Step 1 - Convolution

Edge Enhance:

	0	0	0	
	-1	1	0	
	0	0	0	

Image Source: docs.gimp.org/en/plug-in-convmatrix.html

Step 1 - Convolution

Edge Detect:

	0	1	0	
	1	-4	1	
	0	1	0	

Image Source: docs.gimp.org/en/plug-in-convmatrix.html

Step 1 - Convolution

Emboss:

	-2	-1	0	
	-1	1	1	
	0	1	2	

Image Source: docs.gimp.org/en/plug-in-convmatrix.html

Step 1 - Convolution

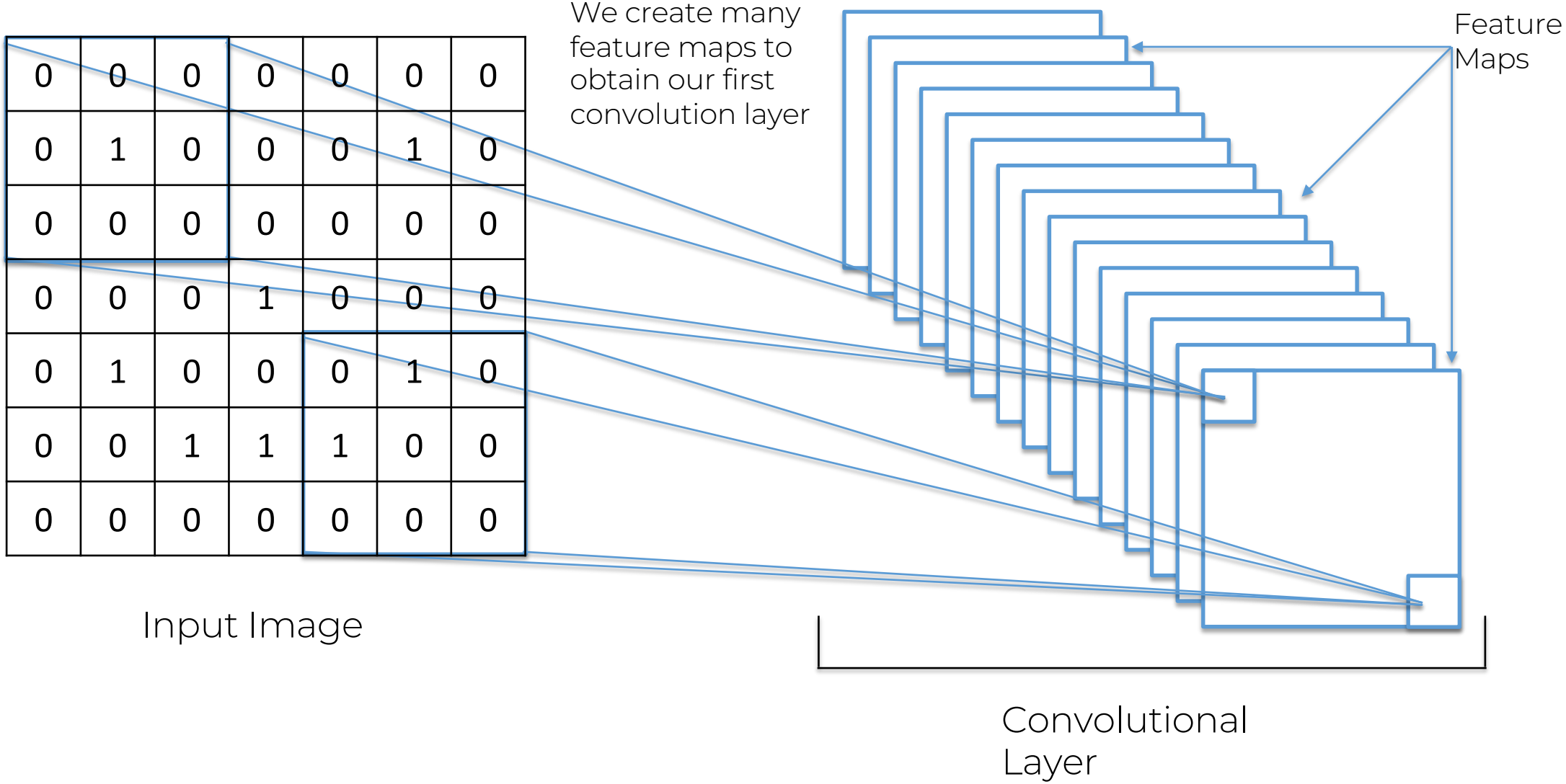
$*$

1	0	-1
2	0	-2
1	0	-1

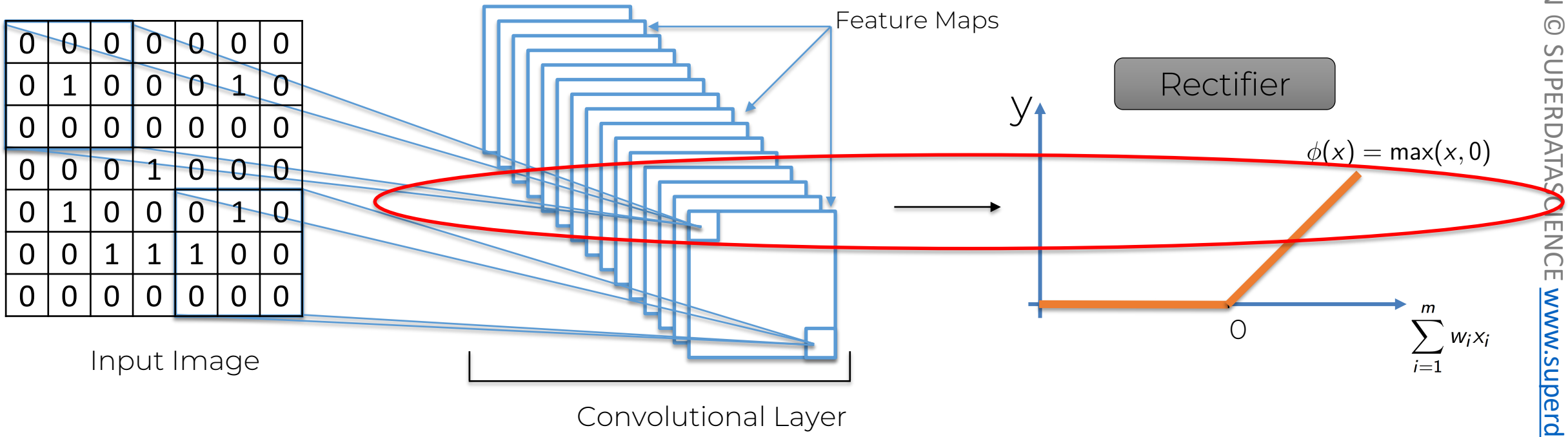
Image Source: eonardoaraujosantos.gitbooks.io

Step 1(B) – ReLU Layer

Step 1(B) – ReLU Layer



Step 1(B) – ReLU Layer



Step 1(B) – ReLU Layer

Image Source: http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf

Step 1(B) – ReLU Layer

Image Source: http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf

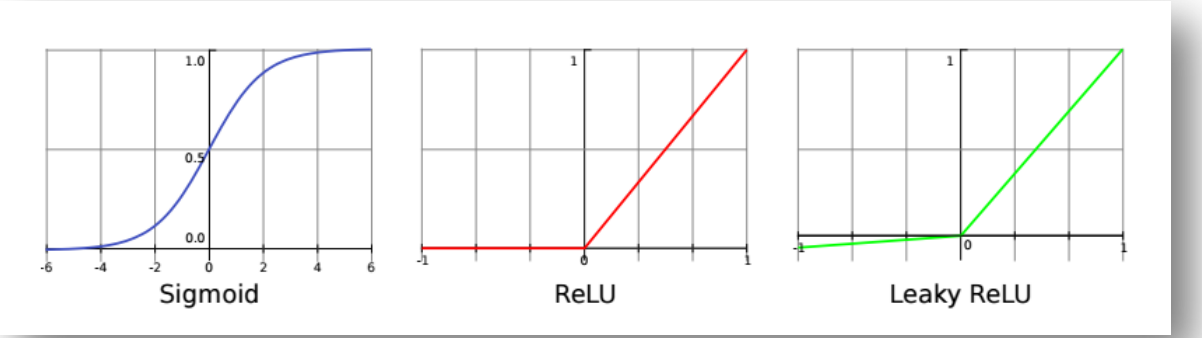
Step 1(B) – ReLU Layer

Image Source: http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf

Step 1(B) – ReLU Layer

Additional Reading:

Understanding Convolutional Neural Networks with A Mathematical Model



By C.-C. Jay Kuo (2016)

Link:

https://arxiv.org/pdf/1609.04112.pdf

Step 1(B) – ReLU Layer

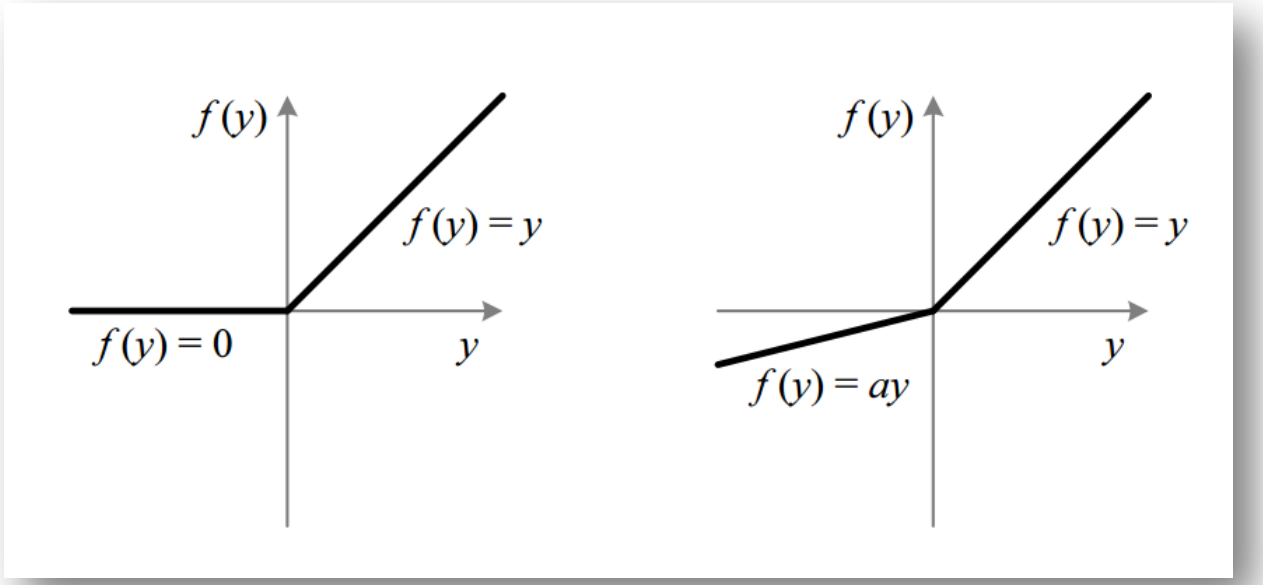
Additional Reading:

Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification

By Kaiming He et al. (2015)

