Name: Pradeesh Kumar G

Reg.No.: 412522248118

Dept : AI-DS - 'C'-section

Roll no: 82.

SECID: SEC22AD173

Subject: ADVANCED MACHINE LEARNING

Subject code: 20AIPC403

Date: 24.04.2024.

Part-A

1. Generative model.

→ A generative model is a type of statistical model that is designed to generate or produce new data that is similar to the data it was trained on.

→ It learns the underlying structure of the data and then uses that knowledge to create new examples that resemble the original data.

→ Generative models are widely used in various fields, including ML, NLP, computer vision etc

→ Example: GAN, chatgpt, Gemini, H_2O.ai etc.

2.

Forward Sampling	~~backward~~ Importance sampling.
• Draws samples directly from the target distribution	• Draw samples from a proposal distribution & reweights them
• simple to implement	• Effective for estimation properties of complex distributi
• suitable for simple distribution where direct sampling is feasible	• Requires choosing a suitable proposal distribution
• Inefficient for high-dimensional or complex distribution	• more efficient than forward sampling when good proposal distribution is choosen
• May require rejection sampling for non-straight forward distribution.	• challenging in high-dimensional space

3. Back Propagation.

→ Back propagation is a key algorithm used in training artificial neural networks, particularly in the context of gradient-based optimization methods like gradient descent.

→ It is a method for efficiently computing the gradient of the loss function with respect to the weights of the network.

4. Approximate inference.

→ Approximate inference refers to a set of techniques used in probabilistic graphical model and Bayesian statics to estimate the posterior distribution of the model's variables when exact inference is computationally infeasible or impractical.

→ It exact inference requires computing the joint distribution of all variables in the model, which can become exponentially complex as the number of variables increases

→ In approximate inference, instead of computing the exact posterior distribution, we aim to finde approximation to this distribution that is computationally tractable and reasonably accurate.

5. Auto regressive model in density estimator.

→ An auto regressive model is a type of generative model that estimates the probability density function (PDF) of a multidimensional random variable by modelling the conditional probabilities of each variabe given the values of previous variables.

→ Autoregressive model are particularly effective for modelling high-dimensional data with complex dependencies, such as images, audio and time series data.

→ They have been successfully applied in various domains, including image processing, generation, speech synthesis, and NLP.

→ Example of auto regressive models:
- Autoregressive Moving Average (ARMA)
- Auto regressive Integrated Moving Average (ARIMA) etc.

Part - B

6.a) Variational auto encoders :=

- VAE interpretable as a directed latent-variable probabilistic graphical model.

- A variational autoencoder (VAE) provides a probabilistic manner for describing an observation in latent space.

- 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 :=

- Autoencoders are a type of neural network that learns the data encodings from the dataset in an unsupervised way.

- ~~It~~ basically contains two parts:

  i) The first one is an encoder which is similar to the convolutional neural network except the last layer.

