Variational Autoencoder

Jia-Wei Liao

Department of Applied Mathematics
National Yang Ming Chiao Tung University

March 27, 2021

Outline

Neural Network (NN)

Question: Why a neural net can approximate functions?

[1] Lu et al., The Expressive power of Neural Networks: A View from the Width, NIPS 2017

Neural Network (NN)

Question: Why a neural net can approximate functions? [1]

Theorem (Universal Approximation Theorem With ReLU Network)

For any Lebesgue-integrable function $f : \mathbb{R}^n \to \mathbb{R}$ and any $\varepsilon > 0$, there exist a fully-connected ReLU network $\mathcal{Q}$ with width $\leq n + 4$ and depth $\leq 4n + 1$ such that the function $F_{\mathcal{Q}}$ represented by this network satisfies

$$\int_{\mathbb{R}^n} |f(x) - F_{\mathcal{Q}}| dx < \varepsilon$$

[1] Lu et al., The Expressive power of Neural Networks: A View from the Width, NIPS 2017

Convolutional Neural Network (CNN)

Manifold structure

Manifold

Let Σ be a topological space, covered by a set of open sets $\Sigma \subset \bigcup_\alpha U_\alpha$. For each open set U_α, there is a homeomorphism $\varphi_\alpha : U_\alpha \to \mathbb{R}^n$, the pair $(U_\alpha, \varphi_\alpha)$ form a chart. The union of charts form an atlas $\mathcal{A} = \{(U_\alpha, \varphi_\alpha)\}$. If $U_\alpha \cap U_\beta \neq \varnothing$, then the chart transition map is given by

$$\varphi_{\alpha\beta} : \varphi_\alpha \left(U_\alpha \cap U_\beta\right) \to \varphi_\beta \left(U_\alpha \cap U_\beta\right),$$

where $\varphi_{\alpha\beta} = \varphi_\beta \circ \varphi_\alpha^{-1}$.

Manifold structure

Manifold

Let Σ be a topological space, covered by a set of open sets $\Sigma \subset \bigcup_\alpha U_\alpha$. For each open set U_α, there is a homeomorphism $\varphi_\alpha : U_\alpha \to \mathbb{R}^n$, the pair $(U_\alpha, \varphi_\alpha)$ form a chart. The union of charts form an atlas $\mathcal{A} = \{(U_\alpha, \varphi_\alpha)\}$. If $U_\alpha \cap U_\beta \neq \varnothing$, then the chart transition map is given by

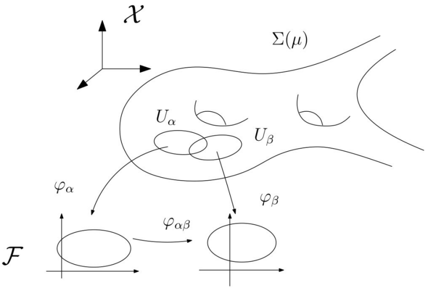
$$\varphi_{\alpha\beta} : \varphi_\alpha \left(U_\alpha \cap U_\beta\right) \to \varphi_\beta \left(U_\alpha \cap U_\beta\right),$$

where $\varphi_{\alpha\beta} = \varphi_\beta \circ \varphi_\alpha^{-1}$.

Manifold assumption

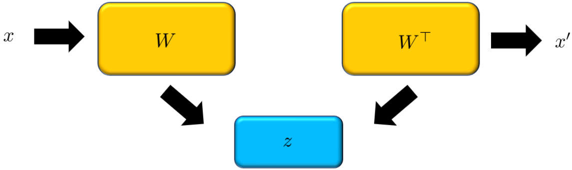
Natural high dimensional data concentrates close to a non-linear low-dimensional manifold.

