Deep Learning with Convolutional Neural Network and Recurrent Neural Network

Ashray Bhandare

Agenda

- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

Agenda

- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

Introduction

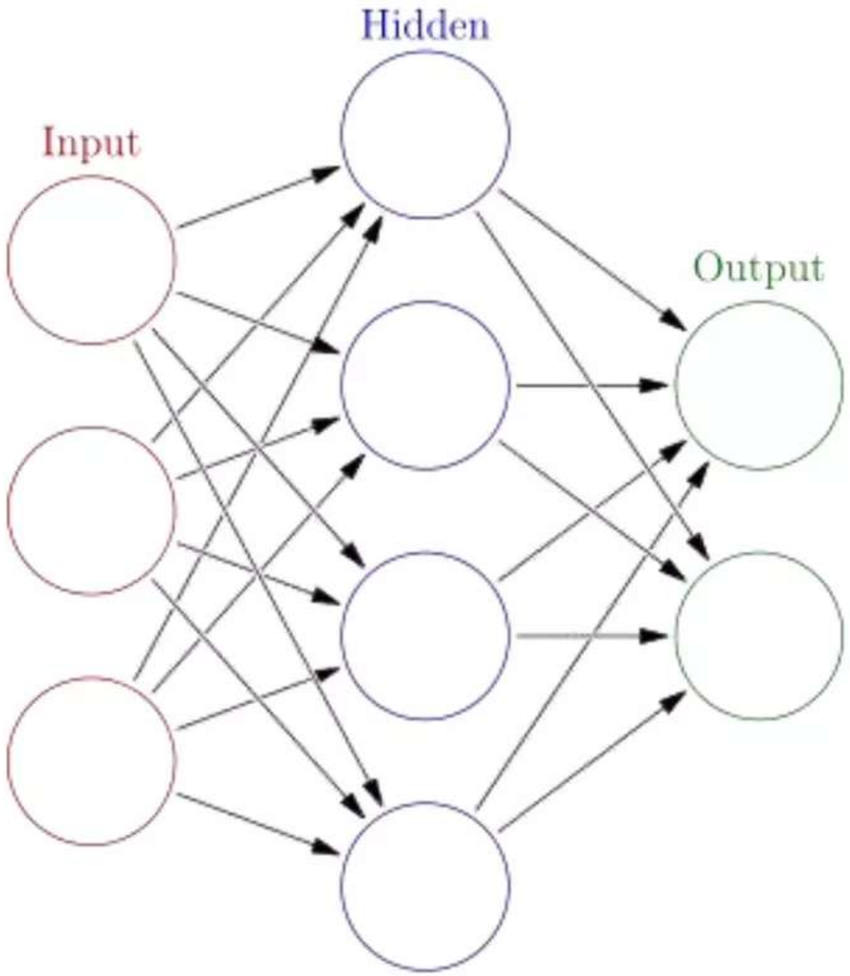
- For a computer to do a certain task, We have to give them a set of instruction that they will follow.

- To give these instruction we should know what the answer is before hand. Therefore the problem cannot be generalized.

- What if we have a problem where we don't know anything about it?

- To overcome this, we use Neural networks as it is good with pattern recognition.

Neural Network Refresher



Backpropagation

$$w = w + \Delta w$$

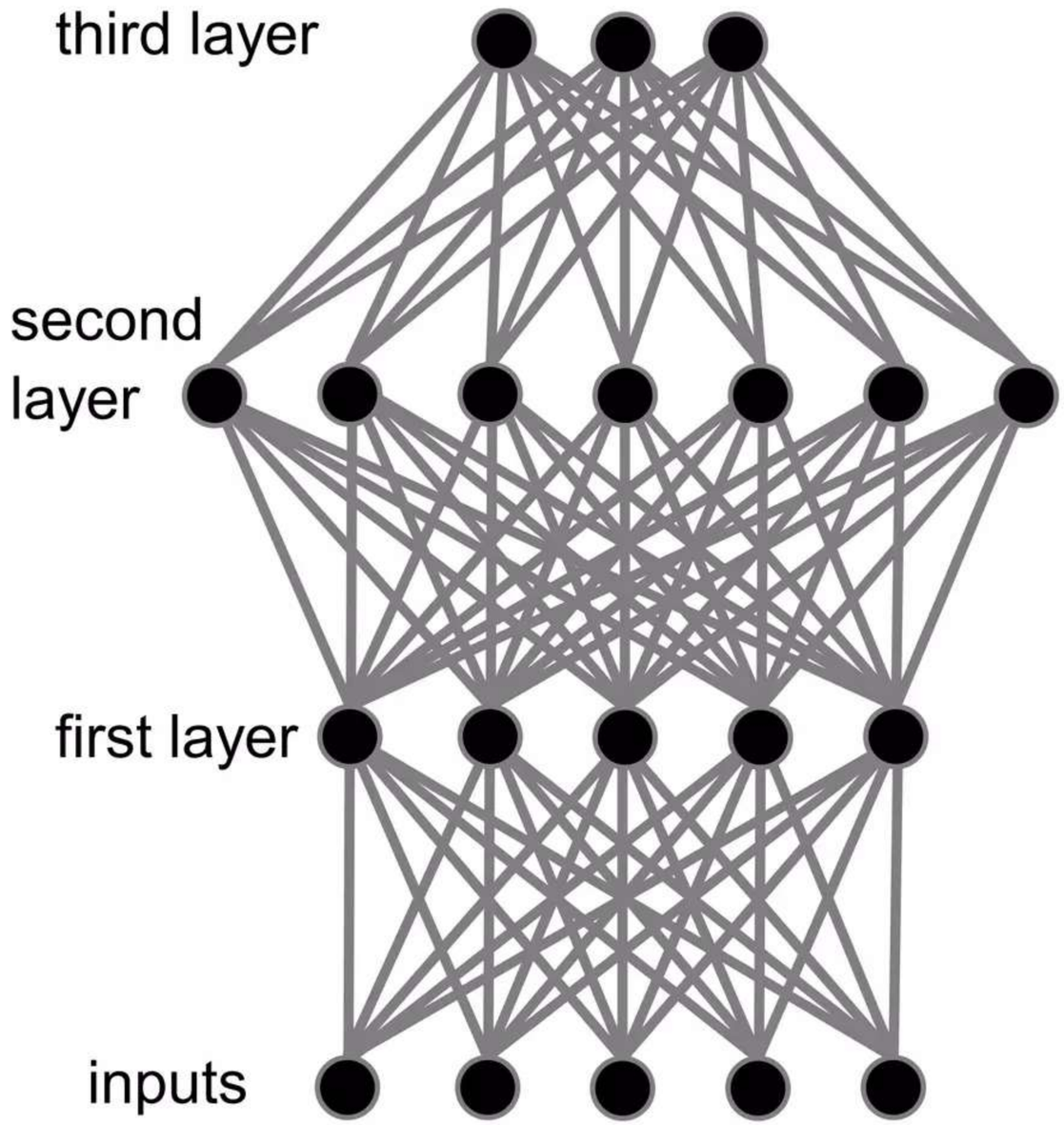
$$\Delta w = n\frac{de}{dw}$$

$$e = (Actual\ Output\ -\ Model\ Output)^2$$

Deep network

If a network has more than three layers, it's deep.

Some 12 or more.



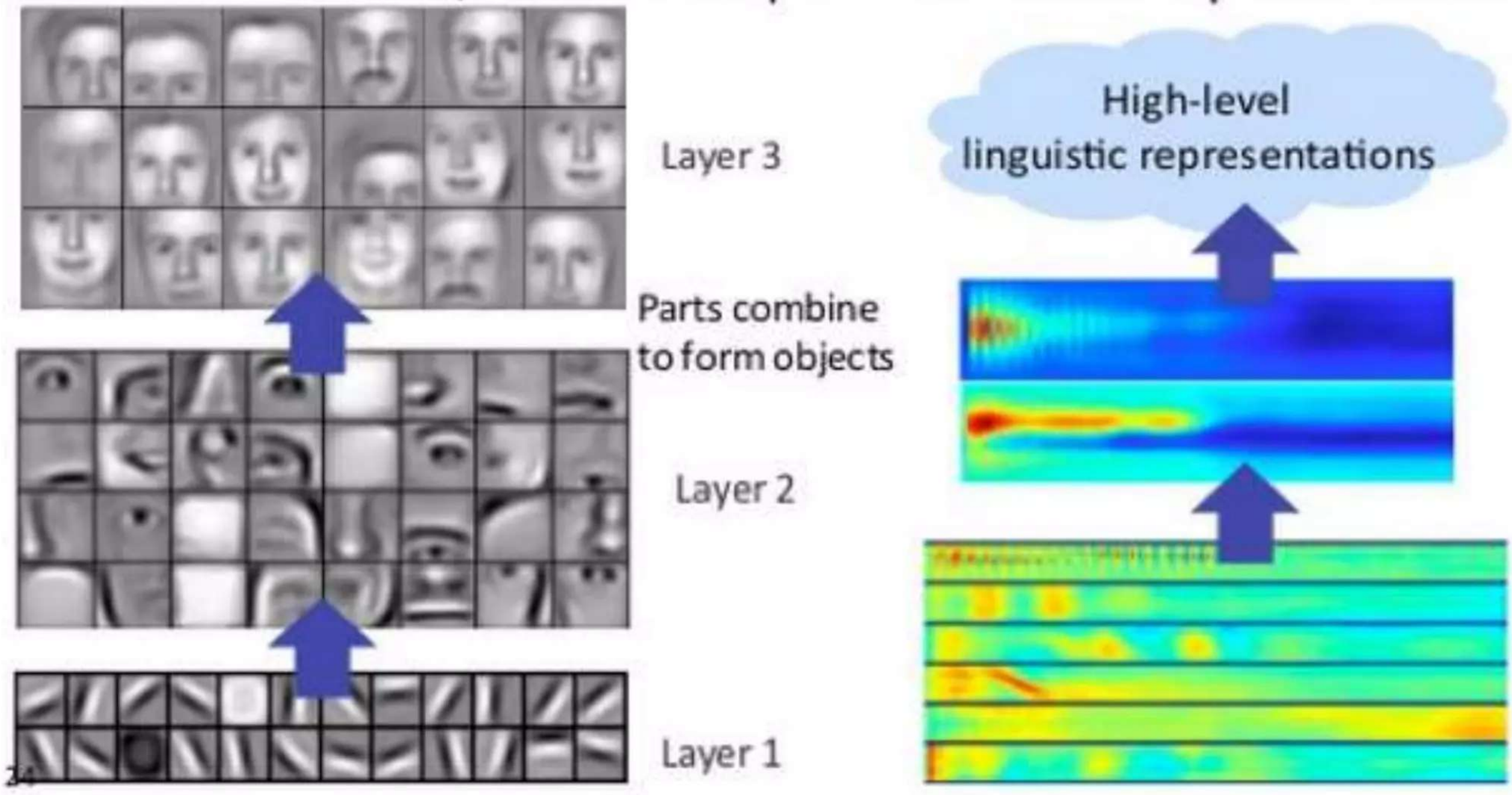
Reasons to go Deep

- Historically, computers have only been useful for tasks that we can explain with a detailed list of instructions.
- Computers fail in applications where the task at hand is fuzzy, such as recognizing patterns.

Pattern Complexity	
Simple	Methods like SVM, regression
Moderate	Neural Networks outperform
Complex	Deep Nets – Practical Choice

Reasons to go Deep

Successive model layers learn deeper intermediate representations



Reasons to go Deep

- A downside of training a deep network is the computational cost.

- The resources required to effectively train a deep net were prohibitive in the early years of neural networks. However, thanks to advances in high-performance GPUs of the last decade, this is no longer an issue

- Complex nets that once would have taken months to train, now only take days.

Agenda

- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

Agenda

- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

How to choose a Deep Net

- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Deep Belief Network (DBN)
- Recursive Neural Tensor Network (RNTN)
- Restricted Boltzmann Machine (RBM)

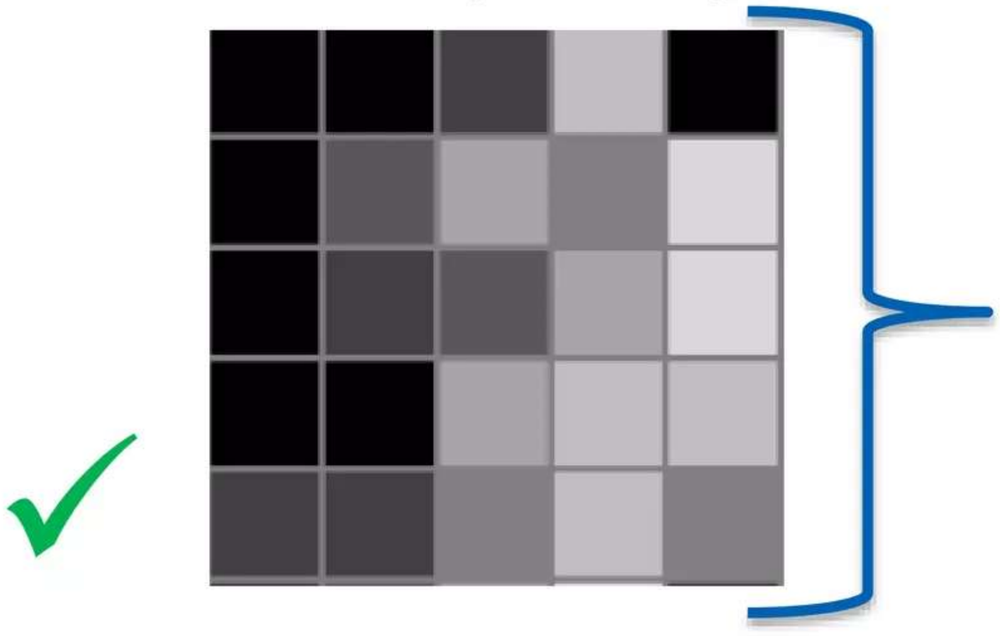
Applications	
Text Processing	RNTN, RNN
Image Recognition	CNN, DBM
Object Recognition	CNN, RNTN
Speech Recognition	RNN
Time series Analysis	RNN
Unlabeled data – pattern recognition	RBM

Agenda

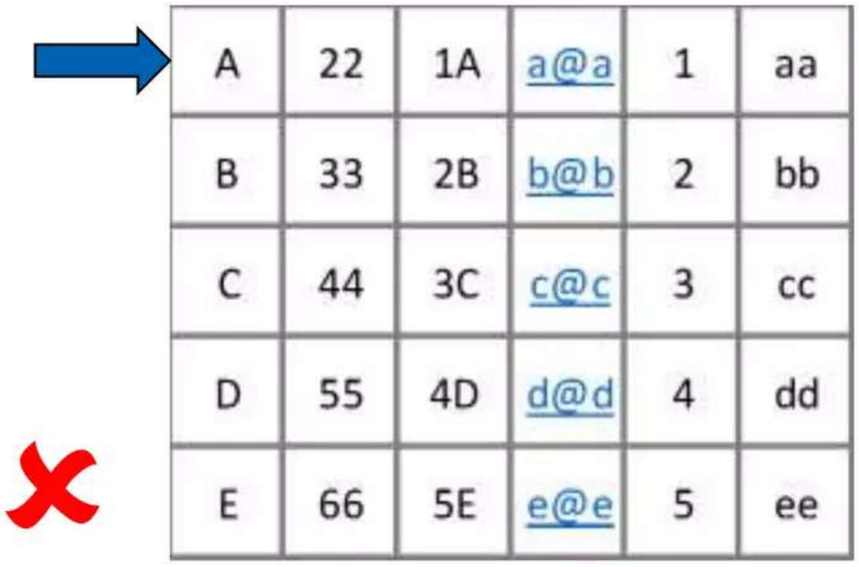
- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

Introduction

- A convolutional neural network (or ConvNet) is a type of feed-forward artificial neural network
- The architecture of a ConvNet is designed to take advantage of the 2D structure of an input image.



VS



A	22	1A	a@a	1	aa
B	33	2B	b@b	2	bb
C	44	3C	c@c	3	cc
D	55	4D	d@d	4	dd
E	66	5E	e@e	5	ee

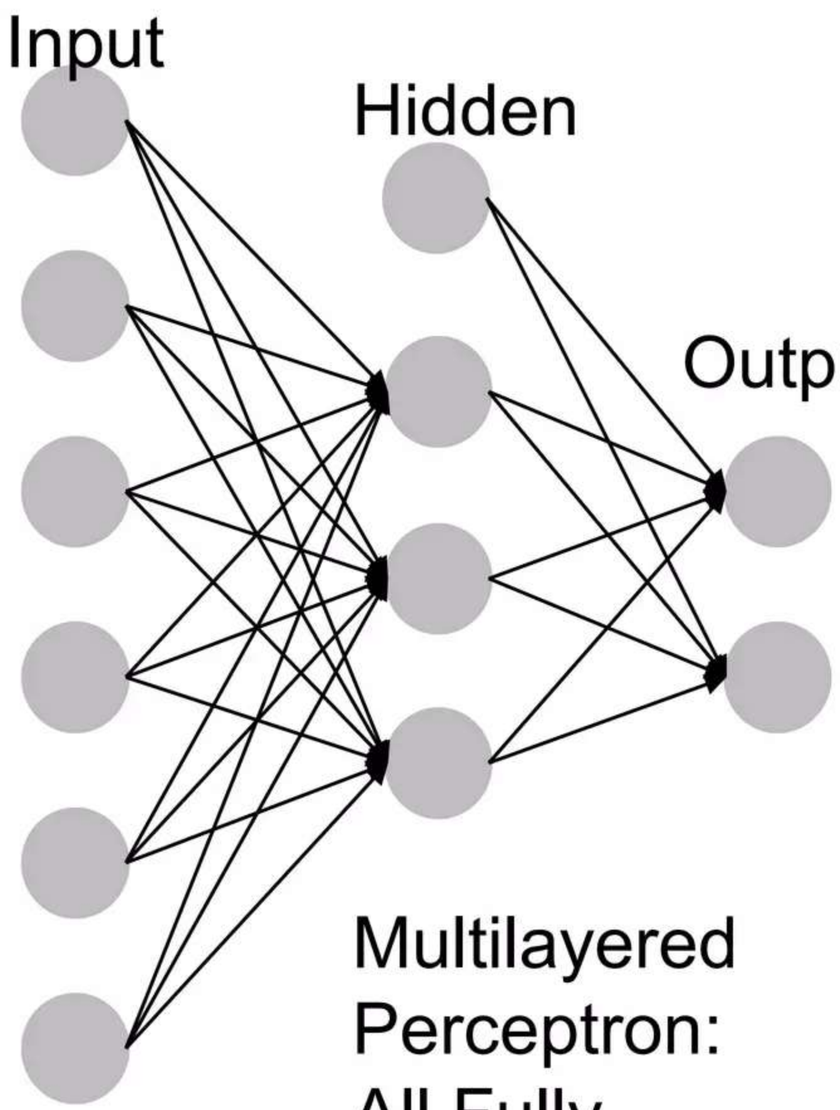
- A ConvNet is comprised of one or more convolutional layers (often with a pooling step) and then followed by one or more fully connected layers as in a standard multilayer neural network.

Motivation behind ConvNets

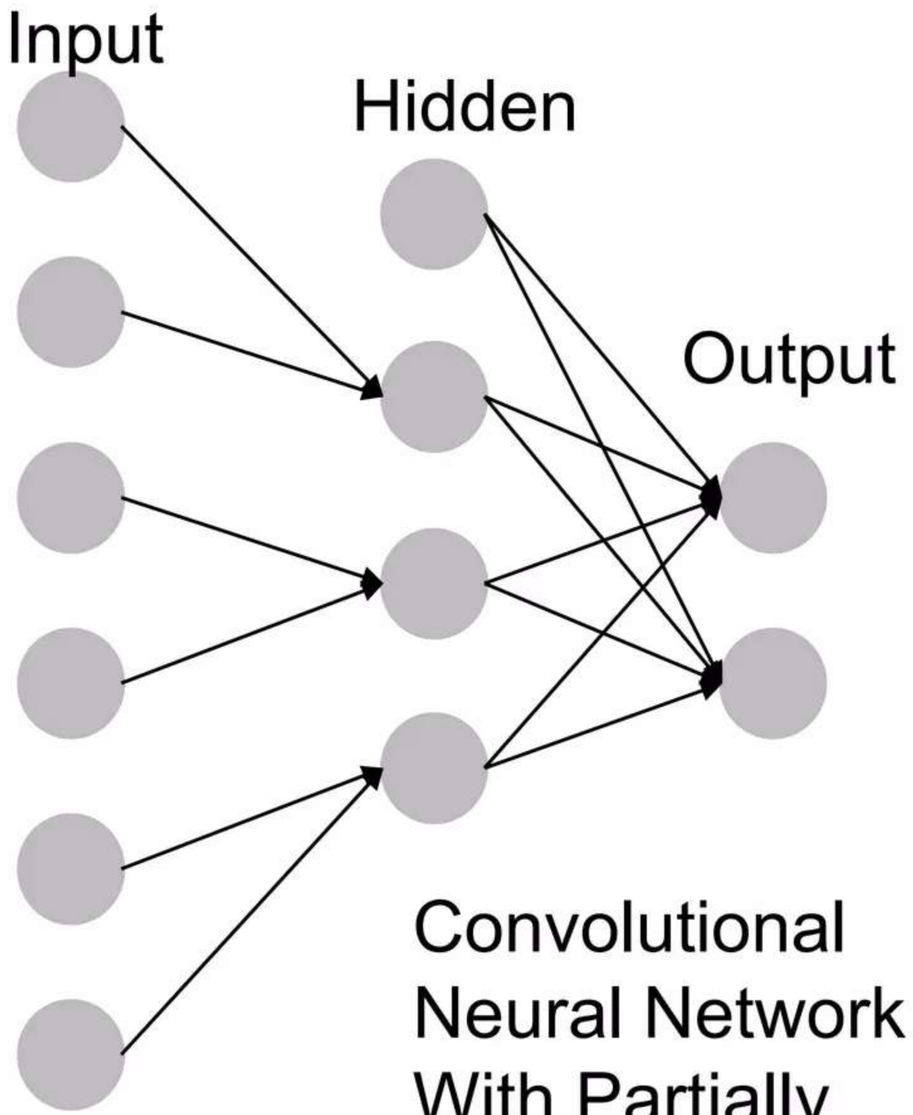
- Consider an image of size 200x200x3 (200 wide, 200 high, 3 color channels)
  - a **single fully-connected neuron** in a first hidden layer of a regular Neural Network would have 200*200*3 = 120,000 weights.
  - Due to the presence of several such neurons, this full connectivity is wasteful and the huge number of parameters would quickly lead to overfitting

- However, in a ConvNet, the neurons in a layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully-connected manner.
  - the final output layer would have dimensions 1x1xN, because by the end of the ConvNet architecture we will reduce the full image into a single vector of class scores (for N classes), arranged along the depth dimension

MLP VS ConvNet



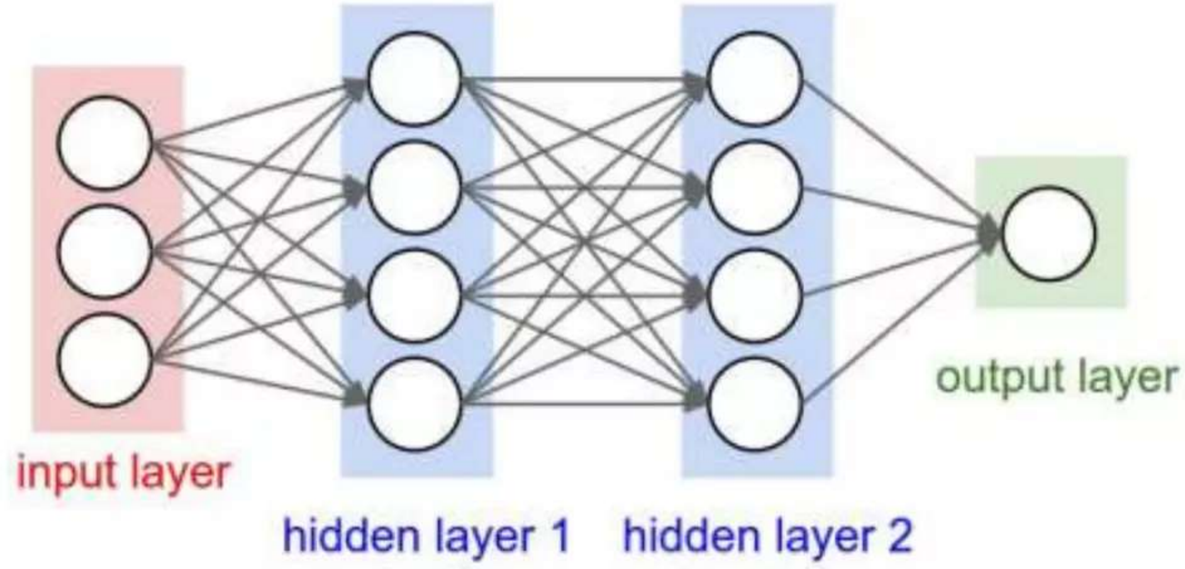
Multilayered Perceptron: All Fully Connected Layers



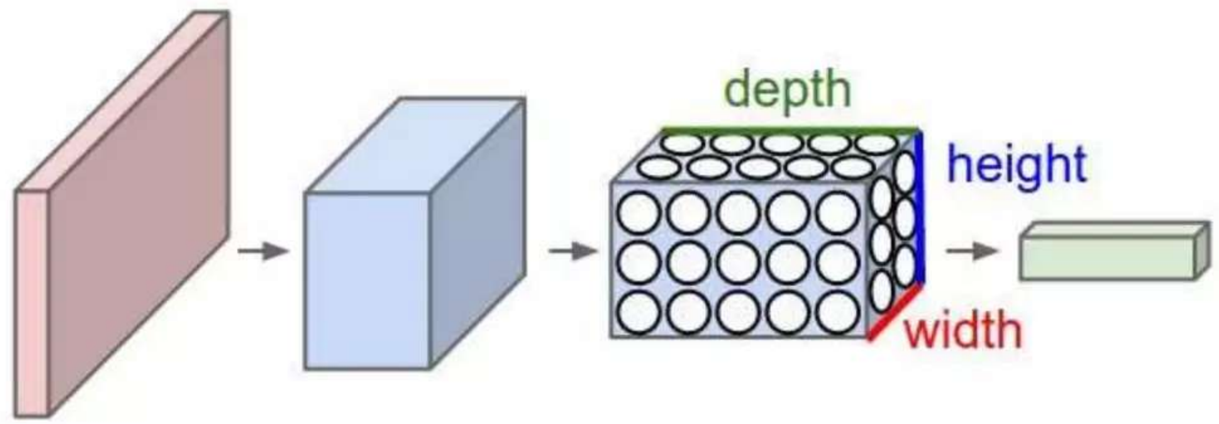
Convolutional Neural Network With Partially Connected Convolution Layer

MLP vs ConvNet

- A regular 3-layer Neural Network.



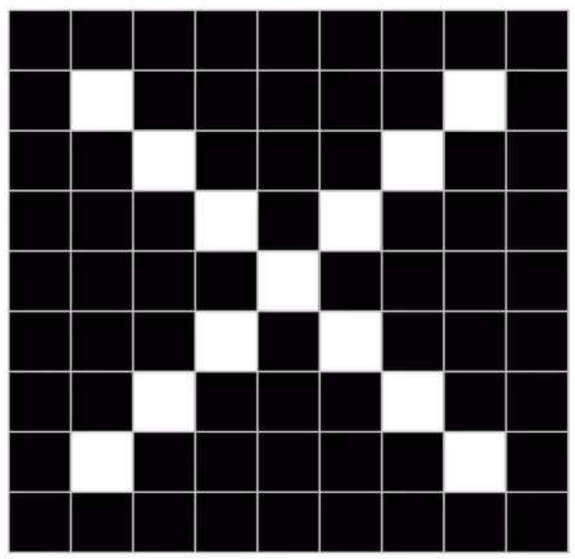
- A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers.



How ConvNet Works

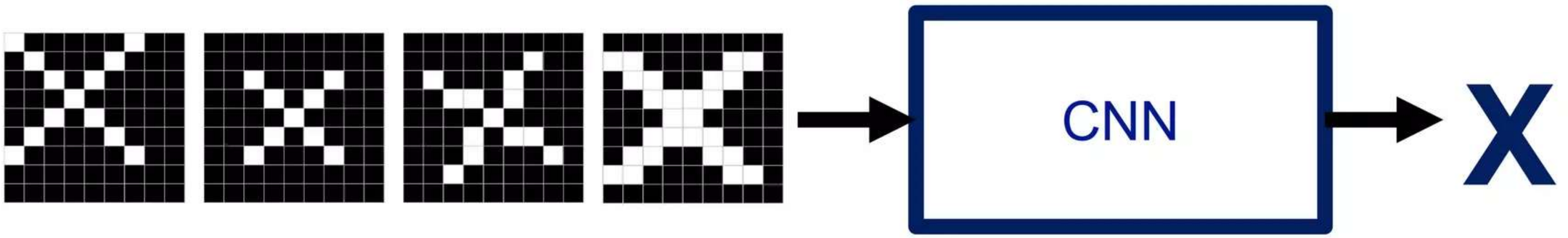
- For example, a ConvNet takes the input as an image which can be classified as ‘X’ or ‘O’

- In a simple case, ‘X’ would look like:



How ConvNet Works

- What about trickier cases?



How ConvNet Works – What Computer Sees

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	1	-1	-1	-1
-1	-1	1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	1	-1	-1
-1	-1	-1	1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

How ConvNet Works

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	1	-1	-1	-1
-1	-1	1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	1	-1	-1
-1	-1	-1	1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

How ConvNet Works – What Computer Sees

- Since the pattern doesnot match exactly, the computer will not be able to classify this as ‘X’

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	X	-1	-1	-1	-1	X	X	-1
-1	X	X	-1	-1	X	X	-1	-1
-1	-1	X	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	X	-1	-1
-1	-1	X	X	-1	-1	X	X	-1
-1	X	X	-1	-1	-1	-1	X	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Agenda

- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

ConvNet Layers (At a Glance)

- CONV layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume.

- RELU layer will apply an elementwise activation function, such as the max(0,x) thresholding at zero. This leaves the size of the volume unchanged.

- POOL layer will perform a downsampling operation along the spatial dimensions (width, height).

- FC (i.e. fully-connected) layer will compute the class scores, resulting in volume of size [1x1xN], where each of the N numbers correspond to a class score, such as among the N categories.

Recall – What Computer Sees

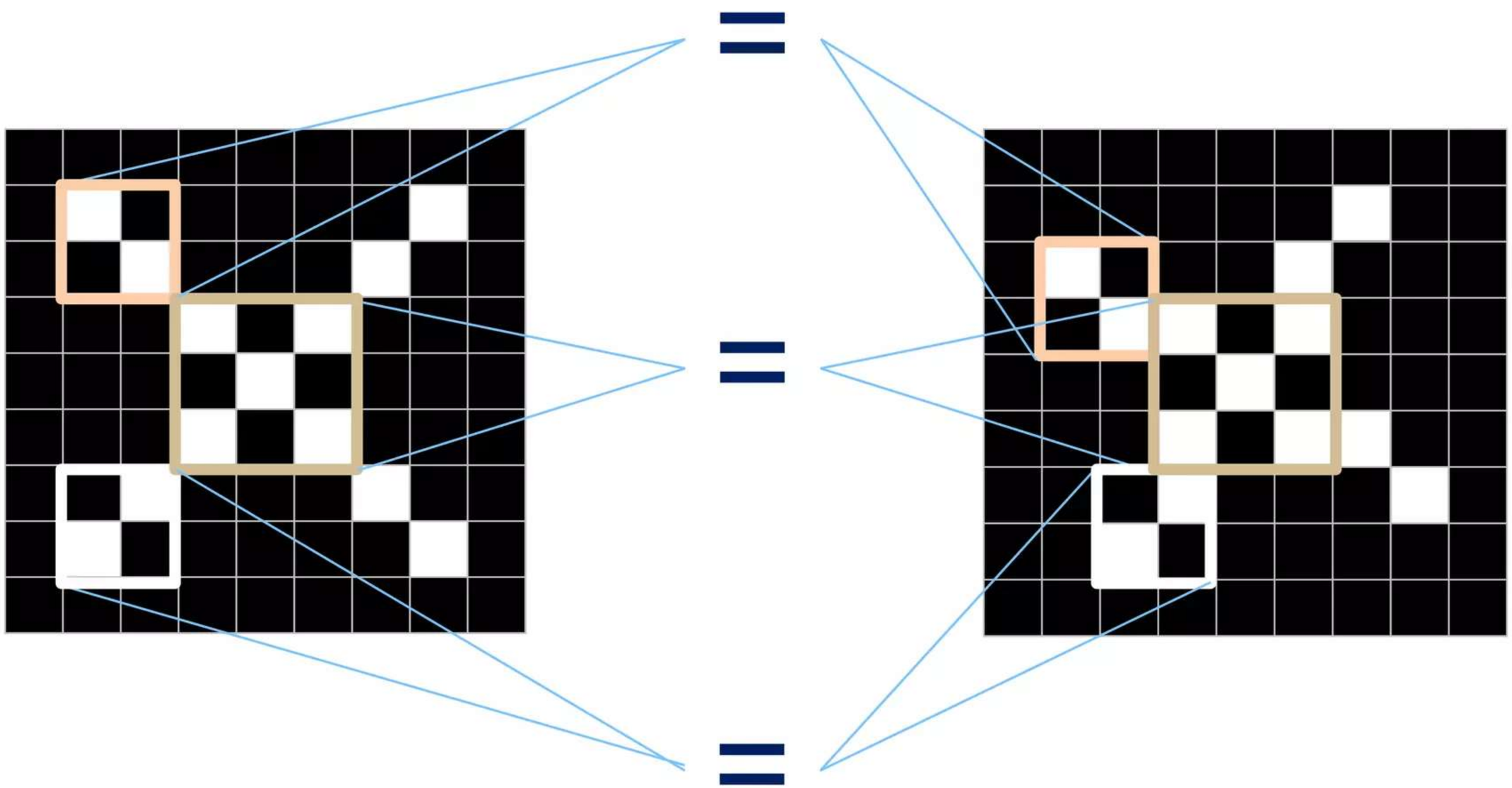
- Since the pattern doesnot match exactly, the computer will not be able to classify this as ‘X’

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	X	-1	-1	-1	-1	X	X	-1
-1	X	X	-1	-1	X	X	-1	-1
-1	-1	X	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	X	-1	-1
-1	-1	X	X	-1	-1	X	X	-1
-1	X	X	-1	-1	-1	-1	X	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

- What got changed?

Convolutional Layer

- Convolution layer will work to identify patterns (features) instead of individual pixels



Convolutional Layer - Filters

- The CONV layer's parameters consist of a set of learnable filters.
- Every filter is small spatially (along width and height), but extends through the full depth of the input volume.
- During the forward pass, we slide (more precisely, convolve) each filter across the width and height of the input volume and compute dot products between the entries of the filter and the input at any position.

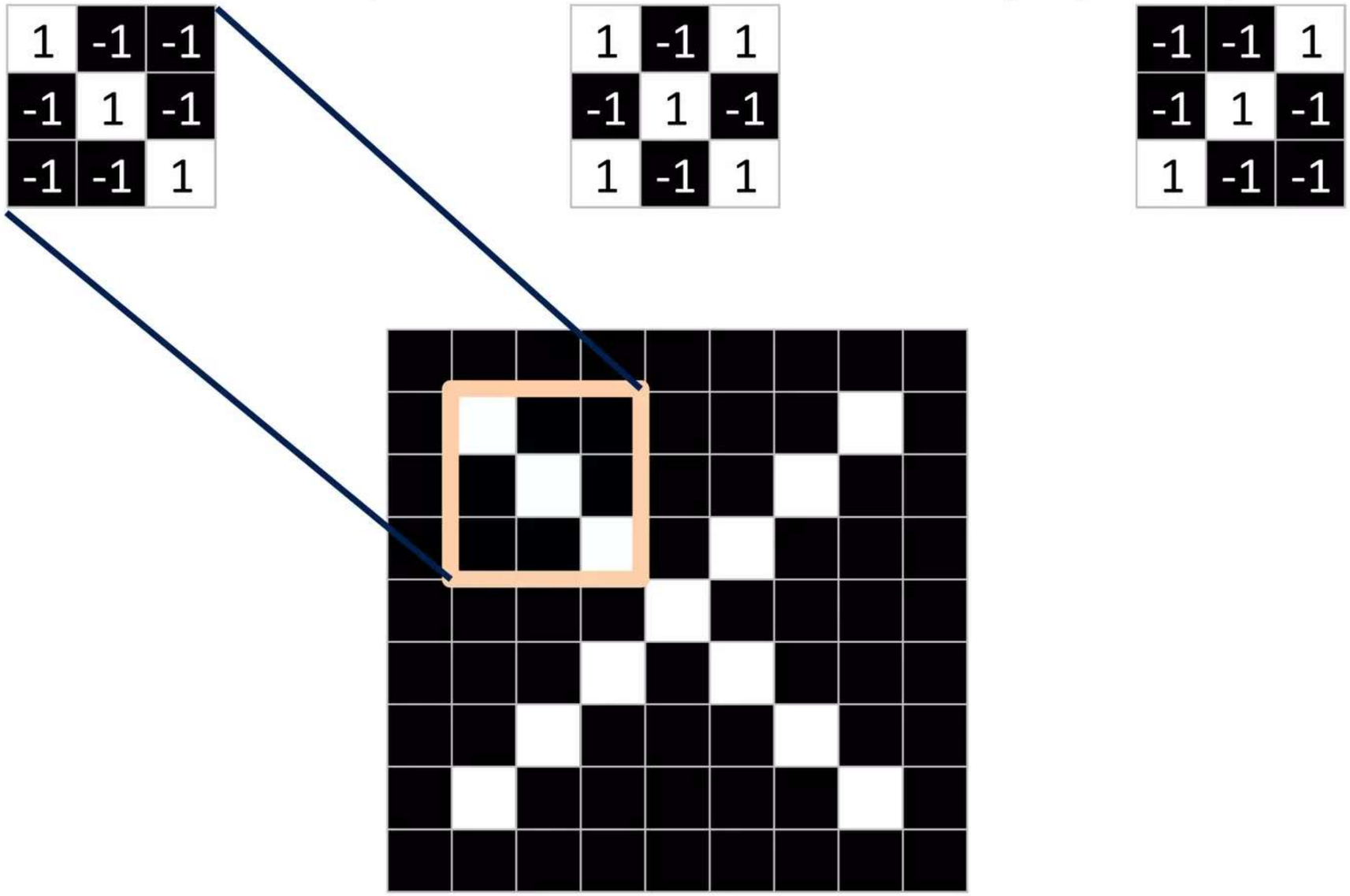
1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

-1	-1	1
-1	1	-1
1	-1	-1

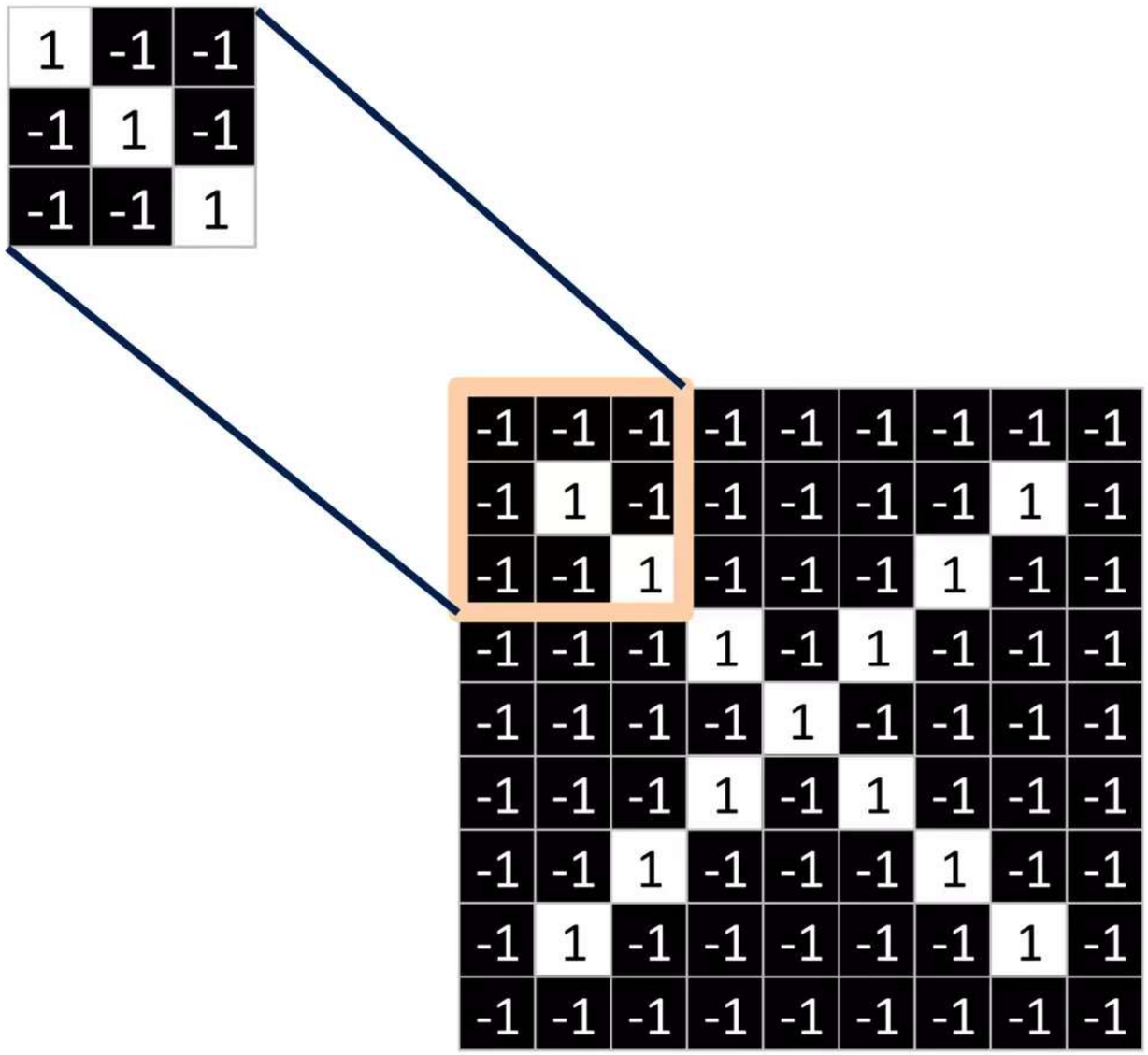
Convolutional Layer - Filters

- Sliding the filter over the width and height of the input gives 2-dimensional activation map that responds to that filter at every spatial position.

