

Image Recognition with Apache Spark and BigDL

Alex Kalinin | VP, AI/Machine Learning | Sizmek

<https://www.linkedin.com/in/alexkalinin/>

<http://alexkalinin.com>

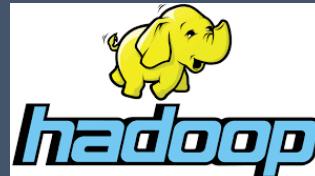


AI-powered marketing and optimization

1,200,000 / s

100,000,000,000 requests per day

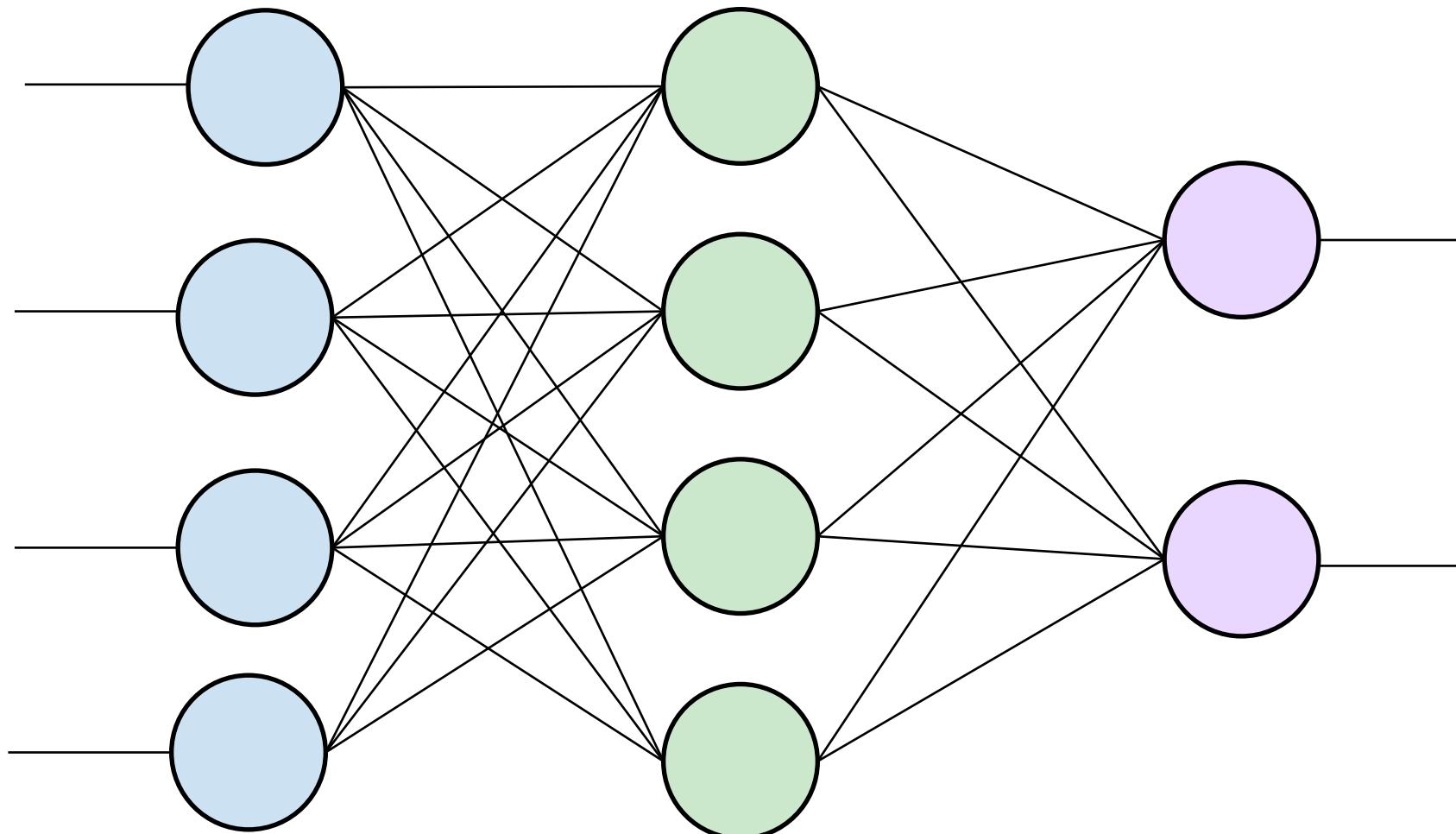
100 PBs of training data



Contact

- GitHub: <https://github.com/alex-kalinin/letnet-bigdl/>
- LinkedIn: <https://www.linkedin.com/in/alexkalinin/>

Feed-forward network



- <http://playground.tensorflow.org/>

$$y = \sum (w_i * x_i)$$

43

?

37

?

45

?

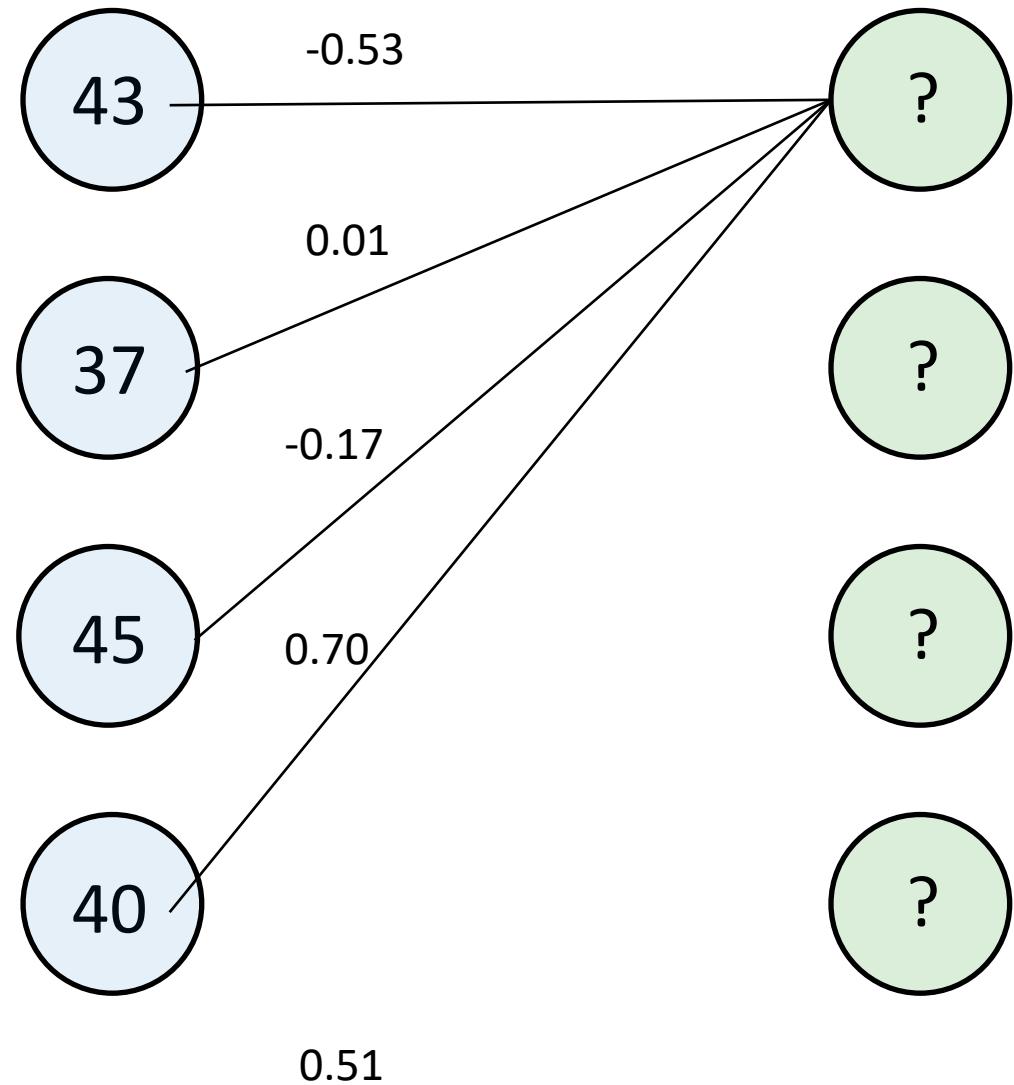
40

?

?

?

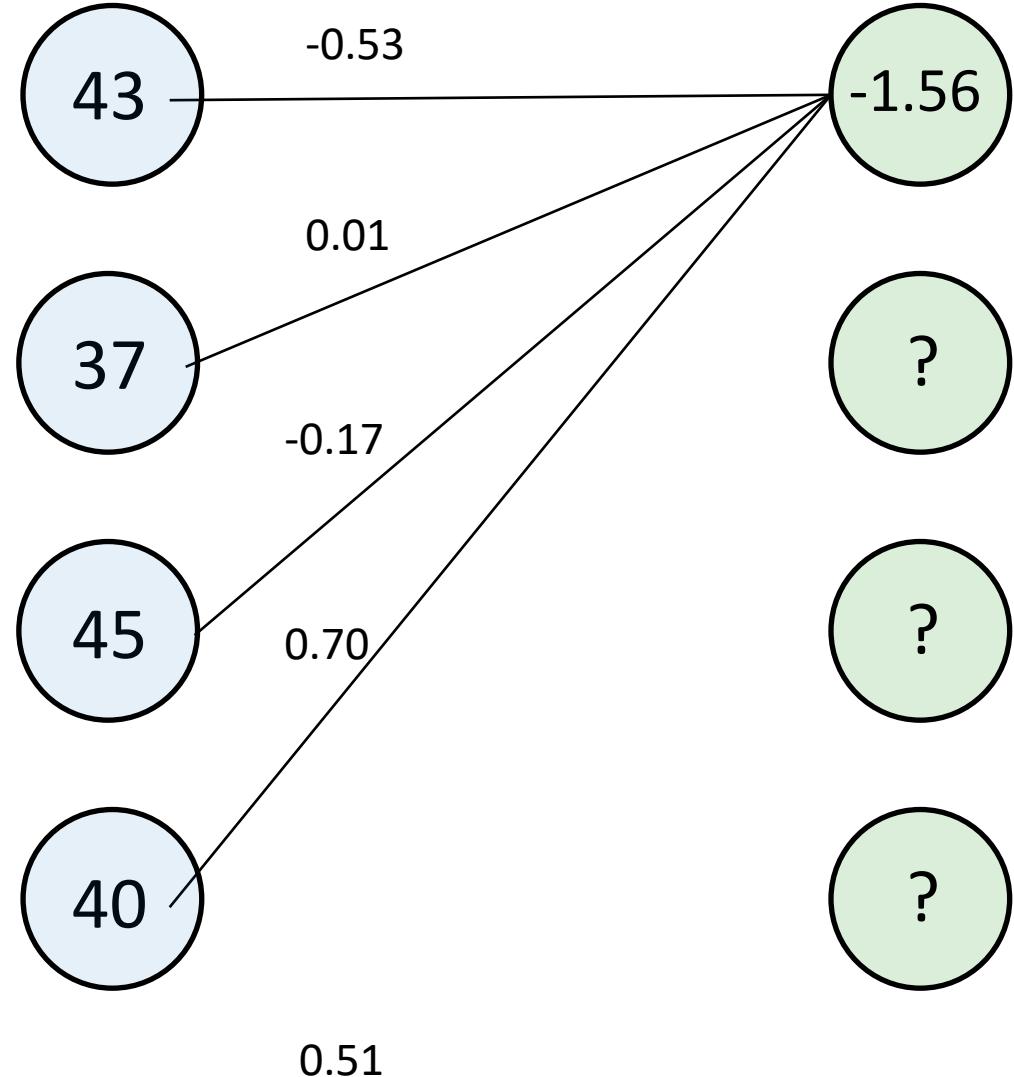
$$y = \sum (w_i * x_i)$$



?

