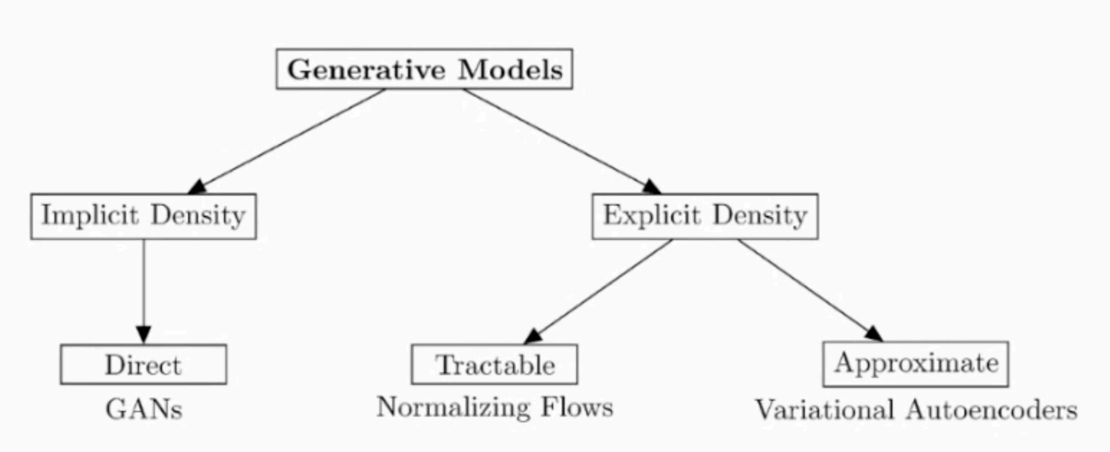
# Application of Deep Generative Modeling in Finance

12/22/20 6:25 PM

*Speaker: Achintya Gopal (Bloomberg Quant Finance)*

## What is generative modeling?

- Any model from which we can sample from




## Amortized Inference

**Example 1: Carbon modeling** (estimate a company's carbon emission)

- Data: Annually reported, 1000 companies report
- Estimation: Distributional estimates per company per year

- Traditional Approach:
  - Partition (bucket) the companies into similar subsets
  - Estimate Gamma GLM on subsets
  - Implicit assumptions
    - Every bucket is independent of every other bucket
  - Problems
    - What about country? Company size? Similar industries?
- Amortized Inference Approach
  - Model: train one model on ALL data
  - Loss: MLE instead of using Mean Squared Error
  - Output: a full distribution, conditioned on features
    - Each company has hundreds of features
    - Linear relationship is not justified for all features
- Benefits for amortized inference
  - Shares statistical strength across data
    - Some industries may have lots of data, some has little. Training a model on all data can help find companies that are similar across industry, rather than just within the industry itself
  - Implicitly buckets the data
  - Allows for more complex distributions
    - For example, multivariate distributions
- Generative Process:
  - $\text{Direct Carbon Emissions} \sim \text{Gamma}(k_1(X), \theta_1(X))$
    $\text{Indirect Carbon Emissions} \sim \text{Gamma}(k_2(X), \theta_2(X))$
  - Predict k1, Theta1, k2, Theta2 from Neural Network or any other models

**Example 2: LQA** (Liquidity Assessment )

- Goal: estimate trade cost
  - C = (P_trade - P_true) / P_true
- Data:
  - Only < 1% securities have more than 5 trades
- Model:
  - Estimate distribution
  - Create buckets based on volumes
  - Y-axis: Relative RMSE

