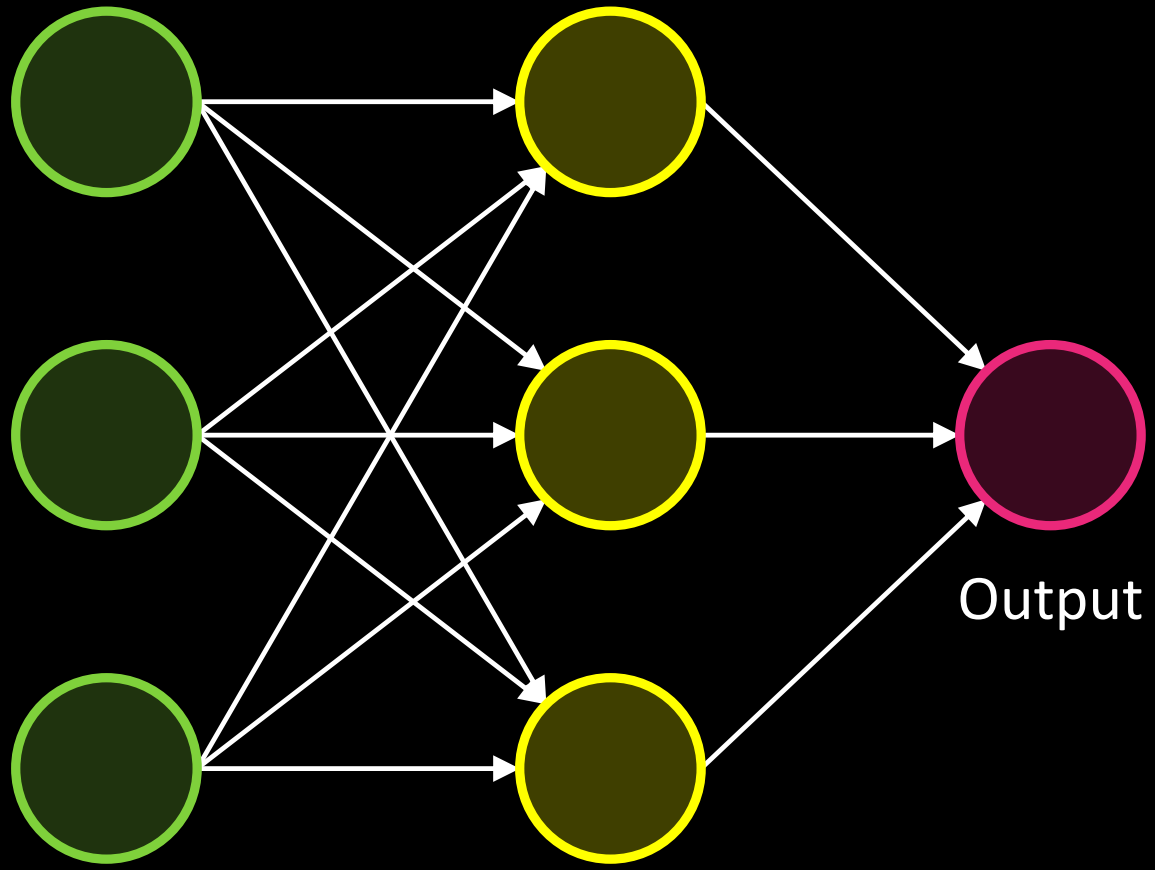
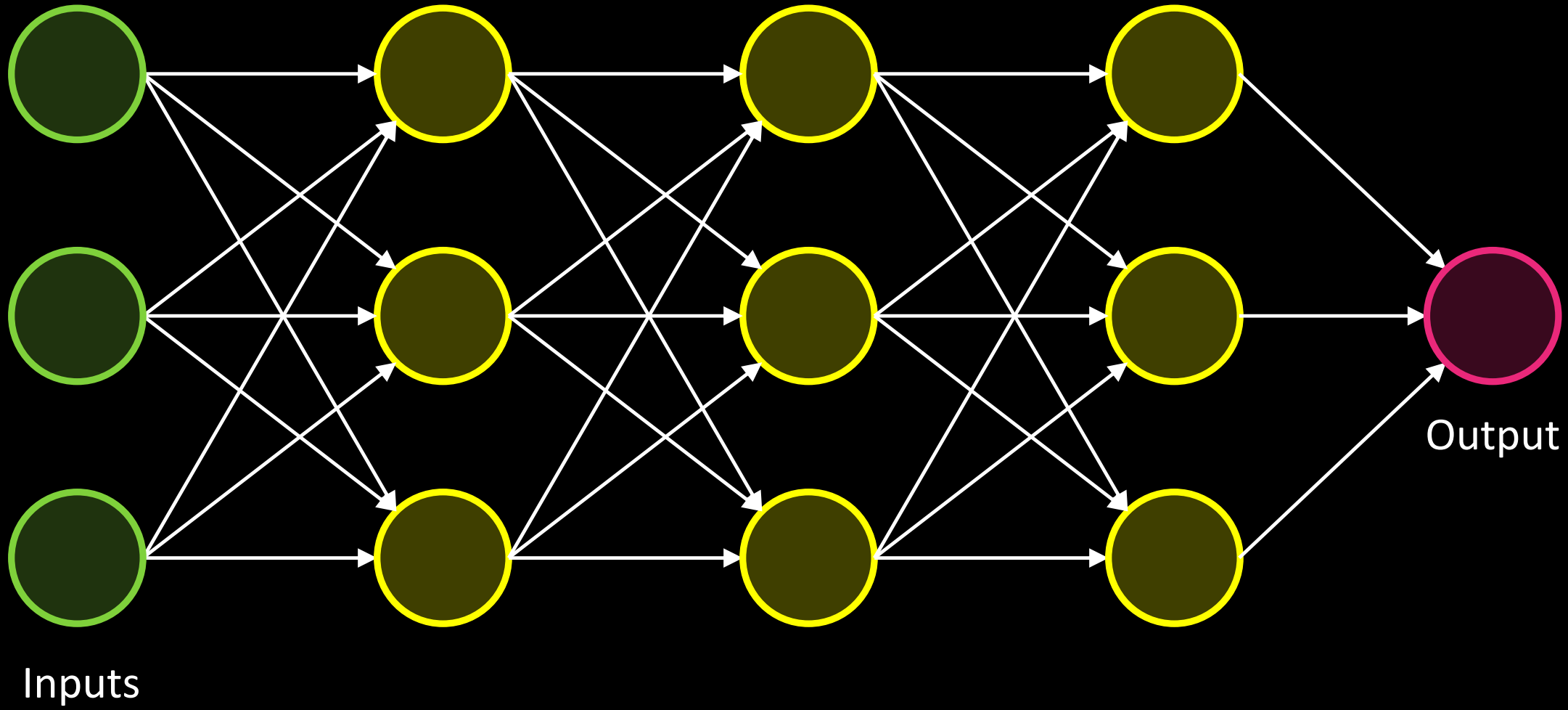
HARVARD
UNIVERSITY

Deep Learning

Mohammed AlQuraishi

MIT Media Lab, 10/19/15


Output
Inputs
Computational Unit


Output
Inputs

1943 – 2006: A prehistory of deep learning
2006 – 2015: A history of deep learning

2006 – 2014: What is deep learning? Take I

2014 – 20XX: What is deep learning? Take II

1943 – 2006: A prehistory of deep learning

2006 – 2015: A history of deep learning

2006 – 2014: What is deep learning? Take I

2014 – 20XX: What is deep learning? Take II

1943: Warren McCulloch and Walter Pitts

1949: Donald Hebb

1954: Farley and Wesley Clark

1958: Frank Rosenblatt (perceptron)

1969: Marvin Minsky and Seymour Papert

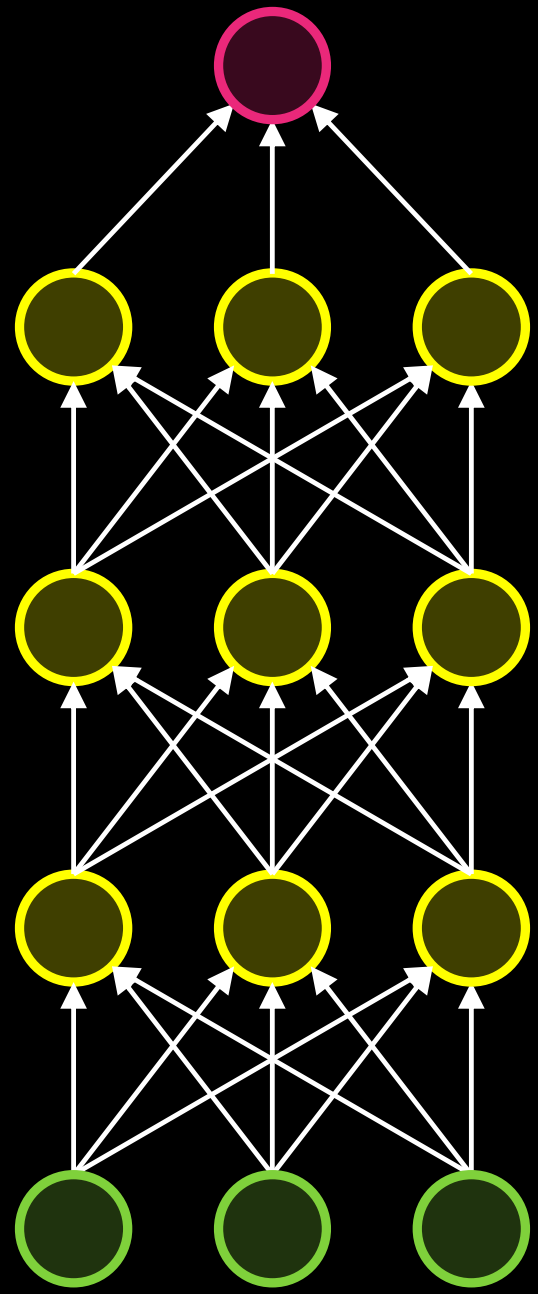
1975: Paul Werbos (backpropagation)

1980s – 2006: Geoff Hinton
Yann LeCun
Yoshua Bengio
Jürgen Schmidhuber

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

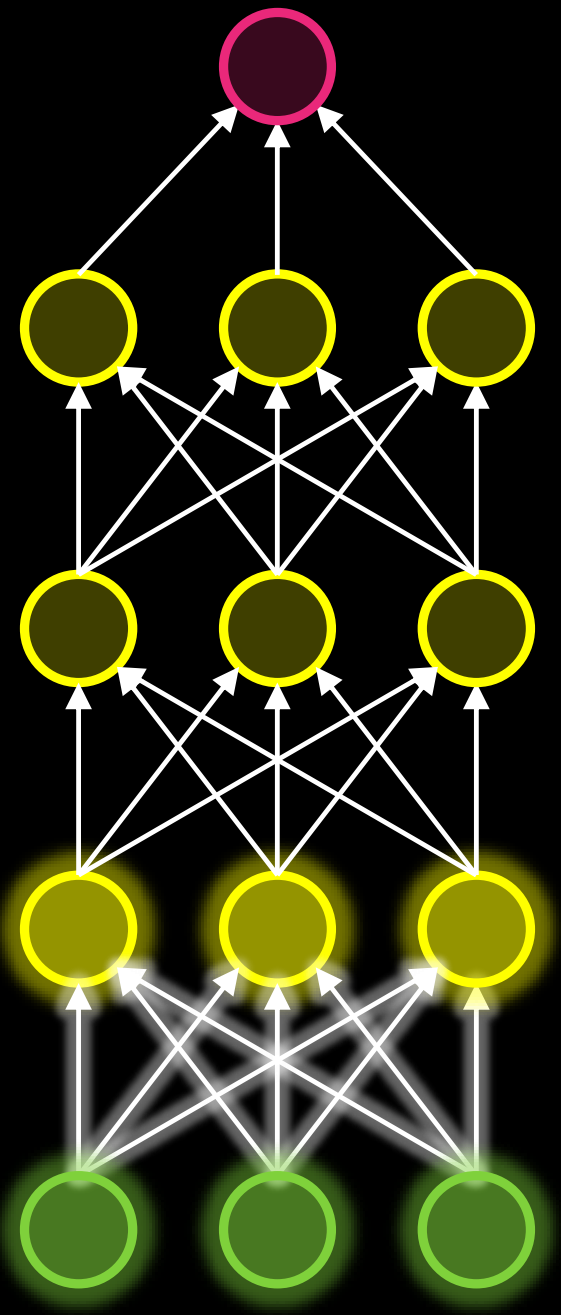
High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.



Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

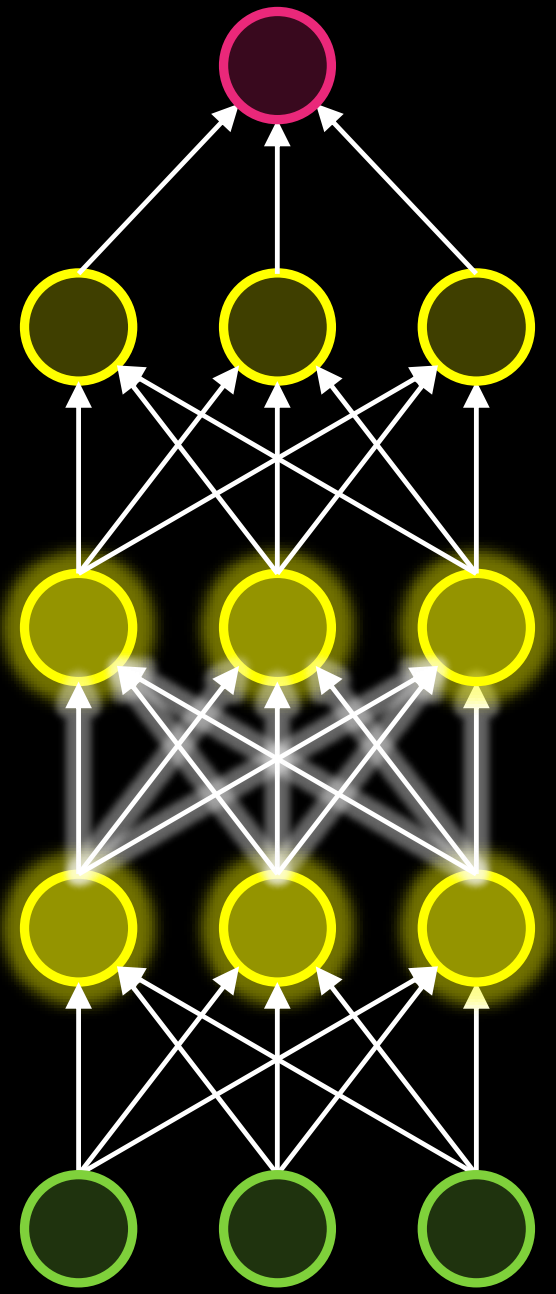
High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.



Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

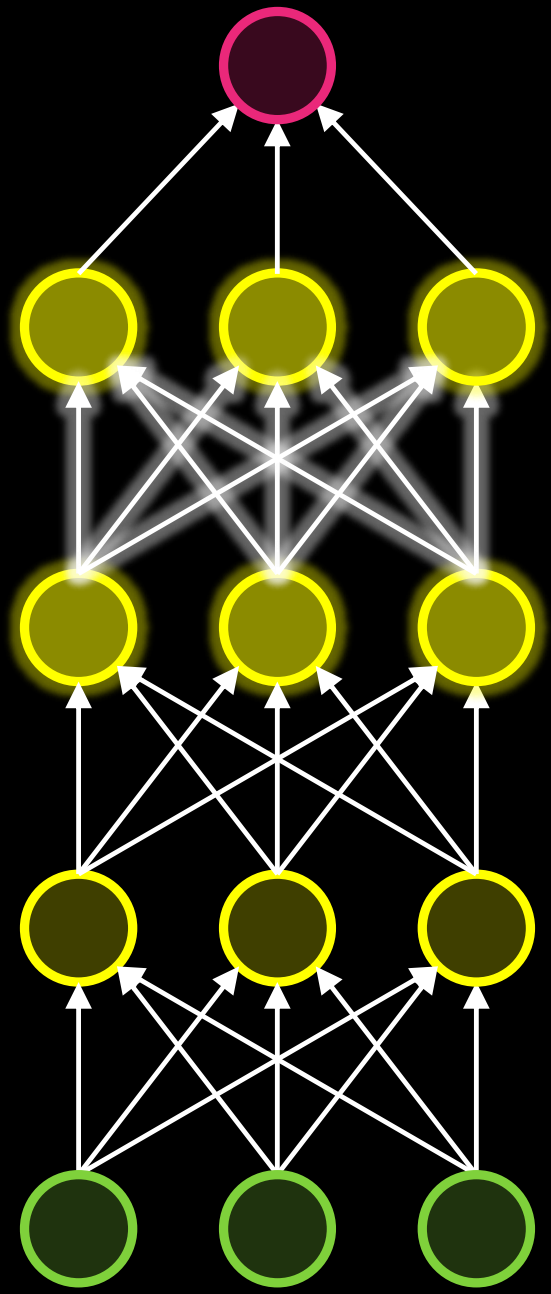
High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.



Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

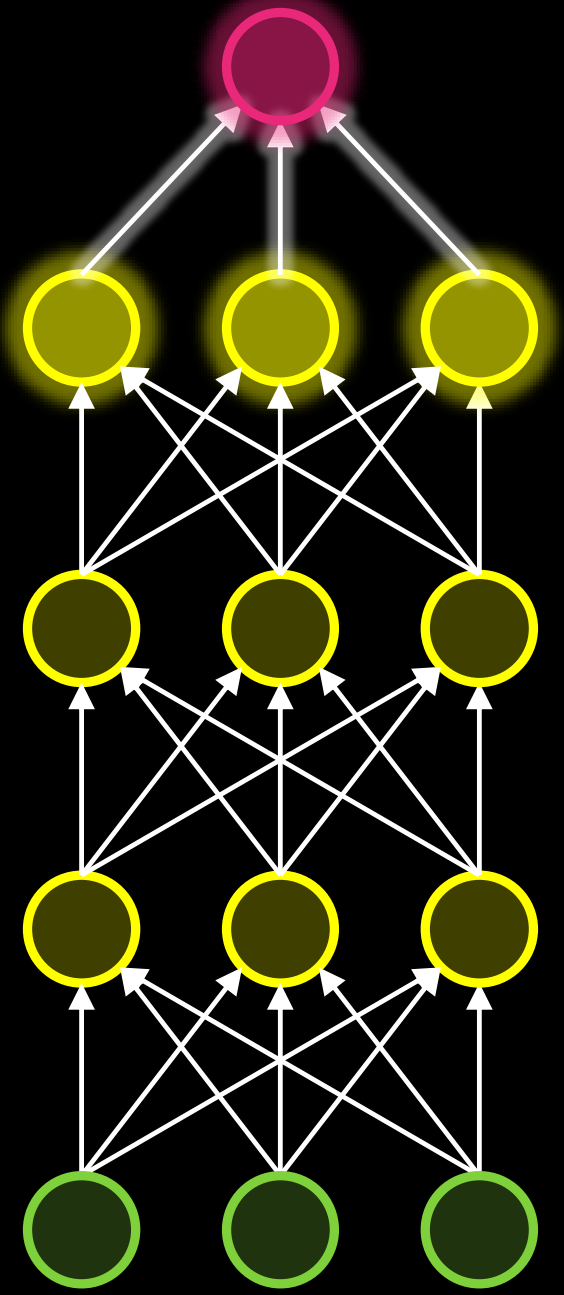
High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.



Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.



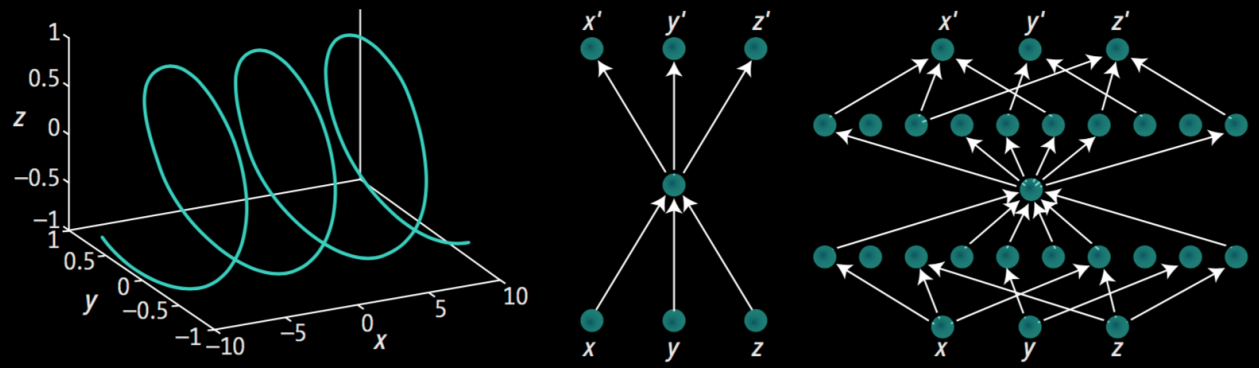
COMPUTER SCIENCE

New Life for Neural Networks

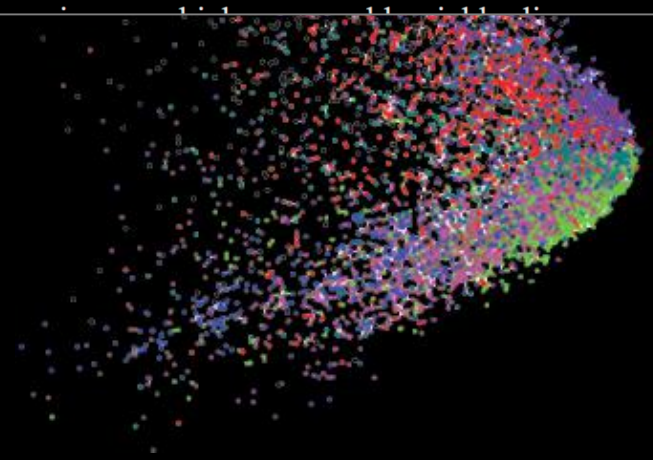
With the help of neural networks, data sets with many dimensions can be analyzed to find lower dimensional structures within them.

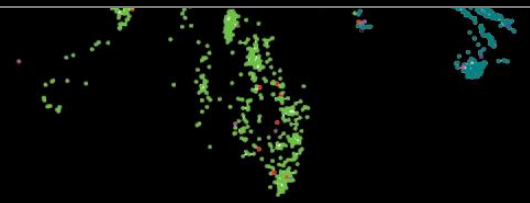
Garrison W. Cottrell

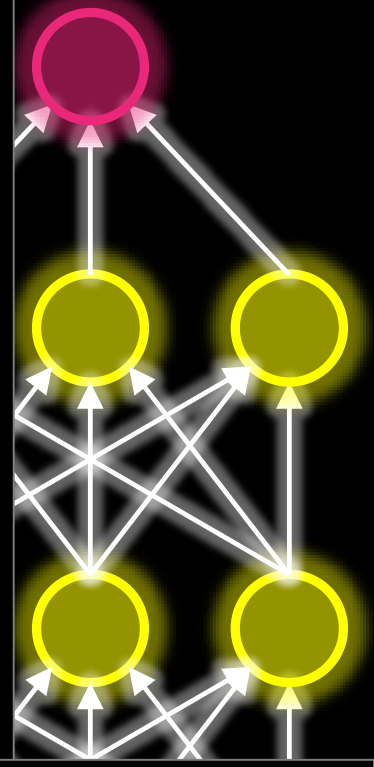
As many researchers have found, the data they have to deal with are often high-dimensional—that is, expressed by many variables—but may contain a great deal of latent structure. Discovering that structure, however, is nontrivial. To illustrate the point, consider a case in the relatively low dimension of three. Suppose you are handed a large number of three-dimensional points in random order (where each point is denoted by its coordinates along the *x*, *y*, and *z* axes): {(–7.4000, –0.8987, 0.4385), (3.6000, –0.4425, –0.8968), (–5.0000, 0.9589, 0.2837), …}. Is there a more compact, lower dimensional description of these data? In this case, the

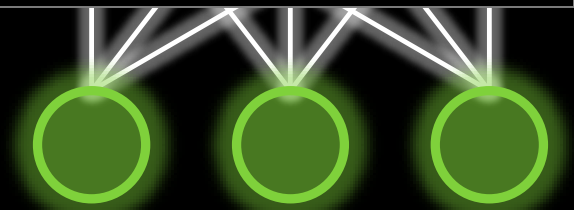


Searching for structure. **(Left)** Three-dimensional data that are inherently one-dimensional. **(Middle)** A simple "autoencoder" network that is designed to compress three dimensions to one, through the narrow hidd… aut… one…





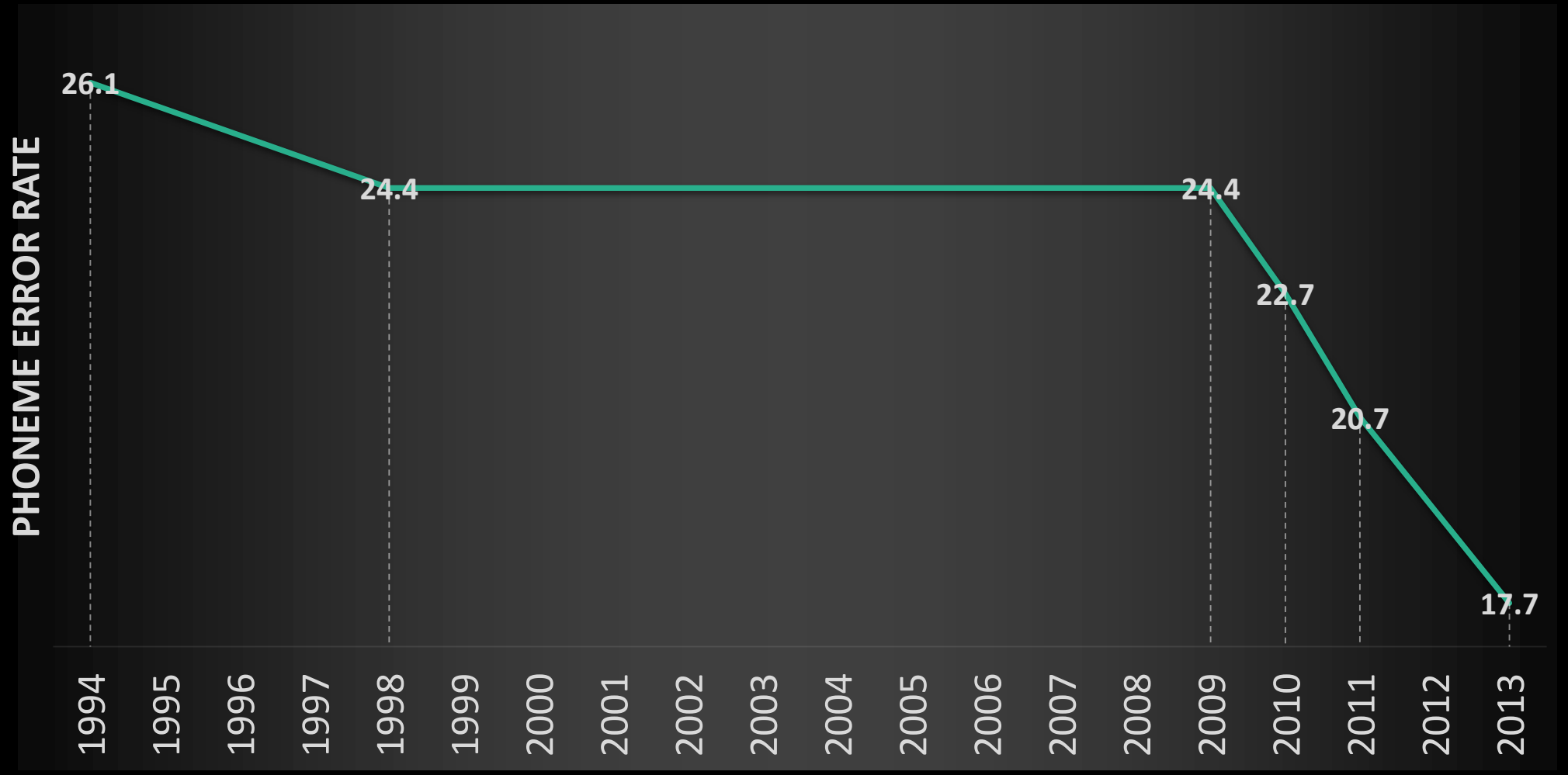




Recent advances in machine learning have caused some to consider neural networks obsolete, even dead. This work suggests that such announcements are premature.

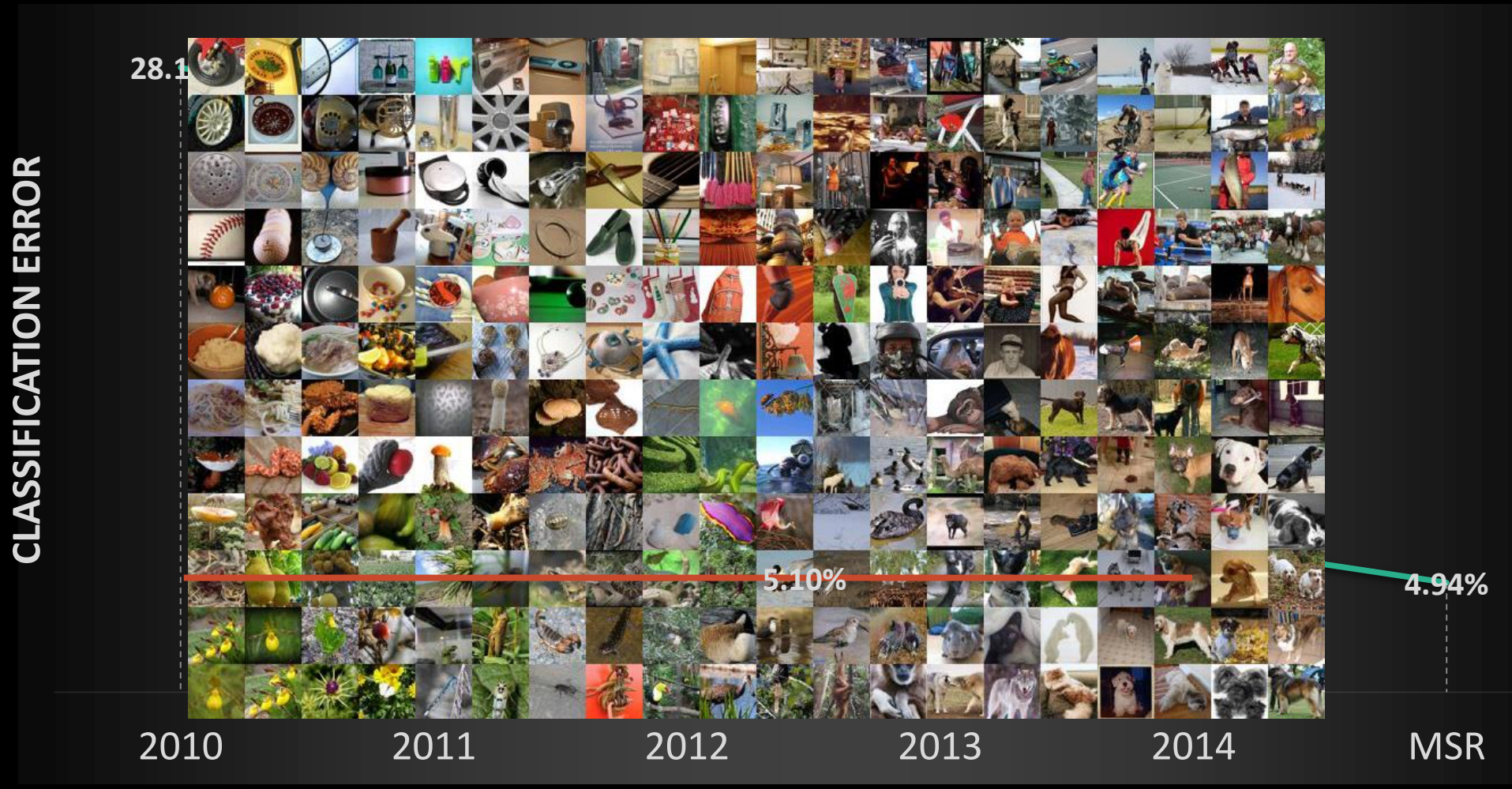
TIMIT

Transcribed Speech of American English



ImageNet

Classification of 1000 Image Categories



The New York Times

SCIENCE

SCIENCE

Researchers Announce Advance in Image-Recognition Software

ROBOTICS

Computer Eyesight Gets a Lot More Accurate

SCIENCE

New Approach Trains Robots to Match Human Dexterity and Speed

CIAL

INTELLIGENCE?

BY GARY MARCUS

2011: Andrew Ng → Google Brain
Mar 2013: Geoff Hinton → Google

Dec 2013: Zuckerberg at NIPS
Dec 2013: Yann LeCun → Facebook

DEEPMIND

Jan 2014: DeepMind → Google


Google
Microsoft
Baidu 百度

1943 – 2006: A prehistory of deep learning

2006 – 2015: A history of deep learning

2006 – 2014: What is deep learning? Take I

2014 – 20XX: What is deep learning? Take II

1943 – 2006: A prehistory of deep learning

2006 – 2015: A history of deep learning

2006 – 2014: What is deep learning? Take I

2014 – 20XX: What is deep learning? Take II

Hal Daumé III @haldaume3 · Sep 23

@ogrisel right we keep coming back to lamppost :) but this is basically saying DL = DNN which is a bandwagon I can't get on

Kyle Kastner @kastnerkyle · Sep 23

@haldaume3 @ogrisel @yoavgo for me DL is close to probabilistic graphical models but trained by backprop vs. Message passing or w/e

Kyle Kastner @kastnerkyle · Sep 23

@haldaume3 @ogrisel @yoavgo this leads to focus on jointly trained systems, weird conditional architectures, differentiability

Supervised Learning

$i = 1, \ldots, N$ $\{x_i, y_i\}$

x_1: (02180, 1 bedroom, 1930) y_1:$500K

x_1:  y_1: “apple”

x_1: Washington D.C. is the capitol of the US.
y_1: Paris est la capitale de la France.

Supervised Learning

$i = 1, \ldots, N$ $\{x_i, y_i\}$ $f: basis\ set$

$$y_i^* \approx \sum_{k=1}^{K} w_k^* x_{ik}$$

$$y_i^* = \langle w^*, x_i \rangle \approx y_i$$

$f = I$

$$y_i^* = \langle w^*, f(x_i) \rangle \approx y_i$$

fixed

$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \| \langle w, f(x_i) \rangle - y_i \|$$

Deep Learning as Adaptive Basis Regression

$i = 1, \ldots, N$ $\quad \{x_i, y_i\}$ $\quad f: basis\ set$

$$y_i^* \approx \sum_{k=1}^{K} w_k^* x_{ik}$$

$$y_i^* = \langle w^*, x_i \rangle \approx y_i$$

$f = I$

$$y_i^* = \langle w^*, f(x_i) \rangle \approx y_i$$

fixed

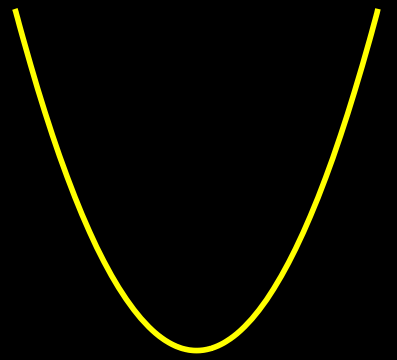
$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \| \langle w, f(x_i) \rangle - y_i \|$$

Deep Learning as Adaptive Basis Regression

$i = 1, \ldots, N$ $\{x_i, y_i\}$ $f: basis\ set$

Representation Learning

Breaks Convexity



$$y_i^* = \langle w^*, f(x_i) \rangle \approx y_i$$

learned

$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \| \langle w, f(x_i) \rangle - y_i \|$$

Deep Learning as Adaptive Basis Regression

$i = 1, \ldots, N$ $\{x_i, y_i\}$ $f: basis\ set$

Feature Selection

$$y_i^* = \langle w^*, f(x_i) \rangle \approx y_i$$

learned

$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \| \langle w, f(x_i) \rangle - y_i \| + reg(w)$$

Deep Learning as a Choice of Hypothesis Space

$i = 1, \ldots, N$ $\quad \{x_i, y_i\}$ $\quad f^*(x_i) \approx y_i$

$\mathcal{F}$: *hypothesis space*

$\mathcal{F} = \{g_a \otimes g_b | g_a, g_b \in \mathcal{G}\}$

$\mathcal{G}$: *atomic funcs* $\quad \otimes$: *operator*

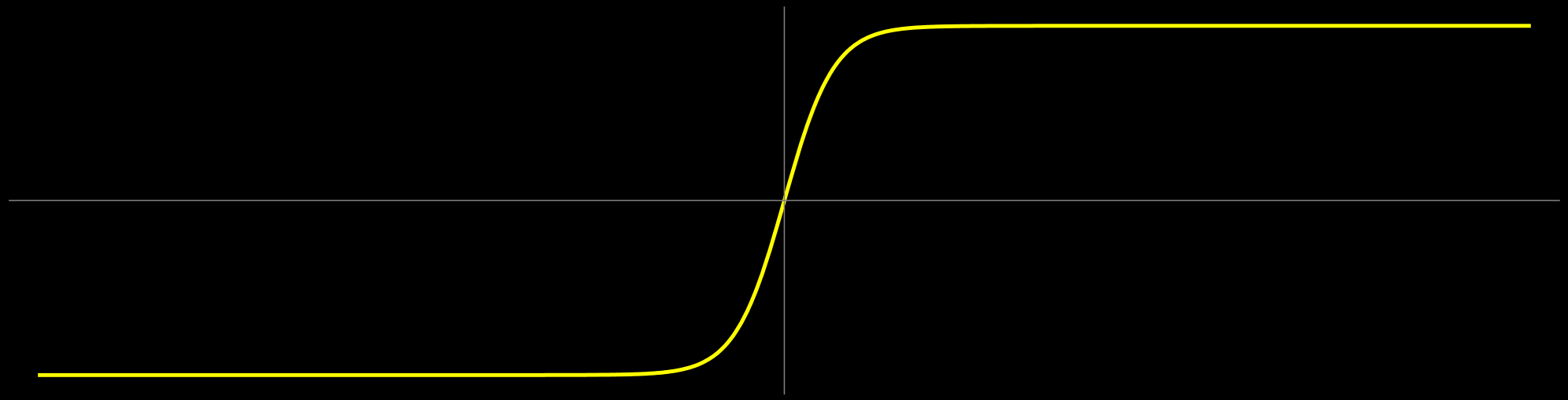
$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{n} \| f(x_i) - y_i \|$$

Linear Regression	**Kernel Methods**	**Deep Learning**
$+$	$\times$	$\circ$
$g_a(x_i) + g_b(x_i)$	$g_a(x_i) g_b(x_i)$	$g_b(g_a(x_i))$

