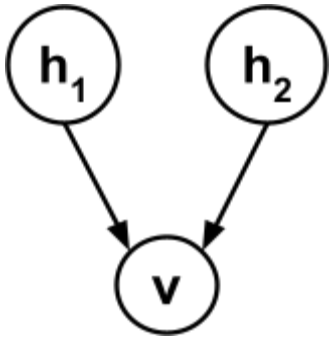
Homework 3

Question 1 - Consider the following Bayes Net where $h_1, h_2 \in \{0, 1\}$ are **latent** binary variables and $v \in R$ is real-valued. The CPDs are defined as



$$Pr(h_1 = 1) = Pr(h_2 = 1) = 0.5$$

$$p(v \mid h_1, h_2) = N(v;\ \mu = w_0 + w_1 h_1 + w_2 h_2;\ \sigma^2)$$

$$= 1/(\sqrt{2\pi}\,\sigma)\ \exp(-\ (v - w_0 - w_1 h_1 - w_2 h_2)^2\ /(2\,\sigma^2))$$

v^i
$v^1 = 1.0$
$v^2 = -1.0$

Derive the Evidence Lower Bound (ELBO) for a fully factorized variational posterior $q(h_1, h_2 \mid v) = q_1(h_1 \mid v)\, q_2(h_2 \mid v)$ for the following data.

Notice that a different q must be used for each training data (non-amortized). Use $Q^i_{jk} = q_j(h_j = k \mid v^i)$ as the parameters of the variational posteriors. Simplify as much as you can.

Question 2 - Coding Practice: Generation of Persian Digits using Image GPT

Implement an Image GPT, an autoregressive model, to generate binary images of Persian digits (0–9).

1.1. Data Preparation

- **You will be provided with a dataset containing binary images of Persian digits.**
- **Each image has a black background and a white foreground representing the digit.**
- **For simplicity and consistent results, you can resize the input image to 28x28.**
- **Shuffle and split the dataset into training, validation, and test sets with a ratio of 85%, 10%, and 5%, respectively.**

1.2. Model Implementation

First, read the article carefully to fully understand the concept and its workings.
To accommodate low-resource environments, we need to implement an extra-small version of Image GPT. For this purpose, you will modify the Image GPT-small architecture by reducing the number of layers according to the configuration provided below.

Embedding dim: 16

Number of layers: 8

Number of heads: 2

Number of vocab: 16

- **Apply any modifications that may enhance performance when working with the binary dataset.**

1.3. Training and Validation

- **Train the model using early stopping, monitoring both the training and validation loss to ensure they decrease monotonically (This lightweight architecture should train quickly, typically in less than an hour, even on a low-resource GPU).**

- **Finally, evaluate and verify the loss on the test set.**

1.4. Sampling

- **After training, perform sampling to generate new binary images of Persian digits.**

The output for class zero (0) would look similar to this example:

For the implementation, you are free to use the official codebase, but you must be able to interpret and explain every line of your code.

references

- Chen, Mark, et al. "Generative pretraining from pixels." *International conference on machine learning*. PMLR, 2020.