?

$$y = \sum (w_i * x_i)$$

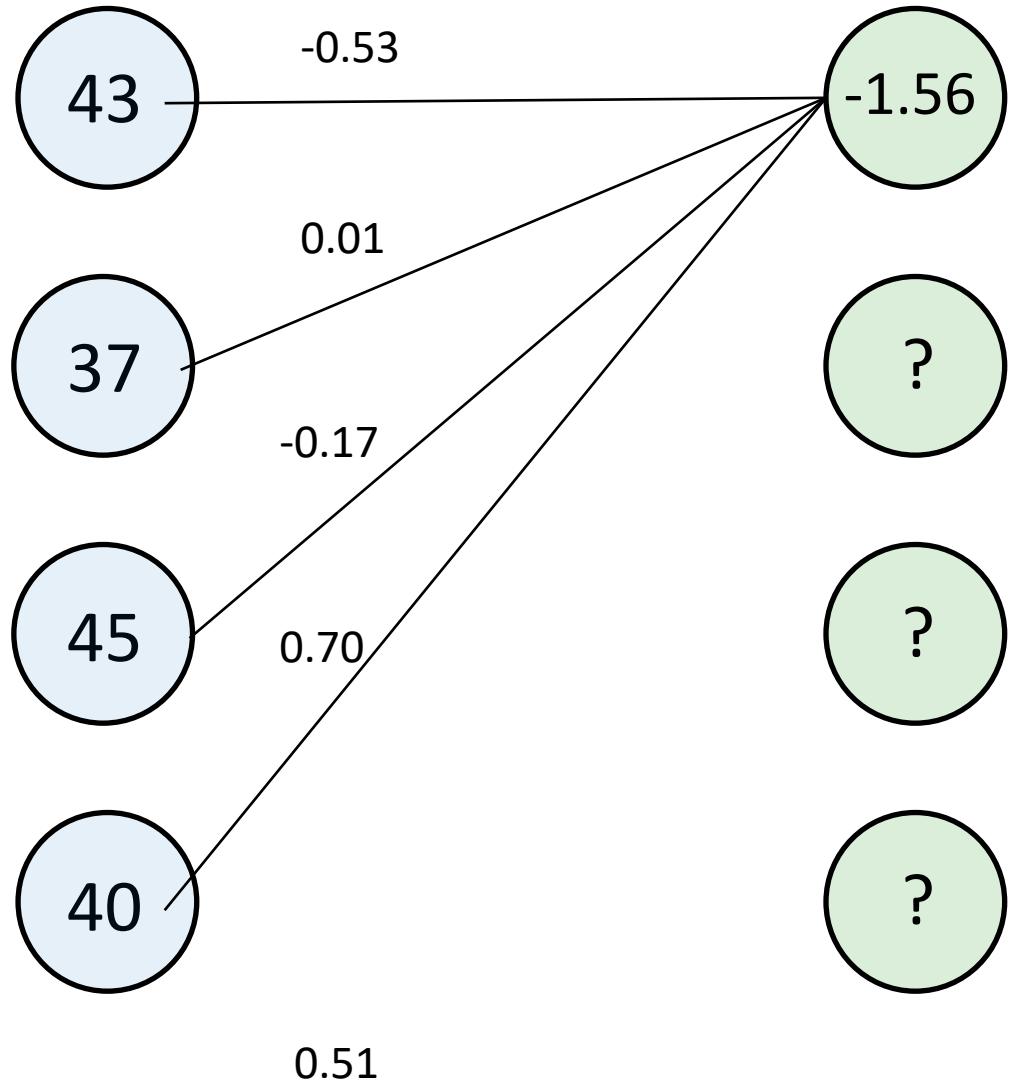


?

?

?

$$y = \text{ReLU}(\sum (w_i * x_i))$$

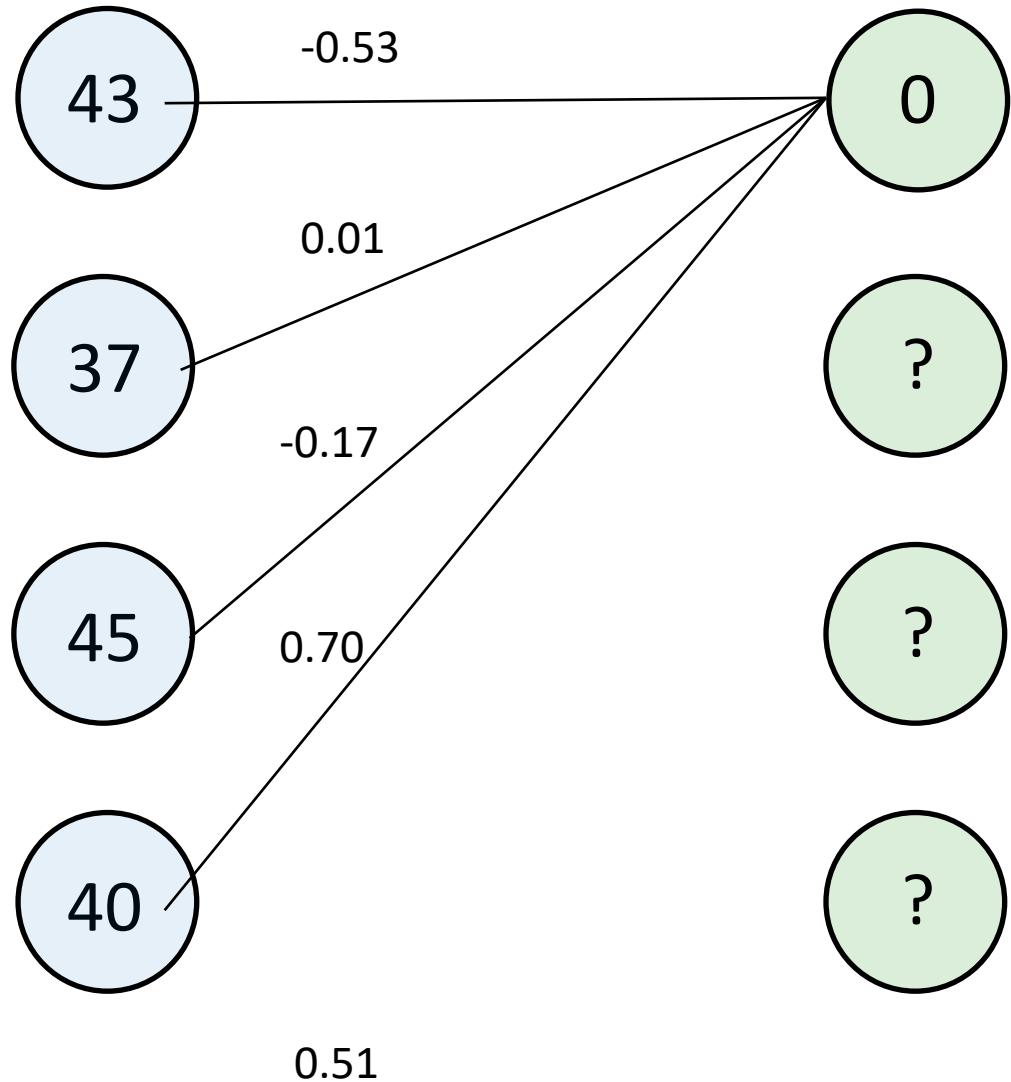


?

?

?

$$y = \text{ReLU}(\sum (w_i * x_i))$$

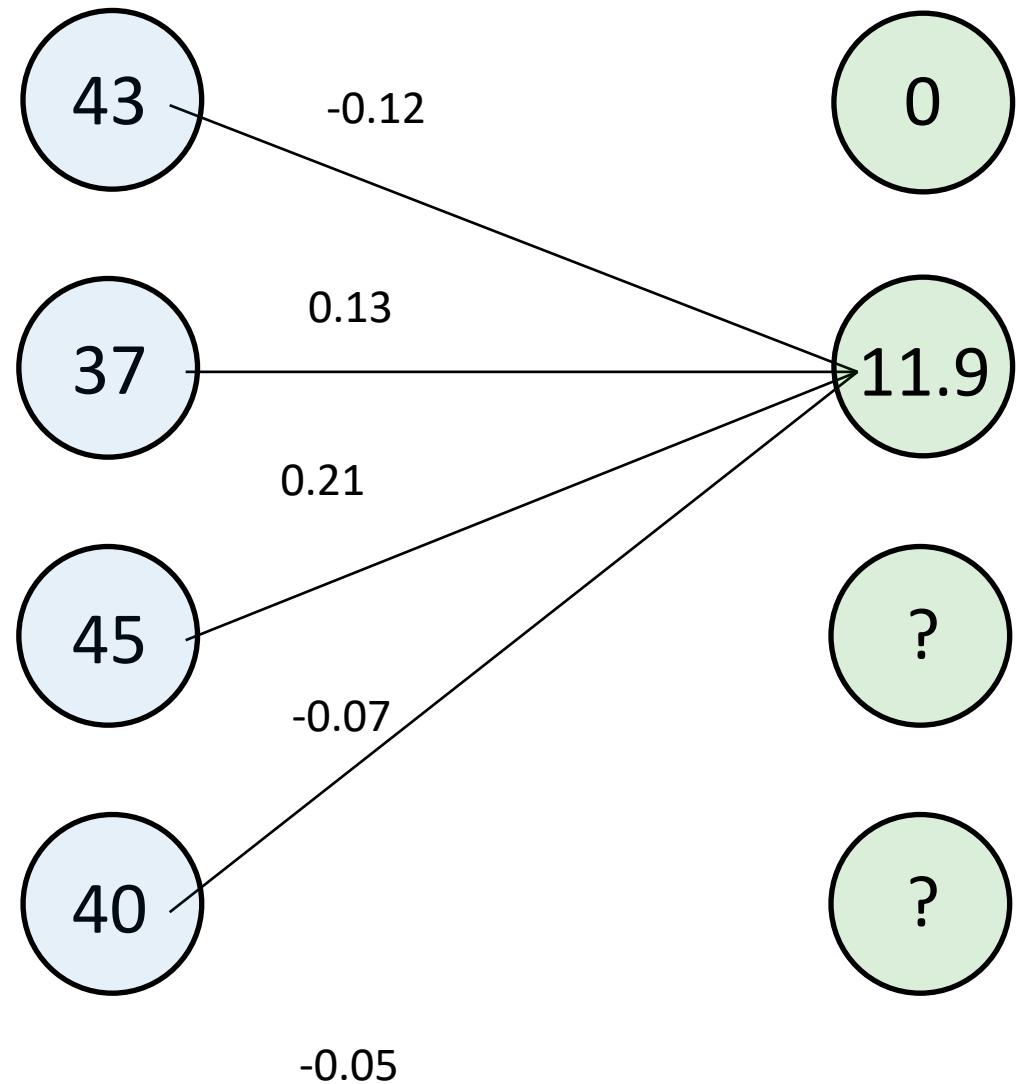


?

?

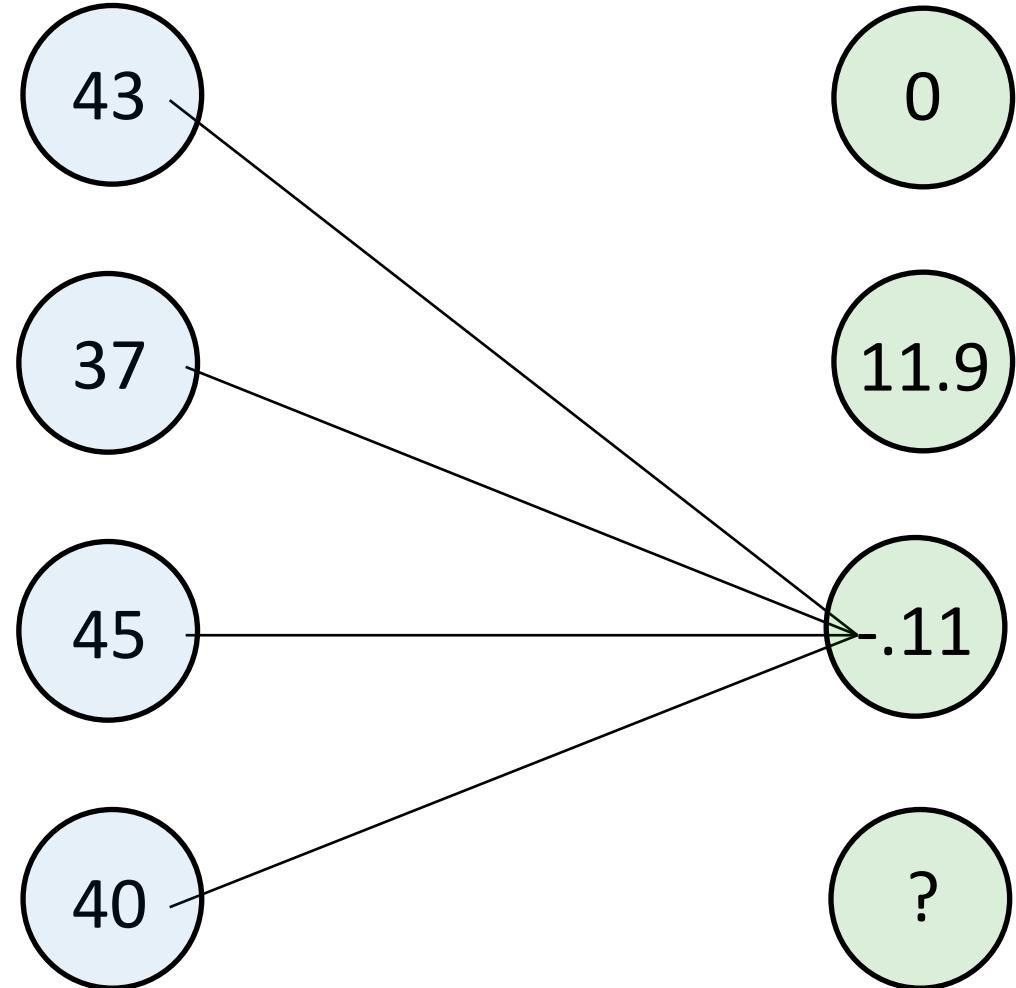
?

$$y = \text{ReLU}(\sum (w_i * x_i))$$



?

