# Convolutional Neural Networks (CNNs)

Computer Engineering Department
Amirkabir University of Technology

Course: **Neural Networks**
Instructor: **Reza Safabakhsh**

# Convolutional Networks

- Neurobiologically motivated, work of Hubel and Wiesel (1962, 1977) on locally sensitive and orientation-selective neurons of the visual cortex of a cat.

  D. Hubel and T. Wiesel (1959, 1962, Nobel Prize 1981): Visual cortex consists of a hierarchy of *simple*, *complex*, and *hyper-complex* cells.

- A special class of multilayer perceptrons. Well suited for pattern classification.

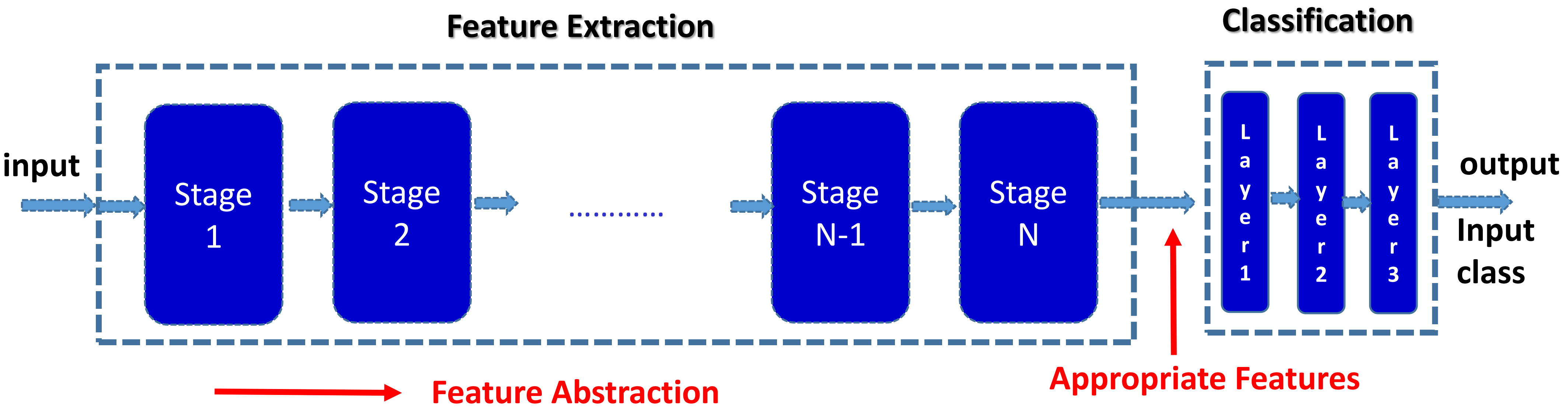
- For recognition of two-dimensional shapes with a high degree of invariance to translation, scaling, skewing, and other forms of distortion.

- Task is learned in a supervised manner.

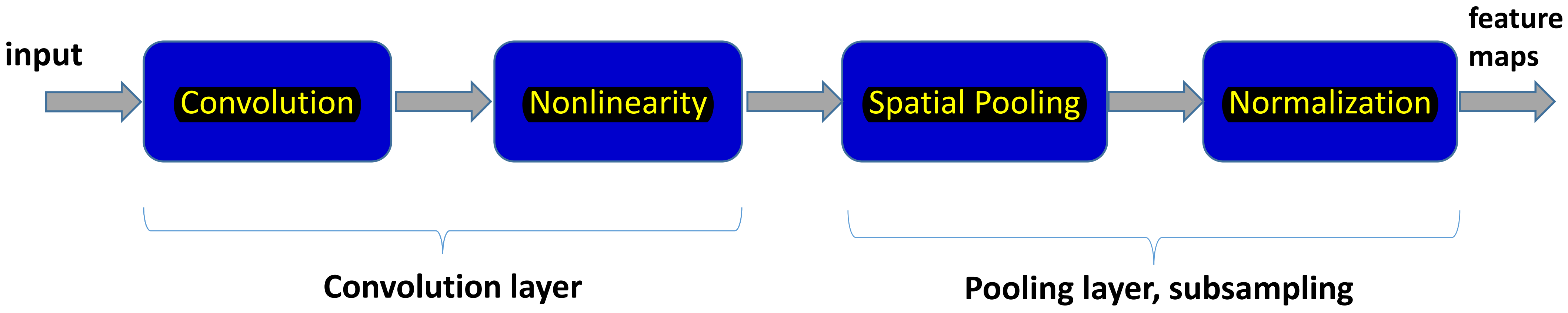
# Network Architechture:

