Outline

Definition

Curse of dimensionality: Large feature dimensionalities $\mathbb{R}^d \not\Leftrightarrow$ benefitial

Definition

Curse of dimensionality: Large feature dimensionalities $\mathbb{R}^d \not\Leftrightarrow$ benefitial

- Data sparcity
- Correlations in the feature space

Definition

Curse of dimensionality: Large feature dimensionalities $\mathbb{R}^d \nLeftrightarrow$ benefitial

- Data sparcity
- Correlations in the feature space

Dimensionality reduction: Decrease the number d of features

Definition

Curse of dimensionality: Large feature dimensionalities $\mathbb{R}^d \not\Leftrightarrow$ benefitial

- Data sparcity
- Correlations in the feature space

Dimensionality reduction: Decrease the number d of features

- $\mathcal{R} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ with $d' \leq d$
- Does not reduce the number of elements in the $\mathcal{D}$
- Unsupervised strategies

Definition

Curse of dimensionality: Large feature dimensionalities $\mathbb{R}^d \not\Leftrightarrow$ benefitial

- Data sparcity
- Correlations in the feature space

Dimensionality reduction: Decrease the number d of features

- $\mathcal{R} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ with $d' \leq d$
- Does not reduce the number of elements in the $\mathcal{D}$
- Unsupervised strategies

Common paradigms:

- Statistical frameworks: PCA, t-SNE
- Neural-based approaches: Autoencoder

Principal Components Analysis

- Linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:

Principal Components Analysis

- Linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:
  - $\rightarrow$ Datum $\mathbf{x} \in \mathbb{R}^d$ projected onto $\mathbb{R}^{d'}$ as: $(\mathbf{x}^T\mathbf{v}_1, \ldots, \mathbf{x}^T\mathbf{v}_{d'})$ with $\mathbf{v} \in \mathbb{R}^d$

Principal Components Analysis

- Linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:
  - $\rightarrow$ Datum $\mathbf{x} \in \mathbb{R}^d$ projected onto $\mathbb{R}^{d'}$ as: $(\mathbf{x}^T\mathbf{v}_1, \ldots, \mathbf{x}^T\mathbf{v}_{d'})$ with $\mathbf{v} \in \mathbb{R}^d$
    - $\Rightarrow$ $\mathbf{x}_{\text{proj}} = \mathbf{x}\mathcal{V}$ with $\mathcal{V} \in \mathbb{R}^{d \times d'}$

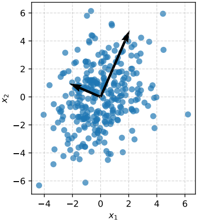
Principal Components Analysis

- Linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:
  - $\rightarrow$ Datum $\mathbf{x} \in \mathbb{R}^d$ projected onto $\mathbb{R}^{d'}$ as: $(\mathbf{x}^T\mathbf{v}_1, \ldots, \mathbf{x}^T\mathbf{v}_{d'})$ with $\mathbf{v} \in \mathbb{R}^d$
    - $\Rightarrow$ $\mathbf{x}_{\text{proj}} = \mathbf{x}\mathcal{V}$ with $\mathcal{V} \in \mathbb{R}^{d \times d'}$
  - $\rightarrow$ If $d' < d \Rightarrow$ Dimensionality reduction

Principal Components Analysis

- Linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:
  - $\rightarrow$ Datum $\mathbf{x} \in \mathbb{R}^d$ projected onto $\mathbb{R}^{d'}$ as: $(\mathbf{x}^T\mathbf{v}_1, \ldots, \mathbf{x}^T\mathbf{v}_{d'})$ with $\mathbf{v} \in \mathbb{R}^d$
    - $\Rightarrow$ $\mathbf{x}_{\text{proj}} = \mathbf{x}\mathcal{V}$ with $\mathcal{V} \in \mathbb{R}^{d \times d'}$
  - $\rightarrow$ If $d' < d \Rightarrow$ Dimensionality reduction

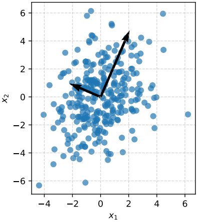
- **Goal**: Find *directions* that capture the most variance in the data

