Representation Learning

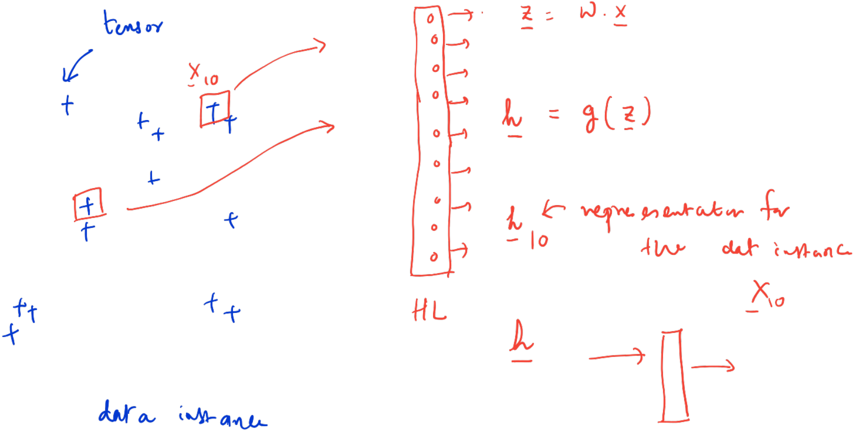
Tirtharaj Dash

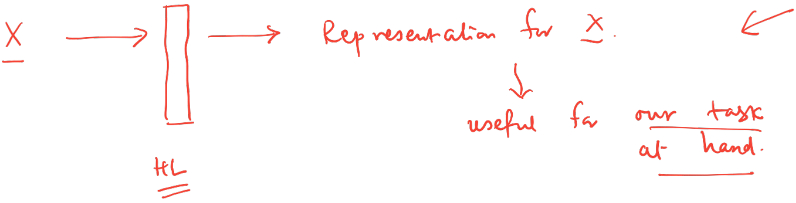
Dept. of CS & IS and APPCAIR
BITS Pilani, Goa Campus

November 23, 2021

Representation learning :-

Learning (compact) numeric representations for inputs.

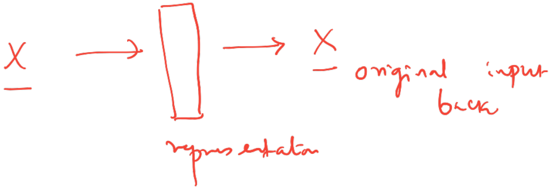

tensor
$\underline{x}_{10}$
data instance
HL
$\underline{z} = W \cdot \underline{x}$
$\underline{h} = g(\underline{z})$
$\underline{h}_{10}$ ← representation for the dat instance $\underline{X}_{10}$
$\underline{h}$



Representation for $\underline{x}$.

useful for our task at hand.

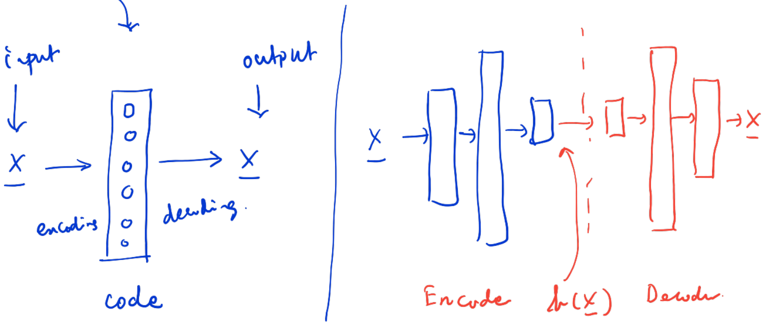
Unsupervised learning :-

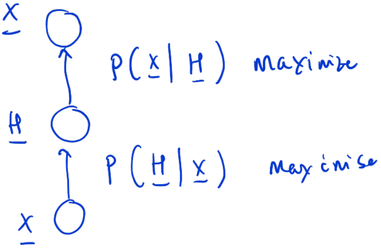


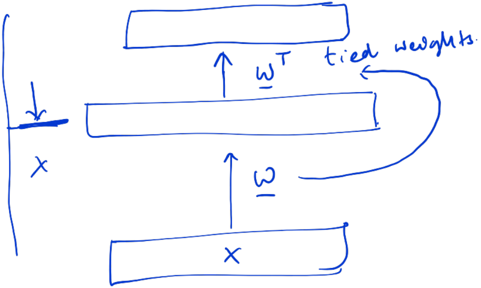
original input back

(1) Autoencoder: It learns $p(\underline{h} \mid \underline{x})$.

given some $\underline{x}$ it produces / constructs a code




$\underline{x}$
$\underline{H}$
$\underline{x}$
$P(\underline{x} \mid \underline{H})$ maximize
$P(\underline{H} \mid \underline{x})$ Maximise



$$\underline{x} \longrightarrow h(\underline{x}) \longrightarrow \hat{\underline{x}}$$

reconstructed data instance

Loss function?

