STOR566: Introduction to Deep Learning

Lecture 12: Generative Models I

Yao Li
UNC Chapel Hill

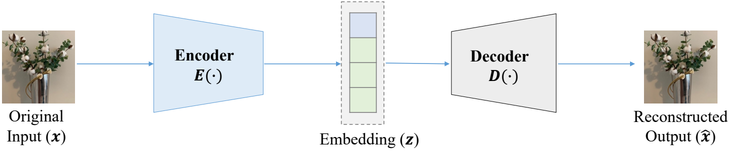
Oct 10, 2024

Unsupervised Learning

- Working with datasets without a **response** variable
- Some Applications:
  - Clustering
  - Data Compression
  - Exploratory Data Analysis
  - Generating New Examples
  - ...
- Example: PCA, K-means, Autoencoders, GAN, etc

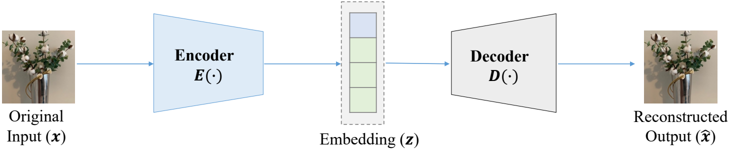
Autoencoder: Basic Architecture

- Autoencoder: A special type of DNN where the target (response) of each input is the input itself.



Autoencoder: Basic Architecture

- Autoencoder: A special type of DNN where the target (response) of each input is the input itself.



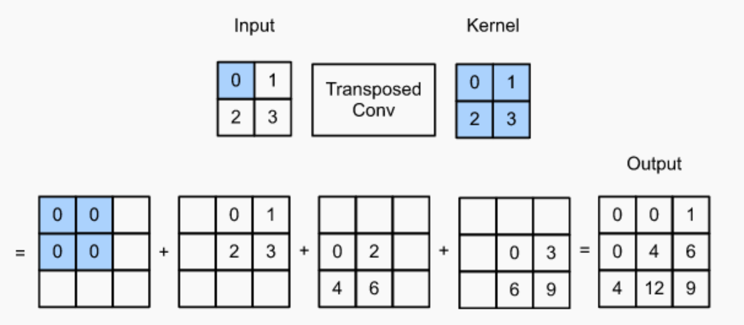
- Objective:

$$\|\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{E}(\boldsymbol{x}))\|^2$$

Encoder: $\boldsymbol{E} : \mathbb{R}^n \to \mathbb{R}^d$

Decoder: $\boldsymbol{D} : \mathbb{R}^d \to \mathbb{R}^n$

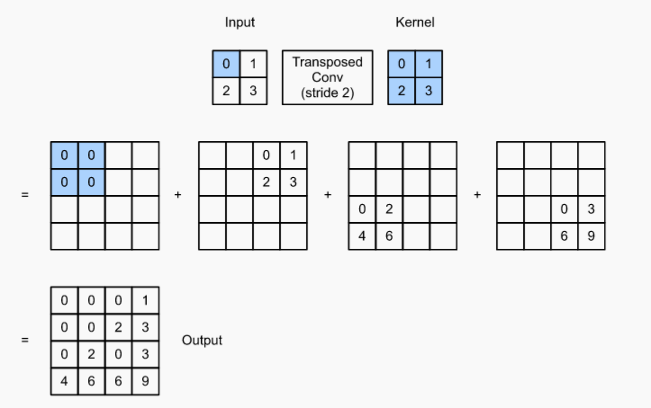
Transposed Convolution



(Figure from Dive into Deep Learning)

- Multiple input and output channels: works the same as the regular convolution
- Number of weights: $k_1 \times k_2 \times d_{in} \times d_{out} + d_{out}$

Transposed Convolution



(Figure from Dive into Deep Learning)

- Strides are specified for the output feature map
- Padding: remove rows and columns from the output

Overfitting

- Overfitting is a problem
- Solutions:
  - Bottleneck layer: a low-dimensional representation of the data ($d < n$)
  - Denoise autoencoder
  - Sparse autoencoder
  - ...