Two main sections:

- Feature extraction: consists of N stages, perform feature abstraction
- Classification: includes a classifier, such as MLP



# Every **stage** consists of all or some of the following steps:




Stage 1: Convolution layer 1 + Pooling layer 1

Stage 2: Convolution layer 2 + Pooling layer 2

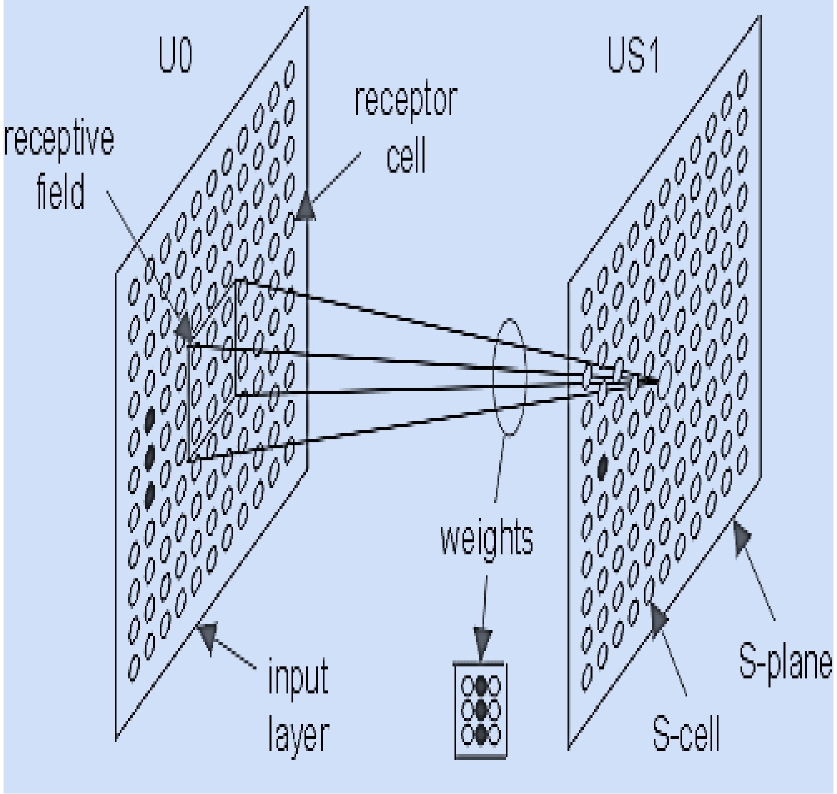
.....

Stage N Convolution layer N + Pooling layer N

# Convolution:

- Every neuron in the convolution layer calculates a convolution in a neighborhood around corresponding input (receptive field).

$$I_j^{(l)} = \sum\nolimits_{i\in recptive\ field} w_{ij}^{(l)}\, x_i^{(l-1)} + b_j^{(l)}$$




- Limited connections to previous layer.

- The same set of weights used for all neurons of this plane (weight sharing). Produces one feature plane.

- K set of weights produce K feature planes.
- Every set of weights applies a filter to the input.
- The set of weights form the filter kernel.
- 3 channels for color input images
- For *FxF* filters, stride can be 1, 2, ..., F
- No overlapping receptive fields for stride=F
- Receptive fields overlap improves the results, but larger feature maps
- Supervised training of filters by back-propagating classification error

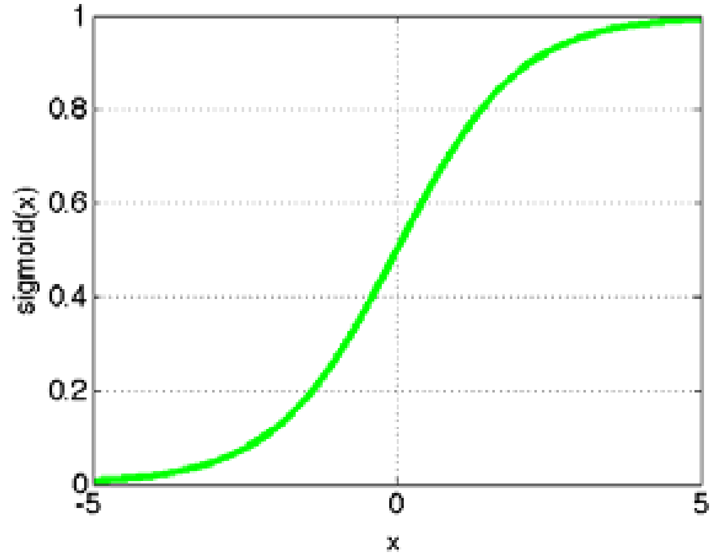
## Nonlinearity:

