tf.talk()

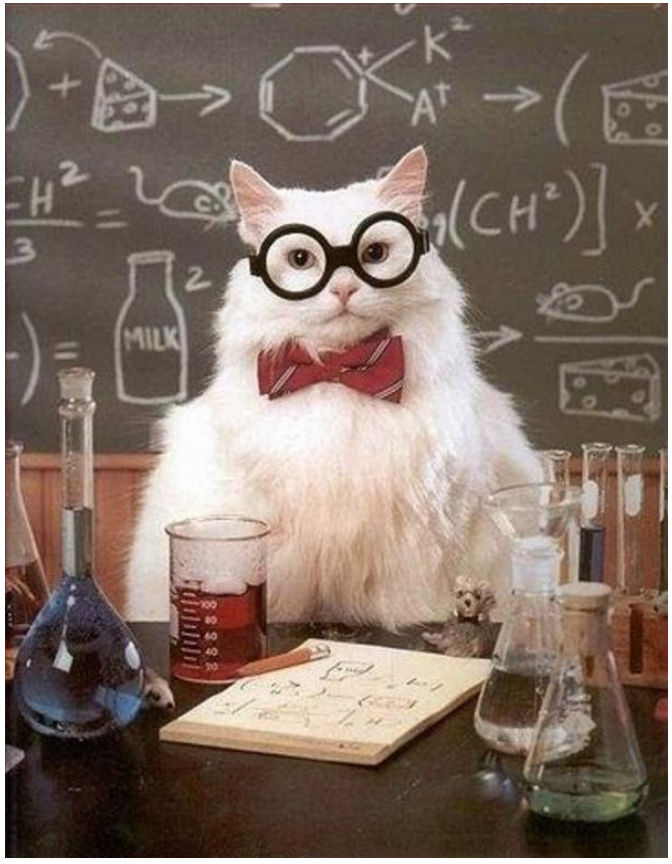
An Introduction to Deep Learning with TensorFlow

Peter Goldsborough

November 29, 2016

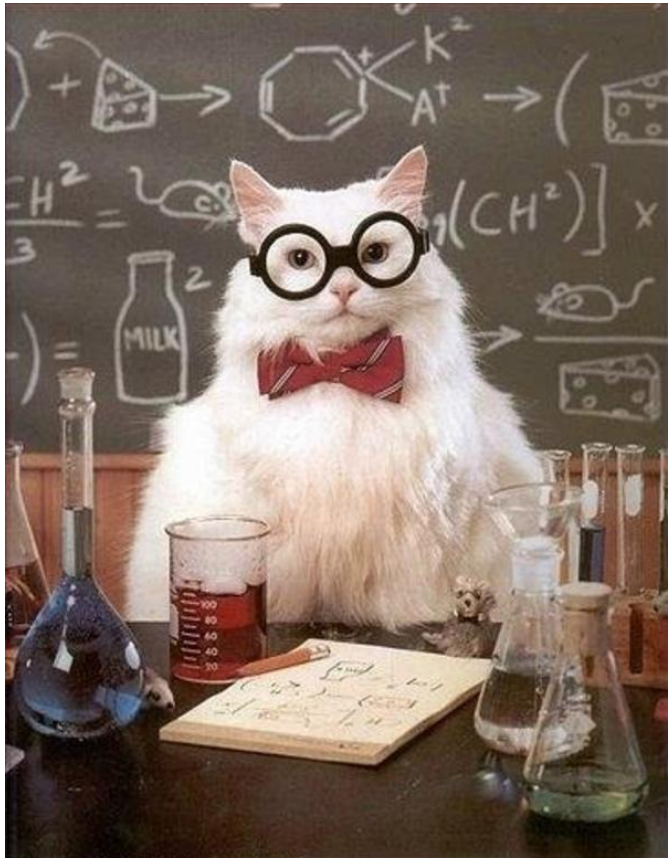
Table of Catents

Table of Catents

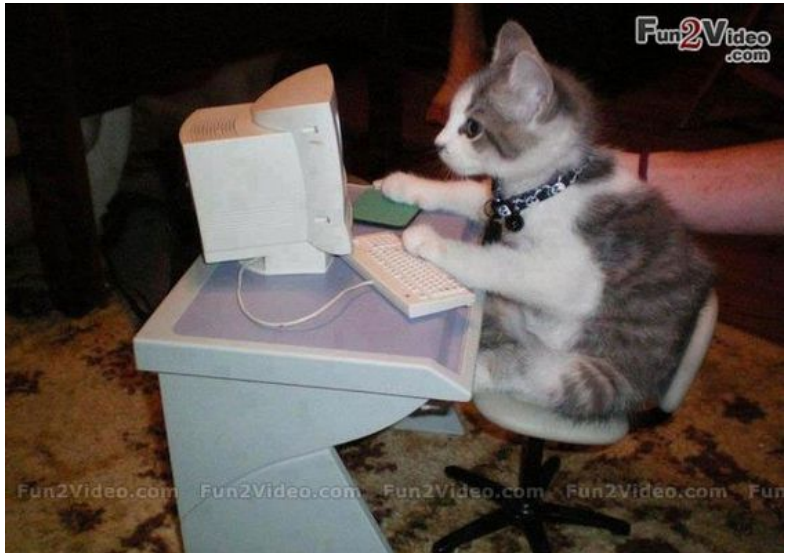


Theory

Table of Catents



Theory



Practice

Background

Background

- CS Student @ TUM

Background

- CS Student @ TUM
- Google & Bloomberg Intern

Background

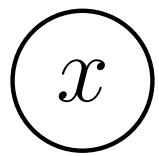
- CS Student @ TUM
- Google & Bloomberg Intern

Seminar Topic: *Deep Learning With TensorFlow*

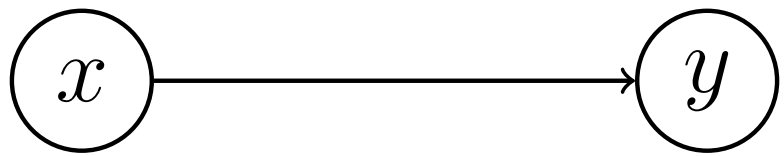
`github.com/peter-can-write/tensorflow-paper`

Neural Networks

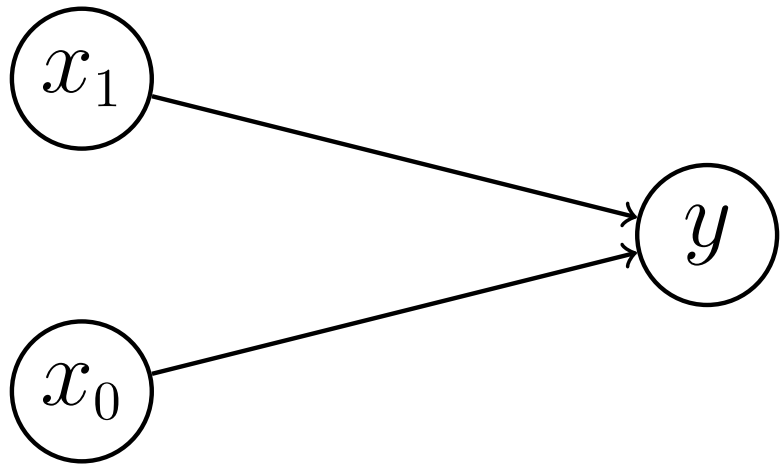
Neural Networks



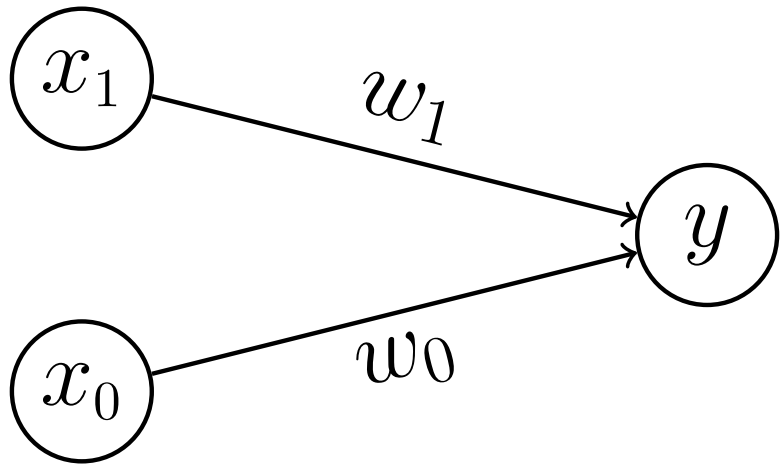
Neural Networks



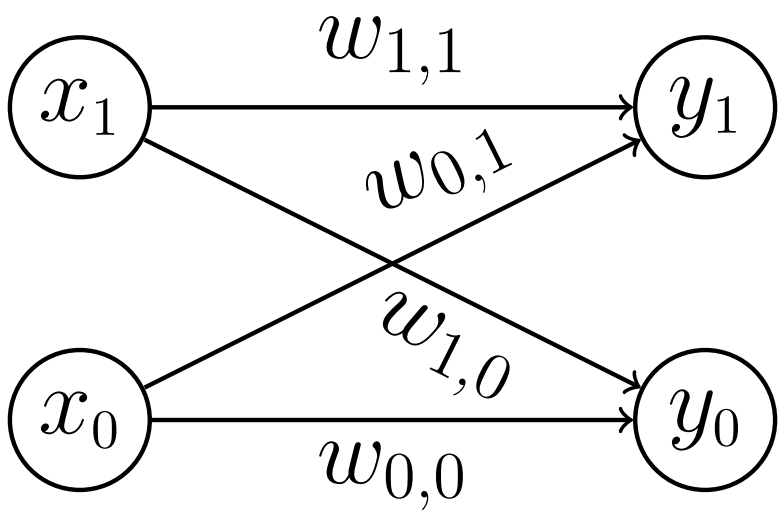
Neural Networks



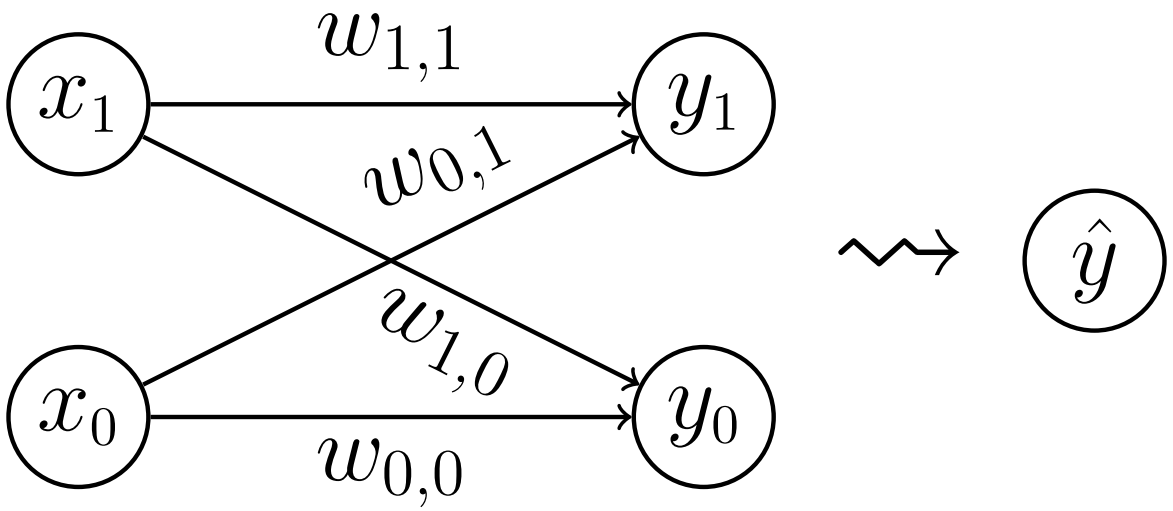
Neural Networks



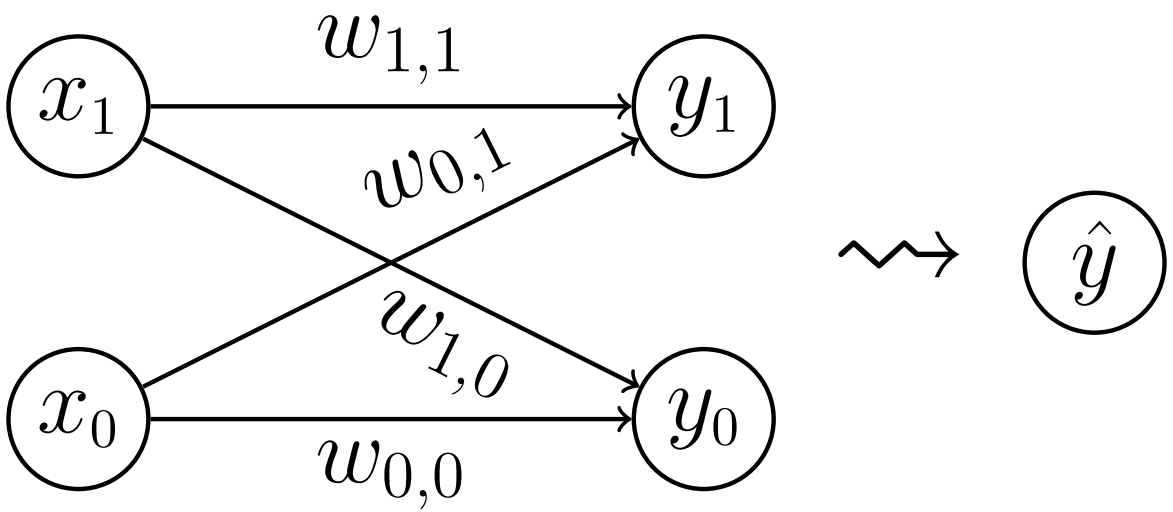
Neural Networks



Neural Networks

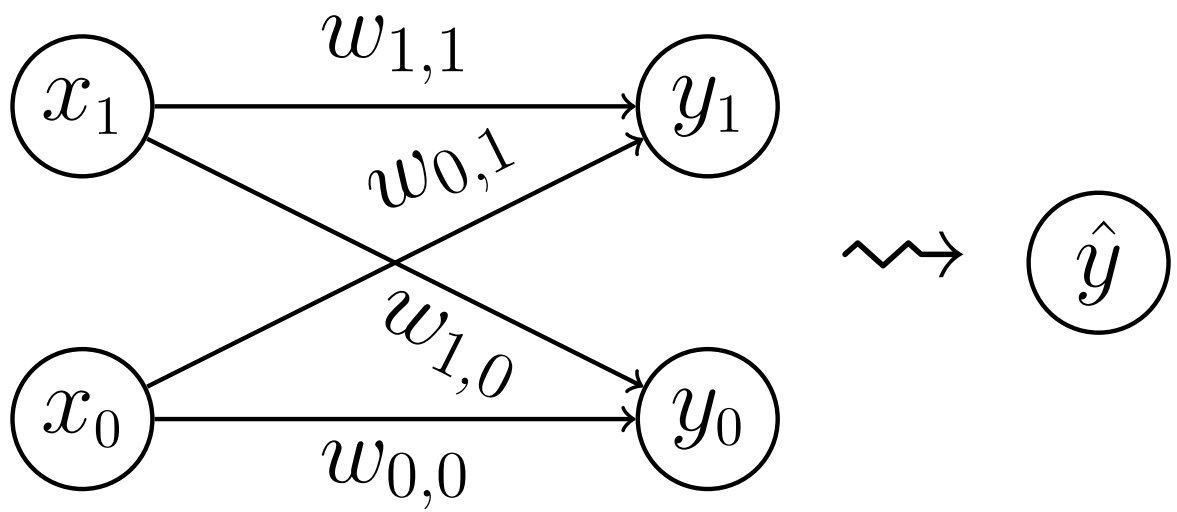


Neural Networks



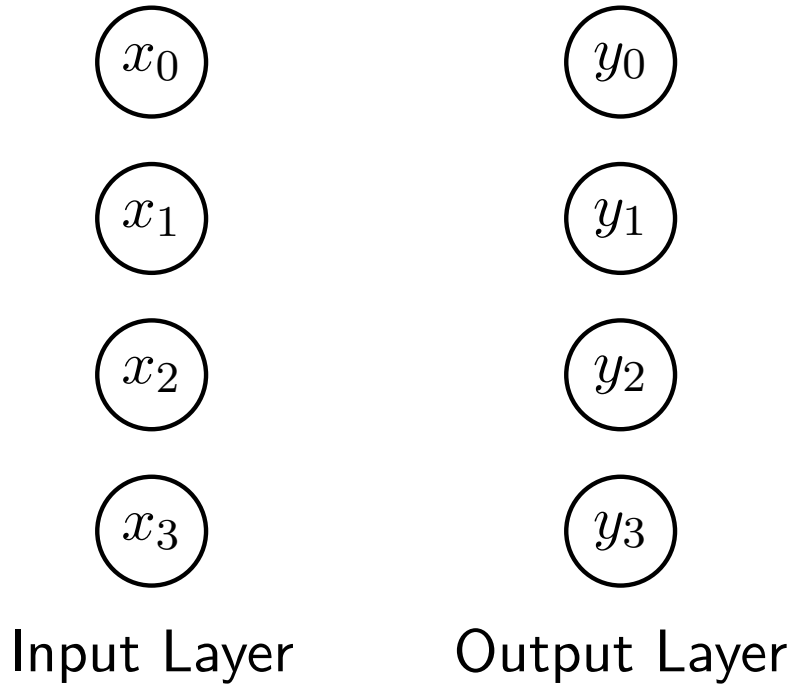
$$\underbrace{\begin{bmatrix} x_0 & x_1 \end{bmatrix}}_{\mathbf{x}} \times \underbrace{\begin{bmatrix} w_{0,0} & w_{0,1} \\ w_{1,0} & w_{1,1} \end{bmatrix}}_{\mathbf{W}} + \underbrace{\begin{bmatrix} b_0 \\ b_1 \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \end{bmatrix}}_{\mathbf{y}}$$

Neural Networks

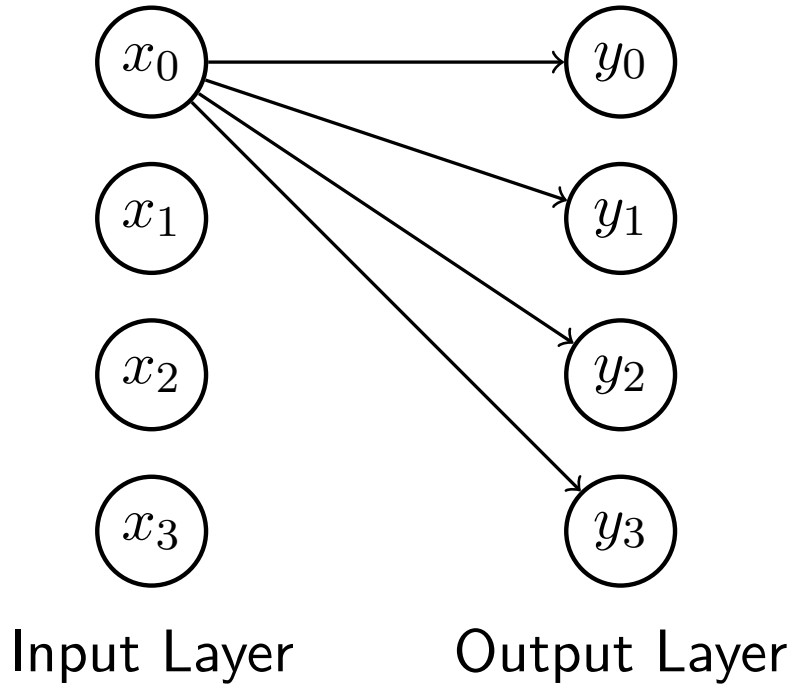


$$\underbrace{\begin{bmatrix} x_0 & x_1 \end{bmatrix}}_{\mathbf{x}} \times \underbrace{\begin{bmatrix} w_{0,0} & w_{0,1} \\ w_{1,0} & w_{1,1} \end{bmatrix}}_{\mathbf{W}} + \underbrace{\begin{bmatrix} b_0 \\ b_1 \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \end{bmatrix}}_{\mathbf{y}} \rightsquigarrow \hat{\mathbf{y}}$$

