# Neural Machine Translation of Indian Languages

Karthik Revanuru Kaushik Turlapty Shrisha Rao

IIIT Bangalore

ACM Compute, 2017

# Agenda

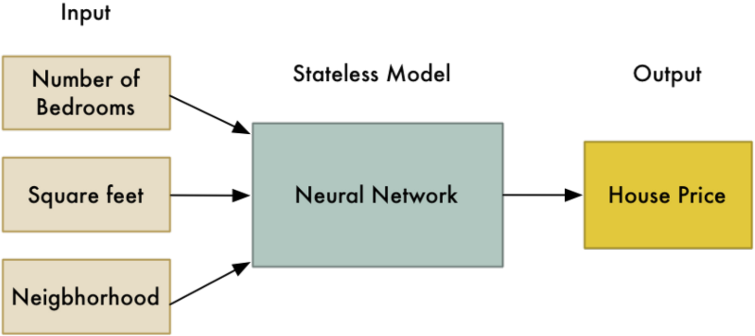
- Introduction
- Neural Networks(Basic Idea)
- Machine Translation using Neural Networks
- Results
- Conclusion

# Introduction

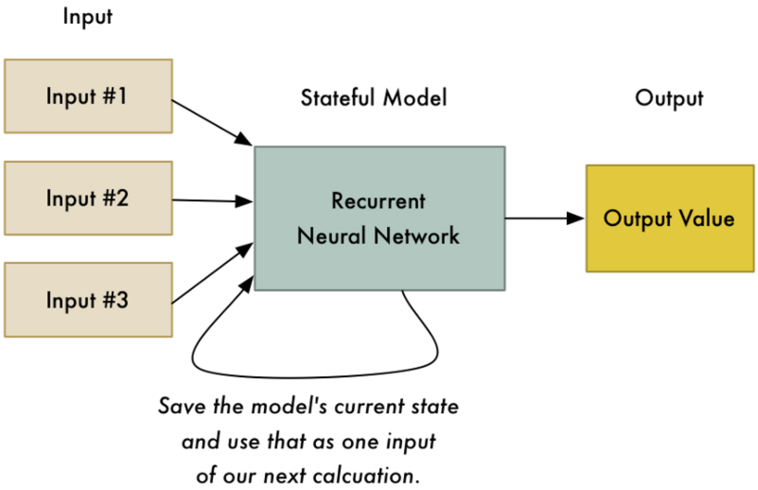
- Machine Translation (MT) is translation of a natural language content from one language to the other.
- It is one of the key areas of Natural language processing (NLP).
- Machine translation is very important to break the language barrier among the people.
- The goal is to develop an automated system without any kind of human intervention.

# Neural Network

- Takes in a list of numbers and calculates a result (based on previous training).



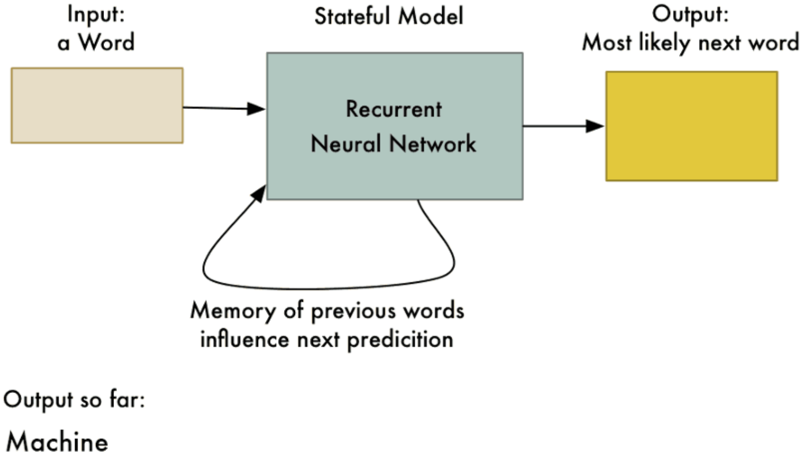
# Recurrent Neural Network




- Tweaked version of a neural network where the previous state of the neural network is one of the inputs to the next calculation.

