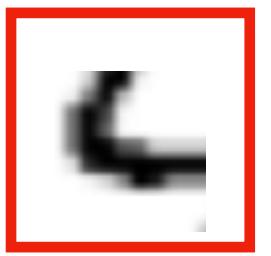
Image



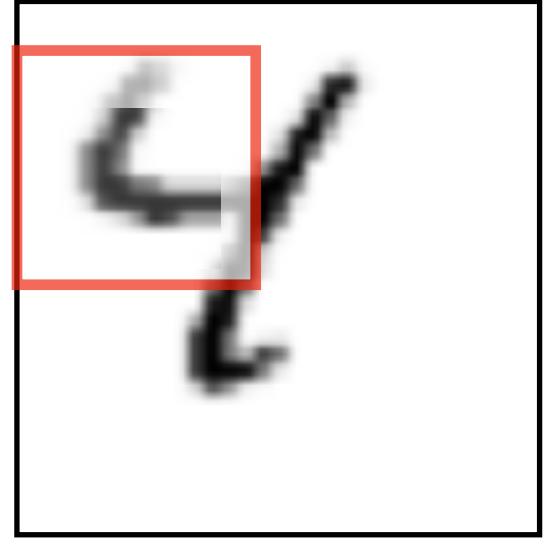
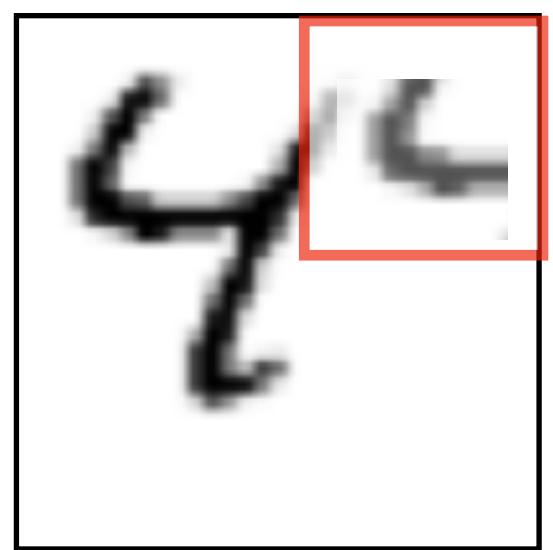
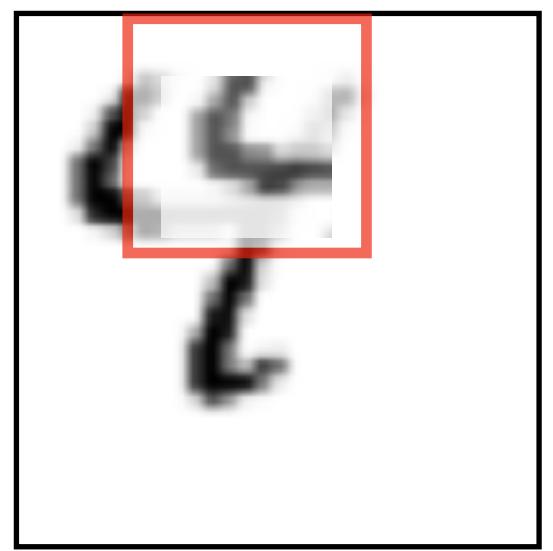
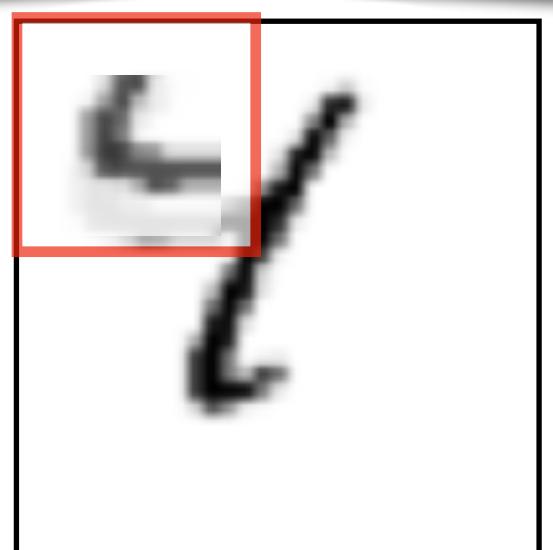
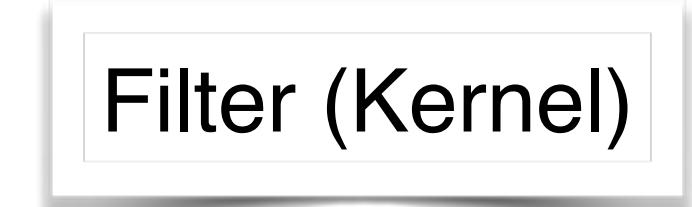
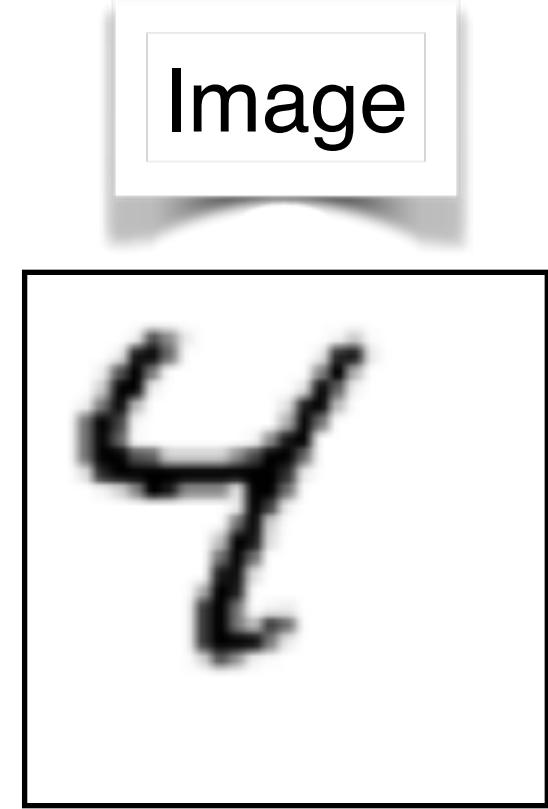
Image



