Computational Intelligence Lab

Cristian Perez Jensen

April 14, 2024

Contents

List of symbols

$\doteq$	Equality by definition
$\approx$	Approximate equality
$\propto$	Proportional to
$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$i : j$	Set of natural numbers between i and j. I.e., $\{i, i+1, \ldots, j\}$
$f : A \to B$	Function f that maps elements of set A to elements of set B
$\mathbb{1}\{\text{predicate}\}$	Indicator function (1 if predicate is true, otherwise 0)

$\boldsymbol{v} \in \mathbb{R}^n$	n-dimensional vector
$\boldsymbol{M} \in \mathbb{R}^{m \times n}$	$m \times n$ matrix
$\mathbf{T} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$	Tensor
$\boldsymbol{M}^\top$	Transpose of matrix $\boldsymbol{M}$
$\boldsymbol{M}^{-1}$	Inverse of matrix $\boldsymbol{M}$
$\det(\boldsymbol{M})$	Determinant of $\boldsymbol{M}$

$\frac{\mathrm{d}}{\mathrm{d}x} f(x)$	Ordinary derivative of $f(x)$ w.r.t. x at point $x \in \mathbb{R}$
$\frac{\partial}{\partial x} f(\boldsymbol{x})$	Partial derivative of $f(\boldsymbol{x})$ w.r.t. x at point $\boldsymbol{x} \in \mathbb{R}^n$
$\boldsymbol{\nabla}_{\boldsymbol{x}} f(\boldsymbol{x}) \in \mathbb{R}^n$	Gradient of $f : \mathbb{R}^n \to \mathbb{R}$ at point $\boldsymbol{x} \in \mathbb{R}^n$
$\boldsymbol{\nabla}_{\boldsymbol{x}}^2 f(\boldsymbol{x}) \in \mathbb{R}^{n \times n}$	Hessian of $f : \mathbb{R}^n \to \mathbb{R}$ at point $\boldsymbol{x} \in \mathbb{R}^n$

$\boldsymbol{\theta} \in \Theta$	Parametrization of a model, where Θ is a compact subset of $\mathbb{R}^K$
$\mathcal{X}$	Input space
$\mathcal{Y}$	Output space
$\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$	Labeled training data

1 Dimensionality reduction

The motivation behind dimensionality reduction is to find a low-dimensional representation of high-dimensional data.[1] This can be split into two goals: (1) compressing the data, while preserving as much as possible of the relevant information, and (2) interpreting the data in low dimensionality is easier than high dimensionality.

[1] Often, the original raw representation is high-dimensional and redundant, *e.g.*, images, audio, time series.

Dimensionality reduction is often performed by an autoencoder, which typically has a bottleneck and aims to predict its input; see Figure 1. In general, we have an encoder F and a decoder G,

$$F : \mathbb{R}^n \to \mathbb{R}^m, \quad G : \mathbb{R}^m \to \mathbb{R}^n.$$

$m \ll n$.

The reconstruction function is then the following function,

$$G \circ F : \mathbb{R}^n \to \mathbb{R}^n,$$

which is ideally the identity function.

1.1 Linear autoencoders

To build a nice theory, we will only consider a single layer linear autoencoder, which means that we have the following functions,

$$F : x \mapsto z = Wx, \quad W \in \mathbb{R}^{m \times n}.$$
$$G : z \mapsto \hat{x} = Vz, \quad V \in \mathbb{R}^{n \times m}.$$

The objective to minimize of the linear encoder is the following,

$$\mathcal{R}(W, V) = \mathcal{R}(P \doteq VW) \doteq \mathbb{E}\left[\frac{1}{2}\|x - Px\|^2\right].$$

Input   Bottleneck   Output

W z V

x $\hat{x}$

Figure 1. Diagram of a single layer linear autoencoder.

