یادگیری عمیق

جلسه ۱۳

شبکه‌های عصبی کانوولوشنال

Convolutional Neural Networks (CNN)

کاظم فولادی قلعه

دانشکده مهندسی، پردیس فارابی

دانشگاه تهران

http://courses.fouladi.ir/deep

یادگیری عمیق

شبکه‌های عصبی کانوولوشنال

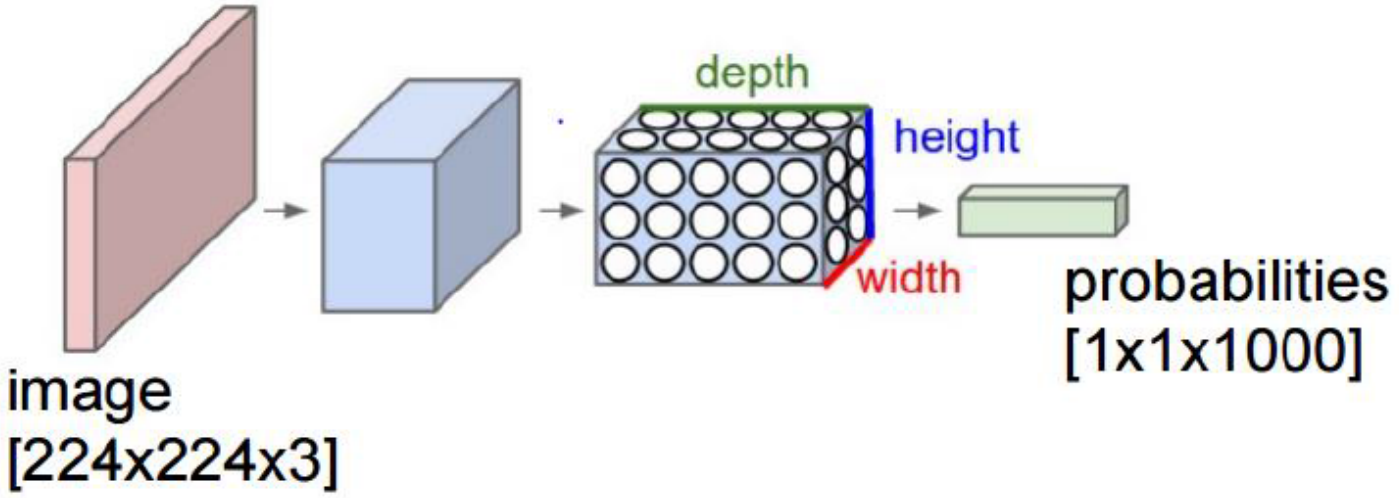
۱

مقدمات

شبکه‌های عصبی کانوولوشنال

CONVOLUTIONAL NEURAL NETWORKS (CNN)

شبکه‌های عصبی کانوولوشنال، یک بازنمایی پیچیده از داده‌های دیداری را
با استفاده از مقدار حجیمی از داده‌ها یاد می‌گیرند.
این شبکه‌ها از سیستم بینایی انسان الهام گرفته شده‌اند و چندین لایه از تبدیلات را یاد می‌گیرند،
که هر یک بر روی دیگری اعمال می‌شود و به‌صورت پیش‌رونده،
بازنمایی پیچیده‌تری از ورودی را استخراج می‌کنند.



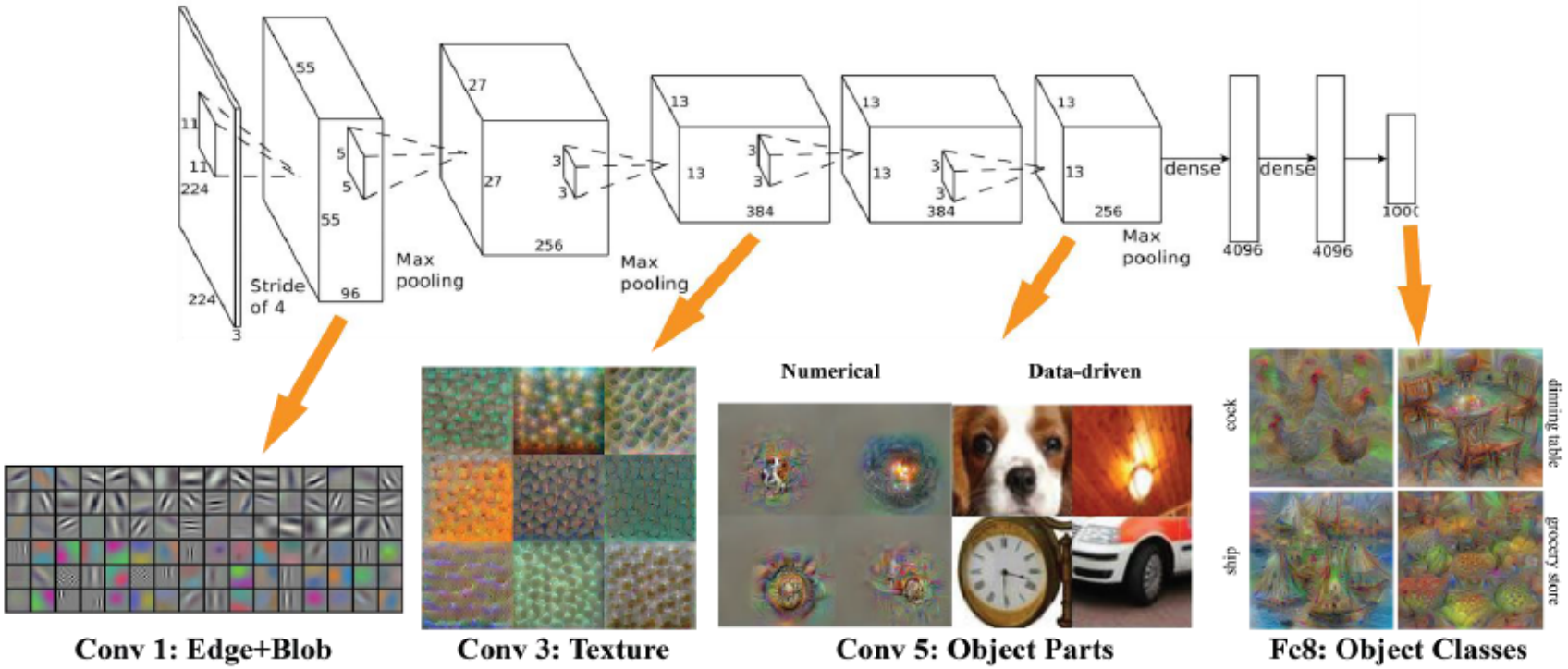
مثال از کاربرد: **دسته‌بندی تصاویر (Image Categorization)**

هر لایه از یک CNN یک حجم سه–بعدی از اعداد را دریافت می‌کند و یک حجم سه–بعدی از اعداد را خروجی می‌دهد.
برای مثال: تصویر یک مکعب $3 \times 224 \times 224$ (RGB) است و به یک بردار 1000 تایی از احتمالات تبدیل می‌شود.

شبکه‌های عصبی کانوولوشنال

لایه‌های اصلی

CONVOLUTIONAL NEURAL NETWORKS (CNN)



لایه‌ی کانوولوشن
Convolution Layer

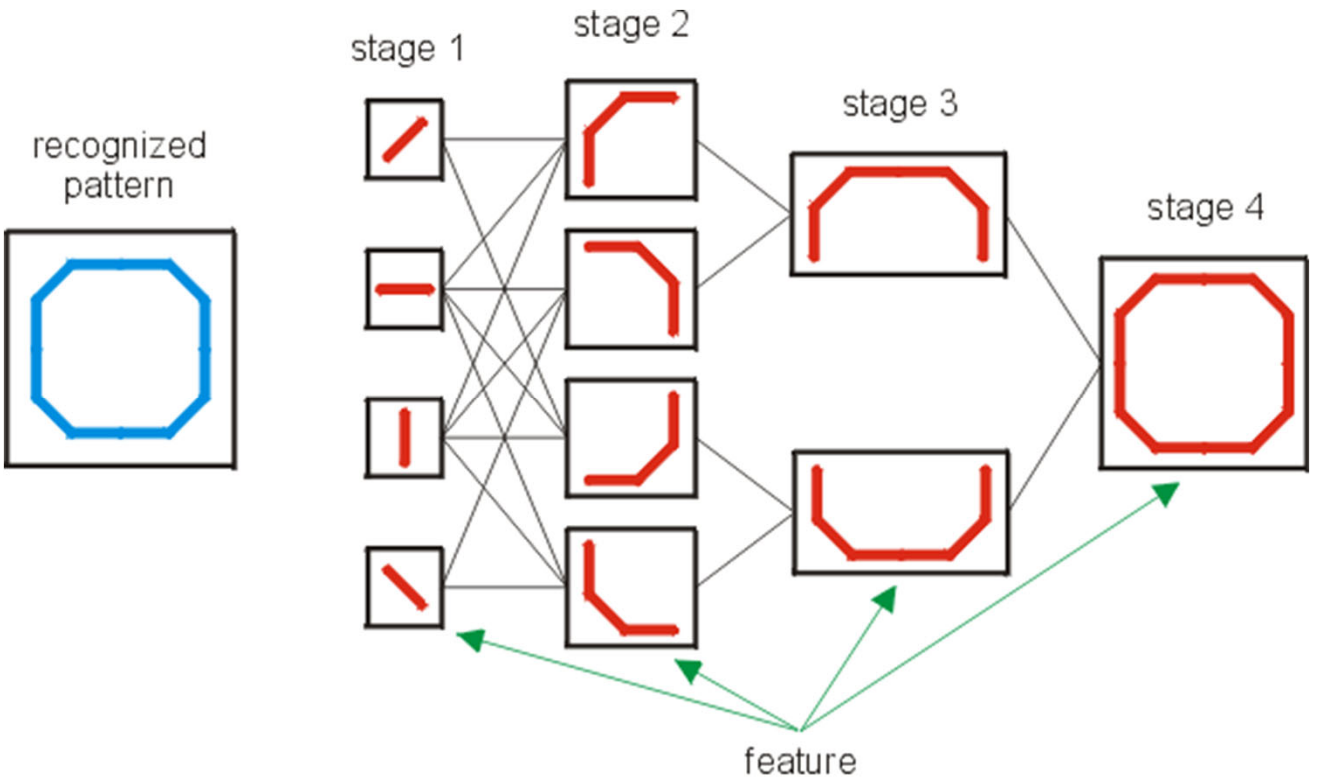
لایه‌ی کانوولوشن، یک آشکارساز ویژگی است که به‌صورت خودکار و جادویی یاد می‌گیرد که با استفاده از کرنل کانوولوشن اطلاعات غیر لازم از یک ورودی را فیلتر کند .

لایه‌ی تلفیق
Pooling Layer

لایه‌ی تلفیق، ماکزیمم یا مقدار متوسط ویژگی‌های خاص بر روی یک ناحیه از داده‌های ورودی را محاسبه می‌کند (⇐ کاهش اندازه‌ی تصاویر ورودی).
همچنین به آشکارسازی اشیا در برخی مکان‌های نامعمول کمک می‌کند و اندازه‌ی حافظه را کاهش می‌دهد.

شبکه‌های عصبی کانوولوشنال

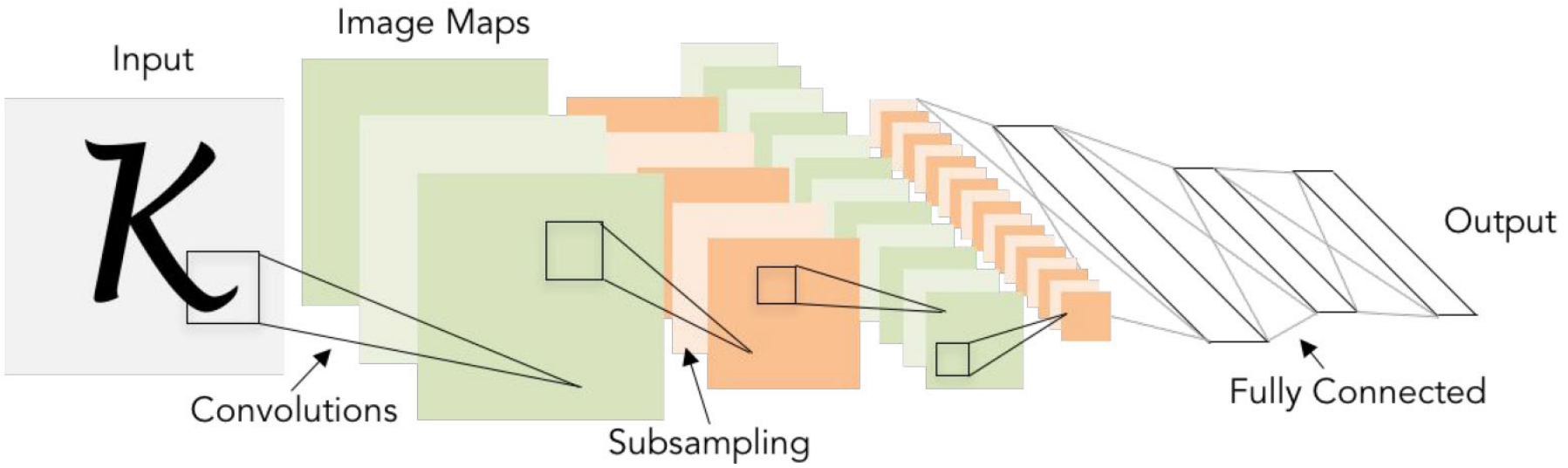
کارکرد نوعی لایه‌ها



شبکه‌های عصبی کانوولوشنال

معماری

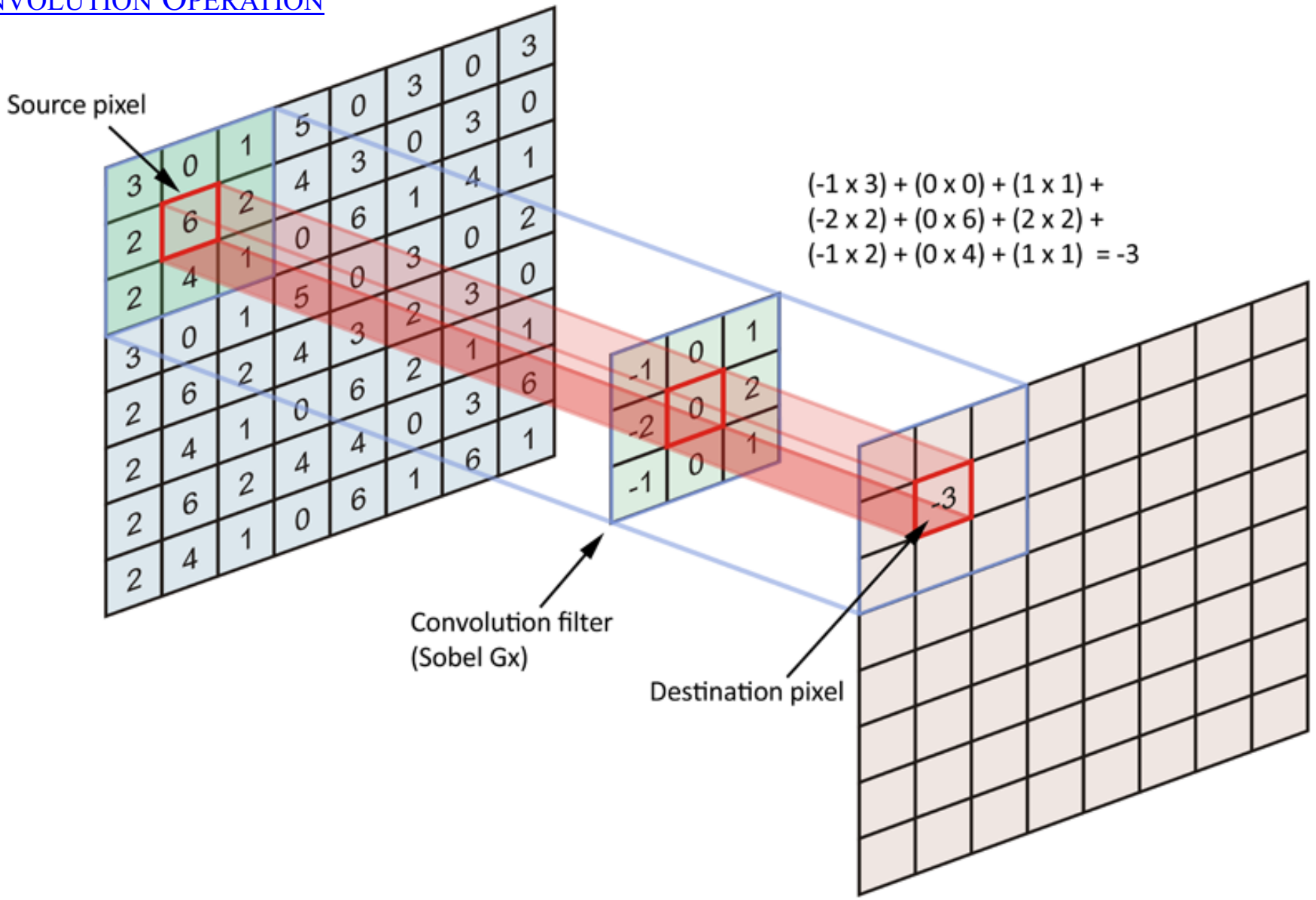
CONVOLUTIONAL NEURAL NETWORKS



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن

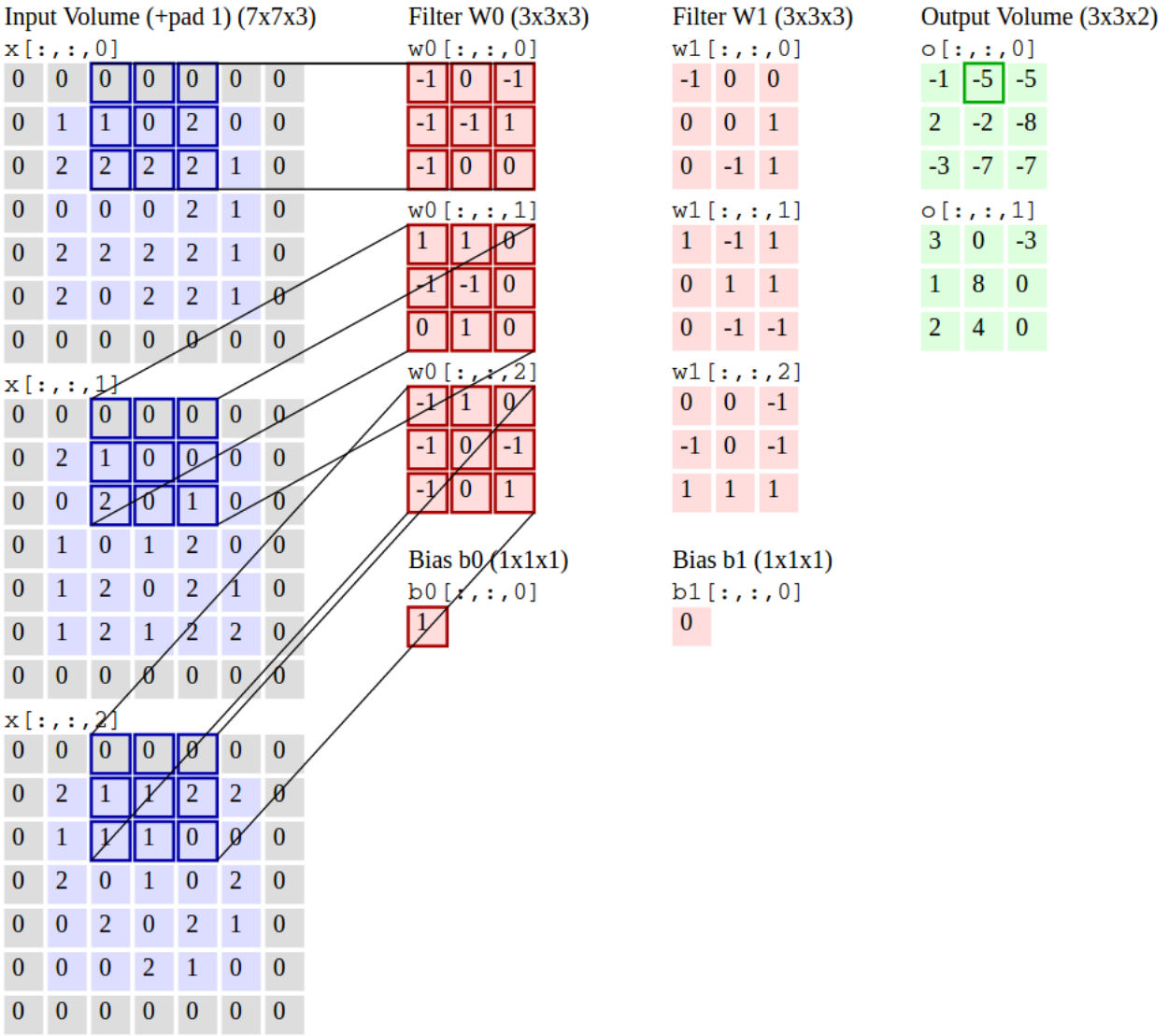
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن

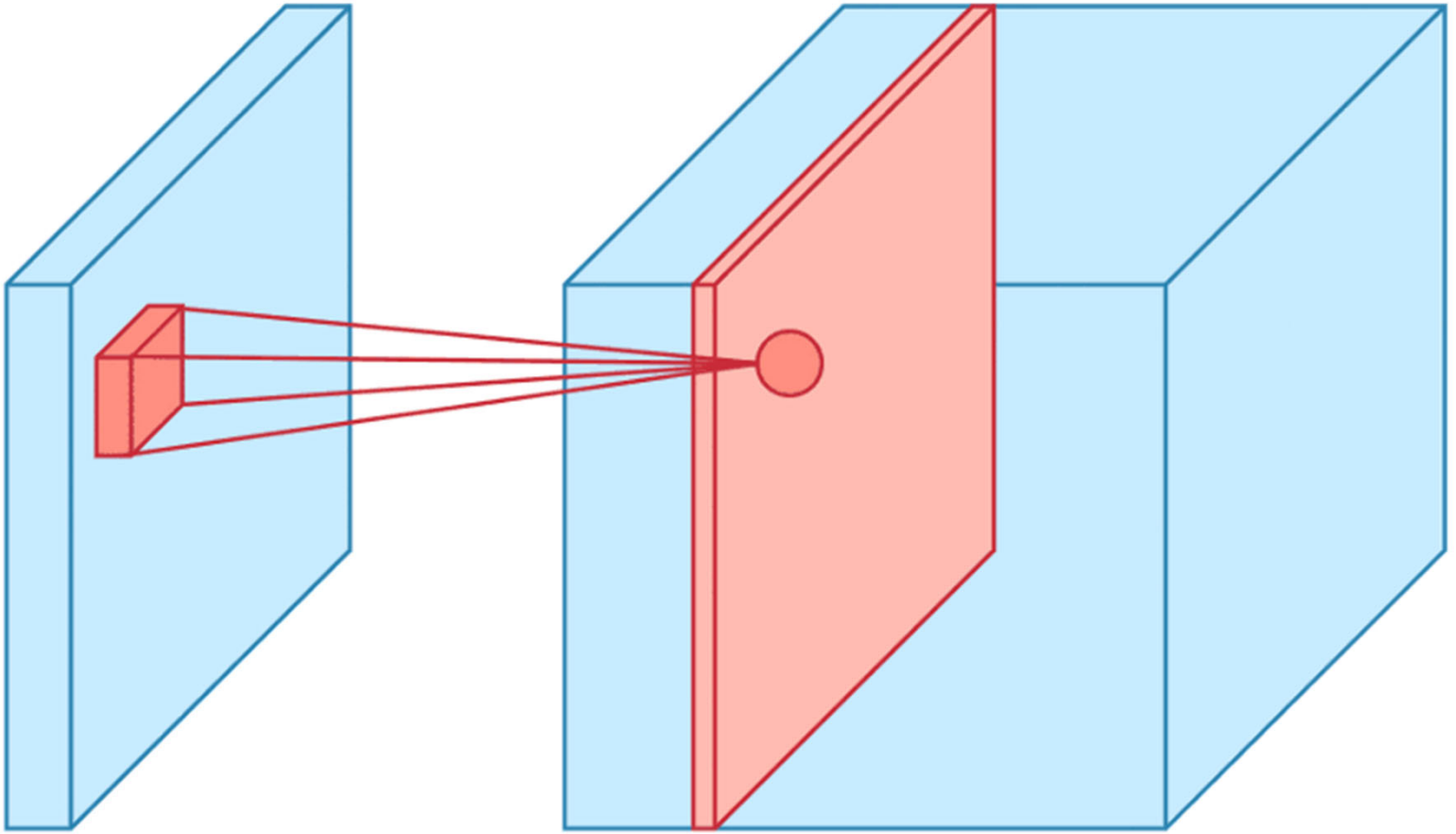
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن

CONVOLUTION OPERATION

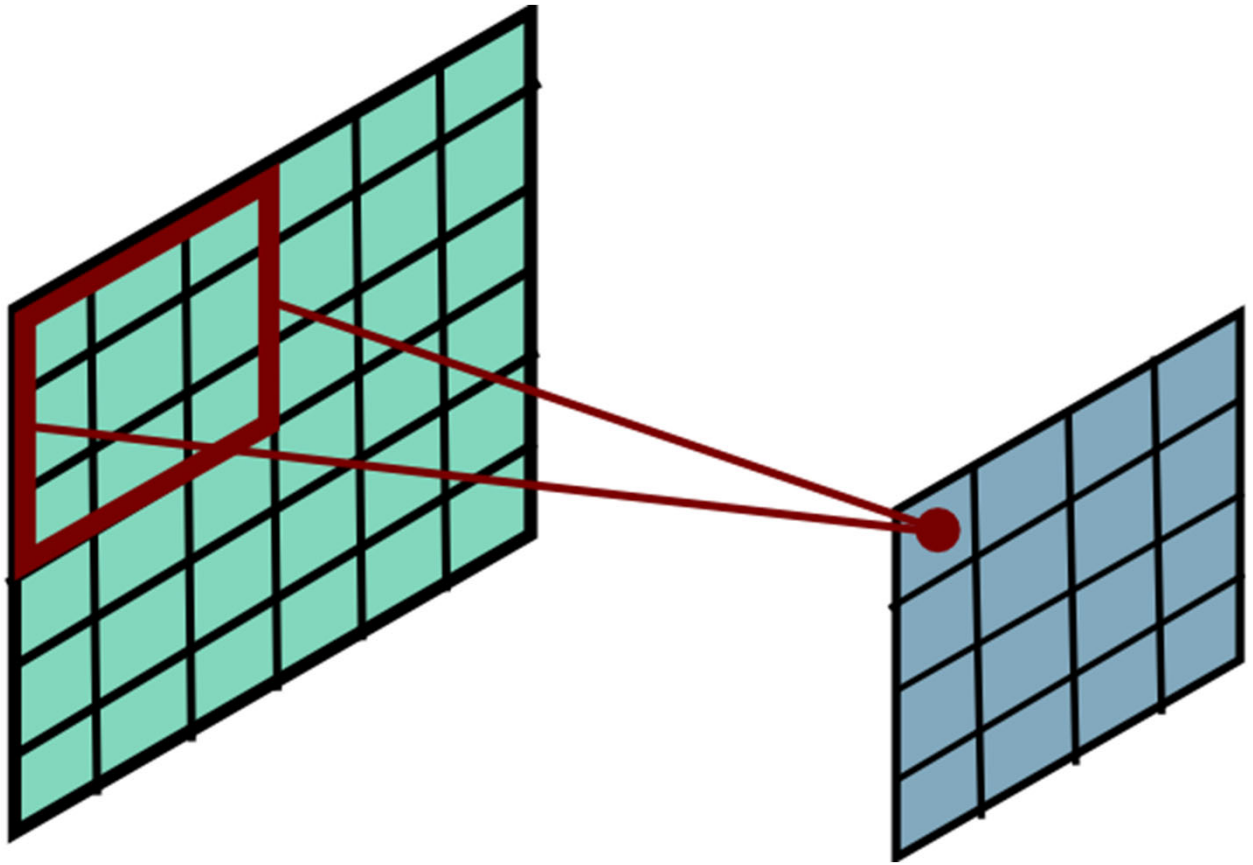


https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۱ از ۱۶)