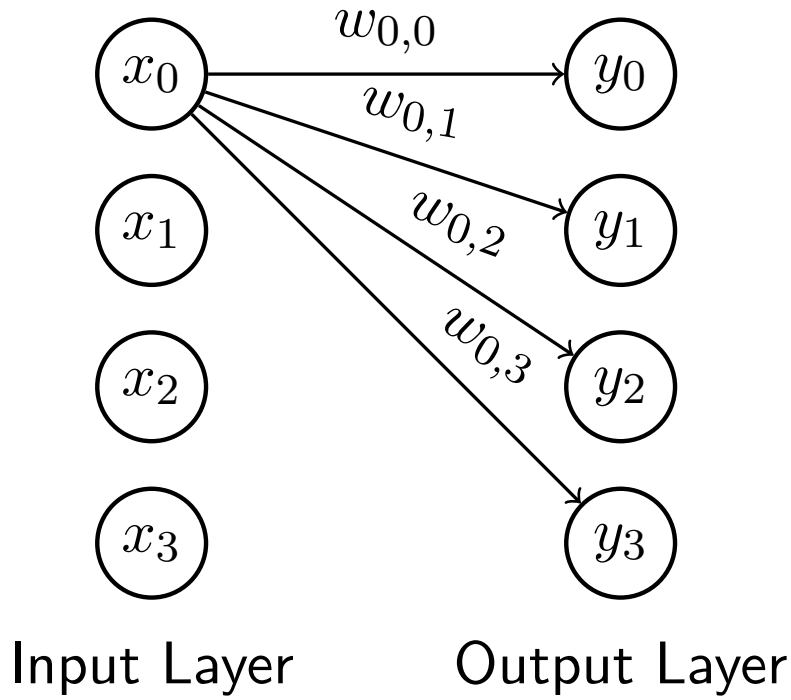
Neural Networks



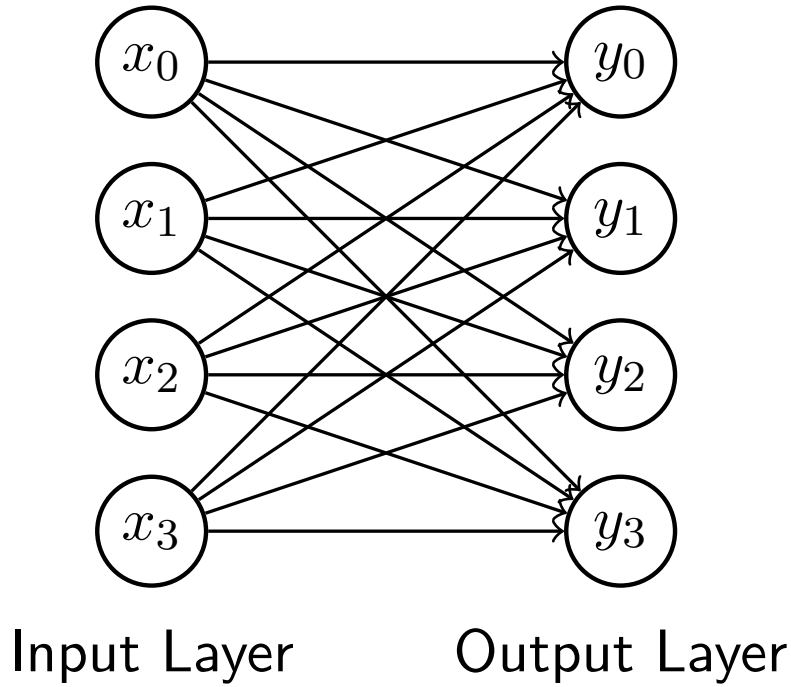
Neural Networks



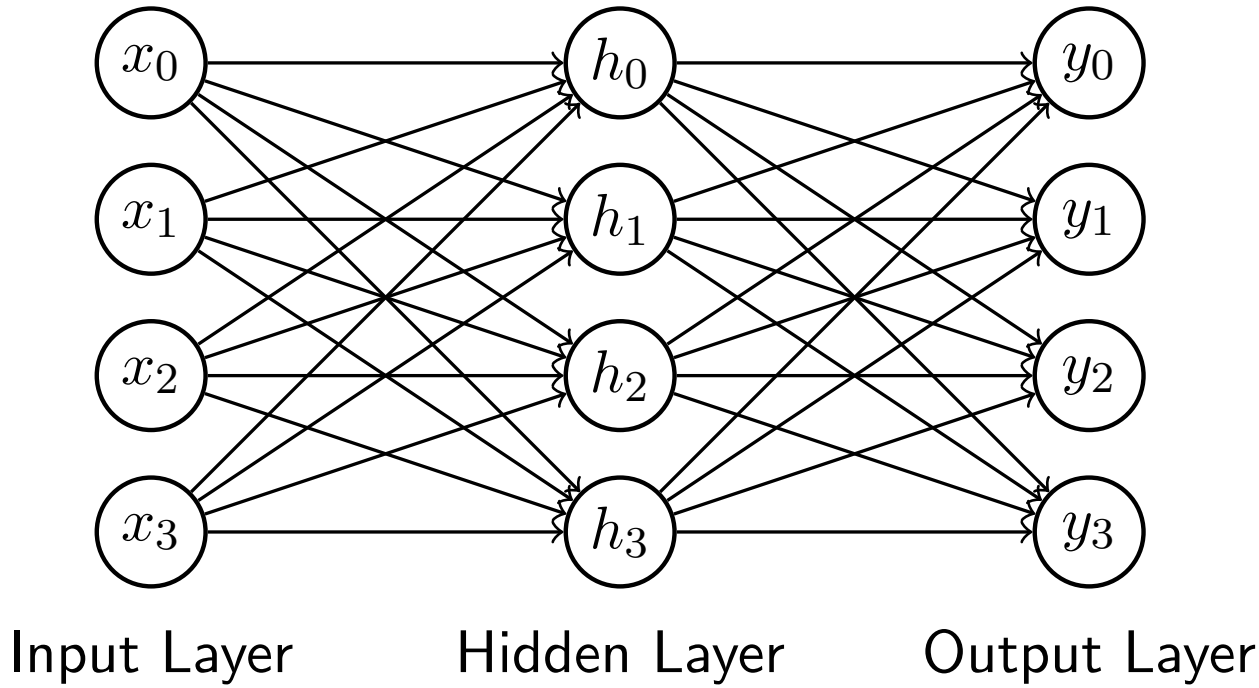
Neural Networks



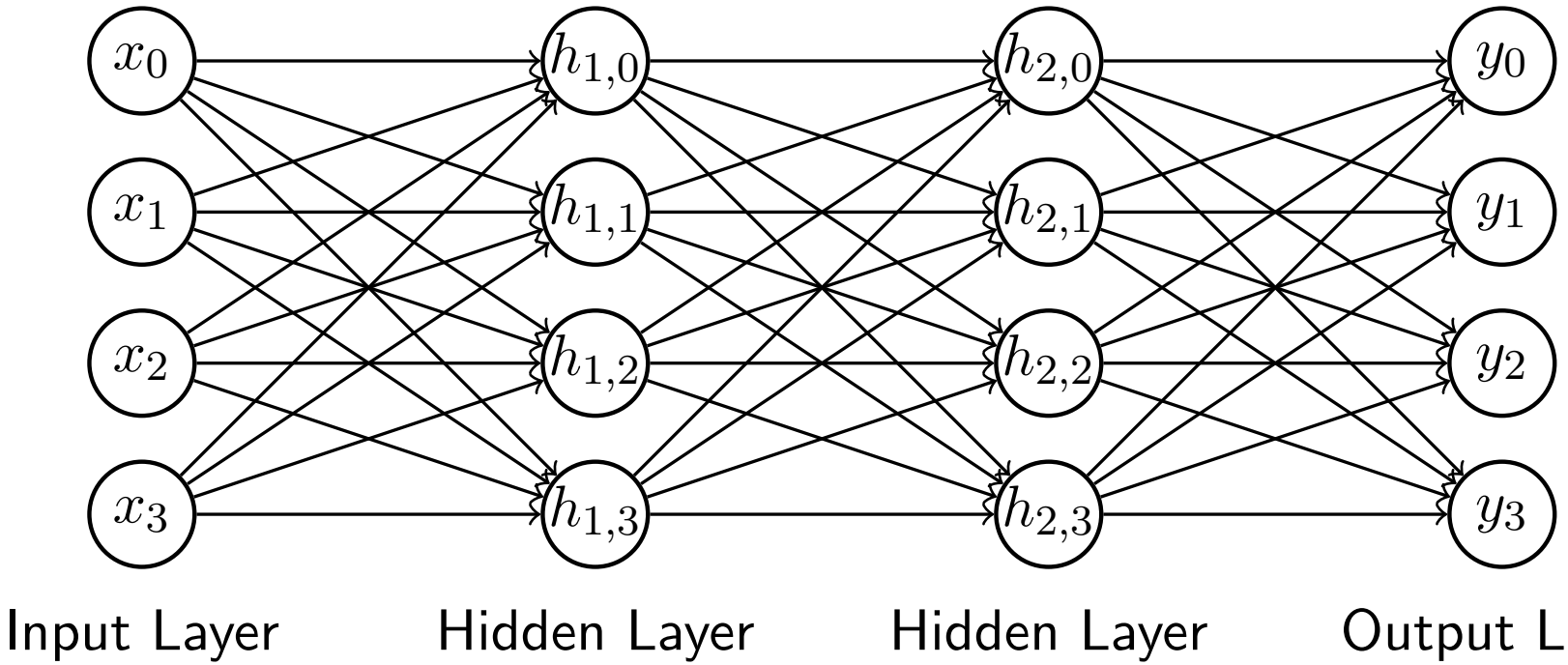
Neural Networks



Deep Neural Networks



Deep Neural Networks



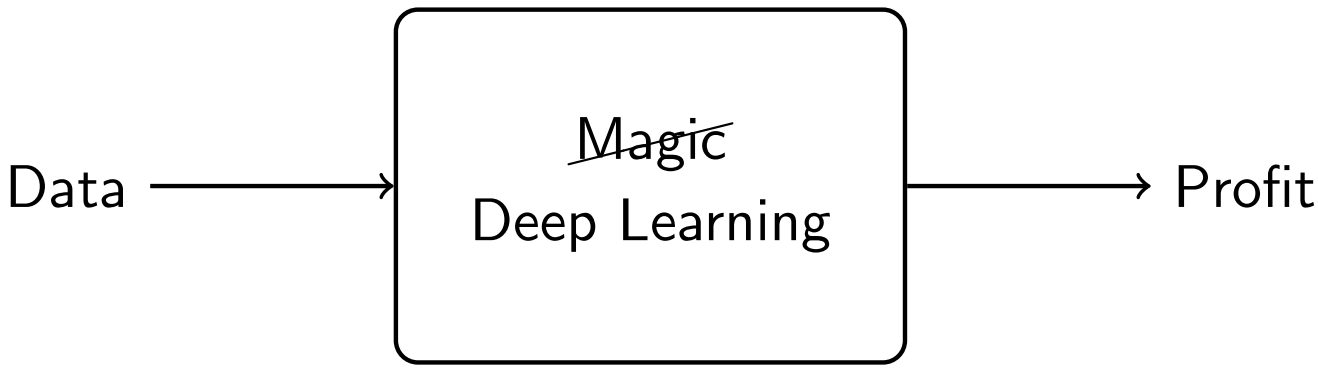
Deep Neural Networks

Deep Learning assumes that data is structured

Deep Learning assumes that data is structured

hierarchically

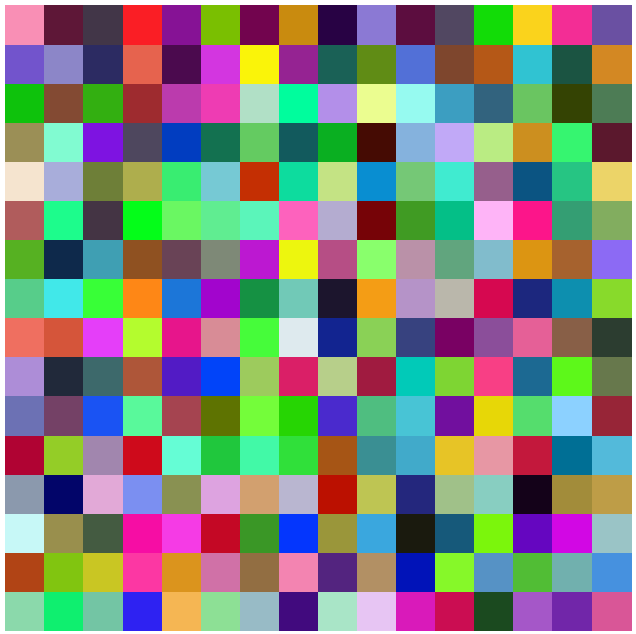
Deep Neural Networks



Deep Neural Networks

Deep Neural Networks

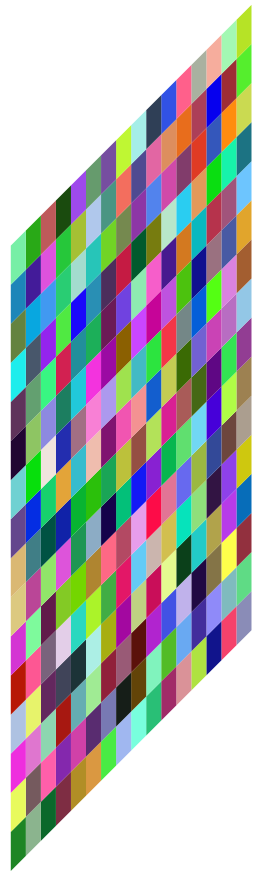
Deep Neural Networks



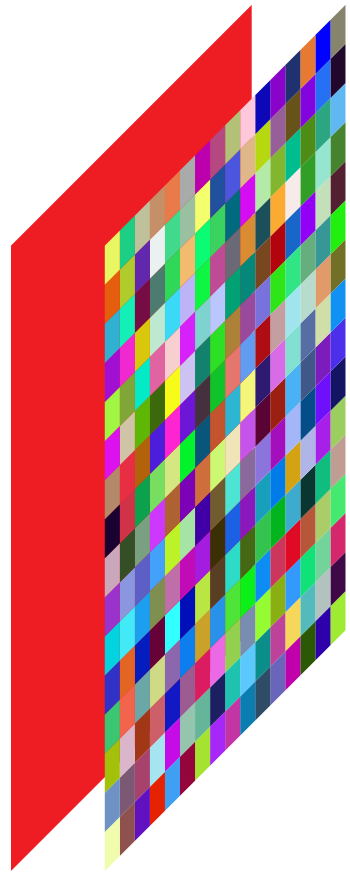
Deep Neural Networks

74	28	75	10	84	58	17	0	60	41	26	10	4	51	21	56
94	70	53	4	7	45	75	23	99	17	97	86	68	59	83	25
68	52	41	27	27	26	9	93	61	50	29	50	22	2	90	32
9	61	49	80	97	4	66	45	59	50	7	30	25	25	76	51
10	89	48	85	41	12	12	86	48	80	55	20	79	46	73	61
99	26	84	66	45	12	18	66	33	18	46	82	9	61	95	97
75	86	55	17	70	92	37	72	95	41	18	91	99	99	23	46
59	50	33	51	49	77	11	81	98	76	30	29	74	53	13	17
57	40	4	25	83	49	66	38	33	49	9	90	55	88	41	79
88	6	83	49	29	53	33	11	67	60	31	32	41	47	1	50
66	44	92	72	67	61	79	43	56	35	92	1	82	17	65	29
34	23	78	56	63	86	47	68	77	22	2	69	36	27	78	65
37	84	46	41	98	49	82	32	17	77	38	66	13	57	46	4
7	44	29	21	98	71	39	52	23	32	16	77	87	35	57	32
25	6	84	55	7	90	74	71	4	27	76	67	34	32	0	14
47	41	75	72	98	75	57	19	94	26	92	97	41	75	1	36

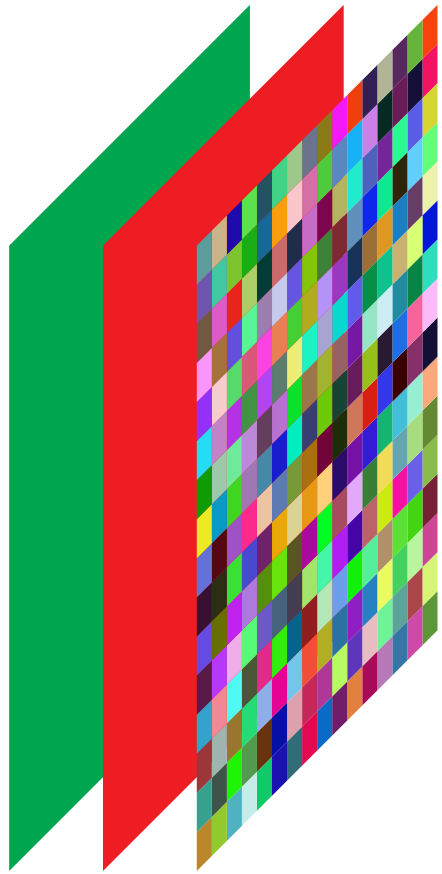
Deep Neural Networks



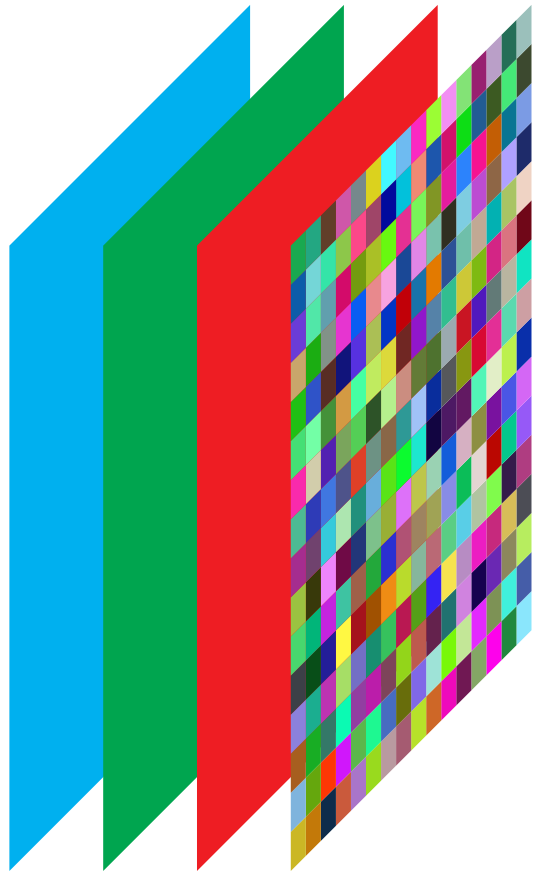
Deep Neural Networks



Deep Neural Networks



Deep Neural Networks



Deep Neural Networks

- We want to classify images into one of k classes

Deep Neural Networks

- We want to classify images into one of k classes
- Extract hierarchical features

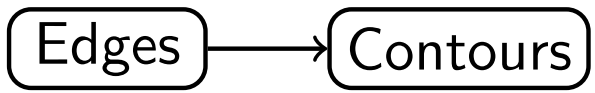
Deep Neural Networks

- We want to classify images into one of k classes
- Extract hierarchical features

Edges

Deep Neural Networks

- We want to classify images into one of k classes
- Extract hierarchical features

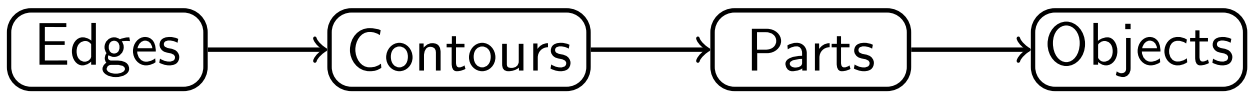


Deep Neural Networks

- We want to classify images into one of k classes
- Extract hierarchical features

Deep Neural Networks

- We want to classify images into one of k classes
- Extract hierarchical features



Deep Neural Networks

Why reinvent the wheel?