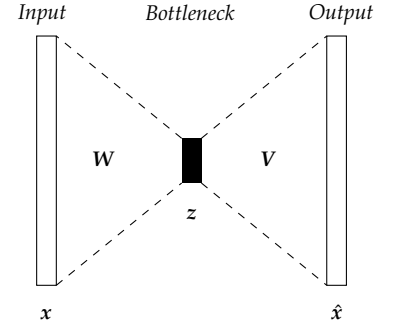> **Corollary.** For centered data, *i.e.* $E[x] = 0$, optimal affine maps degenerate to linear ones.

Proof. Proof by contradiction. Let $a \neq 0$, then

$$\mathbb{E}\left[\|x - (Px + a)\|^2\right] = \mathbb{E}[\langle x - Px - a, x - Px - a\rangle]$$
$$= \mathbb{E}[\langle x - Px, x - Px\rangle + 2\langle x - Px, -a\rangle + \langle -a, -a\rangle]$$
$$= \mathbb{E}\left[\|x - Px\|^2\right] - 2\underbrace{\langle \mathbb{E}[x] - P\mathbb{E}[x], a\rangle}_{=0} + \underbrace{\|a\|^2}_{>0}$$
$$> \mathbb{E}\left[\|x - Px\|^2\right].$$

Thus, the risk is strictly worse if $a \neq 0$. ∎

Thus, we will assume that the data is centered, which makes the analysis easier, since we do not need to consider the affine case.

Note that while the optimal linear reconstruction map P is unique, its parametrization W, V is not unique, since for any invertible matrix $A \in \mathbb{R}^{m \times m}$, we can construct an optimal parametrization,

$$VW = VIW = V(AA^{-1})W = (VA)(A^{-1}W),$$

with $A^{-1}W, VA$. The weight matrices are non-identifiable.

Since P cannot be any $n \times n$ matrix, we want to know how the composition of $W \in \mathbb{R}^{m \times n}$ and $V \in \mathbb{R}^{n \times m}$ characterizes the matrix P and which constraints they impose. The answer to this is that the weight matrices impose a rank constraint on P,

$$\text{rank}(P) = \min\{\text{rank}(W), \text{rank}(W)\} \le \min\{m, n\} = m.$$

Thus, when optimizing for P, we are constrained to matrices with rank less or equal to m.

1.2   Projection

The rank constraint and linearity of P means that the image (column space) of P is a linear subspace $\mathcal{U} \subseteq \mathbb{R}^n$ of dimension at most m. We will break the solution to our problem into two parts: (1) finding the optimal subspace $\mathcal{U}$, and (2) finding the optimal mapping to that subspace.[2]

[2] We do not search for the weight matrices W, V, since they are not unique, but P is unique.

Finding the optimal mapping to a subspace. We will first focus on (2); given subspace $\mathcal{U}$, we need to determine the optimal linear map $P^\star$, such that

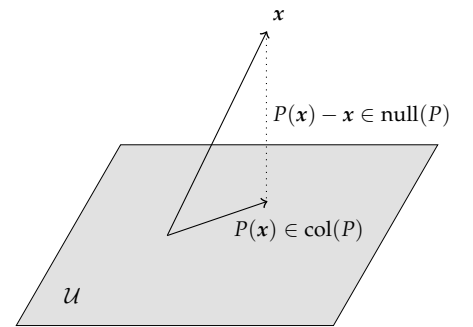$$P^\star = \underset{P}{\text{argmin}} \, \|x - Px\|^2, \quad \text{col}(P) = \mathcal{U}.$$

Definition 1 (Orthogonal projection). A linear transformation $P : \mathcal{V} \to \mathcal{V}$ is called an orthogonal projection onto $\mathcal{U}$ if $\forall x \in \mathcal{V}$:

1. *Projection*: $P(x) \in \mathcal{U}$;

2. *Orthogonality*: $\text{null}(P) \perp \text{col}(P)$, which is equivalent to the following holding, $\langle P(x), y \rangle = \langle x, P(y) \rangle$ (self-adjointness);

3. *Idempotency*: $P(P(x)) = P(x)$.