**Interpretability**

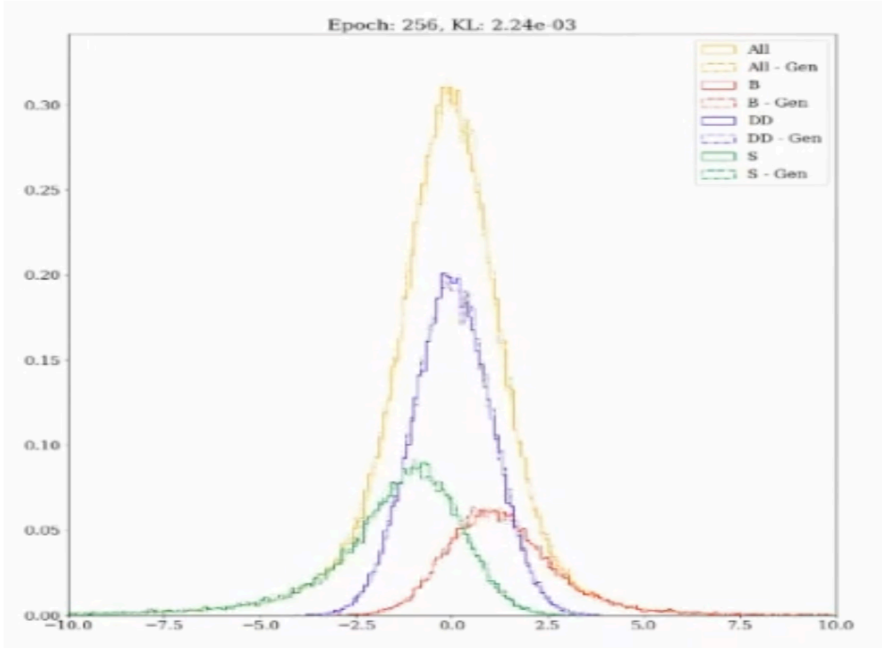
- What is we have a known relationship between some features and targets?
  - E.g. carbon emission is linear to company size
- Update generative model:
  - $\text{Gamma}(a_\phi(X) + S * b_\phi(X), \theta_\phi(X))$
    - Bias term: a
    - Slope between company size and carbon emission: S
    - Company size: b

## Applications of Deep Generative (state of art) Models

1. GANs (Generative Adversarial Networks)
   **Likelihood Free Estimation**

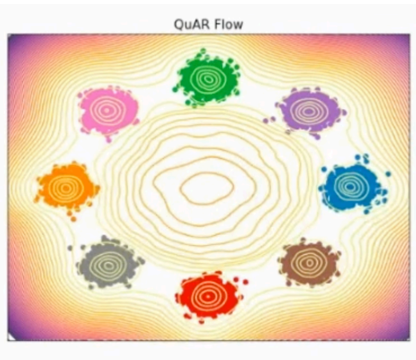
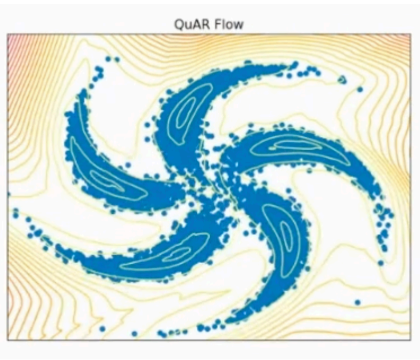
- Idea: Generator is trying to trick Discriminator
- Effectively minimized JS Divergence (JS: similarity between two distributions)

**Example: Directionless trade cost estimation**

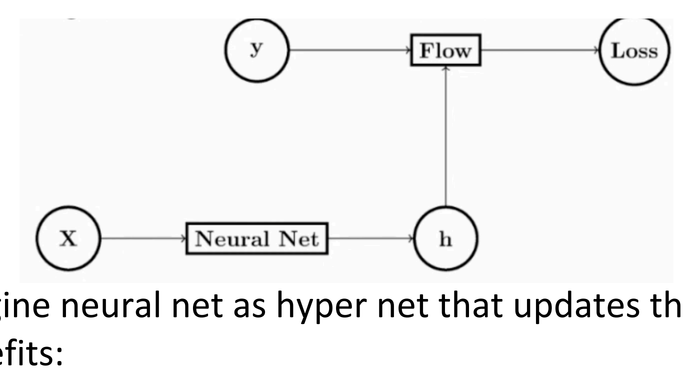
- Data is missing trade direction
  - Mixture of 3: dealer to dealer, dealer to client, client to dealer
  - But don't know which transaction is corresponding to what
- Estimation
  - Mixture of three components
    - Mixture weights (what proportion is for each component)
    - Dealer to Dealer: Normal (0, sigma_deal)
    - Dealer to client (buy): LogNormal (0, sigma_buy) + Normal (0, sigma_marketrisk)
    - Client to dealer (sell): - LogNormal (0, sigma_sell) + Normal (0, sigma_marketrisk)
    - LogNormal is charge (fee) from dealer to client
- Results:
- 

