CONVOLUTIONAL NEURAL NETWORK

CONVOLUTIONAL NEURAL NETWORK

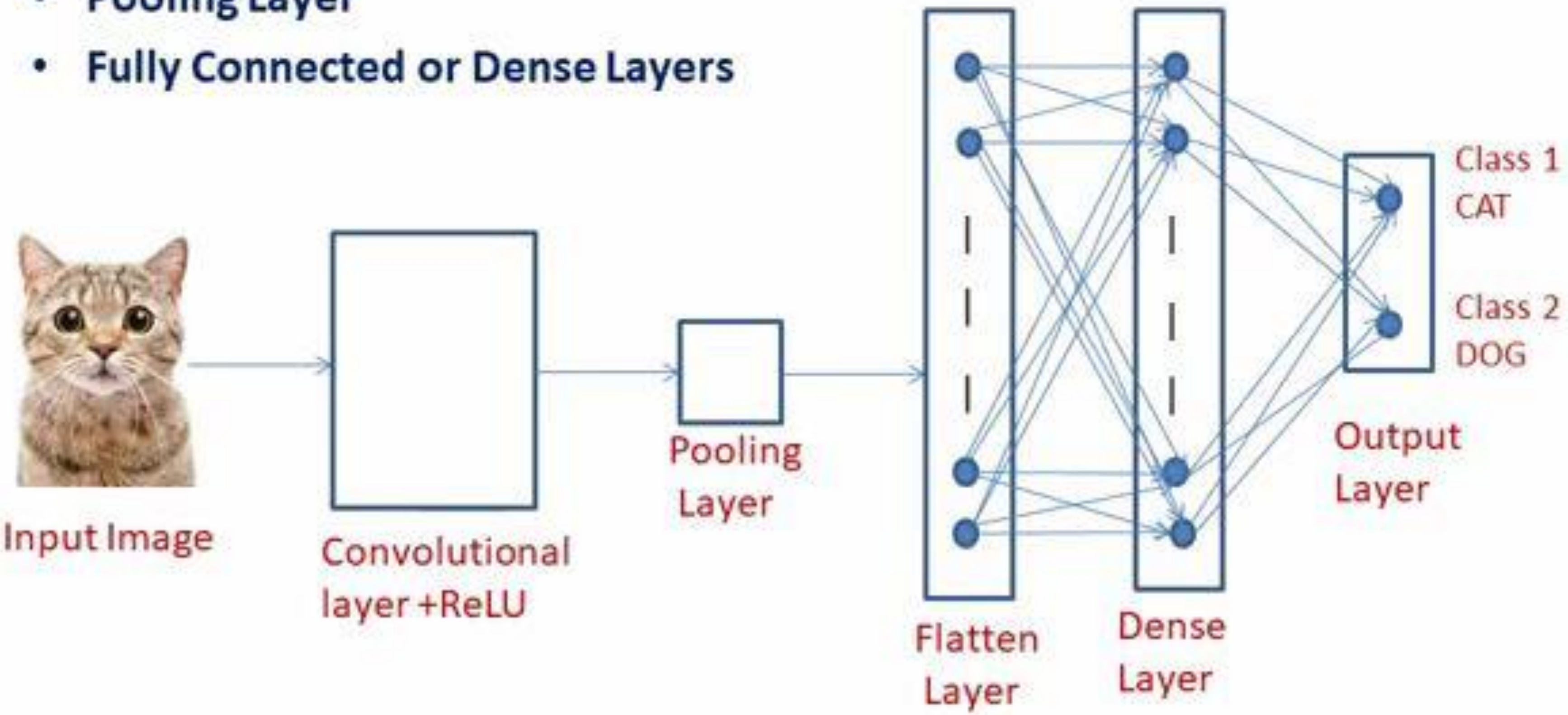
- Convolutional Neural Networks (CNNs) are a category of Neural Network that have proven very effective in areas such as image recognition and classification.

- CNNs have been successful in identifying faces, objects and traffic signs apart from powering vision in robots and self driving cars.

- CNNs are also used in Smart Grid applications as well.

CNN ARCHITECTURE

- CNN has three types of layers to build architectures apart from input layer:
  - **Convolutional Layer (with ReLU activation)**
  - **Pooling Layer**
  - **Fully Connected or Dense Layers**



CONVOLUTIONAL LAYER

- **Convolutional Layer**: The primary purpose of Convolution is to extract features from the input image. Convolution preserves the spatial relationship between input by learning input features.