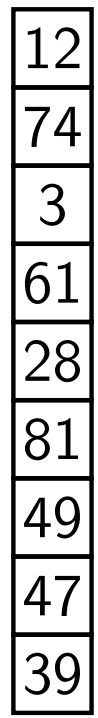
Deep Neural Networks

Why reinvent the wheel?

40	54	64
38	97	19
99	41	59

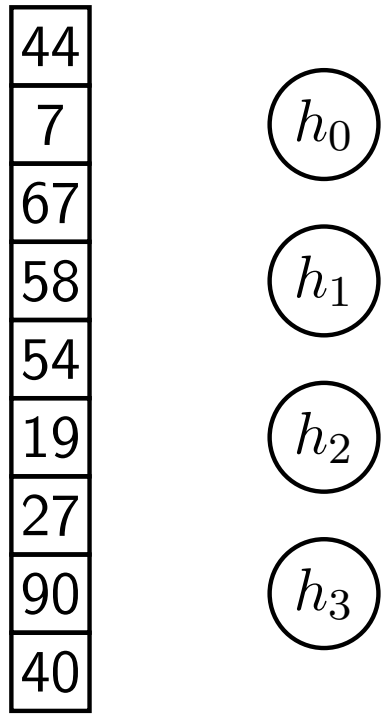
Deep Neural Networks

Why reinvent the wheel?

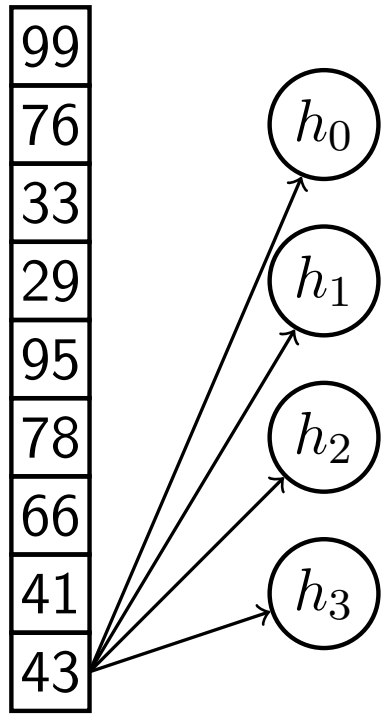


Deep Neural Networks

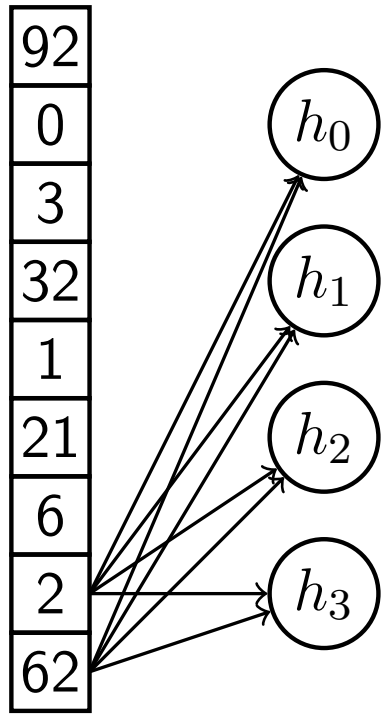
Why reinvent the wheel?



Why reinvent the wheel?

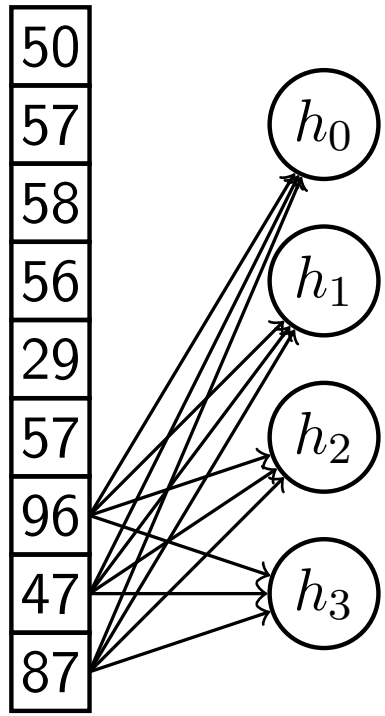


Why reinvent the wheel?



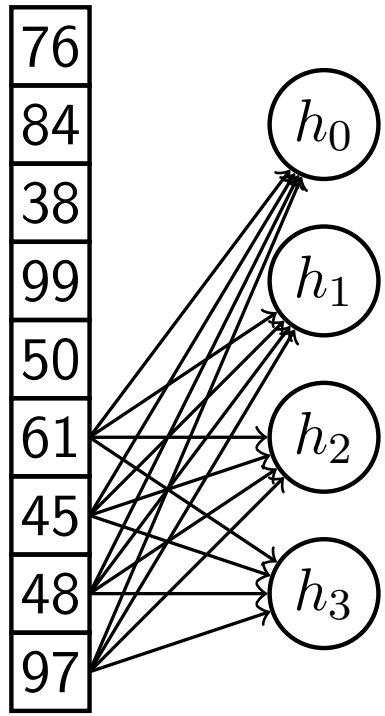
Deep Neural Networks

Why reinvent the wheel?



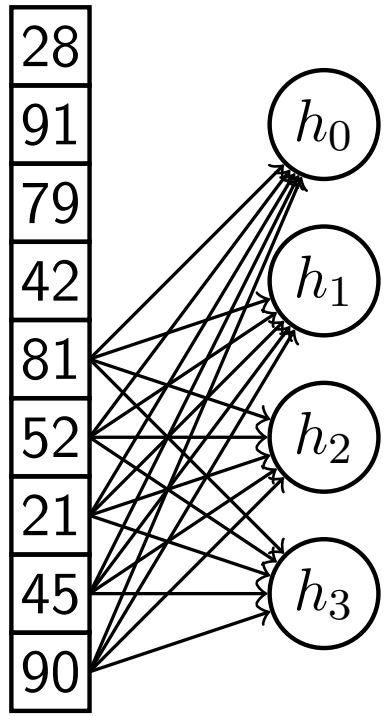
Deep Neural Networks

Why reinvent the wheel?

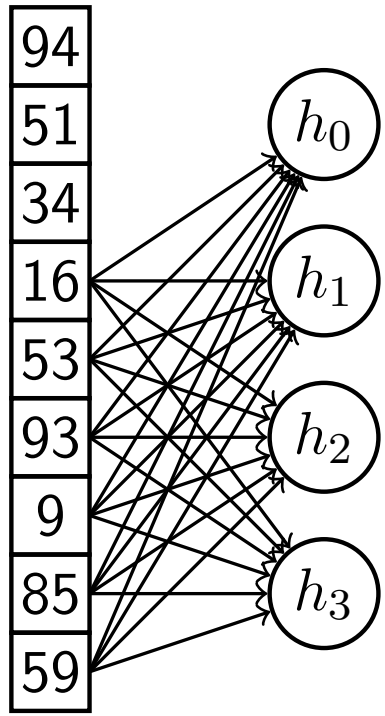


Deep Neural Networks

Why reinvent the wheel?

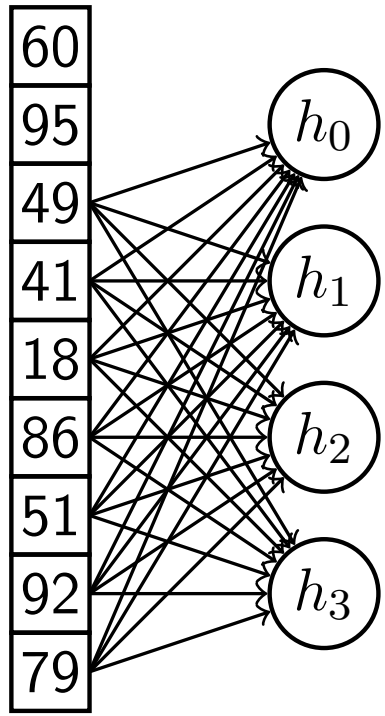


Why reinvent the wheel?



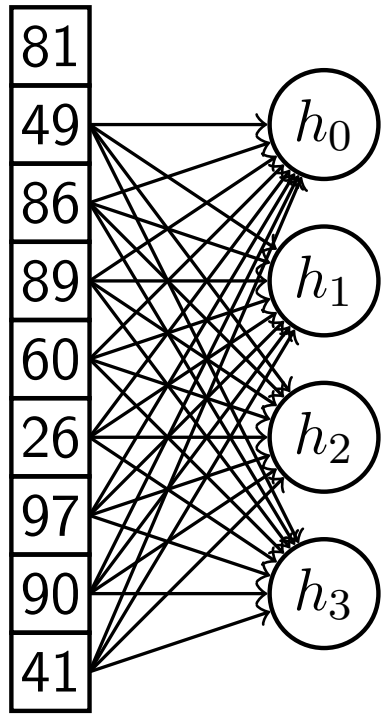
Deep Neural Networks

Why reinvent the wheel?



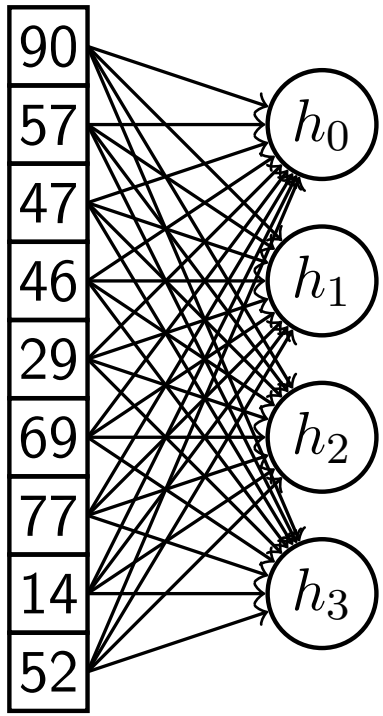
Deep Neural Networks

Why reinvent the wheel?



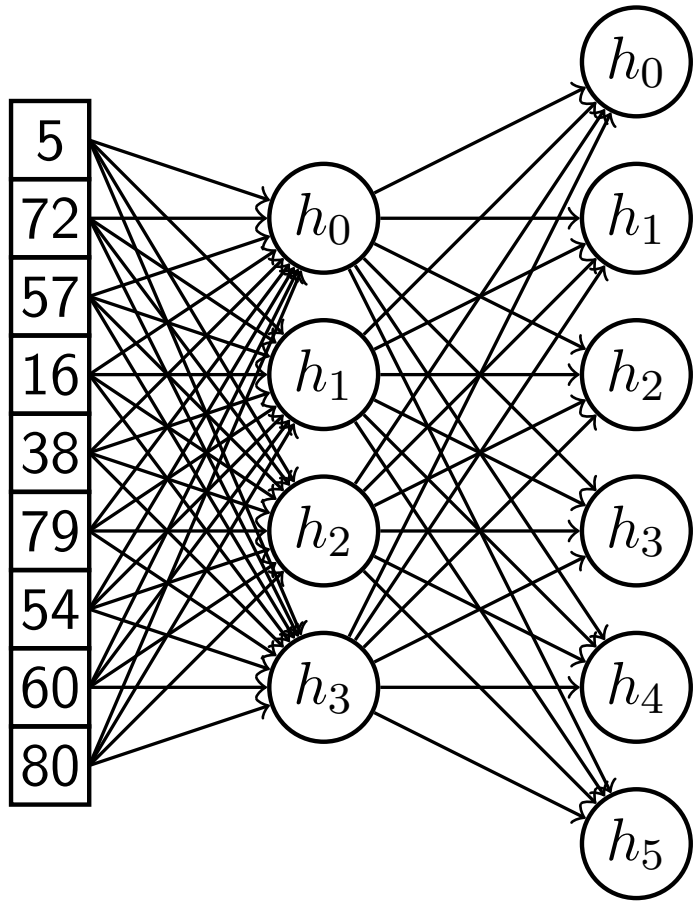
Deep Neural Networks

Why reinvent the wheel?



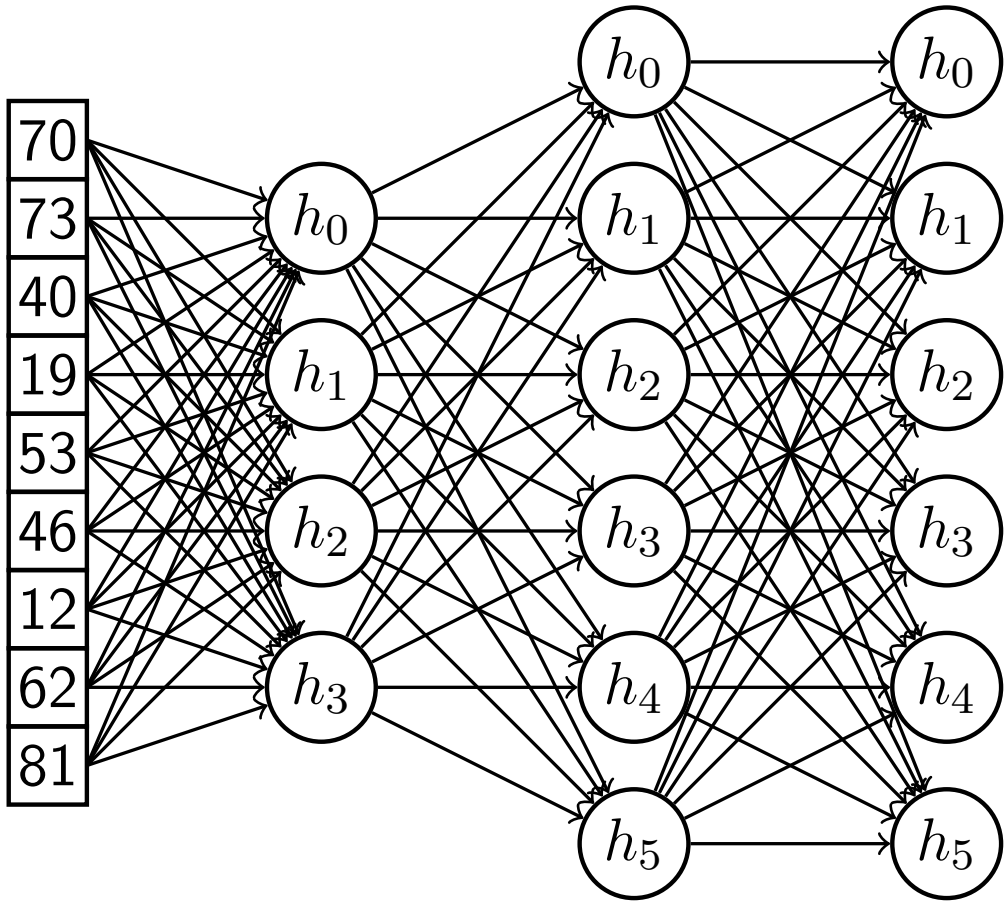
Deep Neural Networks

Why reinvent the wheel?



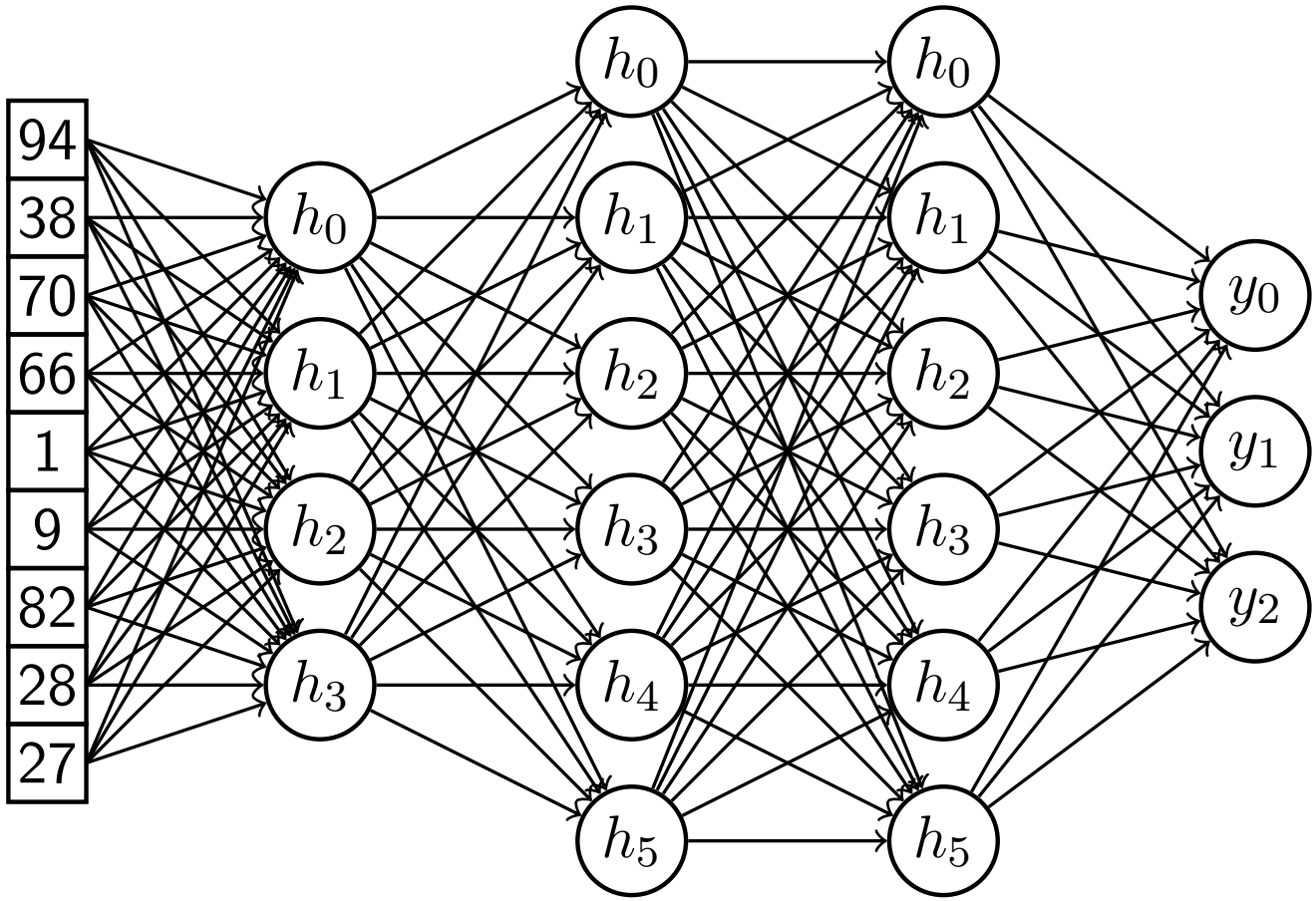
Deep Neural Networks

Why reinvent the wheel?