Deep Learning in Practice

$g_l(x) = \langle w, x \rangle$ basically a rotation

$g_{nl}(x) = \tanh(x)$ basically a threshold



Deep Learning in Practice

$g_l(x) = \langle w, x \rangle$ basically a rotation

$g_{nl}(x) = \tanh(x)$ basically a threshold

$g_l(x)$

Deep Learning in Practice

$g_l(x) = \langle w, x \rangle$ basically a rotation

$g_{nl}(x) = \tanh(x)$ basically a threshold

$g_{ln}(g_l(x))$

Deep Learning in Practice

$g_l(x) = \langle w, x \rangle$ basically a rotation

$g_{nl}(x) = \tanh(x)$ basically a threshold

$$g_l\left(g_{ln}(g_l(x))\right)$$

Deep Learning in Practice

$g_l(x) = \langle w, x \rangle$ basically a rotation

$g_{nl}(x) = \tanh(x)$ basically a threshold

$$g_{nl}\left(g_l\left(g_{ln}(g_l(x))\right)\right)$$

Deep Learning in Practice

$g_l(x) = \langle w, x \rangle$ basically a rotation

$g_{nl}(x) = \tanh(x)$ basically a threshold

$$g_l\left(g_{nl}\left(g_l\left(g_{ln}(g_l(x))\right)\right)\right)$$

Deep Learning in Practice

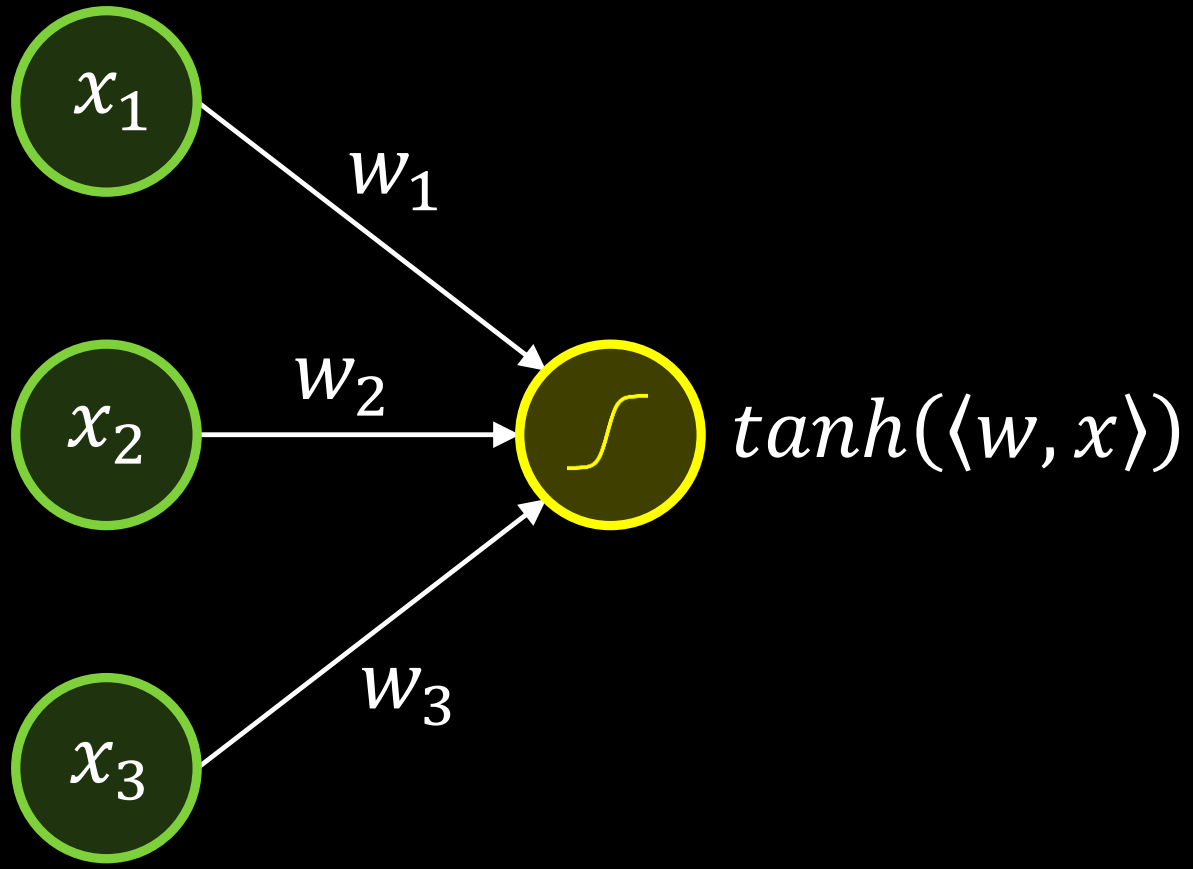
$g_l(x) = \langle w, x \rangle$ basically a rotation

$g_{nl}(x) = \tanh(x)$ basically a threshold

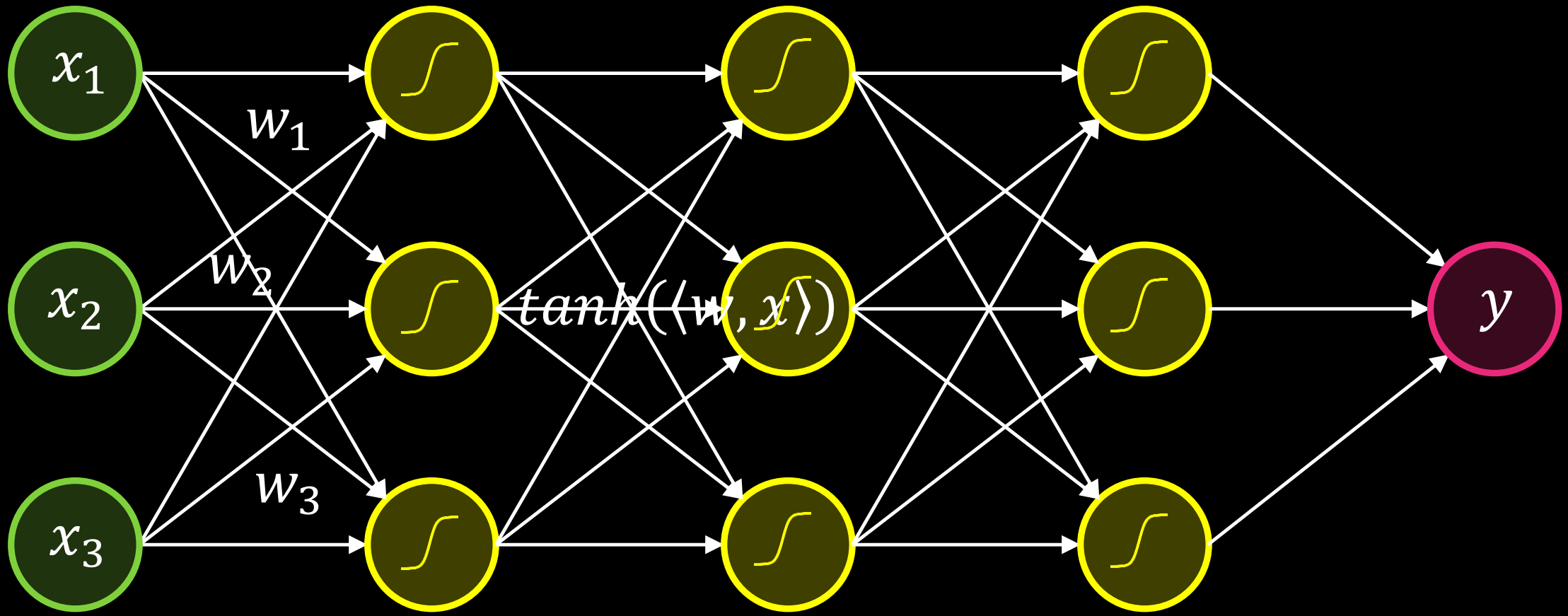
$$g_{nl}\left(g_l\left(g_{nl}\left(g_l\left(g_{ln}(g_l(x))\right)\right)\right)\right)$$

Demo

Standard Neural Architectures: Feedforward

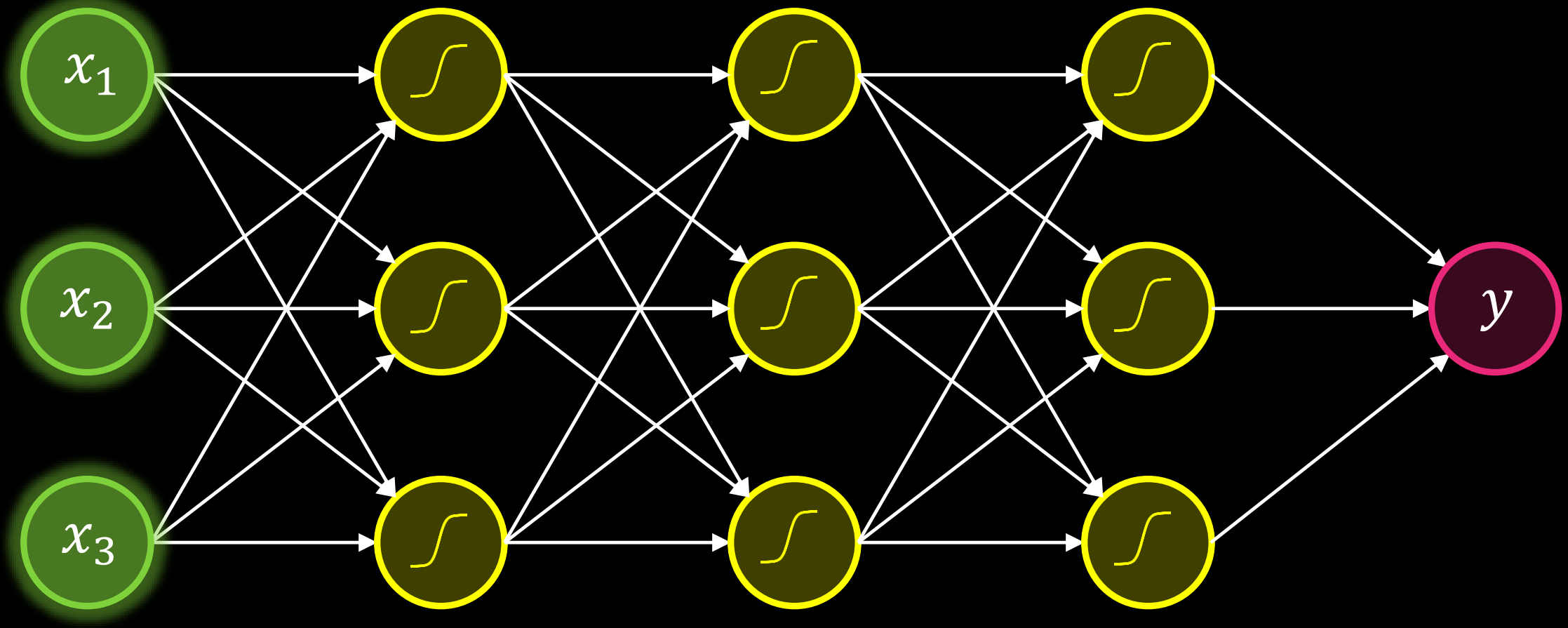


Standard Neural Architectures: Feedforward



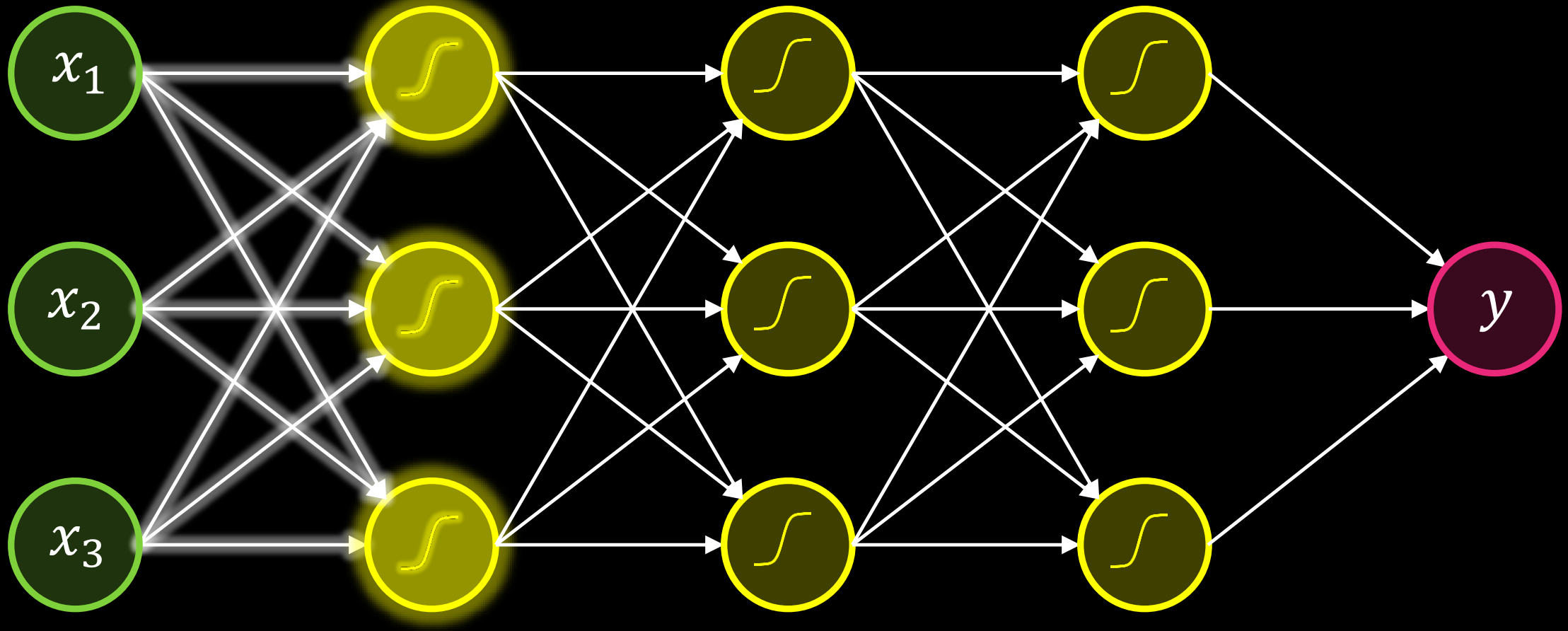
Prediction

Standard Neural Architectures: Feedforward



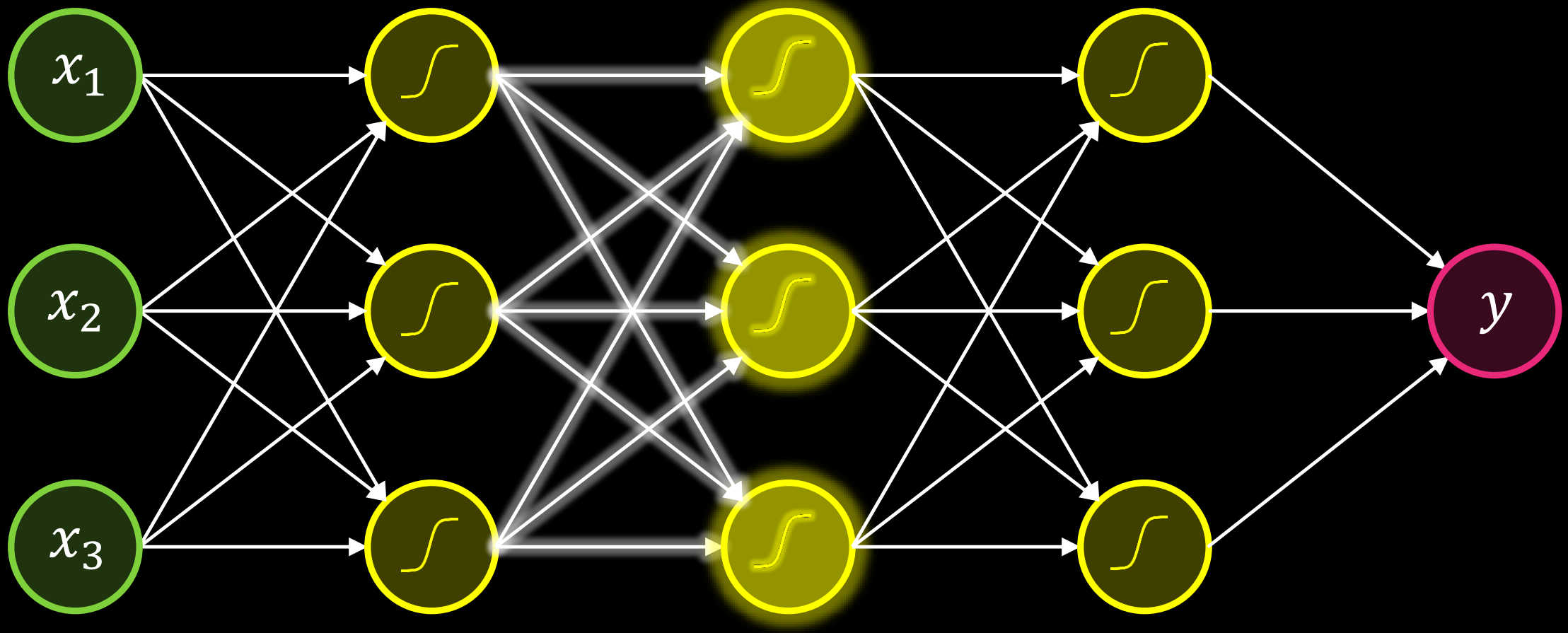
Prediction

Standard Neural Architectures: Feedforward



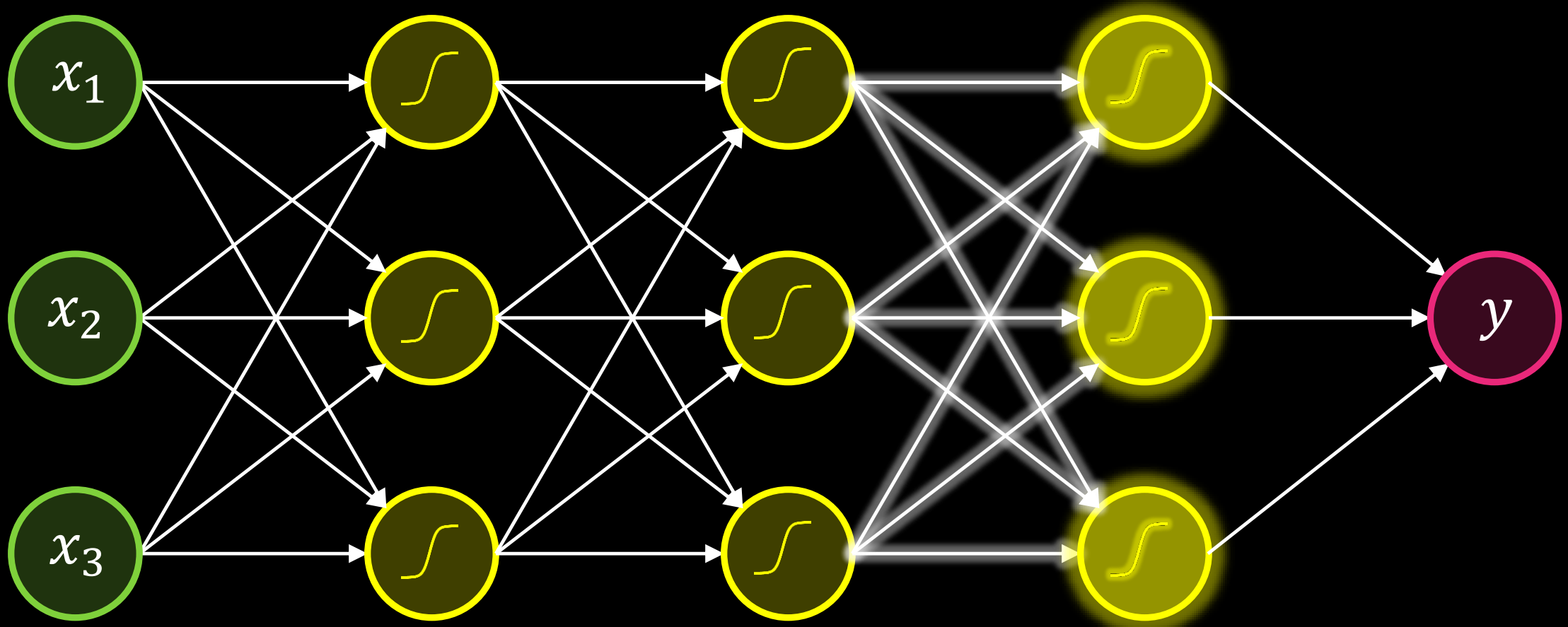
Prediction

Standard Neural Architectures: Feedforward



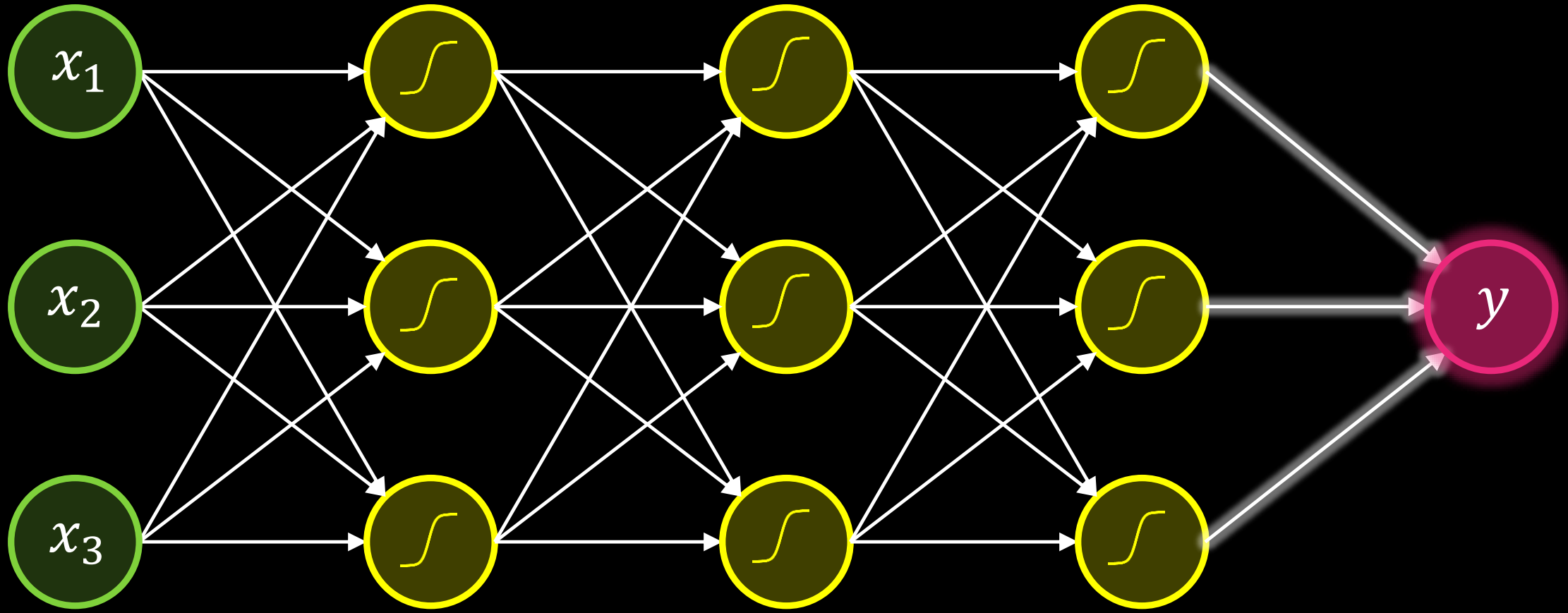
Prediction

Standard Neural Architectures: Feedforward



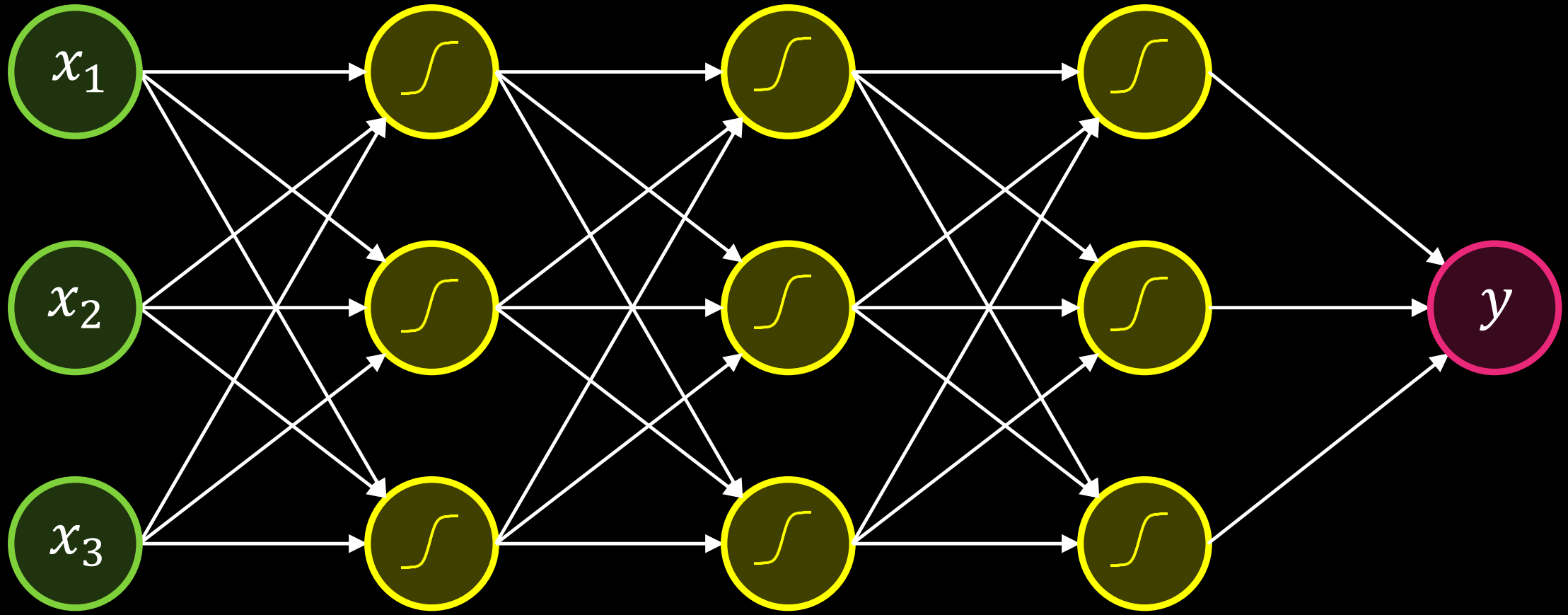
Prediction

Standard Neural Architectures: Feedforward



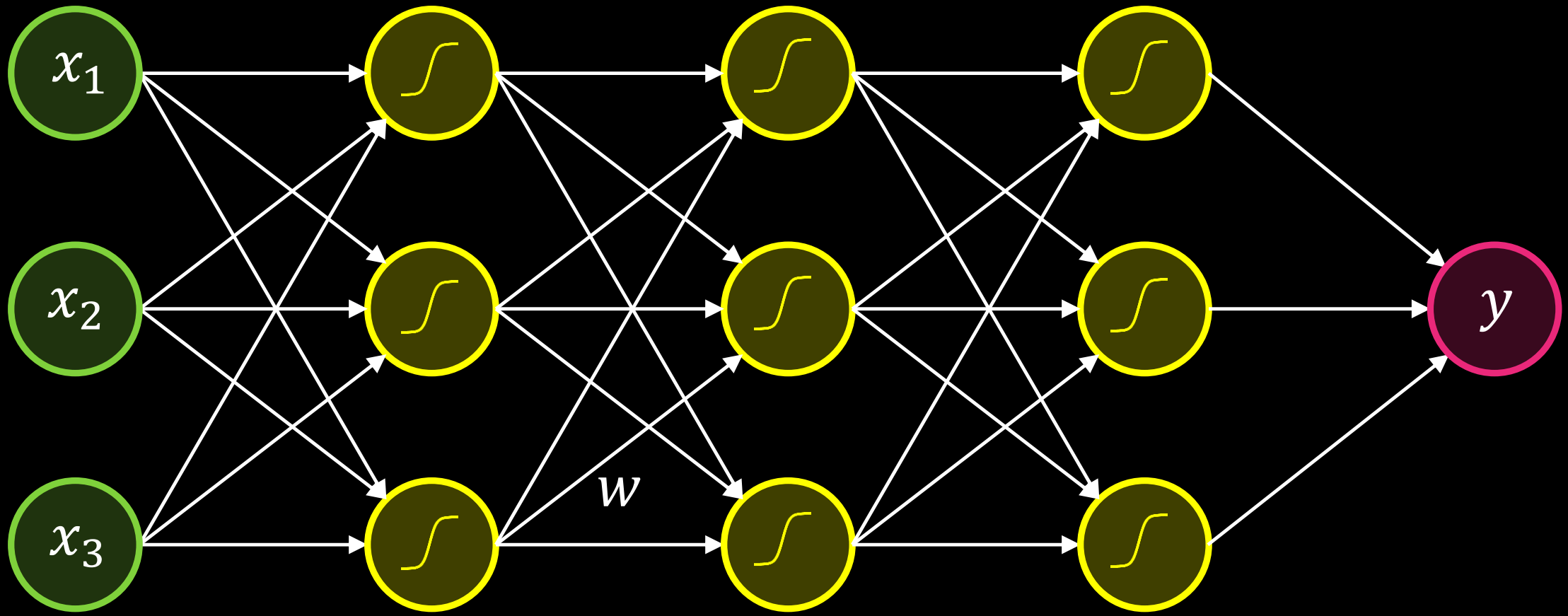
Prediction

Standard Neural Architectures: Feedforward



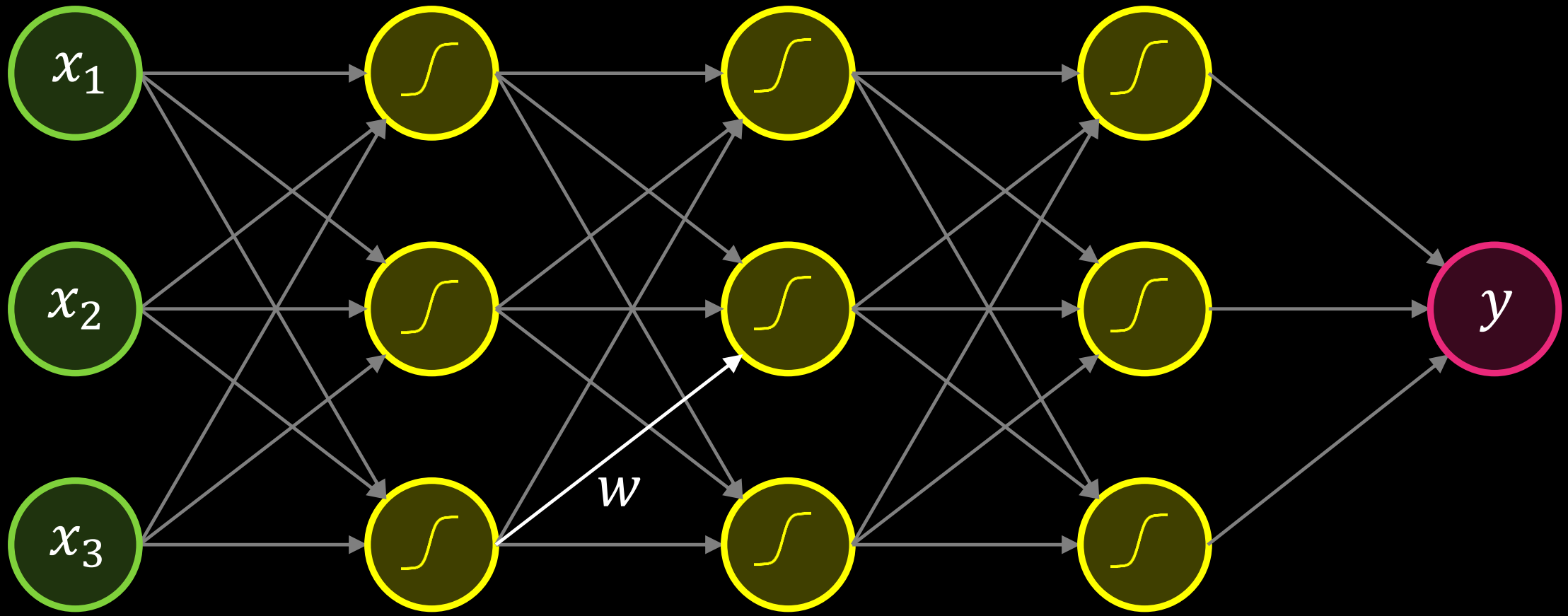
Prediction

Standard Neural Architectures: Feedforward



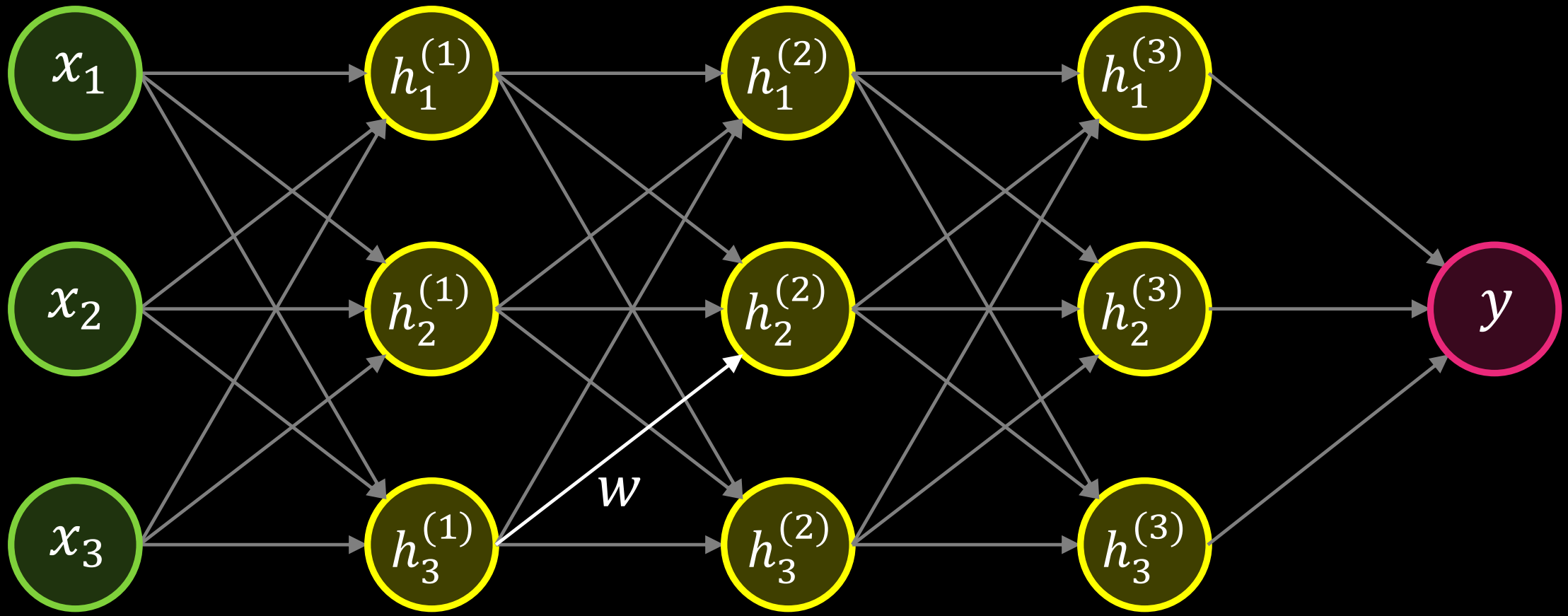
Learning

Standard Neural Architectures: Feedforward



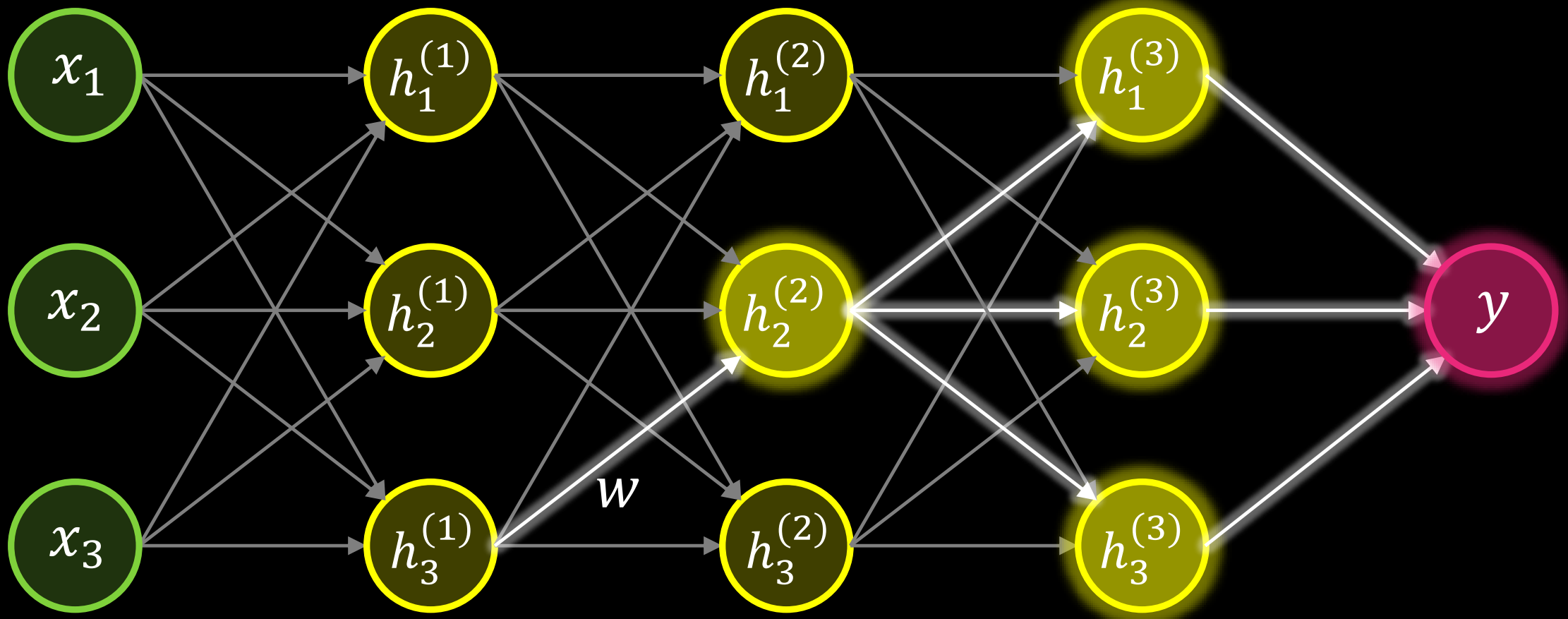
Learning $\left(\frac{\partial y}{\partial w}\right)$

Standard Neural Architectures: Feedforward



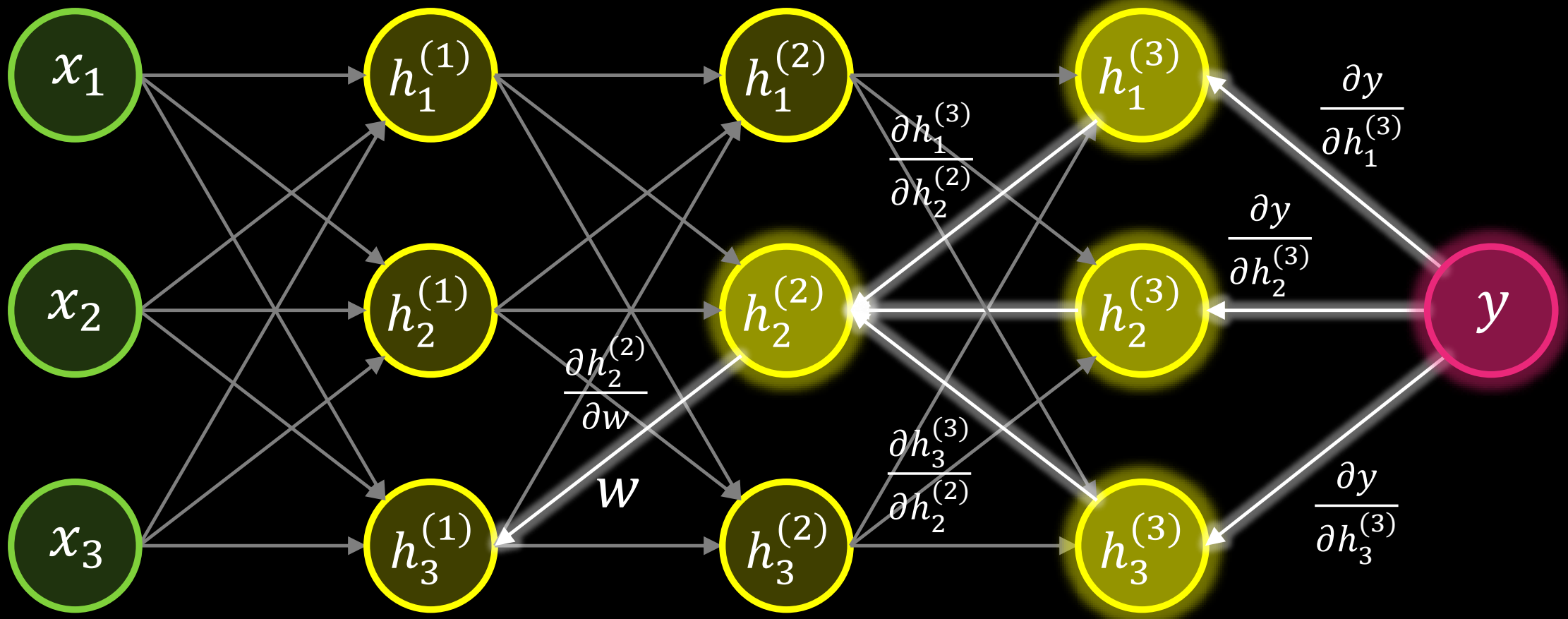
Learning $\left(\frac{\partial y}{\partial w}\right)$

