Generative Adversarial Networks

Vikram Bhat[1]

[1]M.Tech CSE (CS16M057)
Indian Institute of Technology Madras

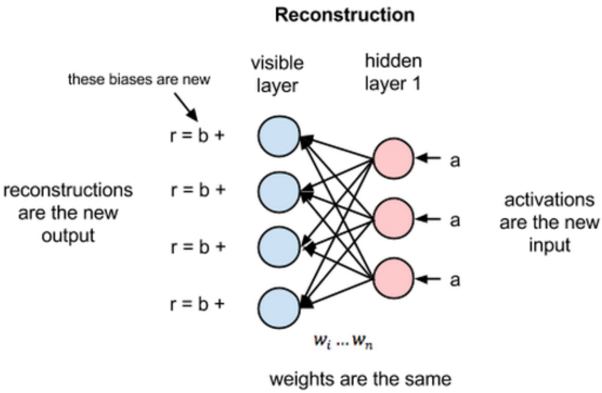
Seminar (CS5970), 2017-18

Outline

Generative Models

- Models which directly try to model the joint distribution of the input with class label in the form $P(x, c_k) = P(x|c_k)P(c_k)$
- Completely ignores the other class training example while model estimation.
- Maximizes log-likelihood $L(D, \theta, c_k) = \sum_{i=1}^{N} log(P(D_i, \theta, c_k))$
- As it models the joint distribution of the class label and training data , it is easier to sample data from the distribution to generate new data.

Example Generative Model: RBM

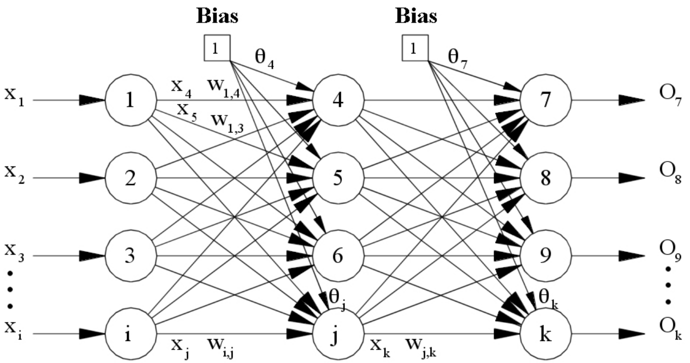


- Restricted Boltzmann Machine (RBM) is a generative model which tries to model $P(x) = \sum^{a} P(x, a) = \frac{1}{Z} \sum_a e^{-E(x,a)}$
- The Model is trained to maximizing log-likelihood which is seemingly difficult in this case , hence it is approximated using gibbs sampling.

Discriminative Models

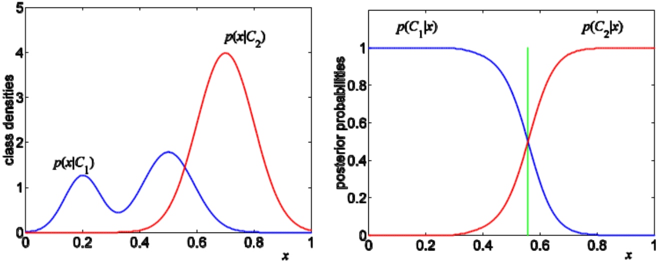
- Models the posterior probabilities given by $P(c_k|x)$ without computing the joint distribution $P(x, c_k)$
- Takes into account all the class examples be it of the targeted class or not and trains using loss function which is minimized when correct classification is done.
- Can be trained with more ease with method like gradient descent and back propagation.
- As it mainly focused on finding the region which divides the class examples into correct classes , it is ideal for classification but bad from generating new data.

Discriminative Model : Feed Forward Neural Network



- Trains to compute $O_i = P(C_i|x, \theta)$ by minimizing the loss function $L(t, x, \theta)$ which is typically squared error or cross entropy.
- Can be trained to desired level accuracy of classification using the backpropagation algorithm and gradient descent algorithms.

Generative vs Discriminative Models



- Discriminative Model creates a simplified distribution which only cares about the region of conflict while Generative Model cares about entire distribution of each class.
- Most of the data points in region don't belong to any class , hence in that case discriminative model is classifying them wrongly.

Generative vs Discriminative Models

- Sample taken from a generative model which is well trained has good chance of being an valid example of the class while in case of Discriminative it is unlikely to be valid.
- As Generative models are more powerful , they pay a price in training as training them is inefficient and often intractable.
- Training discriminative models to desired level of accuracy has been found easier and useful for many classification tasks.

Best of Both Worlds : Related Work

- As we have weighted out all the advantages and disadvantages of both models , the obvious question comes to mind is that can be improve the overall model by combined them both in such a way that they retain their advantages.
- **Deep Boltzmann Machine (DBM)** : It is extension of the RBM to more hidden layers like a stack of RBM's , it retains its generative properties if trained properly. It can also be used for classification by stacking a classification layer on top of it.
- **Deep Belief Networks (DBN)** : It is using stacked RBM with greedy training , which some of the generative properties of the DBM without the computation overhead.
- **Generative Stochastic Networks (GSN)** : It is extension to deep neural networks like Denoising Autoencoders which can be trained by Backpropagation but also requires Markov Chain sampling.

