In the Name of God

Deep Generative Models (25120)

Problem Set 1

Fall Semester 1404-05

Department of Electrical Engineering

Sharif University of Technology

Instructor: Dr. Sajjad Amini

Due on: According to CW

(∗) starred problems are optional!

1. Properties of Gaussian Distributions

In this section we are going to review some basics and take a look at the properties of the Gaussian distribution!

1.1 Linear Transformation on Gaussians

Let $\vec{X} \in \mathbb{R}^d$ be Gaussian: $\vec{X} \sim \mathcal{N}(\vec{\mu}, \Sigma)$. Let A be a fixed matrix $(m \times d)$ and $\vec{b} \in \mathbb{R}^m$.
Show that $\vec{Y} = A\vec{X} + \vec{b}$ is Gaussian and give its mean and covariance.

1.2 Sum of Independent Gaussians

Let $\vec{X} \sim \mathcal{N}(\vec{\mu}_X, \Sigma_X)$ and $\vec{Y} \sim \mathcal{N}(\vec{\mu}_Y, \Sigma_Y)$ be independent vectors in $\mathbb{R}^d$.
Show that $\vec{Z} = (a\vec{X} + b\vec{Y})$ is Gaussian and compute its mean $\vec{\mu}_Z$ and covariance Σ_Z.

1.3 Conditional Gaussian (Marginalization)

Let $\vec{X} \in \mathbb{R}^{d_x}$ and $\vec{Y} \in \mathbb{R}^{d_y}$. Suppose $\vec{X} \sim \mathcal{N}(\vec{\mu}_X, \Sigma_{XX})$ and conditional on $\vec{X}$,
$(\vec{Y} \mid \vec{X}) \sim \mathcal{N}(A\vec{X} + \vec{b}, \Sigma_{Y|X})$ (here A is $d_y \times d_x$ and note that $\Sigma_{Y|X}$ is the covariance matrix for $(\vec{Y} \mid \vec{X})$.
Show the marginal distribution of $\vec{Y}$ is Gaussian and compute its mean and covariance.

1.4 General Joint Gaussian Conditional Distribution

Let $\begin{pmatrix} \vec{X} \\ \vec{Y} \end{pmatrix}$ be jointly Gaussian with mean $\begin{pmatrix} \vec{\mu}_X \\ \vec{\mu}_Y \end{pmatrix}$ and covariance blocks $\Sigma = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix}$.
Assume Σ_{XX} invertible.
Derive the conditional distribution $(\vec{Y} \mid \vec{X} = \vec{x})$ (mean and covariance).

1.5 KL Divergence between Two Multivariate Gaussians (Closed Form)

KL divergence is one of the mathematical tools for measuring the difference between two probability distributions. Write down the KL divergence $D_{KL}\big(\mathcal{N}(\vec{\mu}_0, \Sigma_0) \,||\, \mathcal{N}(\vec{\mu}_1, \Sigma_1)\big)$ for **full-rank** covariances $\Sigma_0, \Sigma_1 \in \mathbb{R}^{d \times d}$. Write it down for the case $d = 1$ as well.

2. Autoregressive (AR) Models of Order p

An Autoregressive model of order p, denoted as AR(p), is used to describe a time-dependent process where the current value depends linearly on its previous p values plus a random error term. The model is expressed as:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

where: y_t is the observation at time t, $\phi_1, \phi_2, \ldots, \phi_p$ are the parameters (coefficients) of the model and ε_t is the error term at time t, assumed i.i.d. with a normal distribution $N(0, \sigma^2)$.

You are required to estimate the parameters $\phi_1, \phi_2, \ldots, \phi_p$ and σ^2 that maximize the likelihood function given the observed data $y_1, y_2, \ldots, y_n$.

2.1 Likelihood Function

Formulate the likelihood function for the AR(p) model based on the assumption that the error terms are normally distributed and independent. Then derive the log-likelihood function in terms of $\phi_1, \phi_2, \ldots, \phi_p$ and σ^2.

2.2 Maximum Likelihood Estimation (MLE)

Find the set of equations for parameter values that maximize the log-likelihood function and provide explicit formulas for the estimators of $\phi_1, \phi_2, \ldots, \phi_p$ and σ^2. Include detailed derivations of the likelihood and log-likelihood functions and explain each step clearly.

2.3 linking AR(p) and Markov property

Prove that an AR(1) process

$$y_t = \phi y_{t-1} + \varepsilon_t,$$

where $\{\varepsilon_t\}$ is a white noise sequence, is a (time-homogeneous) Markov process of order 1.

(a) Derive the one-step transition density $p(y_t \mid y_{t-1})$.

(b) Assuming $|\phi| < 1$, derive the stationary distribution $\pi(y)$ of y_t and show its mean and variance.

3. Nonlinear and Multi-Step Autoregressive Models

In this problem, you will study nonlinear extensions of autoregressive (AR) models and explore how to use them for prediction and sequence generation. You will also examine issues of stability and long-term behavior in autoregressive generative models.

3.1 Nonlinear Autoregressive Model

Consider a sequence of real-valued random variables $x = (x_1, x_2, ..., x_T)$. Suppose we model the conditional distribution of each x_T given its past as:

$$p_\theta(x_t | x_{<t}) = \mathcal{N}(f_\theta(x_{t-1}, ..., x_{t-p}), \sigma^2)$$

where: f_θ is a nonlinear function parameterized by a neural network with parameters θ, σ^2 is a fixed variance and p is the model order. This model is referred to as a nonlinear autoregressive (NAR) model.

