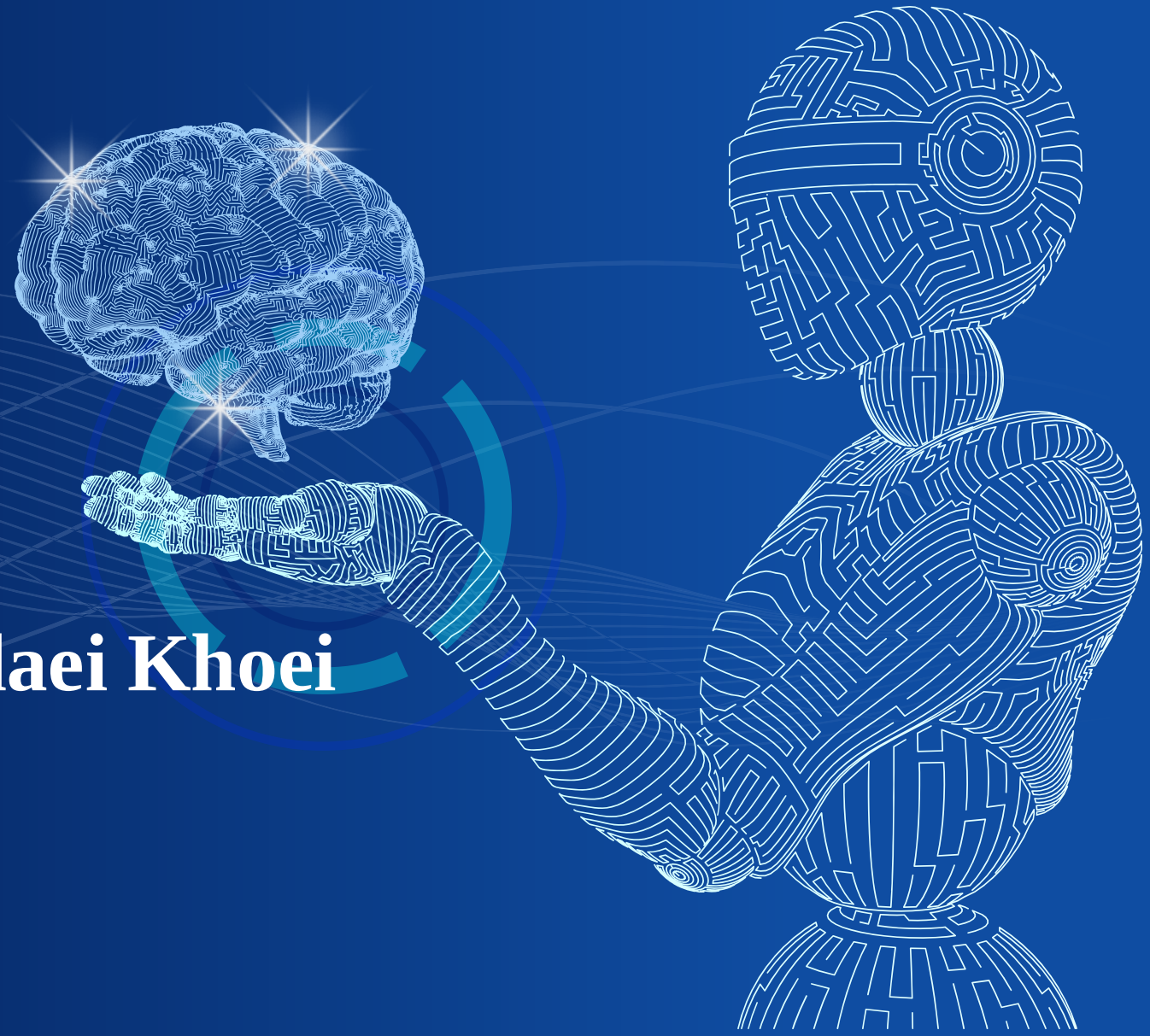
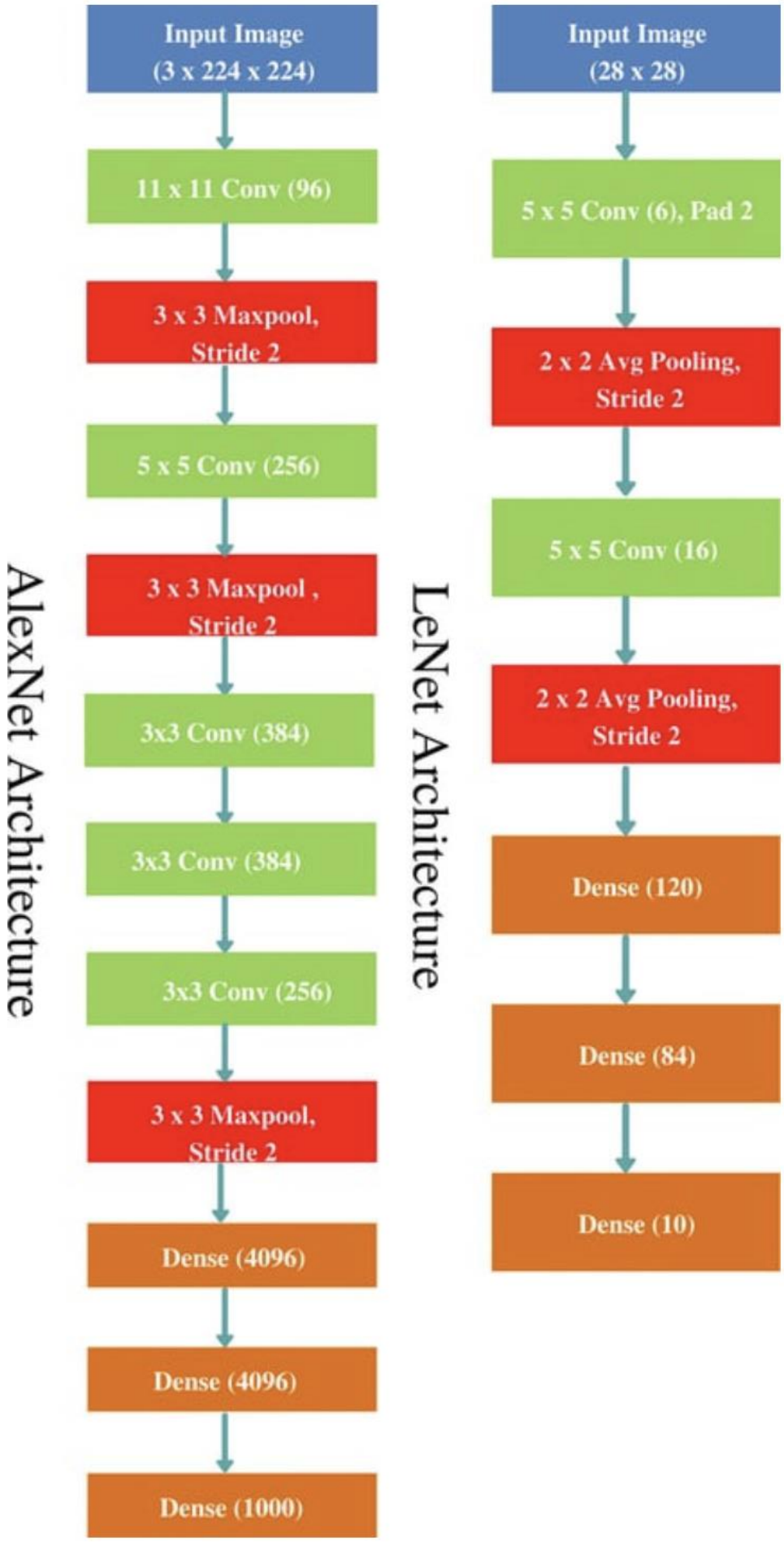
CS 7150: Deep Learning



