28/12/25

Autoencoder (questions)

Q.1 An autoencoder [vanilla autoencoder, where ANN is used for encoder and decoder architecture] has one hidden layer

Given → Input Layer → has 10 neurons
↳ Hidden Layer → has 4 neurons

Then, in output layer → how many neurons will be there?

output layer → 10 neurons (ans)

2. If the autoencoder architecture:

vanilla autoencoder

8 → 3 → 8
input hidden output

How many parameters are there in this autoencoder? (simple vanilla autoencoder)

8 × 3 = 24 weights
+ 3 bias
= 27

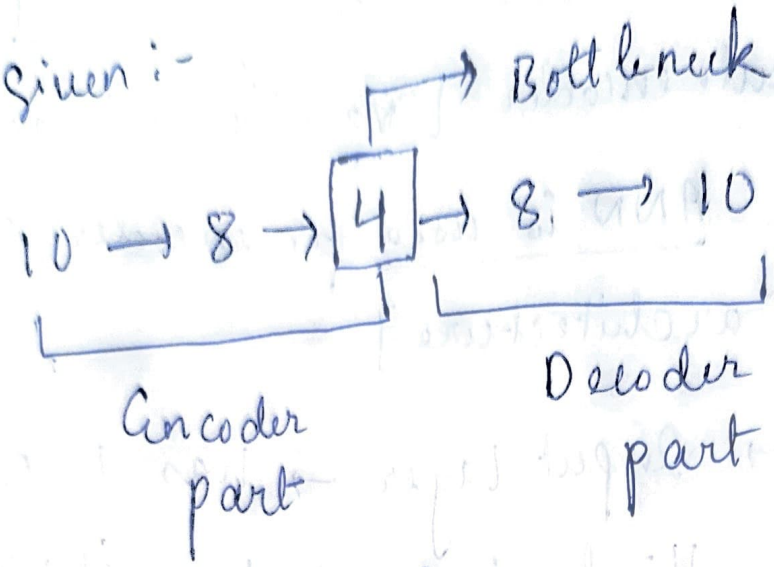
3 × 8 = 24 weights
+ 8 bias
= 32

Total parameters → 27 + 32 = 59 (ans)

3. Stacked autoencoder / deep autoencoder

=

Given :-



Total parameters = ?

i) $10 \rightarrow 8 \rightarrow 4$ (Encoder)

$10 \rightarrow 8 \Rightarrow 10 \times 8 = 80$ weights

$8 \rightarrow 4 \Rightarrow 8 \times 4 = 32$ weights

$10 \rightarrow 8 \rightarrow 4 \Rightarrow 80w + 32w + 8b + 4b$

$\Rightarrow 112w + 12b \Rightarrow \boxed{124}$

ii) Decoder part :- $4 \rightarrow 8 \rightarrow 10$

$\Rightarrow 4 \times 8 = 32w$

$\Rightarrow 8 \times 10 = 80w$

$\Rightarrow 8 + 10 = 18b$

$\Rightarrow 112w + 18b \Rightarrow \boxed{130}$

Total parameters $\Rightarrow$ $124 + 130 = 254$ $\Rightarrow$ 254 parameters (ans)

4. Suppose in an autoencoder :-

Input neurons = 50, hidden neurons = 80

Then, what kind (what type) of autoencoder is it ?

Ans → There are two types :-

- undercomplete autoencoder
- overcomplete autoencoder

→ Here, it is an overcomplete autoencoder [where the latent vector or the hidden neurons consist of more number of neurons in comparison to the input neurons]

→ hidden neurons ≥ (greater than or equal to) input neurons ⇒ overcomplete

→ hidden neurons < input neurons ⇒ undercomplete.