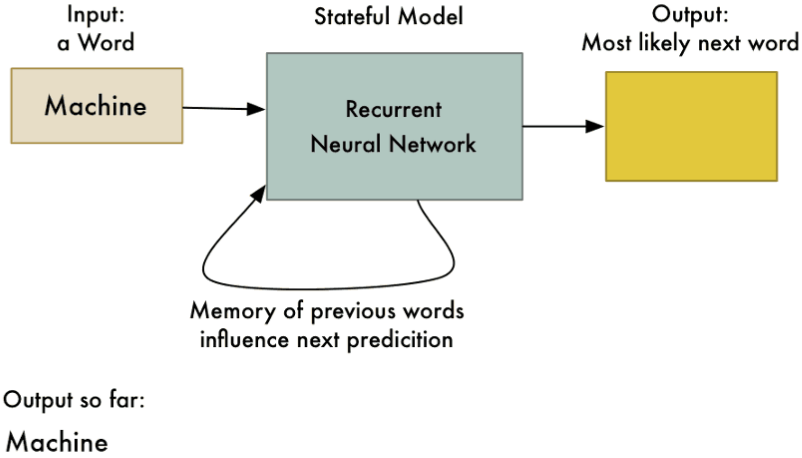
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



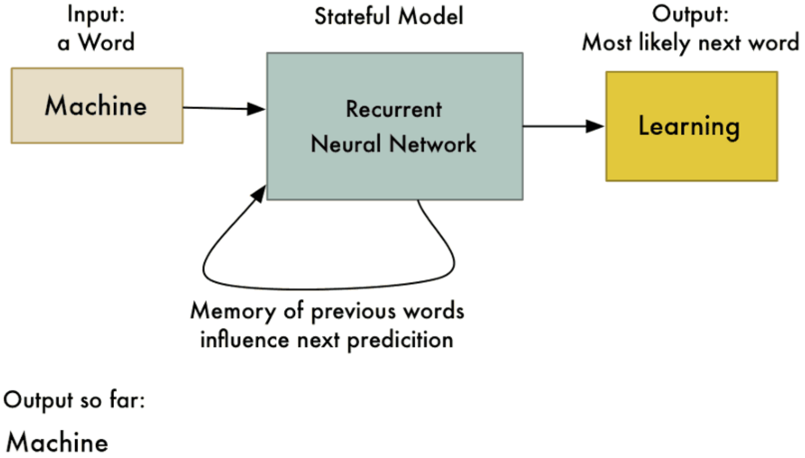
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



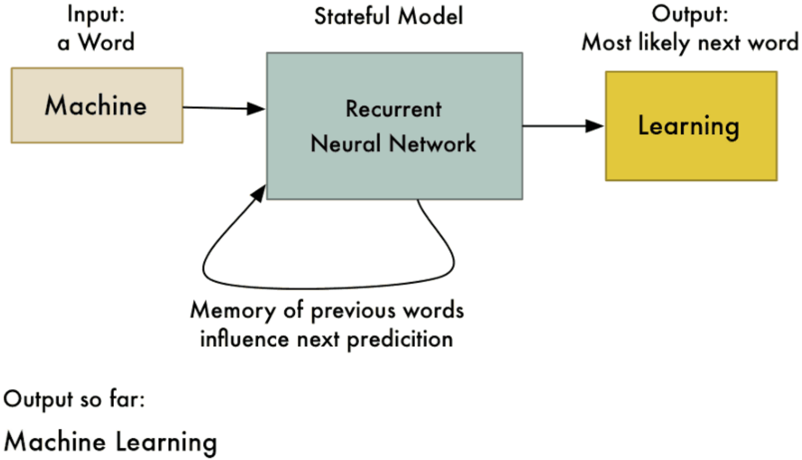
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