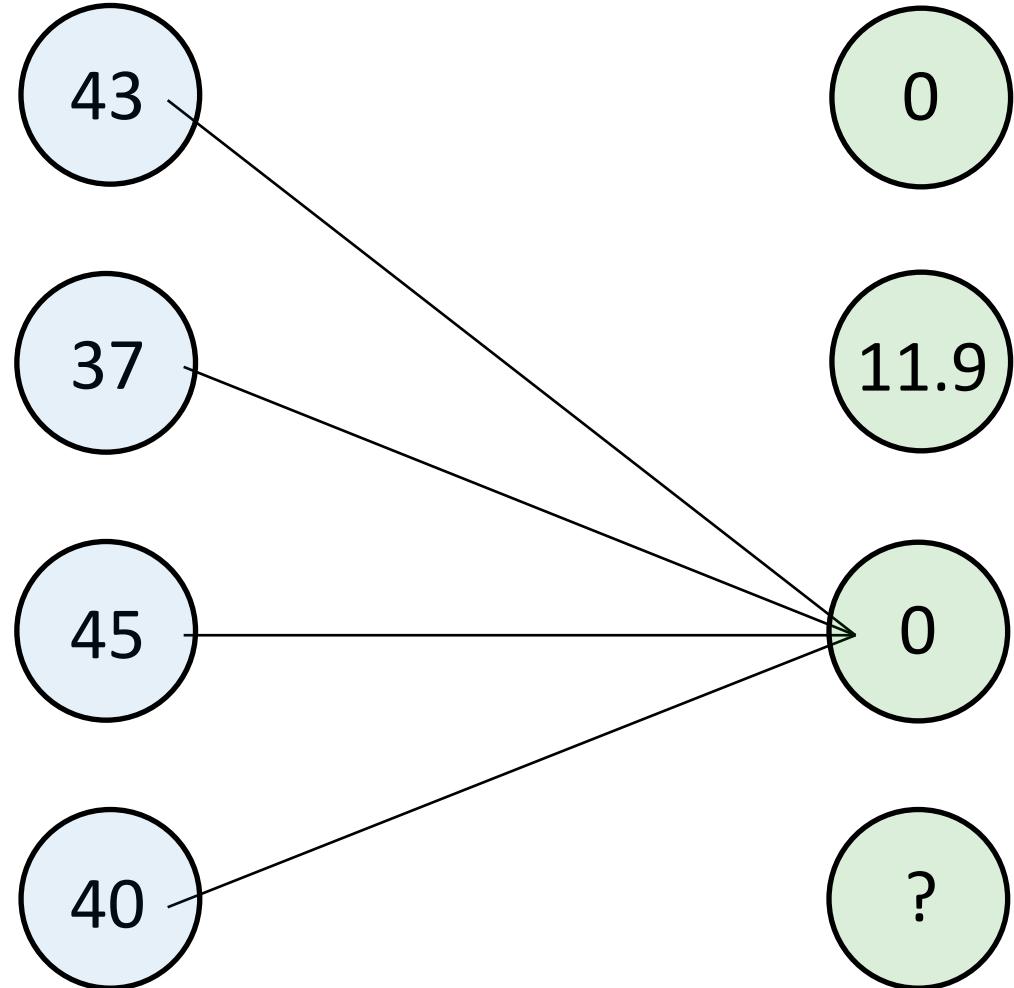
?



$$y = \text{ReLU}(\sum (w_i * x_i))$$

?

?

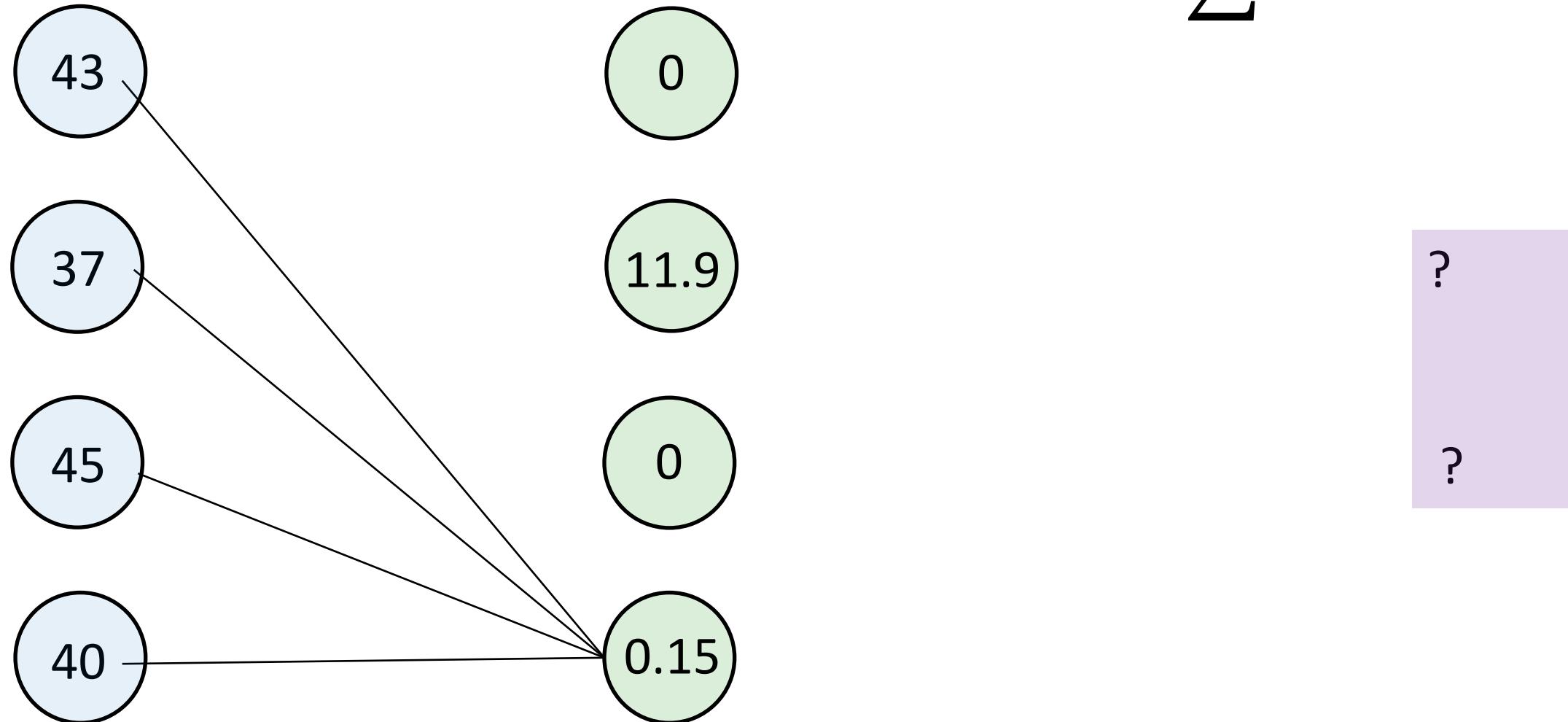


$$y = \text{ReLU}(\sum (w_i * x_i))$$

?

?

$$y = \text{ReLU}(\sum (w_i * x_i))$$



$$y = \text{ReLU}(\sum (w_i * x_i))$$

43

37

45

40

0

11.9

0

0.15

-0.67

?

$$y = \text{ReLU}(\sum (w_i * x_i))$$

43

37

45

40

0

11.9

0

0.15

0
?