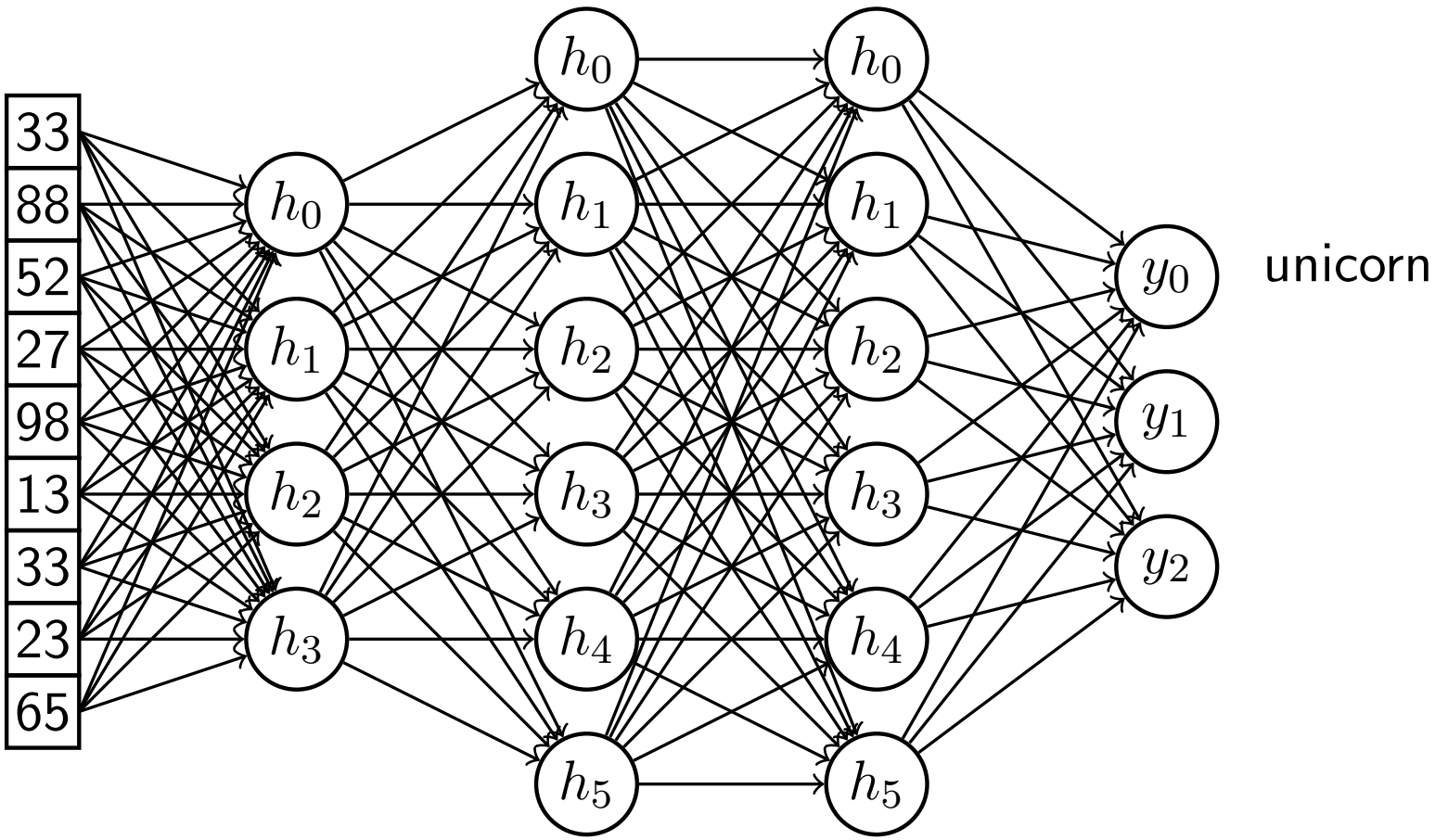
Deep Neural Networks

Why reinvent the wheel?



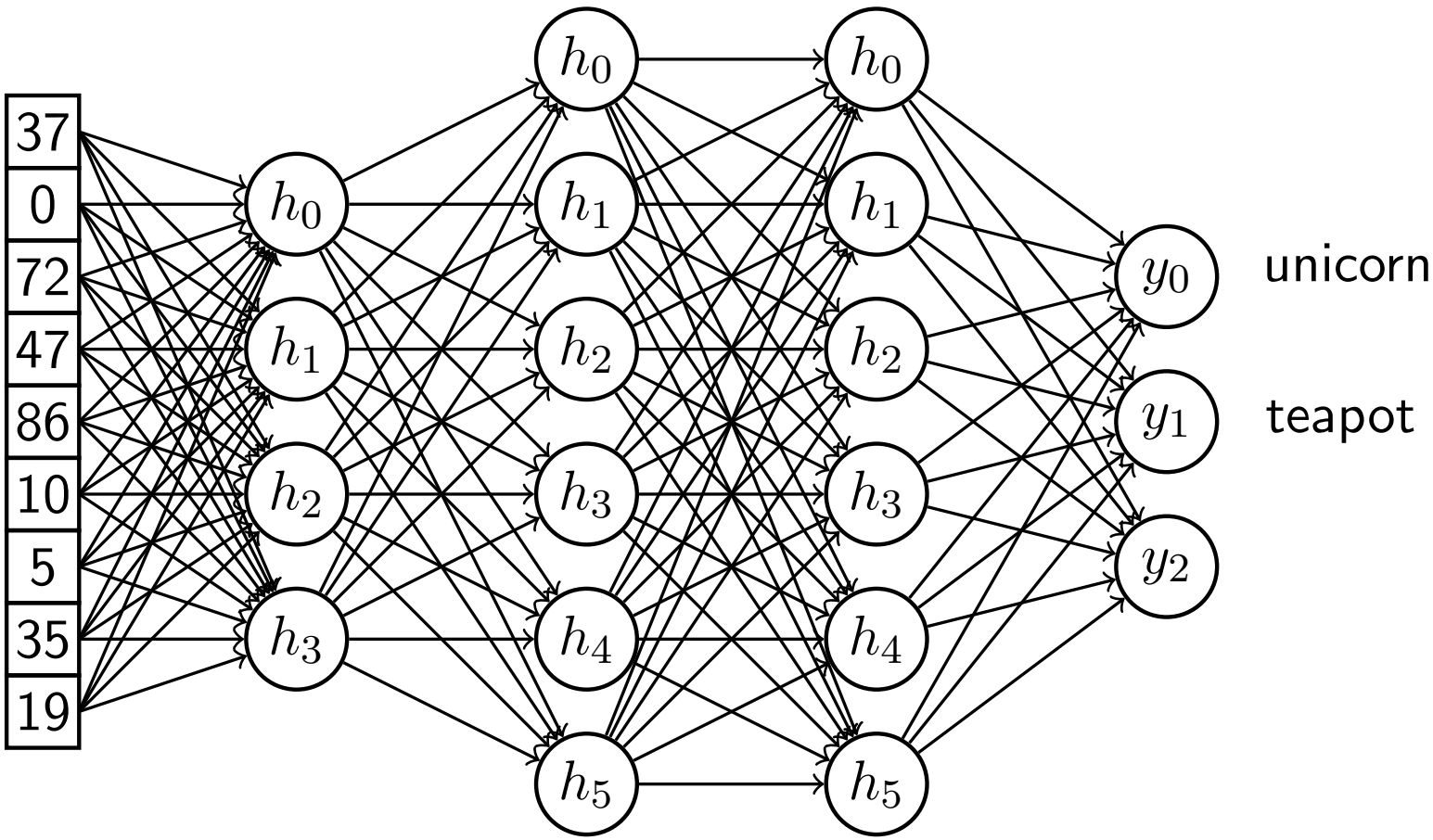
Deep Neural Networks

Why reinvent the wheel?



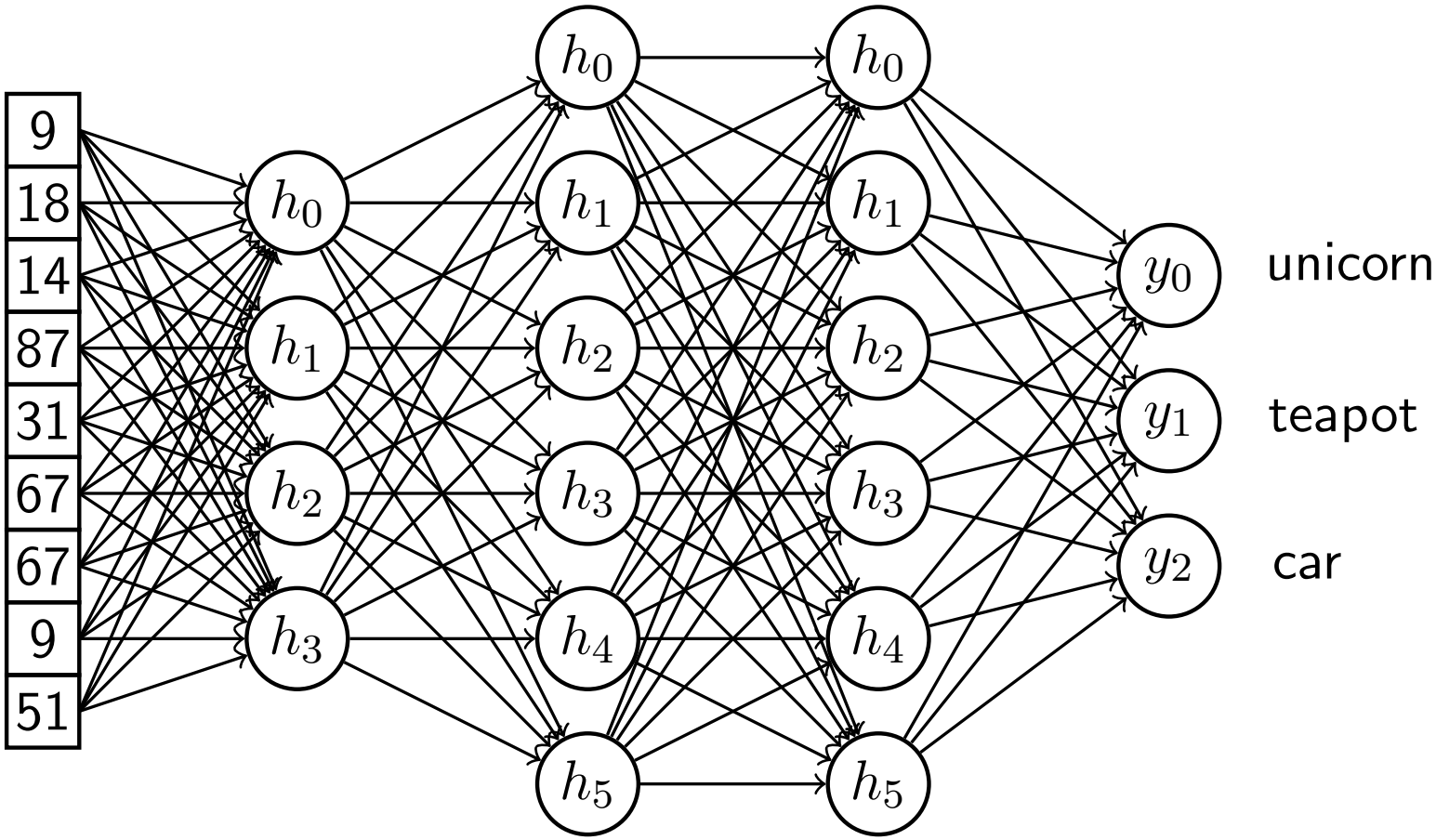
Deep Neural Networks

Why reinvent the wheel?



Deep Neural Networks

Why reinvent the wheel?



Time for ze kitteh

This is a cat ♥

Time for ze kitteh

Still a cat ♥♥

Time for ze kitteh

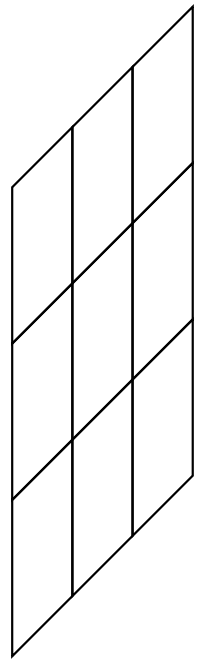
Half cat / half salad ♥♥♥

Time for ze kitteh

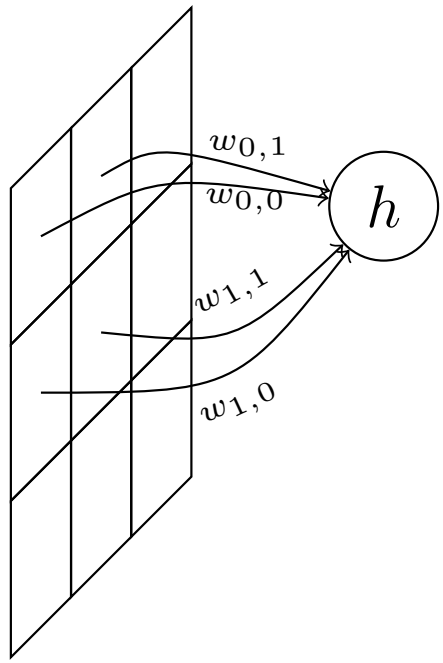
Many cats ♥♥♥♥

Weight Sharing

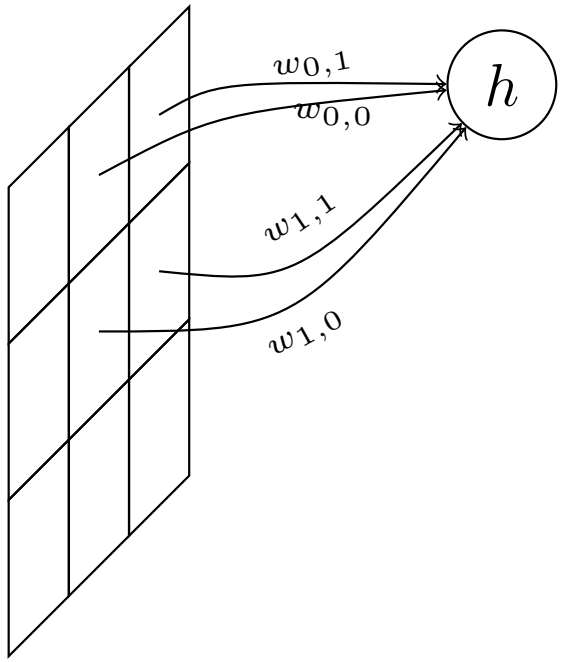
Weight Sharing



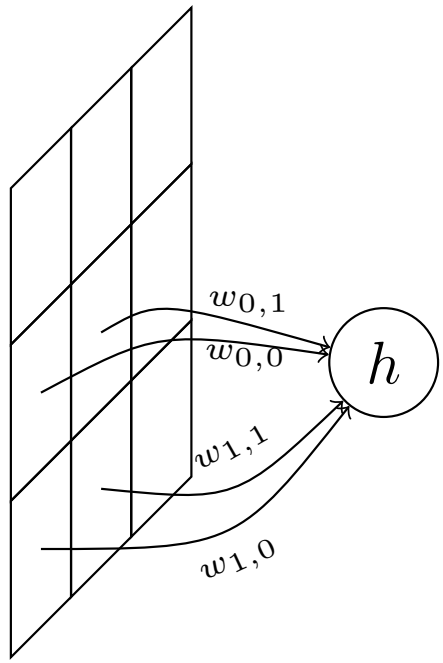
Weight Sharing



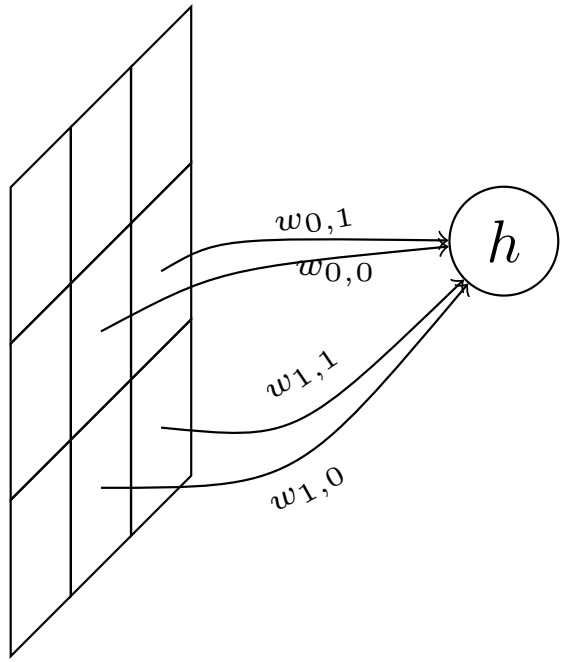
Weight Sharing



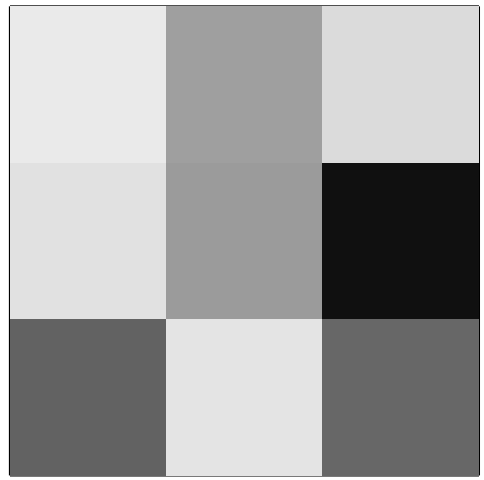
Weight Sharing



Weight Sharing



Convolutional Neural Networks: Mechanics



Image

Convolutional Neural Networks: Mechanics

0.4	0.9	0.1
0.7	0.2	0.6
0.8	0.3	0.5

Image

Convolutional Neural Networks: Mechanics

0.4	0.9	0.1
0.7	0.2	0.6
0.8	0.3	0.5

Image

5.7	2.4
3.1	0.9

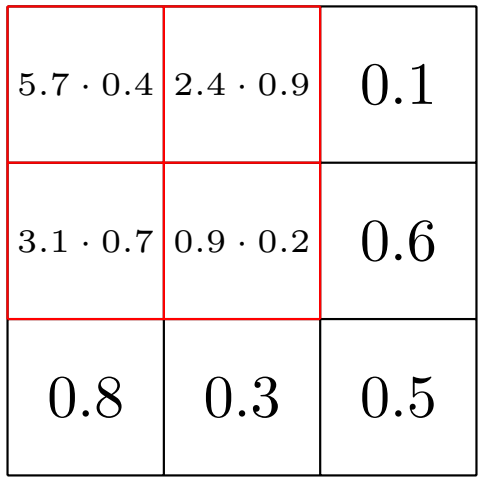
Kernel

Convolutional Neural Networks: Mechanics

$5.7 \cdot 0.4$	$2.4 \cdot 0.9$	0.1
$3.1 \cdot 0.7$	$0.9 \cdot 0.2$	0.6
0.8	0.3	0.5