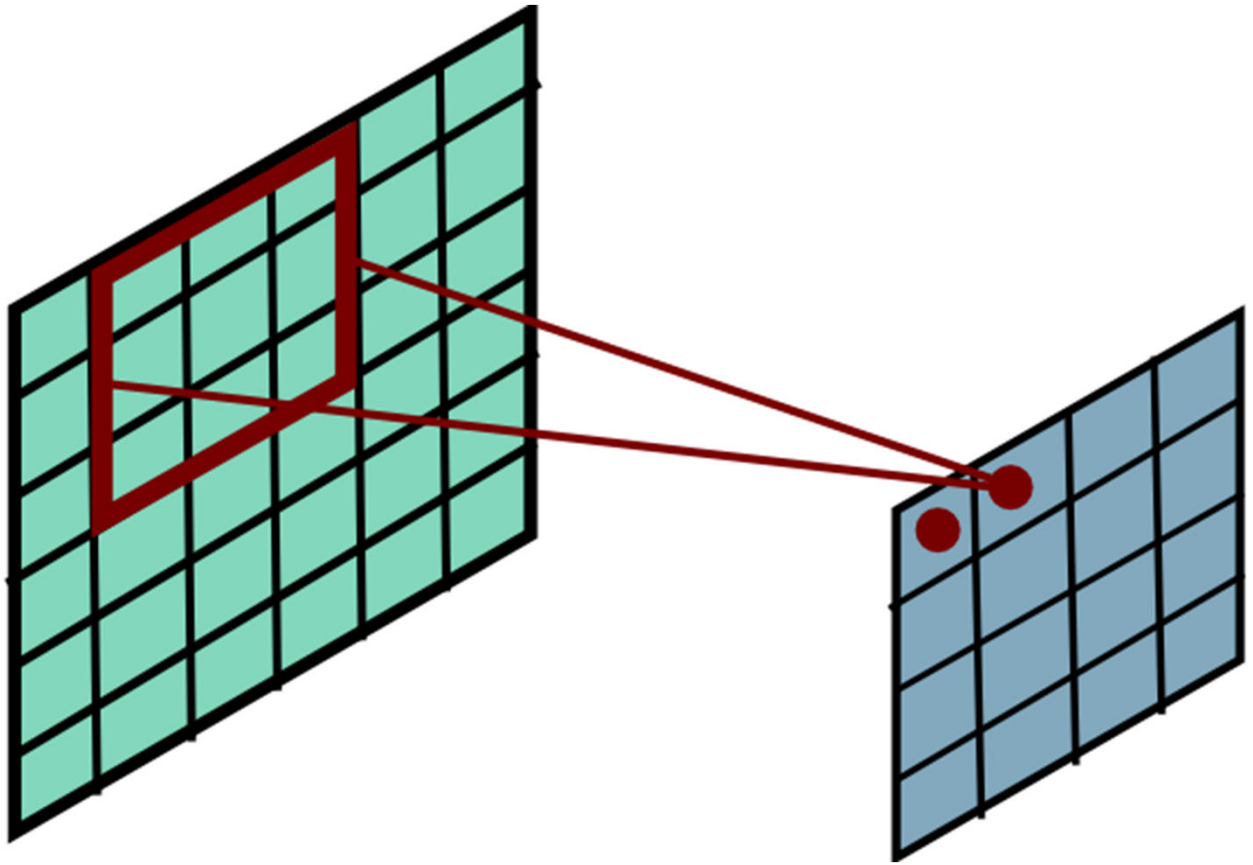
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۲ از ۱۶)

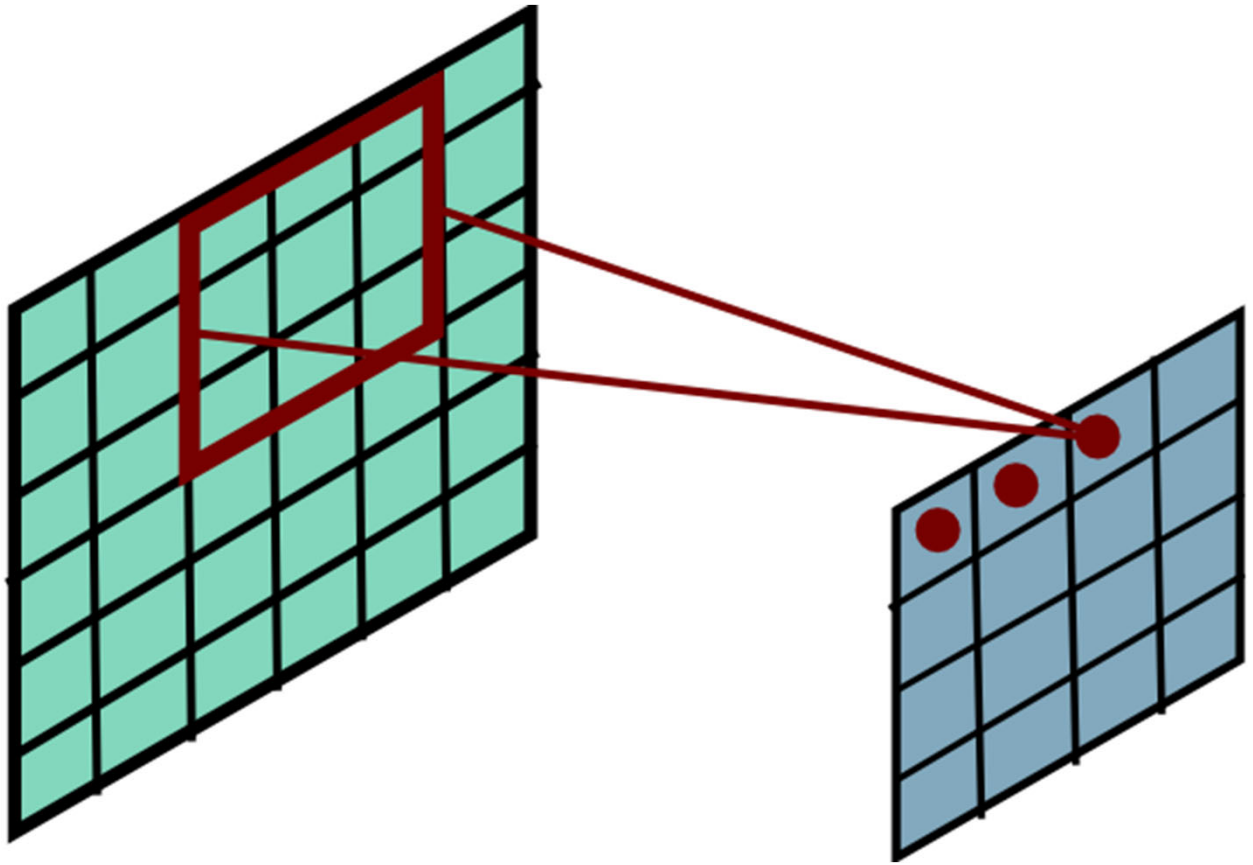
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۳ از ۱۶)

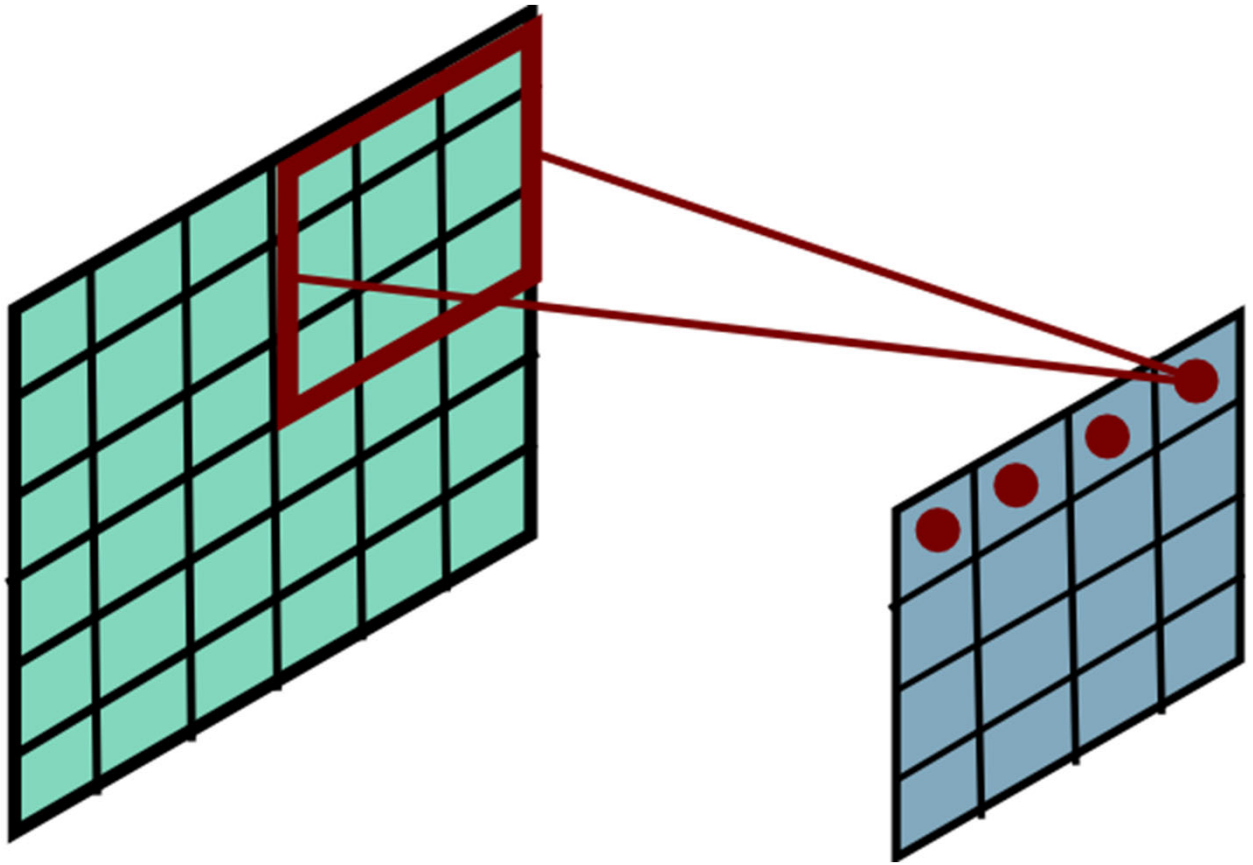
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۴ از ۱۶)

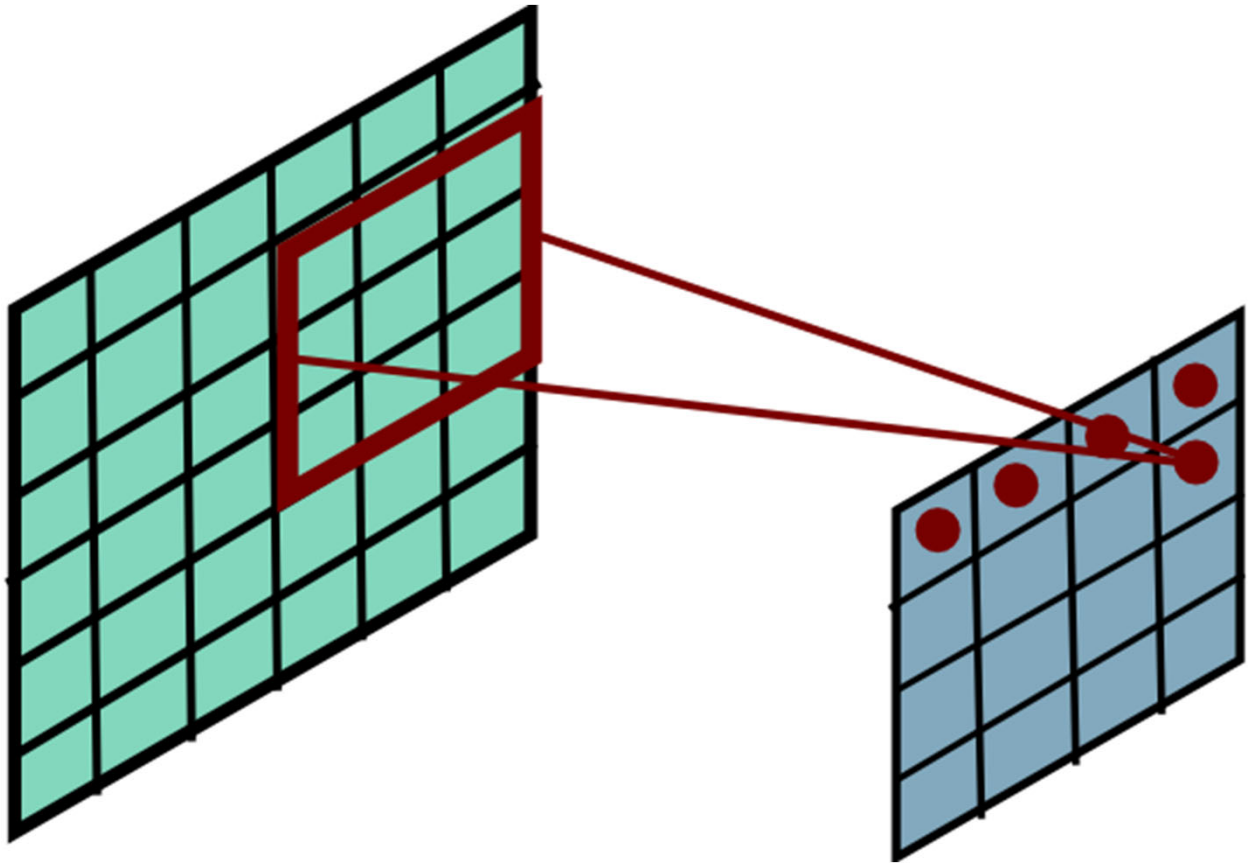
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۵ از ۱۶)

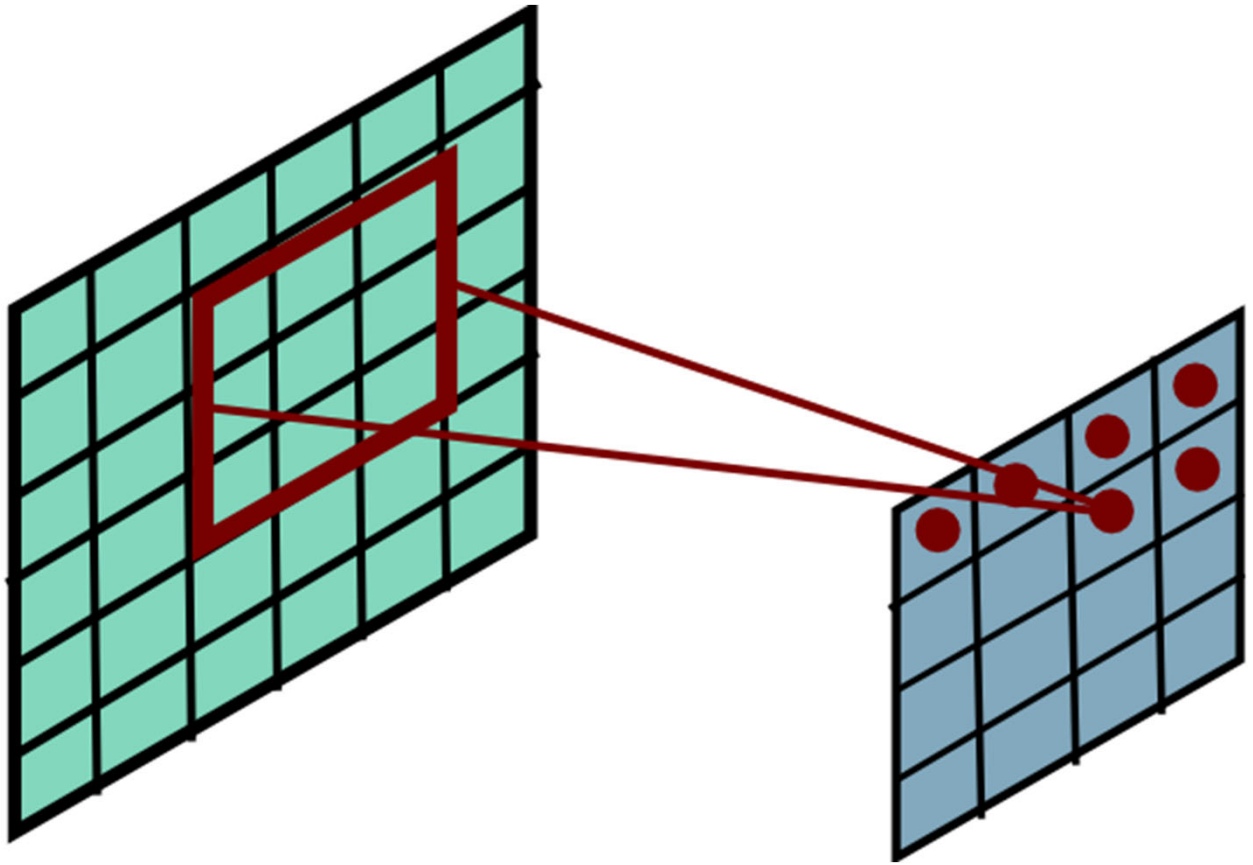
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۶ از ۱۶)

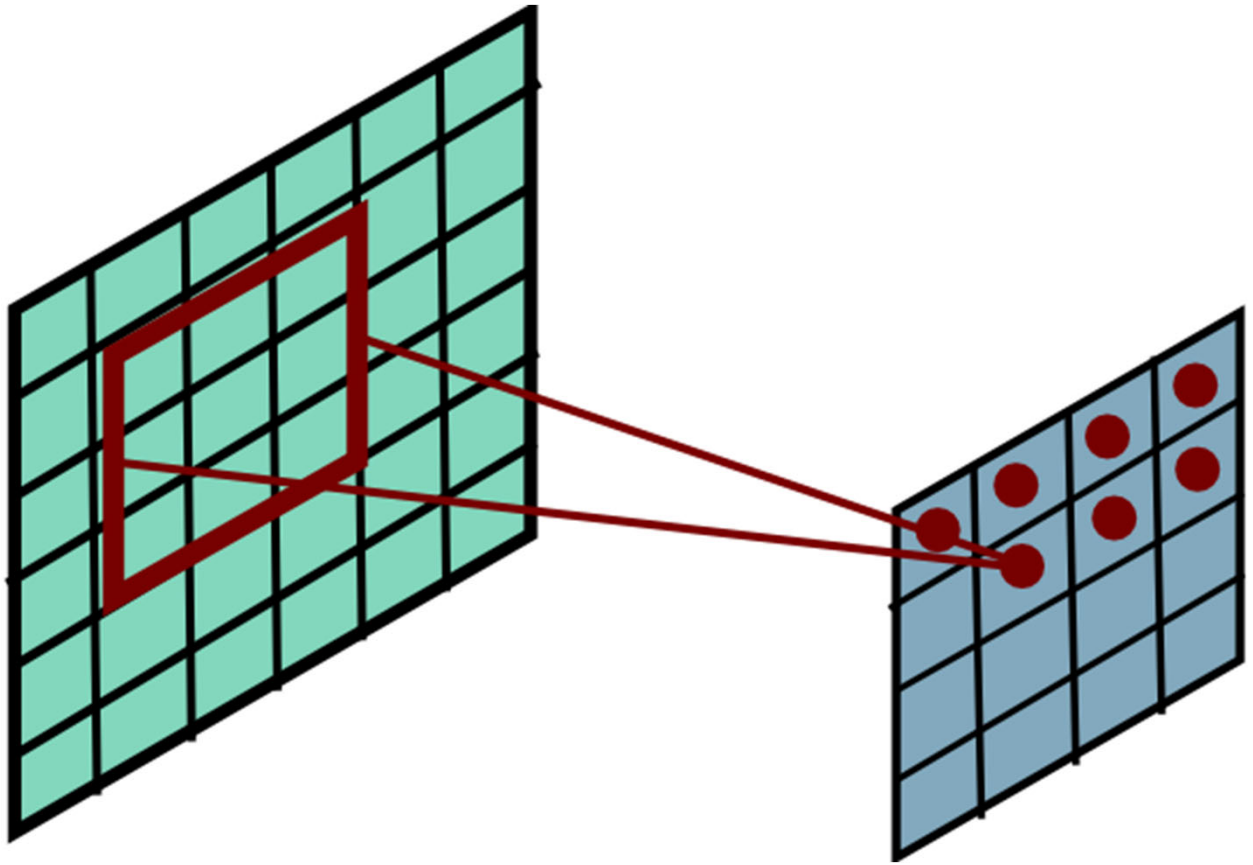
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۷ از ۱۶)

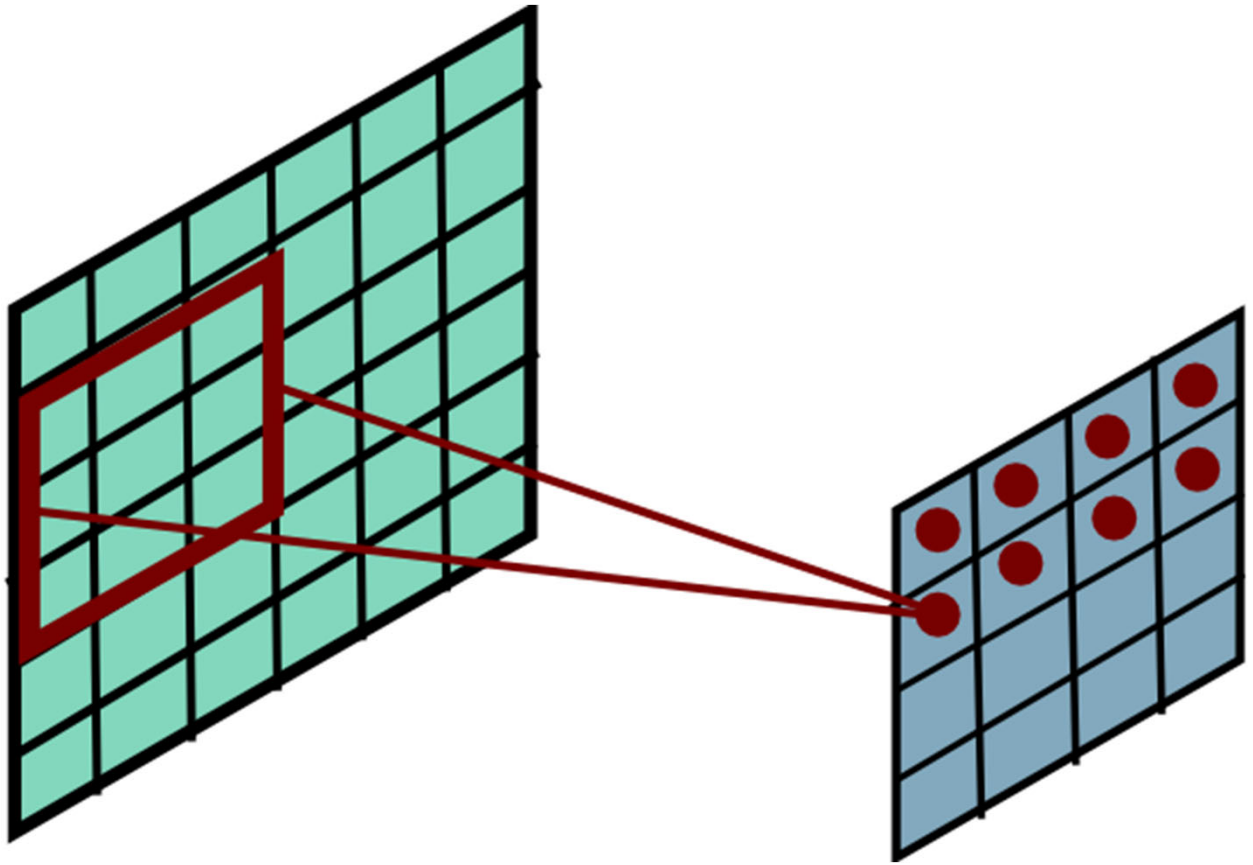
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۸ از ۱۶)

