Principal Component Analysis and Autoencoders

Jean-sébastien Delineau, Eugène Mettraux, Xiao Xiang

EPFL

December 2023

Motivation: PCA

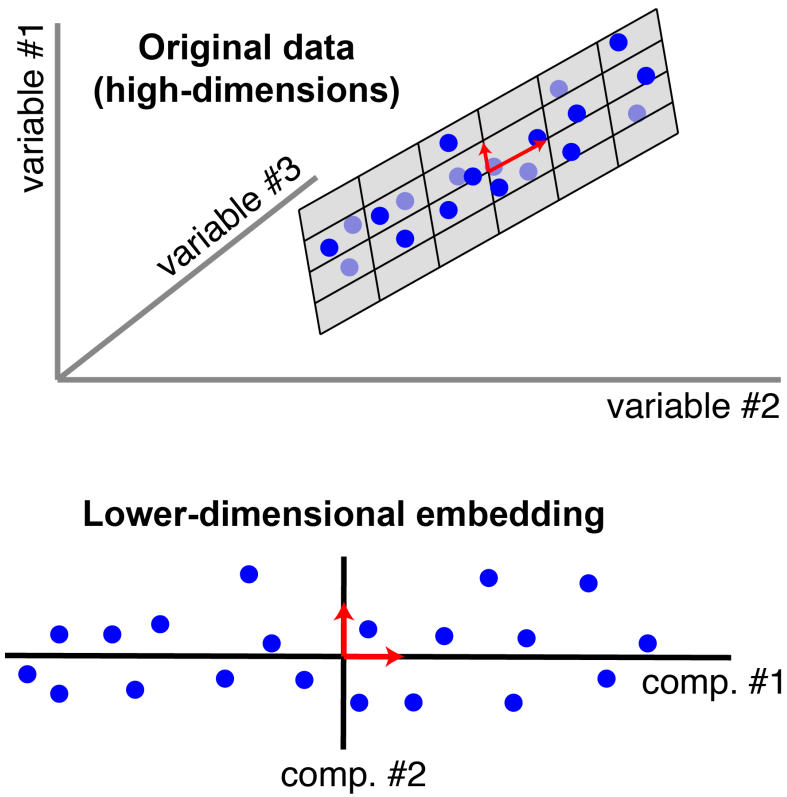


Figure: Dimension Reduction

- Dimensionality Reduction
- Feature Extraction and Data Compression
- Data Visualization:

Motivation: Autoencoders

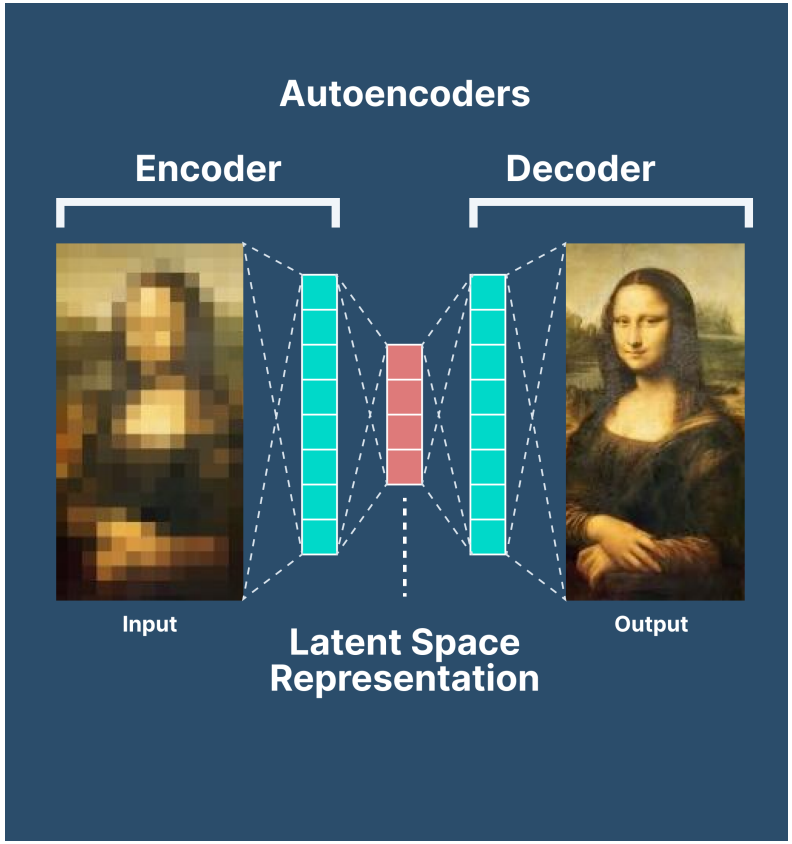


Figure: Denoising propriety of autoencoder

- Data compression
- Image denoising
- Dimensionality reduction

Methodology:

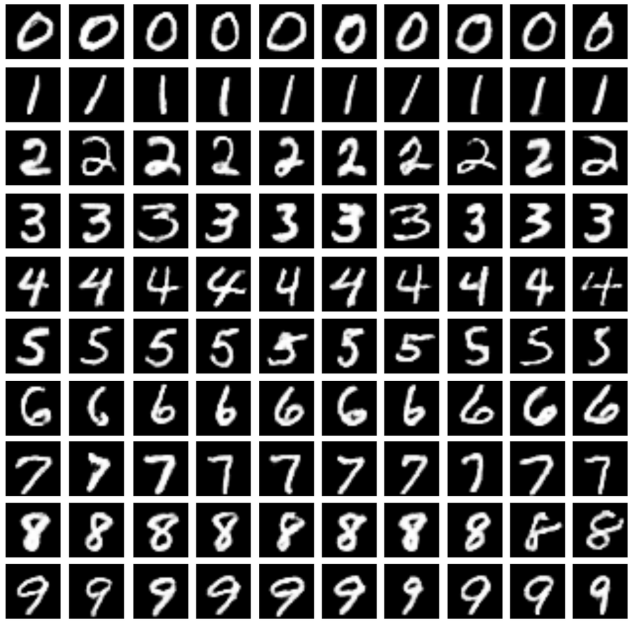


Figure: MNIST data set

- Training set 60'000, test set 10'000
- Image of size 28x28, considered as a vector 1x784
- The data set is centered to zero and normalized to 1

Theory: PCA

Aiming to find an independent component maximizing the variance of the data:

$$\mathbf{p}_1 = \max_{\mathbf{w}_1} \operatorname{cov}(\mathbf{w}_1^T X, \mathbf{w}_1^T X) = \max_{\mathbf{w}_1} \mathbf{w}_1^T X X^T \mathbf{w}_1 \quad \text{s.t} \quad \mathbf{w}_1^T \mathbf{w}_1 = 1.$$

We retrieve such components by applying the SVD on X, since it also retrieves the eigenvectors of the matrix XX^T:

$$X = U\Sigma V^T \implies XX^T = U\Sigma\Sigma^T U^T \quad U \in R^{n\times n}, V \in R^{N\times N}.$$

It can be shown with Eckart-Young theorem that the matrix $U_m = [\mathbf{u}_1, \cdots, \mathbf{u}_m]$ is a solution to:

$$\min_{W \in R^{n\times m}} ||X - WW^T X||_F^2 \quad \text{s.t} \quad W^T W = I_{m\times m}$$

.

Theory: Autoencoders

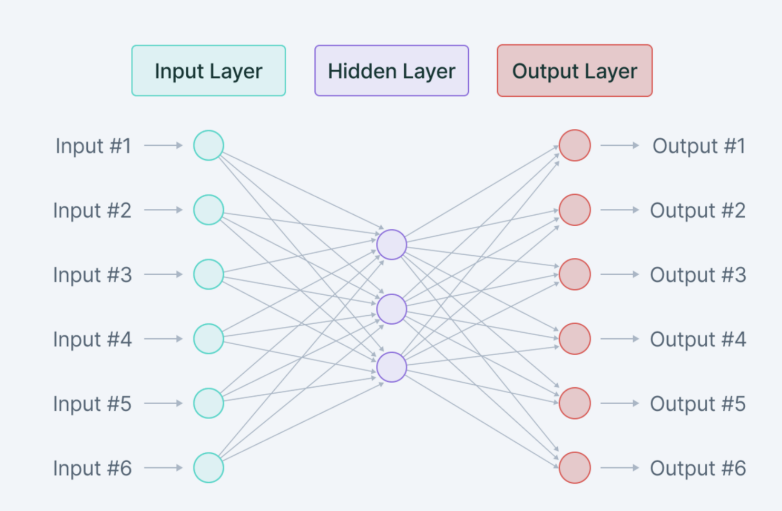


Figure: Single-layer Autoencoder

- Encoding Function

$$\mathbf{a}_i = f\left(\mathbf{W}^{(1)}\mathbf{x}_i + \mathbf{b}^{(1)}\right)$$

- Decoding Function

$$\mathbf{y}_i = g\left(\mathbf{W}^{(2)}\mathbf{a}_i + \mathbf{b}^{(2)}\right)$$

Goal: To learn the parameters $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}$ such that the output $\mathbf{y}$ is a good approximation of the input $\mathbf{x}$:

$$\underset{\mathbf{W}^{(1)},\mathbf{W}^{(2)},\mathbf{b}^{(1)},\mathbf{b}^{(2)}}{\text{minimize}} \frac{1}{N}\Sigma_{i=1}^{N}\left(||\mathbf{x}_i - g(\mathbf{W}^{(2)}f\left(\mathbf{W}^{(1)}\mathbf{x}_i + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)})||^2\right)$$

Goal: Implementation

Goals:

- Compare PCA and Linear Autoencoder on a dimension reduction task
- Apply both methods on a denoising task
- Search for better architectures for denoising task (e.g., Convolutional Autoencoder)

Difficulties encountered

- Setting the hyperparameters such as the learning rate, the number of time steps, the size of the hidden layer, etc.

Result: PCA-dimension of projected space

The added noise increases the MSE, and modifies the convergence.

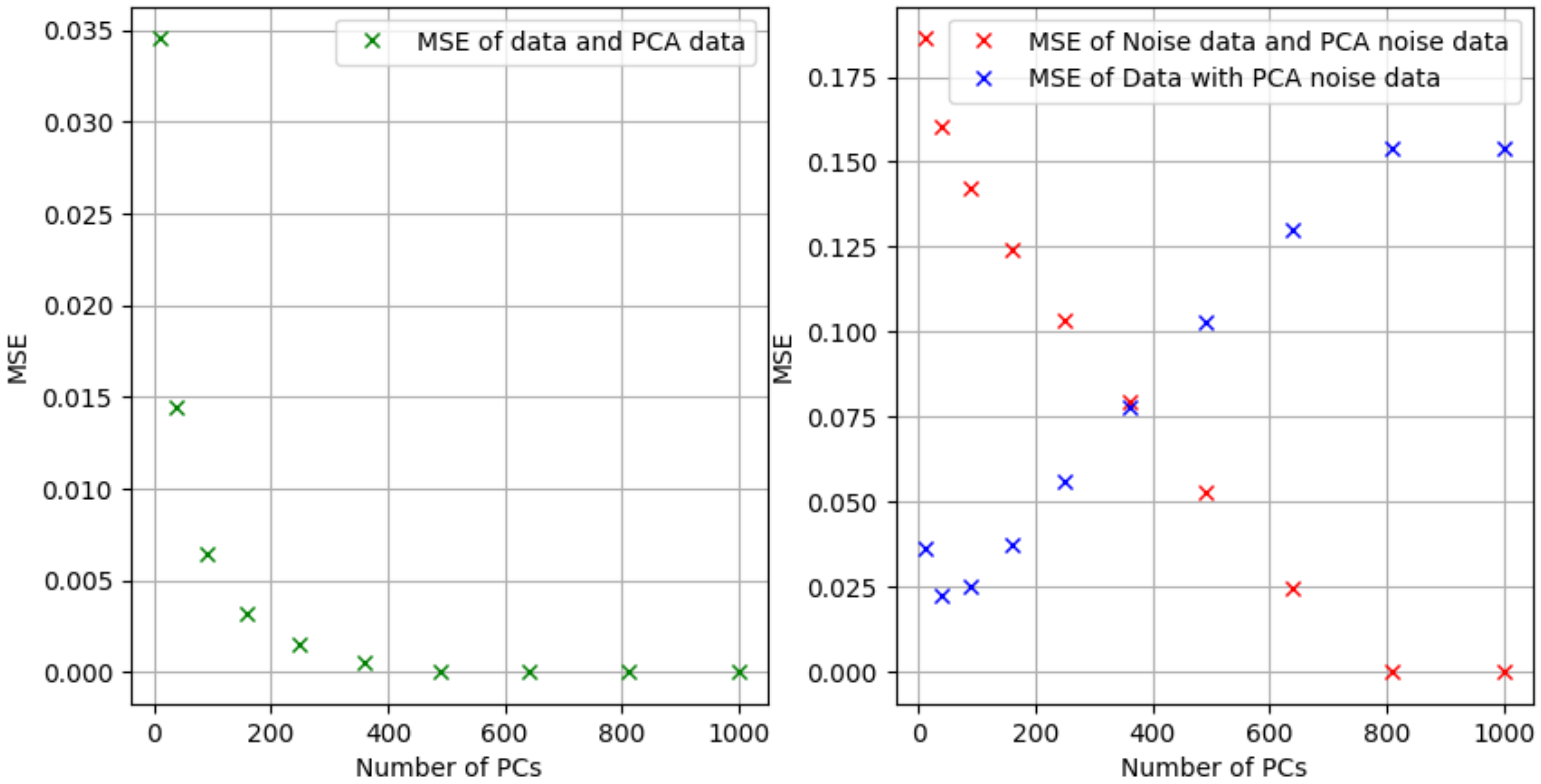
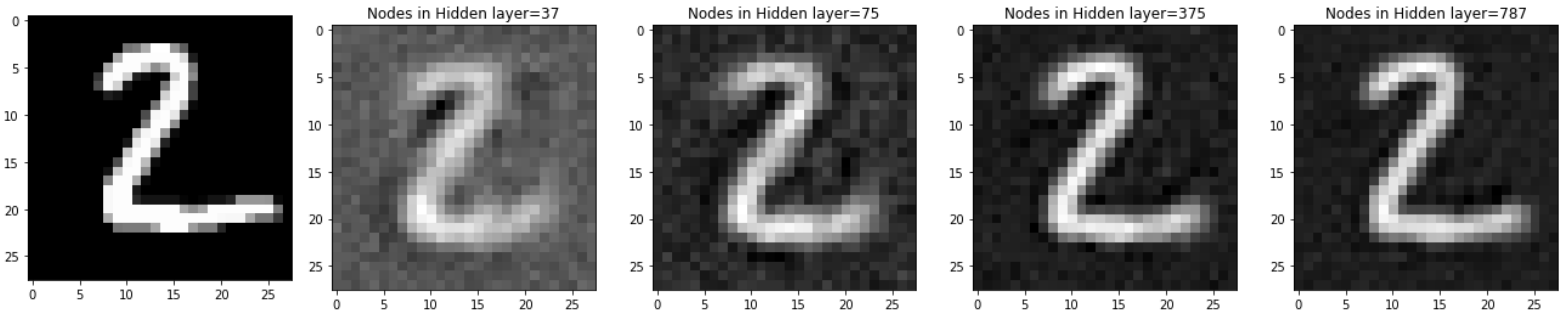


Figure on the right hint that PCA could have some denoising proprieties. The noise is set such that $X'_{i,j} = X_{i,j} + 100 * \mu_{i,j}$ with $\mu_{i,j} \sim \mathcal{N}(0, 1)$

Result: Linear AE on MNIST

The Autoencoder is trained by SGD over $T = 10'000$ steps, with a learning of $\eta = 1$ for different number of nodes in the hidden layer.



Result: Linear AE Iteration

The Autoencoder is trained by SGD over $T = 10'000$ steps, with a learning of $\eta = 1$ for different number of nodes in the hidden layer.

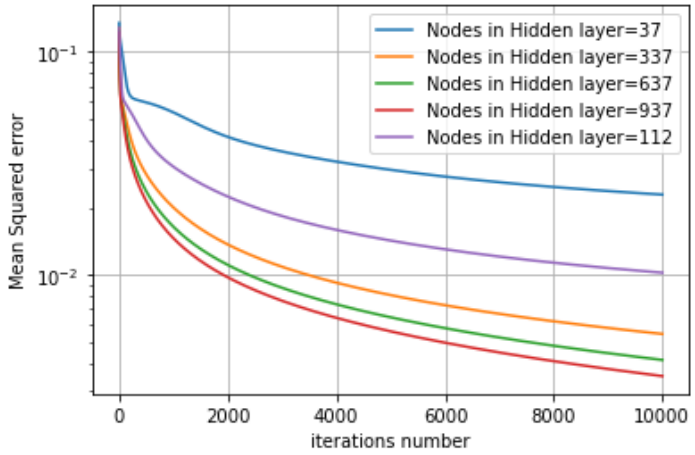


Figure: Evolution of the training error over the training

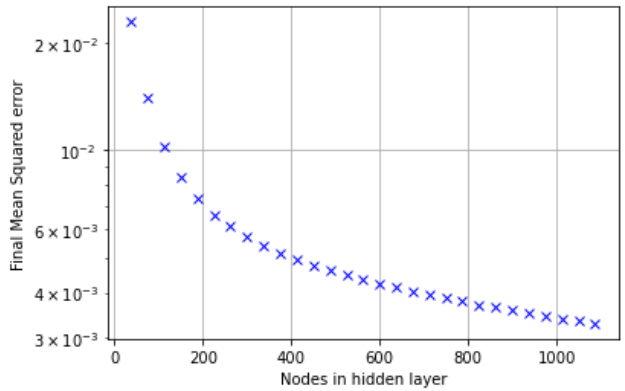


Figure: Final Mean Squared Error as a function of the number of nodes in the hidden layer

Result: Denoising

The dataset is noised such that $X'_{i,j} = X_{i,j} + 100 * \mu_{i,j}$ with $\mu_{i,j} \sim \mathcal{N}(0, 1)$

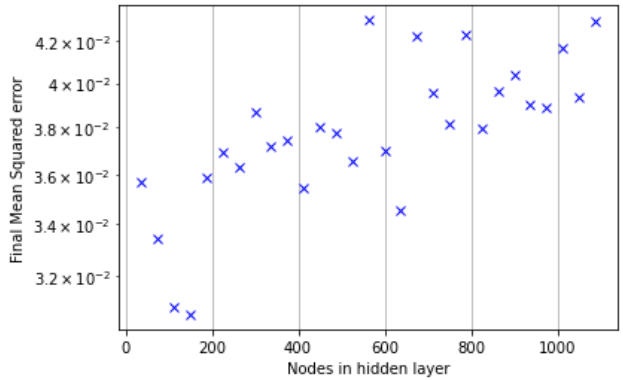
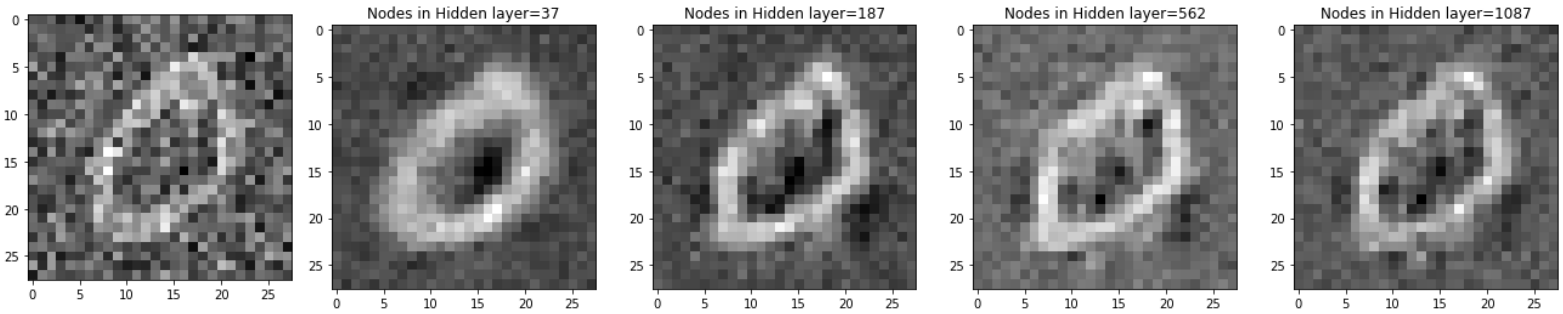


Figure: Mean squared error as a function of the number of nodes in the hidden layer



THANK YOU!

Finding Principle Components

- Maximum Variance Direction

$$\mathbf{p}_1 = \max_{\mathbf{w}_1} \mathbf{w}_1^T X X^T \mathbf{w}_1 \quad \text{s.t} \quad \mathbf{w}_1^T \mathbf{w}_1 = 1.$$

- Recursively finding other components

$$\mathbf{p}_2 = \max_{\mathbf{w}_2} \mathbf{w}_2^T (X - \mathbf{p}_1 \mathbf{p}_1^T X)(X - \mathbf{p}_1 \mathbf{p}_1^T X)^T \mathbf{w}_2 \quad \text{s.t} \quad \mathbf{w}_2^T \mathbf{w}_2 = 1.$$

$\text{cov(X,X)} = \text{E[(X-E[X])(X-E[X])}^T] = E[XX^T] = XX^T$

Singular Value Decomposition

$$X = U\Sigma V^T \implies XX^T = U\Sigma\Sigma^T U^T \quad U \in R^{n\times n}, V \in R^{N\times N}$$

Stochastic gradient Descent

we follow the gradient descent update:

$$\mathbf{W}_{l,j}^{(k)} := \mathbf{W}_{l,j}^{(k)} - \alpha \frac{\partial}{\partial \mathbf{W}_{l,j}^{(k)}} \mathbf{F}(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}),$$

$$\mathbf{b}_{l}^{(k)} := \mathbf{b}_{l}^{(k)} - \alpha \frac{\partial}{\partial \mathbf{b}_{l}^{(k)}} \mathbf{F}(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)})$$

where k denotes the numbering of the layer, α the learning rate, and $\mathbf{F}$ the objective function of the minimization problem.

Back-propogation

For each output unit of the output vector $\mathbf{y}'_l$, we compute the error term defined as:

$$\epsilon_l^{(k=2)} = -(y_l - y'_l)[\mathbf{g}'(\mathbf{W}^{(2)}\mathbf{a} + \mathbf{b}^{(2)})]_l$$

Subsequently, we compute for each node of the hidden layer:

$$\epsilon_j^{(k=1)} = (\Sigma_{l=1}^{m} \mathbf{W}_{l,j}^{(1)} \epsilon_l^{(k=2)})[\mathbf{f}'(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})]_j]$$

Given the error terms, we are able to compute the partial derivatives:

$$\frac{\partial}{\partial \mathbf{W}_{j,l}^{(2)}} \mathbf{F}(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}) = y'_l \epsilon_l^{(k=2)}$$

$$\frac{\partial}{\partial \mathbf{W}_{l,j}^{(1)}} \mathbf{F}(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}) = a_j \epsilon_j^{(k=1)}$$

$$\frac{\partial}{\partial \mathbf{b}_l^{(2)}} \mathbf{F}(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}) = \epsilon_l^{(k=2)}$$

$$\frac{\partial}{\partial \mathbf{b}_j^{(1)}} \mathbf{F}(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}) = \epsilon_j^{(k=1)}$$

With iteration, we are able to train the samples and obtain the results efficiently.

Proof - Equivalence

We start by proving the equivalence of **??** to the following optimization problem:

$$\min_{\mathbf{Z} \in \mathbb{R}^{N \times n}} \quad \|\mathbf{X} - \mathbf{Z}\|_F^2 \quad \text{s.t.} \quad \text{rank}(\mathbf{Z}) \leq m, \quad \text{Im}(\mathbf{Z}) \subseteq \text{Im}(\mathbf{X}). \tag{1}$$

Without loss of generality, we can conveniently choose the activation function f, g to be the identity map. Therefore, **??** can be reduced as:

$$\underset{\mathbf{W}^{(1)},\mathbf{W}^{(2)},\mathbf{b}^{(1)},\mathbf{b}^{(2)}}{\text{minimize}} \frac{1}{N}\Sigma_{i=1}^{N}\left(||\mathbf{x}_i - \mathbf{W}^{(2)}\left(\mathbf{W}^{(1)}\mathbf{x}_i + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)}||^2\right) \tag{2}$$

Additionally, we define two matrices $\mathbf{A} \in \mathbb{R}^{m\times(n+1)}, \mathbf{B} \in \mathbb{R}^{(n+1)\times m}$ such that $\mathbf{A}\mathbf{x}_i = \mathbf{W}^{(1)}\mathbf{x}_i + \mathbf{b}^{(1)}, \mathbf{B}\mathbf{a}_i = \mathbf{W}^{(2)}\mathbf{a}_i + \mathbf{b}^{(2)}$.[2] We thus rewrite eq. (2) as:

$$\underset{\mathbf{A}\in\mathbb{R}^{m\times(n+1)},\mathbf{B}\in\mathbb{R}^{(n+1)\times m}}{\text{minimize}} \frac{1}{N}\Sigma_{i=1}^{N}\left(||\mathbf{x}_i - \mathbf{B}\mathbf{A}\mathbf{x}_i||^2\right) \tag{3}$$

[2] This is allowed as we simply reformulate the question from an affine model setting to a linear model setting

We further construct the feature matrix $\mathbf{X} = [x_1, \cdots, x_N] \in \mathbf{R}^{n \times N}$, so we can write eq. (3) as:

$$\underset{\mathbf{A} \in \mathbb{R}^{m \times (n+1)}, \mathbf{B} \in \mathbb{R}^{(n+1) \times m}}{\text{minimize}} \frac{1}{N} ||\mathbf{X} - \mathbf{X}\mathbf{A}^\top \mathbf{B}^\top||_F^2 \tag{4}$$

which is equivalent to:

$$\underset{\mathbf{A} \in \mathbb{R}^{m \times (n+1)}, \mathbf{B} \in \mathbb{R}^{(n+1) \times m}}{\text{minimize}} \frac{1}{N} ||\mathbf{X} - \mathbf{X}\mathbf{M}||_F^2 \tag{5}$$

Where $\mathbf{M} \in \mathbb{R}^{(n+1) \times (n+1)} = \mathbf{A}^\top \mathbf{B}^\top$ with $\text{rank}(\mathbf{M}) \leq m$
Lastly, we set $\mathbf{Z} = \mathbf{X}\mathbf{M}$. By definition, $\text{Im}(\mathbf{Z}) \subseteq \text{Im}(\mathbf{X})$. Additionally, we can check that $\text{rank}(\mathbf{Z}) \leq \min(\text{rank}(\mathbf{X}), \text{rank}(\mathbf{M})) \leq \text{rank}(\mathbf{M})$.

We further construct the feature matrix $\mathbf{X} = [x_1, \cdots, x_N] \in \mathbf{R}^{n \times N}$, so we can write eq. (3) as:

$$\underset{\mathbf{A} \in \mathbb{R}^{m \times (n+1)}, \mathbf{B} \in \mathbb{R}^{(n+1) \times m}}{\text{minimize}} \frac{1}{N} ||\mathbf{X} - \mathbf{X}\mathbf{A}^\top \mathbf{B}^\top||_F^2 \tag{6}$$

which is equivalent to:

$$\underset{\mathbf{A} \in \mathbb{R}^{m \times (n+1)}, \mathbf{B} \in \mathbb{R}^{(n+1) \times m}}{\text{minimize}} \frac{1}{N} ||\mathbf{X} - \mathbf{X}\mathbf{M}||_F^2 \tag{7}$$

Where $\mathbf{M} \in \mathbb{R}^{(n+1) \times (n+1)} = \mathbf{A}^\top \mathbf{B}^\top$ with $\text{rank}(\mathbf{M}) \leq m$
Lastly, we set $\mathbf{Z} = \mathbf{X}\mathbf{M}$. By definition, $\text{Im}(\mathbf{Z}) \subseteq \text{Im}(\mathbf{X})$. Additionally, we can check that $\text{rank}(\mathbf{Z}) \leq \min(\text{rank}(\mathbf{X}), \text{rank}(\mathbf{M})) \leq \text{rank}(\mathbf{M})$.

We then start to proceed with the remaining proof such that the result from PCA is equivalent to the optimization problem eq. (1):
We first observe that by the Eckart-Young theorem, the matrix $\mathbf{Z}$containing the m principal components from the truncated transformation of PCA satisfies the following optimization:

$$\min_{\mathbf{Z}} \quad \|\mathbf{X} - \mathbf{Z}\|_F^2 \quad \text{s.t.} \quad \text{rank}(\mathbf{Z}) \leq m. \tag{8}$$

and its solution is of the form $\mathbf{Z} = \mathbf{X}_{[m]} = \mathbf{U}_{[m]}\mathbf{S}_{[m]}\mathbf{V}_{[m]}$ with $\mathbf{U} \in \mathbb{R}^{N\times N}$ an orthogonal basis of $\mathbb{R}^N$, $\mathbf{S} \in \mathbb{R}^{N\times n}$ a diagonal matrix, and $\mathbf{V} \in \mathbb{R}^{n\times n}$ an orthogonal basis of $\mathbb{R}^n$ from the single value decomposition.
We then state that the solution to eq. (8) is equivalent to the one to eq. (1).
Indeed, eq. (8) is equivalent to eq. (1) without the image inclusion constraint. We additionally note that $\mathbf{U}_{[m]}$ is by definition in $\text{Im}(\mathbf{X})$.
Therefore, we conclude that the solution to both problems are equivalent.