Artificial Intelligence and Machine Learning

Applications

Lecture Outline

- Linear Regression Tutorial & Playground
- Image Compression
- Logistic Regression Interactive Demo
- Neural Network Regression Tutorial & Playground
- Auto-Encoders

Linear Regression Tutorial

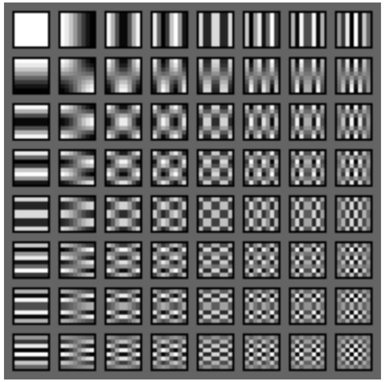
https://mlu-explain.github.io/linear-regression/

Linear Regression Playground

https://observablehq.com/@yizhe-ang/interactive-visualization-of-linear-regression

Image Compression with Linear Regression

- An image can be transformed into the frequency domain and represented as a combination of some basic components.
- Cosine Basis Functions and Discrete Cosine Transforms (DCT) can enable this.
- The human eye is most sensitive to low frequencies. Therefore, most of the high frequencies can be ignored.
- Principal behind JPEG Image Compression.



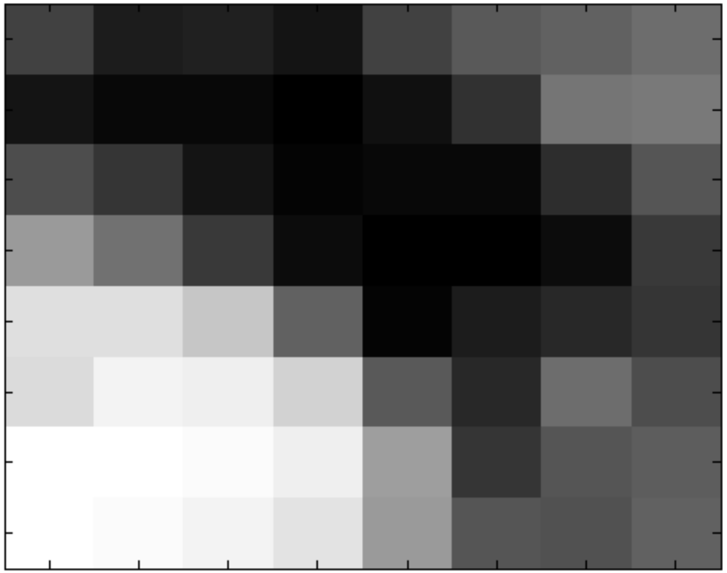
https://bengarney.com/2016/06/25/video-conference-part-2-joint-photographic-hell-for-beginners/

Image Compression with Linear Regression

- Using linear regression, we can learn the weight of each of these cosine basis.
- We only need to store the weights of the basis frequencies.
- The image is reconstructed using these weights.

Image Compression

8 x 8 Pixels



Image

Slide based on Discrete Cosine Transform and Image Compression by Thomas Buck et. Al.

Image Compression

- Gray-Scale Example:
- Value Range 0 (black) --- 255 (white)

63	33	36	28	63	81	86	98
27	18	17	11	22	48	104	108
72	52	28	15	17	16	47	77
132	100	56	19	10	9	21	55
187	186	166	88	13	34	43	51
184	203	199	177	82	44	97	73
211	214	208	198	134	52	78	83
211	210	203	191	133	79	74	86