→ Training process or Forward propagation of a vanilla autoencoder :

a. Binary autoencoder (forward pass)

Given :- Binary input, $x = [1, 0]$

Encoder weight → We

$$W_e = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad b_e = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

b_e ↓ bias of encoder

Decoder weight → $W_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$,

$b_d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Q. → what is the structure / architecture of the auto encoder based on given information.

ans → 2 input neurons, 2 hidden & 2 output: $2 \to 2 \to 2$

what type of auto encoder is this?

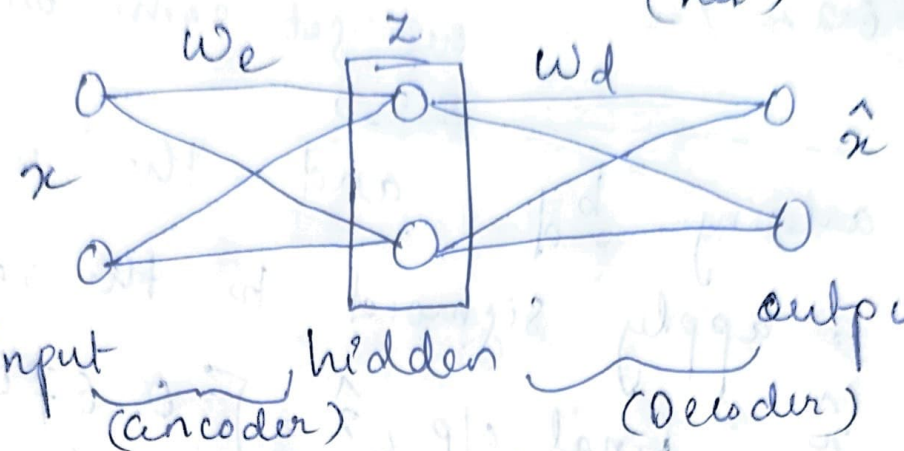
↳ Overcomplete [input ≤ hidden]

Q. what task is it performing?

[It copies the content of the input, directly to the decoder neurons. This architecture can perform a trivial copy operation. input information → passed to hidden layer → decoder to produce a reconstructed output]

↳ Overcomplete auto encoder

⇒ given activation function: Sigmoid (let)



① Equation of encoder

Linear Transformation: $\rightarrow \sigma(W_e \cdot x + b_e)$ ← sigmoid; b_e ← bias of encoder

$= z$; $z \Rightarrow$ o/p of encoder.

② Equation of Decoder :-

z is input to decoder (o/p of encoder)

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

Now, $W_e \cdot x = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$2\times2 \quad 2\times1$

$W_e x + b_e \Rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$z = \sigma\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} \sigma(1) \\ \sigma(0) \end{bmatrix} = \begin{bmatrix} 0.73 \\ 0.5 \end{bmatrix}$

↳ encoder output

$W_d \cdot z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{2\times2} \times \begin{bmatrix} 0.73 \\ 0.5 \end{bmatrix}_{2\times1} = \begin{bmatrix} 0.675 \\ 0.622 \end{bmatrix}$

$\begin{pmatrix} 0.675 \\ 0.622 \end{pmatrix}$ ⇒ we get some output

↳ adding b_d and then we need to apply sigmoid to the result to get $\hat{x}$. Final o/p, $\hat{x} = \begin{bmatrix} 0.675 \\ 0.622 \end{bmatrix}$

$\hat{x} = [0.68, 0.62]$ → reconstructed output

Reconstruction Error

$\hat{x} = [0.68, 0.62]$, $x = [1, 0]$

Loss function used → Binary Cross Entropy

$$L = -\frac{1}{n} \sum [x \log \hat{x} + (1-x) \log (1-\hat{x})]$$

~~$-x \log \hat{x}$~~

Mam's steps that I'm unsure about:-

$-x \log \hat{x} = -1 \cdot \log(0.68) = 0.136$

$-(1-0.62) \log(1-0.62) = 0.257$

Ignore this and calculate in your own way to check with the ans - [Find yourself]

$L = \frac{0.136 + 0.257}{2}$ =

[Not sure of this process]