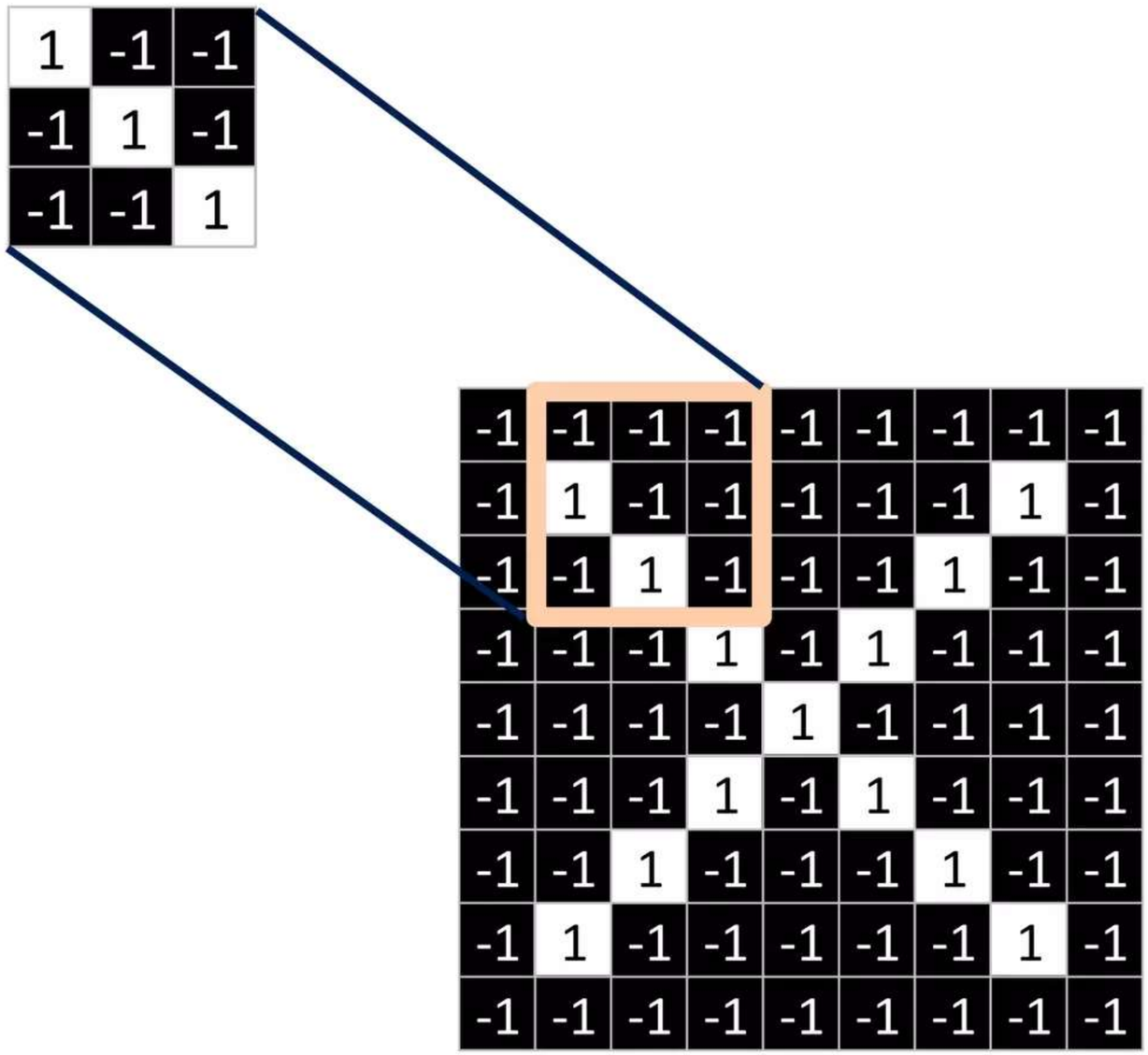


Convolutional Layer – Filters – Navigation Example

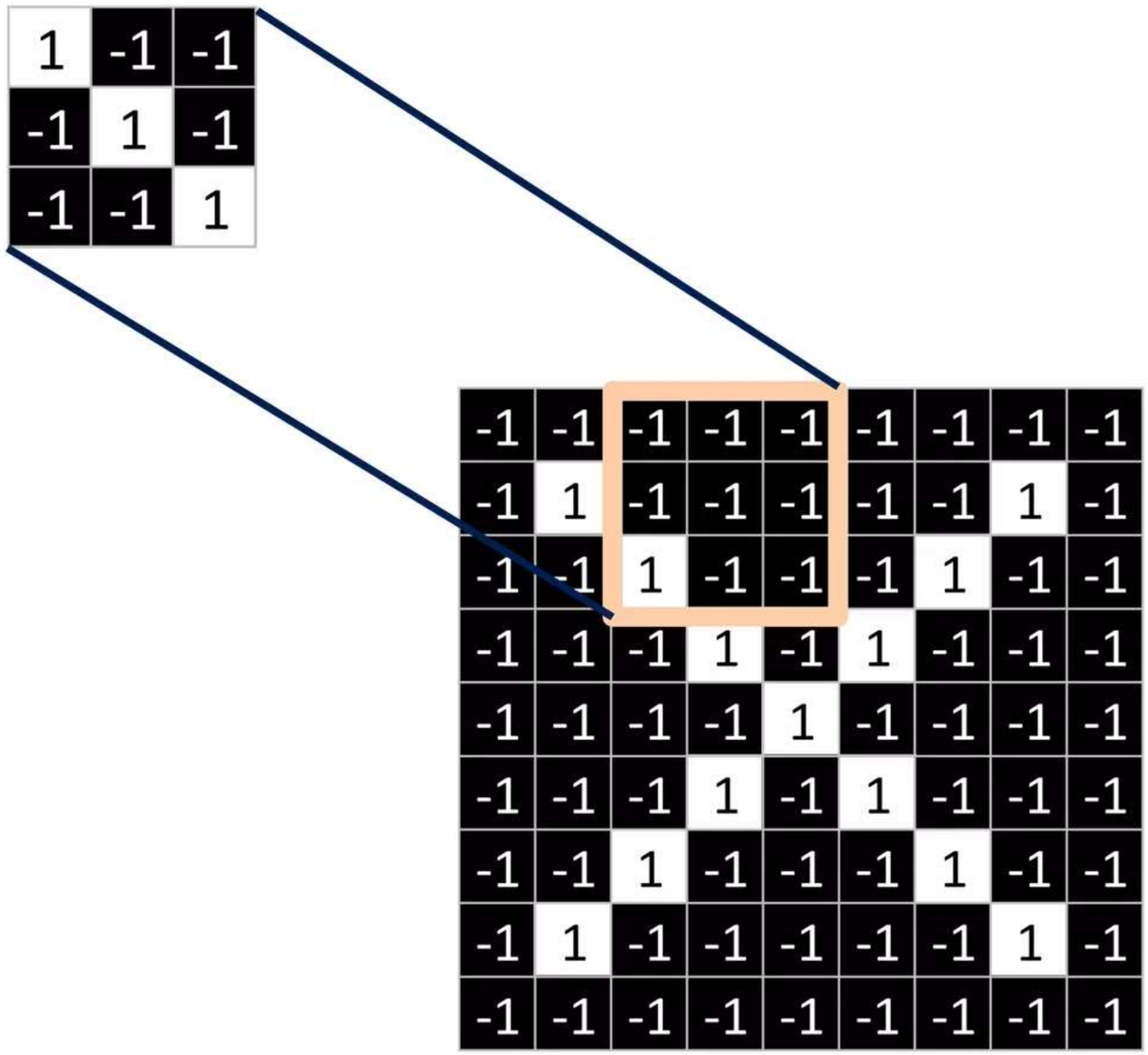
- Strides = 1, Filter Size = 3 X 3 X 1, Padding = 0



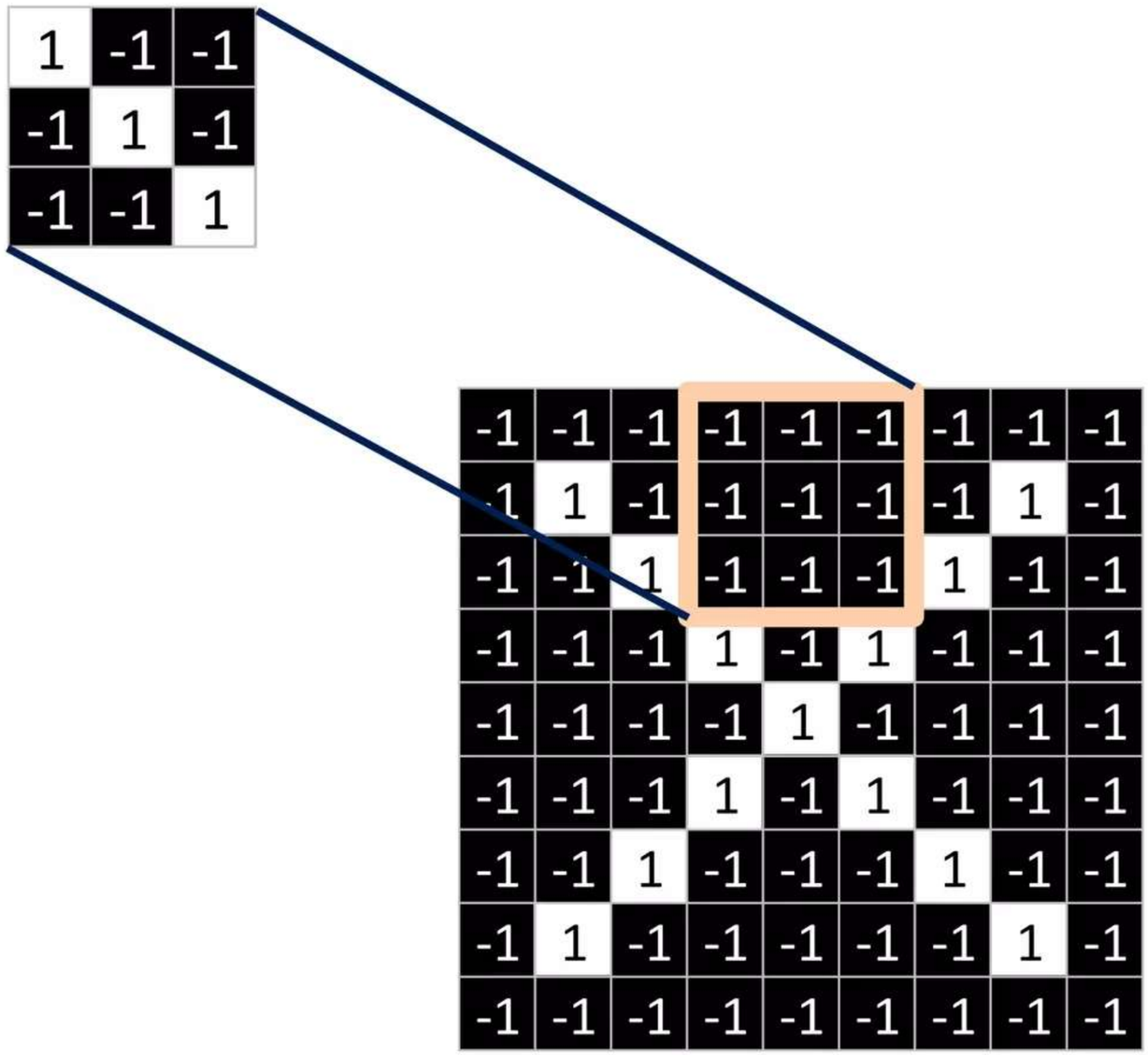
Convolutional Layer – Filters – Navigation Example



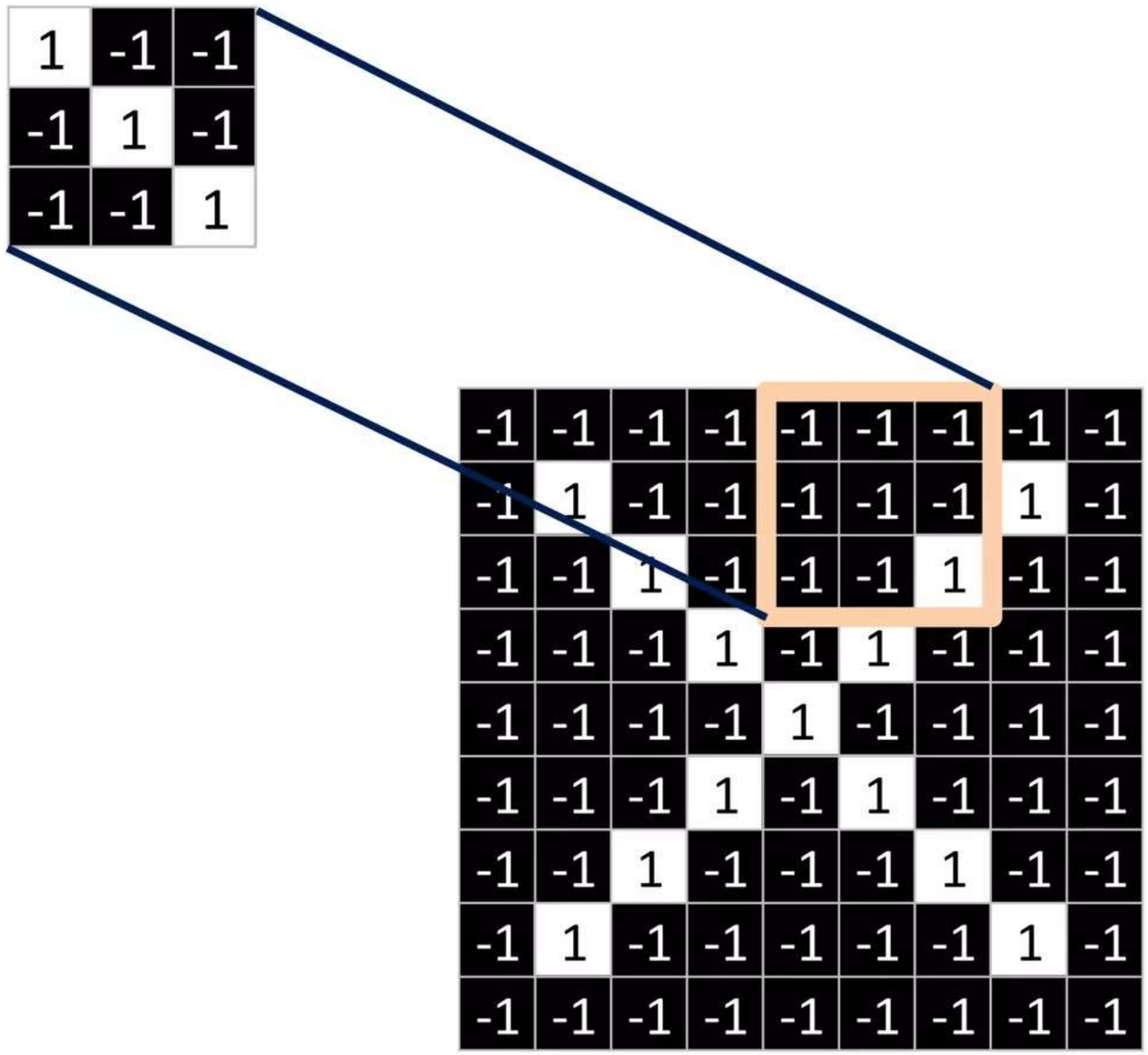
Convolutional Layer – Filters – Navigation Example



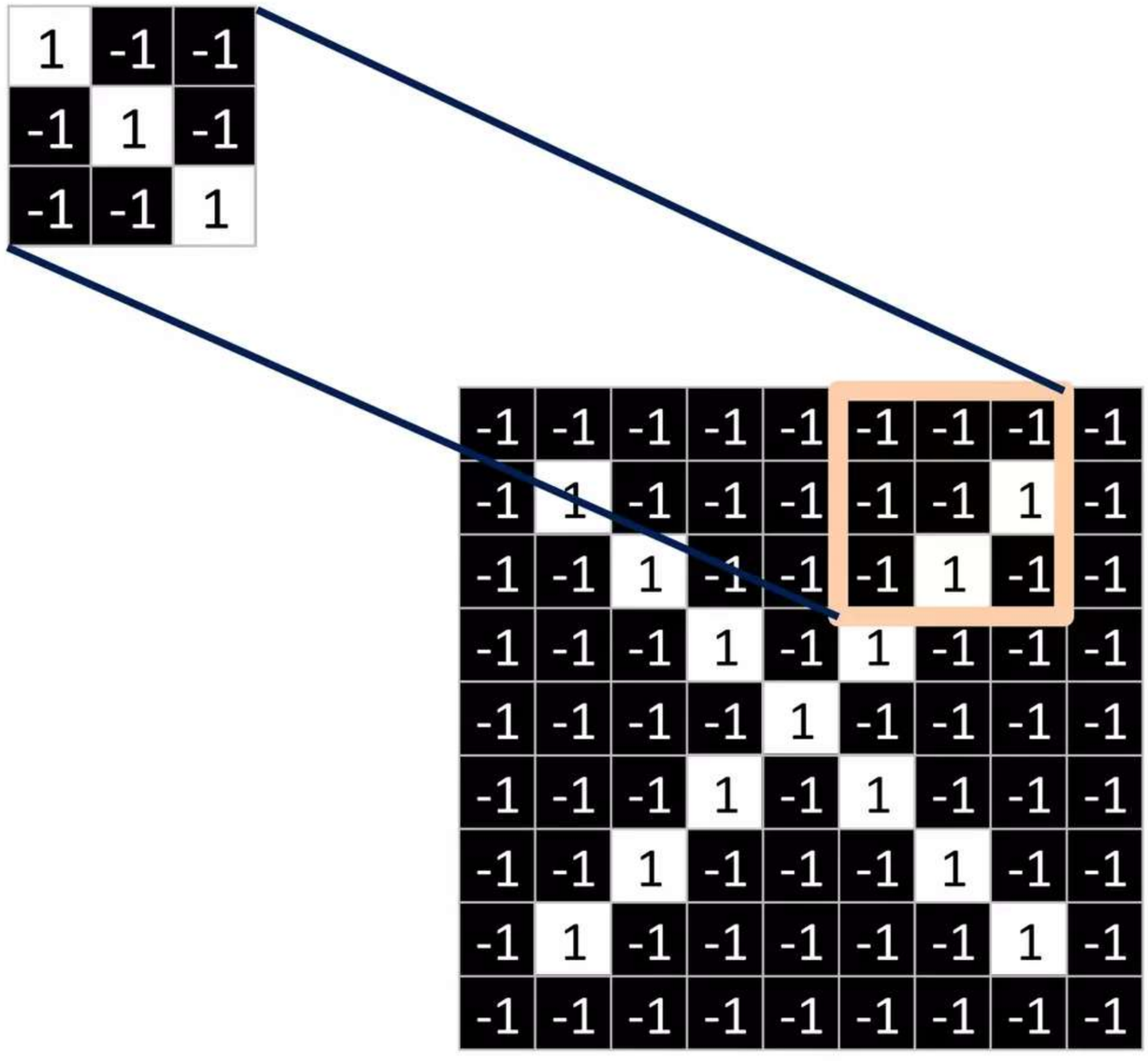
Convolutional Layer – Filters – Navigation Example



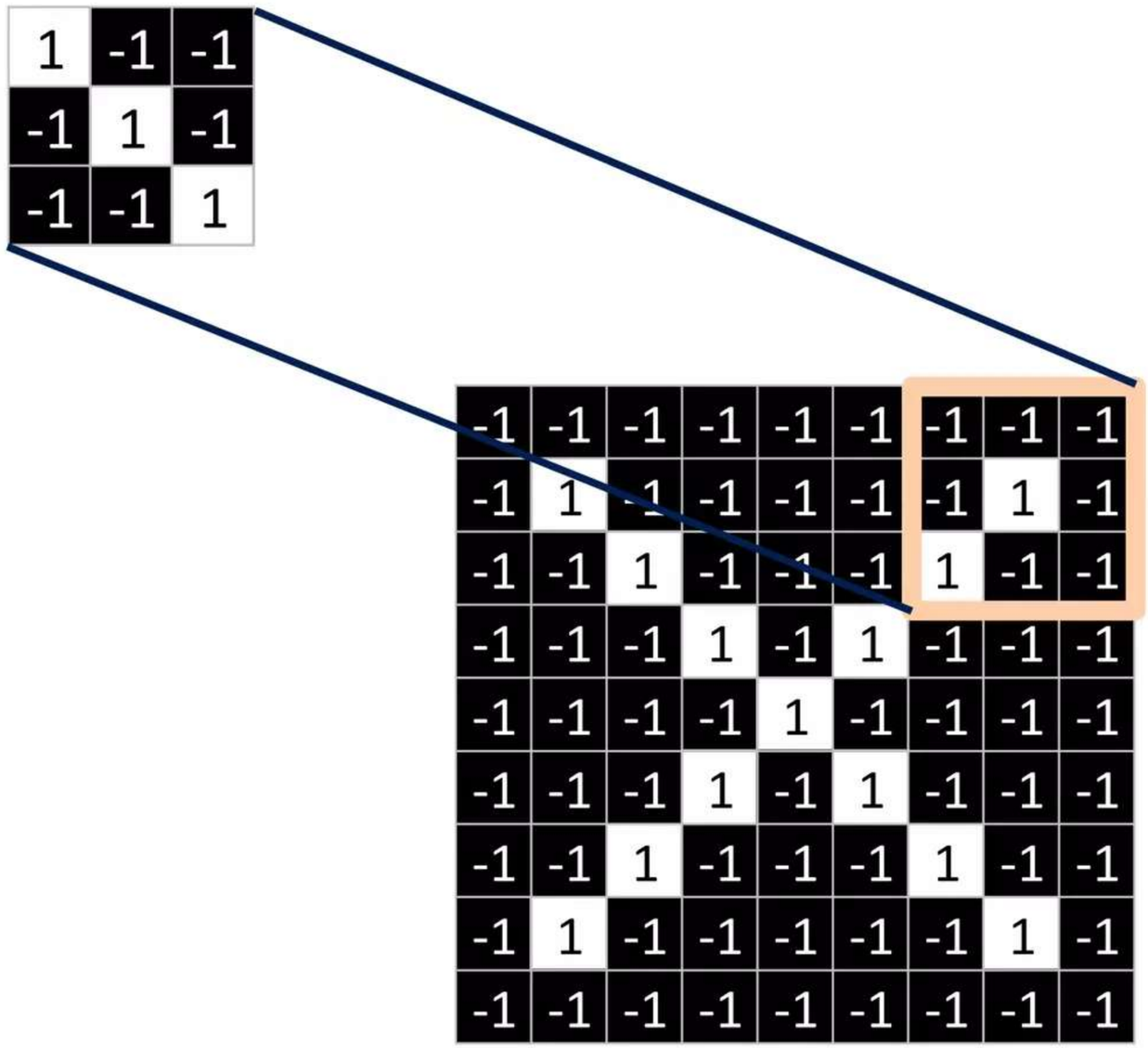
Convolutional Layer – Filters – Navigation Example



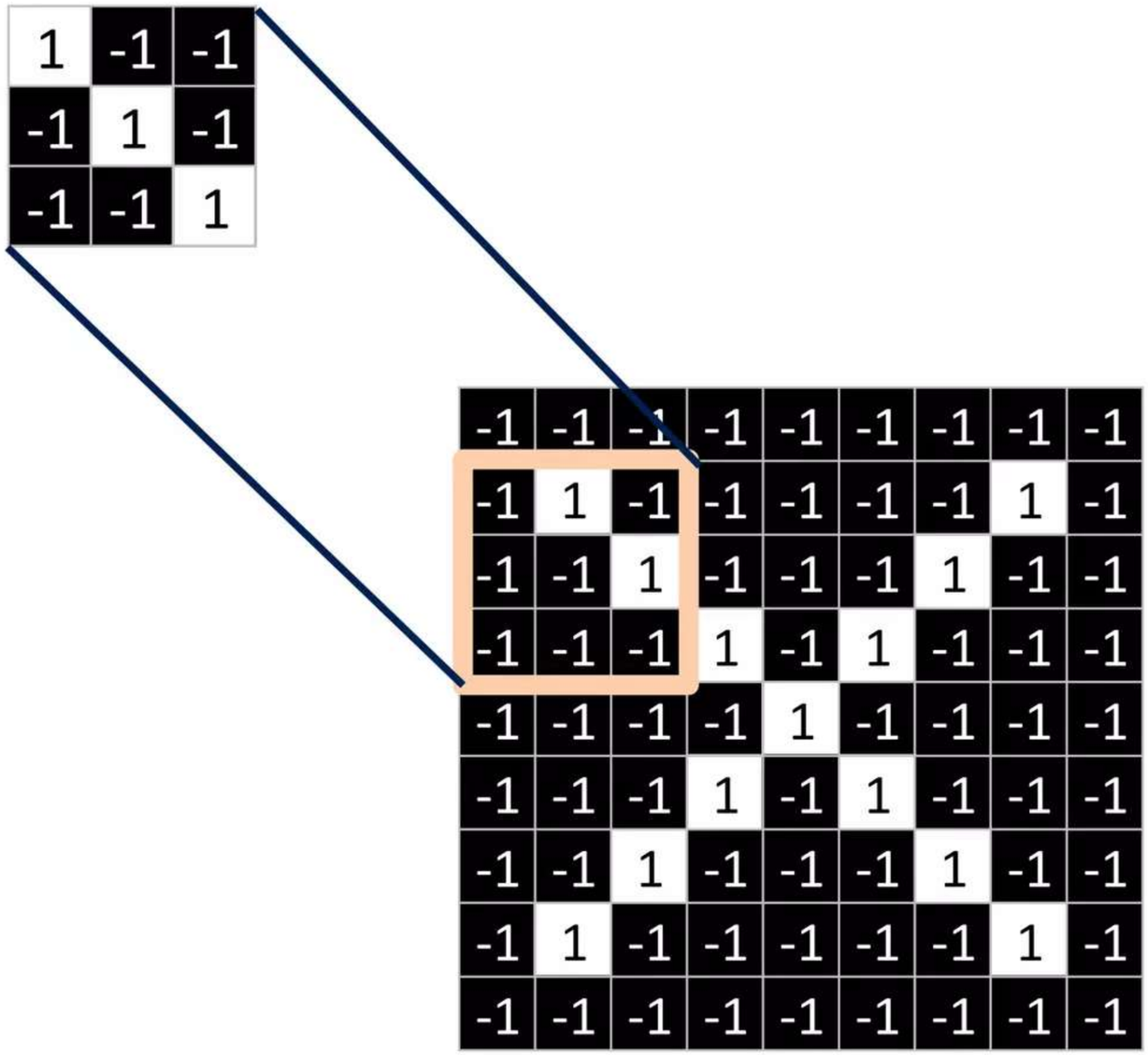
Convolutional Layer – Filters – Navigation Example



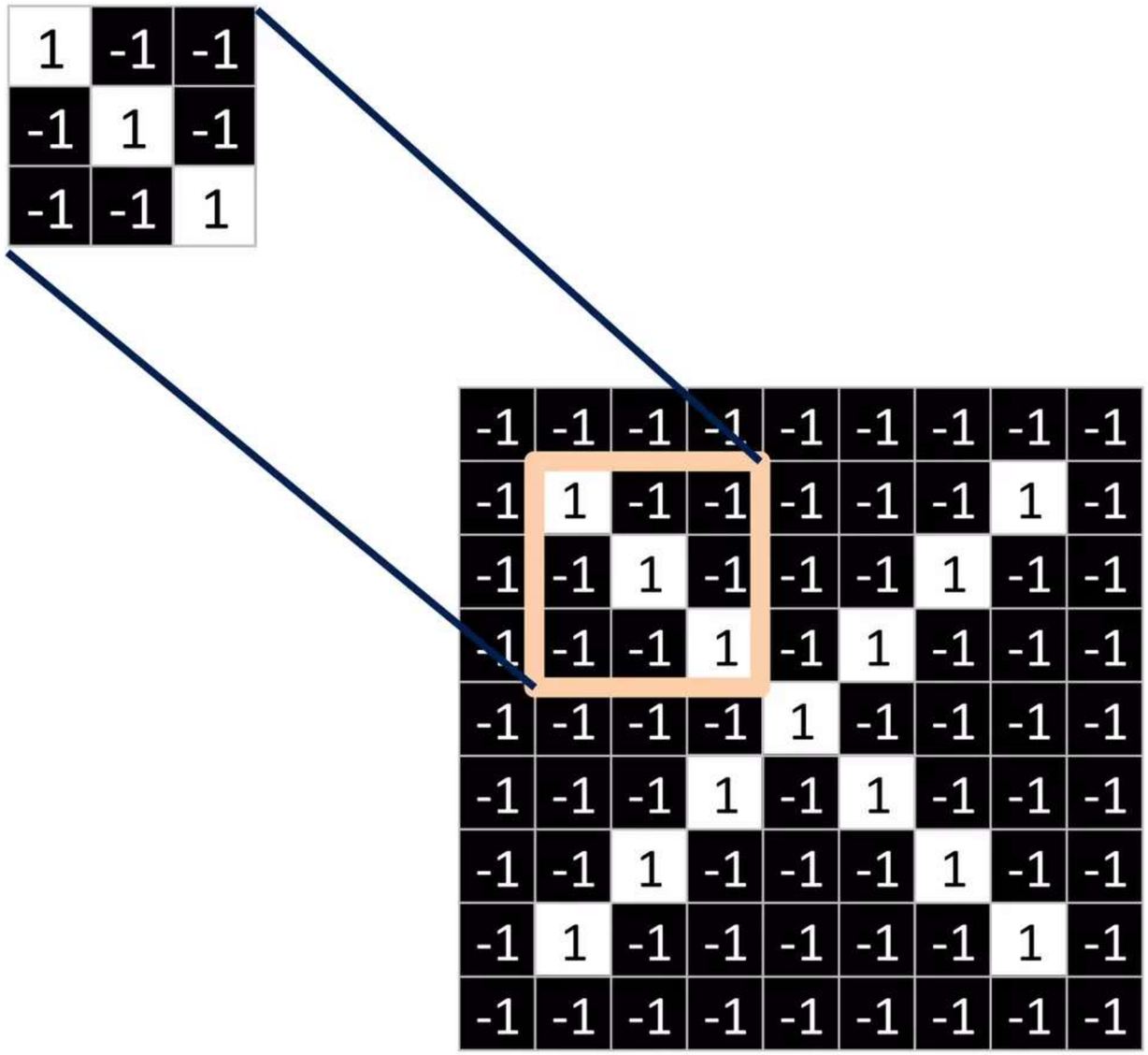
Convolutional Layer – Filters – Navigation Example



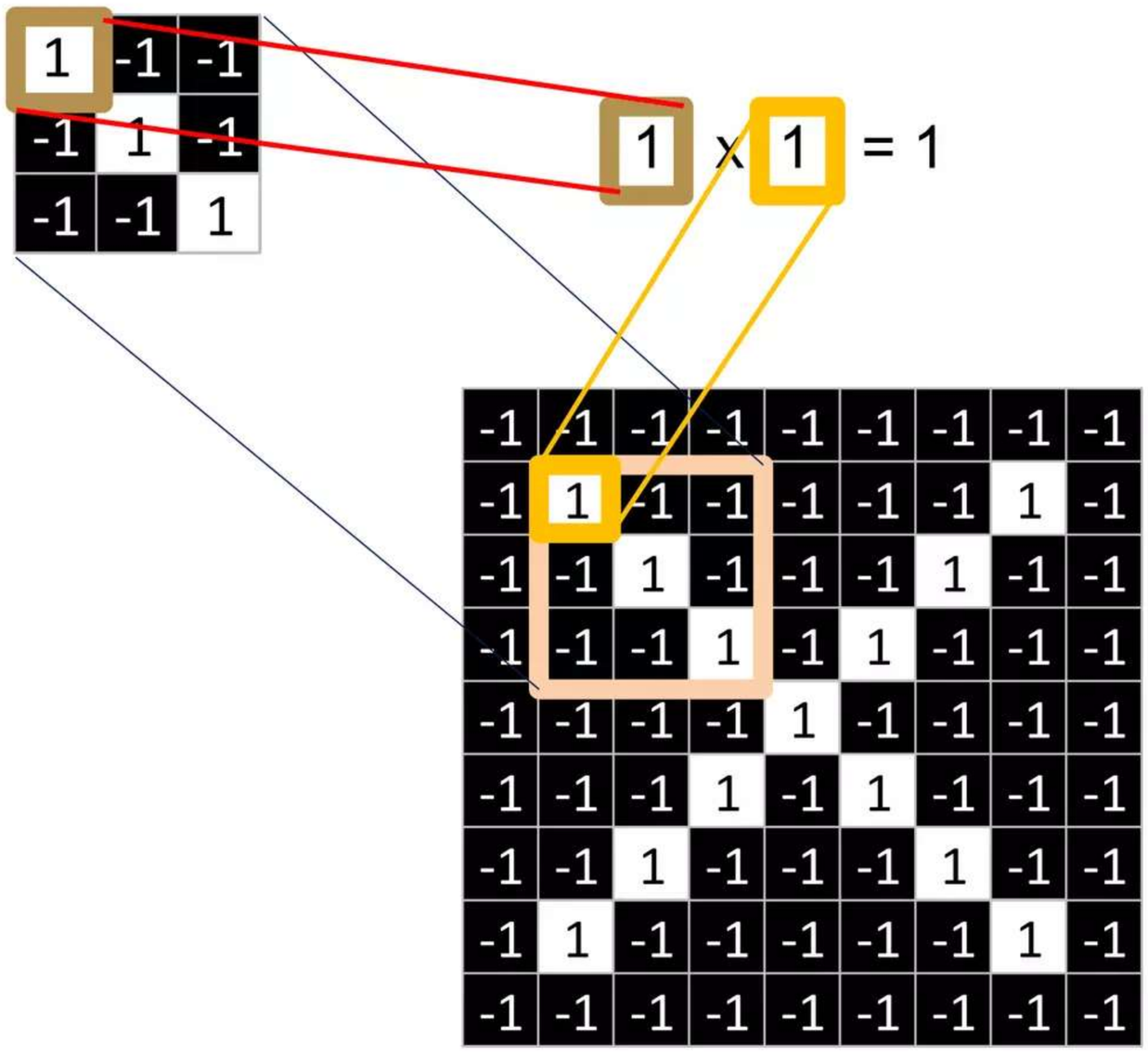
Convolutional Layer – Filters – Navigation Example



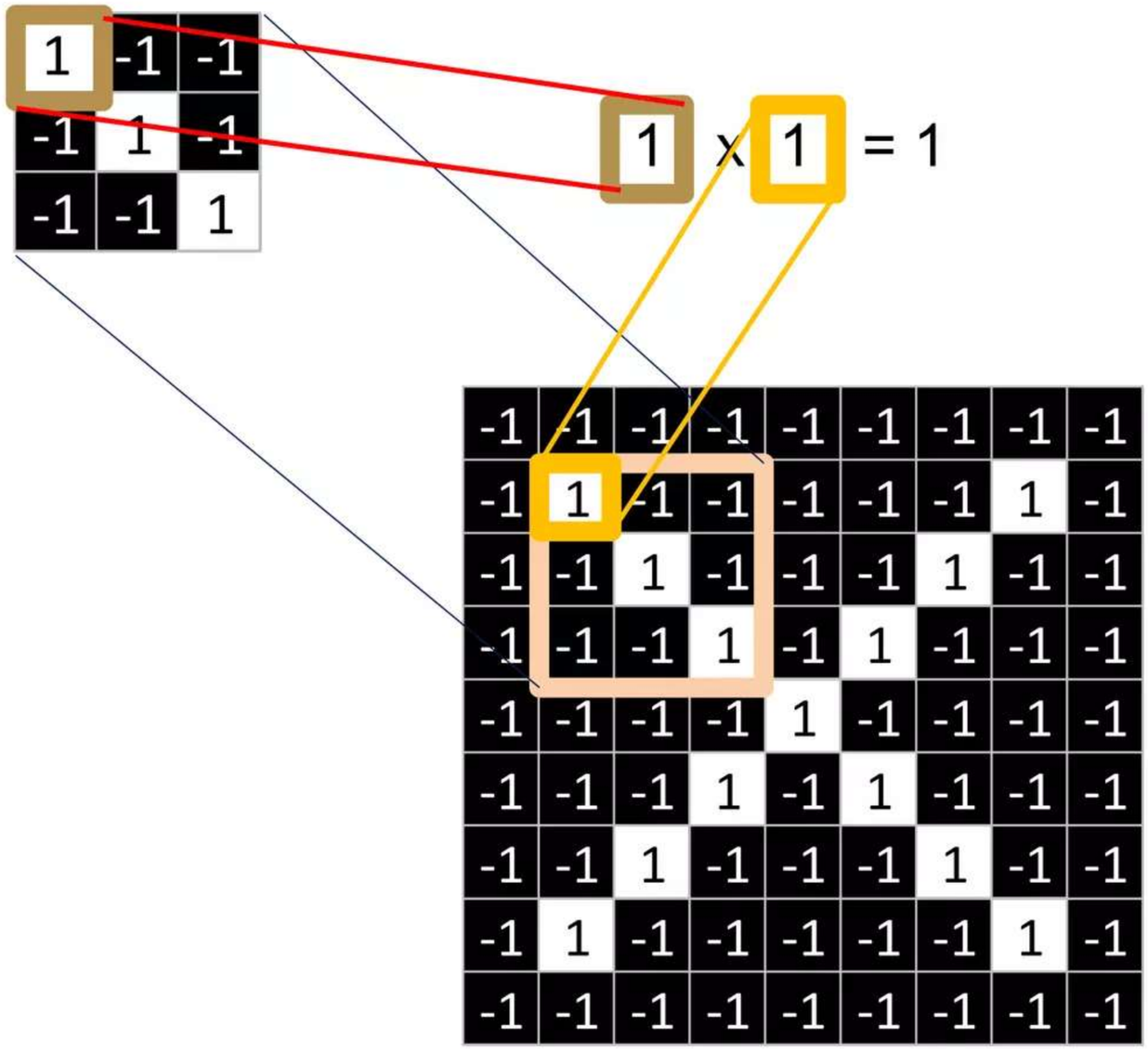
Convolutional Layer – Filters – Navigation Example

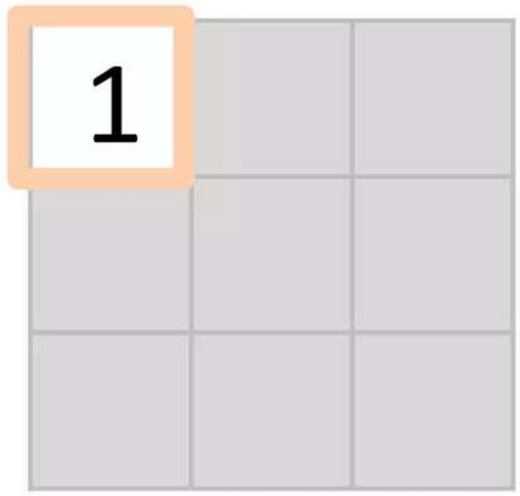


Convolutional Layer – Filters – Computation Example

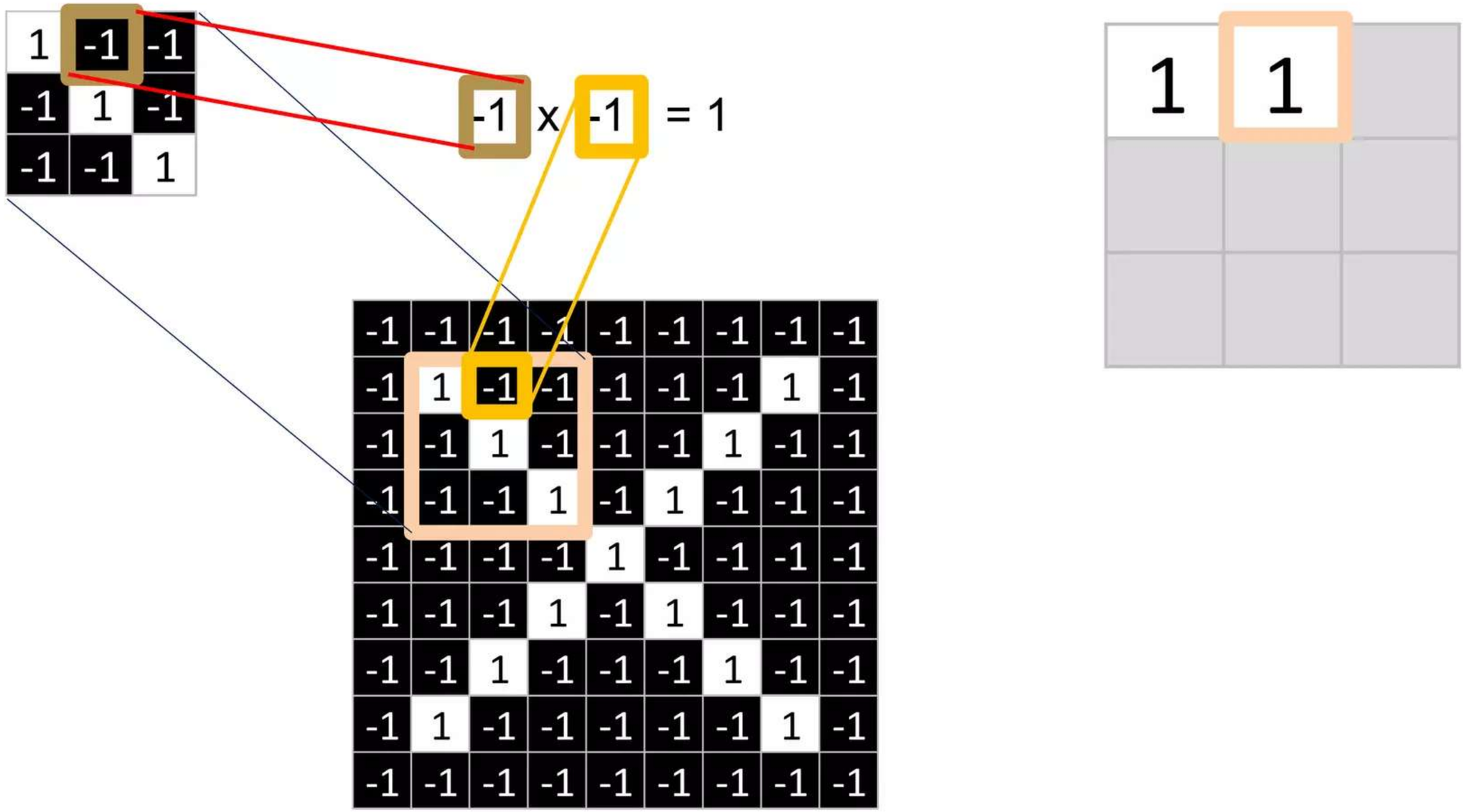


Convolutional Layer – Filters – Computation Example

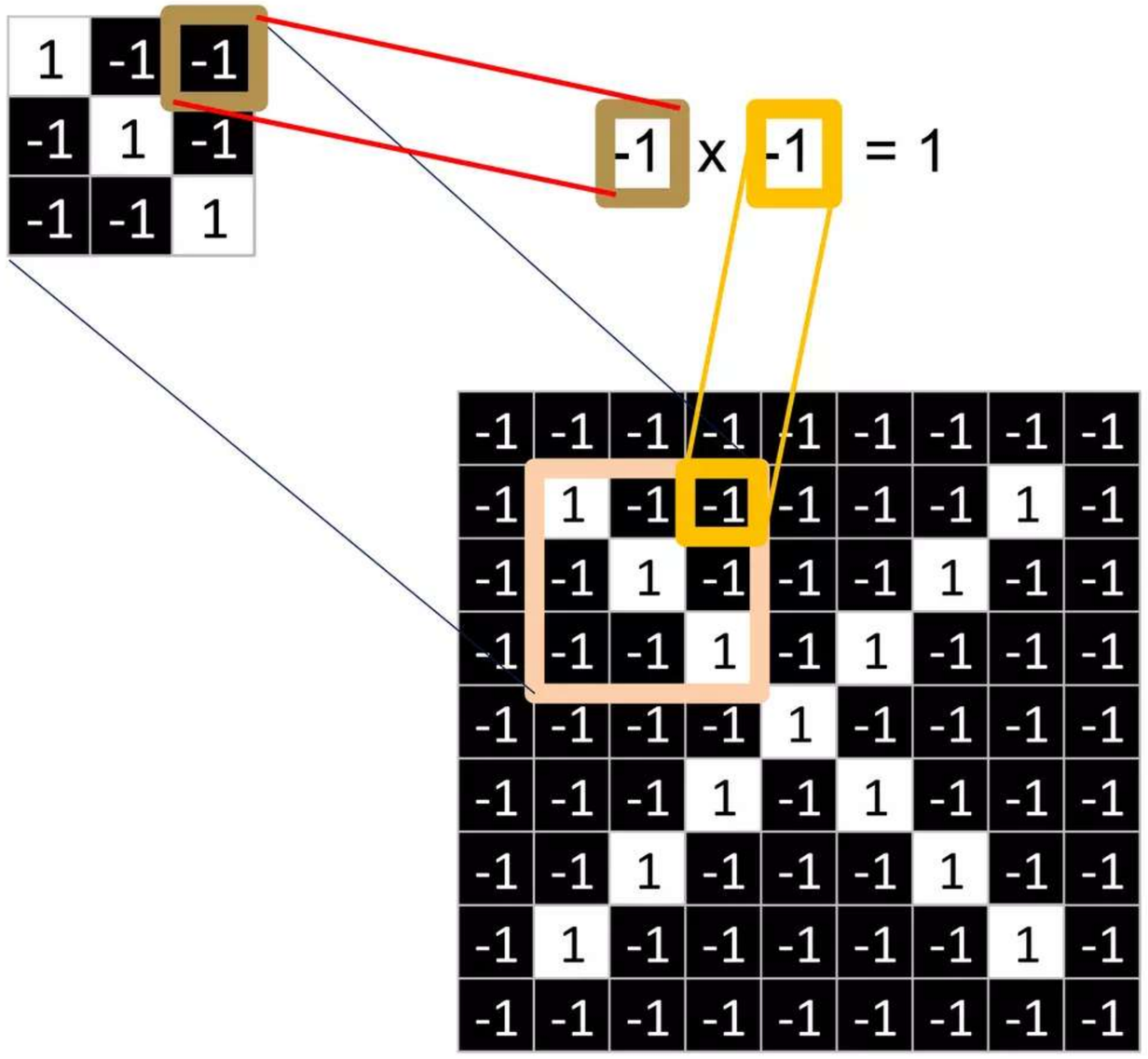


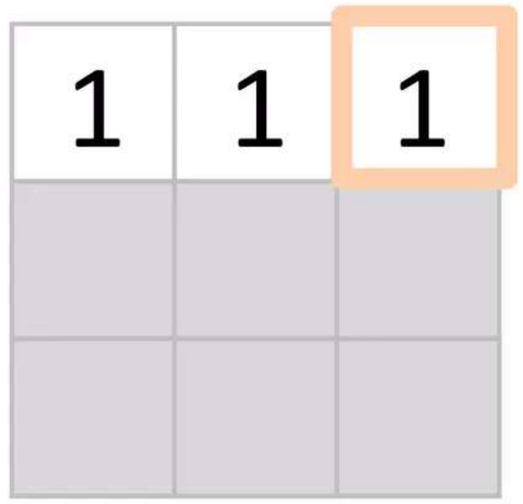


Convolutional Layer – Filters – Computation Example

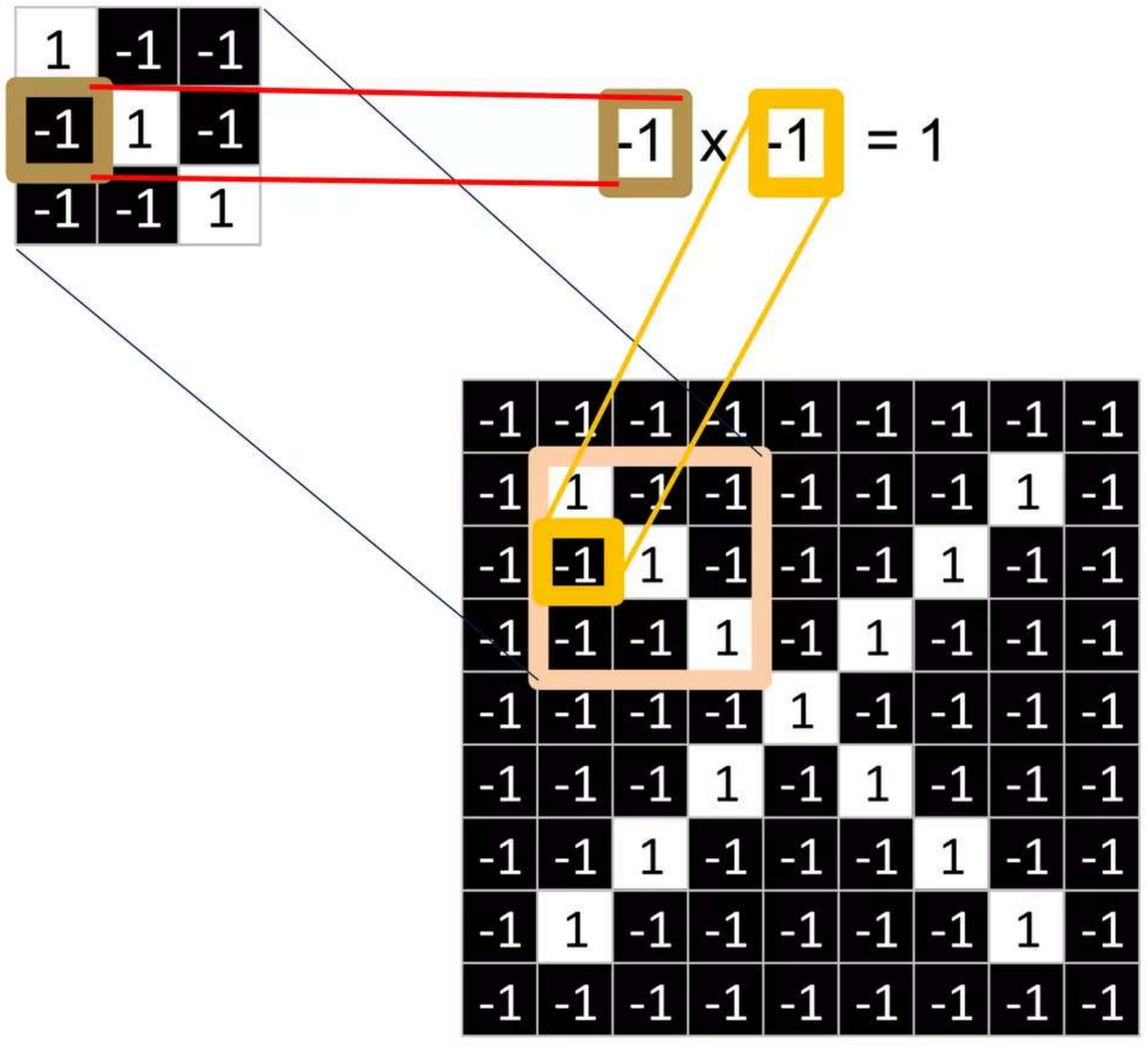


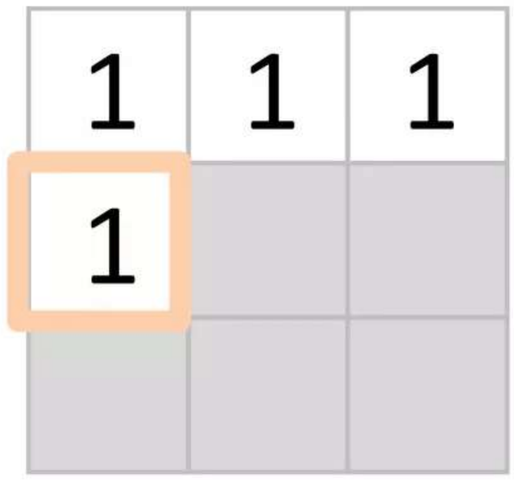
Convolutional Layer – Filters – Computation Example



