CIS 365 Artificial Intelligence

Convolutional Neural Networks

Dr. Denton Bobeldyk

Topics

- Filter Review
- Convolutional Neural networks (CNN)
- CNN layers:
  - Convolutional layers
  - Stride
- Popular CNNs (Alexnet, etc)
- Popular Datasets

Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

Filter Review

https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html

Convolutional Neural Networks

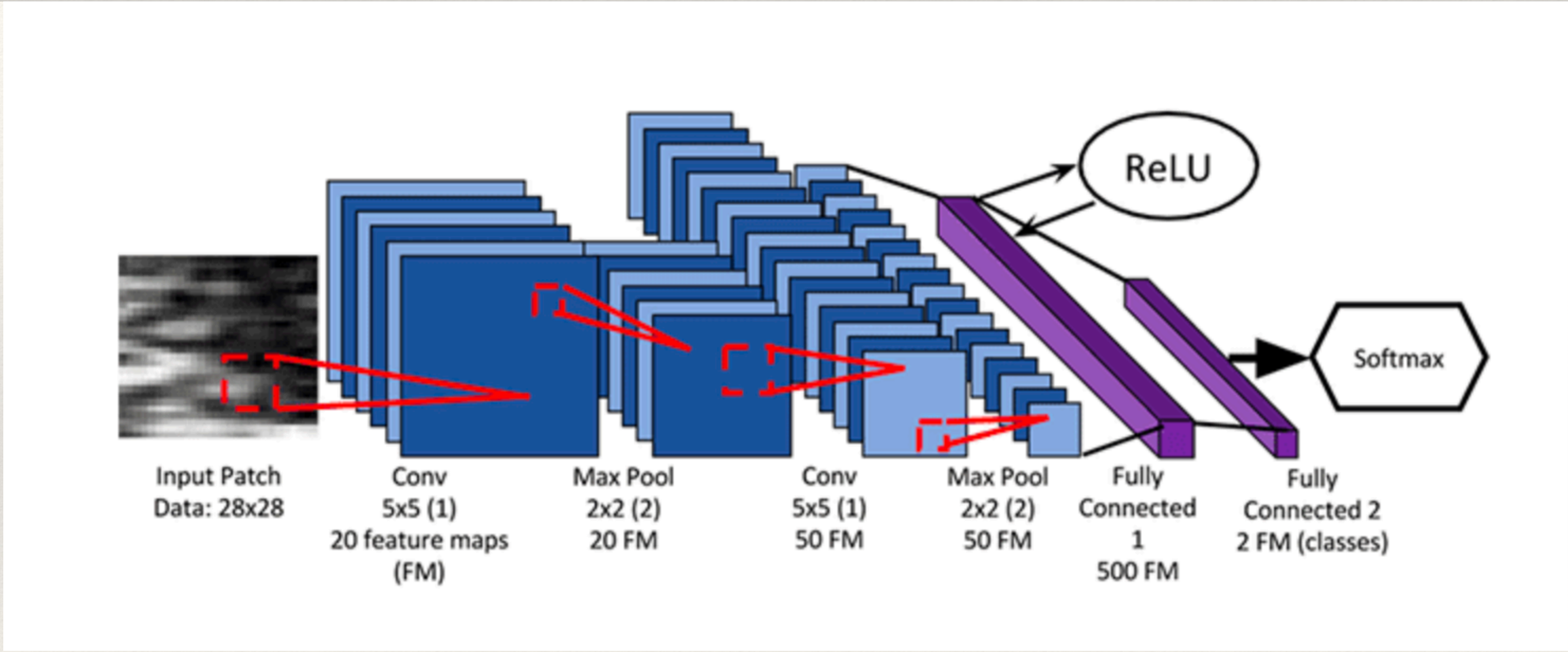


Image sourced from: https://pyimagesearch.com/2021/07/19/pytorch-training-your-first-convolutional-neural-network-cnn/

Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

Convolutional Neural Networks

Convolutional Layers - extract features from the image
Fully connected layers - discriminate between those features

Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

Convolutional Layers

- Applies a filter to the input channel
  - The input channel could be an image or a feature map from a previous convolutional layer