Image

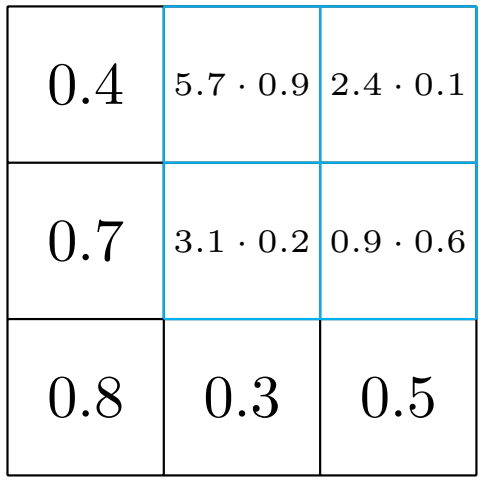
Convolutional Neural Networks: Mechanics



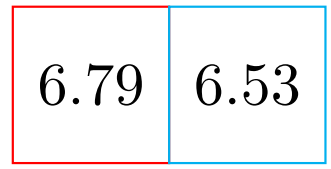
Image

Output

Convolutional Neural Networks: Mechanics

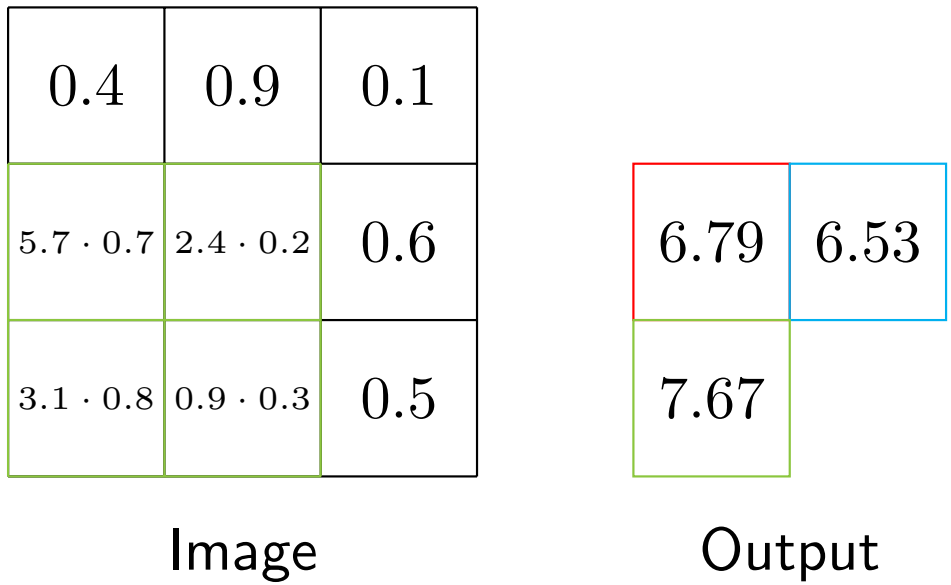


Image

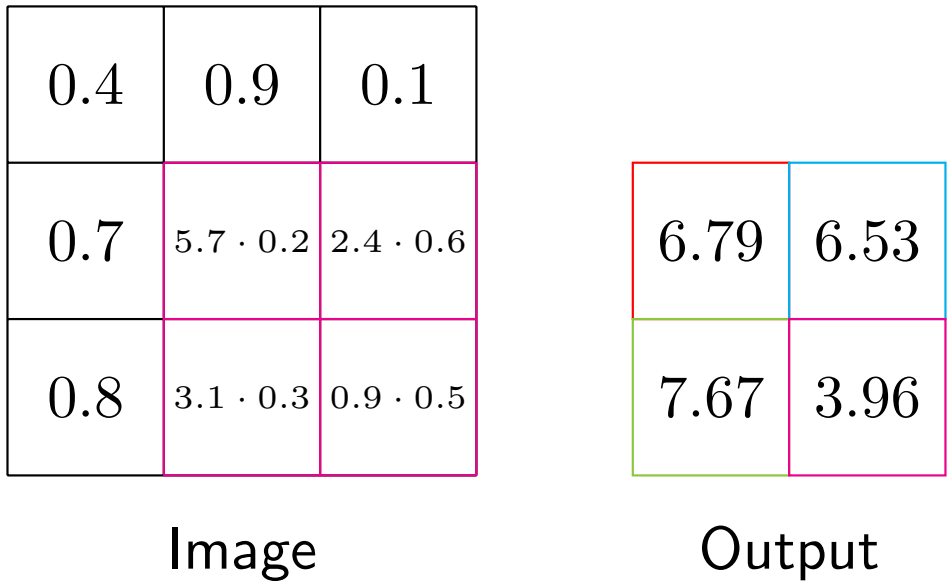


Output

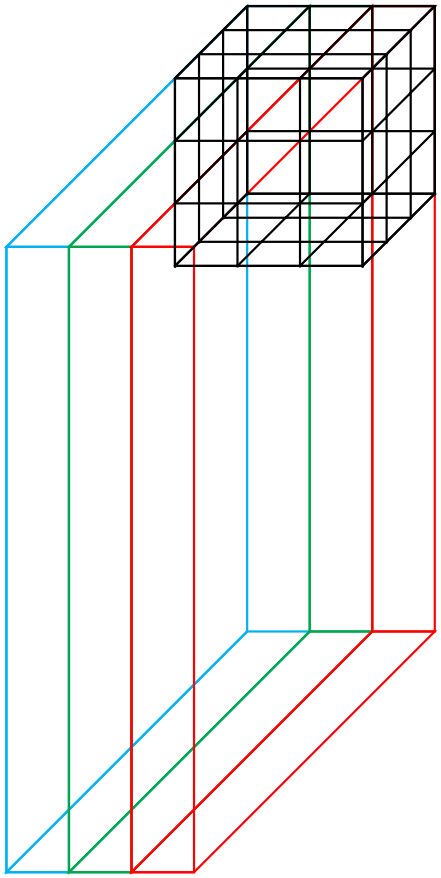
Convolutional Neural Networks: Mechanics



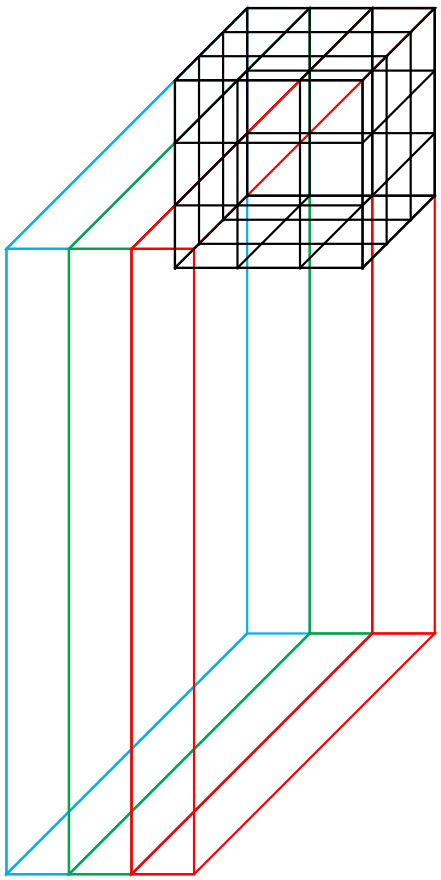
Convolutional Neural Networks: Mechanics

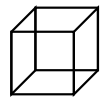


Convolutional Neural Networks: Mechanics

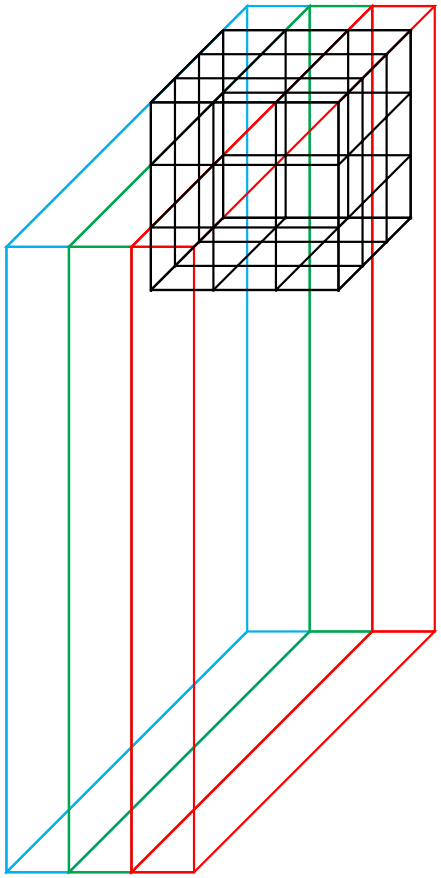


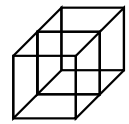
Convolutional Neural Networks: Mechanics



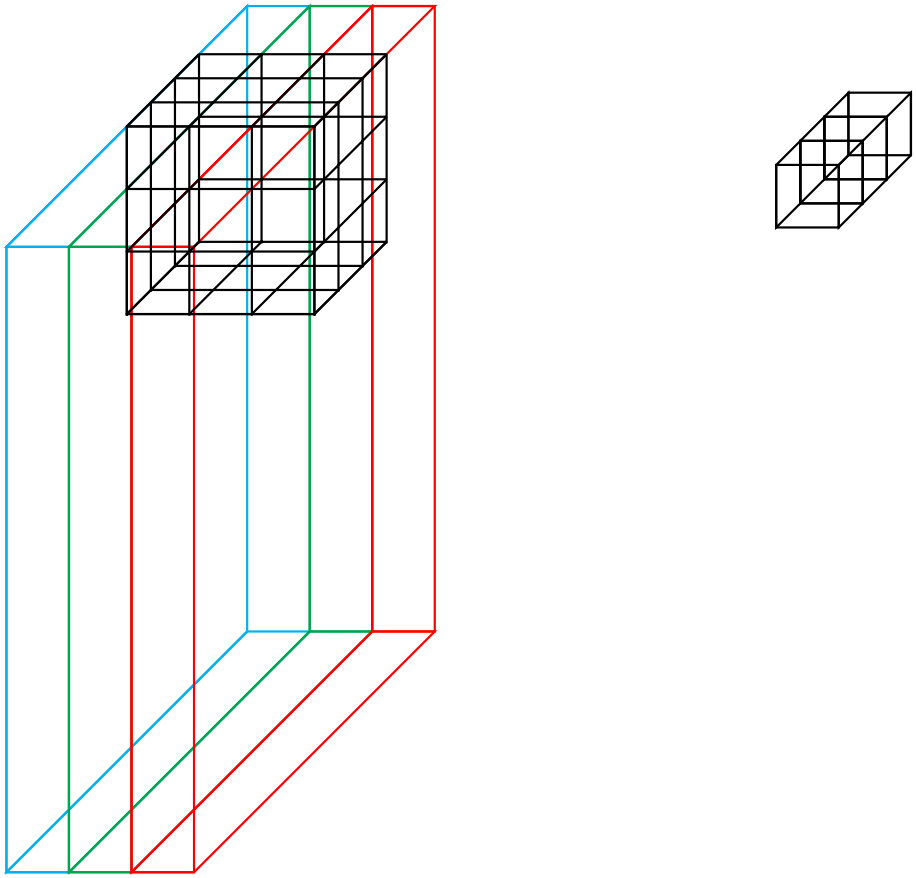


Convolutional Neural Networks: Mechanics

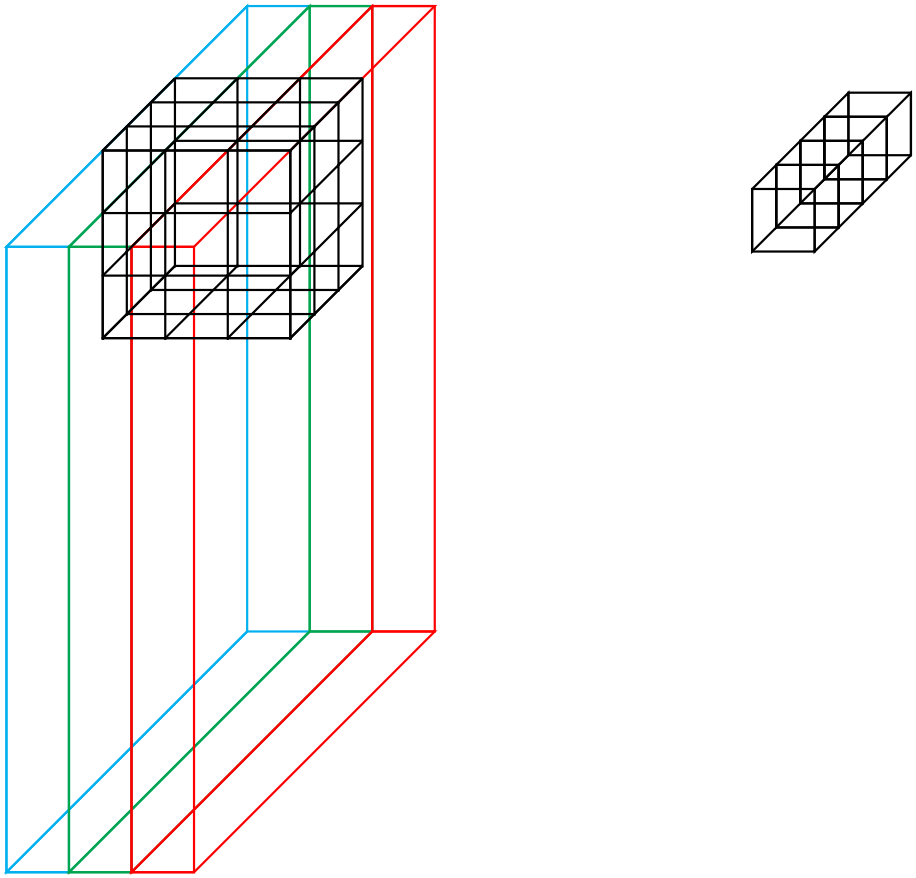




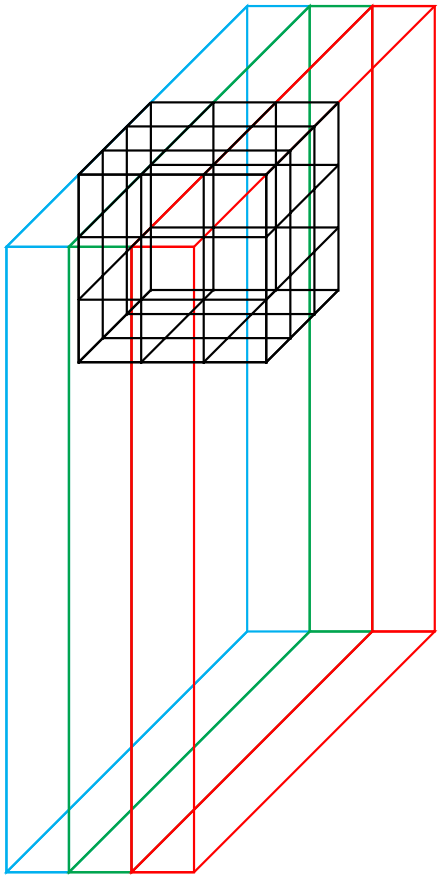
Convolutional Neural Networks: Mechanics

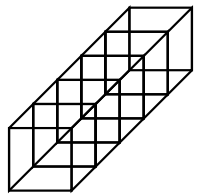


Convolutional Neural Networks: Mechanics

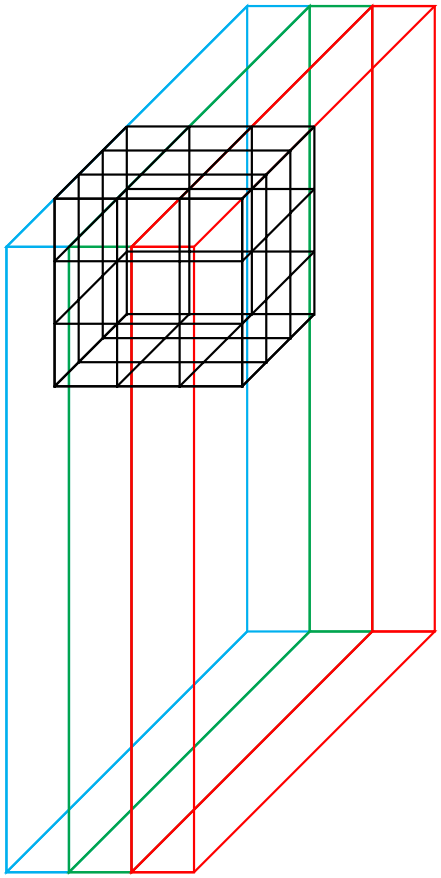


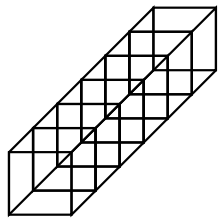
Convolutional Neural Networks: Mechanics



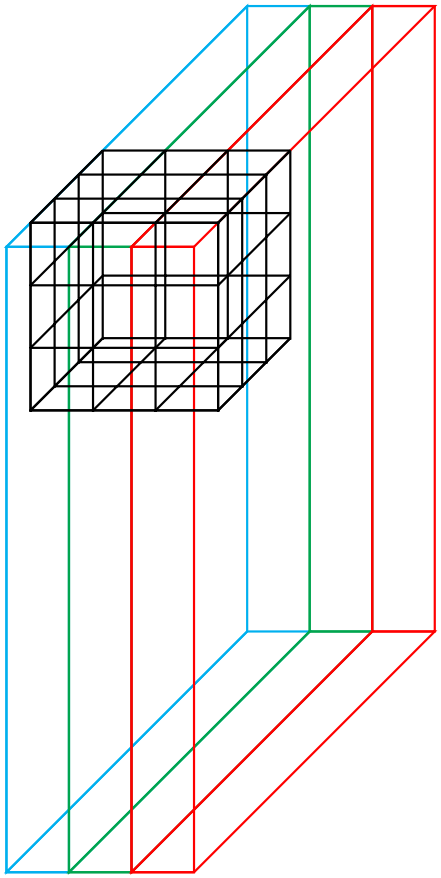


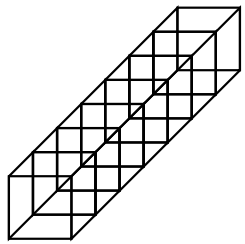
Convolutional Neural Networks: Mechanics