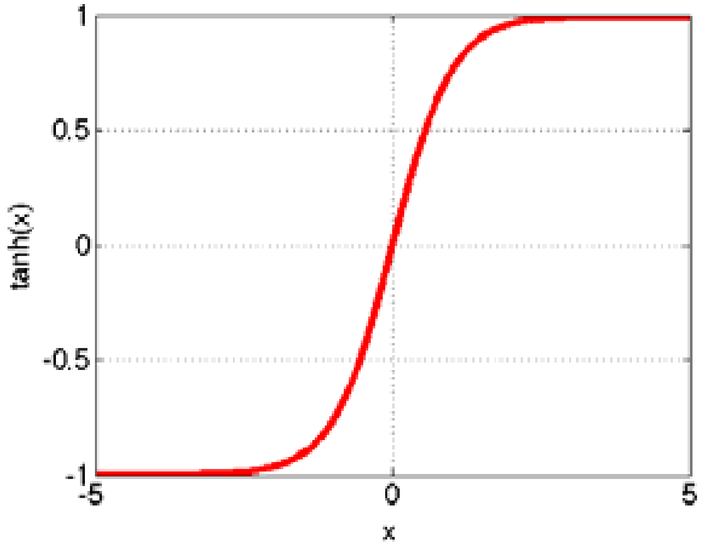
**Use nonlinear activation functions.**

**Options:**

- Tanh

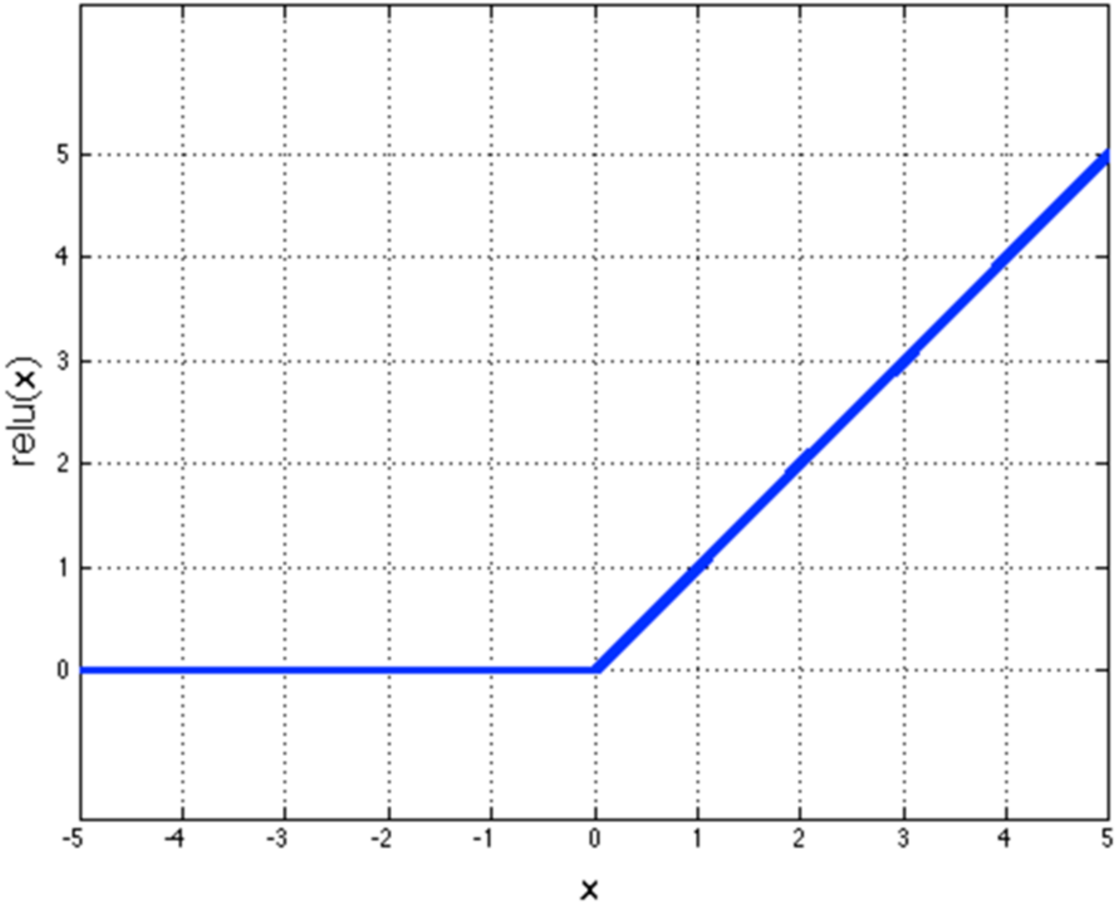
- Sigmoid: 1/(1+exp(-x))