Convolution layer output y = $\sum_{i}^{n} x \ * f$

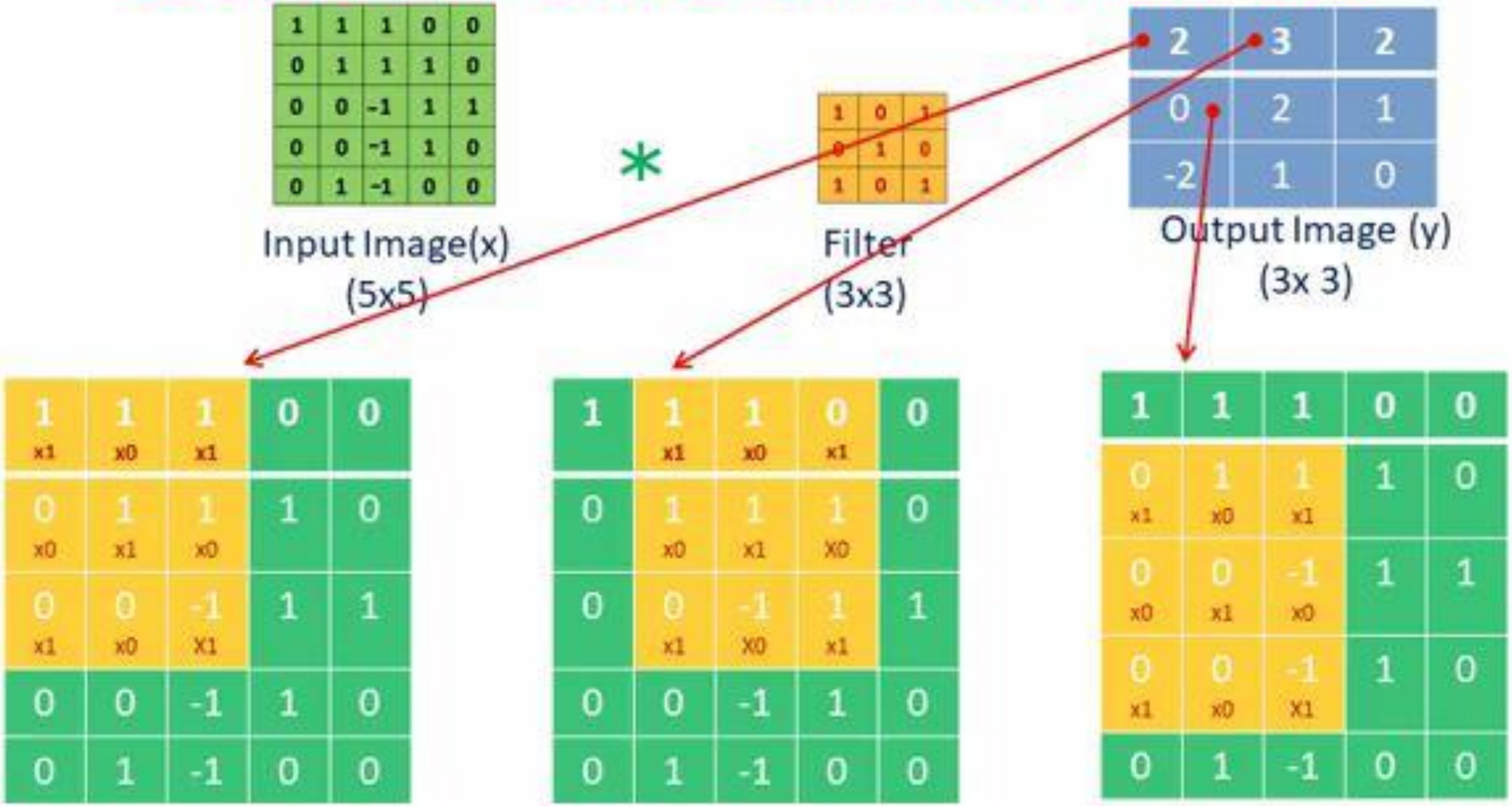
$$\begin{array}{|c|c|c|c|c|}\hline 1&1&1&0&0\\\hline 0&1&1&1&0\\\hline 0&0&-1&1&1\\\hline 0&0&-1&1&0\\\hline 0&1&-1&0&0\\\hline\end{array} \quad * \quad \begin{array}{|c|c|c|}\hline 1&0&1\\\hline 0&1&0\\\hline 1&0&1\\\hline\end{array} \quad = \quad \begin{array}{|c|c|c|}\hline 2&3&2\\\hline 0&2&1\\\hline -2&1&0\\\hline\end{array}$$