Link:

https://arxiv.org/pdf/1502.01852.pdf



Step 2 – Max Pooling

Step 2 - Max Pooling

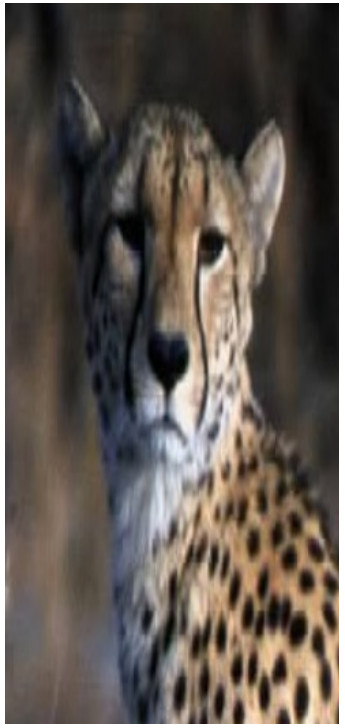


Image Source: Wikipedia

Step 2 - Max Pooling

Image Source: Wikipedia

Step 2 - Max Pooling

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

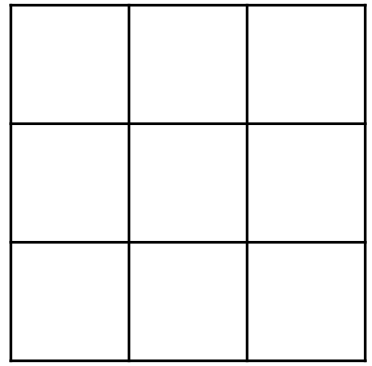
Feature Map

Step 2 - Max Pooling

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Feature Map

Max Pooling



Pooled Feature Map

Step 2 - Max Pooling

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Feature Map

Max Pooling

1		

Pooled Feature Map

Step 2 - Max Pooling

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Feature Map

Max Pooling

1	1	

Pooled Feature Map

Step 2 - Max Pooling

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Feature Map

Max Pooling

1	1	0

Pooled Feature Map

Step 2 - Max Pooling

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Feature Map

Max Pooling

1	1	0
4		

Pooled Feature Map

Step 2 - Max Pooling

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

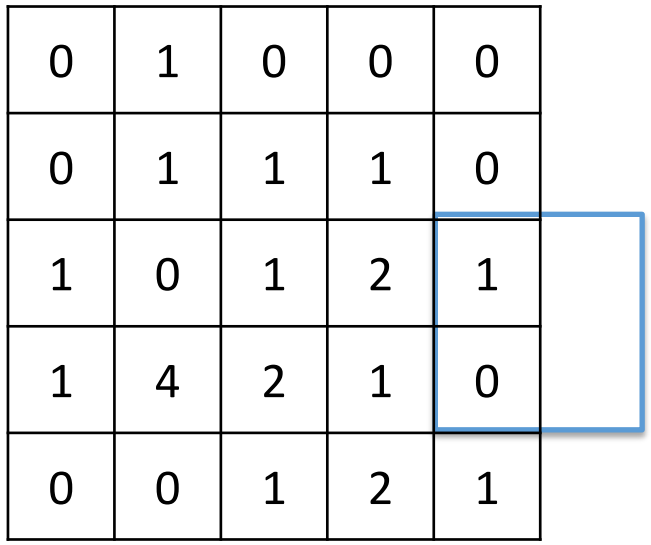
Feature Map

Max Pooling

1	1	0
4	2	

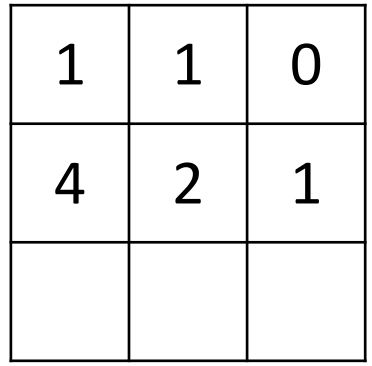
Pooled Feature Map

Step 2 - Max Pooling



Feature Map

Max Pooling



Pooled Feature Map

Step 2 - Max Pooling

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Feature Map

Max Pooling

1	1	0
4	2	1
0		

Pooled Feature Map

Step 2 - Max Pooling

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Feature Map

Max Pooling

1	1	0
4	2	1
0	2	

Pooled Feature Map

Step 2 - Max Pooling

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Feature Map

Max Pooling

1	1	0
4	2	1
0	2	1

Pooled Feature Map

Step 2 - Max Pooling

Additional Reading:

Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition

By Dominik Scherer et al. (2010)

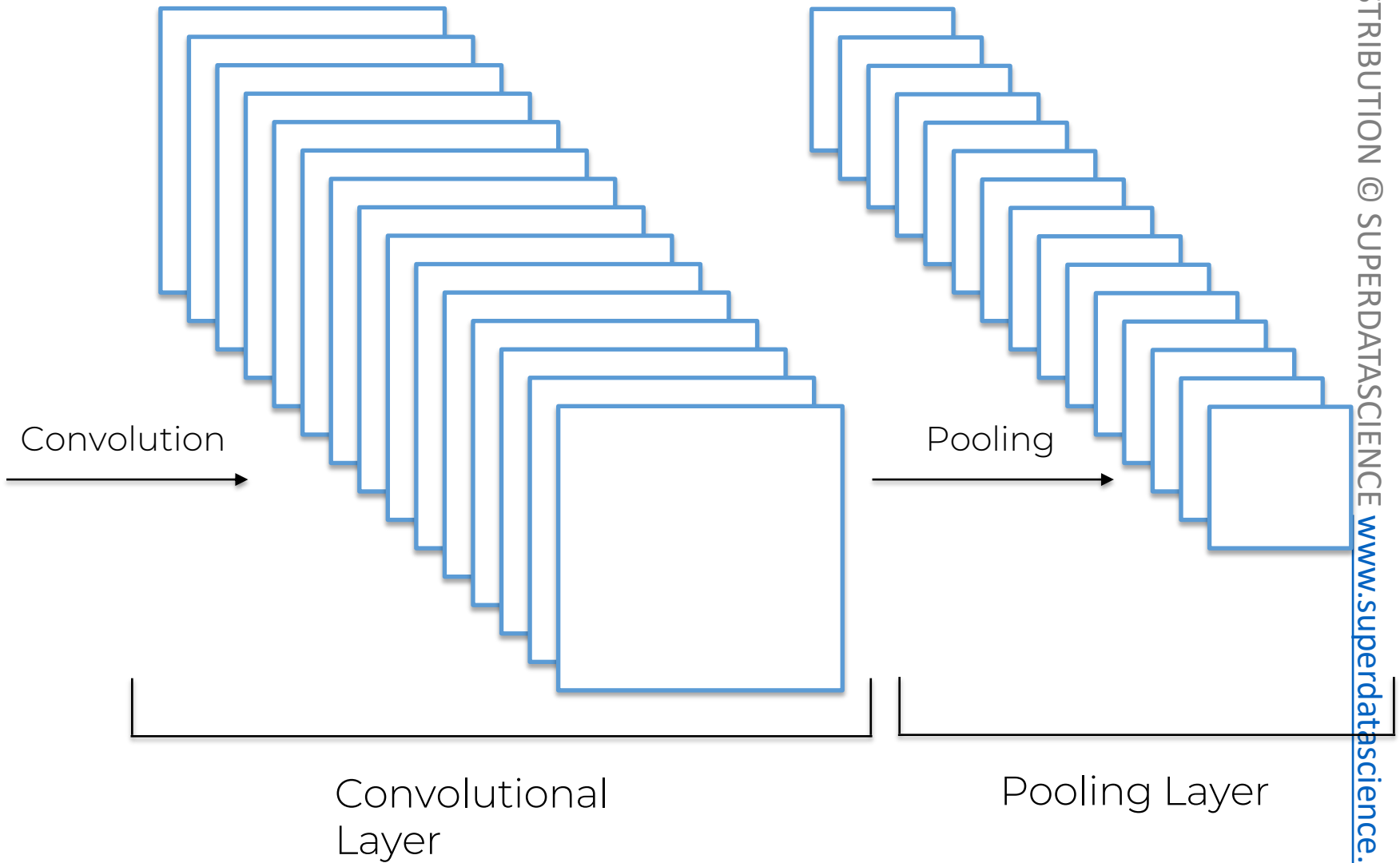
Link:

