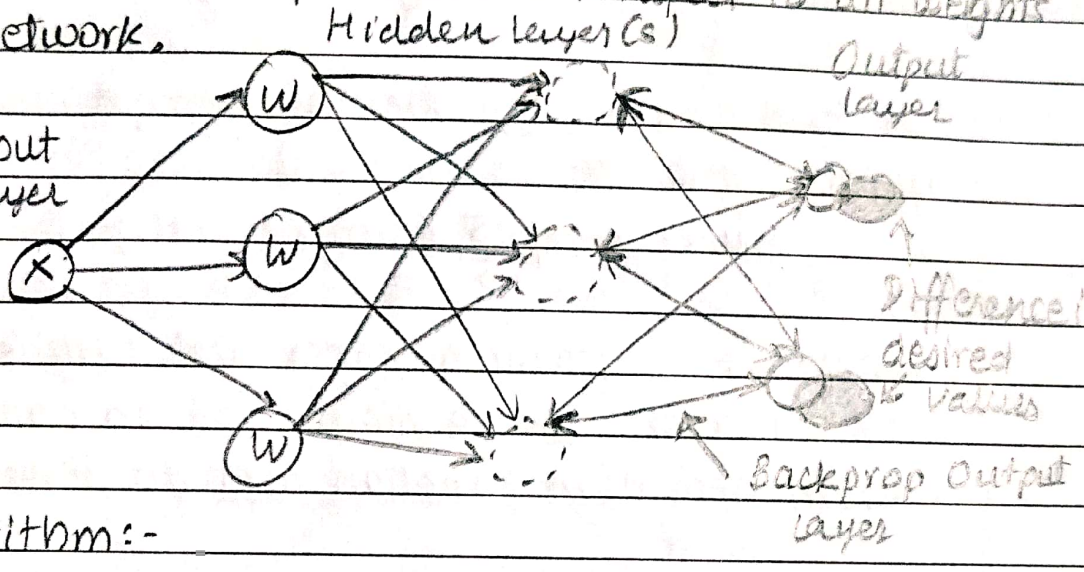
DATA SCIENCE
HONOURS ENDSEM NOTES

* UNIT 3 already studied in ML *

We will only cover concepts not studied

UNIT 4 : Artificial Neural Network

* Back Propagation algorithm :-
- Method of fine tuning weights of a neural network based on errors obtained in previous epoch.
- Proper tuning helps in increasing model generalization.
- Helps calculate loss function with respect to all weights in the network.



- Algorithm :-

1) Inputs X arrive through preconnected path
2) Input is modelled using real weights (randomly selected)
3) Calculate output for every neuron from the input layer to the hidden layer, to the output layer.
4) Calculate error in the outputs :-
   Error = Actual Output - Desired output
5) Travel back from output layer to the hidden layer and adjust weights such that error is decreased.

- Advantages :-
Fast and simple, flexible & standard method
- Disadvantages :-
Sensitive to noise, Performance dependent on input data.

* Generalized Delta learning Rule :-

- Most common method for training backpropagation network.
- Iterative gradient-descent method that minimizes the least-mean-squares (LMS) error in the output.
- Consider in, out and w to be input, output and weights.

If we use an activation function such as sigmoid a single layer network output is given by

$$out_i = Sigmoid\left(\sum_i in_i \cdot w_{ii}\right)$$

Due to derivate properties of sigmoid, weight update equation is

$$\Delta w_{ki} = \eta \sum_p (targ_i - out_i) \cdot f'\left(\sum_i in_i \cdot w_{ii}\right) \cdot in_k$$

where η is learning rates and targ is desired output. This equation can be simplified to contain only neuron activation function and no derivatives.

$$\Delta w_{ki} = \eta \sum_p (targ_i - out_i) \cdot out_i \cdot (1 - out_i) \cdot in_k$$

This is known as generalized Delta learning Rule.

