RECURRENT NEURAL NETWORKS (RNN)

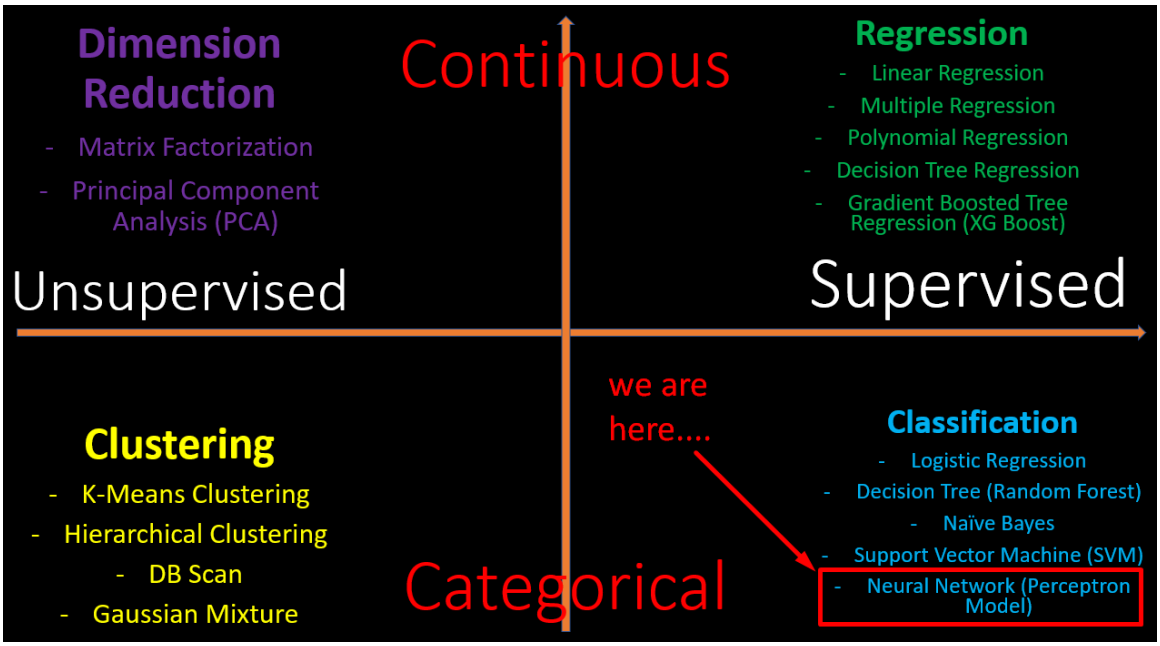
HOW IT WORKS

DR. ALVIN ANG

COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

CONTENTS

COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG



- Above is a table categorizing the different Machine Learning algorithms.
- Objective of Neural Network is to predict a CATEGORY.

COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

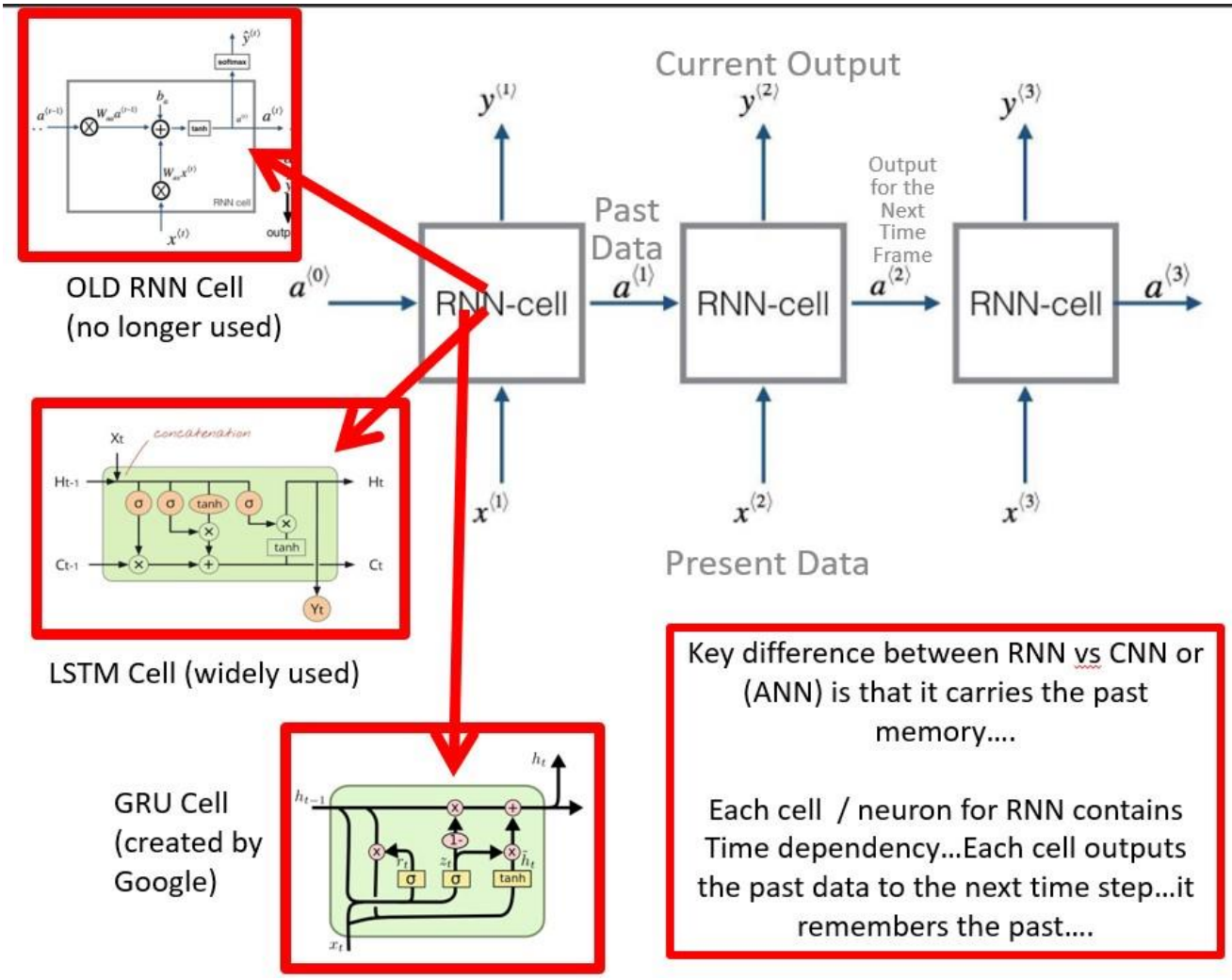
II. WHY RNN?

A. KEY DIFFERENCES BETWEEN CNNS RNN

- CNN is only good for predicting images.
- It cannot predict
    - Word Sentences (what the person is about to say)
    - Names in Context (is apple a computer? Or a fruit?)
    - Music (create new melodies!)
- Thus, we need RNNs.

COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

B. RNN ARCHITECTURE



- Instead of using the Traditional Perceptron (per Cell)…. (the summation of Weights followed by Activation Function of Sigmoid, ReLu, tanh)…..
- People invented new Cells (the most widely used is the LSTM).

COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

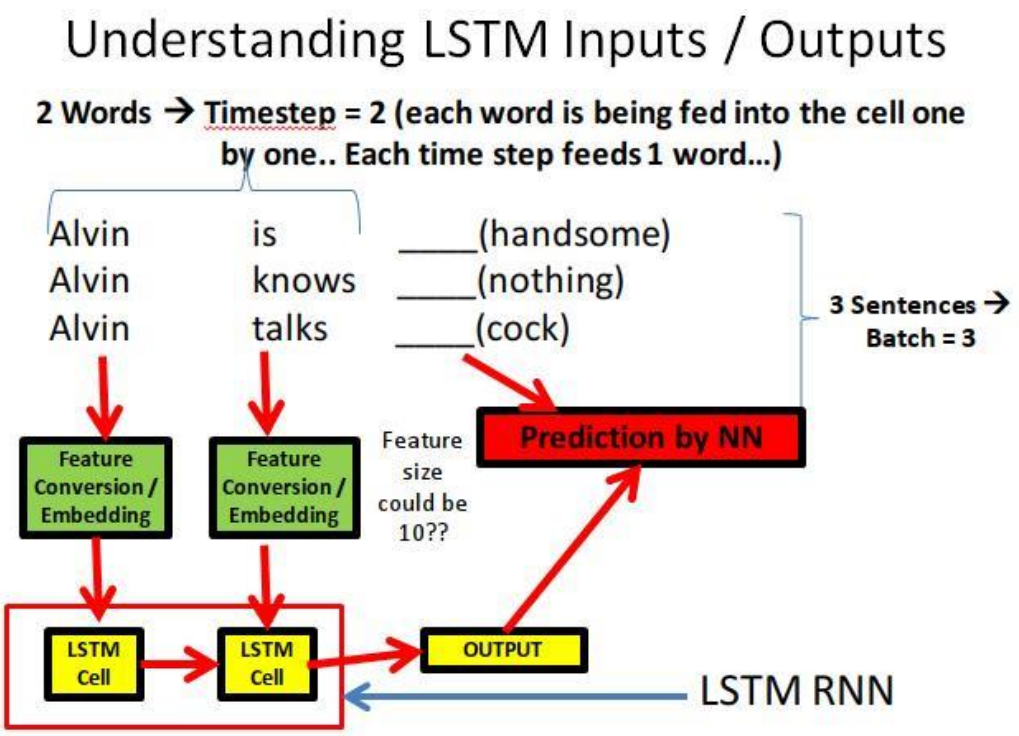
C. WHY LSTM CELL?