http://ais.uni-bonn.de/papers/icann2010_maxpool.pdf

Step 2 - Max Pooling

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

Input Image



Example

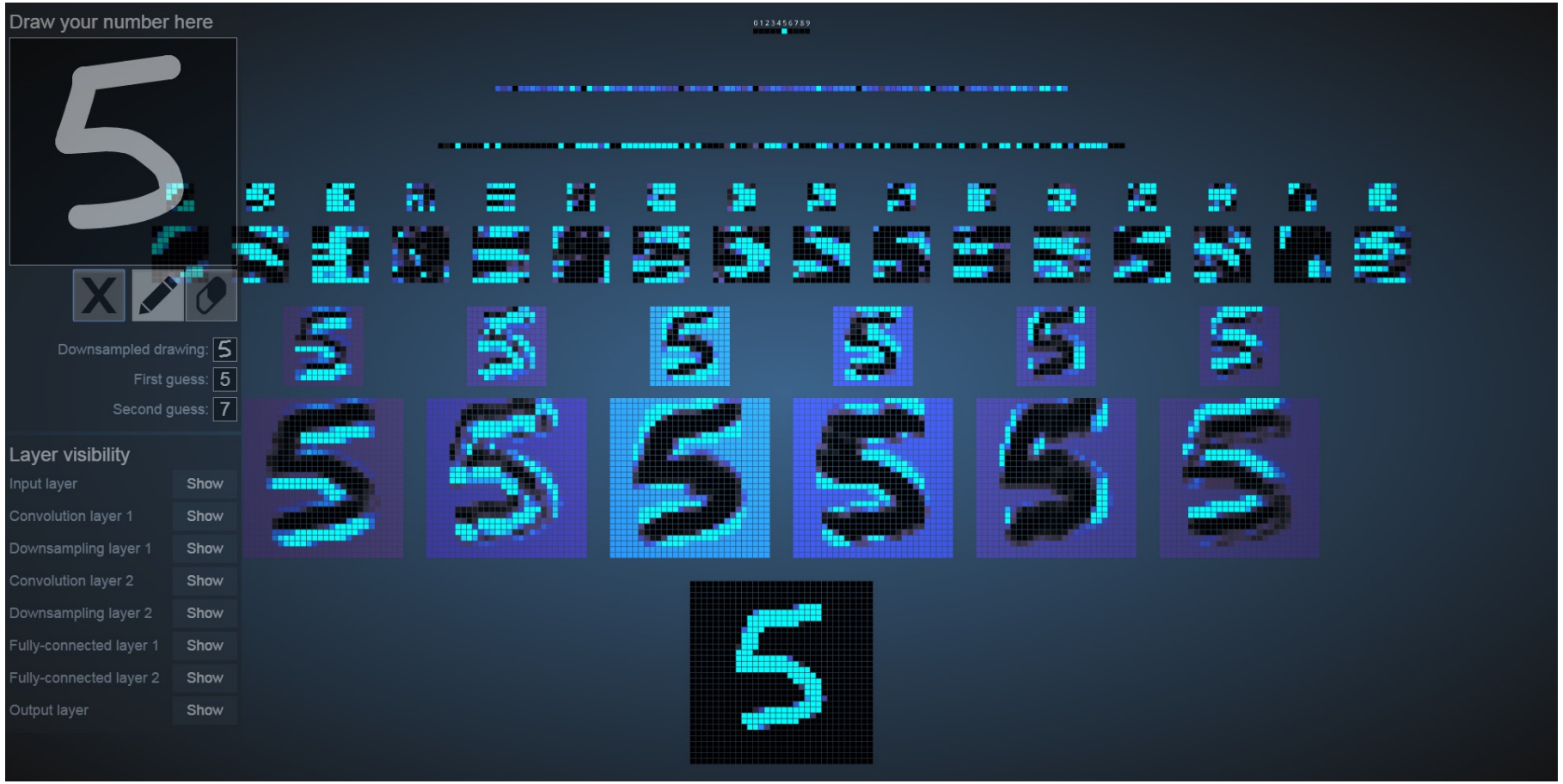


Image Source: scs.ryerson.ca/~aharley/vis/conv/flat.html

Step 3 – Flattening

Step 3 - Flattening

1	1	0
4	2	1
0	2	1

Pooled Feature Map

Step 3 - Flattening

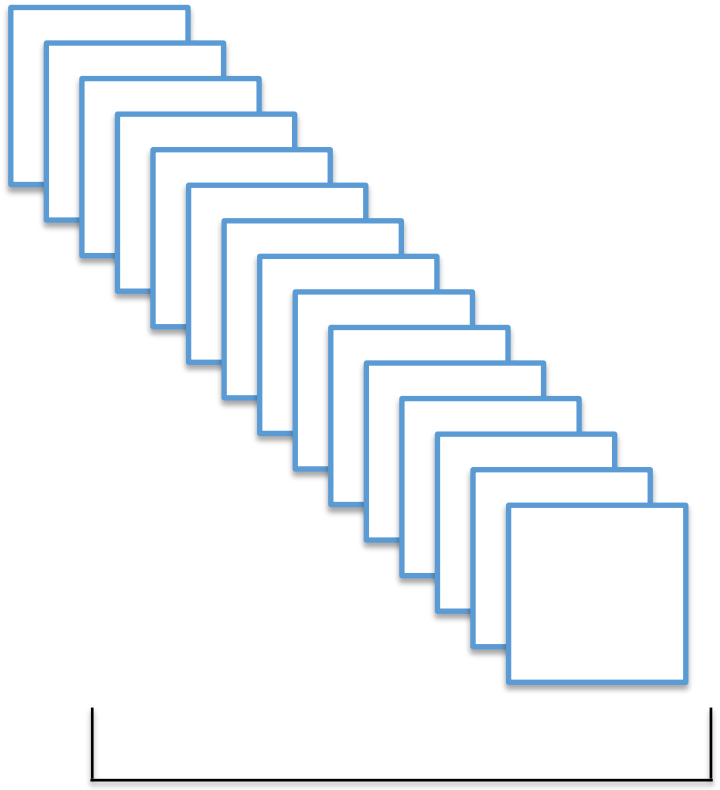
1	1	0
4	2	1
0	2	1

Pooled Feature Map

Flattening

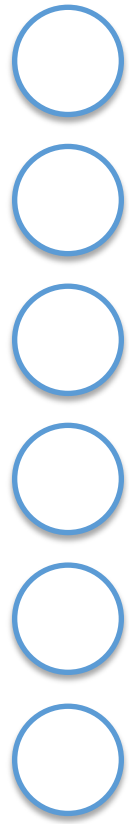
1
1
0
4
2
1
0
2
1

Step 3 - Flattening



Pooling Layer

Flattening

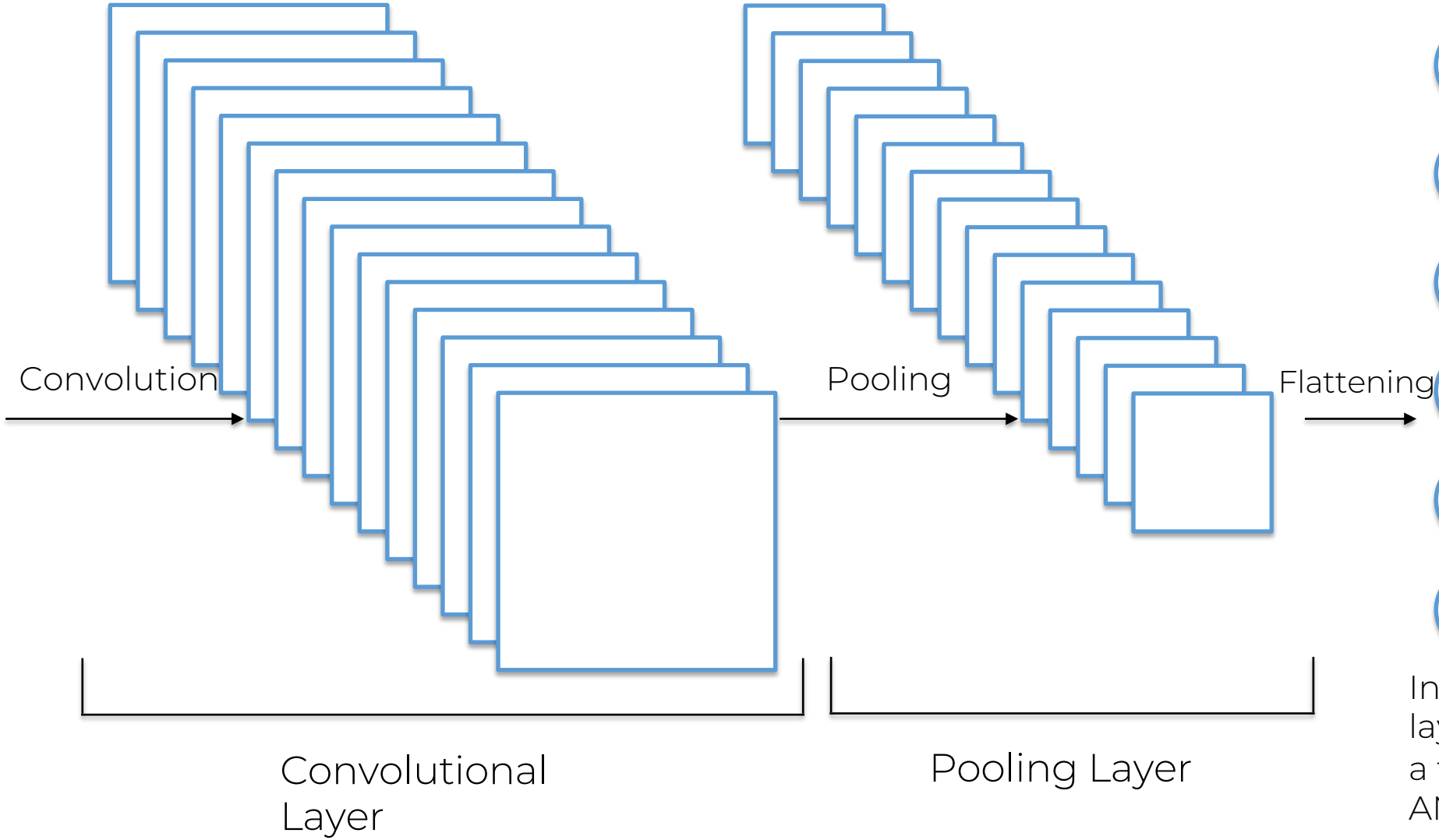


Input layer of a future ANN

Step 3 - Flattening

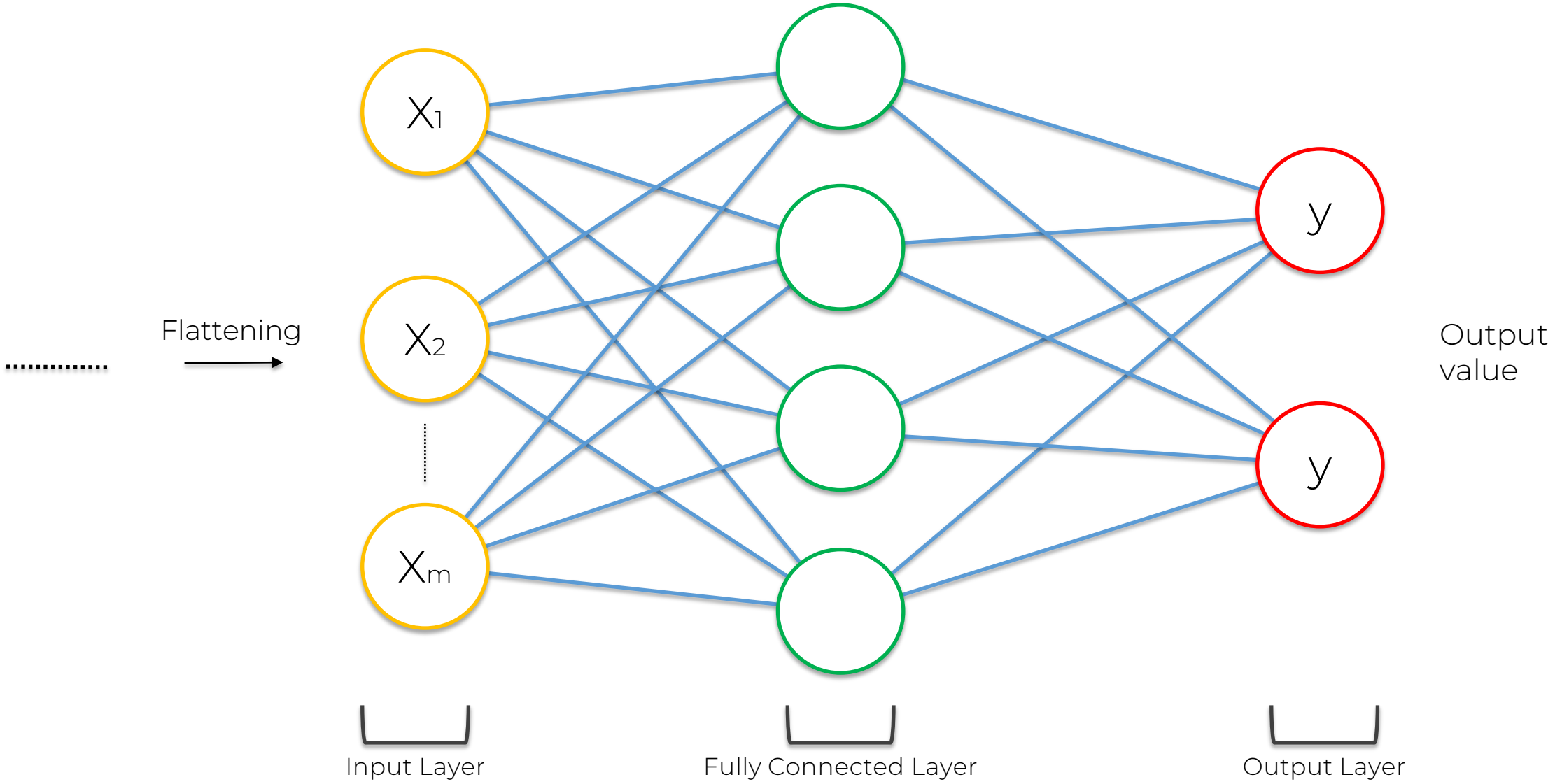
0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Input Image

