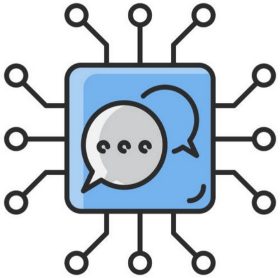
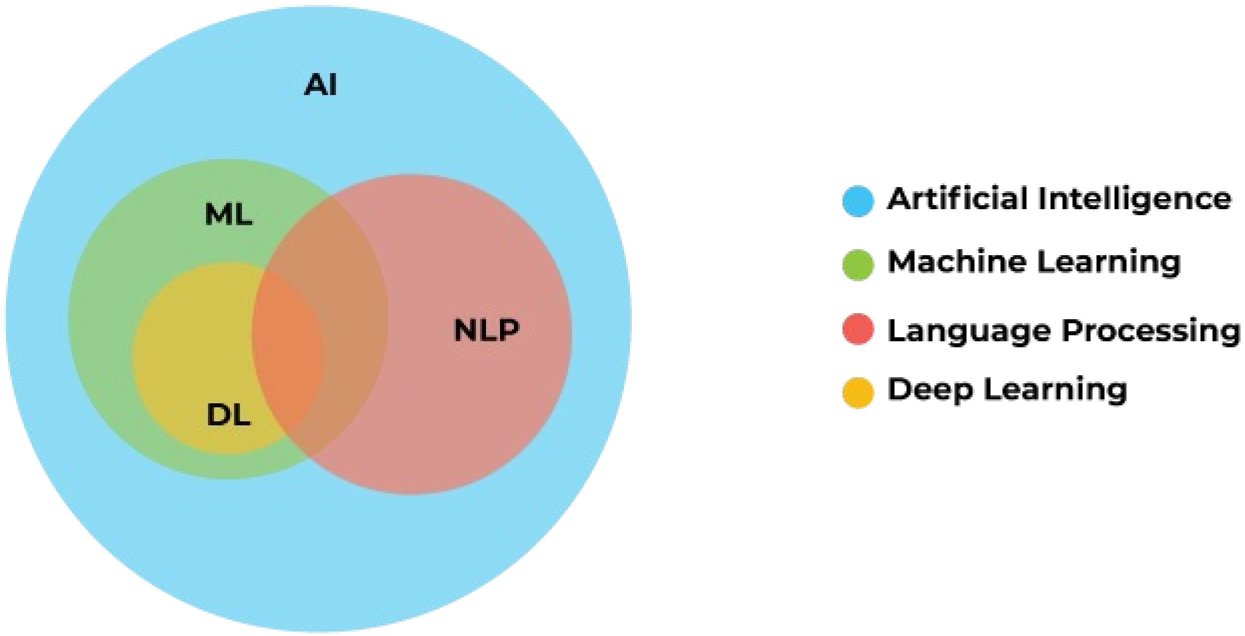
Machine Learning and Deep Learning for NLP

Tushar B. Kute,
http://tusharkute.com



Why ML & DL for NLP?



Why ML & DL for NLP?

Machine learning and Deep Learning for NLP helps data analysts turn unstructured text into usable data and insights.

