EURO^2

Fundamental Concepts of Generative Machine Learning

Erdem Akagündüz, PhD
Graduate School of Informatics, METU, Türkiye

ncc@ulakbim.gov.tr

Lesson 3: Auto-Encoding

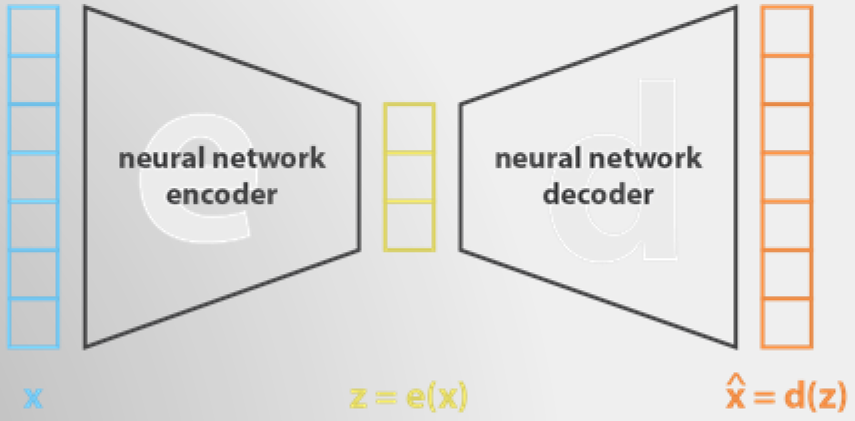
Welcome to **Part III: “Auto-Encoding”**

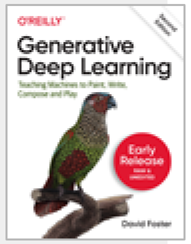
This part includes three subsections:

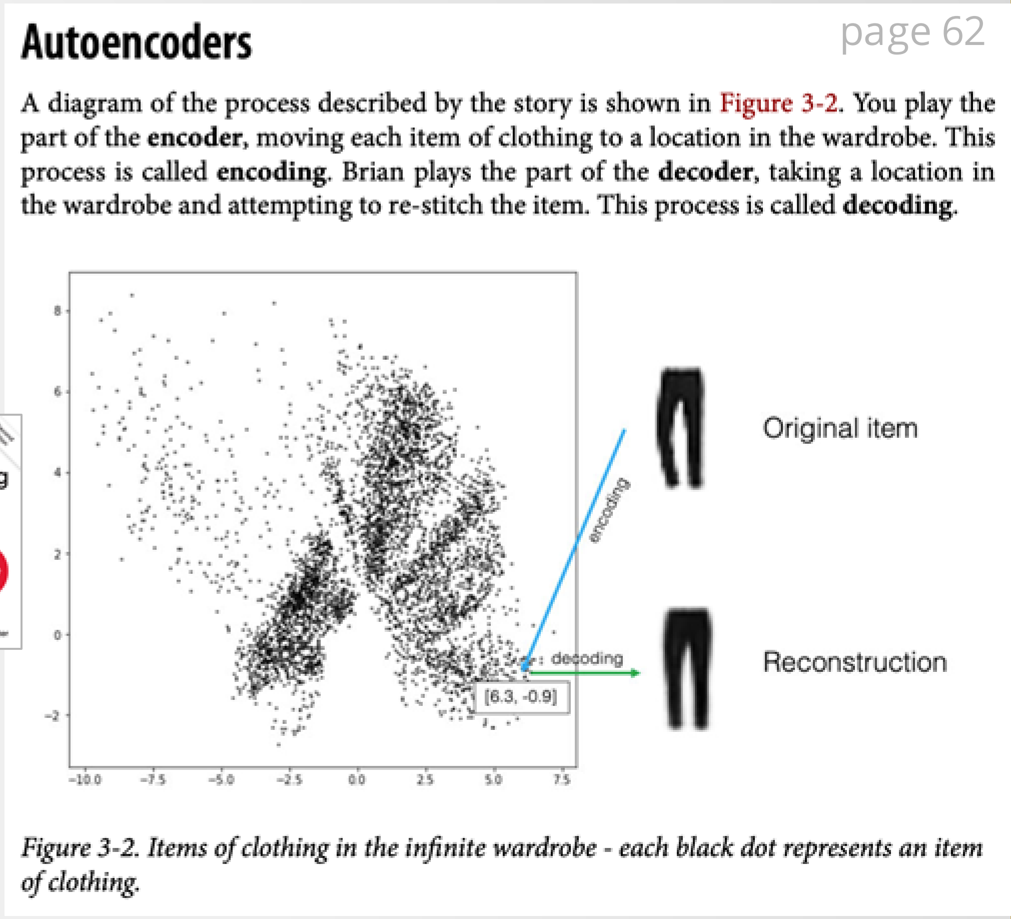
- **Autoencoders and Dimensionality Reduction**
- Variational Inference and VAEs
- Conclusions

Auto-Encoders

- Autoencoders comprise an encoder network that maps input data to a latent representation and a decoder network that reconstructs the input from the latent space.







Autoencoders page 62

A diagram of the process described by the story is shown in Figure 3-2. You play the part of the **encoder**, moving each item of clothing to a location in the wardrobe. This process is called **encoding**. Brian plays the part of the **decoder**, taking a location in the wardrobe and attempting to re-stitch the item. This process is called **decoding**.

Original item
Reconstruction
encoding
decoding
[6.3, -0.9]

Figure 3-2. Items of clothing in the infinite wardrobe - each black dot represents an item of clothing.

