Zhang - Convolutional Neural Networks

Images are grid shape data

MLPs flatten this grid into vectors

MLP vectors are invariant to the order of the "pixels"

Same result regardless of order

However, nearby pixels tend to be related

// important info held in these relationships

CNNs (Le Cun, 1995)

imagenet (Deng 2009)

convnets (krizhevsky, 2012)

Compared to MLP, CNN is more compencent.

- Require fewer params
- convolutions can be parrelized accross gpu

7.1. fully connected to convolutions

Invariance

when recognizing objects in a image, we likely do not want to be too concerned w/ precise location in the image

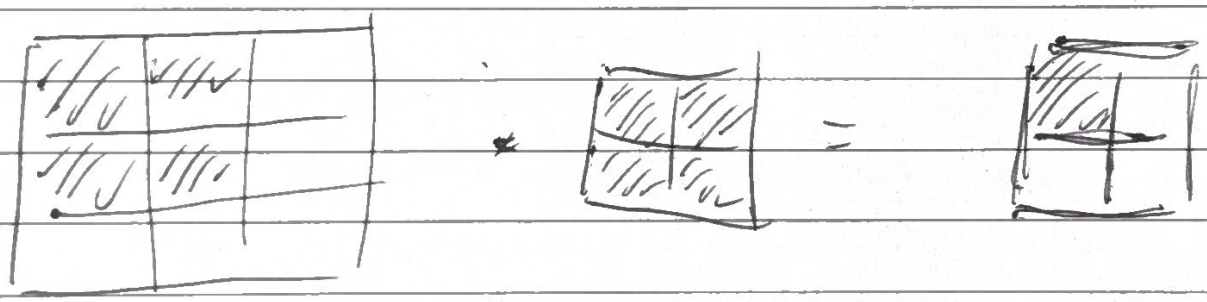
---

7.2. Convolutions for images

Convolutions = cross correlation

- An input layer
- kernal

↳ are combined to produce an output tensor



Aggregates down shape

the two params of a convolution:

- the kernal
- A scalar bias

kernals are generally randomized

Cross correlation vs convolution

A true convolution can be performed by flipping the kernal vertically & horizontally

Same trained kernel

Cross-correlation is the standard procedure

7.2.6 feature map & receptive field

output of a convolution layer is called a feature map

it can be regarded as the learned representations (features) in the spatial Dimensions (w, h)

the receptive field
↳ given an element/Pixel

the RF is the elements from the prev field that went into creating it

7.3 Padding & Stride

P & S offer more control over the size of the output

Assuming the kernal w or h > 1, we tend to "run out" of space to stride & the image shrinks

the shrunk size is lost info

Padding is the most popular tool to handle this

Sometimes we want to reduce dimensions

↳ Strided convolutions

Padding = retain size

Strided Convolutions = Reduce dimensions

7.4 Multiple input & Multiple output Channels

when adding in channels both the input & Hidden reps become 3-D

$3 \times h \times w$

3 = Channel dim

kernel needs to have same number of channels as input (if more than 1 channel)

Multi-channels = Multi filters

each filter acts on each channel

↳ 3 outputs are then concat into 1

this happens for each filter so the output varies from input to output remain the same

each output layer is a product of all the channels

7.4.2. Multi output channels

1 X 1 convolutions have the effect of reducing the number of channels

7.5. Pooling

the deeper we go into the network, the larger the receptive field to which each hidden node is sensitive

Pooling layers

↳ Mitigating sens of convo layers to location

↳ Spatially downsampling representations

7.5.1 Max & Average Pool

Similar to conv layer that is slid over

Contains no parameters

instead deterministic

↳ Avg or max the kernel values

Akin to down-sampling

Can be adjusted w/ Padding & Stride to change output size

unlike convo layers

↳ Pooling acts on each channel

↳ Does not concat or aggregate channels remain

7.6 Convo Neural Networks - (LeNet)

fully functional CNN

high level =
- 2 conv layers
- DenseBlock 3 MLPs

Image → conv → Pool → conv → Pool → MLP x3

Convo layer = conv & sigmoid activation

Pool = Average

↳ Relu & Max not discovered yet

Conv layers have 6 & 16 filters turning the 2-D input into having channels

Pooling is used to downsample by factor of four

2nd conv also has a downsample effect

output of conv is flattend to be pass MLP

↳ take minibatches & flatten

first conv uses padding to compensate to retain size

↳ 2nd foregoes which is who decreases size