NLP In Deep Learning

Sunday, 29 December 2024 9:10 PM

NLP in Deep Learning :-

Text Data → Vectors → Numerical Rep.

1. OHE - one hot encoding
2. BOW
3. TF-IDF
4. Word2Vec, Avg Word2Vec

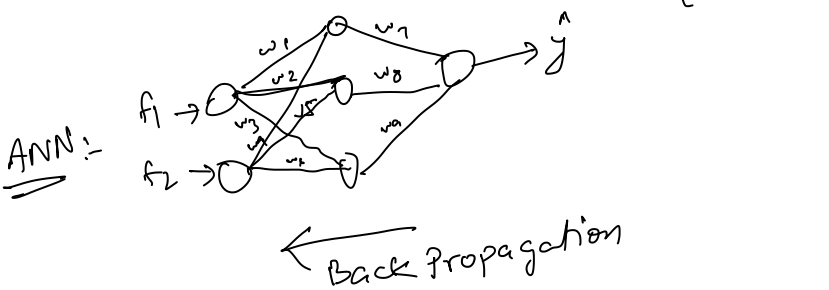
(Sentiment Analysis, Text classification)

ANN → Artificial Neural Network

- Classification
- Regression

Eg. House Price Prediction

f_1 size	f_2 rooms	y price



2. CNN → convolutional Neural Network
   - ↳ Image classification.

   Data - Image, Video frames.

3. Data - Sequential Data
   1. Text generation

Eg. I/P: This is a Apple → sequence → O/P: juice

Eg. Chatbot conversion → Q/A
I/P: Question → O/P: Answer

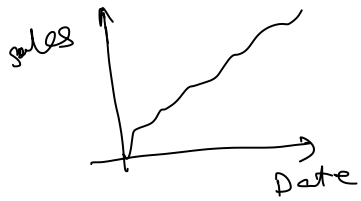
Sequential Data → The food is good.

Eg Language Translation:

English → French.

Eg. Auto suggestion
LinkedIn, Gmail.

Eg. Sales Data.



Sales forecasting on future date

Can we use ANN to solve this problem?
→ which has sequential Data.

NLP : Generative AI → LLM, Multi Model.

① SimpleRNN ⇒ LSTM / GRU RNN
↓
Bidirectional RNN
↓
Encoder Decoder
↓
Self Attension ← Transformers

Can we use ANN ⇒ sequential data.

Dataset :- Sentiment Analysis

Text	O/P
the food is good	1
the food is bad	0
The food is not good	0

① Text preprocessing → Text → Vectors

Vocab: → 4

BOW: convert every sentence we convert to vectors.

	food	good	bad	not
S1	[1	1	0	0]
S2	[1	0	1	0]
S3	[1	1	0	1]

} I/P.

Text Data – Sequence Information is important

Meaning of Sentence is lost.

In case of ANN the data input is all at one, so we are lossing the sequence information.

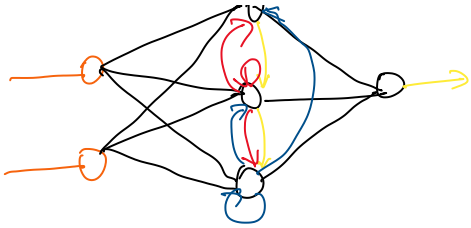
In case of Text Data :- we need to give one word at a time then backpropagation should happen & then another word.