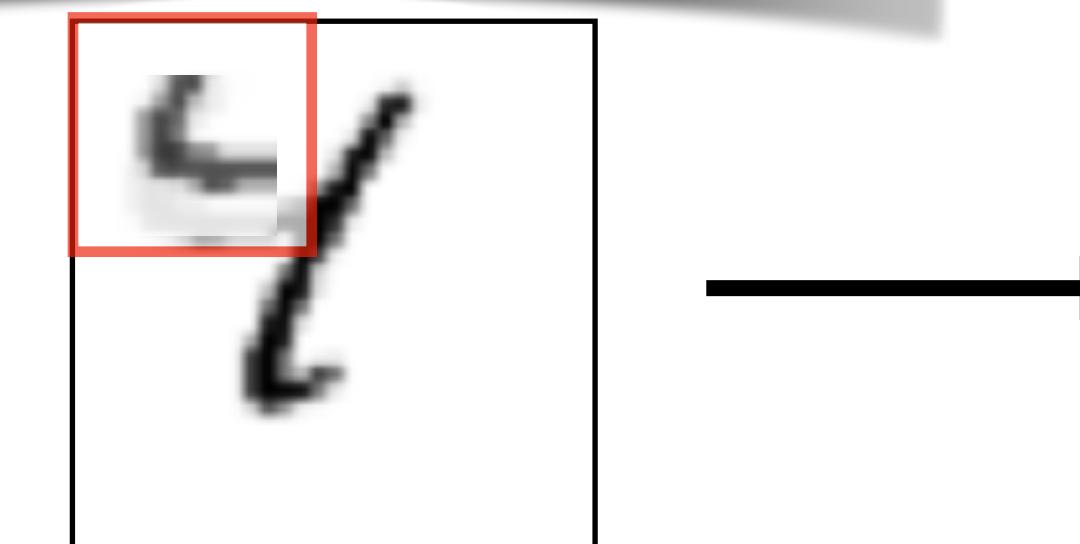
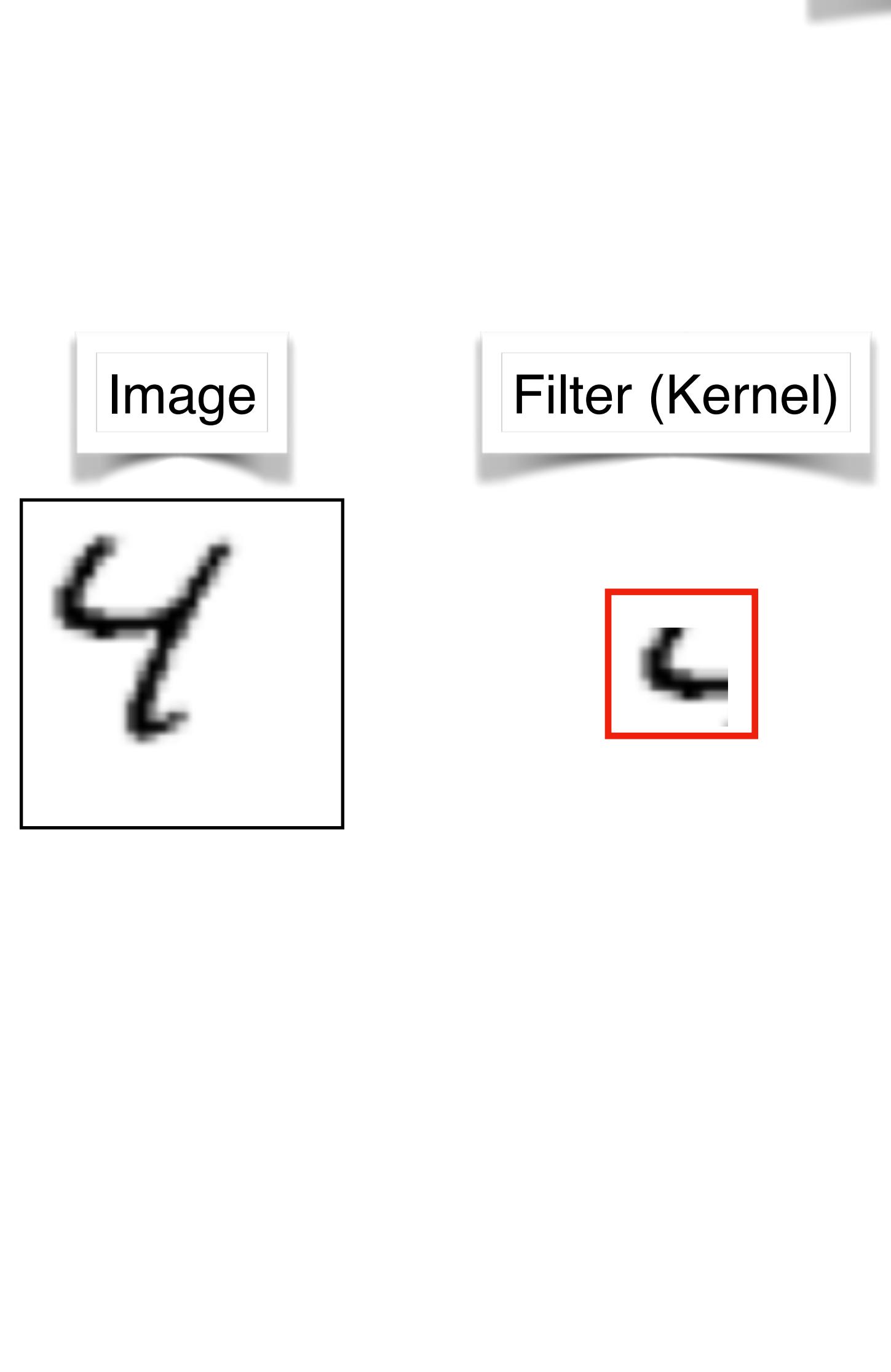
Filter (Kernel)



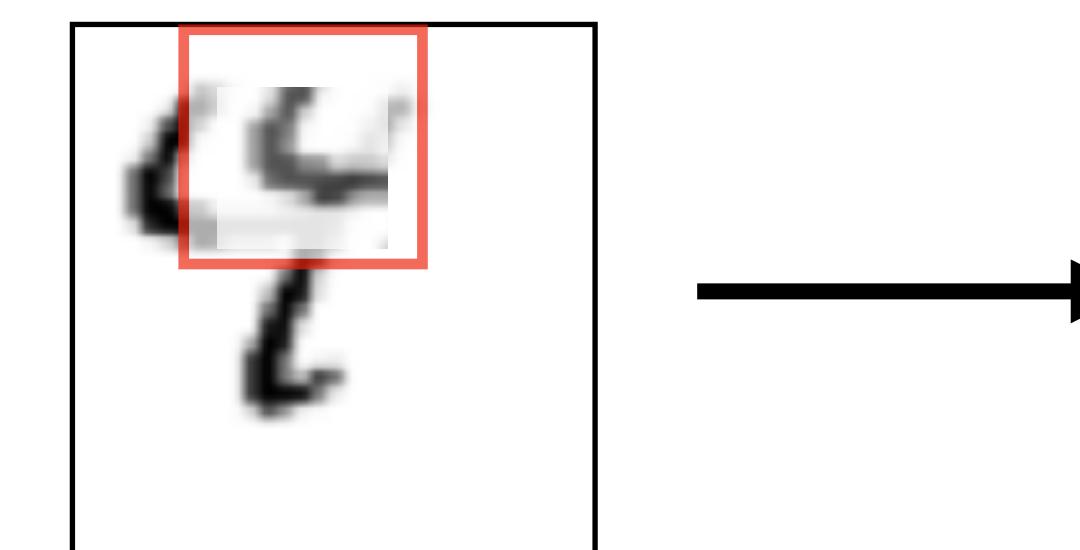
Pass the filter over the image
(Convolutions)



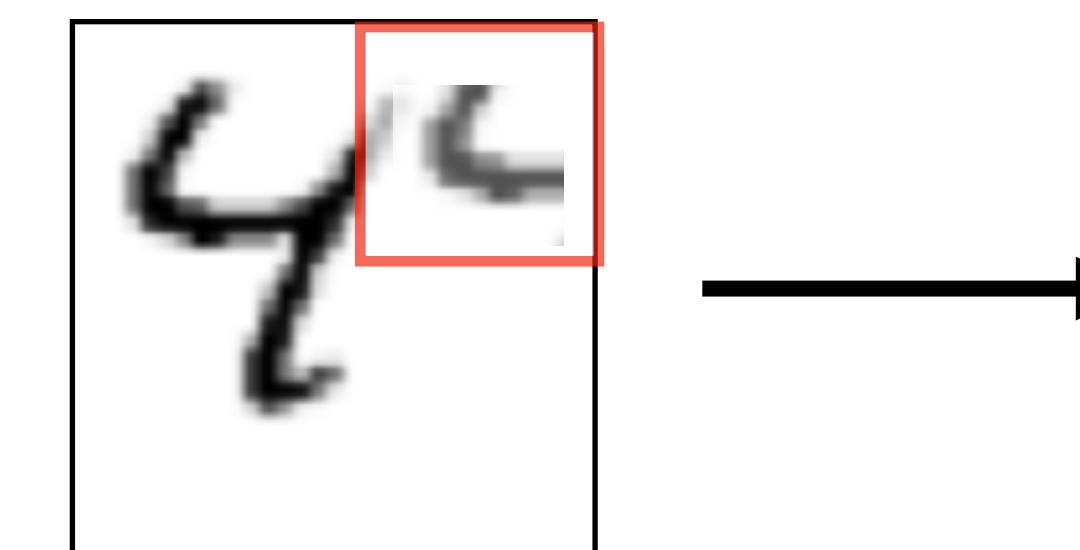
Pass the filter over the image
(Convolutions)