Definition 2. The orthogonal projection to a linear subspace $\mathcal{U} \subseteq \mathbb{R}^n$ is defined as

$$\Pi_{\mathcal{U}} : \mathbb{R}^n \to \mathcal{U}, \quad \Pi_{\mathcal{U}}(x) = \underset{x' \in \mathcal{U}}{\text{argmin}} \, \|x - x'\|.$$

Figure 2. Orthogonal projection of x onto subspace plane $\mathcal{U}$.

Proof. We need to show that the definition of $\Pi_{\mathcal{U}}$ indeed is an orthogonal projection by showing that it adheres to the properties of Definition 1.

1. *Projection*: This is true by definition of the values that the argmin are allowed to take on;

2. *Idempotency*: For all $\boldsymbol{u} \in \mathcal{U}$, $\Pi_{\mathcal{U}}(\boldsymbol{u}) = \operatorname{argmin}_{\boldsymbol{x}' \in \mathcal{U}} \|\boldsymbol{u} - \boldsymbol{x}'\| = \boldsymbol{u}$. Thus, $\Pi_{\mathcal{U}} = \Pi_{\mathcal{U}} \circ \Pi_{\mathcal{U}}$;

3. *Orthogonality*: We need to show that $\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x} \in \mathcal{U}^{\perp}$. Decompose it into $\boldsymbol{u} \in \mathcal{U}$ and $\boldsymbol{u}^{\perp} \in \mathcal{U}^{\perp}$ by $\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x} = \boldsymbol{u} + \boldsymbol{u}^{\perp}$. Then, we only need to show that $\boldsymbol{u} = \boldsymbol{0}$.

   Proof by contradiction. Let $\boldsymbol{u} \neq \boldsymbol{0}$, then

$$
\begin{aligned}
\|\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}\|^2 &= \underbrace{\|\boldsymbol{u}\|^2}_{>0} + \|\boldsymbol{u}^{\perp}\|^2 + 2 \underbrace{\langle \boldsymbol{u}, \boldsymbol{u}^{\perp} \rangle}_{=0} \\
&> \|\boldsymbol{u}^{\perp}\|^2 \\
&= \|\underbrace{\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{u}}_{\in \mathcal{U}} - \boldsymbol{x}\|^2,
\end{aligned}
$$

   which contradicts with

$$
\|\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}\| = \min_{\boldsymbol{x}' \in \mathcal{U}} \|\boldsymbol{x}' - \boldsymbol{x}\| \leq \|\tilde{\boldsymbol{u}} - \boldsymbol{x}\|, \quad \forall \tilde{\boldsymbol{u}} \in \mathcal{U}.
$$

   Hence, $\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x} \in \mathcal{U}^{\perp}$ and is unique;

4. *Linearity*: We need to show homogeneity,

$$
\begin{aligned}
\Pi_{\mathcal{U}}(\alpha \boldsymbol{x}) &= \operatorname*{argmin}_{\boldsymbol{x}'' \in \mathcal{U}} \|\alpha \boldsymbol{x} - \boldsymbol{x}''\| \\
&= \operatorname*{argmin}_{\alpha \boldsymbol{x}' \in \mathcal{U}} \|\alpha \boldsymbol{x} - \alpha \boldsymbol{x}'\| && \boldsymbol{x}' = \tfrac{1}{a} \boldsymbol{x}''. \\
&= \alpha \operatorname*{argmin}_{\boldsymbol{x}' \in \mathcal{U}} |\alpha| \|\boldsymbol{x} - \boldsymbol{x}'\| && \operatorname{argmin}_{\lambda x} f(x) = \lambda \operatorname{argmin}_x f(x). \\
&= \alpha \operatorname*{argmin}_{\boldsymbol{x}' \in \mathcal{U}} \|\boldsymbol{x} - \boldsymbol{x}'\| \\
&= \alpha \Pi_{\mathcal{U}}(\boldsymbol{x}).
\end{aligned}
$$

