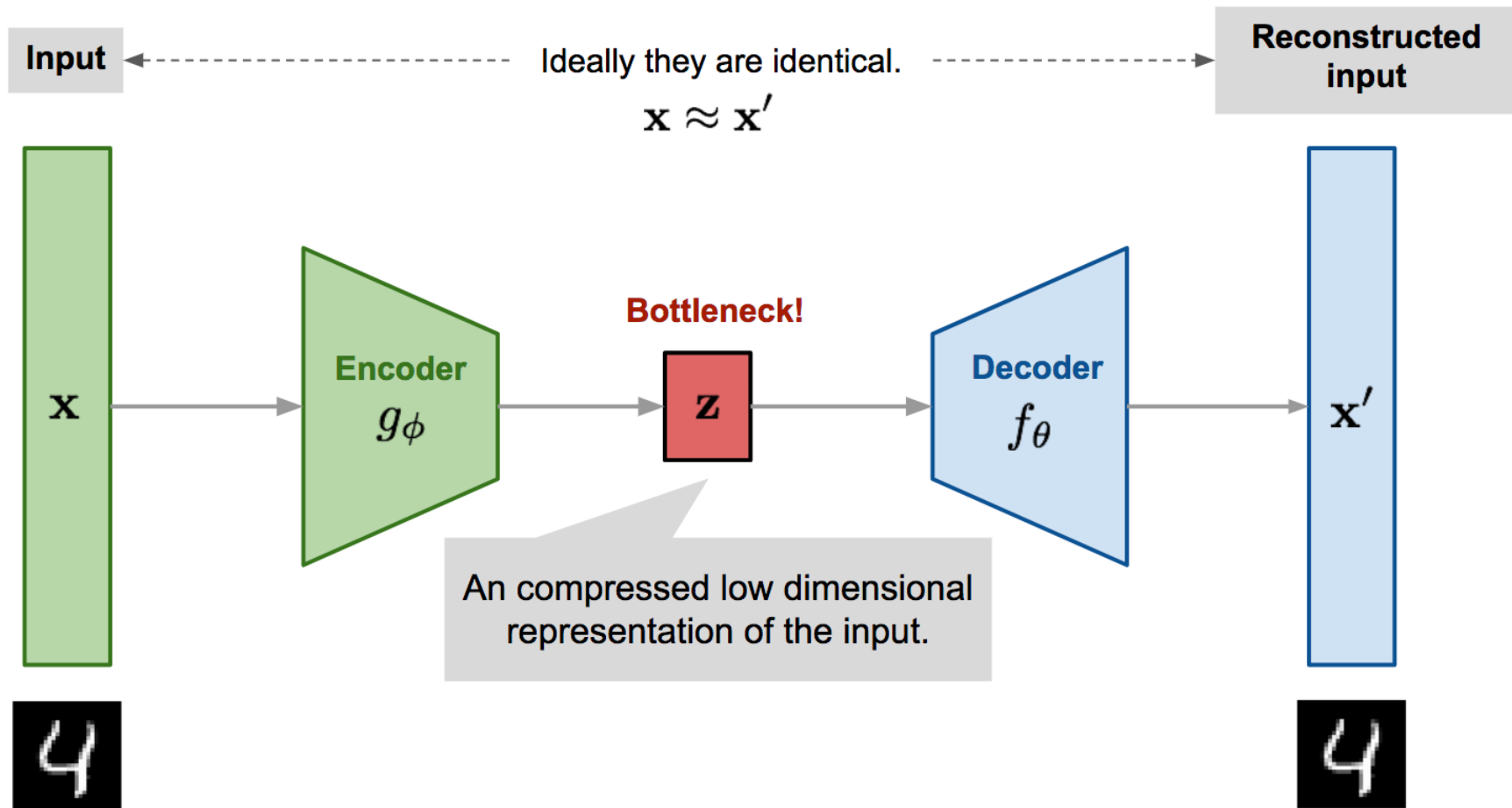


# Autoencoder



# Autoencoder



# Autoencoder

Autoencoder 是一種用於非監督學習的神經網絡模型，主要用於數據的降維或特徵學習。它由編碼器 (Encoder) 和解碼器 (Decoder) 組成。以下是有關 Autoencoder 的詳細說明和相關方程。