$$y = \text{ReLU}(\sum (w_i * x_i))$$

43

37

45

40

0

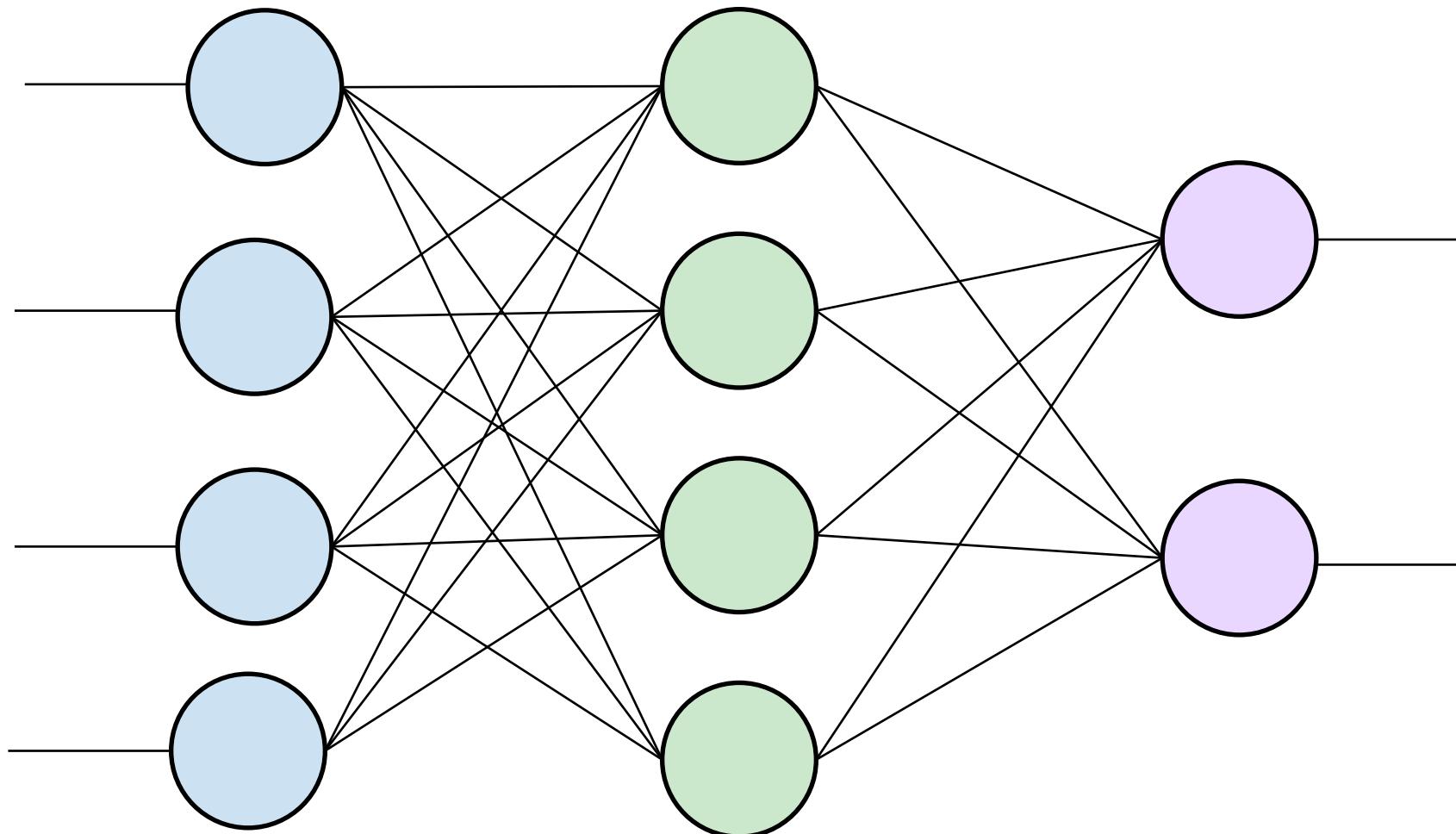
11.9

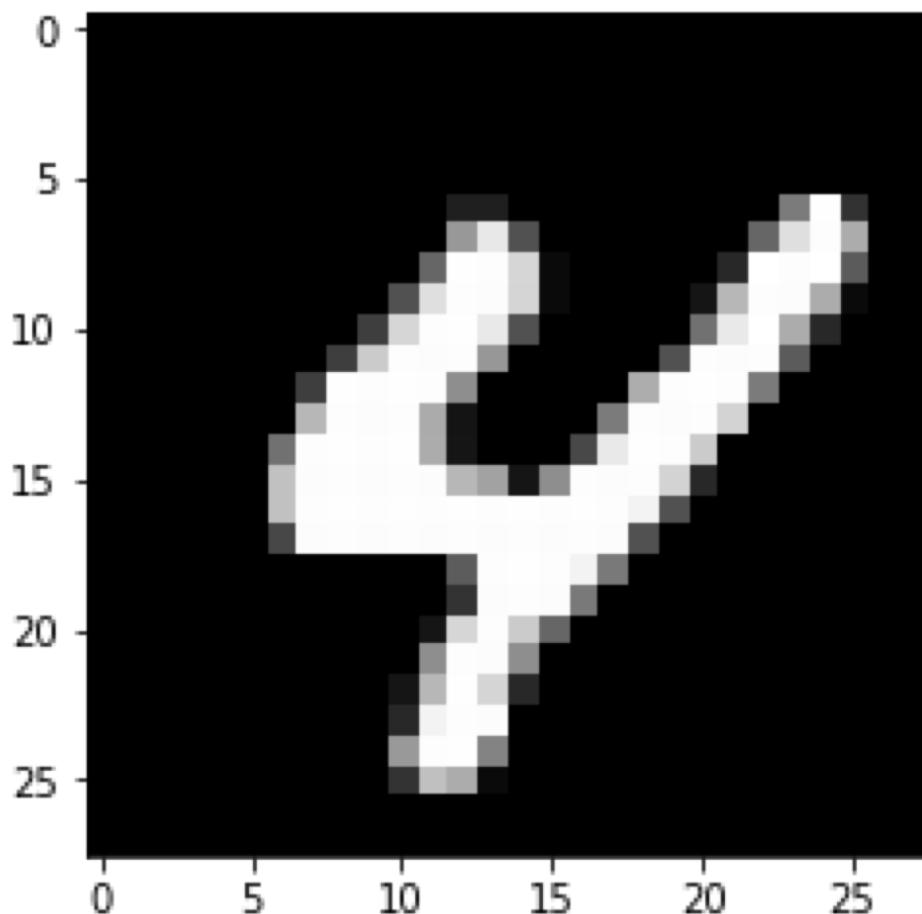
0

0.15

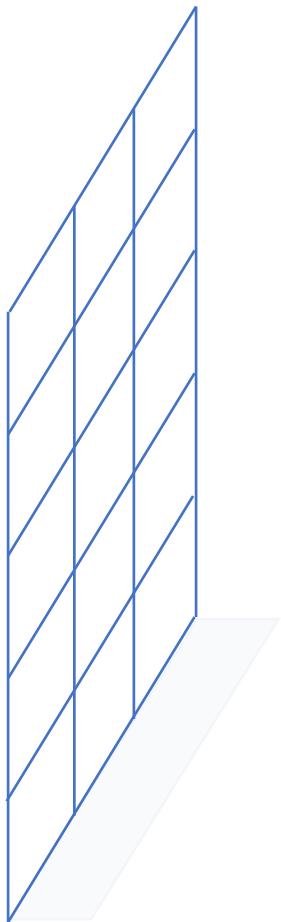
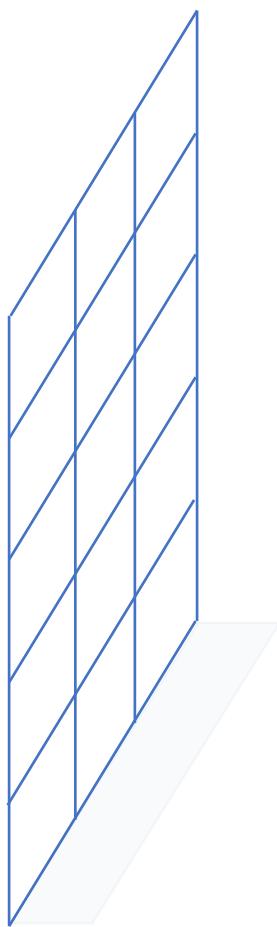
0
0.52

Feed-forward network

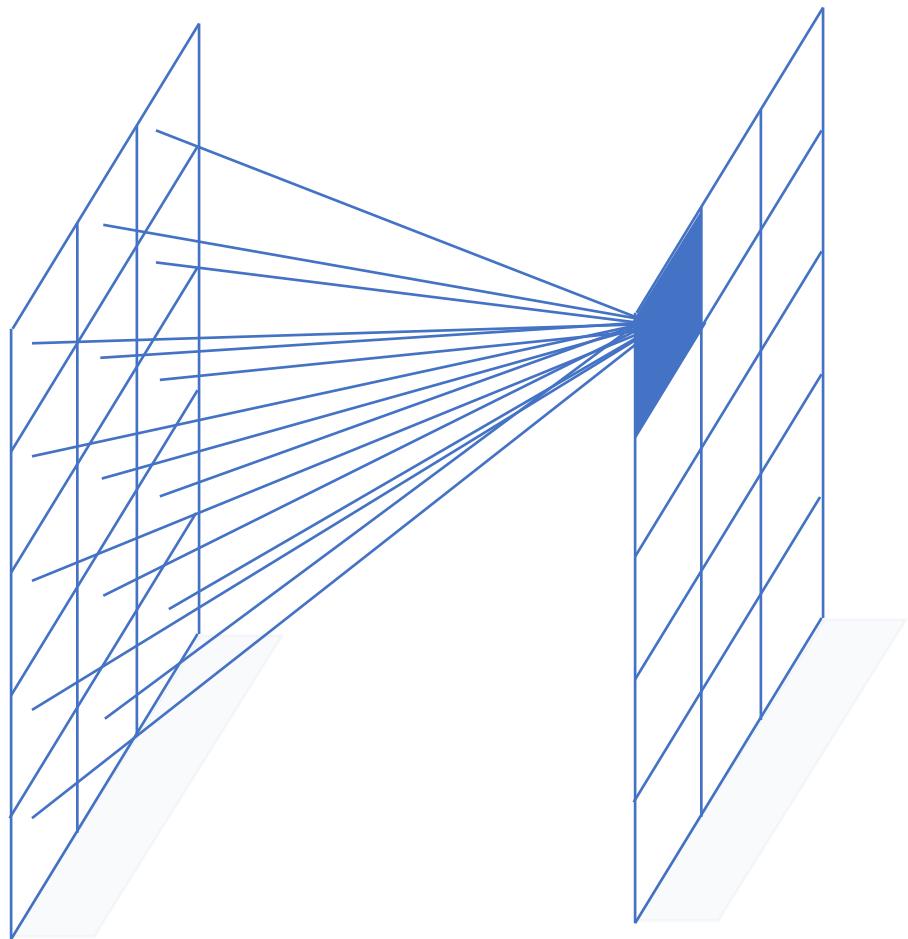




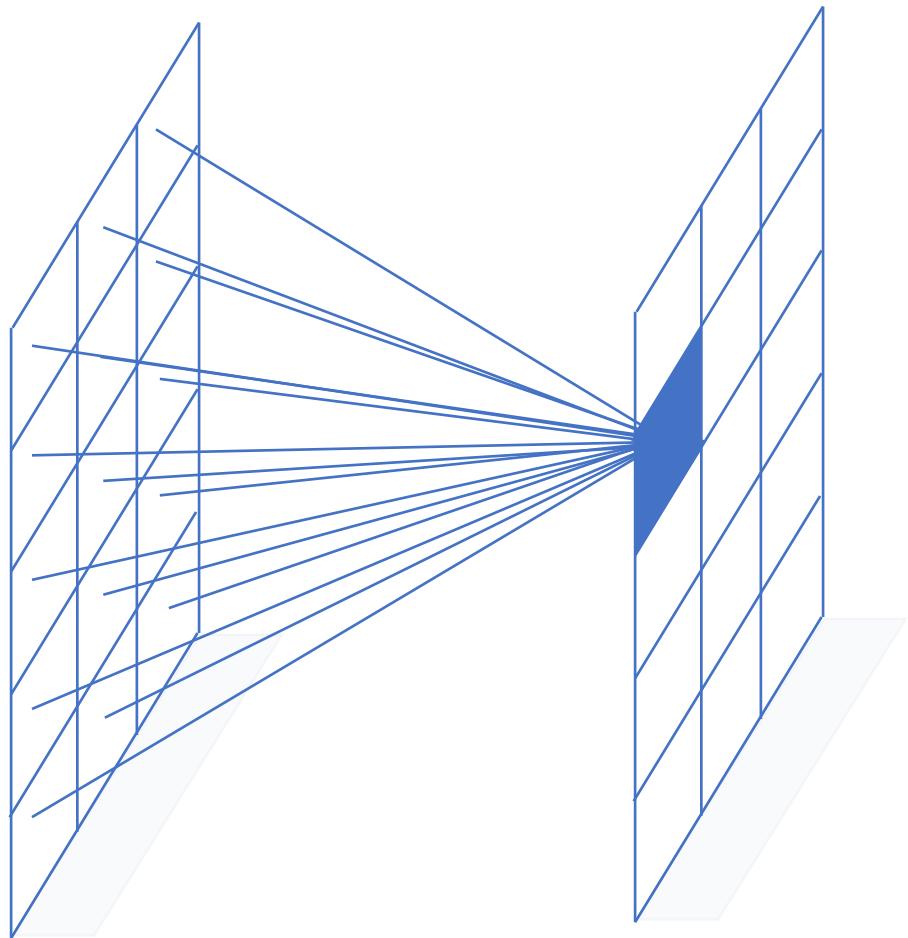
Alex Kalinin
<https://www.linkedin.com/in/alexkalinin/>