Convolutional Layers

- Applies a filter to the input channel

$$out(N_i, C_{out_j}) = bias(C_{out_j}) + \sum_{k=0}^{C_i n-1} weight(C_{out_j}, k) \star input(N_i, k)$$

https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html

Images from Springer Digital Image Processing

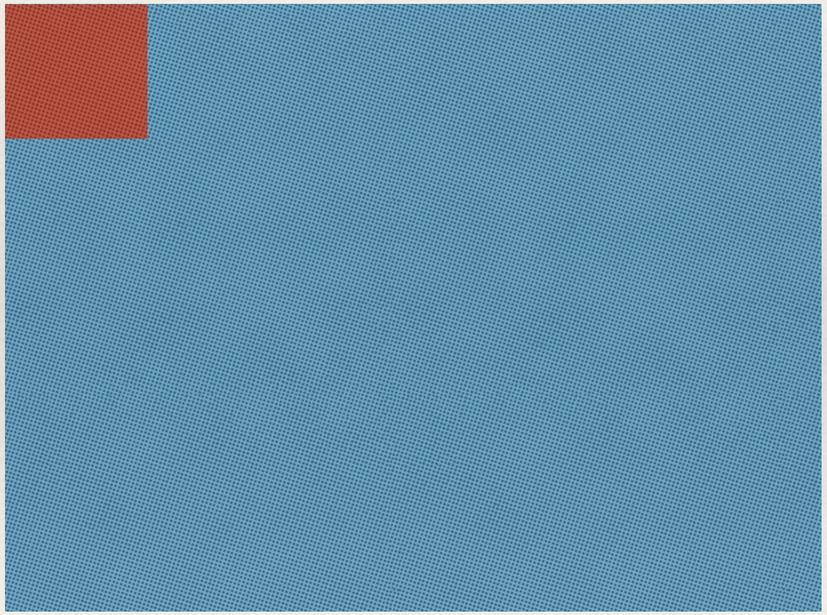
Slides created by Dr. Denton Bobeldyk

Convolutional Layers Parameters

- Stride - the number of pixels by which the convolutional filter (or kernel) is moved across the input image or feature map during the convolution process

Convolutional Layers Parameters

- Stride - the number of pixels by which the convolutional filter (or kernel) is moved across the input image or feature map during the convolution process

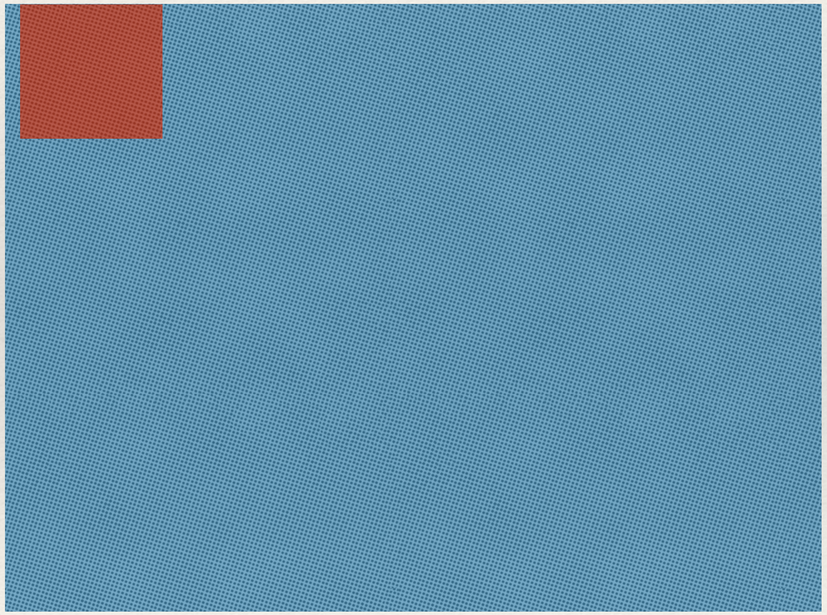


Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

Convolutional Layers Parameters

Stride of 1 pixel ‘slides’ the filter over the image, moving it to the right 1 pixel at a time