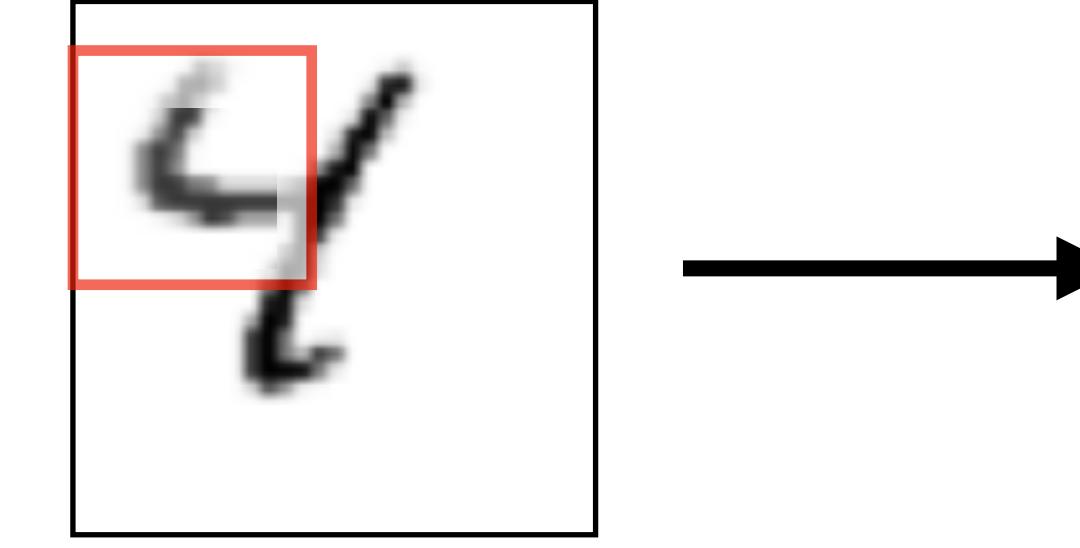
No detection
Output: Low



No detection
Output: Low

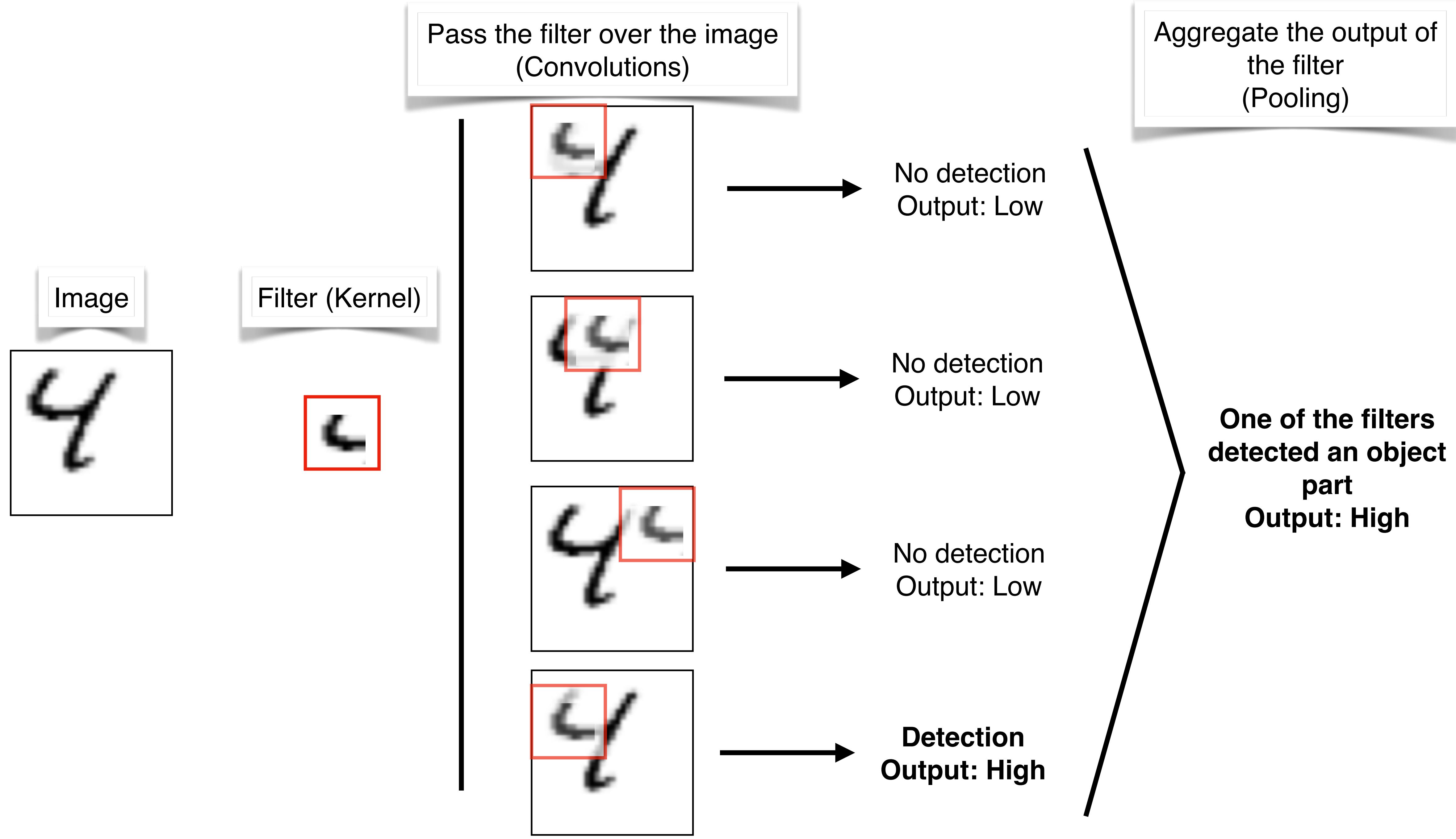


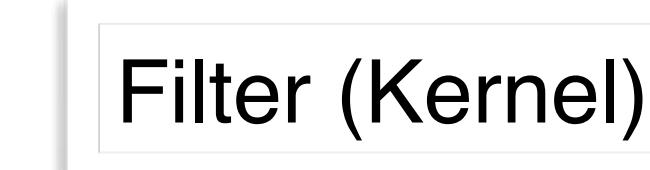
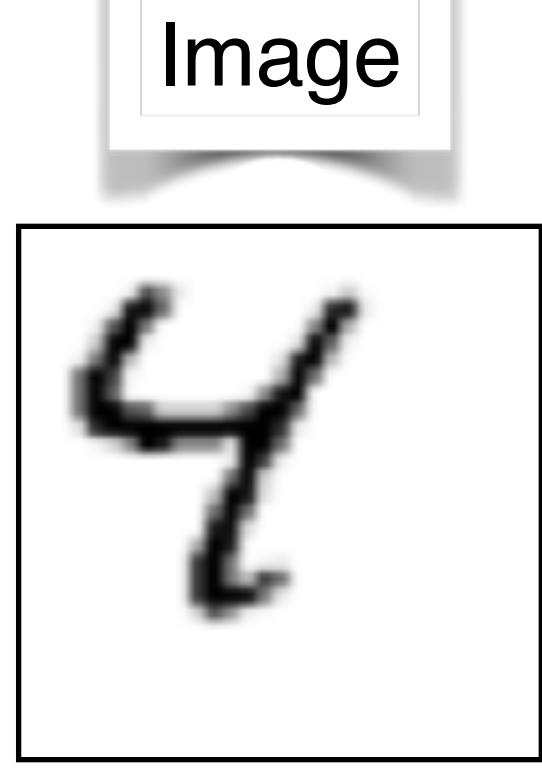
No detection
Output: Low