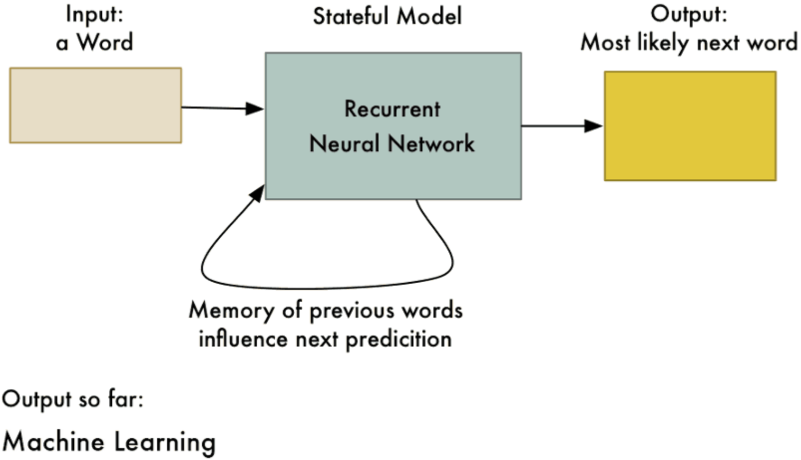
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



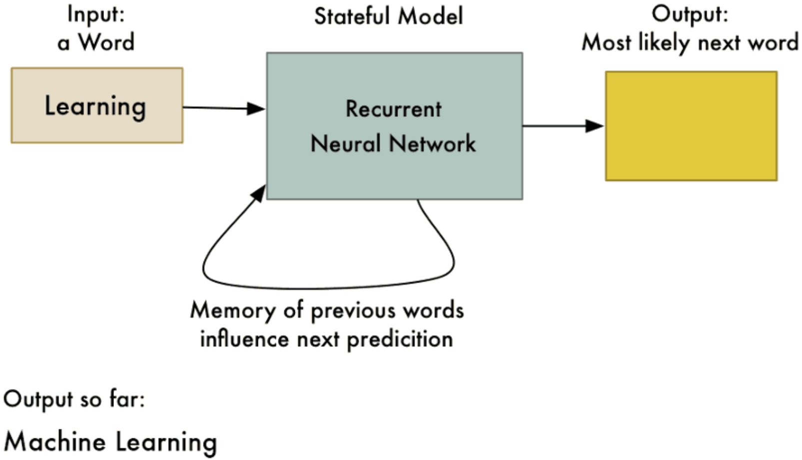
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



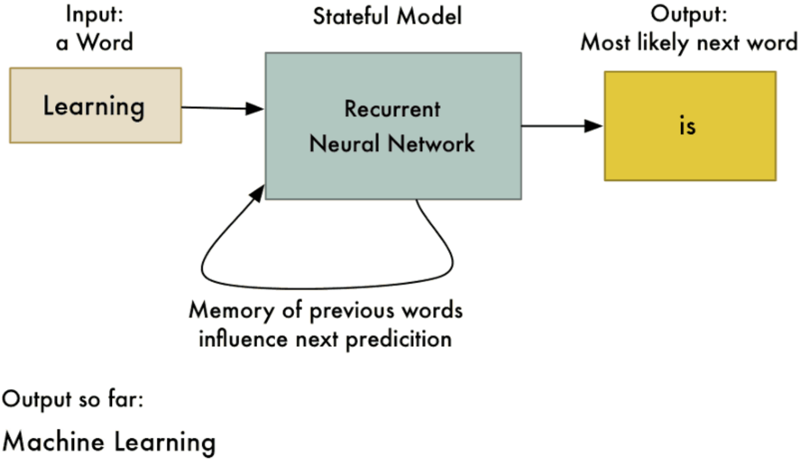
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



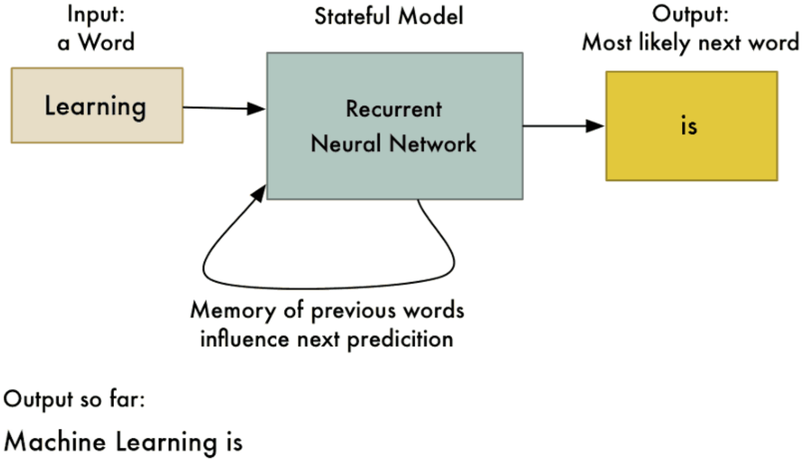
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