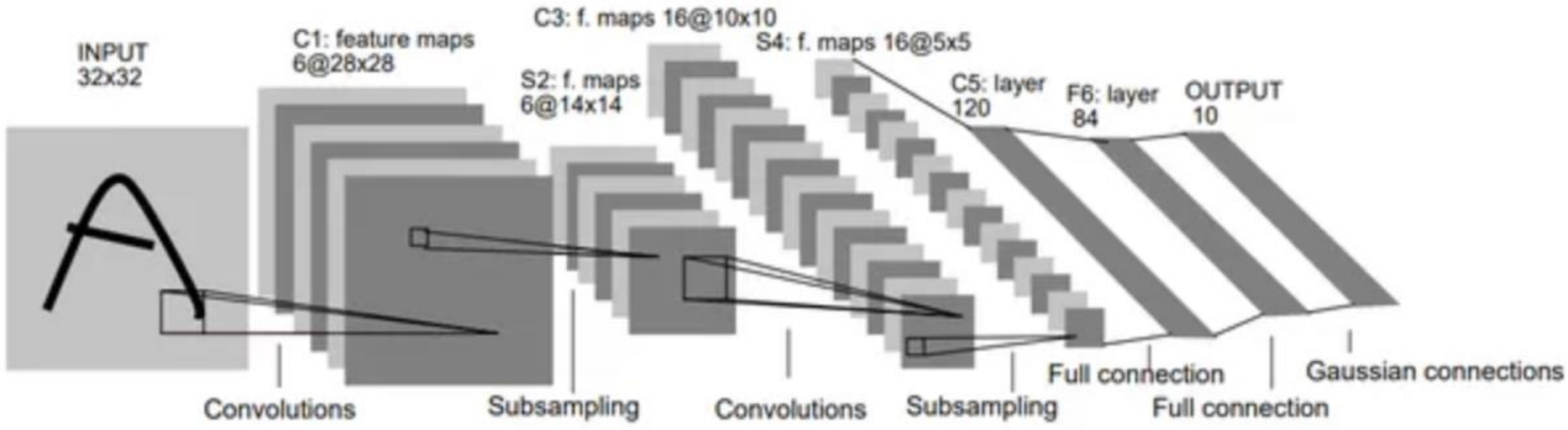
Presented by: Dr. Tala Talaei Khoei


AlexNet Architecture
Input Image
(3 x 224 x 224)
11 x 11 Conv (96)
3 x 3 Maxpool,
Stride 2
5 x 5 Conv (256)
3 x 3 Maxpool ,
Stride 2
3x3 Conv (384)
3x3 Conv (384)
3x3 Conv (256)
3 x 3 Maxpool,
Stride 2
Dense (4096)
Dense (4096)
Dense (1000)
LeNet Architecture
Input Image
(28 x 28)
5 x 5 Conv (6), Pad 2
2 x 2 Avg Pooling,
Stride 2
5 x 5 Conv (16)
2 x 2 Avg Pooling,
Stride 2
Dense (120)
Dense (84)
Dense (10)

Introduction to LeNet-5

- LeNet was introduced in the research paper "Gradient-Based Learning Applied To Document Recognition" in the year 1998 by Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Many of the listed authors of the paper have gone on to provide several significant academic contributions to the field of deep learning.

- LeNet-5 CNN architecture is made up of 7 layers. The layer composition consists of 3 convolutional layers, 2 subsampling layers and 2 fully connected layers.