Input Image(x)
(nxn)

Filter
(fxf)

Output Image (y)
(n-f+1) x (n-f+1)

CONVOLUTIONAL LAYER OPERATION



1	1	1	0	0
0	1	1	1	0
0	0	-1	1	1
0	0	-1	1	0
0	1	-1	0	0

Input Image(x)
(5x5)

1	0	1
0	1	0
1	0	1

Filter
(3x3)

2	3	2
0	2	1
-2	1	0

Output Image (y)
(3x 3)

1 x1	1 x0	1 x1	0	0
0 x0	1 x1	1 x0	1	0
0 x1	0 x0	-1 X1	1	1
0	0	-1	1	0
0	1	-1	0	0

y11 = (1x1) + (1x0)
+ (1x1) + (0x0)
+ (1x1) + (1x0)
+ (0x1) + (0x0)
+ (-1x1) = 2

1	1 x1	1 x0	0 x1	0
0	1 x0	1 x1	1 X0	0
0	0 x1	-1 X0	1 x1	1
0	0	-1	1	0
0	1	-1	0	0

y12 = (1x0) + (1x1)
+ (0x1) + (1x0)
+ (1x1) + (1x0)
+ (0x1) + (-1x0)
+ (1x1) = 3

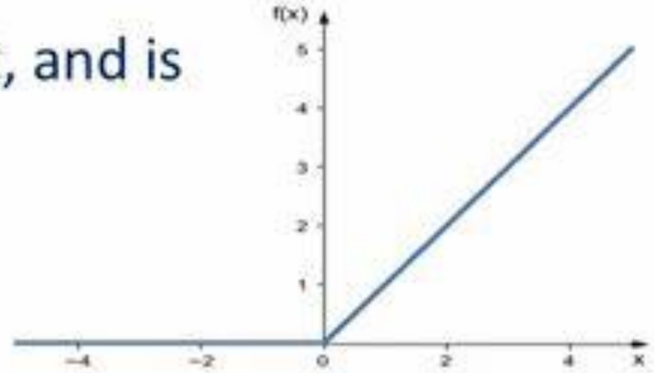
1	1	1	0	0
0 x1	1 x0	1 x1	1	0
0 x0	0 x1	-1 x0	1	1
0 x1	0 x0	-1 X1	1	0
0	1	-1	0	0

y21 = (0x1) + (1x0)
+ (1x1) + (0x0)
+ (0x1) + (-1x0)
+ (0x0) + 0x0)
+ (-1x1) = 0

ReLU Layer

ReLU stands for rectified linear unit, and is a type of **activation function**.