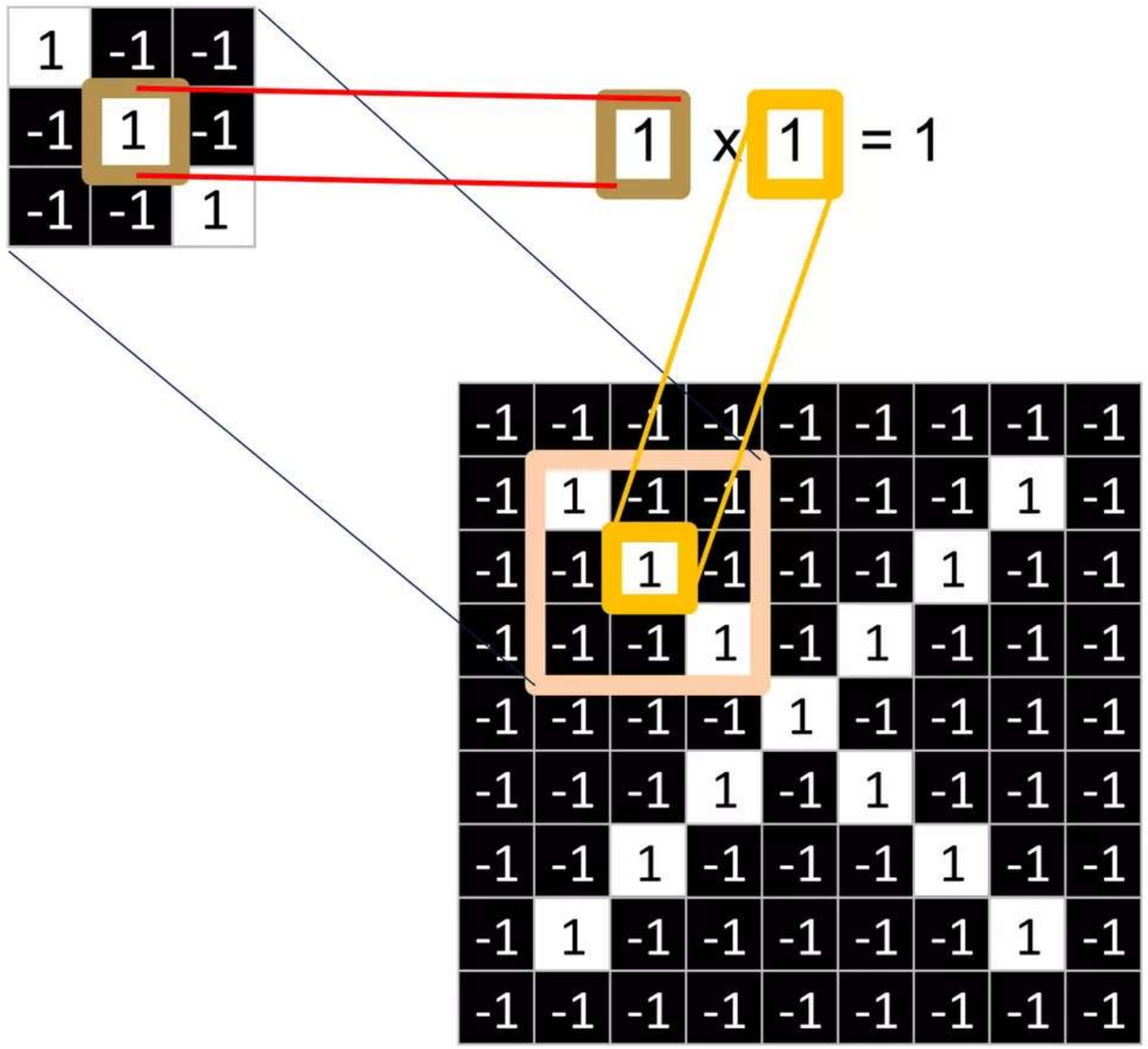


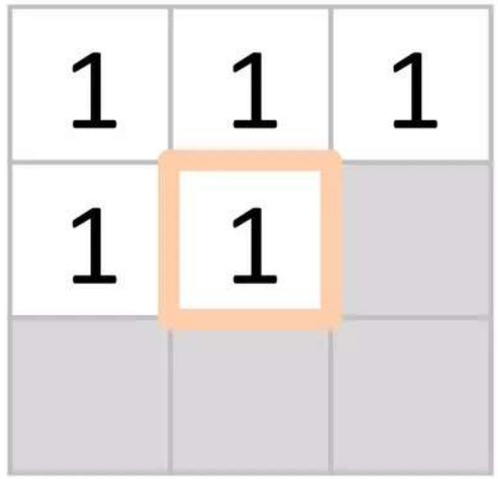
Convolutional Layer – Filters – Computation Example



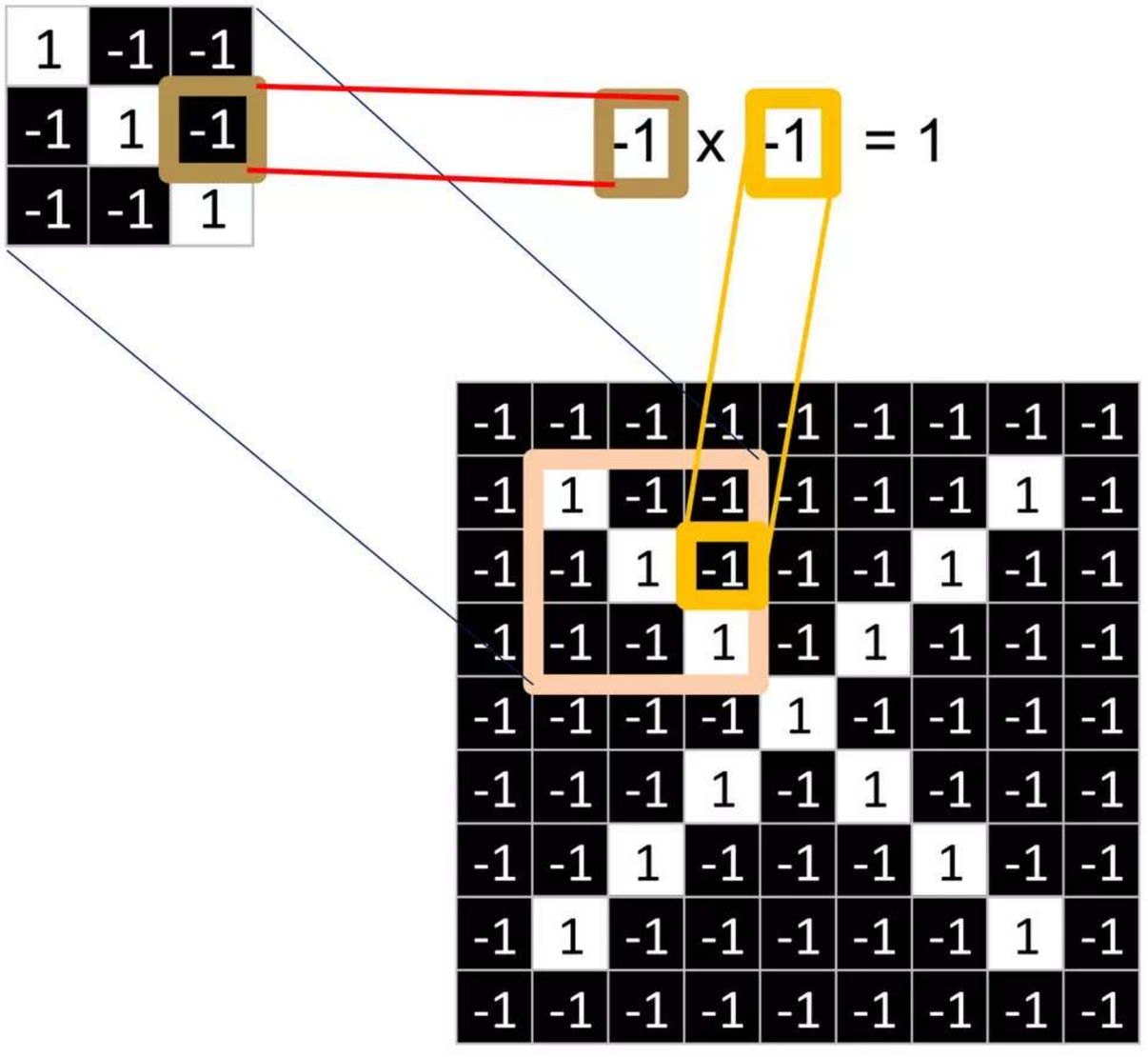


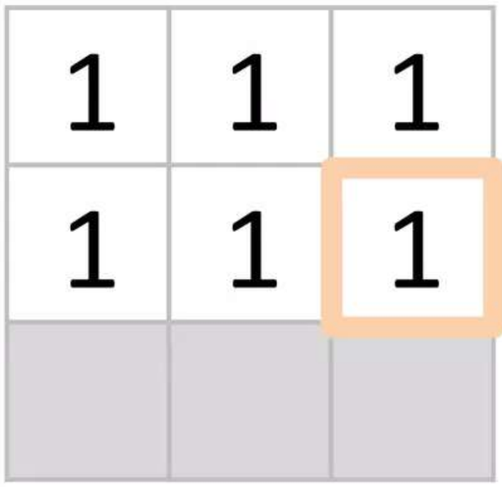
Convolutional Layer – Filters – Computation Example



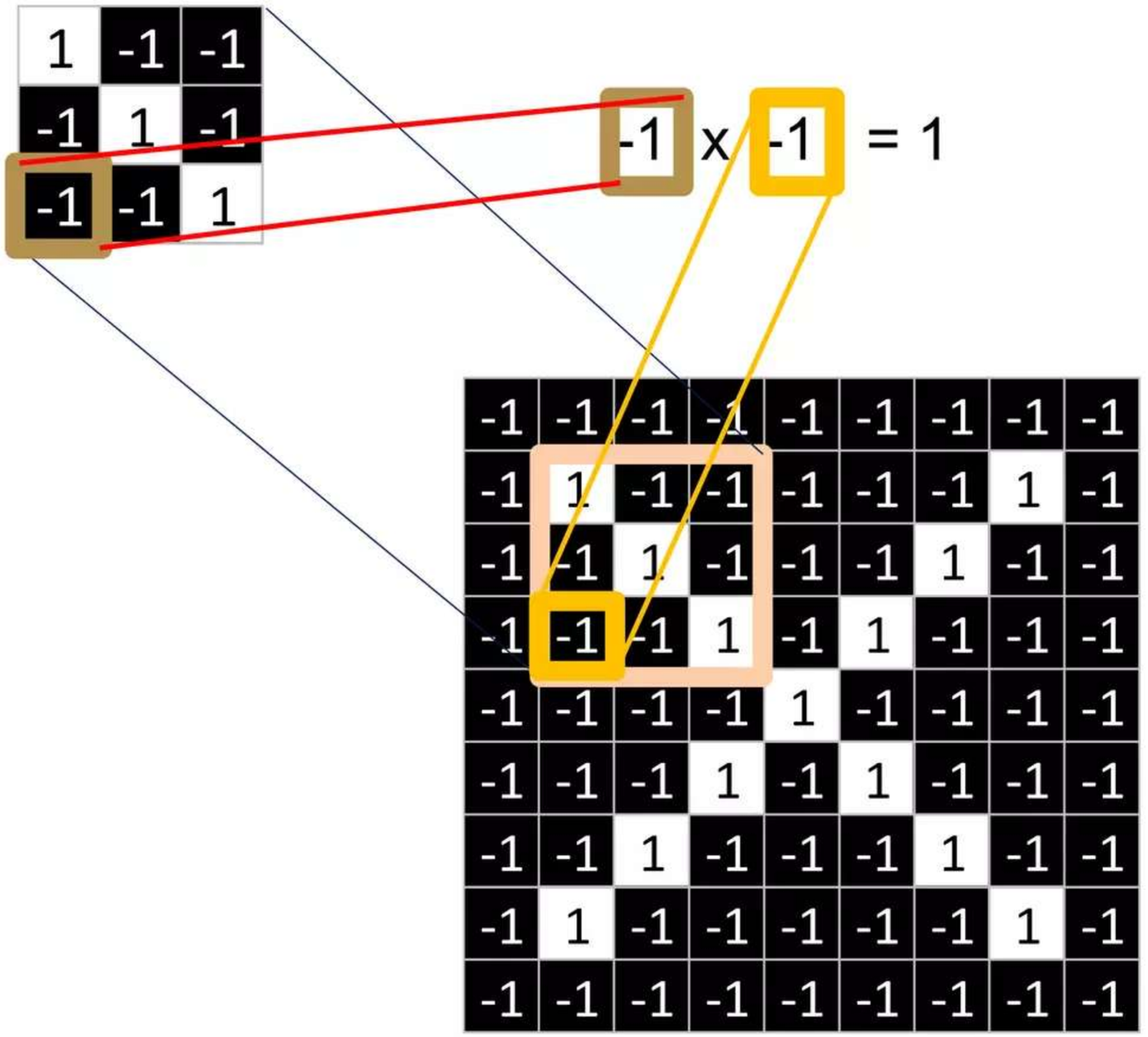


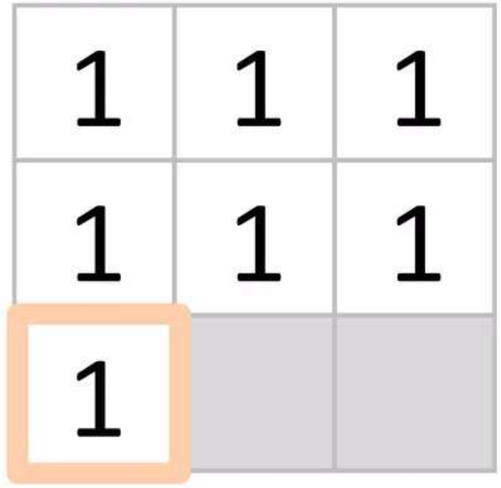
Convolutional Layer – Filters – Computation Example



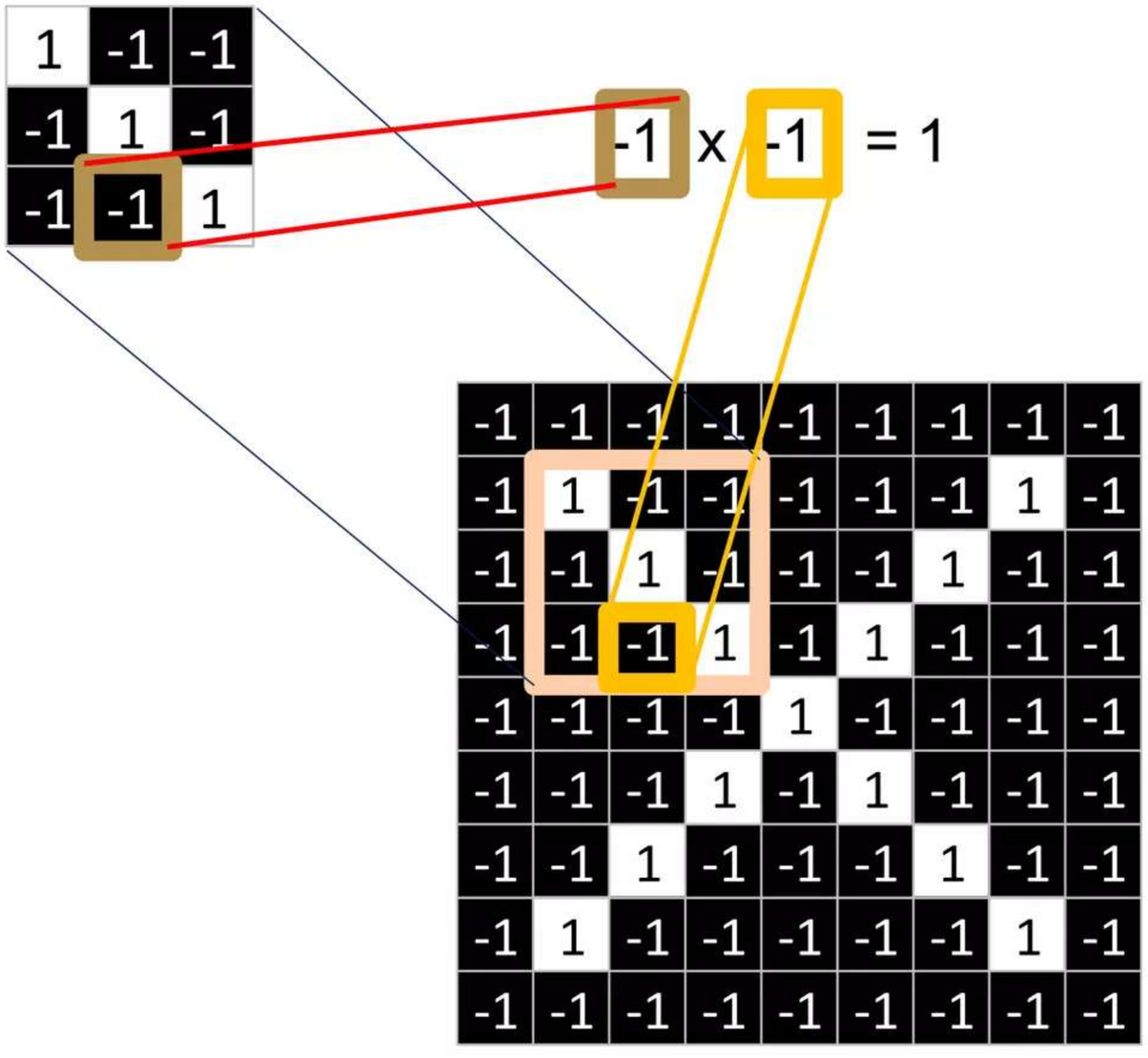


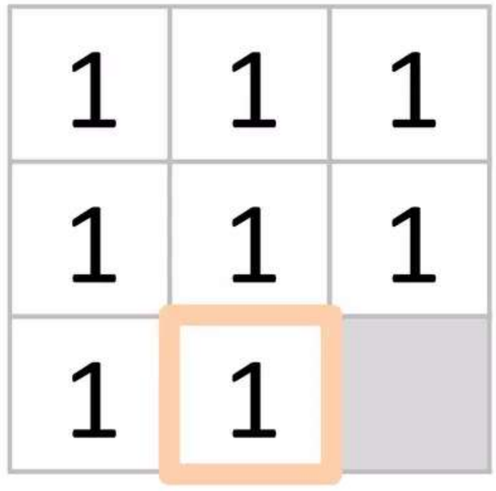
Convolutional Layer – Filters – Computation Example



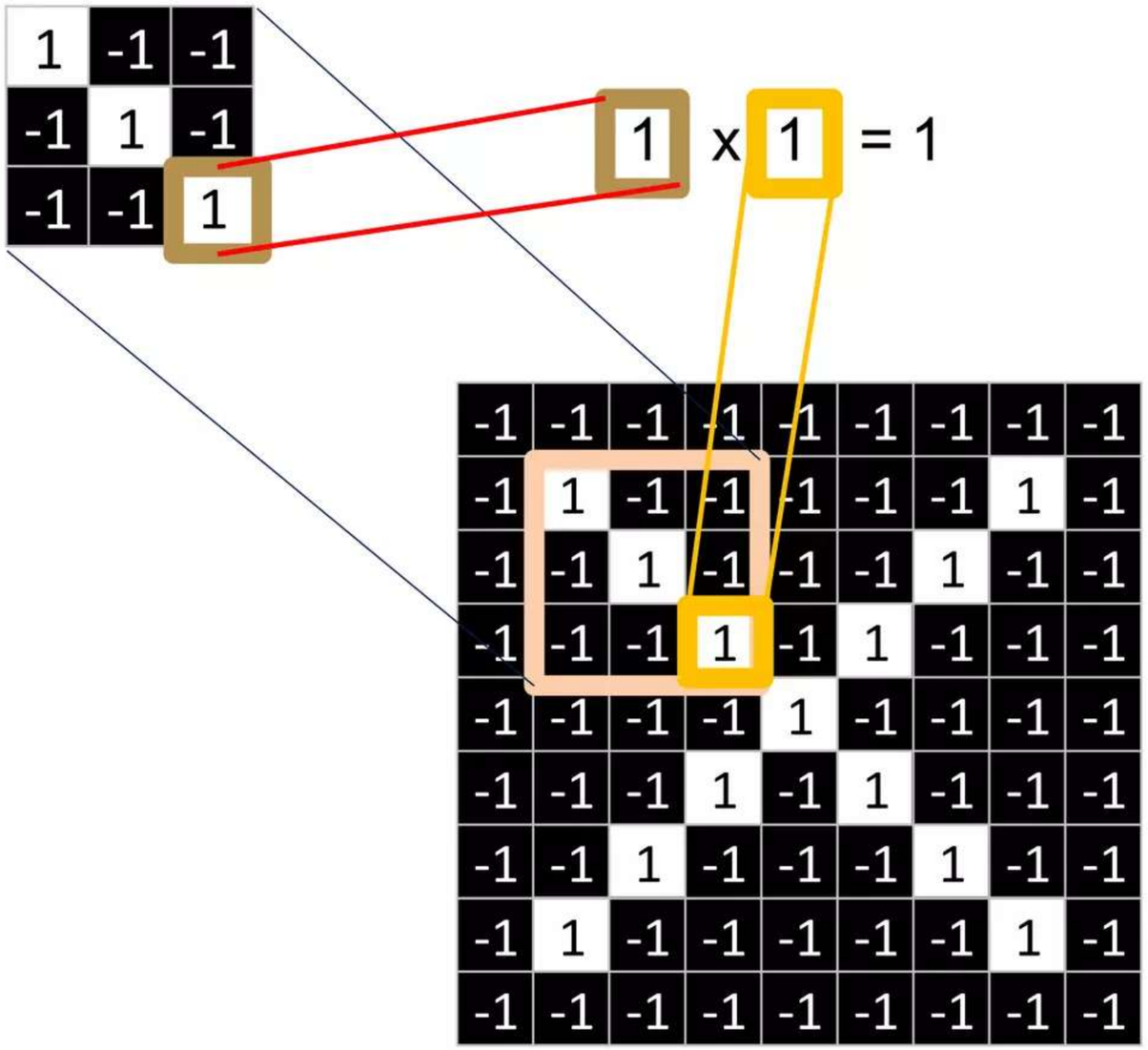


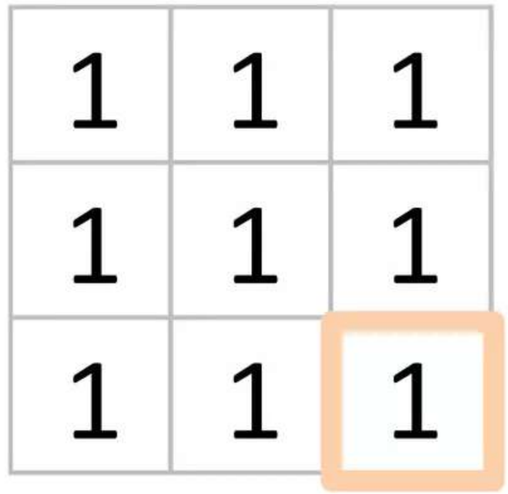
Convolutional Layer – Filters – Computation Example



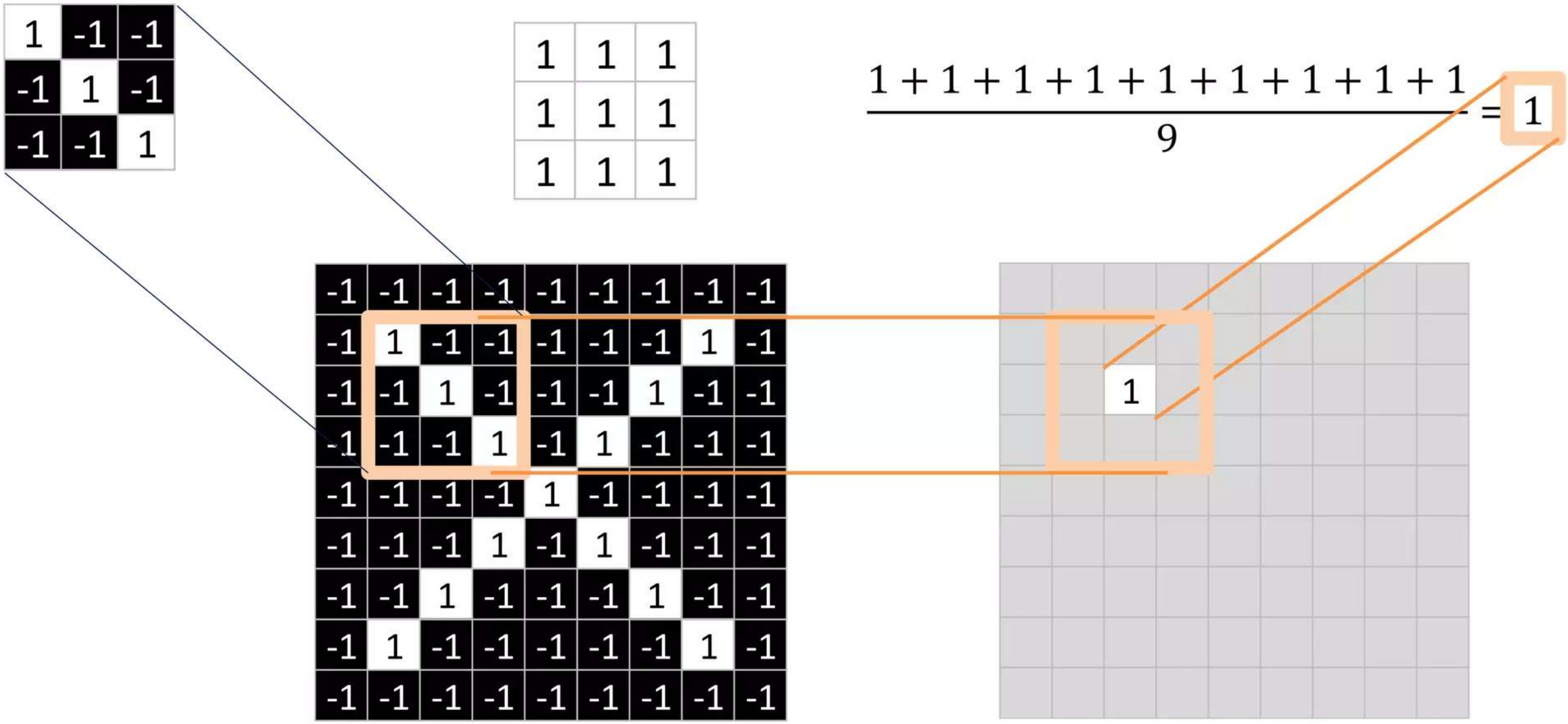


Convolutional Layer – Filters – Computation Example

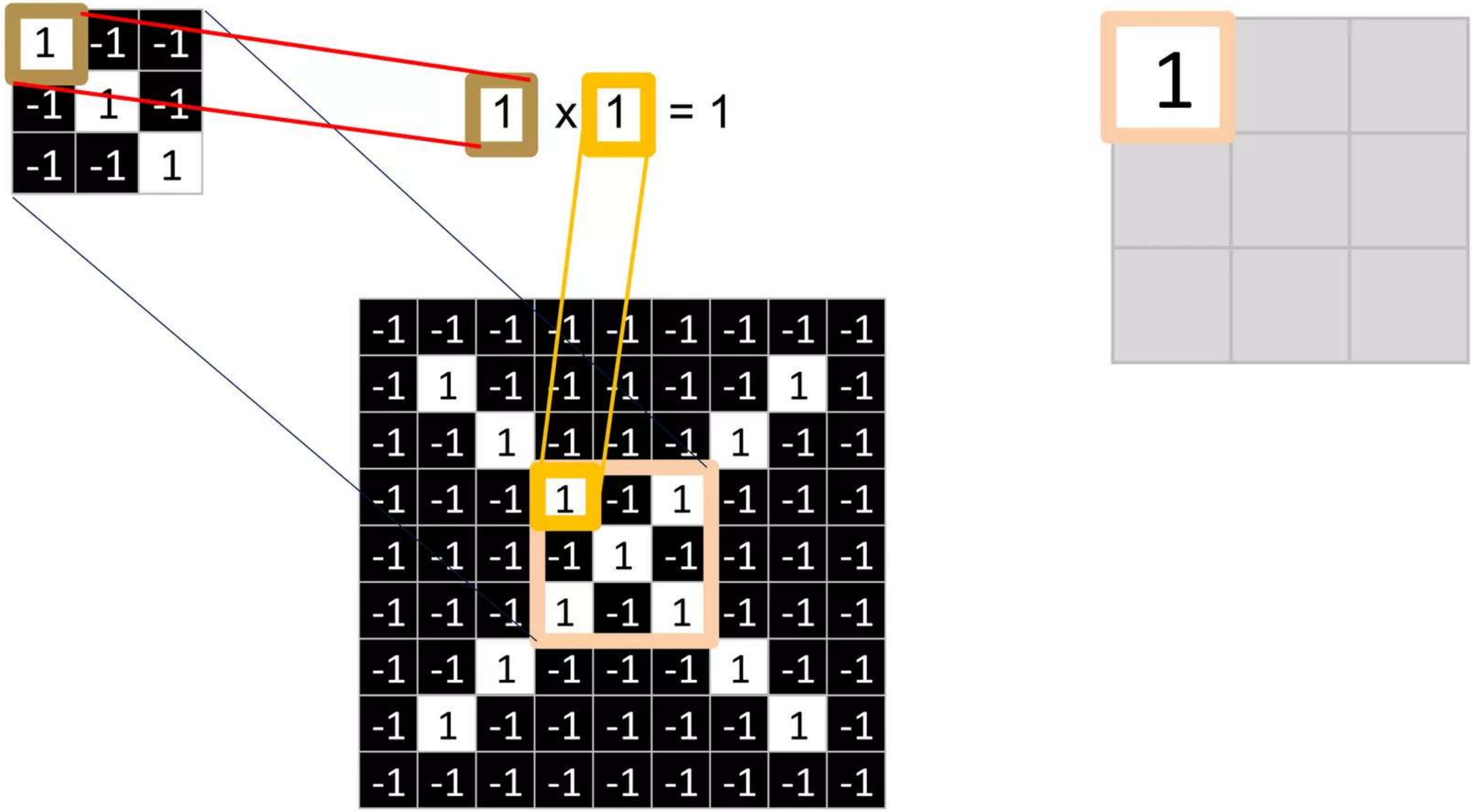




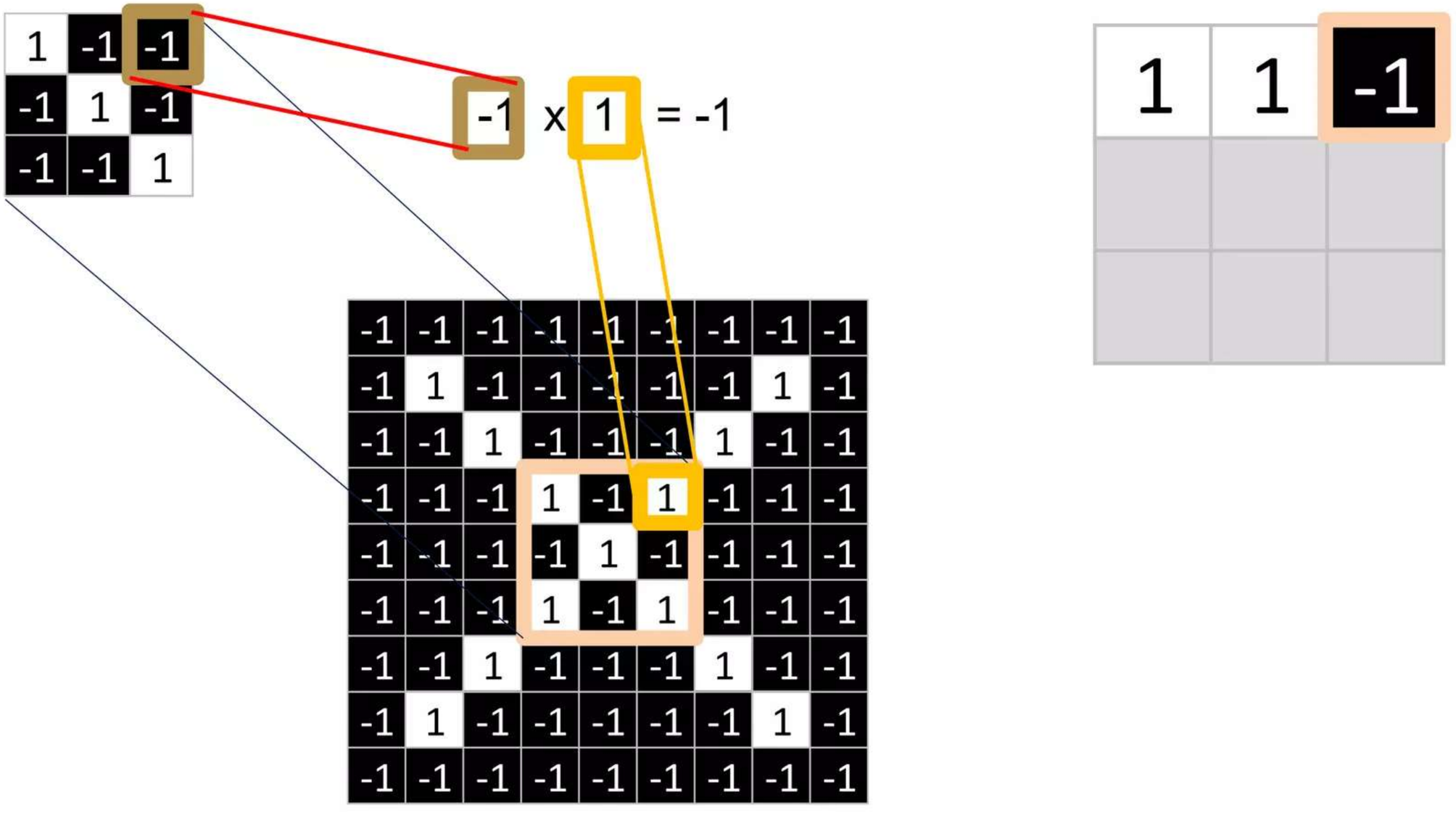
Convolutional Layer – Filters – Computation Example



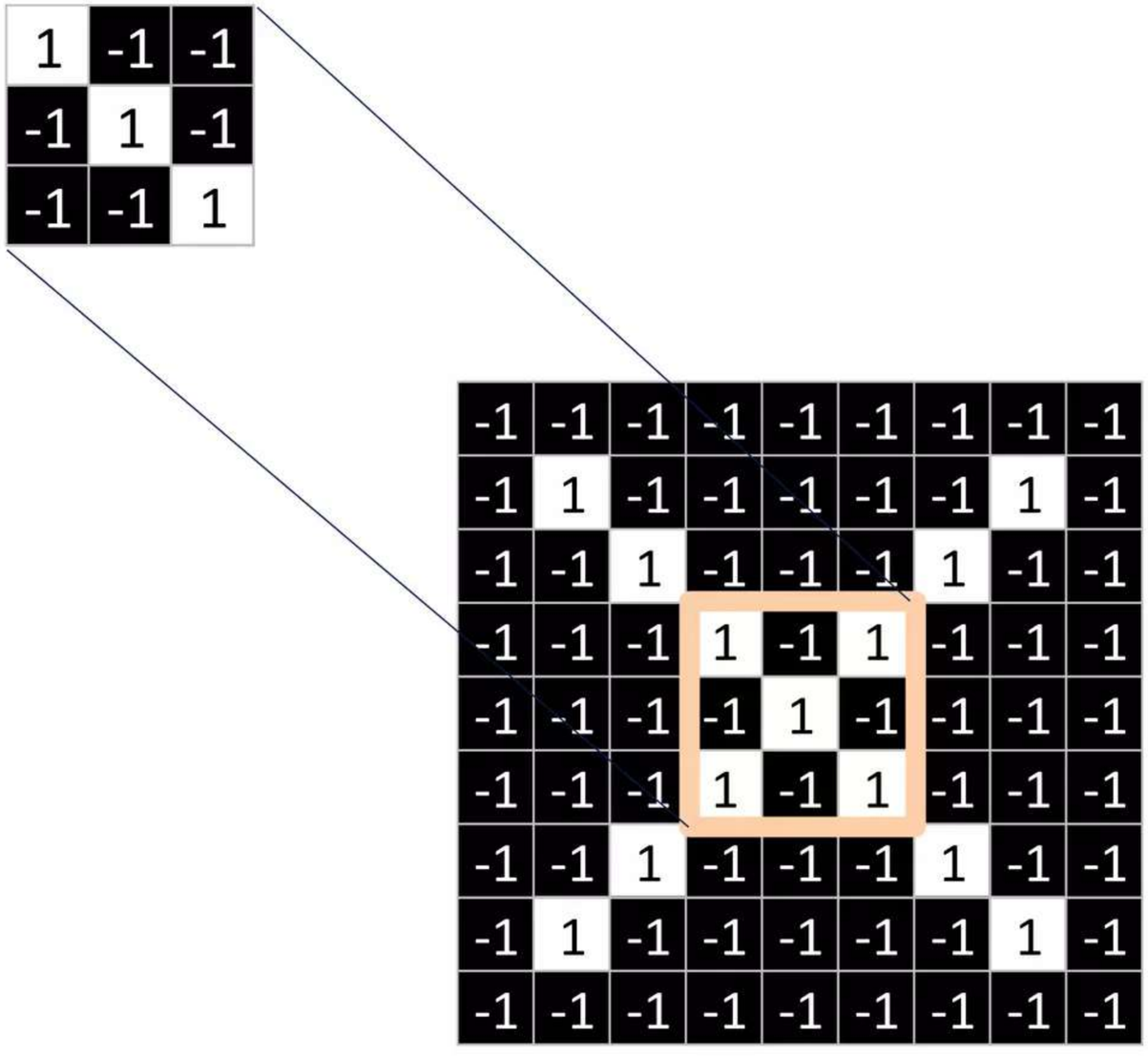
Convolutional Layer – Filters – Computation Example

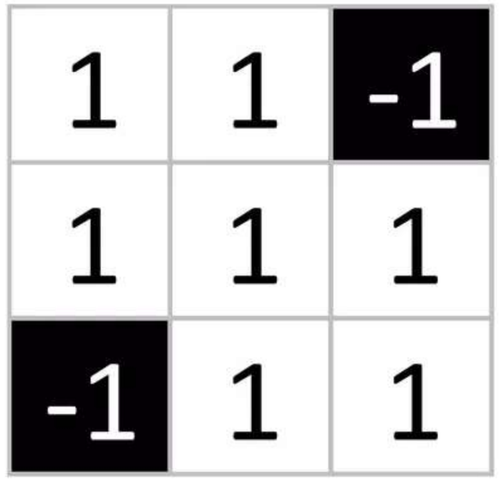


Convolutional Layer – Filters – Computation Example

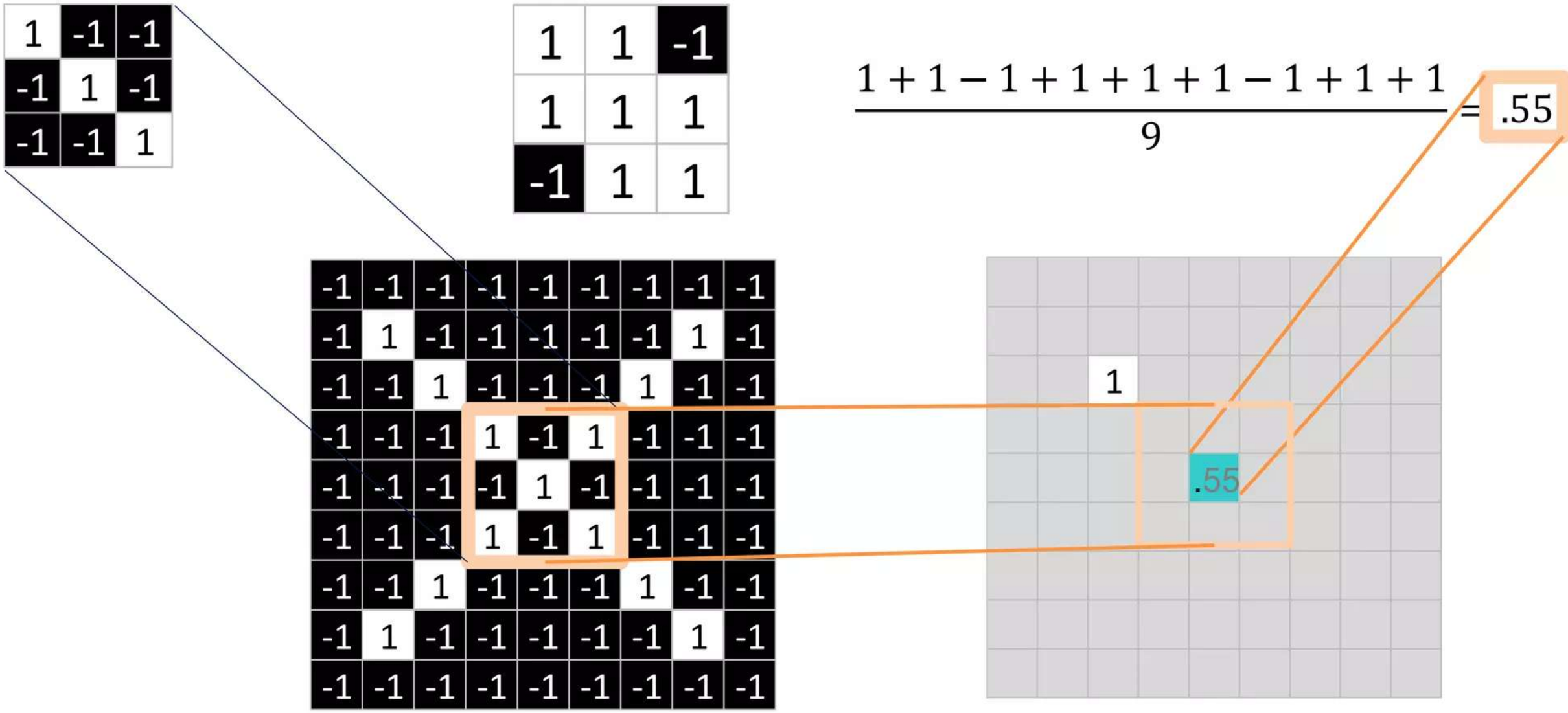


Convolutional Layer – Filters – Computation Example





Convolutional Layer – Filters – Computation Example



Convolutional Layer – Filters – Computation Example

Input Size (W): 9
Filter Size (F): 3 X 3
Stride (S): 1
Filters: 1
Padding (P): 0

9 X 9

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

⊗

1	-1	-1
-1	1	-1
-1	-1	1

=

7 X 7

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

Feature Map Size = 1+ (W – F + 2P)/S
= 1+ (9 – 3 + 2 X 0)/1 = 7

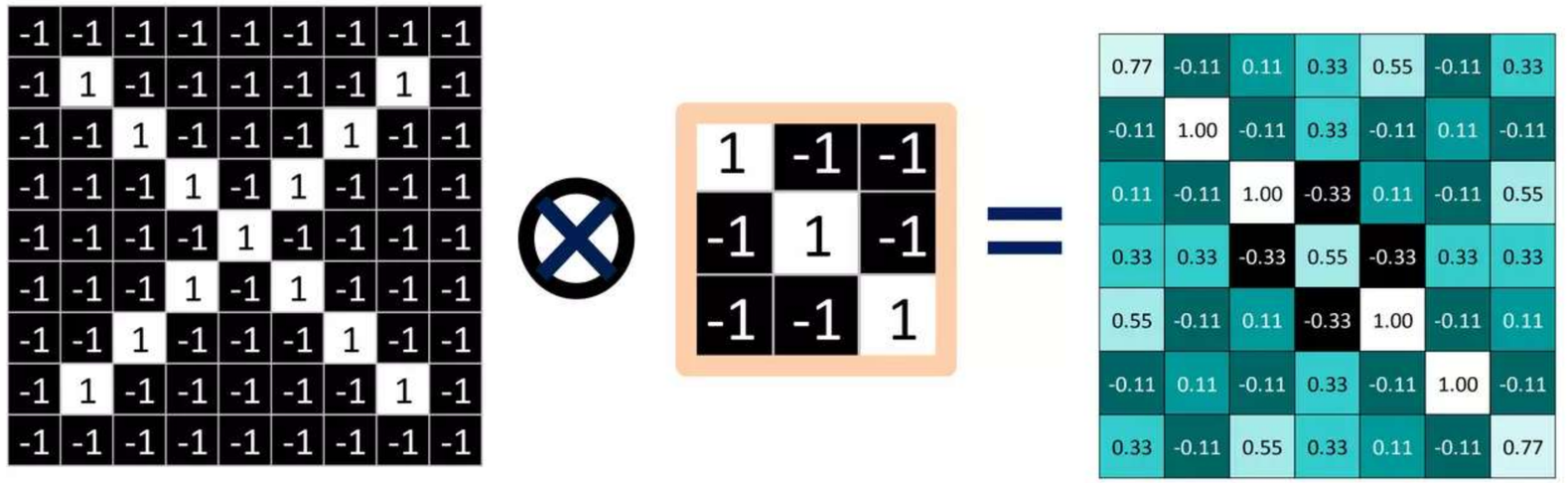
Convolutional Layer – Filters – Output Feature Map

- Output Feature Map of One complete convolution:
  - Filters: 3
  - Filter Size: 3 X 3
  - Stride: 1

- Conclusion:
  - Input Image:

    9 X 9
  - Output of Convolution:

    7 X 7 X 3



Convolutional Layer – Output

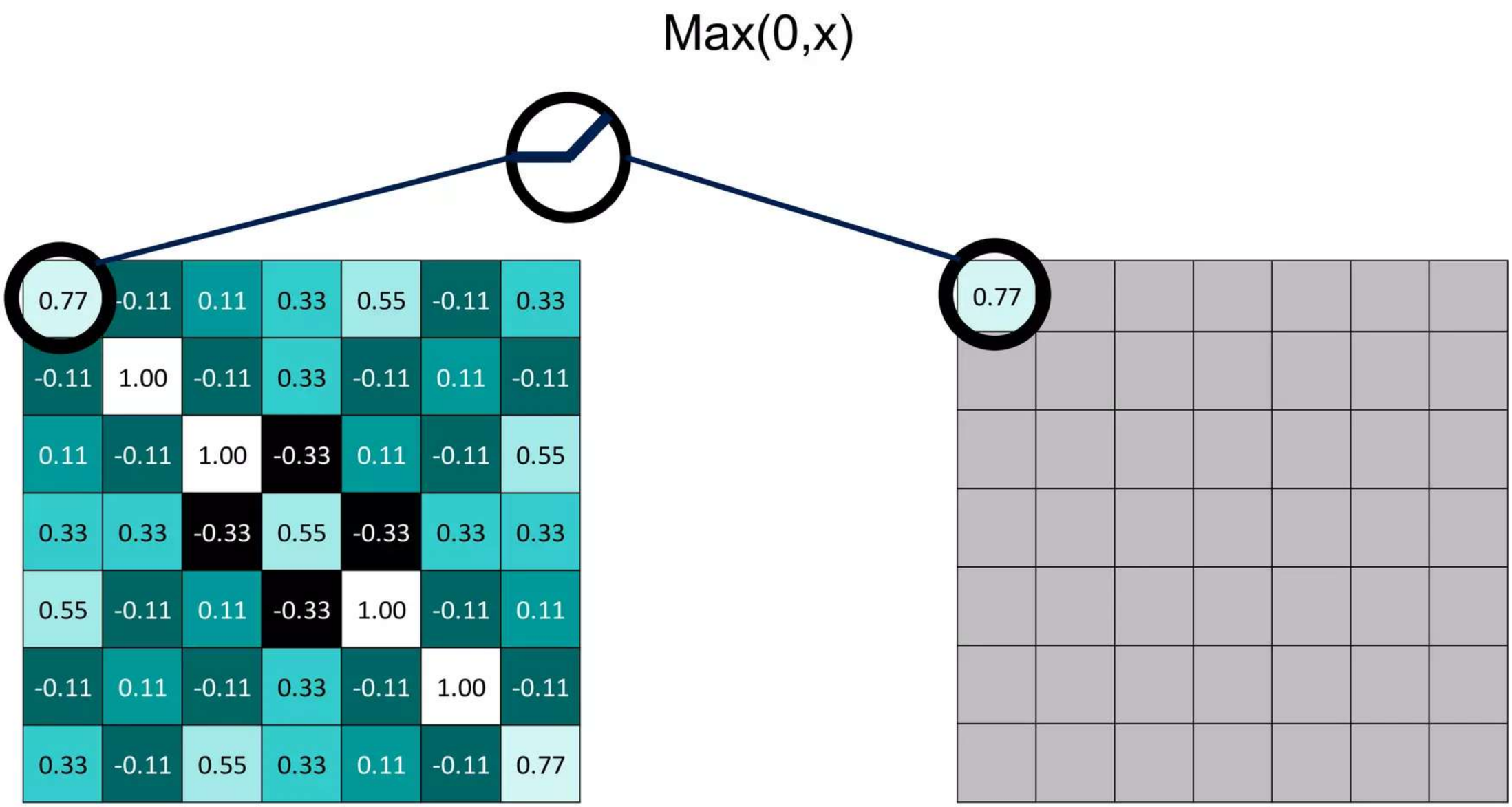
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

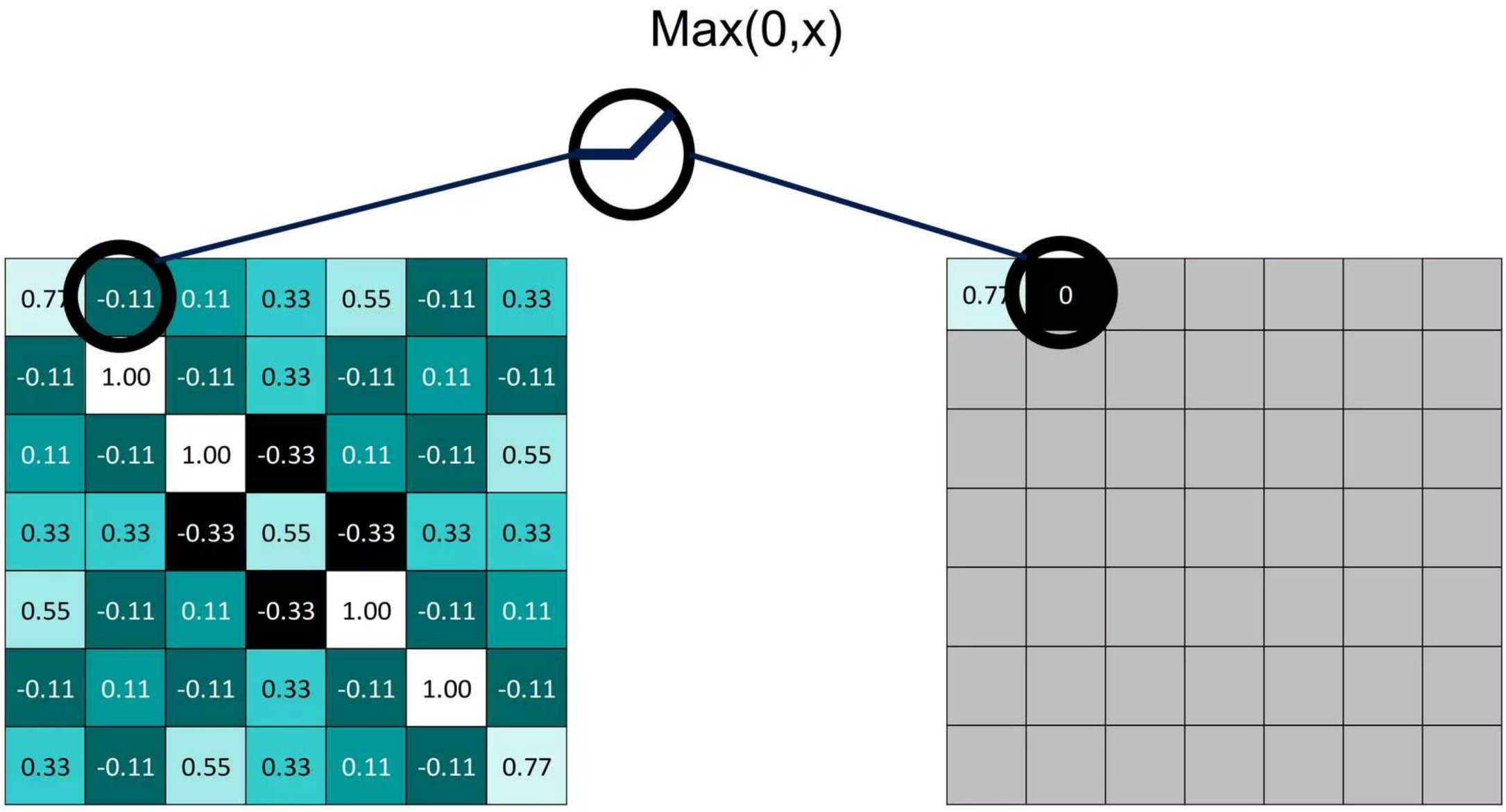
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.11	0.33	-0.77	1.00	-0.77	0.33	-0.11
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33

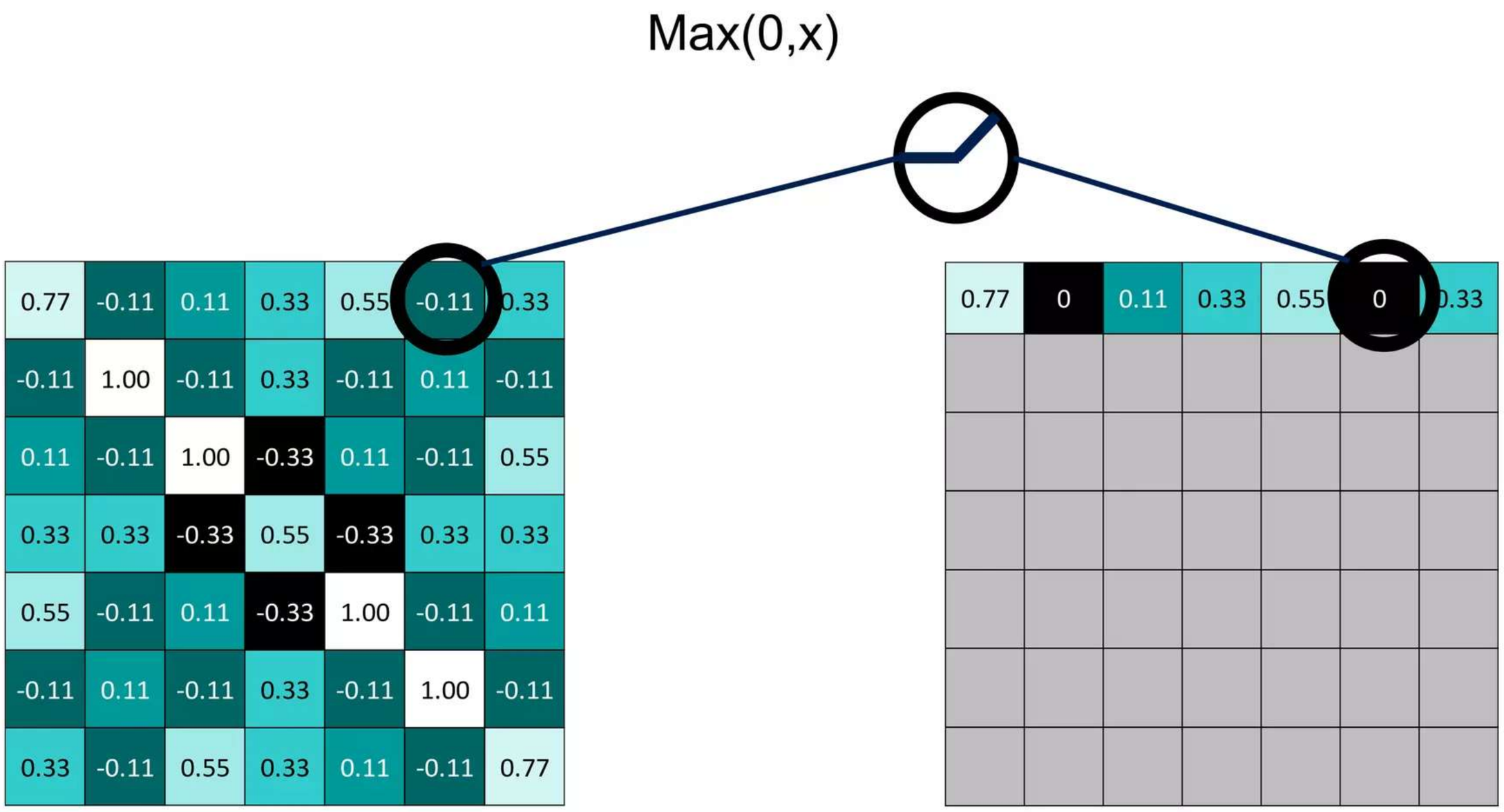
Rectified Linear Units (ReLUs)



Rectified Linear Units (ReLUs)

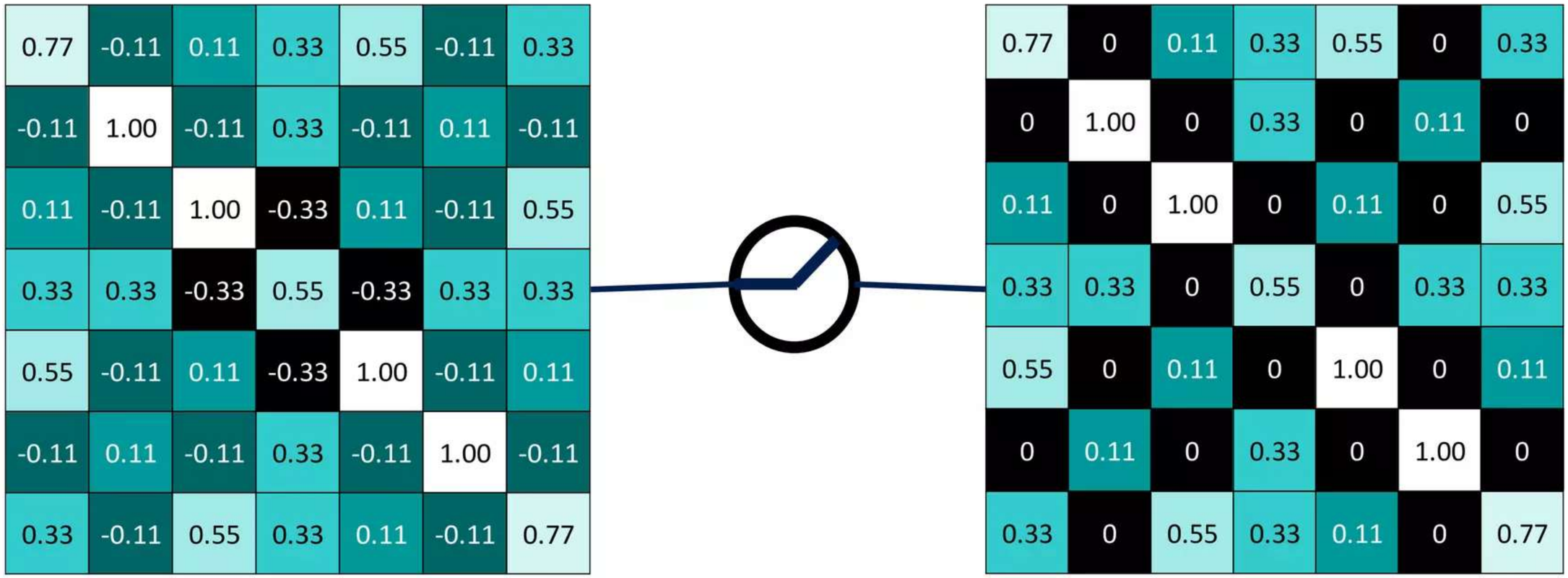


Rectified Linear Units (ReLUs)



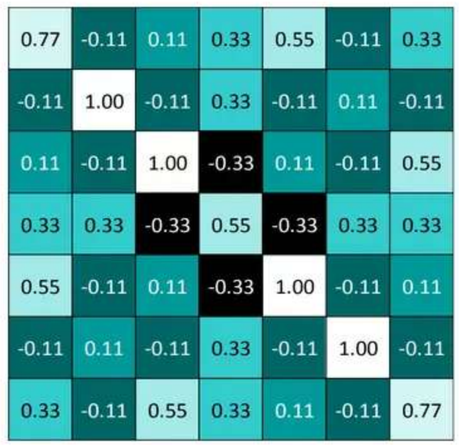
Rectified Linear Units (ReLUs)

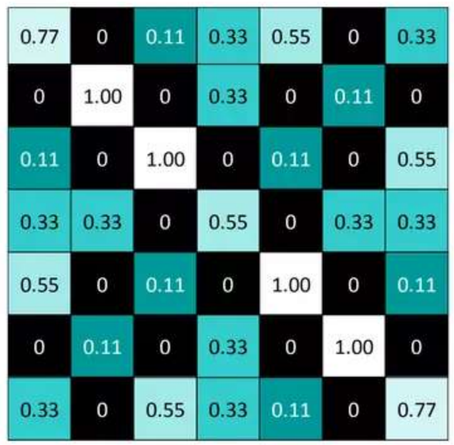
Max(0,x)

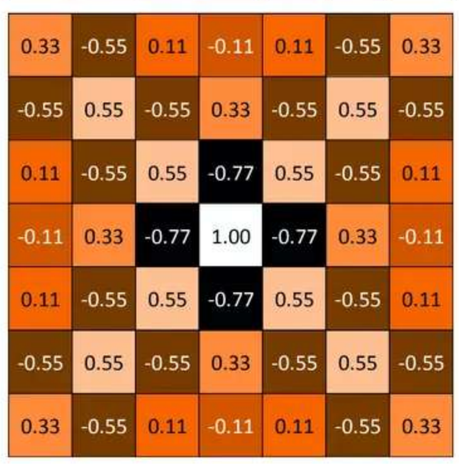


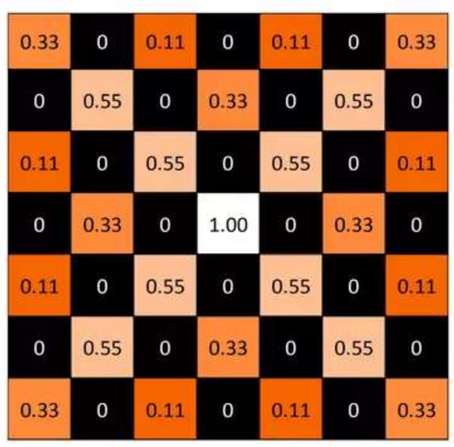
ReLU layer

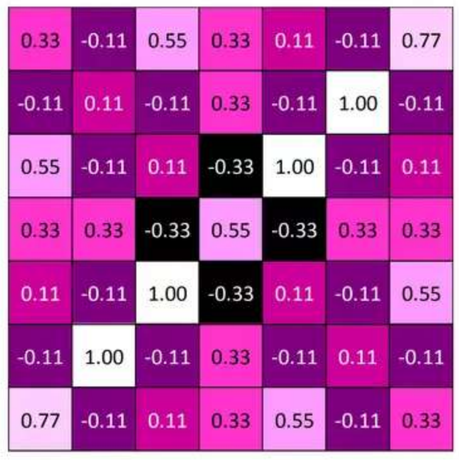
A stack of images becomes a stack of images with no negative values.

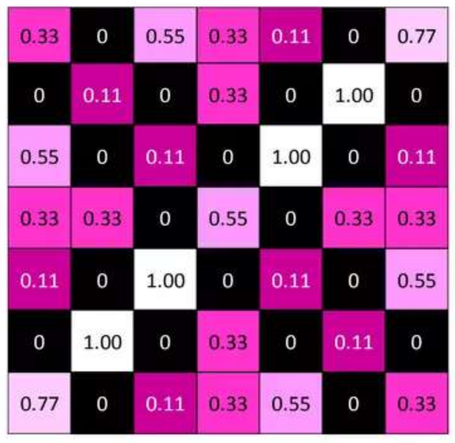










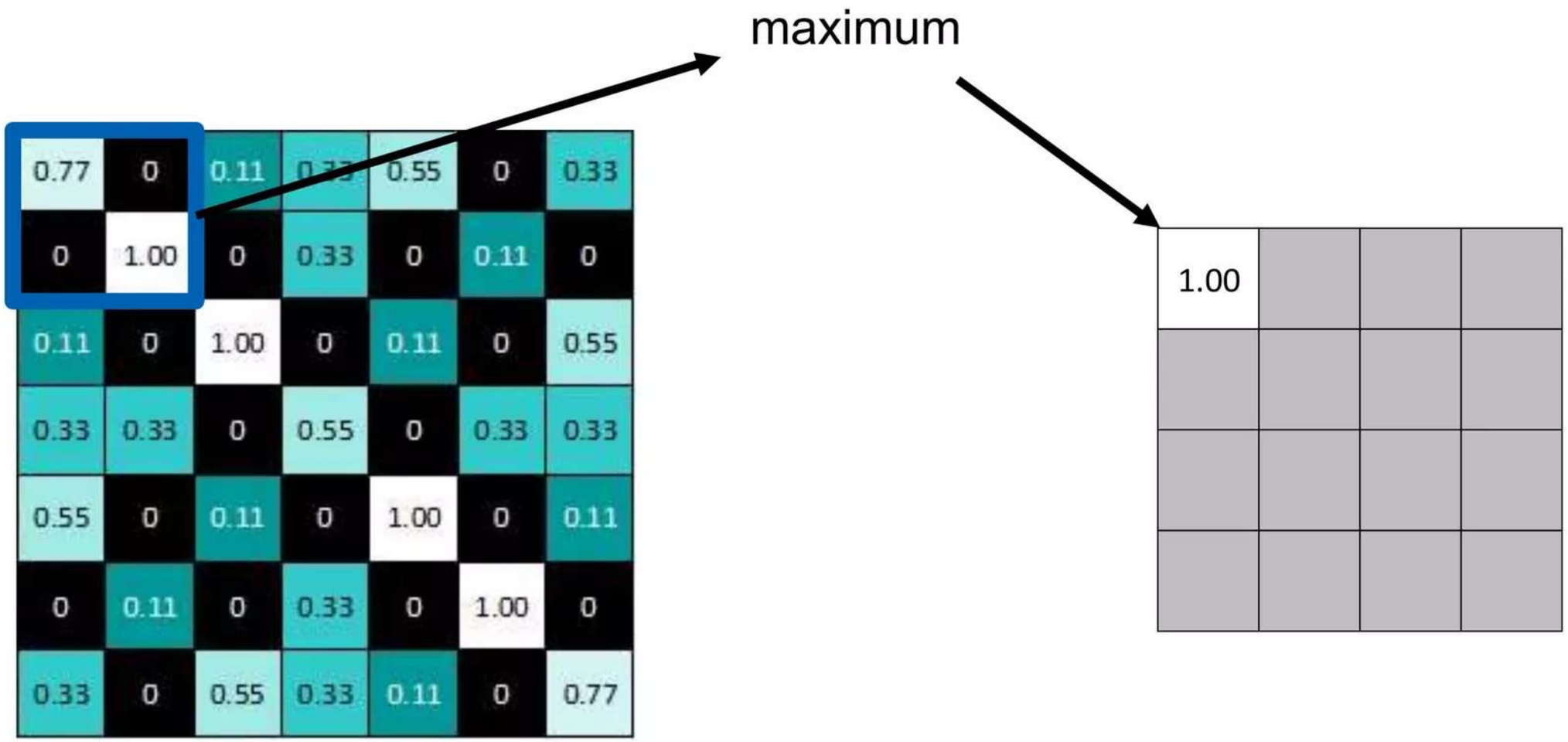


Pooling Layer

- The pooling layers down-sample the previous layers feature map.
- Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network
- The pooling layer often uses the Max operation to perform the downsampling process

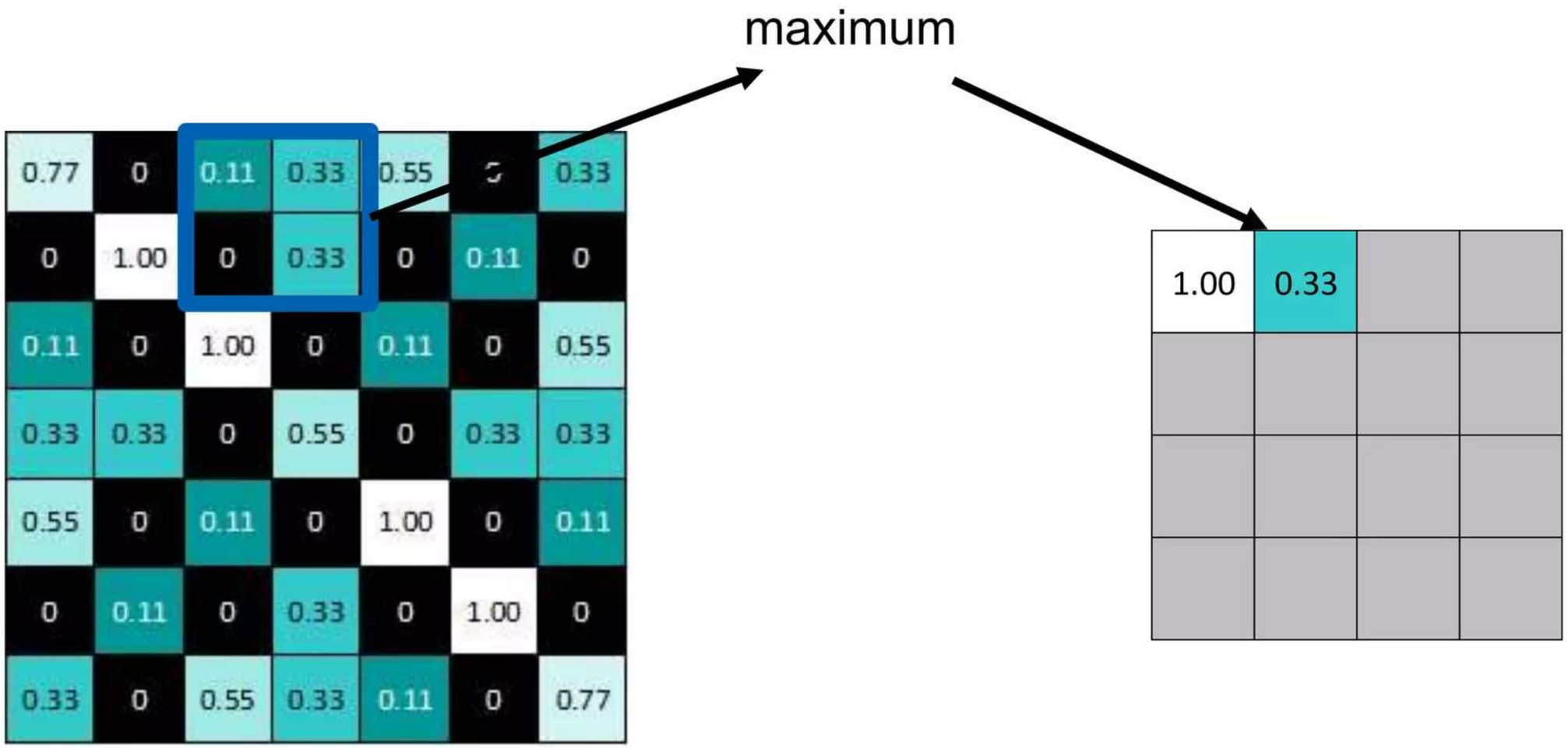
Pooling

- Pooling Filter Size = 2 X 2, Stride = 2



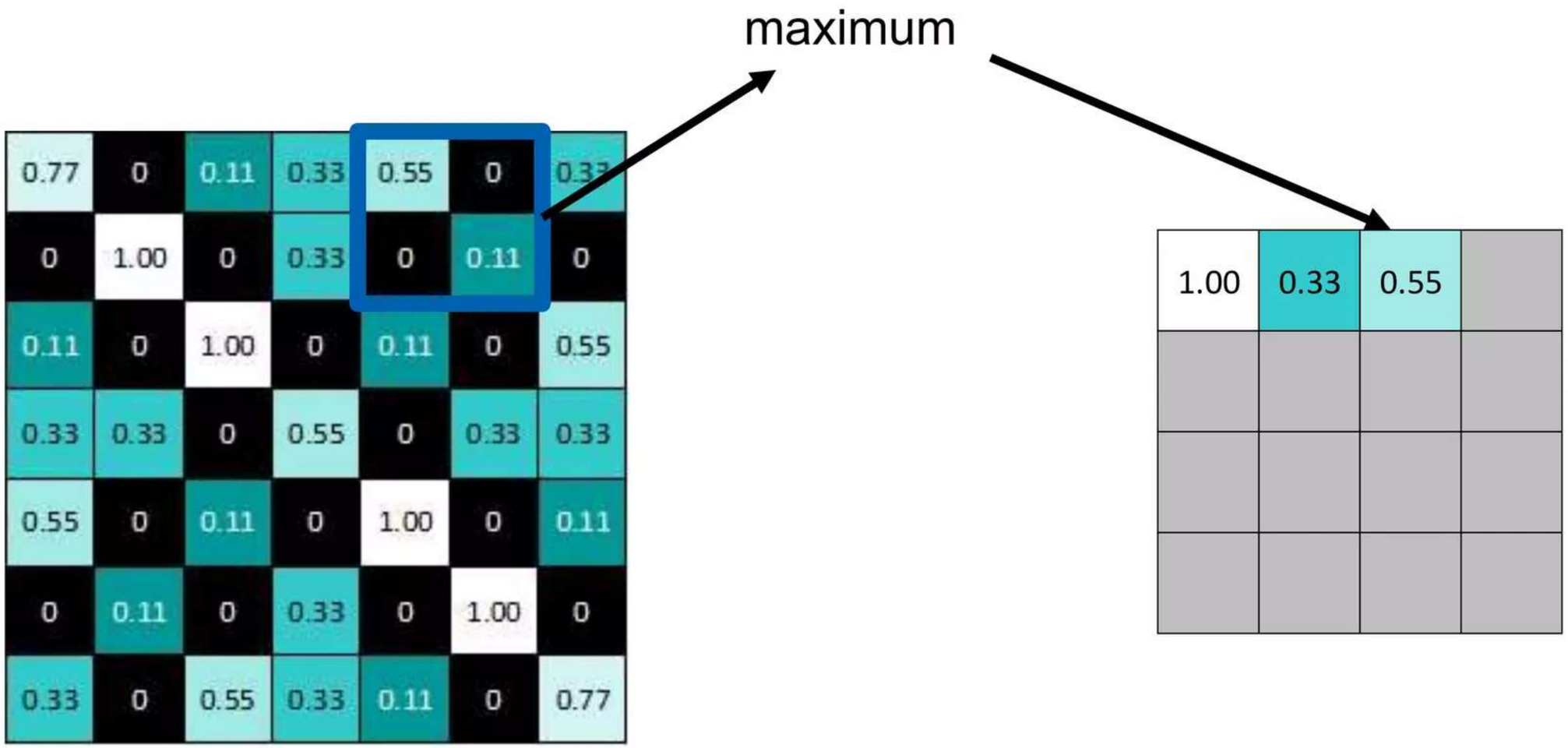
Pooling

- Pooling Filter Size = 2 X 2, Stride = 2



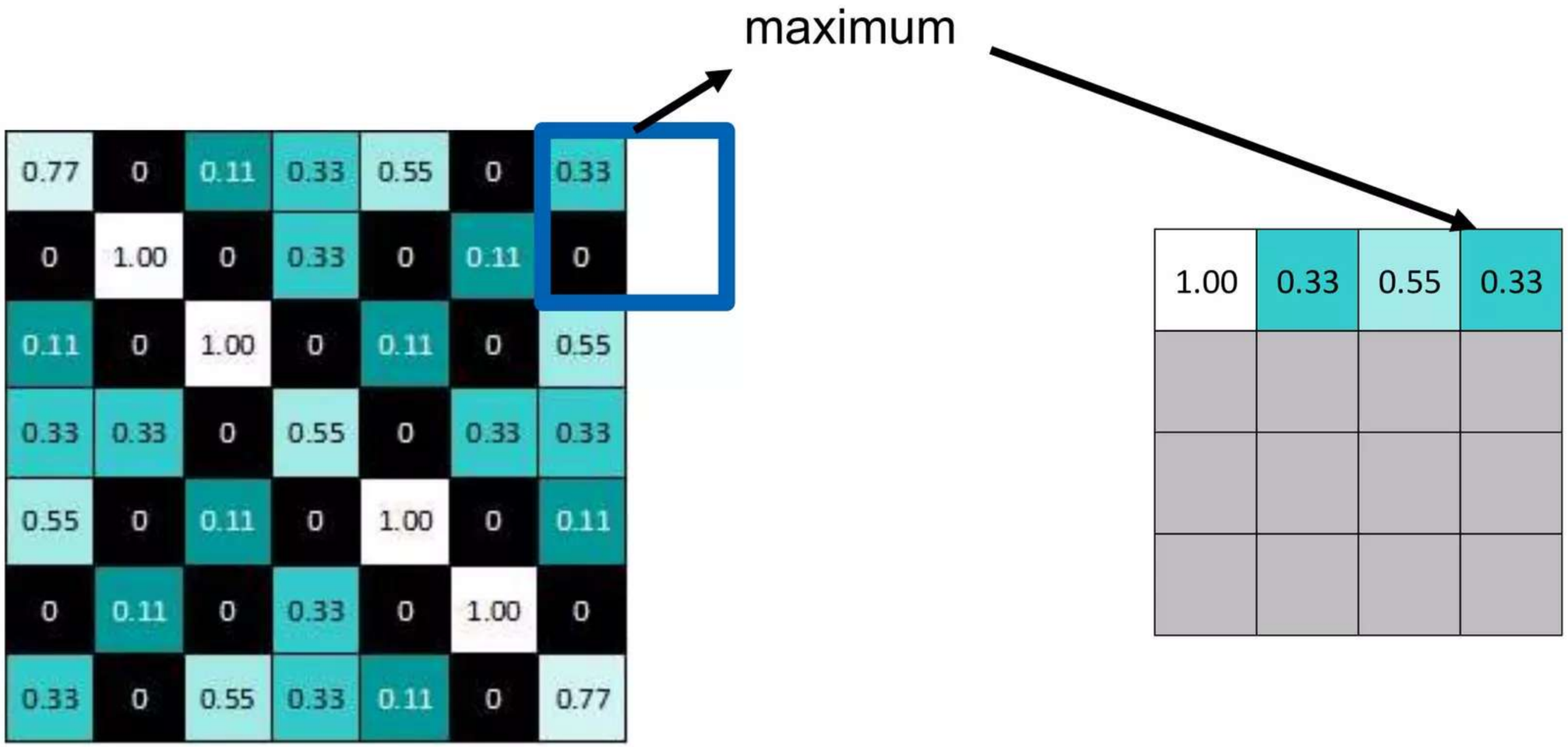
Pooling

- Pooling Filter Size = 2 X 2, Stride = 2



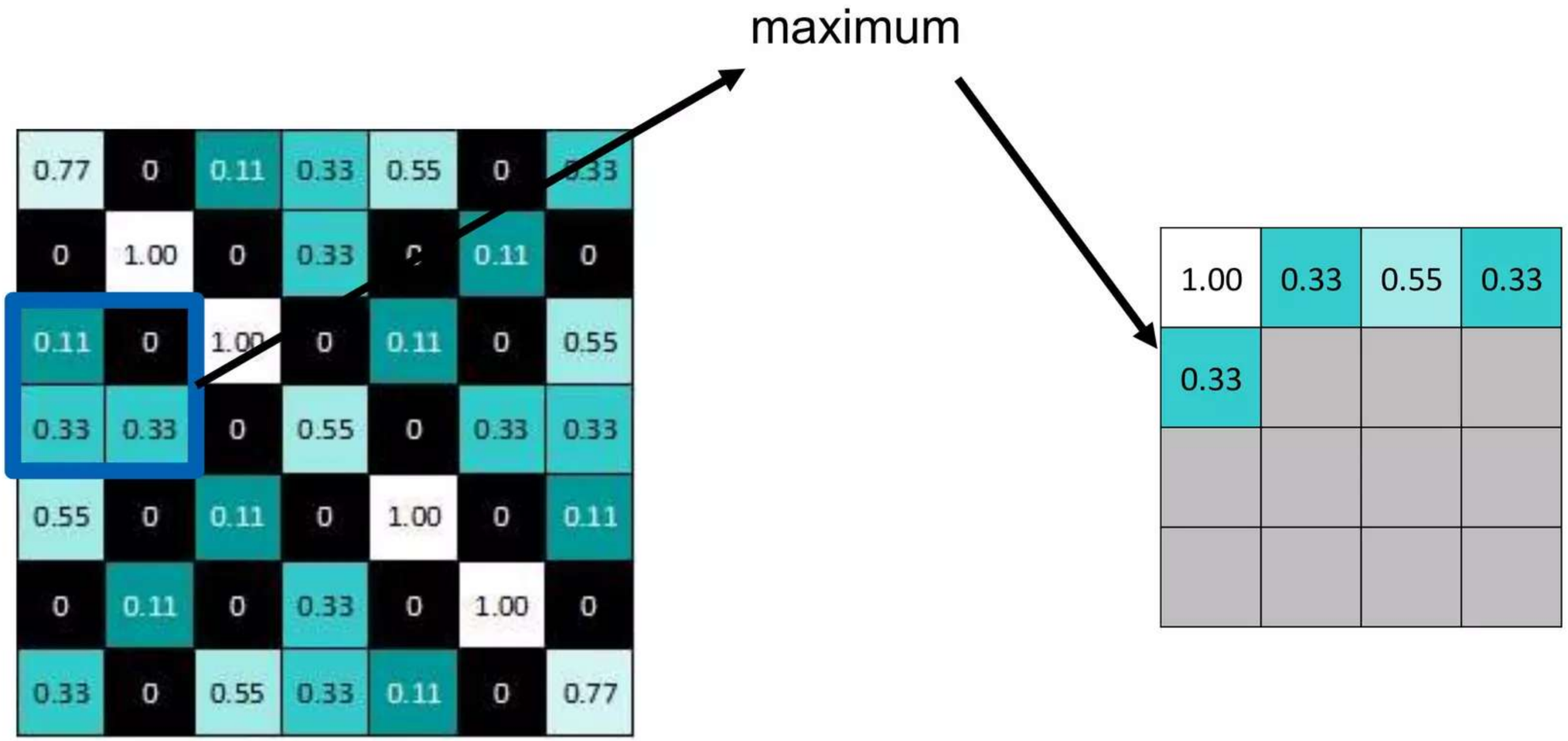
Pooling

- Pooling Filter Size = 2 X 2, Stride = 2



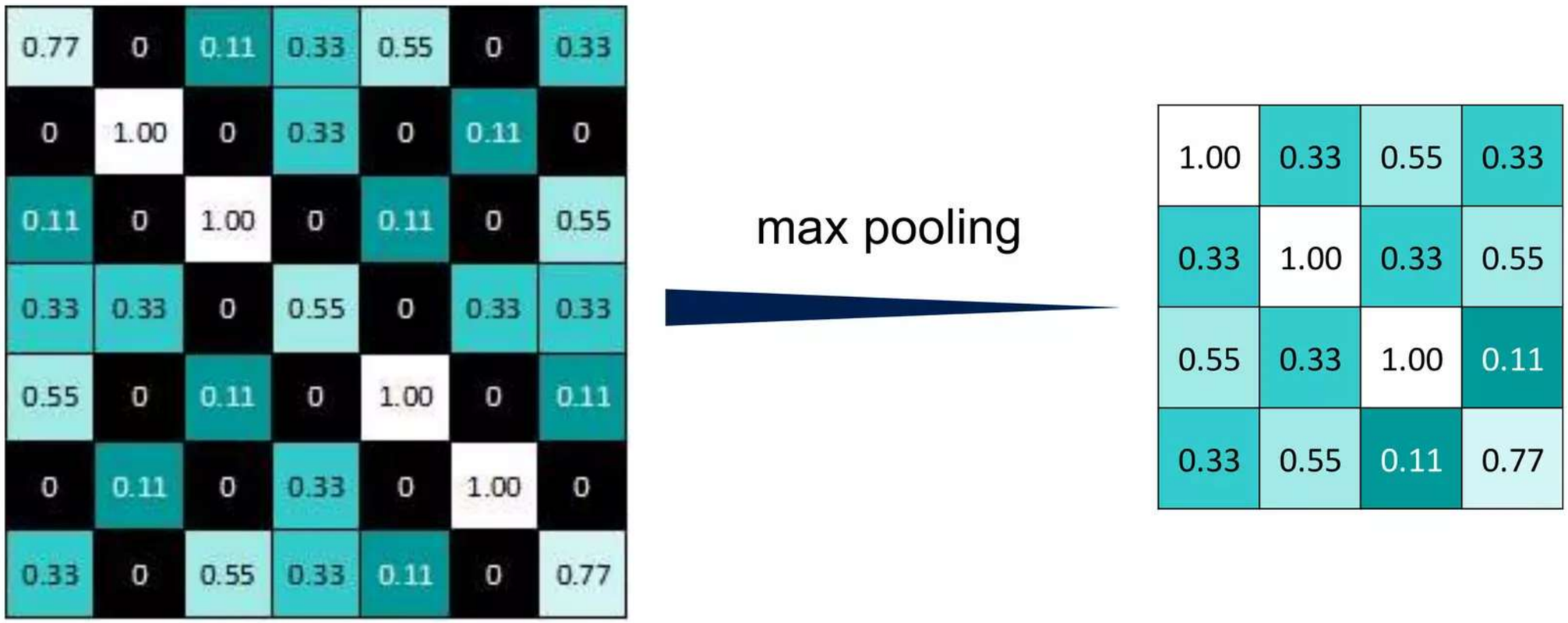
Pooling

- Pooling Filter Size = 2 X 2, Stride = 2



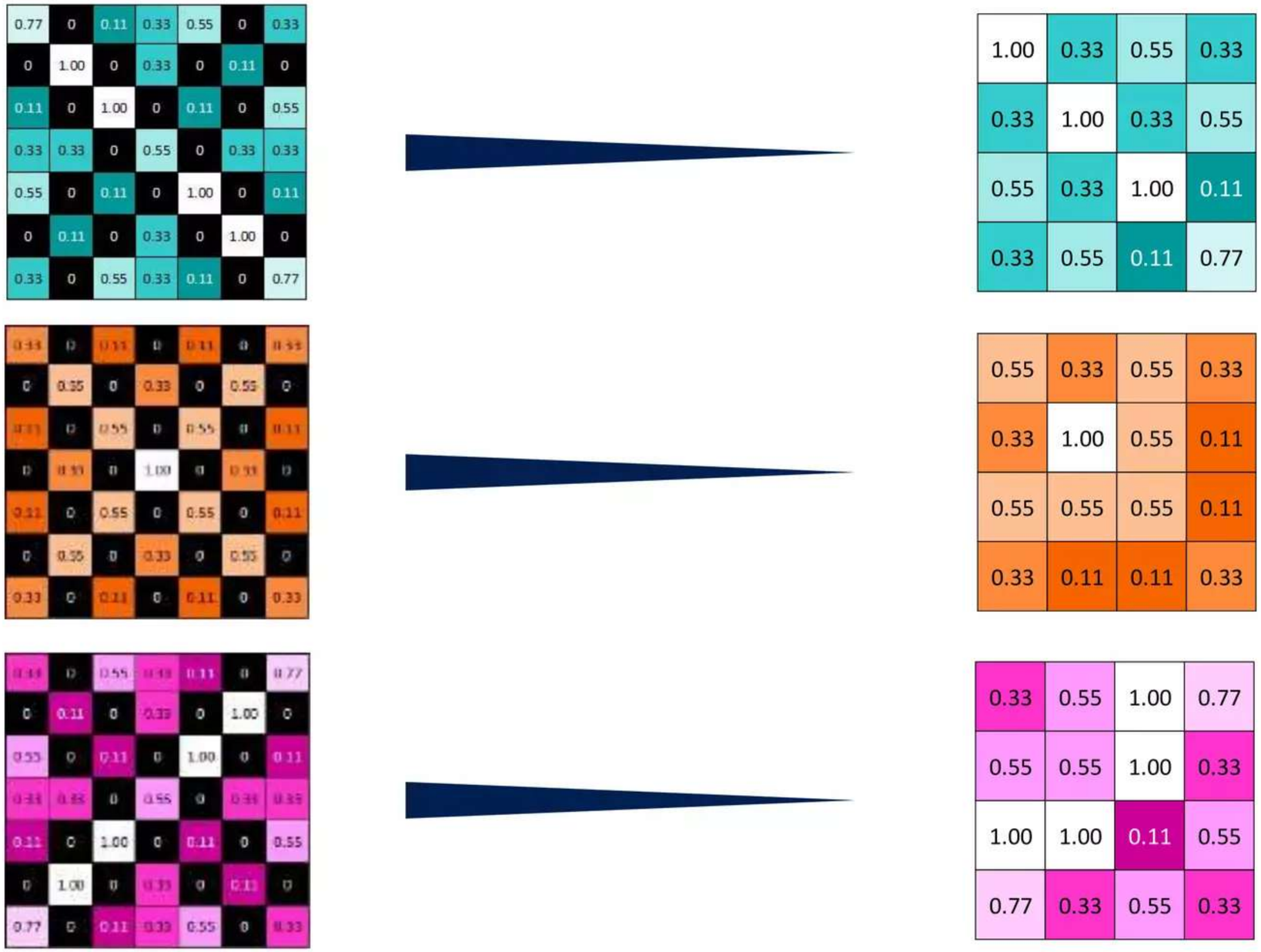
Pooling

- Pooling Filter Size = 2 X 2, Stride = 2



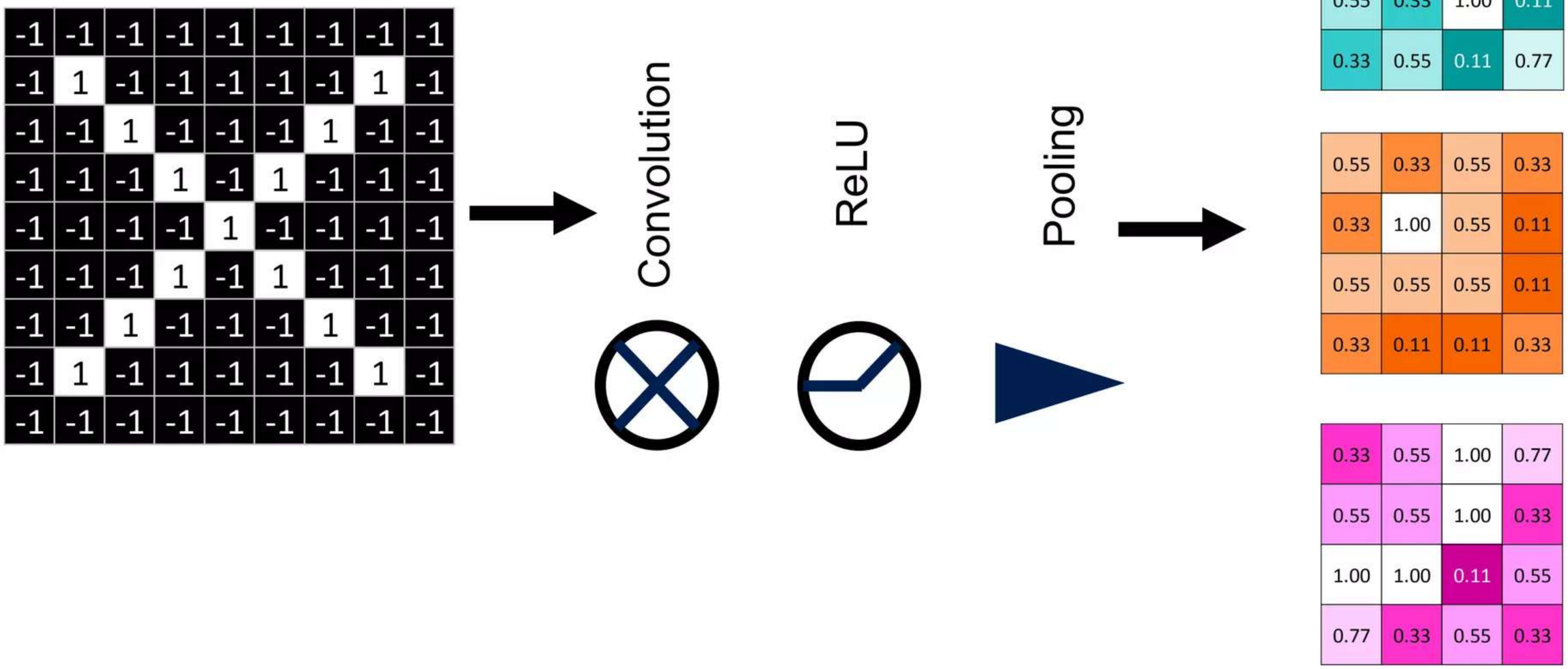
Pooling

A stack of 3 larger images becomes a stack of smaller images.