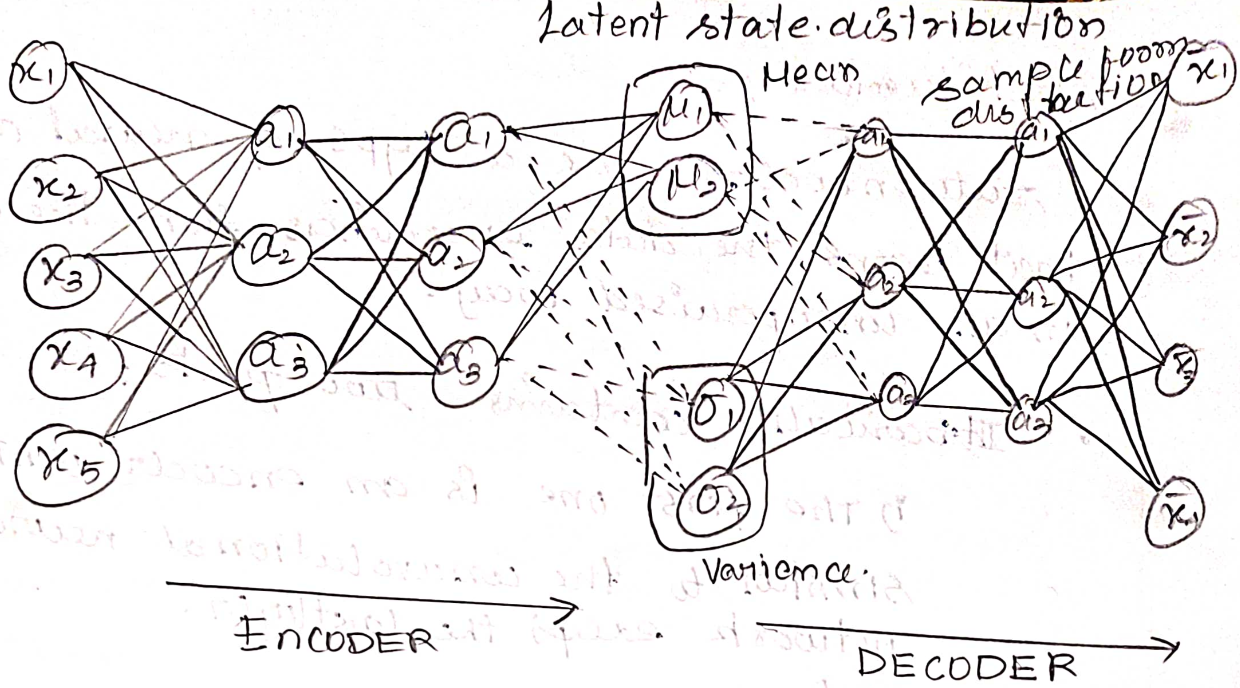
  ii) layer

- The aim is to encode ~~and~~ to learn efficient data encoding from the dataset and pass it into bottleneck architecture.

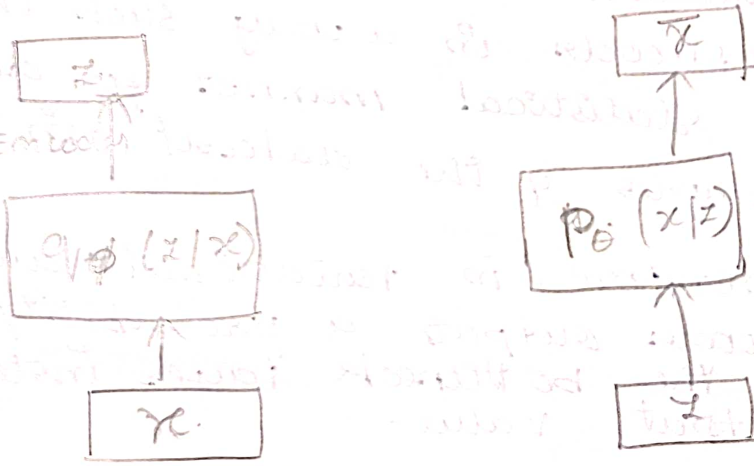
- The other part of the autoencoder is a decoder that uses latent space in the bottleneck layer to regenerate the images similar to the dataset.

- Variational autoencoder is different from autoencoder in a way such that it provides a statistical 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 image using another neural network.
- Components of Variational auto encoder
  - Encoder
  - Decoder
  - Regularised loss function



Encoder:

- It takes training data as input and Produces a latent representation z.

- The latent representation is stochastic i.e. they are parameters of a probability distribution.

- $q\phi$ 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

$$-D_{KL}[q\phi(z|x) \,||\, P_\theta(z)] + E_{q\phi}(z|x)[\log(P_\theta(x|z)]$$

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

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 distribution 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 autoregressive flow for Density estimation.

Masked autoregressive Flow (MAF) is a powerful method for density estimation, particularly in high-dimensional space.

Developed as a type of normalizing flow model, MAF relies on autoregressive modeling to capture the complex dependencies within the data.

1. Intro to Density estimation:

- Density estimation is the task of estimating the probability density function (PDF) of a dataset.
- In high-dimensional spaces, traditional methods like kernel density estimation become computationally infeasible.

2. Normalizing Flows:

- Normalizing flow are generative models that transforms samples from a simple distribution (eg. Gaussian) into samples from a complex target distribution

· These Transformation are typically invertible, allowing for exact computing of densities and sampling.

$$p(x) = \prod_{i=1}^{D} p(x_i | x_{<i})$$

3. Auto regressive Modelling

· Auto regressive models decompose the joint distribution of variables into a product of conditional distributions.