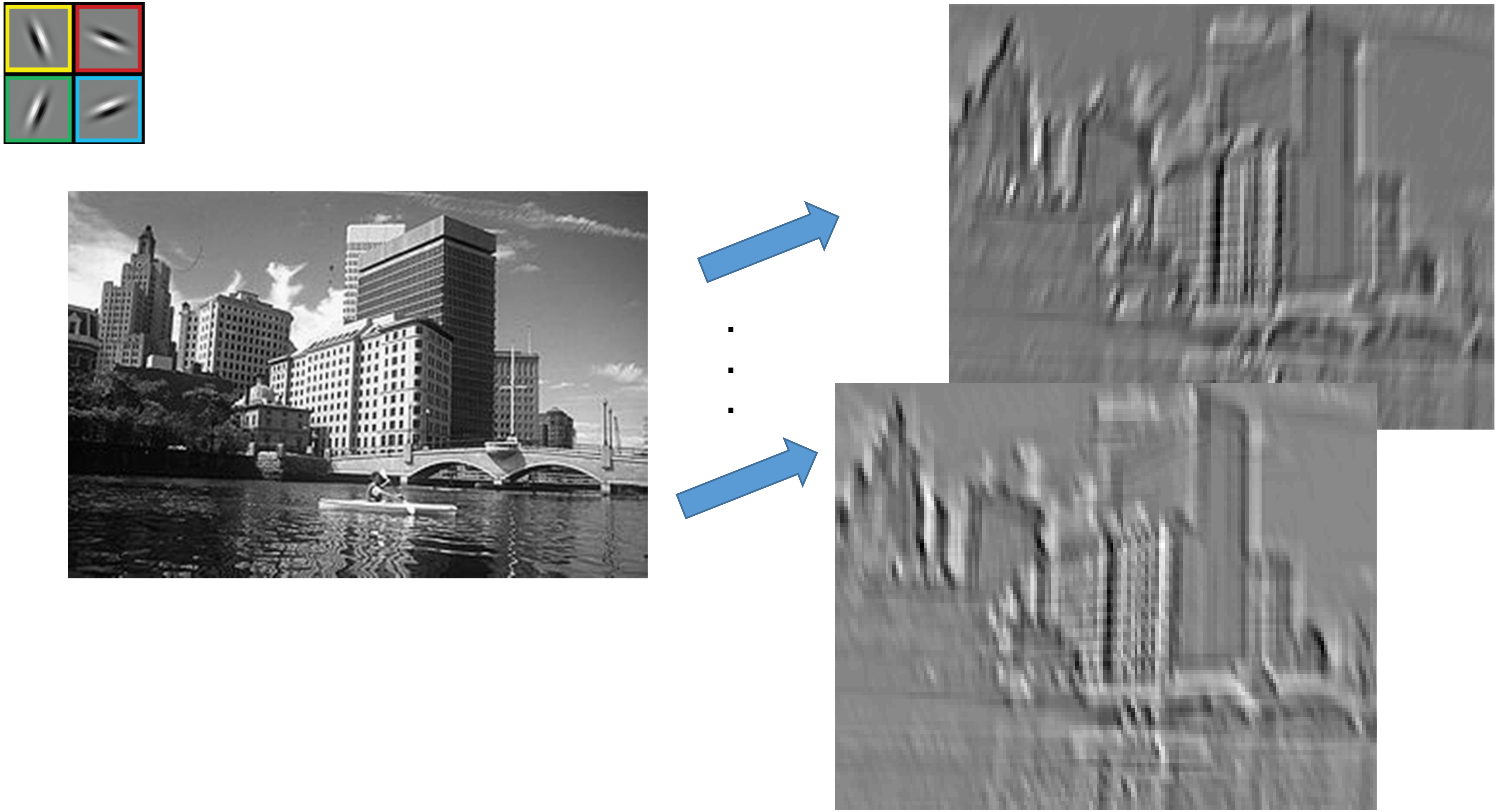
- Rectified linear unit (ReLU)
  - Simplifies backpropagation
  - Makes learning faster
  - Avoids saturation issues