Auto-Encoders

- **Encoder**: A module that compresses the train-validate-test set input data into an encoded representation that is typically several orders of magnitude smaller than the input data.
- **Bottleneck**: A module that contains the compressed knowledge representations and is therefore the most important part of the network.
- **Decoder**: A module that helps the network“decompress” the knowledge representations and reconstructs the data back from its encoded form. The output is then compared with a ground truth.

Autoencoders

page 62

A diagram of the process described by the story is shown in Figure 3-2. You play the part of the **encoder**, moving each item of clothing to a location in the wardrobe. This process is called **encoding**. Brian plays the part of the **decoder**, taking a location in the wardrobe and attempting to re-stitch the item. This process is called **decoding**.

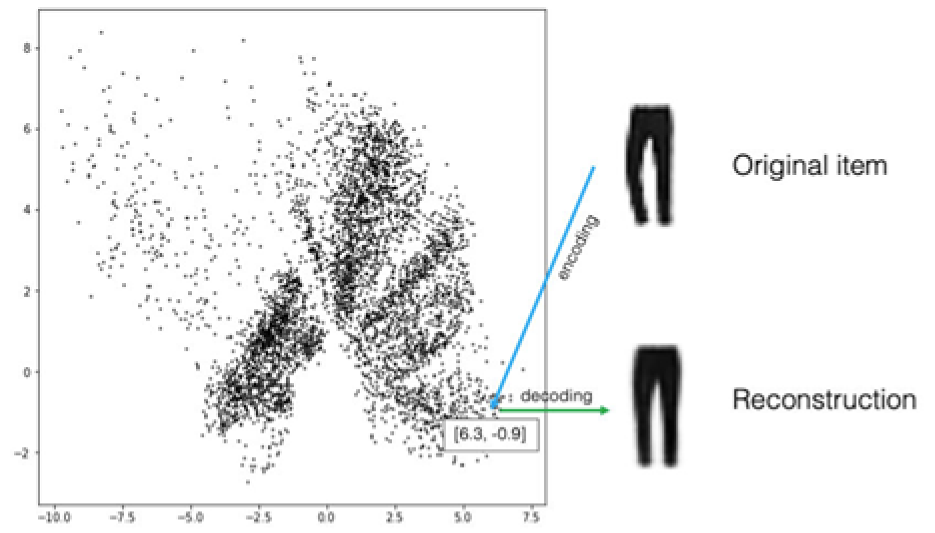
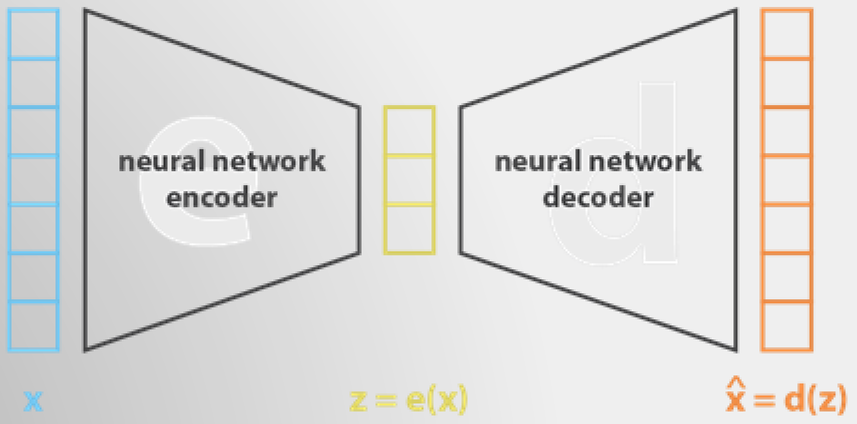
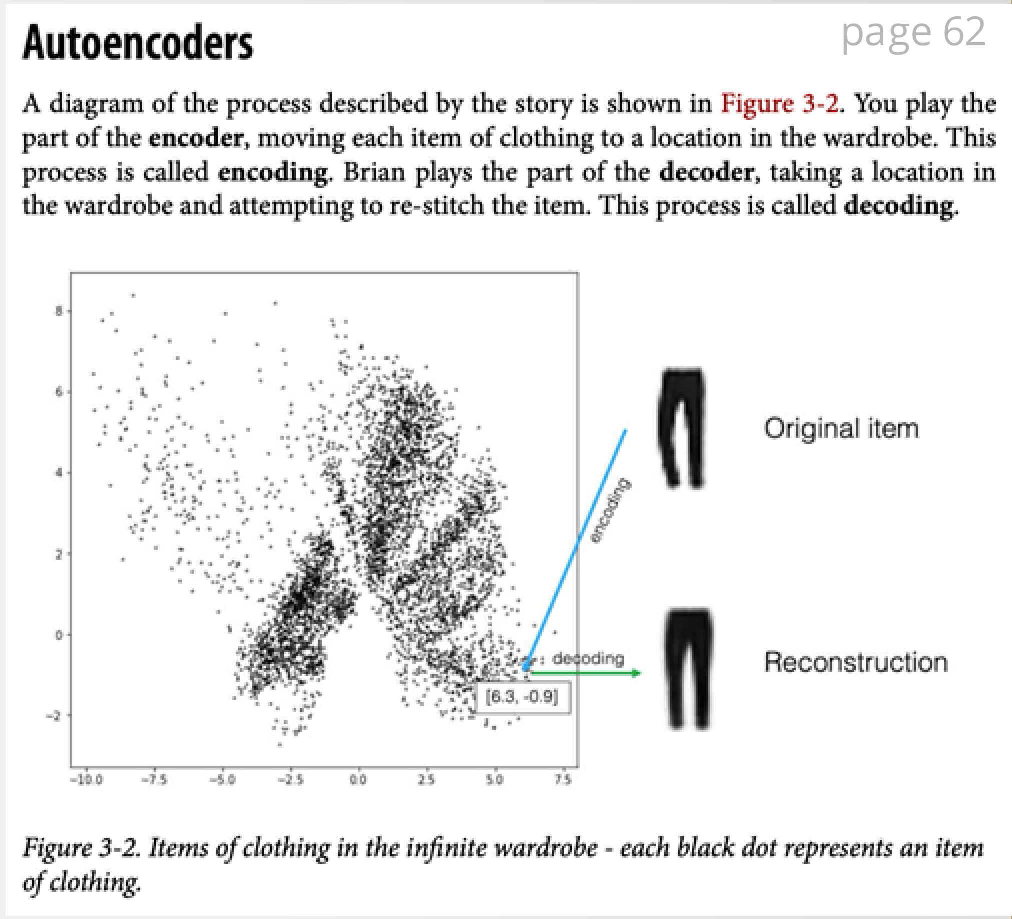


