Topics of This Lecture

• Recap: GANs
• Autoencoders
  – Motivation
  – Regularized Autoencoder
  – Denoising Autoencoder
• Variational Autoencoders (VAE)
  – Autoencoders as Generative Models
  – Intractability
  – Variational Approximation
  – Evidence Lower Bound (ELBO)
• Application Examples

Recap: Generative Adversarial Networks (GANs)

• Conceptual view
  – Simultaneously train an image generator $G$ and a discriminator $D$.
  – Interpreted as a two-player game

Recap: GAN Loss Function

• This corresponds to a two-player minimax game:
  $$\min_D \max_G V(D, G) = \mathbb{E}_{x \sim \text{data}} \log D(x) - \mathbb{E}_{z \sim p(z)} \log (1 - D(G(z)))$$

• Explanation
  – Train $D$ to maximize the probability of assigning the correct label to both training examples and samples from $G$.
  – Simultaneously train $G$ to minimize $\log (1 - D(G(z)))$.

• The Nash equilibrium of this game is achieved at
  $p_D(x) = p_{\text{data}}(x)$ $\forall x$
  $D(x) = \frac{1}{2}$ $\forall x$

GAN Algorithm

```
for k = 1 : max_iterations do
  for i = 1 : num_critic_updates do
    $\nabla_{x} \sum \log D(x) = \log (1 - D(G(z)))$   Discriminator updates
  end

  $\nabla_{z} \sum \log (1 - D(G(z)))$   Generator updates
end
```

This gradient-based updates can use any standard gradient-based learning rate. We used momentum in our experiments.
Recap: Intuition

- Behavior near convergence
  - In the inner loop, $D$ is trained to discriminate samples from data.
  - Gradient of $D$ guides $G$ to flow to regions that are more likely to be classified as data.
  - After several steps of training, $G$ and $D$ will reach a point at which they cannot further improve, because $p_g = p_{data}$.
  - Now, the discriminator is unable to differentiate between the two distributions, i.e., $D(x) = 0.5$.

Topics of This Lecture

- Recap: GANs
- Autoencoders
  - Motivation
  - Regularized Autoencoder
  - Denoising Autoencoder
- Variational Autoencoders (VAE)
  - Autoencoders as Generative Models
  - Intractability
  - Variational Approximation
  - Evidence Lower Bound (ELBO)
- Application Examples
Autoencoders

- After training
  - Throw away the decoder part

```latex
\text{Encoder: 4-layer conv}
\text{Decoder: 4-layer upconv}
```

Variants of Autoencoders

- Analyzing the learning process
  - Learning process minimizes a loss function $L(\mathbf{x}, g(f(\mathbf{x})))$
  - Linear decoder + $L_2$ loss: Autoencoder learns PCA subspace
  - Autoencoders with nonlinear encoder and decoder functions thus learn a more powerful nonlinear generalization of PCA.

- Regularized Autoencoders
  - Include a regularization term to the loss function: $L(\mathbf{x}, g(f(\mathbf{x}))) + \Omega(\mathbf{z})$
  - E.g., enforce sparsity by an $L_1$ regularizer $\Omega(\mathbf{z}) = \lambda \sum_i |z_i|$
### Variants of Autoencoders

**Denoising Autoencoder (DAE)**
- Rather than the reconstruction loss, minimize $L(x, g(f(x)))$
- where $f$ is a copy of $x$ that has been corrupted by some noise.
- Denoising forces $f$ and $g$ to implicitly learn the structure of $p_{data}(x)$.

### Topics of This Lecture

- Recap: GANs
- Autoencoders
  - Motivation
  - Regularized Autoencoder
  - Denoising Autoencoder
- Variational Autoencoders (VAE)
  - Autoencoders as Generative Models
  - Intractability
  - Variational Approximation
  - Evidence Lower Bound (ELBO)
- Application Examples

### Autoencoders as Data Generators

- **Autoencoders**
  - Can reconstruct data and can learn features to initialize a supervised model
  - Features capture factors of variation in training data
  - Can we generate new images from an autoencoder?

- **For this we need to generate samples from the data manifold. How?**

### Probabilistic Spin on Autoencoders

- **Idea:** Sample the model to generate data
- Assume training data $(x^{(i)})_{i=1}^N$ is generated from underlying latent representation $z$.

- Sample from true conditional $p_{θ}(x|z)$
- Sample from true prior $p_{θ}(z)$

- **Idea:** Sample the model to generate data
- We want to estimate the true parameters $θ$ of this generative model.
- **How should we represent the model?**
  - Choose prior $p(z)$ to be simple, e.g., Gaussian
  - Conditional $p(x|z)$ is complex (generates image)
  - Represent with neural network
Variational Autoencoders

- **Probabilistic Spin on Autoencoders**
  - Sample from true conditional $p_F(x|x^{(i)})$
  - Sample from true prior $p_U(x)$

- **Idea**: Sample the model to generate data
  - We want to estimate the true parameters $\theta^*$ of this generative model.