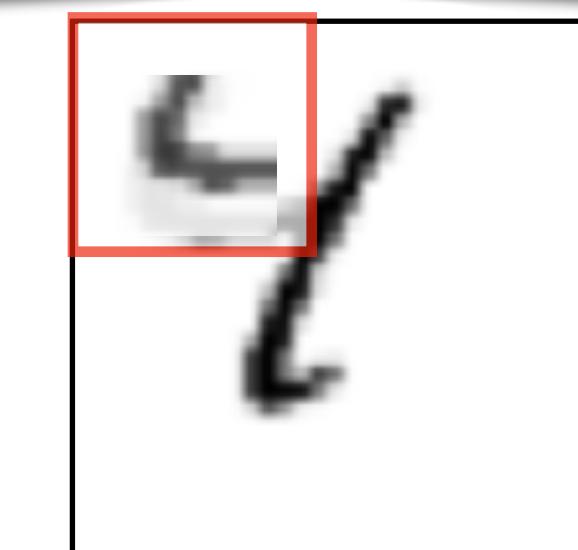
**Detection
Output: High**

...

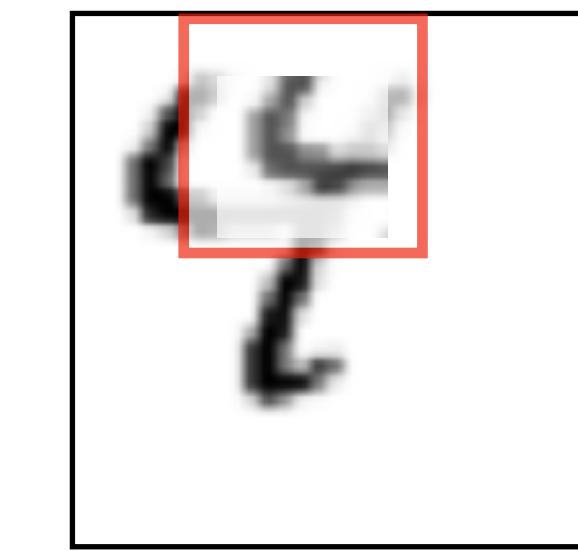




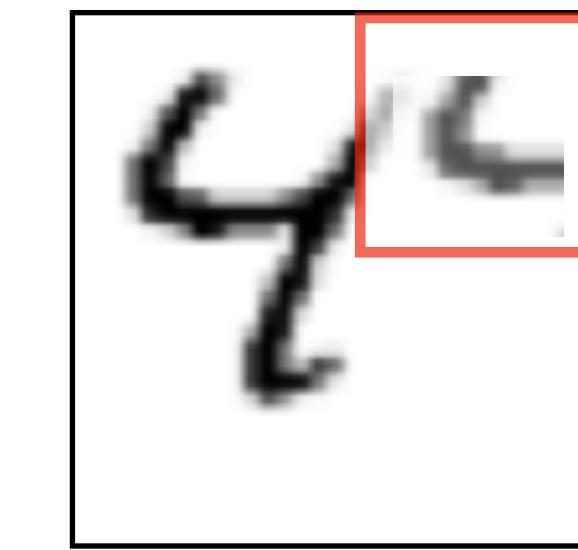
Pass the filter over the image
(Convolutions)