## 1. Autoencoder 結構

Autoencoder 的目標是學習一個函數  $f(x)$ ，使得輸入  $x$  能夠被重建為輸出  $\hat{x}$ 。具體的數學表示為：

$$\hat{x} = g(f(x))$$

其中：

- 編碼器 (Encoder) 將輸入  $x$  映射到一個潛在表示  $z$ ：

$$z = f(x) = \sigma(W_e x + b_e)$$

其中， $W_e$  是編碼器的權重矩陣， $b_e$  是編碼器的偏置， $\sigma$  是激活函數。

# Autoencoder

- 解碼器 (Decoder) 從潛在表示  $z$  重建輸出  $\hat{x}$  :

$$\hat{x} = g(z) = \sigma(W_d z + b_d)$$

其中， $W_d$  是解碼器的權重矩陣， $b_d$  是解碼器的偏置。

## 2. 損失函數

Autoencoder 的目標是使輸入與重建輸出的差異最小化。通常使用均方誤差 (Mean Squared Error, MSE) 作為損失函數： $L(x, \hat{x}) = \|x - \hat{x}\|^2 = \|x - g(f(x))\|^2$

潛在表示 ( Latent Representation ) 是 Autoencoder 中關鍵的部分，指的是數據在經過編碼器壓縮後所形成的低維表示。這個表示保留了輸入數據的核心特徵，並且通過解碼器能夠重建出原始數據。潛在表示通常比原始數據的維度要低，從而實現了數據的降維。

# Autoencoder

## 3. 潛在空間

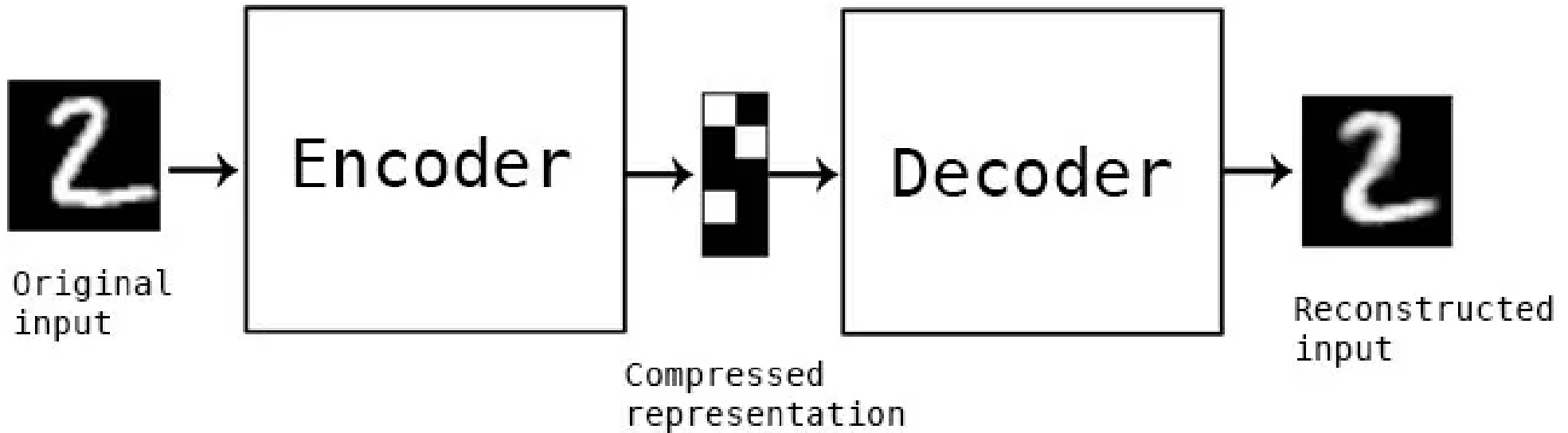
Autoencoder 的核心概念是潛在空間 (Latent Space)，即隱含表示  $z$ ，這個空間中的每個向量是輸入數據的壓縮表示。通過學習有意義的潛在表示，Autoencoder 可以在低維空間中捕捉輸入數據的核心特徵。

