5/30/2020

Forward Pass: Propagation: All data travels forward through nodes to calculate the NN output

- Input Layer: Prepossessing → non-Linearity

  - Equation:
    - input: $X \cdot W$ → data → weight input to hidden
    - output: activation function(input) → sigmoid ϕ or σ

- Hidden Layer: non-linearity / Linear

  - Equ:
    - input: hidden output $\cdot W$ → output from hidden layer → weights from hidden to output layer
    - output: activation / $1 \cdot$ input ← or

Back propagation: data travels in opposite direction to fix/adjust weights & bias

- Output layer:

  Equ input:

  Output error: $(y - \hat{y})$ ← target, output/prediction | label - prediction

  Output error term: $(y - \hat{y}) f'(a_k)$ | label - prediction (error) * derative of activation function of output (gradient)

hidden layer:

hidden error: W_{jk} → weight δ_k ← previous layer error term | Output layer term * weights hidden to output

hidden error term: $\Sigma[W_{j,k} \delta_k] f'(h_j)$ | hidden layer error * derative of activation function of hidden output

(error) (gradient)

6/2/2020 cont.

Filters & Edges in CNNs:

- low pass filters : block high frequencies (vice versa)
- using high pass filters - can block low pass freqs in image
  - used for edge detection
- convolutional kernal : edge detector filter

weights = #'s

ex nij

0	-1	0
-1	4	-1
0	-1	0

$\sum nij = 0$

$(-1)+(-1)+(-1)+(-1)+(+4)+0+0+0+0 = 0$

process ⇓

Kernel ⇒

0	-1	0
-1	4	-1
0	-1	0

Image data ⇒

3	4	7
8	5	6
9	11	10

* Convolution (astrik)

* main save volume data

numbers multiplied

Final ⇒

8	-4	0
-8	20	-6
0	-11	0

$\sum nij = -9$

$(-4)+(-6)+(-8)+(-11)+20 = -9$

Convolution Layer:

- applies many convolutional kernels
- stride the distance from pixel to pixel
  - 9 becomes center pixel value (5) in data image

Max pooling — used for decreasing complexity b/c overfitting

Avg Pooling

** Capsel Networks

6/2/2020 cont.

Filters & Edges in CNNs:

- low pass filters : block high frequencies (vice versa)
- using high pass filters - can block low pass freqs in image
  - used for edge detection
- convolutional kernal : edge detector → filter

weights = #'s

ex nij

0	-1	0
-1	4	-1
0	-1	0

$\sum n_{ij} = 0$

$(-1)+(-1)+(-1)+(-1)+(+4)+0+0+0+0 = 0$

process ⇓

Kernel ⇒

0	-1	0
-1	4	-1
0	-1	0

Image data ⇒

3	4	7
8	5	6
9	11	10

* ↗ Convolution (astrik)

* main same volume data

numbers multiplied

Final ⇒

3	-4	0
-8	20	-6
0	-11	0

$\sum n_{ij} = -9$

$(-4)+(-6)+(-8)+(-11)+20 = -9$

Convolution Layer:

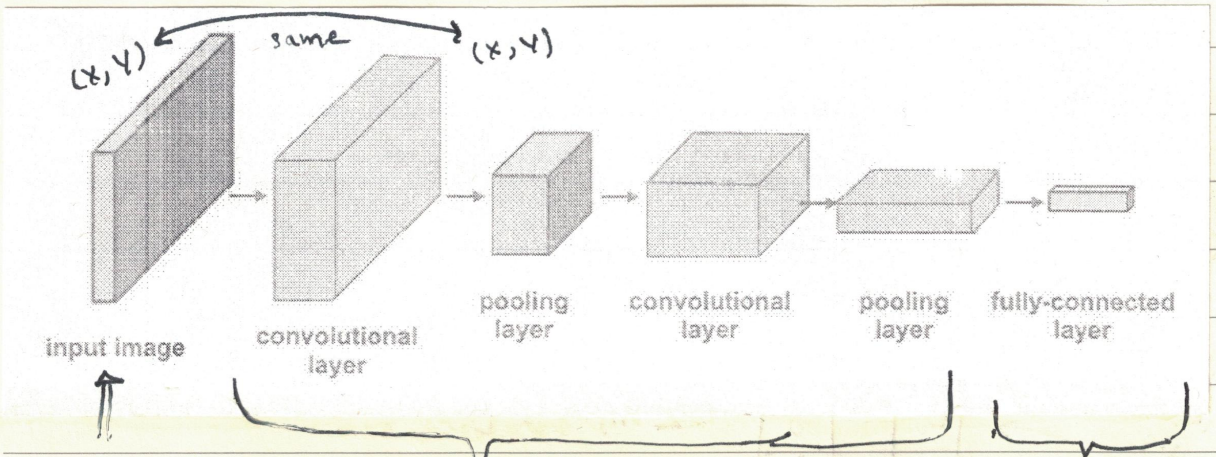
- applies many convolutional kernels
- stride the distance from pixel to pixel
  - 9 becomes center pixel value (5) in data image

Max pooling — used for decreasing complexity s/c overfitting

Avg Pooling

** capsel networks

6/2/2020 cont.



input

feature extraction

(isolation information from image)

Prediction

6/3/2020

Transfer learning: implementing feature extraction from successful models and retraining the dense/fully-connected layer.