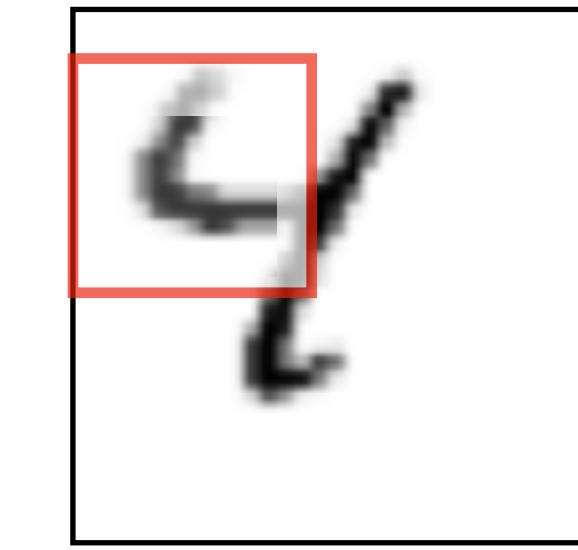
No detection
Output: Low



No detection
Output: Low



No detection
Output: Low



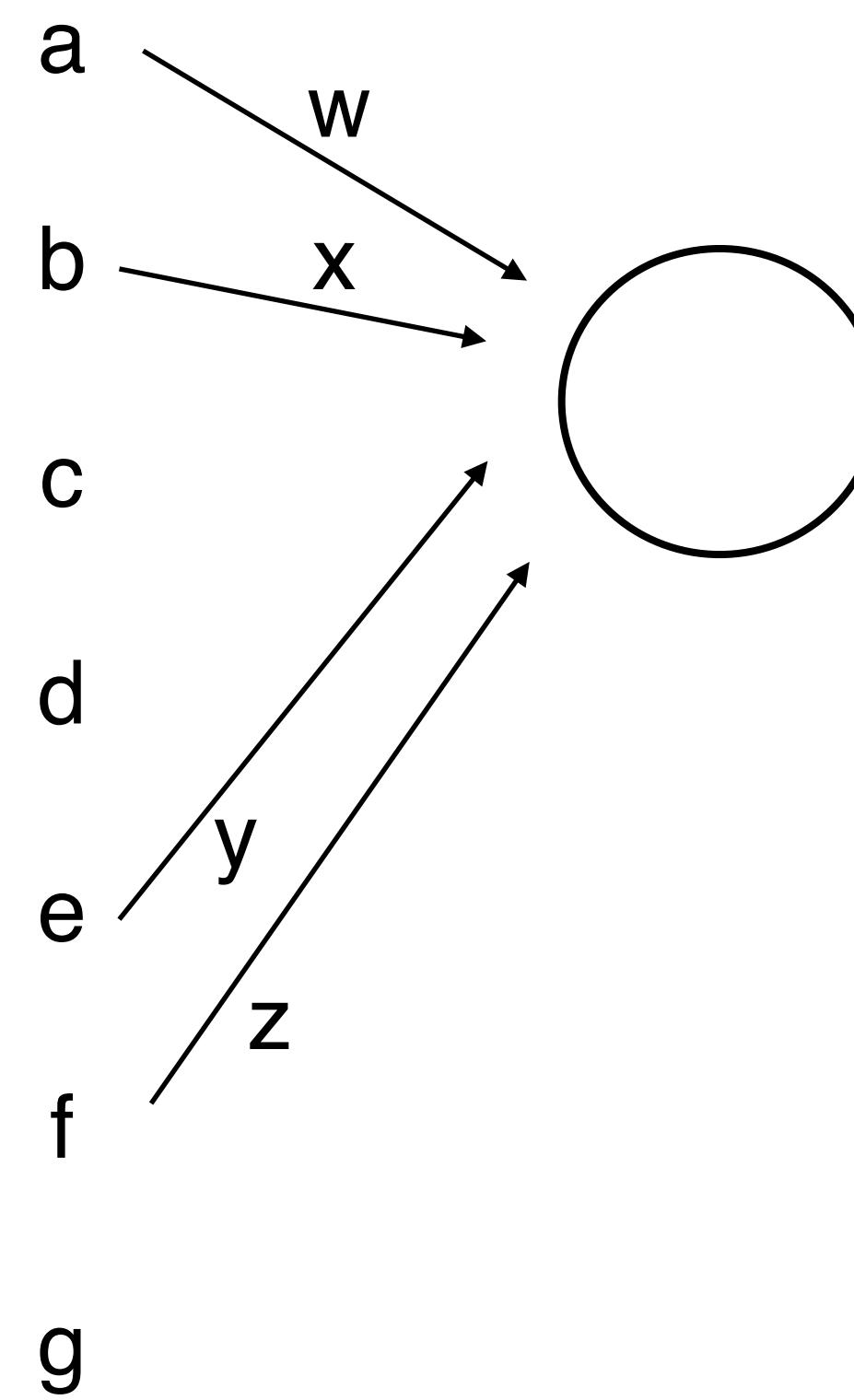
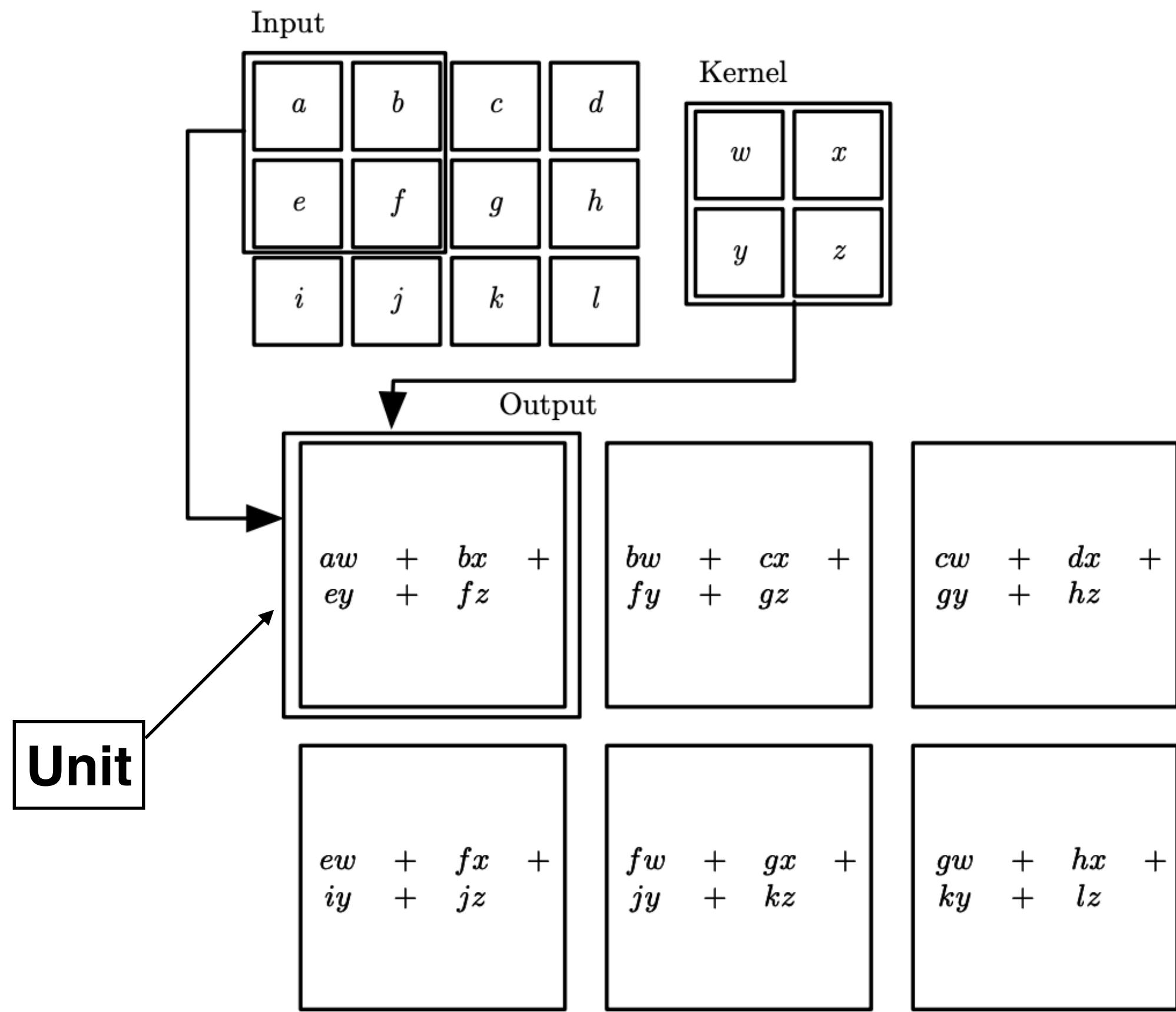
**Detection
Output: High**

Aggregate the output of
the filter
(Pooling)