Figure 3-2. Items of clothing in the infinite wardrobe - each black dot represents an item of clothing.

Auto-Encoders

- The bottleneck layer in autoencoders serves as the latent space, capturing a compressed representation of the input data.





Autoencoders page 62

A diagram of the process described by the story is shown in Figure 3-2. You play the part of the **encoder**, moving each item of clothing to a location in the wardrobe. This process is called **encoding**. Brian plays the part of the **decoder**, taking a location in the wardrobe and attempting to re-stitch the item. This process is called **decoding**.

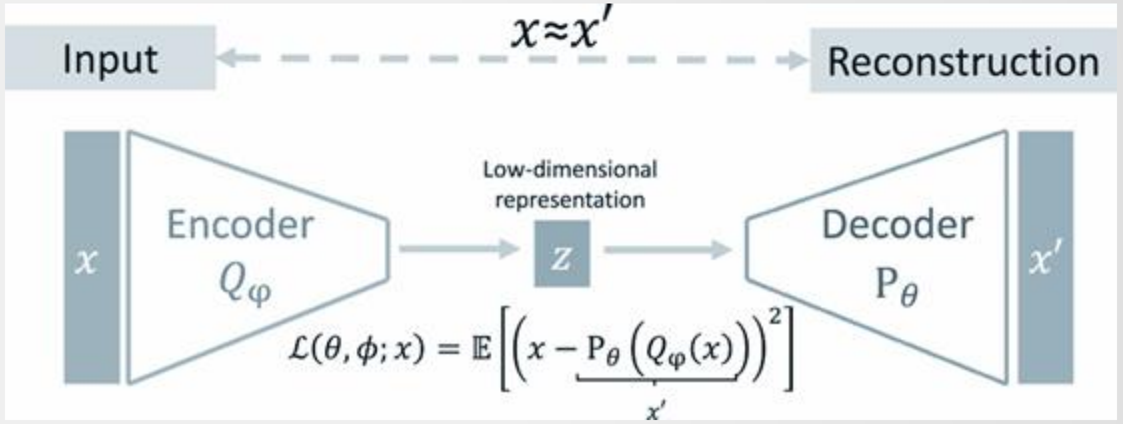
encoding
decoding
[6.3, -0.9]
Original item
Reconstruction

Figure 3-2. Items of clothing in the infinite wardrobe - each black dot represents an item of clothing.

Training Auto-Encoders

There are 4 fundamental hyperparameters for training an autoencoder:

- **Code/Latent Space Size**: The bottleneck size decides how much the data has to be compressed. This can also act as a regularisation term.
- **Number of layers/nodes**: While a higher depth/number of nodes increases model capacity/complexity, a lower depth is faster to process and avoid overfitting.
- **Reconstruction Loss**: This is the most important one. It defines the character of the AutoEncoder.



$$\mathcal{L}(\theta,\phi;x) = \mathbb{E}\left[\left(x - \underbrace{P_\theta\left(Q_\varphi(x)\right)}_{x'}\right)^2\right]$$

Auto-Encoder Loss function

- In an autoencoder, the loss function plays a crucial role in training the model and optimizing the reconstruction process.
- The choice of the loss function depends on the specific objectives and characteristics of the autoencoder.
- There can be many features that shape the loss character of an AE
  - Reconstruction (main)
  - Regularization (main)
  - Characterized loss (optional): Sparsity, Variational loss, other problem dependent

Reconstruction Loss

- The primary objective of an autoencoder is to reconstruct the input data from the compressed latent space.
- The reconstruction loss measures the discrepancy between the original input and the reconstructed output.
- Commonly used reconstruction loss functions depend on the input data and include:
  - mean squared error (continuous data), binary cross-entropy (binary data), Kullback-Leibler Divergence (distributions), Categorical Cross-Entropy (multiple classes), etc.

Regularization Loss

- Autoencoders can be prone to overfitting, especially when the model capacity is high or the dataset is limited.
- Regularization techniques, such as L1 or L2 regularization, drop-out etc, can be incorporated into the loss function to discourage complex or redundant representations.
- Regularization loss plays a vital role in controlling the behavior of an autoencoder and ensuring that the learned representations possess desirable properties.
- The specific choice and application of regularization techniques depend on the characteristics of the data, the desired properties of the representations, and the overall training objectives.

Characterized Loss

- A “”characterized loss” component can be introduced to go beyond traditional objectives like reconstruction or regularization, addressing specific requirements or goals of the task at hand.
- Such as:
  - Sparsity Loss: Only a small subset of the latent variables or activations should be active or non-zero.
    - $\lambda * \sum |z|$
  - Diversity Loss: Generation of diverse outputs by penalizing models that generate repetitive or similar samples.
    - $(1/(n*(n-1)))*\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} d(x_i, x_j)$
  - Consistency Loss: encourages the model to produce consistent predictions across augmented versions of the same sample, improving generalization.

