REAL TIME SPEECH TRANSLATION USING DEEP RECURRENT LSTM

NEURAL NETWORKS

Scope of the project:

The main essence of our project is to solve the problem of communication between the people who speak different languages. Language is a dress of thought. So, if we can break the barriers between different languages then we can express our ideas and thoughts to the whole world. Our project deals with a solution to this problem that is real time speech translation which can be achieved by using Deep Learning, the subfield of Machine Learning. Machine learning is a subfield of Artificial Intelligence. In Deep learning, an artificial neural network replicates the biological nervous system where the interconnected nodes are neurons and edges (connections) are synapse. A type of recurrent neural networks called LSTM (long short term memory) networks helps us in speech translation. In case of speech recognition, we use LSTM (Long Short-Term Memory) networks which are a type of recurrent neural networks. A recurrent neural network remembers the patterns and then generates new patterns and they store memory unlike convolution networks. We can use a device which can translate the other person's words to language we know. This is a simple and convenient solution to everyone and if the device is accessible to everyone. Sometimes having a separate device makes it difficult to carry from one place to another and it becomes difficult to address more than one person at same time. Having something which could be accessed by our mobile and connect multiple people at same time makes speech translation easier. It can be used in conferences, keynotes etc where a lot of people from different communities join together.

Literature survey:

Fredrik Bredmar in his master’s thesis(2016), Speech-to-speech translation using deep learning proposed a solution that has the possibility of using an LSTM neural network model to translate speech-to-speech without the need of a text representation. That is by translating using the raw audio data directly in order to persevere the characteristics of the voice that otherwise get lost in the text transcoding part of the translation process. As part of this research he created a data set of phrases suitable for speech-to-speech translation tasks. The thesis results in a proof of concept system which needs to scale the underlying deep neural network in order to work better. This paper gave us a new idea of translating speech to speech without using text representation and also the idea of preserving the important information about the characteristics of the voice such as the emotion, pitch and accent added new features to our project.

Kostadinov,S.(Dec 12,2017) in his medium article, How Recurrent Neural Networks work discussed the various concepts like how RNN works and how to train them. This helped us in understanding the working of Recurrent neural networks, their implementation and limitations. This article also gives us insight about different softwares, products that work on the similar line.

Lifa Sun, Shiyin Kang, Kun Li, Helen Meng in their paper, voice conversion using deep Bidirectional Long Short-Term Memory based Recurrent Neural Networks proposed a solution that uses Deep Bidirectional Long Short-Term Memory based Recurrent Neural Networks (DB LSTM-RNNs) for voice conversion. They proposed a model to improve the naturalness and continuity of the speech output in voice conversion, a sequence-based conversion method using DBLSTM-RNNs to model not only the frame-wised relationship between the source and the target voice, but also the long-range context-dependencies in the acoustic trajectory. This paper gave us the idea of using bidirectional LSTM and we would like to use it in our project too.

Methodology:

For designing real time speech translators, we are using Long short term memory neural networks. Generally, neural networks have the ability to learn from the training data, they organize the set of instructions by themselves. LSTM networks learn by recognizing patterns and they understand the patterns from the training data and then predict output based on the learnt patterns. The way we learn a new language by understanding its rules and regulations but we don't train our networks with rules. We take training data as different phrases, sentences and commonly used words of different languages that are widely spoken across the world and then we train our network on this data. The more training data we give to it, the more accurately it translates. While training the network, we use data of different voices with various accents. This helps to avoid overfitting. There are multiple layers of LSTM networks with encoder, decoder, and attention modules. Encoder network processes the input whereas the decoder network generates the output. We give vast amounts of data in different languages and we train our neural network. When we train our network we divide each word into word pieces and now when we give any sentence as input, the encoder layer divides each word into word pieces and then passes it to the next layer. In the attention module those pieces of words will be processed and at last decoder will give the output based on output with highest probability at each layer.