## 4. 應用

Autoencoders 被廣泛應用於以下領域：

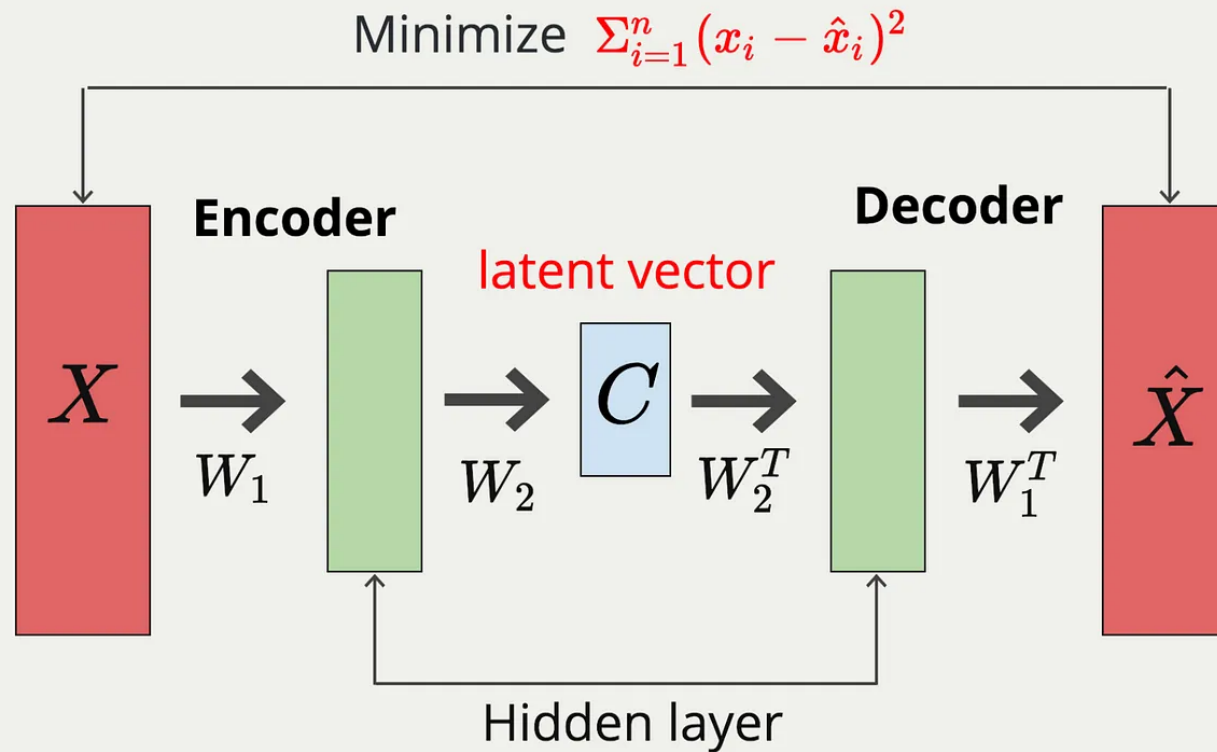
- **降維**：用於數據壓縮和可視化，如 PCA 的非線性替代方案。
- **去噪**：去噪自編碼器 (Denoising Autoencoder) 能夠從有噪聲的數據中重建無噪聲的數據。
- **生成模型**：變分自編碼器 (Variational Autoencoder, VAE) 可以用來生成新數據樣本。

