SRI SAI RAM ENGINEERING COLLEGE

Name: Naaga Dheva Dharshan. N   Class & sec: AI-DS 'C'

Sec id: SEC22AD029   Roll no.: 28   CAT: II

Register no.: 412522243106   Date: 24.04.2024

Course code - name: 20AIPC403 - Advanced Machine Learning

Part - A (5 x 2 = 10 Marks)

1. Generative model.

Generative models are a class of models in unsupervised machine learning space. They help us to model underlying distributions responsible for generating the dataset under consideration. There are two major classifications of generative models: 1. Explicit density
2. Implicit density

2. Forward and importance sampling.

Forward sampling:

* Direct sampling: Samples are drawn directly from the target distribution without the need for an intermediary distribution.

* Suitably for simple distribution: Well-suited for simple target distributions where direct sampling is feasible and efficient.

Importance sampling:

* Indirect sampling: Samples are drawn from a proposal distribution, which might be different from the target distribution then weighted.

* Flexibility for complex distribution: Effective for complex target distributions where direct sampling is im paractical or infeasible, allowing for estimation using a more convenient proposal distribution.

3. Back propagate:

Back propogation is a key algorithm in training neural networks that calculates gradients of the loss function with respect to the model's parameters. It efficiently progates these gradients backward through the network, updating the parameters using gradient descent or its variants. By iteratively adjusting weights based on the calculated gradients, back propogation enables the network to learn and improve its performance on tasks like classification or regression.

4. Approximate inference:

Approximate inference refers to refers to a set of methods used in statistical and machine learning to estimate model performance or make predictions when exact inference is computationally infeasible. These methods aim to approximate the true probability distributions with a more tractable one, trading off some accuracy for computional efficiency.

5. Auto regressive model in density estimator.

Auto regressive model in density estimation predicts the probability distribution of a variable based on its own past values, assuming a conditional independence structure. It models the probability density fuction as a fnction of its own past values, often using techniques like autoregressive neural networks or auto regressive moving average (ARMA) models.

Part- B (2 × 13 = 26 Marks)

6.a) Variational auto encoders.

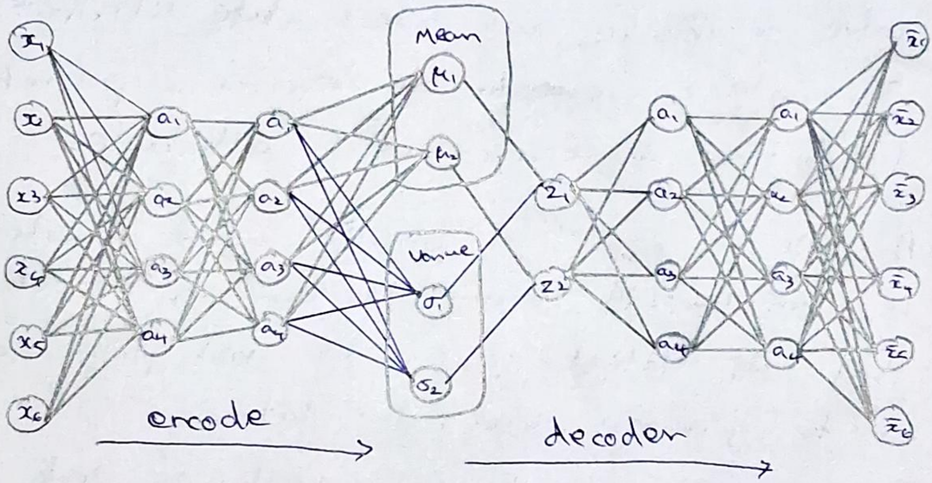
A variational auto encoder (VAE) provides a probabilistic manner for describing an observation in latent space. Thus, rather than building an encoder that output a single value to describe each latent state attribute, we'll formulate our encoder to describe a probability distribution for each latent attribute.