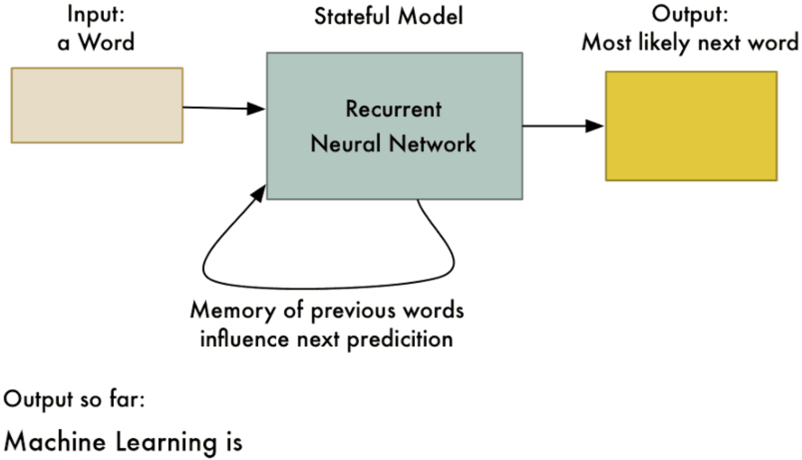
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



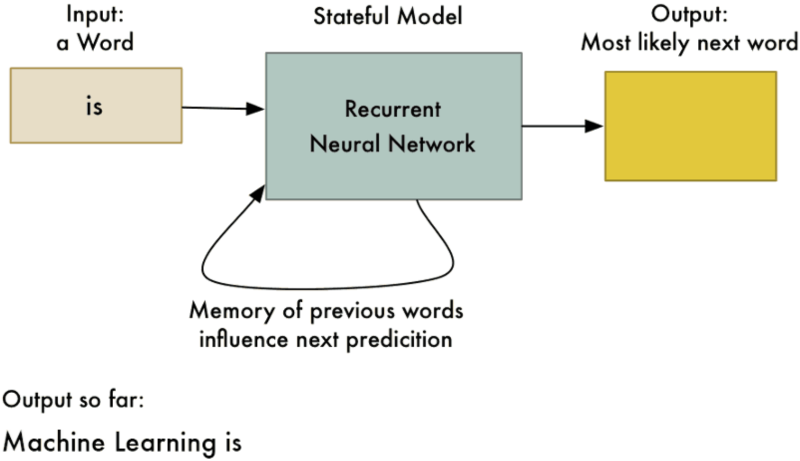
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



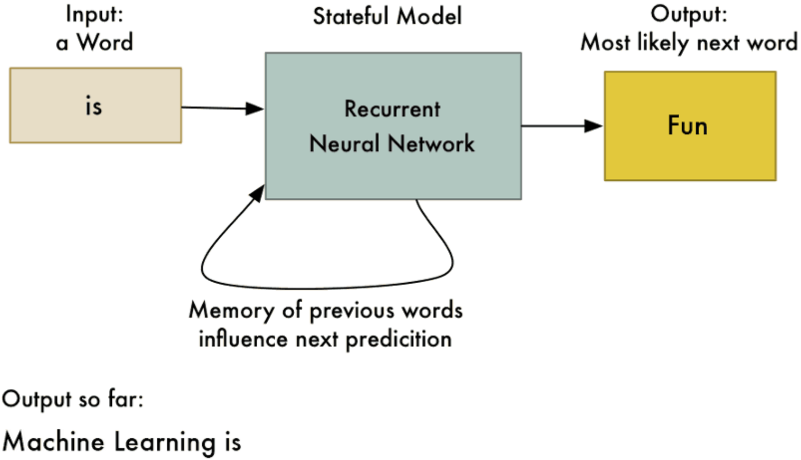
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