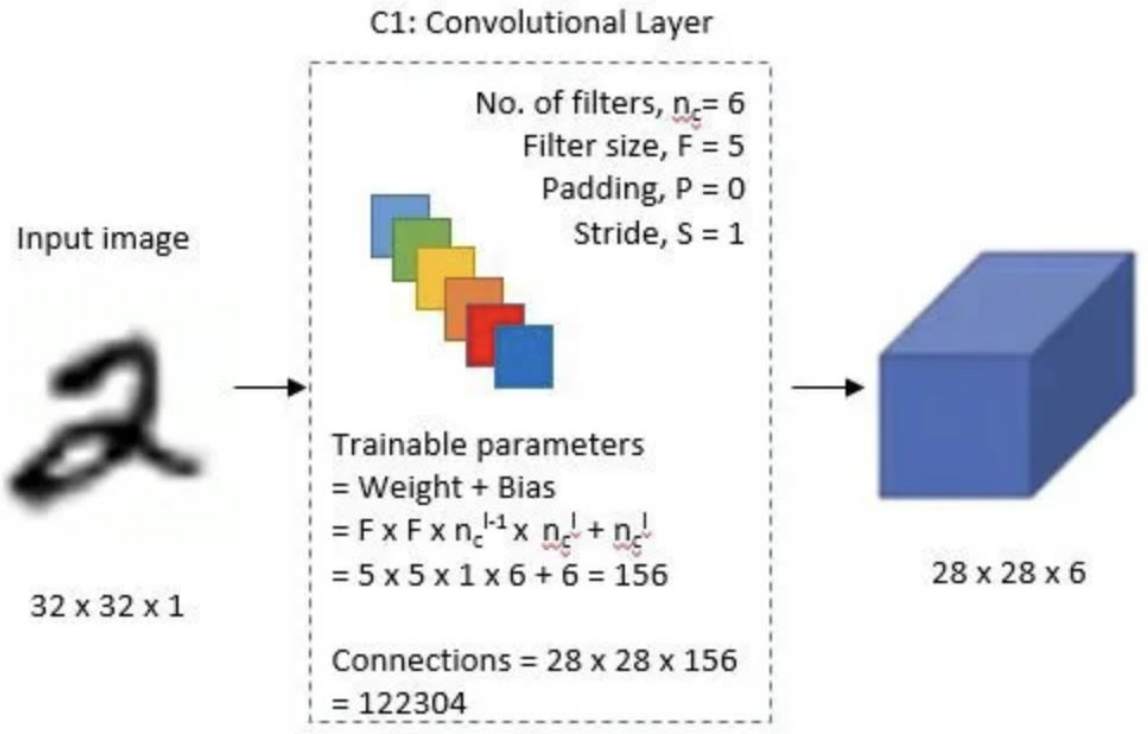


Features of LeNet-5

- Every convolutional layer includes three parts: convolution, pooling, and nonlinear activation functions
- Using convolution to extract spatial features (Convolution was called receptive fields originally)
- The average pooling layer is used for subsampling.
- ‘tanh’ is used as the activation function
- Using Multi-Layered Perceptron or Fully Connected Layers as the last classifier
- The sparse connection between layers reduces the complexity of computation

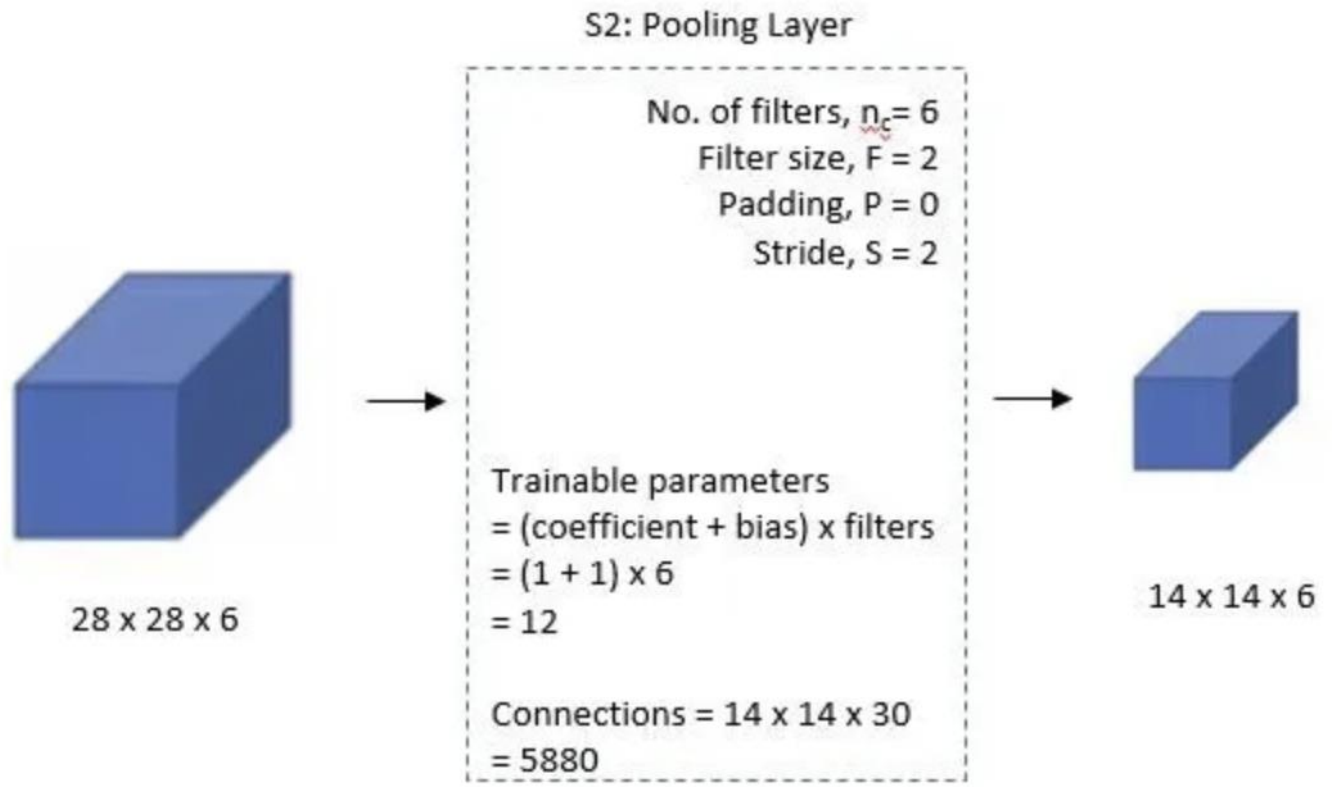
Architecture: First Layer

- A 32x32 grayscale image serves as the input for LeNet-5 and is processed by the first convolutional layer comprising six feature maps or filters with a stride of one. From 32x32x1 to 28x28x6, the image's dimensions shift.



