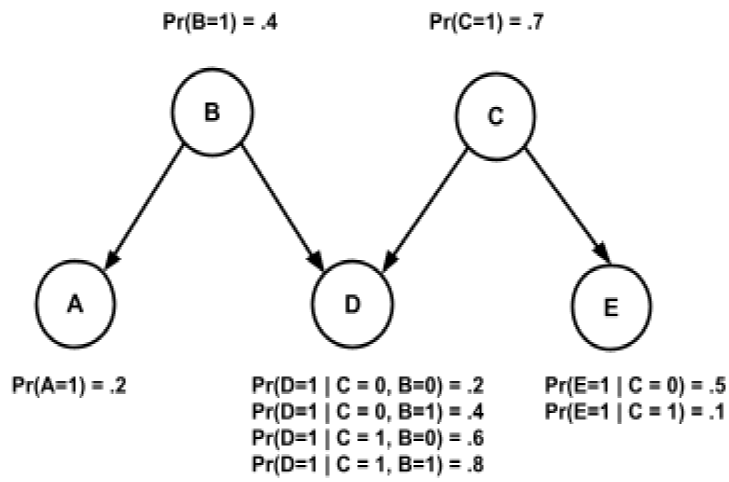
Homework 2

Q1- Bayesian Nets

Given the following network, calculate the probabilities below:

- P(E=1)
- P(C=0)
- P(C=1 | B=1)
- P(C=1 | B=1, D=0)
- P(D=0 | A=1)
- P(E=1 | A=1, D=1)



Q2- Markov chain

Consider a Markov Chain as above, prove

A) $\mathbf{X_t} \perp \mathbf{X_{t-4}} \mid \mathbf{X_{t-2}}$

B) $\mathbf{X_t} \perp \mathbf{X_s} \mid \mathbf{X_r}$ for $\mathbf{s<r<t}$

C) Given $\mathbf{X_{t-1}}$ and $\mathbf{X_{t+1}}$, $\mathbf{X_t}$ is conditionally independent of all other nodes.

You can only make use of the following for the proofs:

- Each node is independent of its non-descendants given its parents, and
- The joint distribution can be written as the product of the CPDs.

Q3- Markov Random Fields

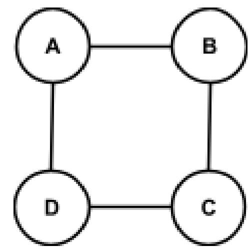
Consider the following Markov Random Field over variables $A, B, C, D \in \{-1, 1\}$. The potential functions are

$\phi_1(A, B) = \exp(A\,B)$

$\phi_2(B, C) = exp(1(B \neq C))$

$\phi_3(C, D) = exp(D + CD)$

$\phi_4(A, D) = exp(-AD)$



where, $1(\cdot)$ is the indicator function.

$$P(A, B, C, D) = 1/Z\ \phi_1(A, B)\ \phi_2(B, C)\ \phi_3(C, D)\ \phi_4(A, D)$$

1. obtain:
   a. the unnormalized measure $Z\,P(A, B, C)$
   b. the unnormalized measure $Z\,P(A, B)$
   c. the unnormalized measure $Z\,P(A)$
   d. the partition function Z (using the fact that $\sum_{A=-1}^{1} P(A) = 1$.
   - **in each case, simply your solution as much as you can**
2. Having Z, obtain the (normalized) distributions $P(A, B, C)$, $P(A, B)$,$P(A)$
3. Derive $P(A \mid B, C, D)$, and $P(A \mid B, D)$. Show that **A** is independent of **C** given **B,D**.