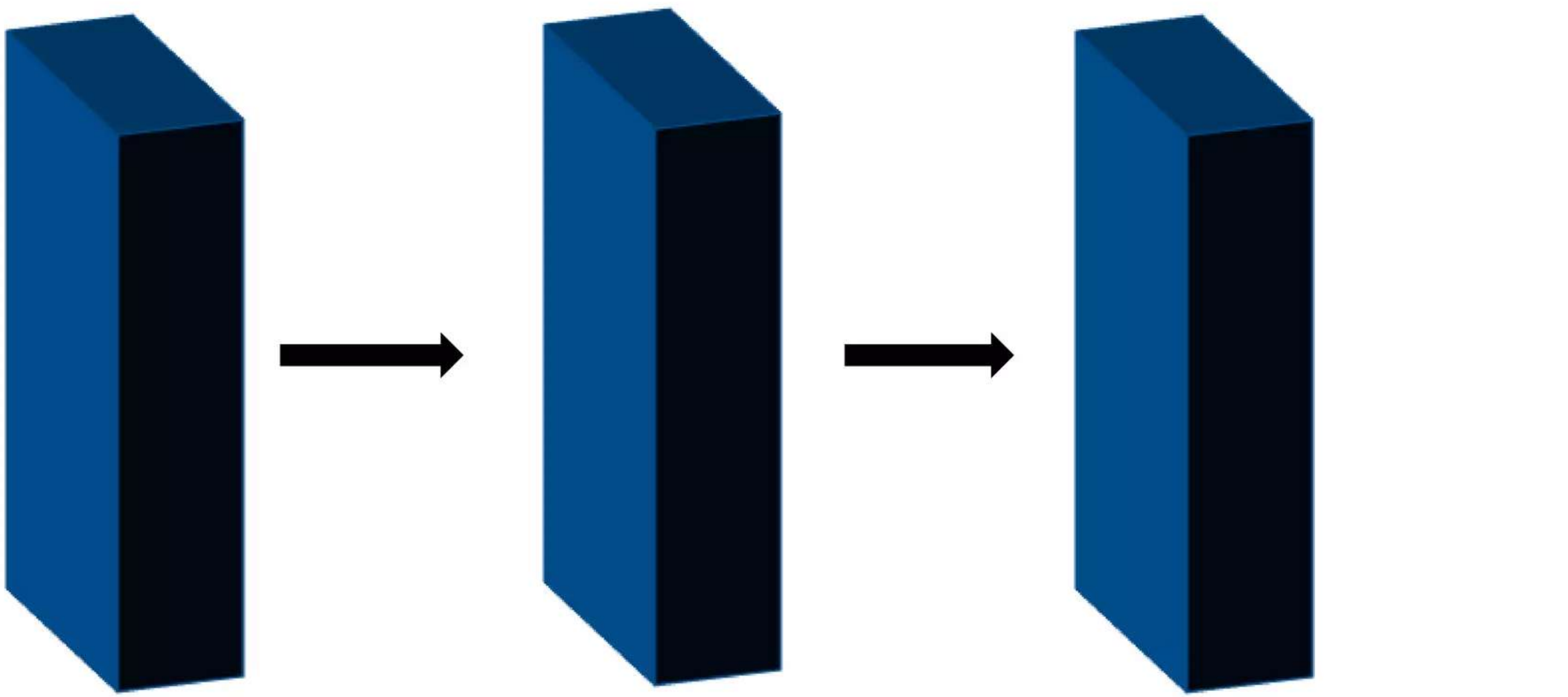


Layers get stacked

The output of one becomes the input of the next.

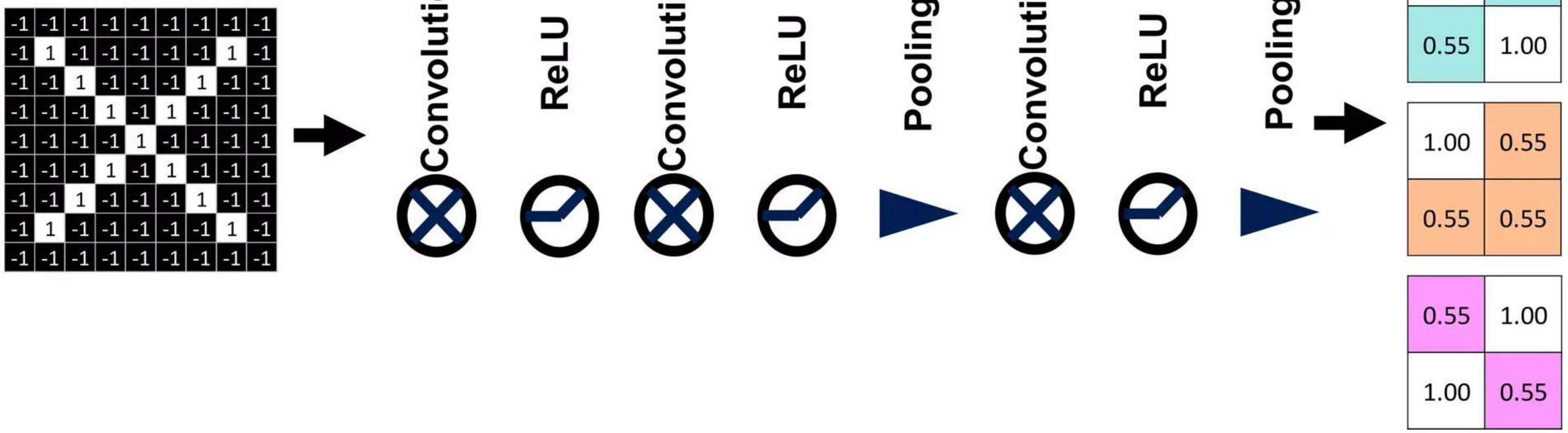


Layers Get Stacked - Example



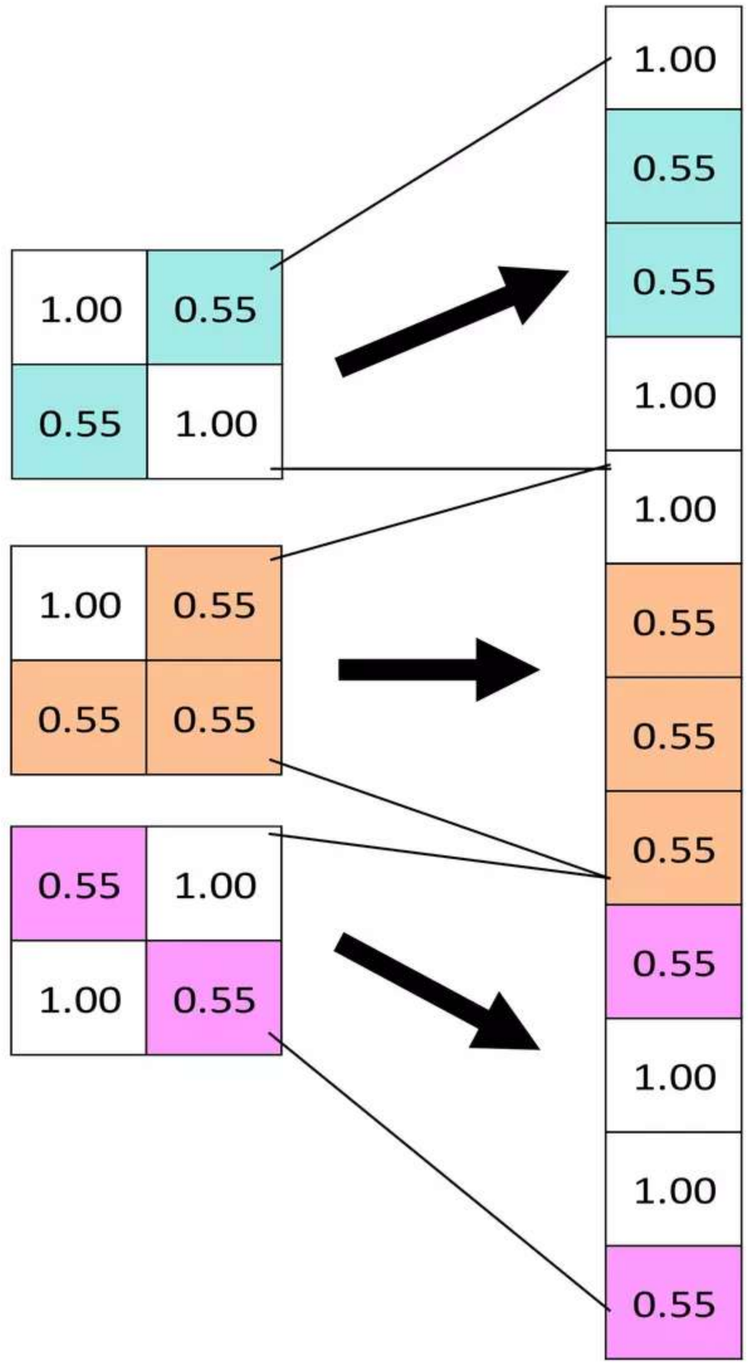
Deep stacking

Layers can be repeated several (or many) times.



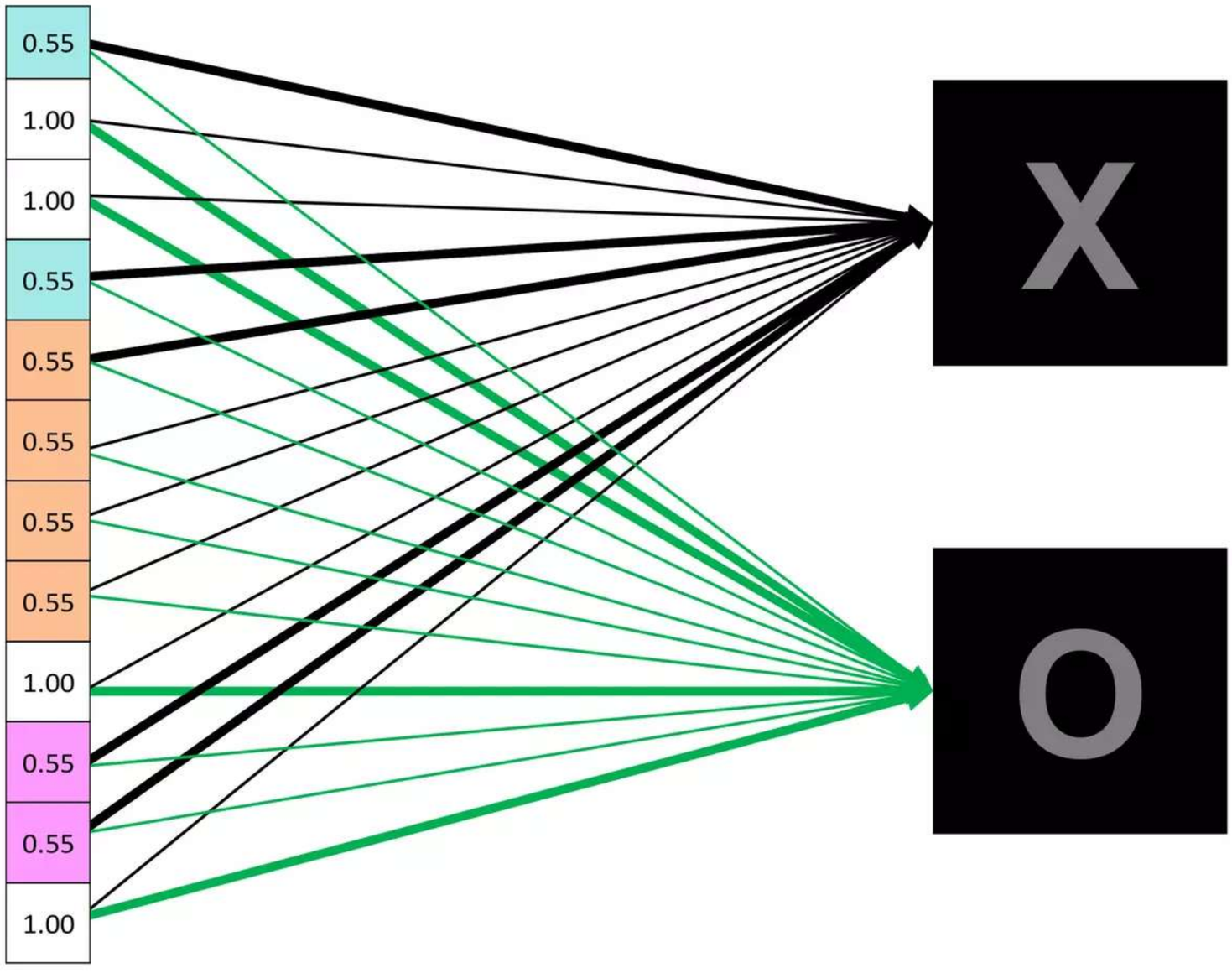
Fully connected layer

- Fully connected layers are the normal flat feed-forward neural network layers.
- These layers may have a non-linear activation function or a softmax activation in order to predict classes.
- To compute our output, we simply re-arrange the output matrices as a 1-D array.

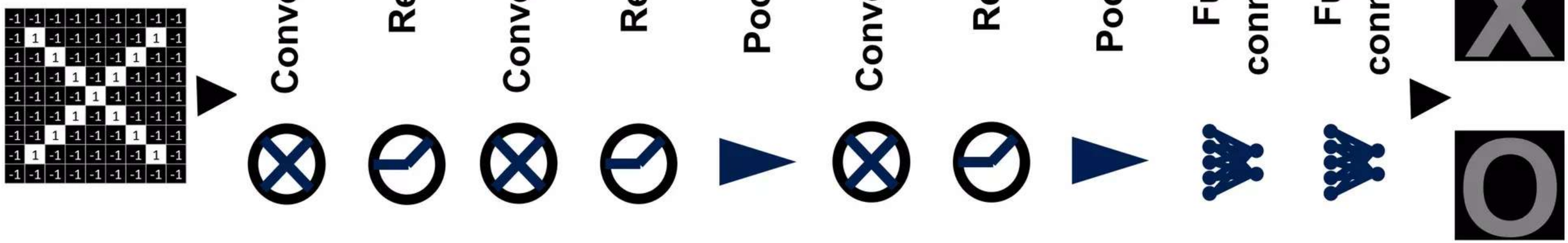


Fully connected layer

- A summation of product of inputs and weights at each output node determines the final prediction



Putting it all together



Agenda

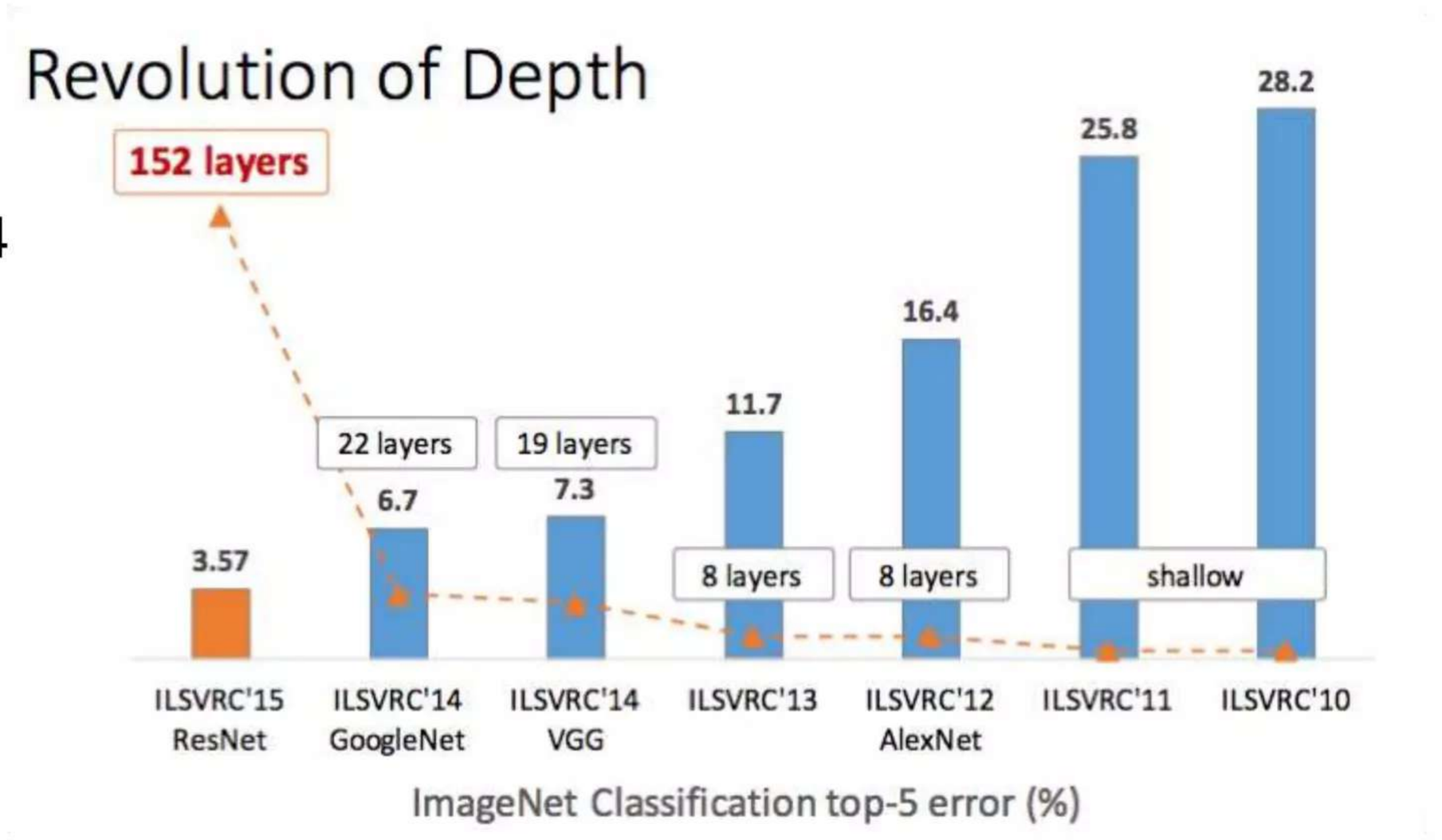
- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

Hyperparameters (knobs)

- Convolution
  - Filter Size
  - Number of Filters
  - Padding
  - Stride
- Pooling
  - Window Size
  - Stride
- Fully Connected
  - Number of neurons

Case Studies

- LeNet – 1998
- AlexNet – 2012
- ZFNet – 2013
- VGG – 2014
- GoogLeNet – 2014
- ResNet – 2015



Agenda

- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

Agenda

- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

Recurrent Neural Network -Intro

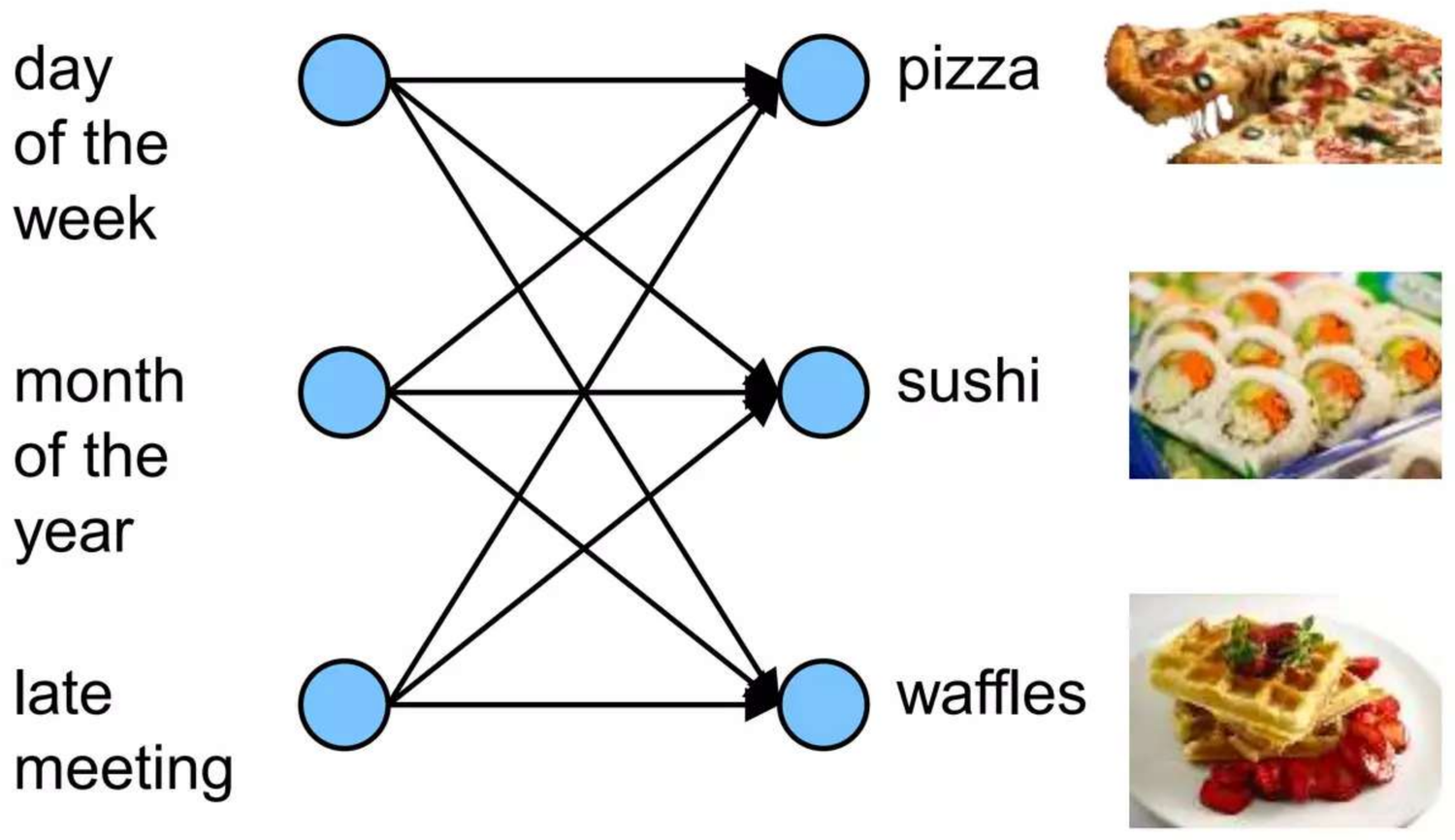
- Humans don’t start their thinking from scratch every second. You don’t throw everything away and start thinking from scratch again. Your thoughts have persistence.

- You make use of context and previous knowledge to understand what is coming next

- Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist.

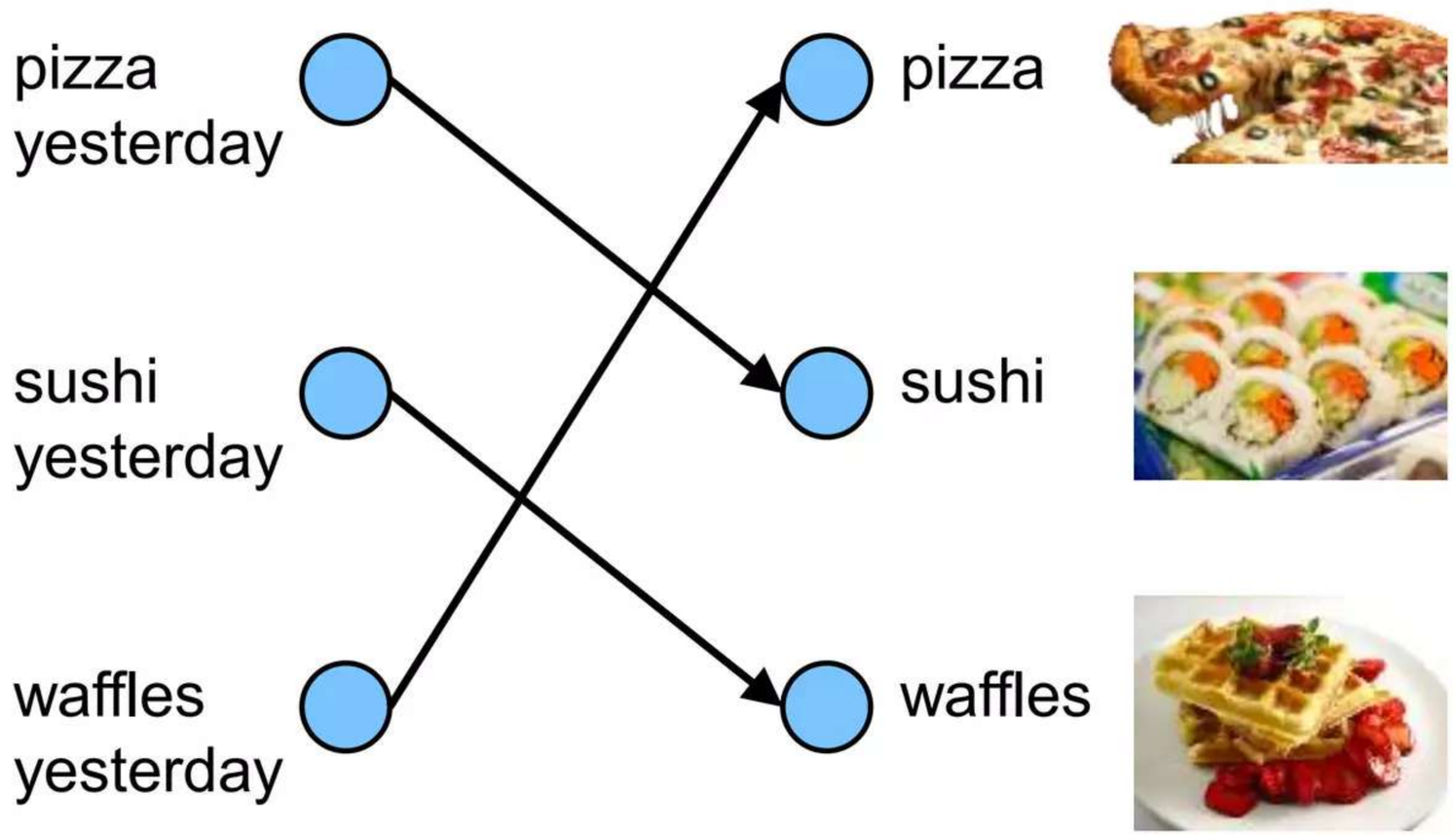
Recurrent Neural Network -Intro

What's for dinner?

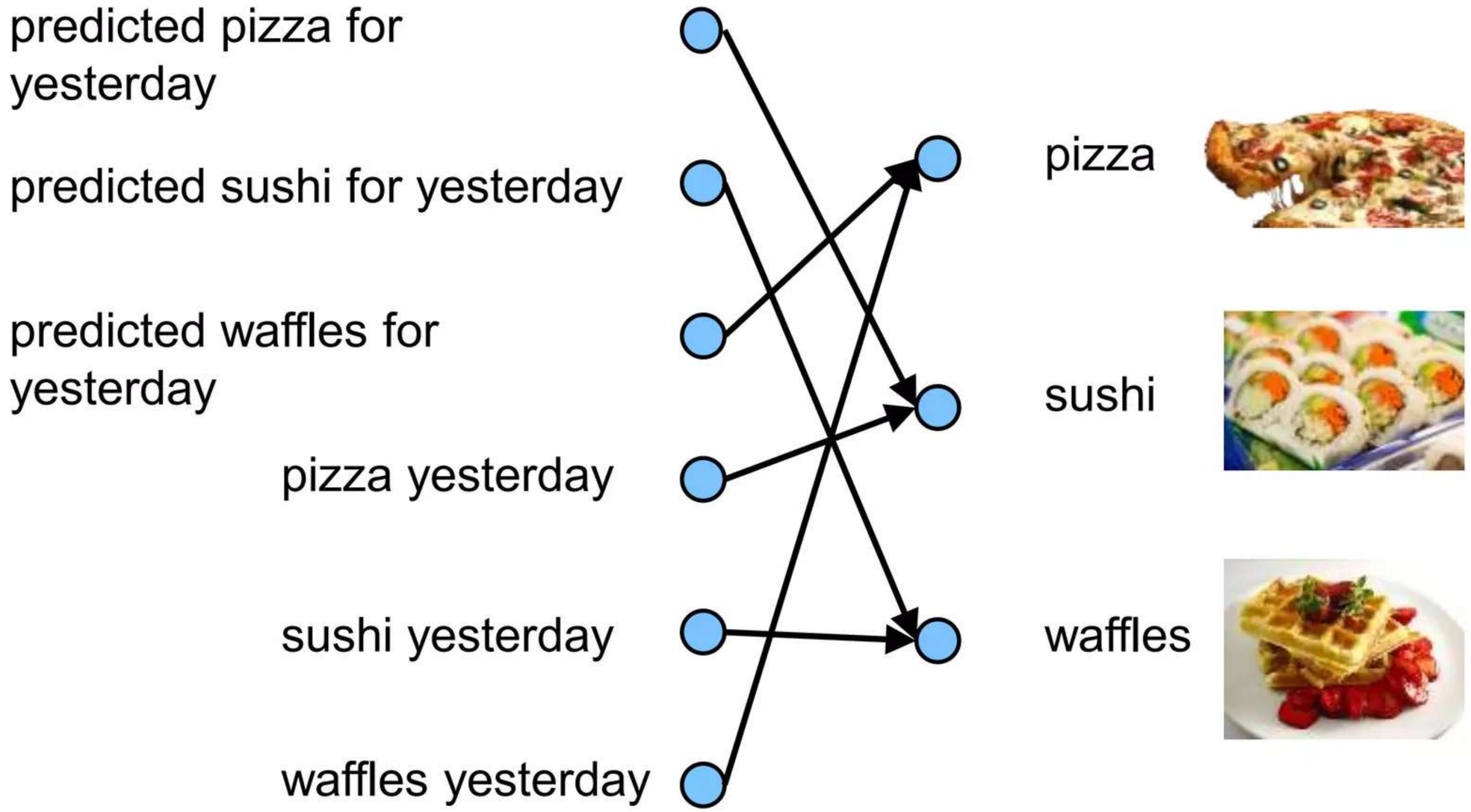


Recurrent Neural Network -Intro

What's for dinner?



Recurrent Neural Network -Intro



Vectors as Inputs

A vector is a list of values

"High is 67 F.
Low is 43 F.
Wind is 13 mph.
.25 inches of rain.
Relative humidity
is 83%."

=

High temperature	67
Low temperature	43
Wind speed	13
Precipitation	.25
Humidity	.83