Architecture: Second Layer

- Then, using a filter size of 22 and a stride of 2, the LeNet-5 adds an average pooling layer or sub-sampling layer. 14x14x6 will be the final image's reduced size.



Architecture: Third Layer

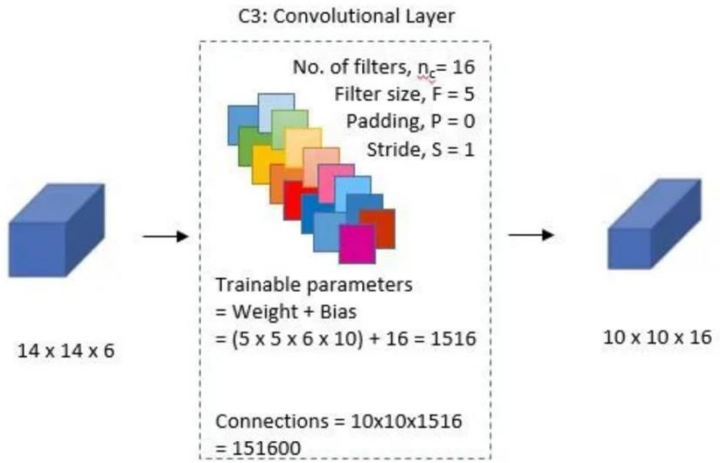
- A second convolutional layer with 16 feature maps of size 55 and a stride of 1 is then present. Only 10 of the 16 feature maps in this layer are linked to the six feature maps in the layer below, as can be seen in the illustration below.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

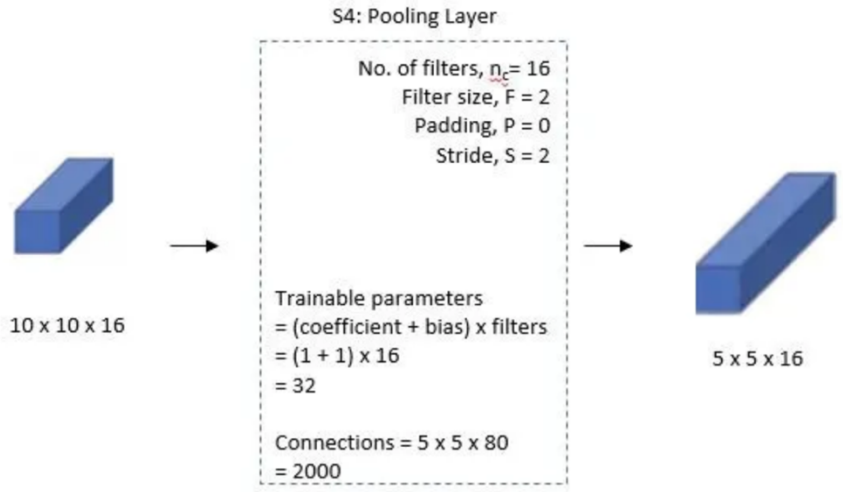
EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

- The primary goal is to disrupt the network's symmetry while maintaining a manageable number of connections. Because of this, there are 1516 training parameters instead of 2400 in these layers, and similarly, there are 151600 connections instead of 240000.

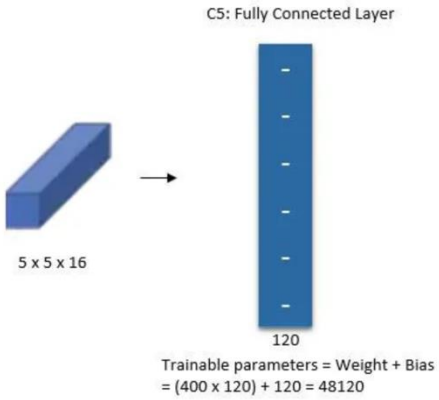


Architecture: Fourth Layer

- With a filter size of 22 and a stride of 2, the fourth layer (S4) is once more an average pooling layer. The output will be decreased to 5x5x16 because this layer is identical to the second layer (S2) but has 16 feature maps.