- **How to train the model?**
  - Learn model parameters to maximize likelihood of training data
  - What is the problem here? Intractable!

- **Picture**

Variational Autoencoders: Intractibility

- **Computing the data likelihood**
  \[ p_F(x) = \int p_F(z)p_F(x | z)dz \]

  - $p_F(x)$ is a simple Gaussian prior.
  - $p_F(x | z)$ is a decoder Neural network.
  - But is intractable to compute $p_F(x | z)$ for every $x$!

  - Posterior density is also intractable

  \[ p_F(x | z) = \frac{p_F(z)p_F(x | z)}{p_F(z)} \]

- **Solution**
  - In addition to the decoder network modeling $p_F(x | z)$, define additional encoder network modeling $q_F(z | x)$ that approximates $p_F(x | z)$.
  - We will see that this allows us to derive a lower bound on the data likelihood that is tractable and that we can optimize.

- **Picture**

Variational Autoencoders: Intractibility

- Since we are modeling probabilistic generation of data, encoder and decoder networks are probabilistic

- **Picture**

Variational Autoencoders

- **We can now work out the log-likelihood**
  \[ \log p_F(x^{(i)}) = \mathbb{E}_{q_F(z | x^{(i)})} [\log p_F(x^{(i)})] \quad (p_F(x^{(i)}) \text{ does not depend on } x) \]

  Taking expectation w.r.t. $z$ (using encoder network) will come in handy later.

- **Picture**

Variational Autoencoders

- **We can now work out the log-likelihood**
  \[ \log p_F(x^{(i)}) = \mathbb{E}_{q_F(z | x^{(i)})} [\log p_F(x^{(i)})] \quad (p_F(x^{(i)}) \text{ does not depend on } x) \]
  \[ = \mathbb{E}_z \left[ \mathbb{E}_{q_F(z | x^{(i)})} [\log p_F(x^{(i)}) | z] \right] \quad \text{(Bayes' Rule)} \]
  \[ = \mathbb{E}_z \left[ \mathbb{E}_{q_F(z | x^{(i)})} [\log p_F(x^{(i)}) | z] q_F(z | x^{(i)}) \right] \quad \text{(Multiply by constant)} \]
  \[ = \mathbb{E}_z [\log p_F(x^{(i)} | z)] - \mathbb{E}_z [\log q_F(z | x^{(i)}) | p_F(x^{(i)})] + \mathbb{E}_z \left[ \log q_F(z | x^{(i)}) \right] \]

  The expectation w.r.t $z$ (using encoder network) lets us write nice KL terms.
Variational Autoencoders

- We can now work out the log-likelihood

\[
\log p(x^{(i)}) = E_{z \sim q(x^{(i)} | x)} \log p(x^{(i)})
\]

\( (p(x^{(i)}) \) does not depend on \( i \) \)

\[
= E_x \log p(x^{(i)} | z)p(z) \frac{q_\phi(z | x^{(i)})}{p_\theta(z)}
\]

(Bayes’ Rule)

\[
= E_x \log p(x^{(i)} | z)p(z) \frac{q_\phi(z | x^{(i)})}{p_\theta(z)}
\]

(Multiply by constant)

\[
= E_x \log p(x^{(i)} | z) - E_x \log q_\phi(z | x^{(i)})p_\theta(z) + E_x \log q_\phi(z | x^{(i)})p_\theta(z)
\]

Decoder network gives \( p_x(x | z) \), can compute estimate of this term through sampling.
(Sampling differentiable through reparametrization trick, see paper)

\begin{align*}
V_i &= \frac{1}{n} \sum_{i=1}^{n} \log p(x^{(i)})
\end{align*}

Want to maximize data likelihood

\[
\mathcal{L}(x^{(i)}, \theta, \phi) \geq 0
\]

Tractable lower bound, which we can take gradient of and optimize

Variational Lower Bound (“ELBO”)

\[
\log \pi_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)}, \theta, \phi)
\]

- Reconstruct the input data
- Make approximate posterior distribution close to prior

Training: Maximize lower bound

\[
\theta^*, \phi^* = \arg \max_{\theta, \phi} \sum_{i=1}^{n} \mathcal{L}(x^{(i)}, \theta, \phi)
\]

Topics of This Lecture

- Recap: GANs
  - Autoencoders
    - Motivation
    - Regularized Autoencoder
    - Denoising Autoencoder
  - Variational Autoencoders (VAE)
    - Autoencoders as Generative Models
    - Intractability
    - Variational Approximation
    - Evidence Lower Bound (ELBO)
- Application Examples
Application Examples

32x32 CIFAR-10

Labeled Faces in the Wild

References

• Variational Auto-Encoders