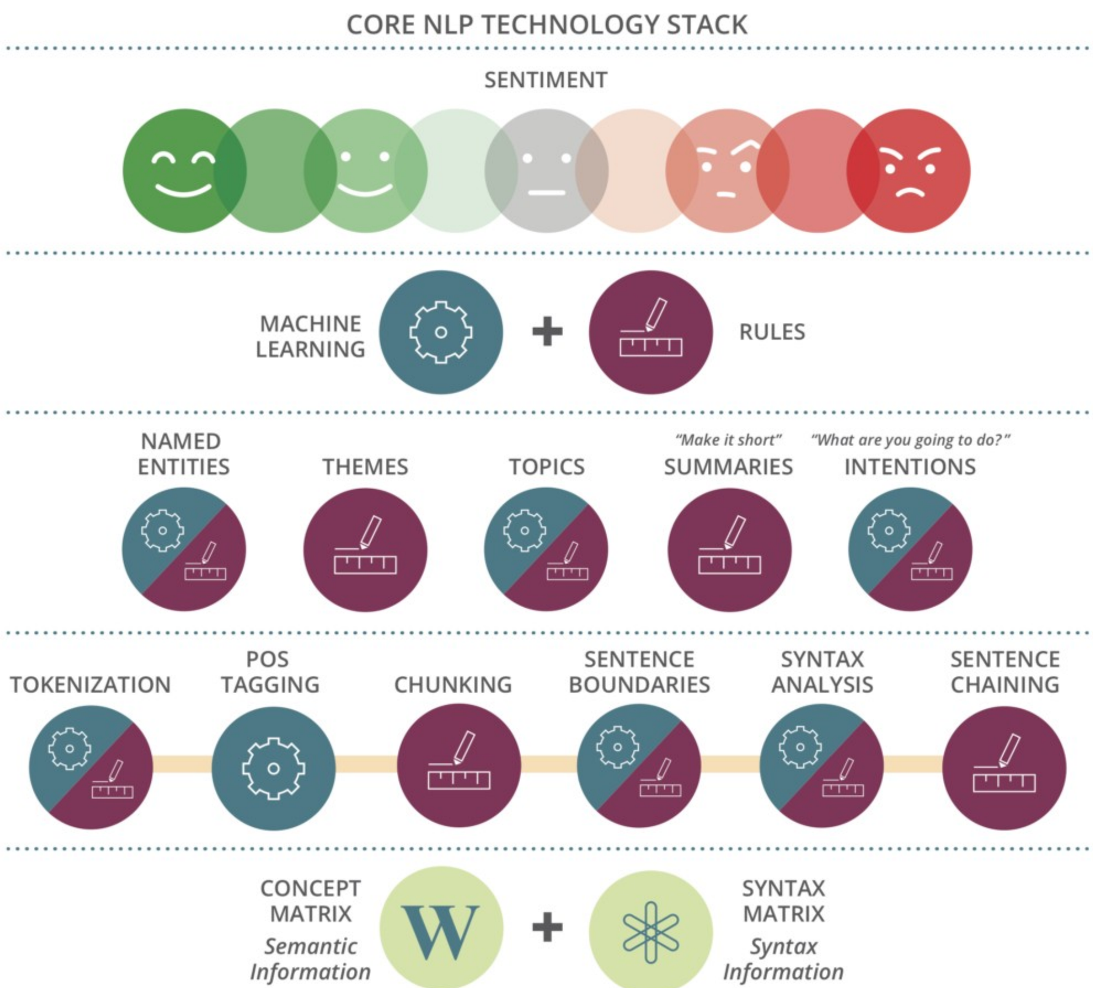
Classification

- Categorization means sorting content into buckets to get a quick, high-level overview of what's in the data.
- To train a text classification model, data scientists use pre-sorted content and gently shepherd their model until it's reached the desired level of accuracy.
- The result is accurate, reliable categorization of text documents that takes far less time and energy than human analysis.

Unsupervised Learning

- Clustering means grouping similar documents together into groups or sets.
- These clusters are then sorted based on importance and relevancy (hierarchical clustering).
- Another type of unsupervised learning is Latent Semantic Indexing (LSI). This technique identifies on words and phrases that frequently occur with each other.
- Data scientists use LSI for faceted searches, or for returning search results that aren't the exact search term.