Standard Neural Architectures: Feedforward



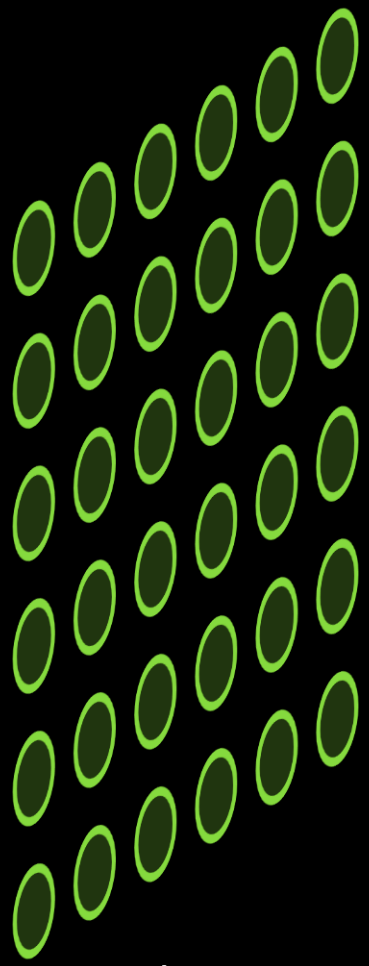
Learning $\left(\frac{\partial y}{\partial w}\right)$

Standard Neural Architectures: Feedforward



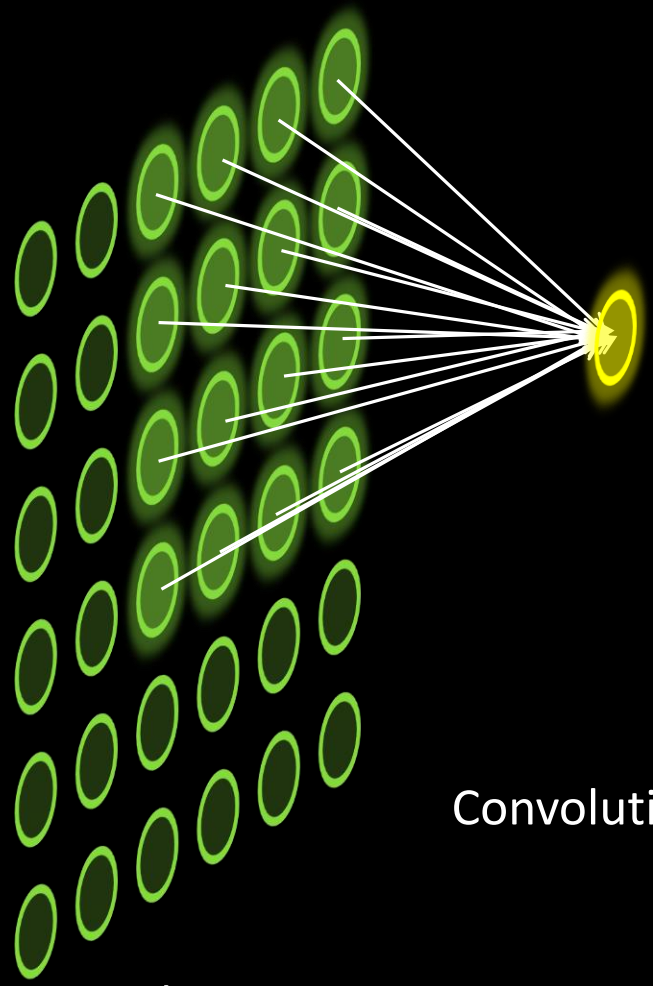
Learning $\left(\frac{\partial y}{\partial w}\right)$

Standard Neural Architectures: Convolutional

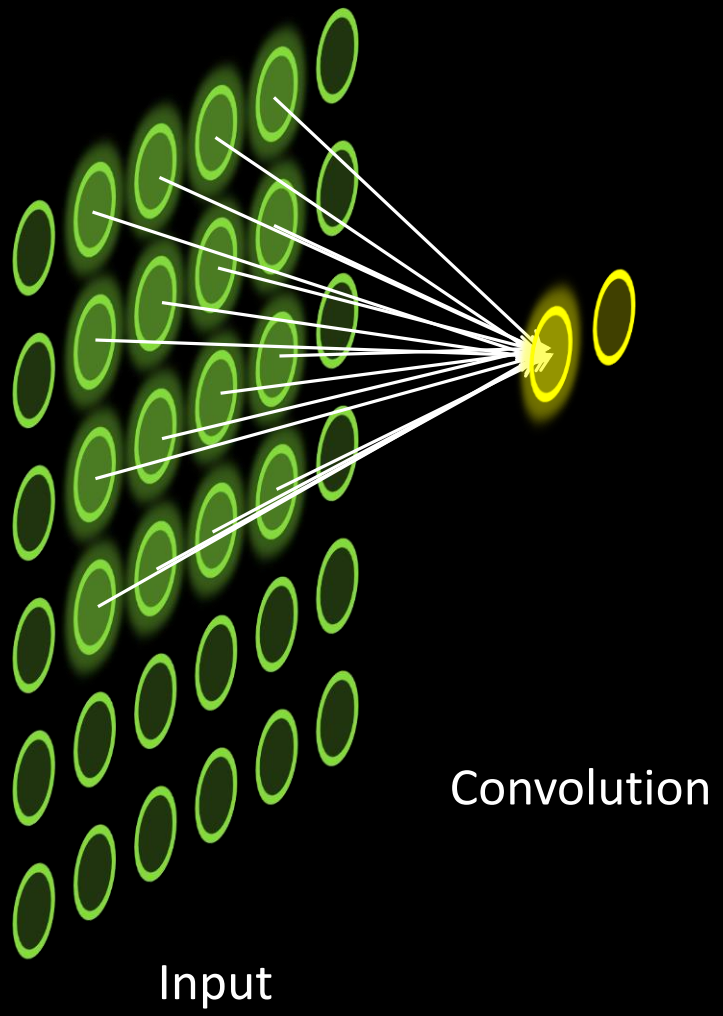


Input

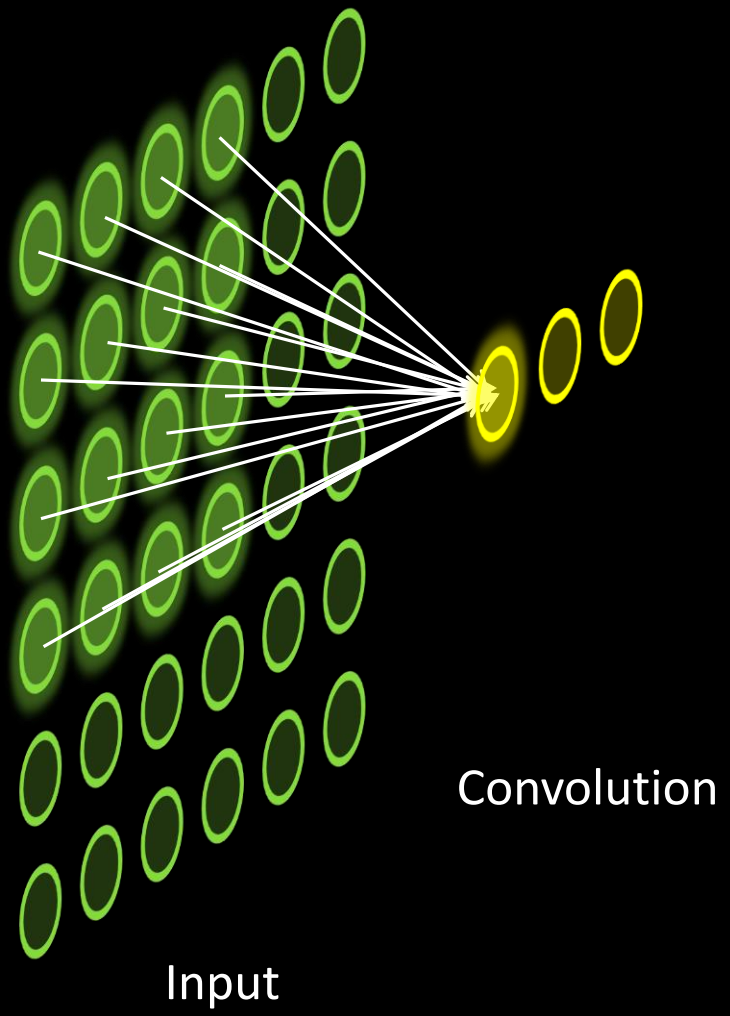
Standard Neural Architectures: Convolutional



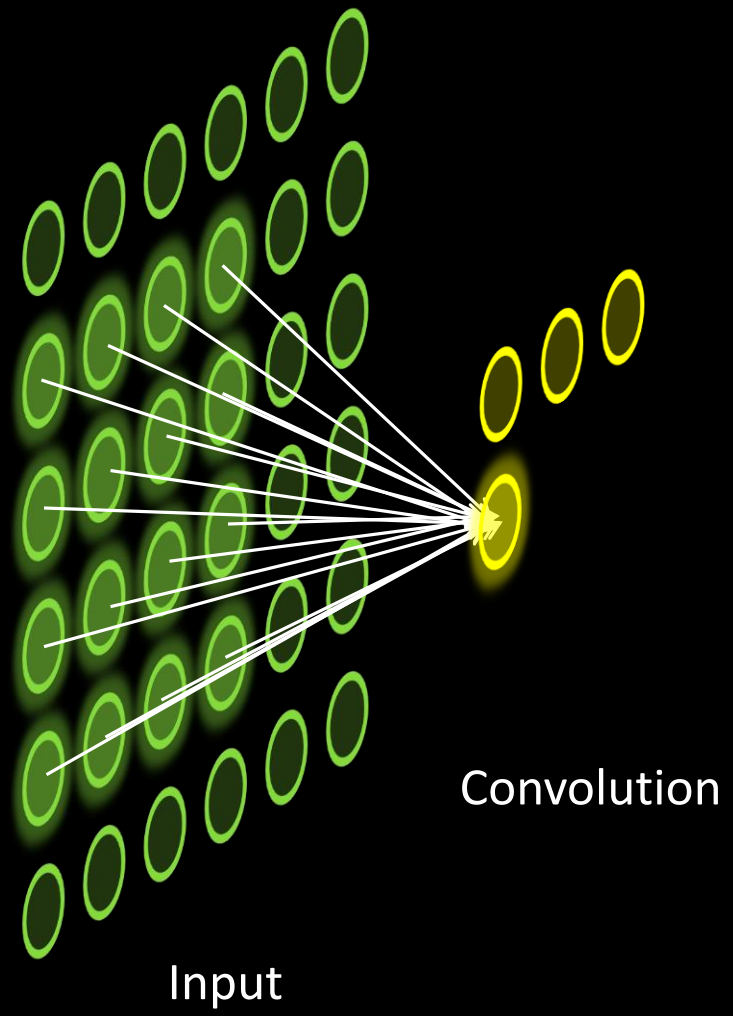
Standard Neural Architectures: Convolutional



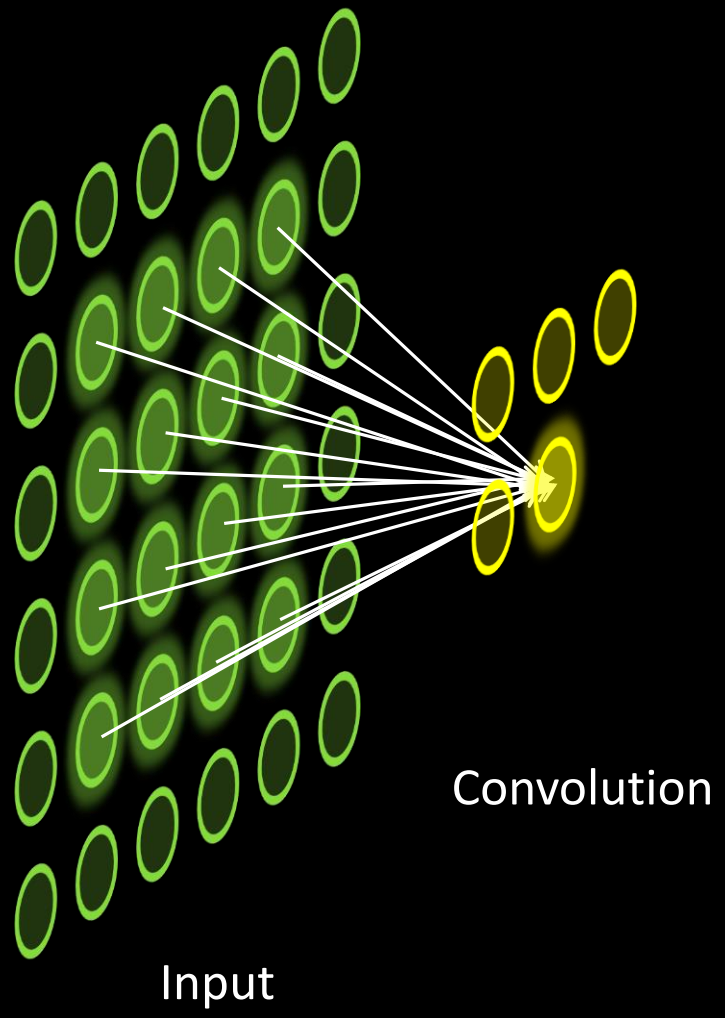
Standard Neural Architectures: Convolutional



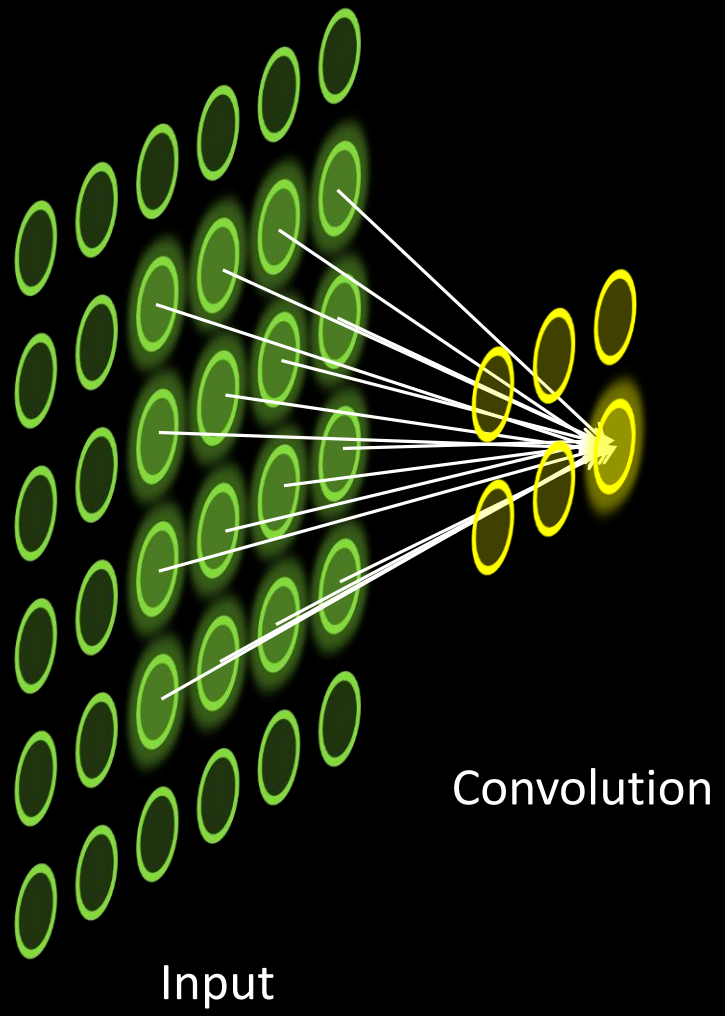
Standard Neural Architectures: Convolutional



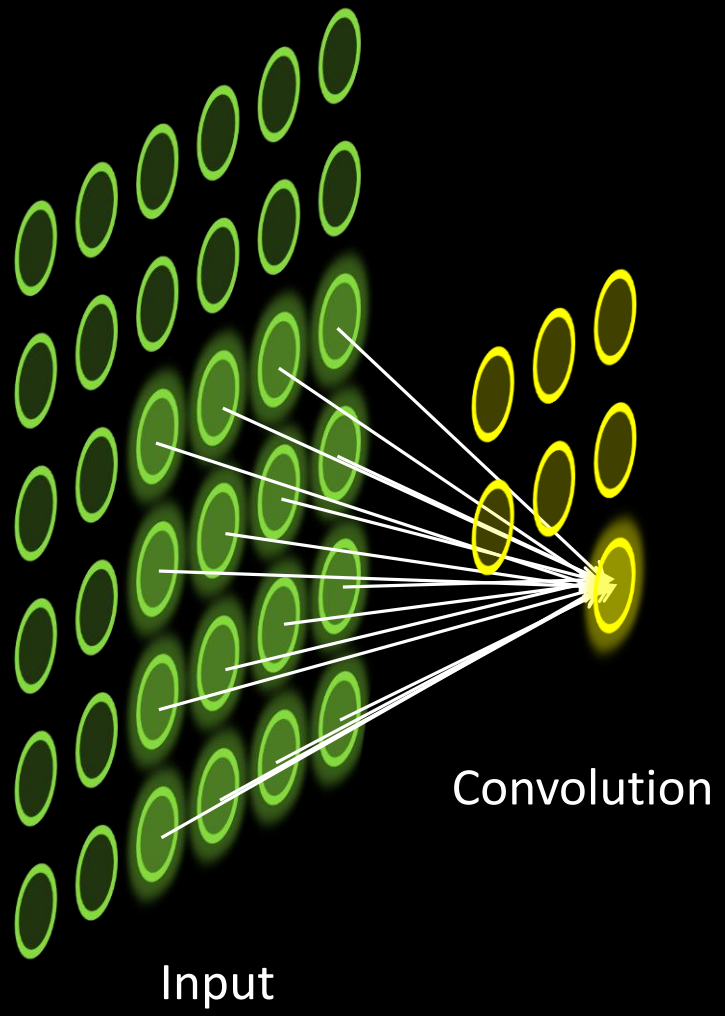
Standard Neural Architectures: Convolutional



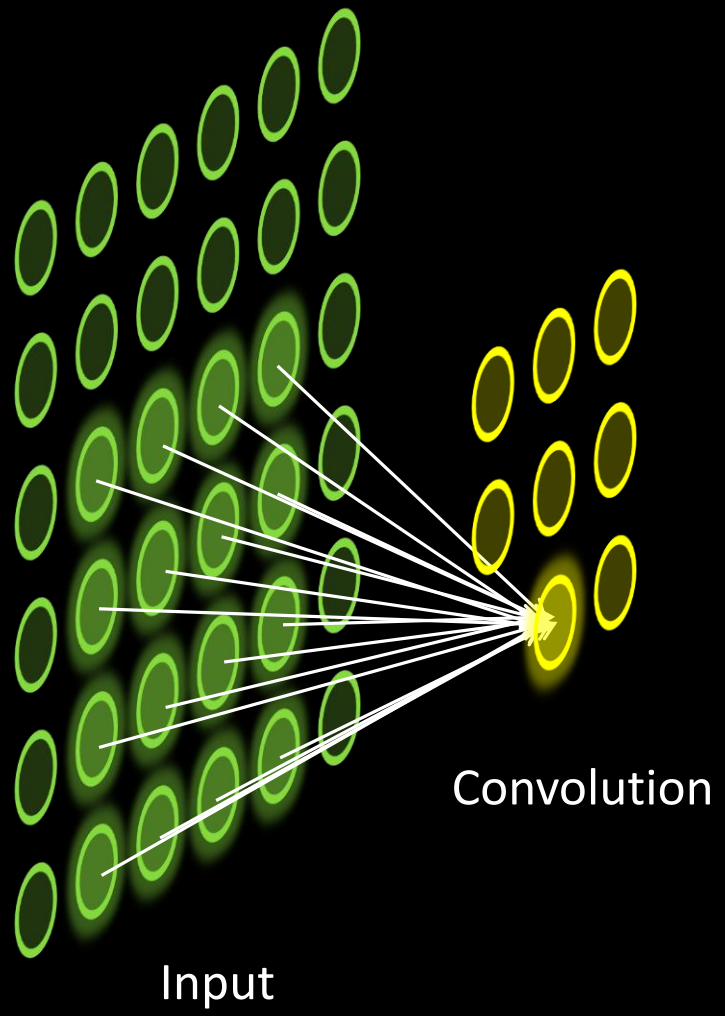
Standard Neural Architectures: Convolutional



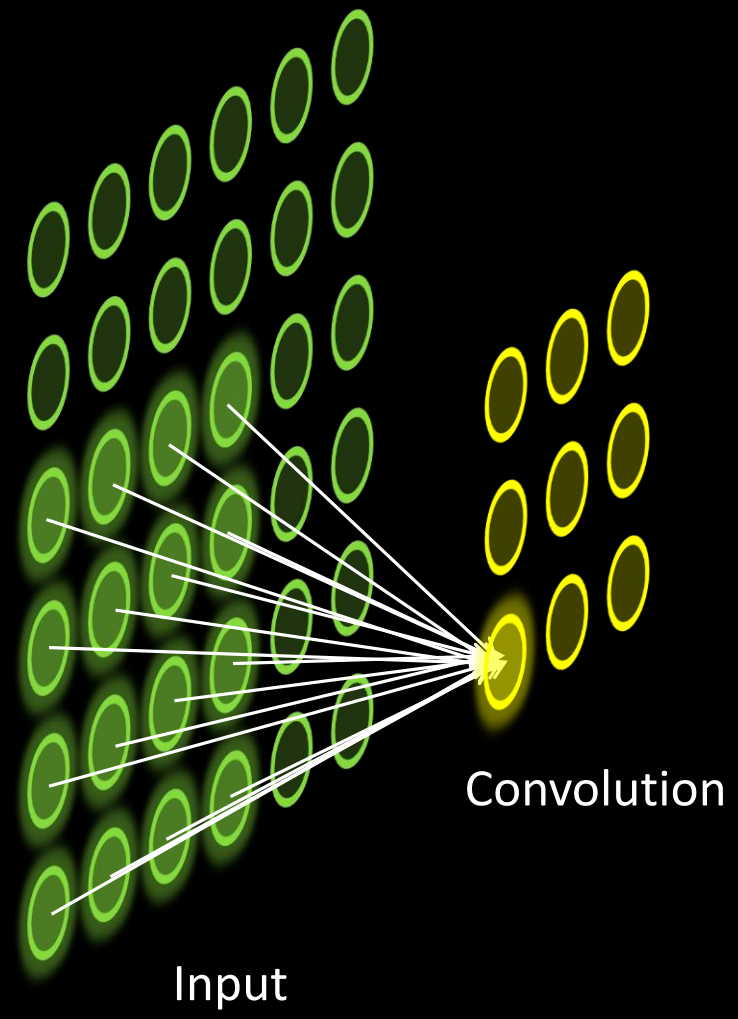
Standard Neural Architectures: Convolutional



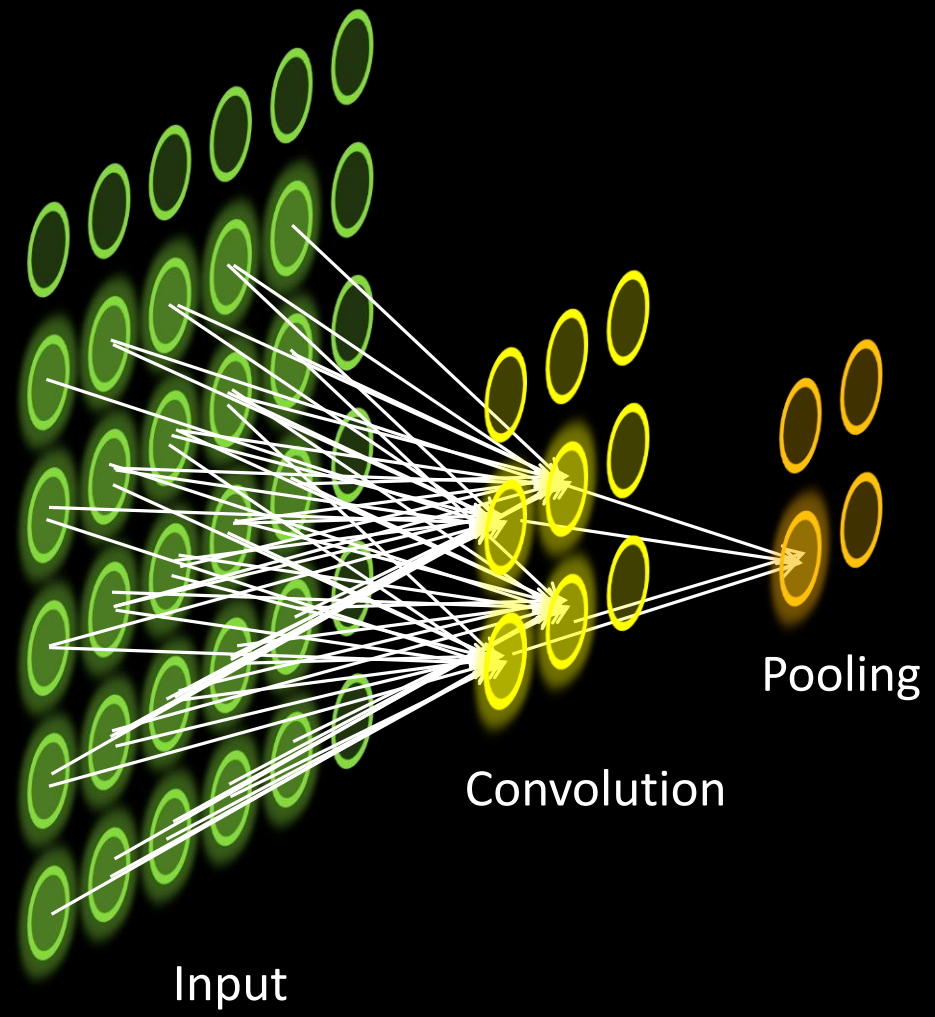
Standard Neural Architectures: Convolutional



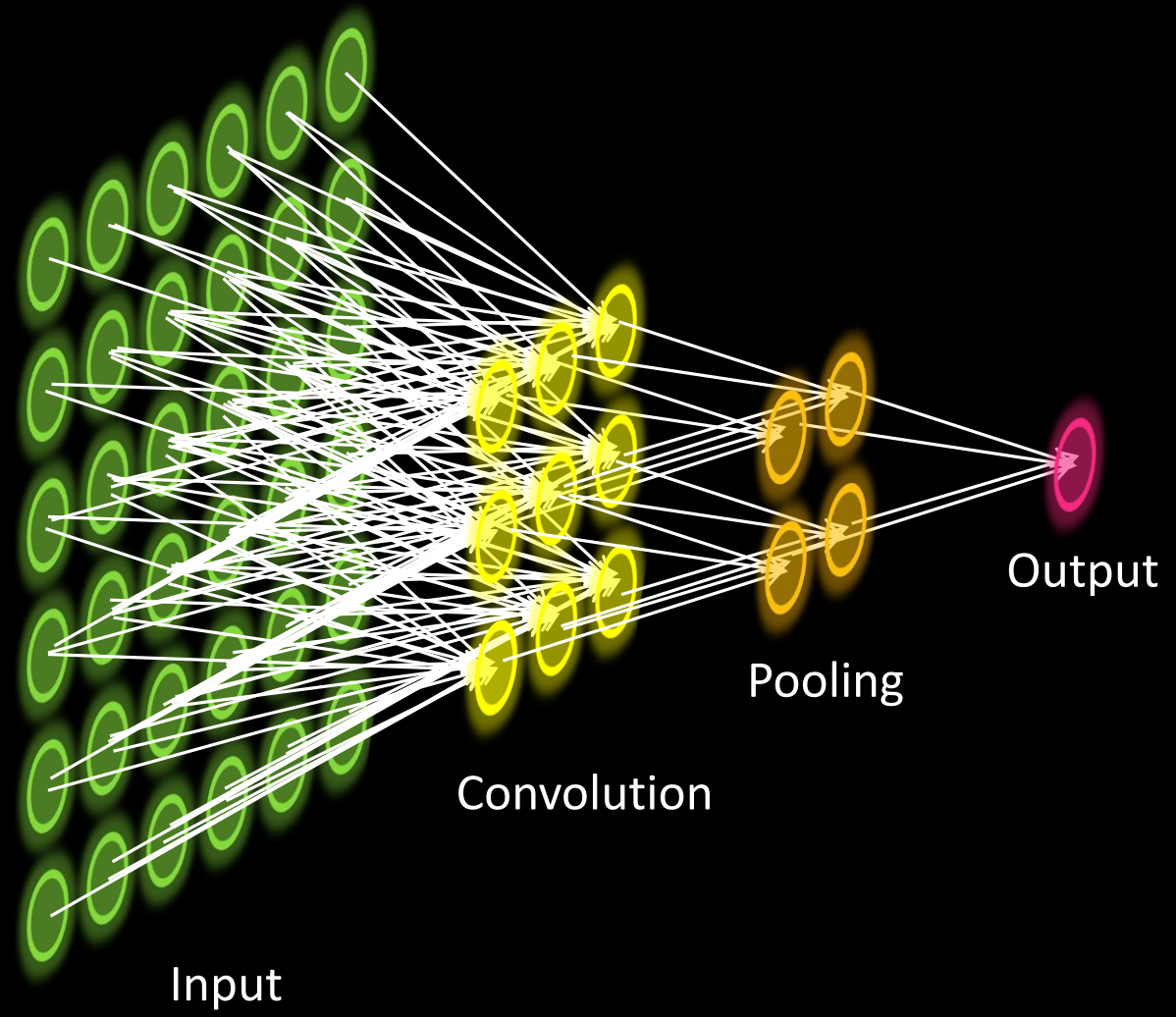
Standard Neural Architectures: Convolutional



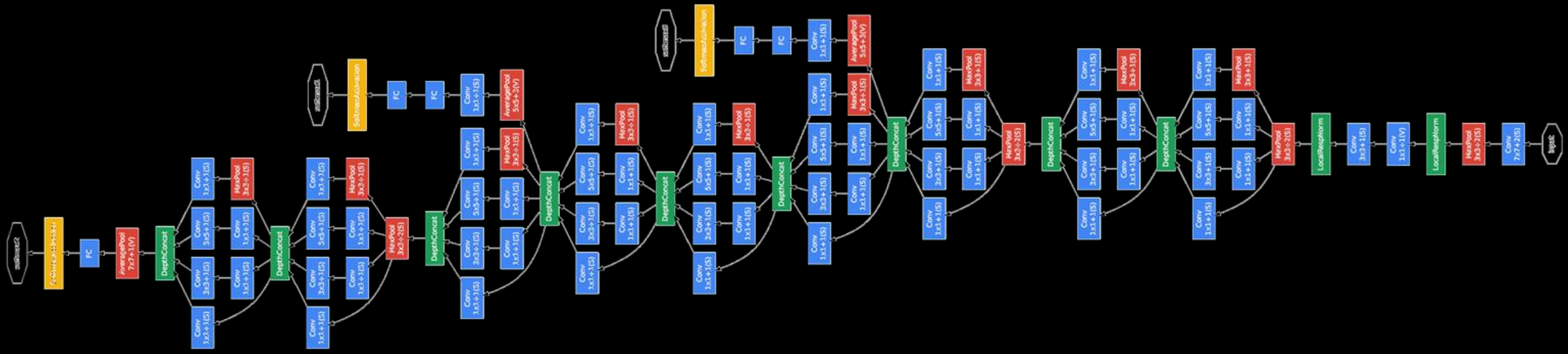
Standard Neural Architectures: Convolutional



Standard Neural Architectures: Convolutional



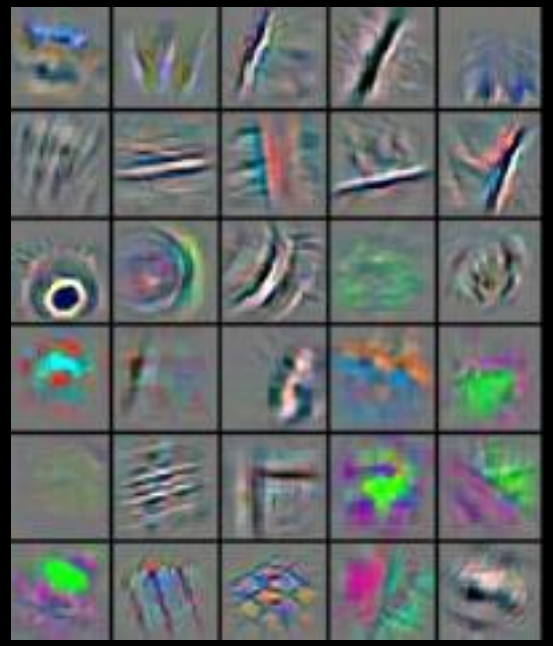
Standard Neural Architectures: Convolutional



GoogleNet

Standard Neural Architectures: Convolutional

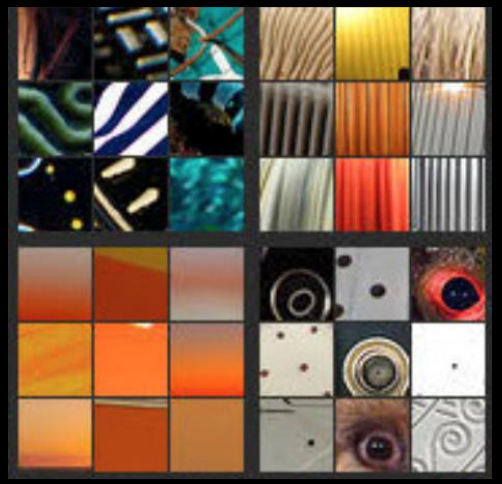
Layer 1



Layer 2

Layer 3

Standard Neural Architectures: Convolutional



Application: Image Recognition

track cycling
cycling
track cycling
road bicycle racing
marathon
ultramarathon

ultramarathon
ultramarathon
half marathon
running
marathon
inline speed skating

heptathlon
heptathlon
decathlon
hurdles
pentathlon
sprint (running)

bikejoring
mushing
bikejoring
harness racing
skijoring
carting

longboarding
longboarding
aggressive inline skating
freestyle scootering
freeboard (skateboard)
sandboarding

ultimate (sport)
ultimate (sport)
hurling
flag football
association football
rugby sevens

demolition derby
demolition derby
monster truck
mud bogging
motocross
grand prix motorcycle racing

telemark skiing
snowboarding
telemark skiing
nordic skiing
ski touring
skijoring

whitewater kayaking
whitewater kayaking
rafting
kayaking
canoeing
adventure racing

arena football
indoor american football
arena football
canadian football
american football
women's lacrosse