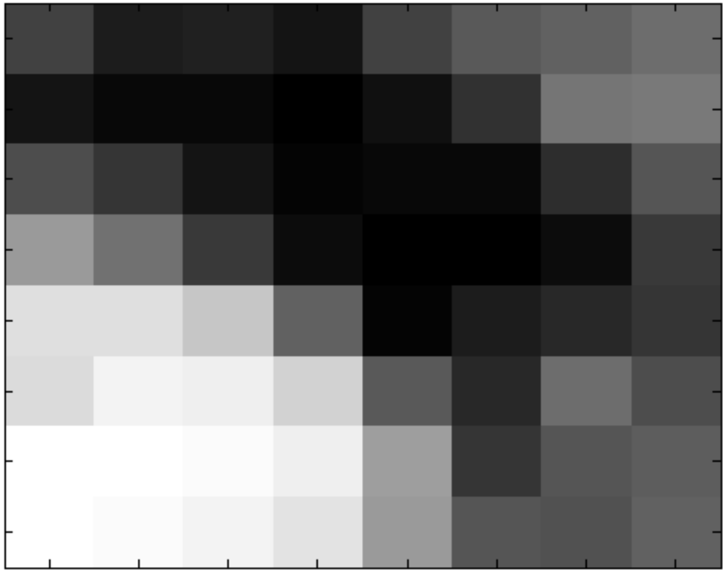
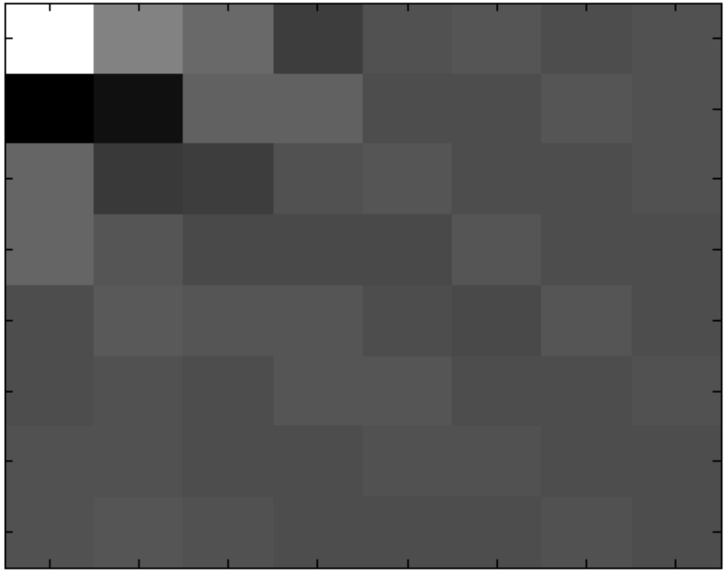


Image Compression

- 2D-DCT of matrix

Numbers are coefficients of polynomial

-304	210	104	-69	10	20	-12	7
-327	-260	67	70	-10	-15	21	8
93	-84	-66	16	24	-2	-5	9
89	33	-19	-20	-26	21	-3	0
-9	42	18	27	-7	-17	29	-7
-5	15	-10	17	32	-15	-4	7
10	3	-12	-1	2	3	-2	-3
12	30	0	-3	-3	-6	12	-1

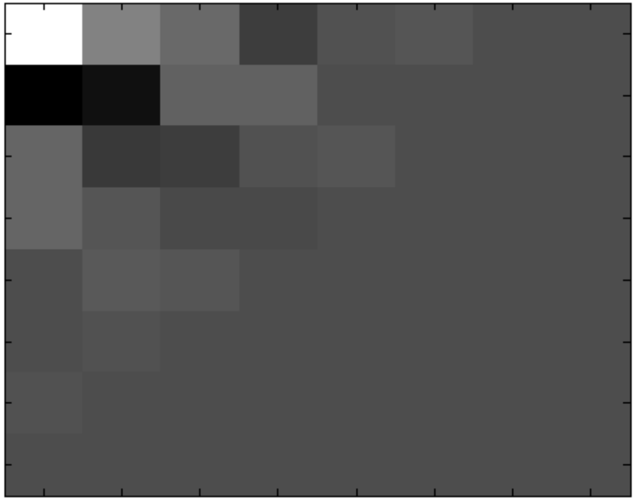


Slide based on Discrete Cosine Transform and Image Compression by Thomas Buck et. Al.

Image Compression

- Cut the least significant components

-304	210	104	-69	10	20	-12	0
-327	-260	67	70	-10	-15	0	0
93	-84	-66	16	24	0	0	0
89	33	-19	-20	0	0	0	0
-9	42	18	0	0	0	0	0
-5	15	0	0	0	0	0	0
10	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



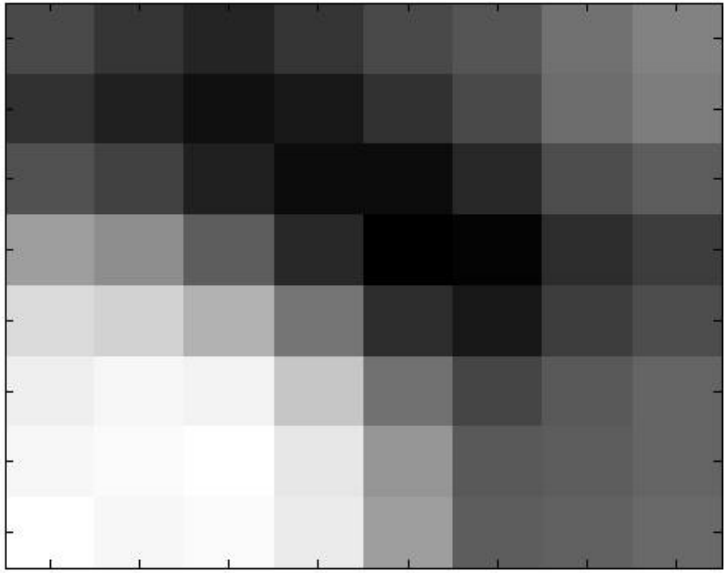
As you can see, we save a little over half the original memory.