Regularization

- Objective:

$$L(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \text{regularizer},$$

Regularization

- Objective:

$$L(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \text{regularizer},$$

$L(\cdot, \cdot)$: captures the distance between the input ($\boldsymbol{x}$) and the output ($\hat{\boldsymbol{x}}$).

  - Example: $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$

Regularization

- Objective:

$$L(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \text{regularizer},$$

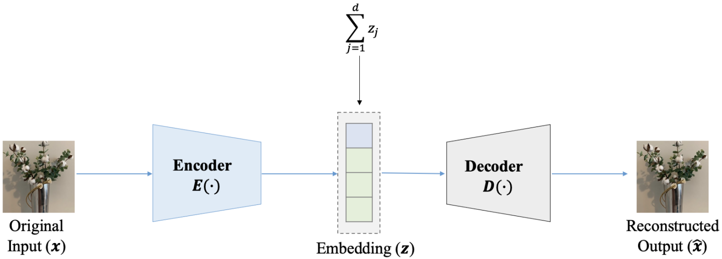
$L(\cdot, \cdot)$: captures the distance between the input ($\boldsymbol{x}$) and the output ($\hat{\boldsymbol{x}}$).

  - Example: $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$

Regularizer example:

  - L_1 penalty: $\sum_j |h_j^l|$
  - h_j^l: hidden output of j-th neuron in l-th layer

Sparse Autoencoder

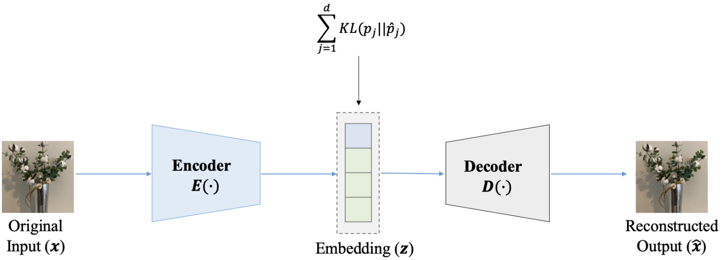


- Objective:

$$\|\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{E}(\boldsymbol{x}))\|^2 + \lambda \sum_j |z_j|$$

- Iterate over layers.

Sparse Autoencoder



- Another regularizer:

$$\|\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{E}(\boldsymbol{x}))\|^2 + \lambda \sum_j KL(p_j||\hat{p}_j)$$

- Convert value of z to $[0, 1]$. (e.g., sigmoid activation)
- p_j: probability of activation for neuron j in the bottleneck layer
- $\hat{p}_j = \frac{1}{B} \sum_{i=1}^{B} z_{ij}$

Denoising Autoencoder

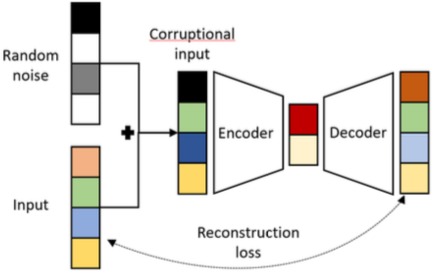


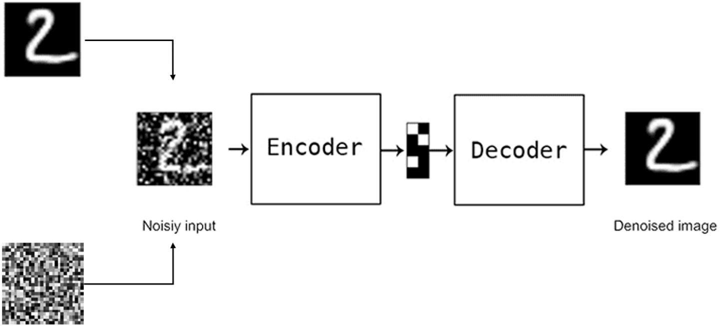
Figure from Bank, Dor, Noam Koenigstein, and Raja Giryes. "Autoencoders." (2020).