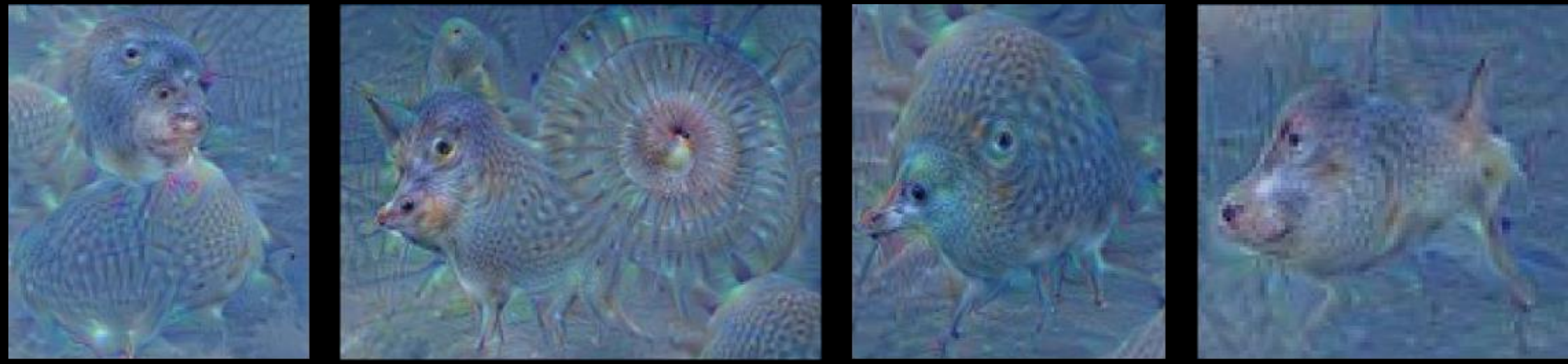
reining
barrel racing
rodeo
reining
cowboy action shooting
bull riding

eight-ball
nine-ball
blackball (pool)
trick shot
eight-ball
straight pool

Application: Deep Dream

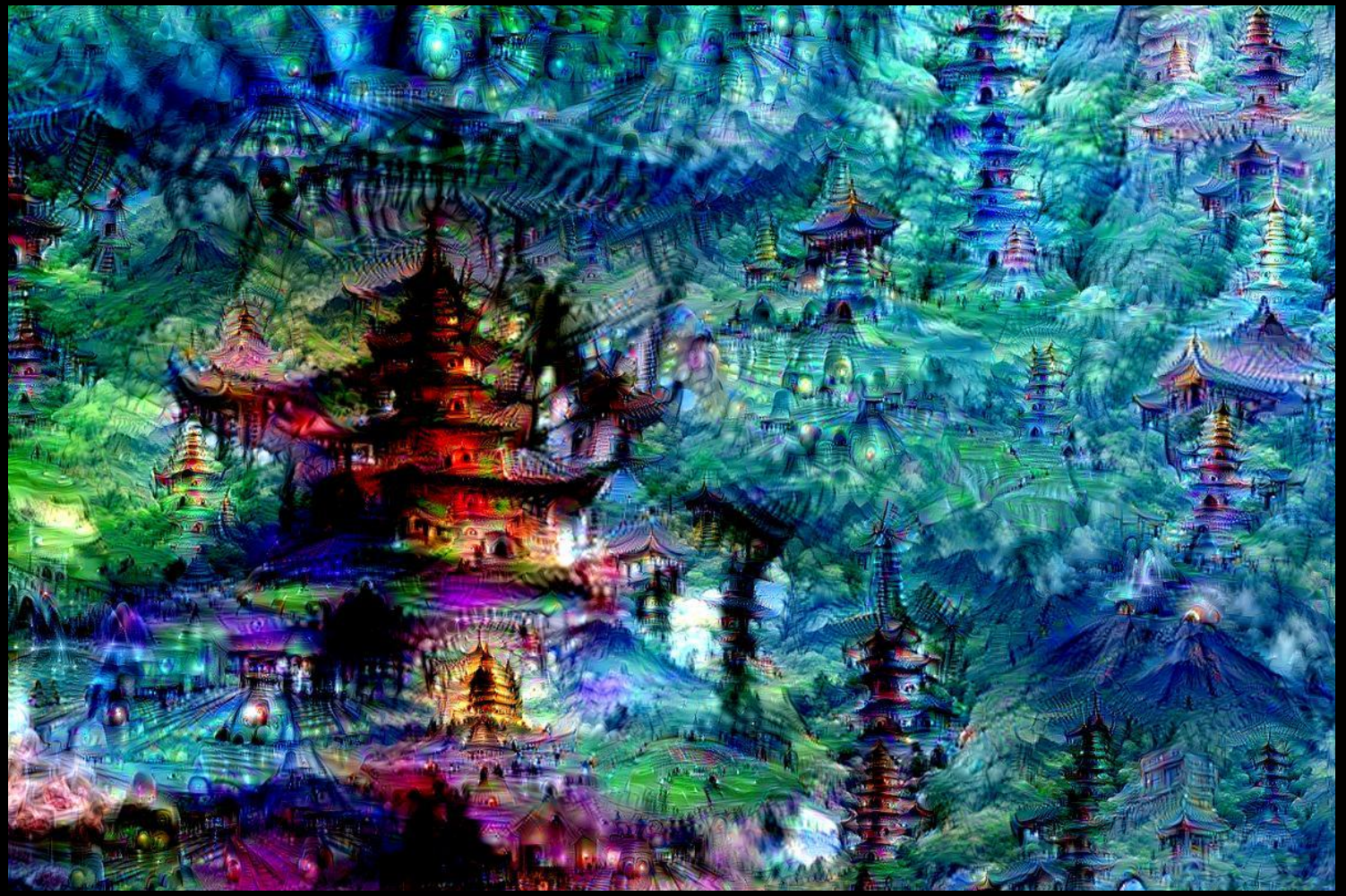
Application: Deep Dream

Application: Deep Dream

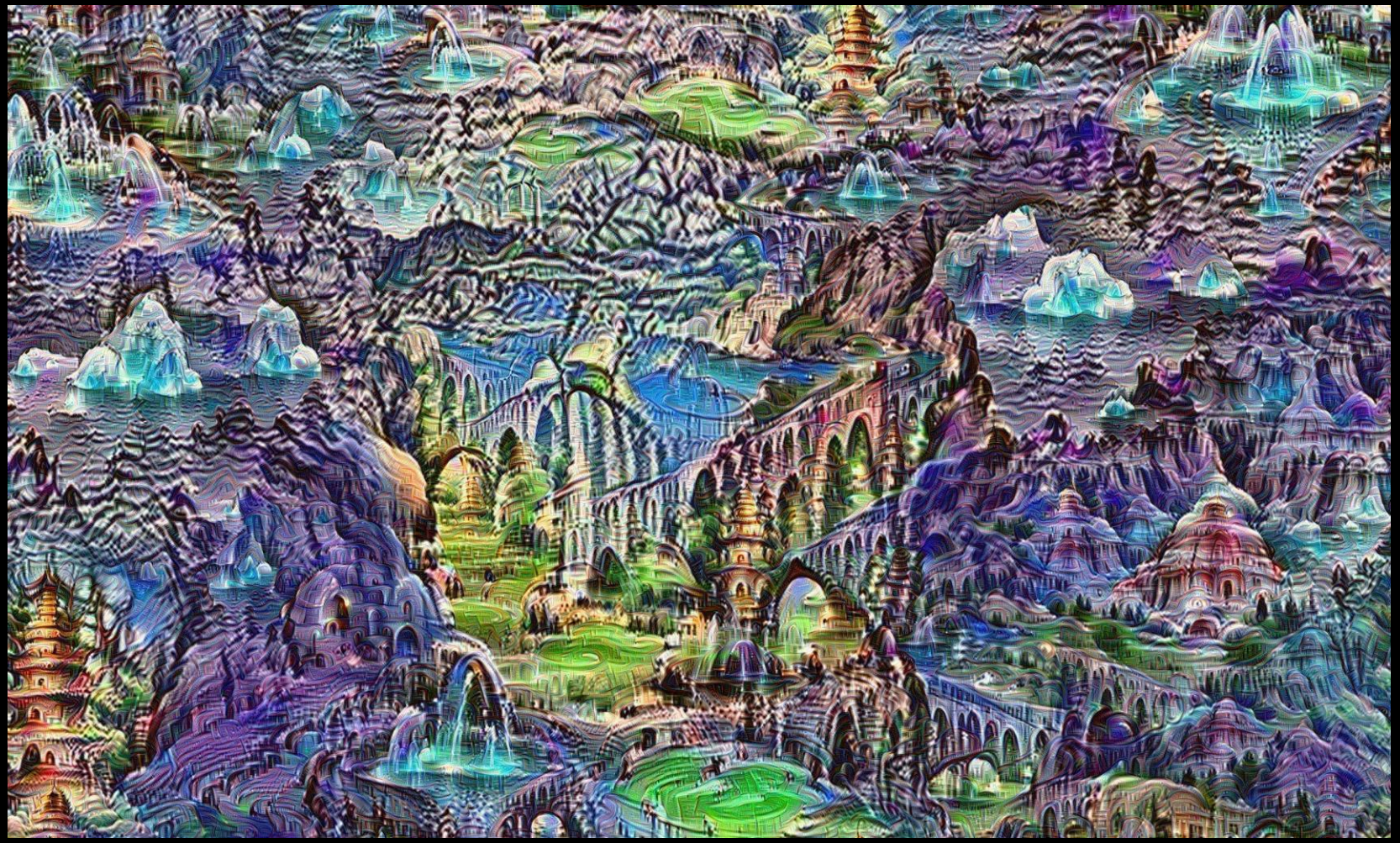


Application: Deep Dream

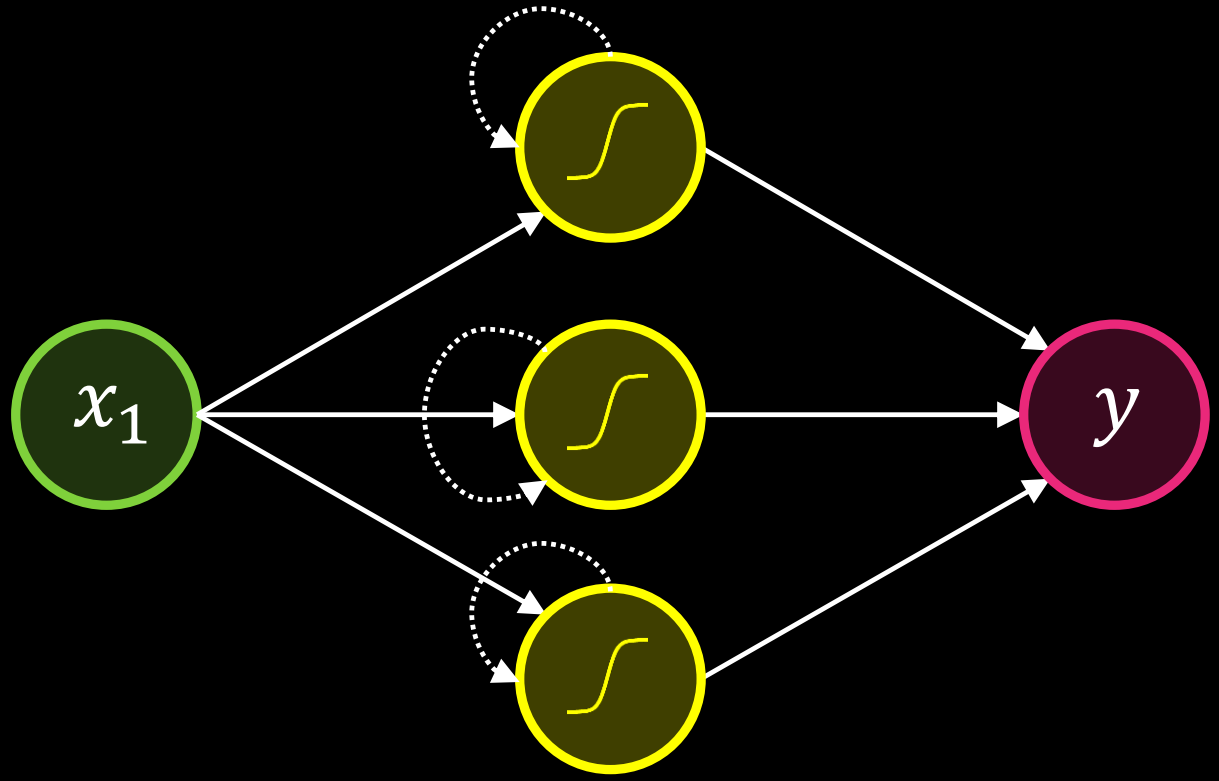
Application: Deep Dream



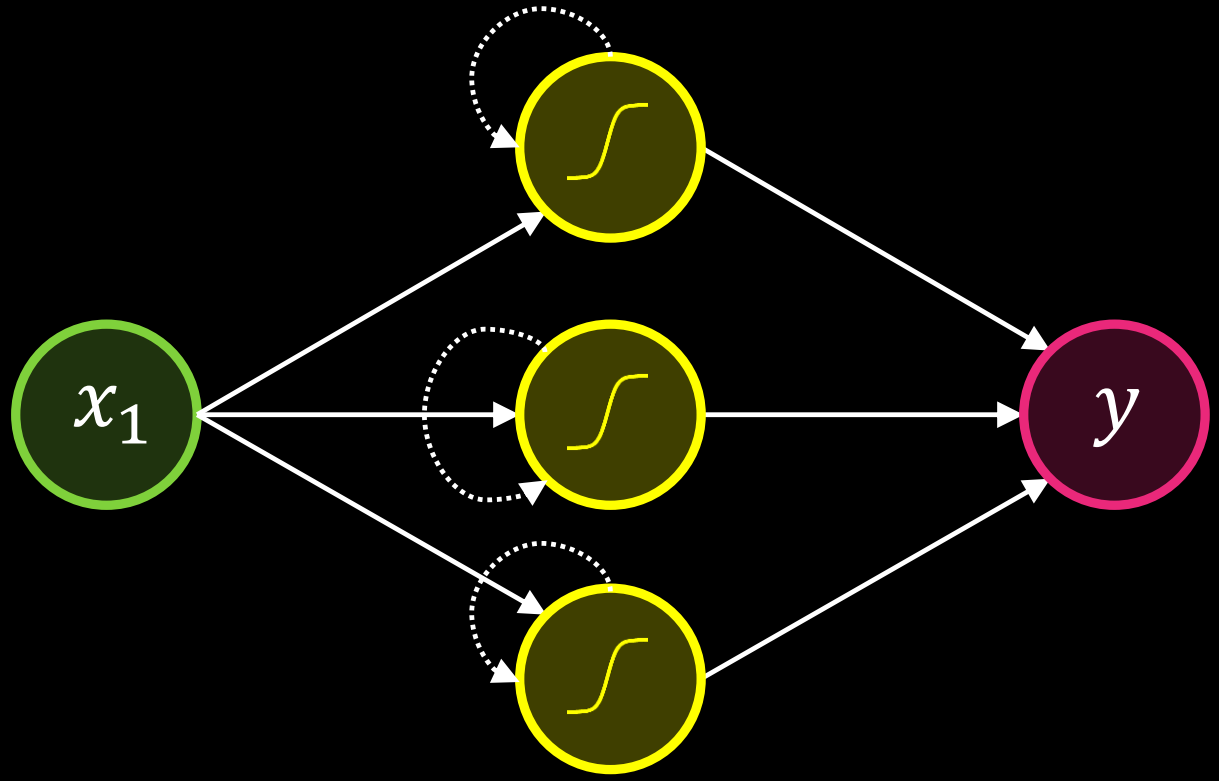
Application: Deep Dream



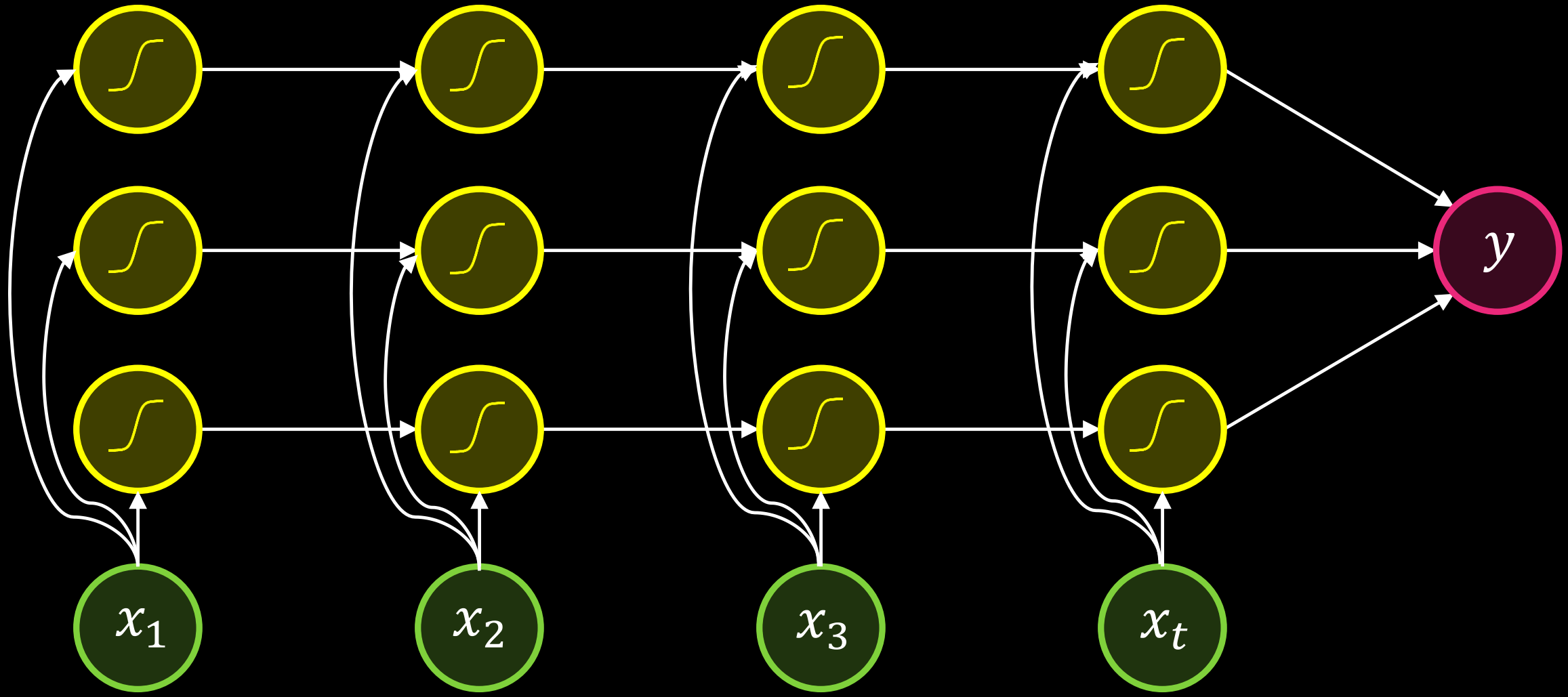
Standard Neural Architectures: Recurrent



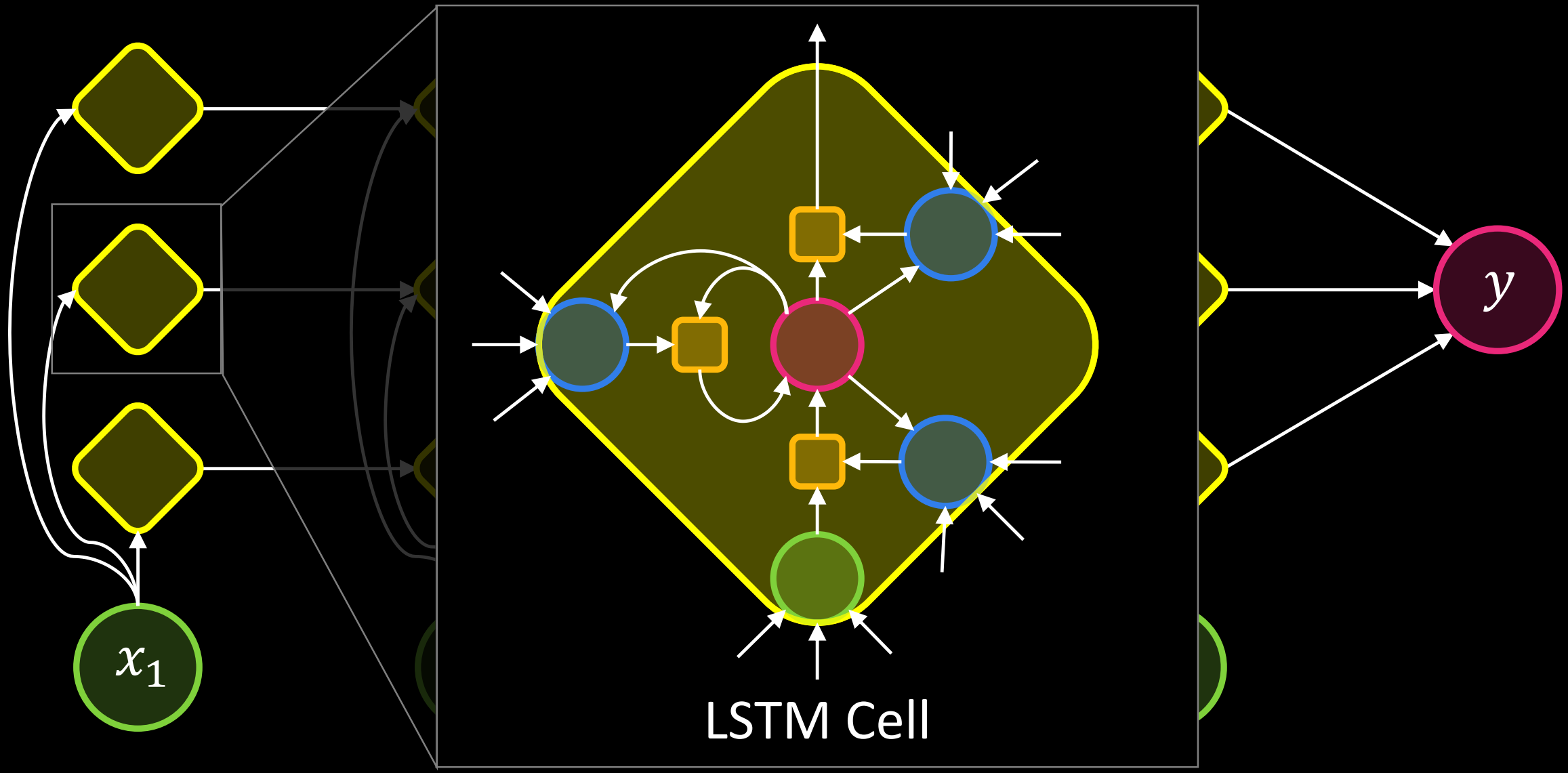
Standard Neural Architectures: Recurrent



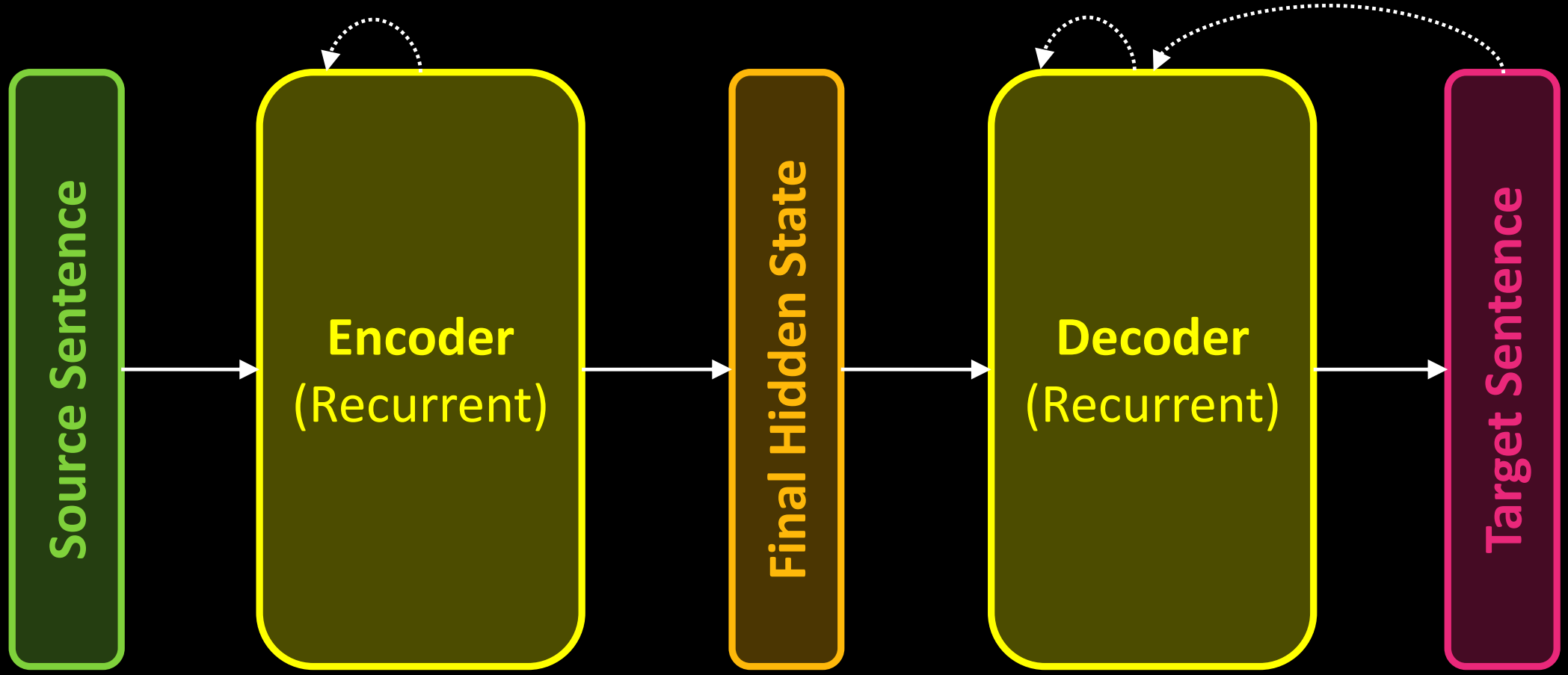
Standard Neural Architectures: Recurrent



Standard Neural Architectures: Recurrent



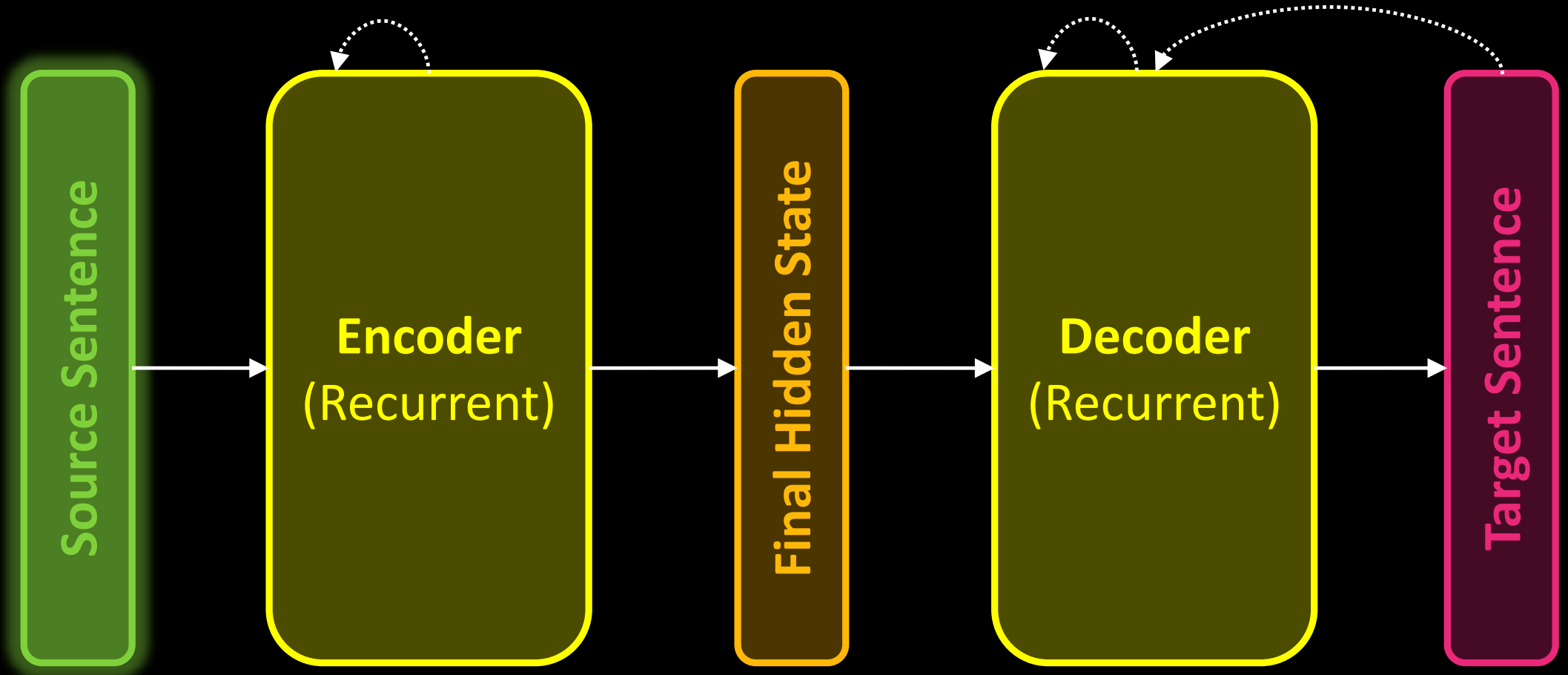
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

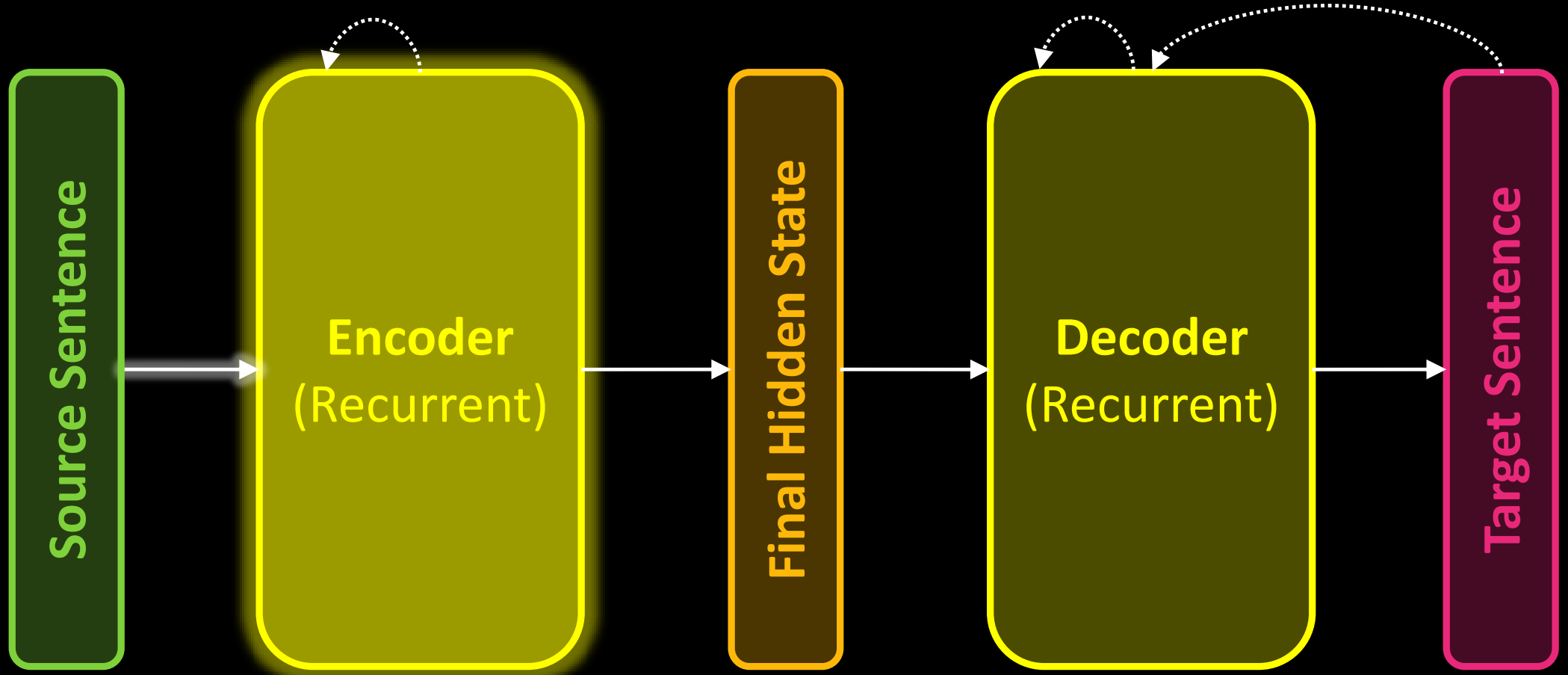
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

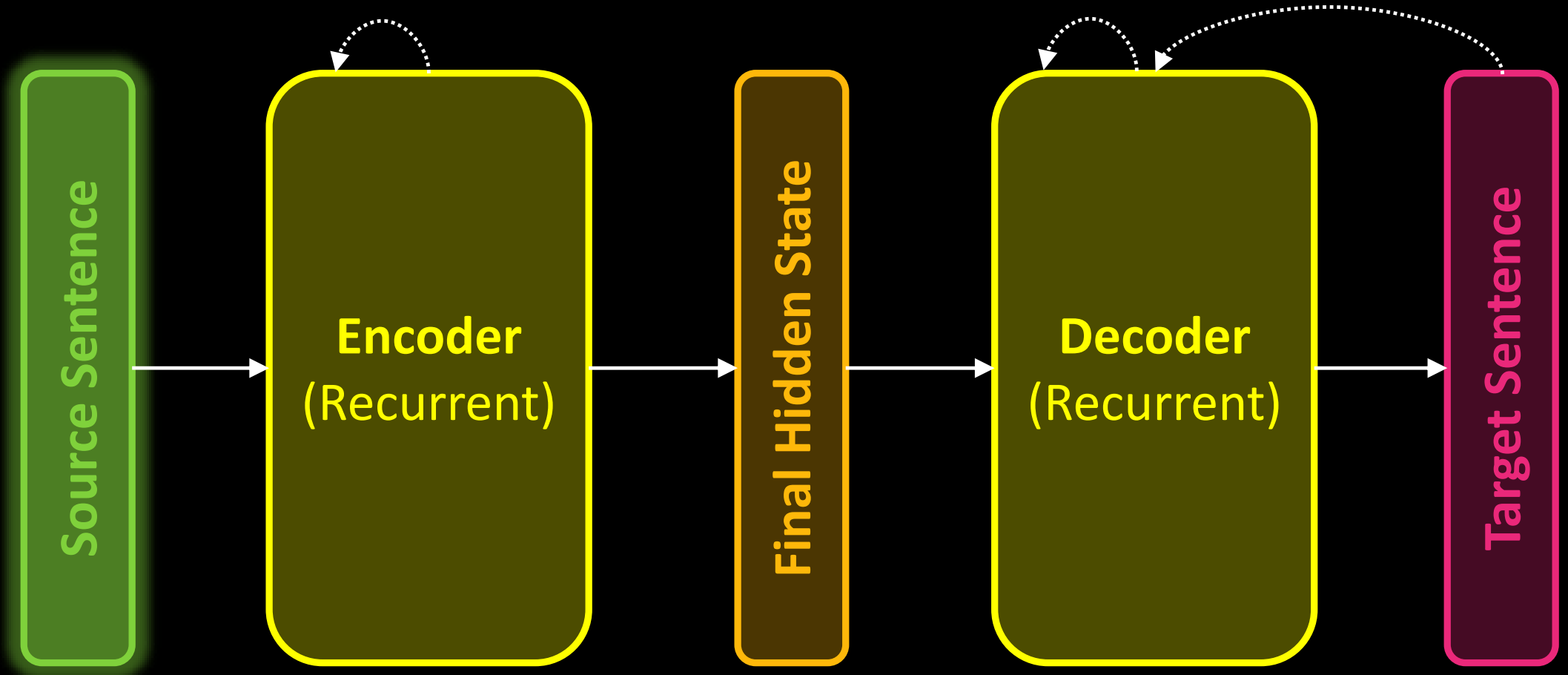
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

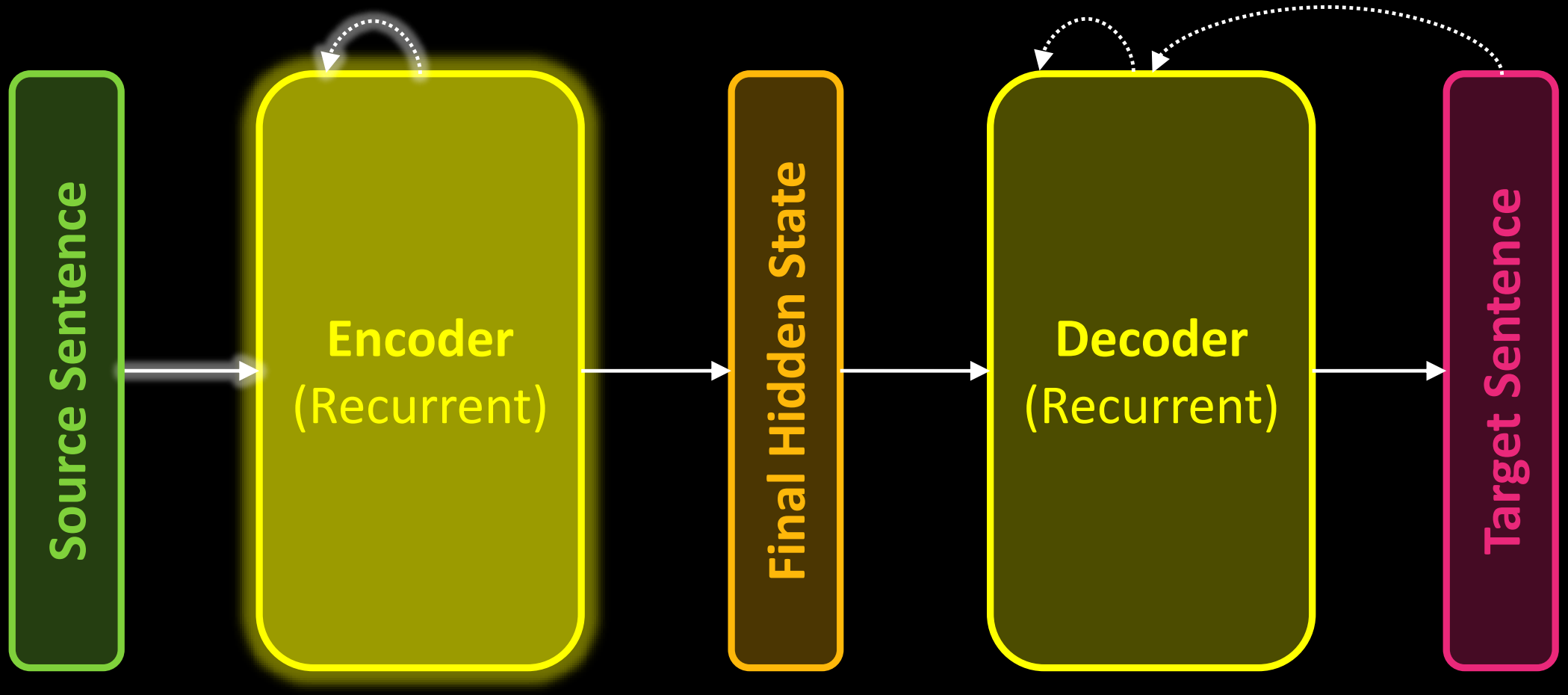
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

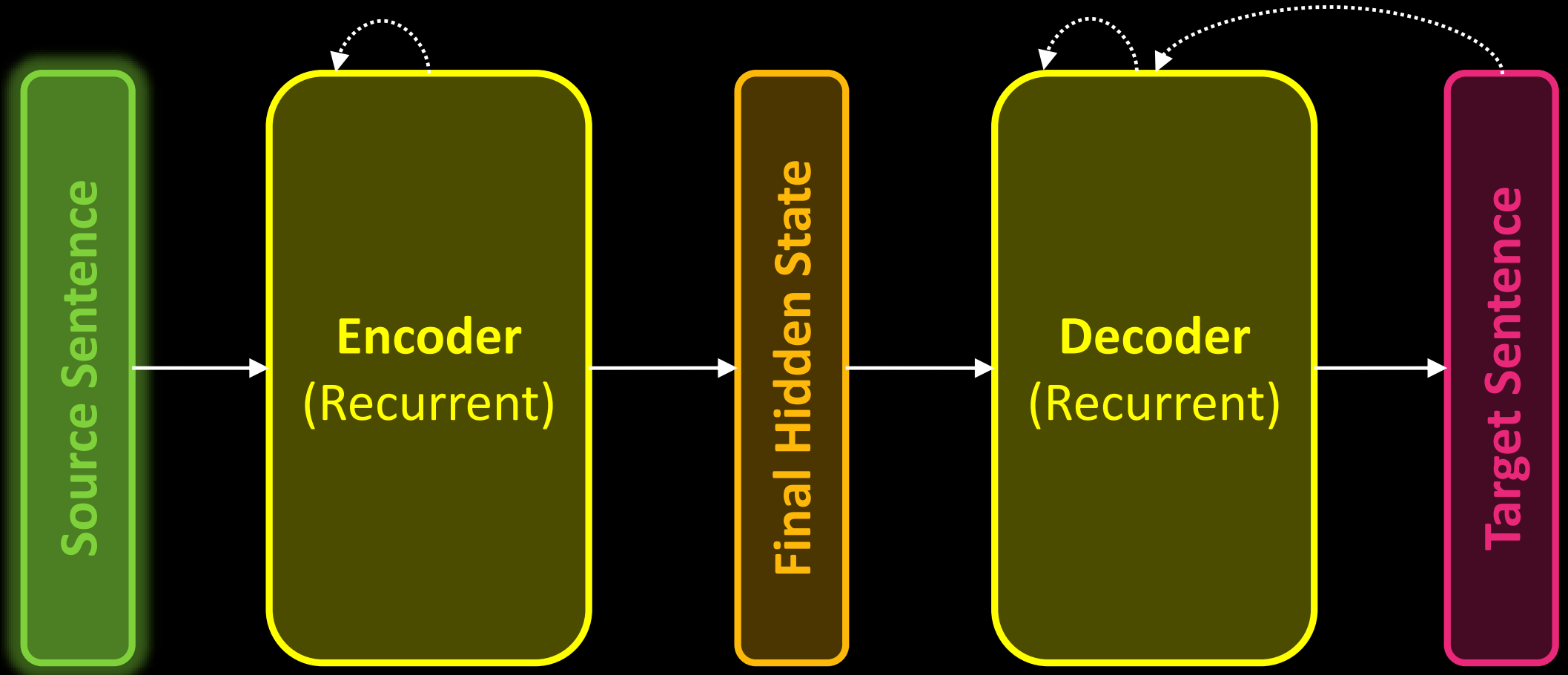
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

