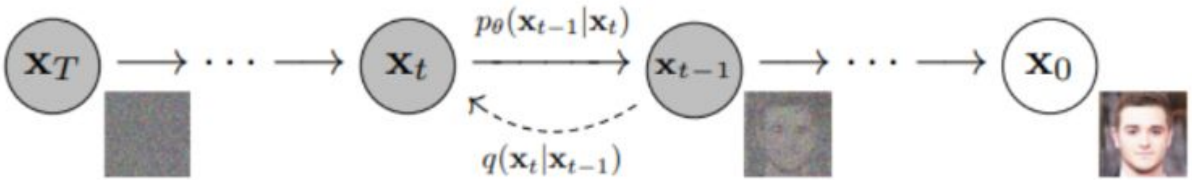
DeepLearning
HW2

Diffusion Model

Denoising Diffusion Probabilistic Model

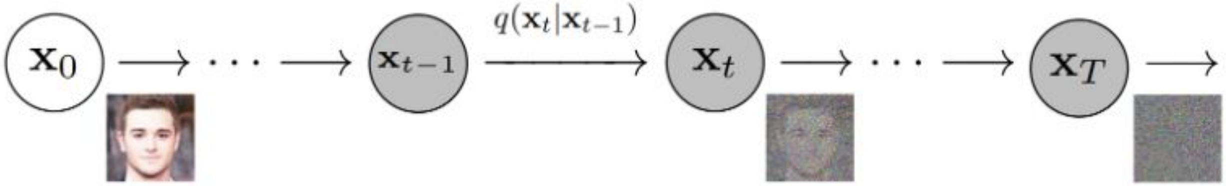
Denoising Diffusion Probabilistic Models

- Diffusion Model consists of two processes:
  - A forward diffusion process that converts data to noise by gradually adding noise to the input
  - A reverse generative process that starts from noise and generates data by denoising one step at a time



DDPM Forward Process

- The forward process or diffusion process, is fixed to a Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$



DDPM Forward Process

- The DDPM forward step and process are defined as:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$$

- We can express $\boldsymbol{x}_t$ as a linear combination of $\boldsymbol{x}_0$ and a noise variable $\boldsymbol{\epsilon}$
- By a sequence of diffusion steps from $\boldsymbol{x}_0$ to $\boldsymbol{x}_t$, we have the Forward/Diffusion Process:

$$q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right) := \prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$$

DDPM Reverse Process

- The reverse process of is also defined as a Markov chain
- Given the forward process above, we define the Reverse step as the inverse step with respect to learned Gaussian transitions parameterized by θ
- Here a neural network $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$ is used to fit the distribution $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right):=\mathcal{N}\left(\mathbf{x}_{t-1} ; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right)$$

DDPM Training Objective

- Diffusion Training Objective: By minimizing the negative log-likelihood (NLL)

$$
\begin{aligned}
\mathbb{E}\left[-\log p_\theta\left(\mathbf{x}_0\right)\right] &\leq \mathbb{E}_q\left[-\log \frac{p_\theta\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)}\right] \\
&= \mathbb{E}_q\left[-\log p\left(\mathbf{x}_T\right)-\sum_{t \geq 1} \log \frac{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}\right] \\
&= \mathbb{E}_q[\underbrace{D_{\mathrm{KL}}\left(q\left(x_T \mid x_0\right) \| p\left(x_T\right)\right)}_{L_T} \\
&\quad+\sum_{t>1} \underbrace{D_{\mathrm{KL}}\left(q\left(x_{t-1} \mid x_t, x_0\right) \| p_\theta\left(x_{t-1} \mid x_t\right)\right)}_{L_{t-1}} \\
&\quad \underbrace{-\log p_\theta\left(x_0 \mid x_1\right)]}_{L_0}=: L
\end{aligned}
$$