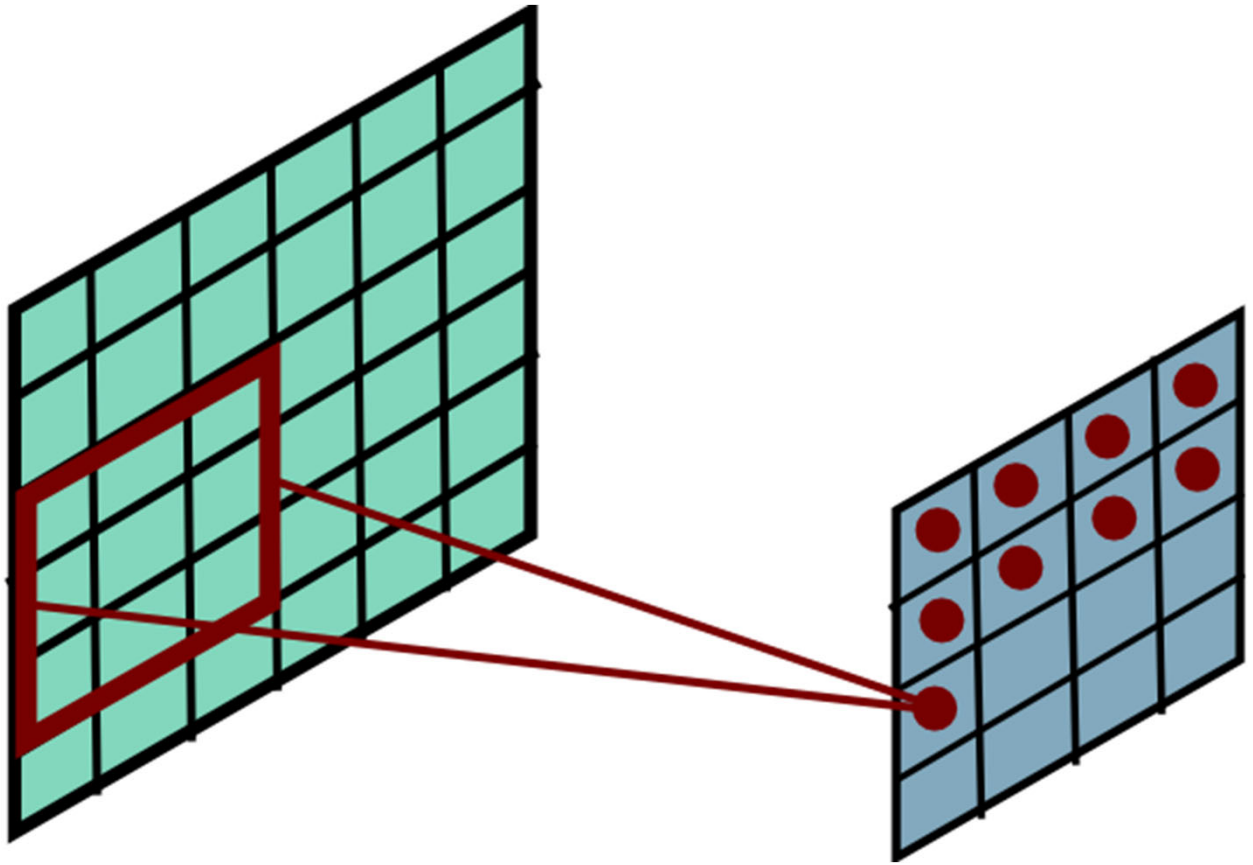
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۹ از ۱۶)

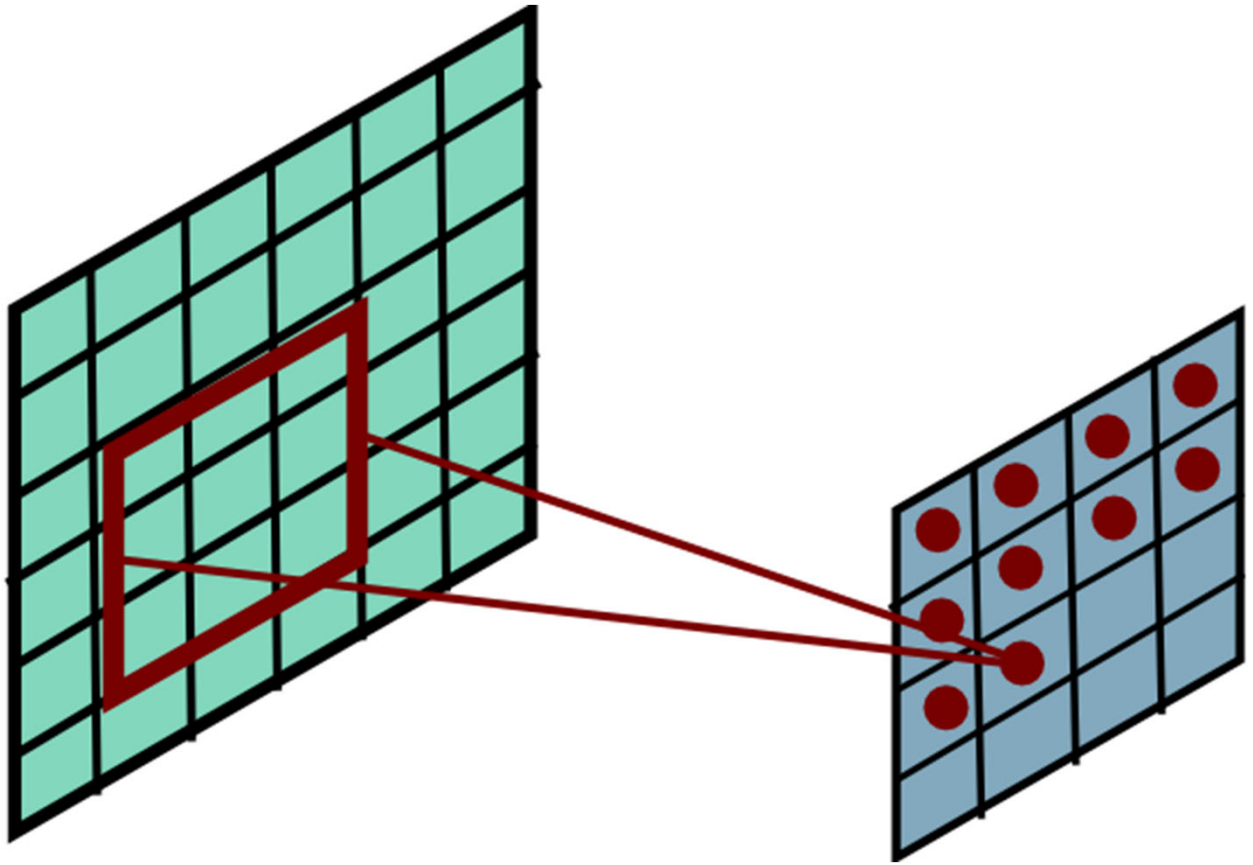
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۱۰ از ۱۶)

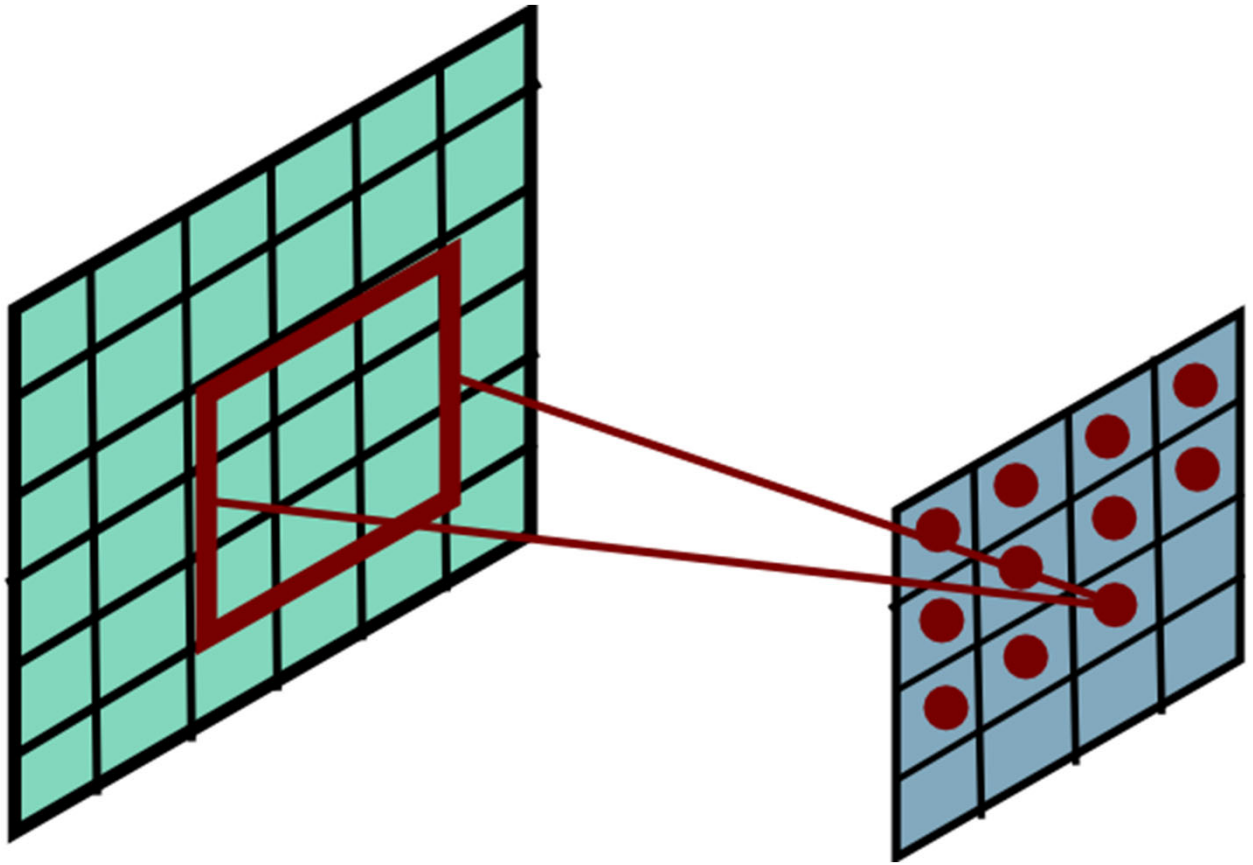
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۱۱ از ۱۶)

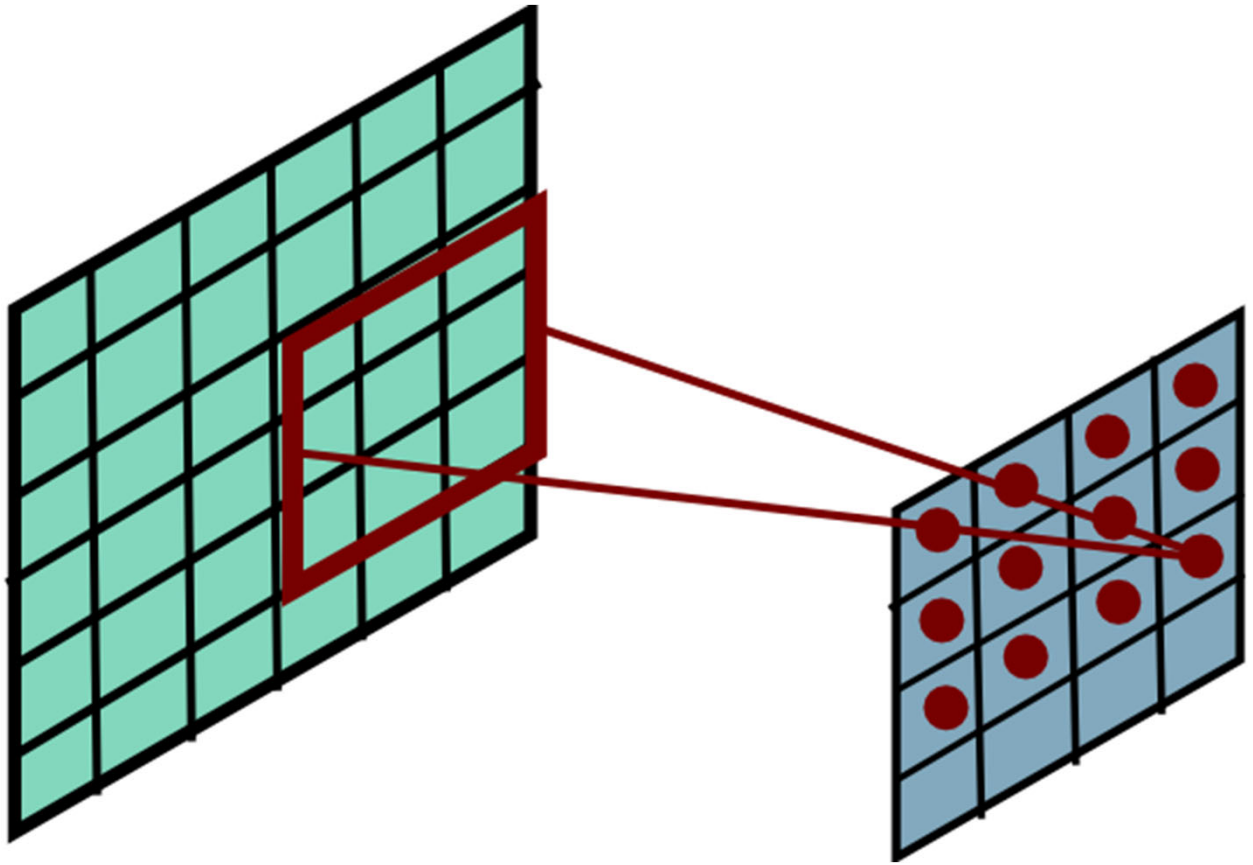
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۱۲ از ۱۶)

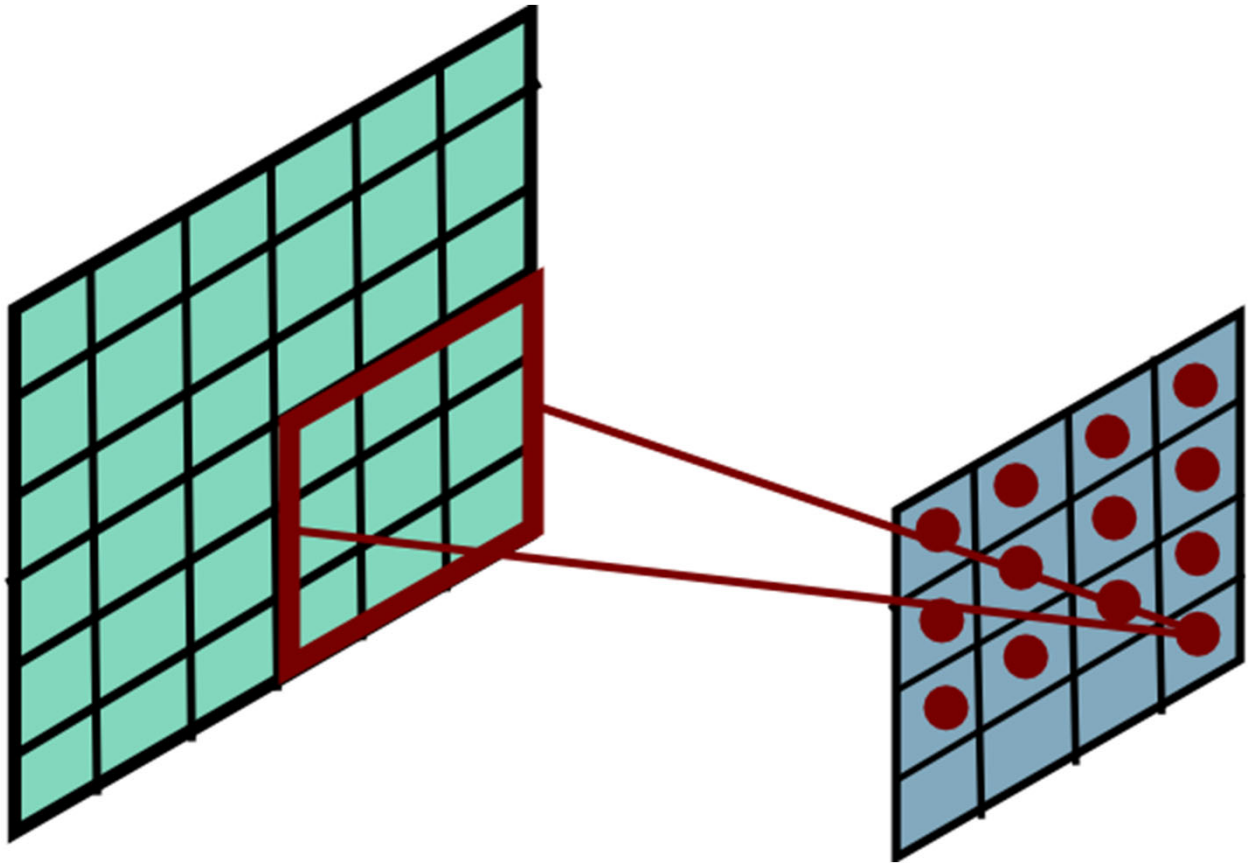
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۱۳ از ۱۶)

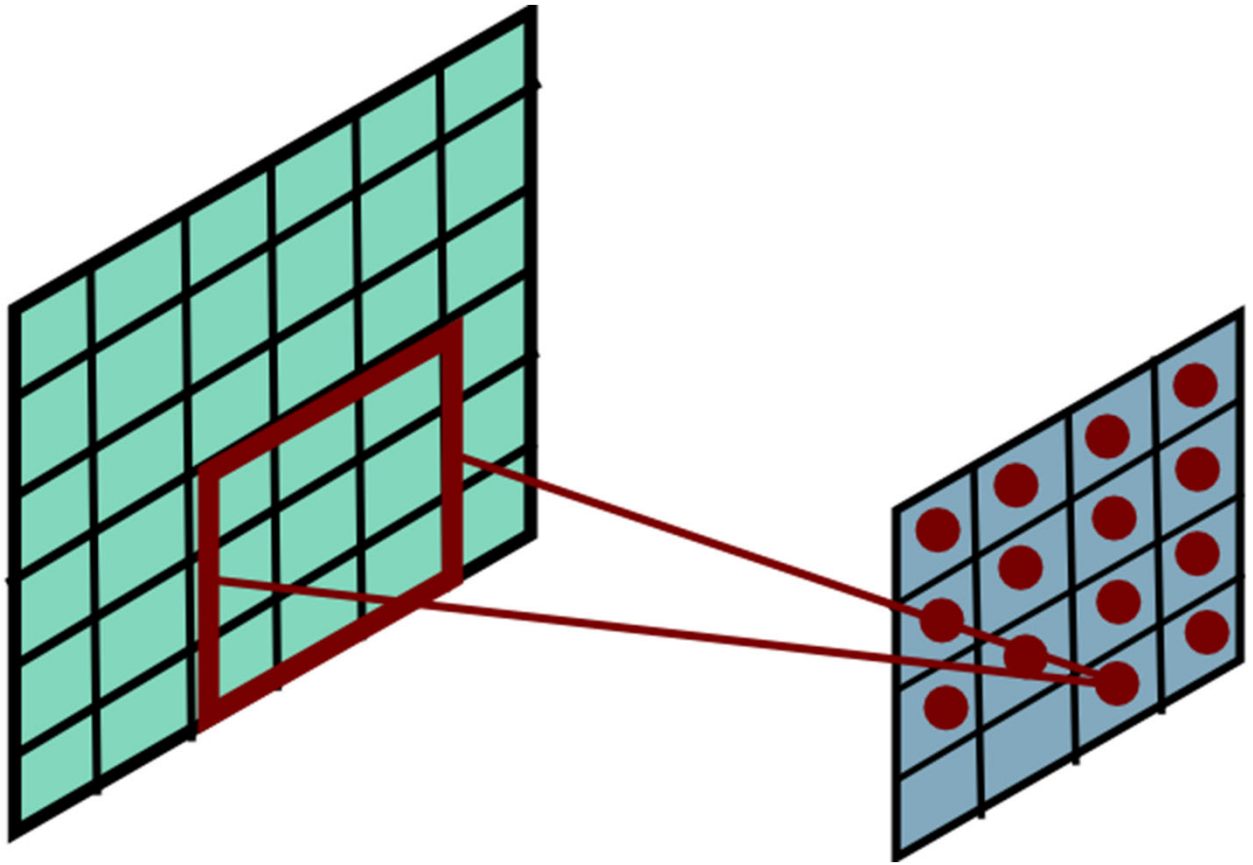
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۱۴ از ۱۶)

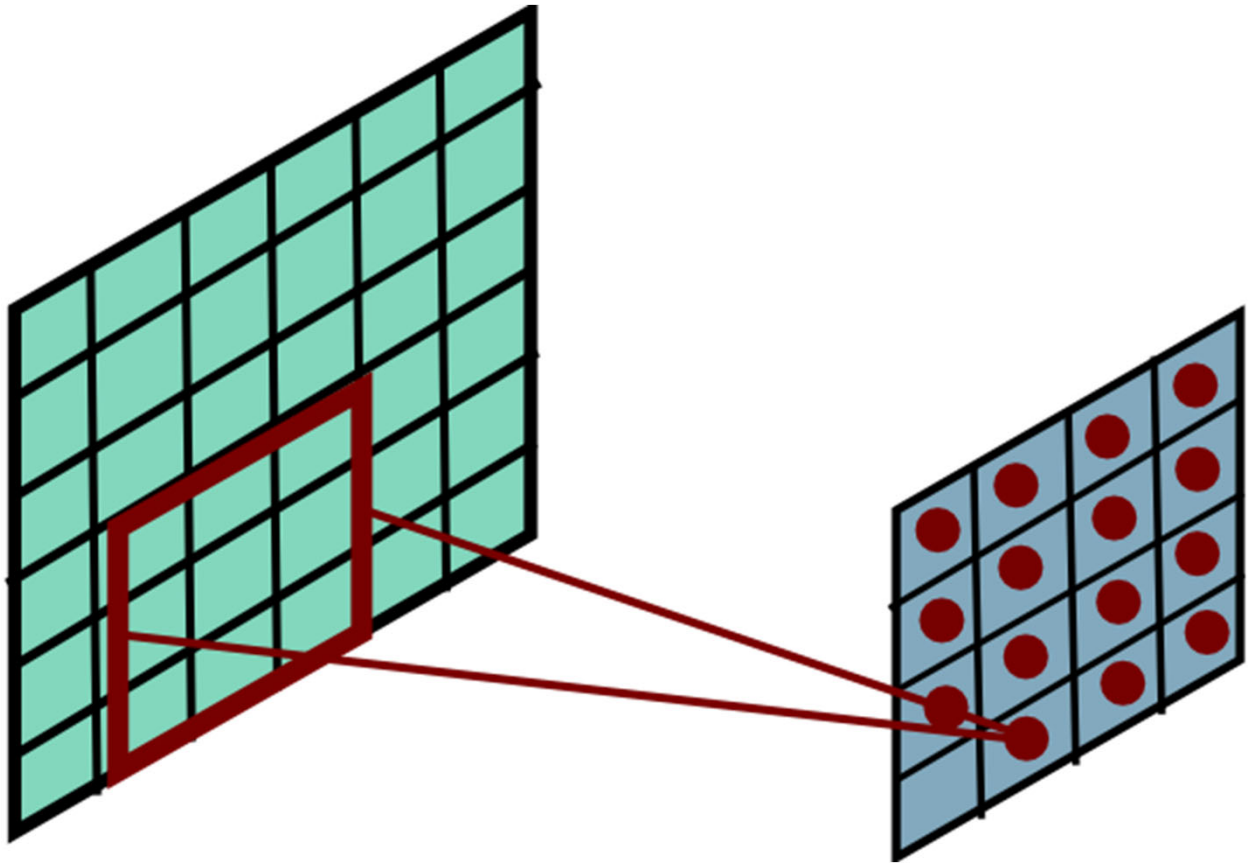
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۱۵ از ۱۶)

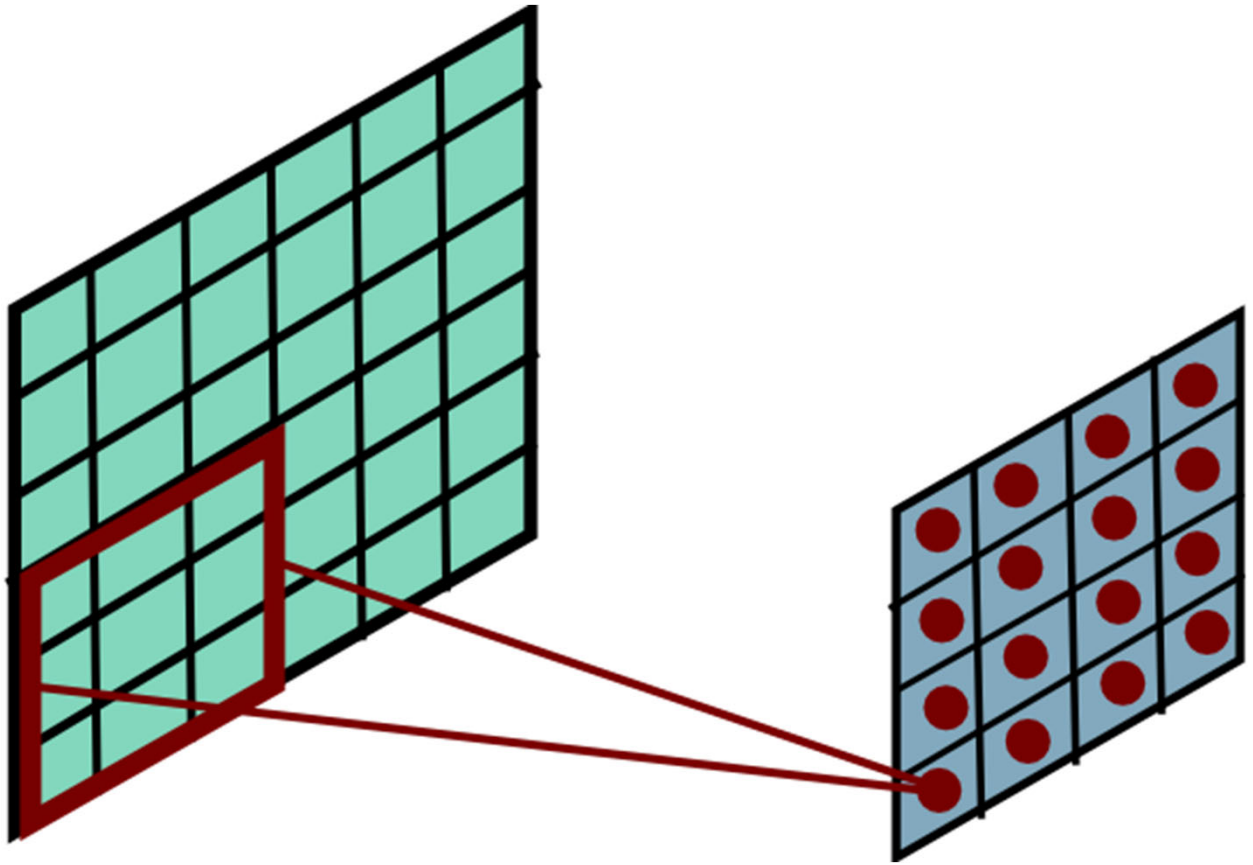
CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

عملیات کانوولوشن (۱۶ از ۱۶)