f(x) $= \max(0, x)$



2	3	2
0	2	1
-2	1	0

Input of ReLU

→

2	3	2
0	2	1
0	1	0

Output of ReLU

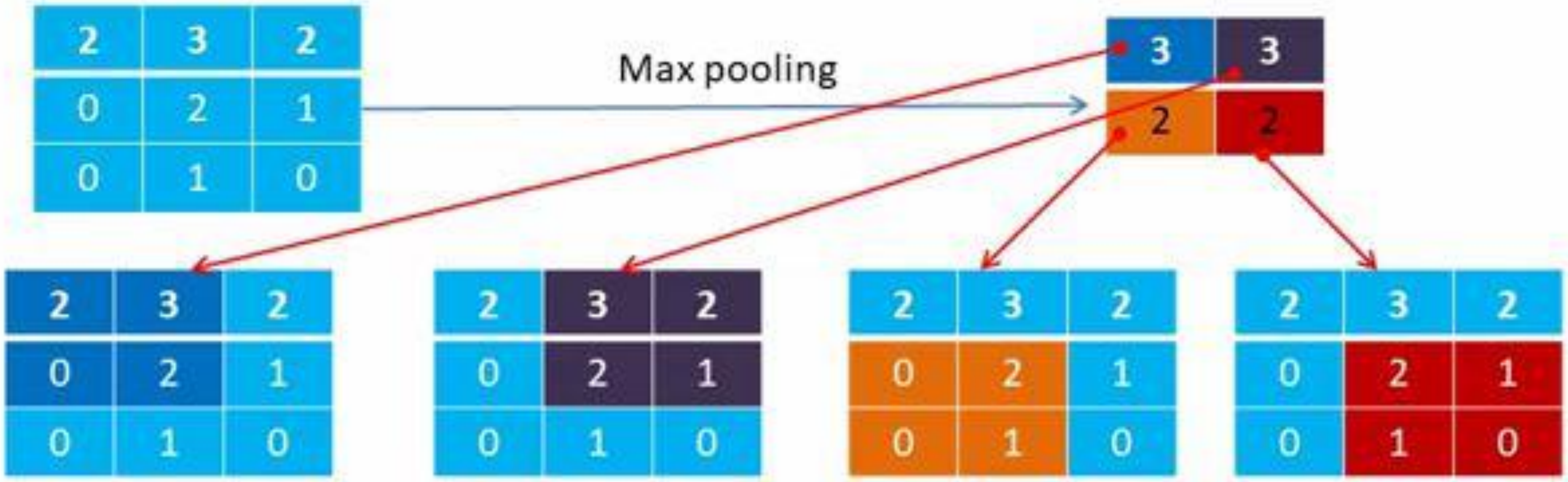
Pooling layers

- Pooling layers would reduce the number of parameters when the inputs are too large.
- Pooling also called down sampling which reduces the dimensionality of each map but retains important information.
- There are three types of pooling namely, Max Pooling, Average Pooling, Sum Pooling.

Max Pooling:

(nxn) (f x f) **(n-f+1)x(n-f+1)**

(3x3) (2 x 2) (2 x 2)



2	3	2
0	2	1
0	1	0

3	3
2	2

2	3	2
0	2	1
0	1	0

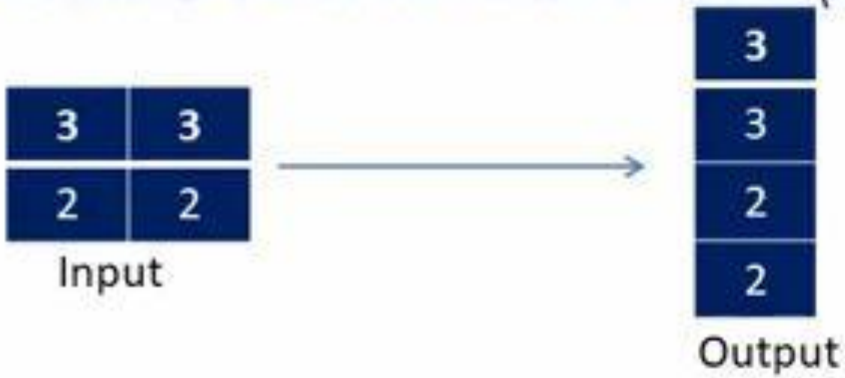
2	3	2
0	2	1
0	1	0

2	3	2
0	2	1
0	1	0

2	3	2
0	2	1
0	1	0

Flatten layer

- The feature matrix will be converted as vector (x1, x2, x3, …).