GAN : Introduction

- GAN pits two multilayered perceptron networks against each other, the **Generator**($G(z, \theta_g)$) tries to fool **Discriminator**($D(x, \theta_d)$) to think that data it generates comes from the real world training sets distribution.
- The job of the Discriminator is to verify the Generator's output and tell it apart from real data given to it in training.
- Both the generator and discriminator carrying on this game until the discrimator cannot tell the real data and generator's data apart.

GAN : Architecture

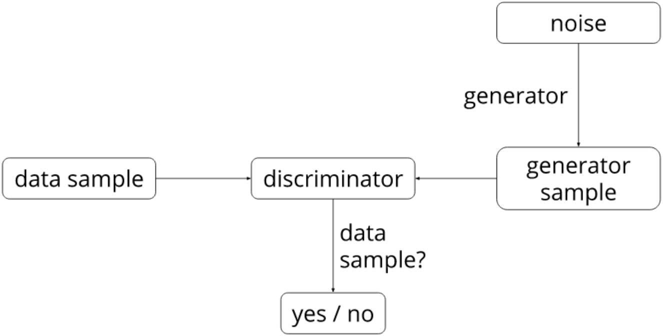


Figure: GAN Architecture

GAN : Architecture

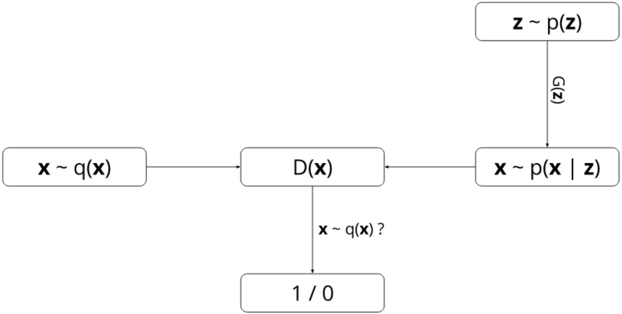


Figure: GAN Probabilistic View

GAN : Working Principle

- $V(D,G) = E_{x\sim p_{data}(x)}[log(D(x))] + E_{z\sim p_g(z)}[log(1 - D(G(z)))]$.
- The Generator and Discriminator play a minimax game on value given on $V(D,G)$.
- Generator tries to minimize $E_{z\sim p_{data}(z)}[log(1 - D(G(z)))]$.
- While Discriminator tries to maximize $V(D,G)$.
- As the value of $log(1 - D(G(z)))$ is very high intially as G is poorly trained the Generator's loss function is saturated so to prevent Generator maximizes on $log(D(G(z)))$ instead.

Training Algorithm

1: **for** number of training iterations **do**

2: **for** k steps **do**

- Sample minibatch of m noise samples $\{z^1, \dots, z^m\}$ from noise prior $p_g(z)$
- Sample minibatch of m data samples $\{x^1, \dots, x^m\}$ from data distribution $p_{data}(x)$
- Update the discriminator by ascending its stochastic gradient:
  $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [log(D(x)) + log(1 - D(G(z^i)))]$

3: **end for**

- Sample minibatch of m noise samples $\{z^1, \dots, z^m\}$ from noise prior $p_g(z)$
- Update the generator by descending its stochastic gradient:
  $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [log(1 - D(G(z^i)))]$

4: **end for**

Optimality

Global Optimality of $p_g = p_{data}$

For G fixed, the optimal discriminator D is

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

Which comes from maximizing

$$V(D,G) = E_{x \sim p_{data}(x)}[log(D(x))] + E_{z \sim p_g(z)}[log(1 - D(G(z)))]$$

$V(D,G)$ is maximized when $D_G^*(x) = \frac{1}{2}$ which only happens at $p_g = p_{data}$

Experiments : MNIST and TFD

Model	MNIST	TFD
DBN	138 ± 2	1909 ± 66
stacked CAE	121 ± 1.6	$\mathbf{2110 \pm 50}$
GSN	214 ± 1.1	1890 ± 29
GAN	$\mathbf{225 \pm 2}$	2057 ± 26

Table: Parzen window-based log-likelihood estimates. The reported numbers on MNIST are the mean log- likelihood of samples on test set, with the standard error of the mean computed across examples. On TFD, we computed the standard error across folds of the dataset, with a different σ chosen using the validation set of each fold

Experiments : Image Generation

Figure: Visualization of samples from the model. a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (CNN)

Advantages and Disadvantages

Advantages

- Involves simple deep neural networks in both the generator and discriminator hence can be trained using backpropagation and gradient descent.
- Faster convergence due to avoidance of maximum likelihood estimation which is intractable.
- Higher accuracy due to avoidance of markov chain sampling based estimation methods.

Disadvantages

- Tendency to overfit the data and converge at mean of target distribution which prevents any meaningful generative power.
- The Generator must be synchronized well with the Discriminator otherwise the gap becomes to big for it to recover and the model progress becomes unpredictable.

Future Work

- A conditional generative model $p(x|c)$ can be obtained by adding c as input to both G and D.
- *Semi-supervised learning:* features from the discriminator or inference net could improve performance of classifiers when limited labeled data is available.
- *Efficiency improvements:* training could be accelerated greatly by devising better methods for coordinating G and D or determining better distributions to sample z from during training.