Dataset similarity

Dataset size	Similar	Different
large	Fine-tune (2)	Fine-tune / retrain (3)
small	End of ConvNet (1)	Start of ConvNet (4)

Cases:

1. Adjust end of ConvNet
   - ~~remove and replace end as~~ Dense layers (add) (not remove dense)
     - randomize dense layer weights → update weights
     - freeze weights in pretrained network - don't update weights
       ↳ prevent overfitting

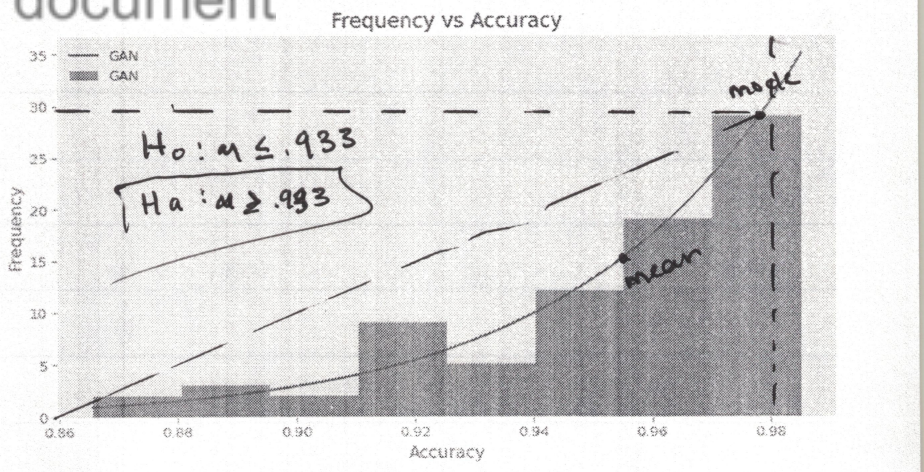
1/16/21

$\bar{X} = 0.9498$

$\sigma_x = 0.0293$

$t = \frac{0.9498 - 0.933}{0.0293/\sqrt{149}} = 6.99$ ave

$P = 1.08\,e^{-10}$



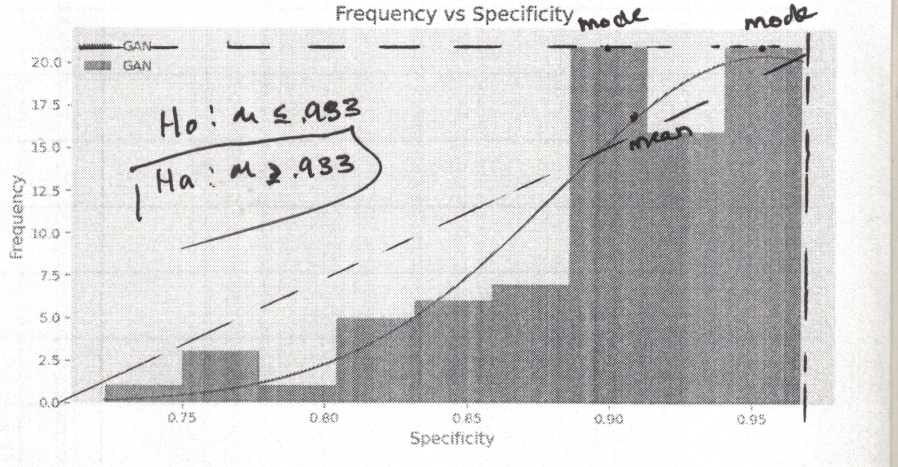
The distributions are subjected to the test, could show that it is stable in performance,