CONVOLUTION OPERATION



شبکه‌های عصبی کانوولوشنال

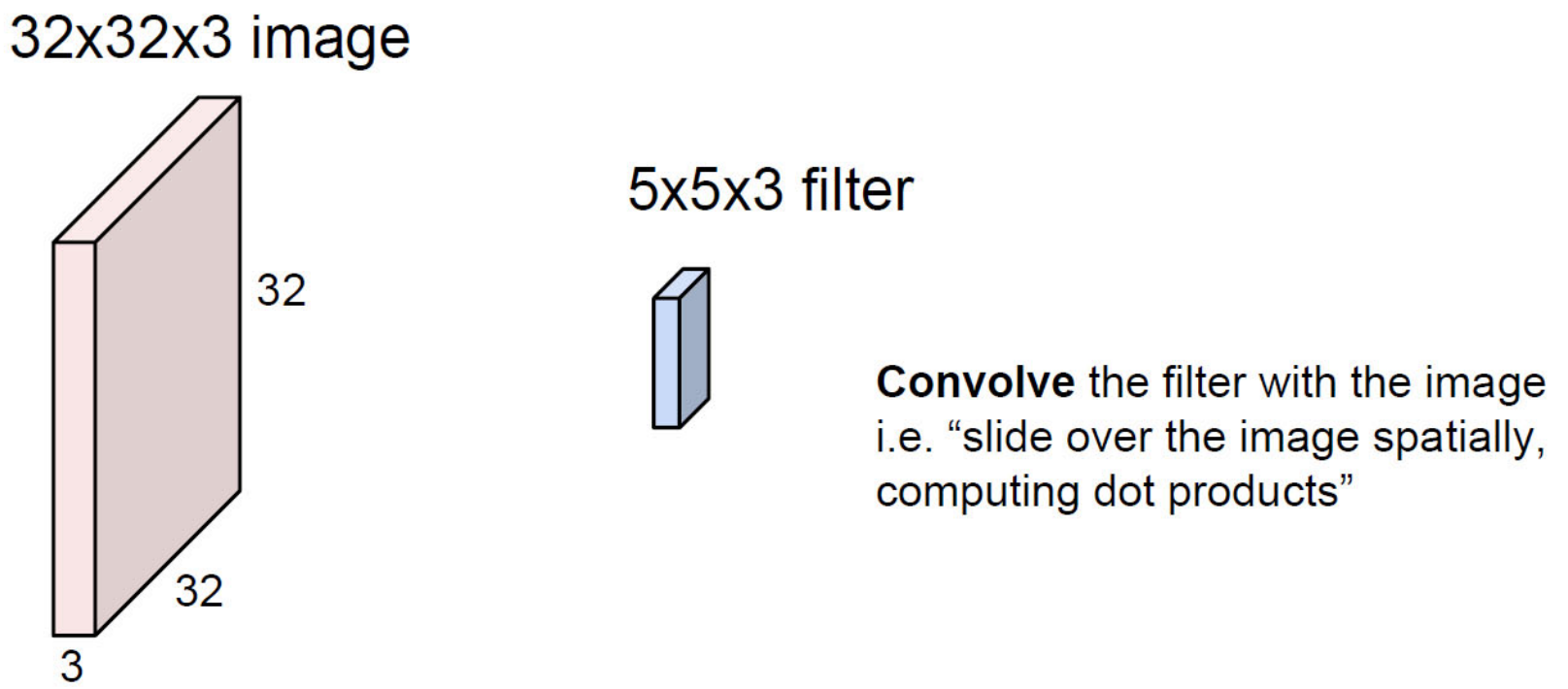
لایه‌ی کانوولوشن

CONVOLUTION LAYER

شبکه‌های عصبی کانوولوشنال

لایه‌ی کانوولوشن

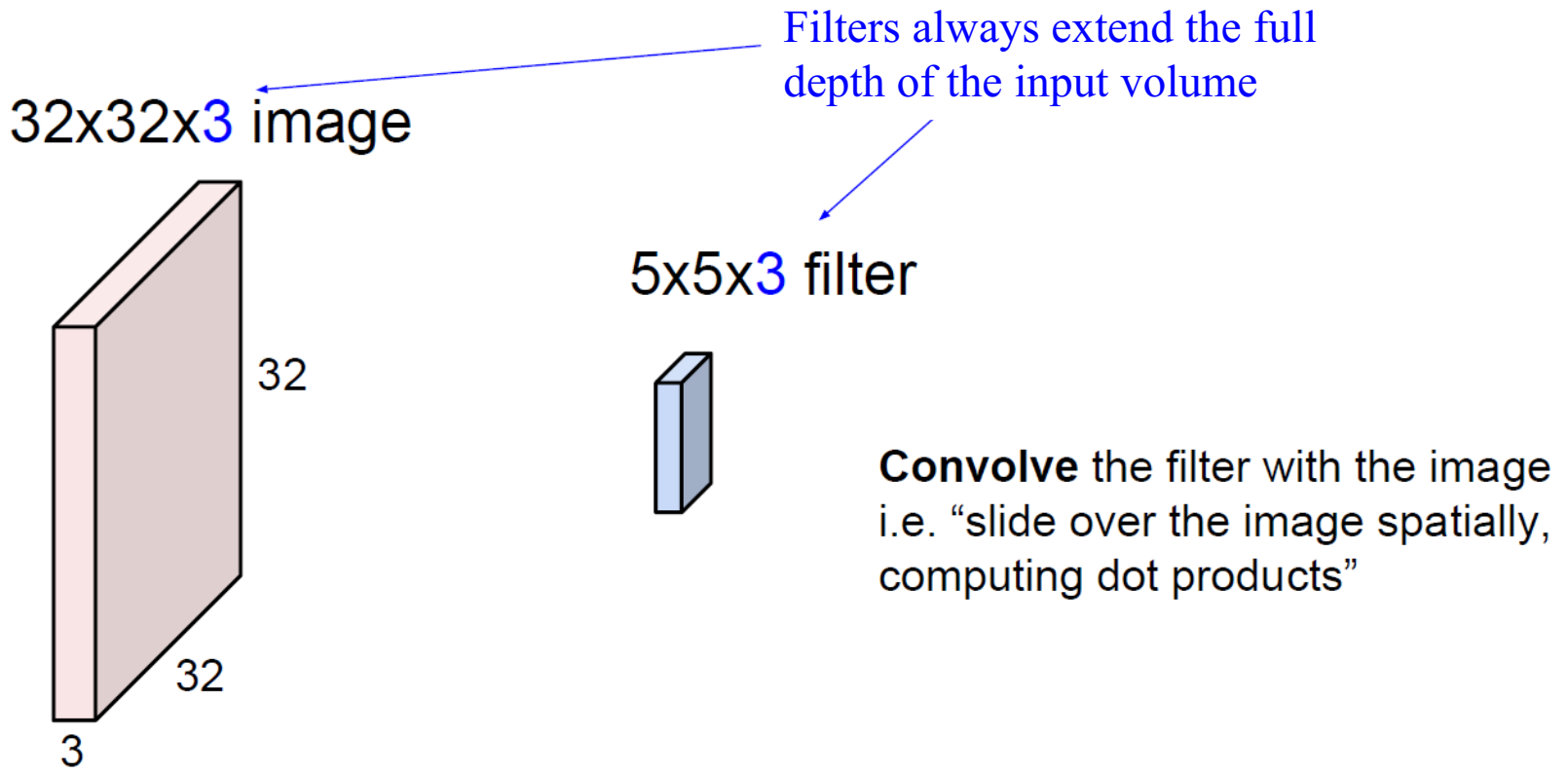
CONVOLUTION LAYER



شبکه‌های عصبی کانوولوشنال

لایه‌ی کانوولوشن

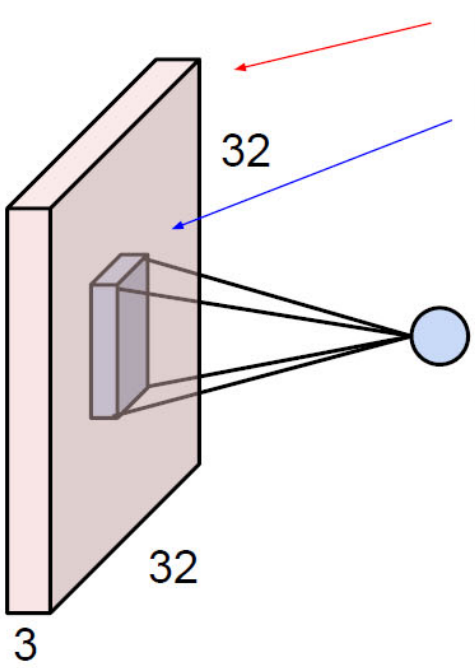
CONVOLUTION LAYER



شبکه‌های عصبی کانوولوشنال

لایه‌ی کانوولوشن

CONVOLUTION LAYER



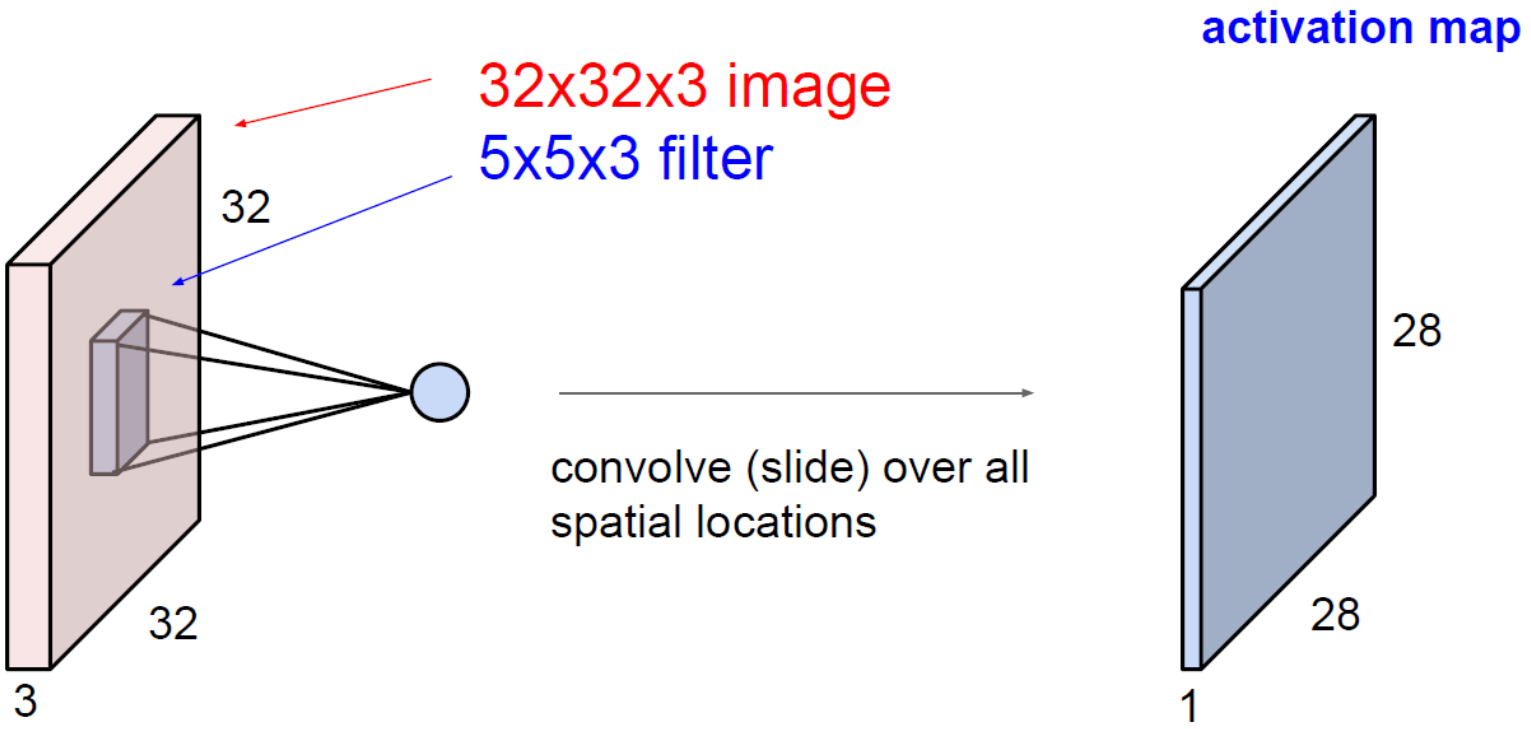
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

شبکه‌های عصبی کانوولوشنال

لایه‌ی کانوولوشن

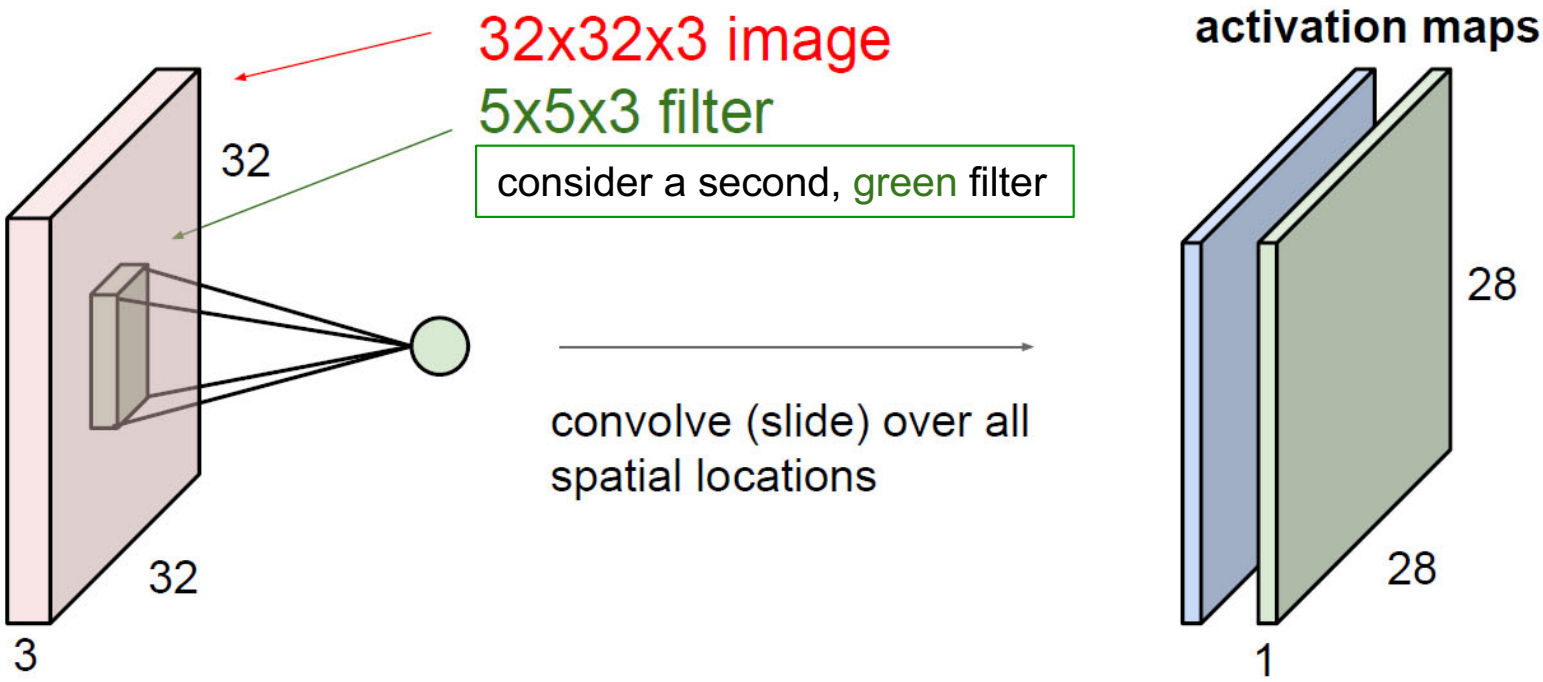
CONVOLUTION LAYER



شبکه‌های عصبی کانوولوشنال

لایه‌ی کانوولوشن

CONVOLUTION LAYER

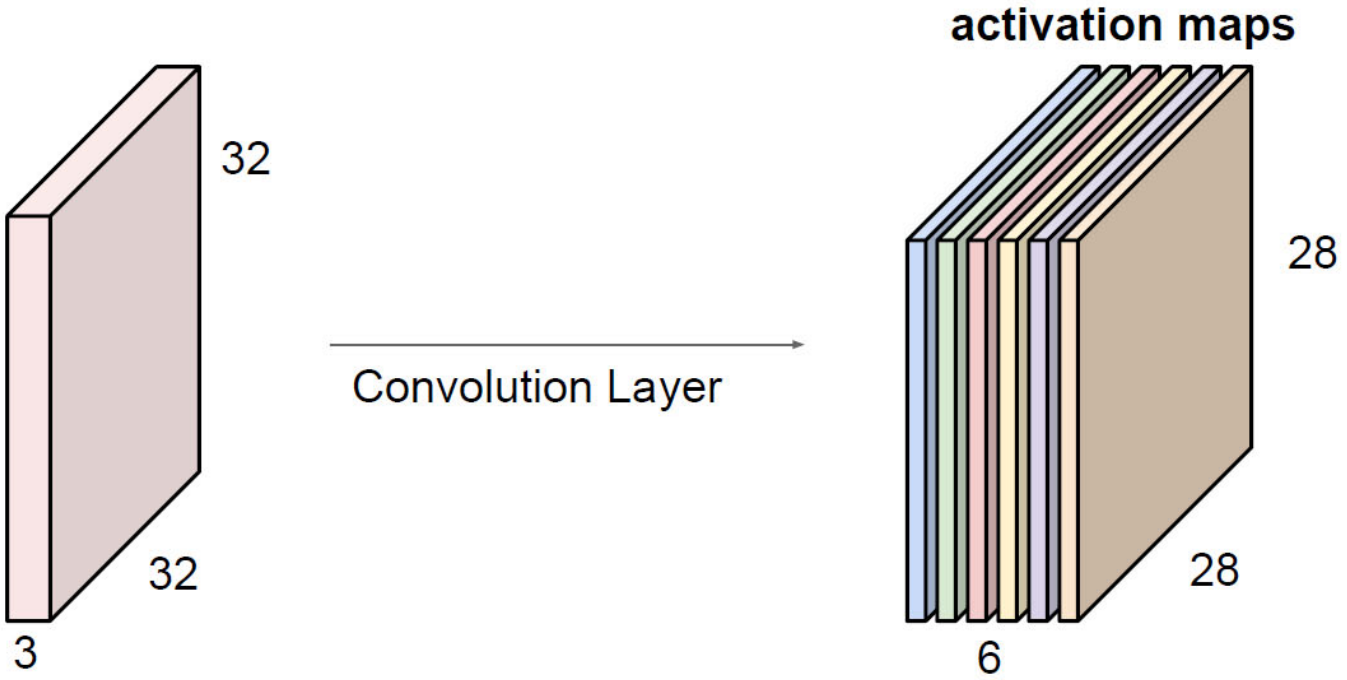


شبکه‌های عصبی کانوولوشنال

لایه‌ی کانوولوشن

CONVOLUTION LAYER

For example, if we had 6 (5x5 filters), we'll get 6 separate activation maps:

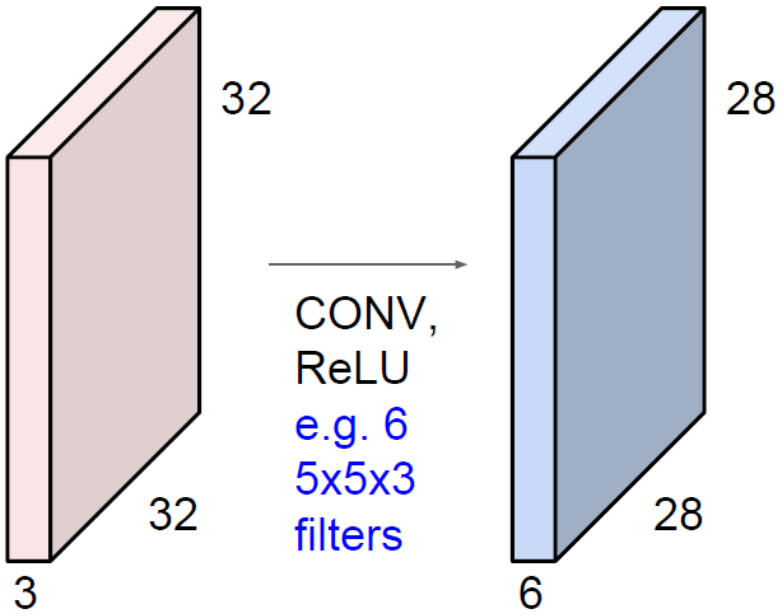


We stack these up to get a "new image" of size 28x28x6!