Requirement analysis:

This model needs a dataset of different phrases, sentences and commonly used words of different languages that are widely spoken across the world. An ideal dataset should have a large set of phrases where each phrase is available in both English and other translating language, phrases of variable length, a natural flow in the speech, as in day-to-day conversation. Inorder to reduce size and fit in the above points of ideal dataset, we can use a movie dialogues which is dubbed in two different language as a dataset. Since, the dataset is very large, computing systems with GPUs are generally required. So, we are trying to use cloud gpu and also reduce the scale of translation to two to three languages. To build these LSTM networks, we can use Tensorflow open source library released by Google. One main advantage with Tensorflow is the ease of assigning the network computations to GPUs and the use of GPU processing is necessary for networks of this scale.

Deliverables:

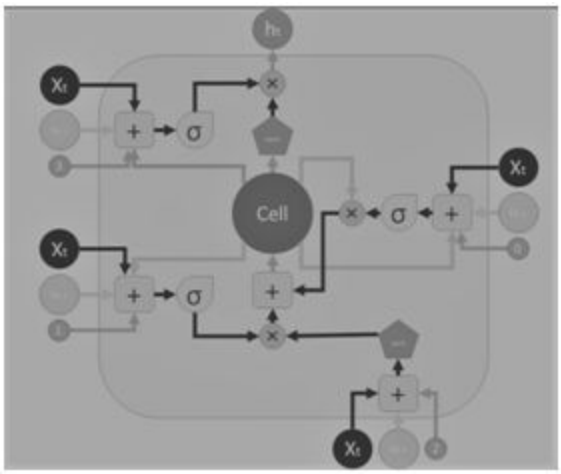
The proposed model helps in real time speech translation from one language to another language. Translation systems rely heavily on the text representation in order to translate. In our model, we are translating using the raw audio data directly in order to persevere the characteristics of the voice that otherwise get lost in the text transcoding part of the translation process. This model helps us bring speech translation more near to the real voice. Thus, This model can help to break the language barriers in the modern world and make communication easy.

Project Design:

The main aim of the project is to solve the problem of communication in different foreign languages. Real time speech translation is the goal of the project. User speaks in the language of his choice and the trained model recognizes the voice through microphone and immediately LSTM networks translate into another language and gives the output of translated input. Right now, we used some google apis and existing translators to design a model which starts with a prompt word such as “hello” and then takes input sentences through the microphone of the laptop and stores the translated output in an mp3 file in the system. For further improvisation of this project we would like to develop a model which translates in real time.

Algorithm:

In this project we are using LSTM(Long Short Term Memory) networks,
They are a special kind of recurrent neural networks which has a capability of remembering what has happened in the previous node for a long time. LSTM networks have four interactive layers and they perform different tasks. They are also composed of three gates input gate, output gate, forget gate and cell. Each of these three gates are short conventional neural networks and they also have activation functions like hyperbolic tangent and sigmoid functions. These gates regulate the flow of data.



This LSTM network has a forget valve, new memory valve, output valve and a cell. We provide each valve with three types of inputs.

h_t1 is output of previous block, x_t is input of this block, c_t1 is memory from previous block, bias we add to regularize in order avoid the problem of overfitting. Here, the plus (+) symbol represents element wise summation and concatenation and cross(x) represent element wise multiplication. The first valve is forget valve it regulates the flow of memory from previous block by multiplying either 0 or 1.h_t1, x_t, c_t1 are added then passed through sigmoid activation function later multiplied with memory from forget valve.Next valve is memory valve in which give same inputs as forget valve and pass through hyperbolic tangent activation this is new memory and memory from cell, h_t1, x_t, c_t1 are summed up now, both new memory and old memory are multiplied then added. Thus memory is regulated.

The final valve is the output valve. The outputs of previous valves are summed up and memory from the cell, activated memory is multiplied to predict the output.