* Limitations of MLP :-

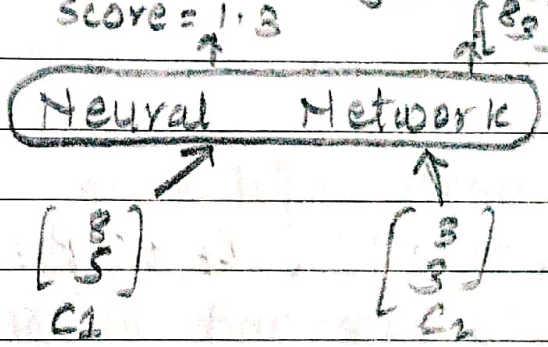
- MLP is Multi-layer perceptron, a class of feed-forward network. It has $\leq$ 3 layers. Every neuron uses a non-linear activation fn. Uses backpropagation for training.
- Limitations :-

1) Computations are difficult & time consuming.
2) Not known to what extent dependent variable is affected by dependent variable.

3] Proper function of model depends on quality
4] It includes too many parameters being fully con

UNIT V: Convolution Neural Network

* Recursive Neural Network:-
- To work with phrases and sentences we need t
relation between word & multi-word expressions.
- For example "consider" == "take into accou
- To solve this problem we use recursive neural
- It uses binary tree to identify phrases or senter
- We recursively merge pairs of representation of
segments to get representation of bigger segm
- The tree if created based on 2 rules:-
  - Semantic representation if two nodes are merge
  - Score of how matching two words are.



* Recurrent Neural Network:-
- Done in ML

* Long-short Term Memory:-
- An advanced RNN, sequential network that allow

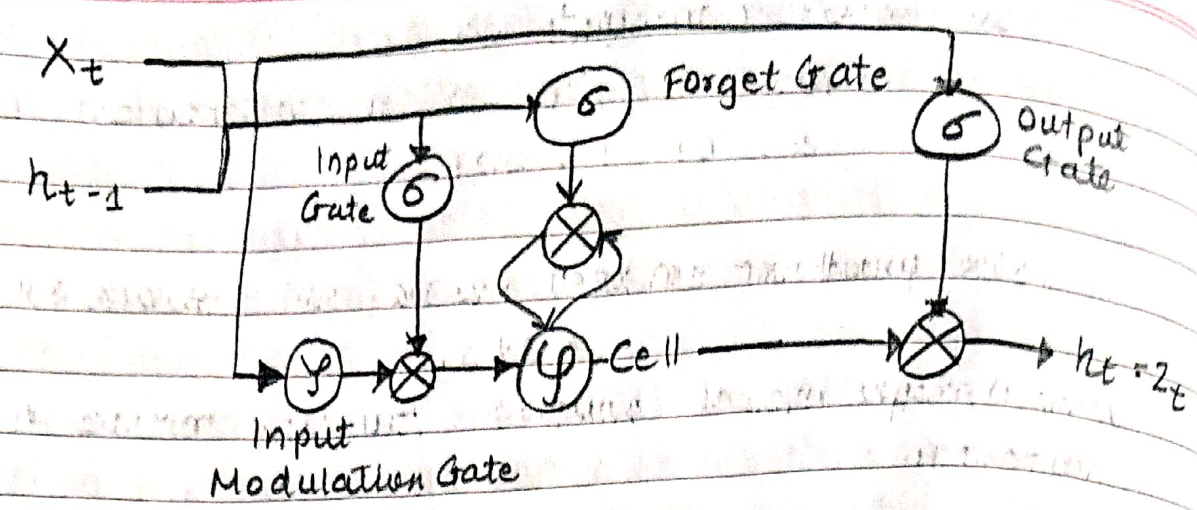
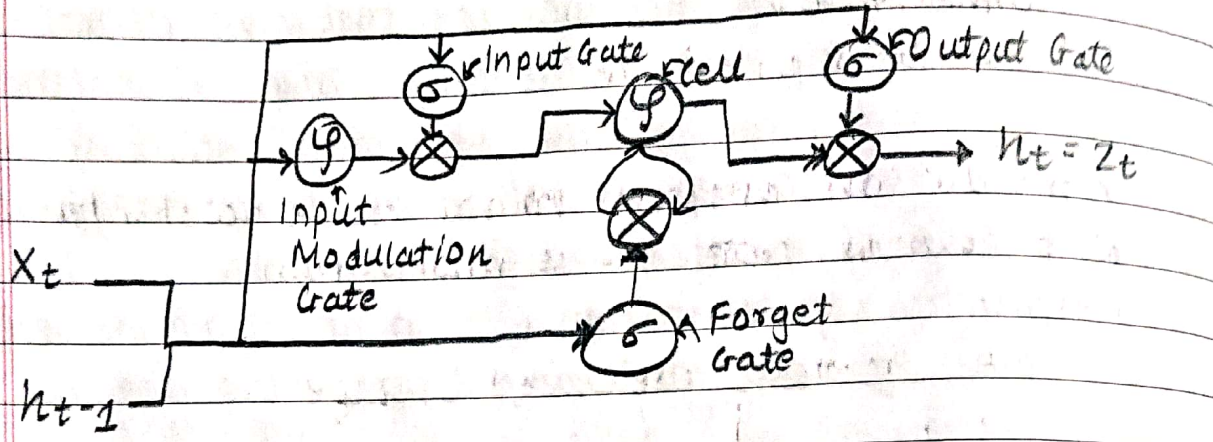