- Conclusion:
  - This is a way that allows us to know the generation process, without computing any closed-form likelihood

2. Normalizing Flows
   **More flexible distribution**
   (If we have no theoretical assumption for output, we should be able to use any free form distribution)

- Idea:
  - Deep learning method for modeling complex distributions
  - Function f(X) that maps complicated distribution to base distribution (e.g. uniform)
  - Likelihood is shown as
    - $\log p(x) = \log p(f(x)) + \log \left| \frac{\partial f(x)}{\partial x} \right|$
- Strength:
  - Can describe very complicated distributions
  - Doesn't require any prior
  - Likelihood doesn't even need to be based on data
- Examples from synthetic data:
  -  

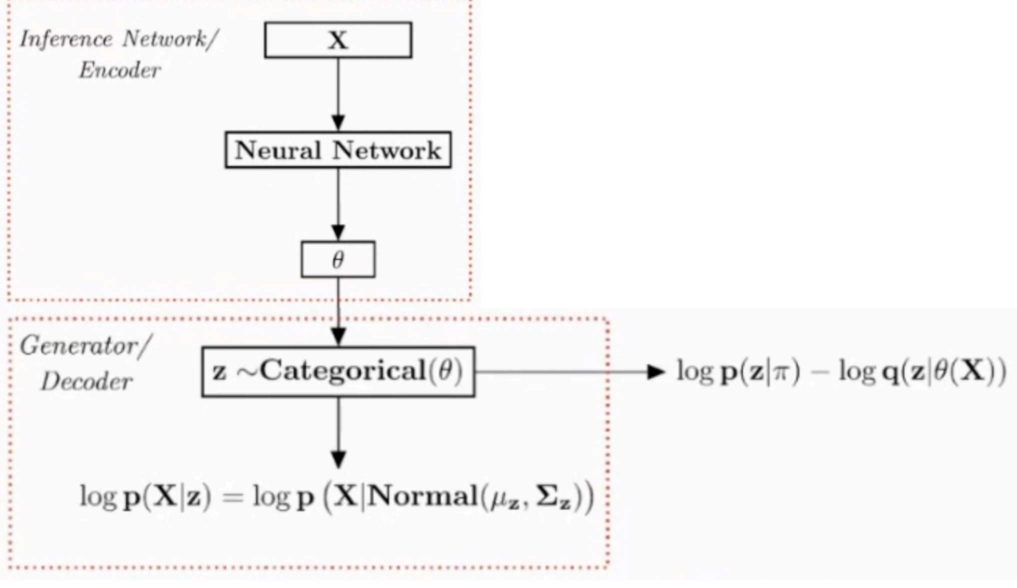
3. Conditional Flow

- Features enter neural networks, output representation (h)
- Flow is conditioned on representation
  - 

- Imagine neural net as hyper net that updates the parameter of the flow
- Benefits:
  - More data driven output distribution
  - Model high dimensional targets
    - More complex correlations

4. Variational Autoencoders (VAE)

- Goal: approximate posterior to perform MLE
- Traditional technique
  - Approximate posterior for all data points
  - Optimize parameters to maximize approximate likelihood
- Idea:
  - combine amortized inference, neural network, and Bayesian inference
  - Still input is features and output a distribution




- Benefits:
  - Uses less memory for posterior
  - Utilize other VAE research

## Summary

- Amortized inference is a lens to view how deep learning can be applied to generative processes
- Distilling deep learning research
  - GANs: likelihood free optimization of generative processes
  - Normalizing Flows: deep learning approach to learn complex distributions
  - VAEs: deep learning approach to enhance variational inference