- Another regularizer:

$$\|\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{E}(\boldsymbol{x} + \delta))\|^2$$

- δ: Random noise

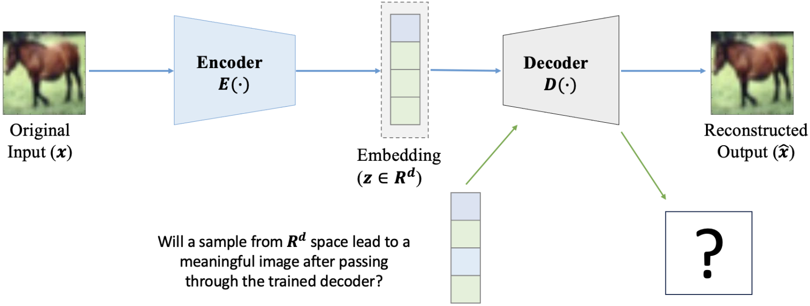
Denoising Autoencoder



- noisy data $\rightarrow$ clean data
- Learn to capture valuable features and ignore noise

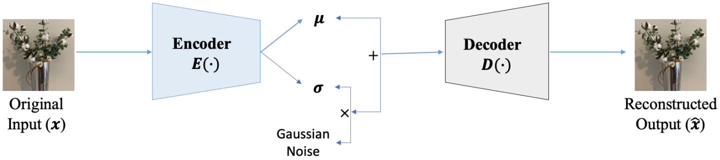
Generative Model

Generative Problem



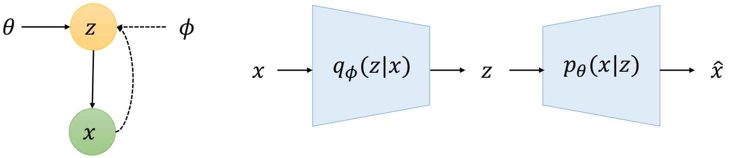
- In general, a trained Vanilla auto-encoder cannot be used to generate new data

Variational Autoencoder (VAE)



- Probabilistic model: will let us generate data from the model
- Encoder outputs $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$
- Draw $\tilde{\boldsymbol{z}} \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma})$
- Decoder decodes this **latent** variable $\tilde{\boldsymbol{z}}$ to get the output

Variational Autoencoder (VAE)



- Maximum likelihood approach: $\Pi_i p(\boldsymbol{x}_i)$
- Variational lower bound as objective:
  - End-to-End reconstruction loss (e.g., square loss)
  - Regularizer: $KL\left(q_\Phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right)$
- Objective:

$$L(\boldsymbol{x}, \hat{\boldsymbol{x}}) + KL\left(q_\Phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right)$$

Variational Lower Bound

- Variational lower bound:

$$\log p(x) \geq E_{q(\boldsymbol{z}|\boldsymbol{x})}\left(\log p(x|z)\right) - KL\left(q(z|x)||p(z)\right)$$

- How to derive the variational lower bound from the likelihood?
- Suggested reading: Kingma et al. (2013). Auto-encoding variational bayes. ICLR.

Re-parameterization Trick

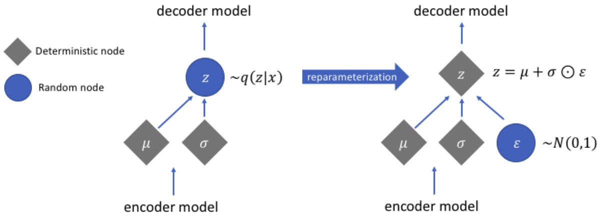
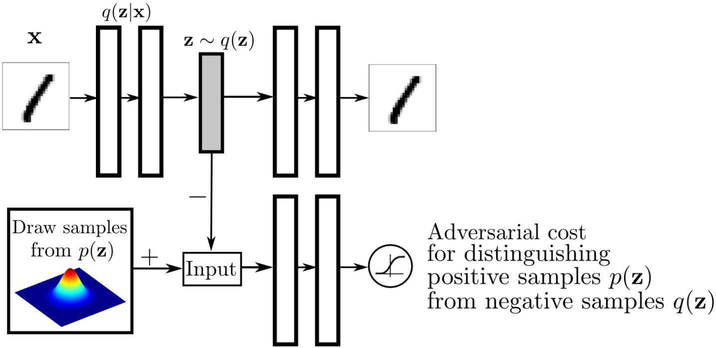