NLP Technology Stack



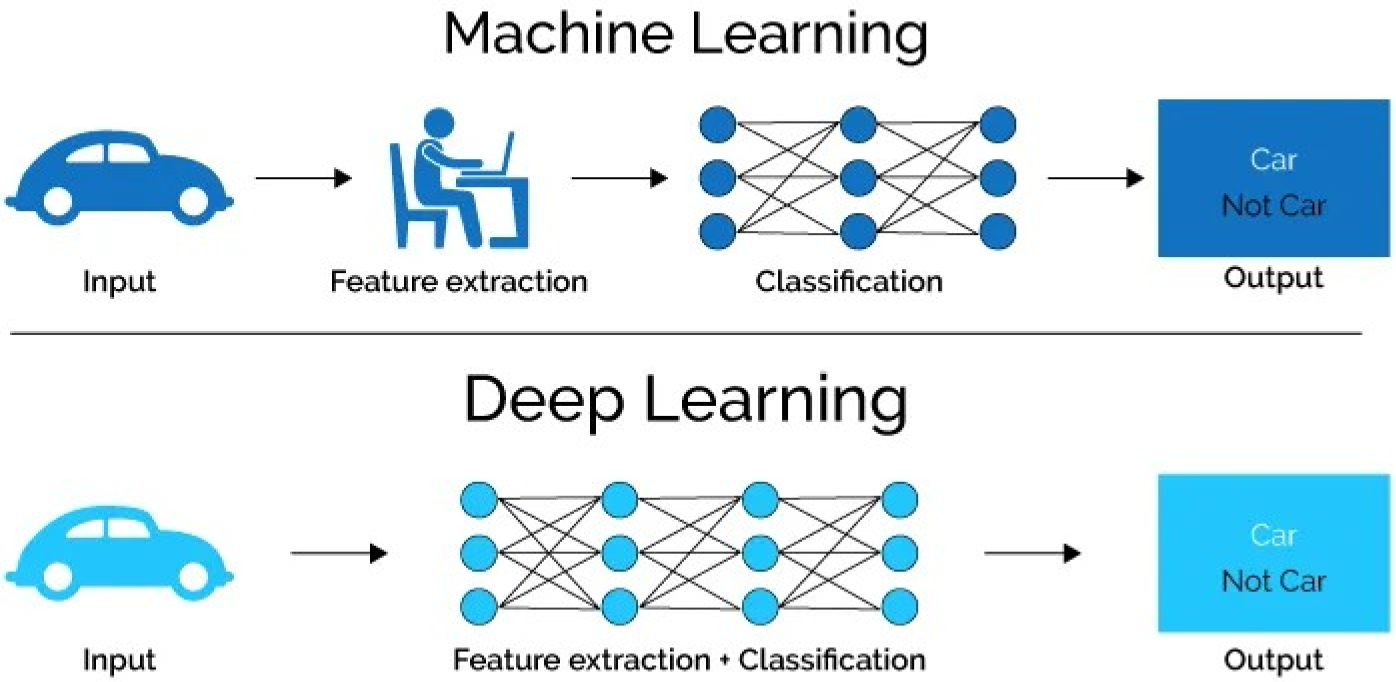
Deep Learning vs. Machine Learning

- Deep Learning can essentially do everything that machine learning does, but not the other way around.
- For instance, machine learning is useful when the dataset is small and well-curated, which means that the data is carefully preprocessed.
- Data preprocessing requires human intervention. It also means that when the dataset is large and complex, machine learning algorithms will fail to extract information, and it will underfit.