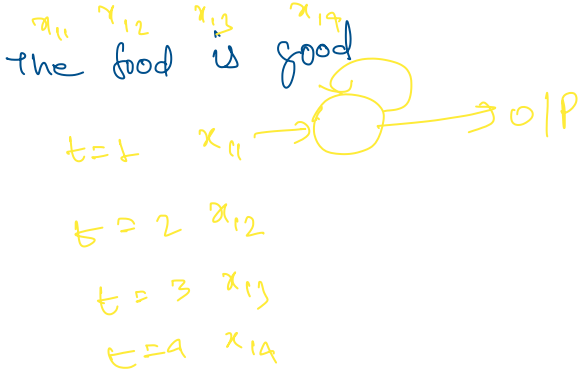
Google Translation:-

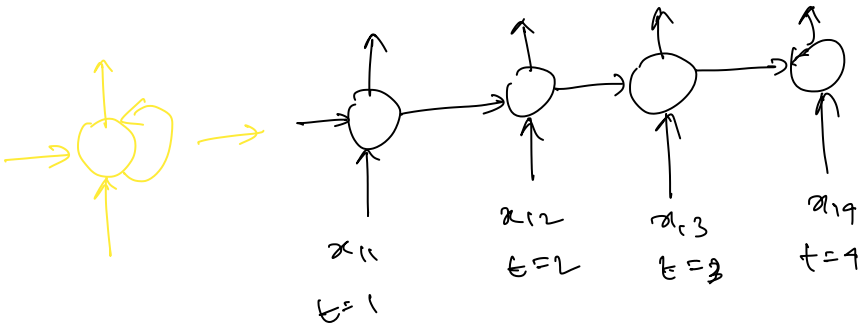
word to word translation takes place

There is another neural network

Simple RNN → Recurrent Neural Network







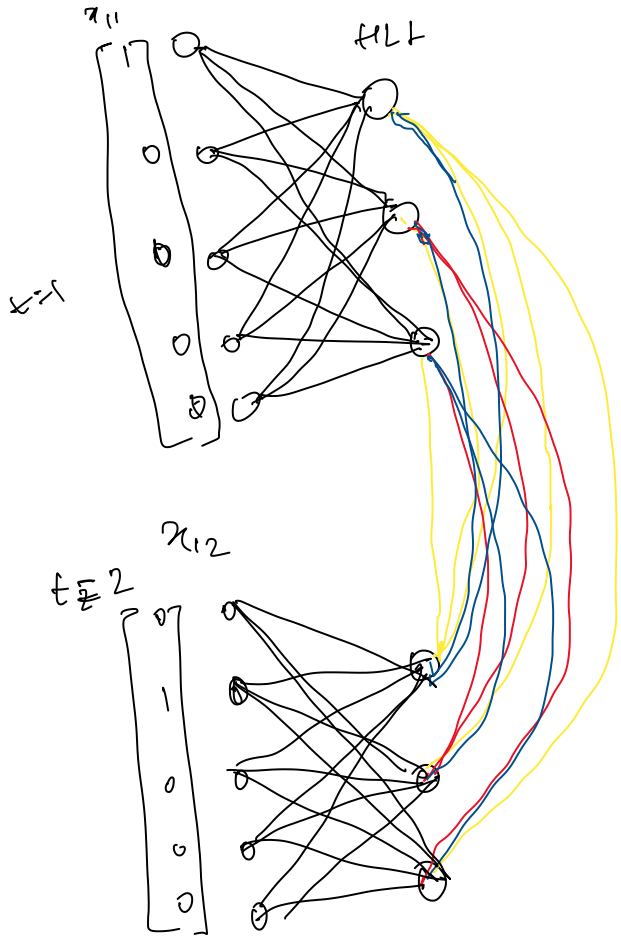
Working of Simple RNN with forword propagation :-

Dataset

	text	O/P
S1	the food is good	1
S2	the food is bad	0
S3	the food is not good	0

OHE

[[1 0 0 0 0],
[0 1 0 0 0],
[0 0 1 0 0],
[0 0 0 1 0],
[0 0 0 0 1]]

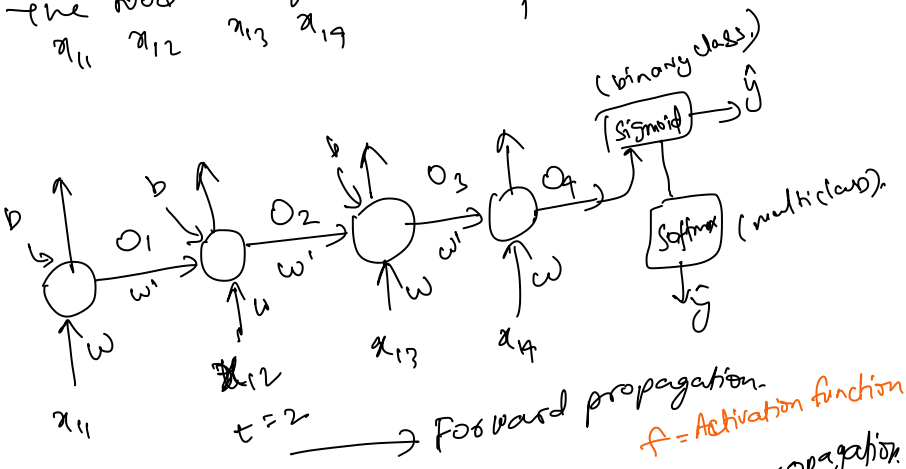

t=1
t=2
t=3
t=4
O/P
3X1

5x3 → 15 wt. + 1 b
3x3 → 9 wt + 1 b
1 b.

⇒ 30 parameters are there.