Manifold structure

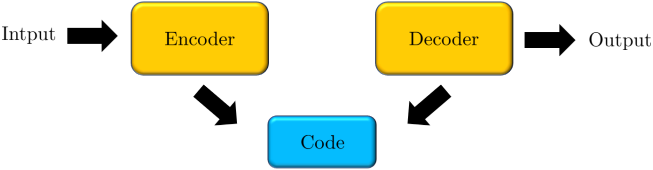


- $\mathcal{X}$ is the ambient space, $\mathcal{F}$ is latent space.
- Σ is a low-dimensional manifold.
- φ_α is encoding map, and φ_α^{-1} is decoding map.
- $x \in \Sigma$ is a sample, and $\varphi_\alpha(x)$ is the code of x.

PCA vs Autoencoder

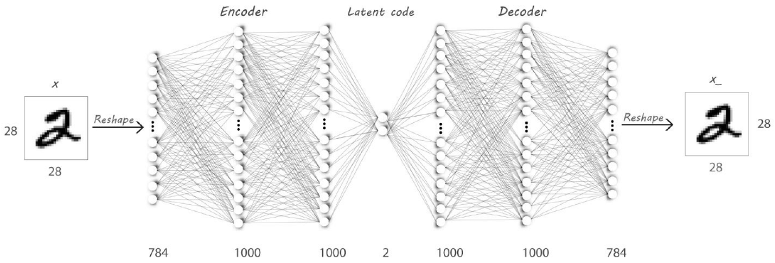


$$z = Wx$$

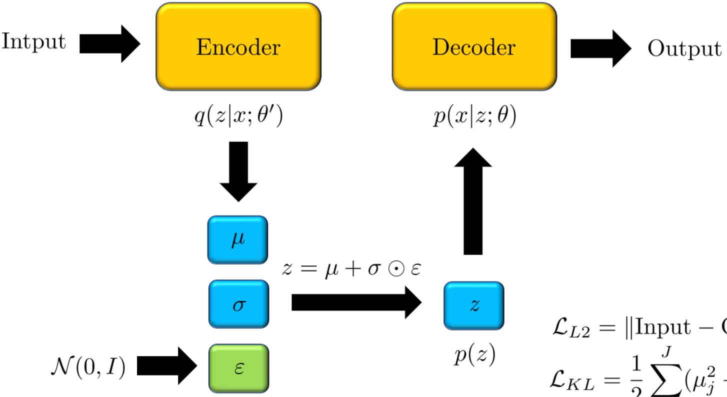


$$\mathcal{L} = \|\text{Input} - \text{Output}\|_2$$

Autoencoder



Variational Autoencoder (VAE)



$$\mathcal{L}_{L2} = \|\text{Input} - \text{Output}\|_2$$

$$\mathcal{L}_{KL} = \frac{1}{2}\sum_{j=1}^{J}(\mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1)$$

Variational Autoencoder (VAE)

- A probabilistic generative model with latent variables that is built on top of end-to-end trainable neural networks.
- By approximation theorey, assume that

$$p(z) = \mathcal{N}(z; 0, I)$$
$$p(\boldsymbol{x}|z) = \mathcal{N}(z; \mu(z), \Sigma(z))$$

Approximation theory

Our goal is to find the probability distribution function that has maximal differential entropy. The problem is

$$\arg\max_{p(x)} H[p(x)]$$

subject to

$$\begin{cases} \int_{x\in X} p(x)dx = 1 \\ E(X) = \mu \\ E[(X-\mu)^2] = \sigma^2 \end{cases},$$

where $H[p(x)] = -\int_{x\in X} p(x)\log p(x)dx$. [2]

[2] $H[p(x)]$ is expectation of the entropy.