   And, we need to show additivity,

$$
\begin{aligned}
\Pi_{\mathcal{U}}(\boldsymbol{x} + \boldsymbol{y}) &= \operatorname*{argmin}_{\boldsymbol{z} \in \mathcal{U}} \|\boldsymbol{z} - (\boldsymbol{x} + \boldsymbol{y})\| \\
&= \operatorname*{argmin}_{\Pi_{\mathcal{U}}(\boldsymbol{x}) + \boldsymbol{y}' \in \mathcal{U}} \|\Pi_{\mathcal{U}}(\boldsymbol{x}) + \boldsymbol{y}' - \boldsymbol{x} - \boldsymbol{y}\| && \boldsymbol{z} \doteq \Pi_{\mathcal{U}}(\boldsymbol{x}) + \boldsymbol{y}' \text{ for some } \boldsymbol{y}' \in \mathcal{U}. \\
&= \Pi_{\mathcal{U}}(\boldsymbol{x}) + \operatorname*{argmin}_{\boldsymbol{y}' \in \mathcal{U}} \|(\boldsymbol{y}' - \boldsymbol{y}) + (\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x})\|^2 && \operatorname{argmin}_{x+\lambda} f(x) = \lambda + \operatorname{argmin}_x f(x). \\
&= \Pi_{\mathcal{U}}(\boldsymbol{x}) + \operatorname*{argmin}_{\boldsymbol{y}' \in \mathcal{U}} \|\boldsymbol{y}' - \boldsymbol{y}\|^2 + \|\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}\|^2 + 2\langle \Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y}' - \boldsymbol{y} \rangle \\
&= \Pi_{\mathcal{U}}(\boldsymbol{x}) + \operatorname*{argmin}_{\boldsymbol{y}' \in \mathcal{U}} \|\boldsymbol{y}' - \boldsymbol{y}\|^2 + 2 \underbrace{\langle \Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y}' \rangle}_{\langle \mathcal{U}^{\perp}, \mathcal{U} \rangle = 0} && \langle \Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y}' - \boldsymbol{y} \rangle = \langle \Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y}' \rangle - \boldsymbol{y}. \\
&= \Pi_{\mathcal{U}}(\boldsymbol{x}) + \Pi_{\mathcal{U}}(\boldsymbol{y}).
\end{aligned}
$$

■

So, we know that $\Pi_{\mathcal{U}}$ is an orthogonal projection. Now, we want to find the matrix P representing that linear transformation.

> Given an orthonormal basis U of $\mathcal{U}$, we can compute the optimal projection matrix,
> $$P = UU^\top.$$
> Note that in this case W and V share parameters, and $UU^\top$ is the optimal weight matrix if we enforce parameter sharing via $V = W^\top$.

Proof. The image of the projection matrix is $\mathcal{U}$,
$$Px = \left(\sum_{i=1}^{m} u_i u_i^\top \right) x = \sum_{i=1}^{m} u_i u_i^\top x = \sum_{i=1}^{m} \langle u_i, x \rangle u_i.$$

Self-adjointness,
$$P^\top = (UU^\top)^\top = UU^\top = P.$$

Idempotency,
$$PP = U \underbrace{U^\top U}_{I_m} U^\top = UU^\top = P.$$

Orthogonality, TODO   ■

> In general, we do not have an orthonormal basis for $\mathcal{U}$. In a non-orthonormal basis V for $\mathcal{U}$, we can recover the projection matrix,
> $$P = VV^+, \quad V^+ \doteq \left(V^\top V \right)^{-1} V^\top.$$

Note that V^+ is the left Moore-Penrose pseudo-inverse of V.

Proof. P is the projection matrix of $\mathcal{U}$,
$$PV = VV^+V = V \left(V^\top V \right)^{-1} V^\top V = V.$$

Together with the rank constraint, this yields $Pu^\perp = 0$ for all $u^\perp \in \mathcal{U}^\perp$.