Characterized Loss: Clustering

IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, VOL. 16, NO. 11, NOVEMBER 2019 1693

Unsupervised Clustering of Seismic Signals Using Deep Convolutional Autoencoders

S. Mostafa Mousavi, Weiqiang Zhu, William Ellsworth, and Gregory Beroza

- A clustering layer, connected to the AE’s bottleneck, assigns the hidden features of each sample to a cluster.
- The clustering loss component L_c is specifically designed to enhance the clustering behavior of the learned latent representations.
- The input is data is seismic waveforms, that reach a seismic station.

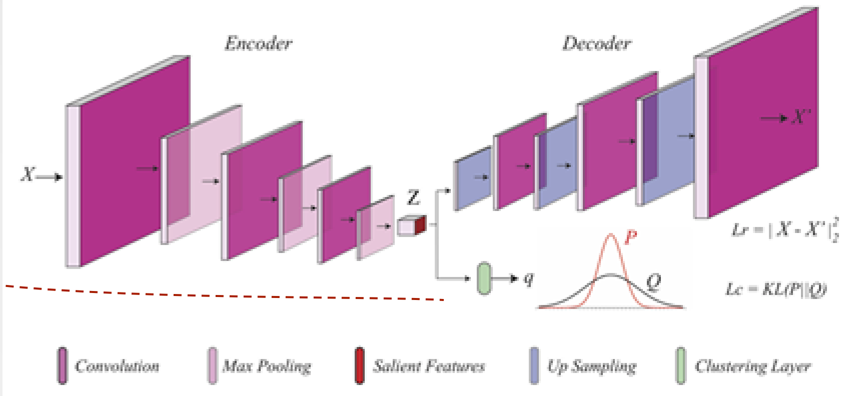
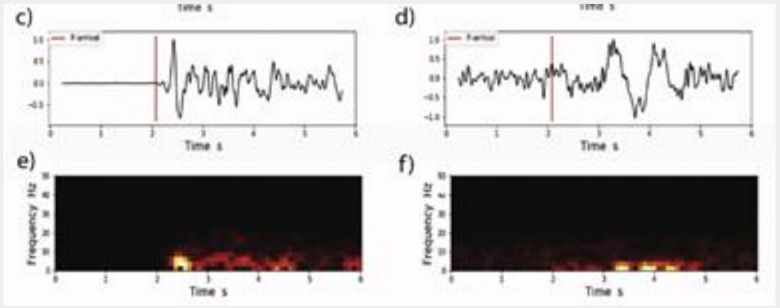


Fig. 1. Network architecture. The encoder and decoder are composed of fully convolutional layers followed by max-pooling and up-sampling layers, respectively. Reconstruction loss (L_r) and the clustering loss (L_c) are given in the figure.



$$L = (1-\lambda)L_r + \lambda L_c \qquad (1)$$

$$q_{ij} = \frac{(1+\| z_i - \mu_j \|^2)^{-1}}{\sum_j (1+\| z_i - \mu_j \|^2)^{-1}}. \qquad (2)$$

The membership probabilities are used to compute an auxiliary target distribution, p_{ij}

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j \left(q_{ij}^2 / \sum_i q_{ij}\right)} \qquad (3)$$

and clustering is performed by minimizing the Kullback–Leibler (KL) divergence between the soft assignments, q_{ij}, and the target distribution, p_{ij}

$$L_c = KL(P \parallel Q) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right). \qquad (4)$$

Characterized Loss: Clustering

- The clustering is done in two steps.
  - In the initial step, they train by setting λ=0 (i.e. only reconstruction)
  - In the next step, the learned features are used to initialize the cluster centroids (μ_j) in the feature space using k -means. Following, cluster assignment and feature learning are jointly performed (λ=0.1)

Or, just for fun (and for a good project) regions can be manually labeled for **regions**, and the second step could be supervised! That would be a mix of unsupervised and supervised learning!

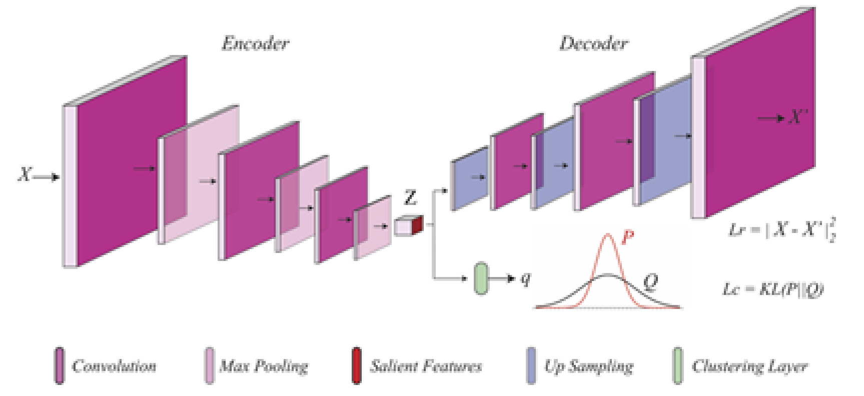


Fig. 1. Network architecture. The encoder and decoder are composed of fully convolutional layers followed by max-pooling and up-sampling layers, respectively. Reconstruction loss (L_r) and the clustering loss (L_c) are given in the figure.