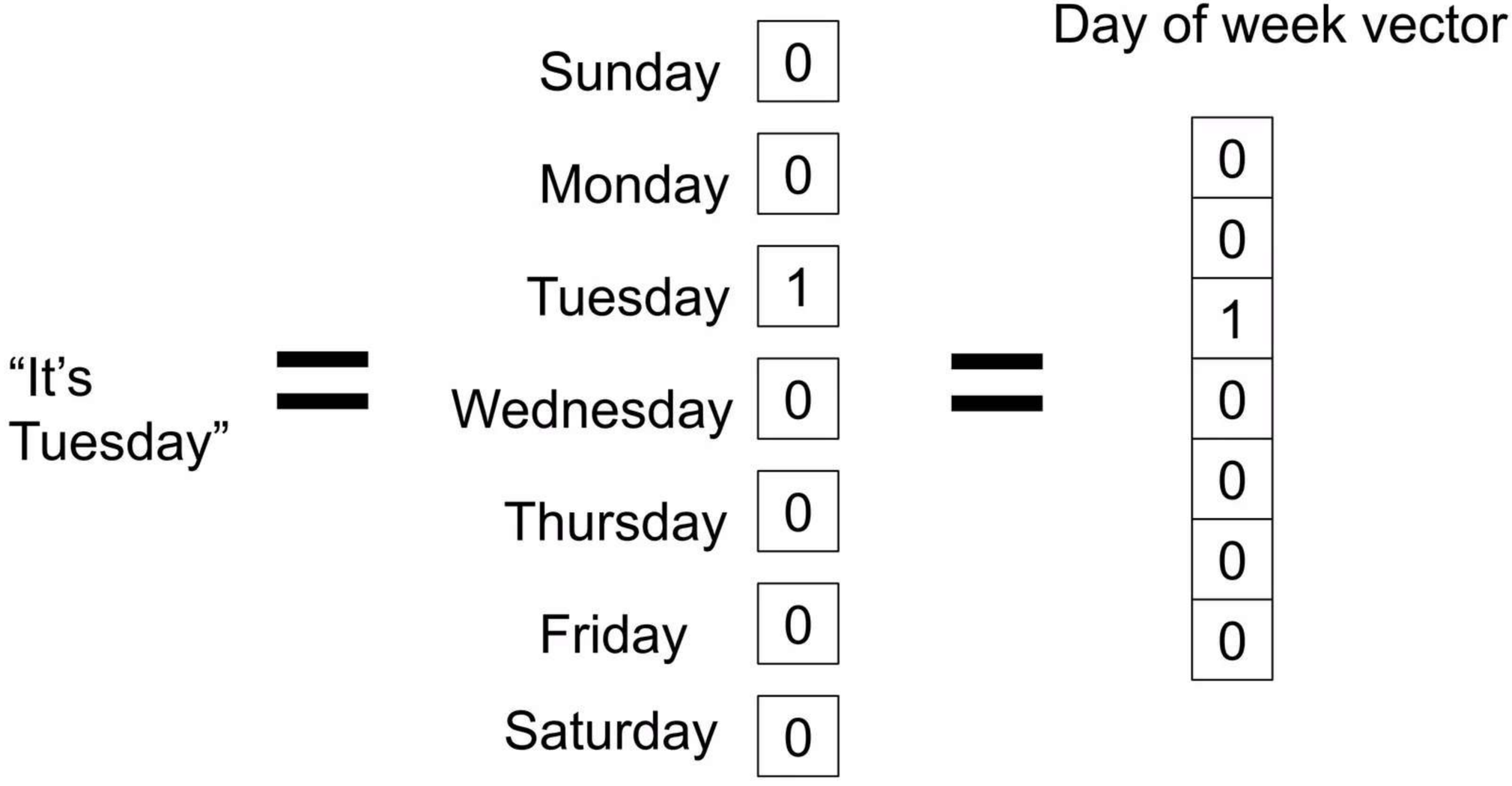
=

Weather vector

67
43
13
.25
.83

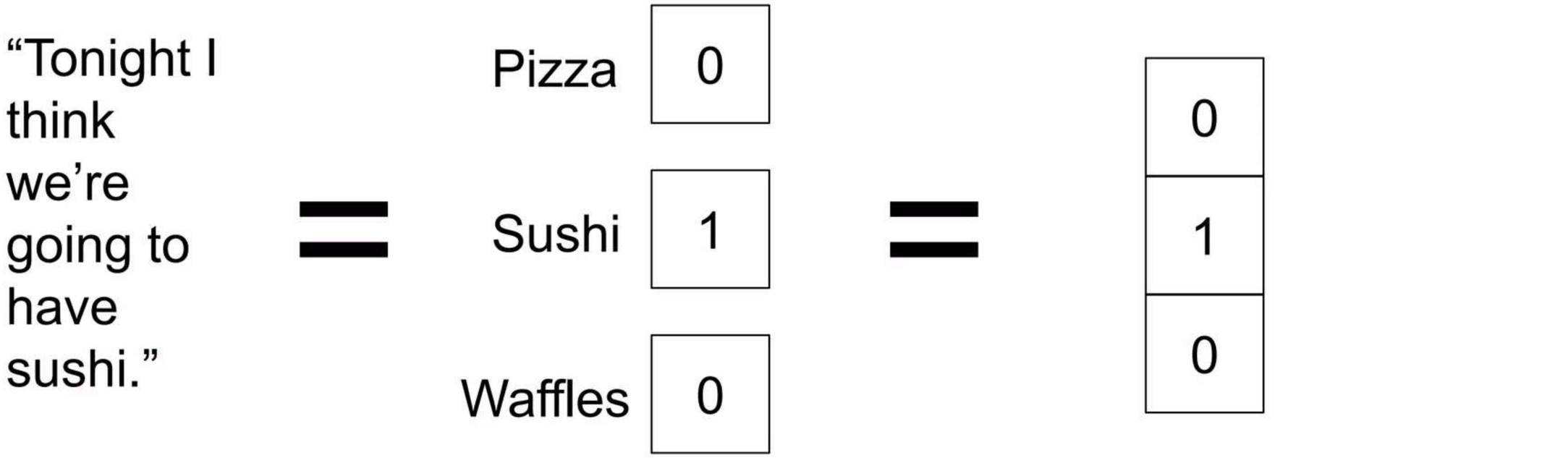
Vectors as Inputs

A vector is a list of values

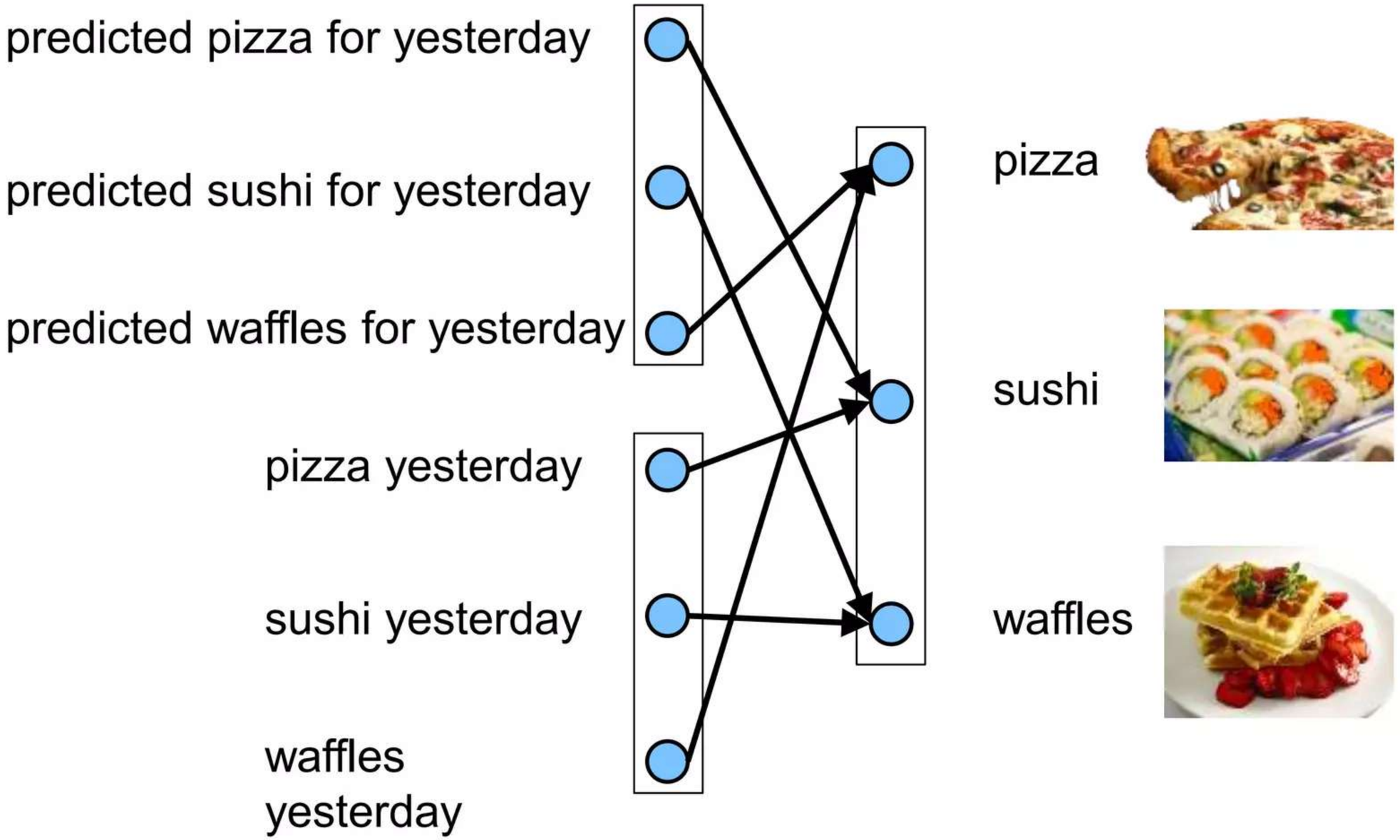


Vectors as Inputs

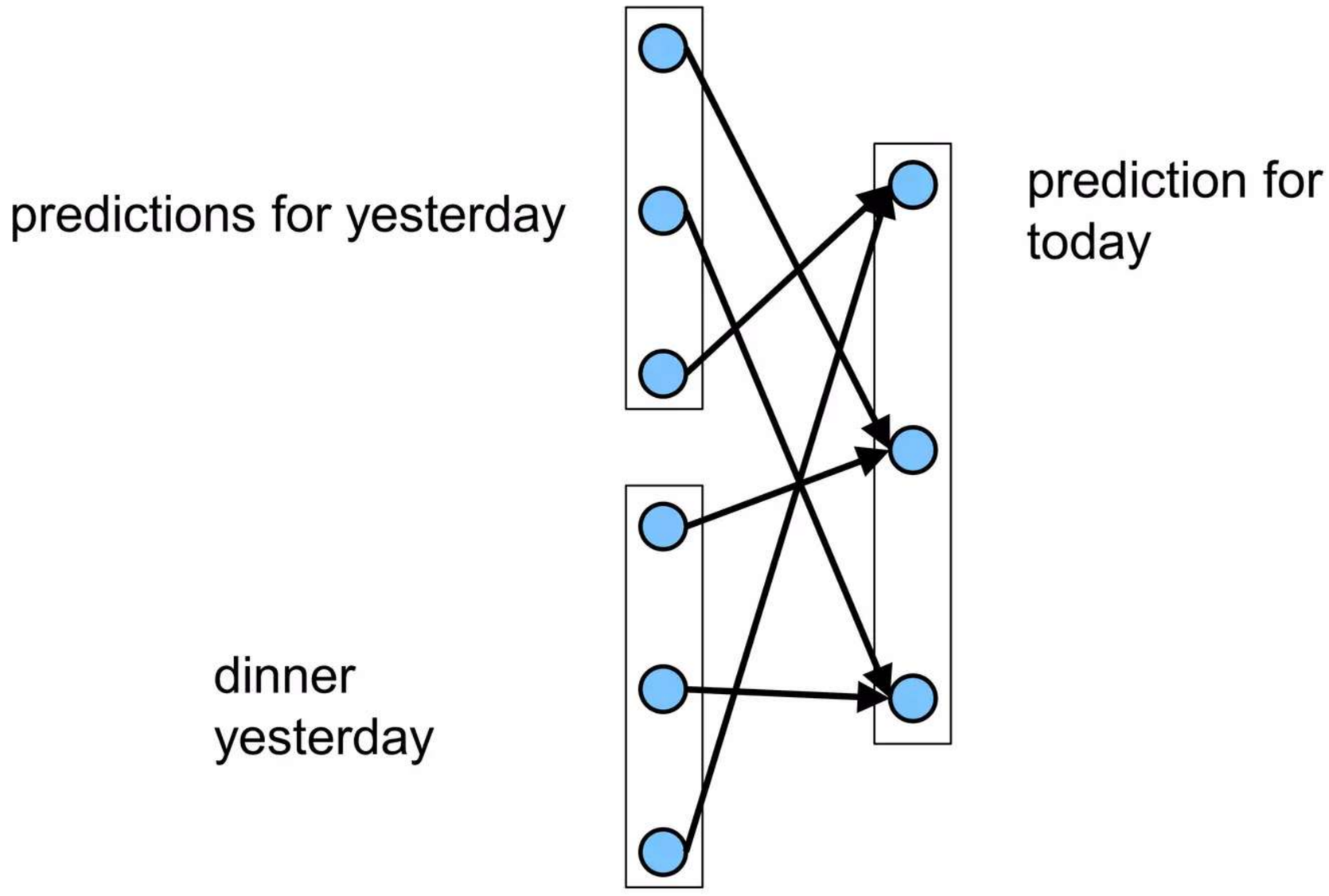
A vector is a list of values



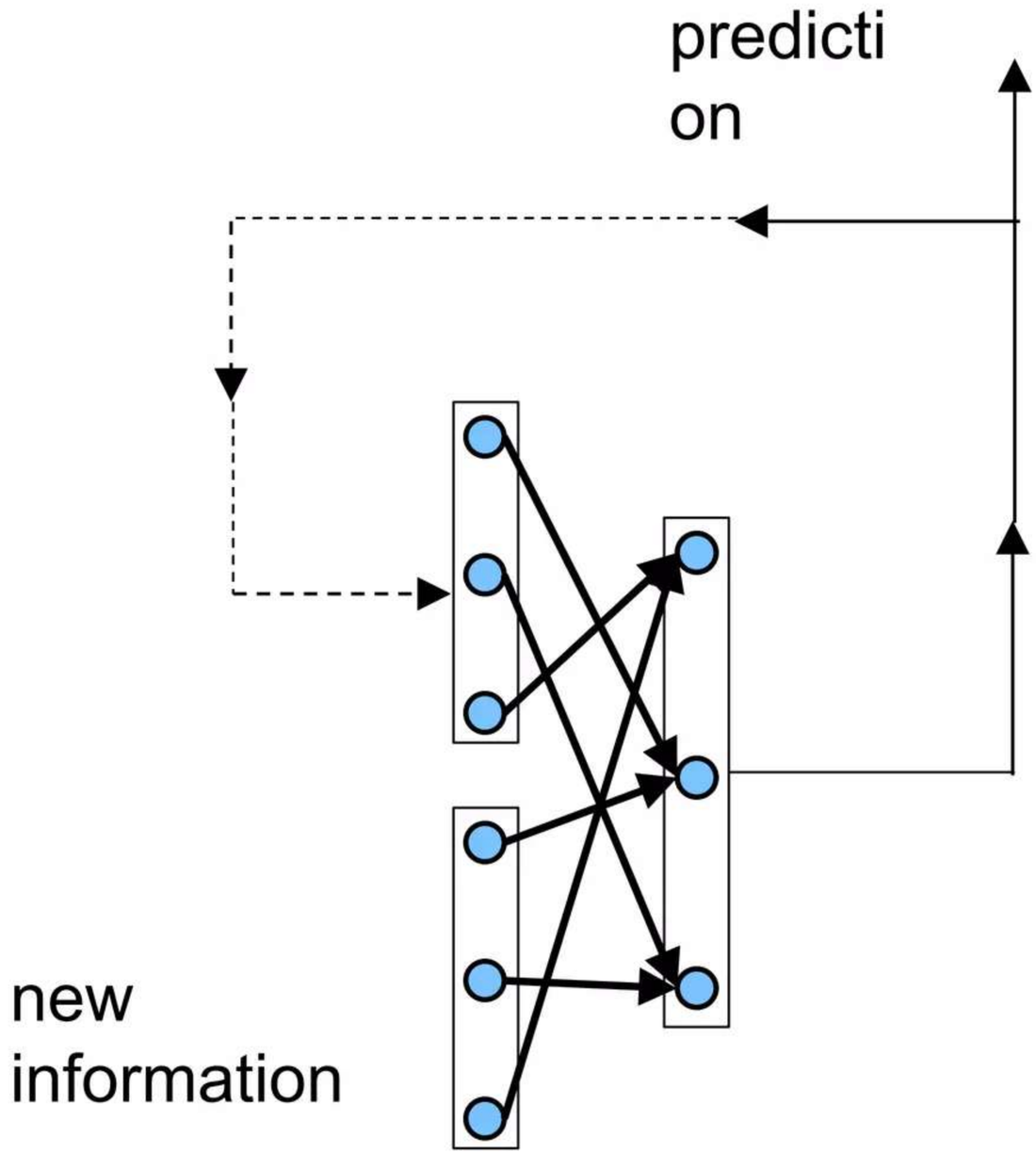
How RNN's Work



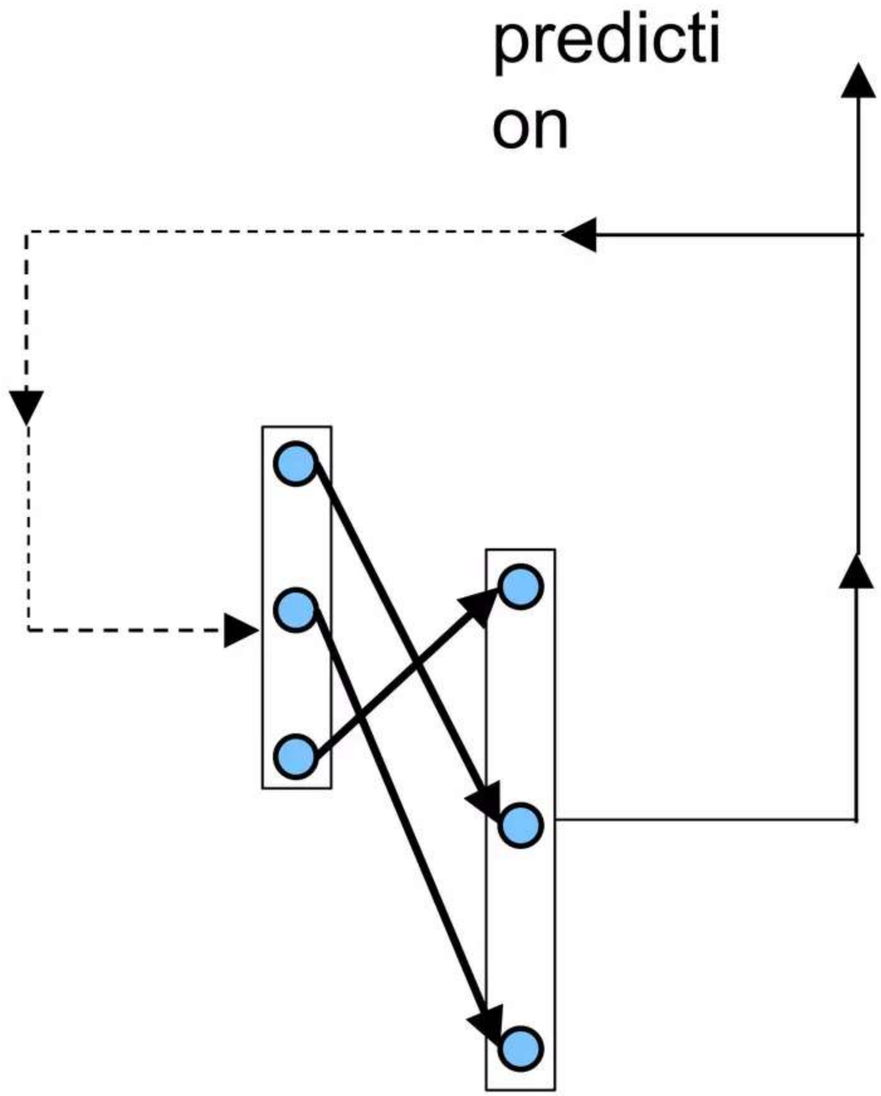
How RNN's Work



How RNN's Work

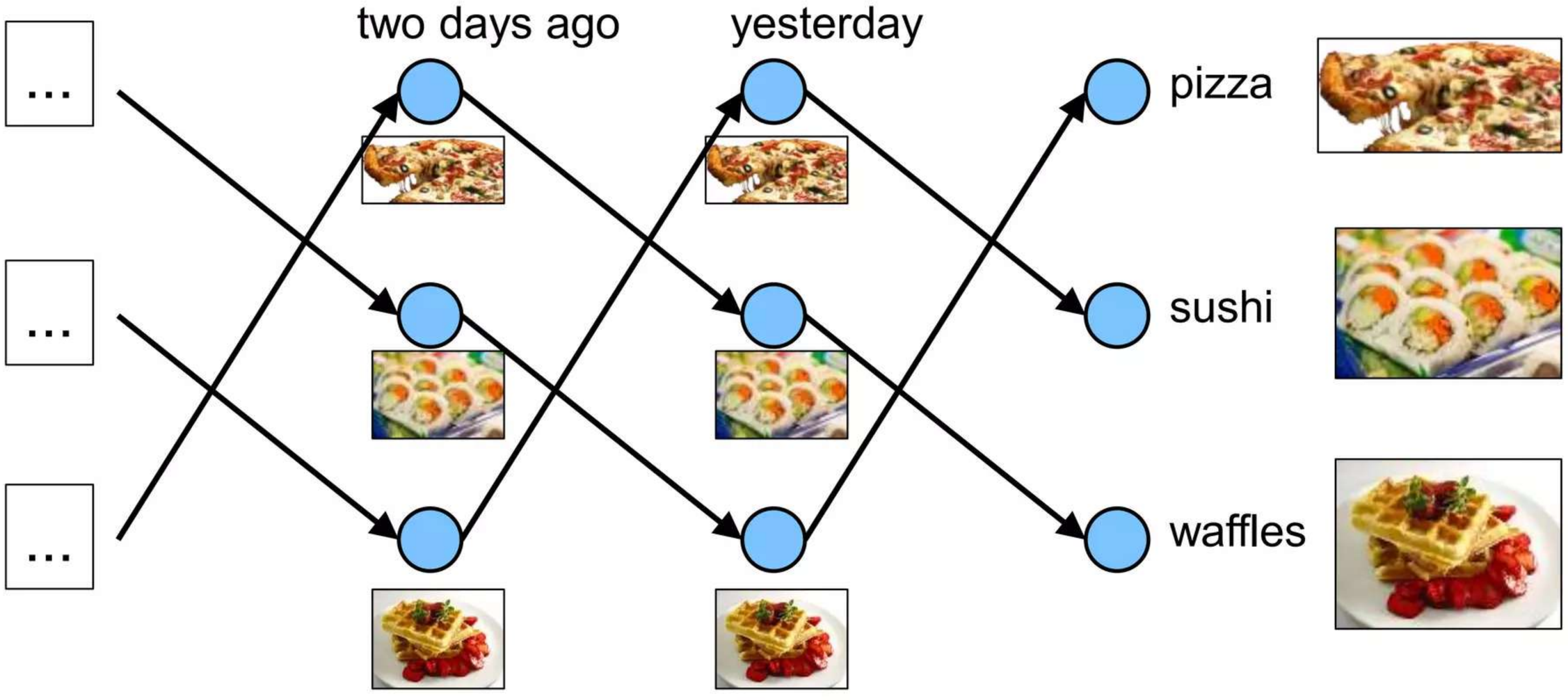


How RNN's Work

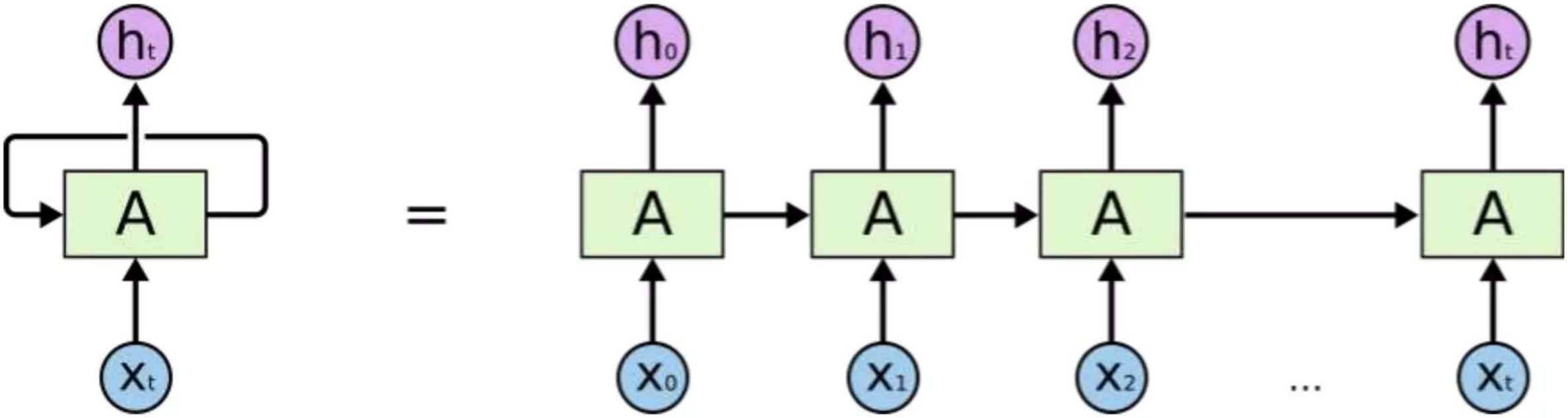


How RNN's Work

Unrolled predictions



RNN's – Overall Structure



These loops make recurrent neural networks seem kind of mysterious.

A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. Consider what happens if we unroll the loop:

RNN's - Example

Consider Simple statements

Harry met Sally.

Sally met James.

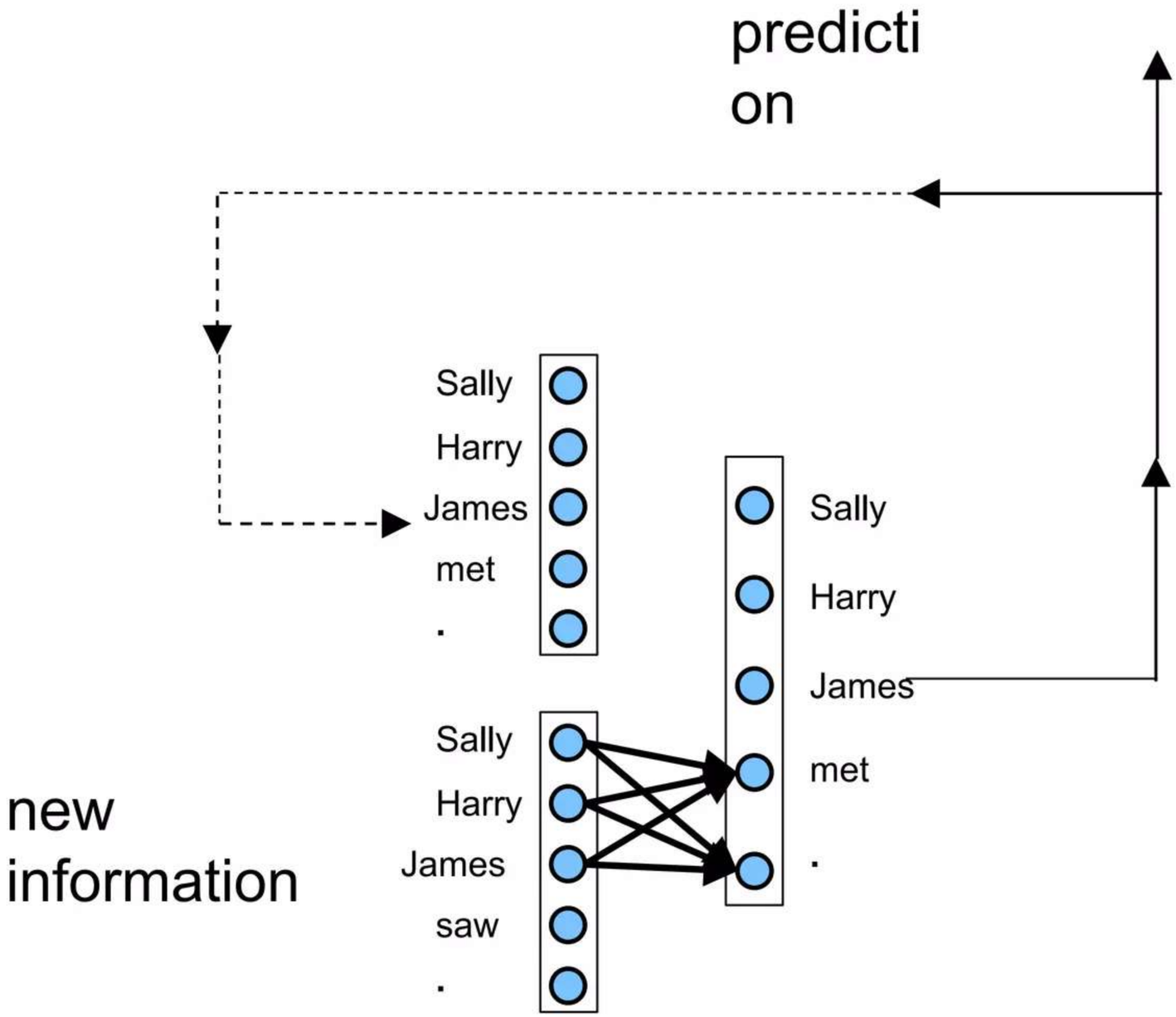
James met Harry.

...

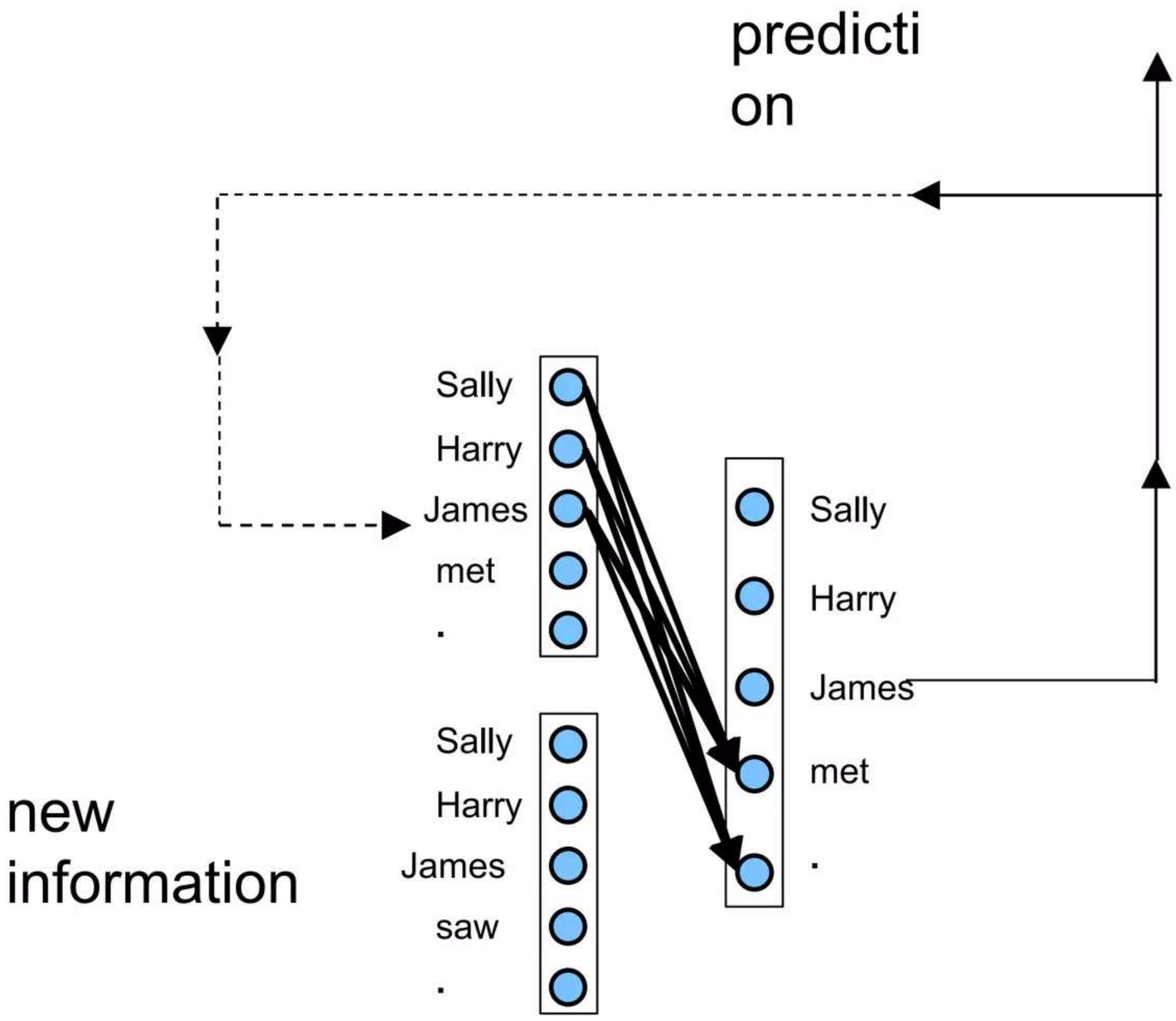
Dictionary : {Harry, Sally, James, met, .}

For the sake of illustration, lets build an RNN that looks at only one previous word.

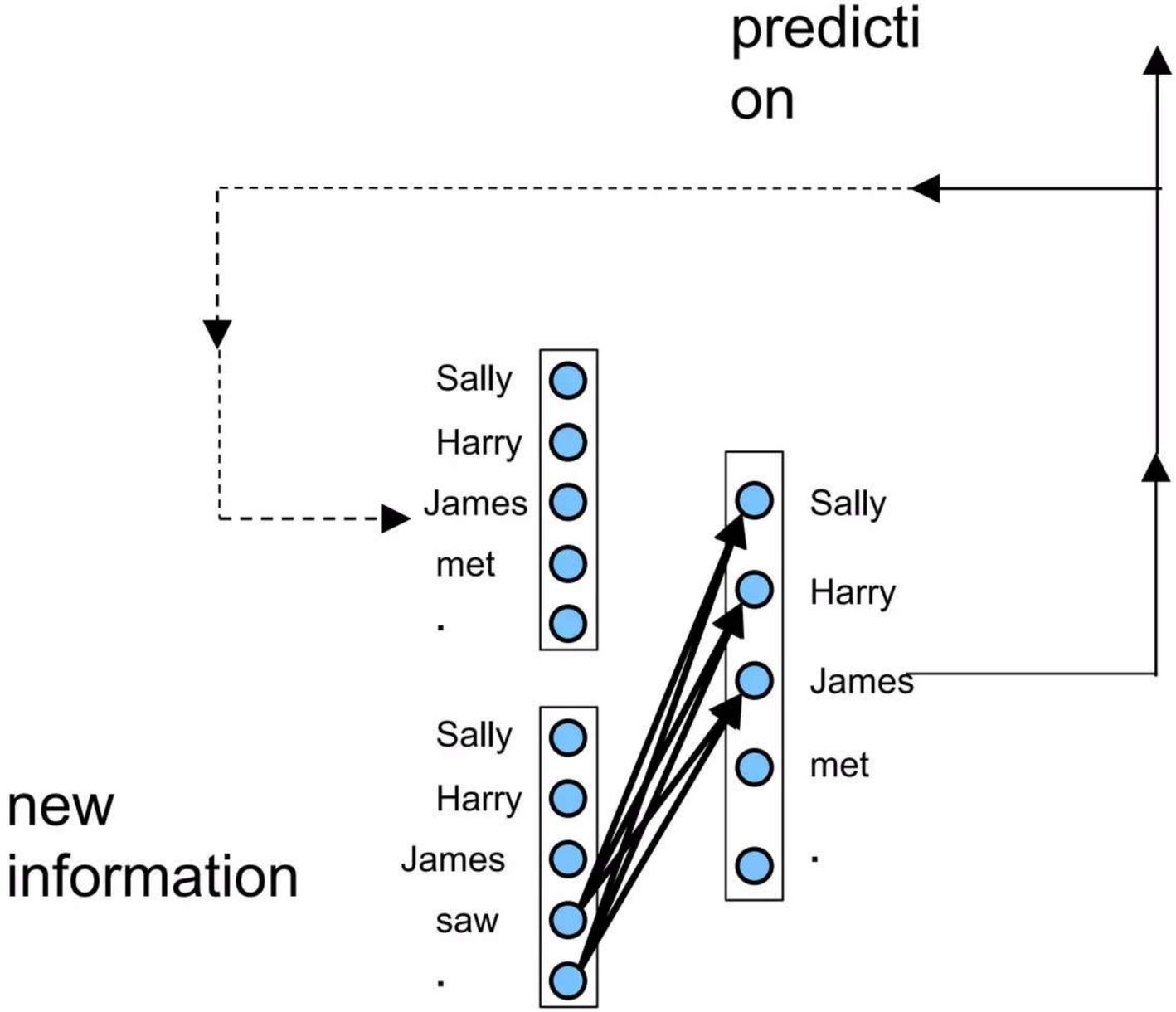
RNN's - Example



RNN's - Example

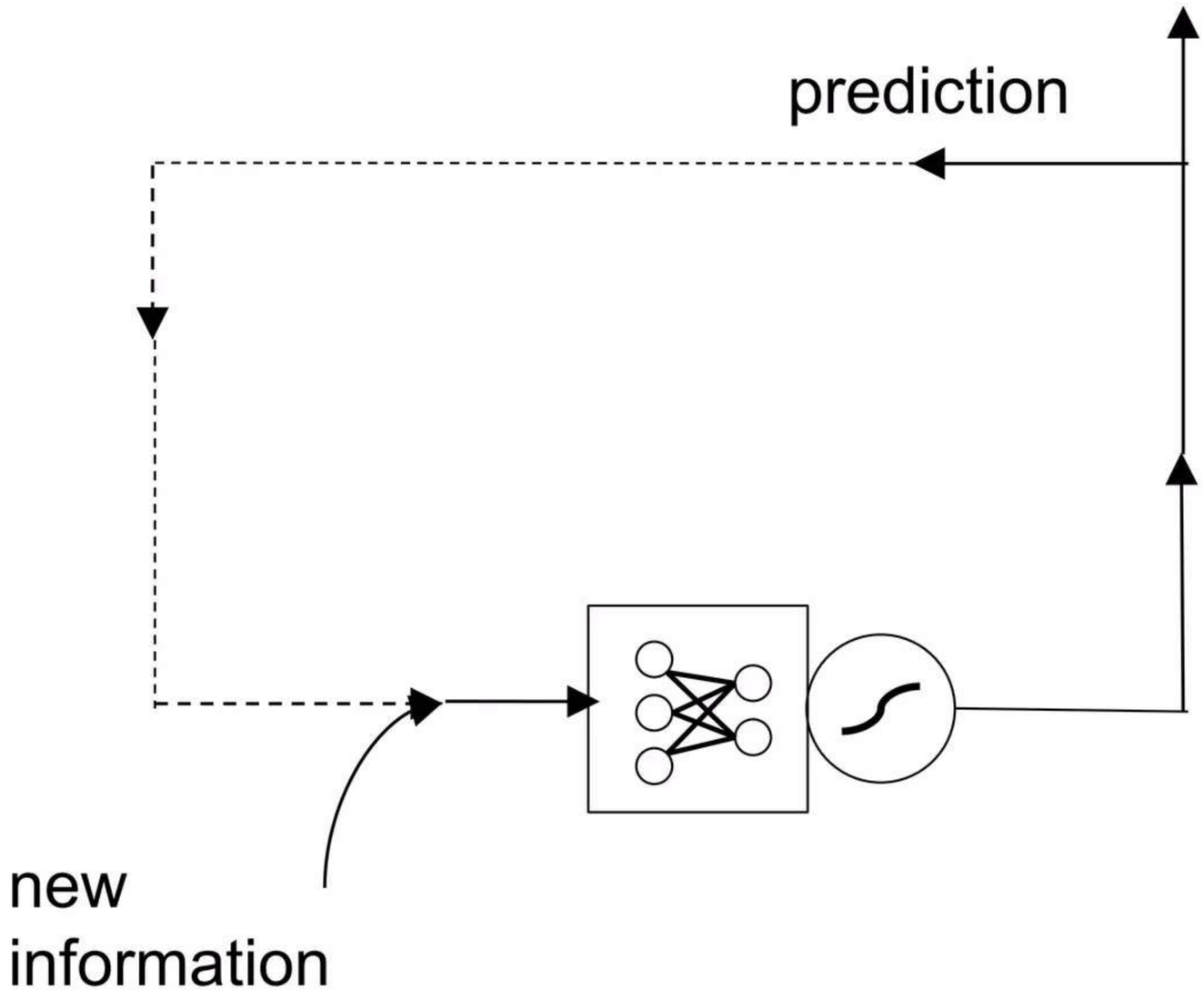


RNN's - Example

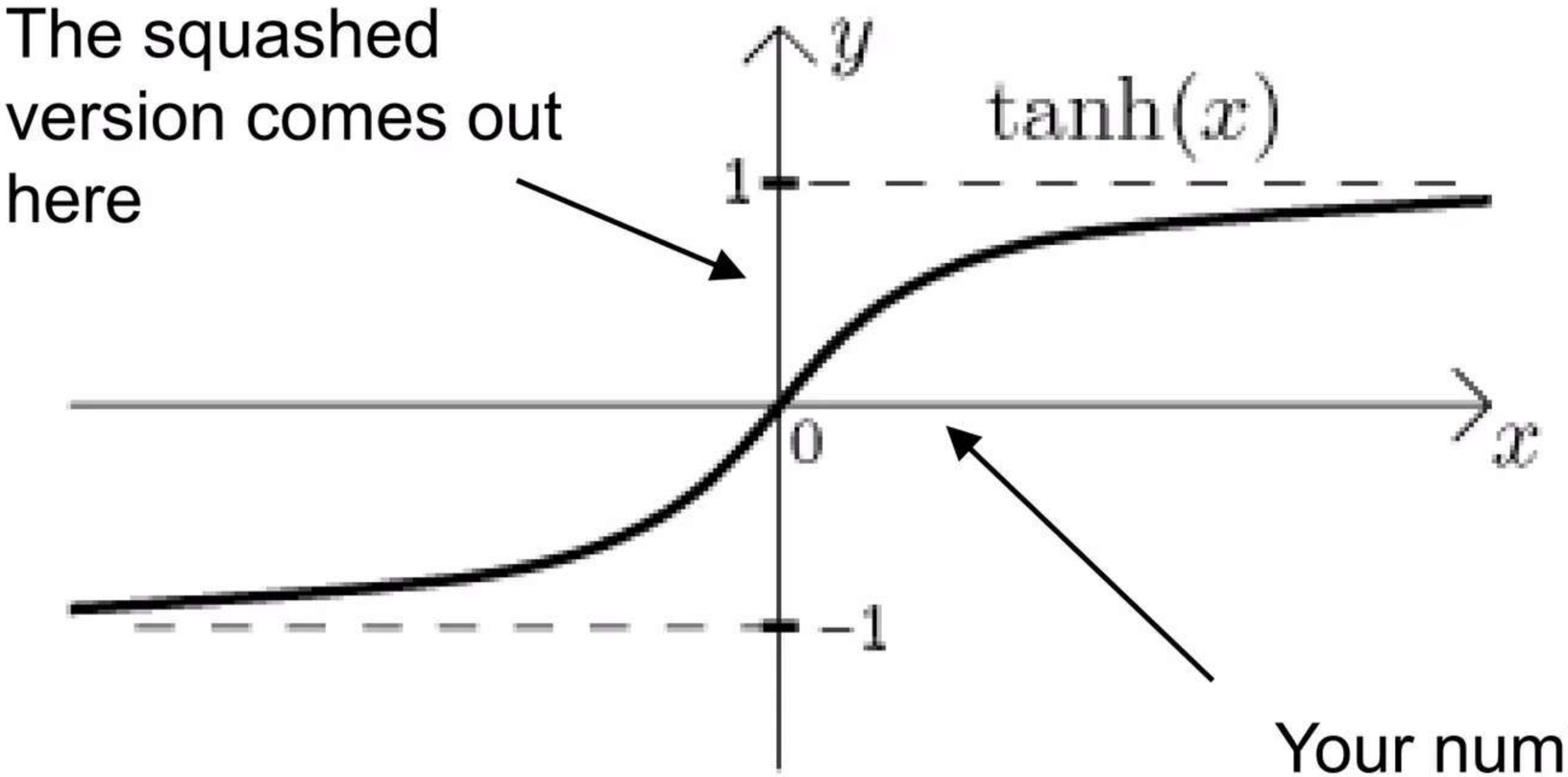


RNN's - Example

recurrent neural network



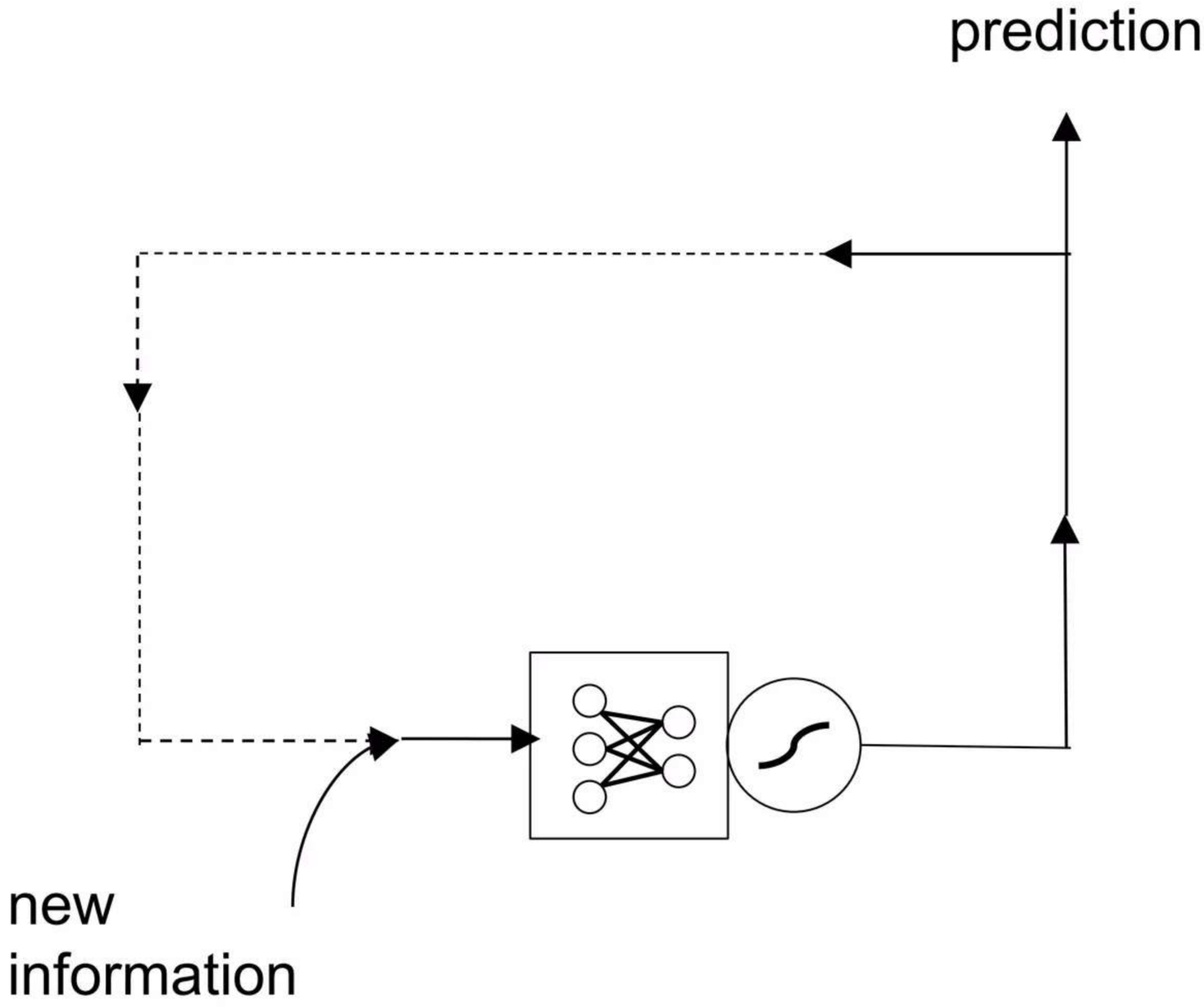
Hyperbolic tangent (tanh) squashing function



No matter what you start with, the answer stays between -1 and 1.

RNN's – Overall Structure

recurrent neural network



Agenda

- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

Why RNN's Fail

Harry met Harry.

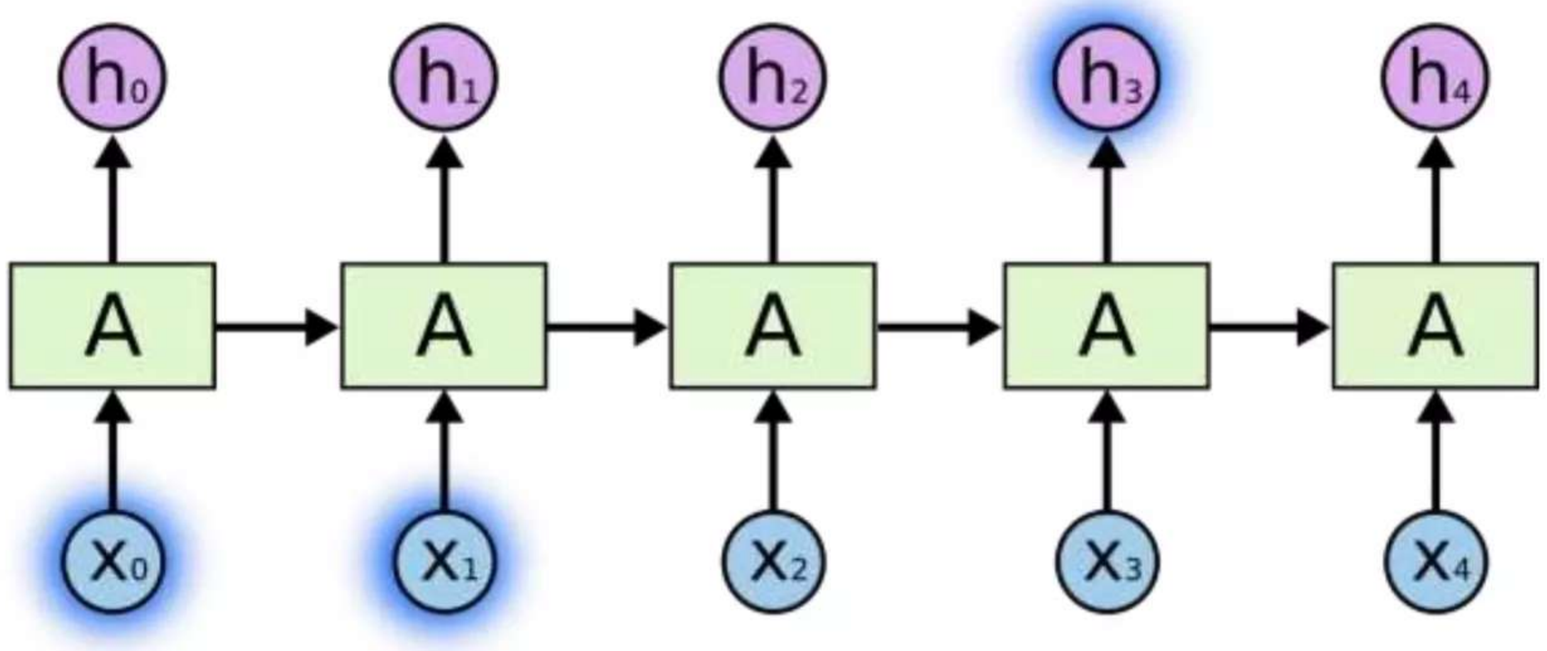
Sally met James met Harry met …

James. Harry. Sally.

This can be easily resolved by going back multiple time steps

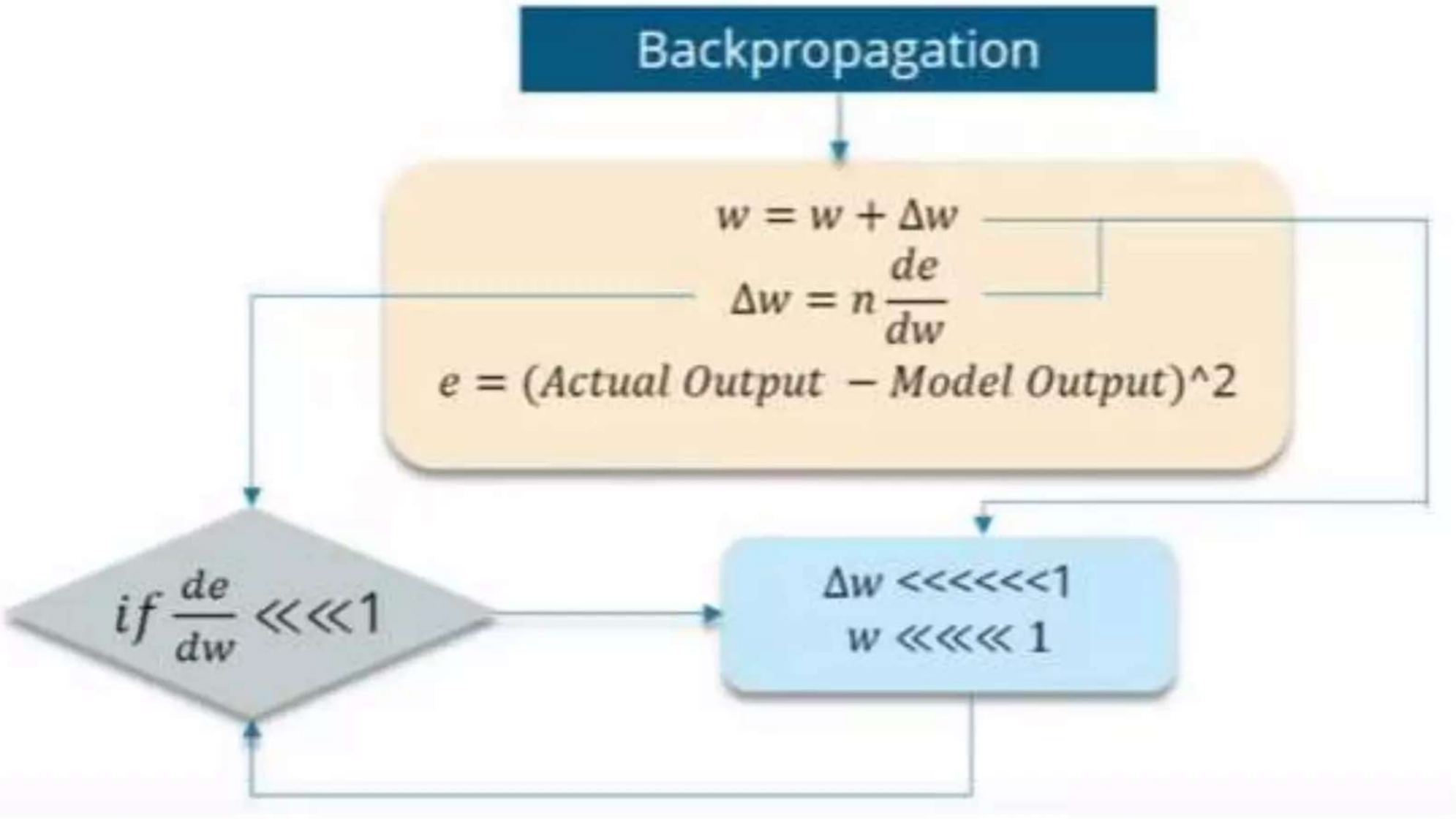
Why RNN's Fail –Vanishing or Exploding Gradient

"the clouds are in the ***sky***"



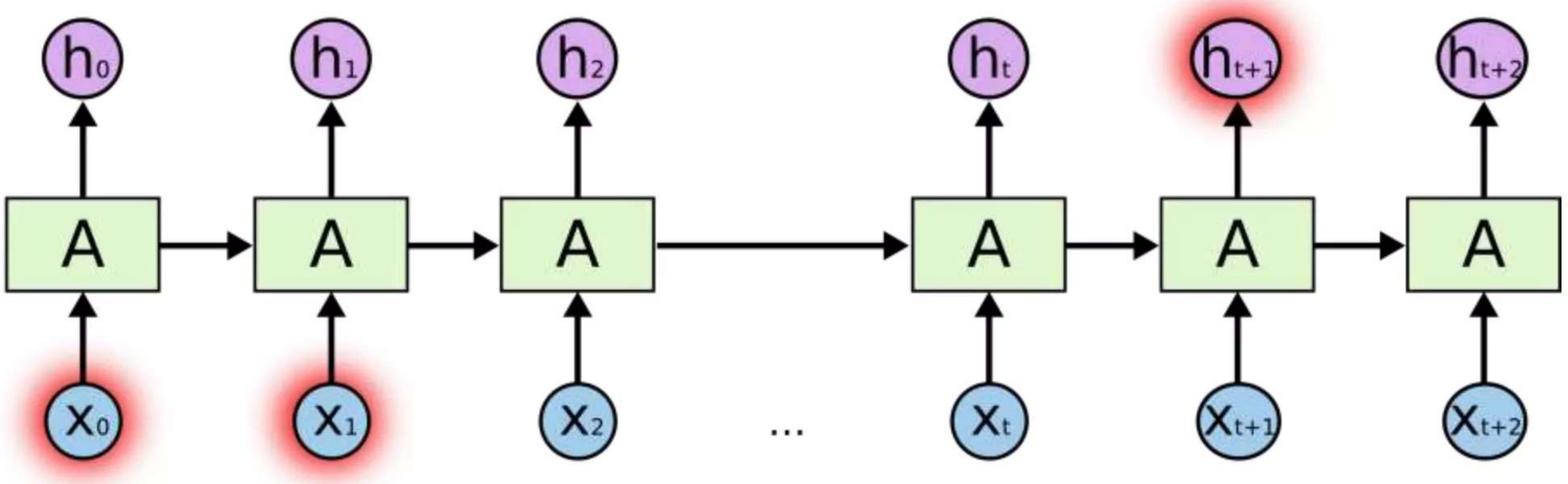
In such cases, where the gap between the relevant information and the place that it's needed is small, RNNs can learn to use the past information.

Why RNN's Fail –Vanishing or Exploding Gradient



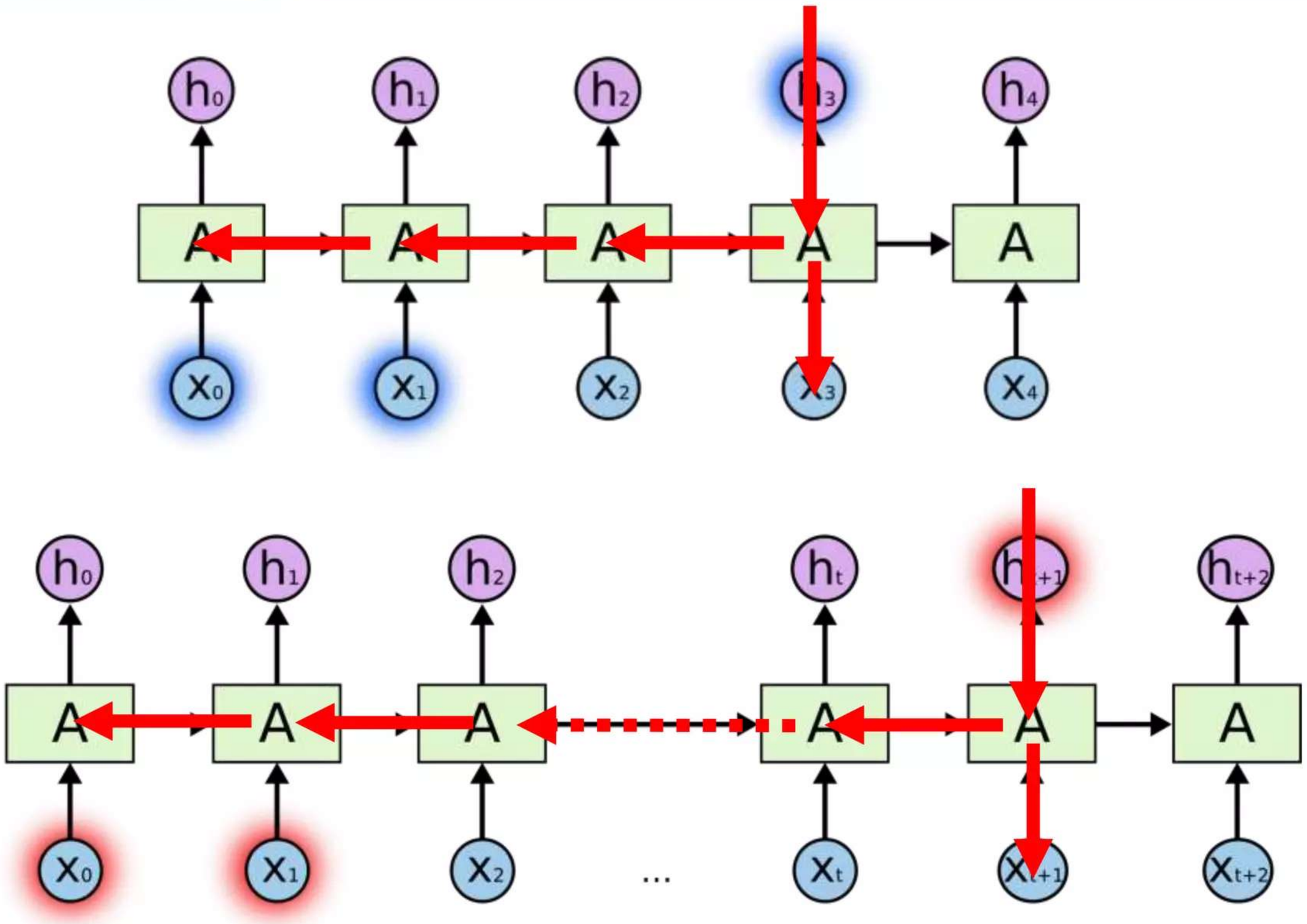
Why RNN's Fail –Vanishing or Exploding Gradient

"I grew up in France. I lived there for about 20 years. The people there are very nice. I can speak fluent **French**"



Unfortunately, as that gap grows, RNNs become unable to learn to connect the information.