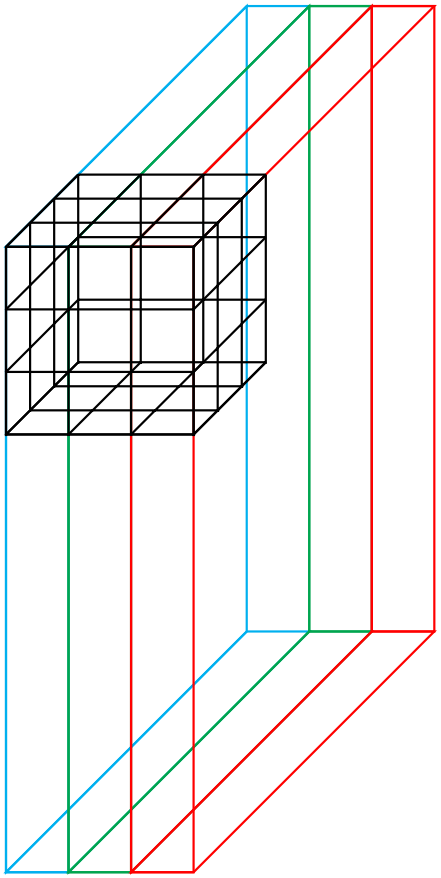


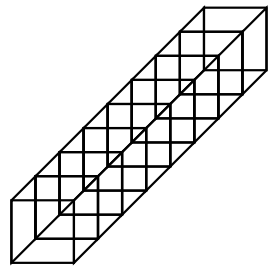
Convolutional Neural Networks: Mechanics



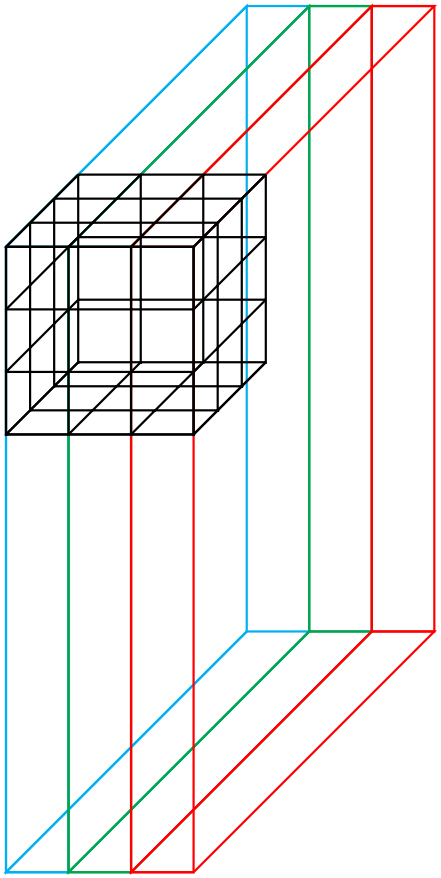


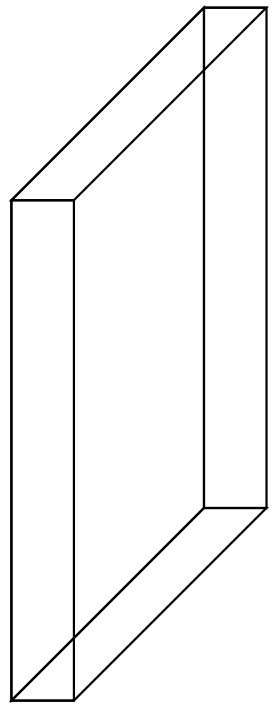
Convolutional Neural Networks: Mechanics



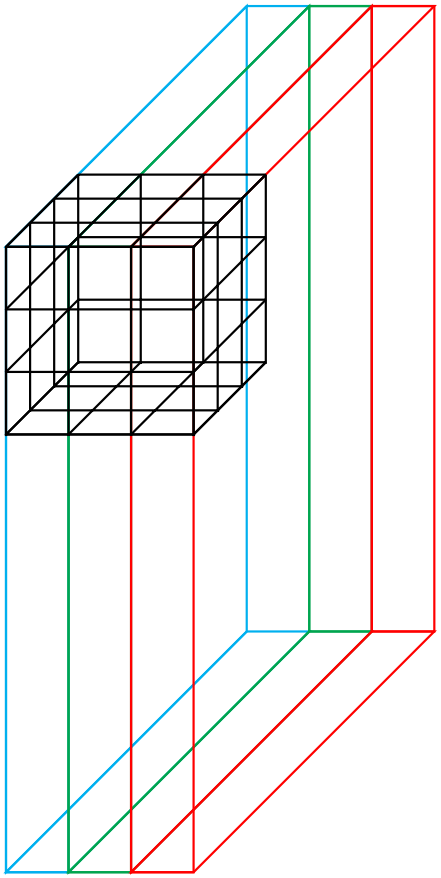


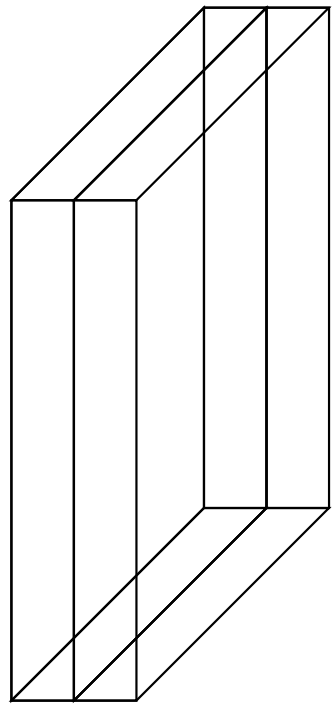
Convolutional Neural Networks: Mechanics



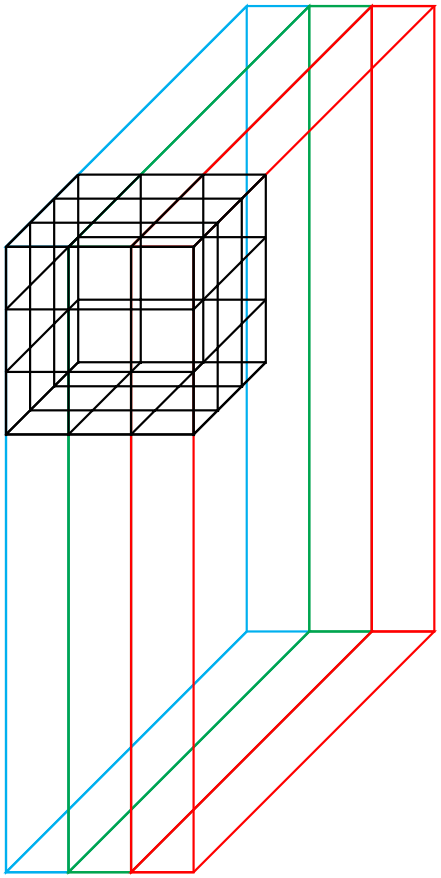


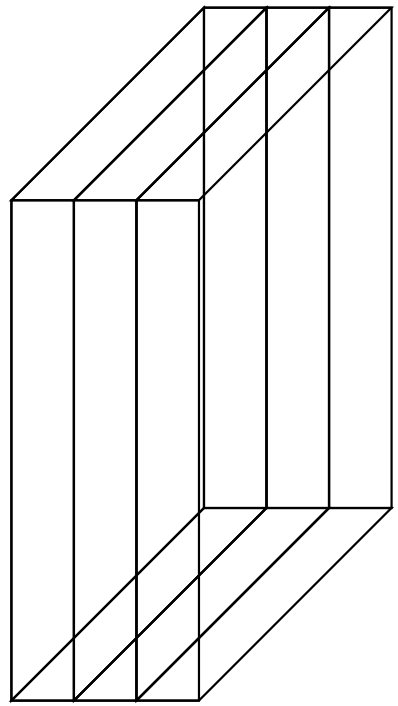
Convolutional Neural Networks: Mechanics



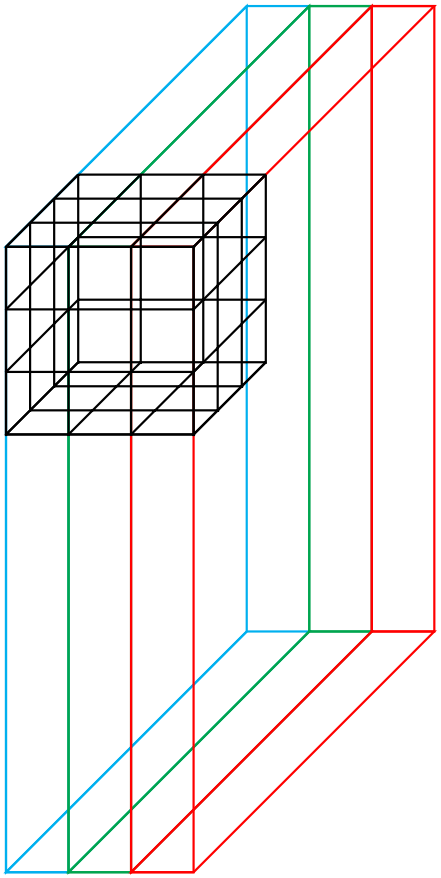


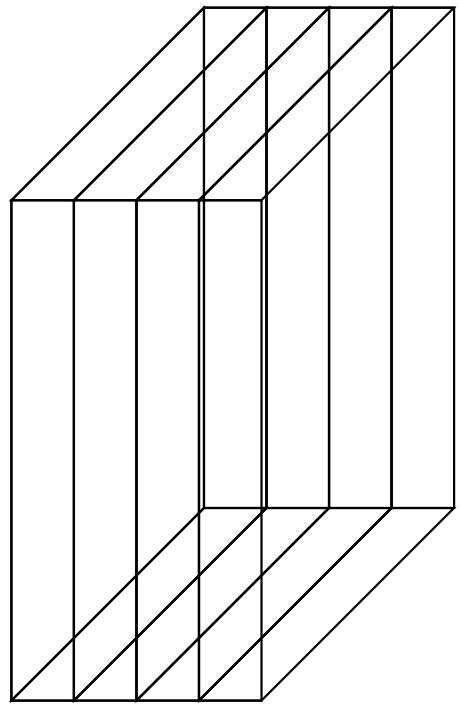
Convolutional Neural Networks: Mechanics





Convolutional Neural Networks: Mechanics





Recipe for a Convolutional Layer

Recipe for a Convolutional Layer

- Ingredients

Recipe for a Convolutional Layer

- Ingredients
  1. Image I with dimensions $w \times h \times d$

Recipe for a Convolutional Layer

- Ingredients
  1. Image I with dimensions $w \times h \times d$
  2. A kernel (filter) K of size $k \times l \times m$

Recipe for a Convolutional Layer

- Ingredients
  1. Image I with dimensions $w \times h \times d$
  2. A kernel (filter) K of size $k \times l \times m$
- Cooking

Recipe for a Convolutional Layer

- Ingredients
  1. Image I with dimensions $w \times h \times d$
  2. A kernel (filter) K of size $k \times l \times m$
- Cooking
  - Put the image into the oven at 150°C

Recipe for a Convolutional Layer

- Ingredients
  1. Image I with dimensions $w \times h \times d$
  2. A kernel (filter) K of size $k \times l \times m$
- Cooking
  - Don't put the image into the oven at 150°C

Recipe for a Convolutional Layer

- Ingredients
  1. Image I with dimensions $w \times h \times d$
  2. A kernel (filter) K of size $k \times l \times m$
- Cooking
  - Don't put the image into the oven at 150°C
  - Slide the kernel across the image

Recipe for a Convolutional Layer

- Ingredients
  1. Image I with dimensions $w \times h \times d$
  2. A kernel (filter) K of size $k \times l \times m$
- Cooking
  - Don't put the image into the oven at 150°C
  - Slide the kernel across the image
  - Compute the "dot product" for each configuration

Convolutional Neural Networks: Pooling

Convolutional Neural Networks: Pooling

66	2
6	32

Convolutional Neural Networks: Pooling

66	2
6	32

Convolutional Neural Networks: Pooling

50	17
66	2

Convolutional Neural Networks: Pooling

9	50
33	66

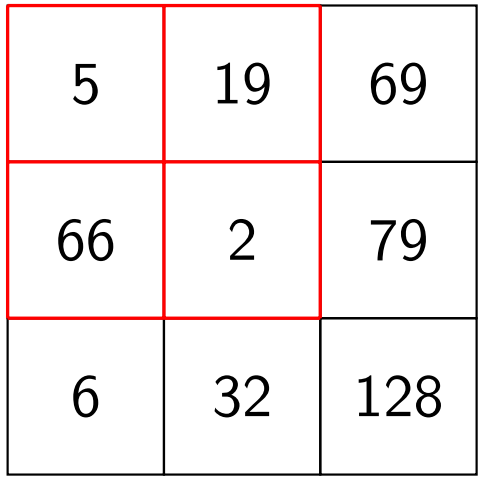
Convolutional Neural Networks: Pooling

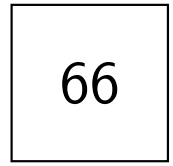
5	19	69
66	2	79
6	32	128

Convolutional Neural Networks: Pooling

5	19	69
66	2	79
6	32	128

Convolutional Neural Networks: Pooling





Convolutional Neural Networks: Pooling

5	19	69
66	2	79
6	32	128

66	79

Convolutional Neural Networks: Pooling

5	19	69
66	2	79
6	32	128

66	79
66	

Convolutional Neural Networks: Pooling

5	19	69
66	2	79
6	32	128

66	79
66	128

Convolutional Neural Networks: Pooling

5	19	69
66	2	79
6	32	128

66	79
66	128

- *Pooling* achieves translational invariance

Convolutional Neural Networks: Pooling

5	19	69
66	2	79
6	32	128

66	79
66	128

- *Pooling* achieves translational invariance
- A form of downsampling

Convolutional Neural Networks: Pooling

5	19	69
66	2	79
6	32	128

66	79
66	128

- *Pooling* achieves translational invariance
- A form of downsampling
- Other pooling functions possible

Convolutional Neural Networks: Architecture

```
INPUT -> [CONV+ -> POOL?]+ -> FC+ -> OUTPUT
```

TensorFlow

TensorFlow

- An open source deep learning library

TensorFlow

- An open source deep learning library
- Released by Google in November 2015

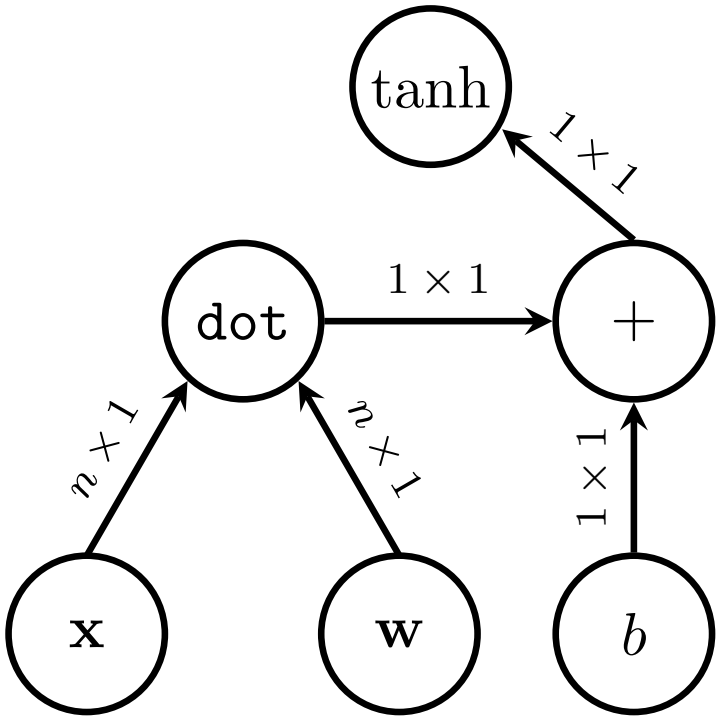
TensorFlow

- An open source deep learning library
- Released by Google in November 2015
- Especially suited to:

  "Large-scale machine learning on heterogeneous distributed systems"