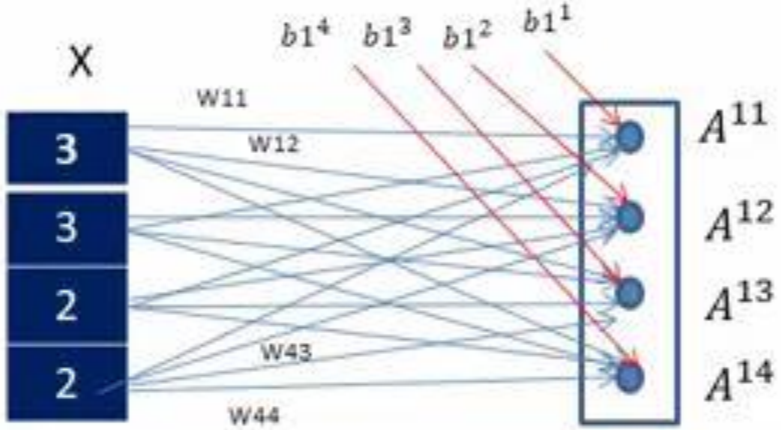
3	3
2	2

Input

3
3
2
2

Output

Fully Connected layer with ReLU

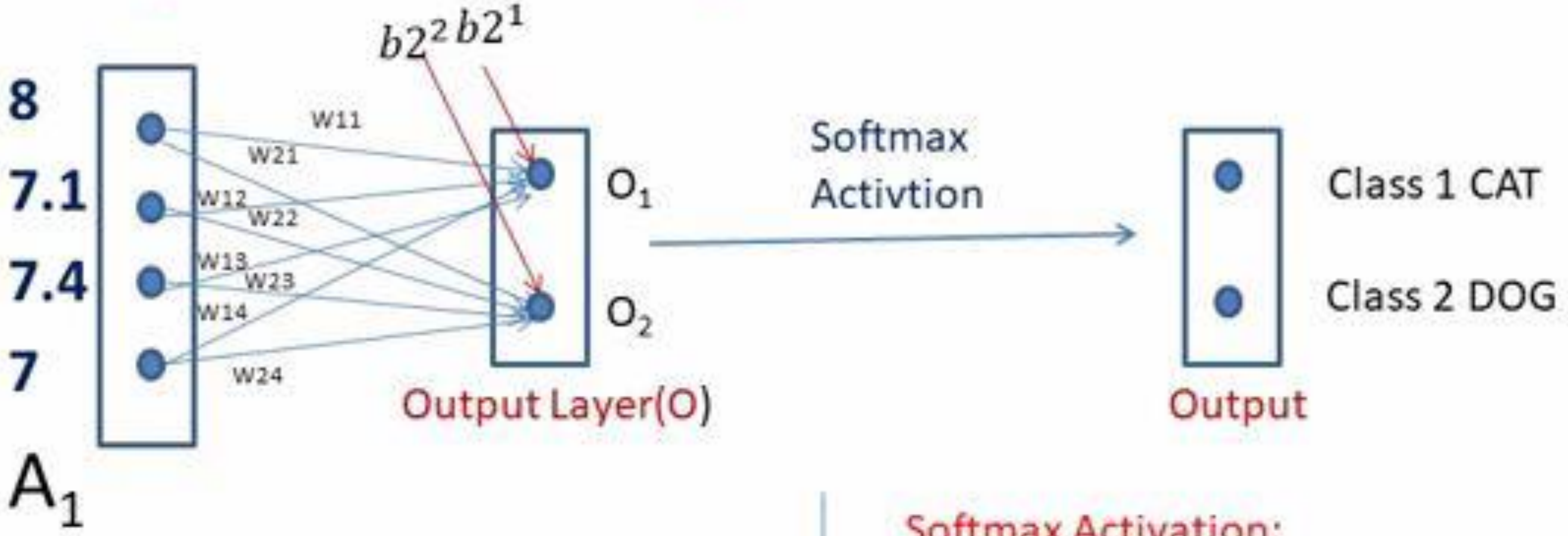


$$A_1 = W_1X + b_1$$

$$\begin{bmatrix}A11\\A12\\A13\\A14\end{bmatrix} = \begin{bmatrix}1.0 & 1.0 & 0.2 & 0.8\\1.0 & 0.5 & 0.5 & 0.8\\0.8 & 1.0 & 0.2 & 0.8\\0.5 & 0.5 & 1.0 & 1.0\end{bmatrix}\begin{bmatrix}3\\3\\2\\2\end{bmatrix} + \begin{bmatrix}0.1\\0.1\\0.1\\0.1\end{bmatrix} = \begin{bmatrix}8\\7.1\\7.4\\7\end{bmatrix}$$

$$\text{ReLU}\begin{bmatrix}8\\7.1\\7.4\\7\end{bmatrix} = \begin{bmatrix}8\\7.1\\7.4\\7\end{bmatrix}$$