Fig: LSTM

Let's redraw to simplify



1. Forget Gate :-
   - Info thats no longer useful is removed with this gate.
   - X_t : Input at a time "t", h_{t-1} : Previous cell output
   - These are fed to gate and multiplied with weights,
   - Result is passed through a binary activation function.
   - If output is 1 info is retained, if 0 it is forgotten.
2. Input Gate :
   - Info. is regulated and filtered are values that need to be remembered.
   - Modulated input and normal input are multiplied together to get useful information.
3. Output Gate :-
   - Extracts useful info from current cell state.
   - output is sent as an input to next cell.

* Gradient Descent Optimization :

- There are various methods that make gradient descent more optimized. Let us see :-

1) Momentum Method :

- To make gradient descent faster, we consider exponentially weighted average of gradients.
- Using average makes converging towards minima faster.
- Pseudocode :-

```
V = 0
for each iteration i :
    compute dW
    V = βV + (1 - β) dW
    W = W - αV
```

- V and dW are analogus to velocity & acceleration respectively
- α is learning rate, β is momentum

2) RMSprop :

- Apply exponentially weighted average to second moment of gradients (dW).
- Pseudocode :-

```
S = 0
for each iteration i :
    compute dW
    S = βS + (1 - β) dW²
    W = W - α [dW / (√S + ε)]
```

3) Adam Optimization :

- Adam Optimization = RMSprop + Momentum + Bias correction
- Pseudocode :

→ continued here ↘

```
V = 0
S = 0
for each iteration i :
    compute dW
    V = β1 V + (1 - β1) dW
    S = β2 S + (1 - β2) dW²
    V = V / (1 - β1^i)
    S = S / (1 - β2^i)
    W = W - α [V / (√S + ε)]
```

UNIT VI : Applications Perspective

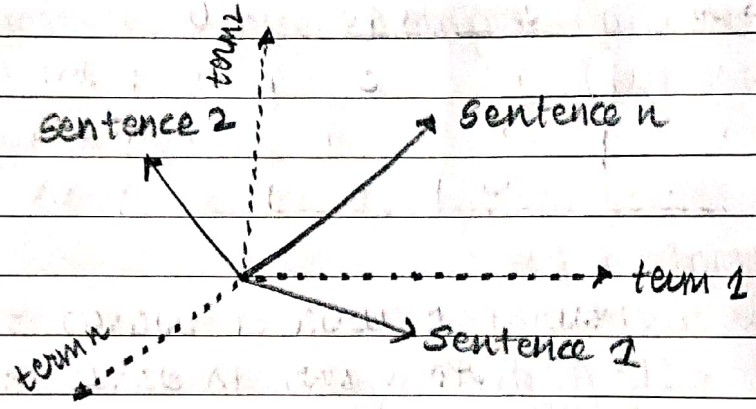
* Text Preprocessing :-

• Tokenization :

- Process of dividing text into meaningful pieces.
- Pieces are called tokens.
- Most common way of this is whitespace/unigram tokenization in which entire text is split into words by splitting them at whitespaces.
- Another type is regular expression tokenization which will use regular expression pattern to get tokens.

• Document representation :-

- In vector space model each term/word is an axis/dimension. The text/document is represented as a vector in multi-dimensional space.