Approximation theory

This constrained optimization problem can be solved by setting up a Lagrangian functional

$$\begin{aligned}\mathcal{L}(p, \lambda_1, \lambda_2, \lambda_3) = &- \int_{x \in X} p(x) \log p(x) dx \\ &+ \lambda_1 \left(\int_{x \in X} p(x) dx - 1 \right) \\ &+ \lambda_2 \left(\int_{x \in X} x p(x) dx - \mu \right) \\ &+ \lambda_3 \left(\int_{x \in X} (x - \mu)^2 p(x) dx - \sigma^2 \right)\end{aligned}$$

We set the functional derivative w.r.t. $p(x)$ to 0

$$\frac{\delta}{\delta p(x)} \mathcal{L} = - \log p(x) - 1 + \lambda_1 + \lambda_2 + \lambda_3 (x - \mu)^2 = 0.$$

Approximation theory

Then we have

$$p(x) = \exp\left(\lambda_1 + \lambda_2 x + \lambda_3 (x-\mu)^2 - 1\right),$$

and take

$$\lambda_1 = 1 - \log \sigma\sqrt{2\pi}, \quad \lambda_2 = 0, \quad \lambda_3 = -\frac{1}{2\sigma^2}.$$

Therefore, we can get

$$p(x) = \mathcal{N}\left(x; \mu, \sigma^2\right).$$

That is, the normal distribution has the maximum entropy. So, when we do not know the true distribution, we can assume the normal distribution.

Maximal Likelihood

- To determine $\boldsymbol{\theta}$, we would intuitively hope to maximize the marginal distribution $p(\boldsymbol{x};\boldsymbol{\theta})$

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \int_{\boldsymbol{z}\in\boldsymbol{Z}} p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})p(\boldsymbol{z})d\boldsymbol{z}$$

- The marginal likelihood is composed of a sum over the marginal likelihoods of individual datapoints

$$\log p(\boldsymbol{x_1},\cdots \boldsymbol{x_N};\boldsymbol{\theta}) = \log \prod_{i=1}^{N} p(\boldsymbol{x_i};\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\boldsymbol{x_i};\boldsymbol{\theta})$$

Maximal Likelihood

Since $\int_{\boldsymbol{z}\in\boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')d\boldsymbol{z} = 1$,

$$\begin{aligned}\log p(\boldsymbol{x};\boldsymbol{\theta}) &= \int_{\boldsymbol{z}\in\boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')\log p(\boldsymbol{x};\boldsymbol{\theta})d\boldsymbol{z} \\ &= \int_{\boldsymbol{z}\in\boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')\log \frac{p(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\theta})}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})}d\boldsymbol{z} \\ &= \int_{\boldsymbol{z}\in\boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')\log \frac{p(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\theta})}{q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')}\cdot\frac{q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})}d\boldsymbol{z} \\ &= \int_{\boldsymbol{z}\in\boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')\log \frac{p(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\theta})}{q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')}d\boldsymbol{z} \\ &\quad+ \int_{\boldsymbol{z}\in\boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')\log \frac{q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})}d\boldsymbol{z}\end{aligned}$$

Maximal Likelihood

$$\log p(\boldsymbol{x};\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{x}, q, \boldsymbol{\theta}) + \mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}') \| p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}))$$

where

$$\mathcal{L}(\boldsymbol{x}, q, \boldsymbol{\theta}) = \int_{\boldsymbol{z}\in\boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}') \log \frac{p(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\theta})}{q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})'} d\boldsymbol{z}$$

$$\mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}') \| p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})) = \int_{\boldsymbol{z}\in\boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}') \log \frac{q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})} d\boldsymbol{z}$$

Note that

$\mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}') \| p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})) \approx 0$ if and only if $p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}) \approx q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')$

Variational lower bound

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{x}, q, \boldsymbol{\theta}) &= \int_{\boldsymbol{z} \in \boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}') \log \frac{p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta})}{q(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}')} dz \\
&= \int_{\boldsymbol{z} \in \boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}') \log \frac{p(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\theta}) p(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}')} dz \\
&= \int_{\boldsymbol{z} \in \boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}') \log \frac{p(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}')} dz \\
&+ \int_{\boldsymbol{z} \in \boldsymbol{Z}} q(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}') \log p(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\theta}) dz \\
&= -\mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}') \| p(\boldsymbol{z})) + E_{z \sim q(z|x; \theta')} [\log p(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\theta}')]
\end{aligned}
$$