$\bar{X} = 0.9030$ ave

$\sigma_x = 0.0547$

$t = \frac{0.903 - 0.933}{\frac{0.0547}{\sqrt{149}}} = -6.69$

$P = 1.65\,e^{-10}$



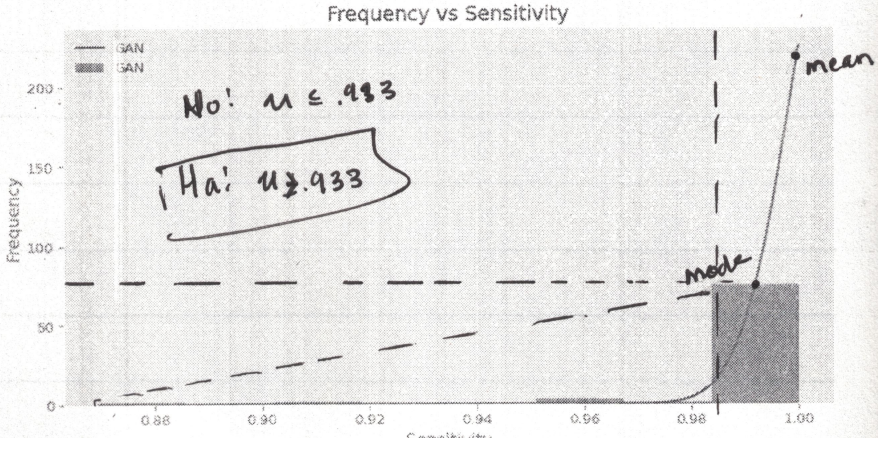
Sensitive is Maximized b/c we want to have all potential anomalies detected

$\bar{X} = 0.9952$

$\sigma_x = 0.0197$

$t = \frac{0.9952 - 0.933}{\frac{0.0197}{\sqrt{149}}} = 38.5$

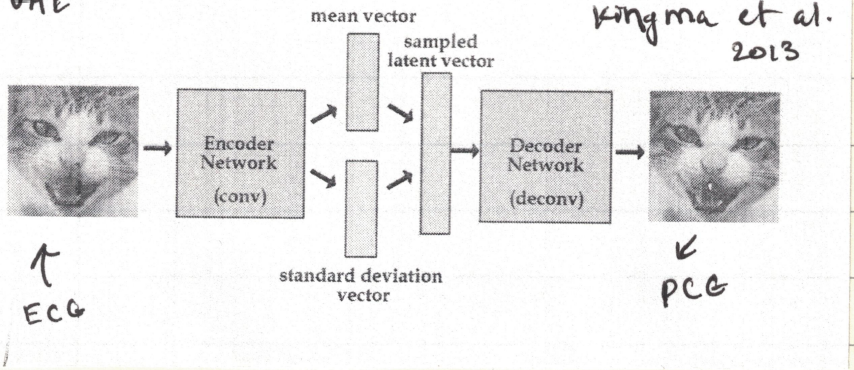
$P = 1.54\,e^{-77}$



2/1/21

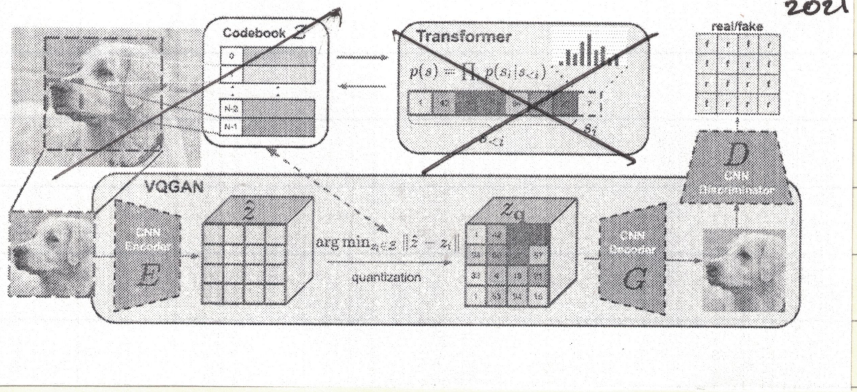
VAE

Kingma et al. 2013



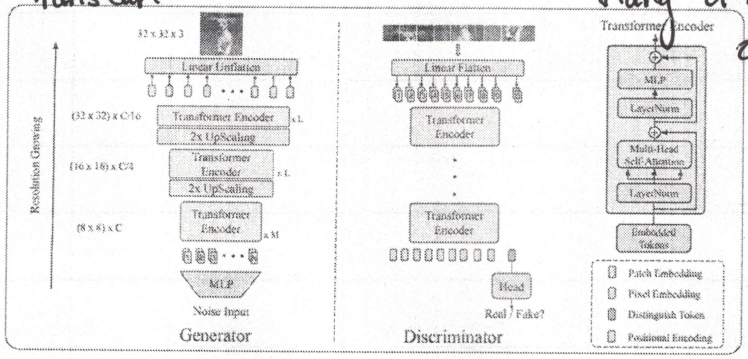
- State of the art autoencoder which is seq to seq
- used for reconstructing sequences ex images

Taming transformers for HR Image Synthesis Esser et al. 2021



- State of the art seq to seq for images
- uses transformer & CNN for latent representation (codebooks)

TransGan Jiang et al. 2021



- State of the art transGans (uses transformers for G & D)