4- Coding Practice: Bayesian Network-Based Generation of Persian Digits

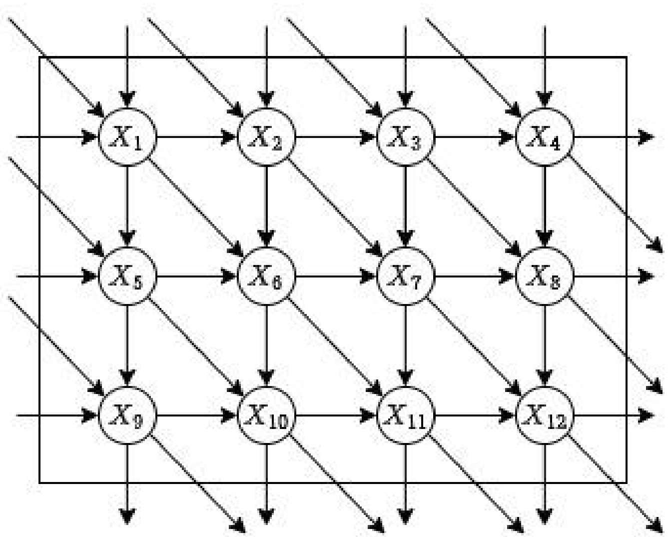
Implement a generative model using Bayesian Networks to generate binary images of Persian digits (0-9).

4.1. 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.**

4.2. Model Implementation



Example of a 3-Connected Graph with 12 Nodes

- **Construct a Bayesian Network where each pixel is a node.**
- **Connect each pixel to its parent pixels: the pixel above, the pixel to the left, and the pixel to the top-left.**
- **For boundary pixels (topmost and leftmost), assume connections to pixels outside the image boundary with a fixed value of 0.**
- **The resultant Bayesian network will be shown as a 3-connected graph (the number of nodes depends on the number of pixels in the image).**
- **Increase the graph connectivity to 8-connected and 15-connected and compare the results.**

4.3. Conditional Probability Distributions (CPDs)

- **Model the CPD for each pixel using a first-order linear sigmoid model (with shared parameters).**

$$P(X_i = 1 \mid parents) = \sigma(w_0 + W_1 X_{left} + W_2 X_{top} + W_3 X_{top-left})$$

Where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function.

- **Extend the model to a multi-layer perceptron (MLP) in later stages, first using one hidden layer and then two hidden layers with ReLU or SiLU activation functions. Compare the results and discuss the impact of hidden layers on the training procedure and the quality of generation for each scenario.**

4.4. Positional Features

- **Add two positional features (x and y coordinates) to each pixel, normalized between 0 and 1, treating them as additional parent nodes for each node.**
- **The leftmost pixel should have x = 0, and the rightmost pixel x = 1. Similarly, the topmost pixel should have y = 0, and the bottommost pixel y = 1.**
- **Consider this step for all settings and monitor the influence of positional embedding on the training procedure and generation. Construct a table for all circumstances.**

4.5. Cost Function (Negative Log-Likelihood)

The model parameters should be trained by minimizing the Negative Log-Likelihood (NLL):

$$L = -\sum_i x_i \log P(x_i \mid parents) + (1 - x_i) \log(1 - P(x_i \mid parents))$$

where the summation is over all pixels in all training images. This loss function encourages the model to maximize the probability of observing the actual data.

4.4. Training and Evaluation

- **Use PyTorch to implement the Bayesian Network and train it on the provided dataset of Persian digits.**
- **Optimize the model parameters using stochastic gradient descent (SGD) or Adam optimizer.**
- **The inference must be run after every epoch to generate digits and observe how the quality of generated images improves throughout training.**

4.7. Comparative Analysis

Students must generate results under different conditions and analyze them:

- **Linear Sigmoid vs. MLP: Train and compare results for the linear sigmoid model and MLP with different numbers of layers.**
- **Different Parent Configurations: Experiment with different numbers of parent nodes per pixel (3, 8, and 15).**
- **With vs. Without Positional Features: Compare results when including positional features versus when they are excluded.**

4.8. Sampling

- **After training, perform ancestral sampling to generate new binary images of Persian digits under each experimental setup.**

Deliverables

- **Well-documented Python code implementing the model.**
- **Generated digit samples for each experimental condition.**