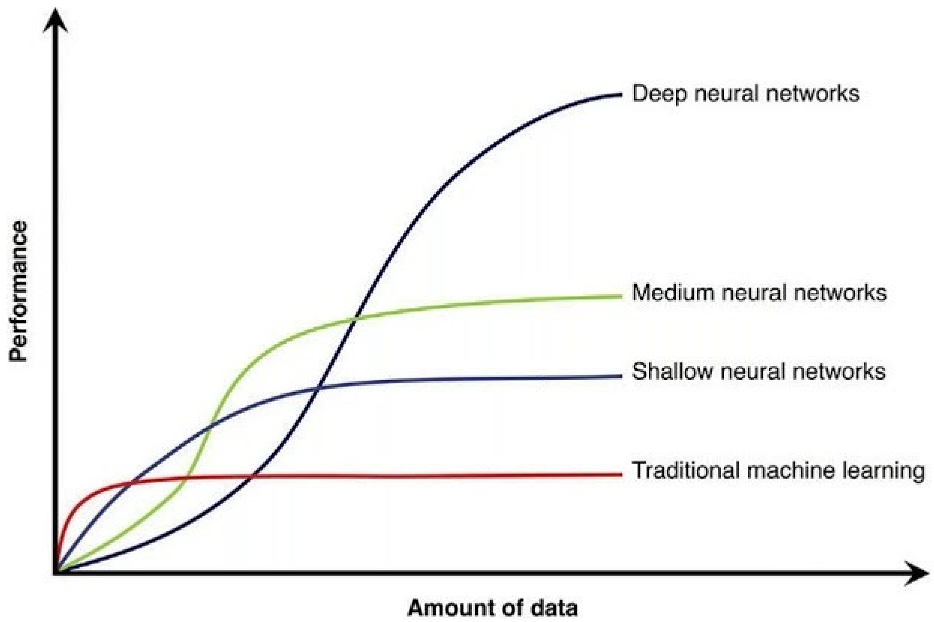
Deep Learning vs. Machine Learning

- Generally, machine learning is alternatively termed shallow learning because it is very effective for smaller datasets.
- Deep learning, on the other hand, is extremely powerful when the dataset is large.
- It can learn any complex patterns from the data and can draw accurate conclusions on its own. In fact, deep learning is so powerful that it can even process unstructured data - data that is not adequately arranged like text corpus, social media activity, etc.
- Furthermore, it can also generate new data samples and find anomalies that machine learning algorithms and human eyes can miss.

Deep Learning vs. Machine Learning



Why Deep Learning ?



Why Now?

- *Algorithm Advancements*
- *GPU Computing*
- *Availability of Larger Training Data*

Deep Learning vs. Machine Learning

- On the downside, deep learning is computationally expensive compared to machine learning, which also means that it requires a lot of time to process.
- Deep Learning and Machine Learning are both capable of different types of learning: Supervised Learning (labeled data), Unsupervised Learning (unlabeled data), and Reinforcement Learning.
- But their usefulness is usually determined by the size and complexity of the data.