⇒ If input is given as Real values to the autoencoder, what is the Loss function is used?

ans → MSE (Mean Squared Error). Let :-

original i/p = x = [3, 5, 2] → 3 neurons in i/p layer &

reconstructed o/p = $\hat{x}$ = [2, 4, 3] → 3 neurons in o/p

Then, MSE :-

$$= \frac{1}{3}\left[(3-2)^2 + (5-4)^2 + (2-3)^2\right]$$

$$= \frac{1}{3}\left\{1^2 + 1^2 + (-1)^2\right\} = \underline{1} \text{ (ans)}$$

Now ⇒ Autoencoders use Unsupervised Learning Algorithm. Why ~~auto~~ so?

ans → We are NOT using any target values or labeled data.

→ If we don't use any labeled data, then for a given dataset [Let 50 features / 100 samples. ⇒ 50 i/p neurons. Let the autoencoder used here ⇒ :- architecture is :-

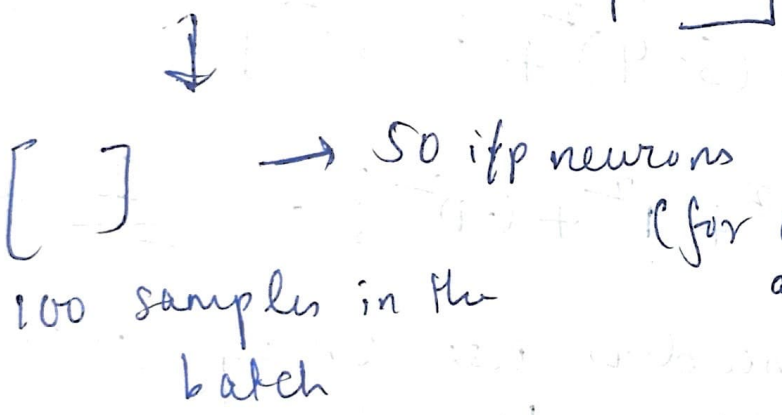
50 → 30 → 20 → [10] → 20 → 30 → 50

encoder | bottleneck | decoder.

Here, we have used a stack autoencoder (deep autoencoder) as we have more no. of features and we need to find out / capture the inherent patterns or features in a multiple level of hierarchy: 1st hidden layer → finds out primitive patterns

2nd hidden layer → less complex patterns

* data is more complex.



Objective → reconstruct the features, when we fed entire data at a time. No labeled data required or true target attr. value. [no need to check if true target value = predicted value. No need.]

→ All data samples is used for training purpose

↓

This is why we call autoencoder as unsupervised learning.

Q. How can an autoencoder be used for classification task?

ans → objective of an autoencoder → reconstruct the data. Such that $\hat{x} \approx x$

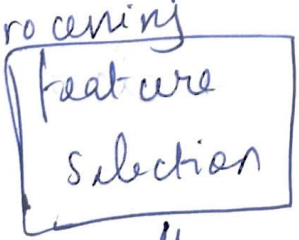
[it cannot reconstruct the data 100% accurately as it loses certain amt. of data during compression. Still, this compressed data (bottleneck) has all the inherent (inherent) patterns of all the i/p data so that we can reconstruct that particular input approximately equal to i/p given to the autoencoder]

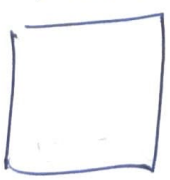
* which part of autoencoder is used as part of classification task?

classification task: feature selection → classifying the data.

↳ classification :-

preprocessing





training → classify an unknown data to its true class.