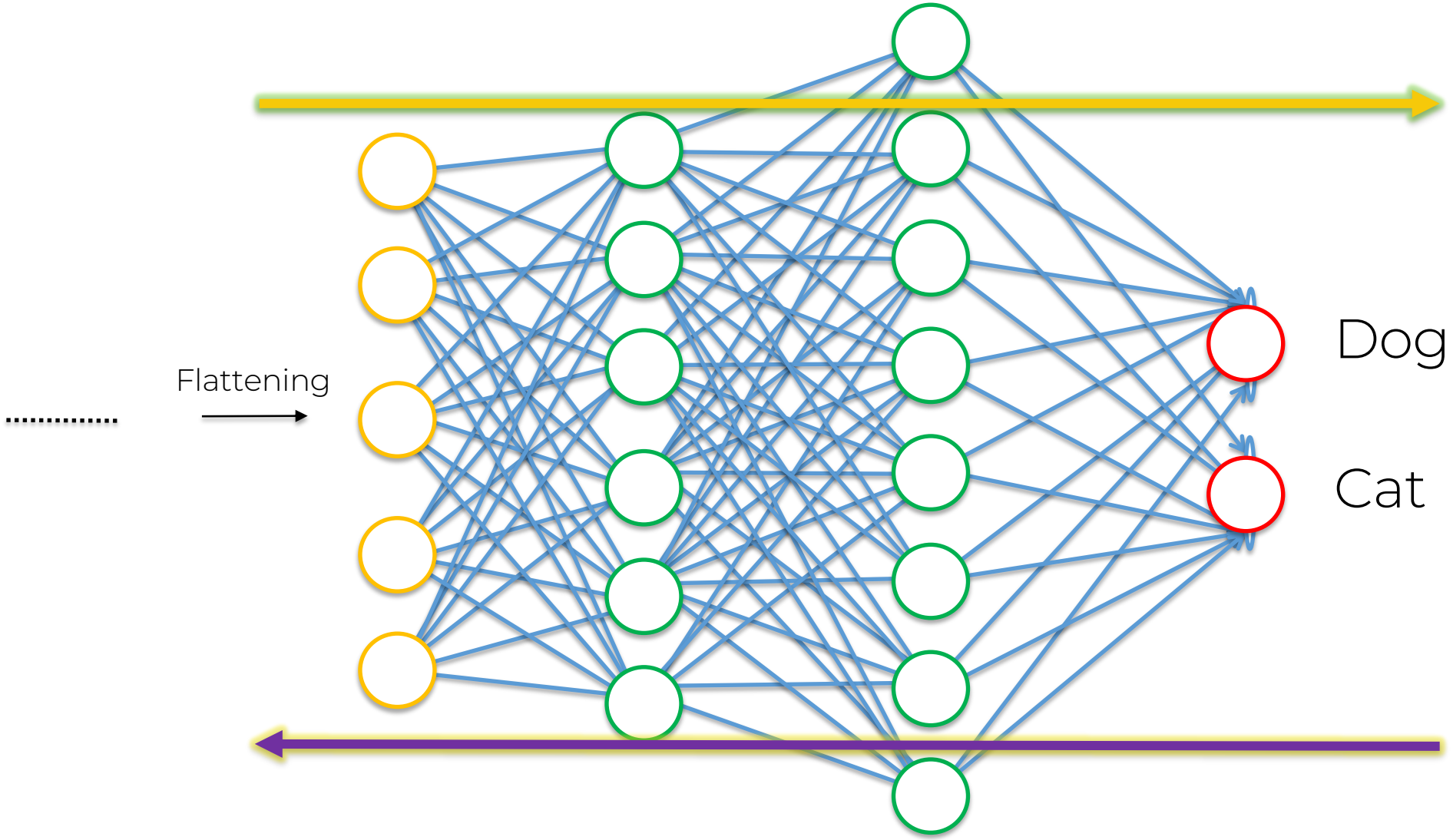


Step 4 – Full Connection

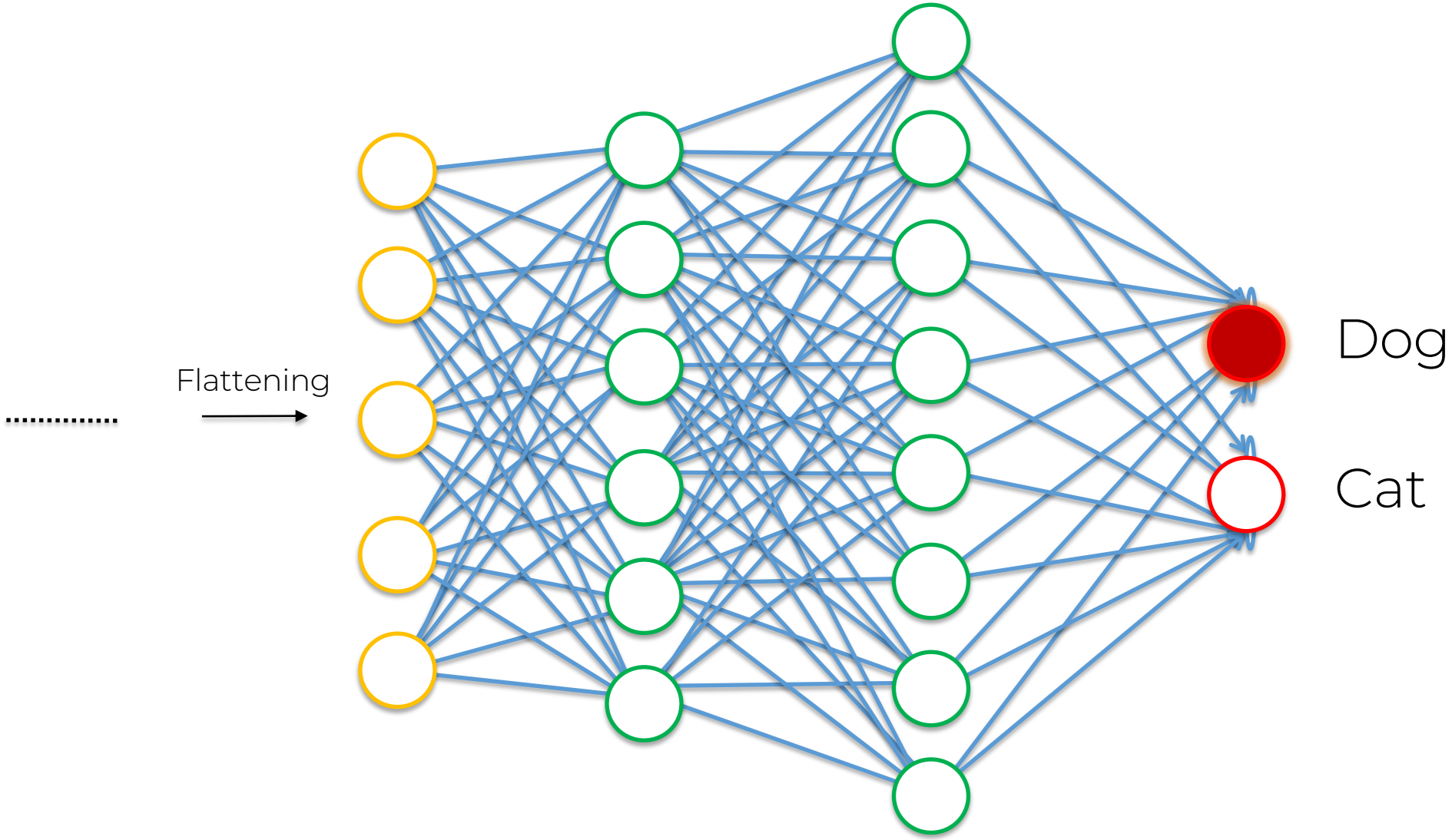
Step 4 - Full Connection



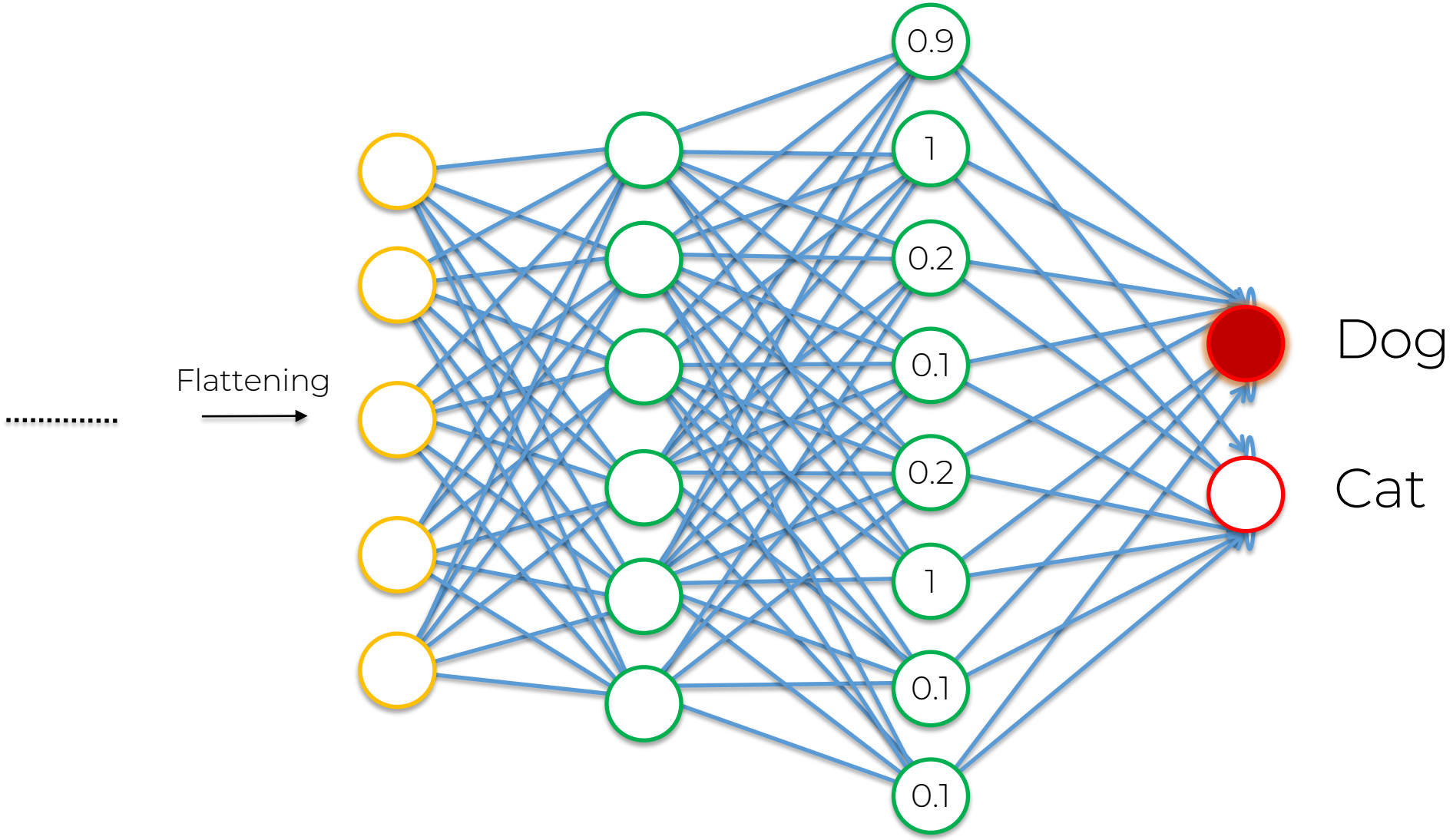
Step 4 - Full Connection



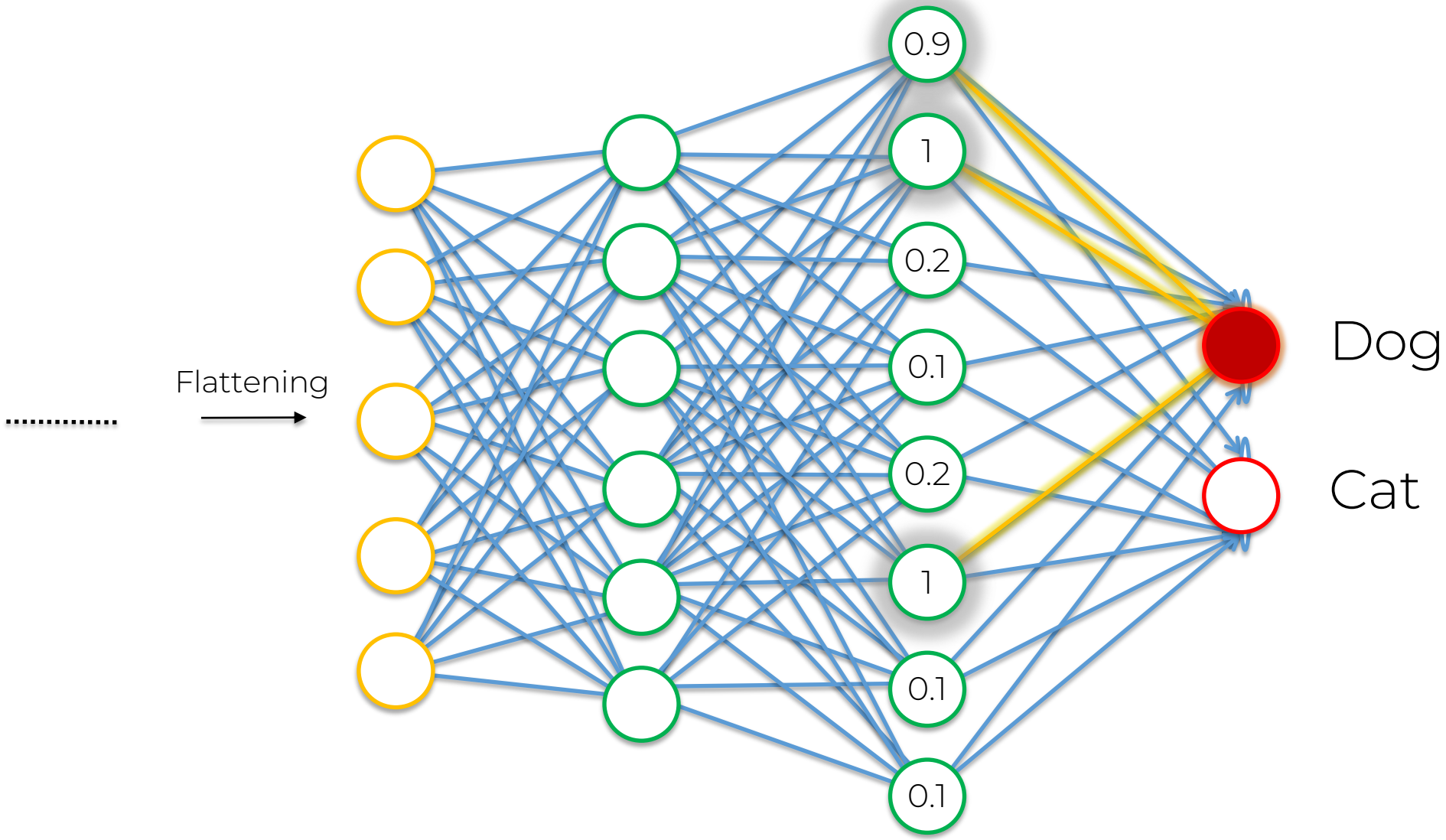