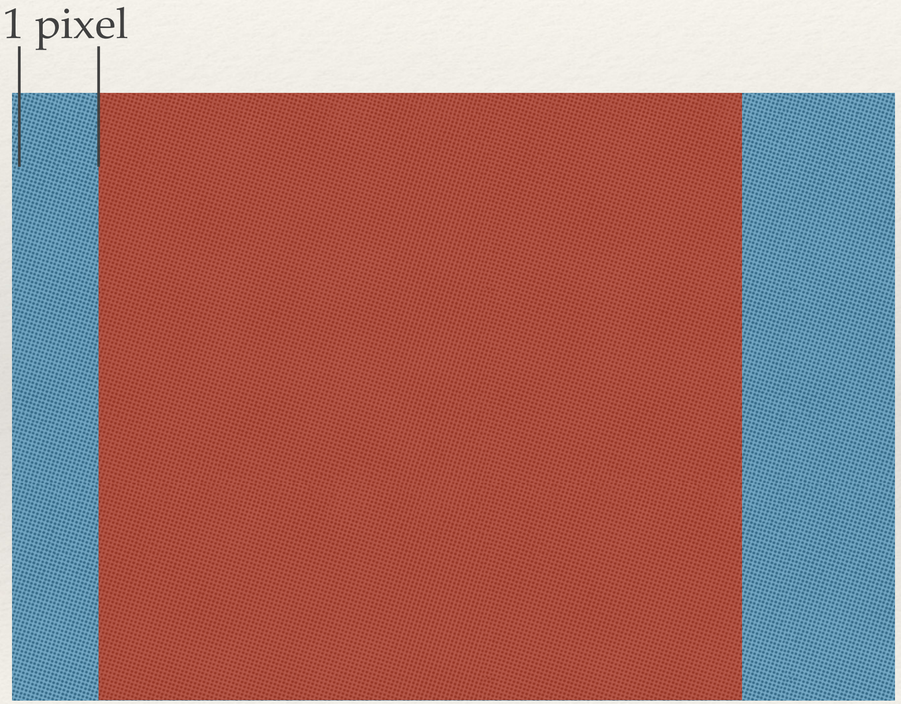


Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

Convolutional Layers Parameters

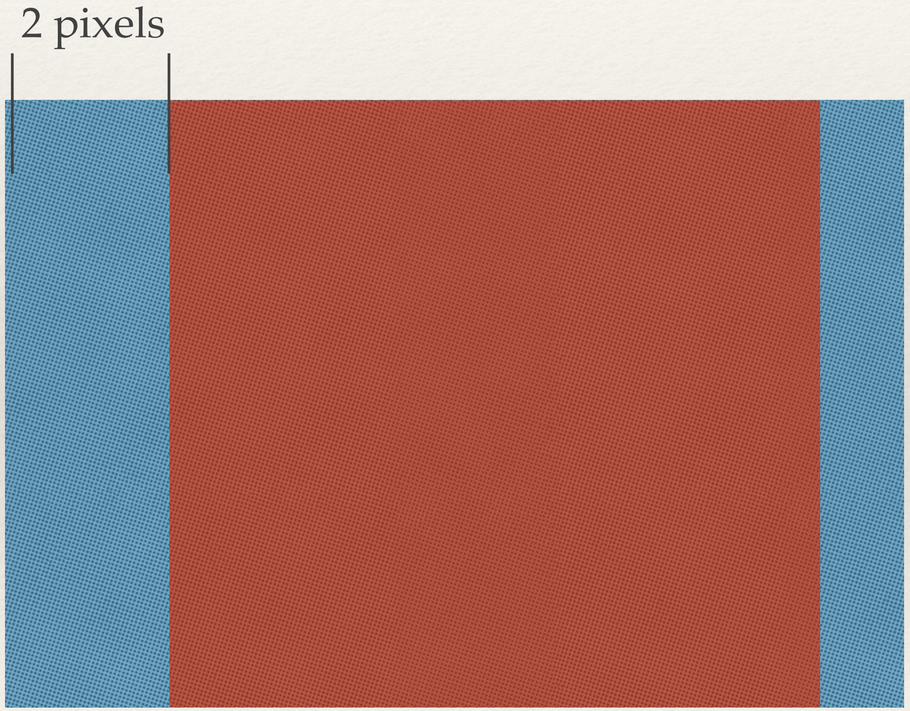
Stride of 1 pixel ‘slides’ the filter over the image, moving it to the right 1 pixel at a time



Zoom in top left corner

Convolutional Layers Parameters

Stride of 2 pixel ‘slides’ the filter over the image, moving it to the right 2 pixel at a time



Zoom in top left corner

Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

Convolutional Layers Parameters

How does the stride value effect the output (feature map generated)?

Convolutional Layers Parameters

How does the stride value effect the output (feature map generated)?

It affects the size of it. The greater the stride, the smaller the output.