Computational Paradigms

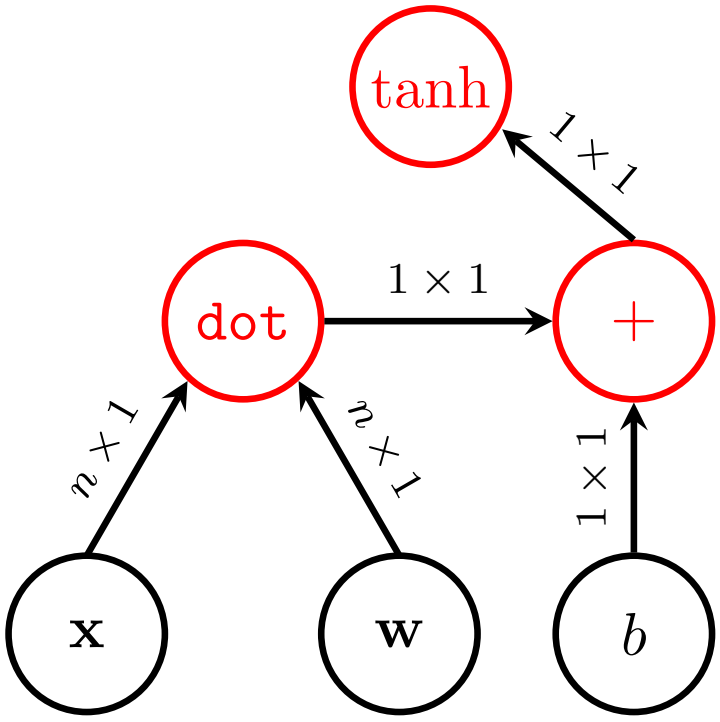
Computational Paradigms



$$\hat{y} = \tanh(\mathbf{x}^\top \mathbf{w} + b)$$

Computational Graphs

Computational Paradigms

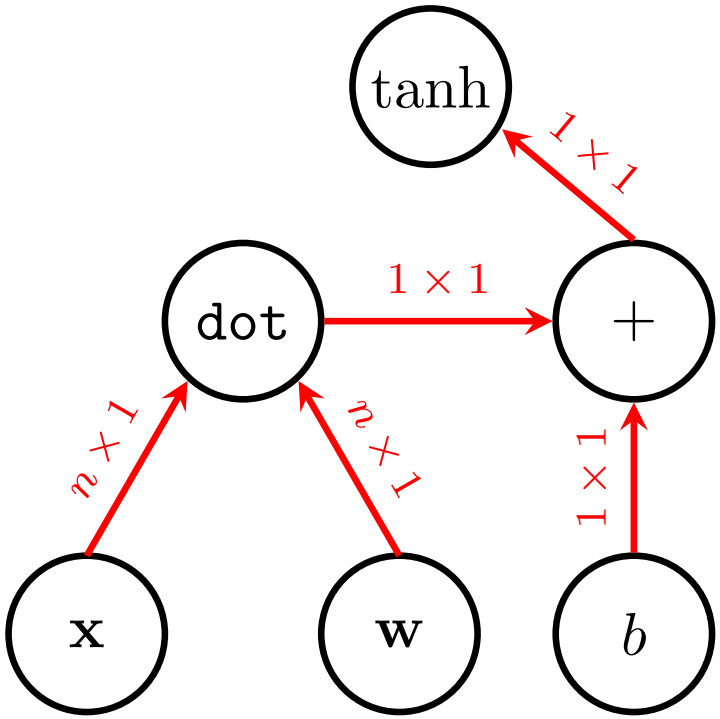


$$\hat{y} = \tanh(\mathbf{x}^\top \mathbf{w} + b)$$

Computational Graphs

1. Operations

Computational Paradigms

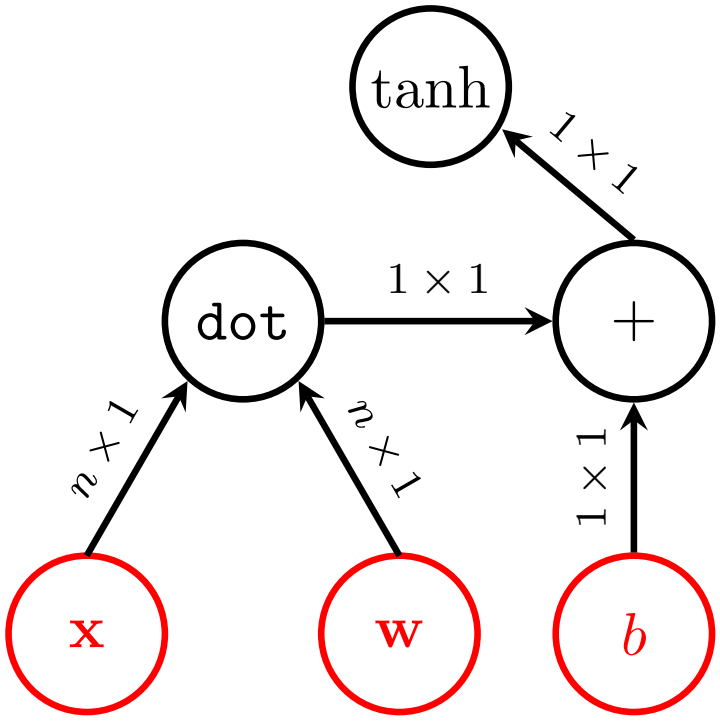


$$\hat{y} = \tanh(\mathbf{x}^\top \mathbf{w} + b)$$

Computational Graphs

1. Operations
2. Tensors

Computational Paradigms

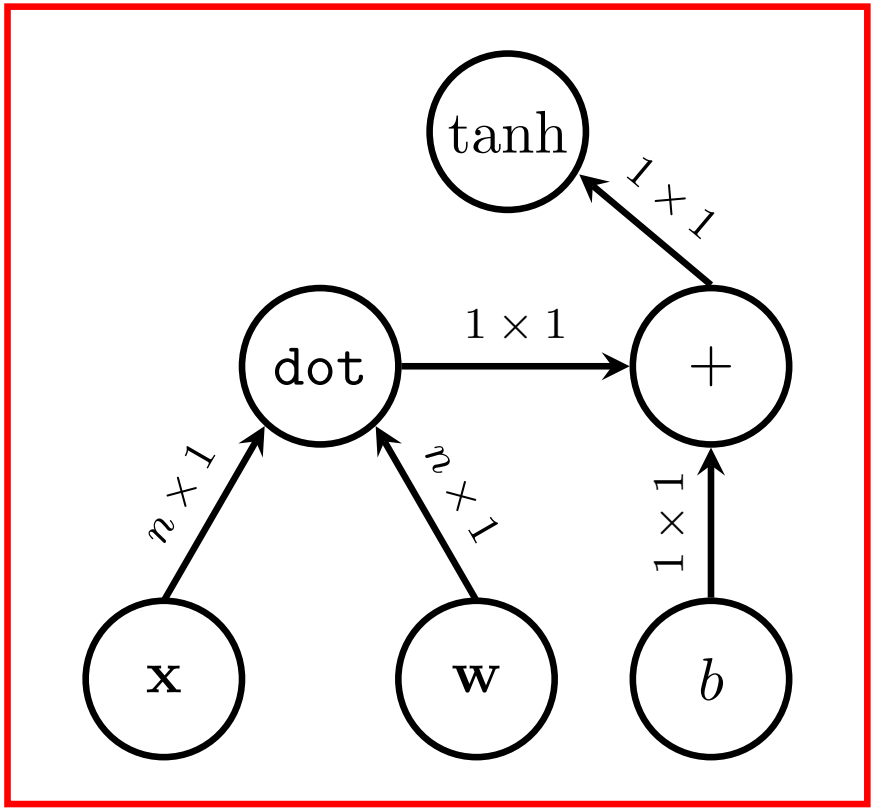


$$\hat{y} = \tanh(\mathbf{x}^\top \mathbf{w} + b)$$

Computational Graphs

1. Operations
2. Tensors
3. Variables

Computational Paradigms



$$\hat{y} = \texttt{session.run}(\tanh(\mathbf{x}^\top \mathbf{w} + b))$$

Computational Graphs

1. Operations
2. Tensors
3. Variables
4. Sessions

Execution Model

Execution Model

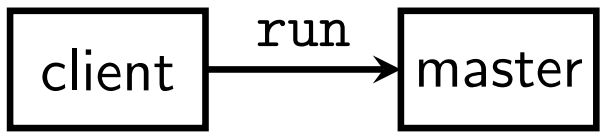
Actors

Execution Model

Actors

1. Client

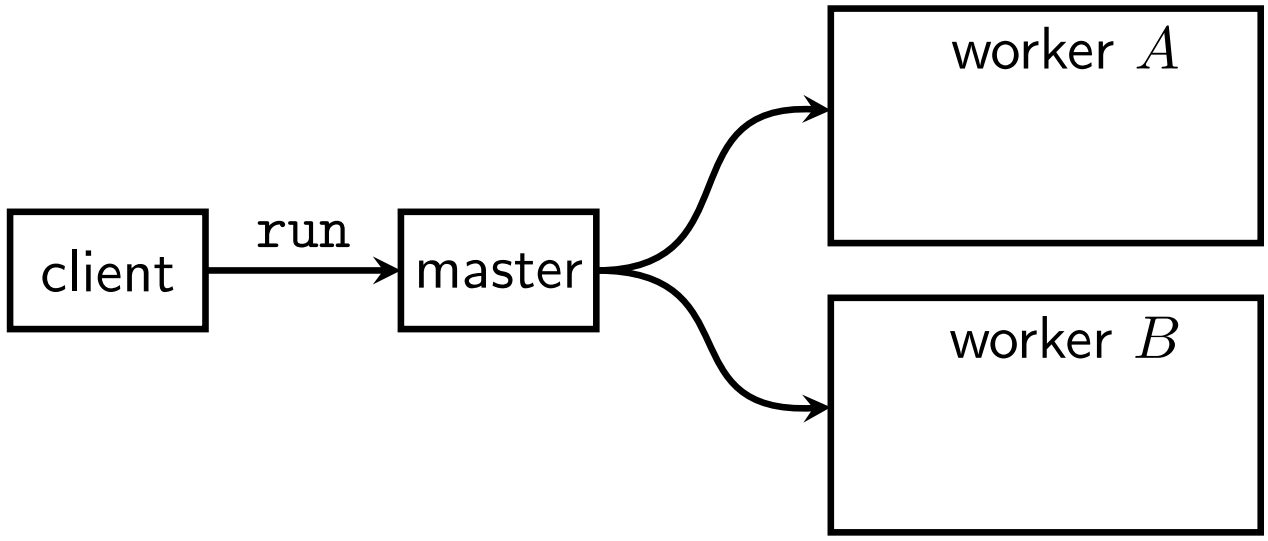
Execution Model



Actors

1. Client
2. Master

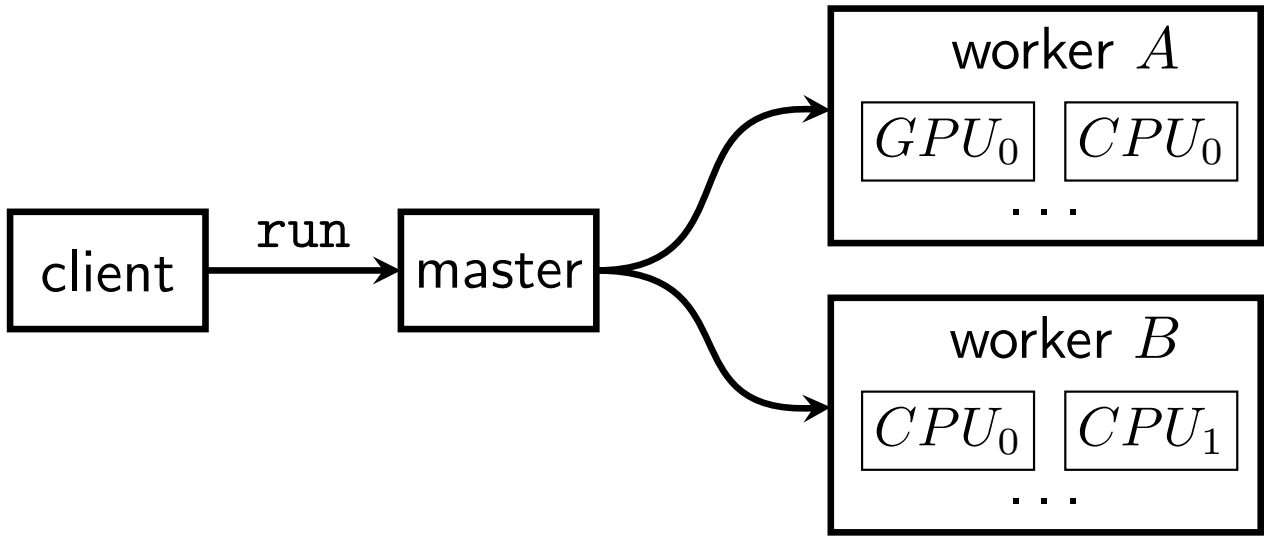
Execution Model



Actors

1. Client
2. Master
3. Workers

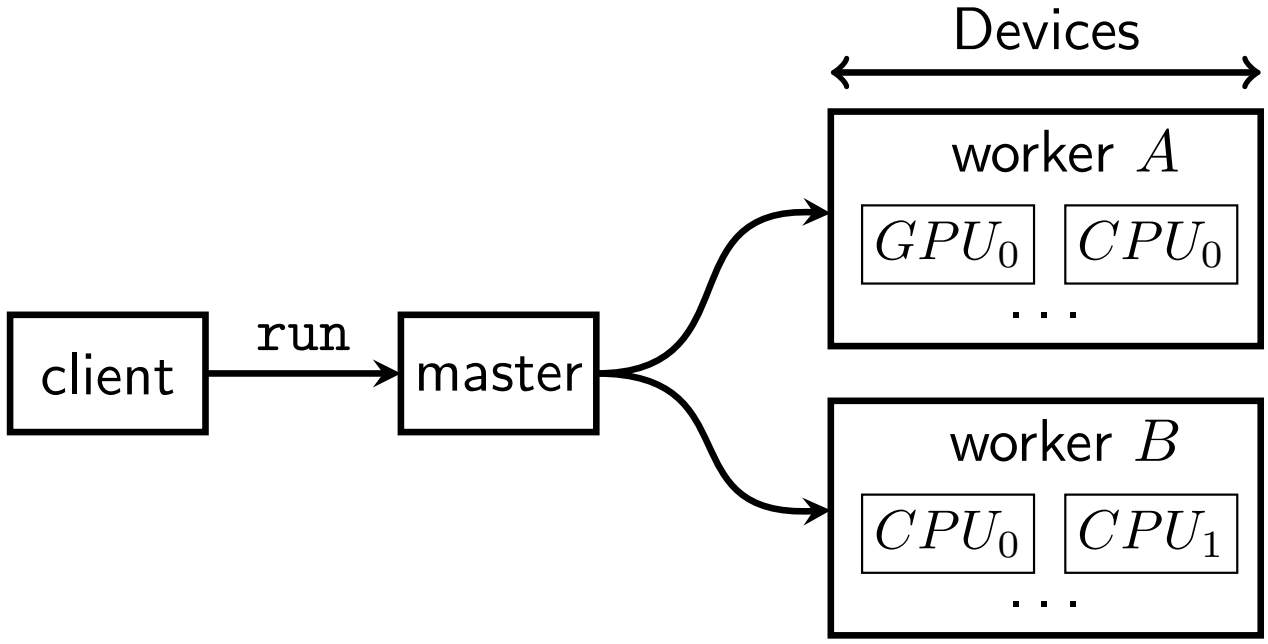
Execution Model



Actors

1. Client
2. Master
3. Workers
4. Devices

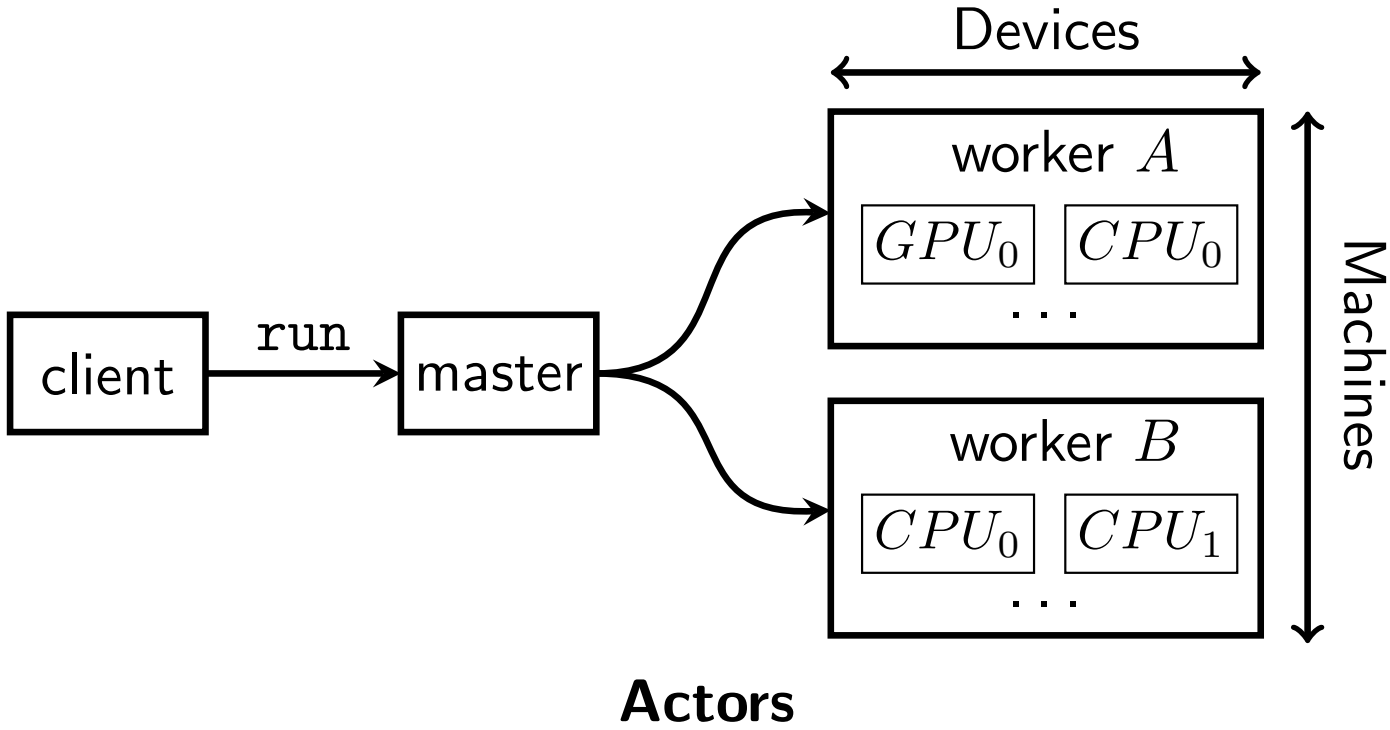
Execution Model



Actors

1. Client
2. Master
3. Workers
4. Devices

Execution Model



1. Client
2. Master
3. Workers
4. Devices

Visualization Tools

Visualization Tools

- Deep Neural Networks have the tendency of being . . . deep

Visualization Tools

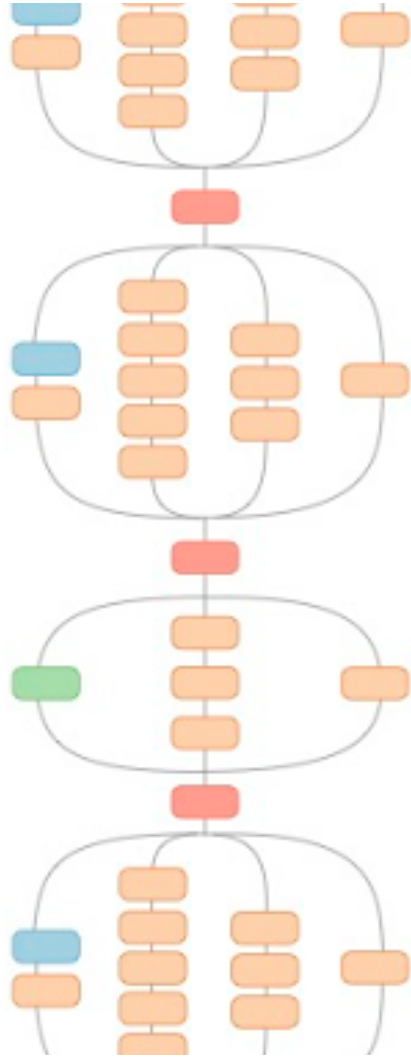
- Deep Neural Networks have the tendency of being . . . deep
- Easy to drown in the complexity of an architecture