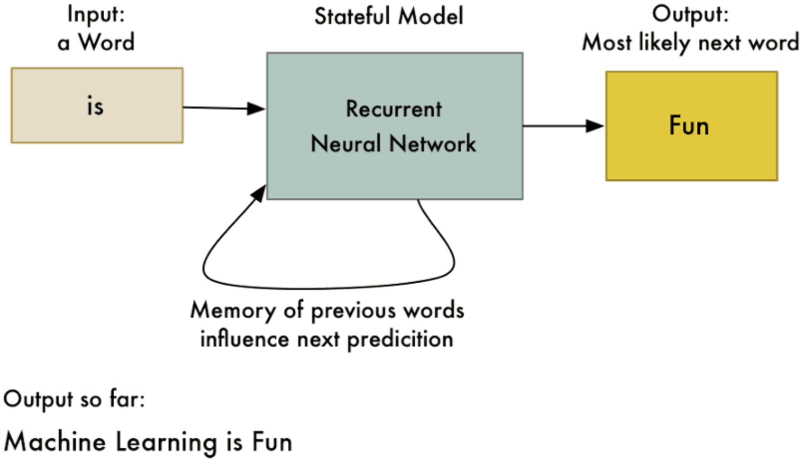
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



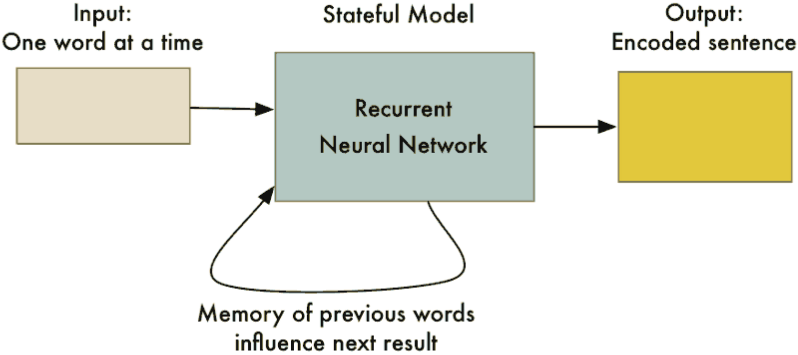
# Recurrent Neural Network

- This trick allows neural networks to learn patterns in a sequence of data.



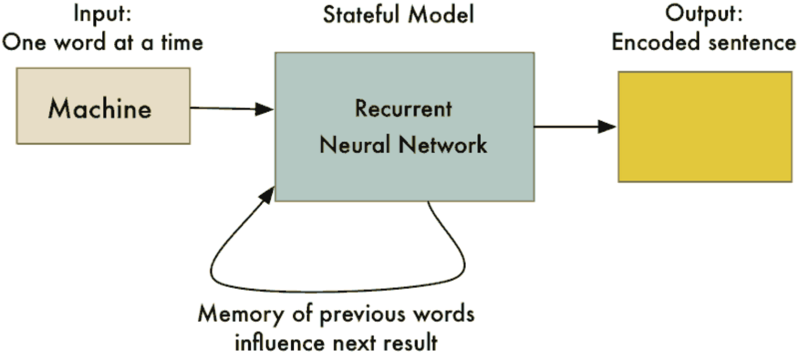
# How to Encode

- We'll feed the sentence into the RNN, one word at time.
- The final result after the last word is processed will be the values that represent the entire sentence.