 Preferred option




The convolution layer output:

$$x_j^{(l)} = f(I_j^{(l)})$$



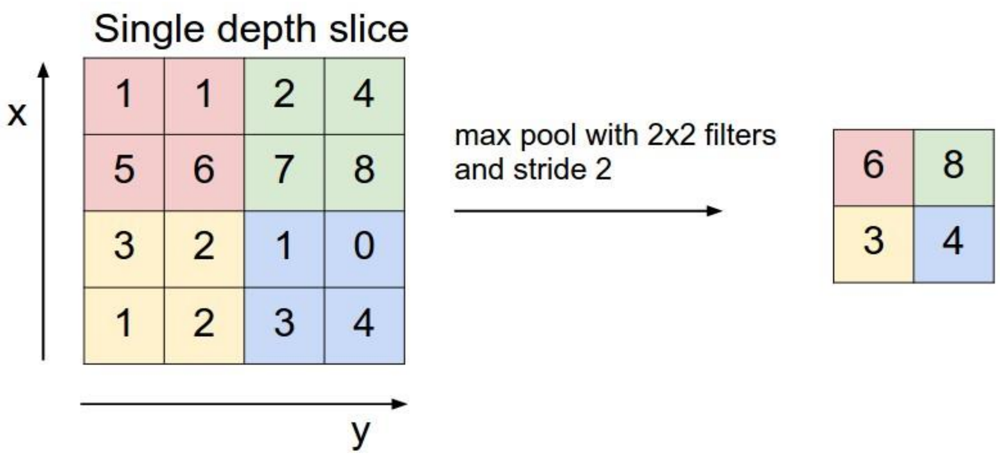
## Spatial Pooling

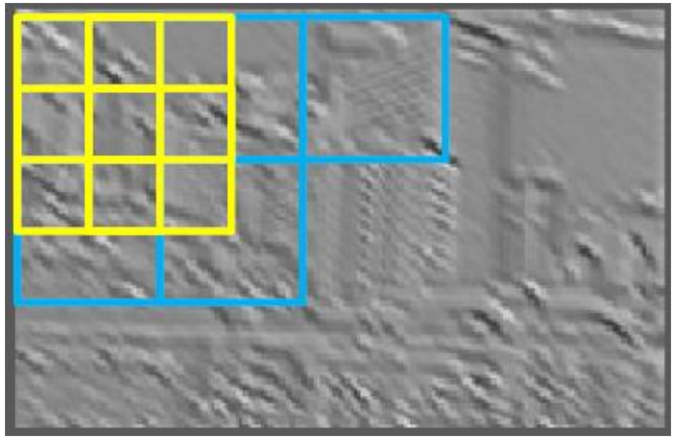
- Sum or max
- Non-overlapping / overlapping regions
- Generally 2x2 with stride of 2, sometimes 3x3 with stride of 2
- Limited connections to previous layer

Role:

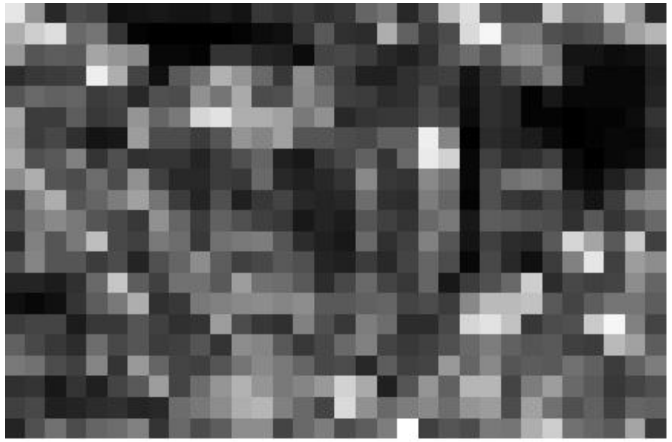
- Invariance to small transformations
- Larger receptive fields (see more of input)