Convolutional Layers Parameters

$$O = \lfloor \frac{(W - K + 2P)}{S} \rfloor + 1$$

O = output size

W = input size

K = kernel/filter size

P = padding (# of pixels added to each edge, if any)

S = stride

Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

CNN Layer Size - Exercise

Calculate the dimensions of the output given the following:

Input image size = 32x32

Kernel size = 3x3

No Padding

Stride = 1

Repeat for Stride = 3

$$O = \frac{(W - K + 2P)}{S} + 1$$

O = output size

W = input size

K = kernel/filter size

P = padding (# of pixels added to each edge, if any)

S = stride

Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

Answer Next Slide

Exercise Result: Stride = 1

$$O = \lfloor \frac{(32 - 3 + 2(0))}{1} \rfloor + 1 \qquad O = \lfloor \frac{(29)}{1} \rfloor + 1 \qquad O = 30$$

Repeat the same computation for the width value. Resulting output matrix is 30x30

$$O = \lfloor \frac{(W - K + 2P)}{S} \rfloor + 1$$

Exercise Result: Stride =3

$$O = \lfloor \frac{(32 - 3 + 2(0))}{3} \rfloor + 1 \qquad O = \lfloor \frac{(29)}{3} \rfloor + 1 \qquad O = 9 + 1 = 10$$

Repeat the same computation for the width value. Resulting output matrix is 10x10

$$O = \lfloor \frac{(W - K + 2P)}{S} \rfloor + 1$$

Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

Convolutional Layers Parameters

Why would you want a larger stride?
Capture larger patterns
Computational efficiency
Memory efficiency
Control overfitting

Convolutional Layers Parameters

Why would you want a larger stride?

Capture larger patterns

Computational efficiency

Memory efficiency

Control overfitting

CNN Design

- Can be difficult to decide how many convolutional layers, feature maps, stride, etc
- ‘Premade’ CNN’s are available that remove the design decisions

PreTrained Models

- Pretrained models: https://pytorch.org/vision/stable/models.html

PreTrained Models

The following classification models are available, with or without pre-trained weights:

- AlexNet
- ConvNeXt
- DenseNet
- EfficientNet
- EfficientNetV2
- GoogLeNet
- Inception V3
- MaxVit
- MNASNet
- MobileNet V2
- MobileNet V3
- RegNet
- ResNet
- ResNeXt
- ShuffleNet V2
- SqueezeNet
- SwinTransformer
- VGG
- VisionTransformer
- Wide ResNet

Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

PreTrained Models

RESNET

The ResNet model is based on the Deep Residual Learning for Image Recognition paper.

• NOTE

The bottleneck of TorchVision places the stride for downsampling to the second 3x3 convolution while the original paper places it to the first 1x1 convolution. This variant improves the accuracy and is known as ResNet V1.5.

Model builders

The following model builders can be used to instantiate a ResNet model, with or without pre-trained weights. All the model builders internally rely on the `torchvision.models.resnet.ResNet` base class. Please refer to the source code for more details about this class.

`resnet18`(*[, weights, progress])	ResNet-18 from Deep Residual Learning for Image Recognition.
`resnet34`(*[, weights, progress])	ResNet-34 from Deep Residual Learning for Image Recognition.
`resnet50`(*[, weights, progress])	ResNet-50 from Deep Residual Learning for Image Recognition.
`resnet101`(*[, weights, progress])	ResNet-101 from Deep Residual Learning for Image Recognition.
`resnet152`(*[, weights, progress])	ResNet-152 from Deep Residual Learning for Image Recognition.

Images from Springer Digital Image Processing

Slides created by Dr. Denton Bobeldyk

PreTrained Model Exercise - 10 minutes

- Visit the following website, after reviewing several models, select one that looks interesting and do a deeper dive. Be prepared to share a 60 second blurb about the model with the class:
- https://pytorch.org/vision/stable/models.html

Popular Datasets

- ImageNet
  - https://image-net.org/about.php
- CiFar-10/100
  - https://www.cs.toronto.edu/~kriz/cifar.html

Popular Datasets - Continued

- ImageNet
  - https://image-net.org/about.php
- CiFar-10/100
  - https://www.cs.toronto.edu/~kriz/cifar.html
- Street View House Numbers
  - http://ufldl.stanford.edu/housenumbers/
- Pascal VOC datasets
  - http://host.robots.ox.ac.uk/pascal/VOC/index.html
- COCO datasets
  - https://cocodataset.org/#home