$$L = (1-\lambda)L_r + \lambda L_c \tag{1}$$

$$q_{ij} = \frac{(1+\| z_i - \mu_j \|^2)^{-1}}{\sum_j (1+\| z_i - \mu_j \|^2)^{-1}}. \tag{2}$$

The membership probabilities are used to compute an auxiliary target distribution, p_{ij}

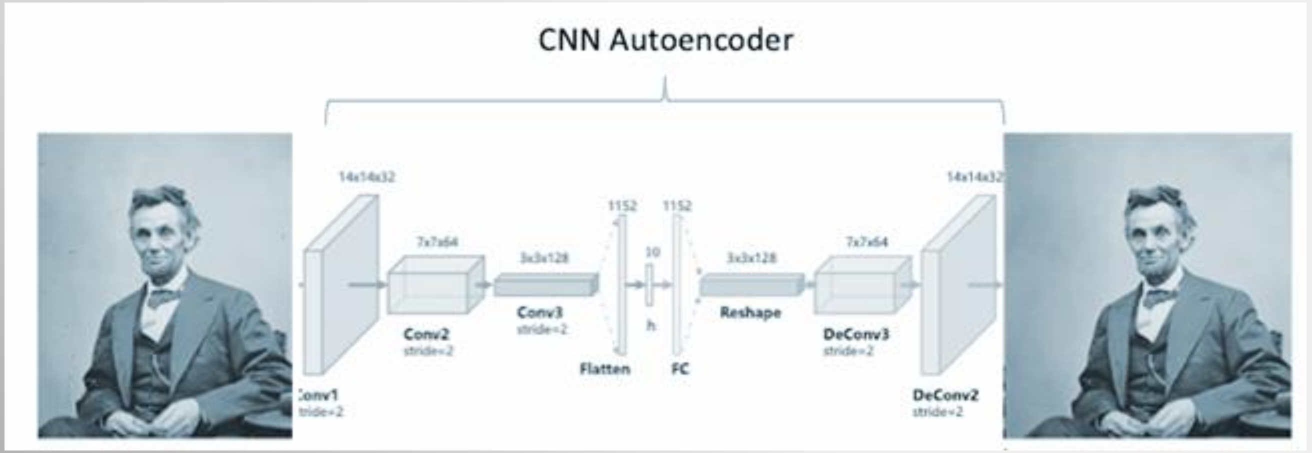
$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j \left(q_{ij}^2 / \sum_i q_{ij}\right)} \tag{3}$$

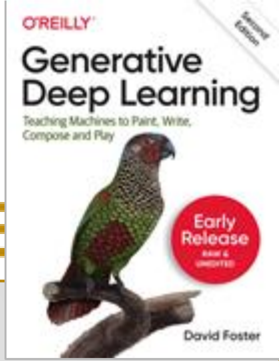
and clustering is performed by minimizing the Kullback–Leibler (KL) divergence between the soft assignments, q_{ij}, and the target distribution, p_{ij}

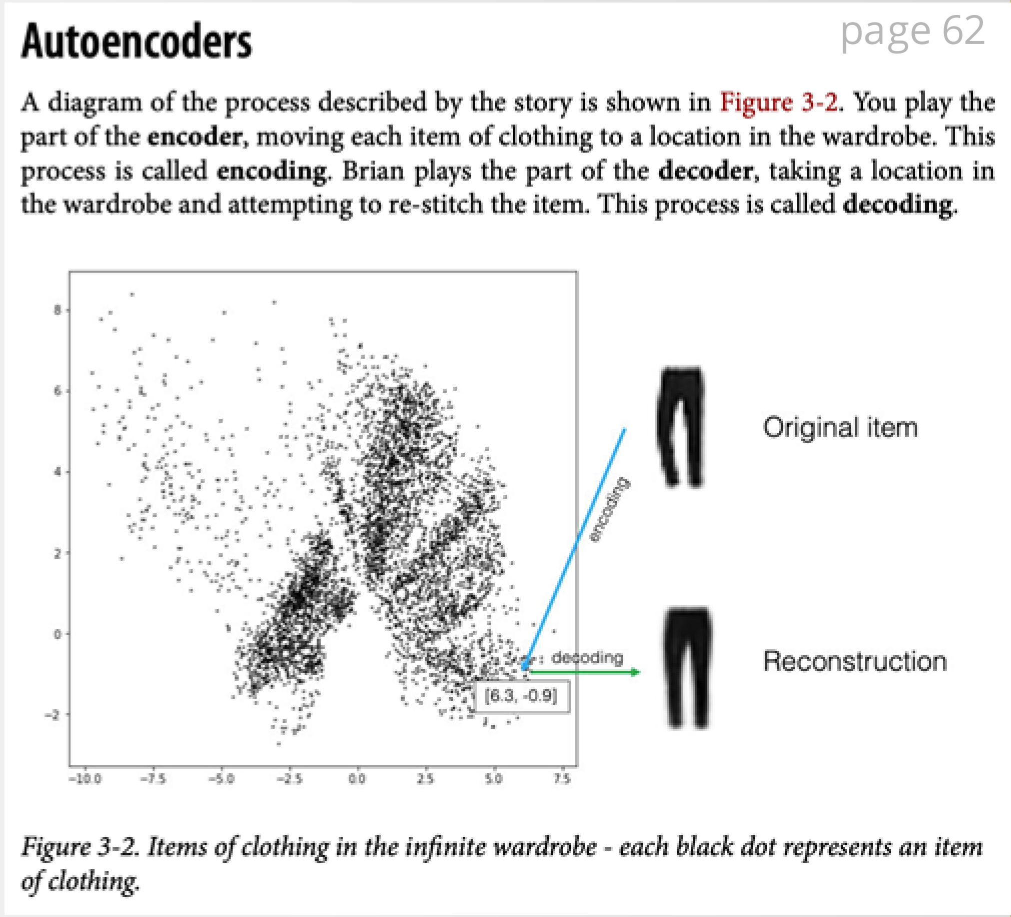
$$L_c = KL(P \parallel Q) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right). \tag{4}$$

Auto-Encoders

- Autoencoders comes in all types
  - **Convolutional,** (even TCN)
  - LSTM
  - Fully-Connected…







page 62

Autoencoders

A diagram of the process described by the story is shown in Figure 3-2. You play the part of the **encoder**, moving each item of clothing to a location in the wardrobe. This process is called **encoding**. Brian plays the part of the **decoder**, taking a location in the wardrobe and attempting to re-stitch the item. This process is called **decoding**.