Self-adjointness,
$$P^\top = \left(V \left(V^\top V \right)^{-1} V^\top \right)^\top = V \left(V^\top V \right)^{-1} V^\top = P.$$

Idempotency,
$$PP = V \left(V^\top V \right)^{-1} V^\top V \left(V^\top V \right)^{-1} V^\top = VV^+ = P.$$

■

Finding the optimal subspace. Now we need to find out which subspace of dimension m or less is optimal to project onto. First, we need to rewrite the objective function to find a new interpretation,

$$
\begin{aligned}
\mathcal{R}(P) &= \frac{1}{2}\mathbb{E}\left[\|x - Px\|^2\right]\\
&= \frac{1}{2}\mathbb{E}[\langle x, x\rangle + 2\langle x, -Px\rangle + \langle -Px, -Px\rangle]\\
&= \frac{1}{2}\mathbb{E}\left[\|x\|^2\right] + \frac{1}{2}\mathbb{E}\left[\|Px\|^2\right] - \mathbb{E}[\langle x, Px\rangle]\\
&= \frac{1}{2}\mathbb{E}\left[\|x\|^2\right] + \frac{1}{2}\mathbb{E}\left[\|Px\|^2\right] - \mathbb{E}[\langle x, P^2x\rangle]\\
&= \frac{1}{2}\mathbb{E}\left[\|x\|^2\right] + \frac{1}{2}\mathbb{E}\left[\|Px\|^2\right] - \mathbb{E}\left[\|Px\|^2\right]\\
&= \frac{1}{2}\mathbb{E}\left[\|x\|^2\right] - \frac{1}{2}\mathbb{E}\left[\|Px\|^2\right].
\end{aligned}
$$

Idempotency of projection matrices.

$\langle x, P^2x\rangle = \langle Px, Px\rangle.$

Because our data is centered, we know the following,

$$
\begin{aligned}
\mathrm{Var}[x] &= \mathbb{E}\left[\|x\|^2\right] - \|\underbrace{\mathbb{E}[x]}_{=0}\|^2\\
&= \mathbb{E}\left[\|x\|^2\right].\\
\mathrm{Var}[Px] &= \mathbb{E}\left[\|Px\|^2\right] - \|\mathbb{E}[Px]\|^2\\
&= \mathbb{E}\left[\|Px\|^2\right] - \|P\underbrace{\mathbb{E}[x]}_{=0}\|^2\\
&= \mathbb{E}\left[\|Px\|^2\right].
\end{aligned}
$$

$$
\mathcal{R}(P) = \frac{1}{2}(\mathrm{Var}[x] - \mathrm{Var}[Px]) \propto -\frac{1}{2}\mathrm{Var}[Px].
$$

Hence, minimizing $\mathcal{R}(P)$ is equivalent to maximizing the variance $\mathrm{Var}[Px]$.

We can further simplify this expression to find a sufficient statistic for the objective function,

$$
\begin{aligned}
-\frac{1}{2}\mathrm{Var}[Px] &= -\frac{1}{2}\mathbb{E}\left[\|Px\|^2\right]\\
&= -\frac{1}{2}\mathbb{E}[\langle x, Px\rangle]\\
&= -\frac{1}{2}\mathbb{E}\left[\mathrm{tr}\left(x^\top Px\right)\right]\\
&= -\frac{1}{2}\mathrm{tr}\left(\mathbb{E}\left[Pxx^\top\right]\right)\\
&= -\frac{1}{2}\mathrm{tr}\left(P\mathbb{E}\left[xx^\top\right]\right).
\end{aligned}
$$

$\|Px\|^2 = \langle Px, Px\rangle = \langle x, P^2x\rangle = \langle x, Px\rangle.$

Cyclic property of trace.

$\mathbb{E}[xx^\top]$ is a sufficient statistic for $\mathcal{R}(P)$.

The optimal projection is fully determined by the covariance matrix $\mathbb{E}[xx^\top]$, together with $\mathbb{E}[x]$ for centering.

1.3 Principal component analysis

Theorem 3 (Spectral theorem)**.** Any symmetric and positive semidefinite matrix Σ can be non-negatively diagonalized with an orthogonal matrix,

$$\Sigma = Q\Lambda Q^\top, \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n),$$

where $\lambda \geq \cdots \geq \lambda_n \geq 0$ and Q is orthogonal.

Remark. Q is composed of ordered eigenvectors of Σ, and Λ is composed of ordered eigenvalues of Σ.