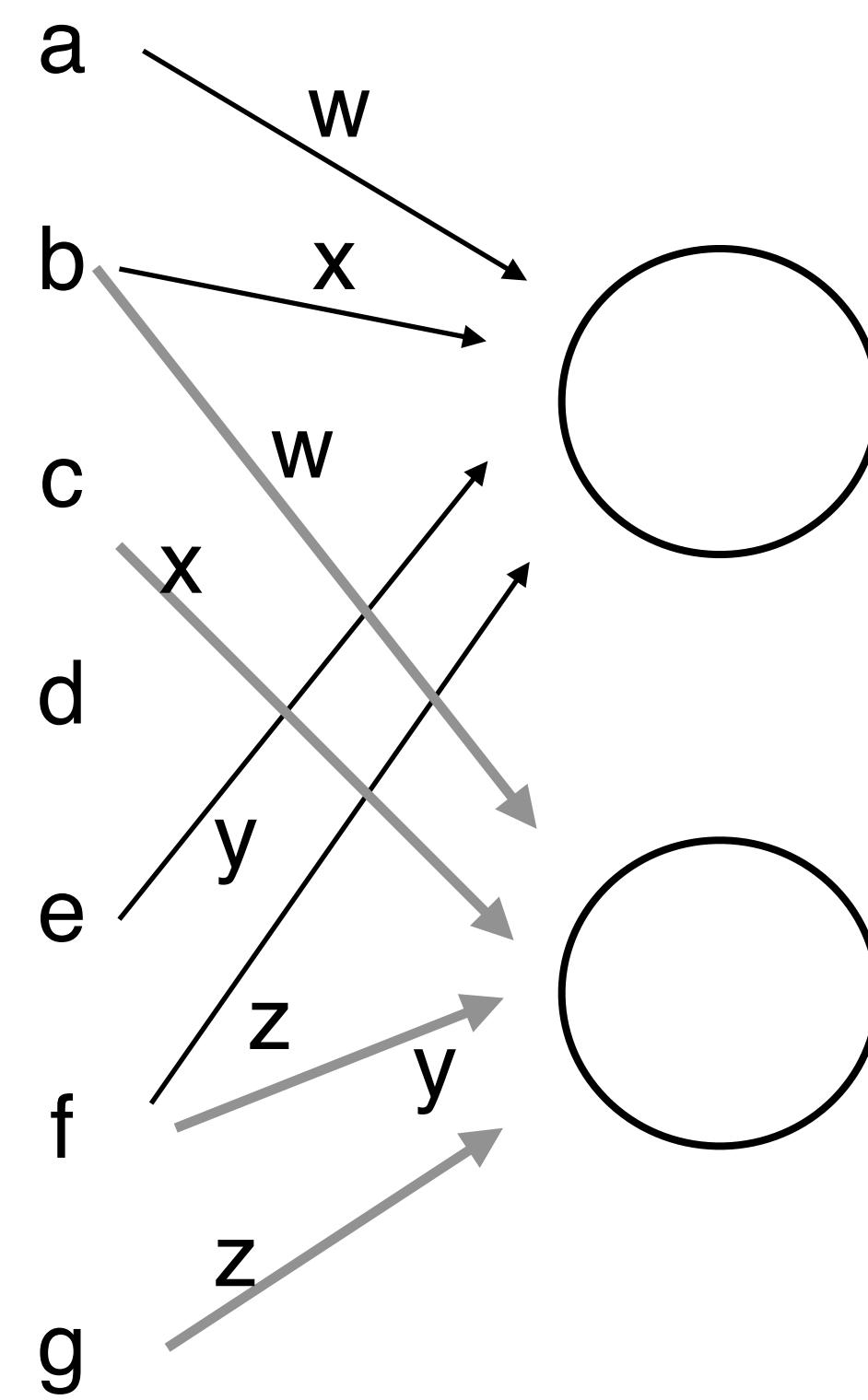
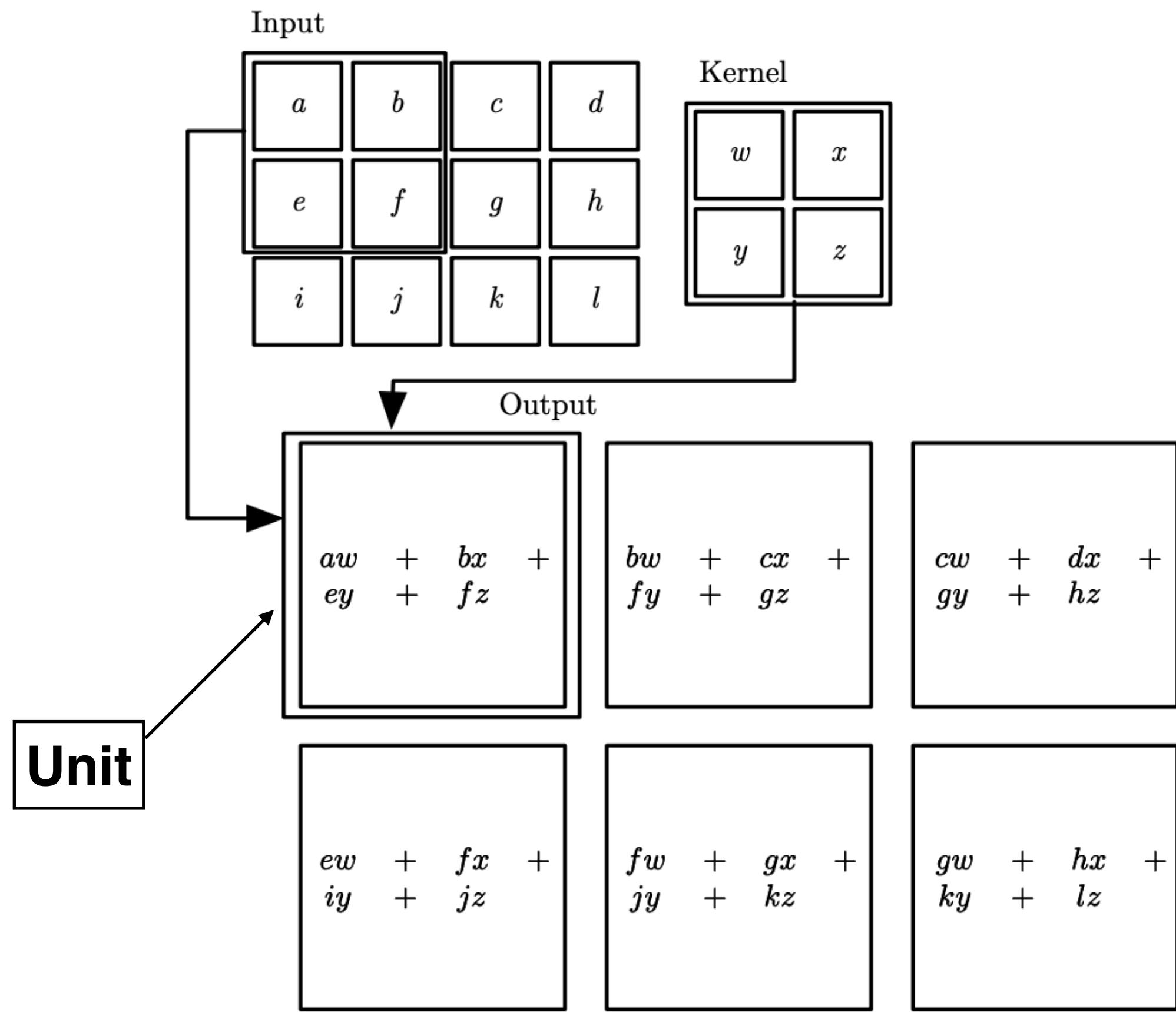
To do well you need to:
1) learn the filters;
2) use many filters.