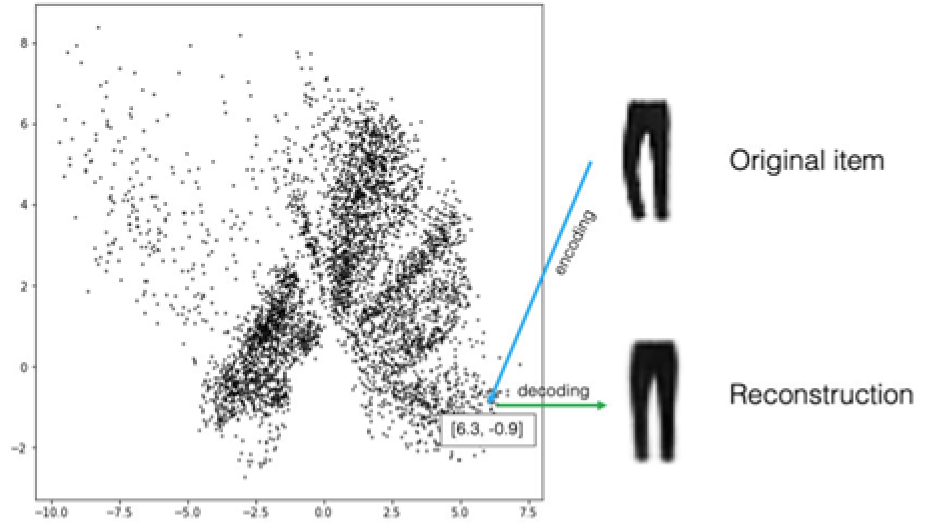
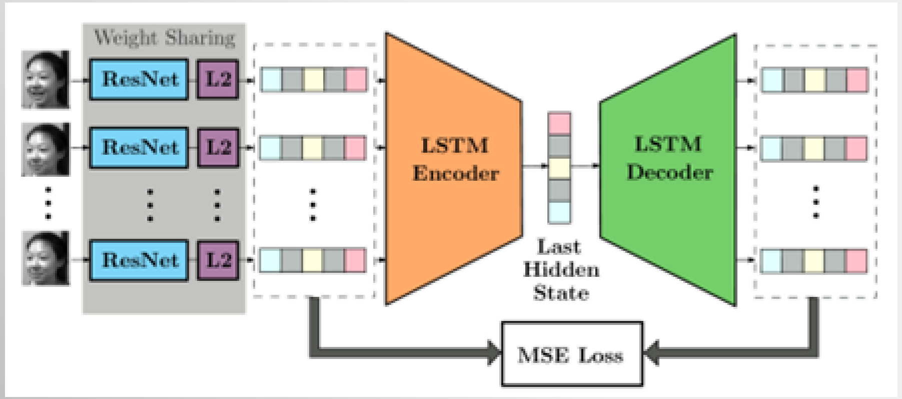
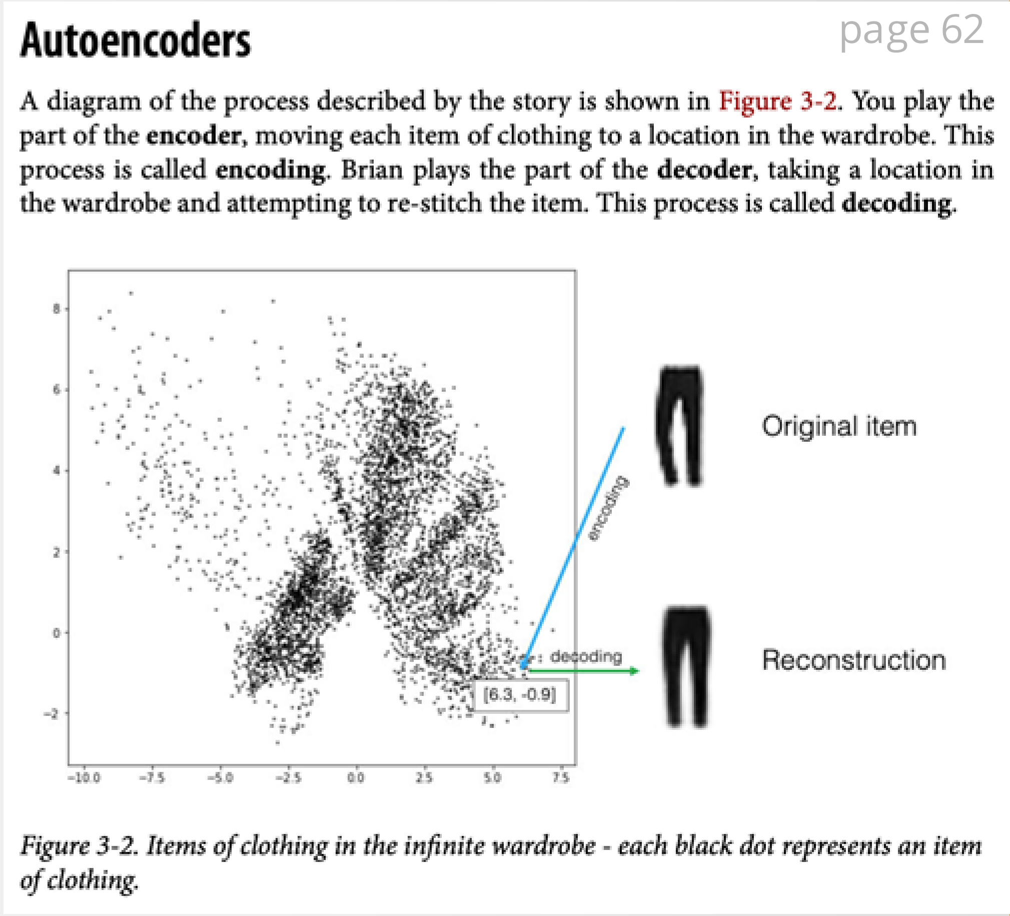