Dataset :-

the	food	is	good.	
x_{11}	x_{12}	x_{13}	x_{14}	O/P
				1



→ Forward propagation.

f = Activation function.

forward propagation

$t=1,\ O_1 = f(x_{11} \cdot w + b_1)$

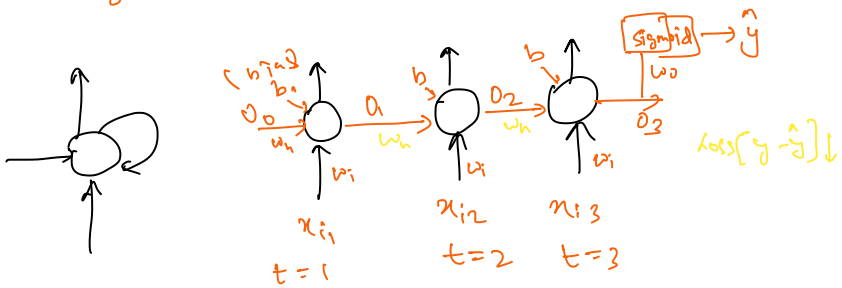
$t=2\ \ O_2 = f(x_{12} \cdot w + O_1 w' + b_1)$

$t=3\ \ O_3 = f(x_{13} \cdot w + O_2 \cdot w' + b)$

$t=4,\ O_4 = f(x_{14} \cdot w + O_3 w' + b)$

RNN Back Propagation with time →

How the back propagation happens and weights are updated. in backpropagation.



Forward Propagation!-

$O_1 = f(x_{i1} w_i + O_0 w_h + b)$

$O_2 = f(x_{i2} w_i + O_1 w_h + b)$

$O_3 = f(x_{i3} w_i + O_2 w_h + b)$

$(O_3 * w_o)$

To reduce the loss we update the weights

$[w_i, w_h, w_o]$ & then we need to

$\hat{y}$ = $\sigma(z^{-3})$ −

do backward propagation to reach global minima

Backward Propagation with time:-

update $[w_i, w_h, w_o]$

① weight updation formula:-

$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w_{old}}$$ → slope of gradient descent.

$$w_{o_{new}} = w_{o_{old}} - \eta \frac{\partial L}{\partial w_{o_{old}}}$$

based on chain Rule:

$$\frac{\partial L}{\partial w_{old}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_{o_{old}}}$$

② update w_h [Hidden layer weights]:

$$w_{h_{new}} = w_{h_{old}} - \eta \frac{\partial L}{\partial w_{h_{old}}}$$

$t=1,2,3$

$t=3 \Rightarrow \frac{\partial L}{\partial w_{h_{old}}} = \left[\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_3} \cdot \frac{\partial O_3}{\partial w_h}\right]_{t=3} + \left[\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_3} \cdot \frac{\partial O_3}{\partial O_2} \cdot \frac{\partial O_2}{\partial w_h}\right]_{t=2}$

$$+ \left[\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_3} \cdot \frac{\partial O_3}{\partial O_2} \cdot \frac{\partial O_2}{\partial O_1} \cdot \frac{\partial O_1}{\partial w_h}\right]_{t=1}$$

③ updating weights w_i :→

$$w_{i_{new}} = w_{i_{old}} - \eta \frac{\partial L}{\partial w_{i_{old}}}$$

$$\frac{\partial L}{\partial w_{i_{old}}} = \left[\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_3} \cdot \frac{\partial O_3}{\partial w_{i_{old}}}\right]_{t=3} + \left[\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_3} \cdot \frac{\partial O_3}{\partial O_2} \cdot \frac{\partial O_2}{\partial w_{i_{old}}}\right]_{t=2}$$

$$+ \left[\frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_3} \cdot \frac{\partial O_3}{\partial O_2} \cdot \frac{\partial O_2}{\partial O_1} \cdot \frac{\partial O_1}{\partial w_{i_{old}}}\right]_{t=1}$$

ANN – Vanishing gradient problem.

We can face the same problem in RNN too.

① The long term dependency cannot be captured by RNN with accuracy.

$$t=1 \quad \frac{\partial L}{\partial W_{hold}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_2} \cdot \frac{\partial O_1}{\partial O_2} \cdot \frac{\partial O_2}{\partial W_{hold}}$$

suppose if the sentence length is 50 words

$$\text{at } t=1, \quad \frac{\partial L}{\partial W_{hold}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_{50}} \cdot \frac{\partial O_{50}}{\partial O_{49}} \cdots \frac{\partial O_3}{\partial O_2} \cdot \frac{\partial O_2}{\partial W_{hold}}$$

$$\frac{\partial O_3}{\partial O_2} = \frac{\partial \sigma(x_3 * W_i + O_2^{0} * W_h + b)}{\partial O_2}$$

Derivative of sigmoid [0 - 0.25]

$$= \sigma'(1 * W_h)$$

$$[0 \to 0.25]$$

At $t=1$ $\quad \frac{\partial L}{\partial W_{hold}} \simeq 0$ [not participating much in updation of value]

⇒ The initial word is not playing significant role in the output.

⇒ The nearest words only have the significant impact.
The chain rule becomes bigger and approximates to 0.

This Problem is known as Vanishing Gradient Problem.

→ In order to solve this we can use other activation function such as Relu.
Leaky Relu.

There is another solution:-

① LSTM RNN → Long short term Memory RNN

② GRU RNN