  C1:3x3, C2:5x5, C3:7x7

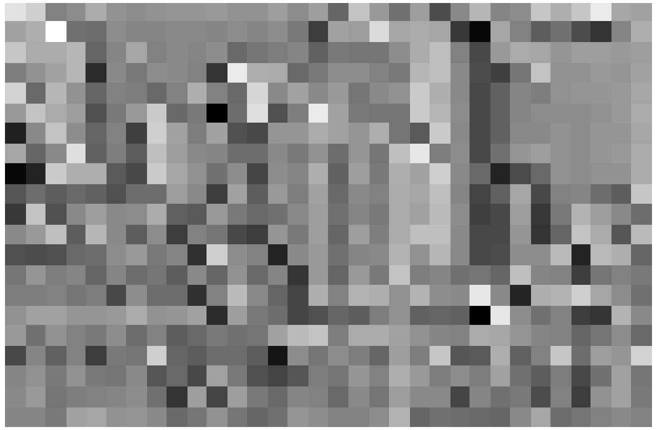

Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters and stride 2
6 8
3 4



**Max**

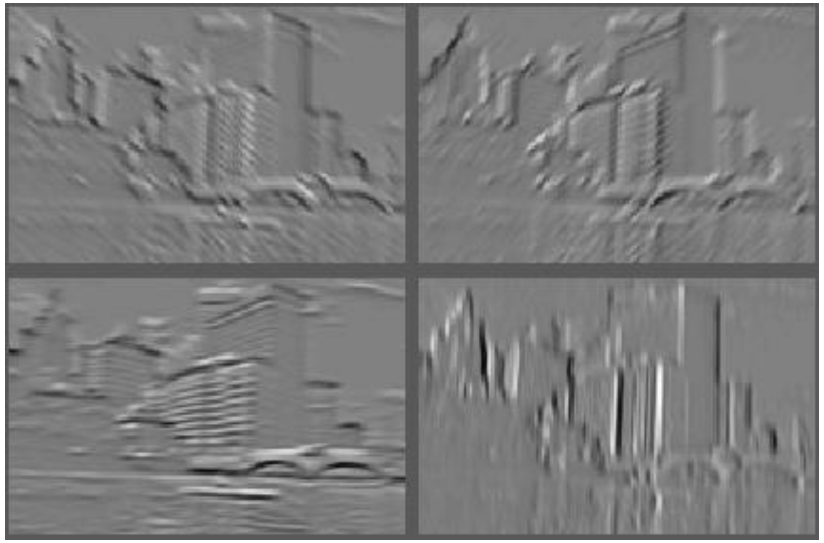


**Sum**

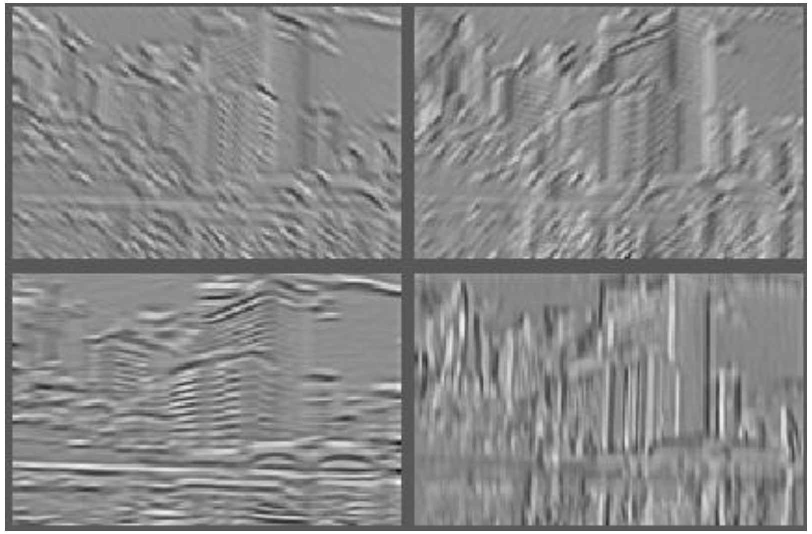


## Normalization

- Contrast normalization within or across feature maps
- Before or after spatial pooling