Why RNN's Fail –Vanishing or Exploding Gradient

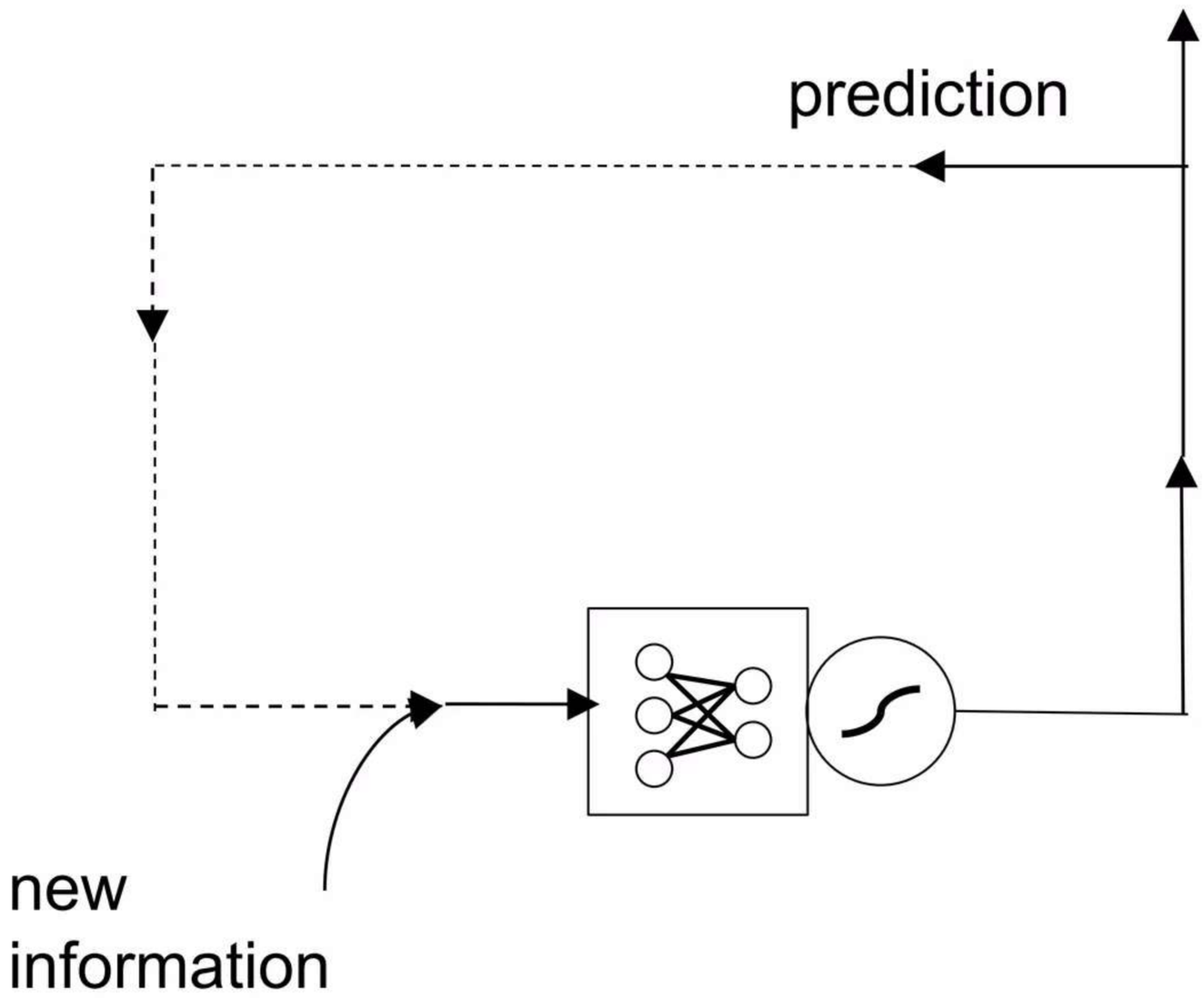


Agenda

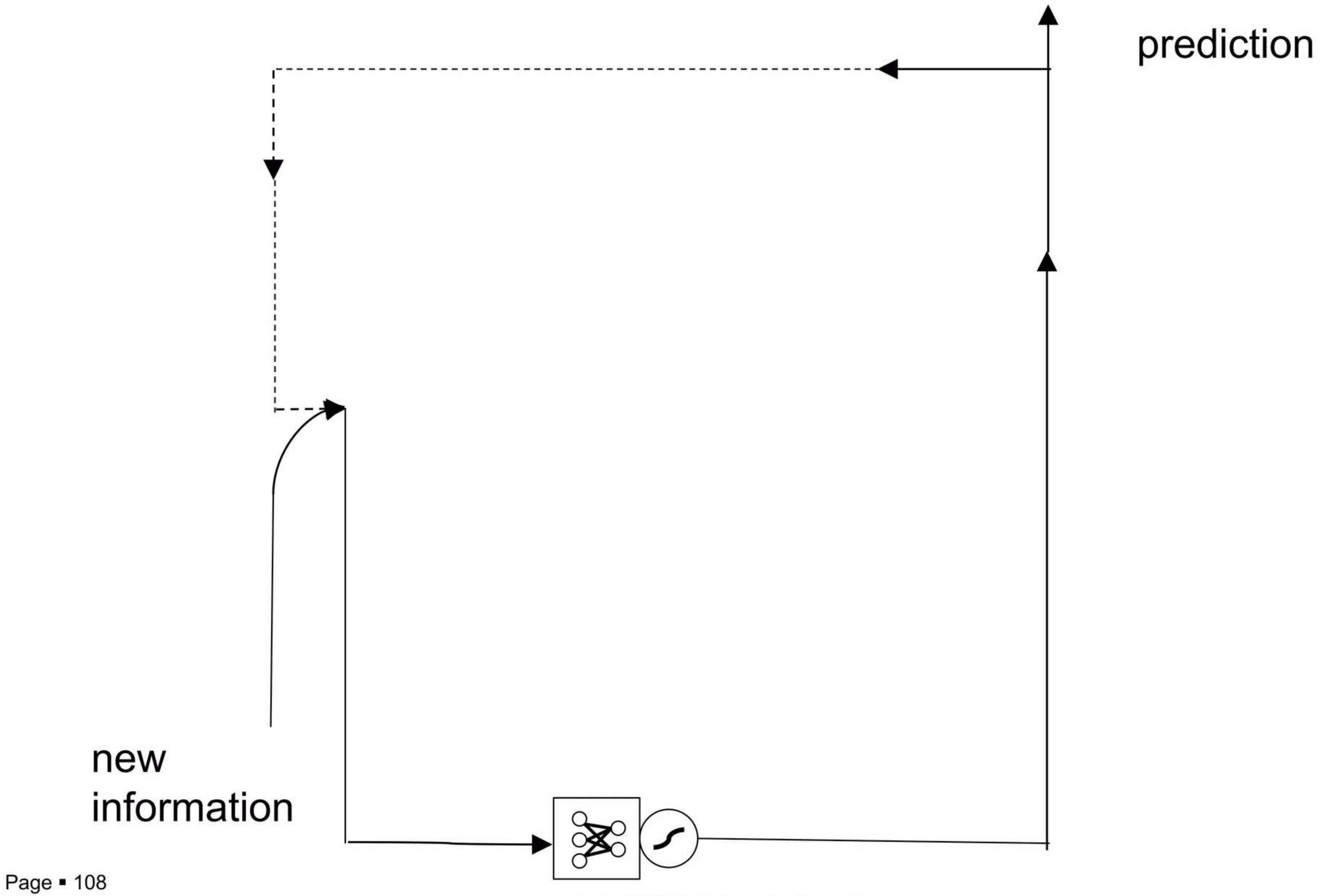
- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

Solution – Long Short Term Memory

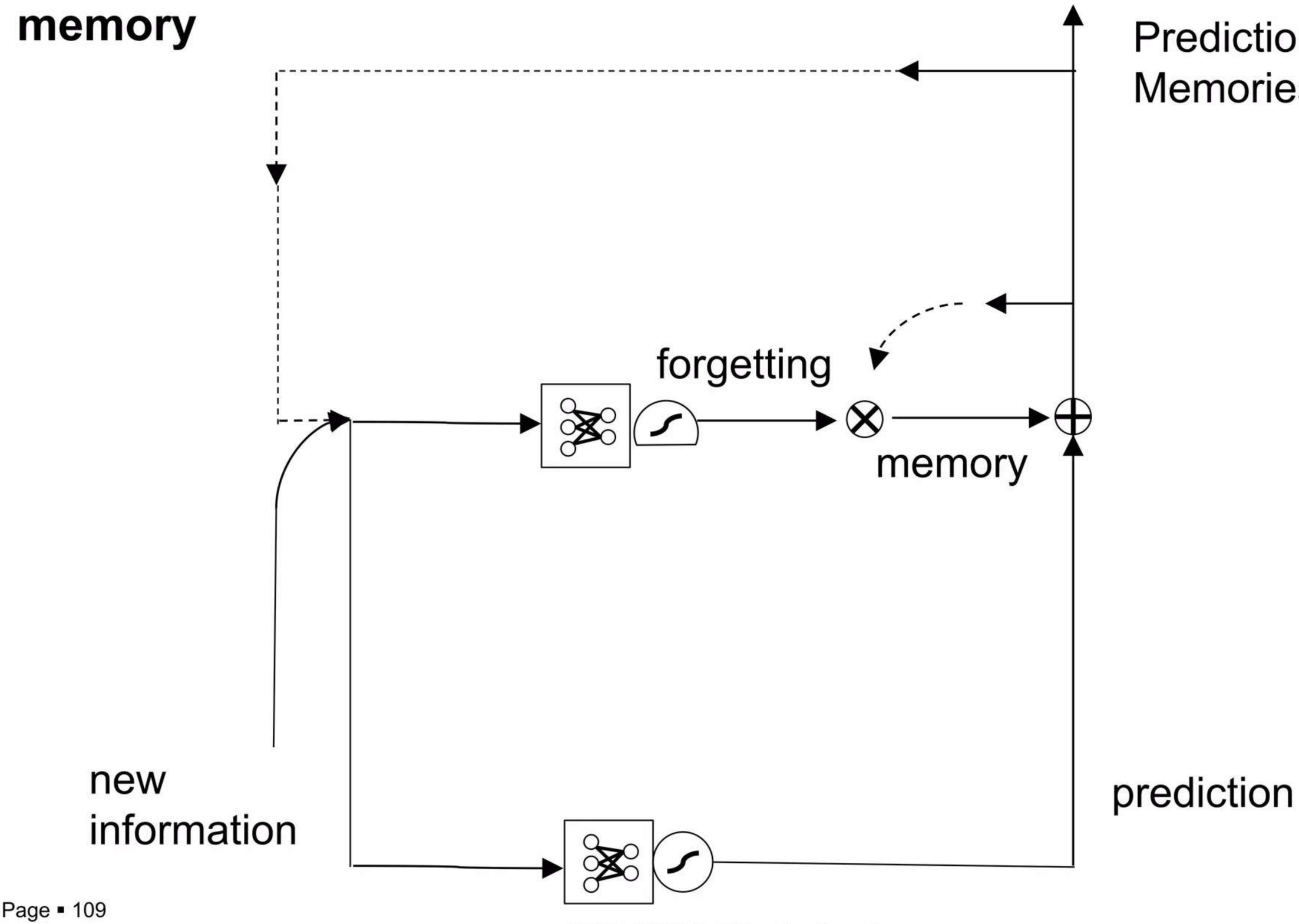
recurrent neural network



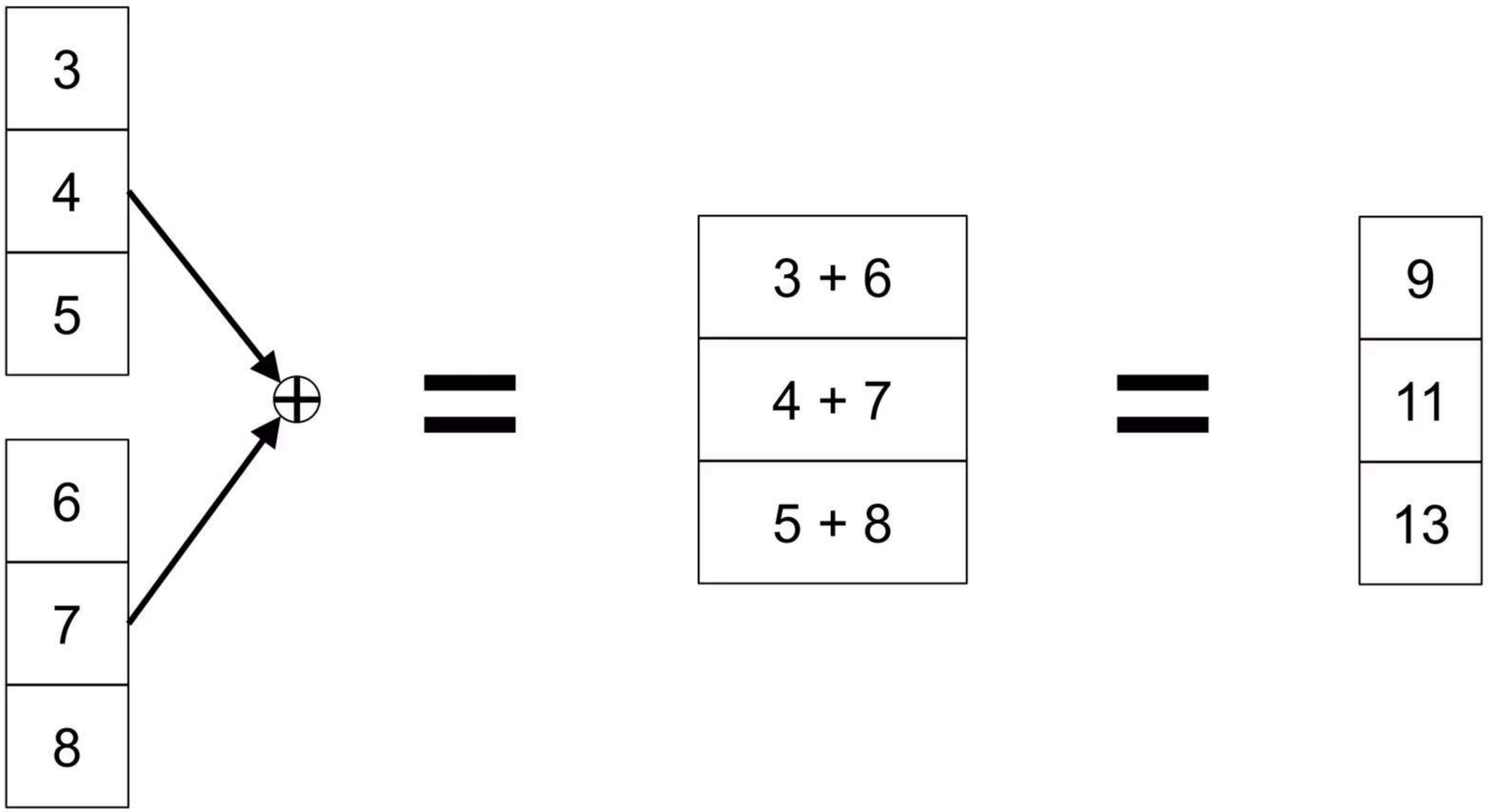
Solution – Long Short Term Memory



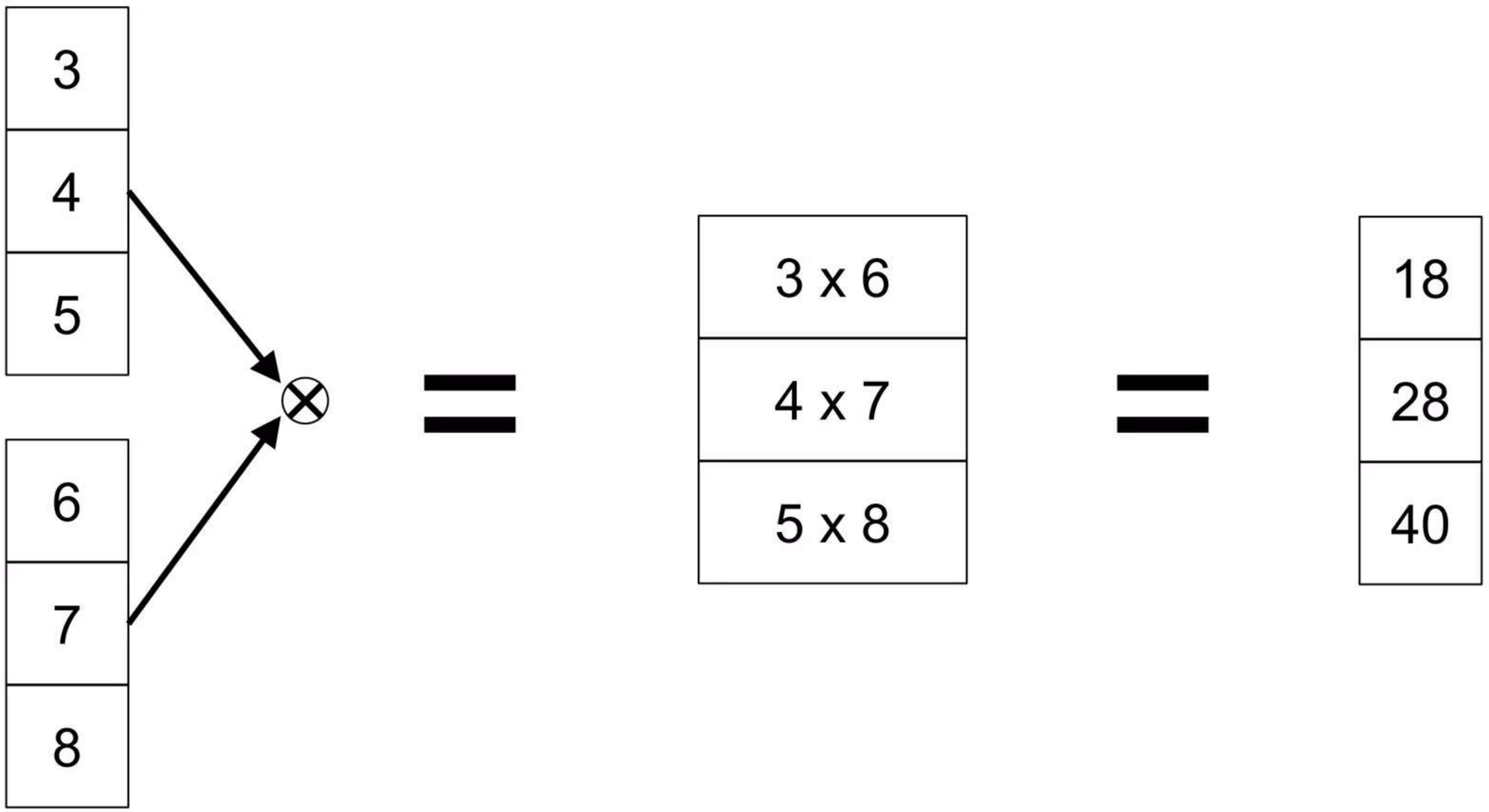
Forget Gate Layer



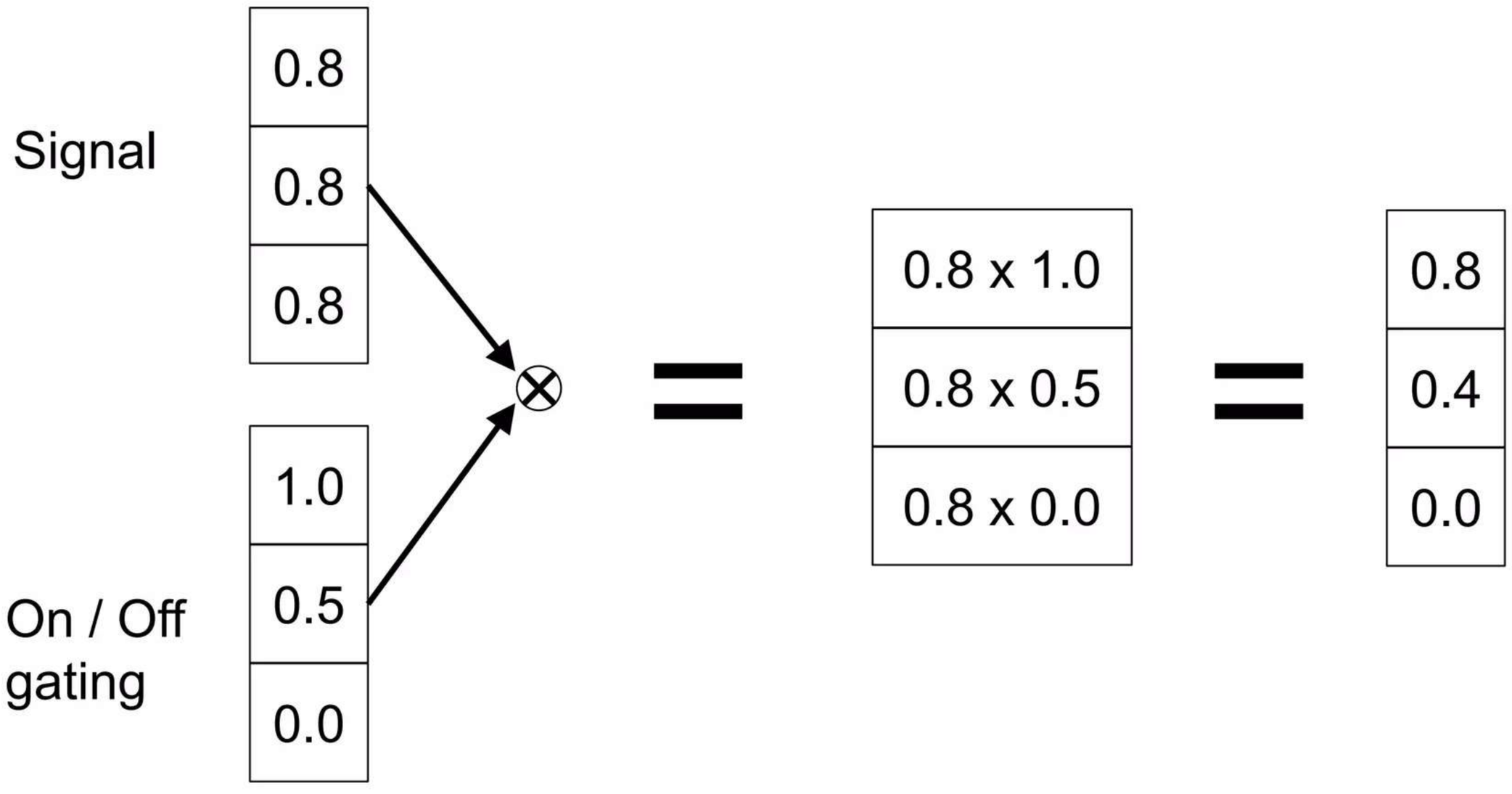
Plus junction: element-by-element addition



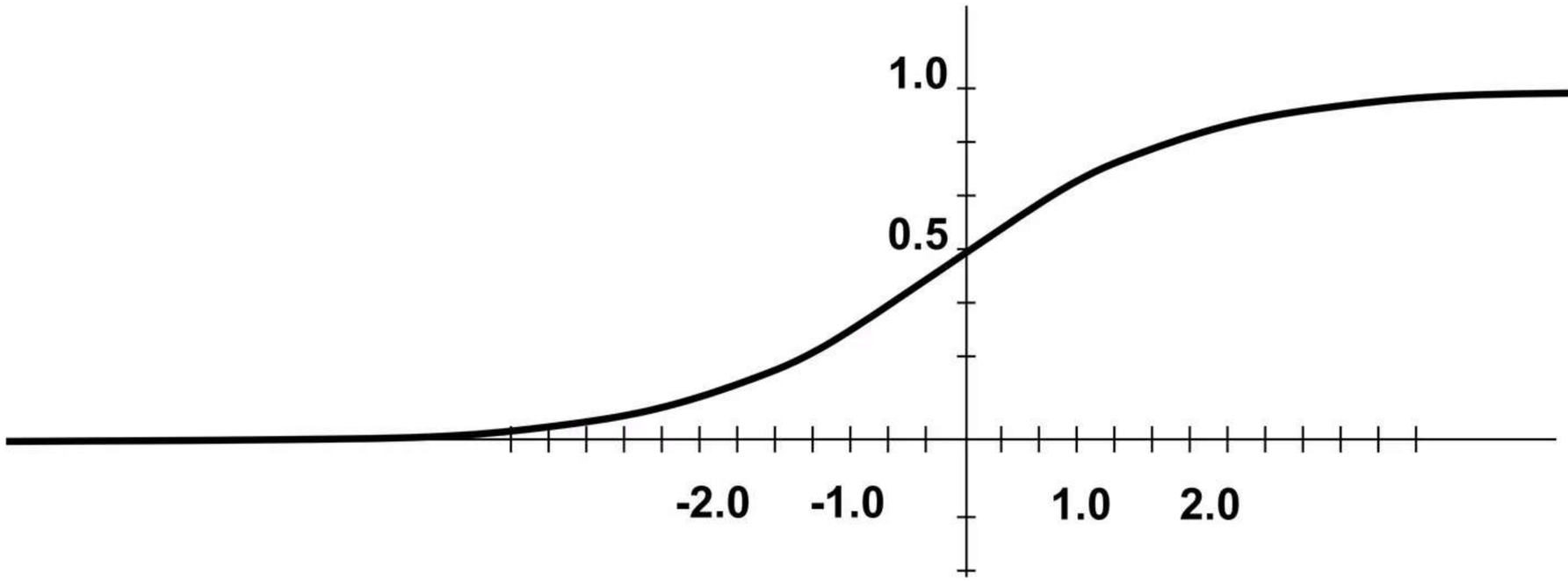
Times junction: element-by-element multiplication



Gating

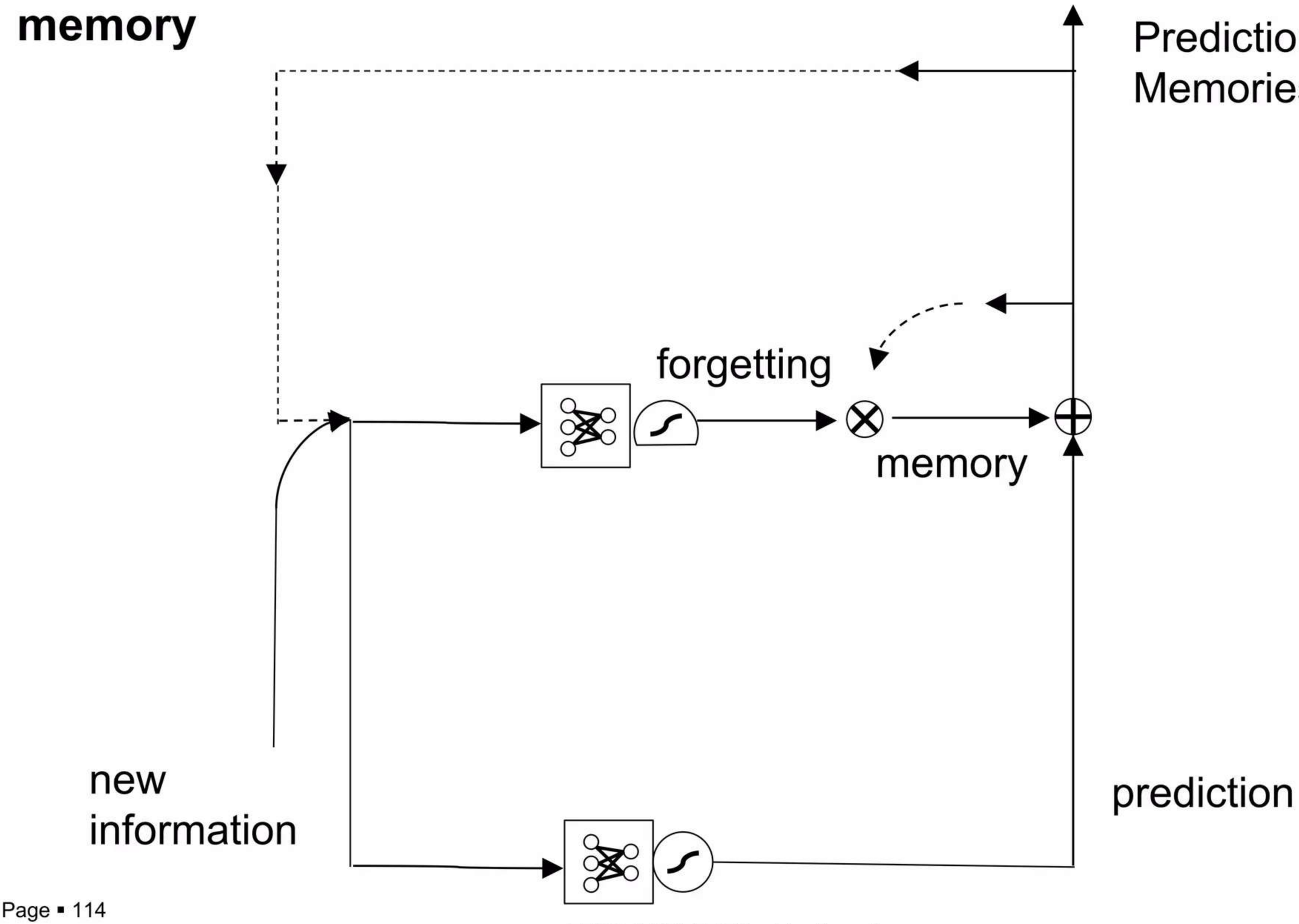

Signal
0.8
0.8
0.8
1.0
0.5
0.0
On / Off gating
0.8 x 1.0
0.8 x 0.5
0.8 x 0.0
0.8
0.4
0.0

Logistic (sigmoid) squashing function

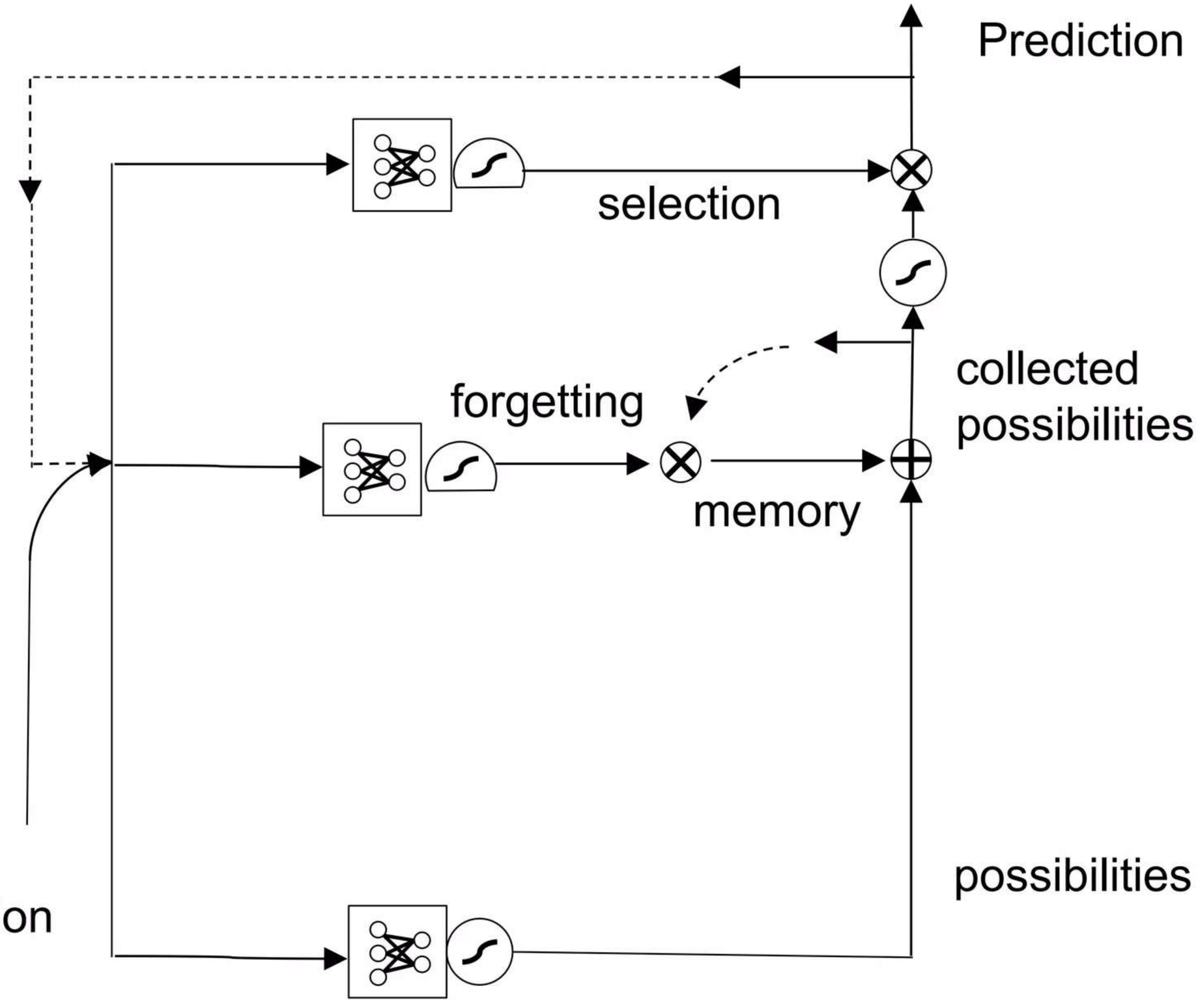


No matter what you start with, the answer stays between 0 and 1.