Fully Connected layer & Softmax Activation



$$O = W_2 A_1 + b_2$$

$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} 0.2 & 0.1 & 0.5 & 0.5 \\ 0.1 & 0.5 & 0.1 & 0.2 \end{bmatrix} \begin{bmatrix} 8 \\ 7.1 \\ 7.4 \\ 7 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

$$= \begin{bmatrix} 9.61 \\ 6.49 \end{bmatrix}$$

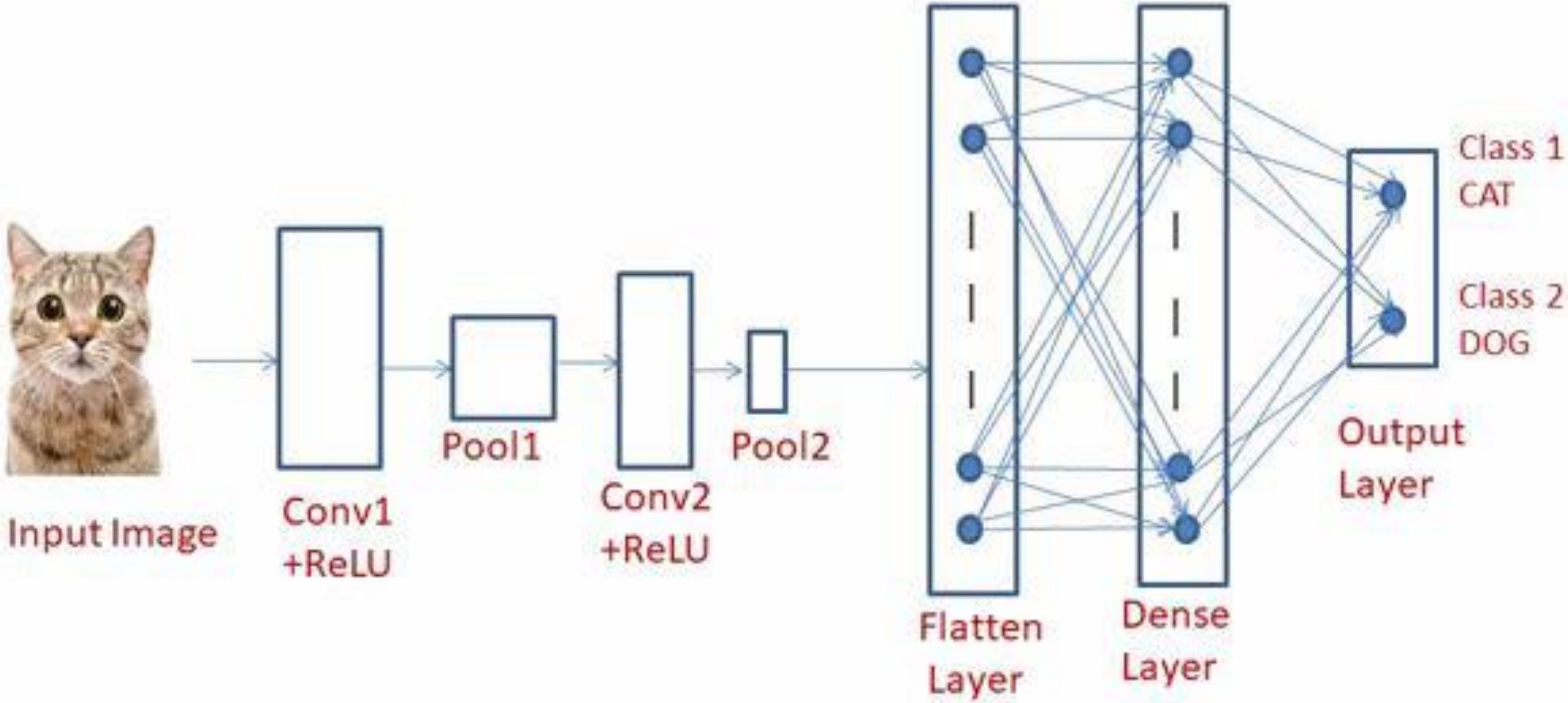
Softmax Activation:

$$\sigma(O_i) = \frac{e^{o_i}}{\sum e^{o_i}}$$

$$\sigma(O1) = \frac{e^{o1}}{e^{o1} + e^{o2}} = \frac{14913.17}{14913.17 + 658.52} = 0.95$$

$$\sigma(O2) = \frac{e^{o2}}{e^{o1} + e^{o2}} = \frac{658.52}{14913.17 + 658.52} = 0.04$$

STACKED CNN ARCHITECTURE


Input Image
Conv1
+ReLU
Pool1
Conv2
+ReLU
Pool2
Flatten
Layer
Dense
Layer
Output
Layer
Class 1
CAT
Class 2
DOG