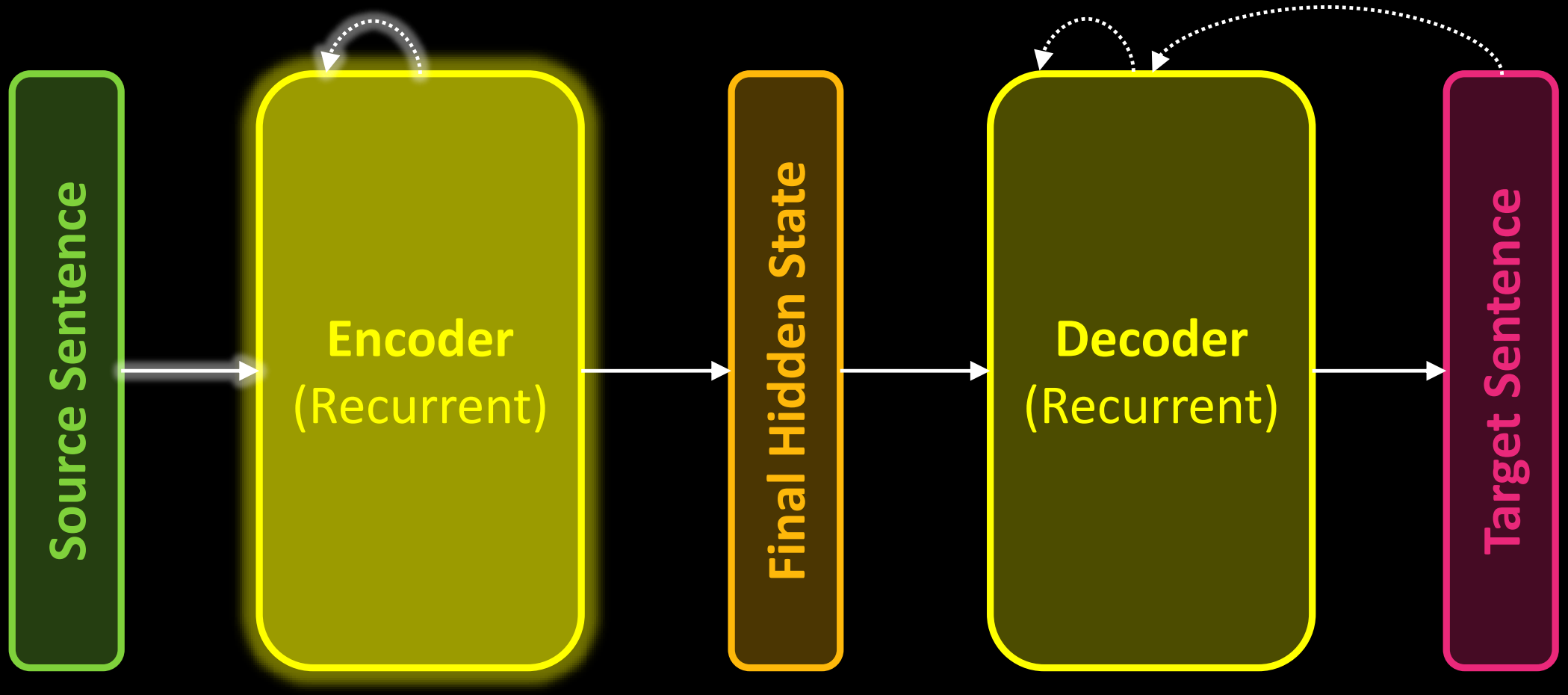
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

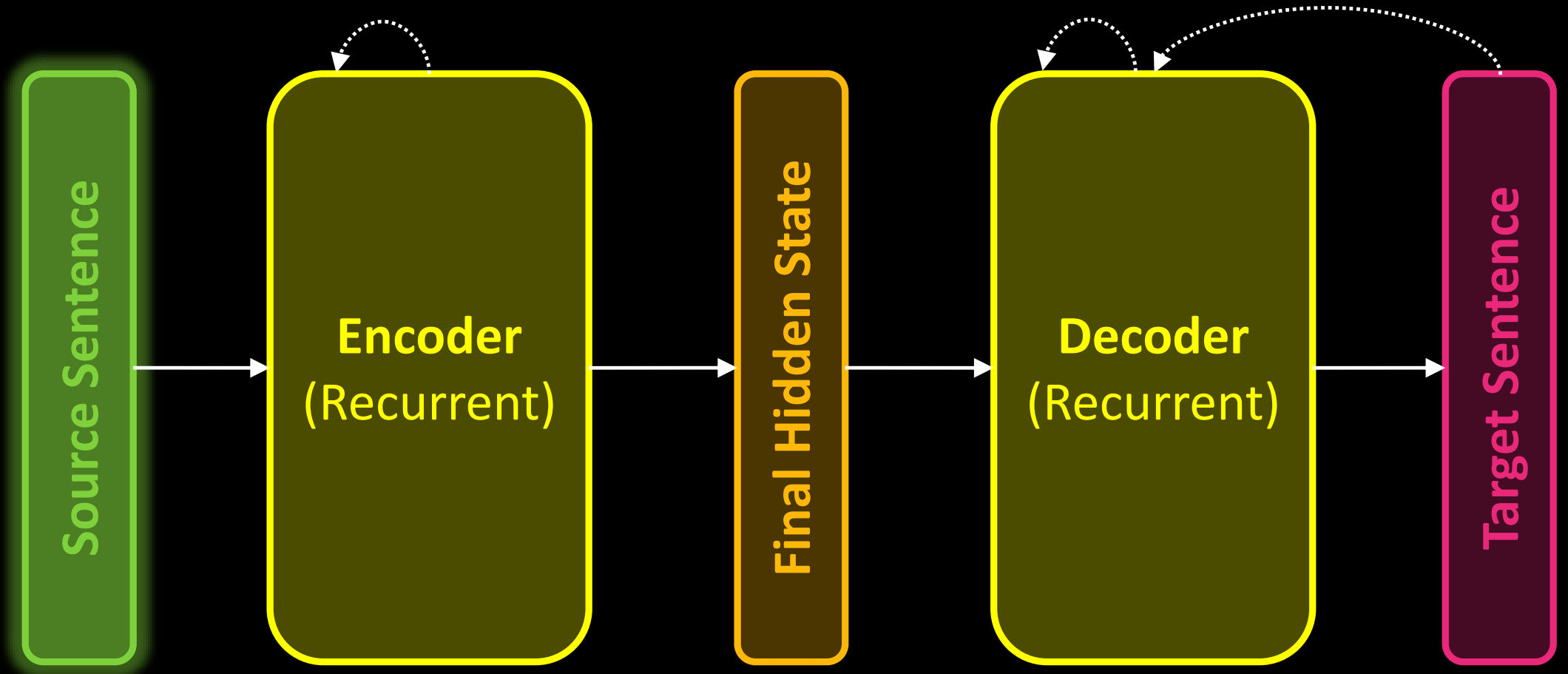
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

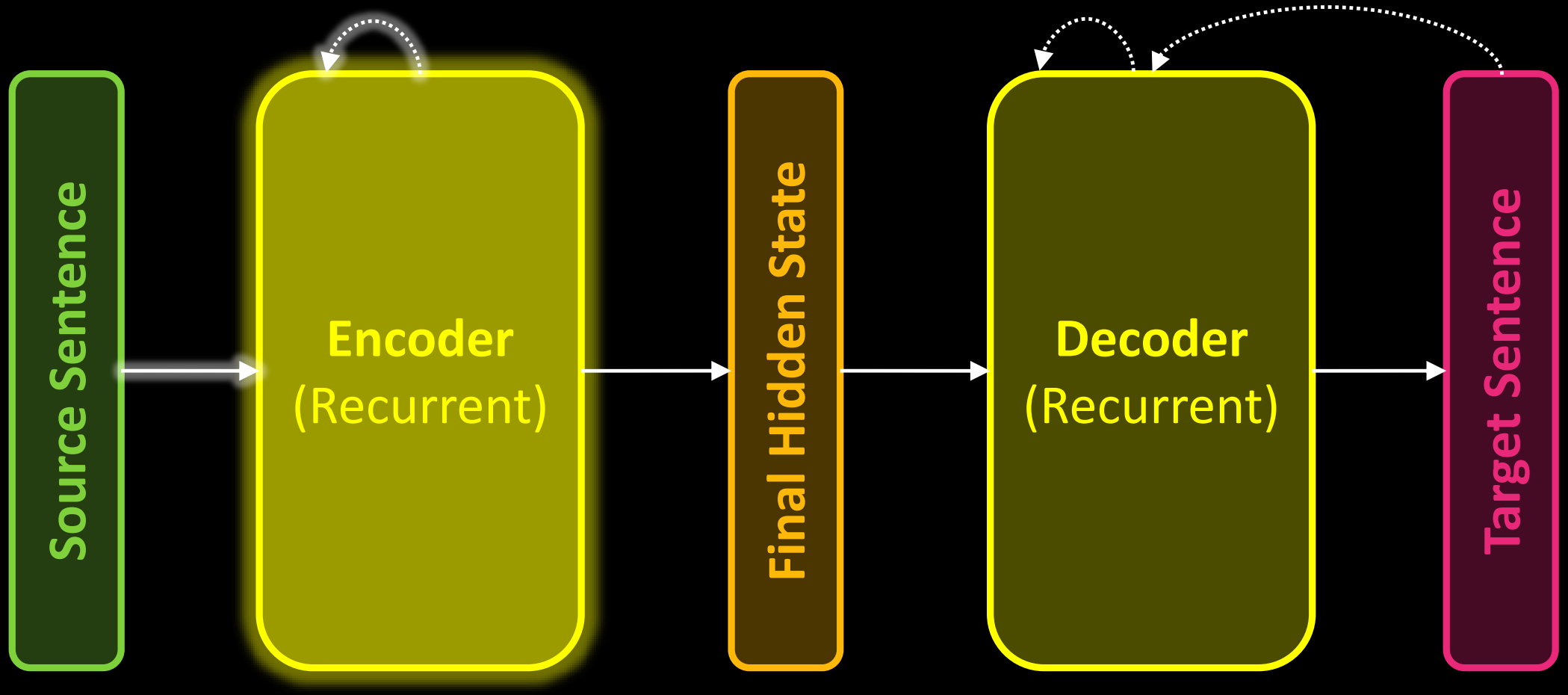
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

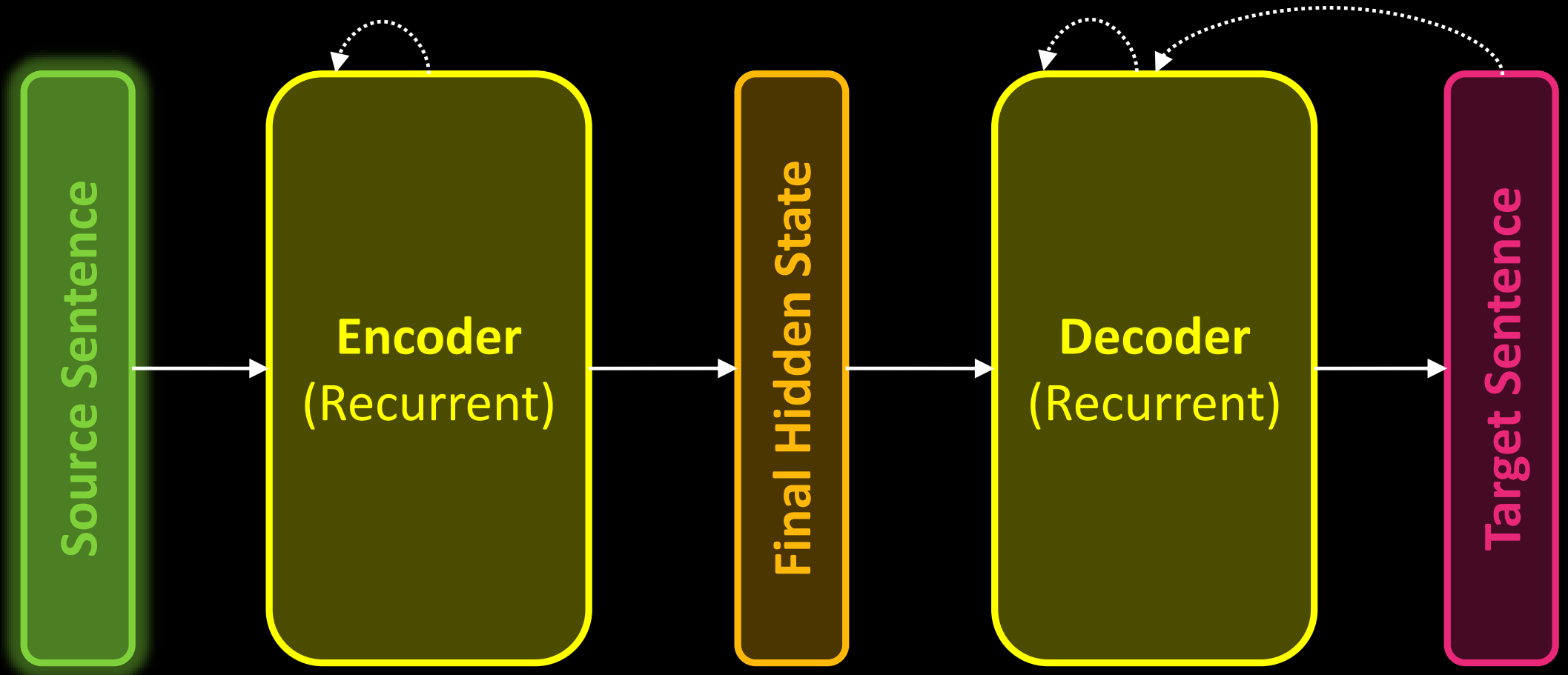
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

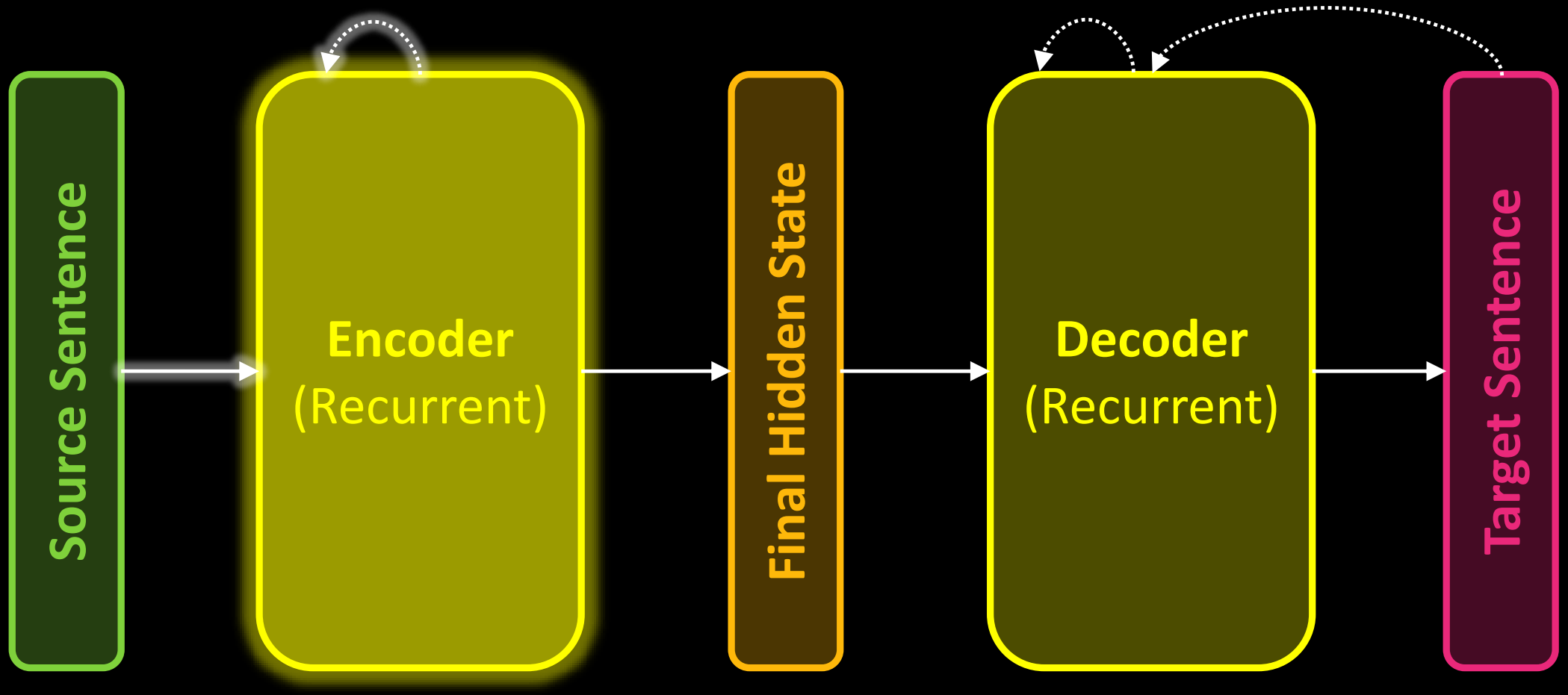
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

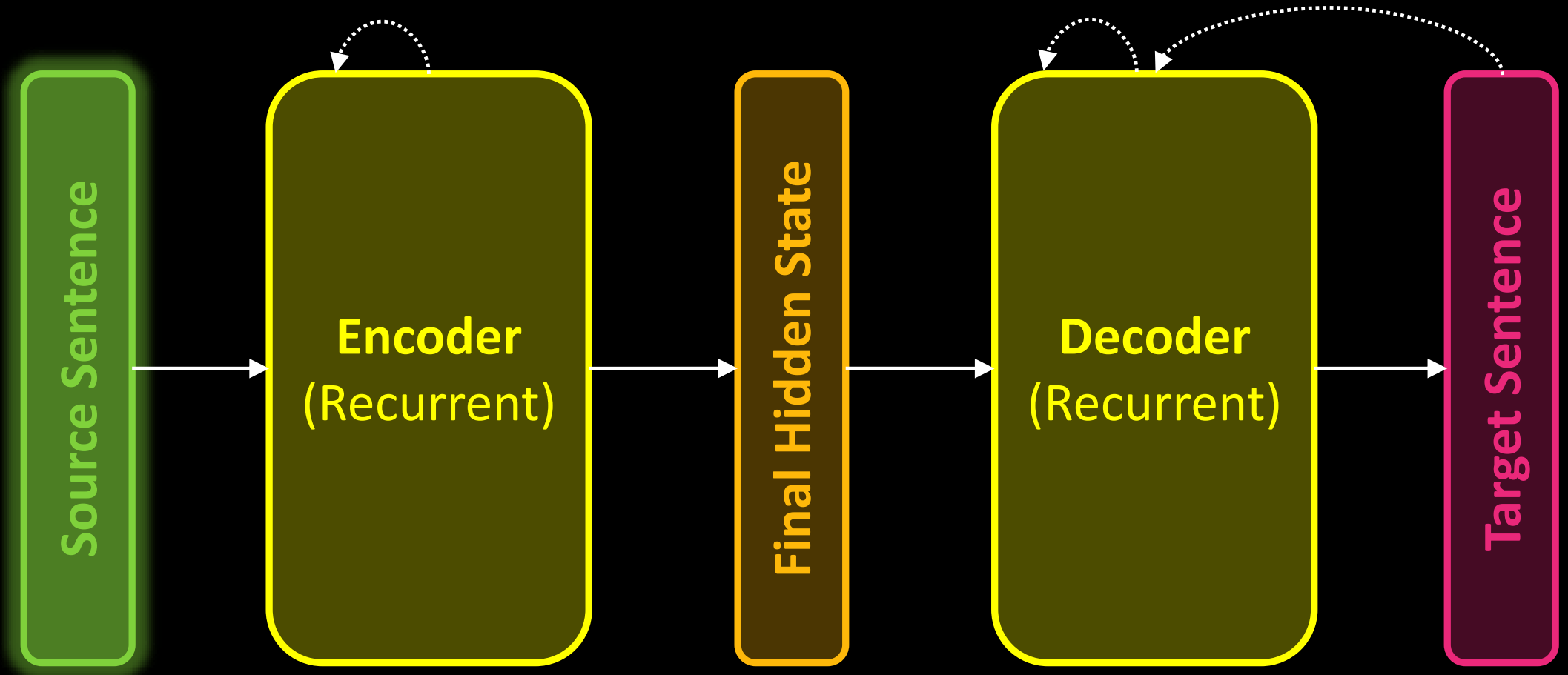
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

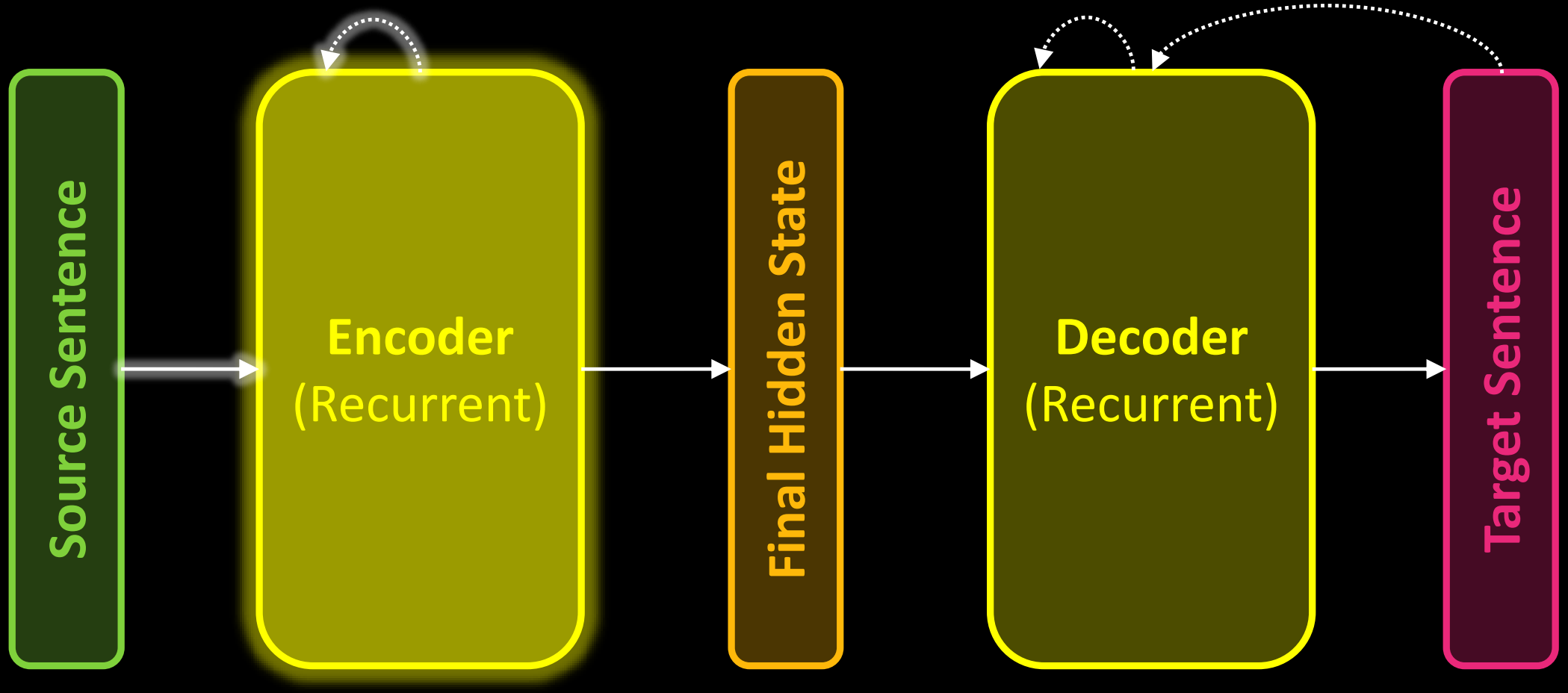
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

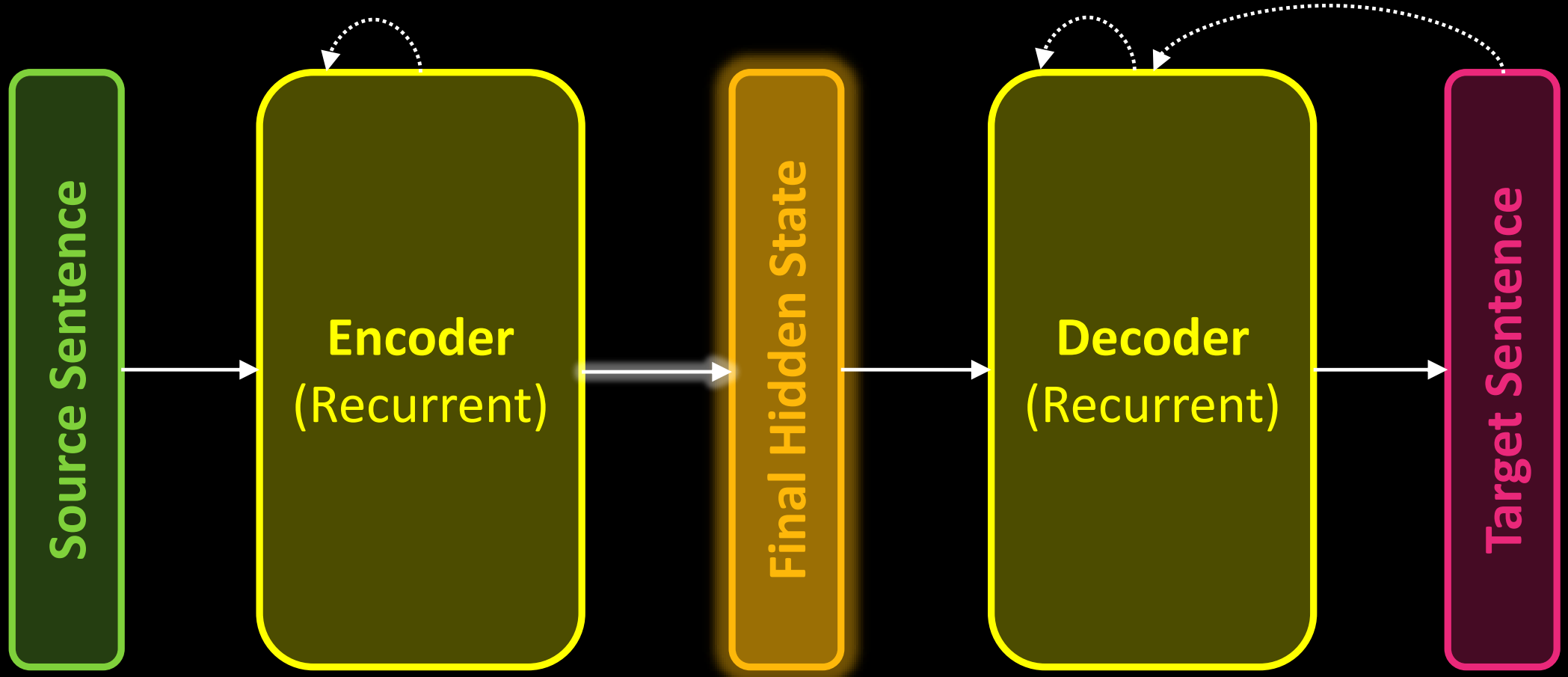
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

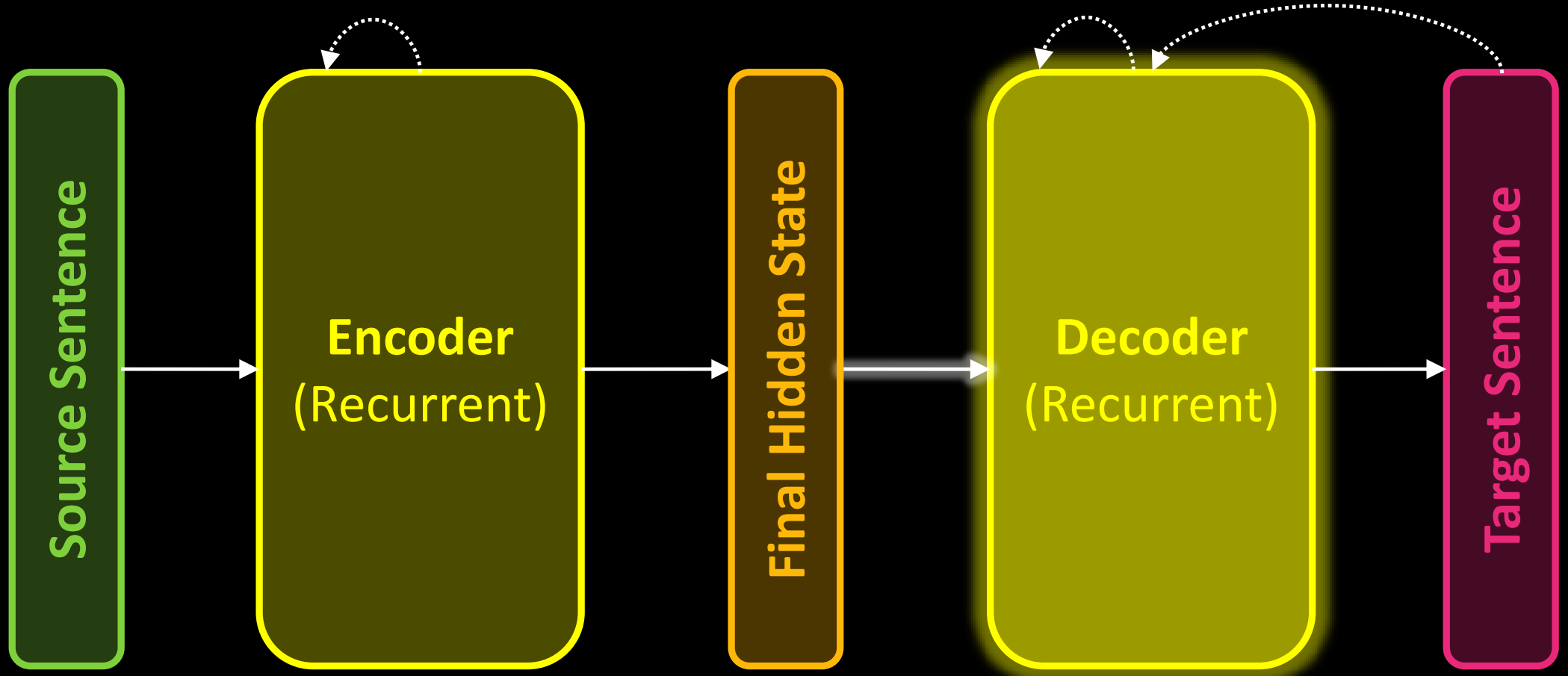
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

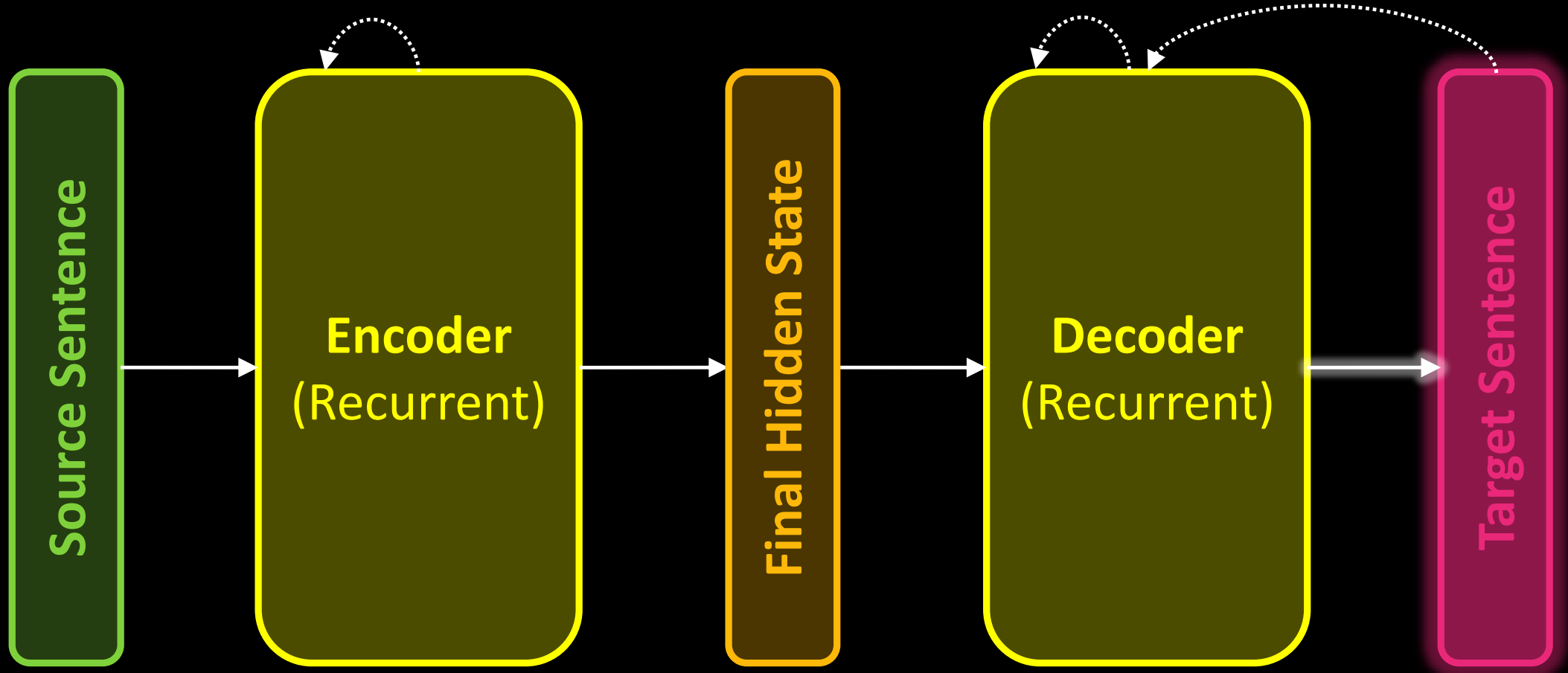
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

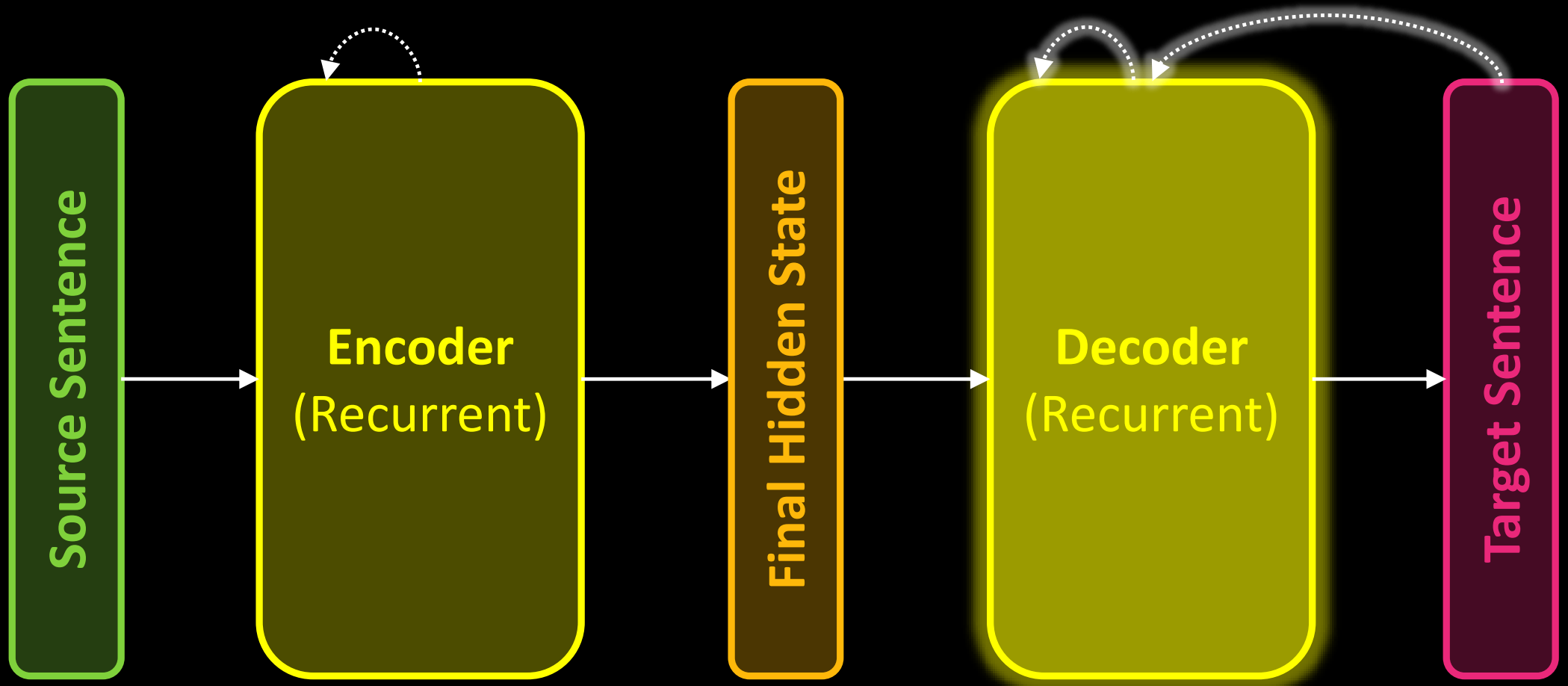
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

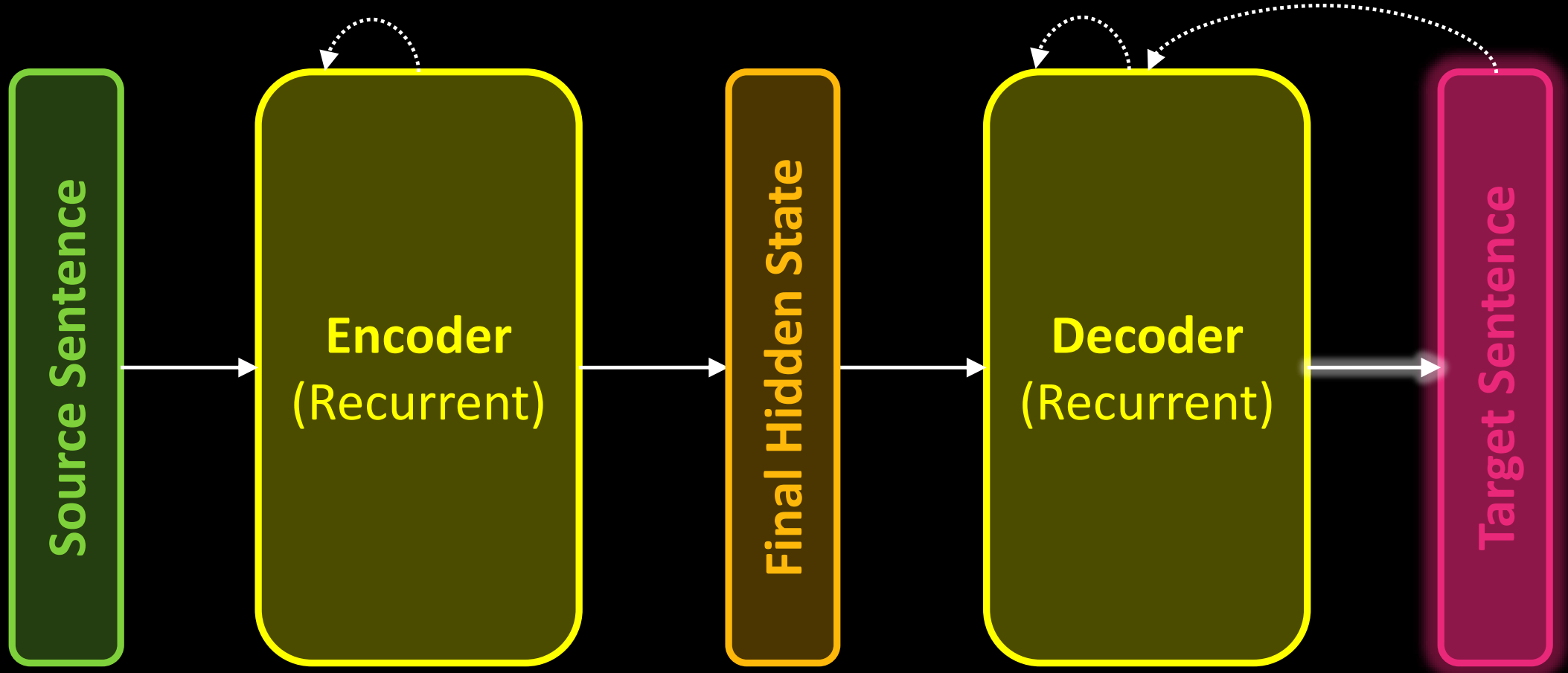
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