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

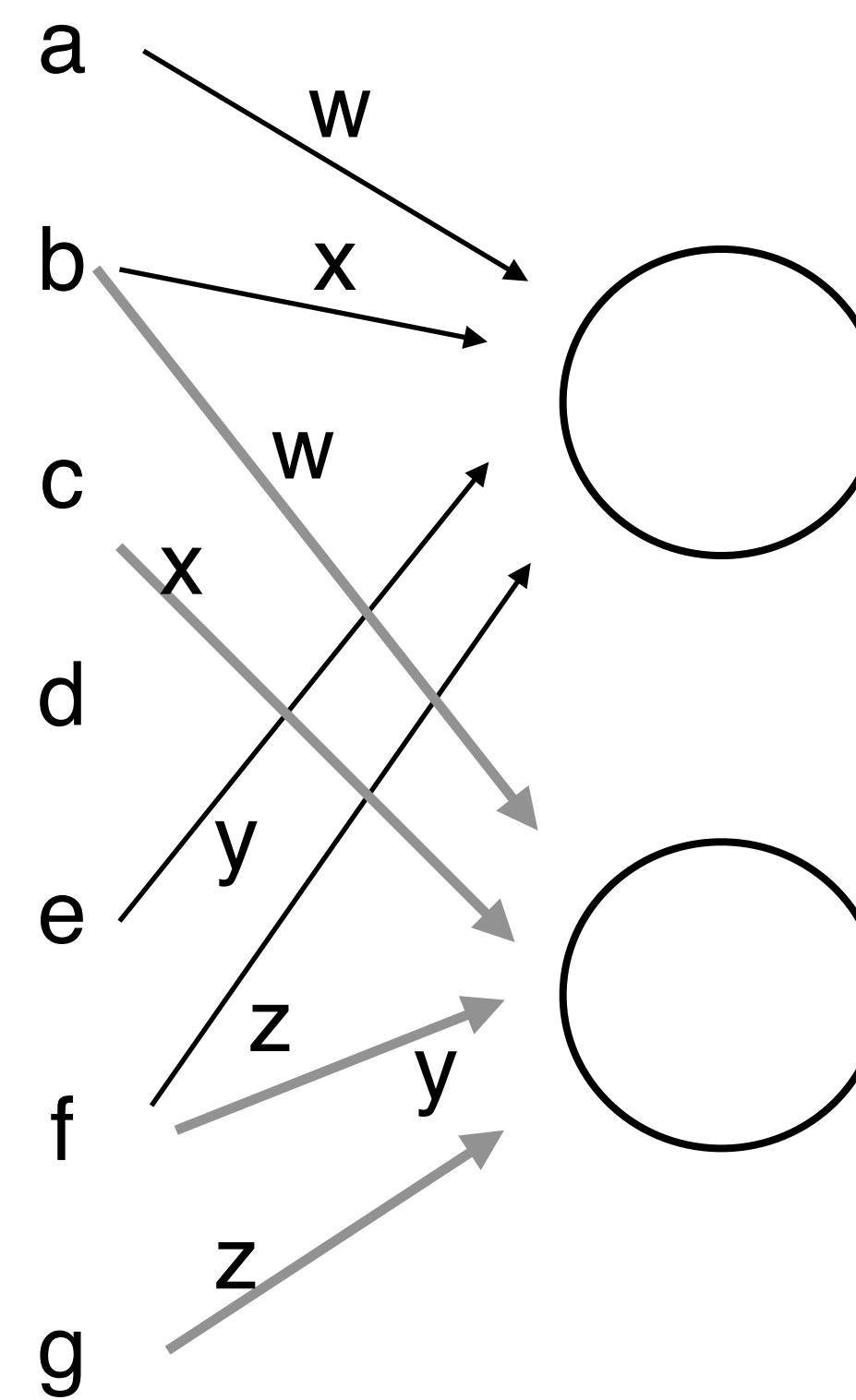
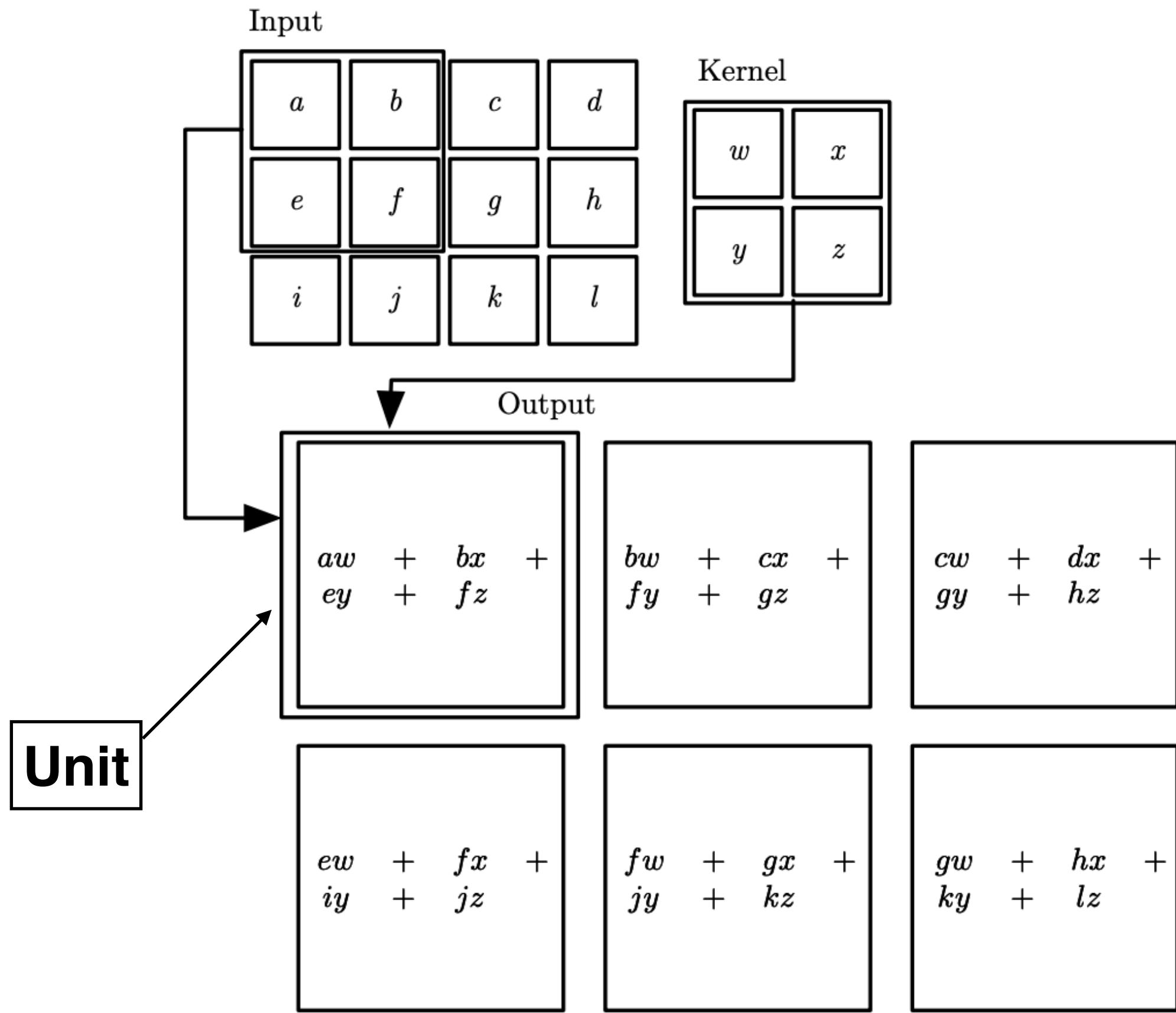
Sparse connections and shared weights



Sparse connections and shared weights



Sparse connections and shared weights



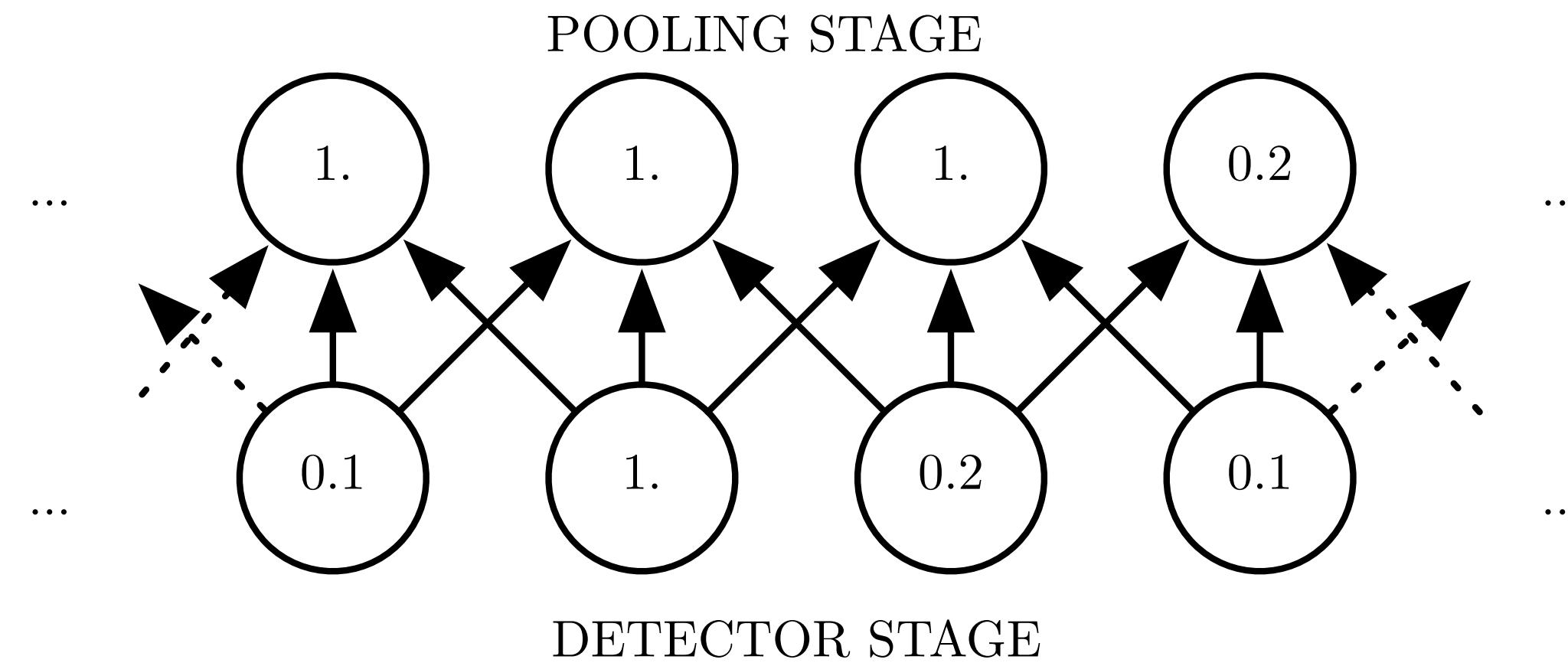
- Kernels induce sparse and shared connections
- Kernels must be small compared to the data

