ABSTRACT

Transliteration is the process of mapping text written in one language into another by means of a pre-defined mapping. It is useful when a user knows a language but does not know how to write its script. Hindi is the lingua-franca of India. It is the most widely spoken and scripted language in India. Transliteration helps people pronounce words and names in foreign languages. Transliteration aims to only change the letters or characters of a source language into corresponding letters of the target language. It does not render meaning unlike translation, which is converts the written or spoken meanings of words or text of a source language into a target language.

Given a word in English, the model represents the same word in Devanagari script (Hindi) using two Recurrent neural networks i.e. an Encoder and a Decoder. The encoder network is that part of the network that takes the input sequence in English and maps it to an encoded representation of the sequence. The encoded representation is then used by the decoder network to generate an output sequence in Hindi. Transliteration can used effectively in the case of nouns. The accuracy of the model can be further increased with attention mechanism used in combination with Encoder-Decoder Model.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." This is Alan Turing's definition of machine learning.

Deep learning is a class of machine learning algorithms that utilizes a hierarchical level of artificial neural networks to carry out the process of machine learning. The artificial neural networks are built like the human brain, with neuron nodes connected together like a web. While traditional programs build analysis with data in a linear way, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach.

The word "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial credit assignment path (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output.

For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited.

Natural language processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyse large amounts of natural language data.

Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts.

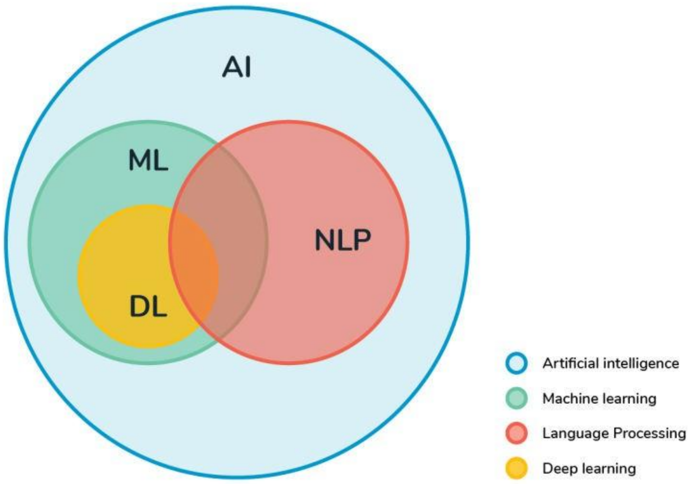

Fig. 1 : Graphical representation of relationship between various fields in artificial intelligence (source: devopedia.org)

1.1 TRANSLITERATION

Transliteration is a problem in natural language processing for transforming the text in one script into another script so as to preserve the phonetic structure of words as closely as possible. It is useful when a user knows a language but does not know how to write its script. Machine transliteration can play an important role in natural language application such as information retrieval and machine translation, especially for handling proper nouns and technical terms, cross-language applications, data mining and information retrieval system.

Transliteration aims to only change the letters or characters of a source language into corresponding letters of the target language. It does not render meaning unlike

Vector is an abstract representation of raw data that reiterates its meaning into a comprehensive form for the machine. It is a kind of text-to-machine translation of data.

The sequence can be described as a collection of data points with some defined order (usually, it is a time-based, there can also be other specific criteria involved). An example of sequence can be time series stock market data - single point shows the current price while its sequence over a certain period shows the permutations of the cost.

Unlike other types of neural networks that process data straight, where each element is processed independently of the others, recurrent neural networks keep in mind the relations between different segments of data, in more general terms, context. Given the fact that understanding of the context is critical in perception of information of any kind, this makes recurrent neural networks extremely efficient at recognizing and generating data based on patterns put into a specific context.

In essence, RNN is the network with contextual loops that enable the persistent processing of every element of the sequence with the output building upon the previous computations, which in other words, means Recurrent Neural Network enables making sense of data. Just like traditional Artificial Neural Networks, RNN consists of nodes with three distinct layers representing different stages of the operation.