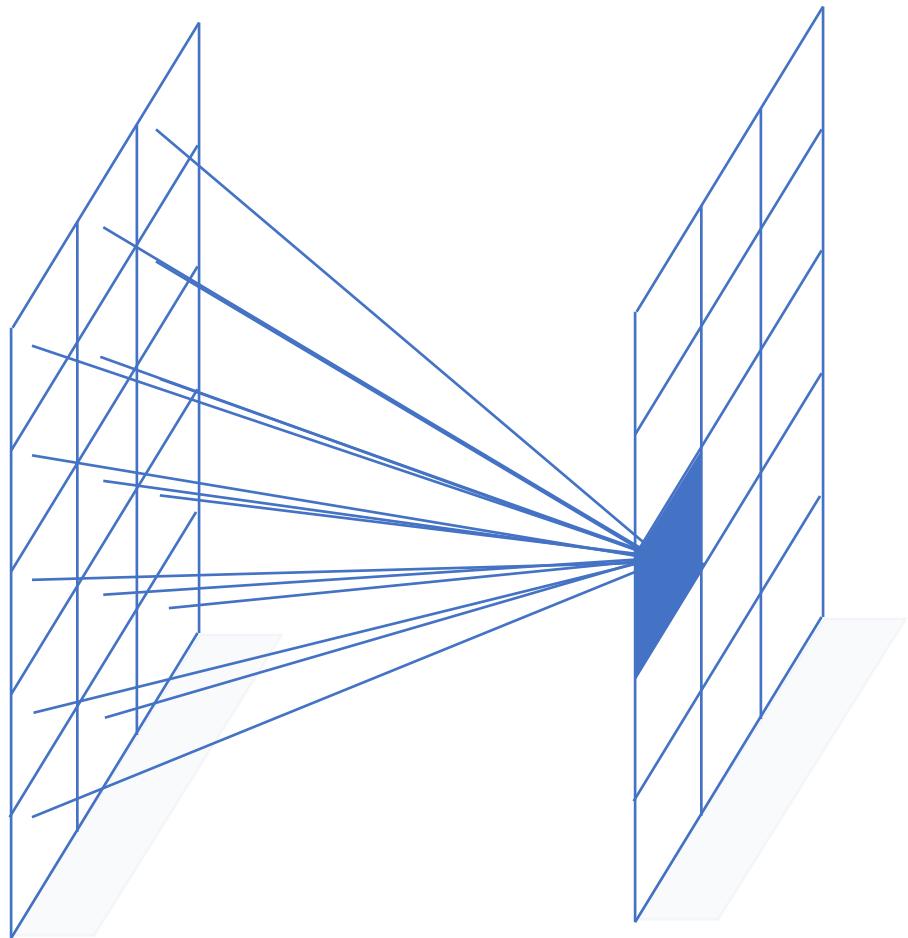
Fully Connected



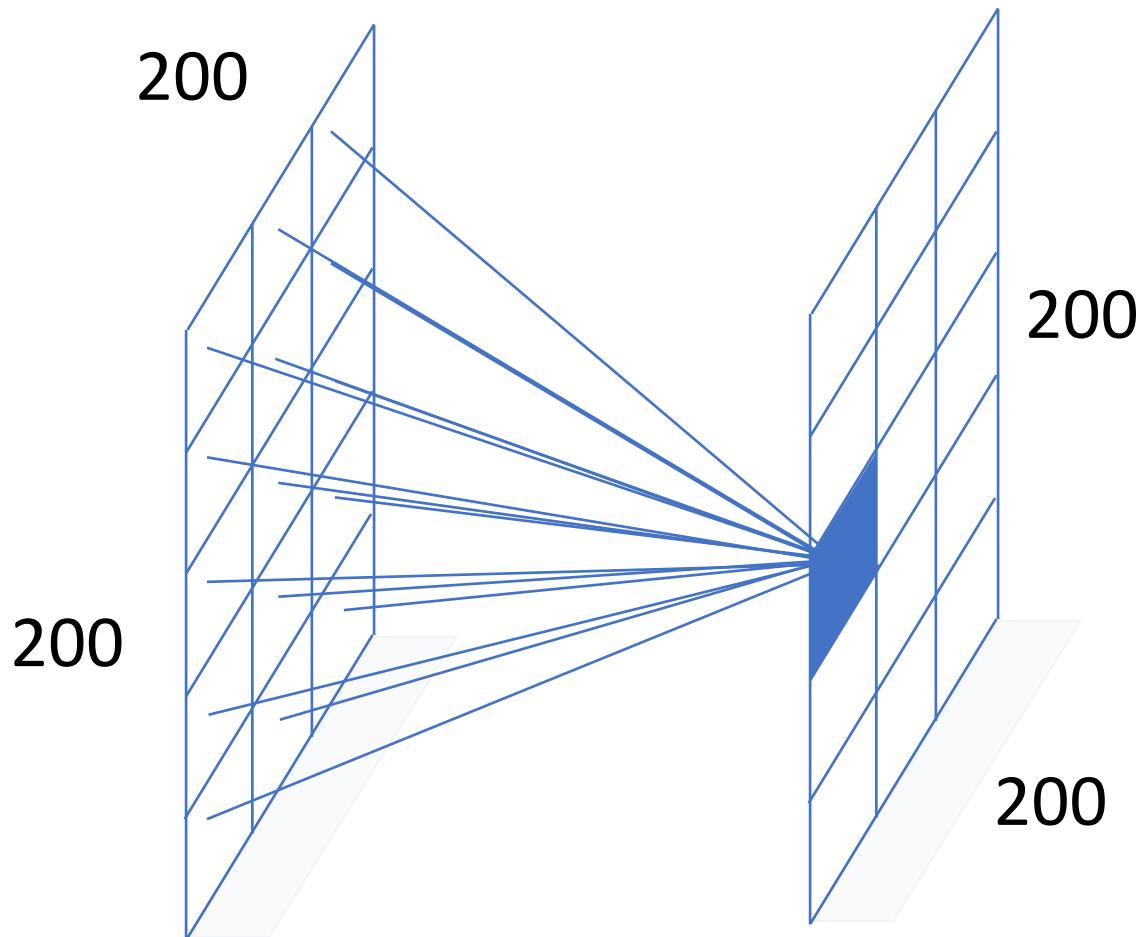
Fully Connected



Fully Connected



Fully Connected



Input Size:

40,000

Connections:

1,600,000,000

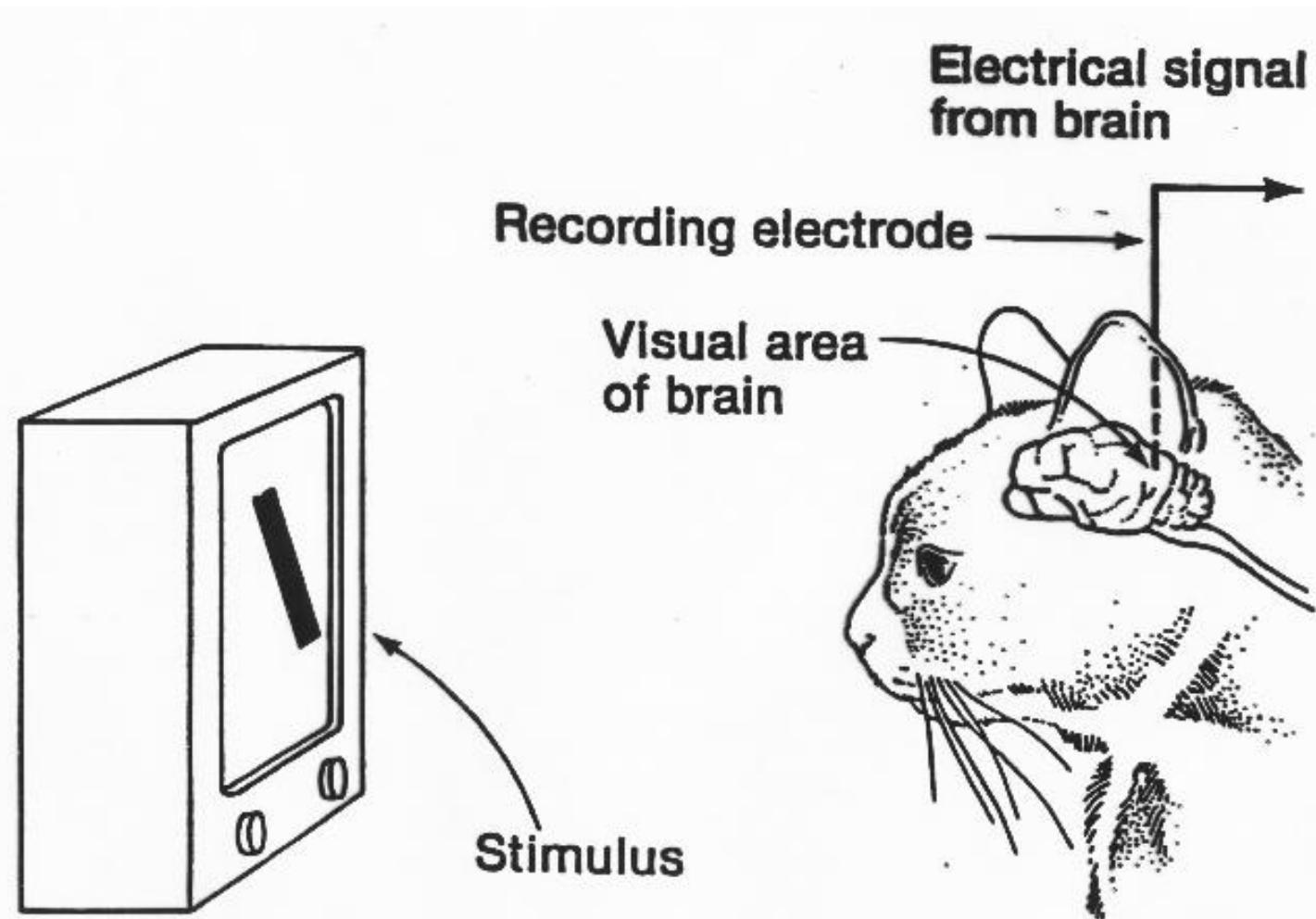
10 layers:

16 billion

Convolutional Neural Networks

- LeNet-5 network developed in 1998 by Yann LeCun
- Torsten Hubel and David Wiesel

Hubel & Wiesel



Alex Kalinin

<https://www.linkedin.com/in/alexkalinin/>



Alex Kalinin

<https://www.linkedin.com/in/alexkalinin/>



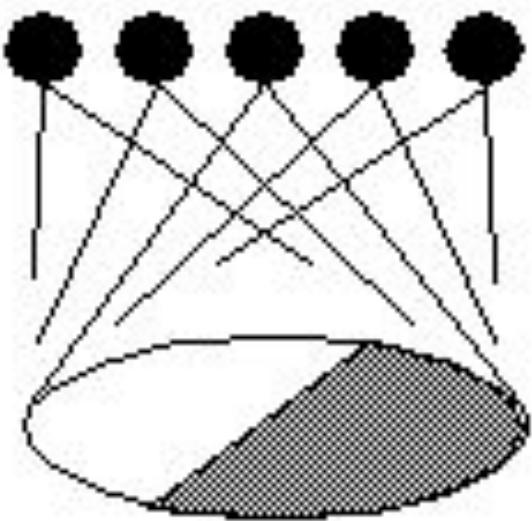
Alex Kalinin

<https://www.linkedin.com/in/alexkalinin/>

Hierarchical Visual Cortex

Hubel & Weisel

topographical mapping

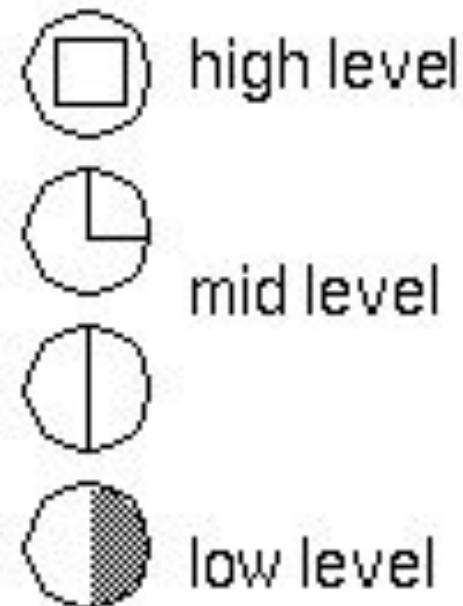
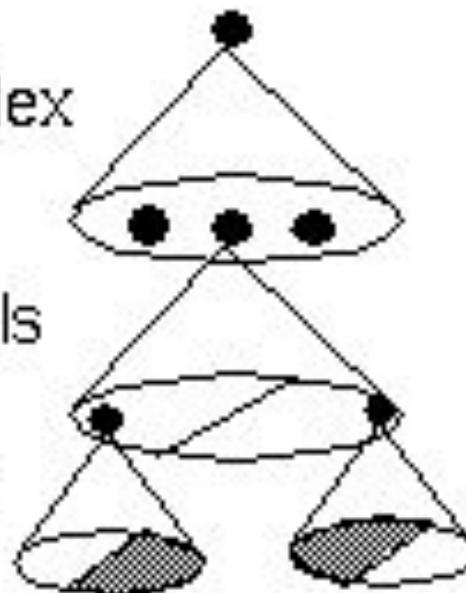