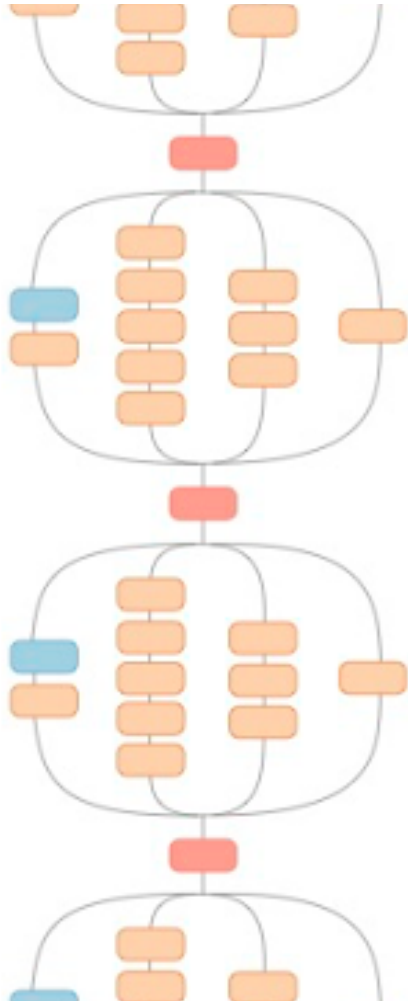
Visualization Tools

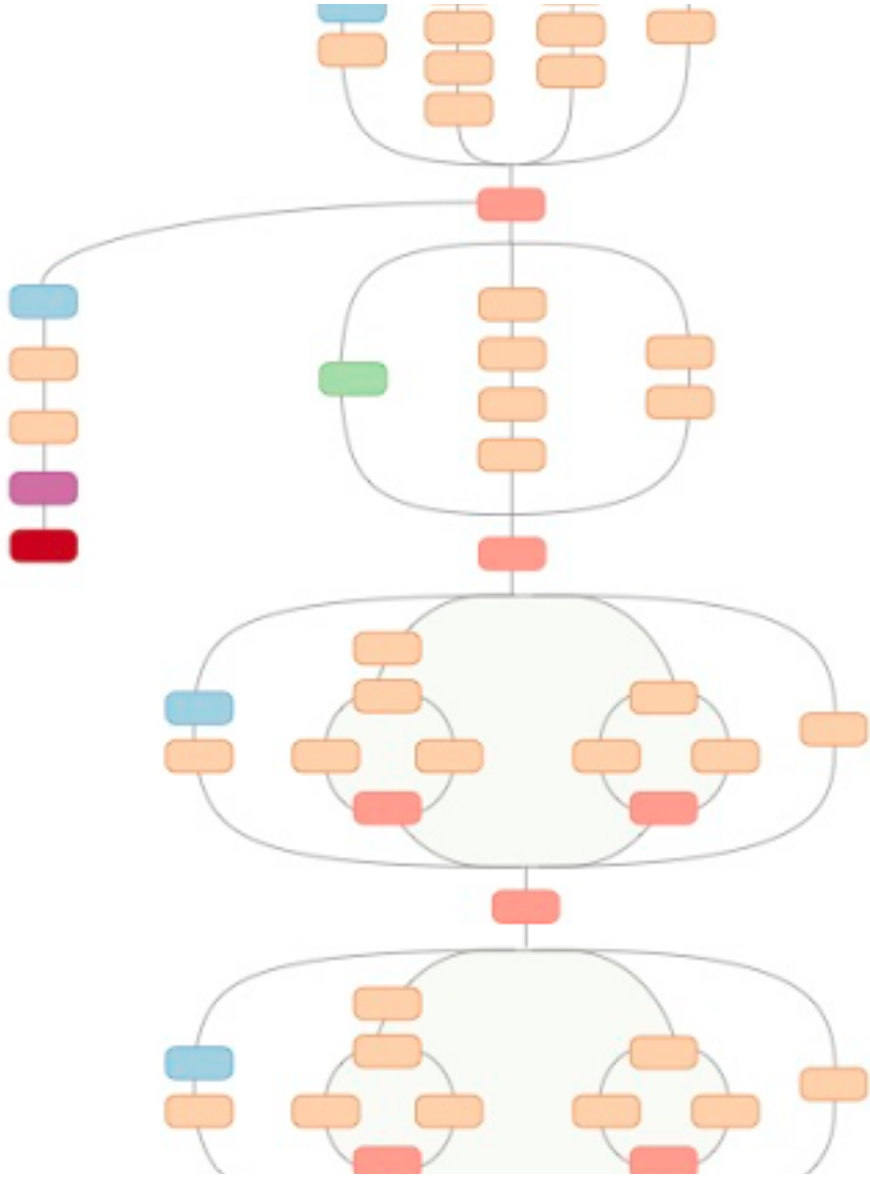
- Deep Neural Networks have the tendency of being . . . deep
- Easy to drown in the complexity of an architecture
- $>$ 36,000 nodes for Google's *Inception* model

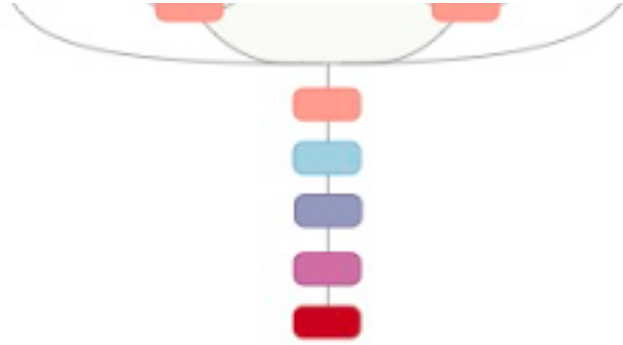
Visualization Tools

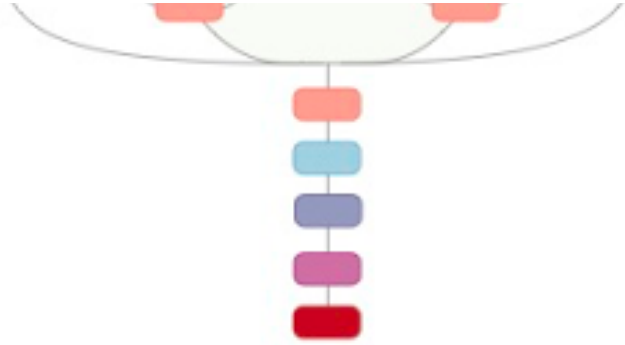
- Deep Neural Networks have the tendency of being . . . deep
- Easy to drown in the complexity of an architecture
- $>$ 36,000 nodes for Google's *Inception* model











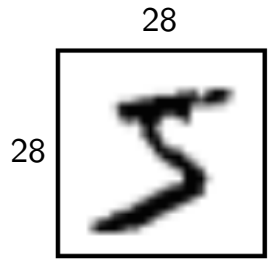
TensorBoard to the Rescue

Walkthrough

Walkthrough

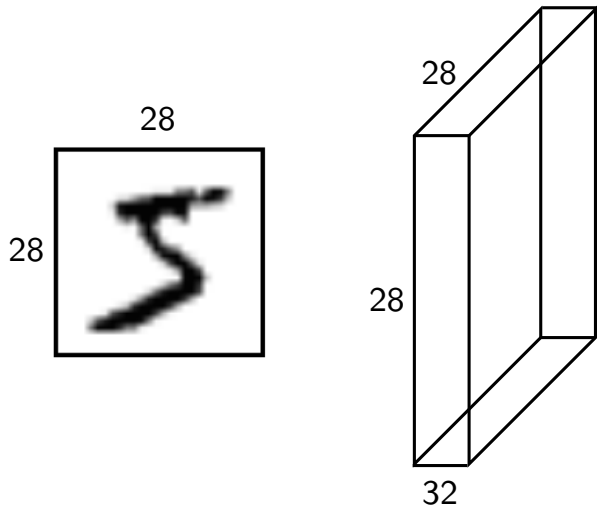
```
INPUT -> [CONV -> POOL]{2} -> FC -> OUTPUT
```

Walkthrough



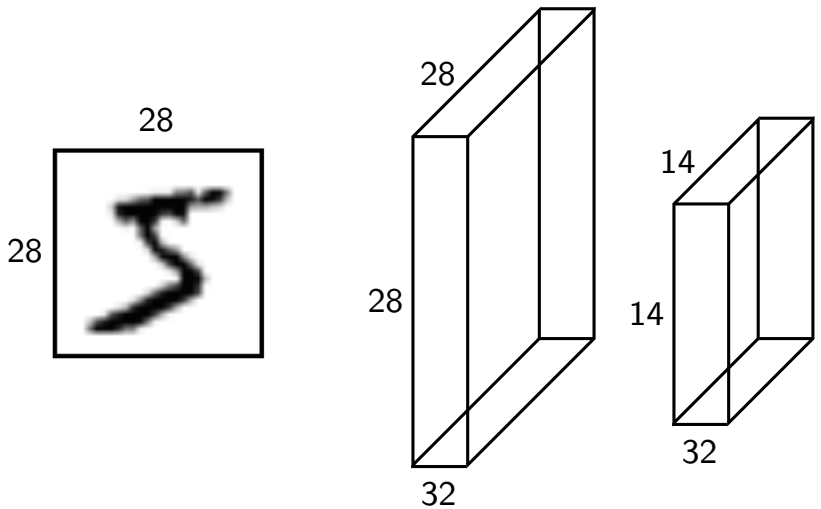

```
INPUT -> [CONV -> POOL]{2} -> FC -> OUTPUT
```

Walkthrough



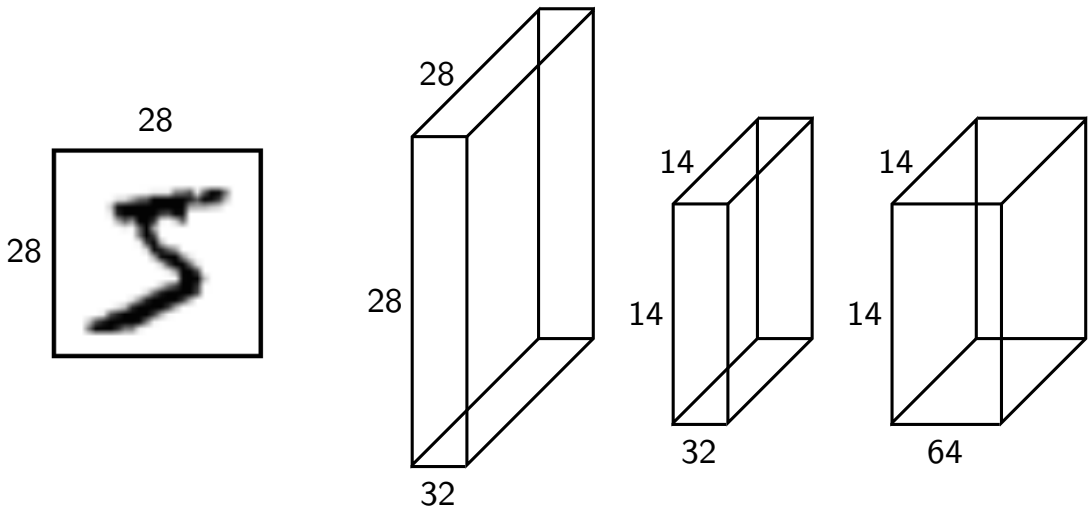

```
INPUT -> [CONV -> POOL]{2} -> FC -> OUTPUT
```

Walkthrough



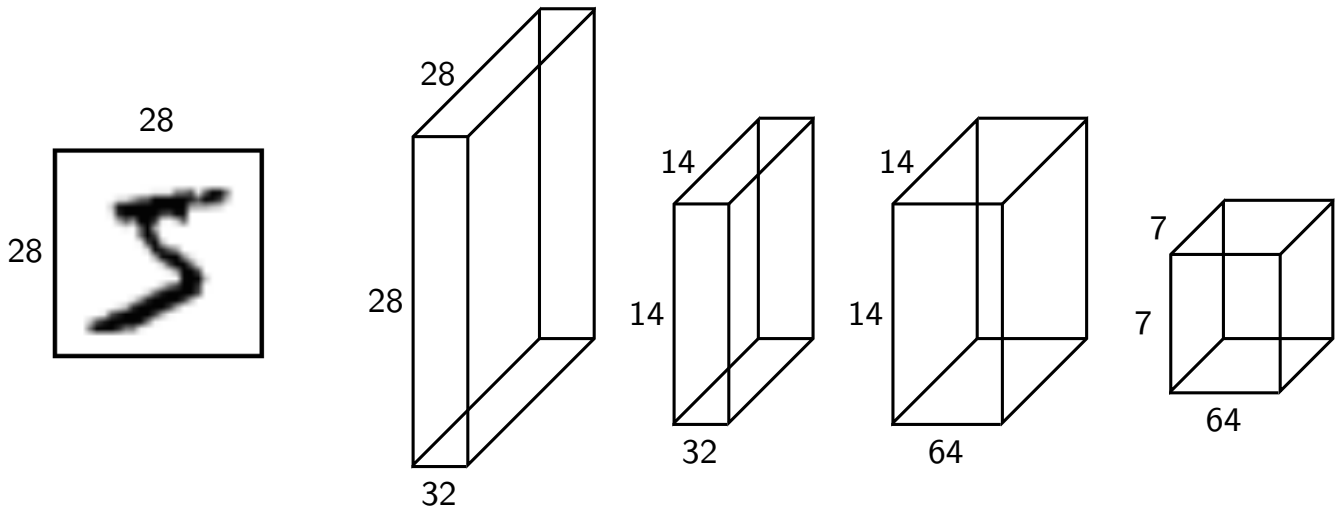

```
INPUT -> [CONV -> POOL]{2} -> FC -> OUTPUT
```

Walkthrough



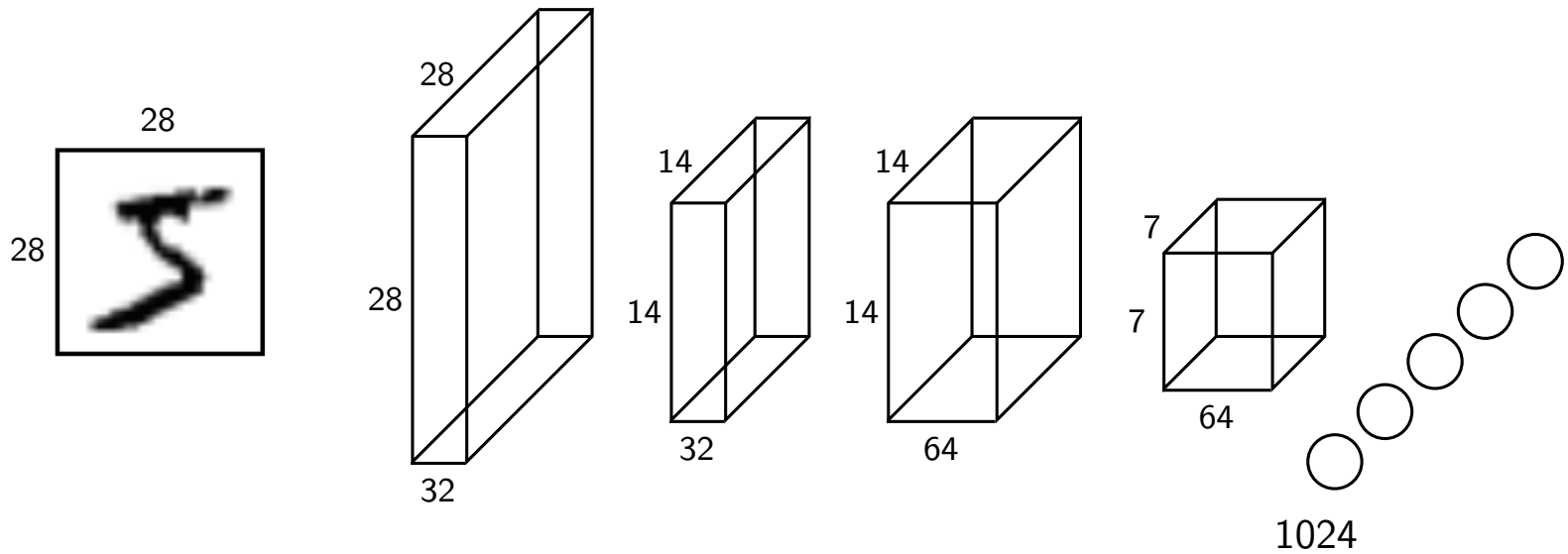

```
INPUT -> [CONV -> POOL]{2} -> FC -> OUTPUT
```

Walkthrough



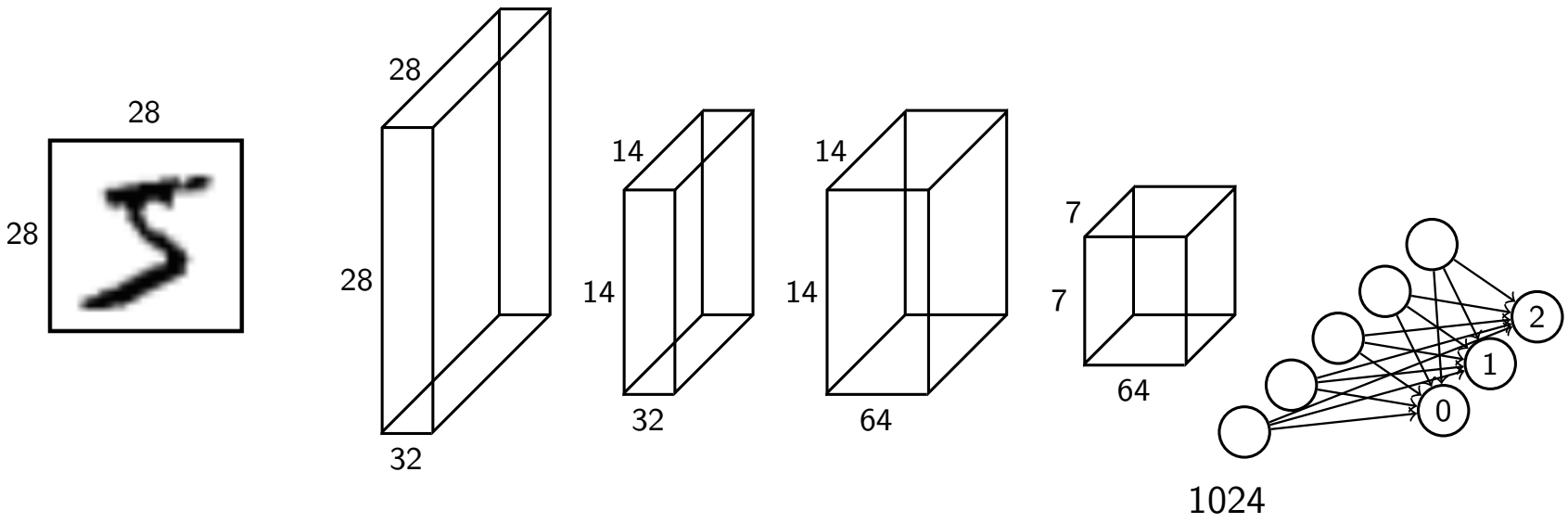

```
INPUT -> [CONV -> POOL]{2} -> FC -> OUTPUT
```

Walkthrough



`INPUT -> [CONV -> POOL]{2} -> FC -> OUTPUT`

Walkthrough



`INPUT -> [CONV -> POOL]{2} -> FC -> OUTPUT`

How do I continue?

Resources

Resources

- MOOCs
  - Machine Learning by Andrew Ng @ Coursera
  - Deep Learning by Google @ Udacity
  - Machine Learning Nanodegree @ Udacity

Resources

- MOOCs
  - Machine Learning by Andrew Ng @ Coursera
  - Deep Learning by Google @ Udacity
  - Machine Learning Nanodegree @ Udacity
- Websites
  - http://colah.github.io
  - http://cs231n.github.io
  - http://karpathy.github.io
  - http://www.deeplearningbook.org
  - https://www.kaggle.com
  - https://www.tensorflow.org

Stay in Touch!

- peter@goldsborough.me
- linkedin.com/in/petergoldsborough
- github.com/goldsborough
- @peterawks

Stay in Touch!

- peter@goldsborough.me
- linkedin.com/in/petergoldsborough
- github.com/goldsborough
- @peterawks

github.com/peter-can-talk/cambridge-2016

Q & A