Figure 3-2. Items of clothing in the infinite wardrobe - each black dot represents an item of clothing.

Auto-Encoders

- Autoencoders comes in all types
  - Convolutional
  - **LSTM**
  - Fully-Connected





Autoencoders

A diagram of the process described by the story is shown in Figure 3-2. You play the part of the **encoder**, moving each item of clothing to a location in the wardrobe. This process is called **encoding**. Brian plays the part of the **decoder**, taking a location in the wardrobe and attempting to re-stitch the item. This process is called **decoding**.

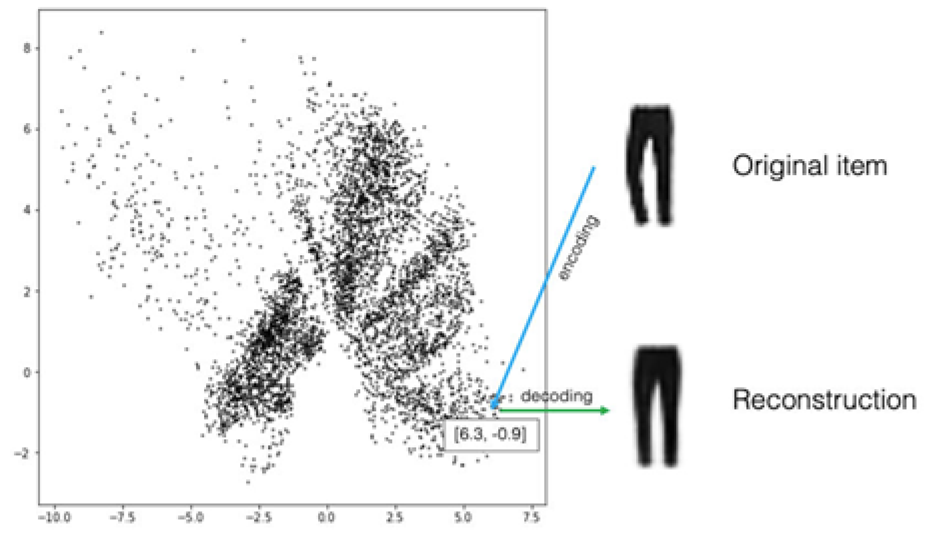
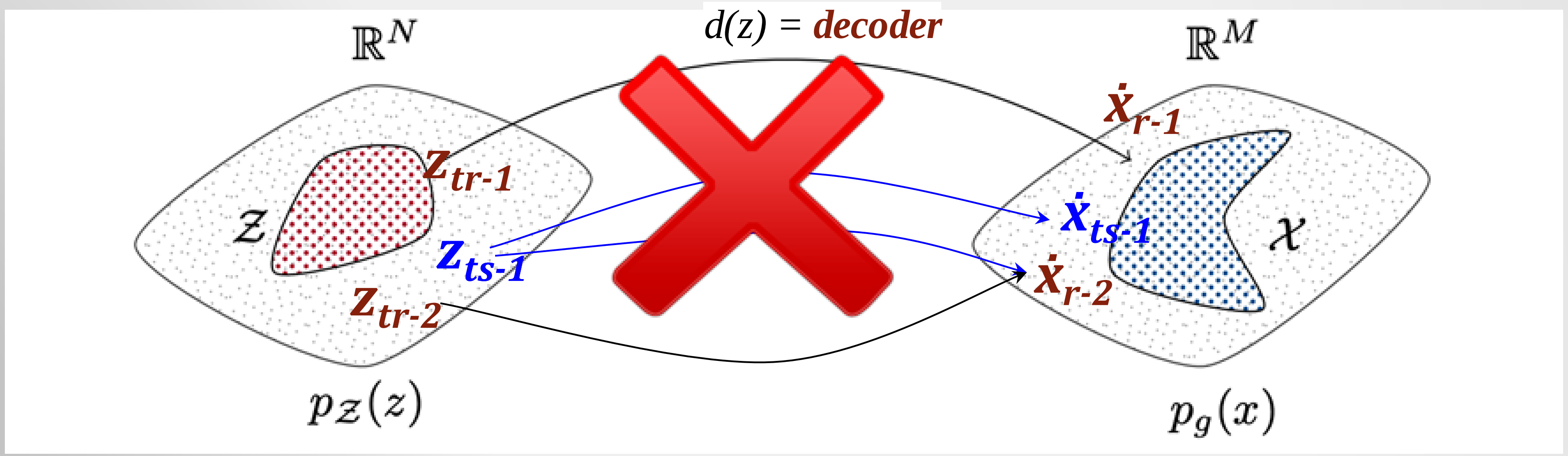


Figure 3-2. Items of clothing in the infinite wardrobe - each black dot represents an item of clothing.