A VAE is an auto encoder whose encodings distribution is regularized during the training in order to ensure that its latent space has good properties allowing us to generate some new data.
It has many applications such as data compression, synthetic data creation etc.

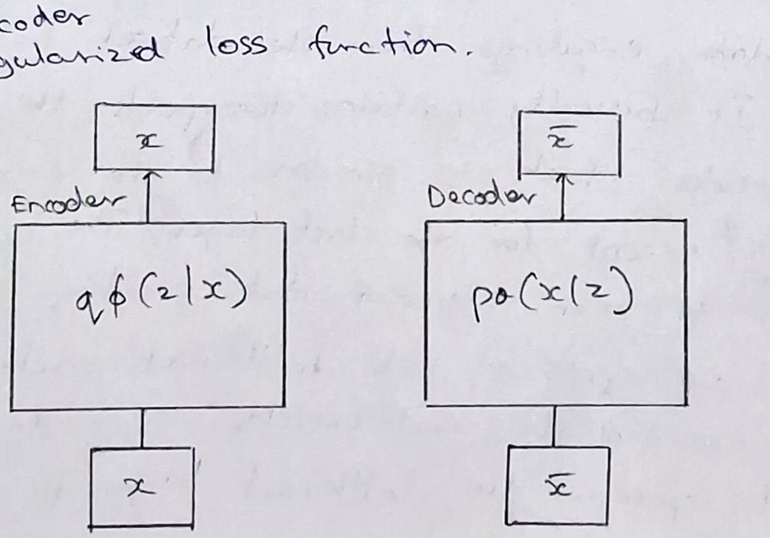
Architecture:

Auto encoders are a type of neural network that learns the data encodings from the dataset in an unsupervised way. It basically contains two parts: the first one is an encoder which is similar to the convolution neural network except for the last layer. The aim of the encoder to learn efficient data encoding is a decoder dataset and pass it into bottleneck architecture. The other part of the auto encoder is a decoder that uses latent space in the bottleneck layer to regenerate the images similar to the dataset. These results back propogate from the neural network in the form of the loss function.

Variational autoencoder is different from autoencoder in a way such that it provides a statistic manner for describing the samples of the dataset in latent space. Therefore, in variational autoencoder, the encoder outputs a probability distribution in the bottleneck layer instead of a single output value.



* Training data is mapped to latent space using neural network.
* The latent space vector is mapped to input to image using another neural network.
* Components of variational autoencoder
  - Encoder
  - Decoder
  - Regularized loss function.



* Encoder takes training data training data as input and produces a latent representation z.
  - q_θ parameters of encoder.

* Decoder:
  - z is sampled from the output of the encoder and gives the input to decoder.
  - Decoder outputs similar to x i.e the regenerated data.

* VAE loss function:

$$-D_{KL}[q_\phi(z|x)||p_\theta(z)] + E_{q_\phi}(z|x)[\log(p_\theta(x|z))]$$

$$D_{KL}[q_\phi(z|x)||p_\theta(z)] = E_{z\sim q}[\log(q_\phi(z|x) - \log(p_\theta(z)]$$

Term 1 represent regularizer and term 2 is data reconstruction loss.

Regularizer is the KL divergence of the output by the encoder network and prior model for the distribution of z. both modeled using gaussian distribution. KL says about how similar the distributions are. If they are exactly same then it will be 0.

There is both a continuous form of KL divergence.

$$D_{KL}(p(x)\,||\,q(x)) = \int_{-\infty}^{\infty} p(x)\ln\frac{p(x)}{q(x)}\,dx$$

And a discrete form of KL divergence:

$$D_{KL}(p(x)||q(x)) = \sum_{x\in X} p(x)\ln\frac{p(x)}{q(x)}$$

7.a) Masked auto regressive flow for density estimation.