· Each variable is modeled as a function of the previous variables, capturing complex dependencies.

$$P(x) = \prod_{i=1}^{D} P(x_i | x_{<i})$$

4. Masked Autoregressive Flow (MAF):

· MAF is a normalizing flow model based on auto regressive modelling.

· It applies a series of invertible transformations to a base distribution to generate samples from the target distribution.

· MAF uses masked neural networks to parameterize the conditional transformation.

5. Key components of MAF:

- Autoregressive coupling layers: These layers decompose the input into two parts: one that is transformed and one that remains unchanged.

- Invertible Transformation: Each coupling layer is designed to be invertible, enabling exact computation of densities and efficient sampling.

6. Training MAF

- MAF is trained by maximizing the likelihood of the training data using techniques like maximum likelihood estimation.

- Gradients are computed using back propagation through the entire model.
- Training often involves optimization algorithm such as stochastic gradient descent (SGD) or variants like adam.

7. Application.

- MAF is applied in various domains requiring density estimation, such as image modelling, text generation and anomaly detection.

- It has shown impressive performance in capturing complex data distributions and generating high-quality samples.

8. Advantages:

MAF can model complex data distributions in high-dimensional spaces, provides exact density computation, and supports efficient sampling.

9. Limitation:

Training can be computationally expensive, especially for large datasets and complex models

Masked autoregressive Flow is an effective approach for density estimation, leveraging autoregressive modelling and invertible transformations to capture complex data distribution

Part-C

8. a) Monte carlo method. (or) Markov chain Monte carlo (MCMC)

Markov chain monte carlo sampling is a powerful computational technique used for approximating complex probability distributions and performing inference in statistical models.

It is widely used in fields such as physics, computational biology, and ML.

The MCMC method combines two key concepts: Markov chains and Monte carlo simulation.

1. Markov chains: A Markov chain is a sequence of random variables where the distribution of each variable depends only on the state of the previous variable.

2. Monte carlo Simulations:

Monte carlo methods are a broad class computational algorithms that rely on repeated random sampling to obtain numerical results. They allow us to approximate complex mathematical Problems using random numbers.

In context of MCMC, these two concepts are combined to create a method where the next sample is generated based on the current sample. This is particularly useful for sampling from complex, high - dimensional distributions.

The MCMC algorithm work by constructing a Markov chain of states in such a way that the stationary distribution of the chain of ~~states~~ is the target distribution we want to sample from. After a sufficient number of steps, the Markov chain will converge to the stationary distribution, and the states of the chain can be used as samples from the target distribution.

There are several types of MCMC methods,

- Metropolis - Hastings algorithm:

This is one of the earliest and simplest MCMC algorithm. It proposes a new state, and then decides whether to accept this new state or stay with the current state

- Gibbs sampling:

This is a special case of the Metropolis - Hastings algorithm that is particularly useful when we can easily sample from the conditional distribution of each variable given the others.

- Hamiltonian Monte Carlo (HMC): This is a more advanced MCMC method that uses concepts from physics to propose new states, which can lead to more efficient sampling for certain types of target distribution.

MCMC methods have revolutionized Bayesian statistics and made it possible to perform complex Bayesian inference that was previously intractable.

Example:

Metropolis - Hastings algorithm.

-> Metropolis - Hastings algorithm is a type of Markov chain Carlo method.

Python implimentation of (MHA)

```
import numpy as np
mu=0
sigma=1
chain = [0.5]
num_samples = 10000
for _ in range(num_samples):
    current_state = chain[-1]
    proposed_state = np.random.normal
        (current_state, 0.5)

    p_acceptt = min(1.0, np.exp(-0.5*((proposed
        state -mu)/sigma)**2)/np.exp(-0.5*((
        current_state -mu)/sigma)**2))
if np.random.rand() < p_accept:
        chain.append(proprosed_state)
else:
    chain.append(current state)
```

This code initializes a Markov chain and then enters a loop where it proposes new state and decideds whether to acupt them. The proposal is just the current state plus some normally - distributed noise, and the acceptance probability is the ratio of the target distribution's density at the proposed state to it density at the current state.

After running this code, the chain list contains a sequence of values that approximates samples from the Normal (0, 1) distribution.

~~Please note that th~~