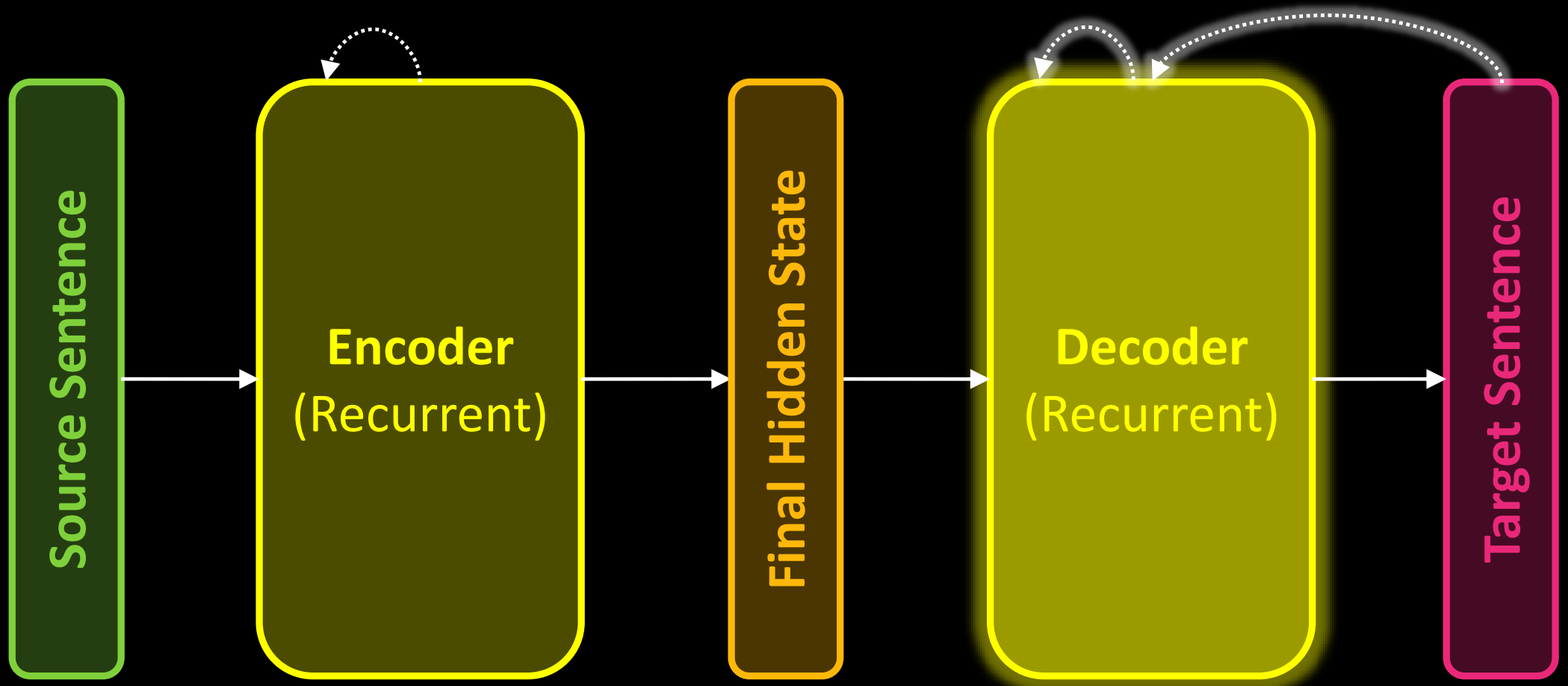
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

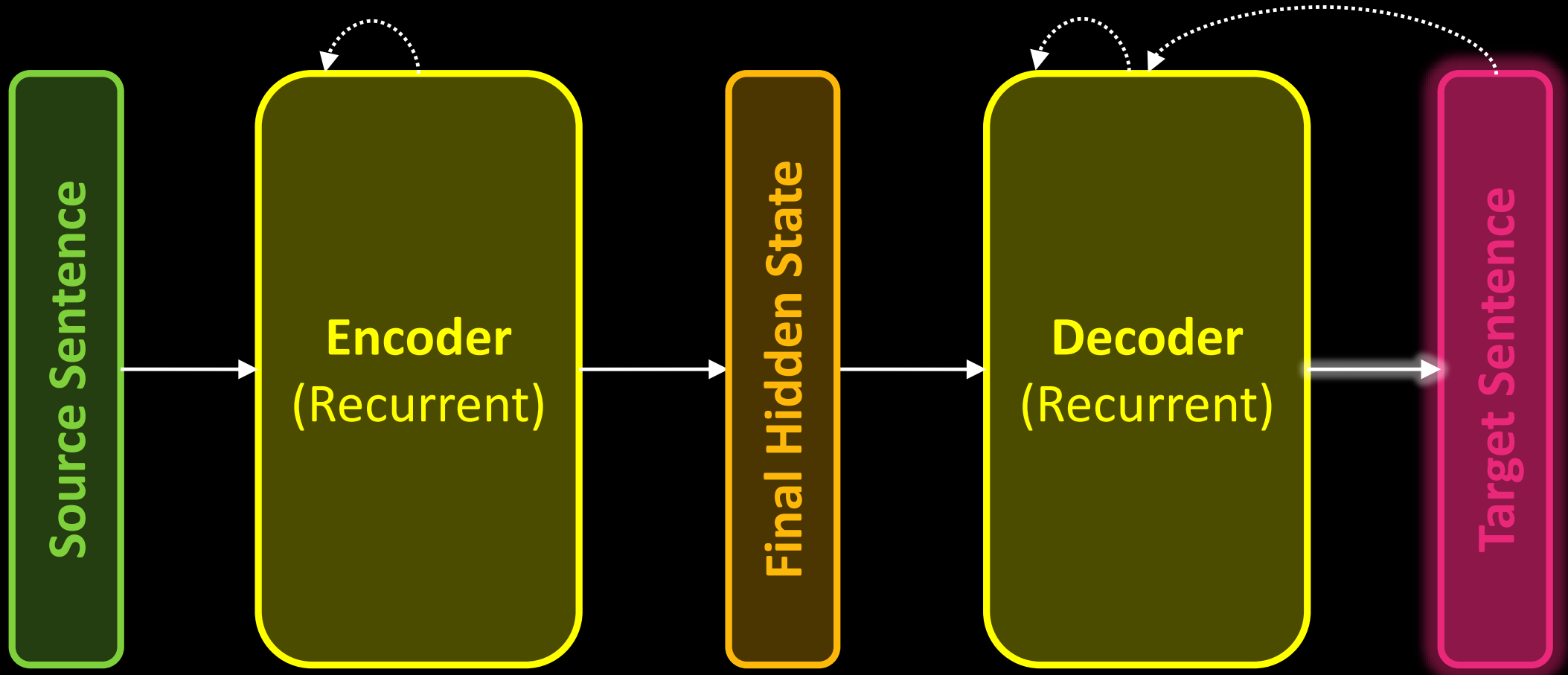
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

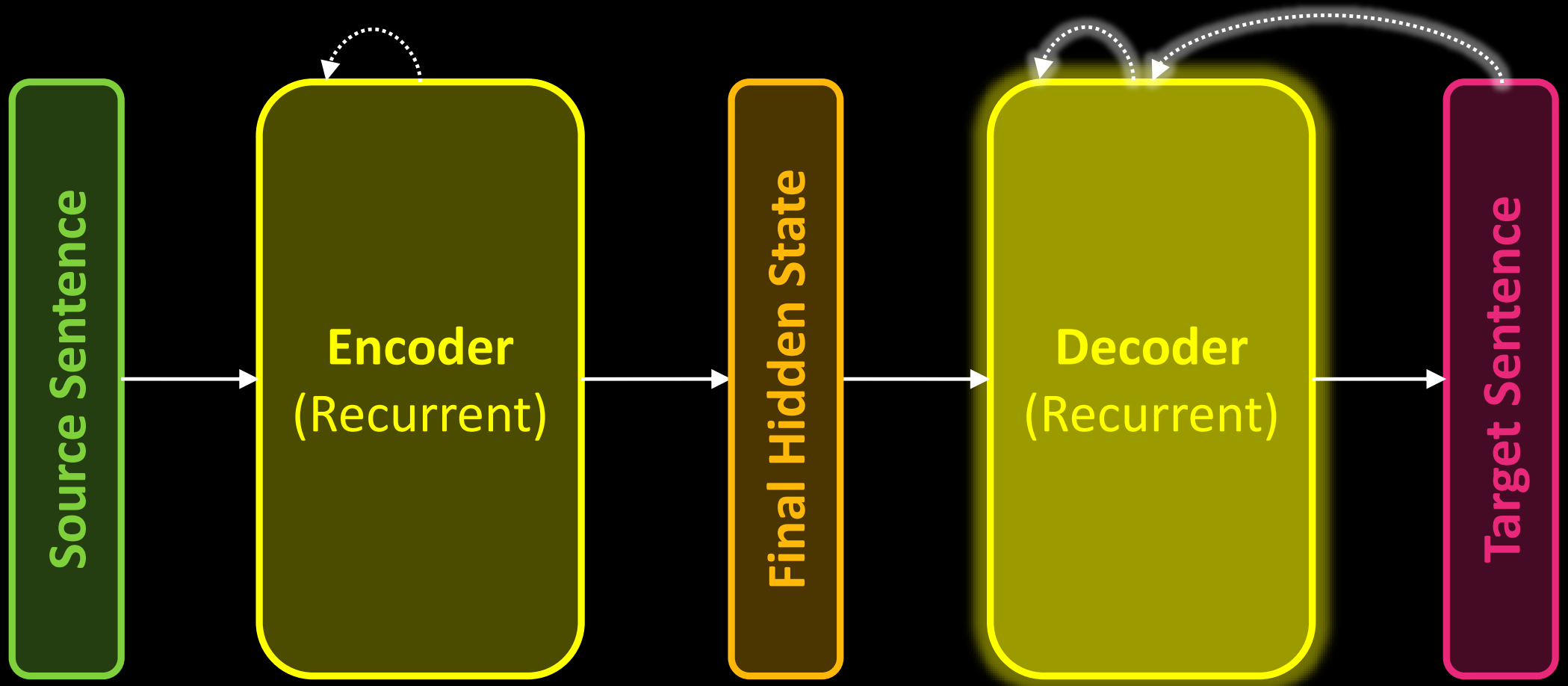
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

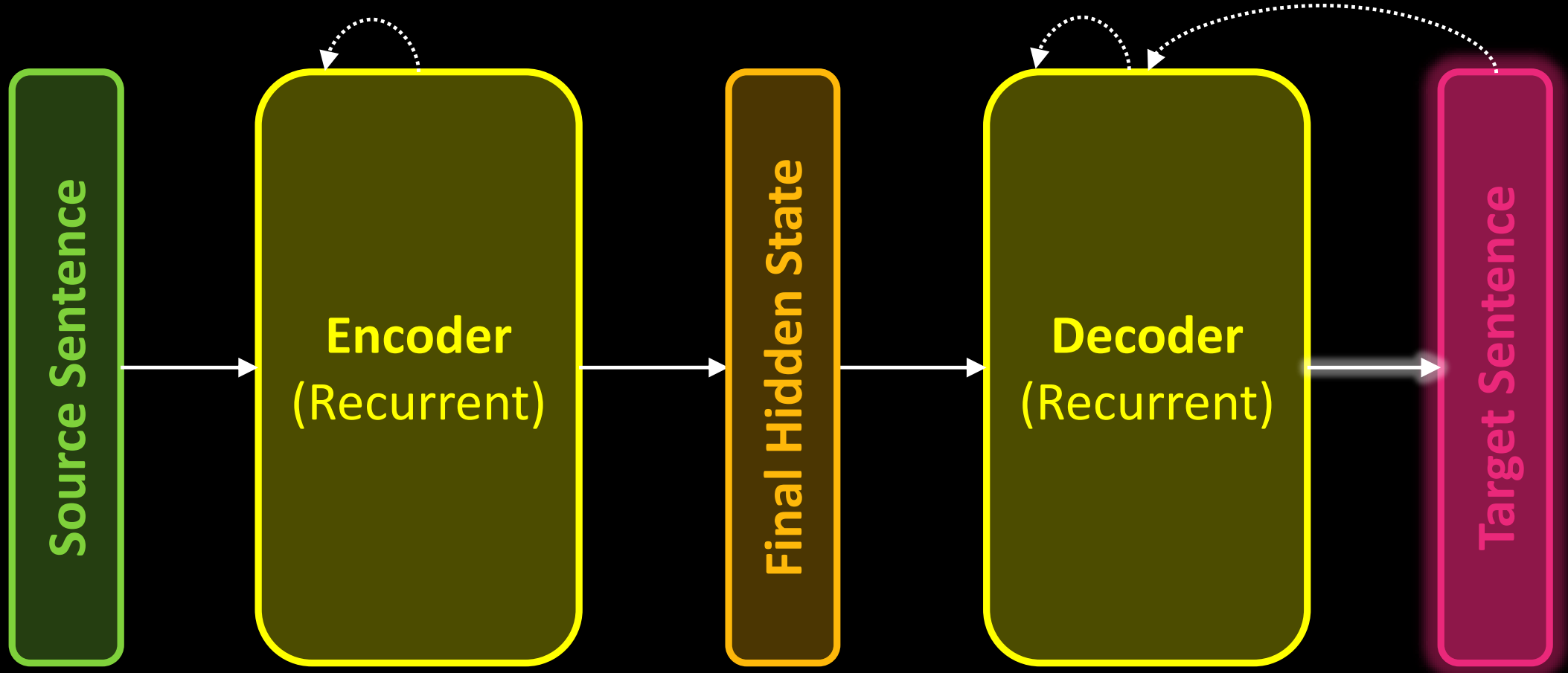
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

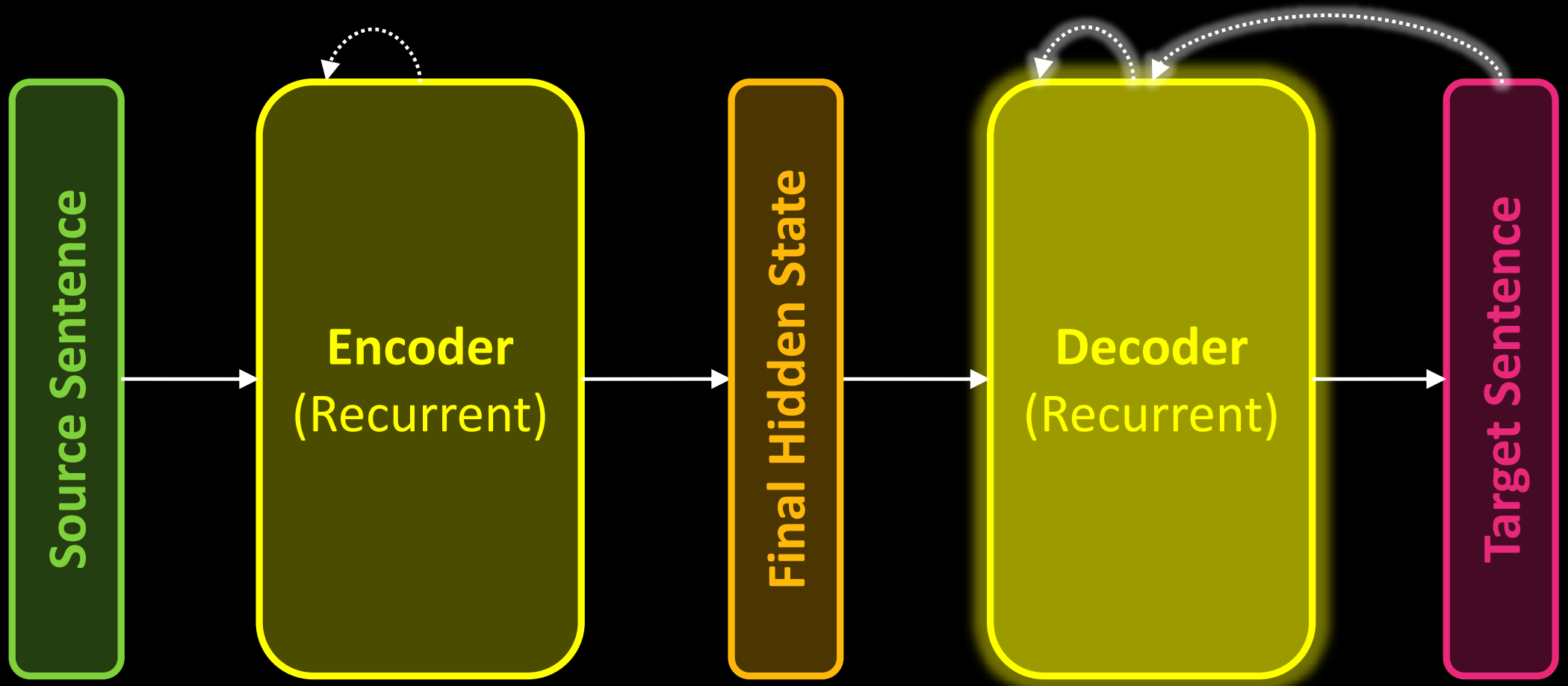
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

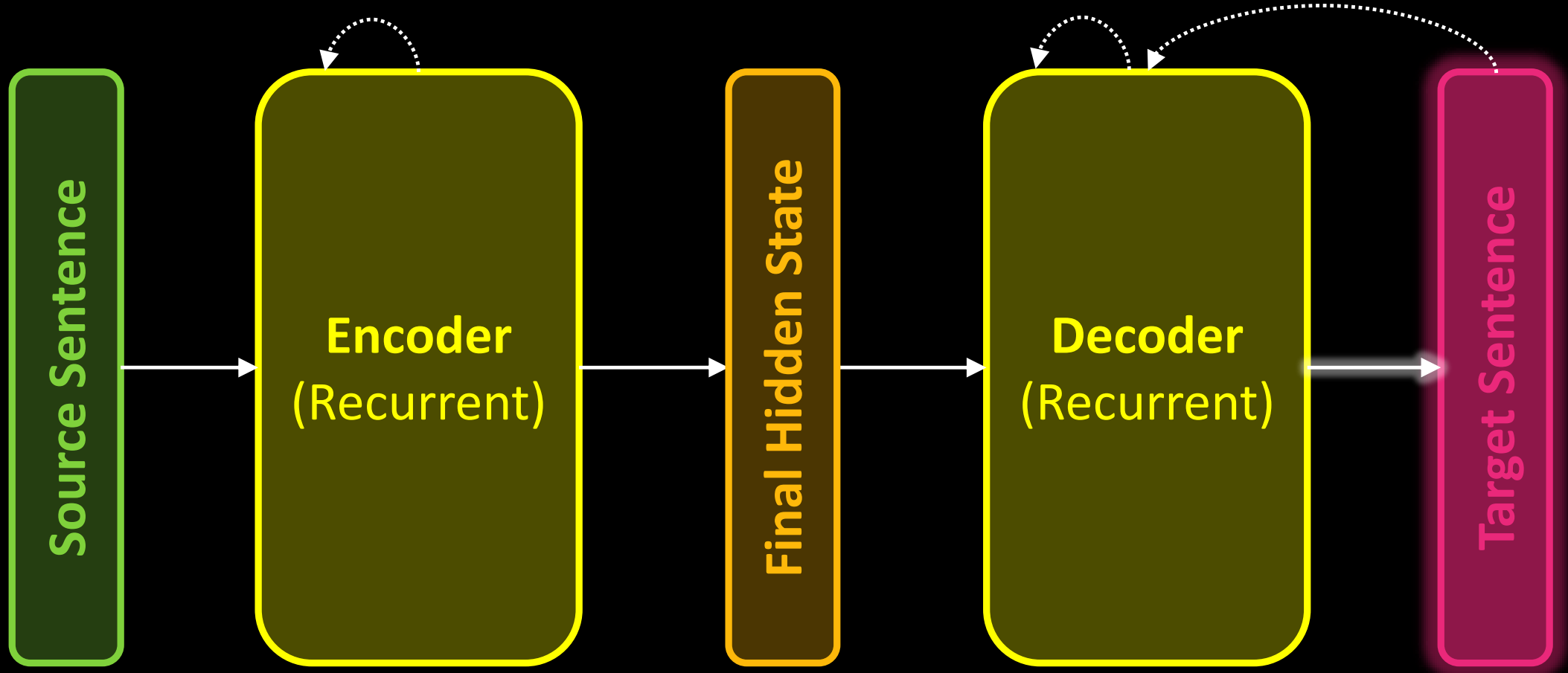
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

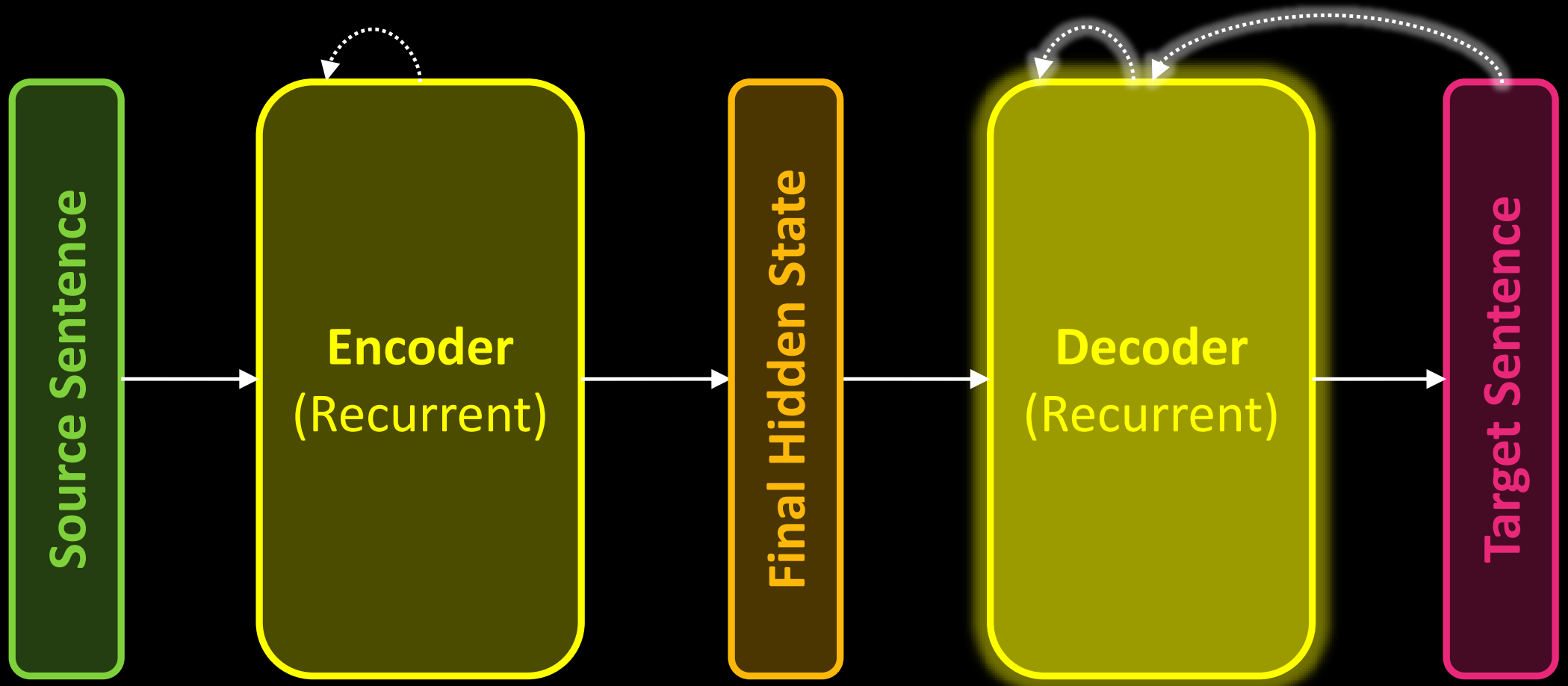
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

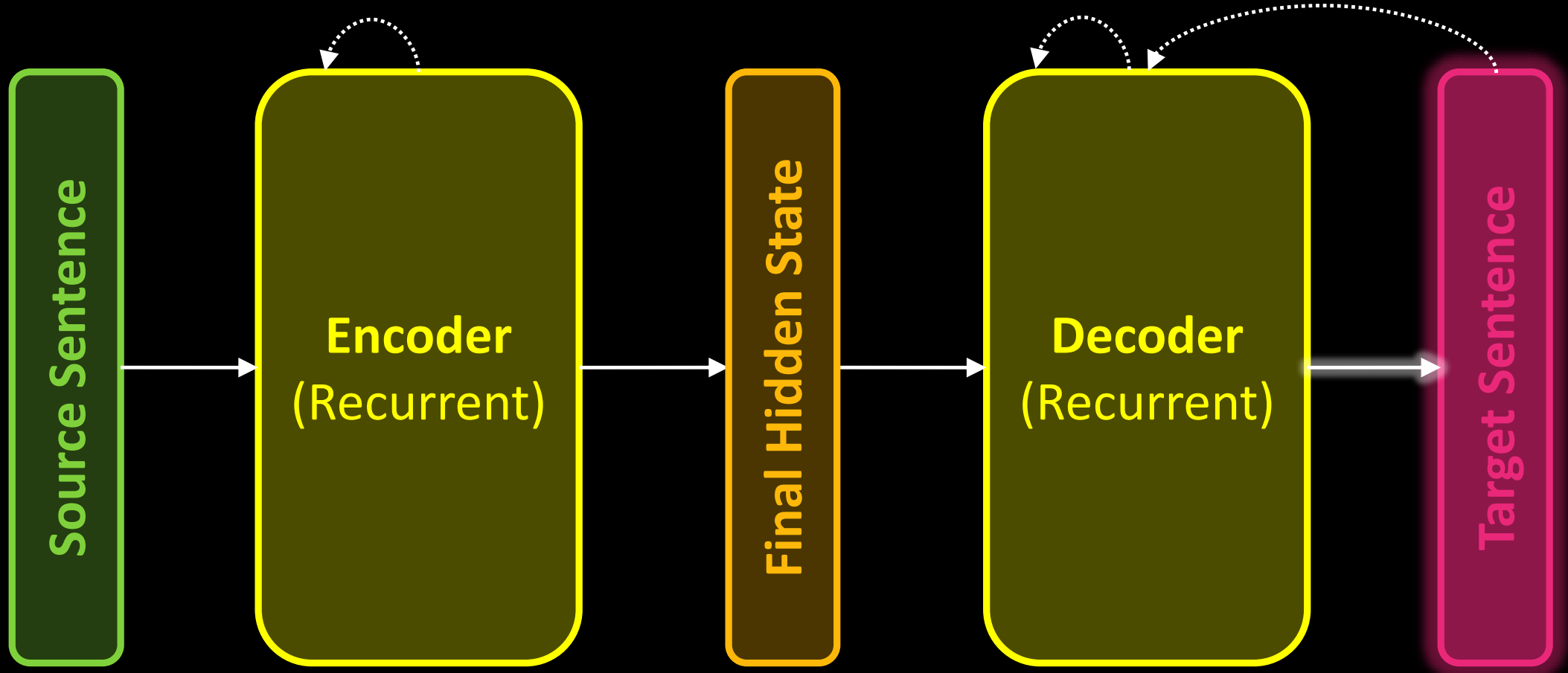
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

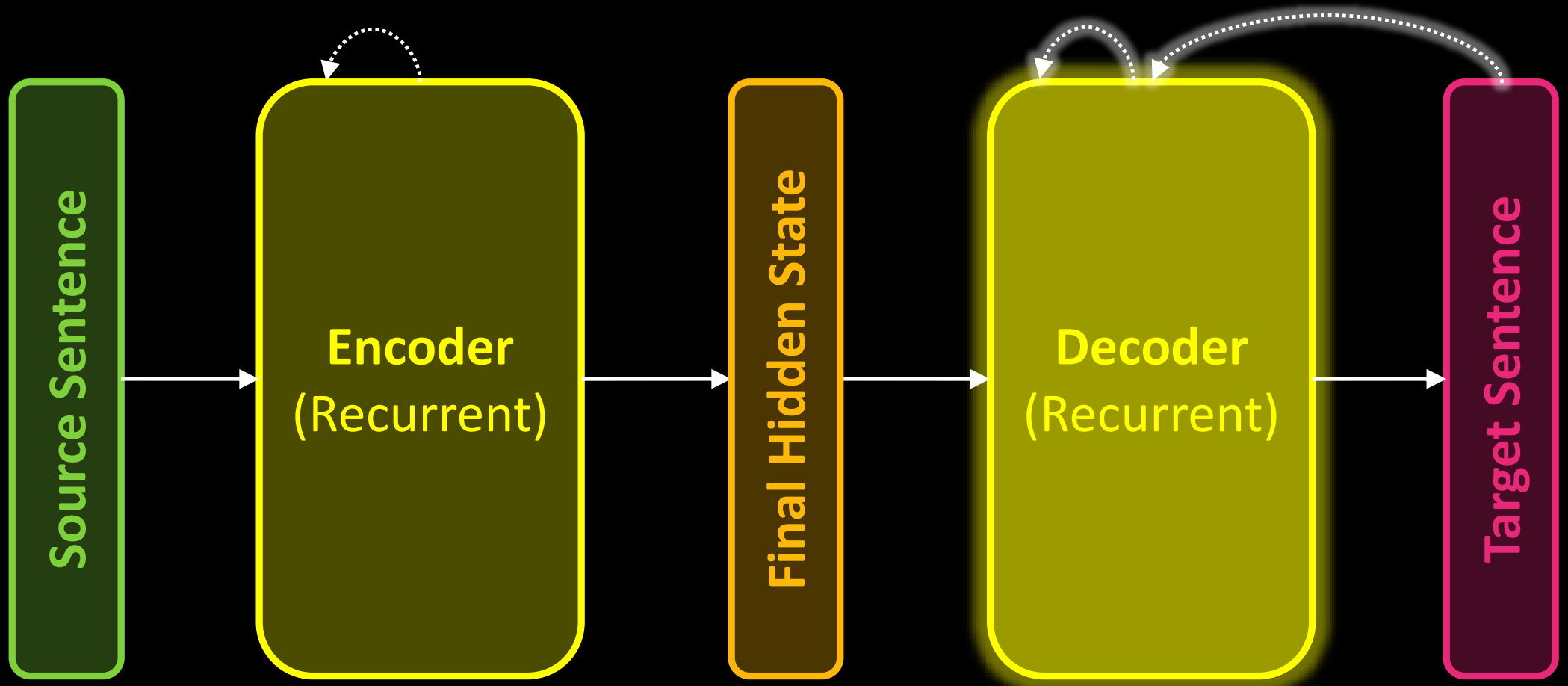
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

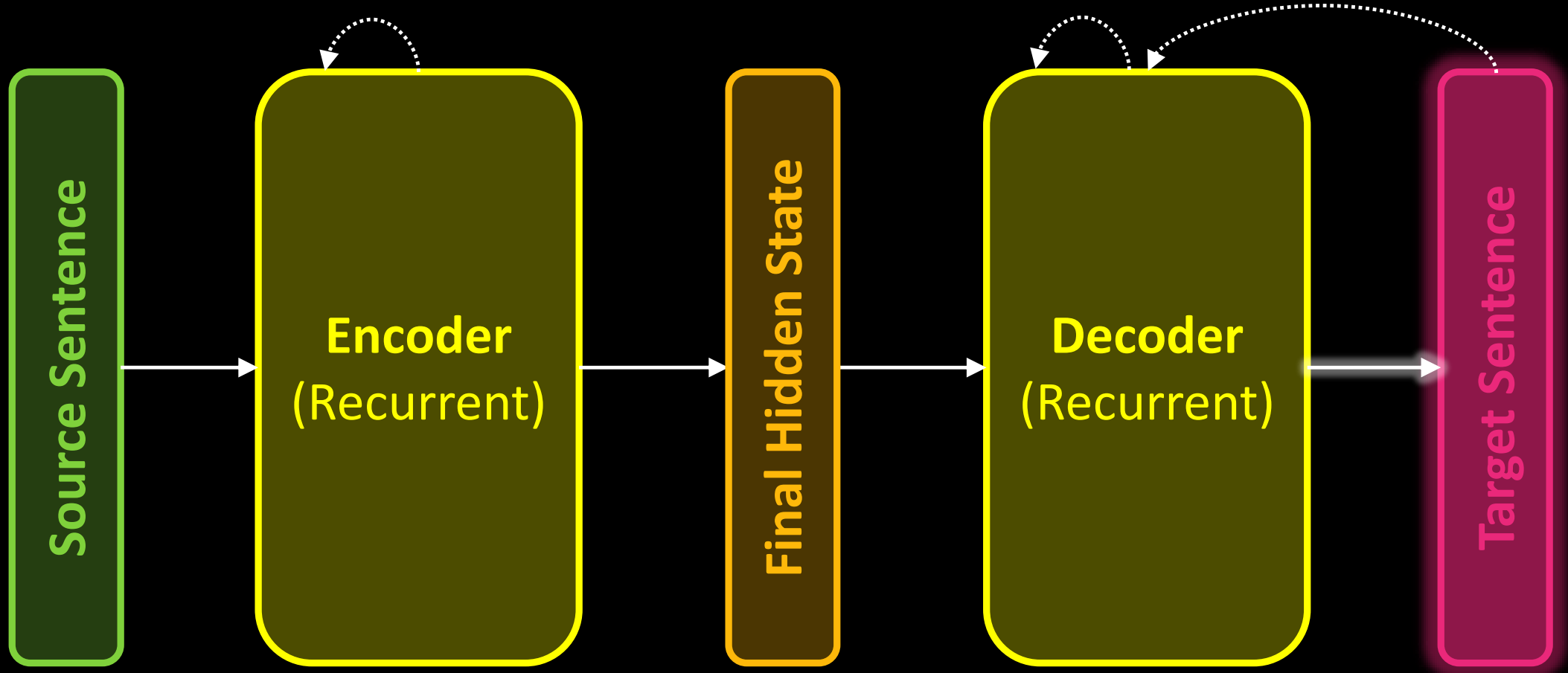
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

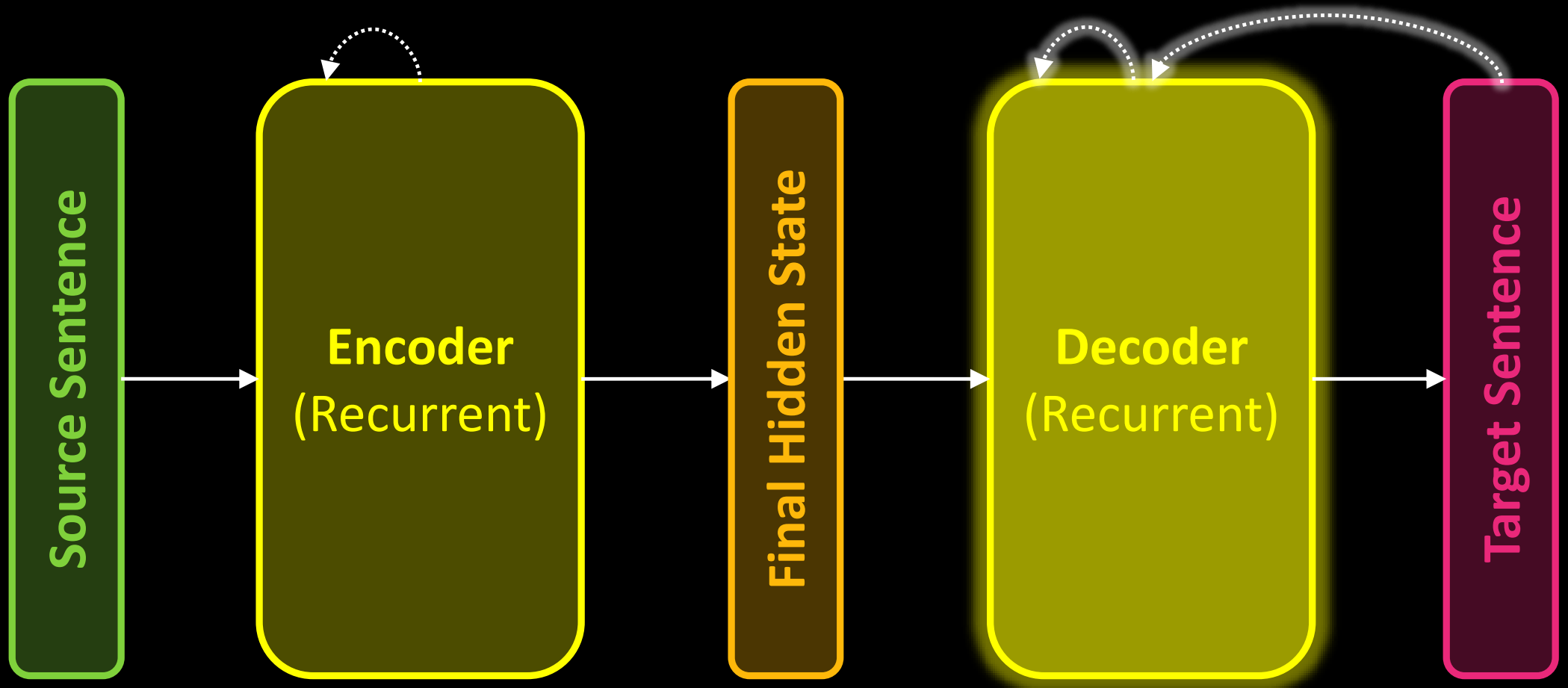
Application: Machine Translation



Paris is the capital of France.

Paris est la capitale de la France.

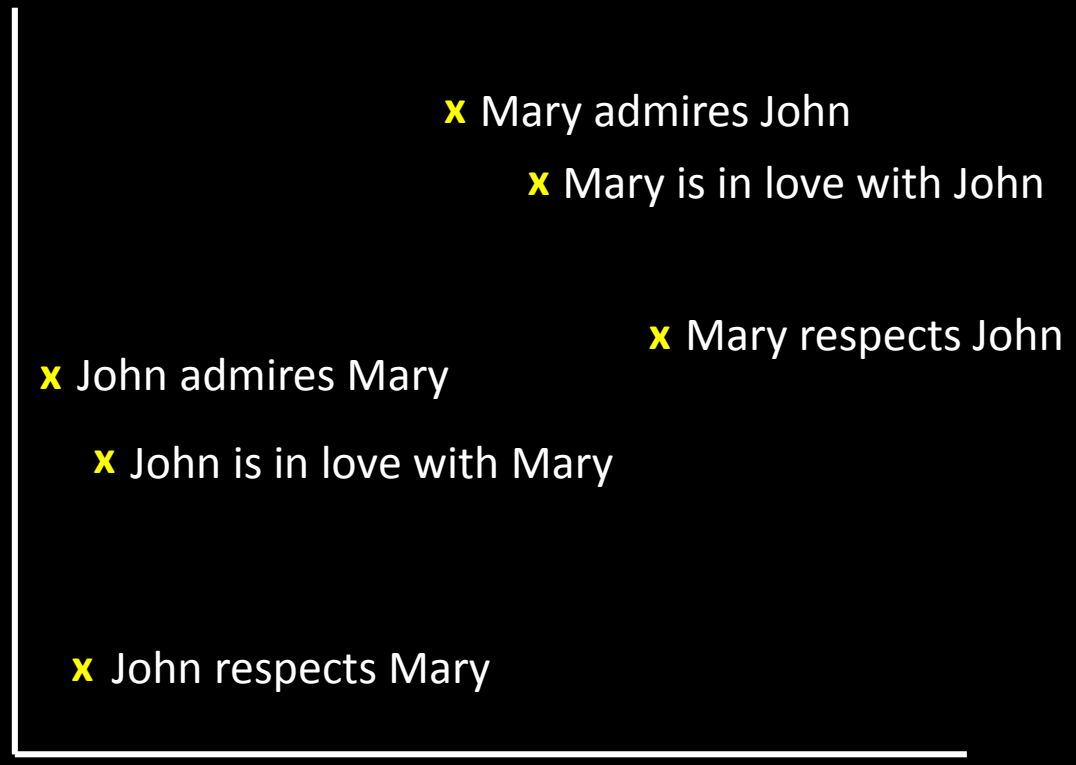
Application: Machine Translation



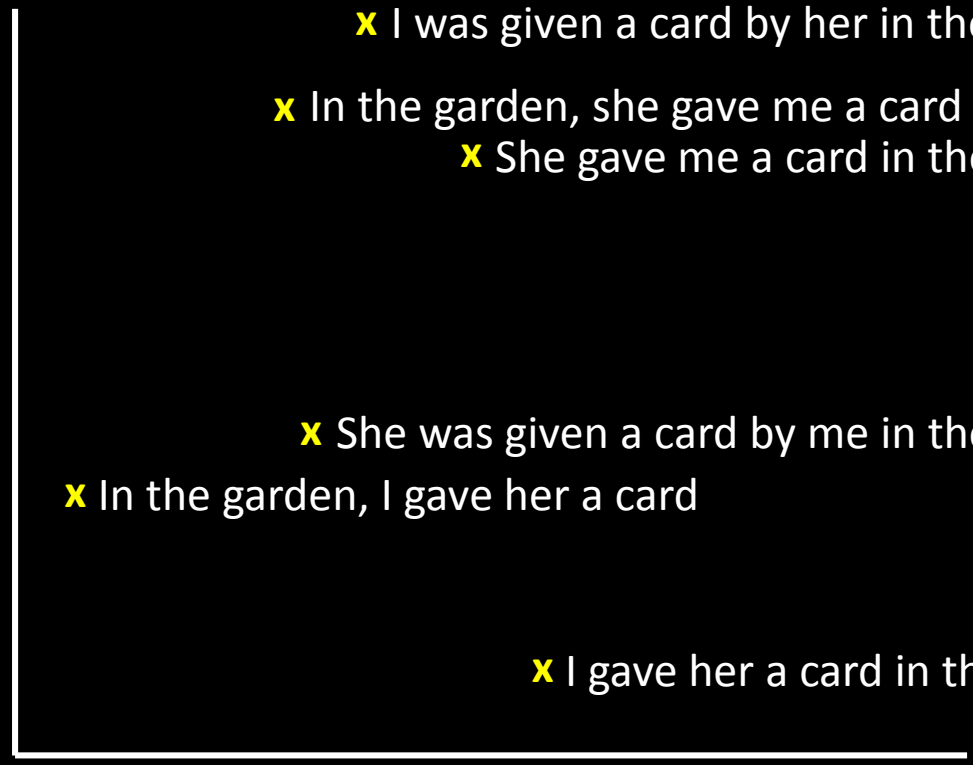
Paris is the capital of France.