① MSE : $\underline{x} \in \mathbb{R}^d$

② NLL : ? $\underline{x} \in \{0,1\}^d$.

Loss : $\mathcal{L}(\underline{x}, \underline{\hat{x}})$ reconstruction error.

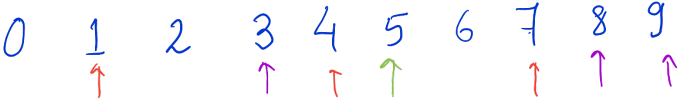
Questions:

(1) What if code is $\underline{\underline{\text{sparse}}}$? $\rightarrow$ length$(h(x))$ > length$(\underline{x})$

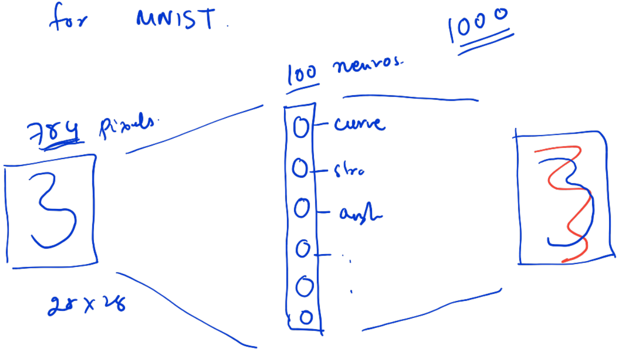
whether the code should be $\underline{\underline{\text{dense}}}$?

MNIST digits. :- hand written digits $\underline{\underline{0-9}}$.

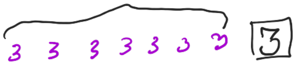
28×28

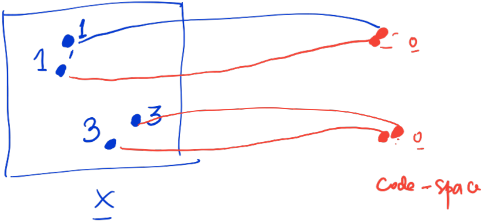


AE for MNIST.



Codes.





How to enforce that the code should be "compact"?

Reconstruction error: $L(\underline{x}, \underline{\hat{x}})$

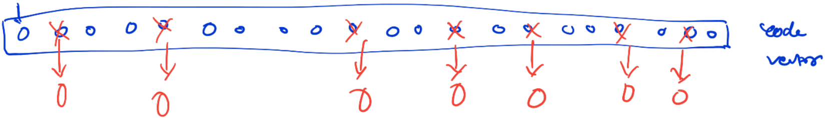
+ λ regularisation.

sparsity penalty.

↑

Sparsity regularisation

top-k features

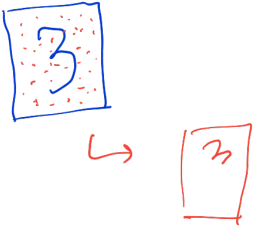


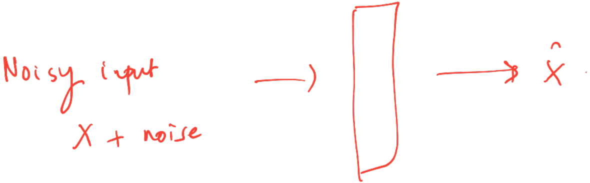
→ Penalising the hidden vector with L_1 or L_2.

(Sparse AE).

Problem: What if the data is noisy?

Assume: that the AE was trained with clean data.





Denoising AE:

1 → provide as input $\boxed{x + \underline{\text{Noise}}}$

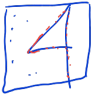
2 → learn the code

3 → try to reconstruct the clean x.

Noise:

① Gaussian noise : $\sigma \sim \mathcal{N}(0, \underbrace{\lambda}_{\text{regularisation parameter}})$

(real-valued input)

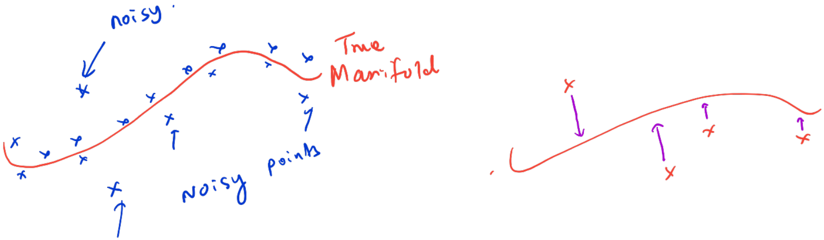


② Masking noise : set a fraction of inputs to 0

(Binary inputs)

③ Salt-pepper noise : set a fraction of inputs to max or min values.

e.g. Blurry image $\xrightarrow{\text{Denoise}}$ Clean image



Contractive autoencoding :

auto encoders-.

AE. : Discriminative models

$P(\underline{H}|\underline{x})$ $\quad$ $p(\underline{x}|\underline{H})$