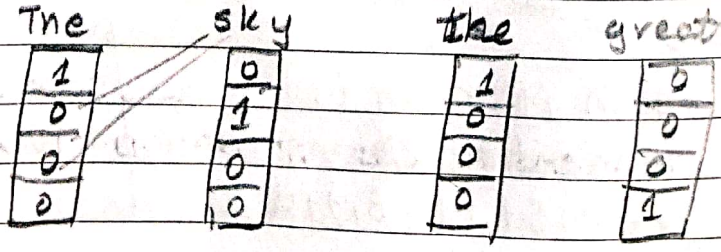


- Some ways :-

• 1-Hot encoding :-

- We represent words with one hot vectors i.e we associate each unique word with an index in the vector.

Ex :



- n-grams :- n-words sequence, we take n words from sentence and consider it as a token.
- n-gram models estimate the following word probability given previous words.
- Bag of words :
- Used in sentiment analysis uses term frequency (TF).
- These are many more...

* Feature Selection / Extraction :
- We convert the text in form of features.
- Common techniques to perform feature extraction are :-

1) Bag of Words :
- Keep frequency count of all unique words and consider it as feature.
- Steps involved :-
  i) Identify unique words from document.
  ii) Find frequencies of unique words from a single sentence.

2) Term Frequency - Inverse Document Frequency (TF-IDF):
- TF : Frequency of each word in document
- IDF : Assign lower weight to words that appear more frequently, basically depicts rarity of word in document.

$$TF = \frac{\text{Term } i \text{ frequency in document } j}{\text{Total words in document } j}$$

$$IDF = \log_2 \left(\frac{\text{Total documents}}{\text{documents with term } i} \right)$$

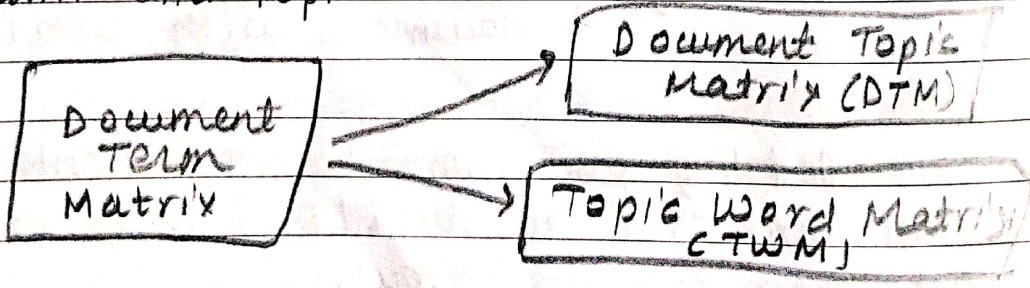
* Topic Modeling algorithm :- Latent Dirichlet Allocation.
- Latent Dirichlet Allocation (LDA) helps extract topics from a given corpus.
- LDA classifies text into a document and words per topic, these are modelled based on dirchlet distributions and processes.

- Assumptions made by LDA:
  i) Documents are a mixture of topics.
  ii) Topics are a mixture of tokens.
- Working of LDA:
  Consider a corpus [collection of Documents] has 3 documents:
  Document 1: Are you watching closely?
  Document 2: You either die a hero or live long enough to see yourself become the villan.
  Document 3: Wake up to reality. Nothing ever goes as planned in this accursed world.

D1, D2, D3 are examples of 3 documents and consider words are represented as Ws and for simplicity we consider there are 5 unique words

	W1	W2	W3	W4	W5
D1	0	0	1	1	1
D2	0	1	1	1	1
D3	1	0	0	0	0

The corpus is now a document-word matrix
LDA converts this matrix into: Document-term matrix and topic-word matrix



DTM: Possible topics
TWM: Words that the topic will contain

* Text Similarity Measures :-
- Used to find similar text.
- Google, Quora, Stackoverflow use it to find similar questions.
- Let us see some similarity measures :-

• Jaccard Similarity :
- Ratio of common words to total words

$$\text{Jaccard Similarity} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

- It does not give importance to duplication of words.

• Cosine Similarity :-

$$\text{Similarity}(A, B) = \frac{A \cdot B}{|A| \times |B|} = \frac{\sum (A_i \times B_i)}{\sqrt{\sum A_i^2} \times \sqrt{\sum B_i^2}}$$

- Suitable when words are repeated and are important.
- It is the ratio of dot product of two vectors of words to their product of magnitude.