Paris est la capitale de la France.

Application: Machine Translation



Application: Machine Translation

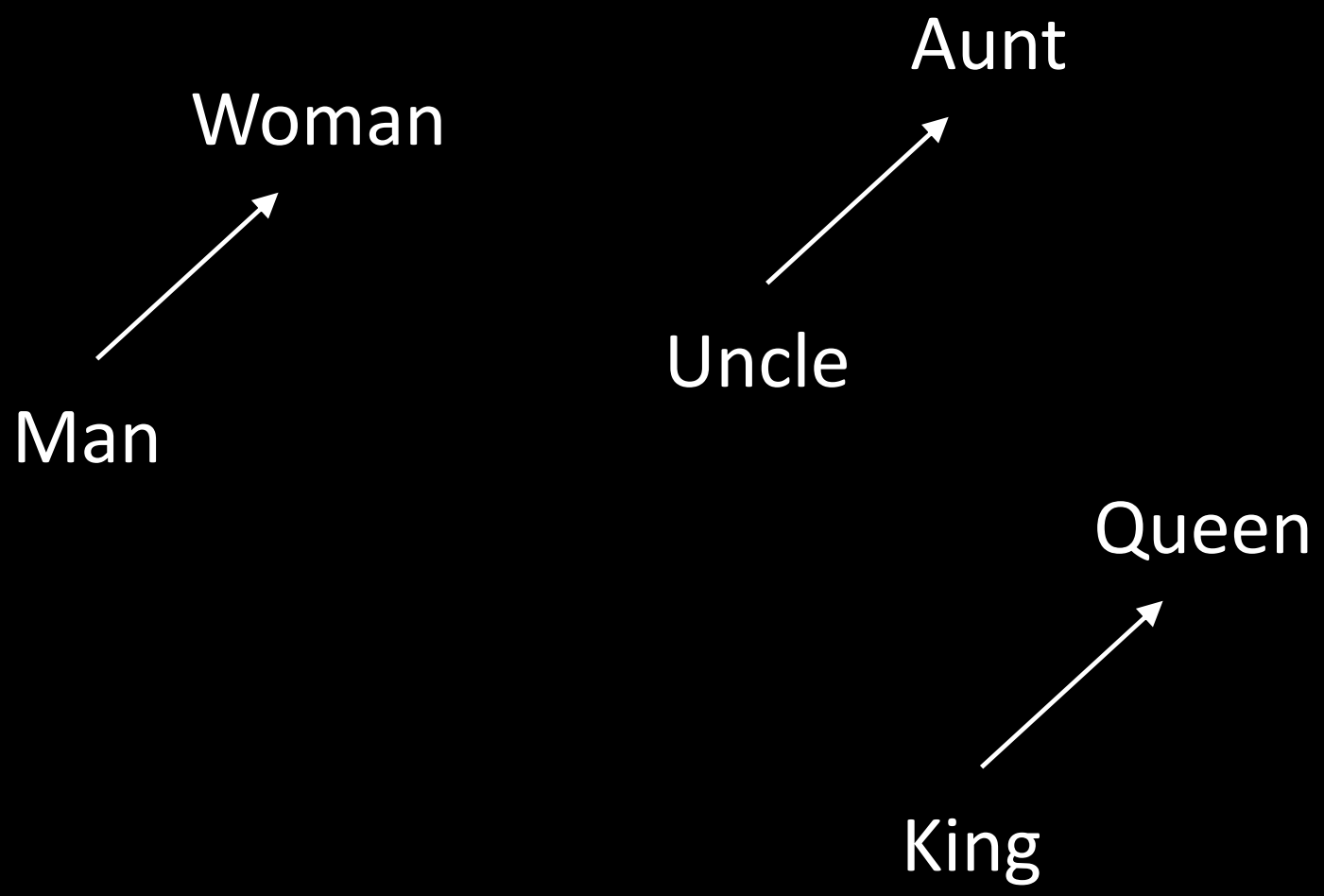


Application: Word Embeddings

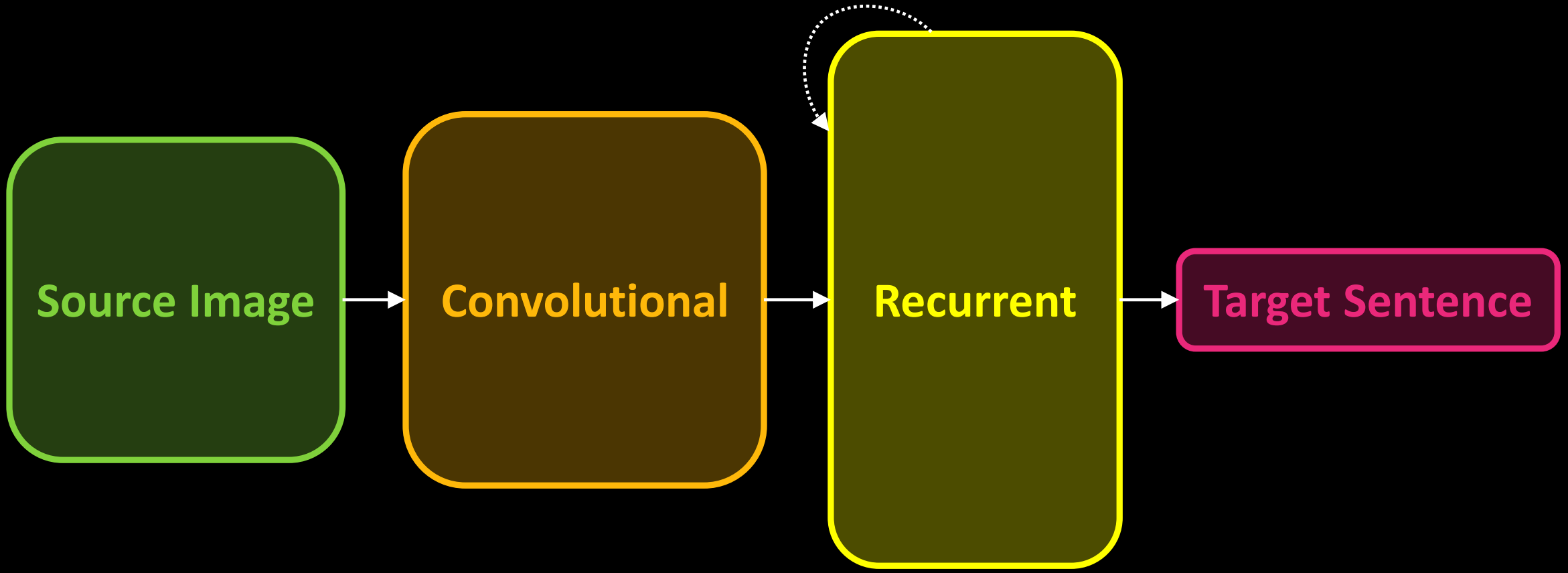
King – Man + Woman = Queen

Paris – France + England = London

Application: Word Embeddings



Application: Caption Generation



Application: Caption Generation

A group of young people playing a game of Frisbee.

Application: Caption Generation

Two hockey players are fighting over the puck.

Application: Caption Generation

A herd of elephants walking across a dry grass field.

Application: Caption Generation

A man flying through the air while riding a snowboard.

Application: Caption Generation

Deep Visual-Semantic Alignments for Generating Image Descriptions **(Stanford)**

Show and Tell: A Neural Image Caption Generator **(Google)**

From Captions to Visual Concepts and Back **(MSR)**

Deep Captioning with Multimodal Recurrent Neural Networks **(UCLA)**

Mind's Eye: A Recurrent Visual Representation for Image Caption Generation **(MSR)**

1943 – 2006: A prehistory of deep learning

2006 – 2015: A history of deep learning

2006 – 2014: What is deep learning? Take I

2014 – 20XX: What is deep learning? Take II

1943 – 2006: A prehistory of deep learning
2006 – 2015: A history of deep learning

2006 – 2014: What is deep learning? Take I

2014 – 20XX: What is deep learning? Take II

Frontiers of Neural Architectures: Memory

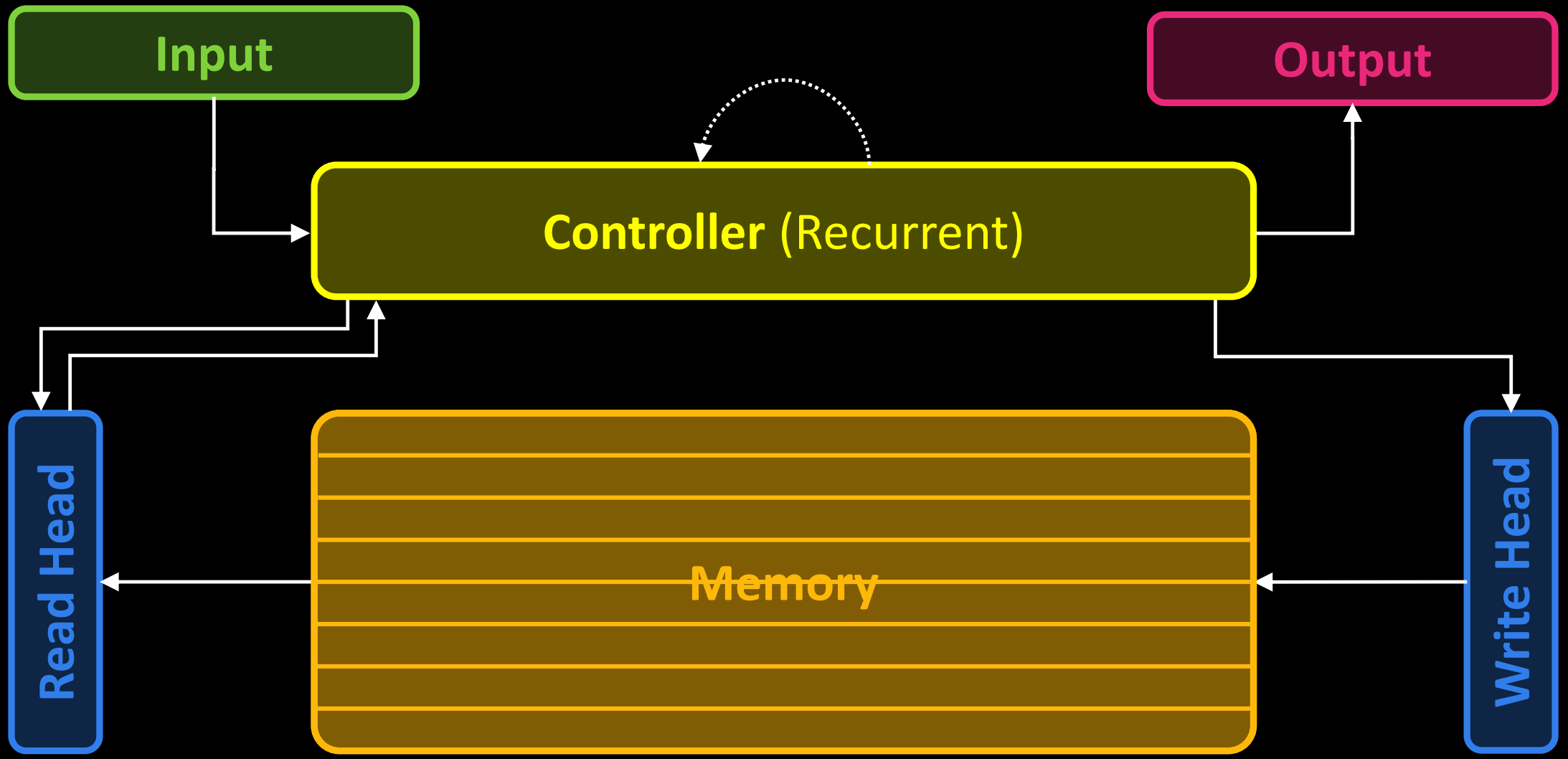
Neural Turing Machines

Alex Graves, Greg Wayne, Ivo Danihelka

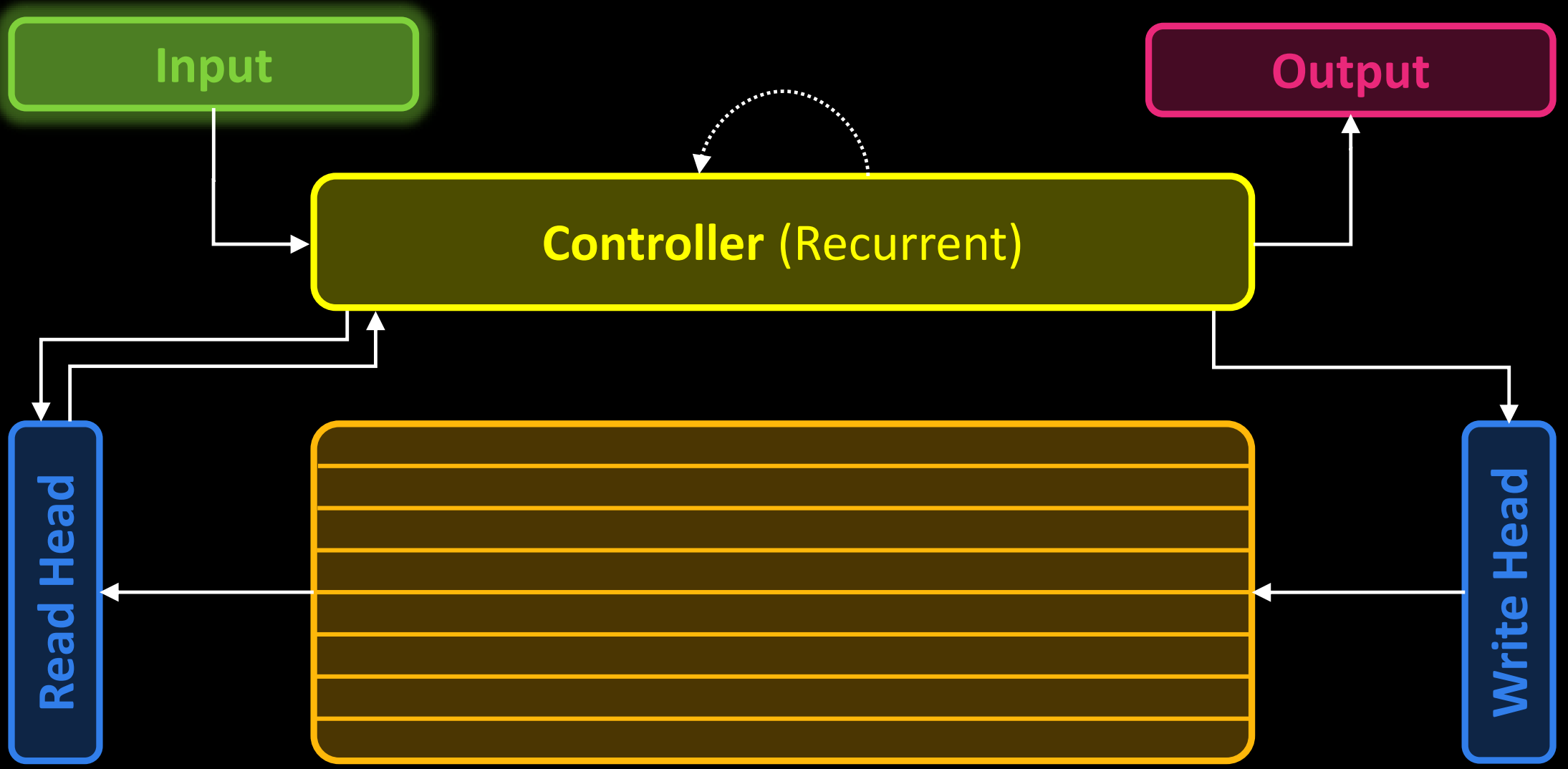
(Submitted on 20 Oct 2014 (v1), last revised 10 Dec 2014 (this version, v2))

We extend the capabilities of neural networks by coupling them to external memory resources, which they can interact with by attentional processes. The combined system is analogous to a Turing Machine or Von Neumann architecture but is differentiable end-to-end, allowing it to be efficiently trained with gradient descent. Preliminary results demonstrate that Neural Turing Machines can infer simple algorithms such as copying, sorting, and associative recall from input and output examples.

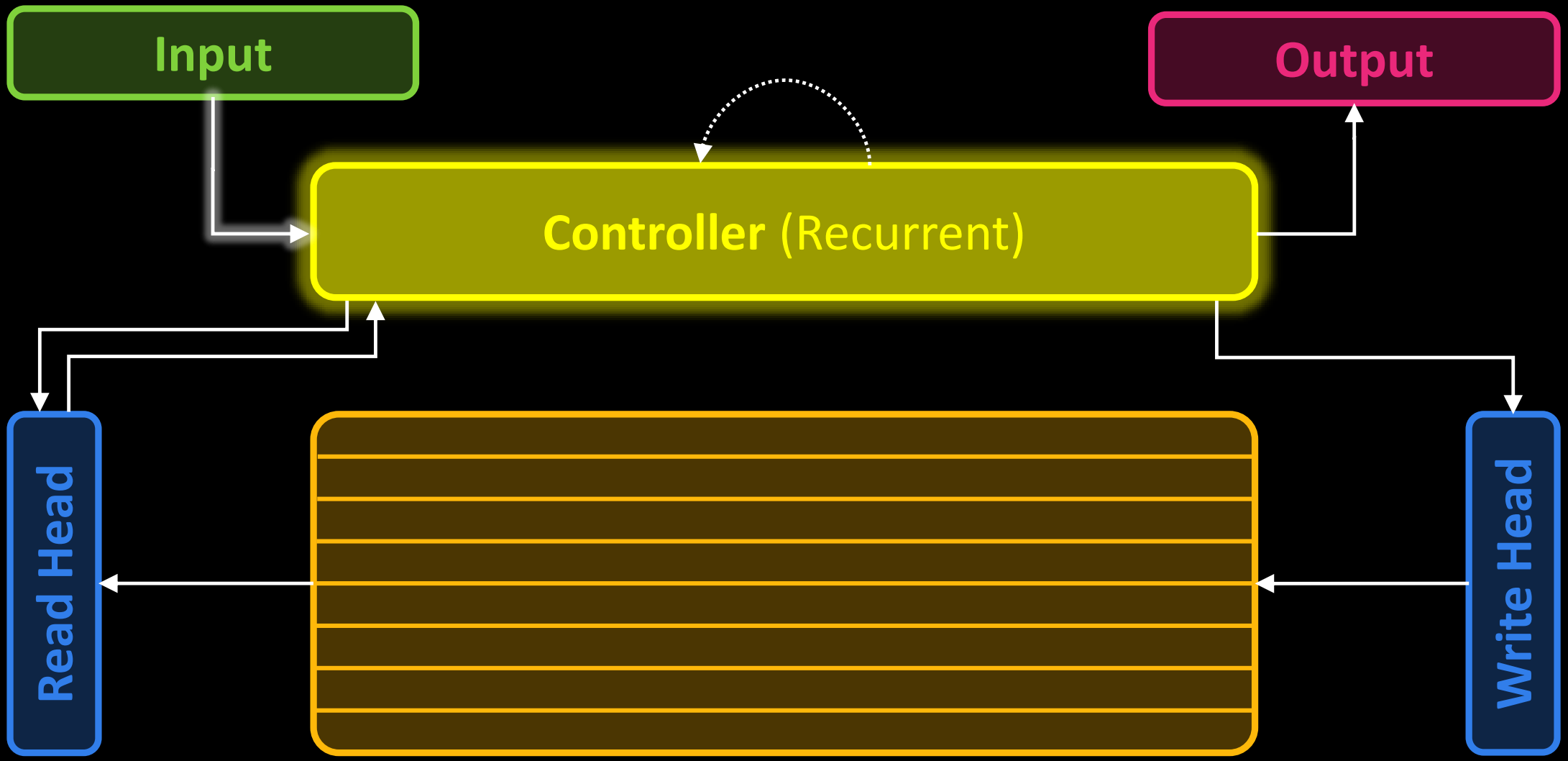
Frontiers of Neural Architectures: Memory



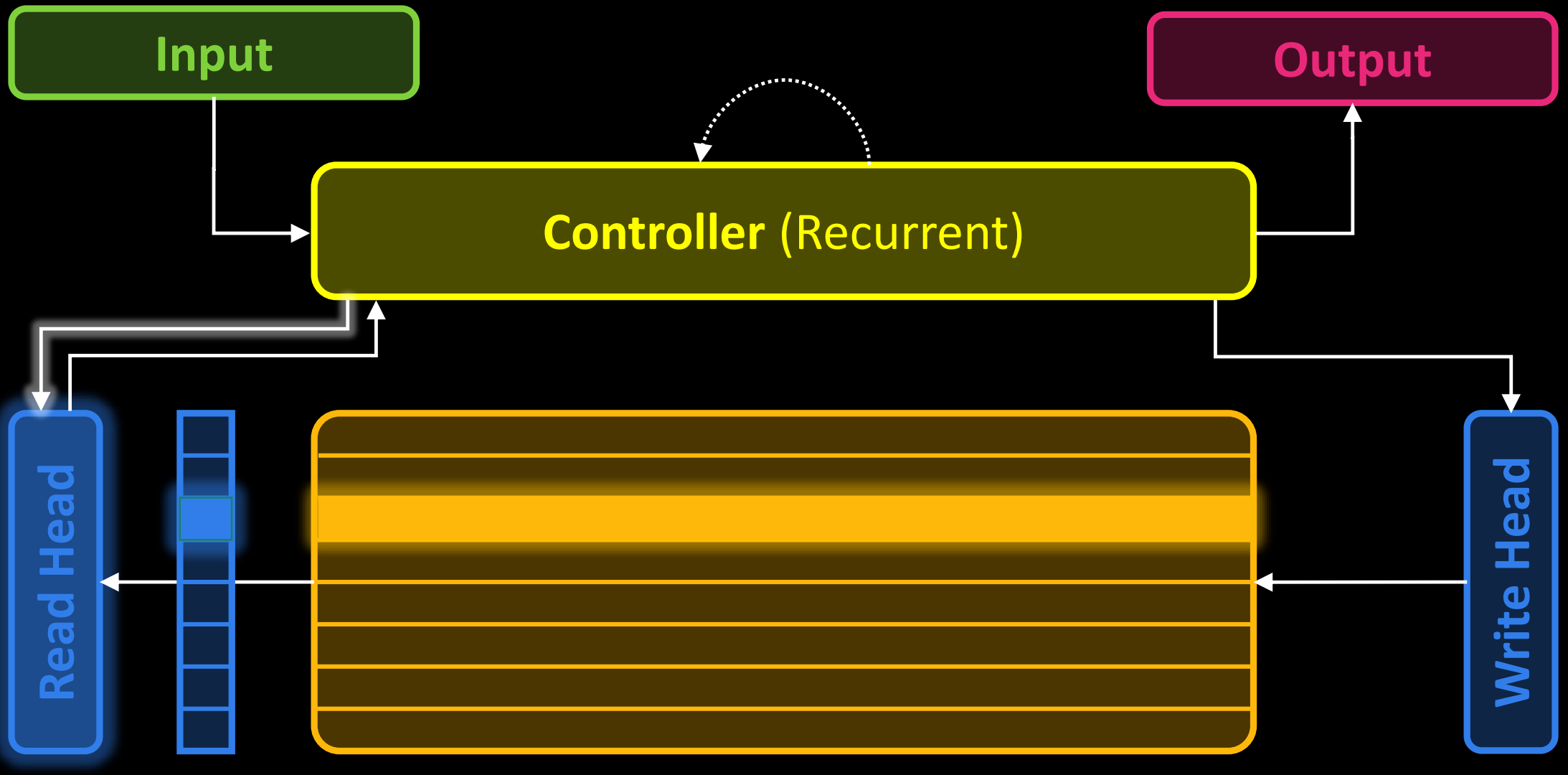
Frontiers of Neural Architectures: Memory



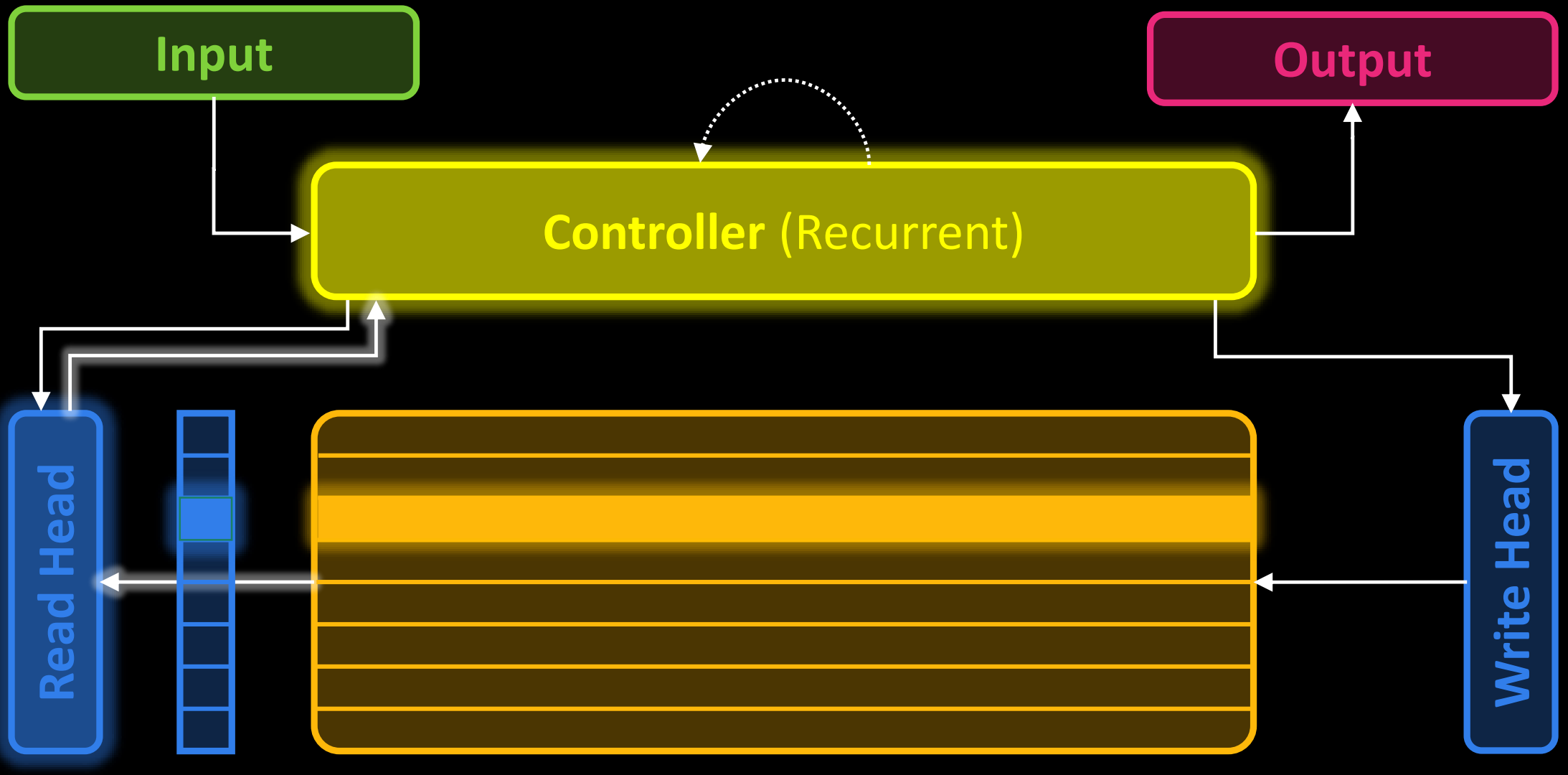
Frontiers of Neural Architectures: Memory



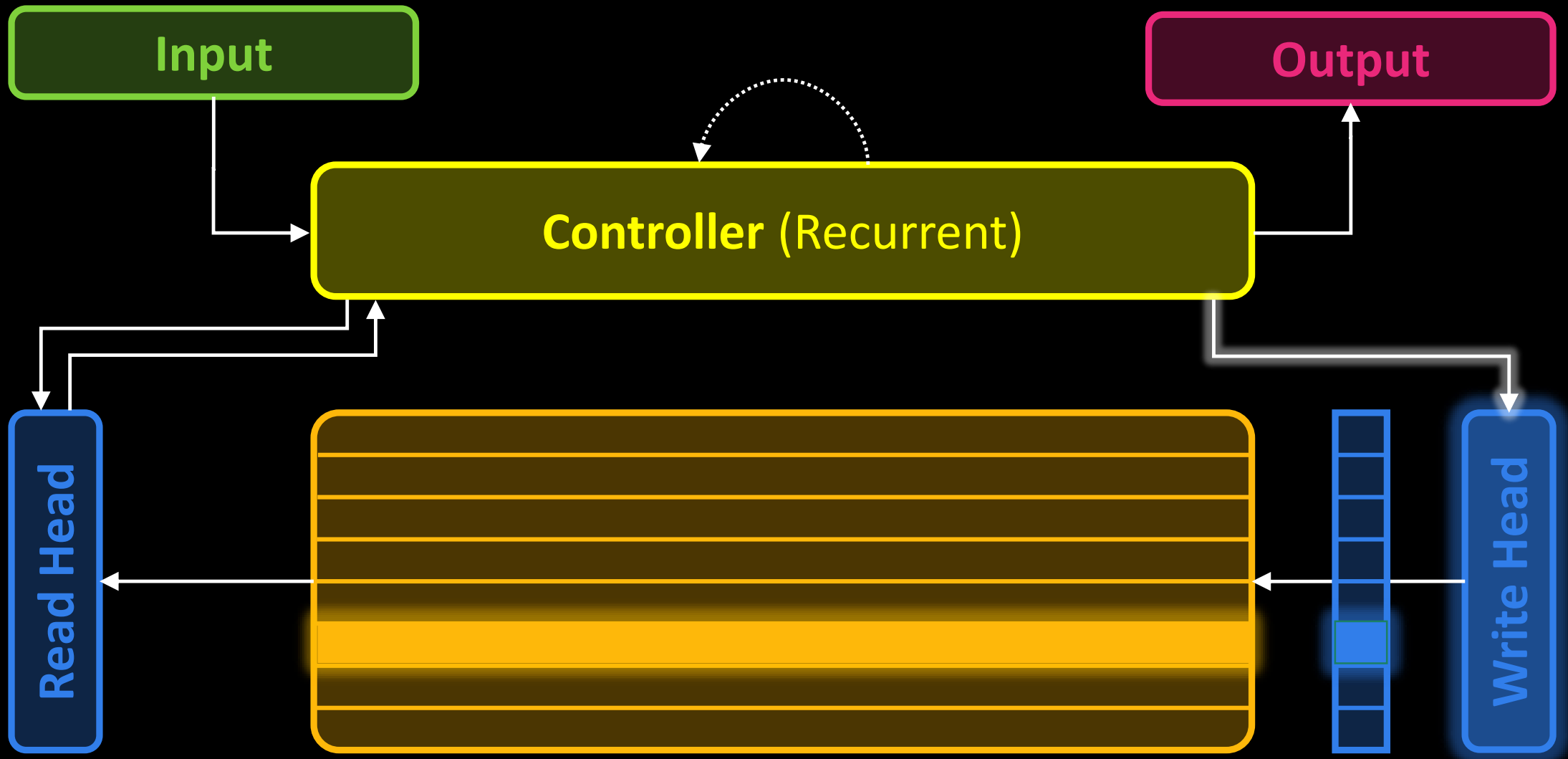
Frontiers of Neural Architectures: Memory



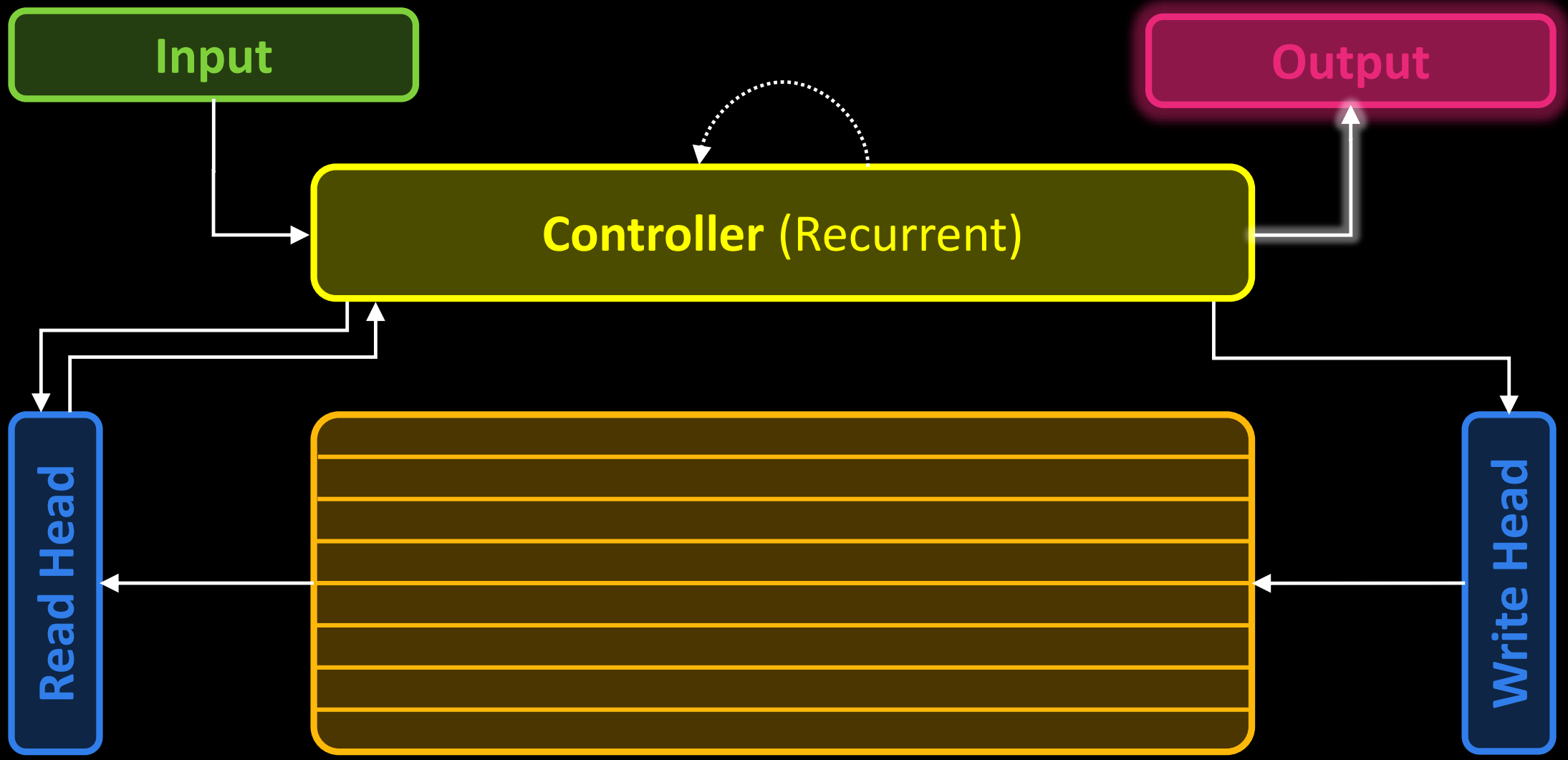
Frontiers of Neural Architectures: Memory



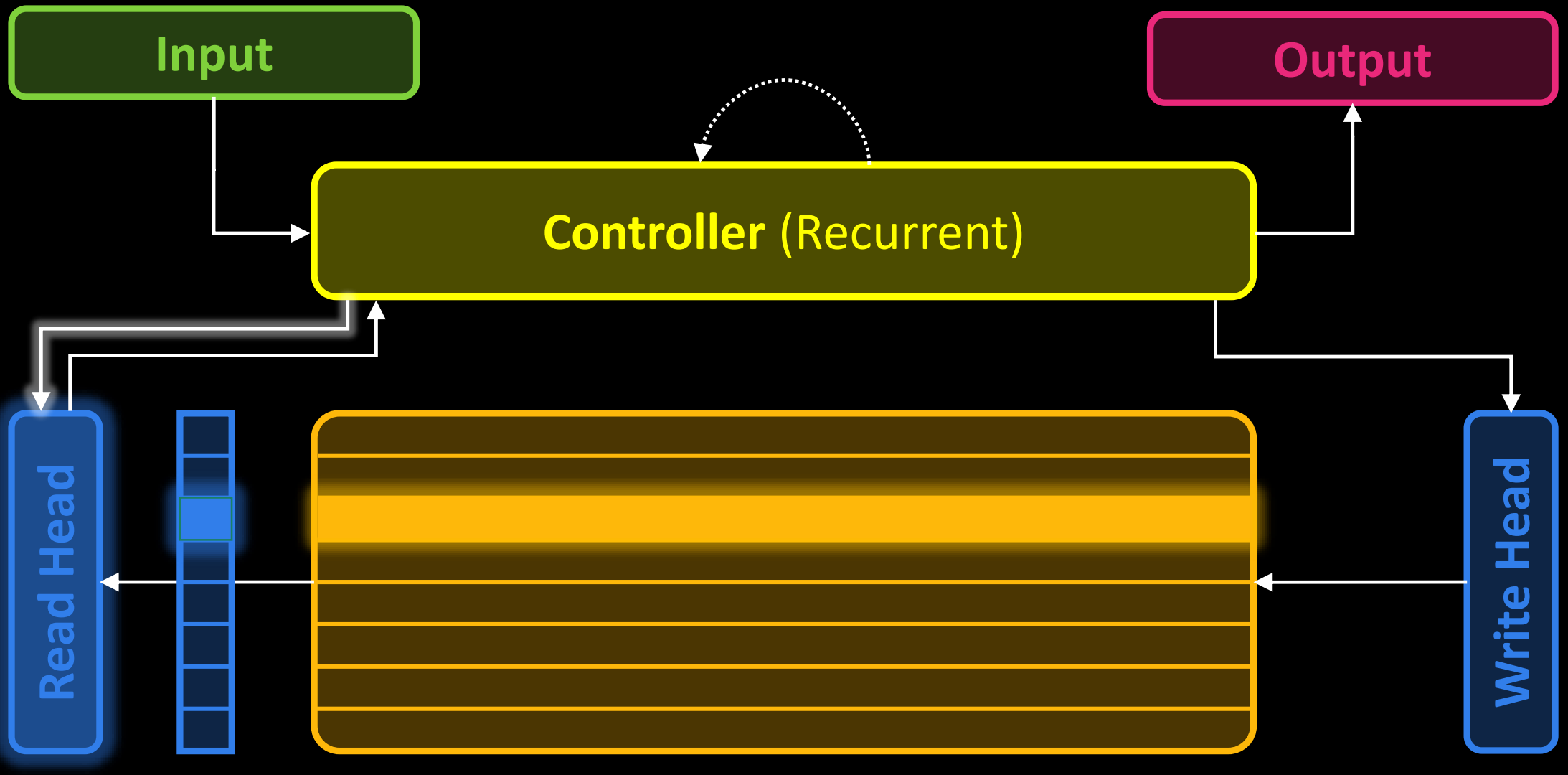
Frontiers of Neural Architectures: Memory