1. There are eight layers of LSTM networks with encoder, decoder, and attention modules.
2. Encoder network processes the input whereas the decoder network generates the output.
3. We give vast amounts of data in different languages and we train our neural network.
4. When we train our network we divide each word into word pieces and now when we give any sentence as input, the encoder layer divides each word into word pieces and then passes it to the next layer.
5. Let (X,Y) be a source and target sentence pair. X=x1, x2, x2….be the different word pieces of the sentence in source sentence and Y=y1, y2, y3….be different word pieces in target sentence.
6. Now each word piece is made as a fixed size vector in the encoder.
7. Then using the chain rule the conditional probability of the sequence P(Y |X),network will find the highest probable word in the target.
8. After that decoder RNN network produces a hidden state yi for the next symbol to be predicted, which then goes through the softmax layer to generate a probability distribution over candidate output symbols.
9. Among the generated outputs only outputs which have highest probability will be active in the next layer of LSTM networks and this process continues over all layers and at last the word will be translated.
10. Generally, these recurrent networks are bidirectional so translation can be achieved easily.Thus LSTM networks translate sentences from one language to another.

Implementation:

Generally, neural networks have the ability to learn from the training data, they organize the set of instructions by themselves. LSTM networks learn by recognizing patterns and they understand the patterns from the training data and then predict output based on the learnt patterns.

The way we learn a new language by understanding its rules and regulations but we don’t train our networks with rules. We take training data as different phrases, sentences and commonly used words of different languages that are widely spoken across the world and then we train our network on this data. The more training data we give to it, the more accurately it translates.

We use some set of mathematical equations to train our data they are:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t tanh(c_t)
\end{aligned}
$$

Here,

W are the different weights which are randomly taken to regulate the path of inputs. Actually, training a network is more about finding the weights which are accurate that minimize the cost functions and predict correct outputs.

b are the biases which we add along with inputs to regularize the outputs in order to avoid overfitting problems. Over fitting is a problem we face when our network memorizes every input and output that are in our training data and predicts exactly the same output values when we take the same inputs again as in our training data. So our network sadly fails when we try to give new inputs which are not in our training data. In order to avoid this problem we add bias along with our inputs so that we could improvise prediction ability of our networks.

We give different words as inputs and then the weights across the edges are put into matrix then they are squashed by functions like sigmoid and hyperbolic tangent. Then we will minimize the cost function with respect weights and we will find a minimum with gradient descent method, here we face problems of exploding and vanishing gradient problems on doing back propagation. We enumerate each word in input by mapping characters in word with a number. Then we have to reshape these characters to fit the LSTM network. In order to do this we have to create one hot encoding to each character as a vector. In one hot encoding, as each unique character is mapped to a unique number, the number becomes index and only that index is represented as 1 and the rest of them are zeros. This is the language of the machine so it could easily understand and later the output is optimized by using a softmax function which converts every number in the output vector to numbers which add to make one. So, they are the same as probabilities and one with the highest probability is our output.

To build these networks we are planning to use an open source library released by Google . The Tensorflow library contains functions to build and run neural networks as smooth and seamless as possible. Since it is well documented and has a good set of examples to build from, including a sequence to sequence example, it became a good choice for this project.

References

1. Bredmar, F.(2017). *Speech-to-speech translation using deep learning*.(Master's Thesis, University of Gothenburg, Gothenburg, Sweden).Retrieved from https://gupea.ub.gu.se/bitstream/2077/51978/1/gupea_2077_51978_1.pdf

2. Kostadinov,S.(Dec 12,2017). *How Recurrent Neural Networks work.* Retrieved from https://towardsdatascience.com/learn-how-recurrent-neural-networks-work-84e975feaaf7

3. Marzouk, Z.(Aug 01, 2017). *Speakeasy: How neural networks are transforming the world of translation.*Retrieved from https://towardsdatascience.com/learn-how-recurrent-neural-networks-work-84e975feaaf7

4. Sun,L.,Kang,S.,& Meng,H.(2015). Voice conversion using deep bidirectional long short-term memory based recurrent neural networks. (Human-Computer Communications Laboratory Department of System Engineering and Engineering Management The Chinese University of Hong Kong, Hong Kong SAR, China).Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.699.5280&rep=rep1&type=pdf

5. Satoshi.(April 31, 2009). *Overcoming the Language Barrier with Speech Translation Technology.* Nakamura in Science & Technology Trends- Quarterly Review.