Lecture 14: Convolutional Neural Networks and Intro to PCA

Nisha Chandramoorthy

October 26, 2023

Motivation for dimensionality reduction

- Map high-dimensional data to a lower-dimensional space

Motivation for dimensionality reduction

- Map high-dimensional data to a lower-dimensional space
- Why? Computational efficiency in downstream tasks

Motivation for dimensionality reduction

- Map high-dimensional data to a lower-dimensional space
- Why? Computational efficiency in downstream tasks
- Visualization, interpretation

Motivation for dimensionality reduction

- Map high-dimensional data to a lower-dimensional space
- Why? Computational efficiency in downstream tasks
- Visualization, interpretation
- Better generalization (avoid overfitting)

Autoencoder decoder

$$(E^*, D^*) = \arg\min_{E,D} \sum_{i=1}^{m} \|x_i - D(E(x_i))\|^2 \tag{1}$$

- Posed as ERM problem.

Autoencoder decoder

$$(E^*, D^*) = \arg\min_{E,D} \sum_{i=1}^{m} \|x_i - D(E(x_i))\|^2 \tag{1}$$

- Posed as ERM problem.
- E is encoder, D is decoder.

Autoencoder decoder

$$(E^*, D^*) = \arg\min_{E,D} \sum_{i=1}^{m} \|x_i - D(E(x_i))\|^2 \tag{1}$$

- Posed as ERM problem.
- E is encoder, D is decoder.
- E maps x to z (latent space), D maps z to $\hat{x}$ (reconstruction).

Autoencoder decoder

$$(E^*, D^*) = \arg\min_{E,D} \sum_{i=1}^{m} \|x_i - D(E(x_i))\|^2 \tag{1}$$

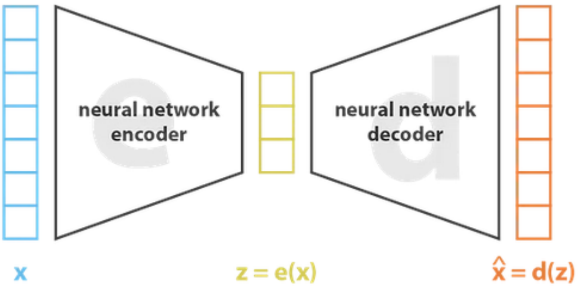
- Posed as ERM problem.
- E is encoder, D is decoder.
- E maps x to z (latent space), D maps z to $\hat{x}$ (reconstruction).
- Both parameterized as Neural Networks.

Variational autoencoders

- Probabilistic encoder and decoder.

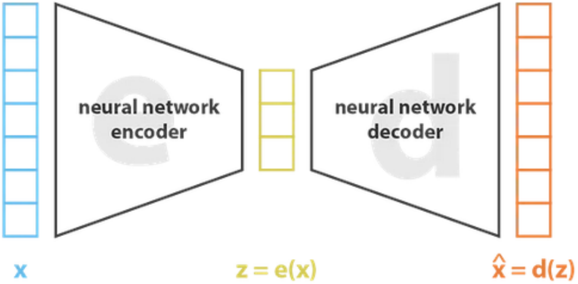
Variational autoencoders

- Probabilistic encoder and decoder.
- Encoder: $q(z|x)$, Decoder: $p(x|z)$



$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

- tends to overfit as a Generative model



$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

- tends to overfit as a Generative model
- VAE: uses VI to regularize the latent space.

Variational Inference

- Minimize KL divergence between $q(z|x)$ and $p(z|x)$.
- Recall KL divergence: $D_{\mathrm{KL}}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx$

Variational Inference

- Minimize KL divergence between $q(z|x)$ and $p(z|x)$.
- Recall KL divergence: $D_{\mathrm{KL}}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx$
- $D_{\mathrm{KL}}(p||q) \geqslant 0$ and $D_{\mathrm{KL}}(p||q) = 0$ iff $p = q$.

Variational Inference

- Minimize KL divergence between $q(z|x)$ and $p(z|x)$.
- Recall KL divergence: $D_{\mathrm{KL}}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx$
- $D_{\mathrm{KL}}(p||q) \geqslant 0$ and $D_{\mathrm{KL}}(p||q) = 0$ iff $p = q$.
-
$$\begin{aligned} D_{\mathrm{KL}}(q_\theta(z|x)||p(z|x)) &= \int q_\theta(z|x) \log \frac{q_\theta(z|x)}{p(z|x)} dz \\ &= E_{z\sim q_\theta(z|x)}[\log q_\theta(z|x)] \\ &- E_{z\sim q_\theta(z|x)}[\log p(z|x)] \\ &= E_{z\sim q_\theta(z|x)}[\log p(x|z)] - D_{\mathrm{KL}}(q_\theta(z|x)||p(z)) \end{aligned}$$

VAEs

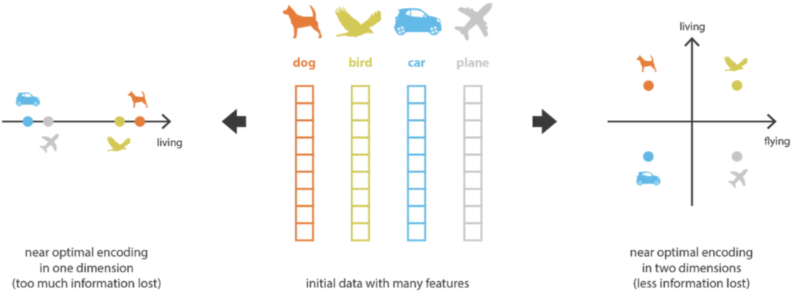
- Conditional Gaussian assumption: $\log p(x|z) = \|x - f(z)\|^2 / c$.