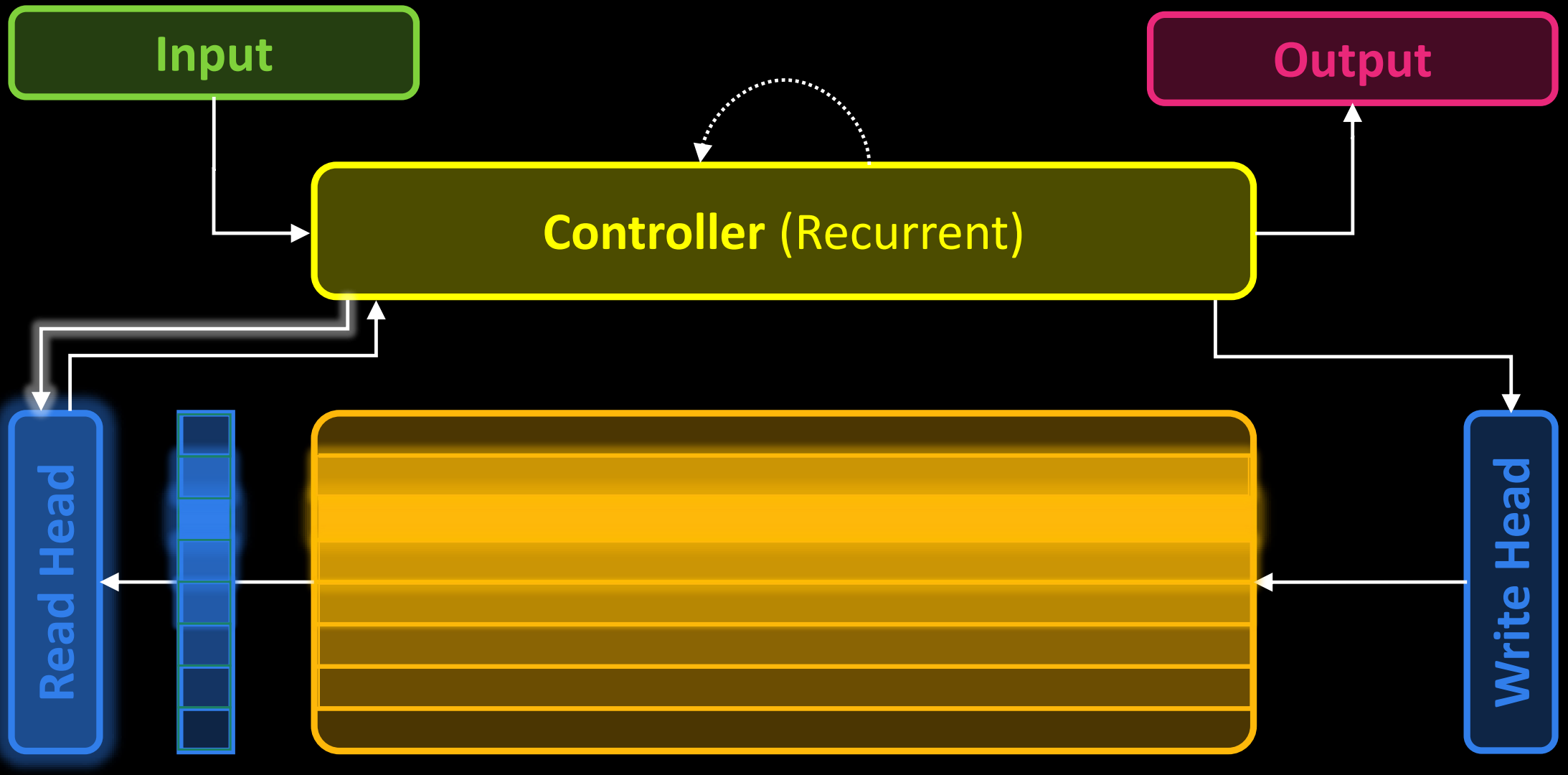
Frontiers of Neural Architectures: Memory



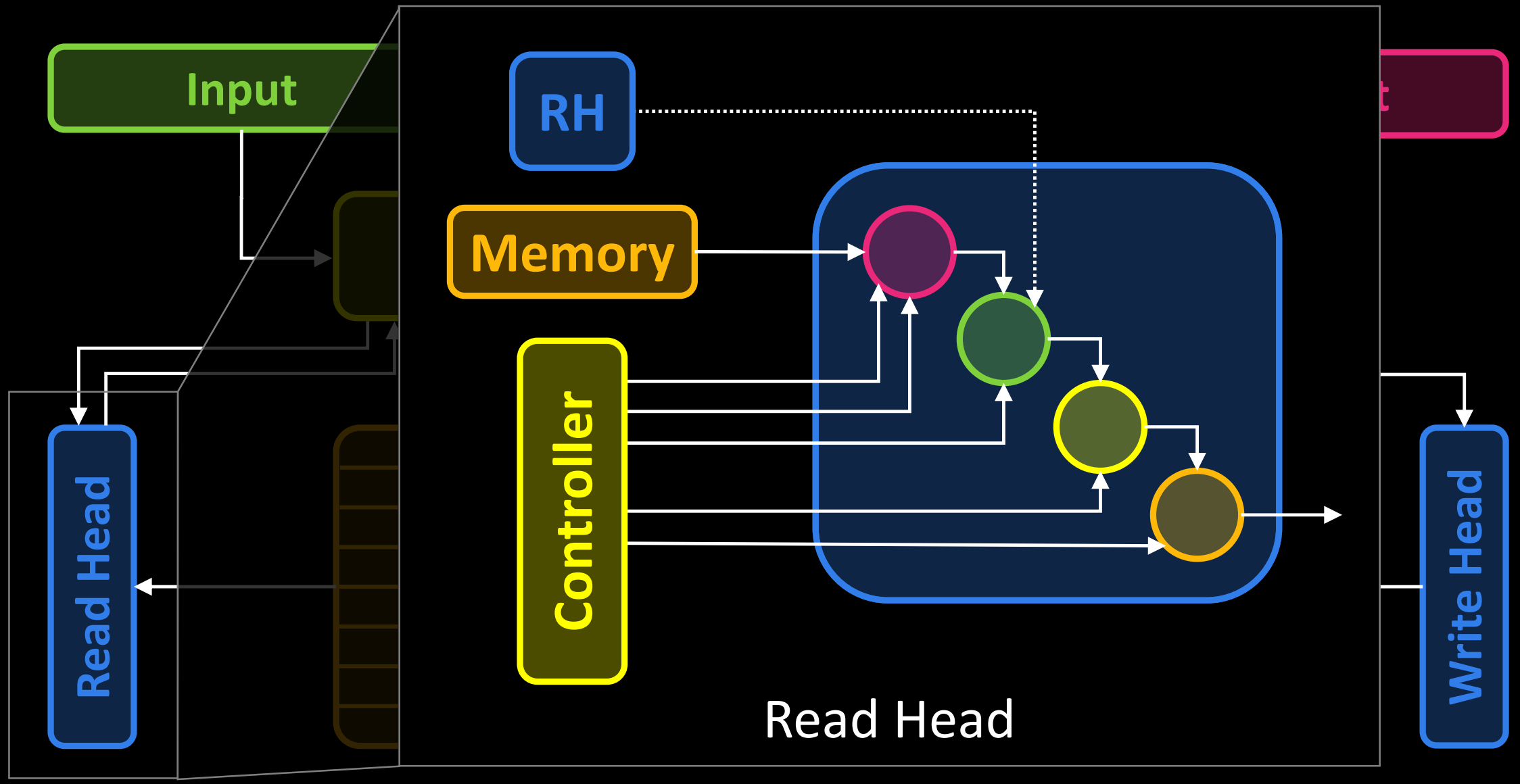
Frontiers of Neural Architectures: Memory



Frontiers of Neural Architectures: Memory



Frontiers of Neural Architectures: Memory



Application: Algorithms

Tasks:

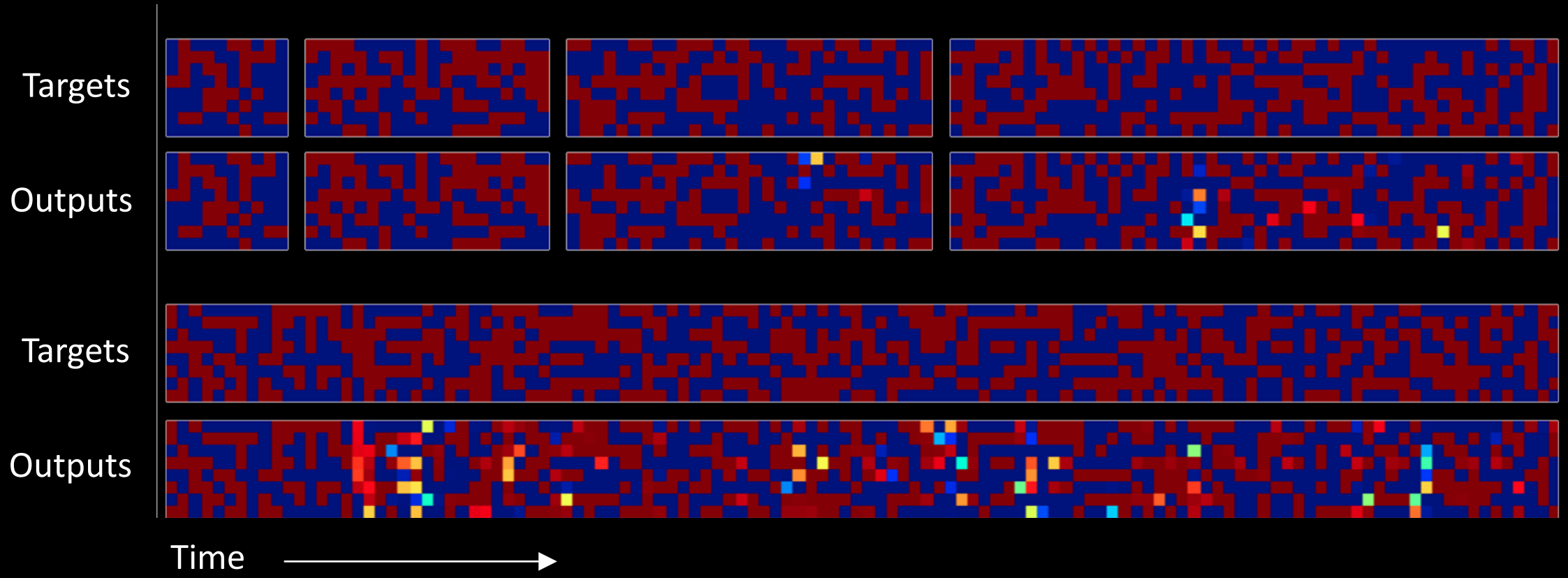
- Copy
- Repeat Copy
- Associative Recall
- Dynamic N-Grams
- Priority Sort

Application: Algorithms

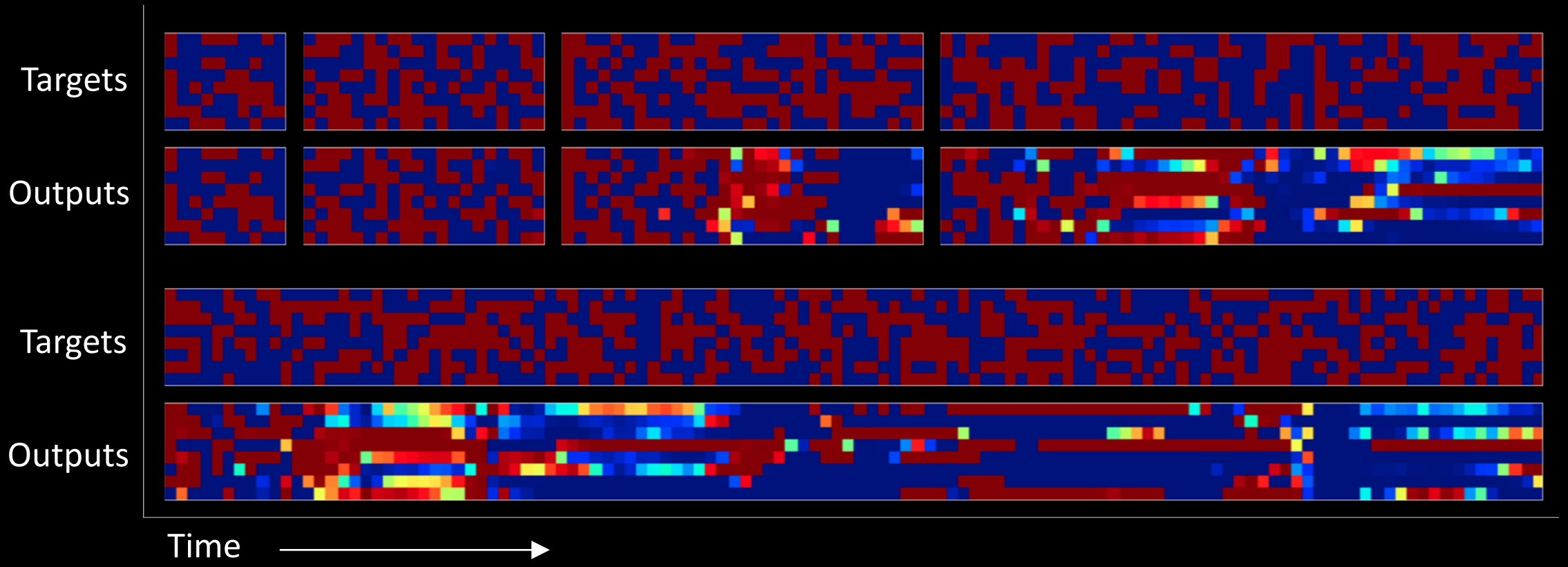
Tasks:

- Copy
- Repeat Copy
- Associative Recall
- Dynamic N-Grams
- Priority Sort

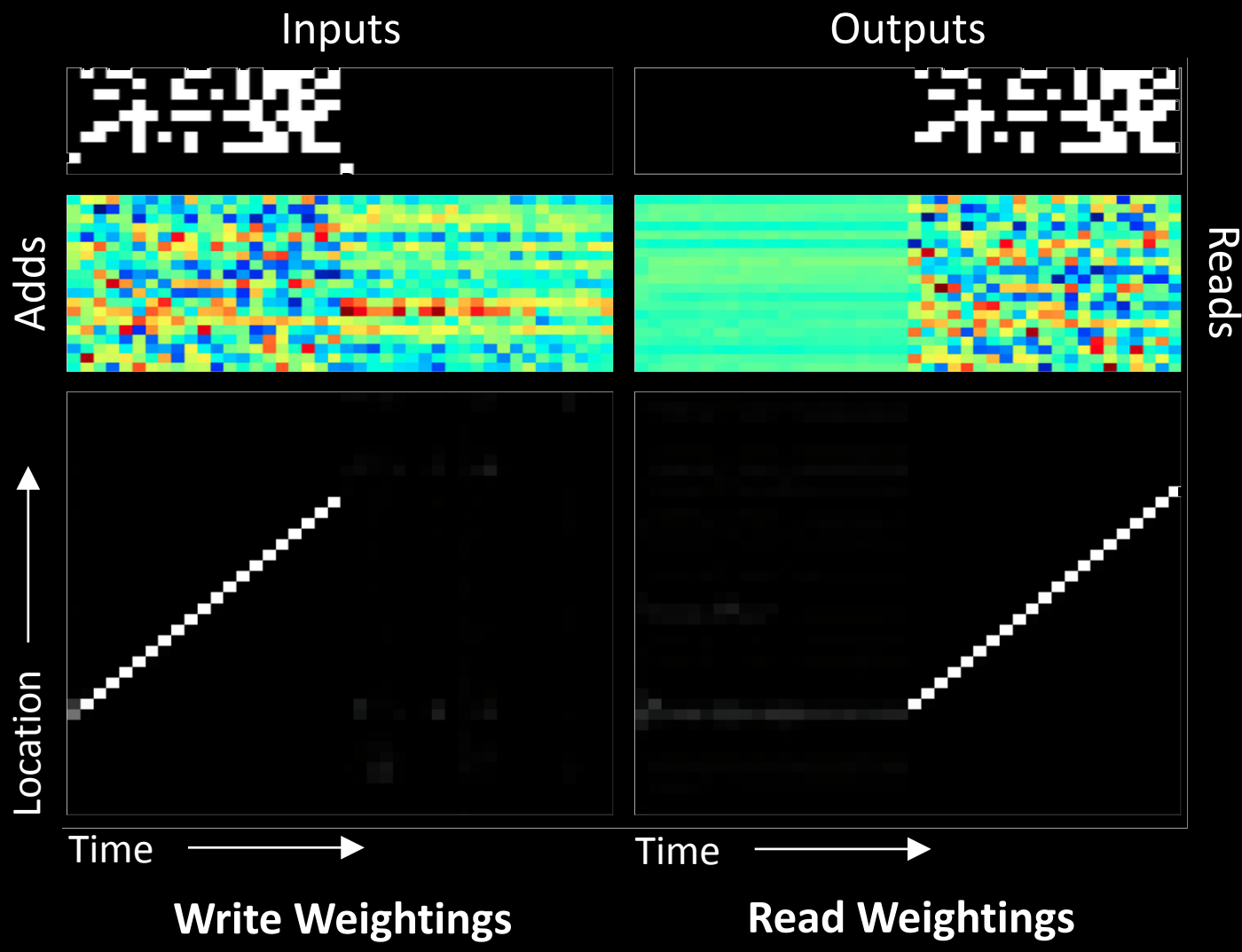
Application: Algorithms



Application: Algorithms



Application: Algorithms



Application: Algorithms

initialise: move head to start location
while input delimiter not seen **do**
    receive input vector
    write input to head location
    increment head location by 1
end while
return head to start location
while true **do**
    read output vector from head location
    emit output
    increment head location by 1
end while

Frontiers of Neural Architectures: Memory

Memory Networks.
Jason Weston, Sumit Chopra, Antoine Bordes.

Teaching Machines to Read and Comprehend.
Karl Moritz Hermann et. al.

Large-scale Simple Question Answering with Memory Networks.
Antoine Bordes et. al.

Learning CFGs: Capabilities and limitations of a recurrent neural network with an external stack memory.
S. Das, C. L. Giles, and G. Z. Sun.

Inferring Algorithmic Patterns with Stack Augmented Recurrent Nets.
Armand Joulin and Tomas Mikolov.

Reinforcement Learning Turing Machine.
Wojciech Zaremba and Ilya Sutskever.

End-To-End Memory Networks.
S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus.

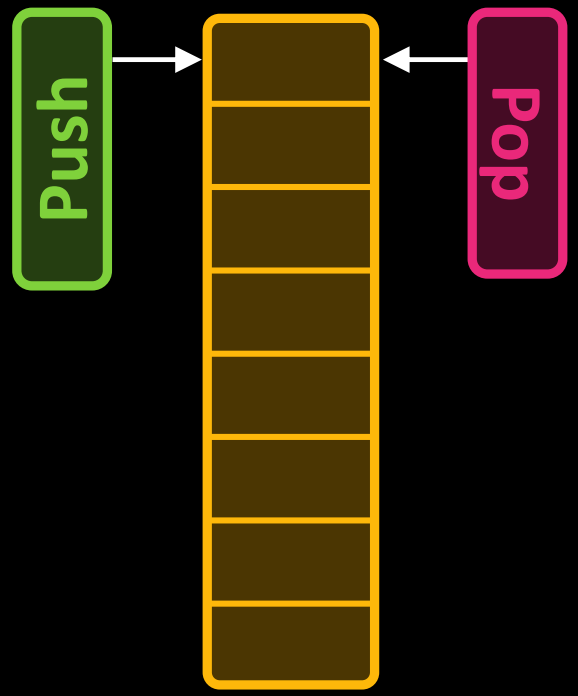
Learning to Transduce with Unbounded Memory.
E. Grefenstette et. al.

Transition-based dependency parsing with stack long short-term memory.
C Dyer et. al.

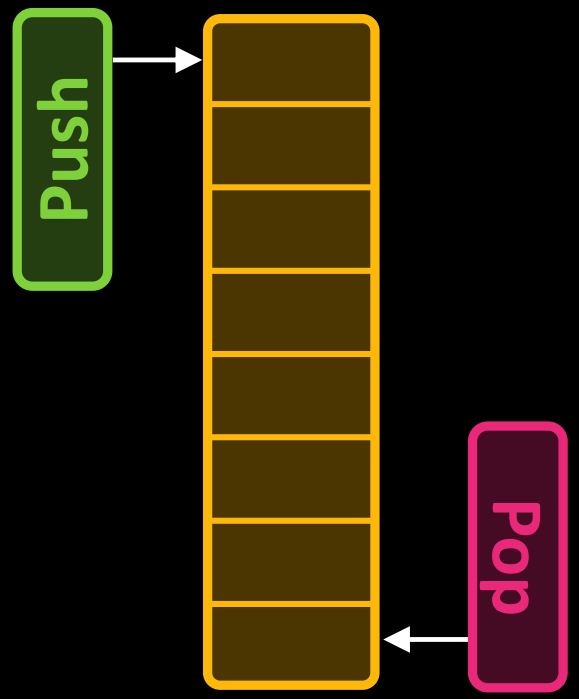
Ask Me Anything: Dynamic Memory Networks for Natural Language Processing.
A Kumar et. al.

Recurrent Neural Networks with External Memory for Spoken Language Understanding.
B Peng et. al.

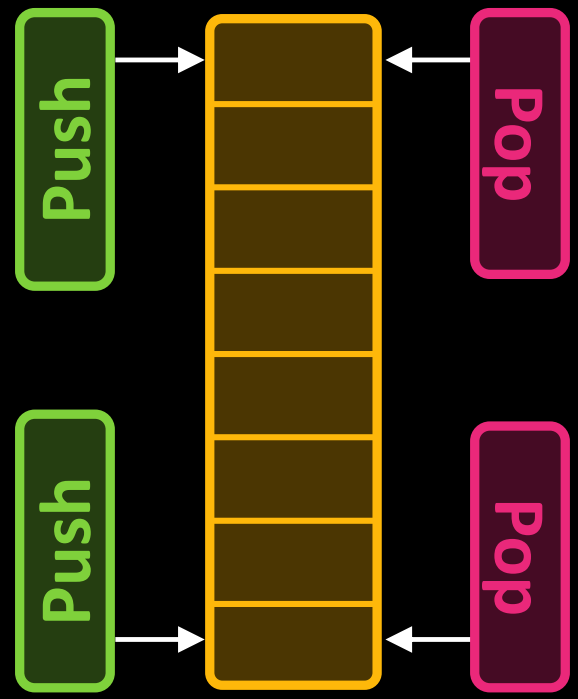
Frontiers of Neural Architectures: Memory



Neural Stack



Neural Queue



Neural DeQueue

Application: Question and Answer

Joe went to the garden then Fred picked up the milk; Joe moved to the bathroom and Fred dropped the milk, and then Dan moved to the living_room.
Where is Dan? A: living room I believe
Where is Joe? A: the bathroom
Fred moved to the bedroom and Joe went to the kitchen then Joe took the milk there and Dan journeyed to the bedroom; Joe discarded the milk.
Where is the milk now ? A: the milk is in the kitchen
Where is Dan now? A: I think he is in the bedroom
Joe took the milk there, after that Mike travelled to the office, then Joe went to the living_room, next Dan went back to the kitchen and Joe travelled to the office.
Where is Joe now? A: I think Joe is in the office

Frontiers of Neural Architectures: Attention

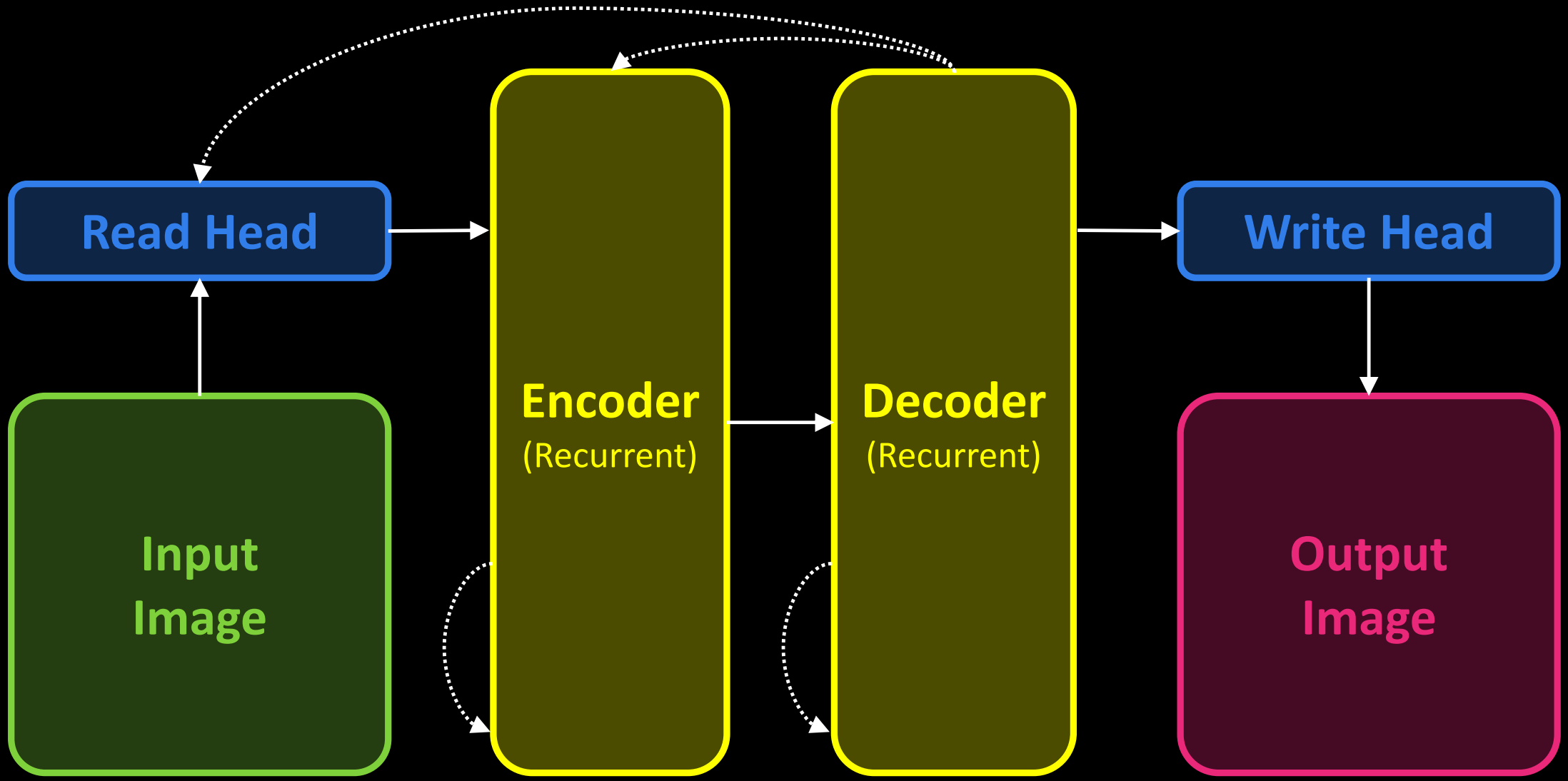
DRAW: A Recurrent Neural Network For Image Generation

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, Daan Wierstra

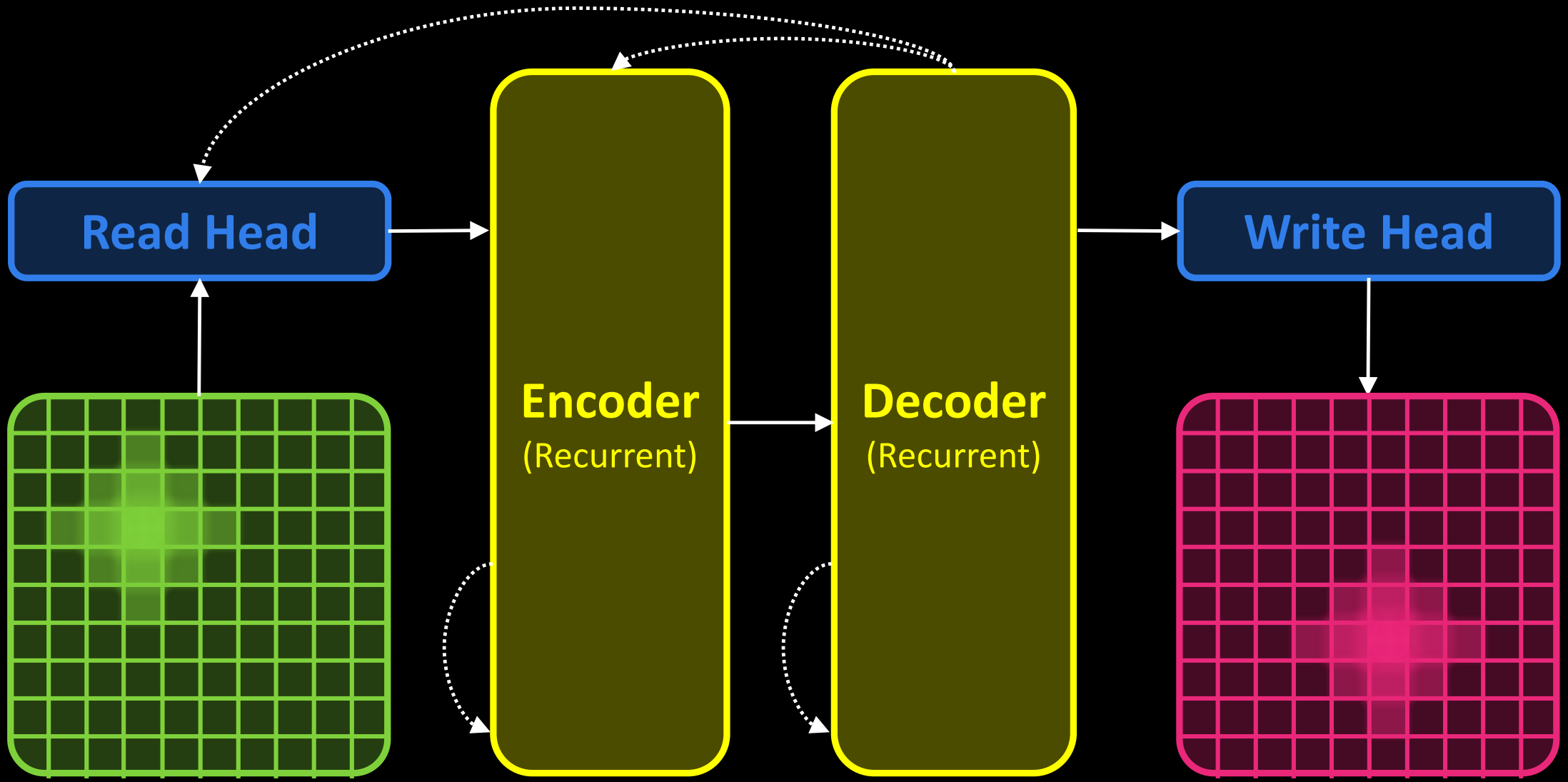
(Submitted on 16 Feb 2015 (v1), last revised 20 May 2015 (this version, v2))

This paper introduces the Deep Recurrent Attentive Writer (DRAW) neural network architecture for image generation. DRAW networks combine a novel spatial attention mechanism that mimics the foveation of the human eye, with a sequential variational auto-encoding framework that allows for the iterative construction of complex images. The system substantially improves on the state of the art for generative models on MNIST, and, when trained on the Street View House Numbers dataset, it generates images that cannot be distinguished from real data with the naked eye.

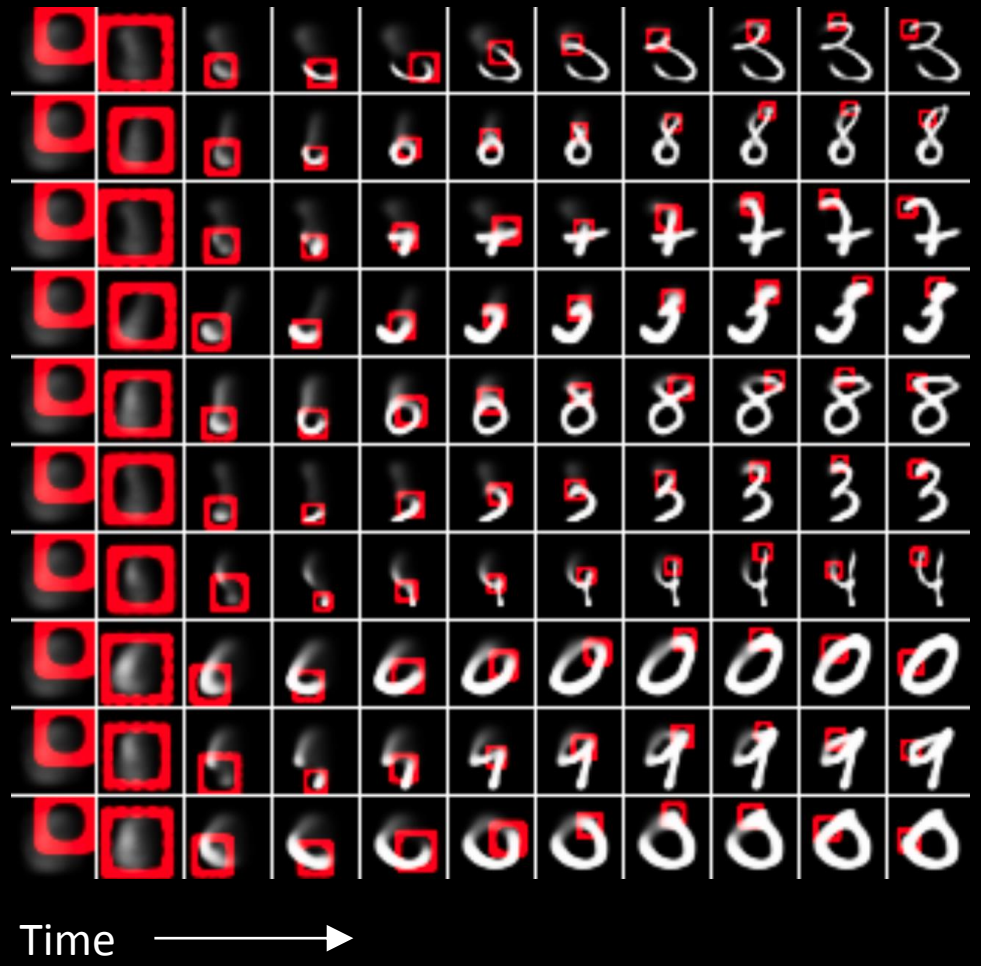
Frontiers of Neural Architectures: Attention



Frontiers of Neural Architectures: Attention



Frontiers of Neural Architectures: Attention



Frontiers of Neural Architectures: Attention

Reading MNIST

Frontiers of Neural Architectures: Reinforcement

LETTER

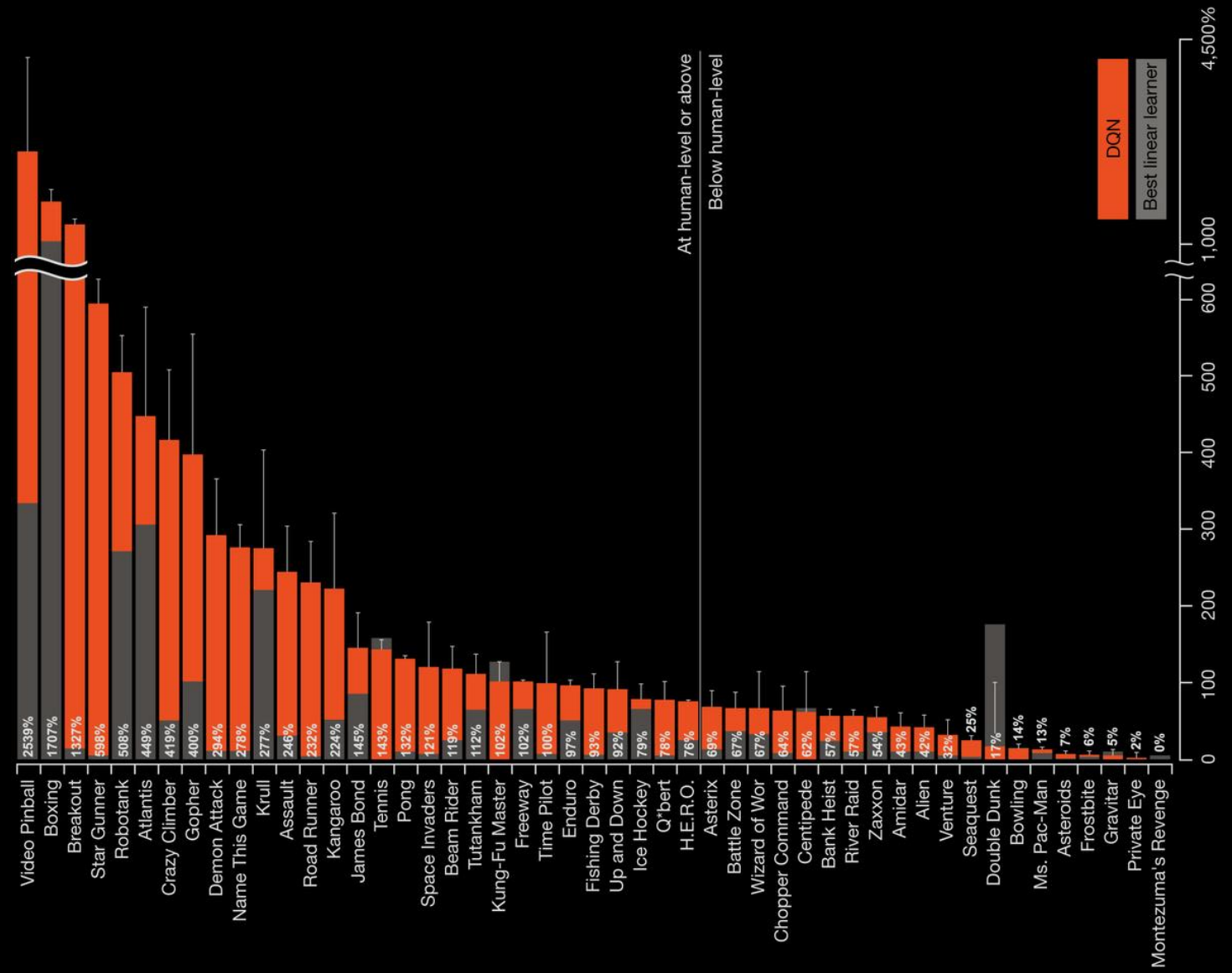
doi:10.1038/nature14236

Human-level control through deep reinforcement learning

Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

Frontiers of Neural Architectures: Reinforcement

Frontiers of Neural Architectures: Reinforcement



Application: Robotics

What is deep learning? Take II

deep learning = non-convex differentiable optimization

deep reinforcement learning = non-convex non-differentiable optimization

deep learning = design of differentiable (and thus trainable) computers

Resources

Review	**Deep Learning**	Yann LeCun, Yoshua Bengio, Geoff Hinton, Nature 521, 436-444	
Classes	**Introductory Deep Learning**	Hugo Larochelle, Université de Sherbrooke	http://bit.ly/1NhkCf2
	Convolutional Nets (CS231n)	Andrej Karpathy, Stanford University	http://cs231n.stanford.edu
	Recurrent Nets (CS224d)	Richard Socher, Stanford University	http://cs224d.stanford.edu
	Deep Reinforcement Learning (CS294)	John Schulman, UC Berkeley	http://rll.berkeley.edu/deeprlcourse/
Book	**Deep Learning**	Yoshua Bengio, Ian J. Goodfellow, and Aaron Courville	http://goodfeli.github.io/dlbook/
Code	**Keras, Lasagne, Blocks**	Basic / off-the-shelf	
	Torch, Theano, Caffe, CGT	Heavy duty	
	Autograd	Prototyping	