* Auto regressive models are among the best performas neural density estimator.
* Types of neural density estimator that model the conditional distribution of each input variable given the previous variable.
* The flexibility of model can be limited becouse the decompose the joint density as a product of conditional and model each conditional in turn.
* masked auto-regressive flow paper increase the flexibility of modelling the random number.
* In high dimensional spaces, traditional computional are infeasible.
* Normalizes flow provide an alternative approach by providing transforming simple distribution is to more complex one.

Normalizing flow:

A normalizing flow represent $p(x)$ an invertible function f of base density $\pi_u(u)$ .ie $x = f(u)$

$$p(x) = \pi_u \left(f^{-1}(x)\right) \left| \det \left(\frac{\partial f^{-1}}{\partial x}\right)\right|$$

$$P(x_1, x_2) = N(x_2 | 0, 4)\ N\left(x_1 \middle| \frac{1}{4} x_2^2\right)$$

Simple distribution $\longrightarrow$ Gaussian distribution

* The transformation are invertible given exact computation.

Autoregressive modeling:

* Decompose the joint distribution of variables into a product of conditional distribution.
* Each variable is modeled as function of previous variable, capturing complex dependencies.

Masked autoregressive flow:

* Auto regressive coupling layers.
  → These layer decompose into parts one is transformed and another remains unchanged.

Invertible transformation:

* Each coupling layer is destroyed to invertible enabling exact computation.

Training MAF:

* MAF is trained by maximizing likelihood of training data using maximum likelihood estimation
* Gradients are computed using back propagation through the entire model.
* Training involves optimization algorithm such as stochastic gradient descent.

$$\left|\det\left(\frac{df^{-1}}{dx}\right)\right| = \exp\left(-\Sigma_i d_i\right) \text{ when } d_i = f_{\theta_i}(x_{i:i-1})$$

Application:

→ Image modelling
→ Text generation

→ Anomaly detection.

Advantages:

* MAF can model complex data distribution and generating high quality samples.
* Exact computed and support efficient sampling.

Limitation:

* Computation can be expensive especially for large data set and complex model.

Conclusion:

• It is aeffective approach for high dimensional data sets.

Part - C (1 × 14 = 14 marks)

8.b) Loss function in GAN:

* GANs try to replicate probability distribution the should use loss function that reflects distance between distribution data generated by GAN and the distribution of read data.

→ Two GAN loss function:

* Minimax loss
* Wasserten loss

one for the generates training and one for discrimination training.

* The generator and discrimator losses derive from a single measure of distance between probability dist.

* Generator can only affect one term in distance measure. the term that reflect the distribution of fake data.
* During generator training we drop the other term, reflects distribution of real data.
* The generator and discrimination losses look different in end, even, they derive through single formula

Minimax loss:

* The generator tries to minimize the following fuction

$$E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

$Dx \rightarrow$ is the discriminator estimate of real instance

$Ex \rightarrow$ expected value

$G(z) \rightarrow$ generator output given noise z

$D(G(z)) \rightarrow$ fake instance in real.

$E_z \rightarrow$ is expected value overall random inputs.

Modification minimax-loss:

GAN paper notes that the above minimax loss function cause GAN to stuck in early stages.