↓

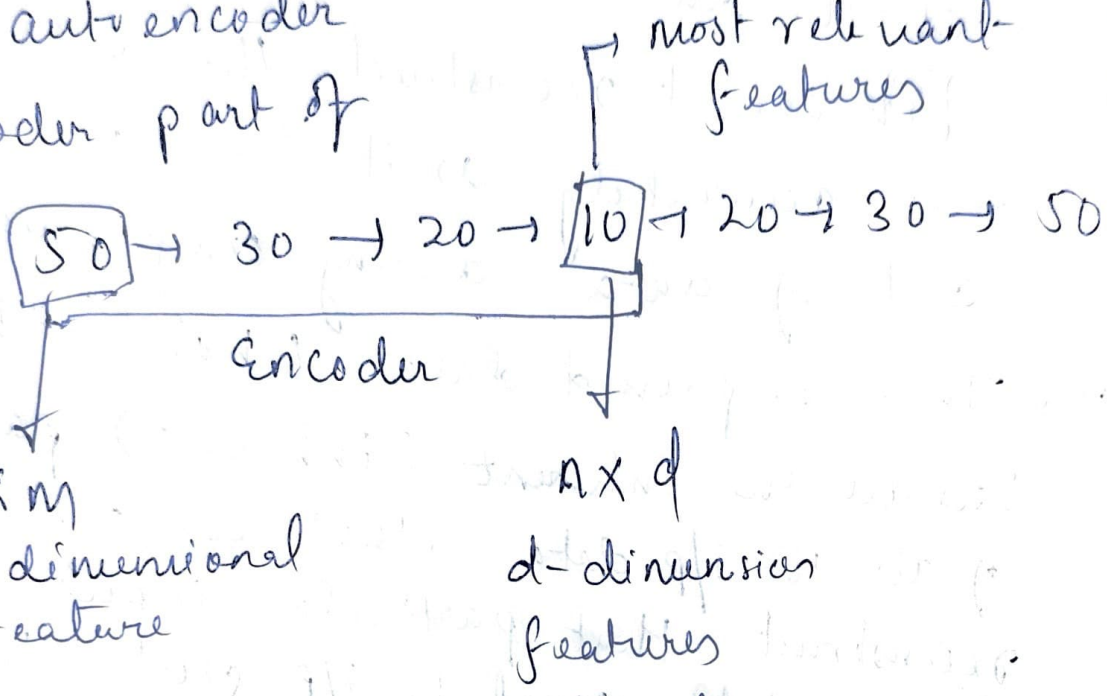
reduces the dimension of data

→ feature selection →

then $d < m$

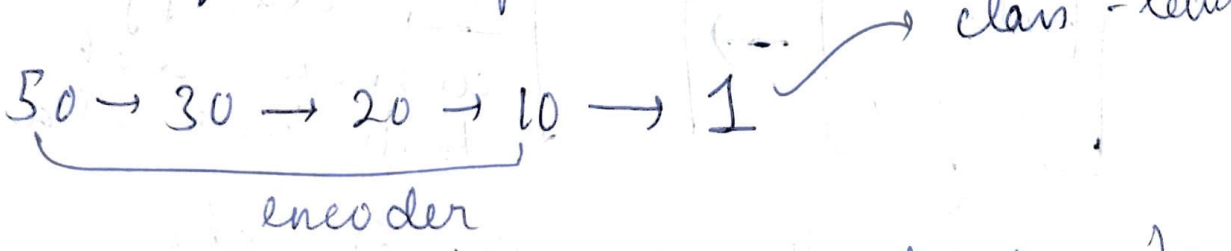
Now, we observe the feature selection part of classification is already being performed by the encoder part of autoencoder

Encoder part of



bottleneck features ⇒ MOST Relevant features as we can reconstruct the entire i/p data almost accurately from these features.

Now, encoder part taken individually and we add to it a single neuron (let) for classification purpose and apply an activation function → it can give the class level of a sample.



* Encoder is used for classification task.

Autoencoder can be used for :-

1. Image reconstruction
2. Text reconstruction

[We cannot use vanilla autoencoder as it uses ANN]. We use variants of autoencoder.

Encoder & decoder architecture

Image reconstruction → CNN

Text Reconstruction → RNN / LSTM / GRU

Same training process for CNN for Encoder & Decoder

—X—