Slide based on Discrete Cosine Transform and Image Compression by Thomas Buck et. Al.

Reconstructing the Image

- New Matrix and Compressed Image

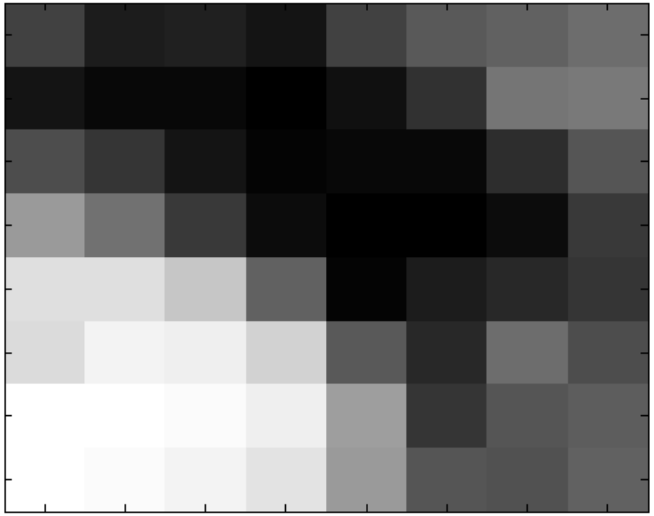
55	41	27	39	56	69	92	106
35	22	7	16	35	59	88	101
65	49	21	5	6	28	62	73
130	114	75	28	-7	-1	33	46
180	175	148	95	33	16	45	59
200	206	203	165	92	55	71	82
205	207	214	193	121	70	75	83
214	205	209	196	129	75	78	85



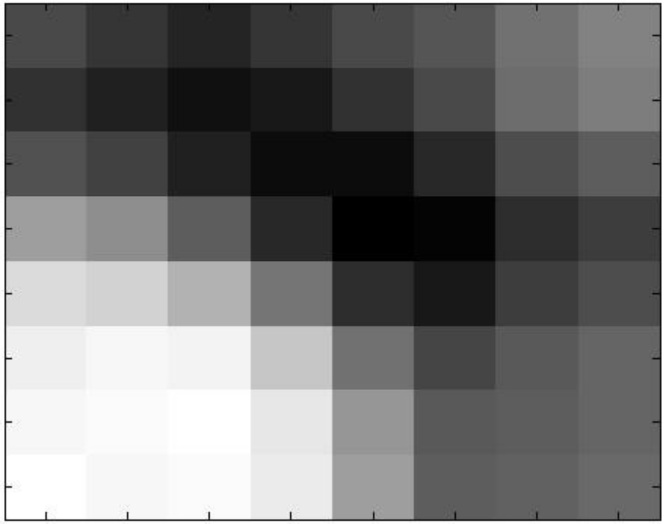
Slide based on Discrete Cosine Transform and Image Compression by Thomas Buck et. Al.

Can You Tell the Difference?

Original



Compressed



Slide based on Discrete Cosine Transform and Image Compression by Thomas Buck et. Al.

Image Compression

Original

Compressed

Slide based on Discrete Cosine Transform and Image Compression by Thomas Buck et. Al.

Logistic Regression Interactive Tutorial

https://mlu-explain.github.io/logistic-regression/

Neural Networks Tutorial

https://mlu-explain.github.io/neural-networks/

Neural Networks Playground

https://playground.tensorflow.org/

Try it Yourself – Digit Classification

https://trekhleb.dev/machine-learning-experiments/#/experiments/DigitsRecognitionMLP