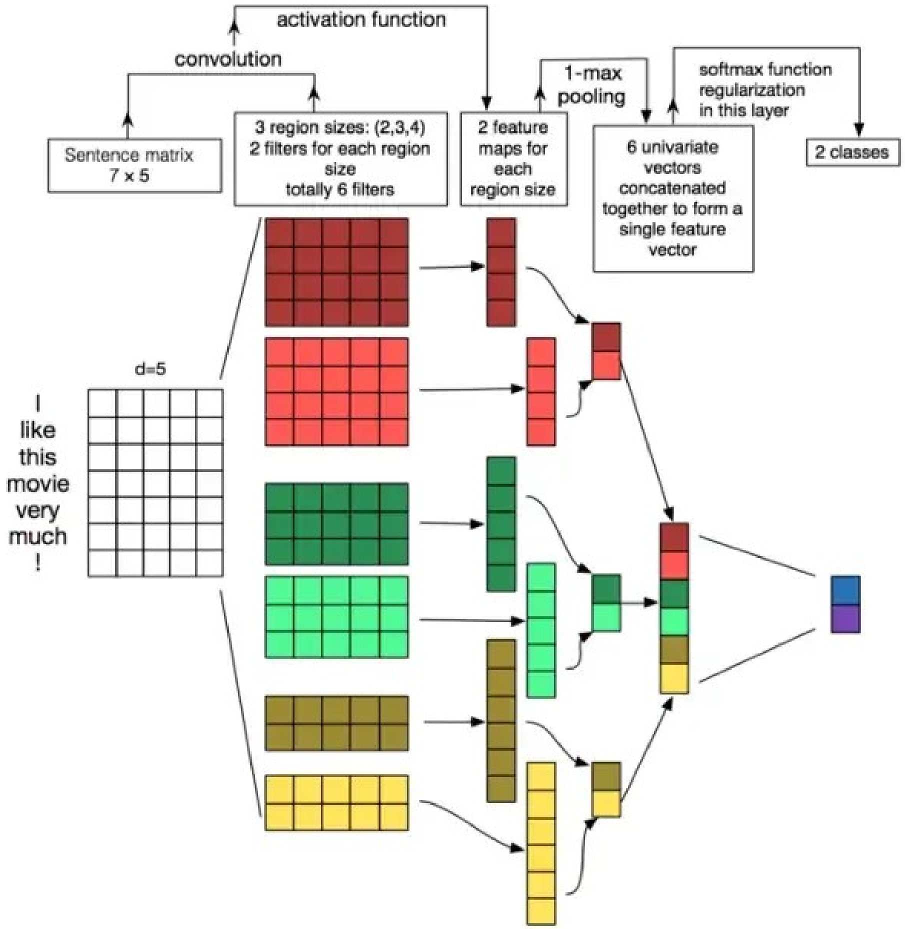
A Quick Summary

- Machine learning requires data preprocessing, which involves human intervention.
- The neural networks in deep learning are capable of extracting features; hence no human intervention is required.
- Deep Learning can process unstructured data.
- Deep Learning is usually based on representative learning i.e., finding and extracting vital information or patterns that represent the entire dataset.
- Deep learning is computationally expensive and time-consuming.

ANN and DNN

- Practical

CNN


activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 x 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I like this movie very much !

Word Embedding

- Word embeddings are a type of word representation that allows words with similar meaning to have a similar representation.
- They are a distributed representation for text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems.

Word Embedding

- Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space.
- Each word is mapped to one vector and the vector values are learned in a way that resembles a neural network, and hence the technique is often lumped into the field of deep learning. Key to the approach is the idea of using a dense distributed representation for each word.
- Each word is represented by a real-valued vector, often tens or hundreds of dimensions. This is contrasted to the thousands or millions of dimensions required for sparse word representations, such as a one hot encoding

Embedding Layer

- An embedding layer, for lack of a better name, is a word embedding that is learned jointly with a neural network model on a specific natural language processing task, such as language modeling or document classification.
- It requires that document text be cleaned and prepared such that each word is one hot encoded. The size of the vector space is specified as part of the model, such as 50, 100, or 300 dimensions.
- The vectors are initialized with small random numbers. The embedding layer is used on the front end of a neural network and is fit in a supervised way using the Backpropagation algorithm.

Embedding Layer

- The one hot encoded words are mapped to the word vectors. If a Multilayer Perceptron model is used, then the word vectors are concatenated before being fed as input to the model.
- If a recurrent neural network is used, then each word may be taken as one input in a sequence.
- This approach of learning an embedding layer requires a lot of training data and can be slow.

Embedding Layer

- if we use one-hot encoding on words in textual data, we will have a dummy feature for each word, which means 10,000 features for a vocabulary of 10,000 words.
- This is not a feasible embedding approach as it demands large storage space for the word vectors and reduces model efficiency.
- Embedding layer enables us to convert each word into a fixed length vector of defined size. The resultant vector is a dense one with having real values instead of just 0’s and 1’s.
- The fixed length of word vectors helps us to represent words in a better way along with reduced dimensions.

Example:

One-hot encoding

		cat	mat	on	sat	the
the	=>	0	0	0	0	1
cat	=>	1	0	0	0	0
sat	=>	0	0	0	1	0
...				...		

Word Embedding

- Word embeddings give us a way to use an efficient, dense representation in which similar words have a similar encoding. Importantly, you do not have to specify this encoding by hand.
- An embedding is a dense vector of floating point values (the length of the vector is a parameter you specify). Instead of specifying the values for the embedding manually, they are trainable parameters (weights learned by the model during training, in the same way a model learns weights for a dense layer).
- It is common to see word embeddings that are 8-dimensional (for small datasets), up to 1024-dimensions when working with large datasets.
- A higher dimensional embedding can capture fine-grained relationships between words, but takes more data to learn.

A 4-D Embedding

A 4-dimensional embedding

cat =>	1.2	-0.1	4.3	3.2
mat =>	0.4	2.5	-0.9	0.5
on =>	2.1	0.3	0.1	0.4
...	...			

- Above is a diagram for a word embedding. Each word is represented as a 4-dimensional vector of floating point values. Another way to think of an embedding is as "lookup table". After these weights have been learned, you can encode each word by looking up the dense vector it corresponds to in the table.