Figure from Jeremy Jordon Blog

- Cannot back-propagate error through random samples
- Reparameterization trick: replace $\tilde{\boldsymbol{z}} \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma})$ with $\boldsymbol{\epsilon} \sim N(0, I)$, $\boldsymbol{z} = \boldsymbol{\epsilon}\boldsymbol{\sigma} + \boldsymbol{\mu}$

Adversarial Autoencoder



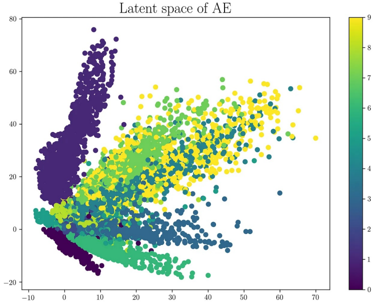
- The top row is a standard autoendoer
- Force the embedding space distribution towards the prior

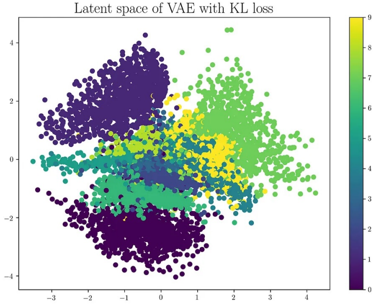
Embedding Space Visualization

Commonly used visualization tools:

- t-SNE (t-Distributed Stochastic Neighbor Embedding)
  - Van der Maaten et al. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(11).
  - Available: sklearn
- UMAP (Uniform Manifold Approximation and Projection)
  - McInnes et al. (2018). UMAP: Uniform Manifold Approximation and Projection. Journal of Open Source Software, 3(29), 861,
  - Availalbe: umap-learn
- PCA (Principal Component Analysis)
  - Available: sklearn

Examples with tSNE





- Embedding space visualization for a Vanilla autoencoer and a VAE trained on MNIST
- VAE: more compact

Examples with PCA

- Problem: Game Result Prediction

Figure: Heroes of the Storm and Dota 2 characters

Assumption

Assumption

We assume a team's score can be written as

$$s_t^+ = \sum_{i \in I_t^+} w_i + \sum_{i \in I_t^+} \sum_{j \in I_t^+} \boldsymbol{v}_i^T \boldsymbol{v}_j$$

- w_i: individual ability of i-th player
- $\boldsymbol{v}_i \in R^d$: teamwork ability of i-th player
- I_t^+: winning team player index set
- s_t^+: winning team score

Team Ability Visualization (PCA)

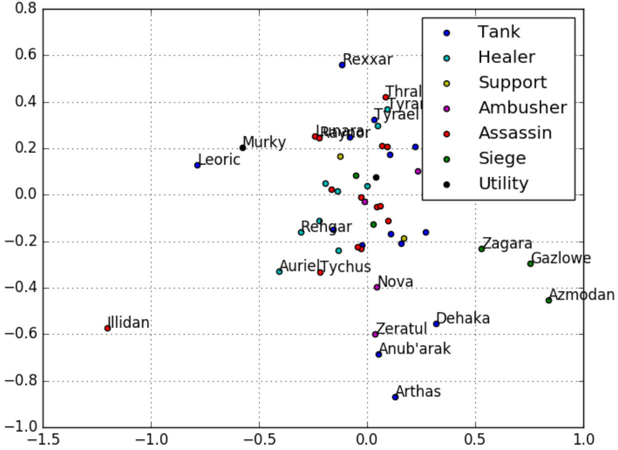


Figure: Projection of team ability vector for each character ($\boldsymbol{v}_i$) to 2-D space. Colors represents the official categorization for these characters.

Conclusions

- Autoencoder
- Regularization
- Variational Autoencoder
- Visualization tools

Questions?