KL term

By previous assumption,

$$p(\boldsymbol{z}) = \log \mathcal{N}(\boldsymbol{z}; 0, I) \text{ and } q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}') = \mathcal{N}(\boldsymbol{z}; \mu, \sigma^2)$$

We can get

$$\begin{aligned}
-\mathrm{KL}(q\|p) &= \int_{\boldsymbol{z}\in Z} q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}') \log \frac{p(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')} d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}\in Z} \mathcal{N}(\boldsymbol{z}; \mu, \sigma^2) \left(\log \mathcal{N}(\boldsymbol{z}; 0, I) - \log \mathcal{N}(\boldsymbol{z}; \mu, \sigma^2)\right) d\boldsymbol{z}
\end{aligned}$$

Note that

- $\int_{\boldsymbol{z}\in Z} \mathcal{N}(\boldsymbol{z}; \mu, \sigma^2) \log \mathcal{N}(\boldsymbol{z}; 0, I) d\boldsymbol{z} = -\frac{J}{2}\log 2\pi - \frac{1}{2}\sum_{j=1}^{J}(\mu_j^2 + \sigma_j^2)$
- $\int_{\boldsymbol{z}\in Z} \mathcal{N}(\boldsymbol{z}; \mu, \sigma^2) \log \mathcal{N}(\boldsymbol{z}; \mu, \sigma^2) d\boldsymbol{z} = -\frac{J}{2}\log 2\pi - \frac{1}{2}\sum_{j=1}^{J}(1 + \log \sigma_j^2)$

KL term

Therefore,

$$-\mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')\|p(\boldsymbol{z})) = \frac{1}{2}\sum_{j=1}^{J}(\mu_j^2 + \sigma_j^2 - \log\sigma_j^2 - 1)$$

Expectation term

By Monte Carlo estimate,

$$E_{z\sim q(z|x;\theta')}[\log p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta}')] \approx \frac{1}{K}\sum_{k=1}^{K}\log p(x|z^{(k)};\theta)$$

where $z \sim q(z|x;\theta')$.

Reparameterization trick

$$z \sim q(z|x;\theta') \Longrightarrow \begin{cases} \text{auxiliary variable: } \varepsilon \sim p(\varepsilon) \\ \text{deterministic variable: } \boldsymbol{z} = g(\boldsymbol{x}, \varepsilon; \theta') \end{cases}$$

For example, we can take

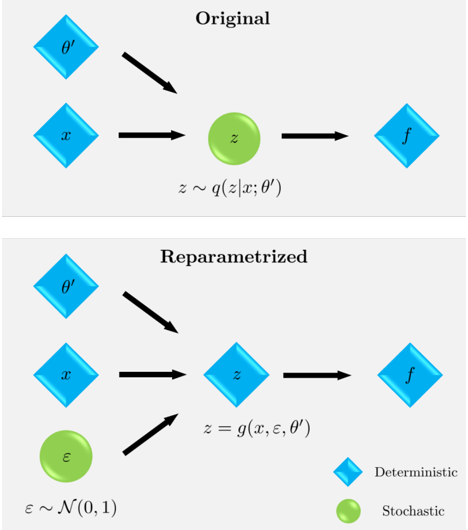
$$p(\varepsilon) = \mathcal{N}(0, I) \text{ and } g(\boldsymbol{x}, \varepsilon; \mu, \sigma) = \mu + \sigma \odot \varepsilon,$$

then

$$E_{z \sim q(z|x;\theta')}[\log p(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\theta}')] \approx \frac{1}{K} \sum_{k=1}^{K} \log p(\boldsymbol{x}|z^{(k)}; \boldsymbol{\theta}')$$

where $z^{(k)} = \mu^{(k)} + \sigma^{(k)} \odot \varepsilon^{(k)}$ and $\varepsilon^{(k)} \sim \mathcal{N}(0, I)$.

Reparameterization trick



Training VAE

- $p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})$ and $q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')$ are modeled by distinct neural networks.
- A by-product of this training process is a stochastic encoder

$$p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta}) \approx q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta}')$$

THE END

Thanks for listening!