Step 4 - Full Connection



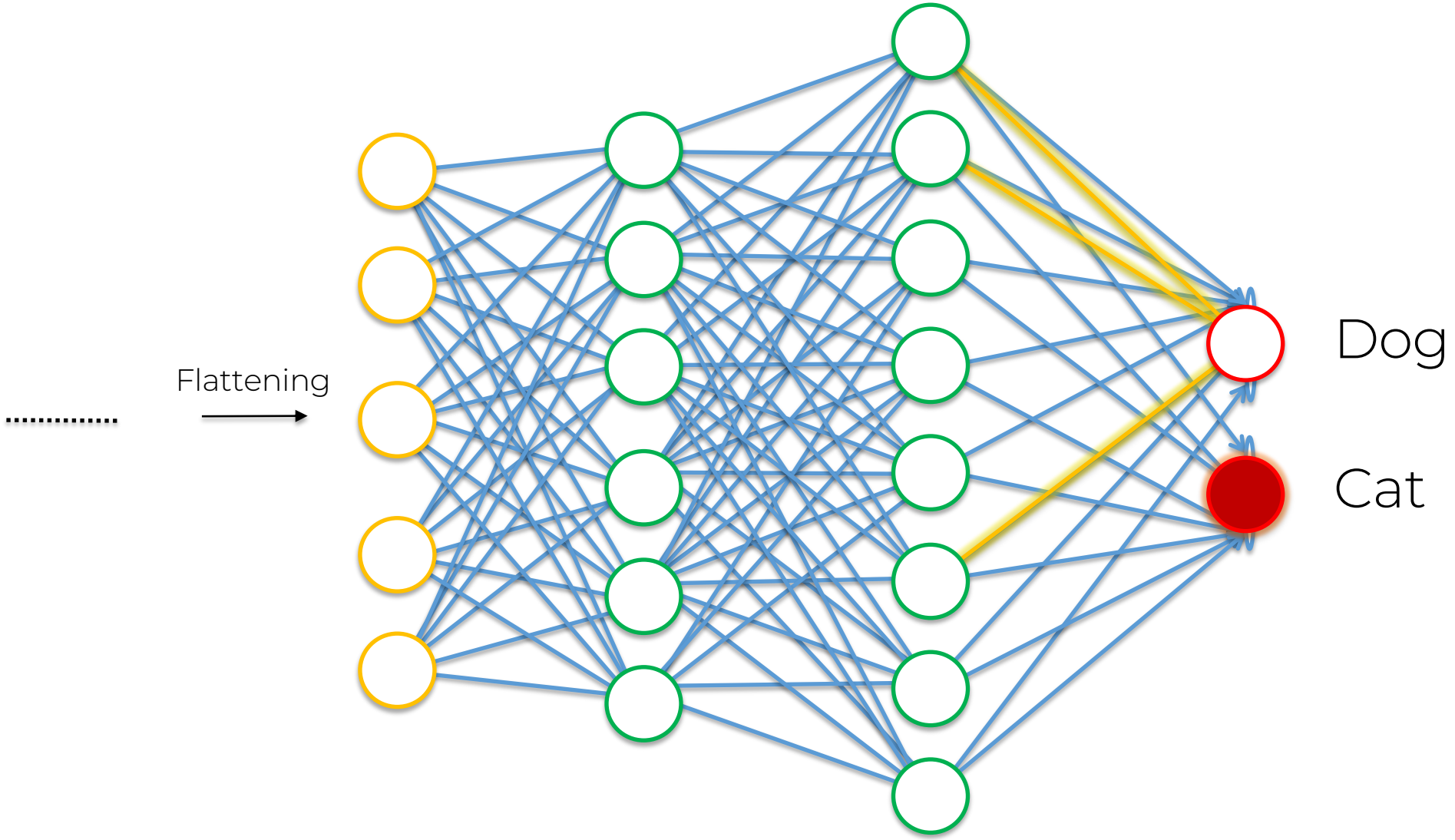
Step 4 - Full Connection



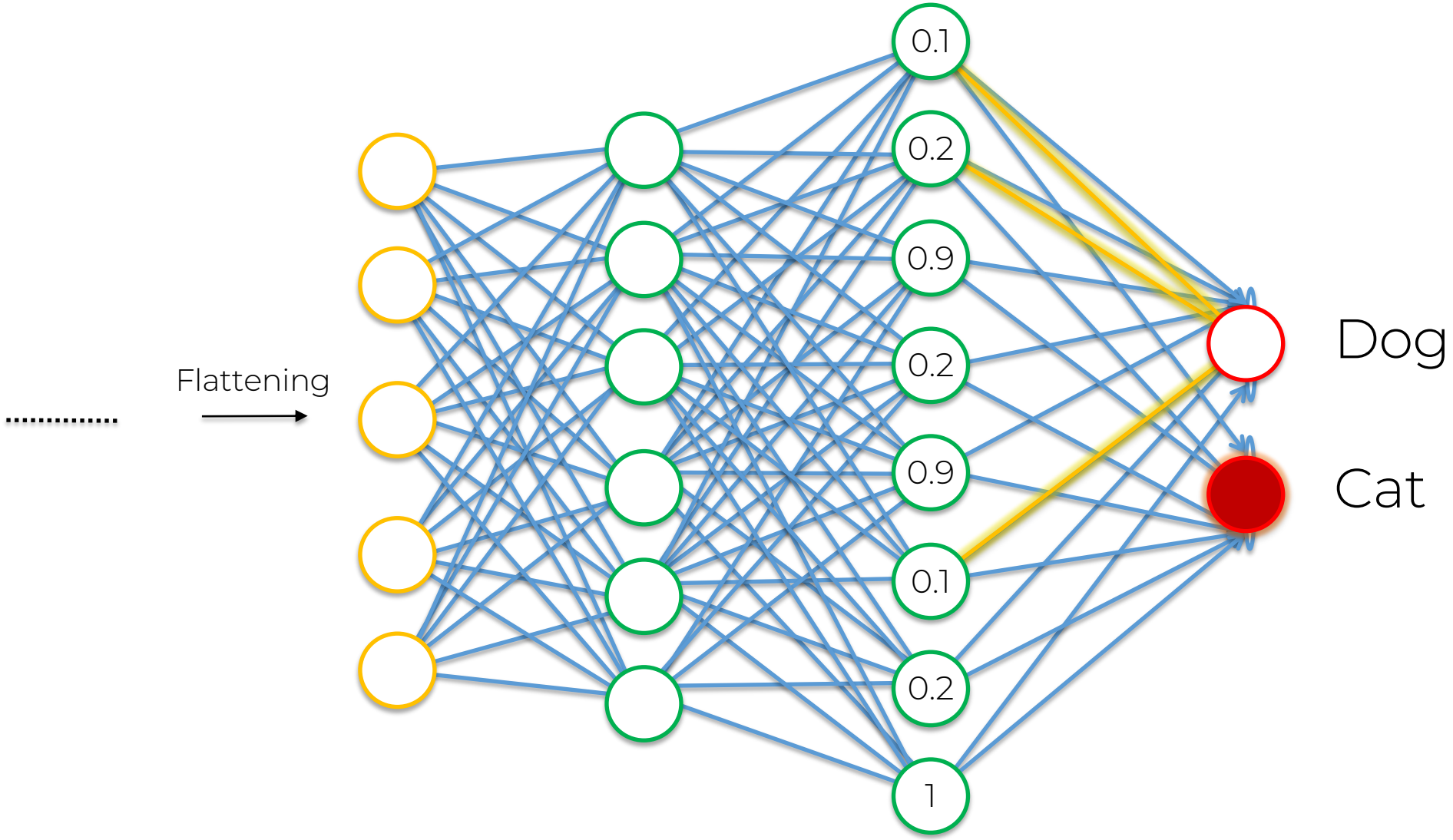
Step 4 - Full Connection



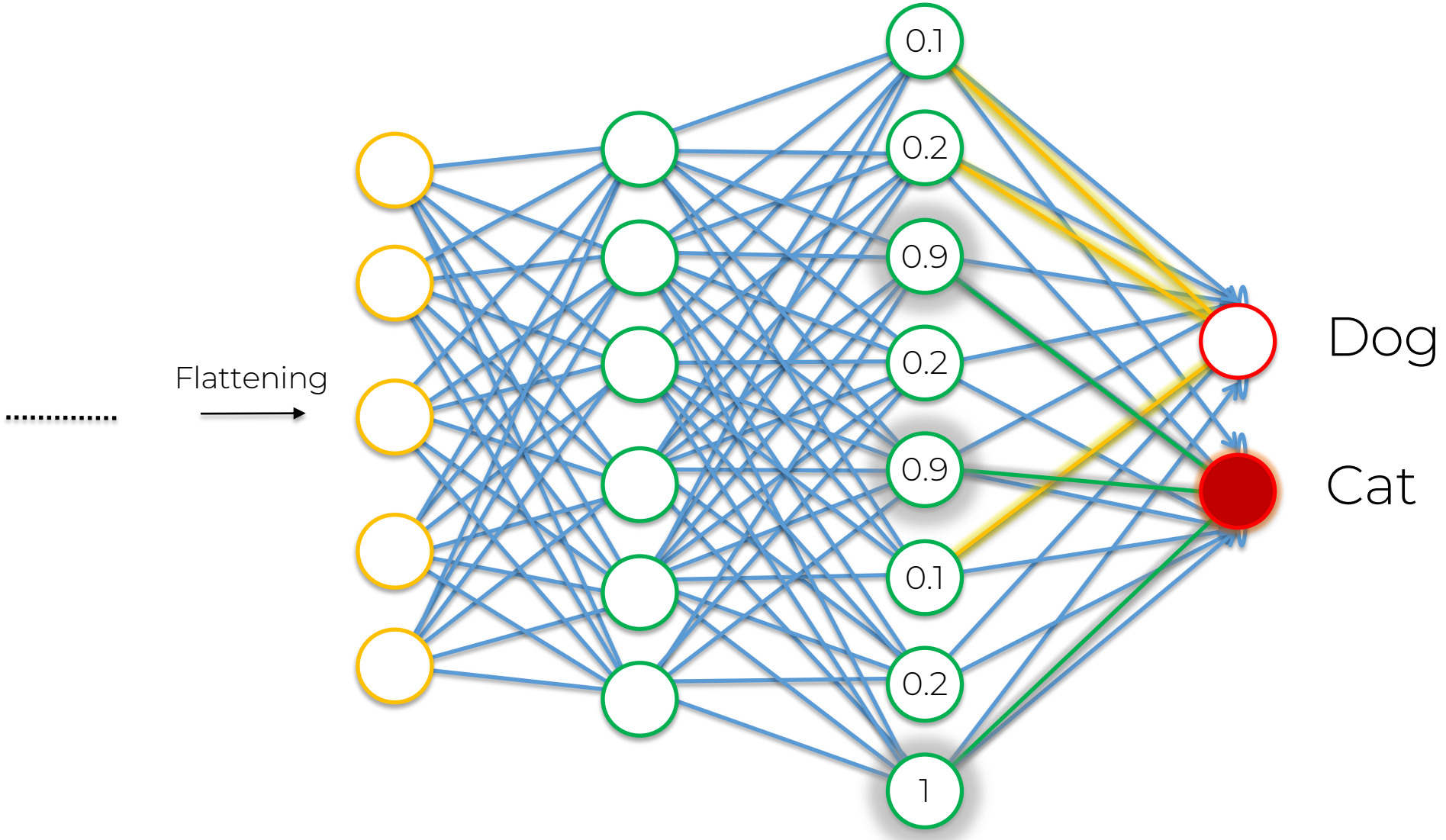