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

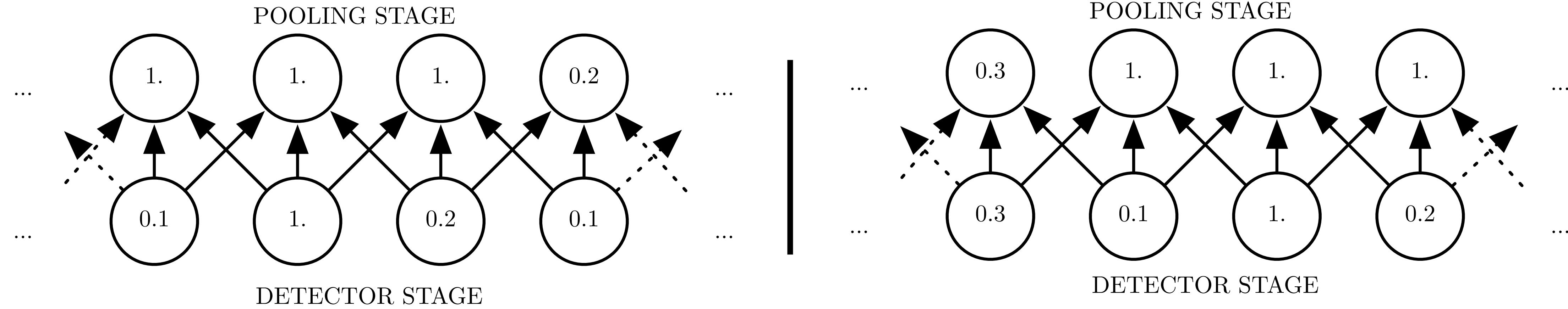
- Make the representation invariant to small translations in the input
 - “Pool” the value of neighbour units
 - E.g., max-pooling takes the max from its input.



[Figure 9.8, Deep Learning, Book]

Pooling

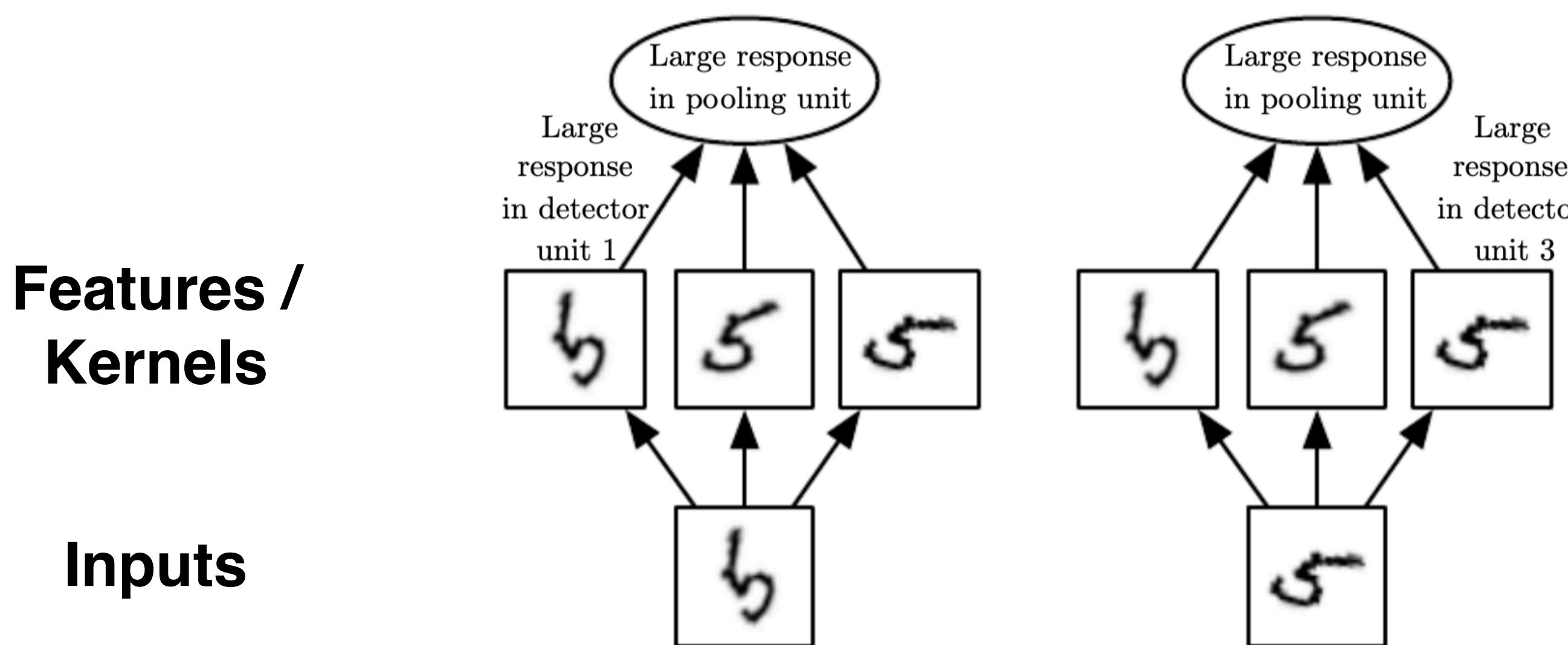
- Make the representation invariant to small translations in the input
 - “Pool” the value of neighbour units
 - E.g., max-pooling takes the max from its input.



[Figure 9.8, Deep Learning, Book]

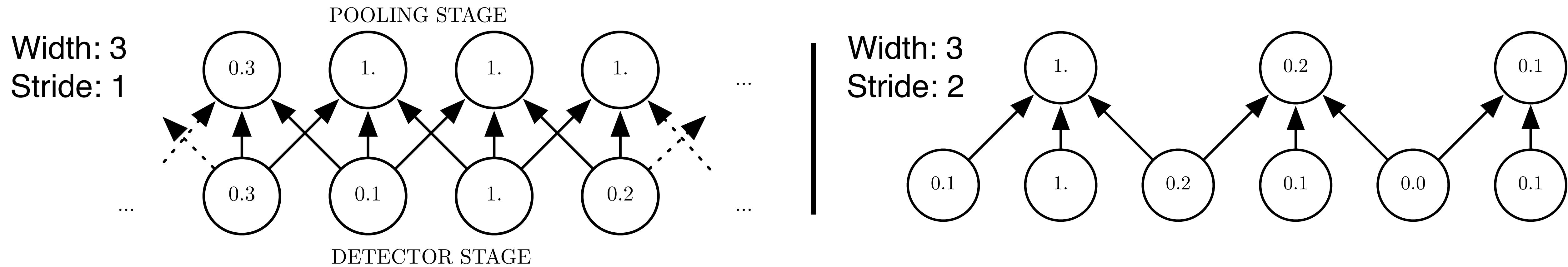
Pooling (II)

- Pooling over different features can learn other types of invariances
- Here, the model learns to be invariant to certain rotations:



Pooling (III)

- Often used to reduce the dimensionality
- Two parameters:
 - **Width:** size of neighbourhood to pool from
 - **Stride:** space between different neighbourhoods



A layer in a CNN

Complex layer terminology