AE problems

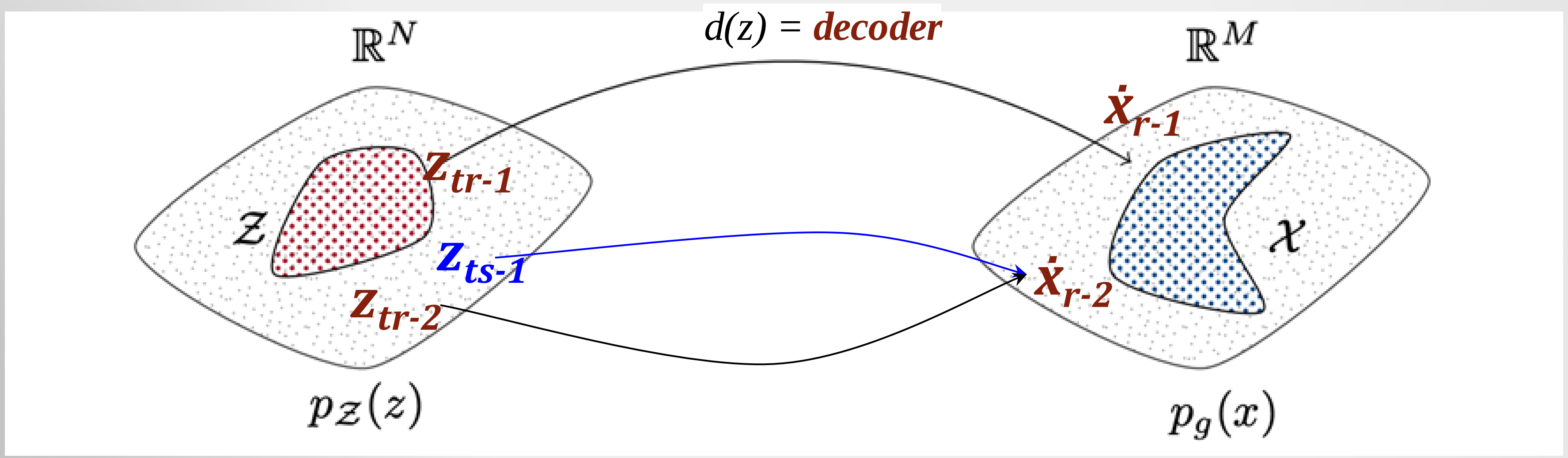
- One of the main limitations of traditional AEs is that they do not necessarily create a continuous latent space.
- This discontinuity makes it difficult for AEs to perform tasks like interpolation or extrapolation, where generating new samples in between or beyond the training data becomes challenging.



because it memorises the training samples

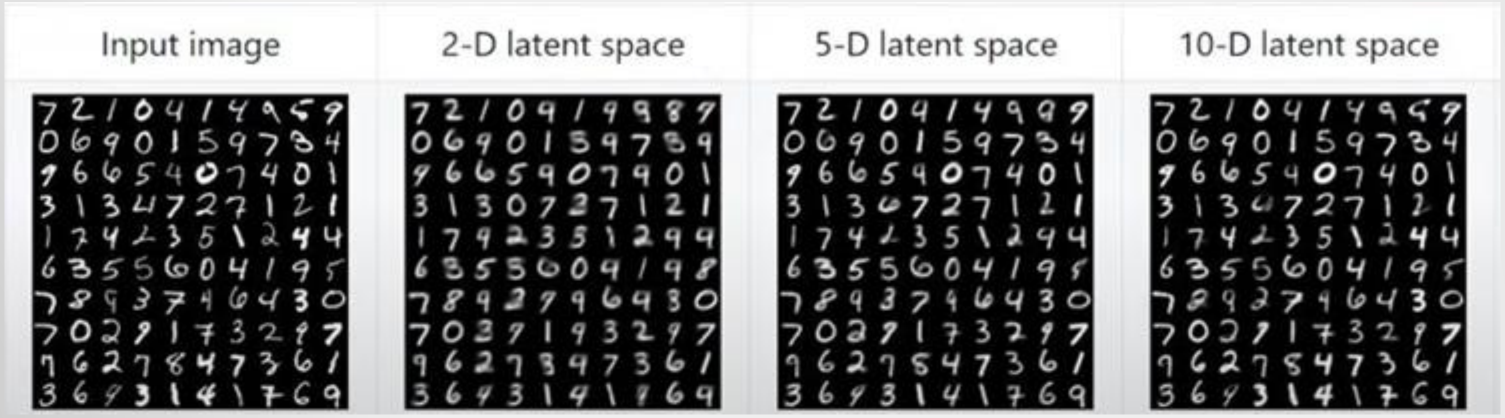
AE problems

- The fundamental problem with autoencoders, for generation, is that the latent space they convert their inputs to and where their encoded vectors lie, may not be continuous, or allow easy interpolation.
- This is fine if you're just replicating the same images.



Latent Space size ?

- Increasing the latent space vector ***z*** dimension size (***n***), will improve the replication/reconstruction success, but will affect the continuity problem even worse.
- Increasing the latent space vector ***z*** dimension size (***n***), will degrade the replication/reconstruction performance.



Next lecture:

- PART III: Auto-Encoding

- Autoencoders and Dimensionality Reduction
- **Variational Inference and VAEs**
- Conclusions

Thanks

Once I became a parent I finally understood the scene where Yoda gets so tired of answering Luke's questions he just dies.

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia