Lecture 29: Auto Encoders

CNN-2D

Applicatn of CNN

→ Lark related image (videos)

- computer vision applicatns — (1) semantic segmentatn
  (2) image classificatn
  (3) object detect
  (4) object recognitn

input mge → convolutnal layer → Pooling layer → flattned → fully connected → o/p

CNN is good for imges

Neural Network ⇒ Auto Encoder

type of neural network designed for data compressn, anomaly detectn, dimensionality reductn and feature learning

why

↳ To get crux (summary) of mformatn

∴ (compressed representatn) so that we can that expand later.

-structured manner

↳ less computational power

Compressed representatn → actual imp re-create

Auto Encoder

(function like which give the same input as like input)

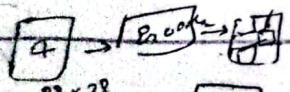
Primary goal

[To encode data into a compact representation and then decode it for re-construction]

Auto encoders

encoder

- converts original data into secret code
- use rules/transform to hide the message



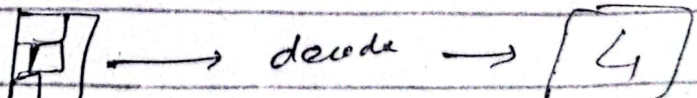
bottleneck/hidden

The encoder is like detective that learn to capture most important feature.

- This part is responsible for compressing the input data

decoder

- like an artist that take compact representation (encoding) from encoder and re-create the original data.
- it is second part of auto-encoder & responsible for generating output from encoder



latent/compressed representation

output

- Reverse the secret code back to original data
- understand the rules to reverse the encoding.

encoder

→ Train of auto-encoder (if we are unable to create the output for input)

encoder is like as a funnel that squeeze a big picture into smaller representation

decoder

enable understanding of hidden message.

It transform the encoding from latent space back into a reconstructed output enabling the decoder network to reconstruct the original input as closely as possible.

Encoding Process

Step 1: input data like image is fed into encoder.

Step 2: encoder consists of layer of neuron that learn patterns and feature in the data.

step 3, pattern are combined and transformed into compact representation in low Dimensional Space (latent space)

so two thing happens

① feature learning (convolution) (feature extract)

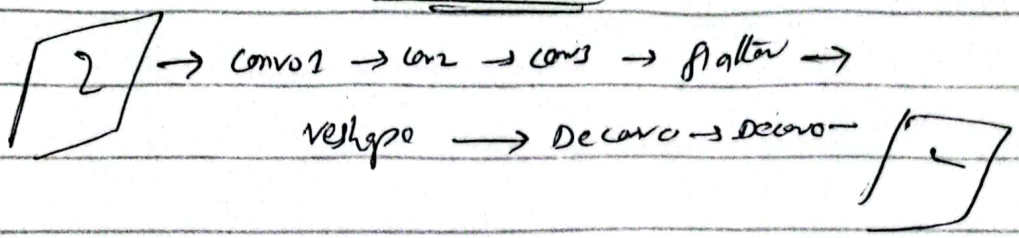
② Compression (reduce) (: pooling layer)

some we can use CNN (convolutional + pooling layers)

21:23

So

Auto-Encoder



28×28×1 → 14×14×32 → 7×7×64 → 3×3×128 → 1152 → 10

(normal image) (flatten (1D vector))

encoding

3×3×128 → 7×7×64 → 14×14×32 → 28×28×1

(normal image)

decoding

Deconvolution

- Takes the compressed feature maps from encoder and expands their spatial dimension
- reverse of convolution (feature → image (deconvolute))
  (image → feature)

Decoding steps

1. The encoded data (latent space repr) feed into decoder
2. The decoder consists of layer that learn to reverse the compression proces
3. constructed features combined to reconstruct original data.

→ we can either use encoder and decoder → combine
→ seperately

→ when we can generate something we can create variation like from old, blurry img to clear img

→ Types of autoencoder
→ use cases of autoencoder
→ modification of autoencoder

As homework