Step 4 - Full Connection



Step 4 - Full Connection

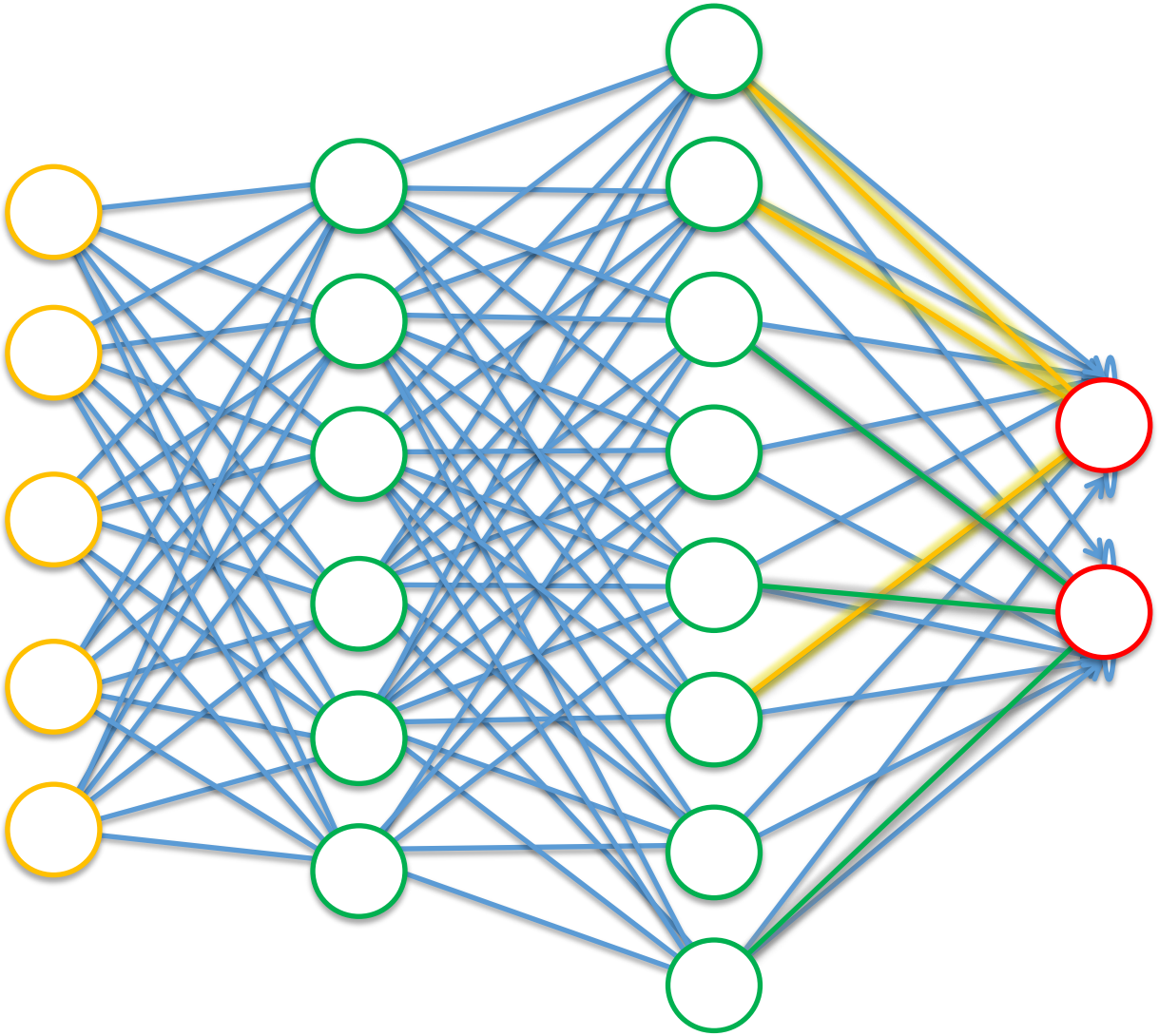


Step 4 - Full Connection



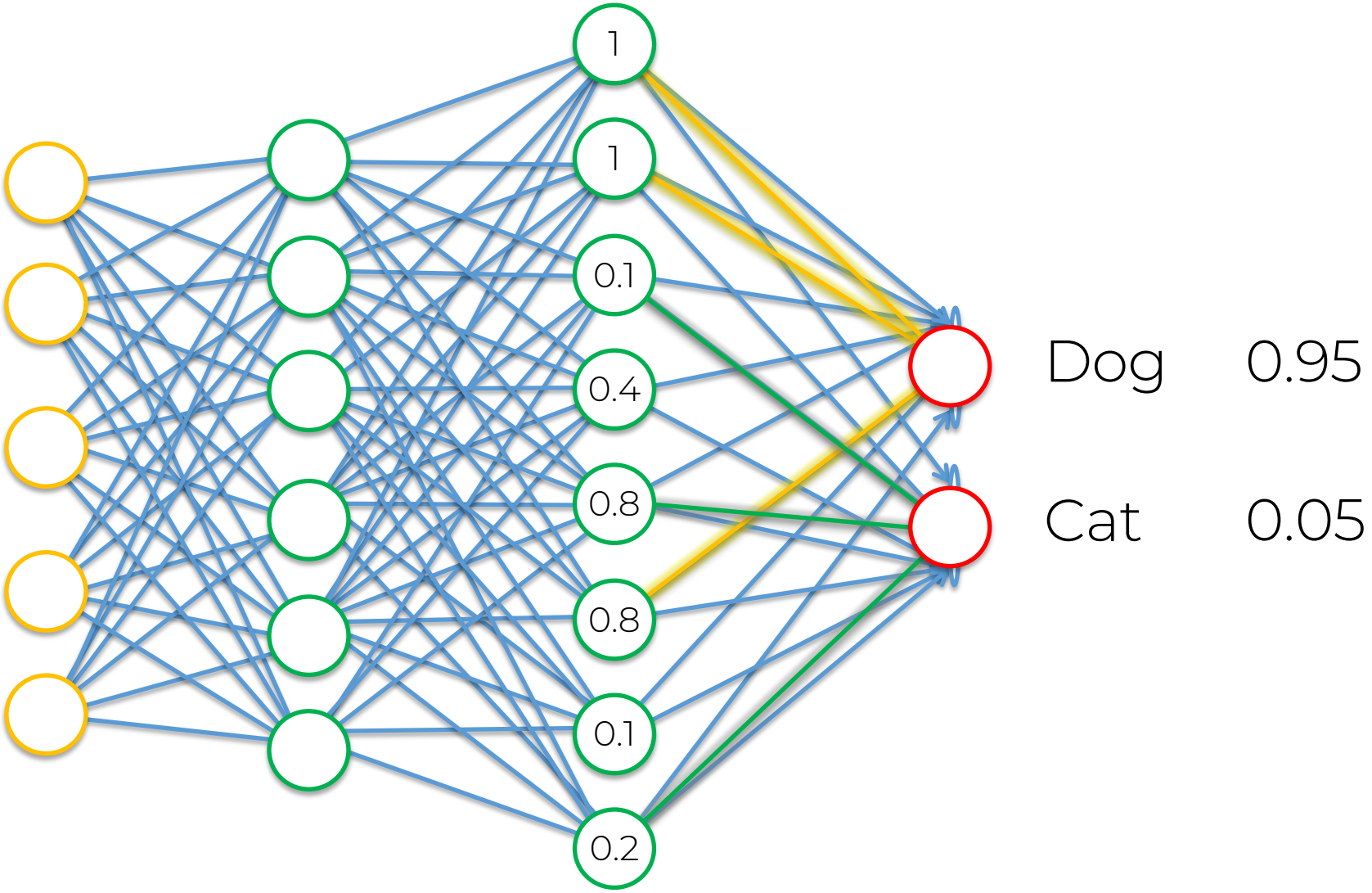
Step 4 - Full Connection

Flattening



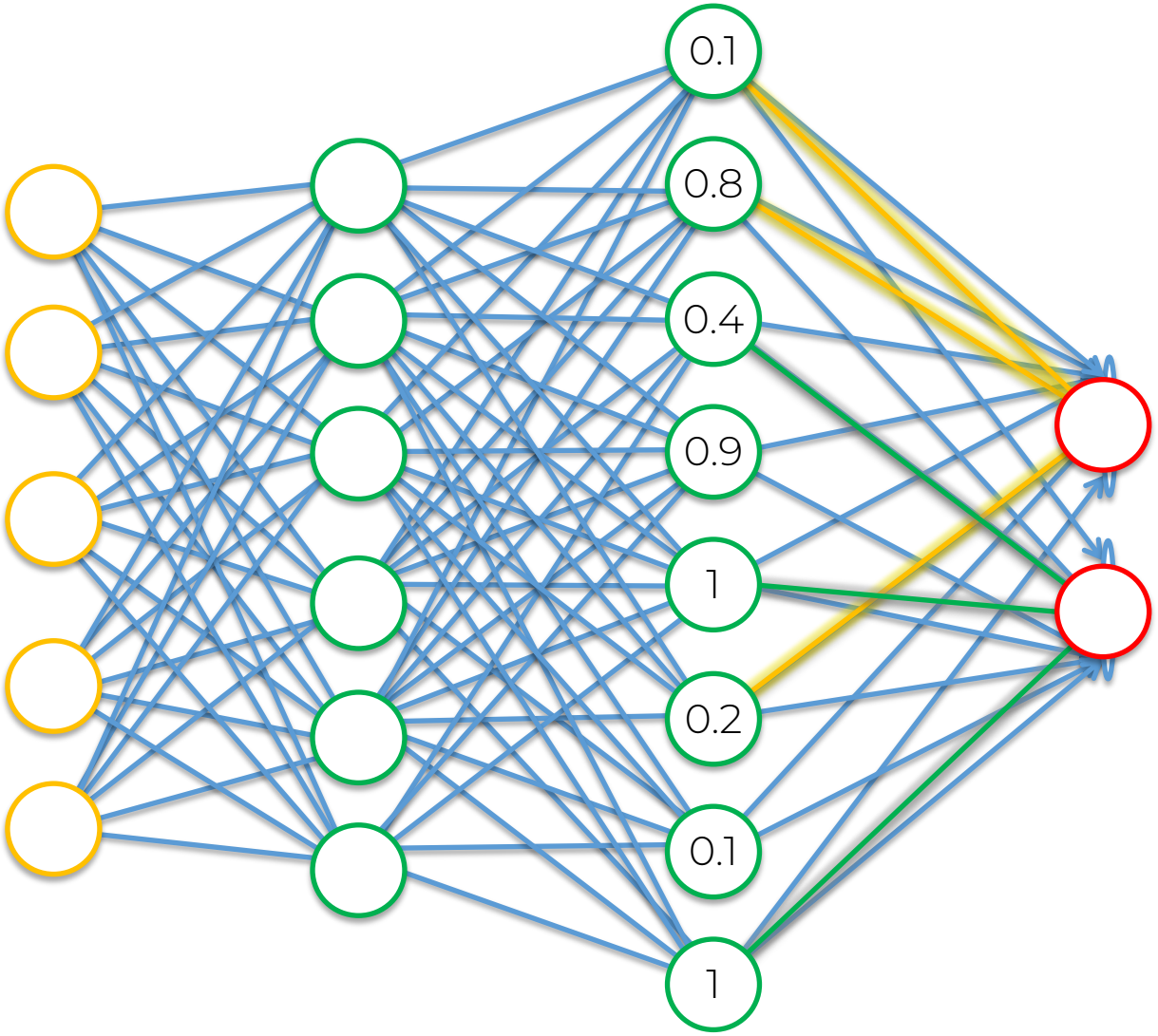
Step 4 - Full Connection

Flattening



Step 4 - Full Connection

Flattening



Dog 0.21

Cat 0.79

Step 4 - Full Connection

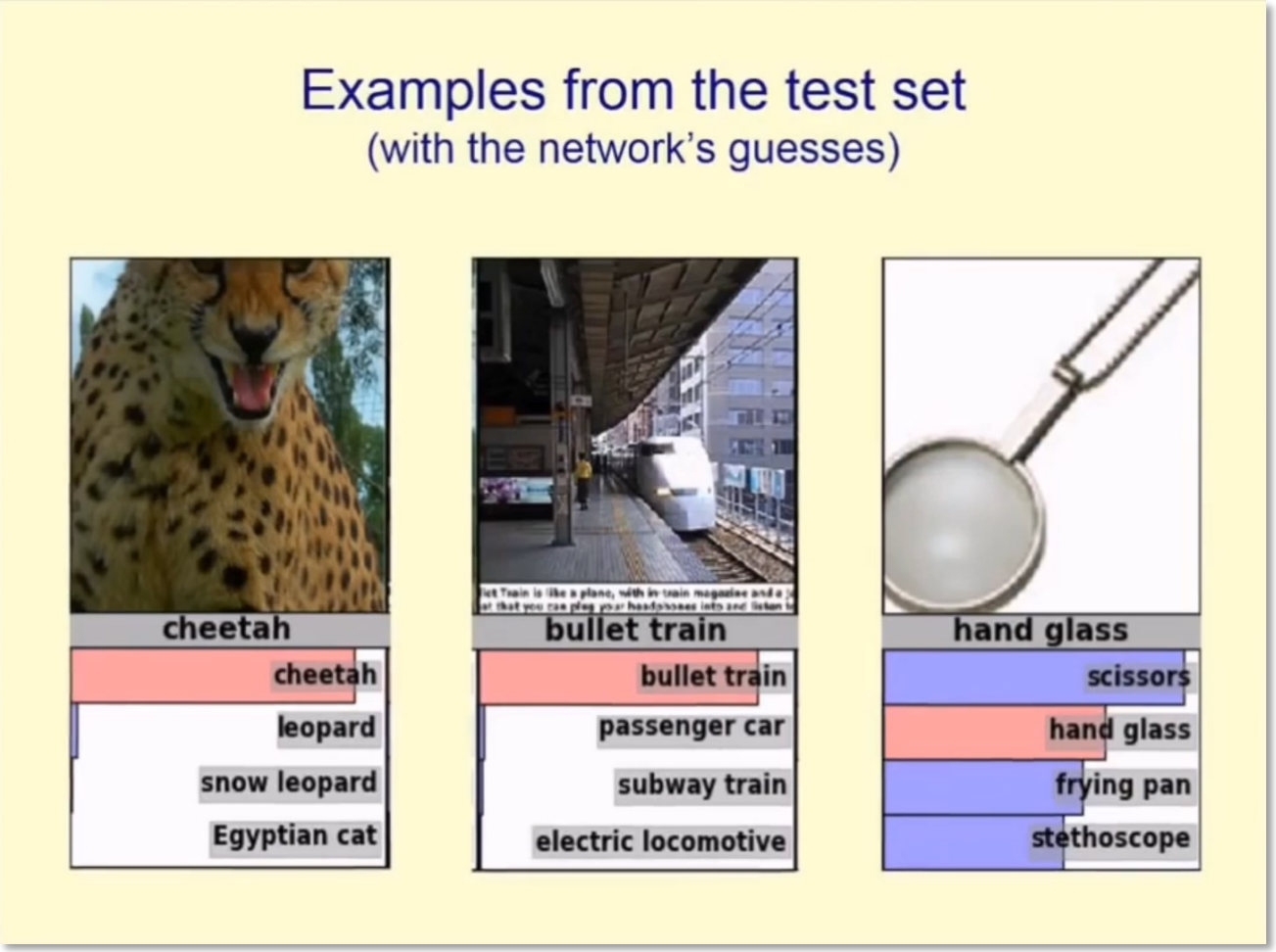
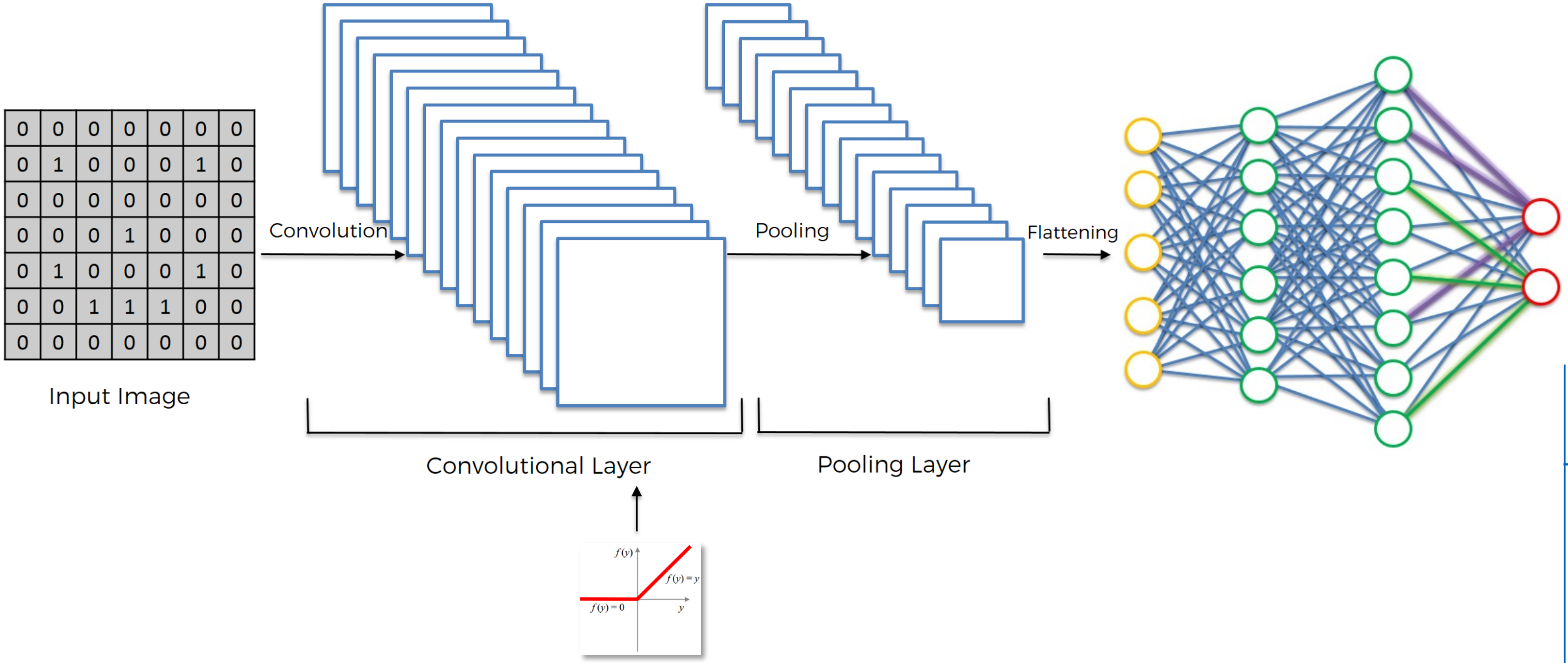