Padding Sequences

```
[1]  seq = [[3, 7], [12, 17, 19], [27, 65, 73, 81]]
```

```
[2]  import tensorflow as tf
     tf.keras.preprocessing.sequence.pad_sequences(seq)
```

```
array([[ 0,  0,  3,  7],
       [ 0, 12, 17, 19],
       [27, 65, 73, 81]], dtype=int32)
```

```
[3]  tf.keras.preprocessing.sequence.pad_sequences(seq, value = 1)
```

```
array([[ 1,  1,  3,  7],
       [ 1, 12, 17, 19],
       [27, 65, 73, 81]], dtype=int32)
```

CNN

- Practical

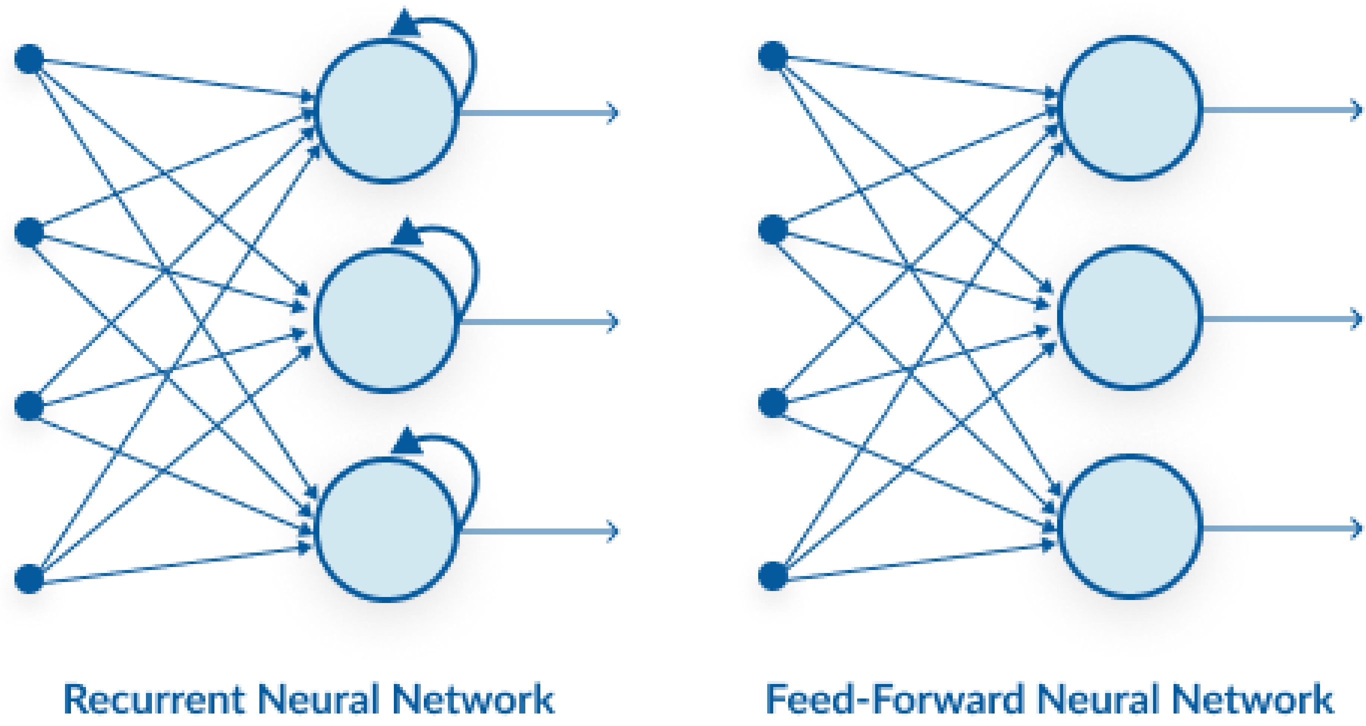
Word2Vec

- Word2Vec is a statistical method for efficiently learning a standalone word embedding from a text corpus.
- It was developed by Tomas Mikolov, et al. at Google in 2013 as a response to make the neural-network-based training of the embedding more efficient and since then has become the de facto standard for developing pre-trained word embedding.
- Additionally, the work involved analysis of the learned vectors and the exploration of vector math on the representations of words.
- For example, that subtracting the man-ness from King and adding women-ness results in the word Queen, capturing the analogy king is to queen as man is to woman.