VAEs

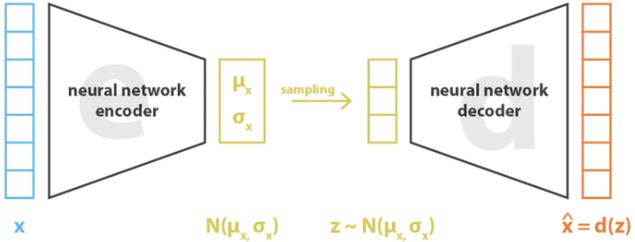
- Conditional Gaussian assumption: $\log p(x|z) = \|x - f(z)\|^2/c$.
- Typically q_θ is parameterized with its mean and variance as Neural Networks.

VAEs

- Conditional Gaussian assumption:
  $\log p(x|z) = \|x - f(z)\|^2/c$.
- Typically q_θ is parameterized with its mean and variance as Neural Networks.
- $$\mathrm{argmax}_{\theta, f} \sum_{i=1}^{m} \log p(x_i|z_i) - D_{\mathrm{KL}}(q_\theta(z_i|x_i)\|p(z_i)) \tag{2}$$



Courtesy: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73



$$\text{loss} = \| x - \hat{x} \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,] = \| x - d(z) \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,]$$

Courtesy: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

PCA

- when E and D are linear, this is equivalent to PCA.

PCA

- when E and D are linear, this is equivalent to PCA.
- $E(x) = Wx$, $D(z) = W^\top z$.

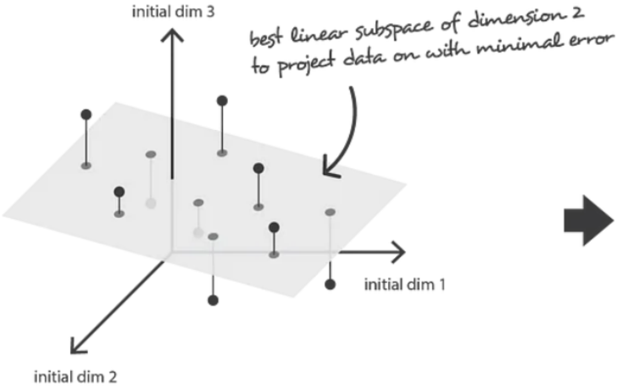
PCA

- when E and D are linear, this is equivalent to PCA.
- $E(x) = Wx$, $D(z) = W^\top z$.
- Let $C = \sum_{i=1}^{m} x_i x_i^\top = X^\top X$ be the data correlation matrix.

PCA

- when E and D are linear, this is equivalent to PCA.
- $E(x) = Wx$, $D(z) = W^\top z$.
- Let $C = \sum_{i=1}^{m} x_i x_i^\top = X^\top X$ be the data correlation matrix.
- C is symmetric and positive semi-definite, $C = V \Lambda V^\top$.
- Theorem PCA: among linear hypothesis classes, $E^* = V^\top$, $D^* = V$, where V is the matrix of eigenvectors of $C = X^\top X$.

Best linear subspace



Courtesy: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

PCA by SVD

- When data are centered, take SVD, $X = U\Sigma V^T$.

PCA by SVD

- When data are centered, take SVD, $X = U\Sigma V^T$.
- Correlation matrix $C = X^T X = V\Sigma^2 V^T$.

PCA by SVD

- When data are centered, take SVD, $X = U\Sigma V^T$.
- Correlation matrix $C = X^T X = V\Sigma^2 V^T$.
- principal components: $XV = U\Sigma$.

PCA by SVD

- When data are centered, take SVD, $X = U\Sigma V^T$.
- Correlation matrix $C = X^T X = V\Sigma^2 V^T$.
- principal components: $XV = U\Sigma$. Numerical stability

Convolutional Neural Networks (source: cs231n.stanford.edu)

- Suitable for image recognition. Won the 2012 ImageNet competition and subsequent ones.
- Three types of layers: convolutional, FC, pooling
- Convolutional layer: accepts a volume of size $W_1 \times H_1 \times D_1$ and outputs a volume of size $W_2 \times H_2 \times D_2$ where $W_2 = (W_1 - F + 2P)/S + 1$ and $H_2 = (H_1 - F + 2P)/S + 1$ and $D_2 = K$.
- K is number of filters, F is filter size, S is stride, P is padding.
- Pooling layer: downsamples along width and height, and optionally along depth.
- FC layer: computes class scores, resulting in volume of size $1 \times 1 \times K$.