شبکه‌های عصبی کانوولوشنال

لایه‌ی کانوولوشن

CONVOLUTION LAYER

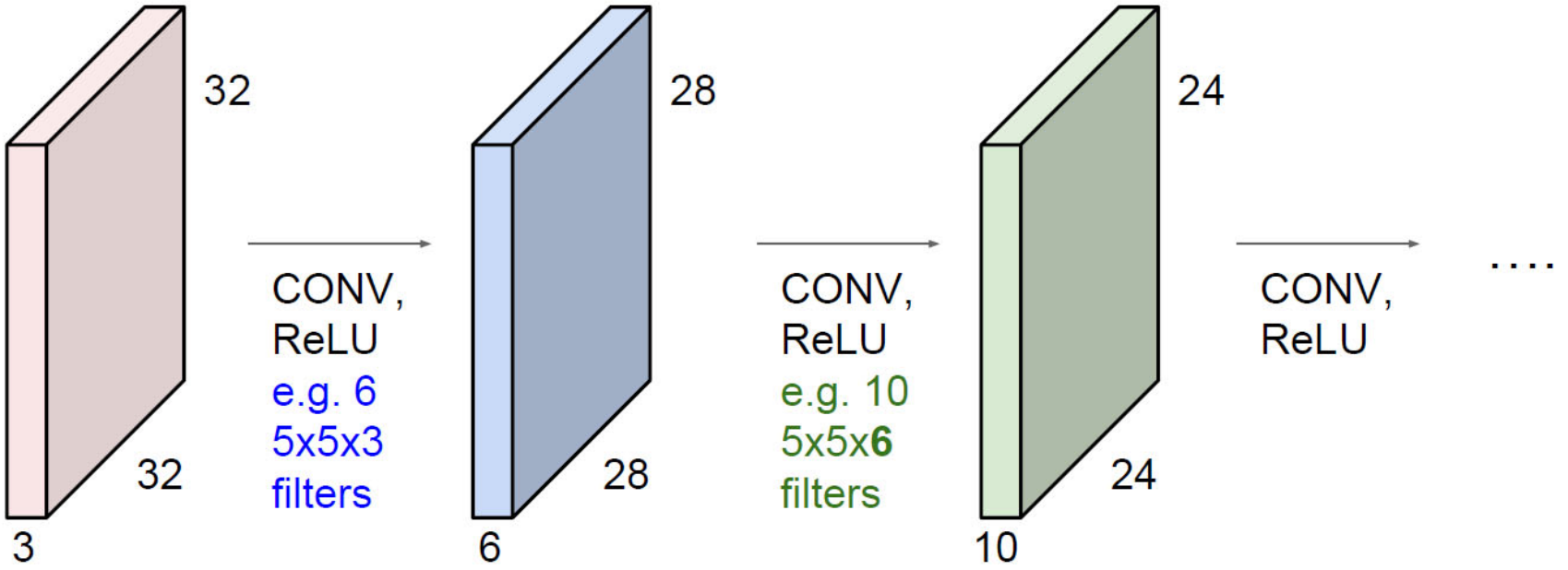


Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions

شبکه‌های عصبی کانوولوشنال

لایه‌ی کانوولوشن

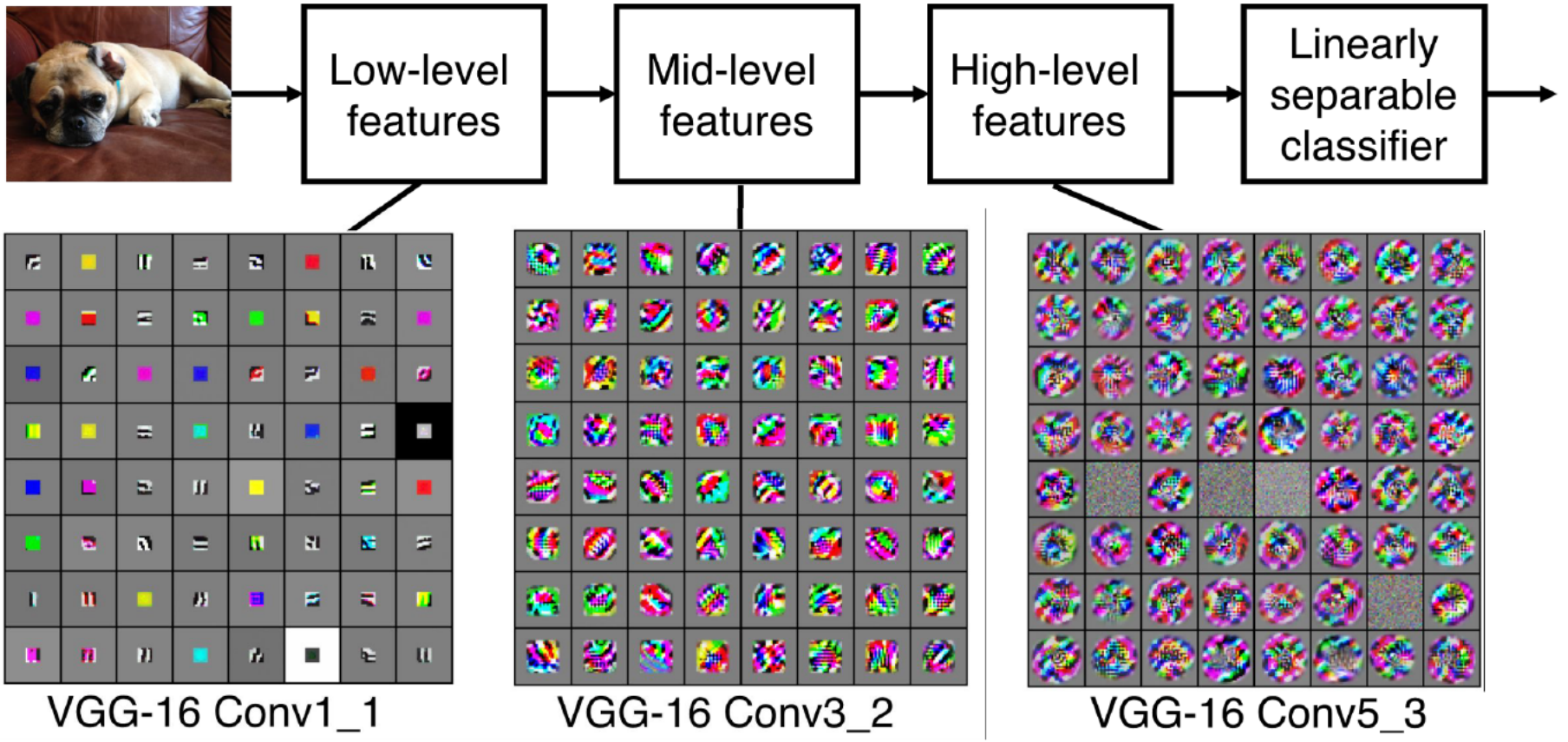
CONVOLUTION LAYER



Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions

شبکه‌های عصبی کانوولوشنال

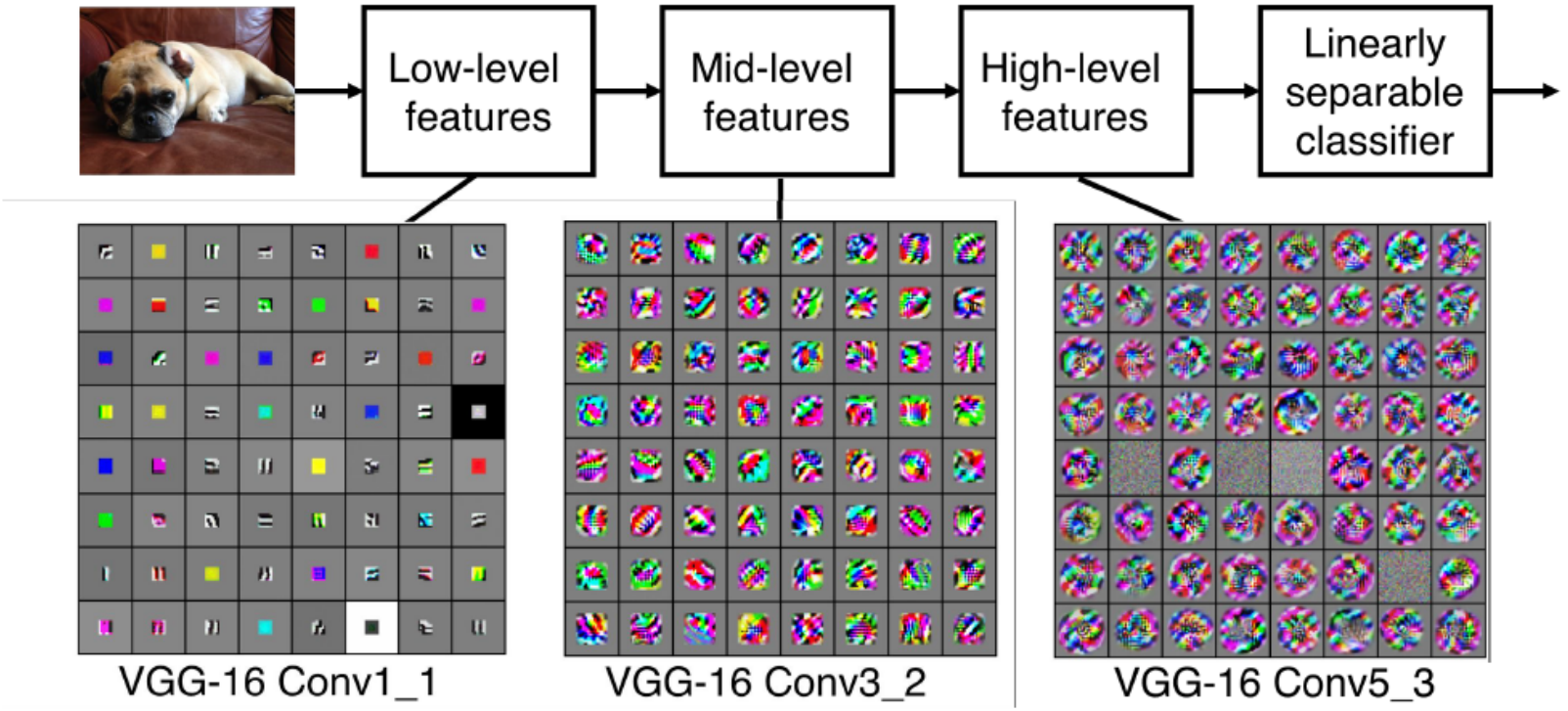
تعبیر لایه‌ها

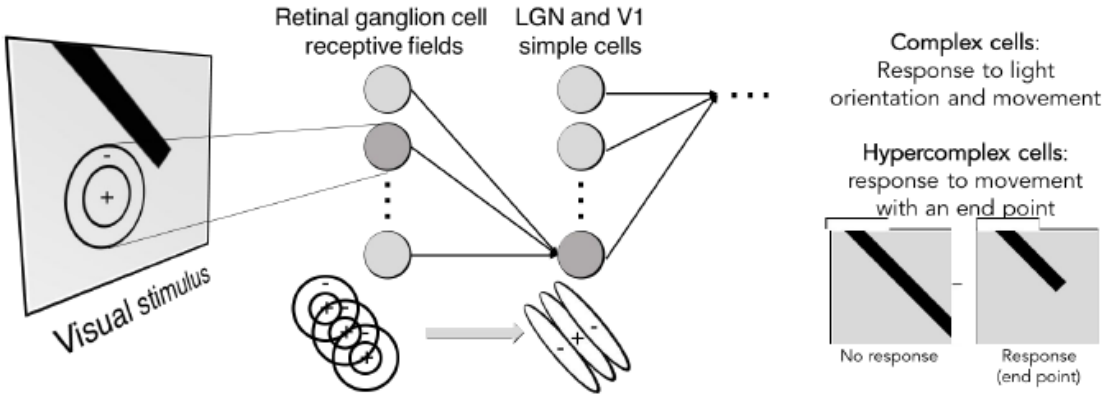


[Simonyan and Zisserman 2014]

شبکه‌های عصبی کانوولوشنال

تعبیر لایه‌ها

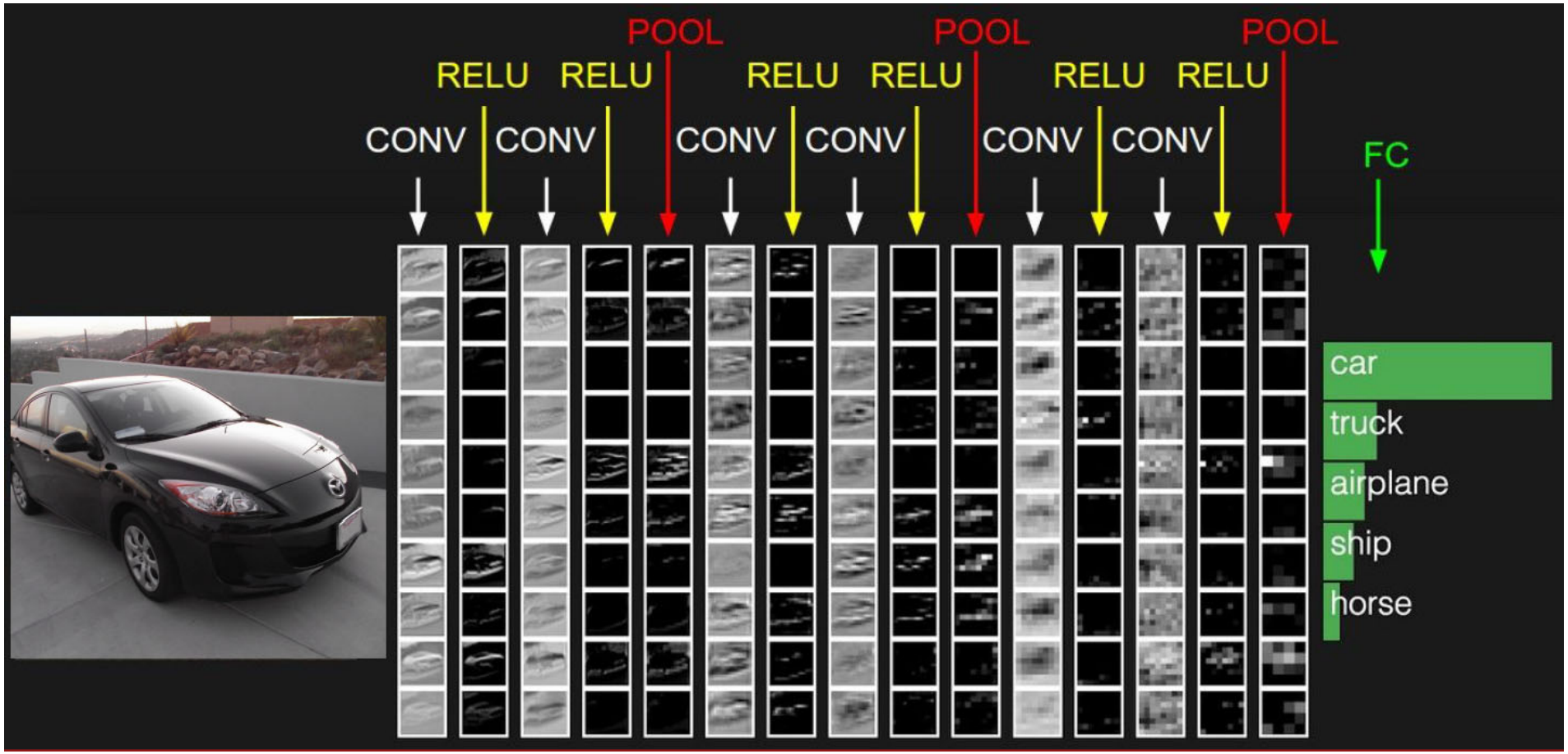




Ref: http://cs231n.stanford.edu/

شبکه‌های عصبی کانوولوشنال

تعبیر لایه‌ها



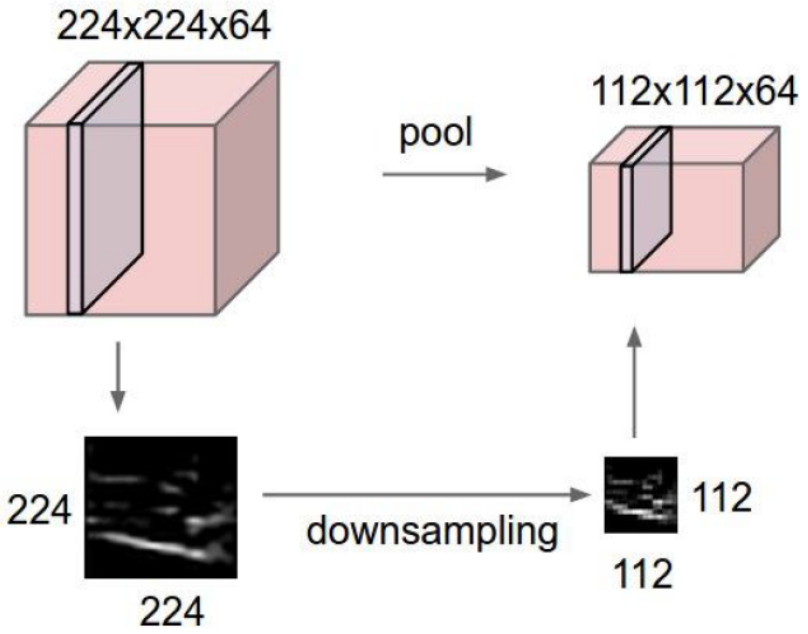
Ref: http://cs231n.stanford.edu/

شبکه‌های عصبی کانوولوشنال

لایه‌ی تلفیق

POOLING LAYER

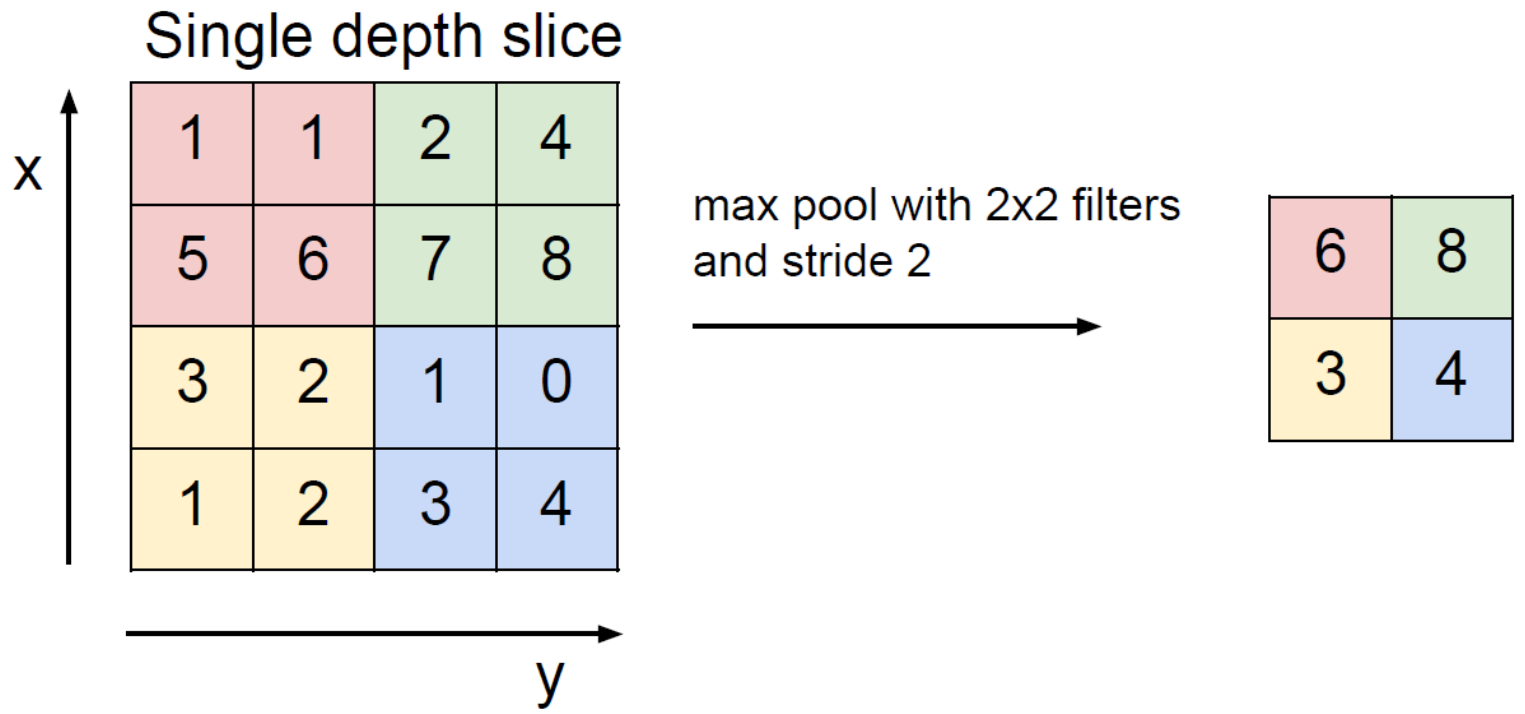
- makes the representations smaller and more manageable
- operates over each activation map independently:



شبکه‌های عصبی کانوولوشنال

لایه‌ی تلفیق : مثال

POOLING LAYER



شبکه‌های عصبی کانوولوشنال

لایه‌ی تلفیق: روش‌ها

POOLING LAYER

Max-pooling: $$h_i^n(r,c) = \max_{\bar{r}\in N(r),\ \bar{c}\in N(c)} h_i^{n-1}(\bar{r},\bar{c})$$

Average-pooling: $$h_i^n(r,c) = \operatorname{mean}_{\bar{r}\in N(r),\ \bar{c}\in N(c)} h_i^{n-1}(\bar{r},\bar{c})$$

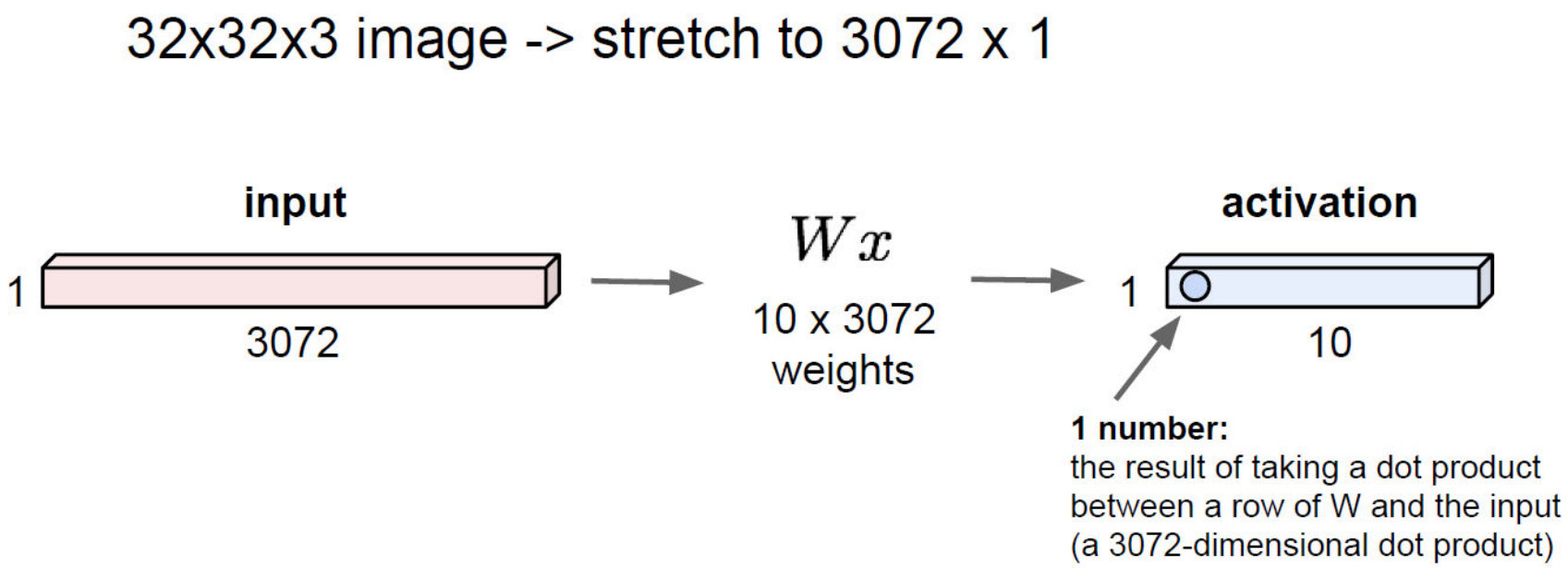
L2-pooling: $$h_i^n(r,c) = \sqrt{\sum_{\bar{r}\in N(r),\ \bar{c}\in N(c)} h_i^{n-1}(\bar{r},\bar{c})^2}$$

L2-pooling over features: $$h_i^n(r,c) = \sqrt{\sum_{j\in N(i)} h_i^{n-1}(r,c)^2}$$

شبکه‌های عصبی کانوولوشنال

لایه‌ی تماماً متصل

FULLY CONNECTED LAYER



Contains neurons that connect to the entire input volume, as in ordinary Neural Networks

[ConvNetJS demo: training on CIFAR-10]

ConvNetJS CIFAR-10 demo

Description

This demo trains a Convolutional Neural Network on the CIFAR-10 dataset in your browser, with nothing but Javascript. The state of the art on this dataset is about 90% accuracy and human performance is at about 94% (not perfect as the dataset can be a bit ambiguous). I used this python script to parse the original files (python version) into batches of images that can be easily loaded into page DOM with img tags.

This dataset is more difficult and it takes longer to train a network. Data augmentation includes random flipping and random image shifts by up to 2px horizontally and verically.

By default, in this demo we're using Adadelta which is one of per-parameter adaptive step size methods, so we don't have to worry about changing learning rates or momentum over time. However, I still included the text fields for changing these if you'd like to play around with SGD+Momentum trainer.

Report questions/bugs/suggestions to @karpathy.

http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

یادگیری عمیق

شبکه‌های عصبی کانوولوشنال

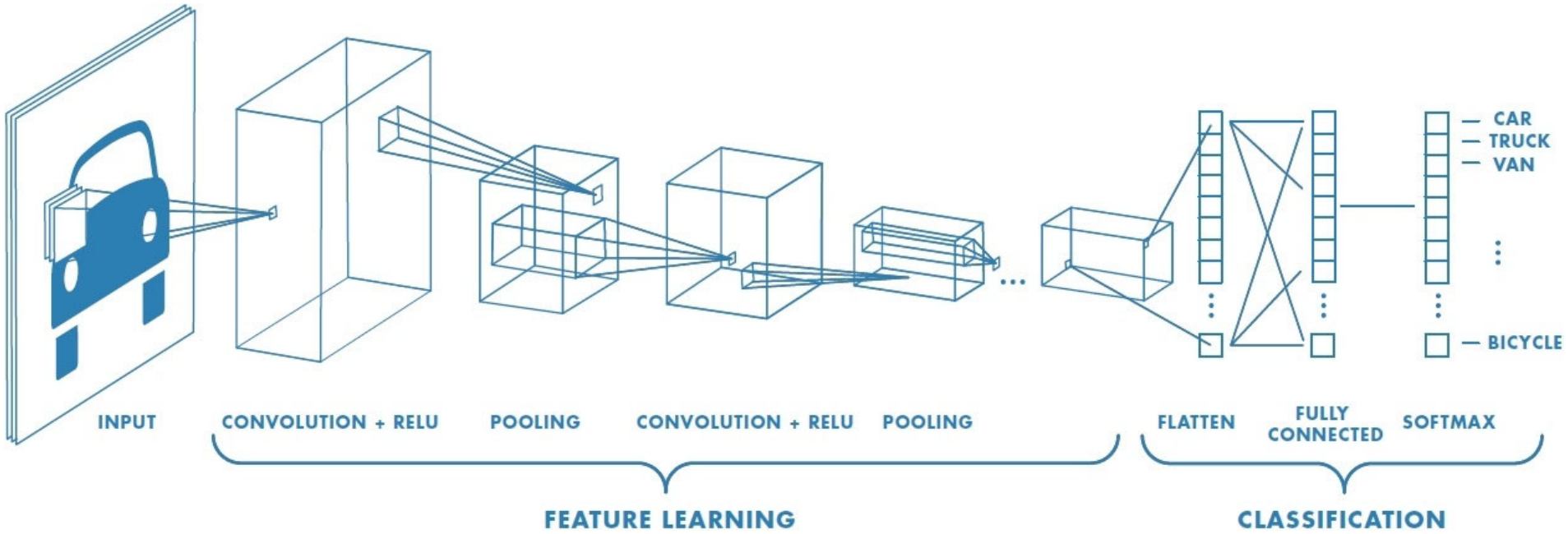
۲

نمونه‌ها

شبکه‌های عصبی کانوولوشنال

معماری

ARCHITECTURE

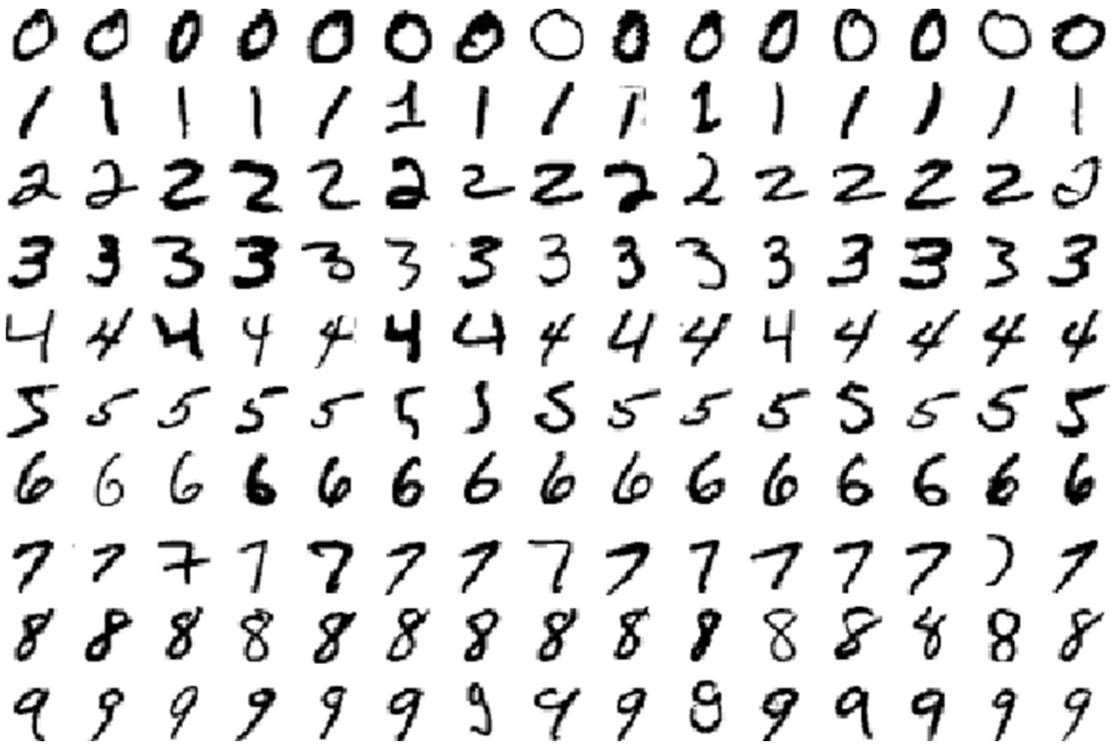


https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html

شبکه‌های عصبی کانوولوشنال برای بازشناسی ارقام دست‌نویس

CONVOLUTIONAL NETWORKS FOR OCR

- **Task:** Correctly recognize the image of hand-written digits.
- **Dataset:** MNIST (28*28 image), 60000 Training samples, 10000 Test.



شبکه‌های عصبی کانوولوشنال برای بازشناسی ارقام دست‌نویس

LeNet-5

CONVOLUTIONAL NETWORKS FOR OCR

- Gradient-based learning applied to document recognition
  [LeCun, Bottou, Bengio, Haffner, 1998]

- Three key ideas:
  - Local Receptive Fields (Convolution, Filters),
  - Shared Weights,
  - Sub-sampling (Pooling).