DDPM Training Objective

- Simplified training objective

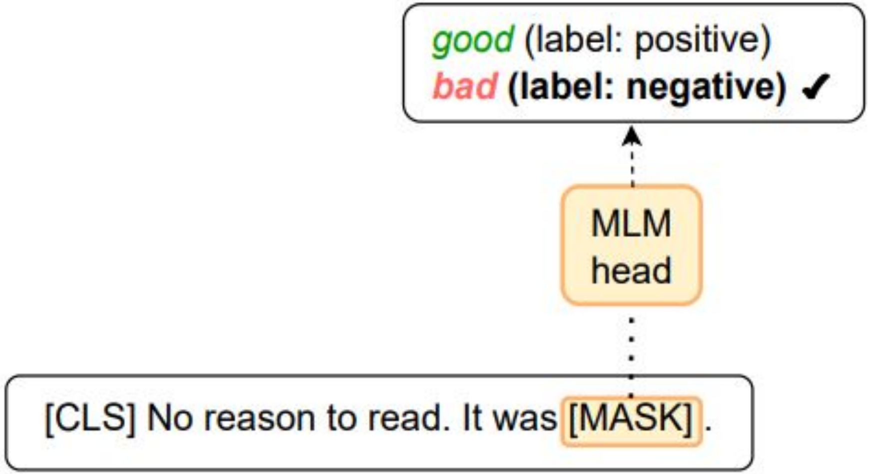
$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right]$$

Utilizing pre-trained Model

Prompt-based Learning

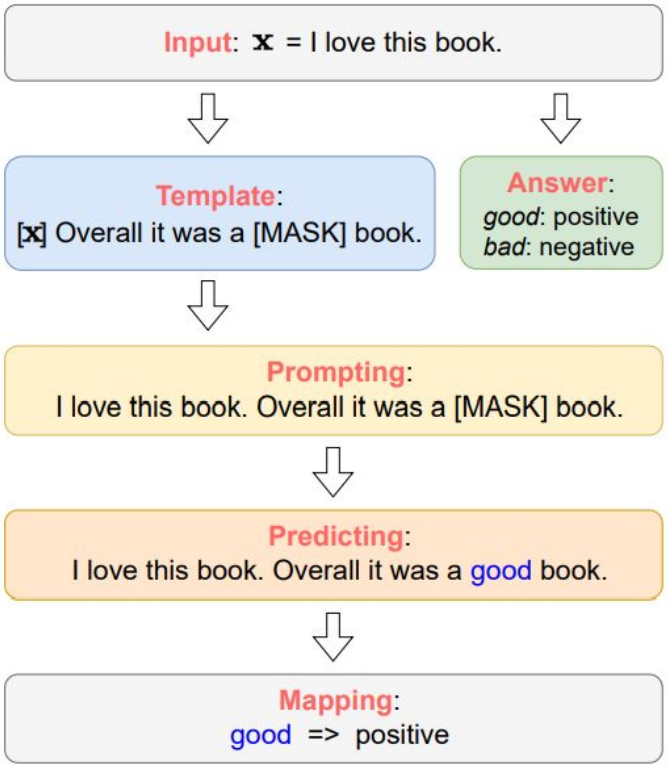
Prompt-based Learning

- Pre-train and Fine-tune has been used as a standard learning paradigm for years in NLP
- Prompt-based learning makes better use of the knowledge from PLM
- The prompt-based learning objective is matched with the pre-training MLM objective



Prompt-based learning

Prompt-based Learning Workflow



Prompting

- Technique of making better use of the knowledge from the pre-trained model by adding additional texts to the input

- Prompting can be divided into two types:
    - **Hard** (discrete) prompt : the template consists of actual tokens
    - **Soft** (continuous) prompt : the template consists of pseudo tokens which are trainable
    - **Hybrid** prompt : the template consists of hard and soft prompt

Input: $\mathbf{x}$

hard prompt $\mathbf{x}$ It was [MASK]

soft prompt $\mathbf{x}\ [v_1\, v_2 \cdots v_k]$ [MASK]

Hard Prompt

- Handcrafting a suitable prompt requires a solid understanding of the task and testing data
- Trial-and-error is required for obtaining an effective prompt, but this is time-consuming
- It is necessary to train the whole model when using hard prompt on a pre-trained language model

Soft Prompt

- Used in natural language processing (NLP) to guide a language model to generate text that is more relevant or specific to a particular task or context

- Different from hard prompt which is providing a specific sentence or instruction to generate a response. Soft prompt is more like guiding the model with a general idea or context

- Soft prompt has been proposed to alleviate the effort of handcrafting prompt

Difference between Fine-Tuning and Prompting