- Derive the NLL(Negative Log-Likelihood) of the observed data sequence $x_1, x_2, ..., x_T$ under this model.

- Then explain how the gradients of the NLL with respect to θ can be computed using backpropagation through time.
- Discuss how the autoregressive structure affects gradient flow, especially for large p or long sequences T.

3.2 Architecture for large vs. small context

Assume that the model order p can vary from very small (e.g. $p = 1$) to very large ("$p \gg 1$", full context).

- When $p \gg 1$, what types of neural architectures would you recommend for the nonlinear function f_θ? Discuss design choices and practical limitations.
- When p is small, what architectures are more suitable and why? Compare them in terms of accuracy, computational efficiency, and training stability.

3.3 Multi-Step Prediction

In practice, we often want to predict not just the next value x_{t+1}, but several steps into the future. Let $\hat{x}_{t+k}$ denote the model's k-step-ahead prediction at time t, obtained recursively by feeding back previous predictions into the model.

- Write down the recursive equations for generating k-step predictions $\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+k}$ given the trained model $f_\theta(.)$ and the most recent p true observations $x_{tp+1:t}$.
- Explain mathematically why prediction errors in autoregressive models tend to accumulate over multiple steps ahead & Propose a way to mitigate this issue when training or evaluating the model.

3.4 Conditional Nonlinear Autoregressive Modeling

Suppose now you want to model a sequence $x = (x_1, x_2, ..., x_T)$ conditional on another sequence $c = (c_1, c_2, ..., c_T)$. Define the conditional AR model:

$$p_\theta(x_t | x_{<t}, c_{1:t}) = \mathcal{N}(f_\theta(x_1, ..., x_{t-p}; c_{1:t}), \sigma^2)$$

- Derive the expression for the conditional log-likelihood of the full sequence $x_{1:T}$ given $c_{1:T}$, and write the training objective that maximizes this conditional likelihood.
- Explain how this model could be used for conditional generation (e.g., generating speech given text, or motion given control inputs).
- Compare conditional and unconditional AR models in terms of: The factorization of their joint distributions, The dependence structure, The types of data they can model.

3.5 KL Divergence between two conditional NAR models

Consider two conidiotnal autoregressive models, one parameterized by f_θ and another by g_ϕ. Write down the expression for the Kullback-Leibler (KL) divergence between the two distributions $p_\theta(x)$ and $q_\phi(x)$. Discuss the computational challenges of directly evaluating this divergence and propose a practical approximation method for high-dimensional sequences.
(Hint: You can describe Monte Carlo sampling here.)

4. Real NADE Parameters

In this problem, you will study an extension of the Real NADE model. Recall that, given an autoregressive model

$$p(x) = p(x_1)p(x_2 \mid x_{<2}) \dots p(x_i \mid x_{<i}) \dots p(x_n \mid x_{<n})$$

and Real NADE models the conditional distribution as

$$p(x_1) = \mathcal{N}(x_1 \mid \mu_1, \exp(s_1))$$

$$\dots$$

$$p(x_i \mid x_{<i}; W, c, v_i, b_i, u_i, d_i) = \mathcal{N}\left(x_i \mid v_i^\top h_i + b_i, \exp(u_i^\top h_i + d_i)\right),$$

where $h_i, c, v_i, u_i \in \mathbb{R}^d$. Now, we would like to make $p(x_i \mid x_{<i})$ follow a mixture of Gaussians:

$$p(x_i \mid x_{<i}) = \sum_{c=1}^{C} \pi_i^c \mathcal{N}(\mu_i^c, (\sigma_i^c)^2),$$

where $\sum_{c=1}^{C} \pi_i^c = 1$. Now the question is: How do you propose to parameterize $\pi_i^c, \mu_i^c, \sigma_i^c, \forall c \in \{1, \dots, C\}$ as a function of h_i? Describe the parameters required and the total number of parameters required for a single $p(x_i \mid x_{<i})$.

5. (∗) Autoregressive Graph Neural Networks for Sequential Data

For this problem, we will explore the concepts of Graph Neural Networks (GNNs) and the Message Passing framework.

A **Graph Neural Network (GNN)** is a class of neural networks designed to operate on graph-structured data, where nodes represent entities and edges represent relations between them. Unlike traditional neural networks that assume fixed-size vector inputs, GNNs leverage the graph topology to learn node-, edge-, or graph-level representations.

For a multivariate time series with d variables, treat each variable as a node in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let the autoregressive model update node features via

$$h_t^v = \text{GNN}\big(h_{t-1}^v, \{h_{t-1}^u : u \in N(v)\}\big).$$

1. Derive the joint likelihood factorization

$$p(x_{1:T}) = \prod_{t=1}^{T} \prod_{v \in \mathcal{V}} p(x_t^v \mid x_{<t}^v, x_{<t}^{N(v)}).$$

2. Suppose that at test time, only a subset of node values is observed. Let $m_t^v \in \{0, 1\}$ denote a binary mask indicating whether node v's value at time t is *observed* ($m_t^v = 1$) or *missing* ($m_t^v = 0$). The observed set is $\mathcal{O}_t = \{v : m_t^v = 1\}$ and the missing set is $\mathcal{M}_t = \{v : m_t^v = 0\}$.

   - Propose a training objective that remains well-defined when only the observed node features $x_t^{\mathcal{O}_t}$ are available, and discuss how masking can be incorporated during training, encouraging robustness to missing inputs.
   - Describe how, at generation (or test) time, one can infer or estimate the missing node values $x_t^{\mathcal{M}_t}$ using the learned conditional distributions.