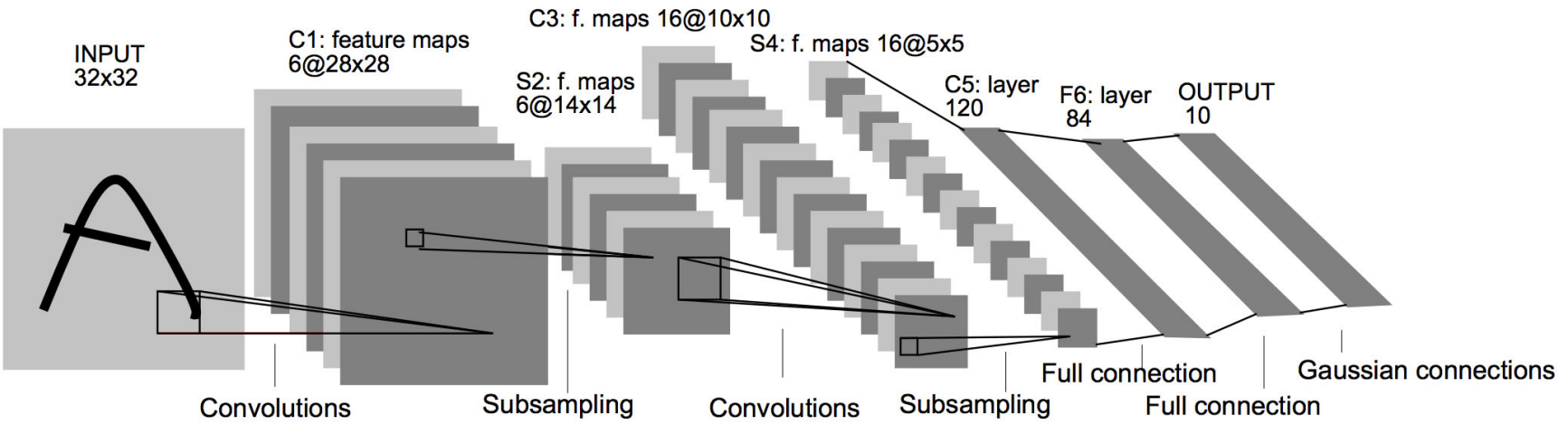


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

شبکه‌های عصبی کانوولوشنال برای بازشناسی ارقام دست‌نویس

LeNet-5

INPUT

- **Input**: 32 × 32 pixel image.
- Largest character is 20 × 20 (All important info should be in the center of the receptive field of the highest level feature detectors)
- Black and White pixel values are normalized:
  E.g. White = –0.1, Black =1.175 (Mean of pixels = 0, Std of pixels =1)

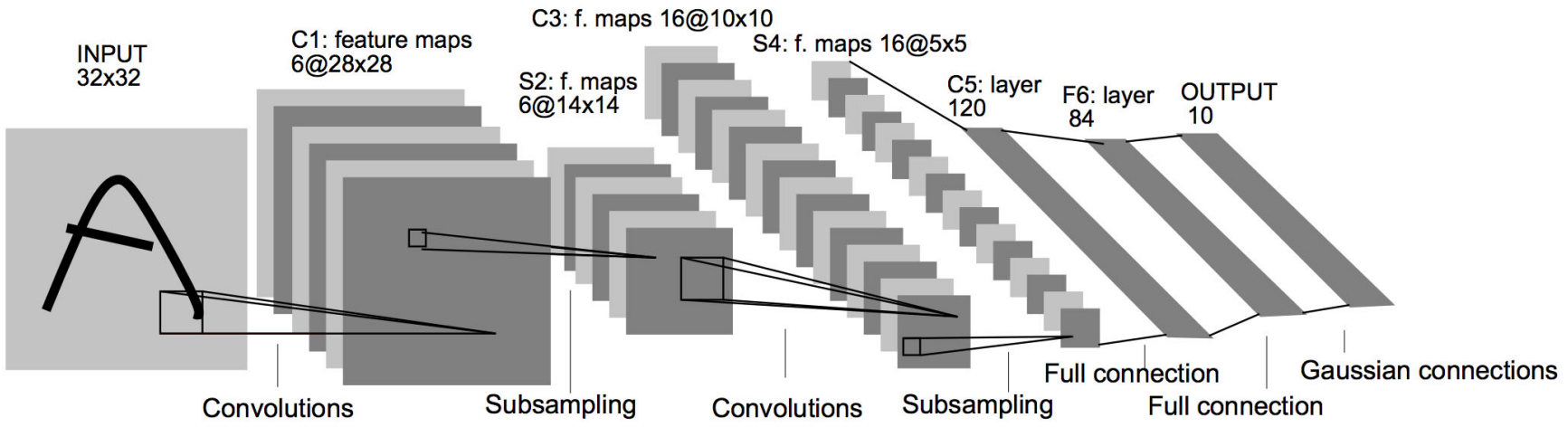


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

شبکه‌های عصبی کانوولوشنال برای بازشناسی ارقام دست‌نویس

LeNet-5

LAYER C1

- C1: Convolutional layer with 6 feature maps of size 28×28. $C1_k$ ($k = 1..6$)
  - Formula for output size = $(N - F + 2P) / S + 1$
  - N: input size, F: size of the filter, P: Padding, S: Stride
- Each unit of C1 has a 5×5 receptive field in the input layer.
- $(5\times5 + 1)\times6 = 156$ parameters to learn
- Connections: $28\times28\times(5\times5 + 1)\times6 = 122304$
- If it was fully connected we had $(32\times32 + 1)\times(28\times28)\times6 = 4{,}821{,}600$ parameters

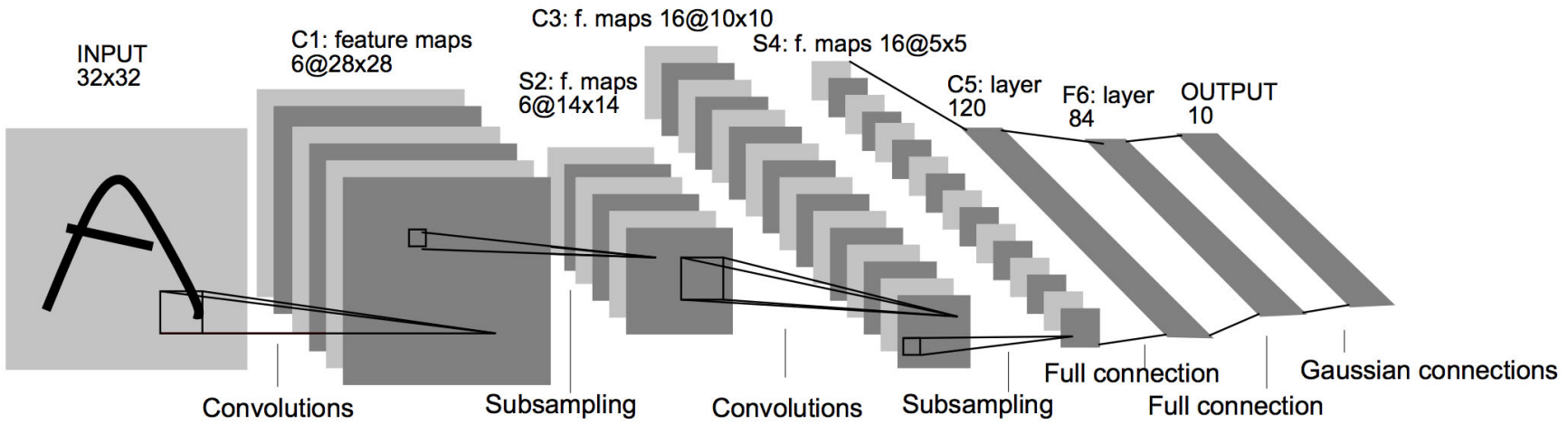


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

شبکه‌های عصبی کانوولوشنال برای بازشناسی ارقام دست‌نویس

LeNet-5

LAYER S2

- S2: Subsampling layer with 6 feature maps of size 14×14, 2×2 non overlapping receptive fields in C1 Layer
- Averages the input, multiplies by a coefficient and adds a bias.
  - Both these coefficient and bias are learnt.
- S2: $6\times2 = 12$ trainable parameters.
- Connections: $14\times14\times(2\times2 + 1)\times6 = 5880$

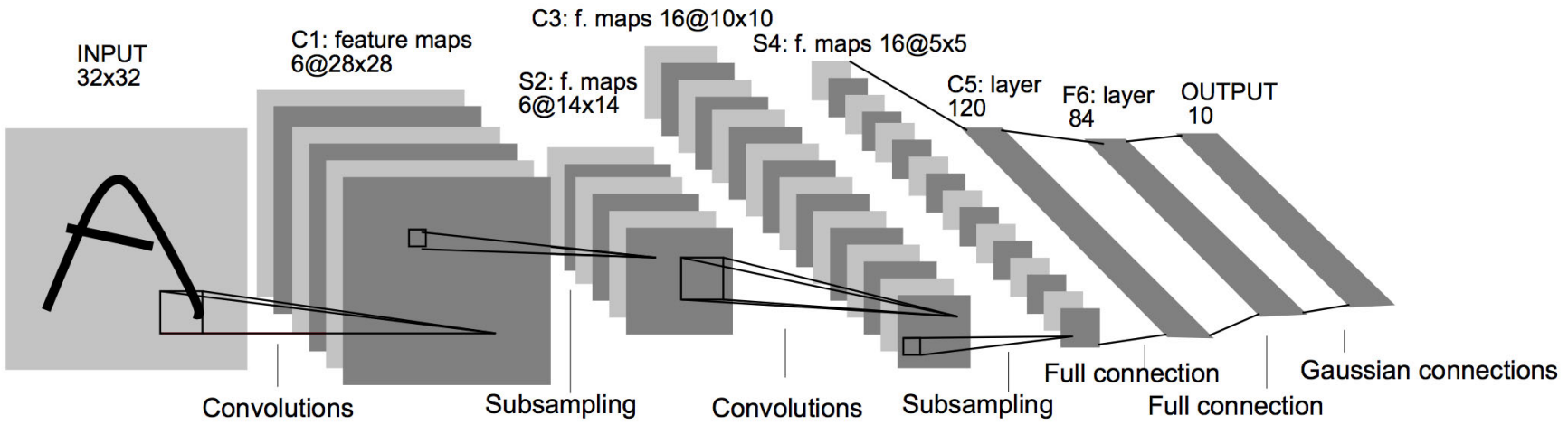


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

شبکه‌های عصبی کانوولوشنال برای بازشناسی ارقام دست‌نویس

LeNet-5

LAYER C3

- C3: Convolutional layer with 16 feature maps of size 10×10
- Each unit in C3 is connected to several! 5×5 receptive fields at identical locations in S2
- Layer C3: 1516 trainable parameters.
- Connections: 151600

	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.

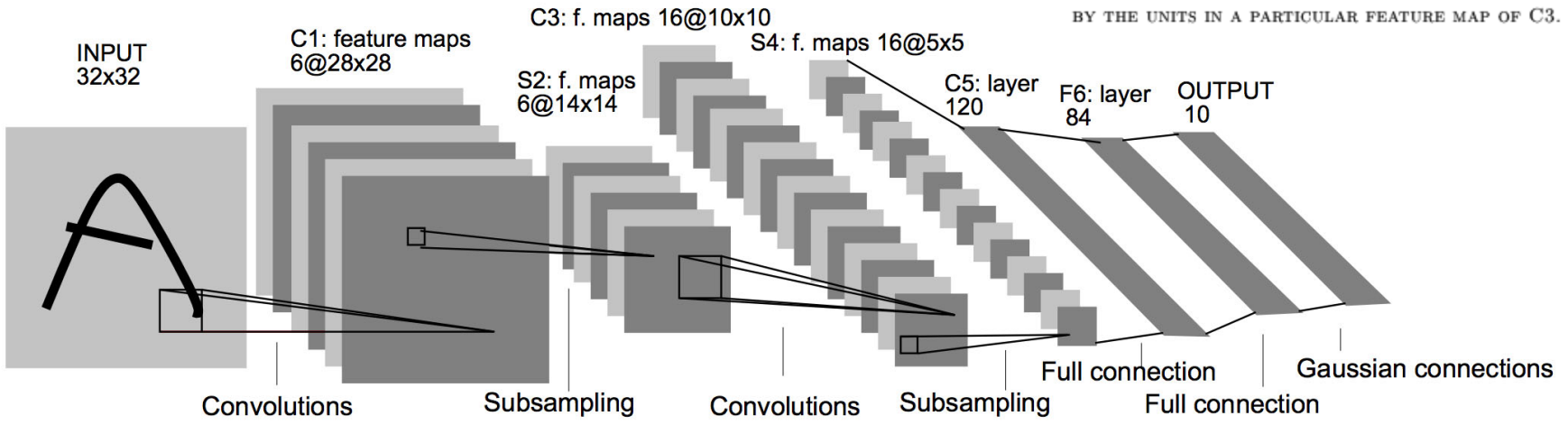


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

شبکه‌های عصبی کانوولوشنال برای بازشناسی ارقام دست‌نویس

LeNet-5

LAYER S4

- S4: Subsampling layer with 16 feature maps of size 5×5
- Each unit in S4 is connected to the corresponding 2×2 receptive field at C3
- Layer S4: 16×2 = 32 trainable parameters.
- Connections: 5×5×(2×2+1)×16 = 2000

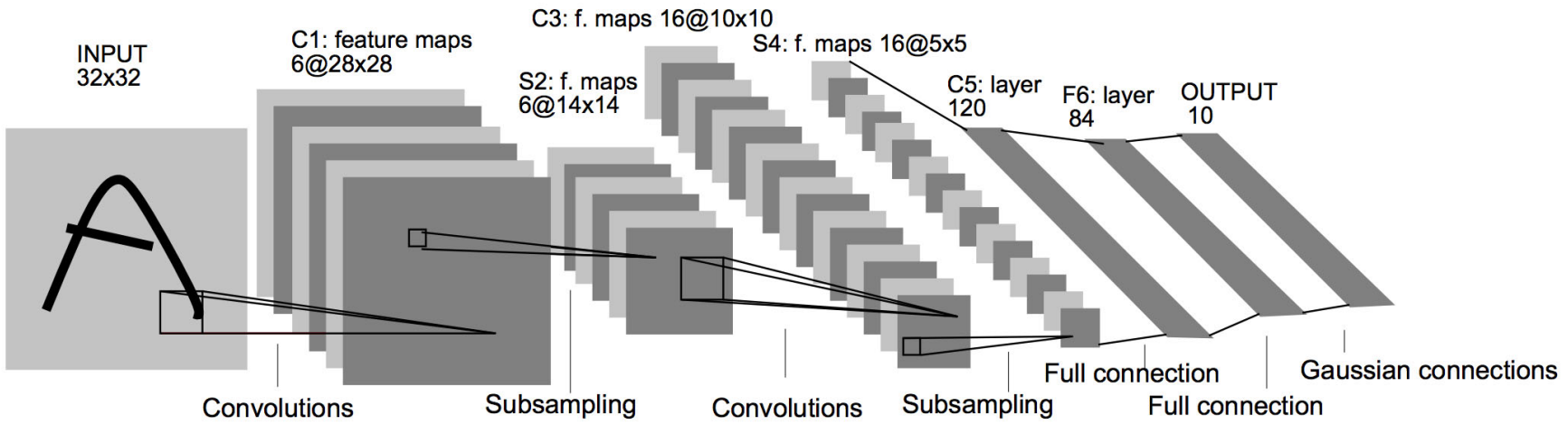


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeNet-5

LAYER C5

- C5: Fully connected layer with 120 feature maps of size 1×1
- Each unit in C5 is connected to all 16, 5×5 receptive fields in S4
- Layer C5: $120 \times (16 \times 25+1) = 48120$ trainable parameters and connections (Fully connected)

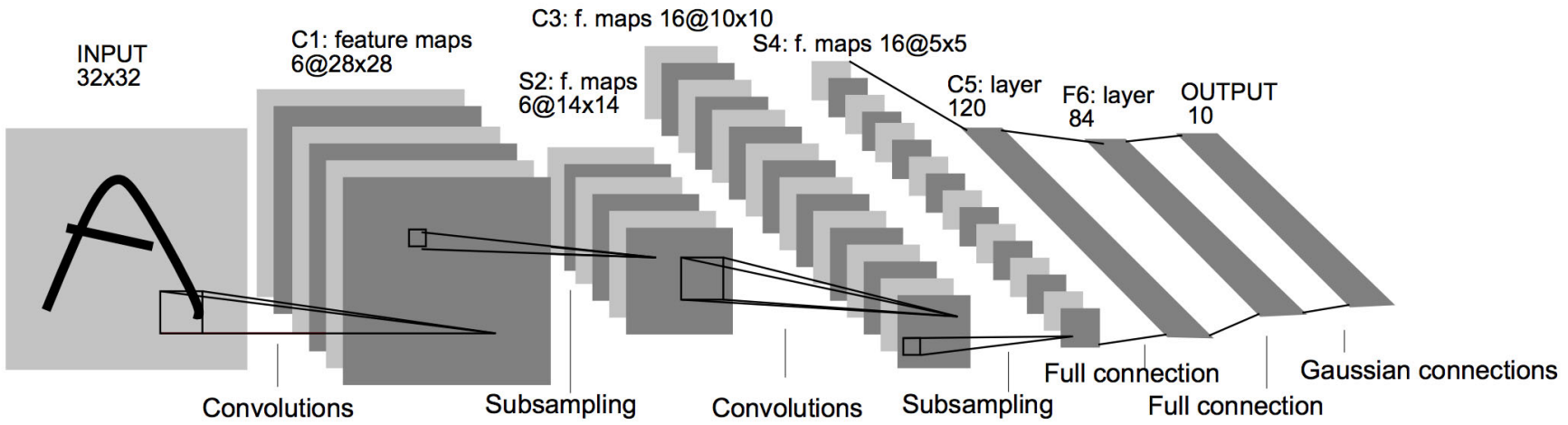


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

شبکه‌های عصبی کانوولوشنال برای بازشناسی ارقام دست‌نویس

LeNet-5

LAYER F6

- F6: 84 fully connected units.
  $84 \times (120 + 1) = 10164$ trainable parameters and connections.
- Output layer: 10 RBF (One for each digit) $84 = 7 \times 12$, stylized image
- Weight update: Backpropagation

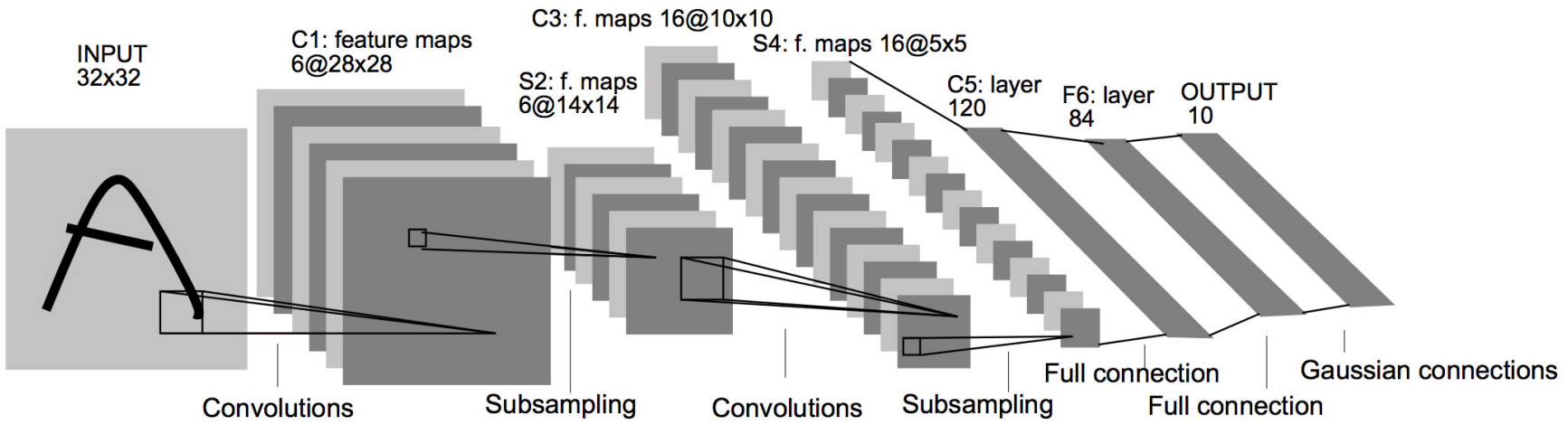


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

شبکه‌های عصبی کانوولوشنال برای بازشناسی ارقام دست‌نویس

LeNet-5

RESULTS

- LeNet achieves 0.8% error rate on test set.

- SVM got 1.1%, a slightly modified version known as ν-SVM got 0.8%.
  but the computation cost is very high. It is almost twice as expensive as LeNet.
- LeNet was trained for 2-3 days on CPU (200 MHz processor).
- Currently it can be trained in less than 30 minutes.

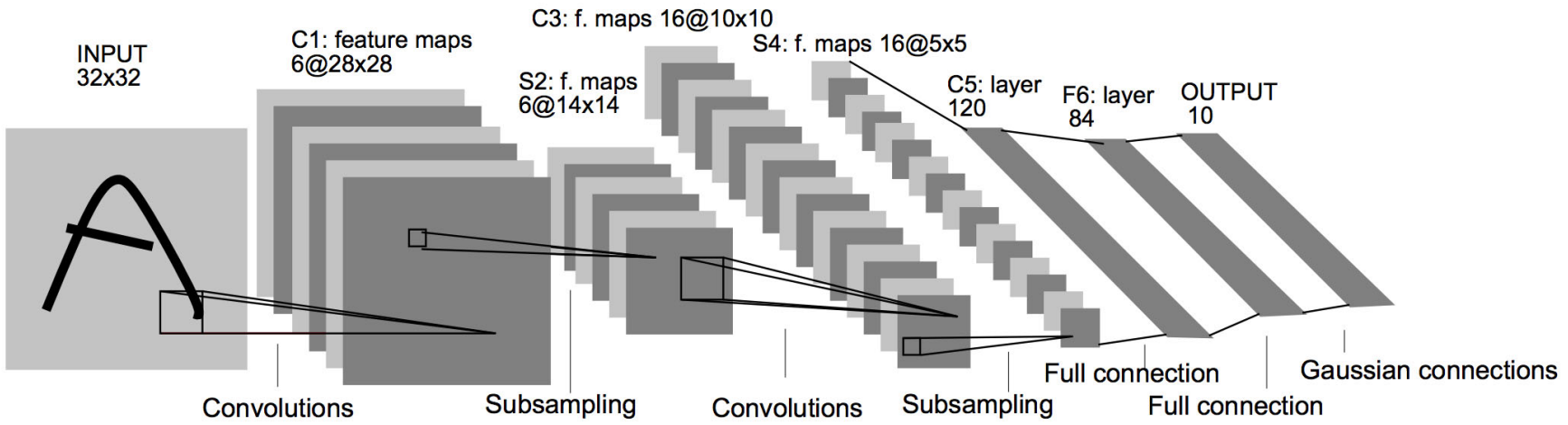


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

CONVOLUTIONAL NETWORKS FOR IMAGE CLASSIFICATION

هدف: بازشناسی اشیای موجود در یک تصویر

Tags	Confidence
racer, race car, racing car	0.7962
sports car, sport car	0.1341
cab, hack, taxi, taxicab	0.0278
convertible	0.0165
car wheel	0.0080

شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

IMAGENET

- Over 15M labeled high resolution images.
- Roughly 22K categories
- Collected from web and labeled by Amazon Mechanical Turk.

شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

IMAGENET LARGE SCALE VISUAL RECOGNITION CHALLENGE (ILSVRC)

- Annual competition of image classification at large scale.
- 1.2M training images in 1K categories.
- 50K validation images, 150K testing images.
- Classification: make 1 (Top-1 error) /5 (Top-5 error) guesses about the image label.

شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

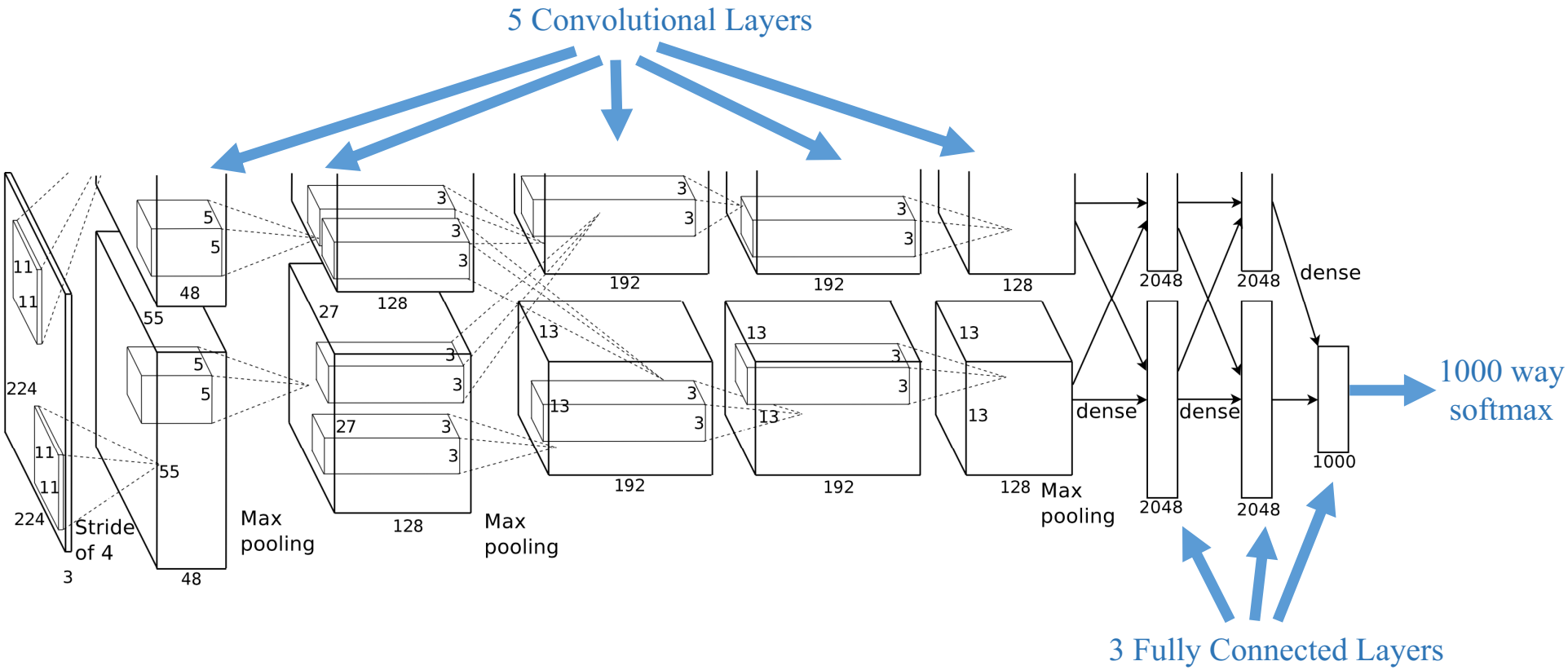
ALEXNET

- Similar framework to LeCun’98 but,
- Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
- More data (10^6 vs. 10^3 images)
- GPU implementation (50x speedup over CPU)
- Trained on two GPUs for a week
- Better regularization for training (DropOut)

شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

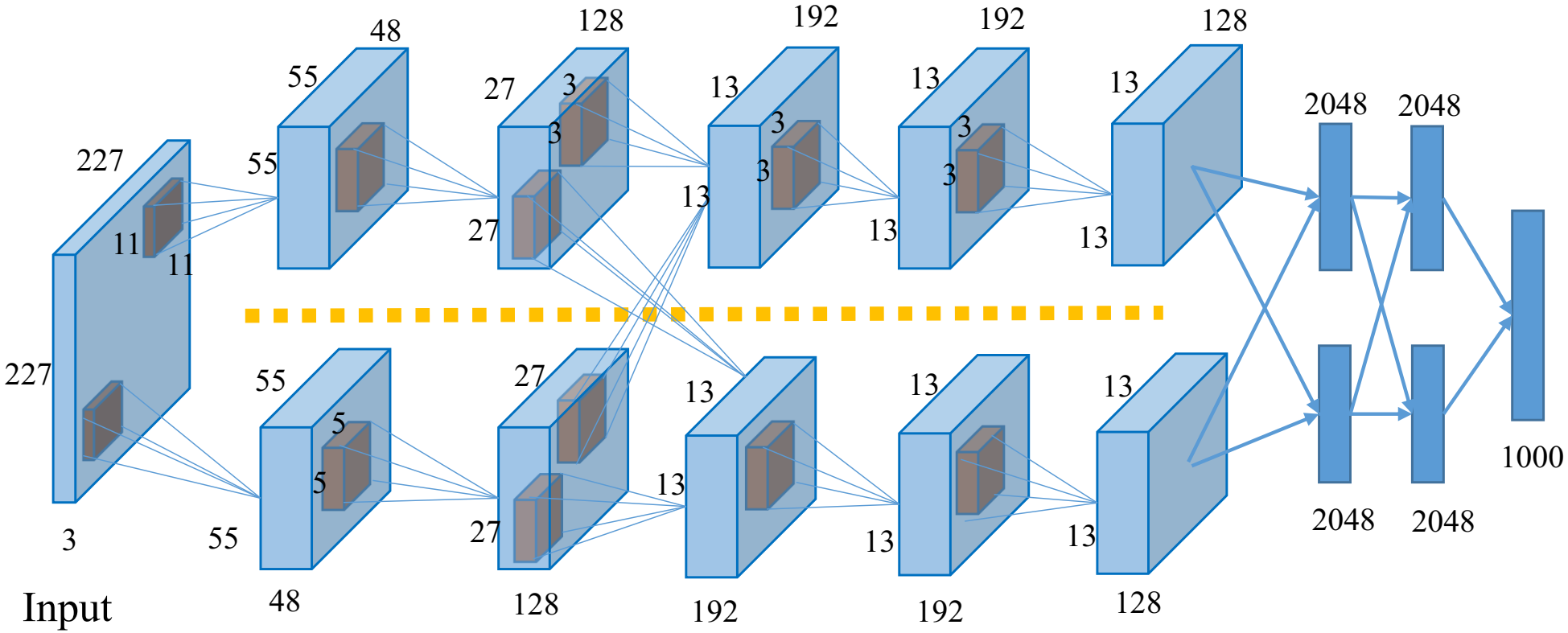
AlexNet: Architecture



شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

AlexNet: Architecture

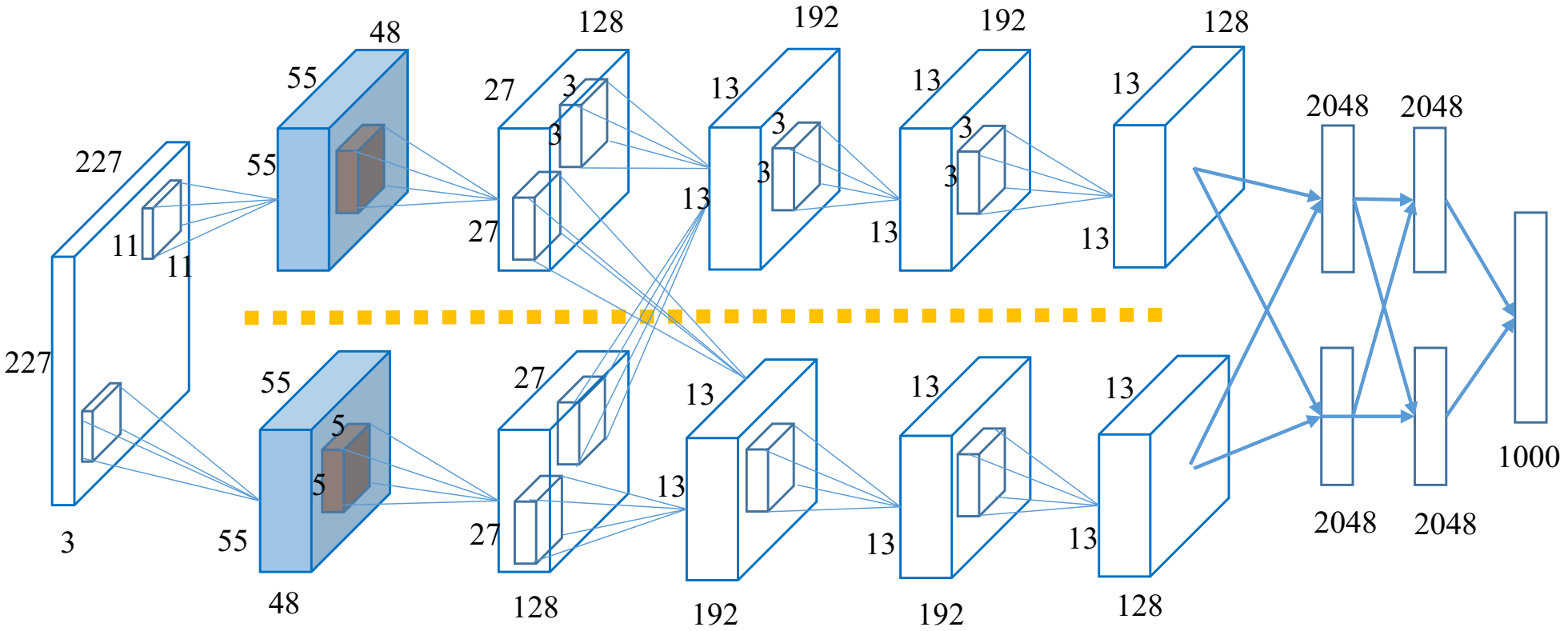


شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

AlexNet: Architecture

- $55 \times 55 \times 96 = 290{,}400$ neurons, each having $11 \times 11 \times 3 = 363$ weights + 1 bias
- $290400 \times 364 = 105{,}705{,}600$ parameters in first layer alone if fully connected.



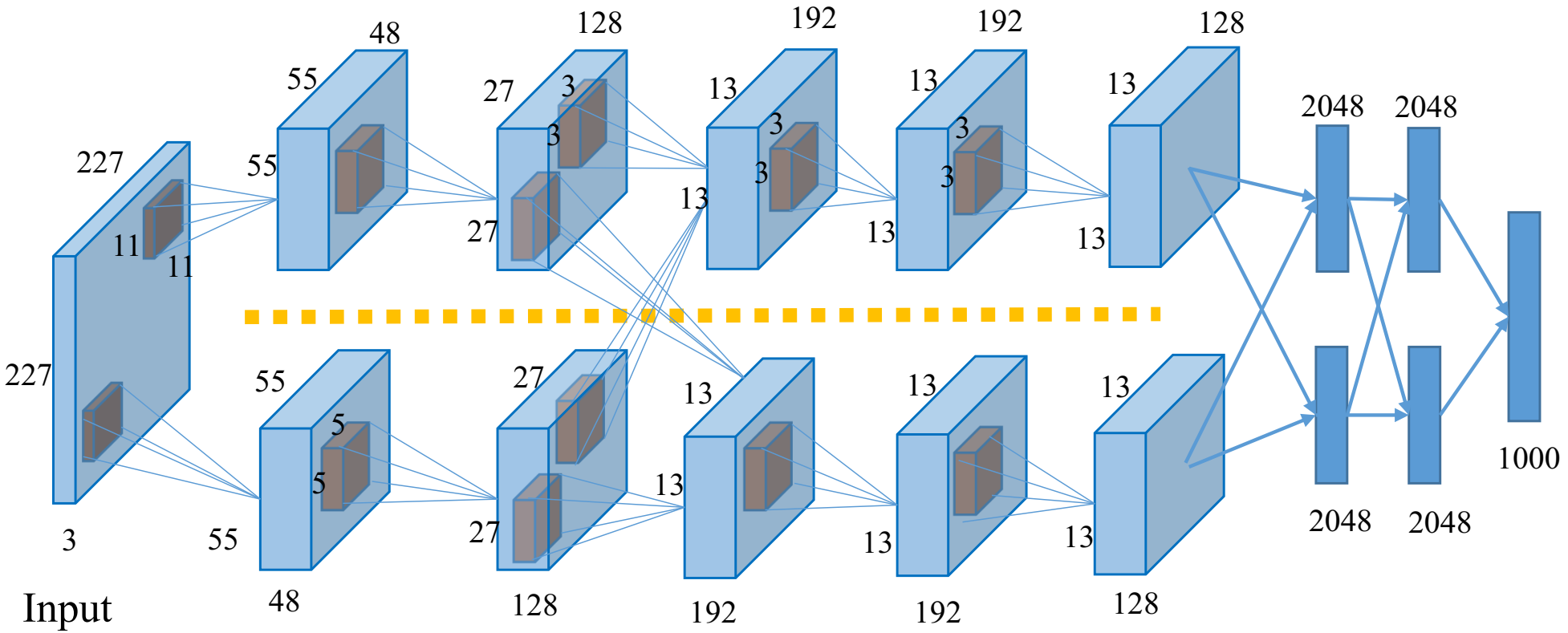
شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

AlexNet: Architecture

for layer details refer to:

https://github.com/BVLC/caffe/blob/master/models/bvlc_alexnet/deploy.prototxt

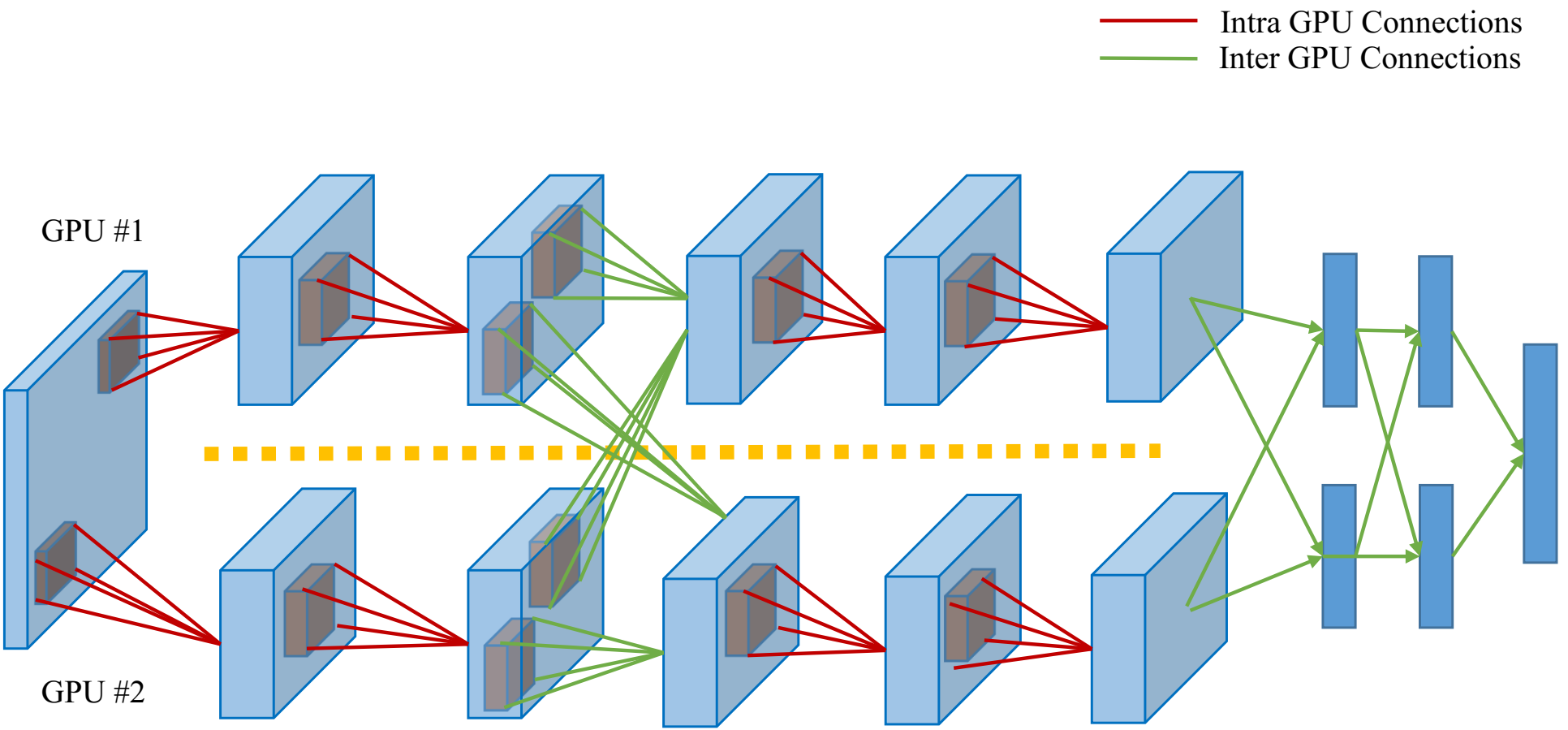


شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

AlexNet: Architecture

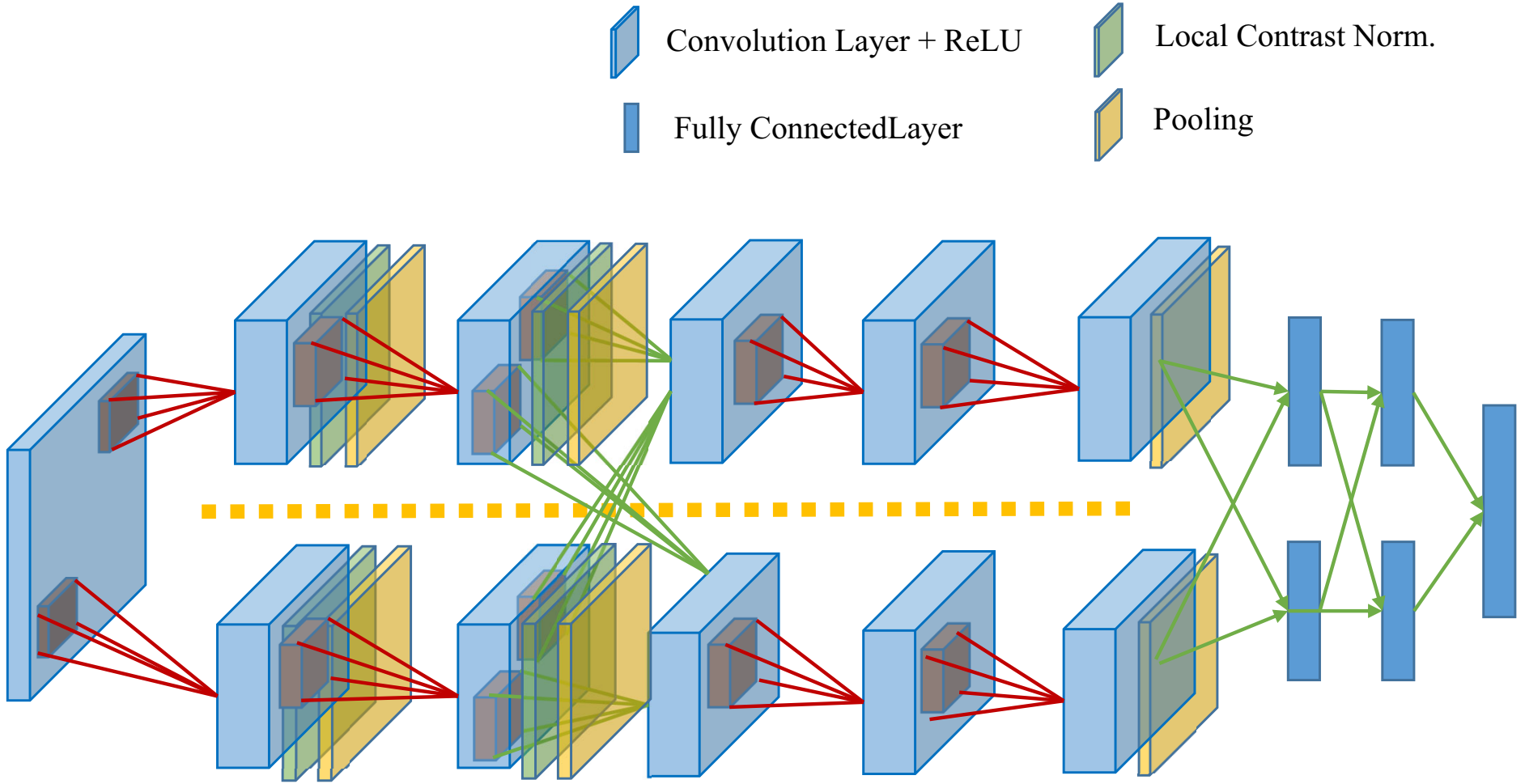
Top-1 and Top-5 error rates decreases by 1.7% and 1.2% respectively, comparing to the net trained with one GPU and half neurons



شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

ALEXNET: ARCHITECTURE

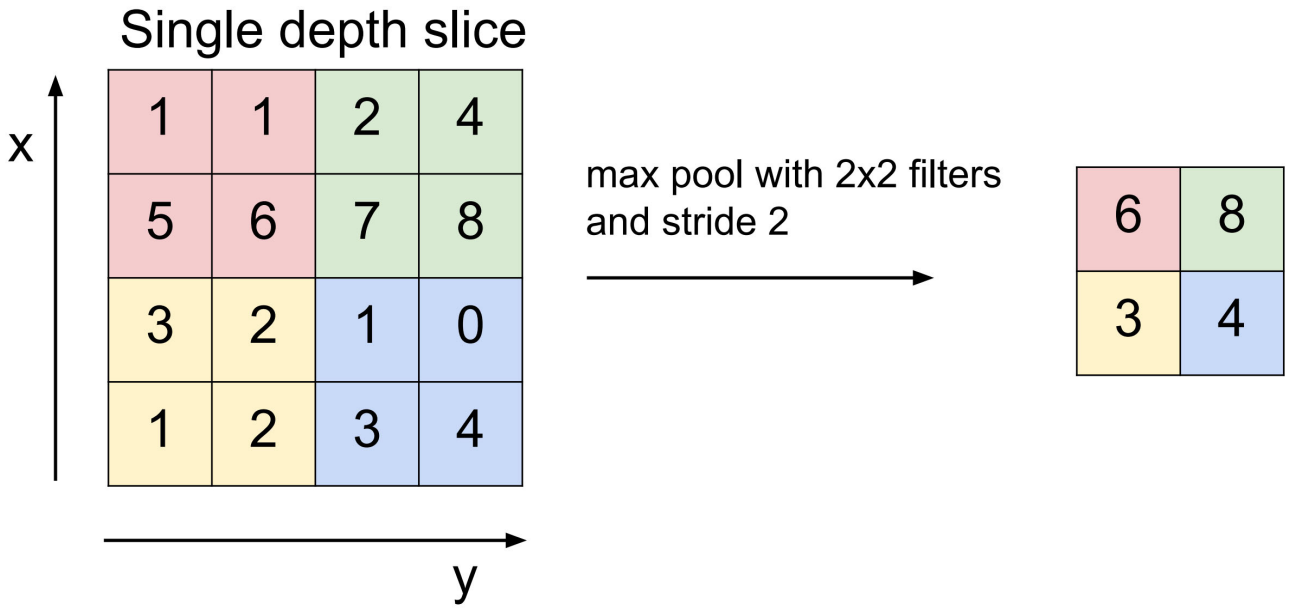


شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

AlexNet: Max Pooling

Makes the representation smaller and more manageable without loosing too much information.



شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

AlexNet: Data Augmentation

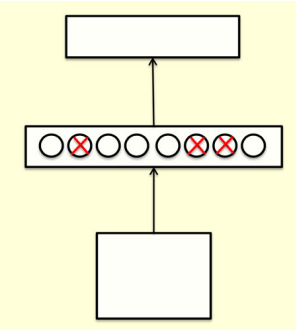
- Easiest way to reduce overfitting on image data is to artificially enlarge the dataset using label-preserving transformations.

- Two forms of data augmentation:
  - Image Translation and Horizontal Reflection.
  - Changing RGB intensities.

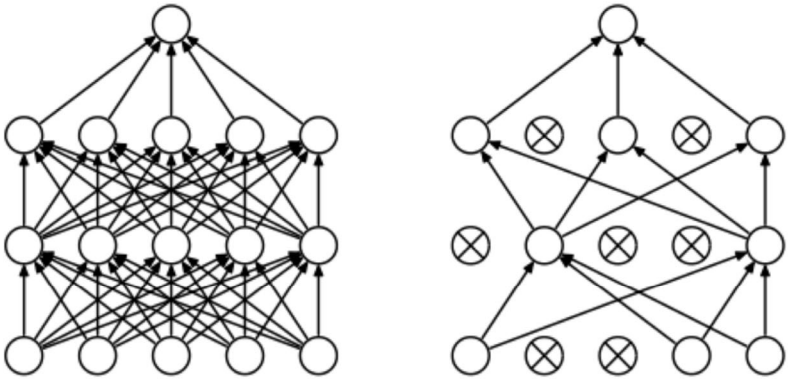
شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

AlexNet: Dropouts : An efficient way to average many large neural nets



- Consider a neural net with one hidden layer.
- Each time we present a training example,
  we randomly omit each hidden unit with probability 0.5.
- Equivalent to randomly sampling from 2^h different units.
- All architectures share weights.
- We sample from 2^h models i.e. only a few of the models ever get trained.
  They only get one training example
- Due to sharing of weights, the model is strongly regularized.
  - Pulls the weights towards what other models want.
  - Better than L2 and L1 that pull weights towards zero.



Standard Neural Net

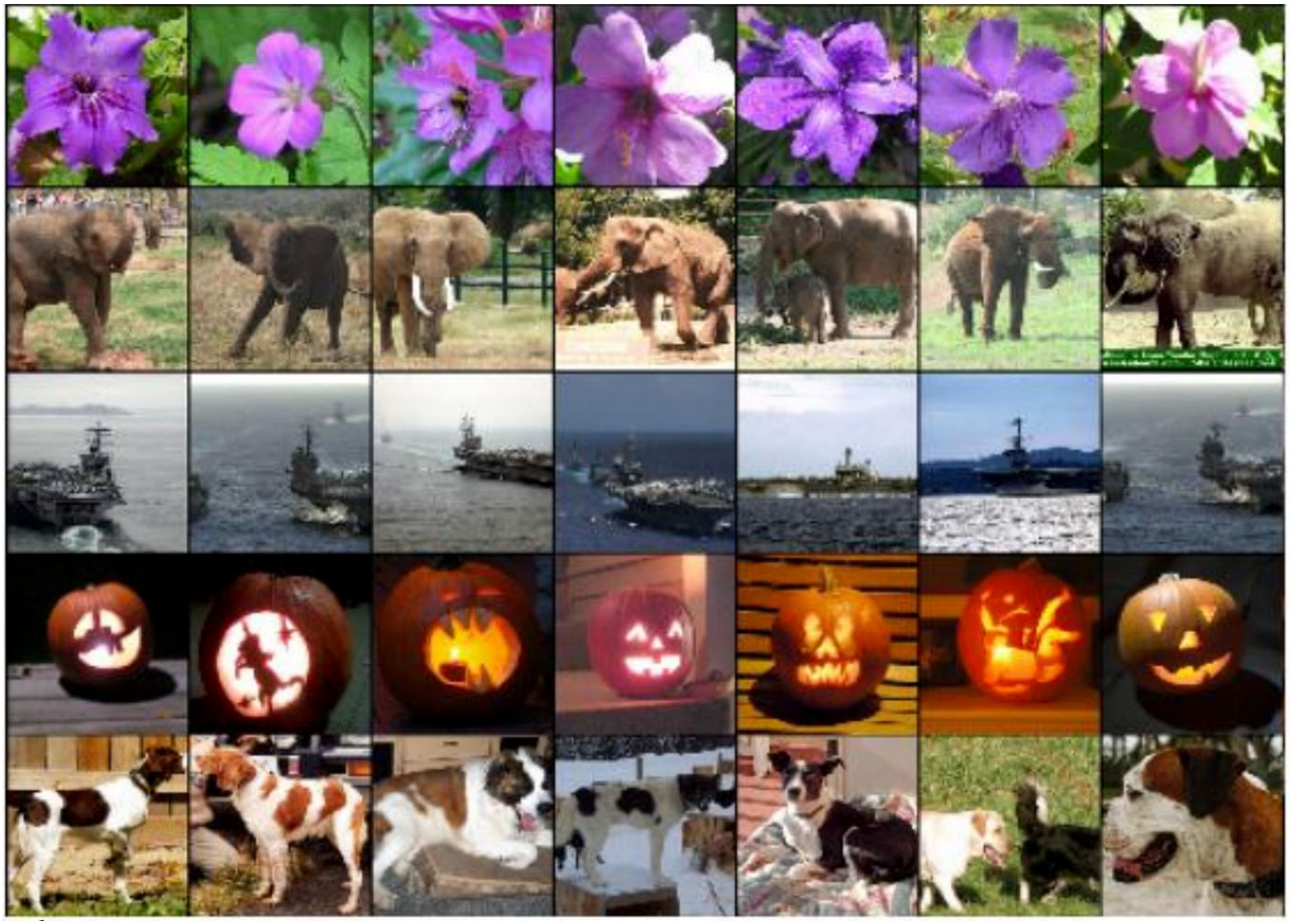
After applying dropout.

شبکه‌های عصبی کانوولوشنال برای طبقه‌بندی تصاویر

ImageNet

AlexNet: Results

six training images that produce feature vectors in the last hidden layer
with the smallest Euclidean distance from the feature vector for the test image.



یادگیری عمیق

شبکه‌های عصبی کانوولوشنال

۳

یادگیری انتقالی با CNN

یادگیری انتقالی با CNN

TRANSFER LEARNING WITH CNN

یادگیری انتقالی *Transfer Learning*	انتقال مدل یادگیری شده برای یک وظیفه برای انجام وظیفه‌ای دیگر

می‌توان یک شبکه‌ی عصبی کانوولوشنال که برای یک وظیفه‌ی خاص آموزش دیده است
را با مقدار کمی تغییر برای یک وظیفه‌ی دیگر استفاده کرد.