featural hierarchy

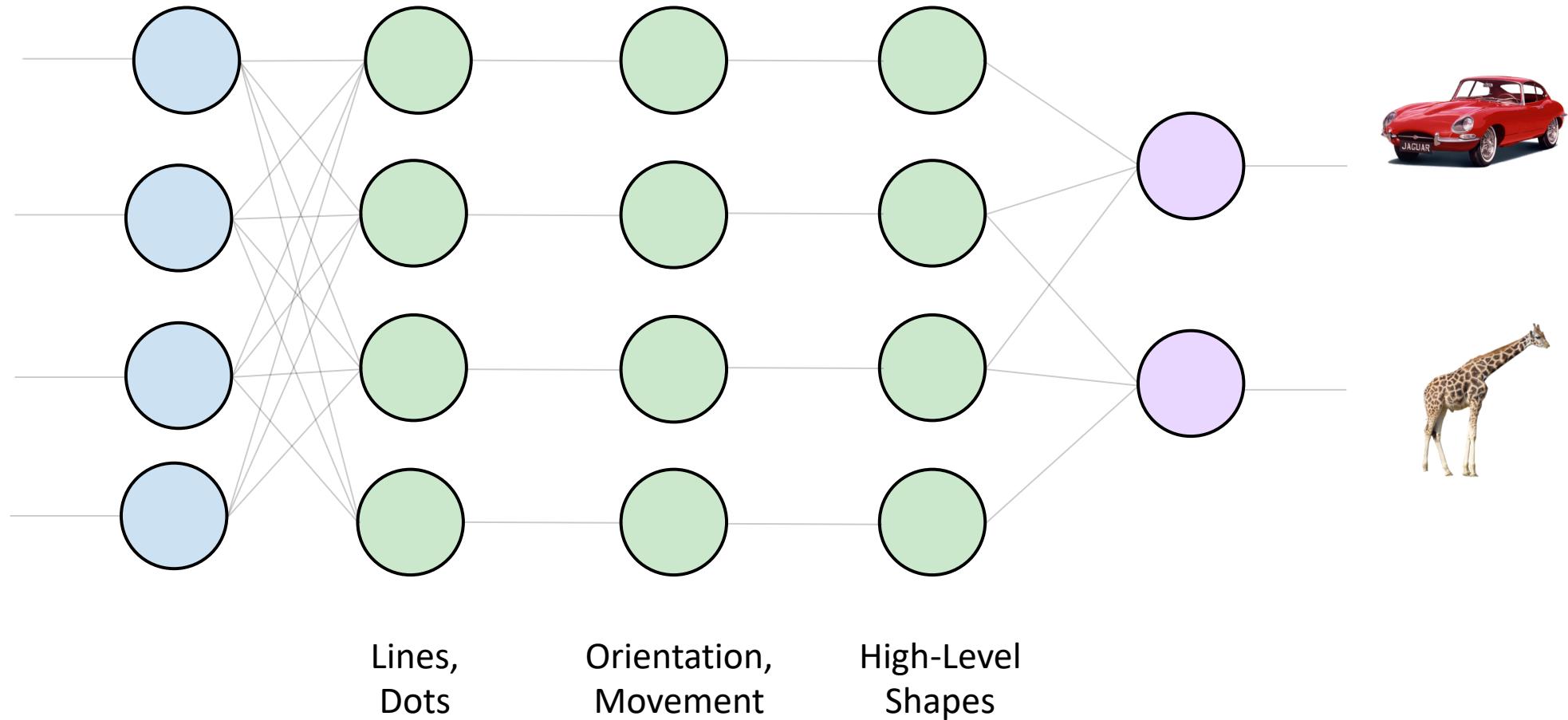
hyper-complex
cells

complex cells

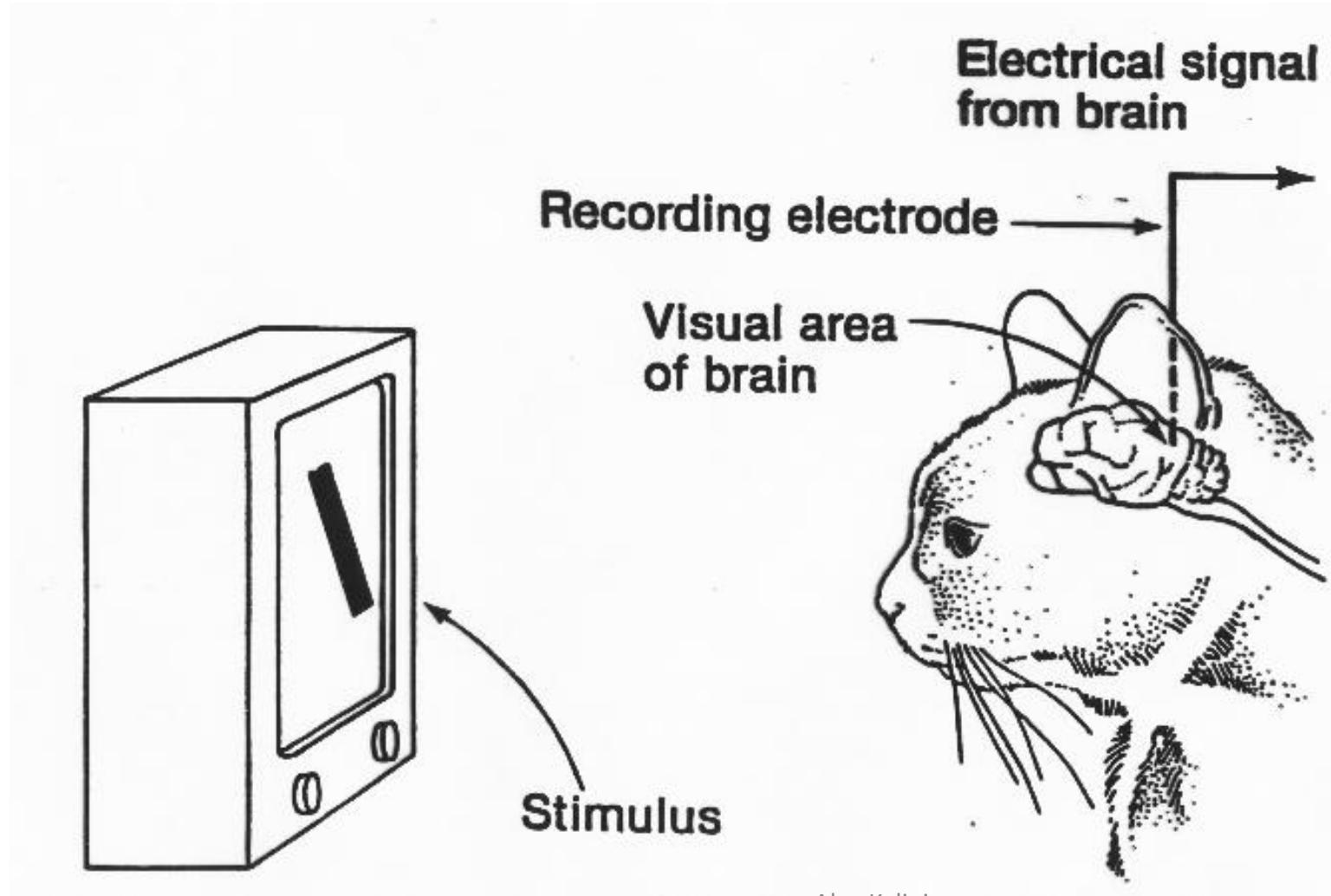
simple cells



Hierarchical Visual Cortex



Local Receptive Fields

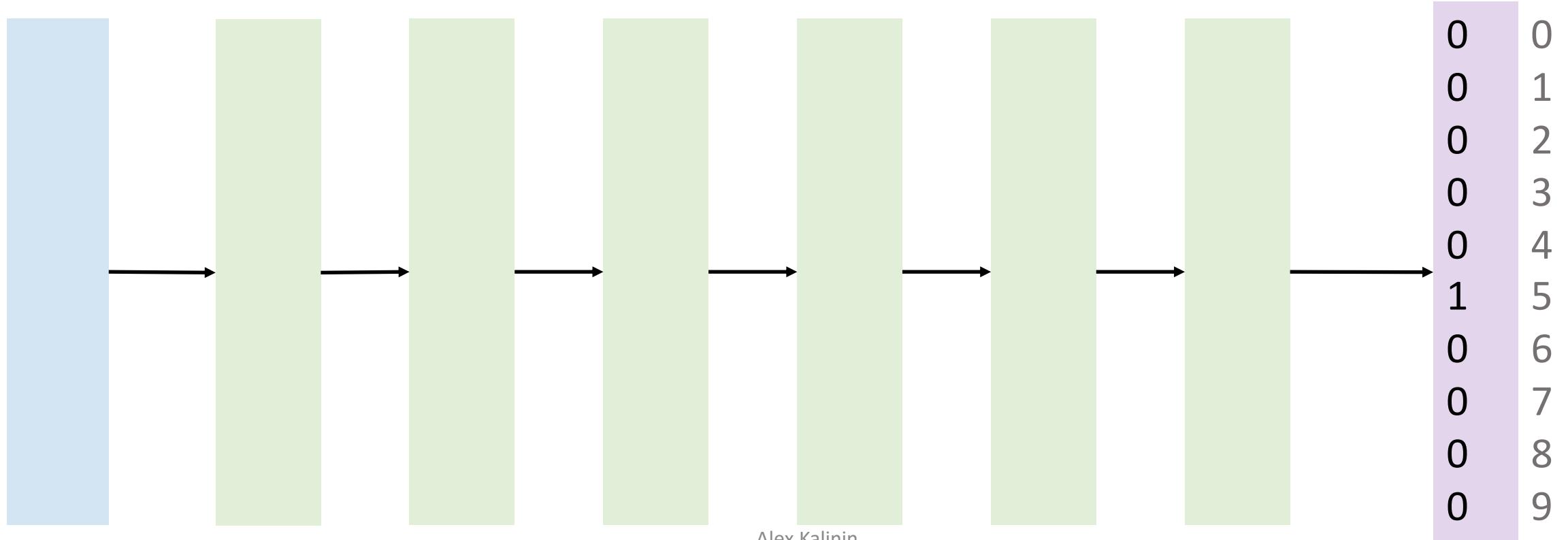


Alex Kalinin

<https://www.linkedin.com/in/alexkalinin/>

Convolutional Network

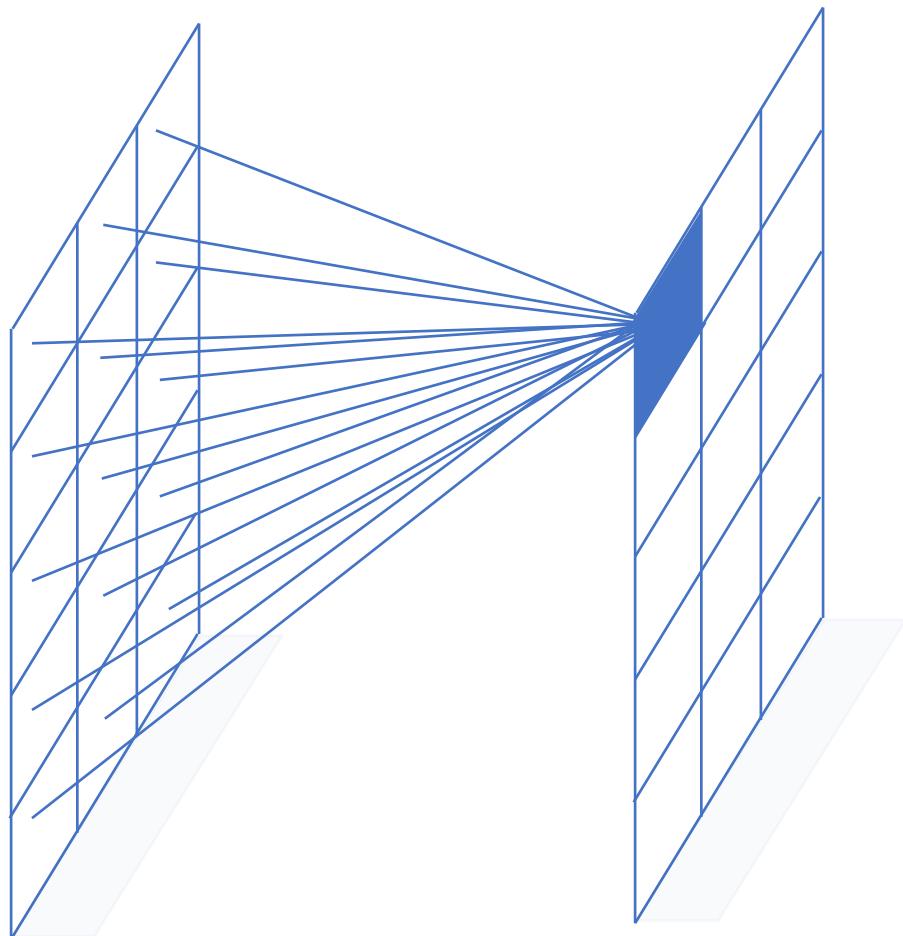
- Hierarchical processing
- Localized receptive fields