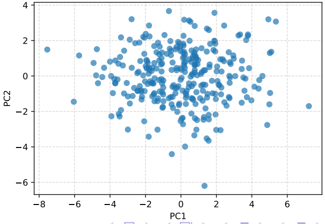


Principal Components Analysis

- Linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:
  - $\rightarrow$ Datum $\mathbf{x} \in \mathbb{R}^d$ projected onto $\mathbb{R}^{d'}$ as: $(\mathbf{x}^T\mathbf{v}_1, \ldots, \mathbf{x}^T\mathbf{v}_{d'})$ with $\mathbf{v} \in \mathbb{R}^d$
    - $\Rightarrow$ $\mathbf{x}_{\text{proj}} = \mathbf{x}\mathcal{V}$ with $\mathcal{V} \in \mathbb{R}^{d \times d'}$
  - $\rightarrow$ If $d' < d \Rightarrow$ Dimensionality reduction

- **Goal**: Find *directions* that capture the most variance in the data
  - $\rightarrow$ These *directions* constitute the coordinates of the new space

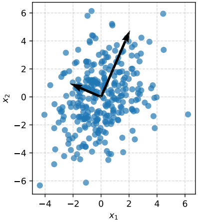


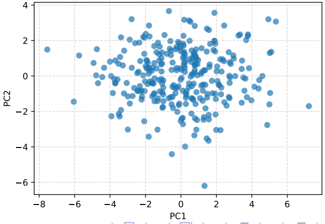


Principal Components Analysis

- Linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:
  - $\rightarrow$ Datum $\mathbf{x} \in \mathbb{R}^d$ projected onto $\mathbb{R}^{d'}$ as: $(\mathbf{x}^T\mathbf{v}_1, \ldots, \mathbf{x}^T\mathbf{v}_{d'})$ with $\mathbf{v} \in \mathbb{R}^d$
    - $\Rightarrow$ $\mathbf{x}_{\text{proj}} = \mathbf{x}\mathcal{V}$ with $\mathcal{V} \in \mathbb{R}^{d \times d'}$
  - $\rightarrow$ If $d' < d \Rightarrow$ Dimensionality reduction

- **Goal**: Find *directions* that capture the most variance in the data
  - $\rightarrow$ These *directions* constitute the coordinates of the new space
  - $\rightarrow$ Correspond to the *eigenvectors* of the data

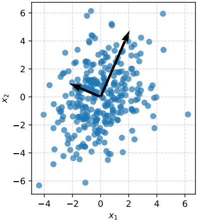


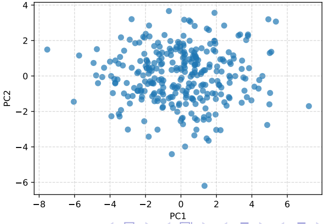


Principal Components Analysis

- Linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:
  - → Datum $\mathbf{x} \in \mathbb{R}^d$ projected onto $\mathbb{R}^{d'}$ as: $(\mathbf{x}^T\mathbf{v}_1, \ldots, \mathbf{x}^T\mathbf{v}_{d'})$ with $\mathbf{v} \in \mathbb{R}^d$
    - $\Rightarrow$ $\mathbf{x}_{\text{proj}} = \mathbf{x}\mathcal{V}$ with $\mathcal{V} \in \mathbb{R}^{d \times d'}$
  - → If $d' < d \Rightarrow$ Dimensionality reduction

- **Goal**: Find *directions* that capture the most variance in the data
  - → These *directions* constitute the coordinates of the new space
  - → Correspond to the *eigenvectors* of the data
  - → Known as Principal Components





Mathematical formulation

- Initial assortment $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{D}|}$ with $\mathbf{x}_i \in \mathbb{R}^d$

Mathematical formulation

- Initial assortment $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{D}|}$ with $\mathbf{x}_i \in \mathbb{R}^d \Rightarrow \mathbf{D} \in \mathbb{R}^{|\mathcal{D}| \times d}$

Mathematical formulation

- Initial assortment $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{D}|}$ with $\mathbf{x}_i \in \mathbb{R}^d \Rightarrow \mathbf{D} \in \mathbb{R}^{|\mathcal{D}|\times d}$
  - $\rightarrow$ Column-wise mean zero $\Rightarrow \sum_{i=1}^{|\mathcal{D}|} x_{ij} = 0 \qquad$ with $1 \leq j \leq d$

Mathematical formulation

- Initial assortment $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{D}|}$ with $\mathbf{x}_i \in \mathbb{R}^d \Rightarrow \mathbf{D} \in \mathbb{R}^{|\mathcal{D}| \times d}$
  - $\rightarrow$ Column-wise mean zero $\Rightarrow \sum_{i=1}^{|\mathcal{D}|} x_{ij} = 0 \qquad \text{with } 1 \leq j \leq d$

- Covariance matrix (Σ): pairwise relationships between features

$$\Sigma = \frac{1}{|\mathcal{D}| - 1} \mathbf{D}^T \mathbf{D} \qquad \left(\in \mathbb{R}^{d \times d} \right)$$

Mathematical formulation

- Initial assortment $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{D}|}$ with $\mathbf{x}_i \in \mathbb{R}^d \Rightarrow \mathbf{D} \in \mathbb{R}^{|\mathcal{D}| \times d}$
  - $\rightarrow$ Column-wise mean zero $\Rightarrow \sum_{i=1}^{|\mathcal{D}|} x_{ij} = 0 \qquad$ with $1 \leq j \leq d$

- Covariance matrix (Σ): pairwise relationships between features

$$\Sigma = \frac{1}{|\mathcal{D}| - 1} \mathbf{D}^T \mathbf{D} \qquad \left(\in \mathbb{R}^{d \times d} \right)$$

- Variance on the projected space: $\mathrm{Var}(\mathbf{Dv}) = \mathbf{v}^T \Sigma \mathbf{v}$

Mathematical formulation

- Initial assortment $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{D}|}$ with $\mathbf{x}_i \in \mathbb{R}^d \Rightarrow \mathbf{D} \in \mathbb{R}^{|\mathcal{D}| \times d}$
  $\rightarrow$ Column-wise mean zero $\Rightarrow \sum_{i=1}^{|\mathcal{D}|} x_{ij} = 0 \quad$ with $1 \leq j \leq d$

- Covariance matrix (Σ): pairwise relationships between features

$$\Sigma = \frac{1}{|\mathcal{D}| - 1} \mathbf{D}^T \mathbf{D} \qquad \left(\in \mathbb{R}^{d \times d} \right)$$

- Variance on the projected space: $\mathrm{Var}(\mathbf{Dv}) = \mathbf{v}^T \Sigma \mathbf{v}$
  $\rightarrow$ Key condition: *eigenvector* must maximize $\mathrm{Var}(\mathbf{Dv})$

Mathematical formulation

- First *eigenvector*: variance condition

Mathematical formulation

- First *eigenvector*: variance condition

$$\mathbf{v}_1 = \underset{||\mathbf{v}||=1}{\arg\max}\, \mathbf{v}^T \Sigma \mathbf{v}$$

Mathematical formulation

- First *eigenvector*: variance condition

$$\mathbf{v}_1 = \arg\max_{||\mathbf{v}||=1} \mathbf{v}^T \Sigma \mathbf{v}$$

- Second *eigenvector*: variance condition + orthogonal to v_1

Mathematical formulation

- First *eigenvector*: variance condition

$$\mathbf{v}_1 = \arg\max_{||\mathbf{v}||=1} \mathbf{v}^T \Sigma \mathbf{v}$$

- Second *eigenvector*: variance condition + orthogonal to v_1

$$\mathbf{v}_2 = \arg\max_{||\mathbf{v}||=1, \mathbf{v}\perp\mathbf{v}_1} \mathbf{v}^T \Sigma \mathbf{v}$$

Mathematical formulation

- First *eigenvector*: variance condition

$$\mathbf{v}_1 = \underset{||\mathbf{v}||=1}{\arg\max}\, \mathbf{v}^T \Sigma \mathbf{v}$$

- Second *eigenvector*: variance condition + orthogonal to v_1

$$\mathbf{v}_2 = \underset{||\mathbf{v}||=1, \mathbf{v}\perp\mathbf{v}_1}{\arg\max}\, \mathbf{v}^T \Sigma \mathbf{v}$$

- Third *eigenvector*: variance condition + orthogonal to $\mathbf{v}_1$, $\mathbf{v}_2$

$$\mathbf{v}_3 = \underset{||\mathbf{v}||=1, \mathbf{v}\perp\{\mathbf{v}_1, \mathbf{v}_2\}}{\arg\max}\, \mathbf{v}^T \Sigma \mathbf{v}$$

Mathematical formulation

- First *eigenvector*: variance condition

$$\mathbf{v}_1 = \arg\max_{||\mathbf{v}||=1} \mathbf{v}^T \Sigma \mathbf{v}$$

- Second *eigenvector*: variance condition + orthogonal to v_1

$$\mathbf{v}_2 = \arg\max_{||\mathbf{v}||=1, \mathbf{v}\perp\mathbf{v}_1} \mathbf{v}^T \Sigma \mathbf{v}$$

- Third *eigenvector*: variance condition + orthogonal to $\mathbf{v}_1$, $\mathbf{v}_2$

$$\mathbf{v}_3 = \arg\max_{||\mathbf{v}||=1, \mathbf{v}\perp\{\mathbf{v}_1,\mathbf{v}_2\}} \mathbf{v}^T \Sigma \mathbf{v}$$

- d'-th *eigenvector*: variance condition + orthogonal to $\{\mathbf{v}_i\}_{i=1}^{d'-1}$

$$\mathbf{v}_{d'} = \arg\max_{||\mathbf{v}||=1, \mathbf{v}\perp\{\mathbf{v}_i\}_{i=1}^{d'-1}} \mathbf{v}^T \Sigma \mathbf{v}$$

Mathematical formulation

- Vector $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{d'} \Rightarrow$ *eigenvectors* sorted by the encoded variance

Mathematical formulation

- Vector $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{d'} \Rightarrow$ *eigenvectors* sorted by the encoded variance
  - $\rightarrow$ Relates to the compression degree of the Dimesionality Reduction

Mathematical formulation

- Vector $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{d'} \Rightarrow$ *eigenvectors* sorted by the encoded variance
  $\rightarrow$ Relates to the compression degree of the Dimesionality Reduction

- How much information are we keeping? $\rightarrow$ Cumulative Variance

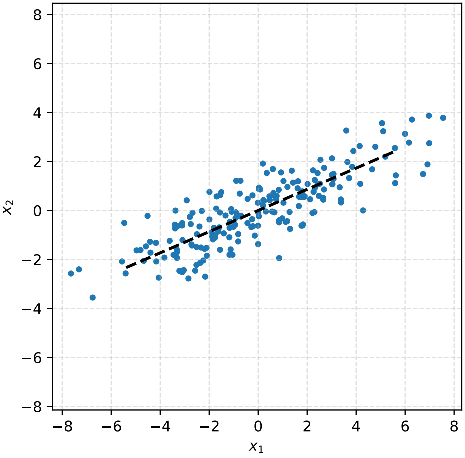
Mathematical formulation

- Vector $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{d'} \Rightarrow$ *eigenvectors* sorted by the encoded variance
  $\rightarrow$ Relates to the compression degree of the Dimesionality Reduction

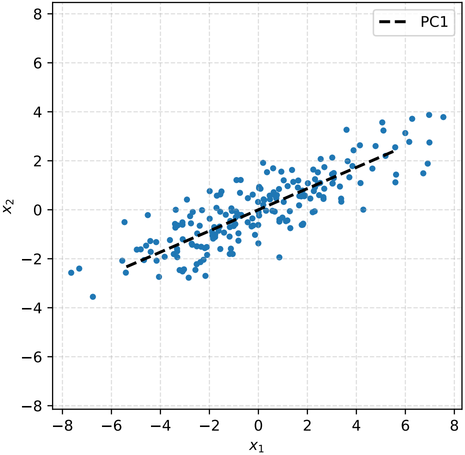
- How much information are we keeping? $\rightarrow$ Cumulative Variance

$$\mathrm{CumVar}(d') = \frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{j=1}^{d} \lambda_j} \quad \text{where } \lambda_j = \mathbf{v}_j^T \Sigma \mathbf{v}_j$$

Example - Single component



Example - Two components



t-Stochastic Nearest Embedding

- Non-linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:

t-Stochastic Nearest Embedding

- Non-linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:
  - → Typically, $d' = \{2, 3\}$
  - → Most commonly used for visualization purposes

t-Stochastic Nearest Embedding

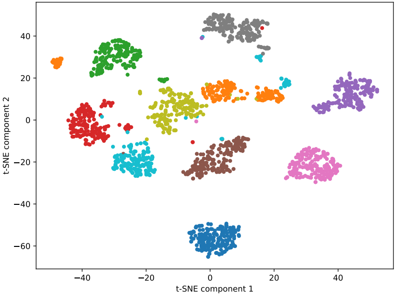
- Non-linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:
  - $\rightarrow$ Typically, $d' = \{2, 3\}$
  - $\rightarrow$ Most commonly used for visualization purposes