Transfer Learning

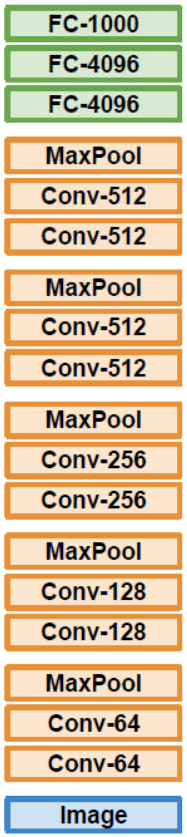
"You need a lot of a data if you want to train/use CNNs"

Ref: http://cs231n.stanford.edu/

Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

1. Train on Imagenet

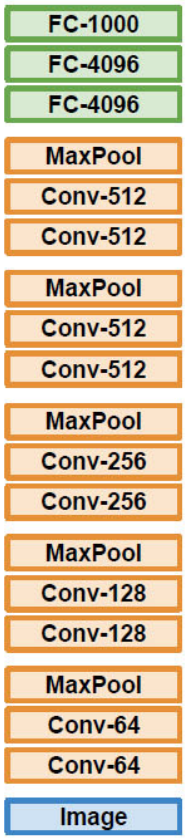


Ref: http://cs231n.stanford.edu/

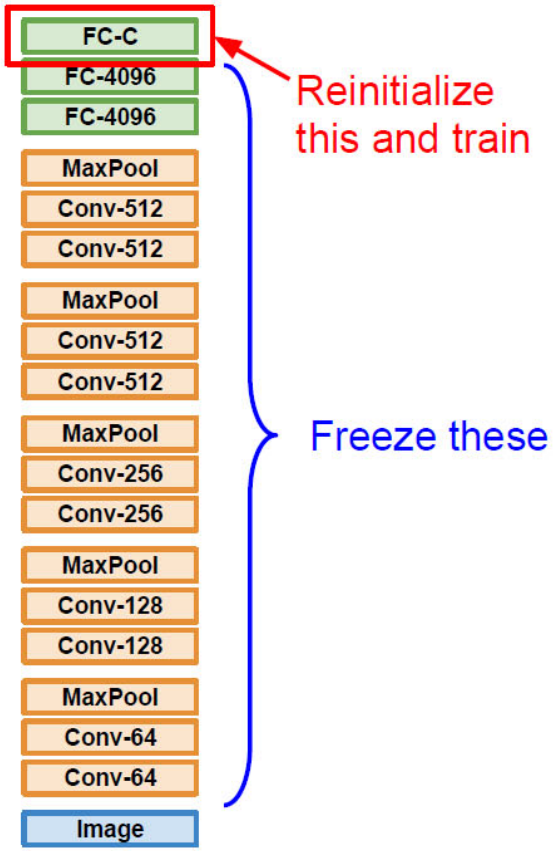
Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

1. Train on Imagenet



2. Small Dataset (C classes)

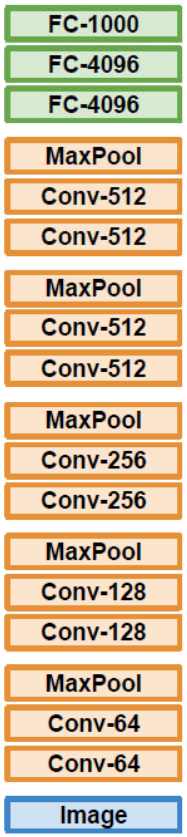


Ref: http://cs231n.stanford.edu/

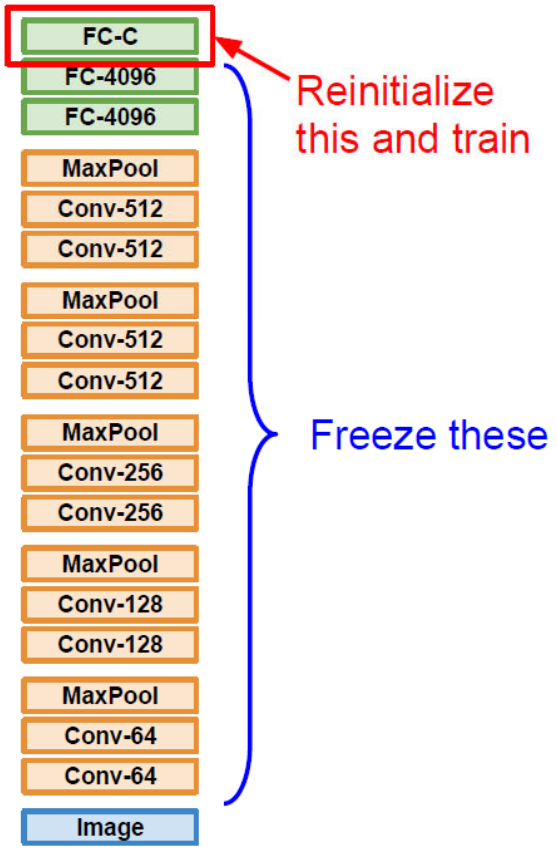
Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

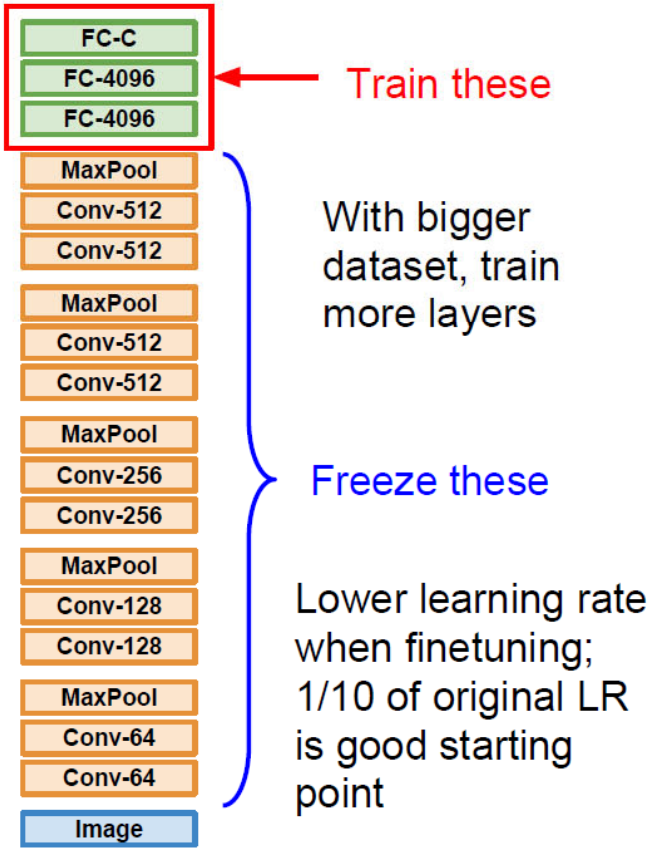
1. Train on Imagenet



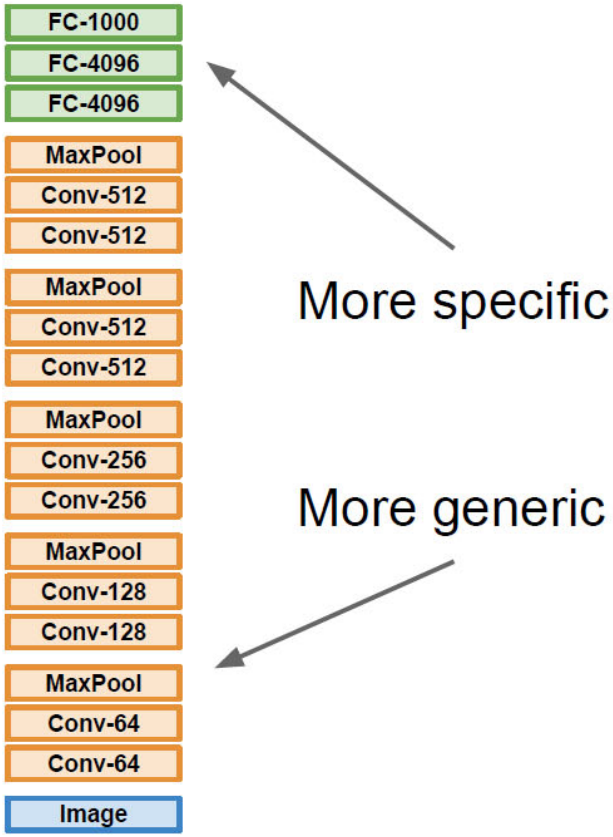
2. Small Dataset (C classes)



3. Bigger dataset

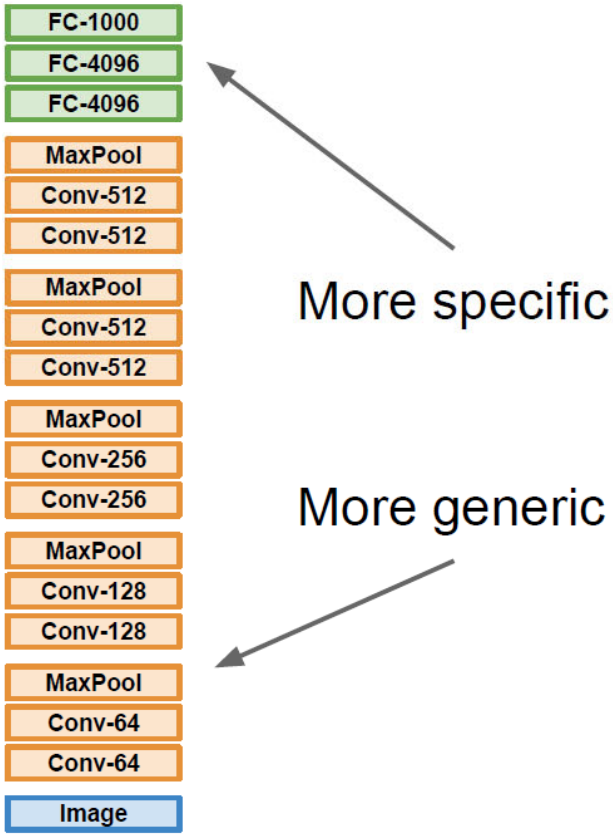


Ref: http://cs231n.stanford.edu/



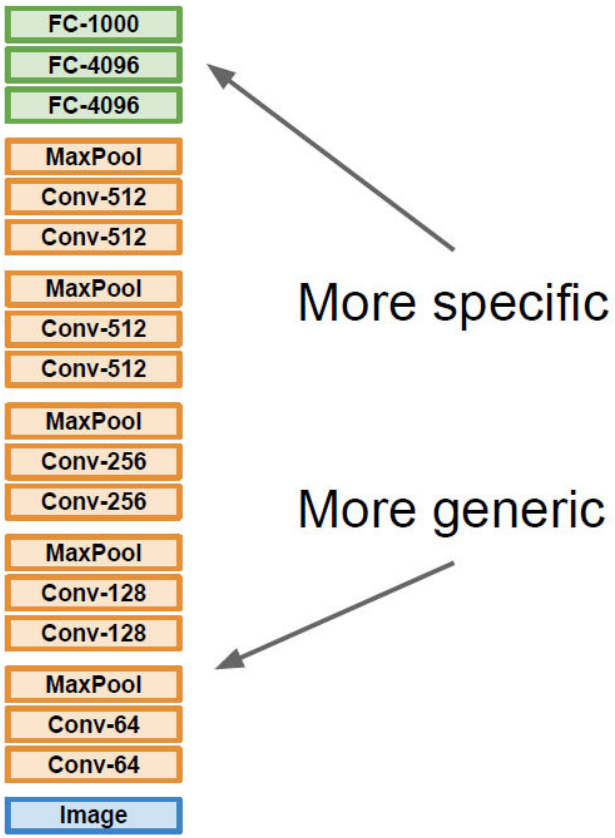
	very similar dataset	very different dataset
very little data	?	?
quite a lot of data	?	?

Ref: http://cs231n.stanford.edu/



	very similar dataset	very different dataset
very little data	Use Linear Classifier on top layer	?
quite a lot of data	Finetune a few layers	?

Ref: http://cs231n.stanford.edu/

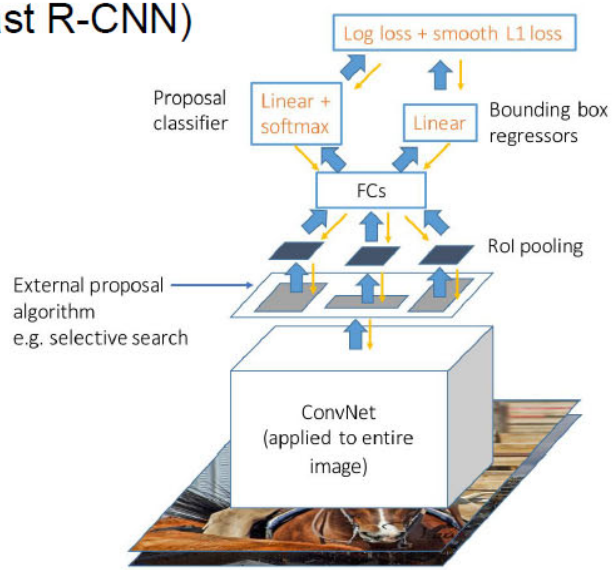


	very similar dataset	**very different dataset**
very little data	Use Linear Classifier on top layer	You're in trouble… Try linear classifier from different stages
quite a lot of data	Finetune a few layers	Finetune a larger number of layers

Ref: http://cs231n.stanford.edu/

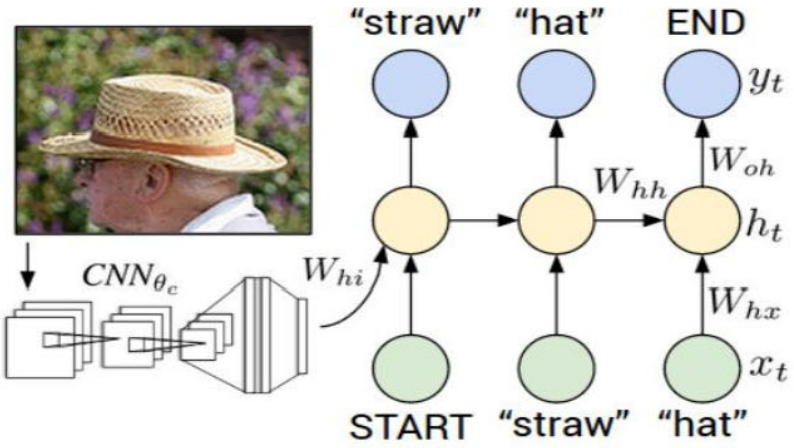
Transfer learning with CNNs is pervasive…
(it's the norm, not an exception)

Object Detection
(Fast R-CNN)



Girshick, "Fast R-CNN", ICCV 2015
Figure copyright Ross Girshick, 2015. Reproduced with permission.

Image Captioning: CNN + RNN



Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015
Figure copyright IEEE, 2015. Reproduced for educational purposes.

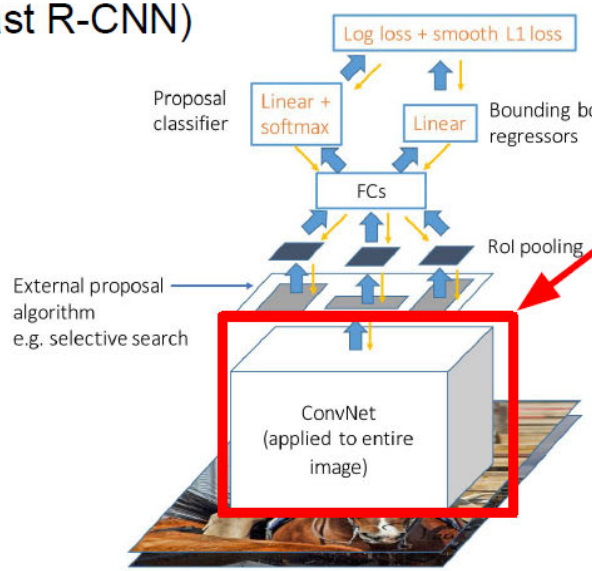
Ref: http://cs231n.stanford.edu/

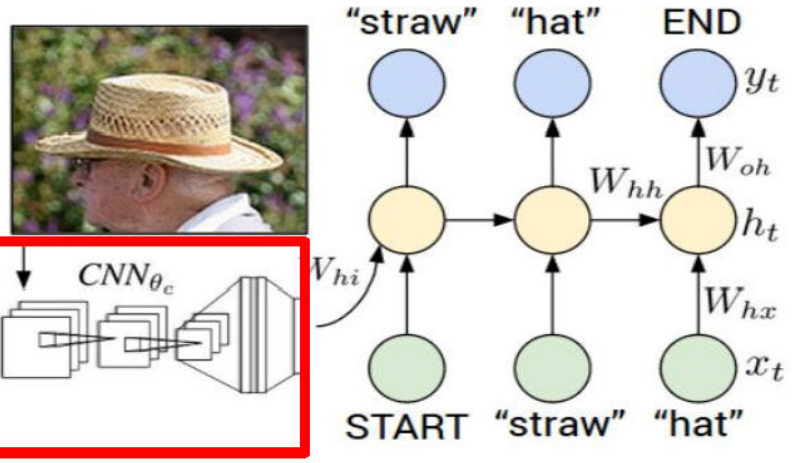
Transfer learning with CNNs is pervasive…
(it's the norm, not an exception)

Object Detection
(Fast R-CNN)

CNN pretrained
on ImageNet

Image Captioning: CNN + RNN





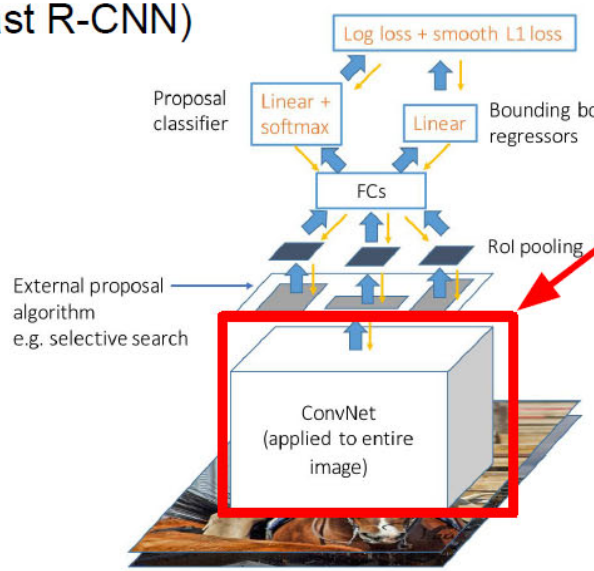
Girshick, "Fast R-CNN", ICCV 2015
Figure copyright Ross Girshick, 2015. Reproduced with permission.

Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015
Figure copyright IEEE, 2015. Reproduced for educational purposes.

Ref: http://cs231n.stanford.edu/

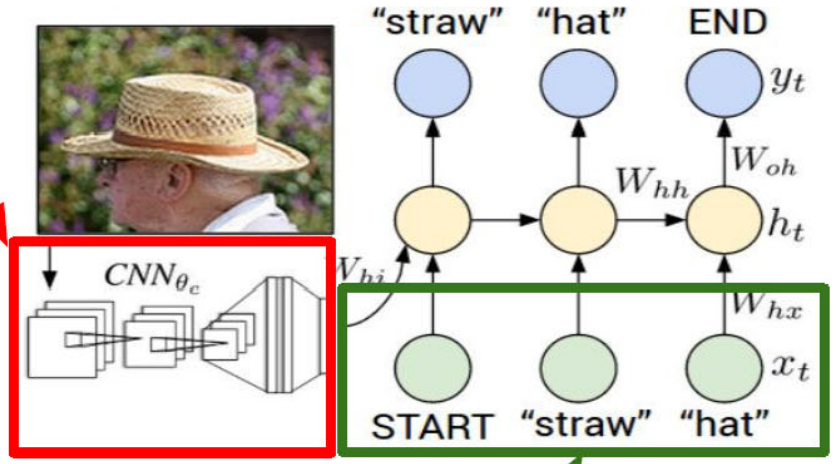
Transfer learning with CNNs is pervasive…
(it's the norm, not an exception)

Object Detection
(Fast R-CNN)



CNN pretrained
on ImageNet

Image Captioning: CNN + RNN



Word vectors pretrained
with word2vec

Girshick, "Fast R-CNN", ICCV 2015
Figure copyright Ross Girshick, 2015. Reproduced with permission.

Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015
Figure copyright IEEE, 2015. Reproduced for educational purposes.

Ref: http://cs231n.stanford.edu/

Takeaway for your projects and beyond:

Have some dataset of interest but it has < ~1M images?

1. Find a very large dataset that has similar data, train a big ConvNet there
2. Transfer learn to your dataset

Deep learning frameworks provide a "Model Zoo" of pretrained models so you don't need to train your own

Caffe: https://github.com/BVLC/caffe/wiki/Model-Zoo
TensorFlow: https://github.com/tensorflow/models
PyTorch: https://github.com/pytorch/vision

Ref: http://cs231n.stanford.edu/

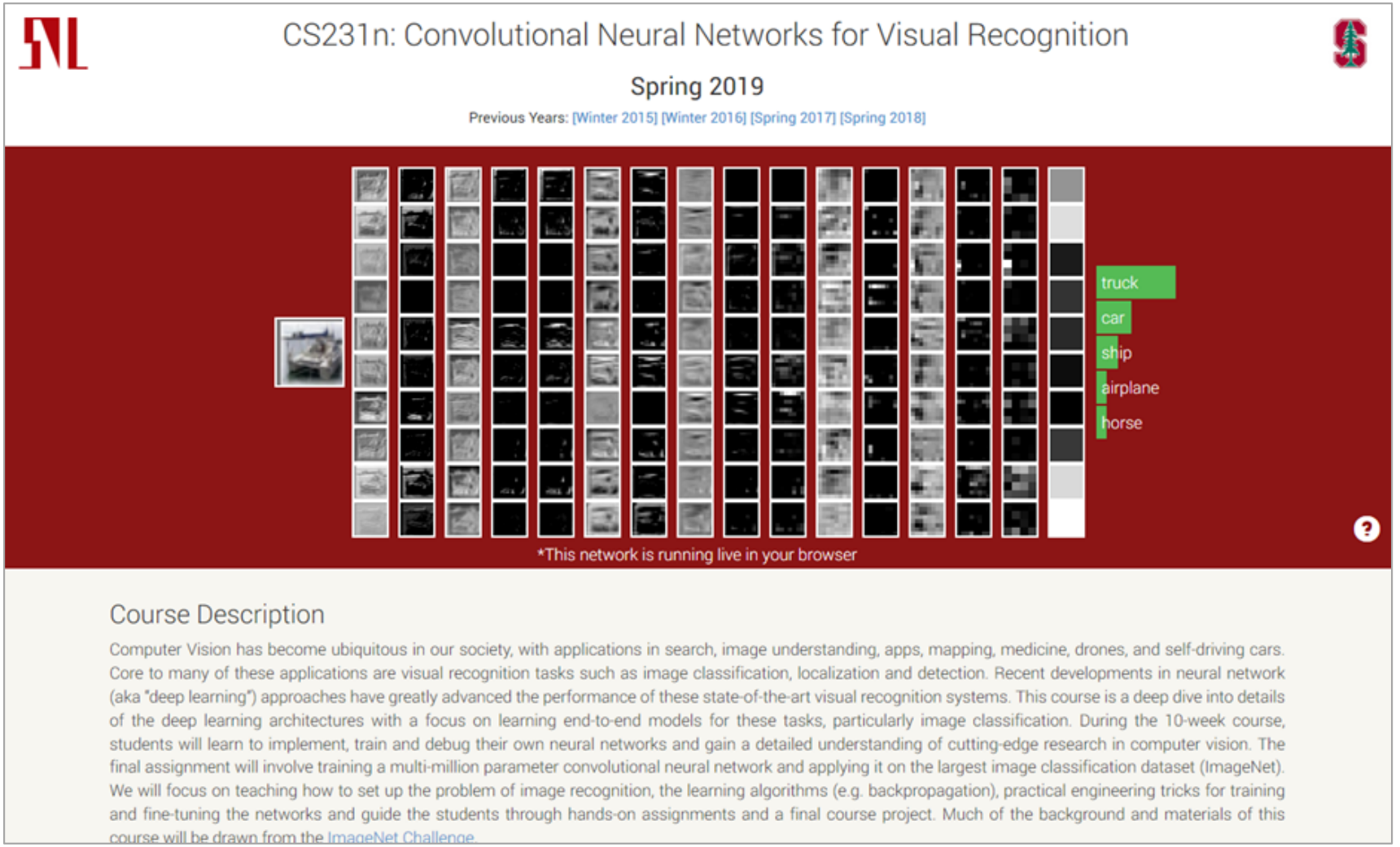
یادگیری عمیق

شبکه‌های عصبی کانوولوشنال

۴

منابع

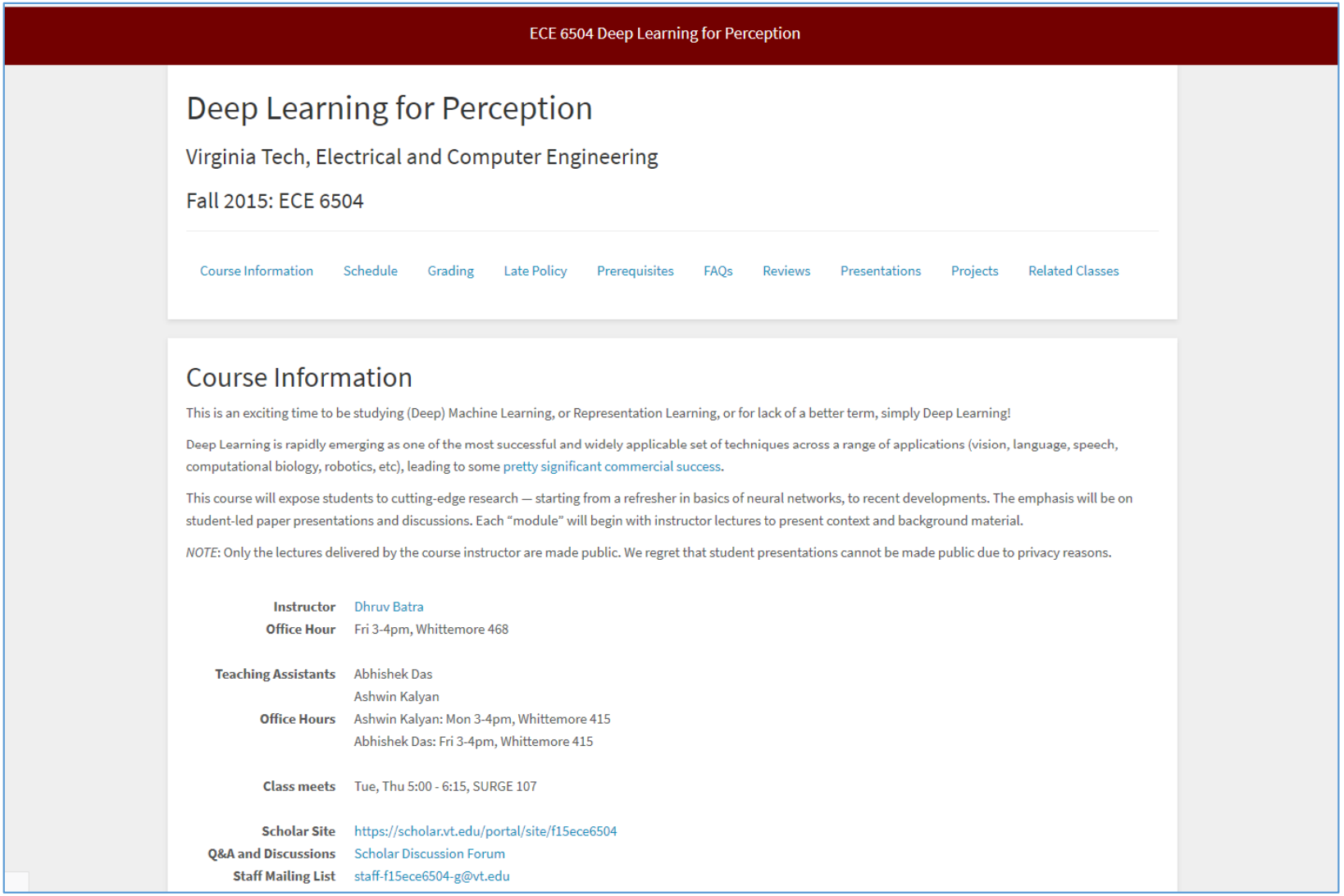
منبع اصلی



http://cs231n.stanford.edu

http://cs231n.github.io/convolutional-networks/

منبع کمکی



https://computing.ece.vt.edu/~f15ece6504/