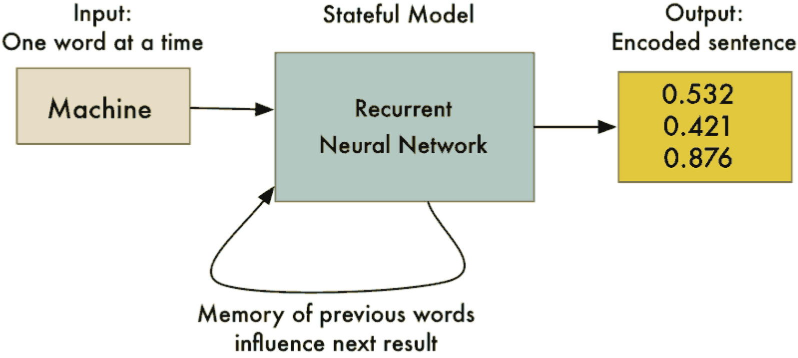
# How to Encode

- We'll feed the sentence into the RNN, one word at time.
- The final result after the last word is processed will be the values that represent the entire sentence.



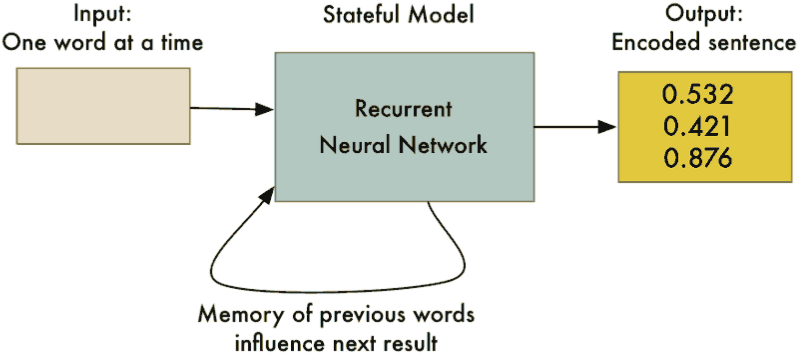
# How to Encode

- We'll feed the sentence into the RNN, one word at time.
- The final result after the last word is processed will be the values that represent the entire sentence.



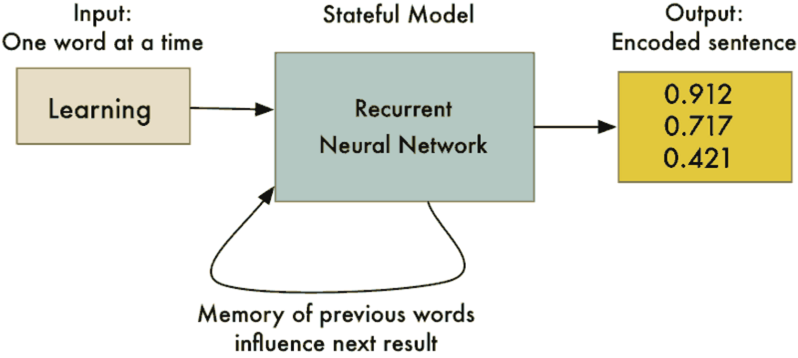
# How to Encode

- We'll feed the sentence into the RNN, one word at time.
- The final result after the last word is processed will be the values that represent the entire sentence.



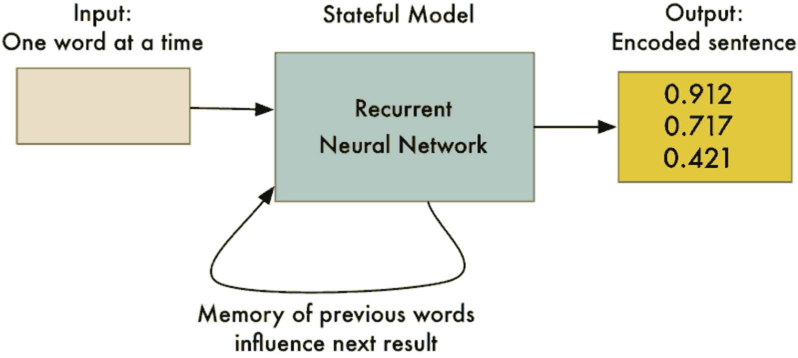
# How to Encode

- We'll feed the sentence into the RNN, one word at time.
- The final result after the last word is processed will be the values that represent the entire sentence.