Feature Maps



Feature Maps
After Contrast Normalization

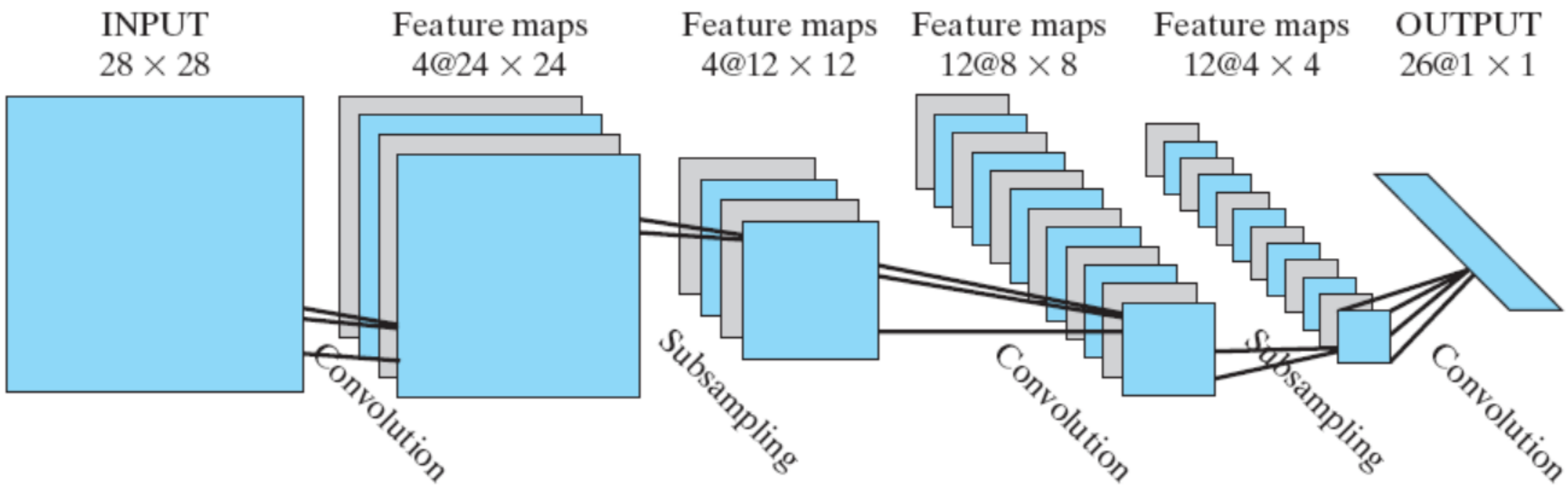
## Network Training

- Often the BP algorithm has been used.
- Very slow: training on ImageNet data (e.g. 1.2 million images from 1000 classes) takes 2 to 3 weeks on multiple GPUs.
- A large training set is required.
- If the training set is not large, create new samples by rotating the existing samples
- Use pre-trained networks as your initial network and adapt and re-train (transfer learning).

# Example: Recognition of handwritten characters

❖ The input layer: made up of 28 X 28 sensory nodes, receives the images of different characters that have been centered and normalized in size.

❖ The first hidden layer performs convolution. It consists of four feature maps. Each feature map consists of 24 X 24 neurons. Each neuron is assigned a receptive field of size 5 X 5.



❖ The second hidden layer performs subsampling and local averaging.

- Consists of four feature maps, each feature map is made up of 12 X 12 neurons.

- Each neuron has a receptive field of size 2 X 2, a trainable coefficient, a trainable bias, and a sigmoid activation function.

❖ The third hidden layer performs a second convolution.

- Consists of 12 feature maps Each feature map consists of 8 X 8 neurons.

- Each neuron in this hidden layer may have synaptic connections from several feature maps in the previous hidden layer.

❖ The fourth hidden layer performs a second subsampling and local averaging.

- Consists of 12 feature maps, each feature map consisting of 4 X 4 neurons.