# Autoencoder

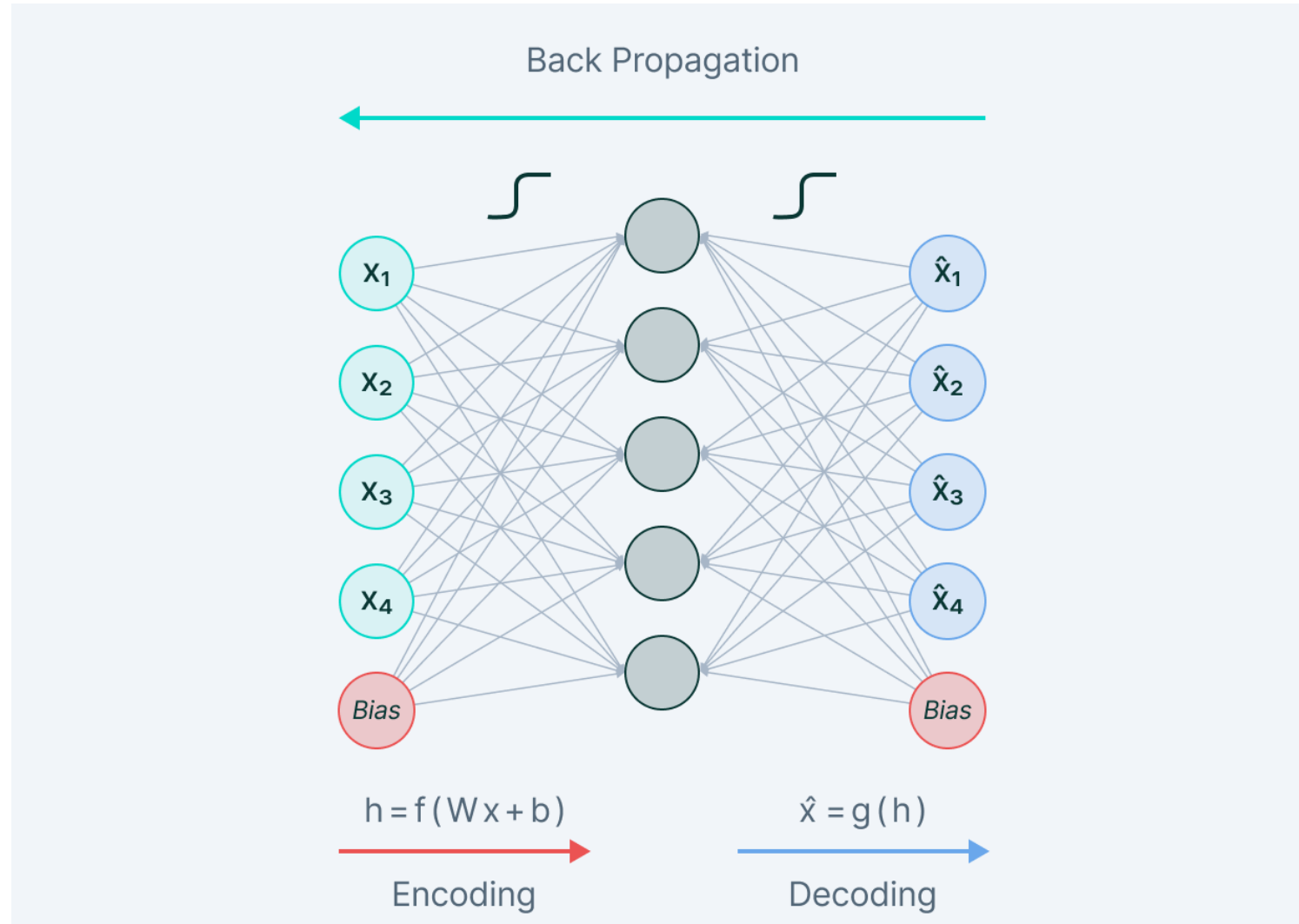


# Autoencoder

- Loss :  $\sum_{i=1}^n (x_i - \hat{x}_i)^2$

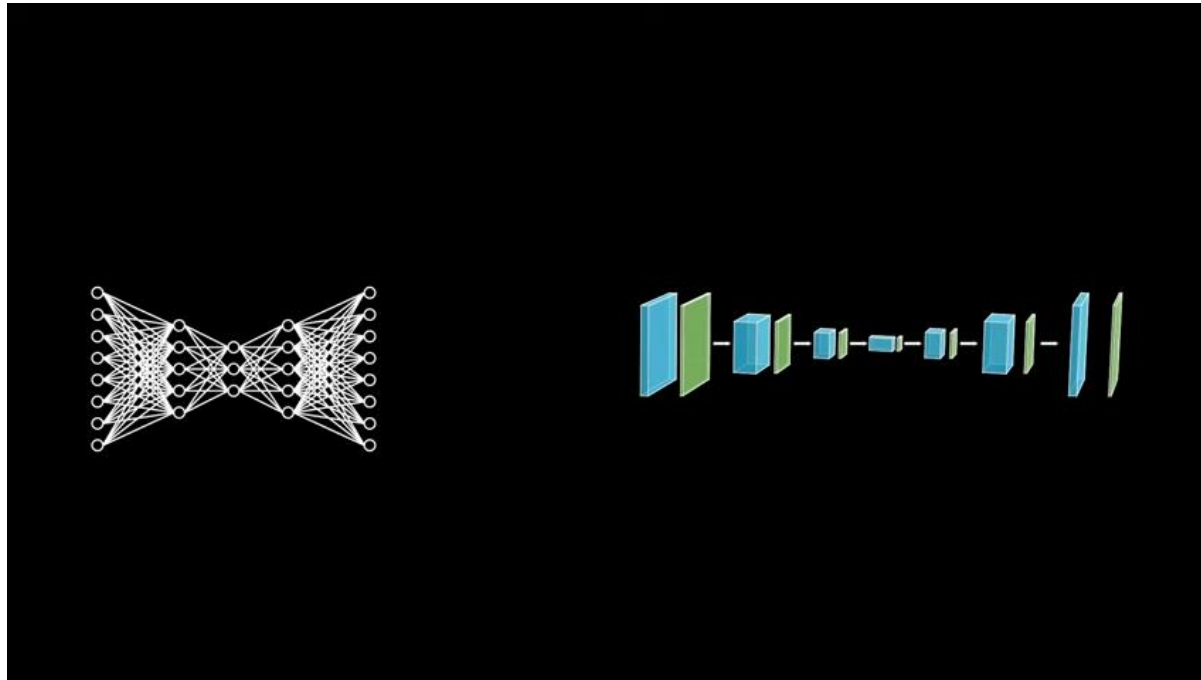


# Autoencoder



# Autoencoder

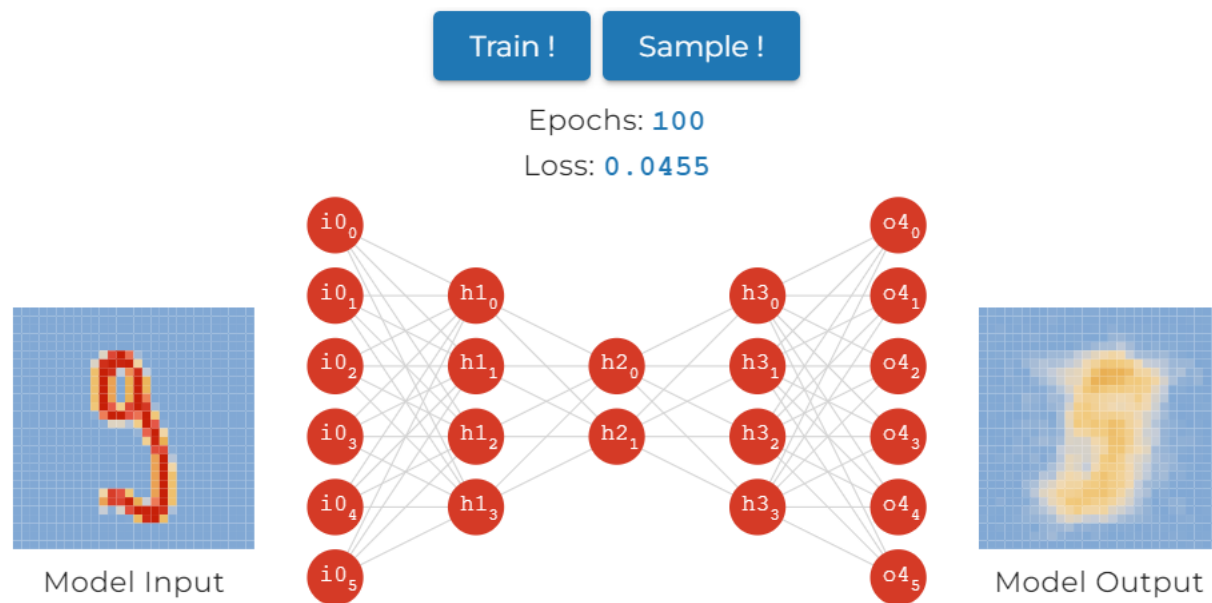
 11:40



# Latent Space



# Sampling



# Backpropagation

在自編碼器 ( Autoencoder ) 中，反向傳播 ( Backpropagation ) 是一種優化權重的技術，使得自編碼器能夠最小化輸入和重建輸出之間的差異。以下是自編碼器中反向傳播的過程，並輔以數學公式來說明。