The neurons are spread over the temporal scale (i.e., sequence) separated into three layers.

The layers are:

- Input layer represents information to be processed;
- A hidden layer represents the algorithms at work;
- Output layer shows the result of the operation;

A single time step of the input is supplied to the network i.e. x_t is supplied to the network. We then calculate its current state using a combination of the current input and the previous state i.e. we calculate H_t .The current H_t becomes H_{t-1} for the next time step. We can go as many time steps as the problem demands and combine the information from all the previous states. Once all the time steps are completed the final current state is used to calculate the output O_t.

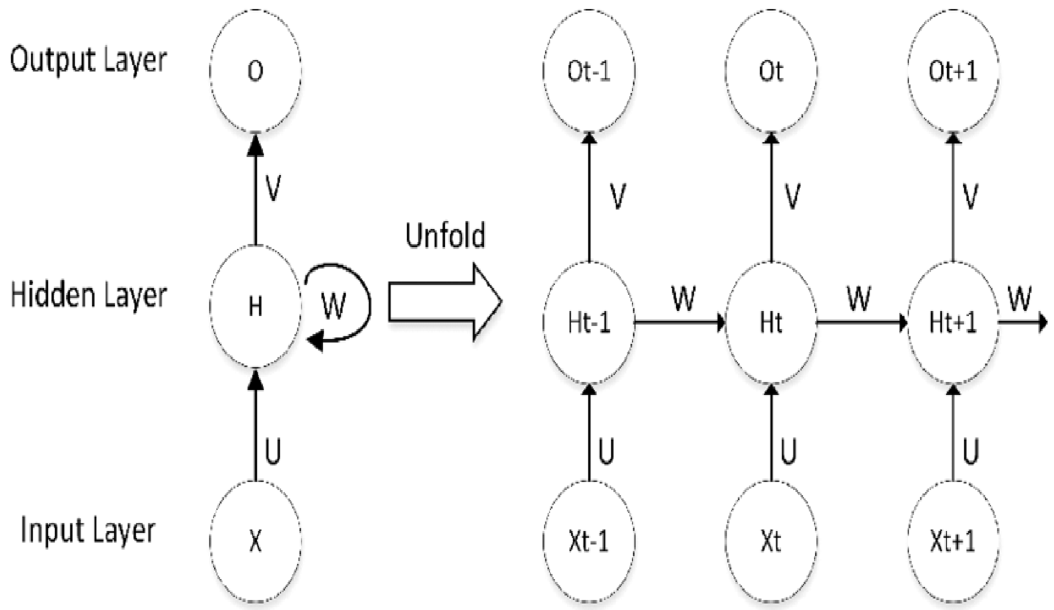

Fig 3: Architecture of Recurrent Neural Network (source: medium.com)

1.3.2.1 LONG SHORT TERM MEMORY

Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. LSTM was designed by Hochreiter & Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give efficient performance. LSTM can by default retain the information for long period of time.

It is used for processing, predicting and classifying on the basis of time series data. The core concept of LSTM's are the cell state, and it's various gates. The cell state act as a transport highway that transfers relative information all the way down the sequence chain. You can think of it as the "memory" of the network. The cell state, in theory, can carry relevant information throughout the processing of the sequence. So even information from the earlier time steps can make it's way to later time steps, reducing the effects of short-term memory. As the cell state goes on its journey, information get's added or removed to the cell state via gates. The gates are different neural networks that decide which information is allowed on the cell state. The gates can learn what information is relevant to keep or forget during training.