Try it Yourself – Sketch Recognition

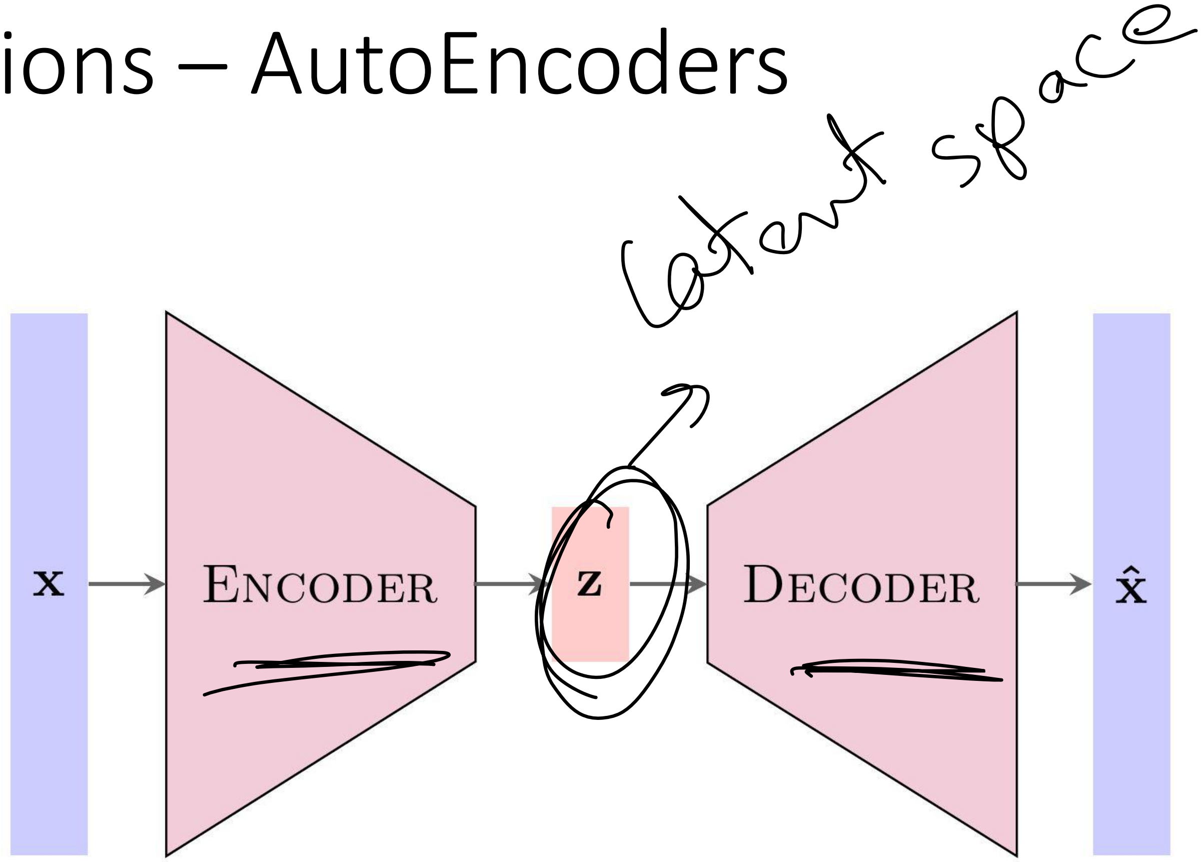
https://trekhleb.dev/machine-learning-experiments/#/experiments/SketchRecognitionMLP

Applications – AutoEncoders

أكاديمية كاوست
KAUST ACADEMY

- Autoencoders are a type of neural networks where the input is also the output.
- They come under unsupervised learning and there are no labels involved.
- An autoencoder consists of two parts: encoder and decoder.
- The idea here is that you take a higher dimensional input, project it into a lower dimensional space and then project it back into the input space.