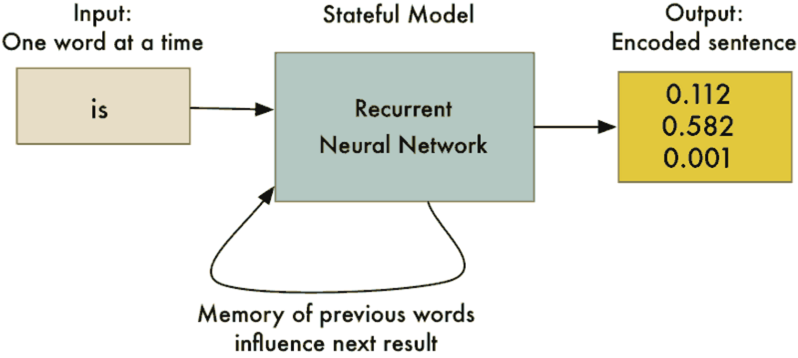
# How to Encode

- We'll feed the sentence into the RNN, one word at time.
- The final result after the last word is processed will be the values that represent the entire sentence.



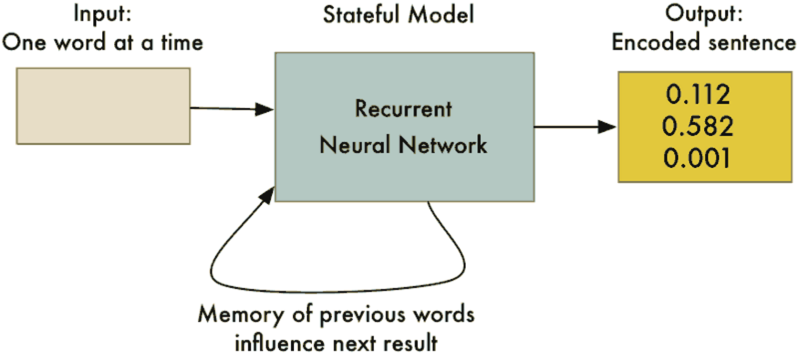
# How to Encode

- We'll feed the sentence into the RNN, one word at time.
- The final result after the last word is processed will be the values that represent the entire sentence.



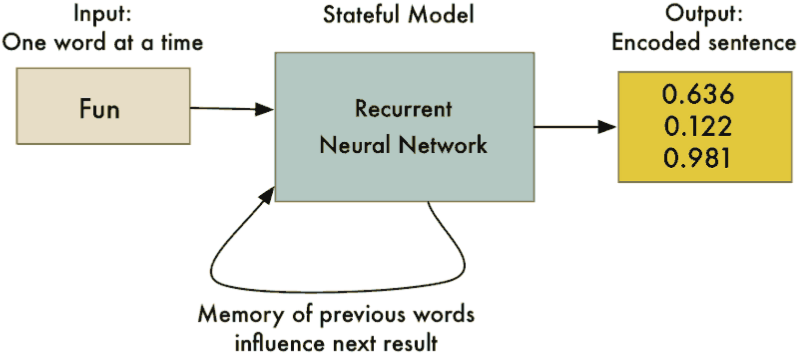
# How to Encode

- We'll feed the sentence into the RNN, one word at time.
- The final result after the last word is processed will be the values that represent the entire sentence.



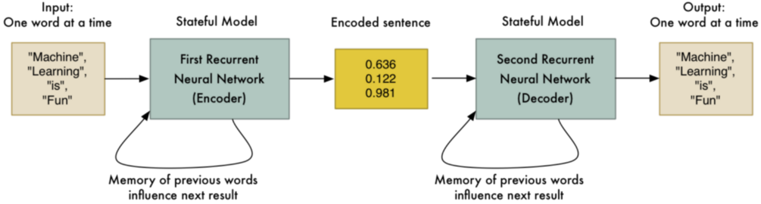
# How to Encode

- We'll feed the sentence into the RNN, one word at time.
- The final result after the last word is processed will be the values that represent the entire sentence.