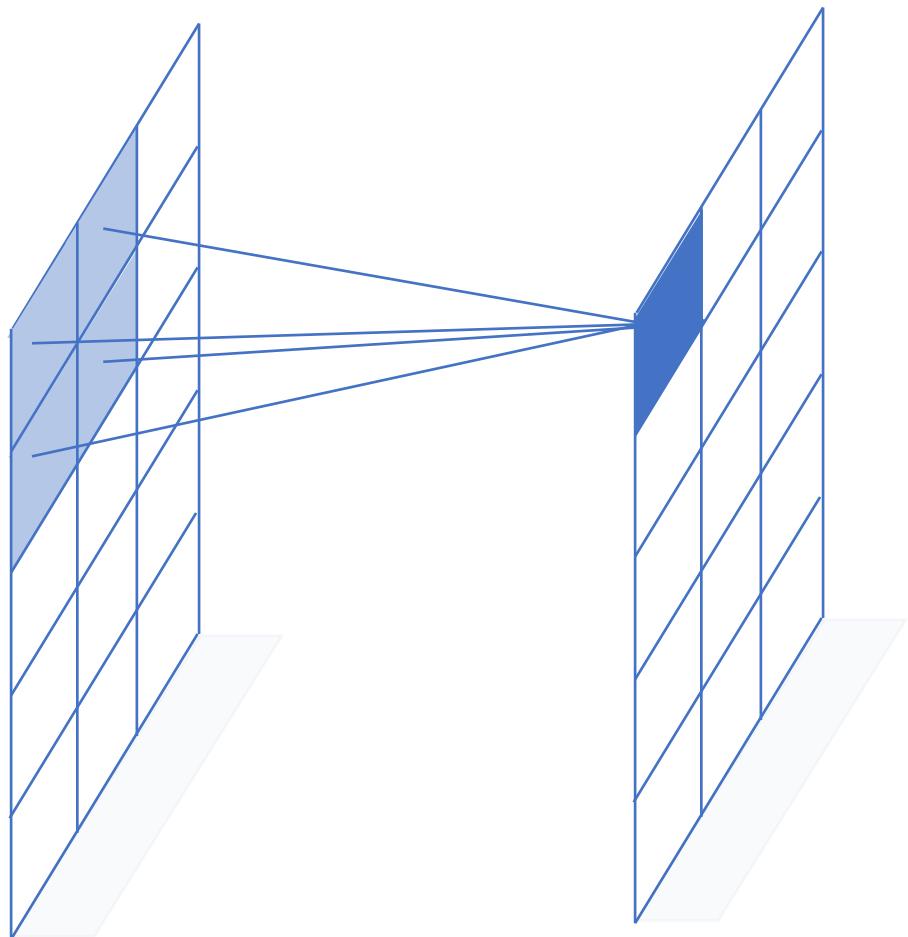
Alex Kalinin

<https://www.linkedin.com/in/alexkalinin/>

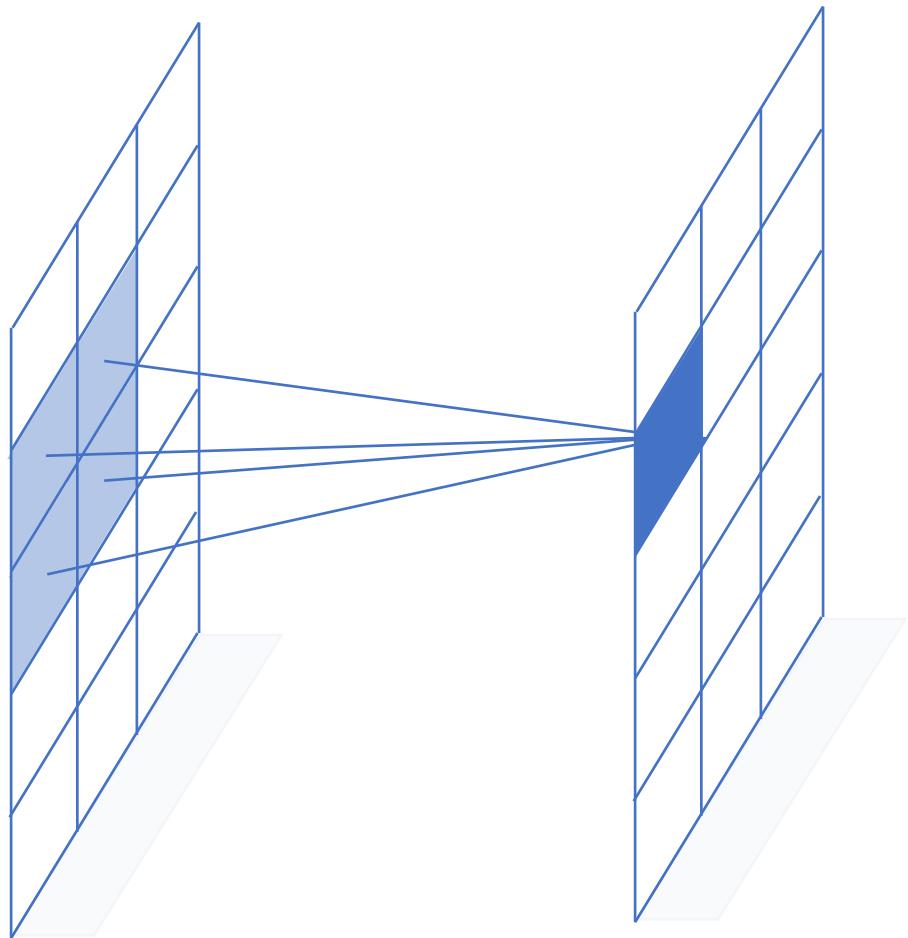
Fully Connected



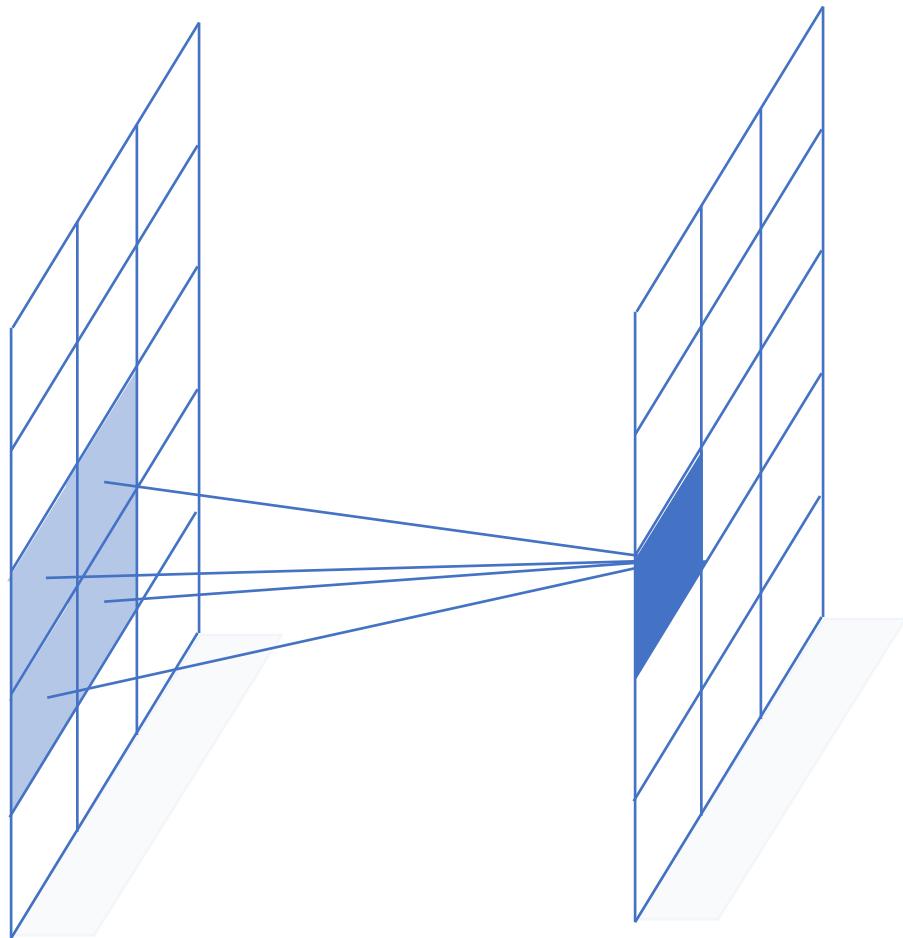
Convolution



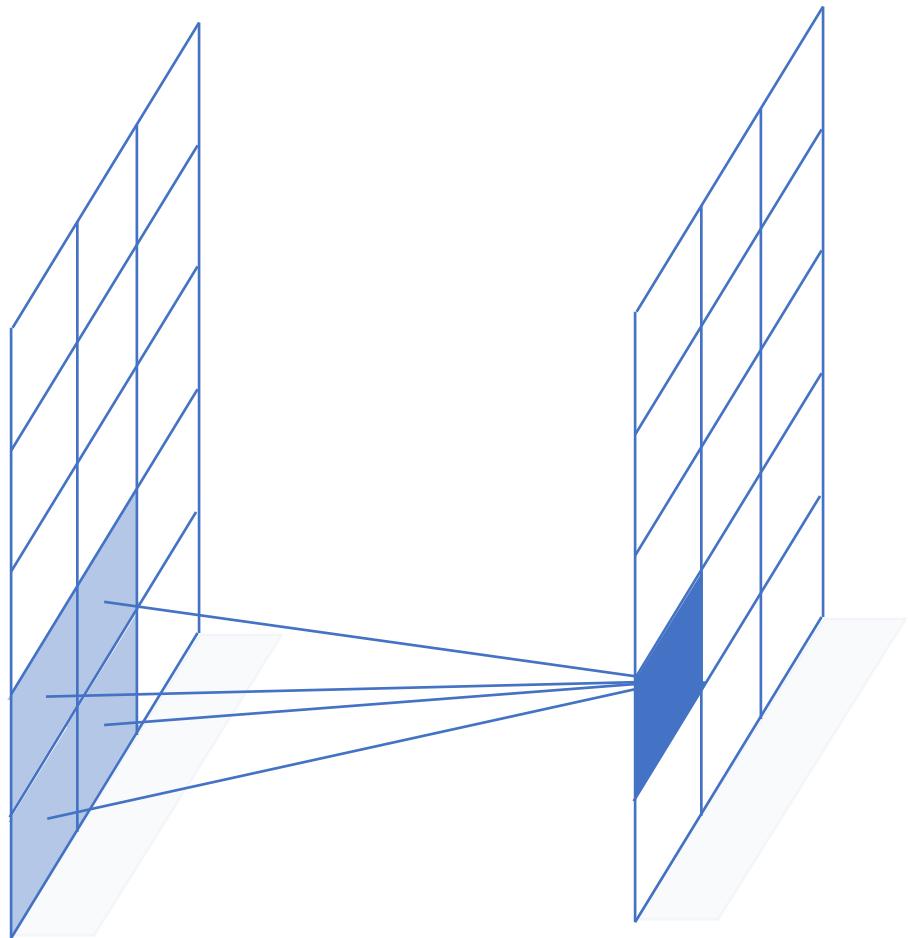
Convolution



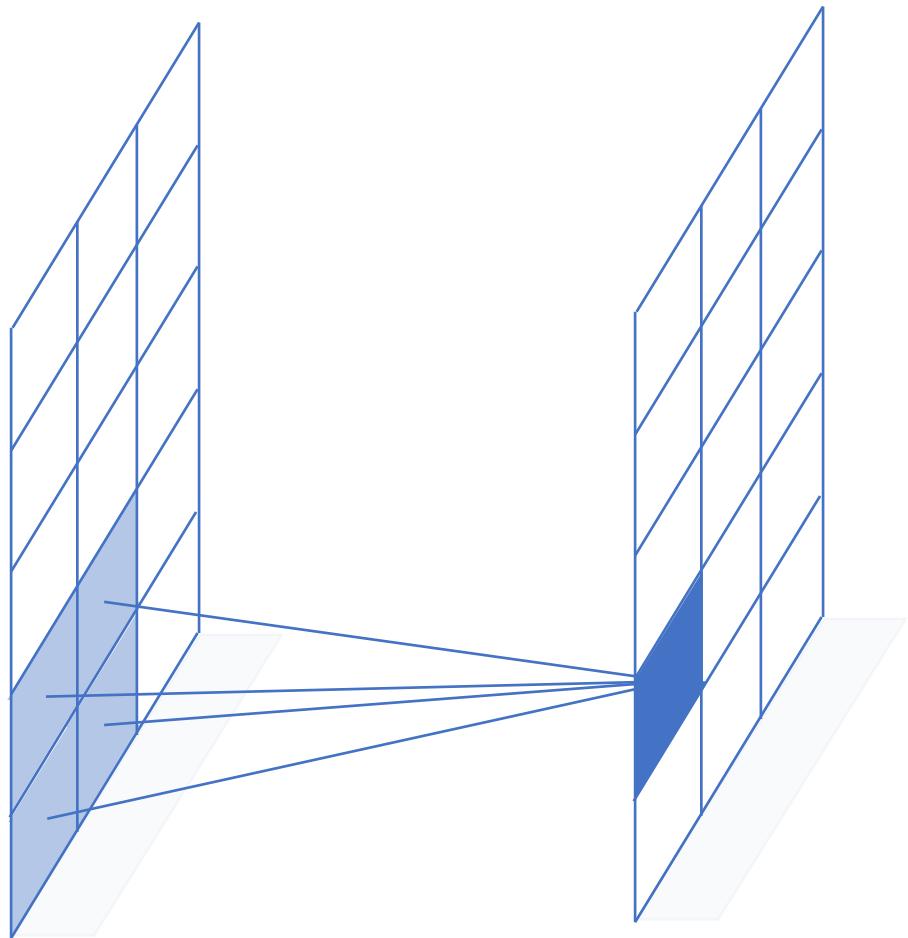
Convolution



Convolution



Convolution

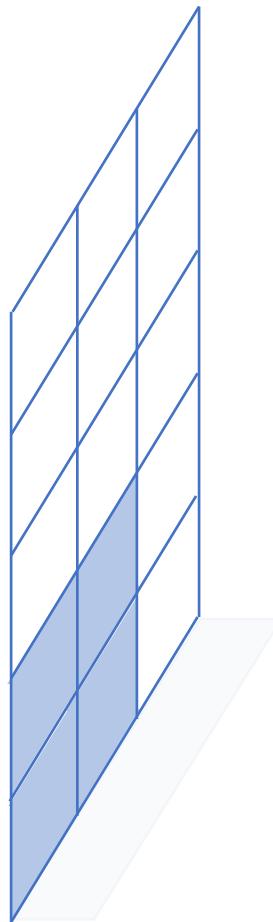


Only four weights

Alex Kalinin

<https://www.linkedin.com/in/alexkalinin/>

Convolution



Filter

0.10	-0.06
0.24	0.17



Vertical filter

Horizontal filter



Alex Kalinin

<https://www.linkedin.com/in/alexkalinin/>



Vertical filter



Horizontal filter



0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0

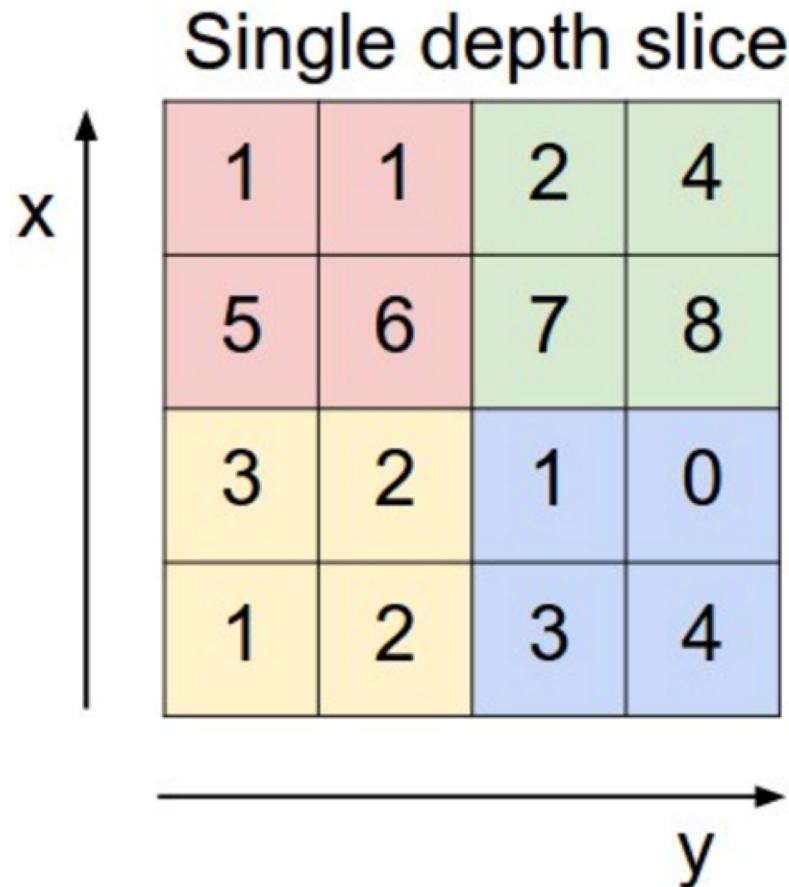


0	0	0	0	0
0	0	0	0	0
1	1	1	1	1
0	0	0	0	0
0	0	0	0	0

Alex Kalinin

<https://www.linkedin.com/in/alexkalinin/>

Pooling



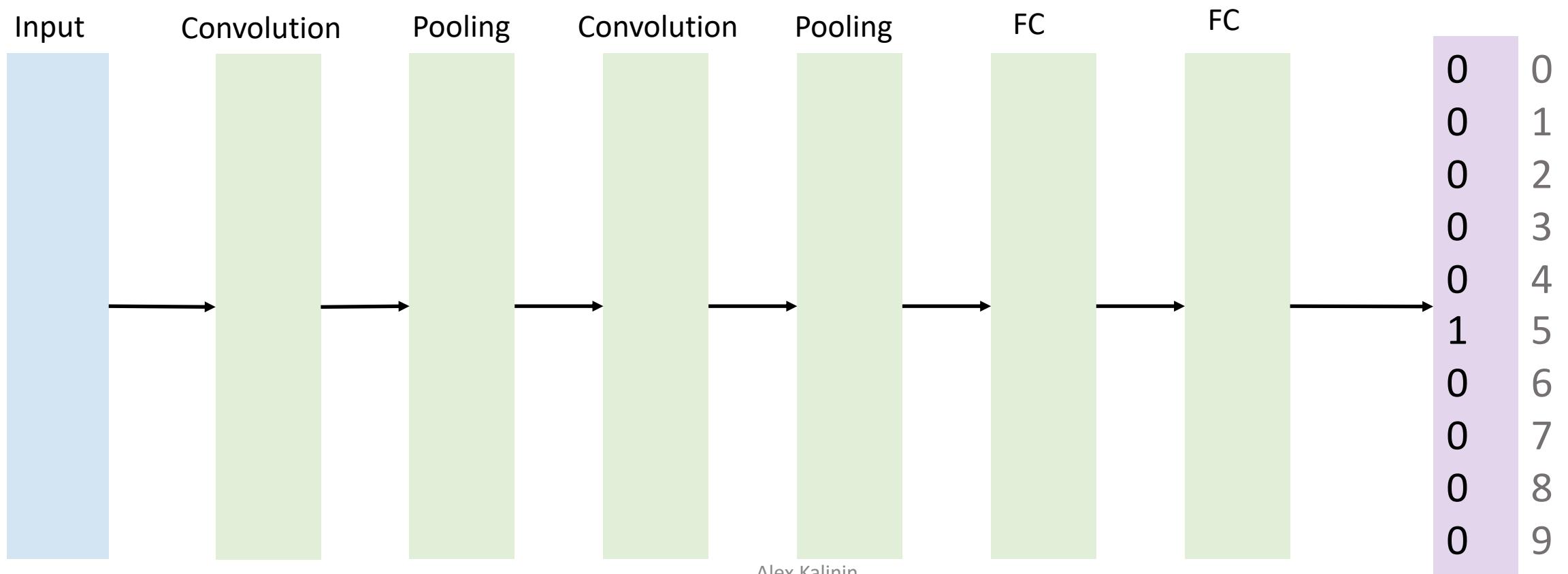
max pool with 2x2 filters
and stride 2