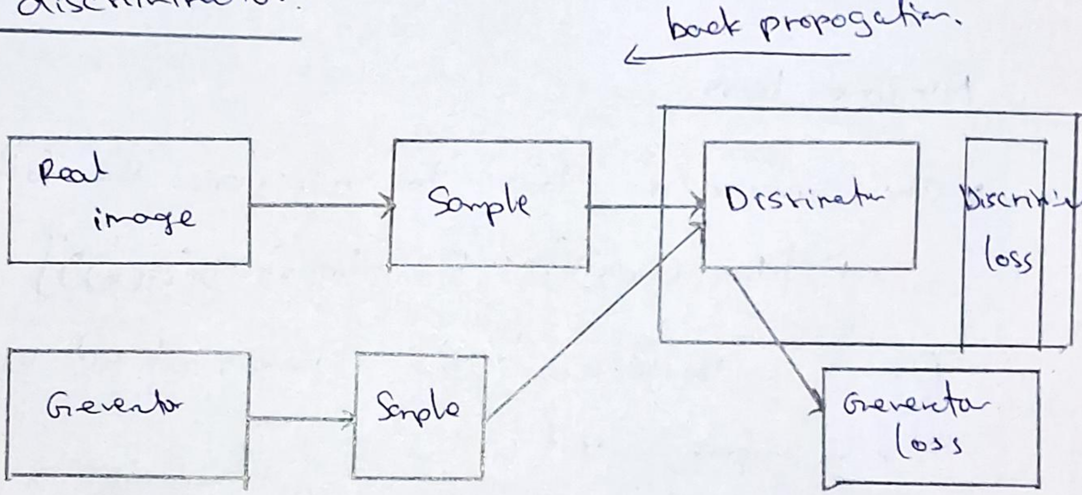
Wasserstein loss:

* The loss function depends on modification of GAN scheme called wasserstein GAN.
* The generator parts of a GAN learn to fake data by incorporating feedback from the discrimination.

Training the generator

* Sample random noise.
* Produce generation from sample random noise.
* Real or fake
* Calculate loss function
* Back propogate through discrimination.

The discriminator:



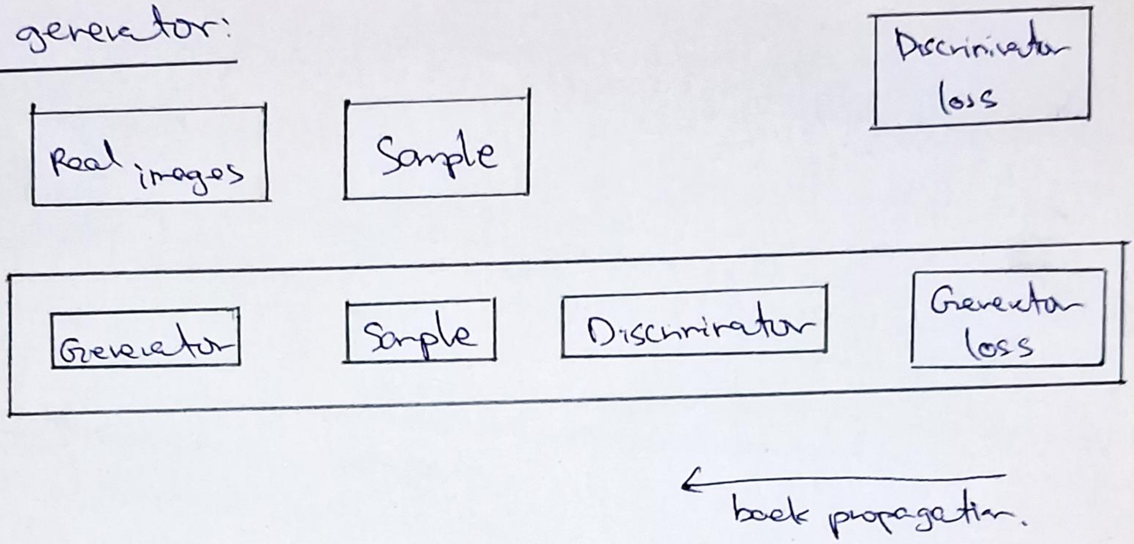
The loss fuction themselves are deceptively simple:

critic loss: $D(x) - D(G(z))$

Generator loss: $D(G(z))$

* $D(x)$ is critic output
* $G(z)$ is generator's output.
* $D(G(z))$ is critic output
* Output of critic does not have to be between 1 and 0.

The generator:



* The discriminator in a GAN is simply a classifier. It tries to distinguish real data from data created by the generator.

Training the model discriminates:

- classifier both real and fake data.
* Discriminator loss normlize the discriminates for misclassifying a real or fake or real data.
* Discriminator updates its weight through back propogation from discrimination less though discriminator network.