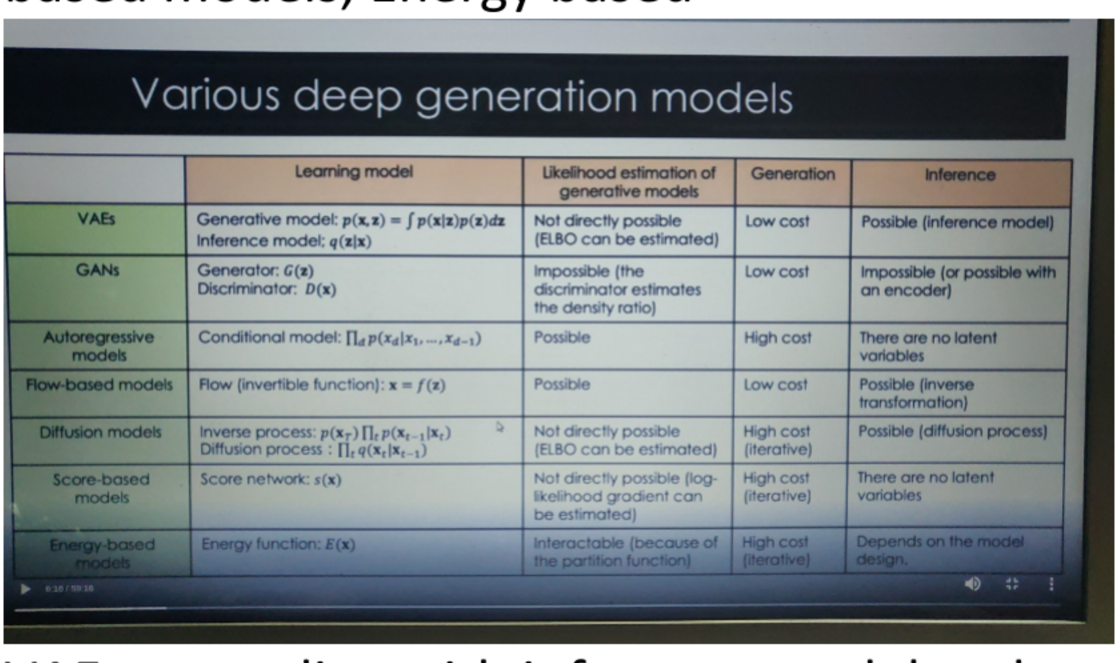
Deep Probabilistic Generative Models for Robotics-2

30 October 2020 16:44

- Probabilistic generated models assume that **observed variable** are generated from some stochastic models
- **latent variables** are often assumed to be the factors behind the observed variables
- Generative models can generate unseen data
- Can estimate how well the sample fits the model
- Outlier detection and anomaly detection
- Imputation(completing missing values of data)
- Deep Generative models-> prob distributions are parameterized by DNNs
    - Can be learned ennd to end from complex inputs
- DGM's -> VAE, GAN, Autoregressive models, FLow based models, Diffusion Models, score based models, Energy based
- 

Various deep generation models

	Learning model	Likelihood estimation of generative models	Generation	Inference
VAEs	Generative model: $p(\mathbf{x},\mathbf{z}) = \int p(\mathbf{x}\|\mathbf{z})p(\mathbf{z})d\mathbf{z}$ Inference model: $q(\mathbf{z}\|\mathbf{x})$	Not directly possible (ELBO can be estimated)	Low cost	Possible (inference model)
GANs	Generator: $G(\mathbf{z})$ Discriminator: $D(\mathbf{x})$	Impossible (the discriminator estimates the density ratio)	Low cost	Impossible (or possible with an encoder)
Autoregressive models	Conditional model: $\prod_d p(x_d\|x_1,\dots,x_{d-1})$	Possible	High cost	There are no latent variables
Flow-based models	Flow (invertible function): $\mathbf{x} = f(\mathbf{z})$	Possible	Low cost	Possible (inverse transformation)
Diffusion models	Inverse process: $p(\mathbf{x}_T)\prod_t p(\mathbf{x}_{t-1}\|\mathbf{x}_t)$ Diffusion process : $\prod_t q(\mathbf{x}_t\|\mathbf{x}_{t-1})$	Not directly possible (ELBO can be estimated)	High cost (iterative)	Possible (diffusion process)
Score-based models	Score network: $s(\mathbf{x})$	Not directly possible (log-likelihood gradient can be estimated)	High cost (iterative)	There are no latent variables
Energy-based models	Energy function: $E(\mathbf{x})$	Interactable (because of the partition function)	High cost (iterative)	Depends on the model design.

- VAE-> encoding with inference model and decoding with generative model
    - In GMS, representation learning is equivalent to inference
- Disentangled representation -> can be obtained by regularizing the inference
- Joint DGs -> modelling with joint distributions of modality
- Missing modality problem
    - JMVAE (jOINT multimodal variational autoencoder)
    - Obtain joint representation and perform bidirectional generation
- DNN implemented using tensorflow, keras, pytroch but many of deep generative model studies uses them as libraries but since they are not treated as probabilistic models it is difficult to implement complex deep generative models their implementation differs from person to person and tanks to difficult to read
    - **probabilistic programming languages** libraries for designing and learning about probabilistic models bye Ruby stone pytorch tensorflow flow probability and Edward based on tensorflow
- Implementing complex DGMs
    - Since they are difficult to implement free have to focus on some features of deep generative models one of them is that the Dnn composed the DGM's as are encapsulated by probability distributions. second model structure and regularisation are described in the objective function which is loss function which is optimised using gradient methods
- Pixyz- pytorch based library specialized for DGM

    Step by step implementation of API- Model , loss, distributed
    Code easy to read, reuse
    Speed not much slower than pytorch implementation
    Can be applied to bayesian Deep Learning

SERKET

- Multimodal information
- Ex- spatial concept acquisition- robot builds a map using SLAM and simultaneously learns
    - space name and appearance (MLDA)
    - Space region (GMM)
    - Parameters of speech recognition(Language models)
- Multimodal Learning = complementary learning
    - Speech recognition and clustering (LDA)
- Models are constructed by connecting small scale models
    - Latent variables ae shared with two models
    - Shared variable is determined with mutual influence-> parameters are optimized complementarily
- SERKET
    - Provides modularizing, connecting, inference
    - Shared latent variables are optimized with mutual inference on each model
    - Each module has observation and latent variables
    - Connected modules hierarchically
    - Parameter optimization
        - Message Passing (MP) approach
        - Sampling Importance Resampling(SIR) approach
    - Examples
        - VAE+GMM
        - VAE+GMM+MLDA
        - VAE+GMM+MLDA+MM
        - Model for language Acquistion
        - Model for learning object feature extractor by robots(MLDA+VAE)
- CONCLUSION
    - Easy to construct the integrated models
    - Accuracy improved by mutual influence of modules

-