Applications – AutoEncoders



https://github.com/JamesAllingham/LaTeX-TikZ-Diagrams

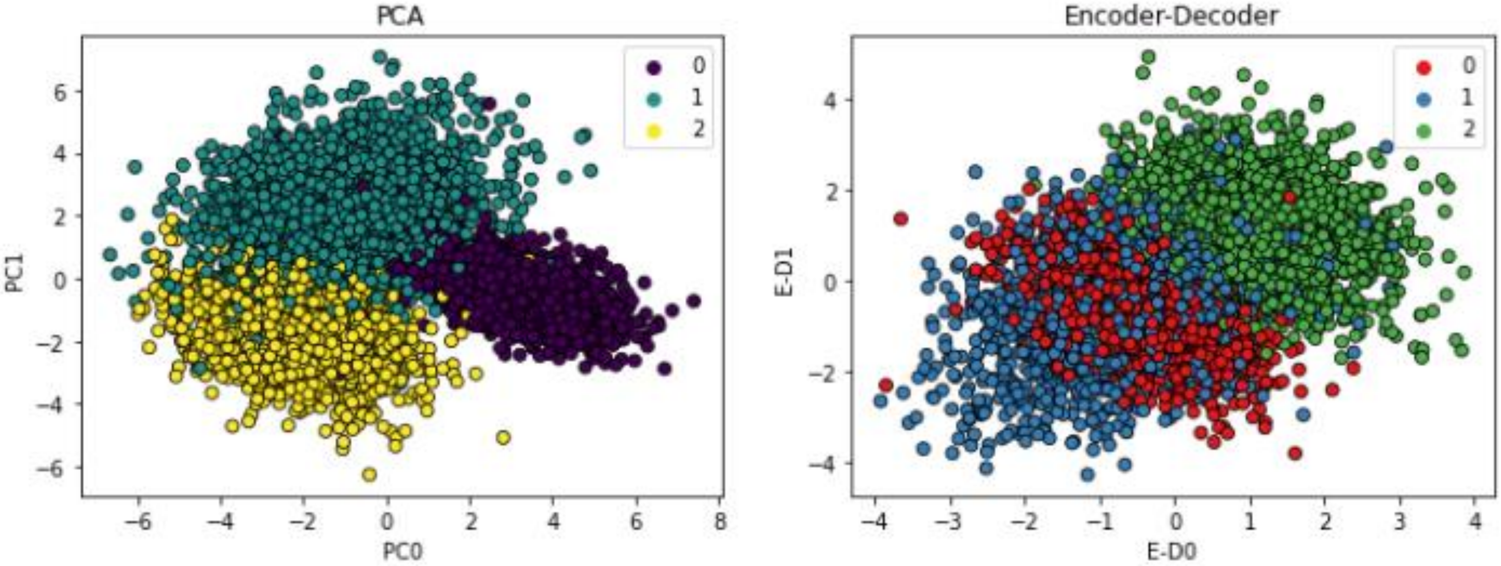
Applications – AutoEncoders

- The autoencoder model tries to minimize the reconstruction error (RE).
- Typically, mean squared is used as the loss function for autoencoders.
- The objective is to minimize the following:

$$L(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^{N} ||x_i - \hat{x}_i||^2$$

AutoEncoders for Dimensionality Reduction

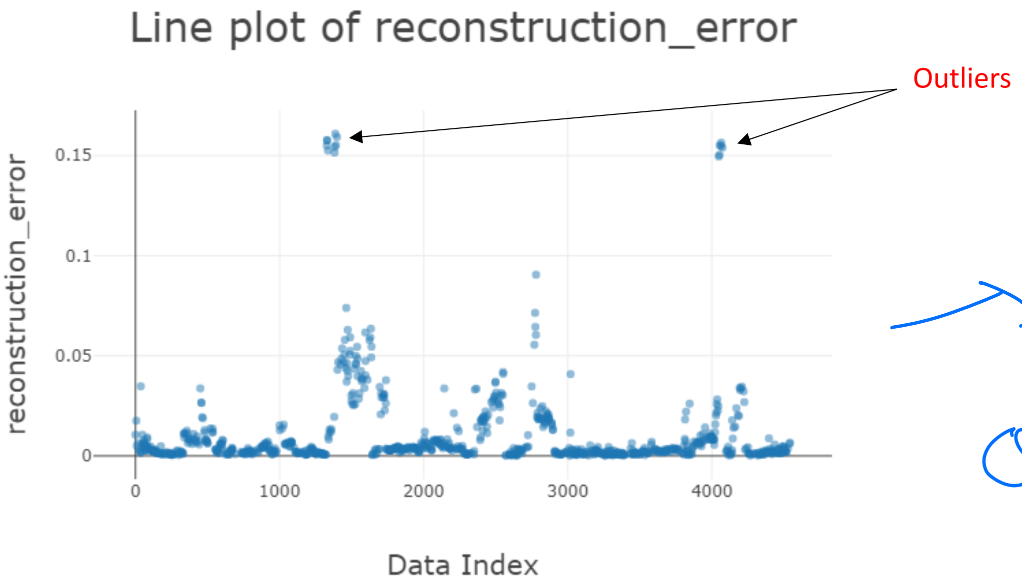
- The encoder output can be used for dimensionality reduction



https://towardsdatascience.com/dimensionality-reduction-with-autoencoders-versus-pca-f47666f80743