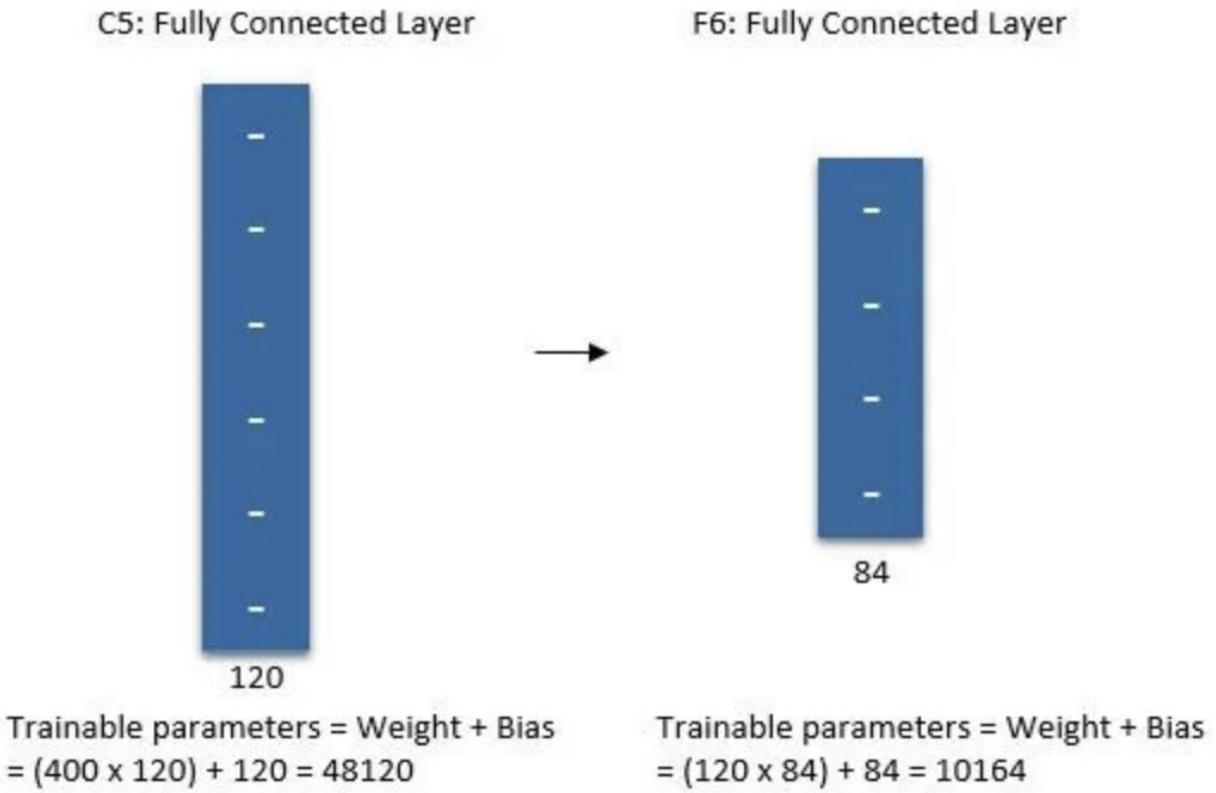


- With 120 feature maps, each measuring 1 x 1, the fifth layer (C5) is a fully connected convolutional layer. All 400 nodes (5x5x16) in layer four, S4, are connected to each of the 120 units in C5's 120 units.



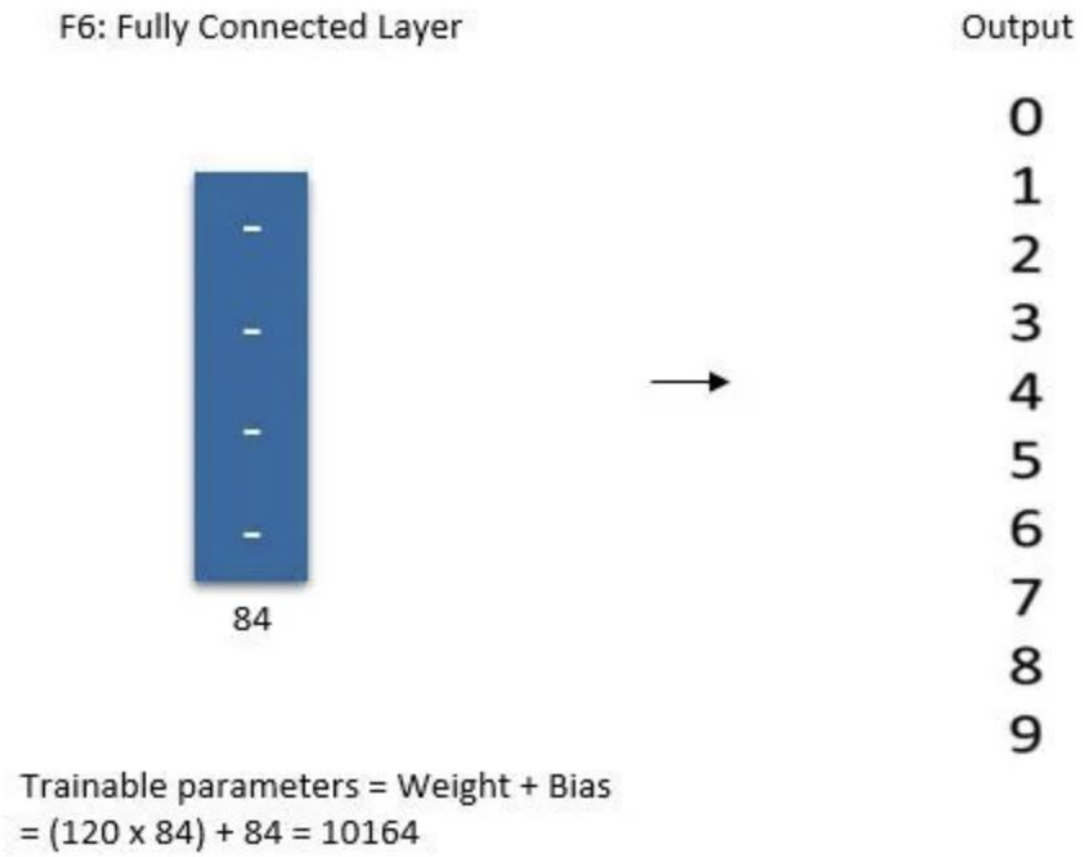
Architecture: Fifth Layer

- A fully connected layer (F6) with 84 units makes up the sixth layer.



Output Layer

The SoftMax output layer, which has 10 potential values and corresponds to the digits 0 to 9, is the last layer.



Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

Please refer to Codes on Canvas

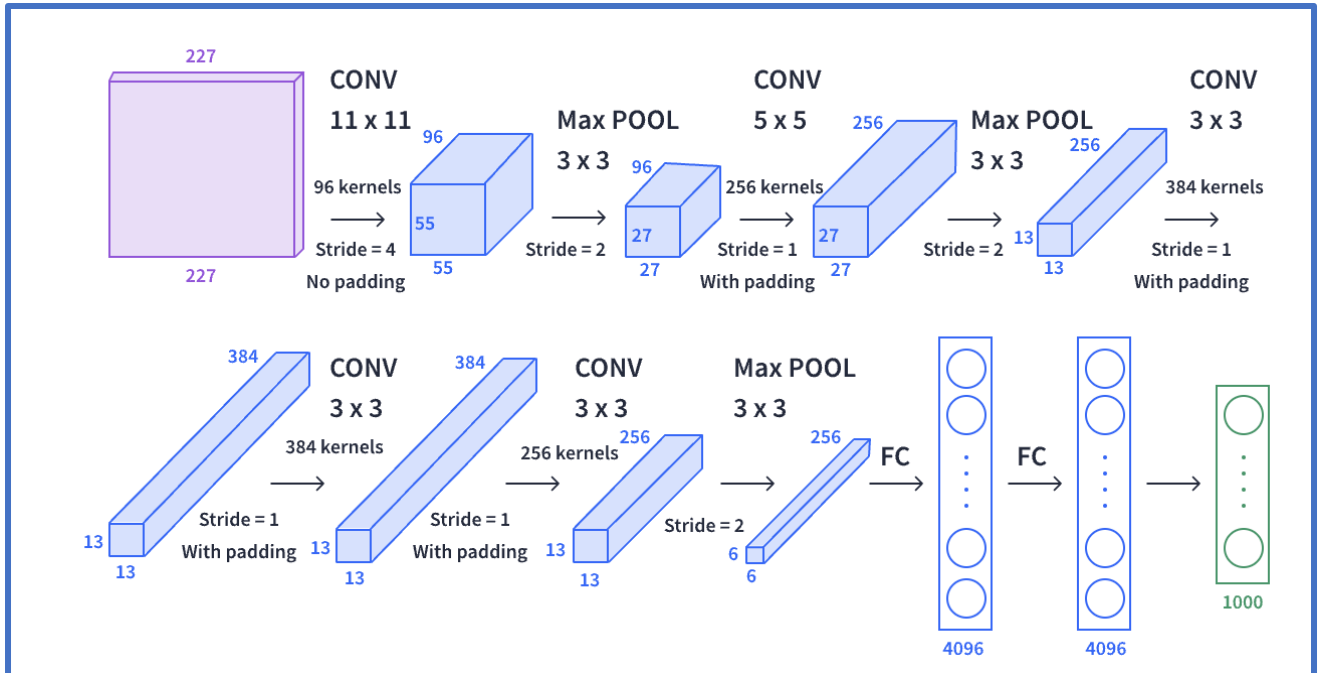
Introduction to Alex Neural Networks

- AlexNet won the Imagenet large-scale visual recognition challenge in 2012. The model was proposed in 2012 in the research paper named Imagenet Classification with Deep Convolution Neural Network by Alex Krizhevsky and his colleagues.

- In this model, the depth of the network was increased in comparison to Lenet-5.

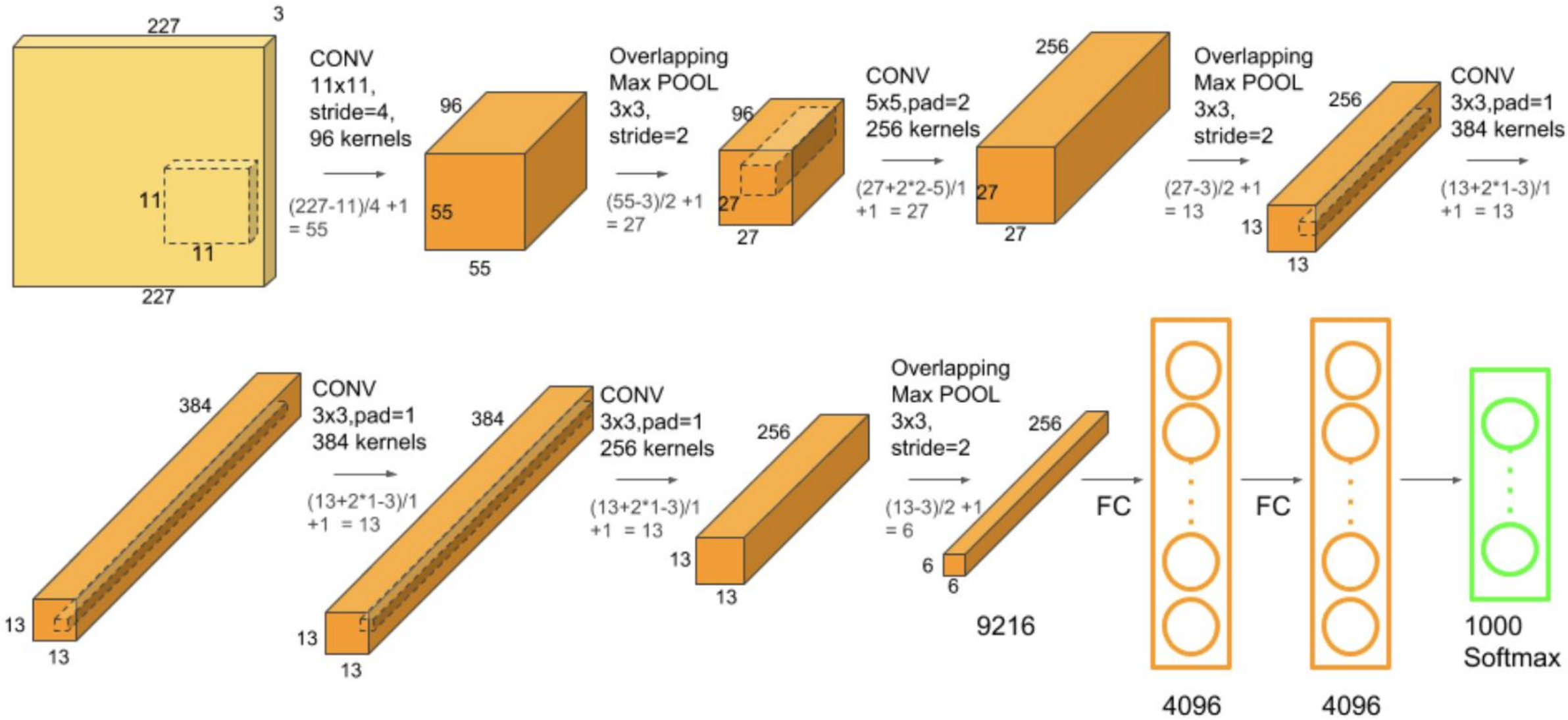
- The AlexNet has eight layers with learnable parameters. The model consists of five layers with a combination of max pooling followed by 3 fully connected layers, and they use Relu activation in each of these layers except the output layer.

- They found out that using the relu as an activation function accelerated the speed of the training process by almost six times. They also used the dropout layers, that prevented their model from overfitting.



Architecture of AlexNet

- The AlexNet architecture consists of eight layers in total.
- The first five layers are convolutional layers.
- The sizes of the convolutional filters are 11×11, 5×5, 3×3, 3×3, and 3×3 for the respective convolutional layers.
- Some of the convolutional layers are followed by max-pooling layers, which help reduce spatial dimensions while retaining important features.
- The activation function used in the network is the Rectified Linear Unit (ReLU), known for its superior performance compared to sigmoid and tanh functions.
- After the convolutional layers, there are three fully connected layers.
- The network's parameters can be tuned based on the training performance.
- The AlexNet can be used with transfer learning, utilizing pre-trained weights on the ImageNet dataset to achieve exceptional performance. However, in this article, we will define the CNN without using pre-trained weights, following the proposed architecture.


227
3
227
11
11
CONV
11x11,
stride=4,
96 kernels
(227-11)/4 +1
= 55
96
55
55
Overlapping
Max POOL
3x3,
stride=2
(55-3)/2 +1
= 27
96
27
27
CONV
5x5,pad=2
256 kernels
(27+2*2-5)/1
+1 = 27
256
27
27
Overlapping
Max POOL
3x3,
stride=2
(27-3)/2 +1
= 13
256
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
256 kernels
(13+2*1-3)/1
+1 = 13
256
13
13
Overlapping
Max POOL
3x3,
stride=2
(13-3)/2 +1
= 6
256
6
6
9216
FC
4096
FC
4096
1000
Softmax

The Network Structure of AlexNet

Size / Operation	Filter	Depth	Stride	Padding	Number of Parameters	Forward Computation
3* 227 * 227						
Conv1 + Relu	11 * 11	96	4		(11*11*3 + 1) * 96=34944	(11*11*3 + 1) * 96 * 55 * 55=105705600
96 * 55 * 55						
Max Pooling	3 * 3		2			
96 * 27 * 27						
Norm						
Conv2 + Relu	5 * 5	256	1	2	(5 * 5 * 96 + 1) * 256=614656	(5 * 5 * 96 + 1) * 256 * 27 * 27=448084224
256 * 27 * 27						
Max Pooling	3 * 3		2			
256 * 13 * 13						
Norm						
Conv3 + Relu	3 * 3	384	1	1	(3 * 3 * 256 + 1) * 384=885120	(3 * 3 * 256 + 1) * 384 * 13 * 13=149585280
384 * 13 * 13						
Conv4 + Relu	3 * 3	384	1	1	(3 * 3 * 384 + 1) * 384=1327488	(3 * 3 * 384 + 1) * 384 * 13 * 13=224345472
384 * 13 * 13						
Conv5 + Relu	3 * 3	256	1	1	(3 * 3 * 384 + 1) * 256=884992	(3 * 3 * 384 + 1) * 256 * 13 * 13=149563648
256 * 13 * 13						
Max Pooling	3 * 3		2			
256 * 6 * 6						
Dropout (rate 0.5)						
FC6 + Relu					256 * 6 * 6 * 4096=37748736	256 * 6 * 6 * 4096=37748736
4096						
Dropout (rate 0.5)						
FC7 + Relu					4096 * 4096=16777216	4096 * 4096=16777216
4096						
FC8 + Relu					4096 * 1000=4096000	4096 * 1000=4096000
1000 classes						
Overall					62369152=62.3 million	1135906176=1.1 billion
Conv VS FC					Conv:3.7million (6%) , FC: 58.6 million (94%)	Conv: 1.08 billion (95%) , FC: 58.6 million (5%)

What is AlexNet used for?

AlexNet is primarily used for image recognition tasks in the field of computer vision. It has been applied to various domains, including autonomous vehicles, robotics, and medical imaging. Its deep architecture allows the network to learn complex features and representations from large-scale image datasets, making it suitable for a wide range of applications.

Why AlexNet is better than CNN?

AlexNet is a specific type of convolutional neural network (CNN) that introduced several innovations that improved its performance compared to traditional CNNs. These innovations include the use of rectified linear units (ReLU) as activation functions, dropout layers for regularization, and the use of graphics processing units (GPUs) for parallel computation. These advancements allowed AlexNet to achieve higher accuracy in image classification tasks compared to previous CNN models.

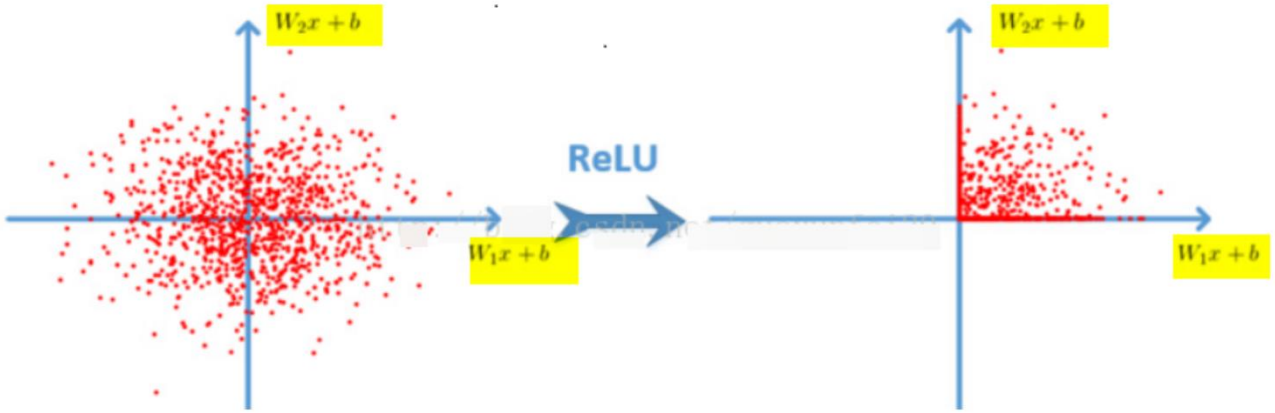
Can AlexNet be used for other tasks besides image recognition?

While AlexNet was originally designed for image recognition tasks, its deep learning architecture can be adapted for other tasks as well. Transfer learning techniques can be applied to fine-tune the pre-trained AlexNet model for tasks like object detection, segmentation, and even non-vision tasks such as natural language processing or speech recognition. However, it is essential to consider the specific requirements and constraints of the target task when adapting AlexNet for different applications.

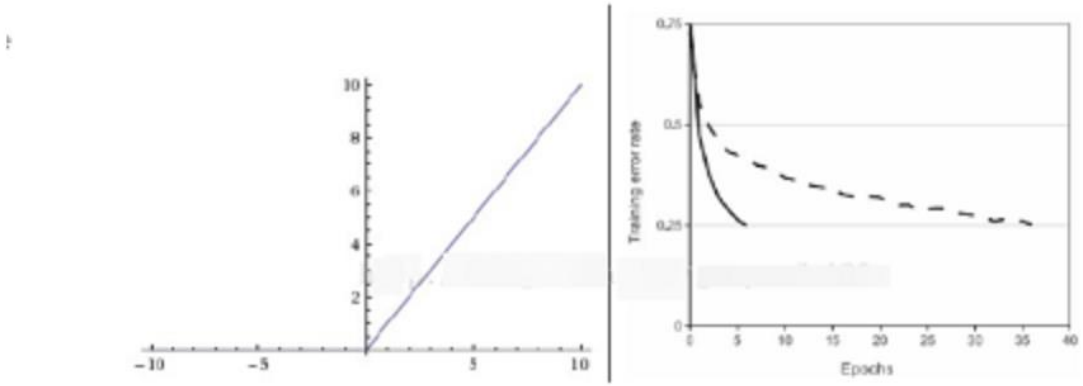
Why does AlexNet achieve better results?

Relu activation function is used:

- Relu function: f (x) = max (0, x)



- ReLU-based deep convolutional networks are trained several times faster than tanh and sigmoid- based networks. The following figure shows the number of iterations for a four-layer convolutional network based on CIFAR-10 that reached 25% training error in tanh and ReLU:



Left: Rectified Linear Unit (ReLU) activation function, which is zero when x < 0 and then linear with slope 1 when x > 0. **Right:** A plot from Krizhevsky et al. (pdf) paper indicating the 6x improvement in convergence with the ReLU unit compared to the tanh unit.

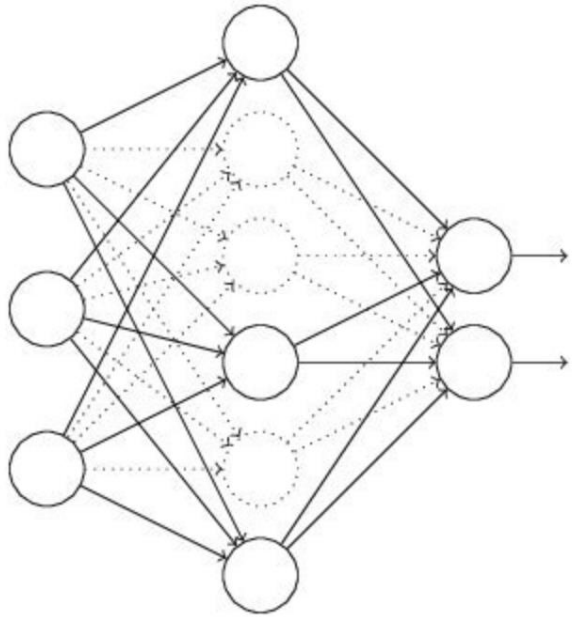
Standardization (Local Response Normalization):

After using ReLU f (x) = max (0, x), you will find that the value after the activation function has no range like the tanh and sigmoid functions, so a normalization will usually be done after ReLU. One method in neuroscience is called "Lateral inhibition", which talks about the effect of active neurons on its surrounding neurons.

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

Dropout:

Dropout is also a concept often said, which can effectively prevent overfitting of neural networks. Compared to the general linear model, a regular method is used to prevent the model from overfitting. In the neural network, Dropout is implemented by modifying the structure of the neural network itself. For a certain layer of neurons, randomly delete some neurons with a defined probability, while keeping the individuals of the input layer and output layer neurons unchanged, and then update the parameters according to the learning method of the neural network. In the next iteration, rerandom Remove some neurons until the end of training.



Please refer to Codes on Canvas


THANK YOU
Any Questions?