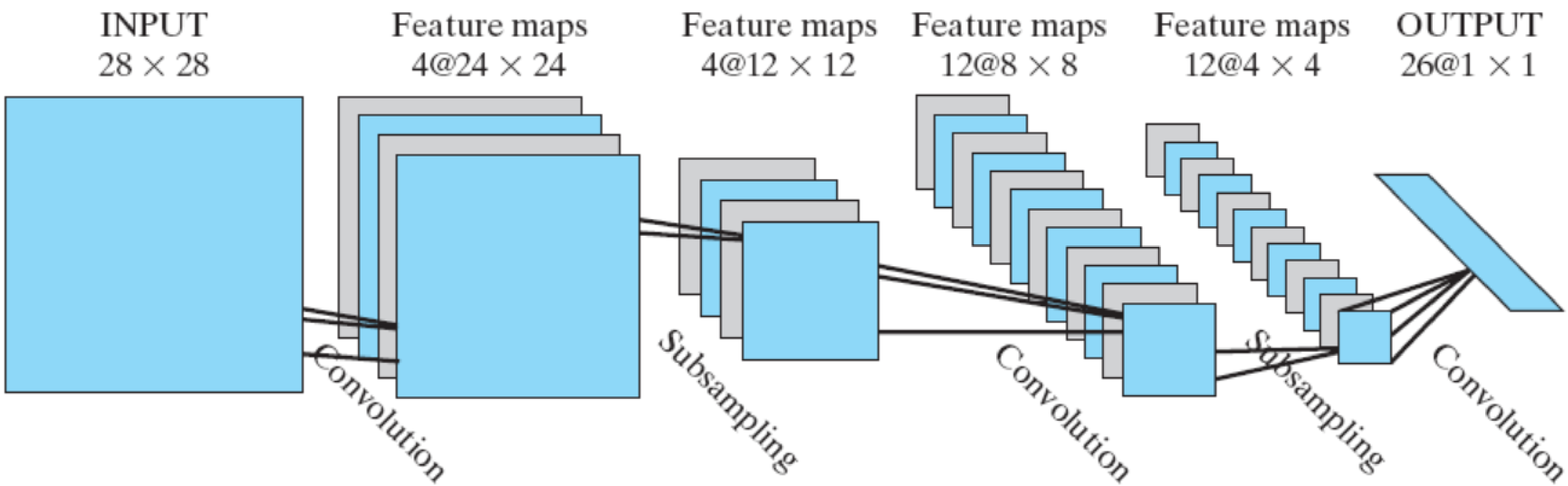
❖ The output layer performs one final stage of convolution.

- Consists of 26 neurons, with each assigned to one of 26 possible characters.

- Each neuron is assigned a receptive field of size 4 X 4.

❖ Contains approximately 100,000 synaptic connections, but only about 2,600 free parameters. This dramatic reduction in the number of free parameters is achieved through the use of weight sharing.

❖ The synaptic weights and bias levels can be learned by cycling the simple back-propagation algorithm through the training sample.




receptive field 5 X 5

receptive field 2 X 2

receptive field 5 X 5

receptive field 2 X 2

receptive field 4 X 4

# Popular CNNs

- **LeNet**

  First successful applications developed by Yann LeCun in 1990's. The best known is the LeNet architecture that was used to read zip codes, digits, etc.

- **AlexNet**

  Popularized CNN in Computer Vision, developed by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton.

  In ImageNet ILSVRC challenge in 2012 significantly outperformed the second runner-up (top 5 error of 16% compared to runner-up with 26% error).

  The Network had a similar architecture basic as LeNet, but was deeper, bigger, and featured Convolutional Layers stacked on top of each other.

- **ZF Net**

  The ILSVRC 2013 winner was a CNN from Matthew Zeiler and Rob Fergus. An improvement on AlexNet by tweaking the architecture hyperparameters, in particular by expanding the size of the middle convolutional layers.

- **GoogLeNet**

  The ILSVRC 2014 winner was a CNN from Szegedy et al. from Google. Main contribution: development of an *Inception Module* that dramatically reduced the number of parameters in the network (4M, compared to AlexNet with 60M).

- **VGGNet**

The runner-up in ILSVRC 2014 was the network from Karen Simonyan and Andrew Zisserman, known as the VGGNet.

Main contribution: showing that the depth of the network is a critical component for good performance.

Despite its slightly weaker classification performance, VGG *features* outperform those of GoogLeNet in transfer learning tasks.

VGG is currently the most preferred choice in the community when extracting CNN features from images.

In particular, their pre-trained model is available for plug and play use in Caffe.

A downside of the VGGNet is that it is more expensive to evaluate and uses a lot more memory and parameters (140M).

## CNN packages:

- Cuda-convnet2 by Alex Krizhevsky is a ConvNet implementation that supports multiple GPUs.
- ConvNetJS CIFAR-10 demo allows you to play with ConvNet architectures and see the results and computations in real time, in the browser.
- Caffe, one of the most popular ConvNet libraries, (Y. Jia, Berkeley).
- Example Torch 7 ConvNet that achieves 7% error on CIFAR-10 with a single model.
- Ben Graham's Sparse ConvNet package, which Ben Graham used to great success to achieve less than 4% error on CIFAR-10.
- Overfeat (NYU)
- MatConvNet, Matlab implementation, includes 8 models, pre-trained with ImageNet, 6 VGG-based models(Oxford), 2 Caffe-based models(Berkeley), CPU and GPU, http://www.vlfeat.org/matconvnet/quick/

# THE END