A 2x2 matrix representing the max pool output. The top-left cell contains 6 (pink), the top-right cell contains 8 (light green), the bottom-left cell contains 3 (yellow), and the bottom-right cell contains 4 (light blue). A horizontal arrow points from the bottom-left corner of the input grid to this output matrix.

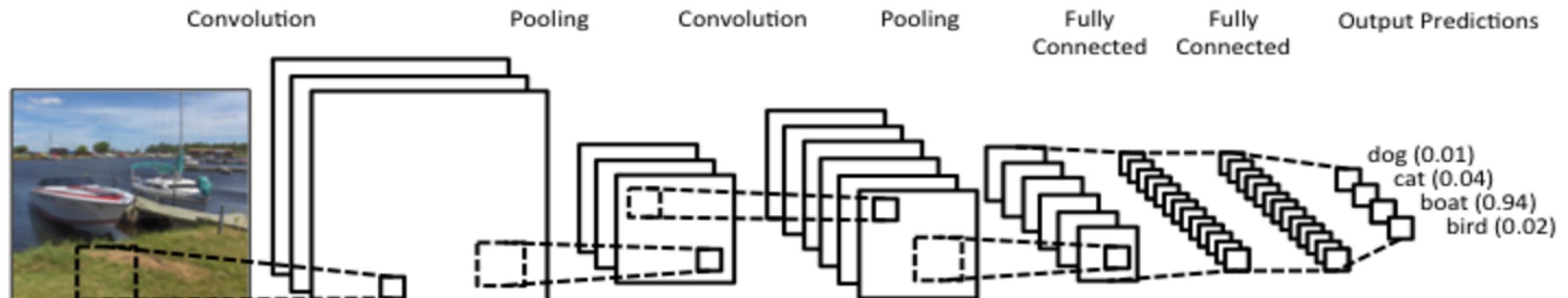
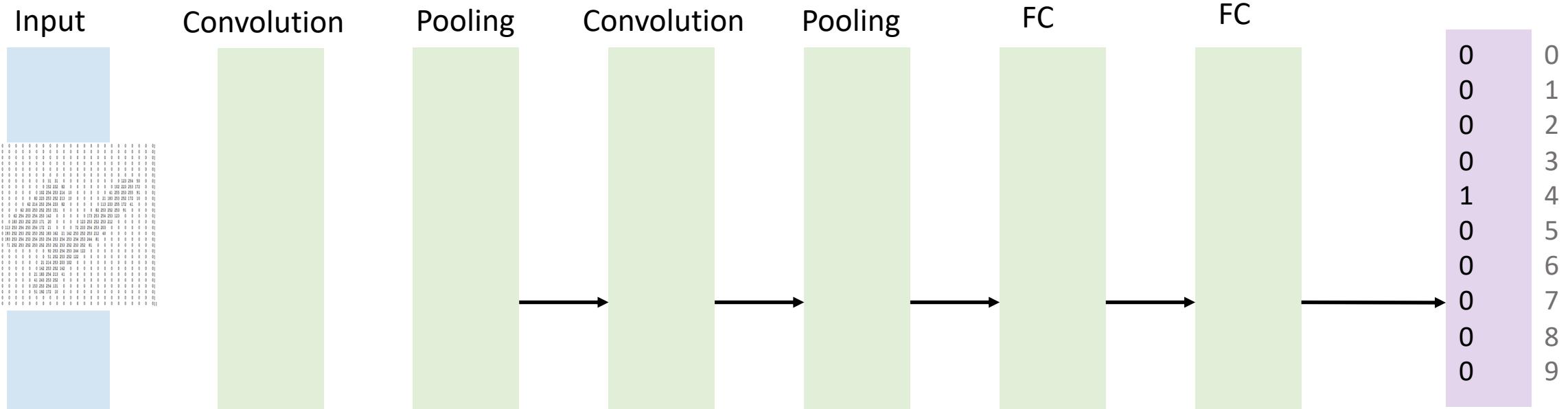
6	8
3	4

Convolutional Network

- Hierarchical processing
- Localized receptive fields



Convolutional Network



Alex Kalinin

<https://www.linkedin.com/in/alexkalinin/>

Workshop

Access

- Use <https://goo.gl/DTb6CK>
- Download Docker **Locally** (do this at home)
 - <https://hub.docker.com/r/kalininalex/bigdl-workshop/>
- Docker in your **AWS** (now is good)
- **git clone https://github.com/alex-kalinin/lenet-bigdl**

Questions?

- GitHub: <https://github.com/alex-kalinin/letnet-bigdl/>
- LinkedIn: <https://www.linkedin.com/in/alexkalinin/>

ConvNetJS

- <http://cs.stanford.edu/people/karpathy/convnetjs/>
- <http://scs.ryerson.ca/~aharley/vis/conv/>