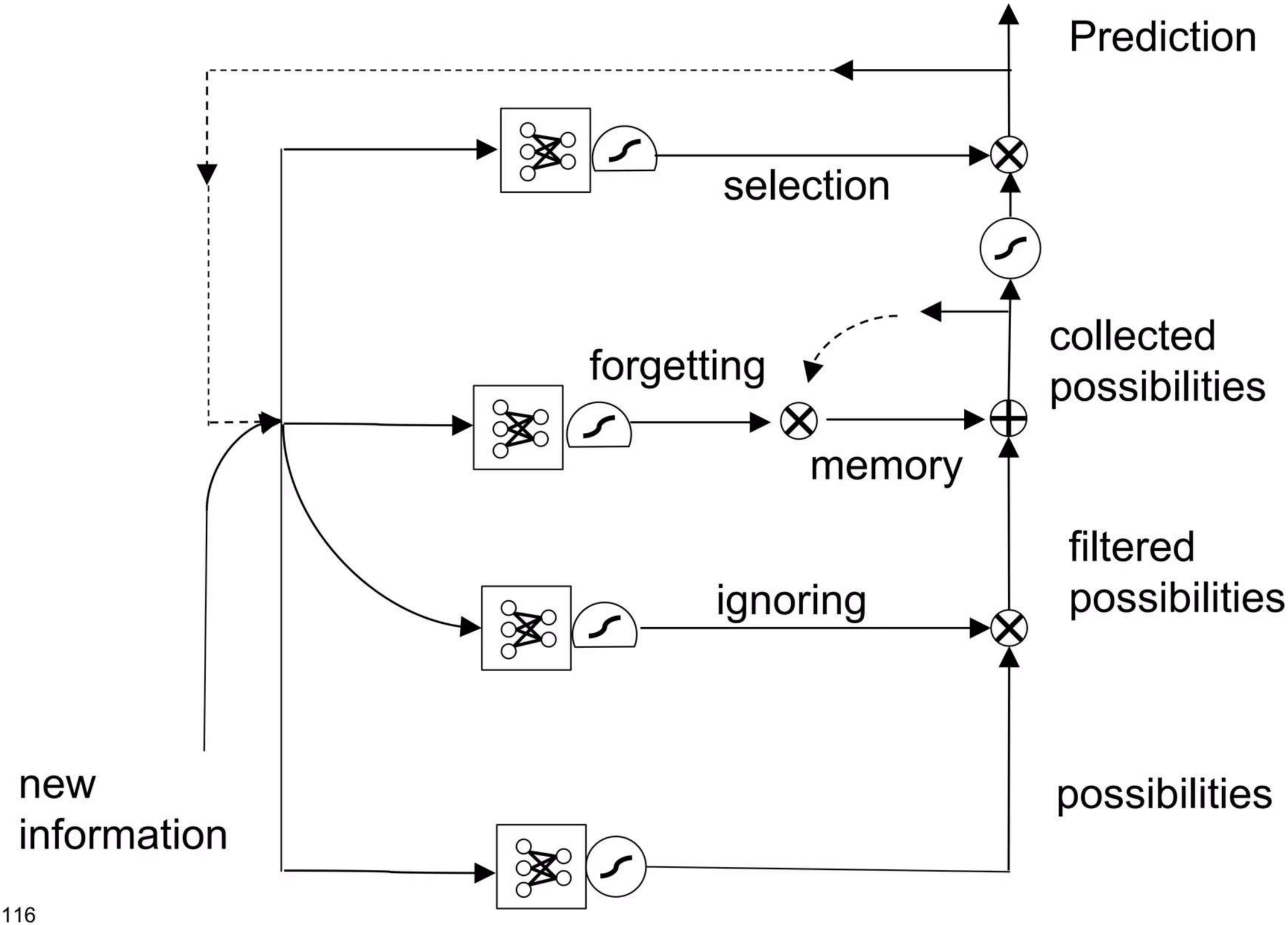
Forget Gate Layer



Long Short Term Memory



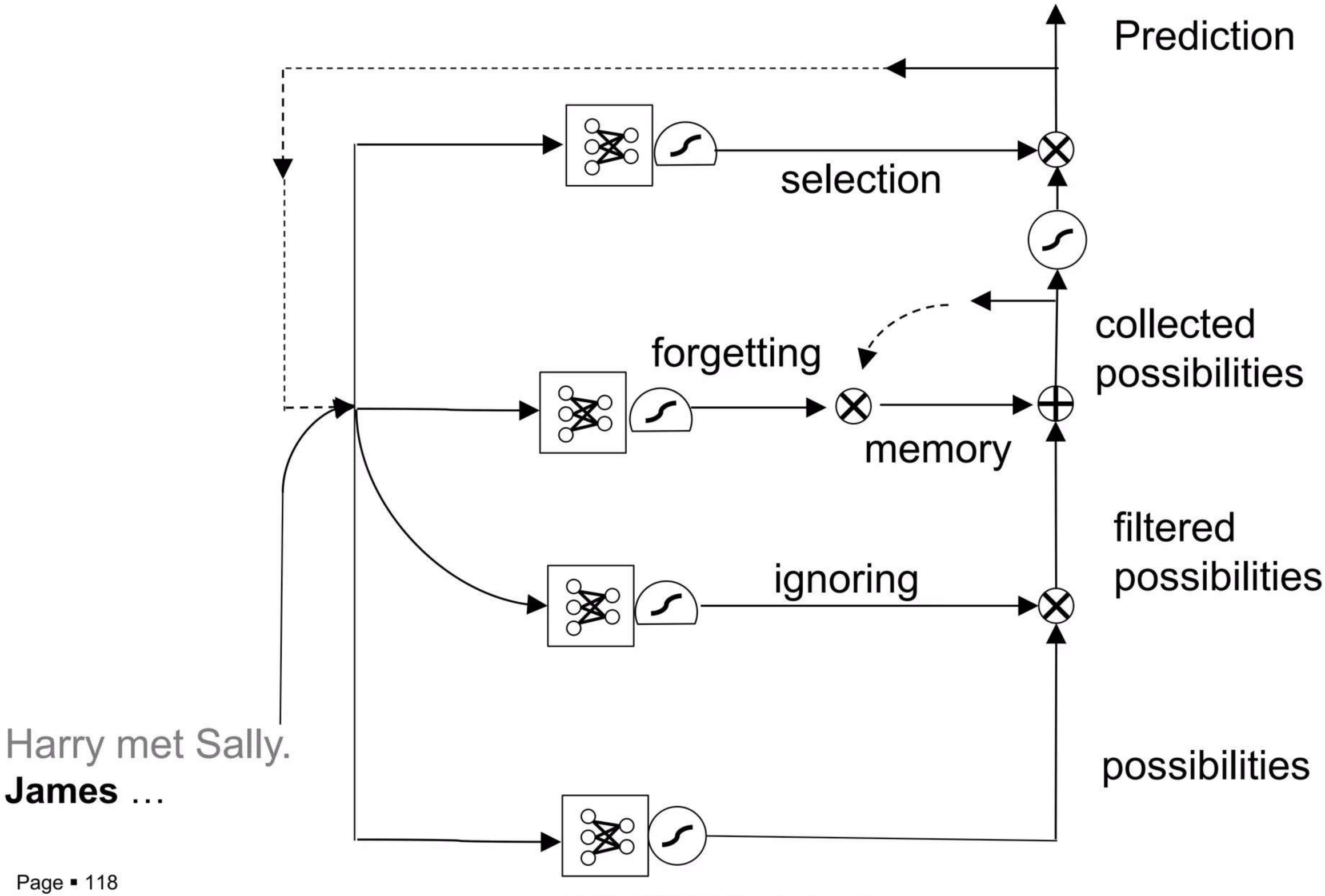
Long Short Term Memory



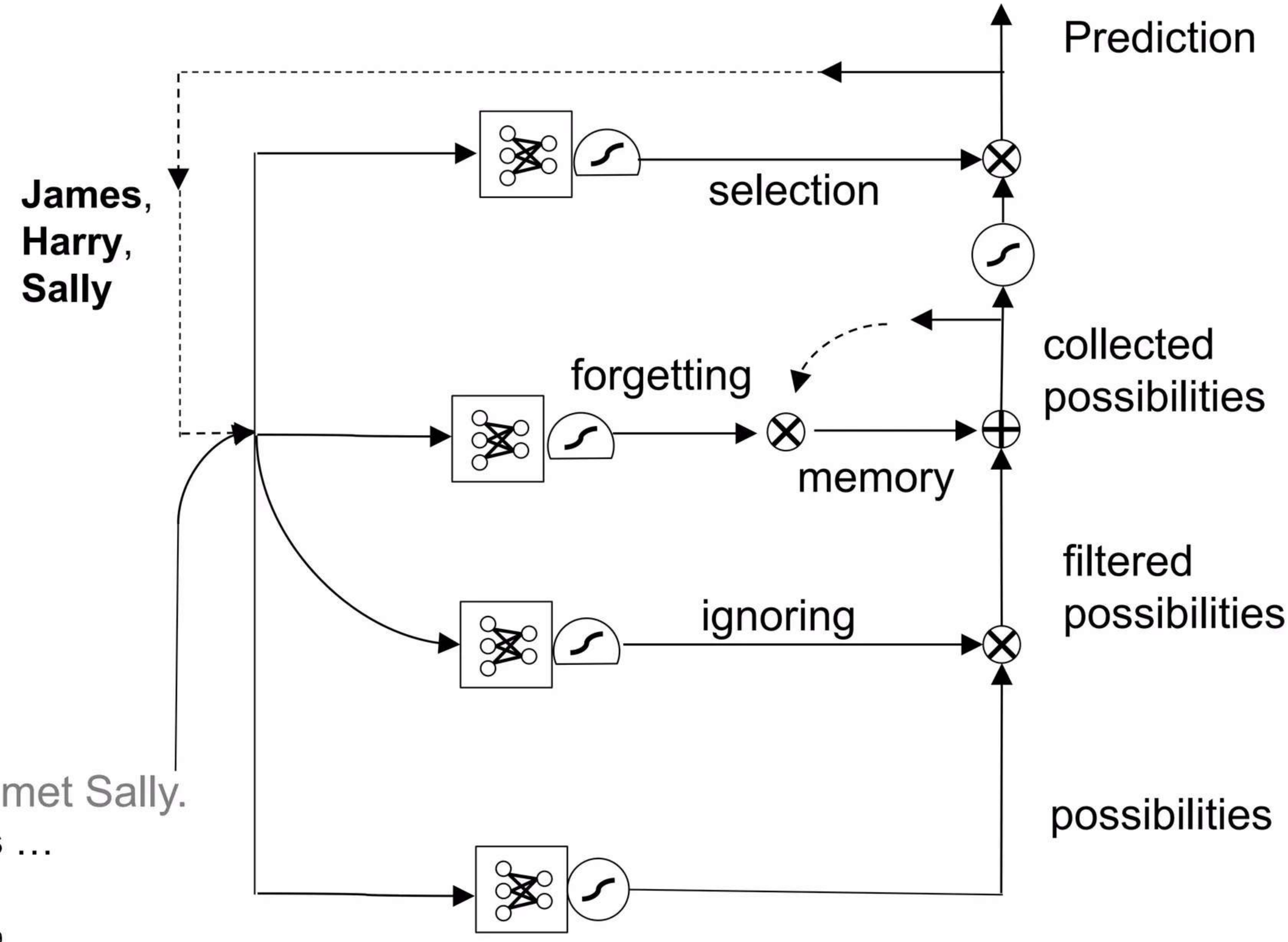
Agenda

- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

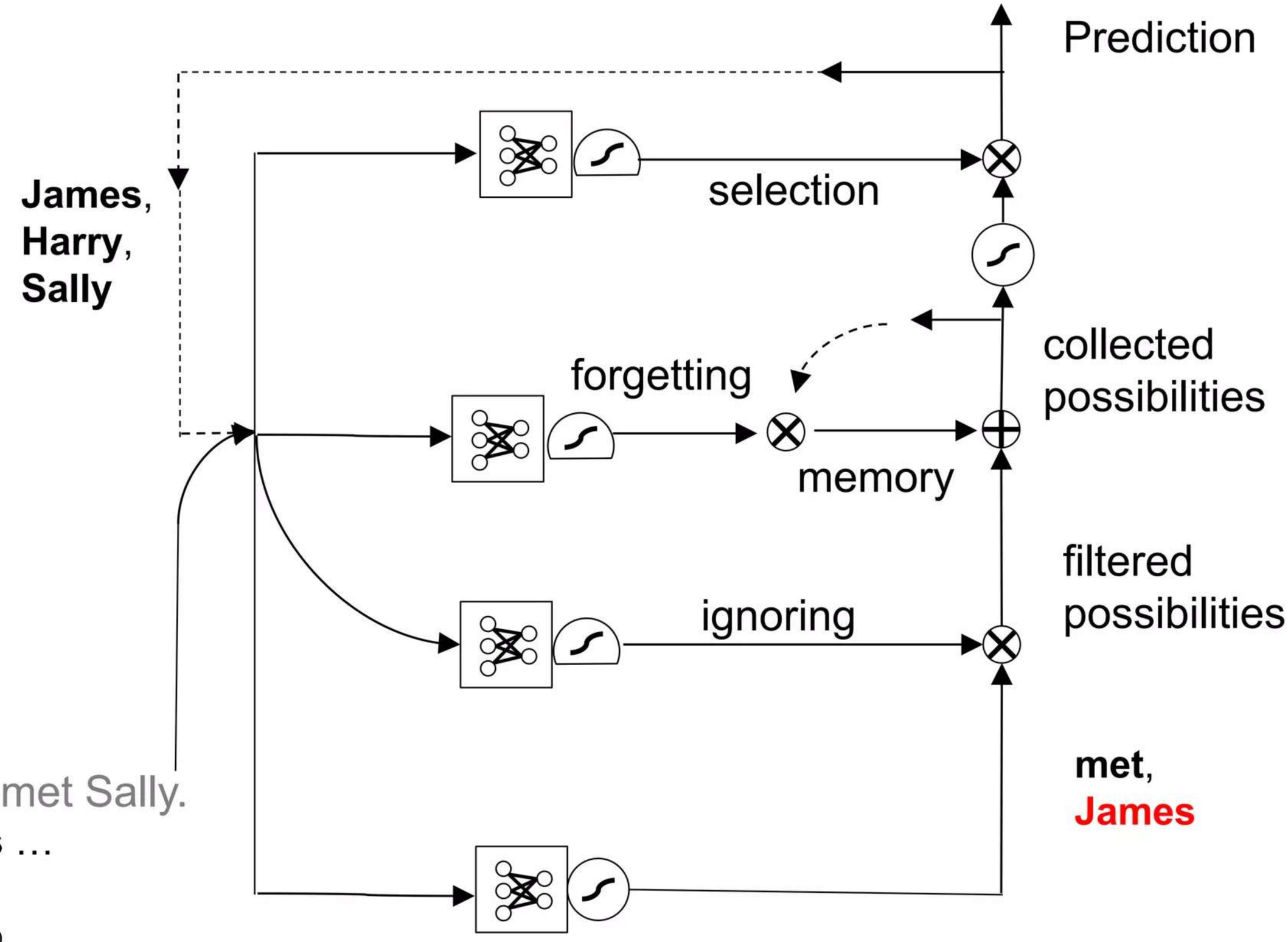
LSTM in Action



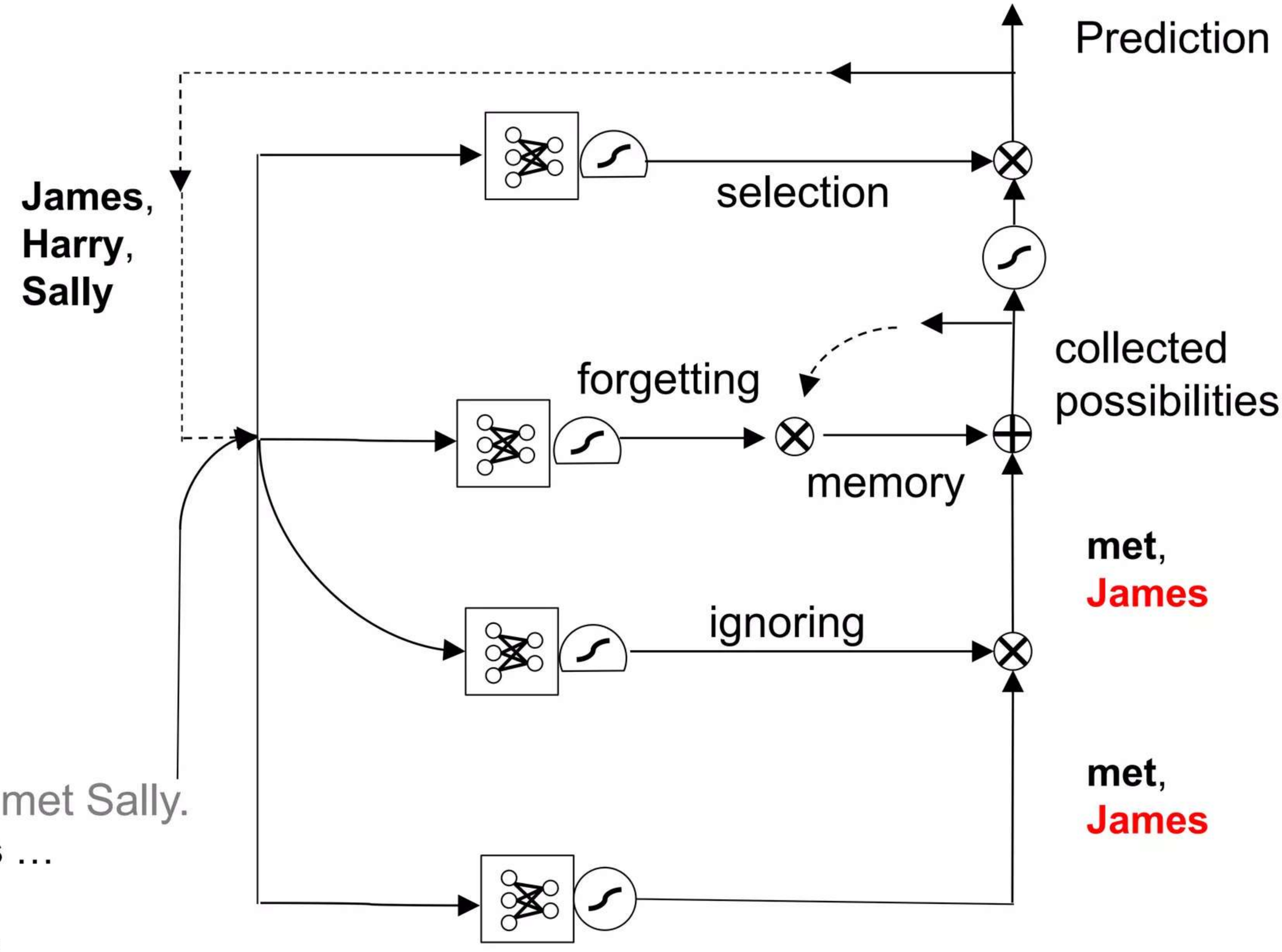
LSTM in Action



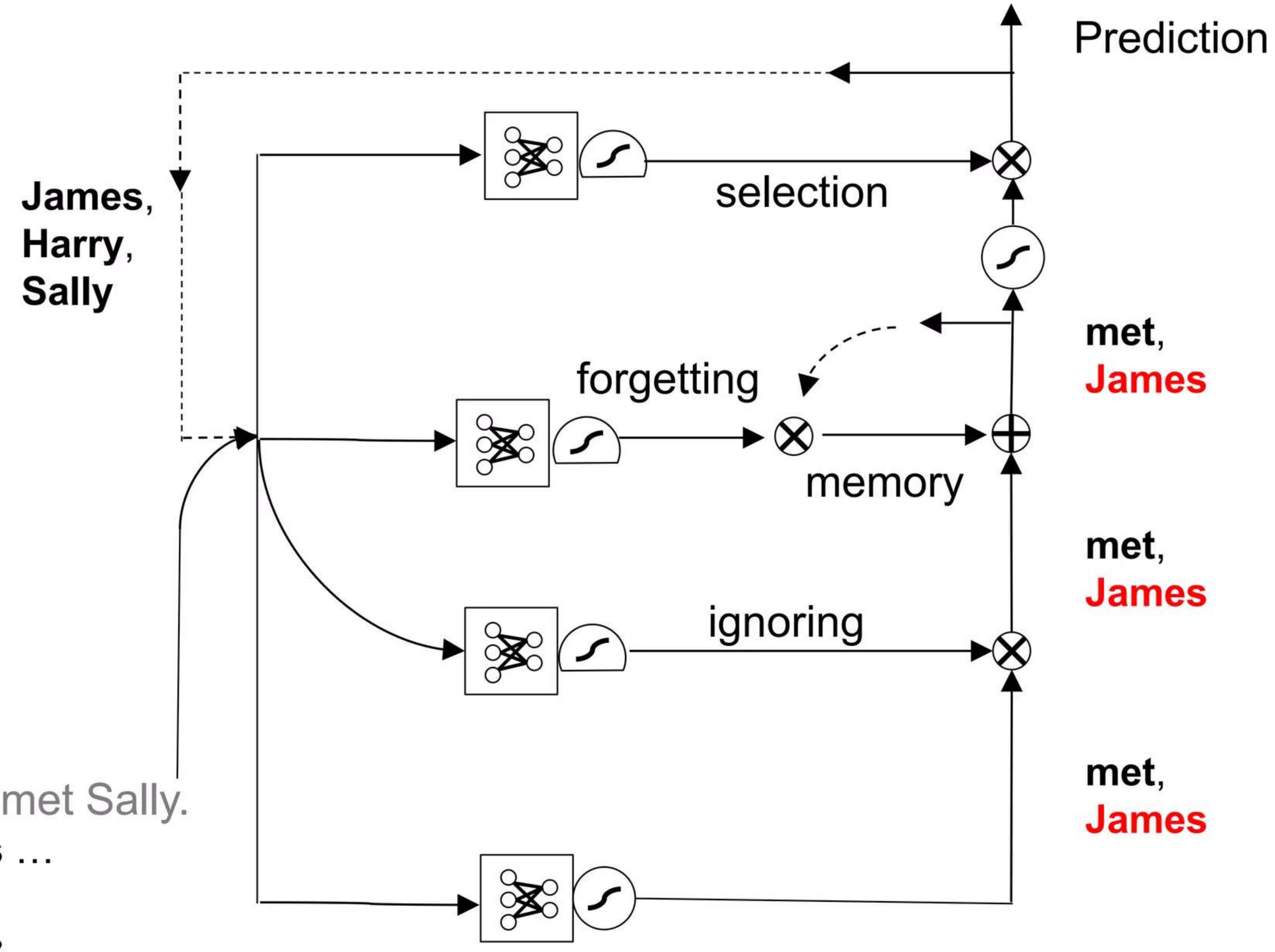
LSTM in Action



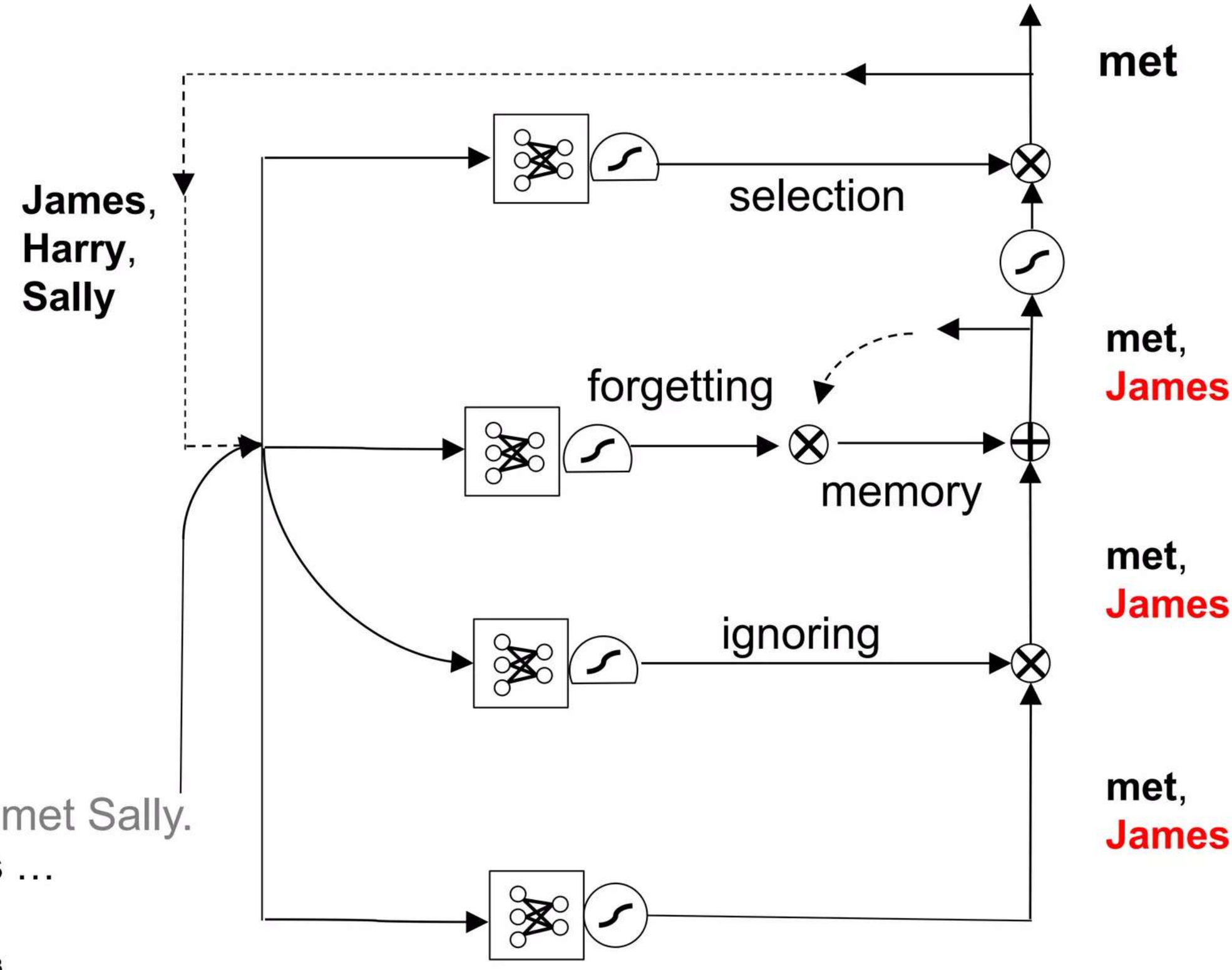
LSTM in Action



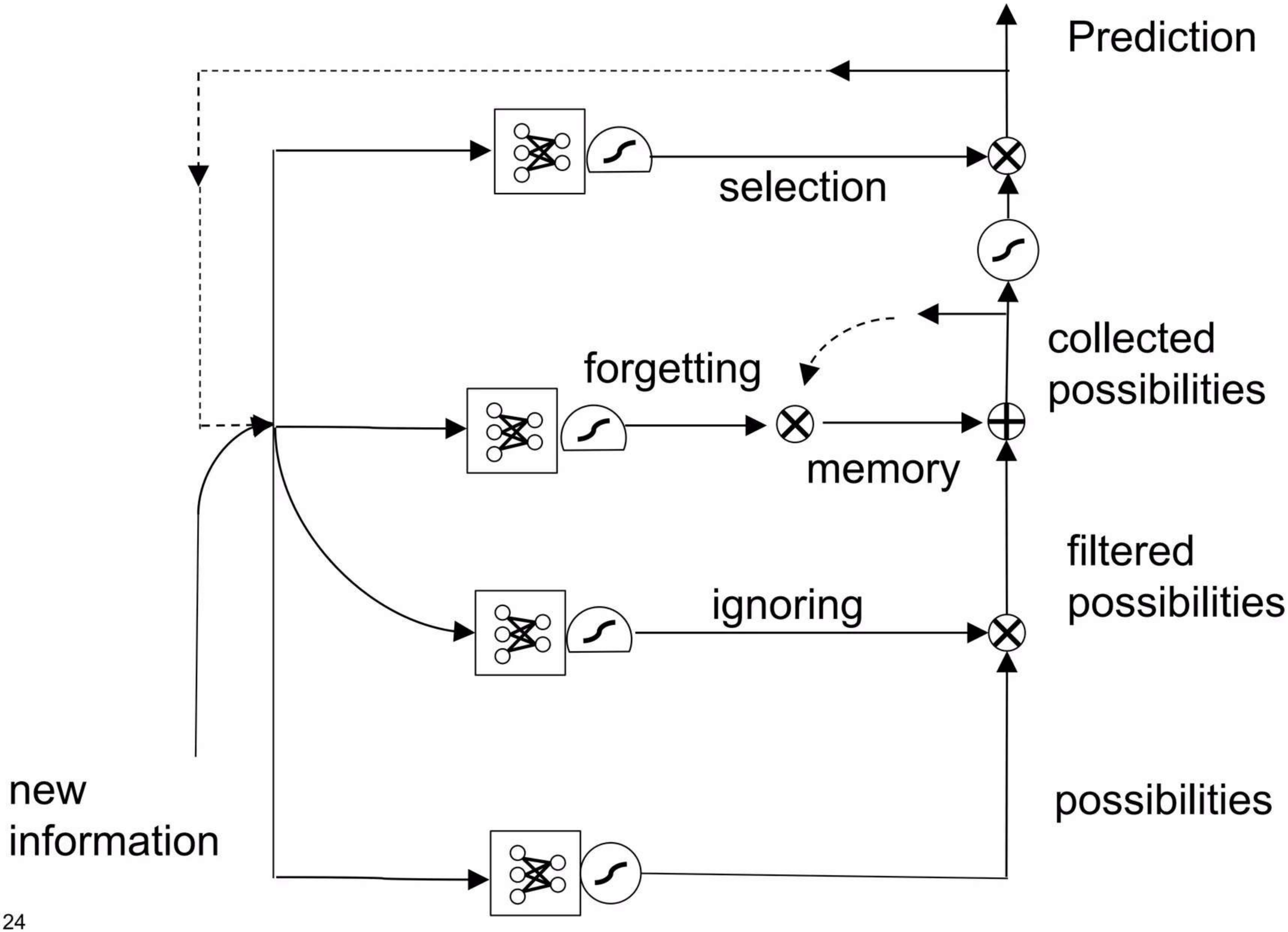
LSTM in Action



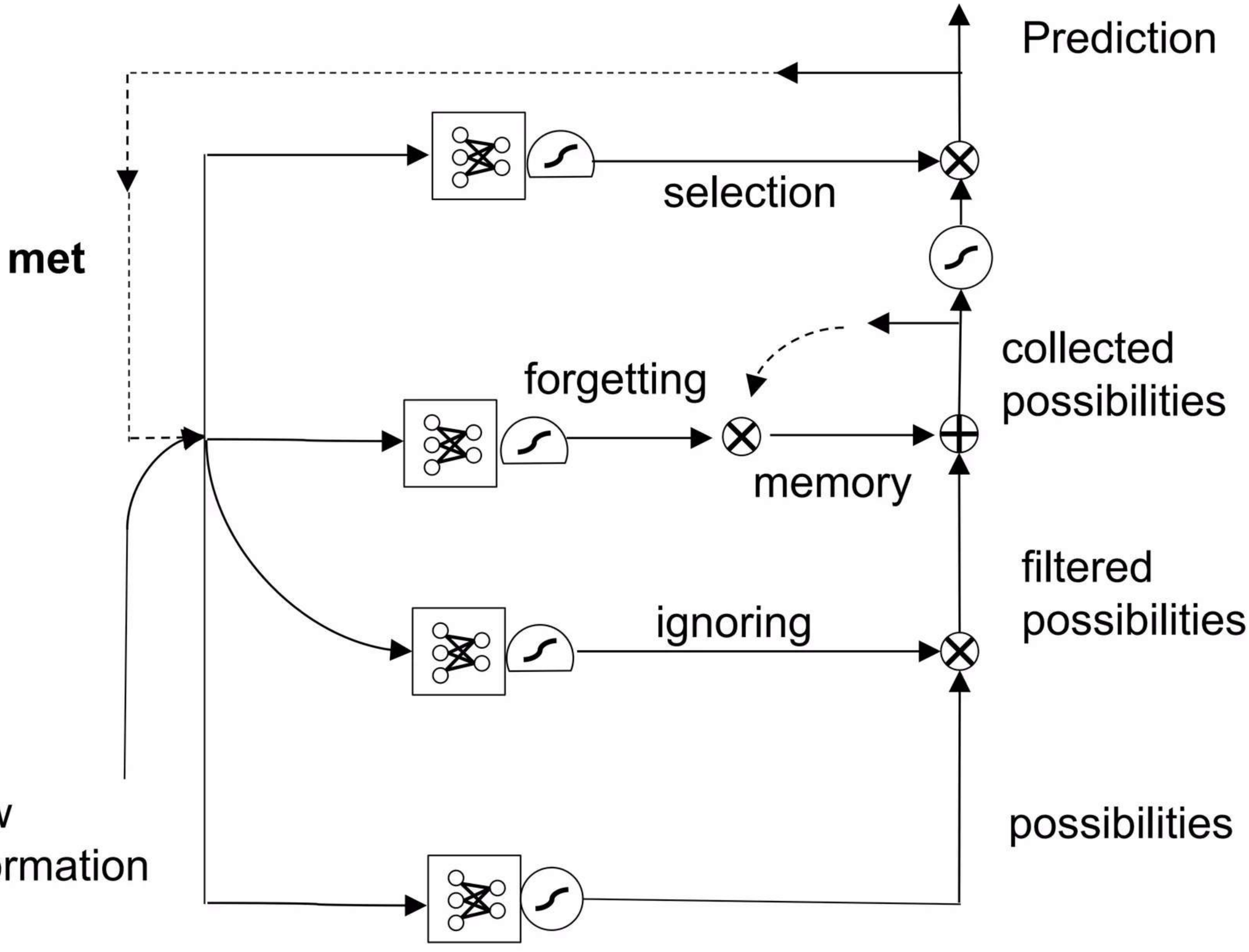
LSTM in Action



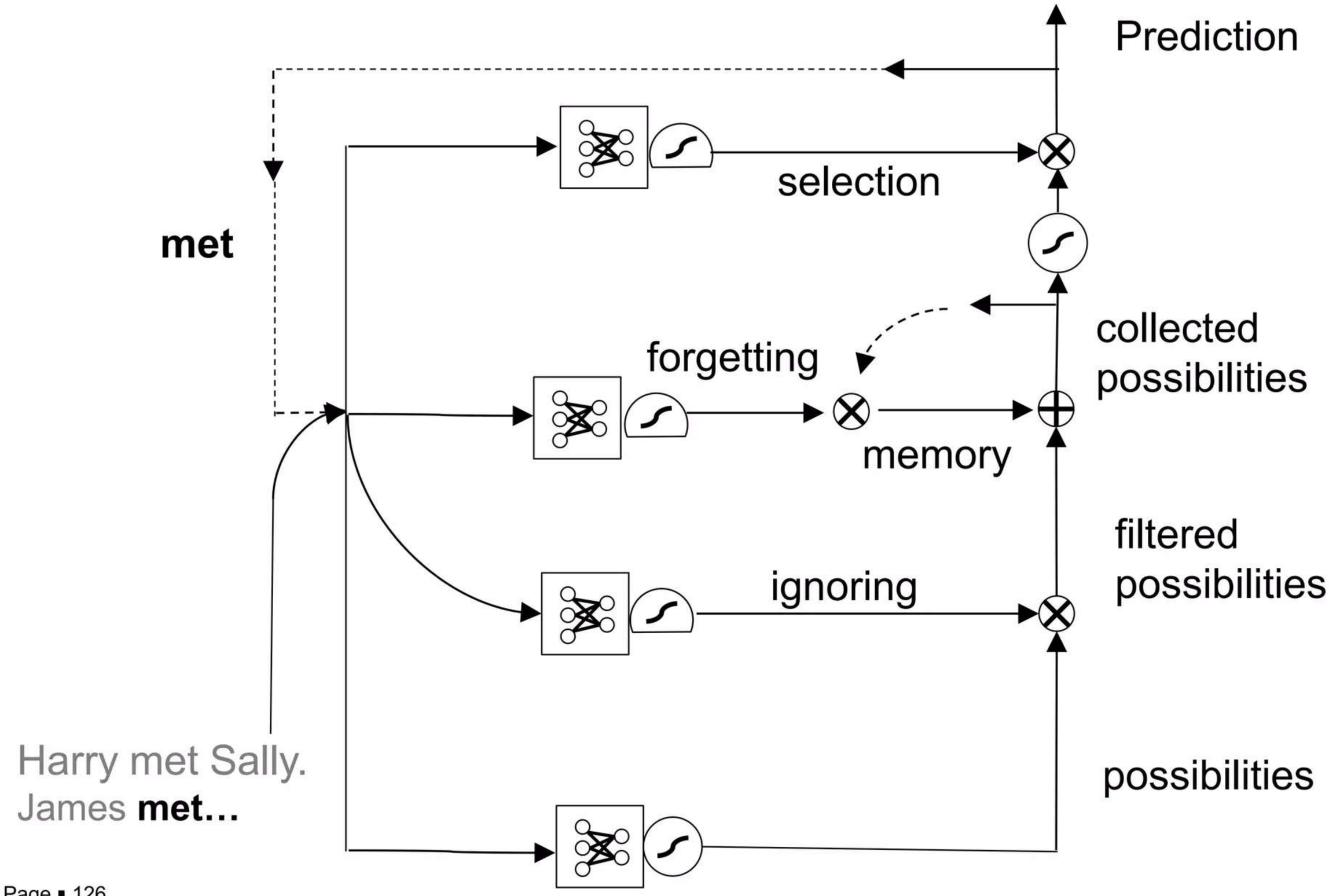
LSTM in Action



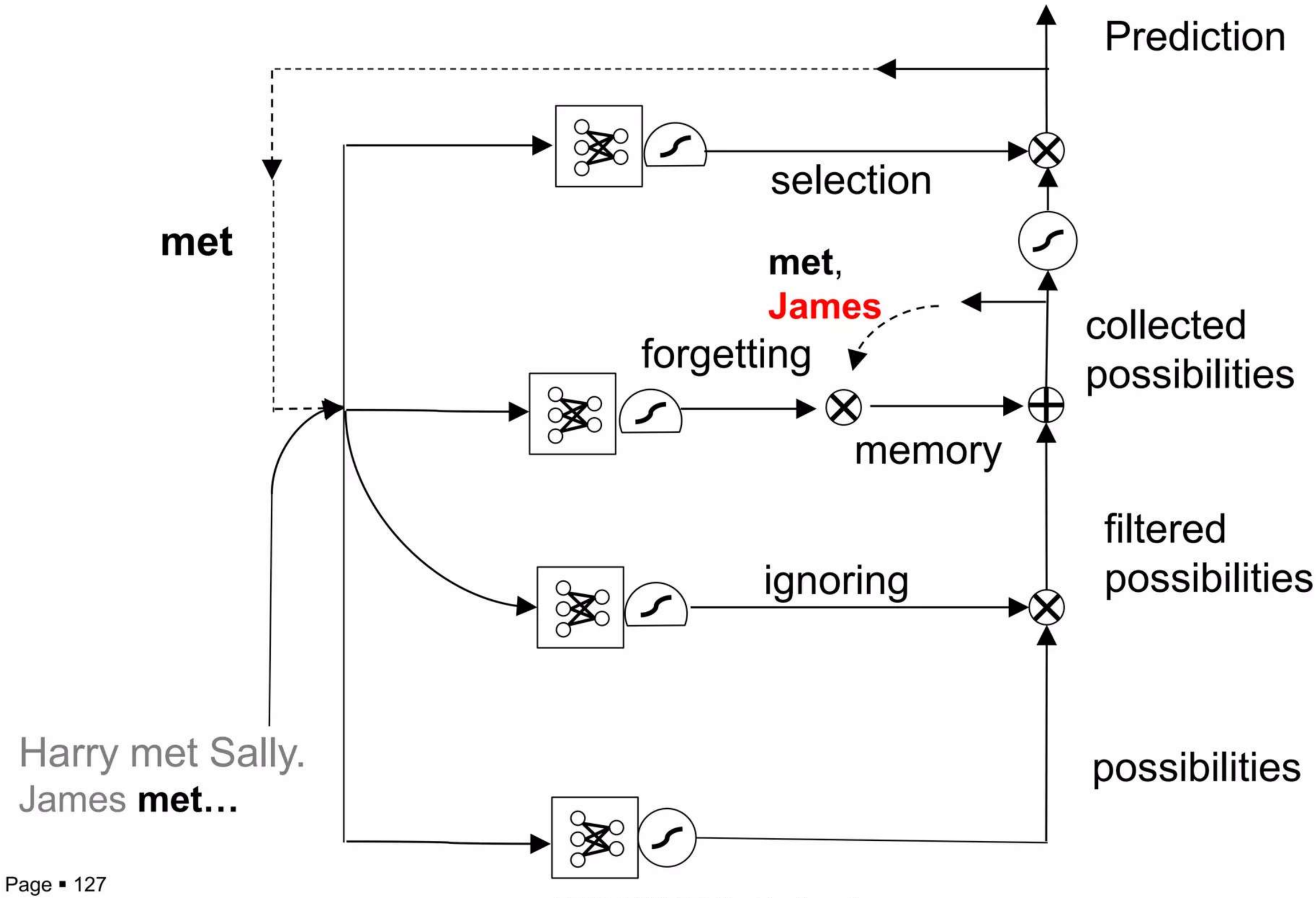
LSTM in Action



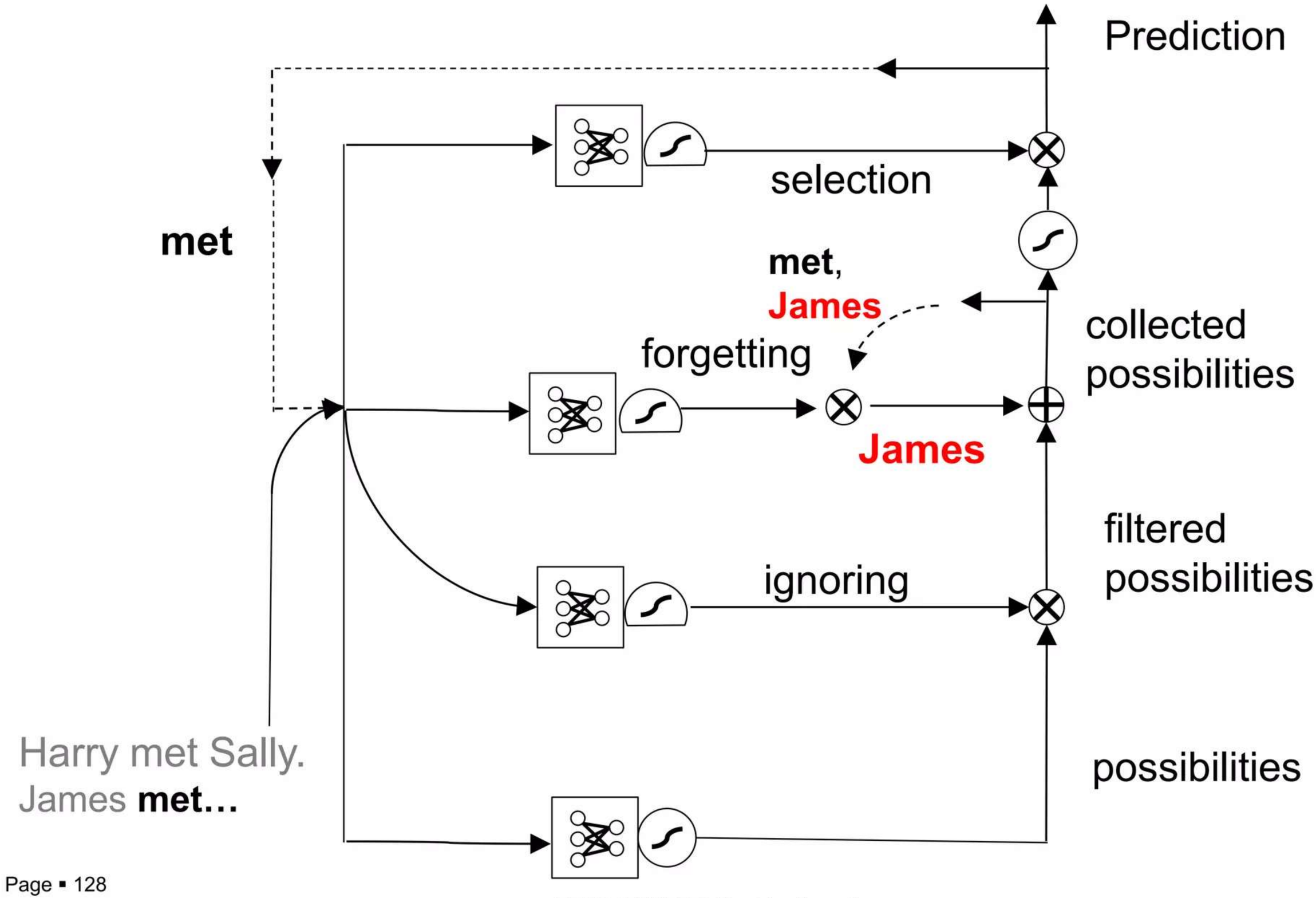
LSTM in Action



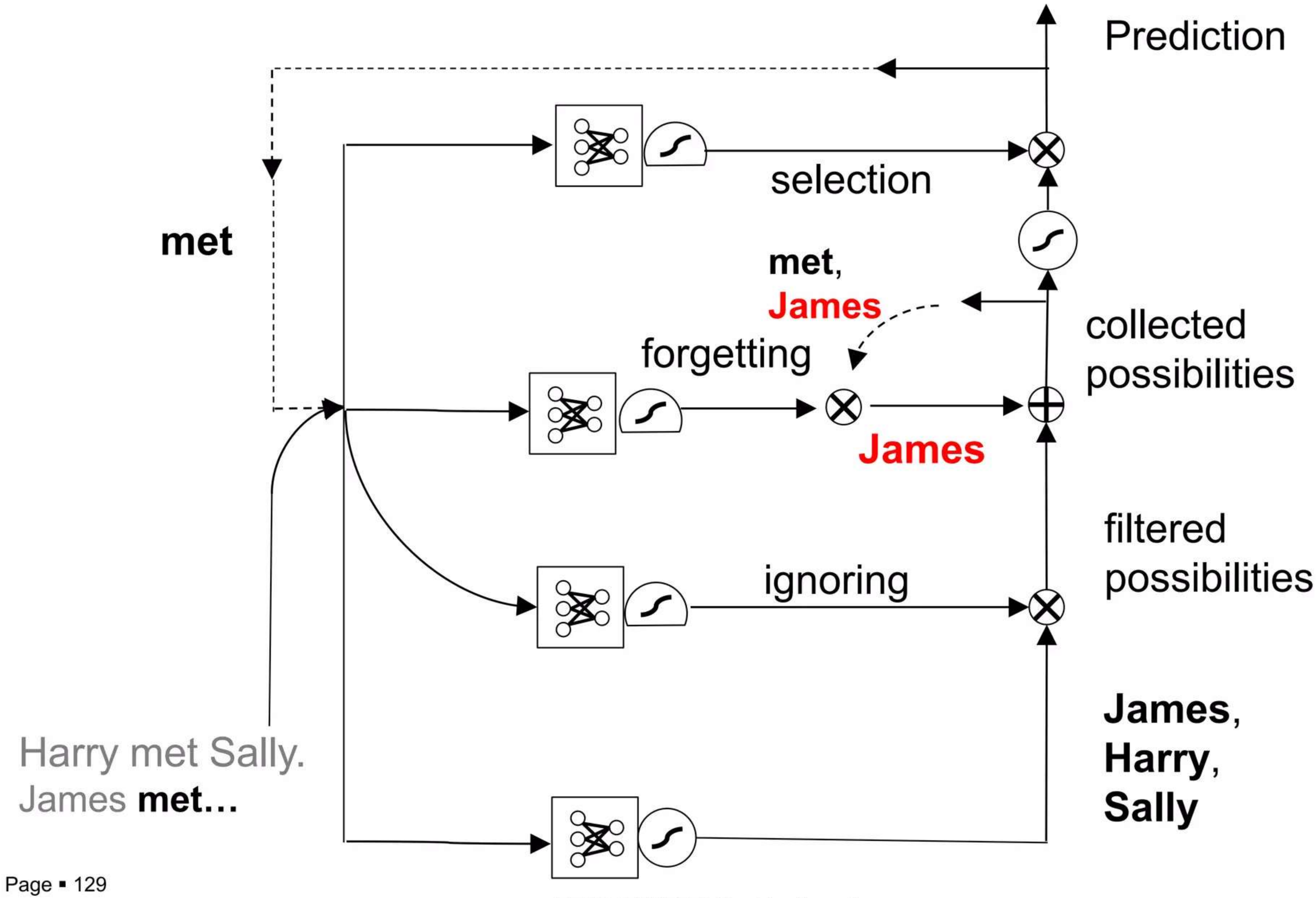
LSTM in Action



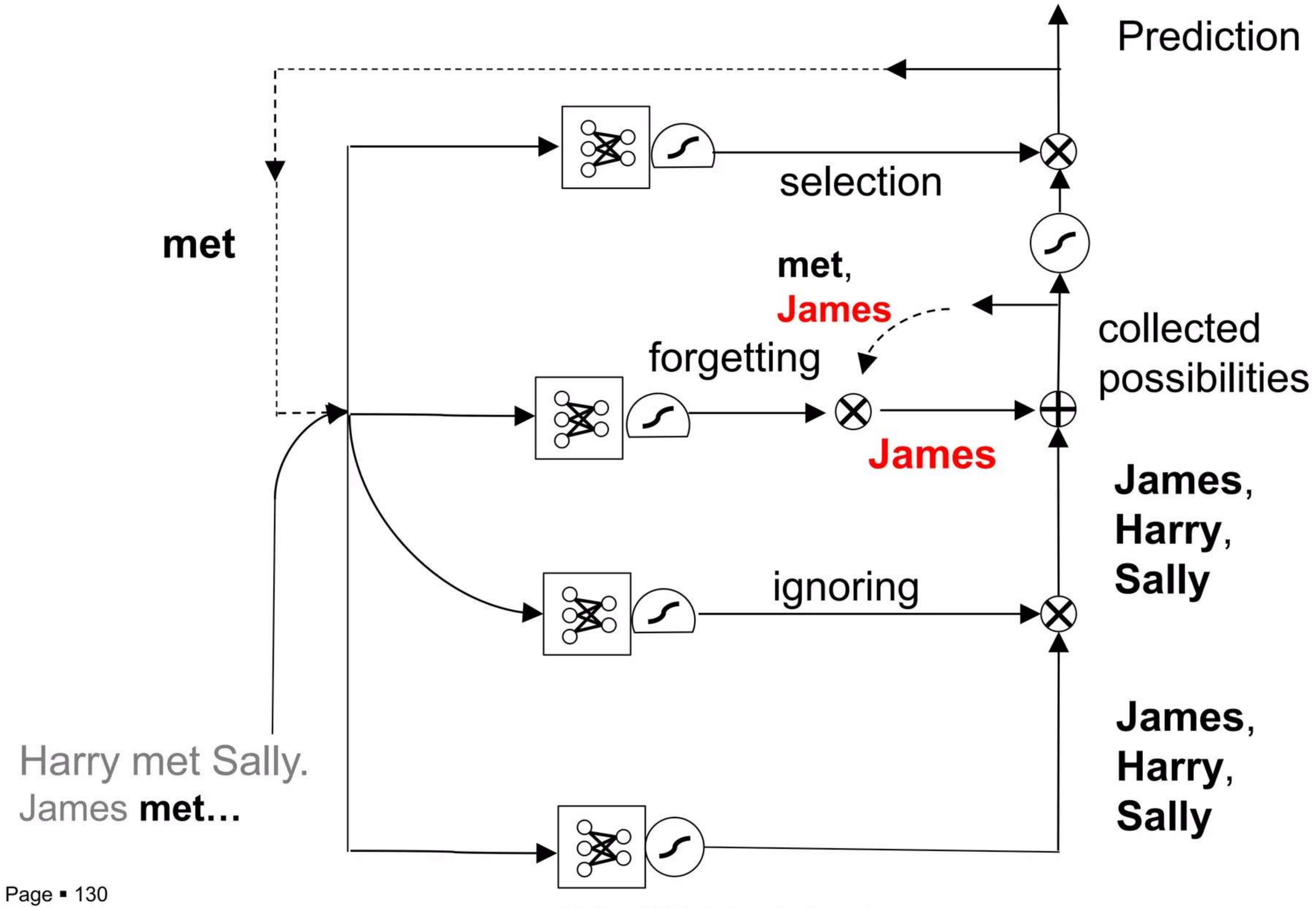
LSTM in Action



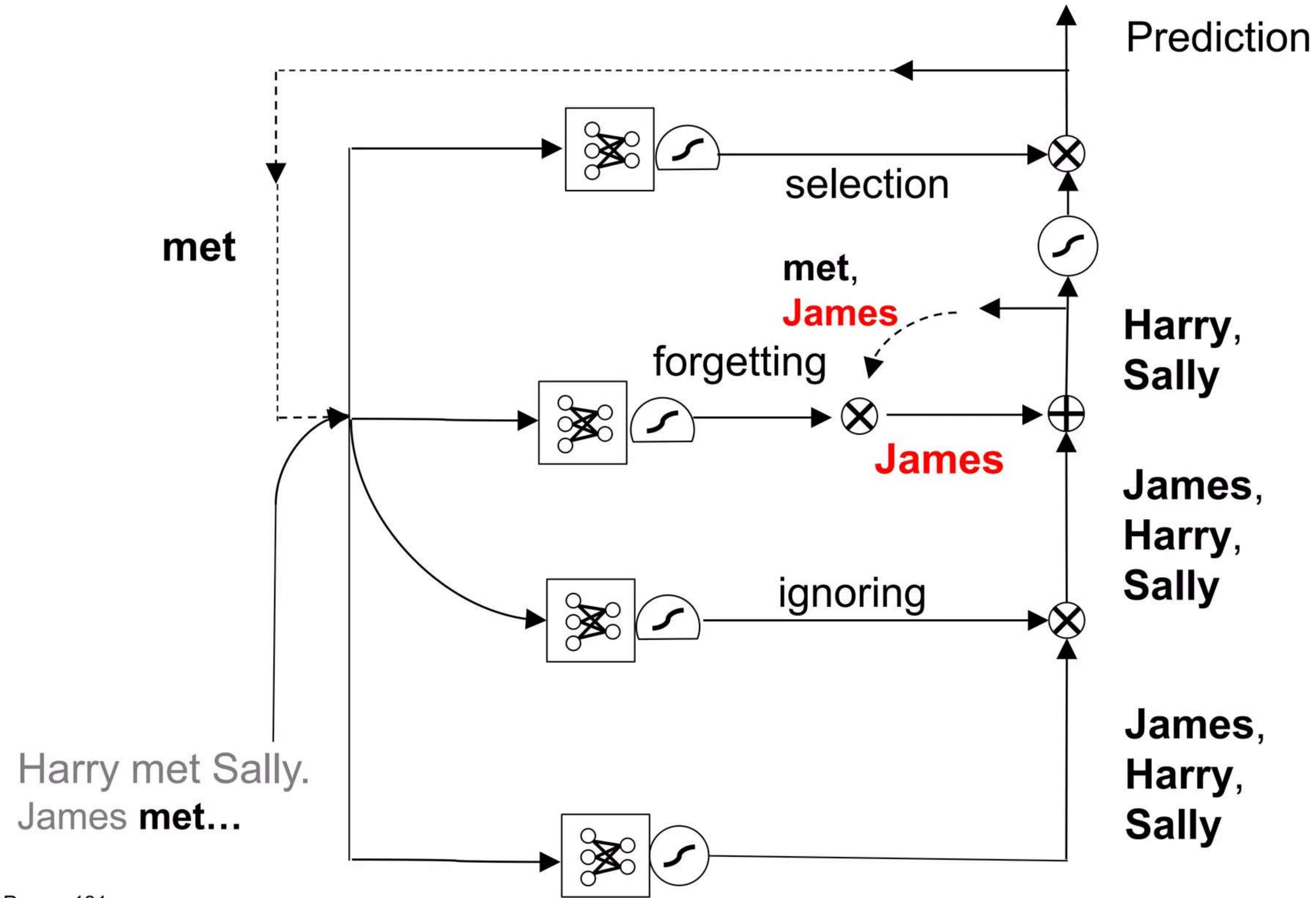
LSTM in Action



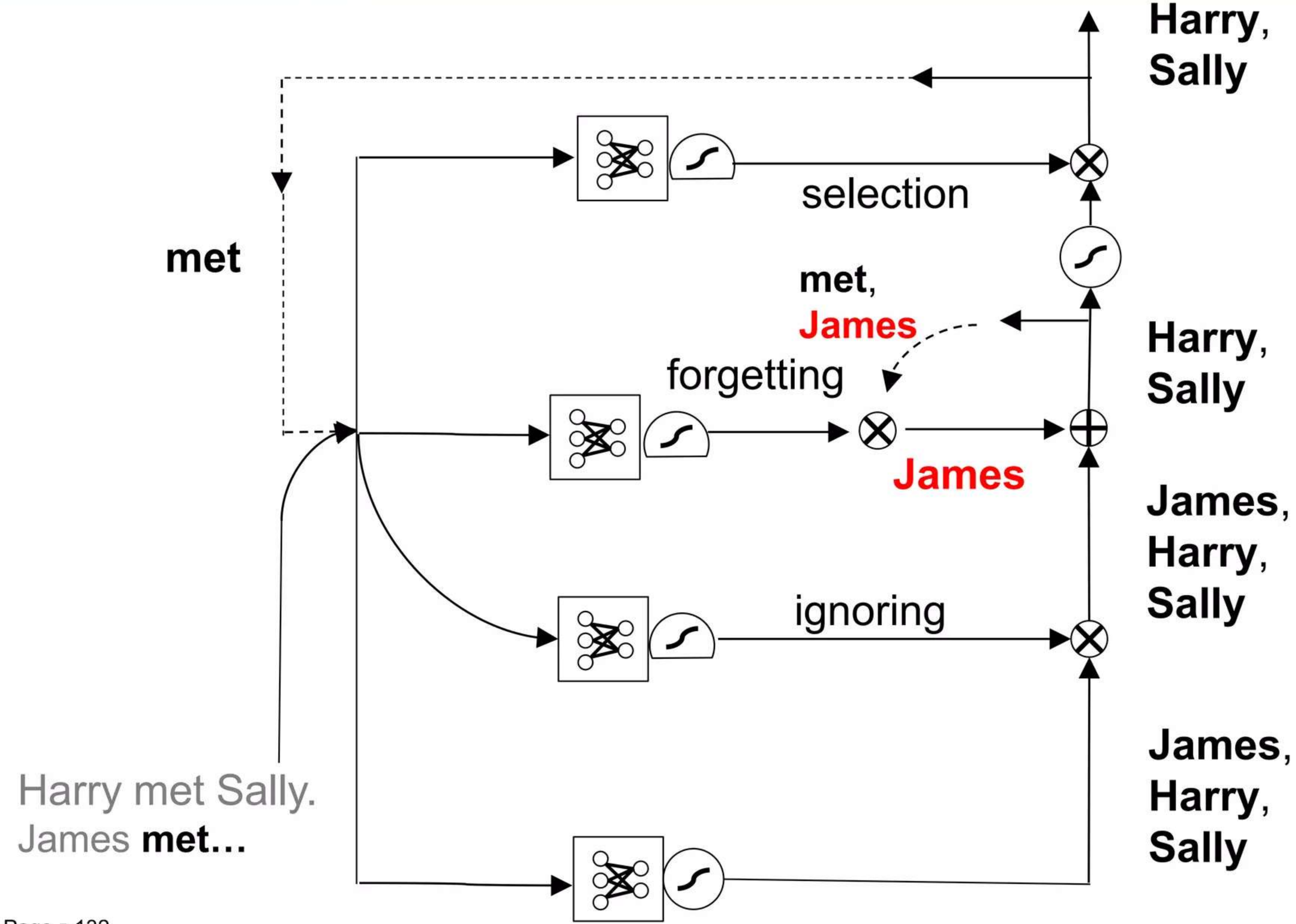
LSTM in Action



LSTM in Action



LSTM in Action



Agenda

- Introduction to Deep Learning
  - Neural Nets Refresher
  - Reasons to go Deep
- Demo 1 – Keras
- How to Choose a Deep Net
- Introcuction to CNN
  - Architecture Overview
  - How ConvNet Works
- ConvNet Layers
  - Convolutional Layer
  - Pooling Layer
  - Normalization Layer (ReLU)
  - Fully-Connected Layer
- Hyper Parameters
- Demo 2 – MNIST Classification
- Introduction to RNN
  - Architecture Overview
  - How RNN's Works
  - RNN Example
- Why RNN's Fail
- LSTM
  - Memory
  - Selection
  - Ignoring
- LSTM Example
- Demo 3 – Imdb Review Classification
- Image Captioning

References

- Karpathy, A. (n.d.). CS231n Convolutional Neural Networks for Visual Recognition. Retrieved from http://cs231n.github.io/convolutional-networks/#overview
- Rohrer, B. (n.d.). How do Convolutional Neural Networks work?. Retrieved from http://brohrer.github.io/how_convolutional_neural_networks_work.html
- Brownlee, J. (n.d.). Crash Course in Convolutional Neural Networks for Machine Learning. Retrieved from http://machinelearningmastery.com/crash-course-convolutional-neural-networks/
- Lidinwise (n.d.). The revolution of depth. Retrieved from https://medium.com/@Lidinwise/the-revolution-of-depth-facf174924f5#.8or5c77ss
- Nervana. (n.d.). Tutorial: Convolutional neural networks. Retrieved from https://www.nervanasys.com/convolutional-neural-networks/
- Olah, C. (n.d.). Tutorial: Understanding LSTM Networks. Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- Rohrer, B. (n.d.). Tutorial: How Recurrent Neural Networks and Long Short-Term Memory Work. Retrieved from https://brohrer.github.io/how_rnns_lstm_work.html
- Serrano, L. (n.d.). A friendly introduction to Recurrent Neural Networks. Retrieved from https://www.youtube.com/watch?v=UNmqTiOnRfg&t=87s

Thank you