AutoEncoders for Outlier Detection

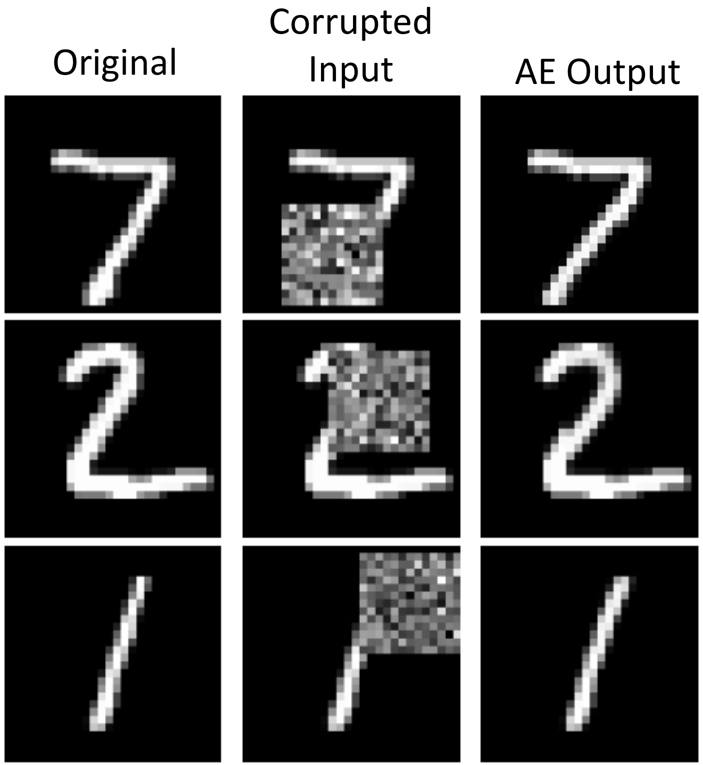
- The reconstruction error can be used to detect outliers. Out of distribution samples will have high Reconstruction error.

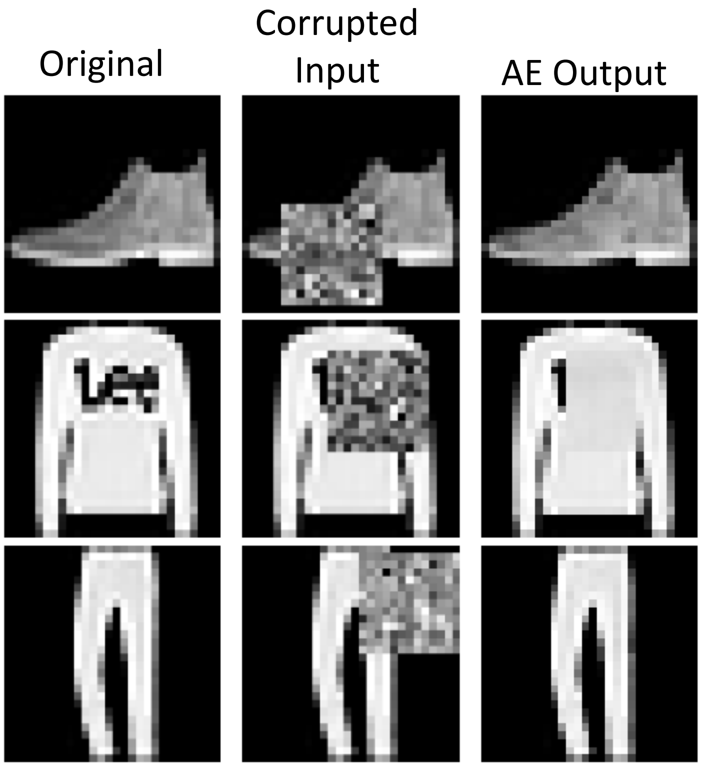


→ Identify outlier

https://towardsdatascience.com/dimensionality-reduction-with-autoencoders-versus-pca-f47666f80743

AutoEncoders for Image Completion





Smieja et. al. Can auto-encoders help with filling missing data?

Summary

Auto- encoder → Encoder
→ Decoder.

Latent space : compress version of the input data

- Encoder : map the input into a lower dimension.
  (use one or hidden Layers with a reduced number of neuron compared,

to the input data

- Decoder: reconstruct the original input data

(consist of hidden layers with a larger number of neurons than the encoder).

Example: - Reducing data dimension

- Enhance image.

* Limitation for reducing the dimension:

may fail to unseen data outside the training.

↳ testing

- A.E ⟶ non linear pattern

PCA ⟶ limited to linear transformation