Theorem 4 (PCA theorem)**.** The variance maximizing projection matrix P for a covariance matrix $\mathbb{E}[xx^\top] = Q\Lambda Q^\top$ as in the spectral theorem is given by

$$P = UU^\top, \quad U = Q \begin{bmatrix} I_m \\ \mathbf{0} \end{bmatrix}.$$

Proof.

$$
\begin{aligned}
\mathrm{Var}[Px] &= \mathrm{tr}\Big(P\mathbb{E}[xx^\top]\Big) \\
&= \mathrm{tr}\Big(UU^\top Q\Lambda Q^\top\Big) && P = UU^\top, \mathbb{E}[xx^\top] = Q\Lambda Q^\top. \\
&= \mathrm{tr}\Big(\big(Q^\top U\big)\big(Q^\top U\big)^\top \Lambda\Big). && \text{Cyclic property.}
\end{aligned}
$$

This term is maximized by $Q^\top U = \begin{bmatrix} I_m & \mathbf{0} \end{bmatrix}^\top$. ∎

1.4 Learning algorithms

Eigenvalue decomposition of the (symmetric) sample covariance matrix has $\mathcal{O}(n^3)$ complexity. Furthermore, the complexity of computing $\mathbb{E}[xx^\top]$ is $\mathcal{O}(Nn^2)$.[3] This is quite costly, thus we need to search for algorithms that have lower runtime complexity.

[3] Typically, $N \gg n$.

Power method. The power method is a recursive algorithm for computing principal eigenvectors. It initializes a vector at random $v^{(0)} \sim \mathcal{N}(\mathbf{0}, I)$. Then, it iteratively improves this guess,

$$v^{(t+1)} = \frac{Av^{(t)}}{\|Av^{(t)}\|}.$$

The computational complexity of this algorithm is $\mathcal{O}(Tn^2)$.

> **Lemma 5.** Let u_1 be the unique principal eigenvector of a diagonalizable matrix A with eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. If $\langle v_0, u_1 \rangle \neq 0$, then
> $$\lim_{t \to \infty} v^{(t)} = u_1.$$

Proof. We can decompose vectors as a linear combination of eigenvectors, $v^{(0)} = \sum_{i=1}^{n} \alpha_i u_i$. Then,

$$v^{(k)} \propto A^k v^{(0)}$$

$$= \sum_{i=1}^{n} \alpha_i \lambda_i^k u_i \qquad\qquad A^k v^{(0)} = \sum_{i=1}^{n} \alpha_i A^k u_i.$$

$$\propto \alpha_1 u_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^k u_i. \qquad\qquad \text{Divide by } \lambda_1.$$

$\lambda_i / \lambda_1 < 1$ for $i > 1$, thus the sum goes to 0 and $v^{(k)} \to u_1$. ∎

We can use this algorithm to also compute the next principal eigenvectors by factoring out u_1 and then doing the algorithm again to recover u_2, and continue doing that until we have the m principal eigenvectors.

Thus, the total complexity of finding the m principal eigenvectors is $\mathcal{O}(Tmn^2)$. However, this does not get rid of the $\mathcal{O}(Nn^2)$ complexity for computing the sample covariance matrix.

Gradient descent. By treating the autoencoder as a neural network, we can use deep learning techniques, such as gradient descent. Gradient descent iteratively updates the weights by

$$P^{(t+1)} = P^{(t)} - \eta \nabla_P \mathcal{R}(P).$$

The gradient is computed by $(P - I)xx^\top$. The problem with this is that we cannot constrain P to be a projection. Thus, we actually need to update V and W. Thus, by the chain rule for matrix derivatives,

$$\nabla_W \mathcal{R}(W, V) = (P - I)xx^\top W^\top$$
$$\nabla_V \mathcal{R}(W, V) = V^\top (P - I)xx^\top.$$

The complexity for T iterations is then $\mathcal{O}(T(m + k)n^2)$, where k is the batch size.

1.5 Non-linear autoencoders

We can get much better performance by considering non-linearities, and we can easily use gradient descent to work with non-linear architectures.