## 自編碼器架構簡介

自編碼器主要包含兩部分：

1. **編碼器 ( Encoder )**：將輸入數據  $x$  壓縮到隱含層表示  $h$ ，通常維度比原始輸入小。

$$h = f(x) = \sigma(W^{(1)}x + b^{(1)})$$

其中， $W^{(1)}$  和  $b^{(1)}$  是編碼器的權重和偏置。

2. **解碼器 ( Decoder )**：將隱含層表示  $h$  轉換回重建的輸出  $\hat{x}$ ，以接近於原始輸入。

$$\hat{x} = g(h) = \sigma(W^{(2)}h + b^{(2)})$$

其中， $W^{(2)}$  和  $b^{(2)}$  是解碼器的權重和偏置。

# Backpropagation

## 2. 前向傳播：

- **編碼器**：將輸入  $x$  投影到隱含表示  $h$ 。

$$h = f(x) = \sigma(W^{(1)}x + b^{(1)})$$

- **解碼器**：將隱含表示  $h$  解碼為重建輸出  $\hat{x}$ 。

$$\hat{x} = g(h) = \sigma(W^{(2)}h + b^{(2)})$$

## 3. 反向傳播：計算梯度：使用鏈式法則分別計算損失對解碼器和編碼器的權重和偏置的梯度。

- 對於解碼器權重  $W^{(2)}$  和偏置  $b^{(2)}$  的梯度：

$$\frac{\partial L}{\partial W^{(2)}} = \delta^{(2)} h^T$$

$$\frac{\partial L}{\partial b^{(2)}} = \delta^{(2)}$$

其中， $\delta^{(2)} = (\hat{x} - x) \circ \sigma'(W^{(2)}h + b^{(2)})$ 。

# Backpropagation

- 對於編碼器權重  $W^{(1)}$  和偏置  $b^{(1)}$  的梯度：

$$\frac{\partial L}{\partial W^{(1)}} = \delta^{(1)} x^T$$

$$\frac{\partial L}{\partial b^{(1)}} = \delta^{(1)}$$

其中， $\delta^{(1)} = ((W^{(2)})^T \delta^{(2)}) \circ \sigma'(W^{(1)}x + b^{(1)})$ 。

4. 更新權重和偏置：使用梯度下降法更新權重和偏置，學習率為  $\eta$ ：

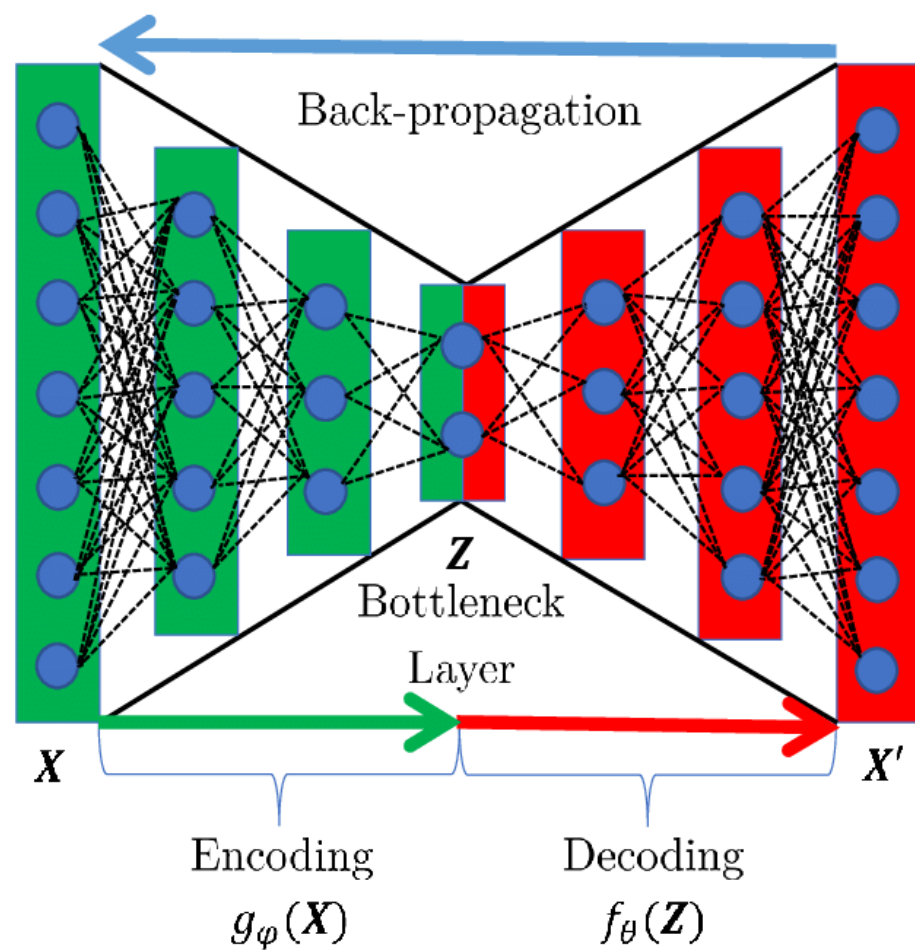
$$W^{(1)} \leftarrow W^{(1)} - \eta \frac{\partial L}{\partial W^{(1)}}$$