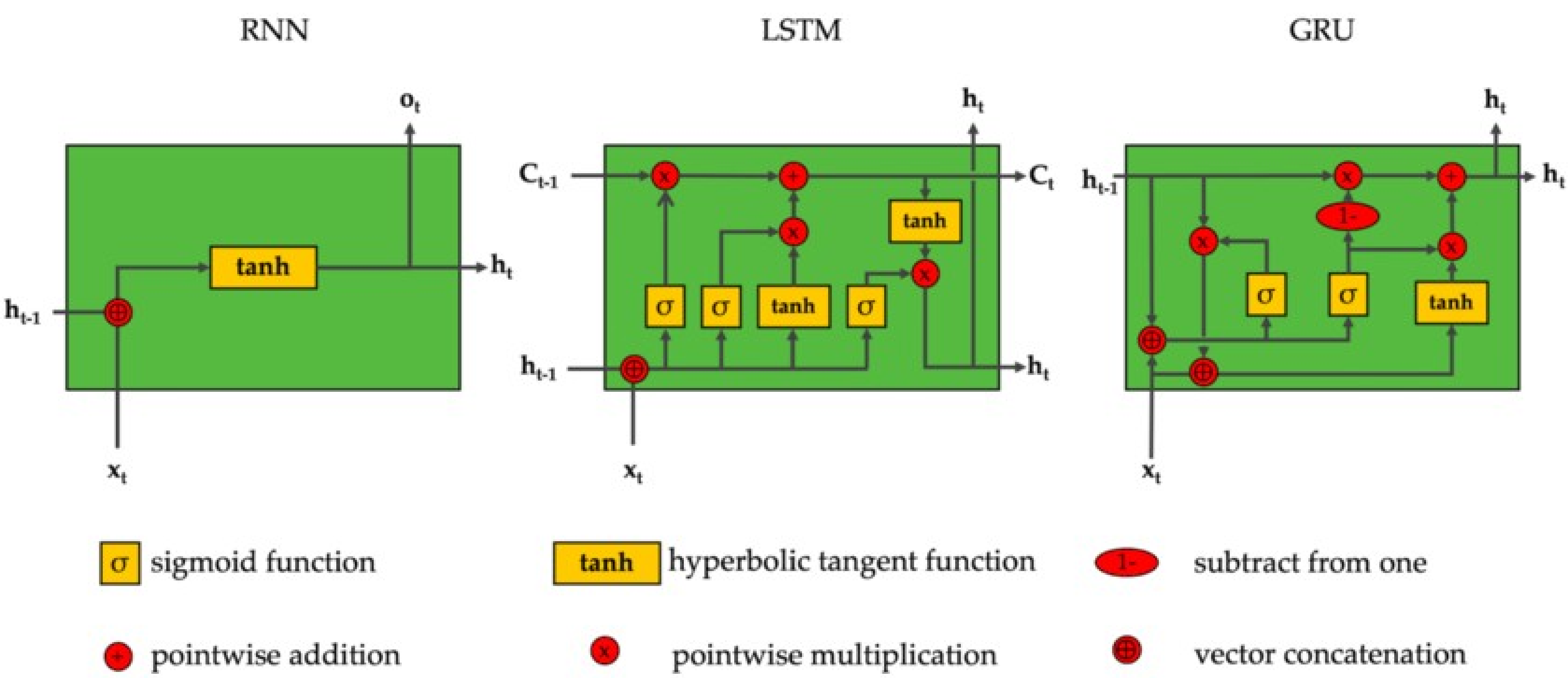
RNN

- A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes can create a cycle, allowing output from some nodes to affect subsequent input to the same nodes.
- This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs.
- This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

RNN



RNN



RNN
LSTM
GRU
o_t
h_t
h_{t-1}
x_t
tanh
C_{t-1}
C_t
σ
1-
sigmoid function
hyperbolic tangent function
subtract from one
pointwise addition
pointwise multiplication
vector concatenation

RNN

- Practical

Thank you

@mitu_skillologies

@mITuSkillologies

@mitu_group

@mitu-skillologies

@MITUSkillologies

kaggle

@mituskillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

@mituskillologies

contact@mitu.co.in

tushar@tusharkute.com