- **Key premise**: preserving local neighborhoods
  - $\rightarrow$ Points close in $\mathbb{R}^d$ remain close in $\mathbb{R}^{d'}$

t-Stochastic Nearest Embedding

- Non-linear projection from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$:
  - $\rightarrow$ Typically, $d' = \{2, 3\}$
  - $\rightarrow$ Most commonly used for visualization purposes

- **Key premise**: preserving local neighborhoods
  - $\rightarrow$ Points close in $\mathbb{R}^d$ remain close in $\mathbb{R}^{d'}$



Mathematical formulation

Models the problem from a probabilistic perspective:

Mathematical formulation

Models the problem from a probabilistic perspective:

1. Distribution $P \Rightarrow$ similarities in high-dimensional space
2. Distribution $Q \Rightarrow$ similarities in low-dimensional space

Mathematical formulation

Models the problem from a probabilistic perspective:

1. Distribution $P \Rightarrow$ similarities in high-dimensional space
2. Distribution $Q \Rightarrow$ similarities in low-dimensional space

Task: allocate elements in Q to resemble distribution in P

Mathematical formulation

Models the problem from a probabilistic perspective:

1. Distribution $P \Rightarrow$ similarities in high-dimensional space
2. Distribution $Q \Rightarrow$ similarities in low-dimensional space

Task: allocate elements in Q to resemble distribution in P

- Q is considerably less dimensional than P

Mathematical formulation

Models the problem from a probabilistic perspective:

1. Distribution $P \Rightarrow$ similarities in high-dimensional space
2. Distribution $Q \Rightarrow$ similarities in low-dimensional space

Task: allocate elements in Q to resemble distribution in P

- Q is considerably less dimensional than P
- Optimization goal:

$$\min \sum_{\mathbf{x} \in \mathcal{D}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{Q(\mathbf{x})}$$

Mathematical formulation

Models the problem from a probabilistic perspective:

1. Distribution $P \Rightarrow$ similarities in high-dimensional space
2. Distribution $Q \Rightarrow$ similarities in low-dimensional space

Task: allocate elements in Q to resemble distribution in P

- Q is considerably less dimensional than P
- Optimization goal:

$$\min \sum_{\mathbf{x} \in \mathcal{D}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{Q(\mathbf{x})} = \min D_{KL}(P||Q)$$

$D_{KL} \rightarrow$ Kullback-Leibler Divergence

Autoencoders

- Neural *disposition* devised for learning efficient representations

Autoencoders

- Neural *disposition* devised for learning efficient representations
  - $\rightarrow$ Typically known as latent representation / embedding

Autoencoders

- Neural *disposition* devised for learning efficient representations
  - $\rightarrow$ Typically known as latent representation / embedding
  - $\rightarrow$ Unsupervised $\Rightarrow$ Which is the target for training the model?

Autoencoders

- Neural *disposition* devised for learning efficient representations
  - $\rightarrow$ Typically known as latent representation / embedding
  - $\rightarrow$ Unsupervised $\Rightarrow$ Which is the target for training the model?
  - $\rightarrow$ *Shallow* and *deep* models

Autoencoders

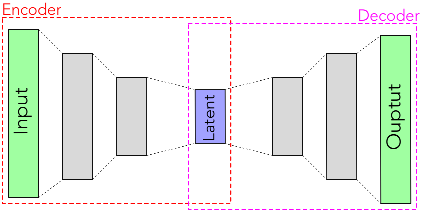
- Neural *disposition* devised for learning efficient representations
  - $\rightarrow$ Typically known as latent representation / embedding
  - $\rightarrow$ Unsupervised $\Rightarrow$ Which is the target for training the model?
  - $\rightarrow$ *Shallow* and *deep* models

- Exhibits two parts:

Autoencoders

- Neural *disposition* devised for learning efficient representations
  - → Typically known as latent representation / embedding
  - → Unsupervised ⇒ Which is the target for training the model?
  - → *Shallow* and *deep* models

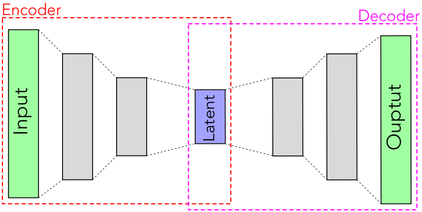
- Exhibits two parts:
  1. Encoder: Reduces the dimensionality of the data
  2. Decoder: Recovers the original dimensionality of the data