Image Source: a talk by Geoffrey Hinton

Summary

Summary

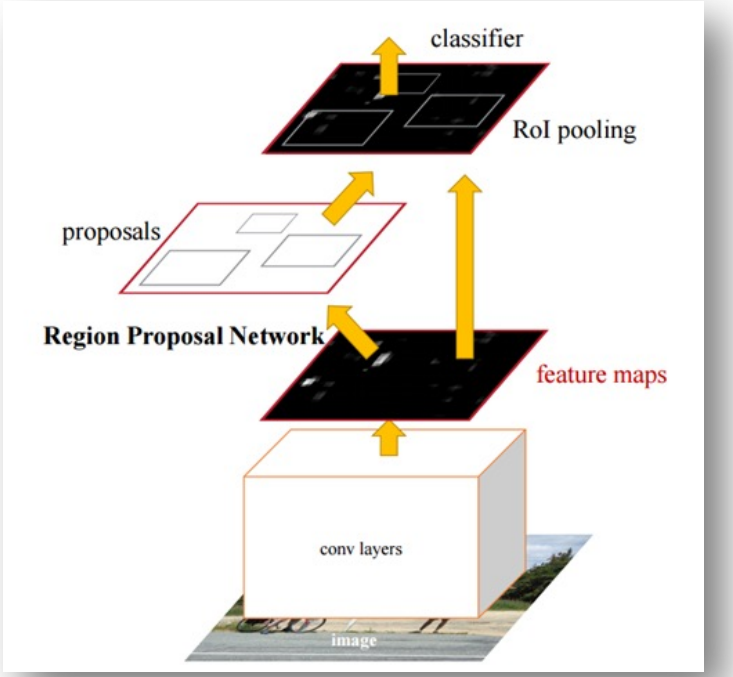


Summary

Additional Reading:

The 9 Deep Learning Papers You Need To Know About (Understanding CNNs Part 3)

Adit Deshpande (2016)



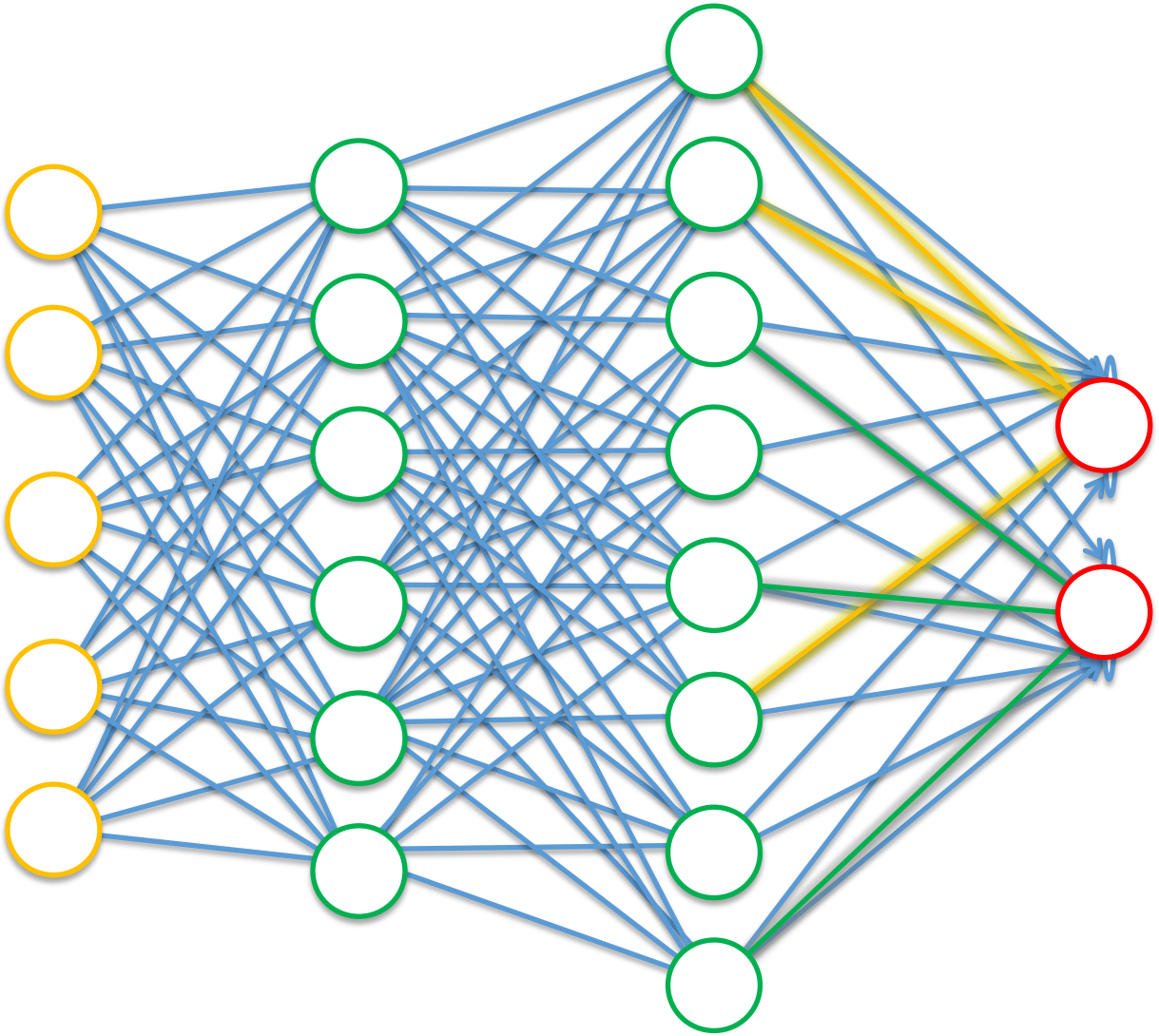
Link:

https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html

Softmax & Cross-Entropy

Softmax & Cross-Entropy

Flattening

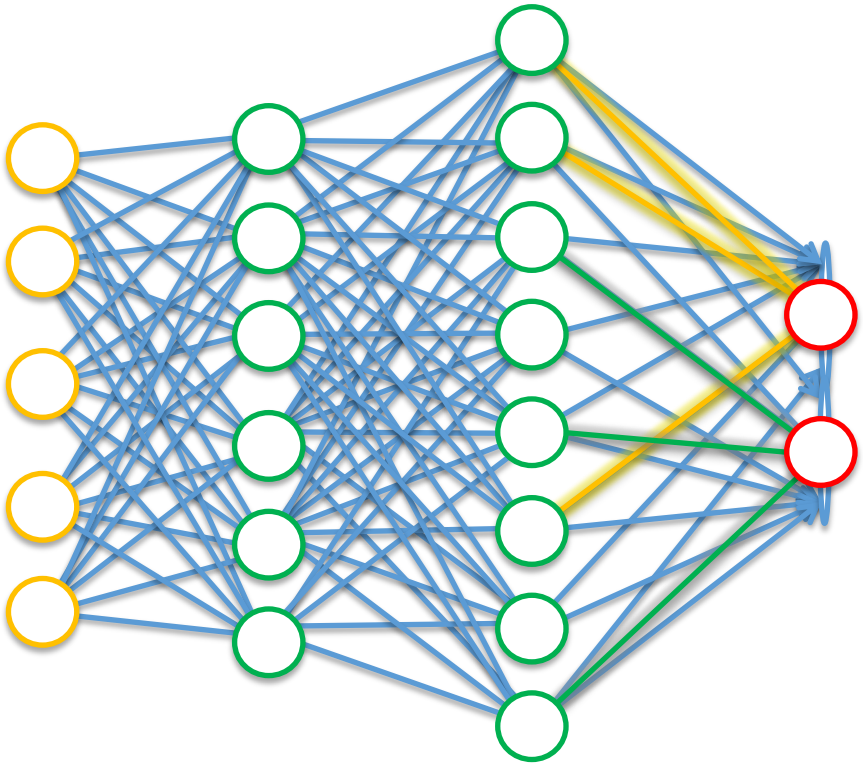


Dog 0.95

Cat 0.05

Softmax & Cross-Entropy

Flattening



$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

Dog → z_1 → 0.95

Cat → z_2 → 0.05

Softmax & Cross-Entropy

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

$$H(p,q) = -\sum_x p(x)\ \log q(x)$$

Softmax & Cross-Entropy

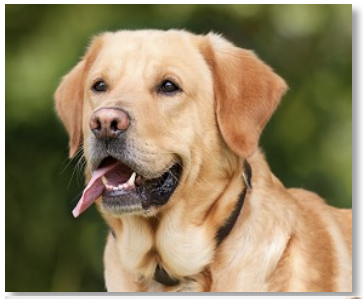
Dog 0.9

Cat 0.1

$$H(p,q) = -\sum_{x} p(x)\ \log q(x)$$

1
0

Softmax & Cross-Entropy

		NN1	NN2
	Dog	1	0.9	0.6
	Cat	0	0.1	0.4
	Dog	0	0.1	0.3
	Cat	1	0.9	0.7
	Dog	1	0.4	0.1
	Cat	0	0.6	0.9

Softmax & Cross-Entropy

NN1

Row	Dog ^	Cat^	Dog	Cat
#1	0.9	0.1	1	0
#2	0.1	0.9	0	1
#3	0.4	0.6	1	0

Classification Error

1/3 = 0.33

Mean Squared Error

0.25

Cross-Entropy

0.38

NN2

Row	Dog ^	Cat^	Dog	Cat
#1	0.6	0.4	1	0
#2	0.3	0.7	0	1
#3	0.1	0.9	1	0

1/3 = 0.33

0.71

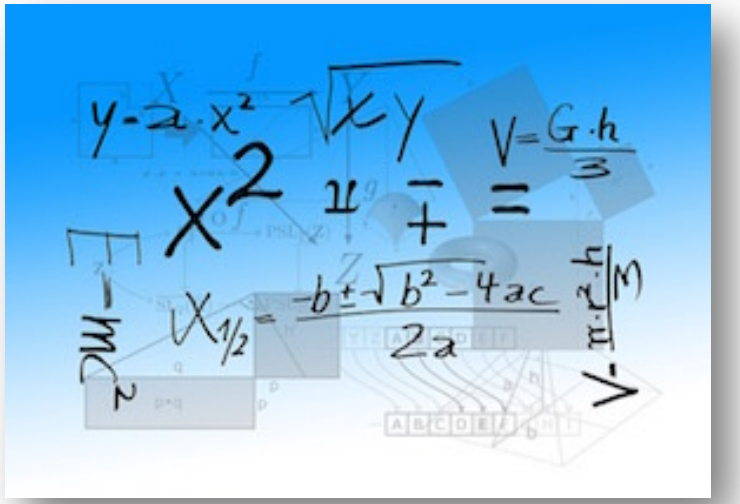
1.06

Softmax & Cross-Entropy

Additional Reading:

A Friendly Introduction to Cross-Entropy Loss

By Rob DiPietro (2016)



Link:

https://rdipietro.github.io/friendly-intro-to-cross-entropy-loss/

Softmax & Cross-Entropy

Additional Reading:

How to implement a neural network Intermezzo 2

By Peter Roelants (2016)

$$\begin{aligned}
\frac{\partial \xi}{\partial z_i} &= -\sum_{j=1}^{C} \frac{\partial t_j log(y_j)}{\partial z_i} = -\sum_{j=1}^{C} t_j \frac{\partial log(y_j)}{\partial z_i} = -\sum_{j=1}^{C} t_j \frac{1}{y_j} \frac{\partial y_j}{\partial z_i} \\
&= -\frac{t_i}{y_i} \frac{\partial y_i}{\partial z_i} - \sum_{j \neq i}^{C} \frac{t_j}{y_j} \frac{\partial y_j}{\partial z_i} = -\frac{t_i}{y_i} y_i (1 - y_i) - \sum_{j \neq i}^{C} \frac{t_j}{y_j} (-y_j y_i) \\
&= -t_i + t_i y_i + \sum_{j \neq i}^{C} t_j y_i = -t_i + \sum_{j=1}^{C} t_j y_i = -t_i + y_i \sum_{j=1}^{C} t_j \\
&= y_i - t_i
\end{aligned}$$

Link:

http://peterroelants.github.io/posts/neural_network_implementation_intermezzo02/