LSTM has a chain structure that contains four neural networks and different memory blocks called **cells**. Information is retained by the cells and the memory manipulations are done by the **gates.** There are three gates in long short-term memory Recurrent-Neural-networks–

1. **Forget Gate:** The information that no longer useful in the cell state is removed with the forget gate. Two inputs x_t (input at the particular time) and h_{t-1} (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for the output 1, the information is retained for the future use.
2. **Input gate:** Addition of useful information to the cell state is done by input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs h_{t-1} and x_t, Then, a vector is created using *tanh* function that gives output from -1 to +1, which contains all the possible values from h_{t-1} and x_t . Atlast, the values of the vector and the regulated values are multiplied to obtain the useful information.
3. **Output gate:** The task of extracting useful information from the current cell state to be presented as an output is done by output gate. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter the values to be remembered using inputs h_{t-1} and x_t . Atlast, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

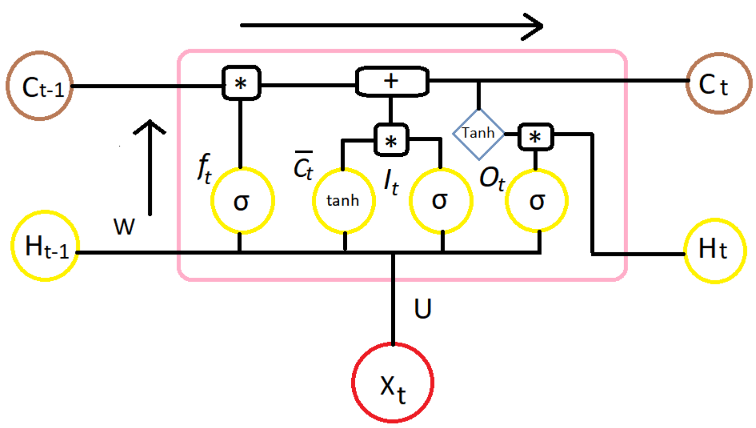

Fig 4: Architecture of LSTM cell (source: medium.com)

1.3.2.2 GATED RECURRENT UNIT NETWORKS

To solve the Vanishing-Exploding gradients problem often encountered during the operation of a basic Recurrent Neural Network, many variations were developed. One of the most famous variations is the LSTM. One of the lesser known but equally effective variations is the **Gated Recurrent Unit Network(GRU)**.

Unlike LSTM, it consists of only three gates and does not maintain an Internal Cell State. The information which is stored in the Internal Cell State in an LSTM recurrent unit is incorporated into the hidden state of the Gated Recurrent Unit. This collective information is passed onto the next Gated Recurrent Unit. The different gates of a GRU are as described below:-

1. **Update Gate(z):** It determines how much of the past knowledge needs to be passed along into the future. It is analogous to the Output Gate in an LSTM recurrent unit.
2. **Reset Gate(r):** It determines how much of the past knowledge to forget. It is analogous to the combination of the Input Gate and the Forget Gate in an LSTM recurrent unit.

The basic work-flow of a Gated Recurrent Unit Network is similar to that of a basic Recurrent Neural Network when illustrated, the main difference between the two is in the internal working within each recurrent unit as Gated Recurrent Unit networks consist of gates which modulate the current input and the previous hidden state.

GRU's has fewer tensor operations; therefore, they are a little speedier to train then LSTM's. There isn't a clear winner which one is better. Researchers and engineers usually try both to determine which one works better for their use case.

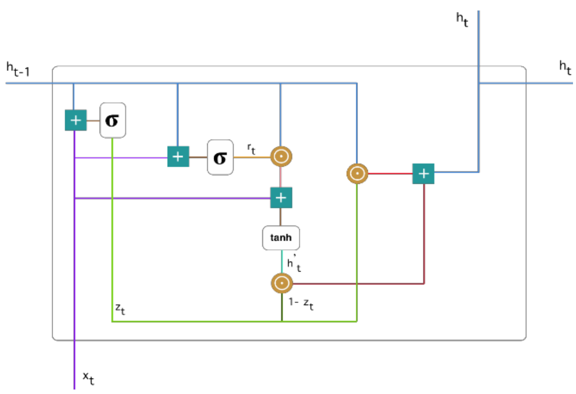

Fig 5: Architecture of Gated Recurrent Unit Network (source: medium.com)