ML - Week 9 - Beyond supervised learning

Recap -

raw data to model input

- Data collection
- Post collection
  - Combine? aggreg? structure?
- Pre-model -
  - Any domain specific preprocess?
  - feature extraction?

types - take raw data & plot/visualise
smooth? variance? change point/threshold

Hand crafting features -

- Shape features - Area, roundness
- Colour features - mean, std dev

Early ML & DL models were created uses Hand crafted features relying on domain ~~error~~ expertise

---

Deep learning

why "Deep"? the first layers of the network act as feature extractors. learn low level features

latter = higher level info + Actual classify/regress

Representation / feature learning

representation learning = A seperate split task of learning feature before the actual supervised learning task

~~Autoencoders~~
can use self-supervised learning for this purpose — Autoencoders (NO labels)

collecting labels on data is not trivial

Representation learning can be done using labelled or unlabelled data

features also known as-
- Embeddings
- encodings
- latent variables
  - ↳ Hidden variables

Supervised representation learning =
encoder ⟶ classifier

unsepervised rep learning:
- Auto-encoder
- encoder output = features
- Second part = decoder

Autoencoder = task reconstruction

$$L_2 = \| \underbrace{f_{w_d}}_{\text{decode}}\left(\underbrace{g_{w_d}}_{\text{encode}}(x)\right) - \underbrace{x}_{\text{input}} \|^2$$

Dimension reduction in AEs

end of encoder is at a diff / smaller Dimension vs input

decoder then expands back to input size

Transfer learning

Re-use of a trained encoder in a target task

representation learning task = encoder / decoder task

After use can transfer the use of the trained encoder to the supervised learning task (~~target task~~)

target task = the main purpose of the work

encoder → classifier

transfer learning = Pre-training

Ways to use an encoder:

(Data Processing)

- use as an offline feature extractor
  - enrich the dataset & use as inputs

- Integrate directly into the model
  (1) encoder → classifier
  freeze the encoder weights

(2) Fine-tune the encoder
- Pre-trained encoder weights are updated during the target task

Pretraining vs target tasks

types of input data need to be the same

but domain doesnt need to be the same
i.e. images but content is diff

CNNs as feature extractors

w/ CNNs the feature extraction is built in

does not need domain feature extraction

Generative Models

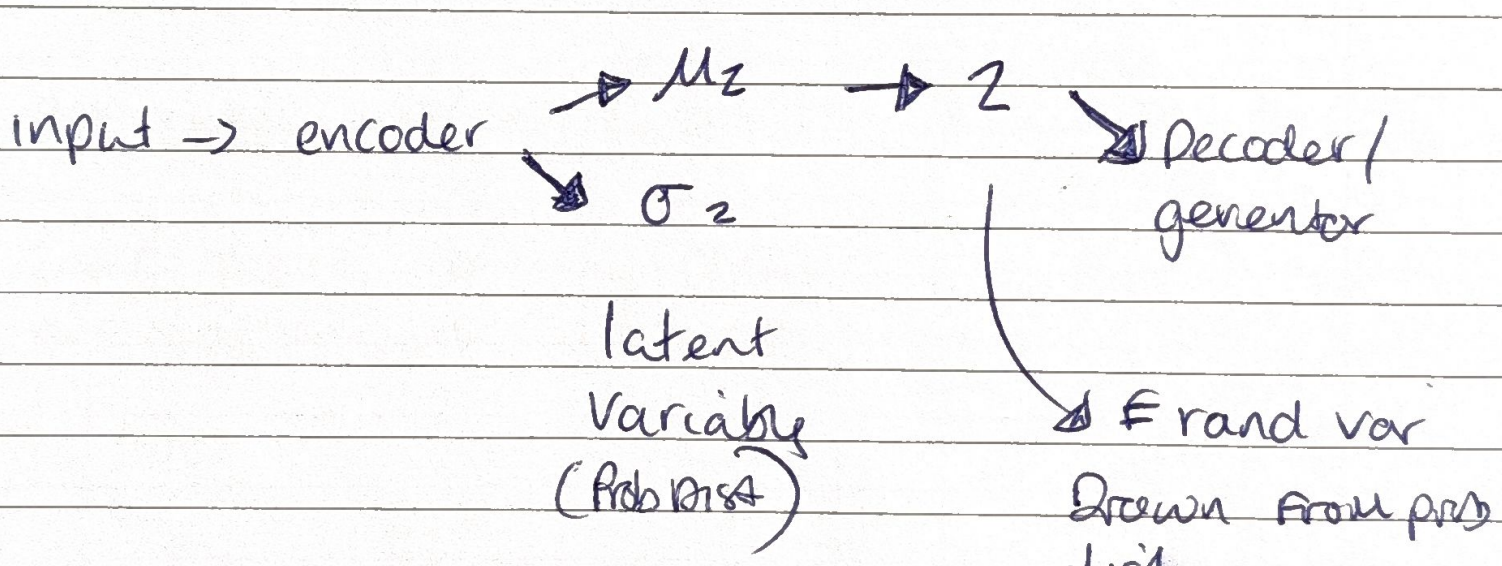
Covering two types (but there are more)

- variational autoencoder (VAE)
- generative adversarial (GAN)

Autoencoders are deterministic

Variational Autoencoder (VAE)

in VAE the latent variables are not deterministic - they are more probabilistic (from prob dist)

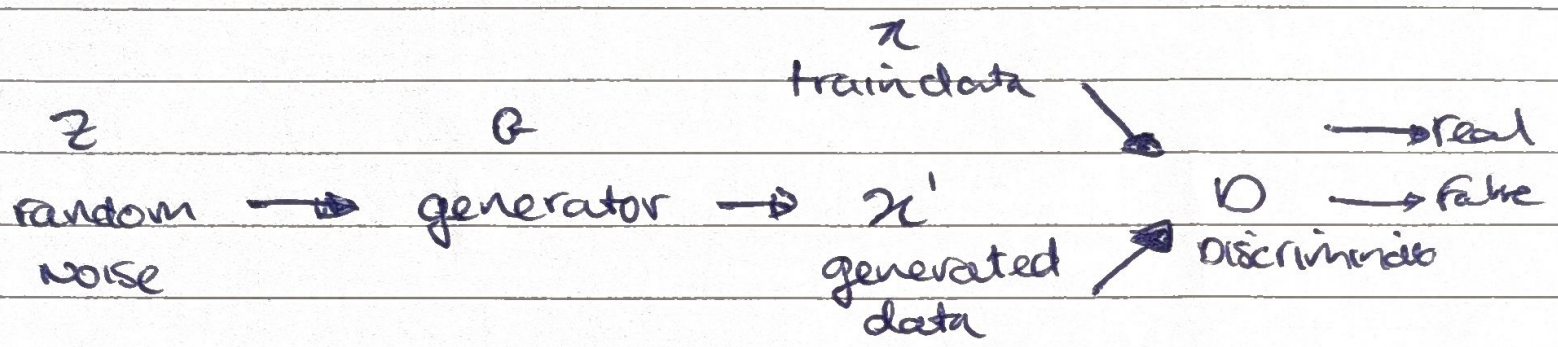


unseen data can be generated by sampling from the ~~probbb~~ Prob dist

Generative Adversial Network (GAN)

has two portions to the model:

- Generator (encode) & discriminator (decode)



the aim of the generator is to create something that looks real

generator wants to fool the discriminator

GANS have lots of issues in practice & are not often used in research