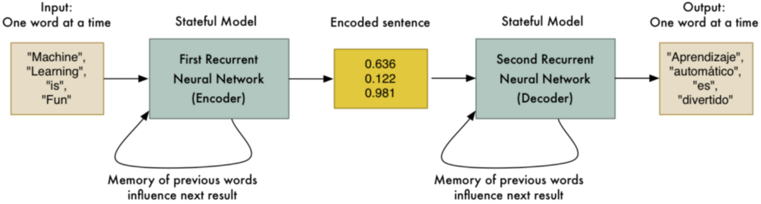
# Lets Translate!

- We know how to use an RNN to encode a sentence into a set of unique numbers.
- What if we take two RNNs and hooked them up end-to-end ?



# Big Idea!

- What if (and heres the big idea!) we could train the second RNN to decode the sentence into some-other language instead of English ?
- We could use our parallel corpora training data to train it.



# Long Term Dependency Problem

- The Clouds are in the **Sky**
- **I** grew up in AP. **I** speak fluent **Telugu**.
- Long Short Term Memory **LSTM**

# Telugu to Hindi Evaluation

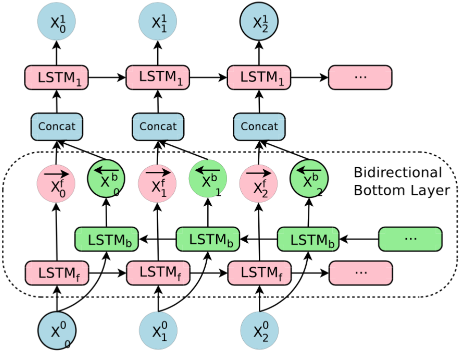
- **Dataset[1] Description**
  Training data = 73180
  Validation data = 10000
  Test data = 10000
- **RNN ( RNN_Size = 500, layers = 2, SGD)**
  BLEU score = 9.90

[1] Dataset is provided by TDIL DC

# Tamil to Hindi Evaluation

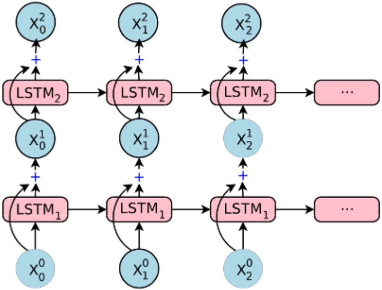
- **Dataset Description**
  Training data = 18000
  Validation data = 4556
  Test data = 2000
- **RNN ( RNN_Size = 500, layers = 2, SGD, 50 epochs)**
  BLEU score = 9.06

# Bidirectional encoder



# Residual connections

- Input of a layer is element-wise added to the output before feeding to the next layer.



# Attention Model

- **Motivation** Problem of Long Term dependencies
- **Basic Idea** Instead of single vector representation for each sentence keep around vectors for every word in the input sentence.

# Telugu to Hindi Evaluation

- **Dataset Description**
  Training data = 73180
  Validation data = 10000
  Test data = 10000
- **BRNN with Residual Connections and Attention Model**
  BLEU score = 14.6

# Telugu to Hindi Evaluation

| Model | BLEU Score |
|---|---|
| 2 Layer LSTM + SGD | 12.67 |
| 4 Layer LSTM + SGD | 4.94 |
| 2 Layer (Bi-dir) LSTM +SGD + Res + Attention | 13.89 |
| 4 layers (Bi-dir) LSTM +SGD+ Res + Attention | 14.16 |

# Extension to other language pairs

| Language Pair | Our Best Model | Google Translate[2] |
|---|---|---|
| Punjabi-Hindi | 46.47 | 17.46 |
| Gujarati-Hindi | 35.69 | 4.87 |
| Urdu-Hindi | 22.47 | 5.79 |
| Tamil-Hindi | 7.56 | 2.65 |

Table: Comparison with Google Translate

[2] Accessed on 15-06-2017

# Conclusion

- This is the first work to apply NMT on Indian Language Pairs.
- We have trained our models using eight different configurations, and evaluated them using five different standard evaluation metrics.
- Our models are easier to train than deeper models, as they have a simpler architecture, require fewer resources, and take less time.
- Our models have consistently outperformed Google Translate.

# Future Work

- Cloud based applications.
- Federated learning - Offline Translation.
- Zero Shot Translation.

# Thank you !