$$b^{(1)} \leftarrow b^{(1)} - \eta \frac{\partial L}{\partial b^{(1)}}$$

$$W^{(2)} \leftarrow W^{(2)} - \eta \frac{\partial L}{\partial W^{(2)}}$$

$$b^{(2)} \leftarrow b^{(2)} - \eta \frac{\partial L}{\partial b^{(2)}}$$

# Backpropagation



# Video Autoencoder

視訊自編碼器 ( Video Autoencoder ) 是一種自編碼器的擴展，主要用於視訊數據的壓縮與重建。視訊數據包含時間序列，因此視訊自編碼器通常結合時序模型來處理空間和時間上的特徵。以下是視訊自編碼器的結構和反向傳播的數學公式。

## 視訊自編碼器架構

視訊自編碼器一般包含以下部分：

1. **編碼器 ( Encoder )**：將輸入視訊序列  $X = \{x_1, x_2, \dots, x_T\}$  映射到低維的隱含表示  $H = \{h_1, h_2, \dots, h_T\}$ 。
2. **解碼器 ( Decoder )**：將隱含表示  $H$  解碼回重建的視訊序列  $\hat{X} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_T\}$ ，期望重建的序列接近於原始輸入序列。

視訊自編碼器的編碼和解碼器可以是基於卷積神經網絡 ( CNN ) 處理空間特徵，再結合循環神經網絡 ( RNN ) 如LSTM來捕捉時間序列特徵。

# Video Autoencoder

## 視訊自編碼器的數學表示

假設一段視訊的每個幀為  $x_t$ ，且  $t \in \{1, 2, \dots, T\}$ ：

1. **編碼器**：將每個輸入幀  $x_t$  經過卷積操作得到特徵，並用LSTM等模型生成隱含表示：

$$h_t = f(x_t, h_{t-1}) = \sigma(W^{(1)}x_t + \underline{U^{(1)}h_{t-1}} + b^{(1)})$$

其中， $W^{(1)}$  和  $U^{(1)}$  是權重矩陣， $b^{(1)}$  是偏置， $\sigma$  表示非線性激活函數。

2. **解碼器**：將隱含表示  $h_t$  解碼回重建的視訊幀  $\hat{x}_t$ ：

$$\hat{x}_t = g(h_t, \hat{x}_{t-1}) = \sigma(W^{(2)}h_t + \underline{U^{(2)}\hat{x}_{t-1}} + b^{(2)})$$

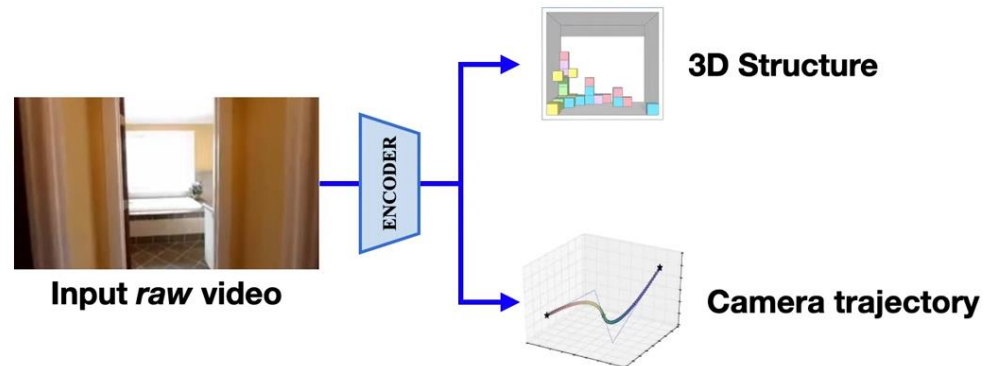
其中， $W^{(2)}$  和  $U^{(2)}$  是解碼器的權重矩陣， $b^{(2)}$  是偏置項。

# Video Autoencoder



## Objective

In this work, we learn to separate **3D structure** from **Camera Motion** without any human annotations



Variational Autoencoder