Autoencoders

- Neural *disposition* devised for learning efficient representations
  - → Typically known as latent representation / embedding
  - → Unsupervised ⇒ Which is the target for training the model?
  - → *Shallow* and *deep* models

- Exhibits two parts:
  1. Encoder: Reduces the dimensionality of the data
  2. Decoder: Recovers the original dimensionality of the data



- Once trained, we are only interested in the Encoder
  - → Decoder in this case is used for training purposes

Mathematical formulation

Goal: Learn two mapping functions:

Mathematical formulation

Goal: Learn two mapping functions:

- Encoder: $E_{\phi_E} : \mathbb{R}^d \to \mathbb{R}^{d'}$

Mathematical formulation

Goal: Learn two mapping functions:

- Encoder: $E_{\phi_E} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$
- Decoder: $D_{\phi_D} : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$

Mathematical formulation

Goal: Learn two mapping functions:

- Encoder: $E_{\phi_E} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$
- Decoder: $D_{\phi_D} : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$

Given an input datum $\mathbf{x}$:

- Embedded representation: $\mathbf{x}^e = E_{\phi_E}(\mathbf{x})$
- Reconstructed datum: $\tilde{\mathbf{x}} = D_{\phi_D}(\mathbf{x}^e) = D_{\phi_D}(E_{\phi_E}(\mathbf{x}))$

Mathematical formulation

Goal: Learn two mapping functions:

- Encoder: $E_{\phi_E} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$
- Decoder: $D_{\phi_D} : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$

Given an input datum $\mathbf{x}$:

- Embedded representation: $\mathbf{x}^e = E_{\phi_E}(\mathbf{x})$
- Reconstructed datum: $\tilde{\mathbf{x}} = D_{\phi_D}(\mathbf{x}^e) = D_{\phi_D}(E_{\phi_E}(\mathbf{x}))$

Optimization: minimize the difference between input and reconstruction

$$\underset{\phi_E,\phi_D}{\arg\min}\, \mathcal{L}\left(\mathbf{x}, \tilde{\mathbf{x}}\right) = \underset{\phi_E,\phi_D}{\arg\min}\, \mathcal{L}\left(\mathbf{x}, D_{\phi_D}\left(E_{\phi_E}(\mathbf{x})\right)\right)$$

Final considerations

- Typically symmetrical but not a necessary condition
  - → Decoder $D_{\phi_D}(\cdot)$ *undoes* the mapping by the encoder $E_{\phi_E}(\cdot)$

Final considerations

- Typically symmetrical but not a necessary condition
  - → Decoder $D_{\phi_D}(\cdot)$ *undoes* the mapping by the encoder $E_{\phi_E}(\cdot)$

- There exist other versions of these structures:
  - → Variational AE: Assumes data follows a particular statistical distribution
  - → Denoising AE: Trained to remove noise from input data

Final considerations

- Typically symmetrical but not a necessary condition
  - → Decoder $D_{\phi_D}(\cdot)$ *undoes* the mapping by the encoder $E_{\phi_E}(\cdot)$

- There exist other versions of these structures:
  - → Variational AE: Assumes data follows a particular statistical distribution
  - → Denoising AE: Trained to remove noise from input data

- Related to PCA:
  - → Single hidden layer with linear activation ⇒ Equivalent approaches

Outline

Other unsupervised learning tasks

1. Outlier detection
   - Detect elements whose features remarkably differ from the rest
   - **Strategies**: PCA error, Autoencoder, Isolation Forest

Other unsupervised learning tasks

1. Outlier detection
   - Detect elements whose features remarkably differ from the rest
   - **Strategies**: PCA error, Autoencoder, Isolation Forest

2. Generative modeling
   - Generate new samples similar to the existing ones
   - **Strategies**: Variational AE (decoder $D_{\phi_D}(\cdot)$)

Other unsupervised learning tasks

1. Outlier detection
   - Detect elements whose features remarkably differ from the rest
   - **Strategies**: PCA error, Autoencoder, Isolation Forest

2. Generative modeling
   - Generate new samples similar to the existing ones
   - **Strategies**: Variational AE (decoder $D_{\phi_D}(\cdot)$)

3. Motif discovery
   - Mining patterns in time-series data collections

T6: Unsupervised learning

Fundamentos del Aprendizaje Automático

Curso 2025/2026