Key Problem for OLD RNN Cell

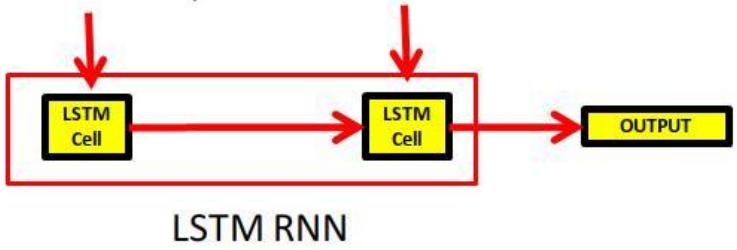
- It has a Long Term Dependency Issue
- It can only remember 3 to 5 words beforehand...
- E.g. *"Alvin holds a Pink Color NRIC. He is super handsome. Alvin is a _____"*(RNN is supposed to predict the word in the blank)
- Can you predict the word? YES! "SINGAPOREAN"
- But the RNN only remembers *"super handsome. Alvin is..."*
- If RNN cannot recall that I hold a Pink Color NRIC, how can it predict I'm Singaporean?
- It only recalls I'm SUPER HANDSOME!!! So it can only predict that I AM SUPER HANDSOME!!!
- HAHAHAHAHAHAHAHAHAAAAAAAAAAAAAAAAAAAAHAHAHAHAAAA
- (now new LSTM cell can remember 30 to 50 words)
- But GRU Cell is performance is about same as LSTM... also around 30 to 50 words....so we ignore....
- Most powerful Cell now is called the TRANSFORMER ARCHITECTURE!!!

COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

D. WHAT ARE THE INPUTS AND OUTPUTS OF A LSTM CELL?



- Input = a matrix of [batch, feature size, time step]
  - Batch = 3
  - Feature size = 10 (maybe?)
  - Time step = 2

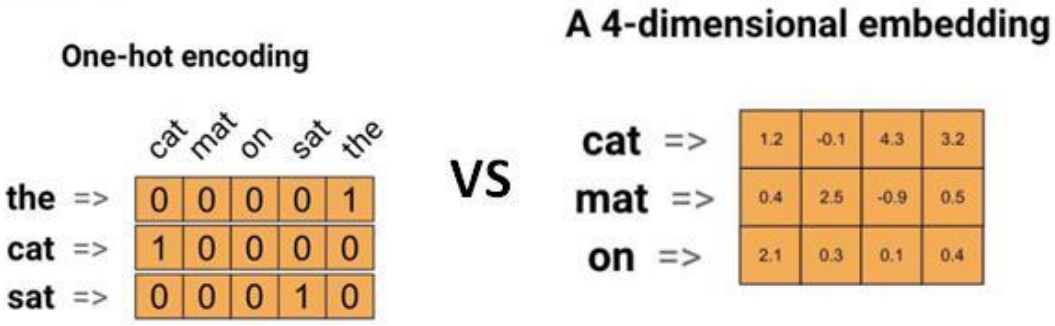


- Output = [batch size, hidden size]
  - Hidden size is the memory size

COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

What is Feature Conversion / Vectorization / Word Embedding?

- Feature Conversion / Embedding means Vectorization of the word into a number so that the computer can "understand" / "know" it's a word...
- Else computer only understands numbers.... (ASCII code)



COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

III. UNDERSTANDING RNN INPUTS AND OUTPUTS WITH PYTHON

FILE:
https://www.alvinang.sg/s/Understanding_LSTM_Input_and_Output_RNN_by_Dr_Alvin_Ang.ipynb

```
import tensorflow as tf
import numpy as np

batch = 3
feature = 10
# the vectorization size of the word / input
#(it depends on how the word has been Embedded / Vectorized for us to know the feature size...)

hidden_size = 20
# memory size
#you decide this.. no real way to know it...

timesteps = 2
# sequence length (or no. of words)

inputs = np.random.randn(batch, timesteps, feature)

lstm = tf.keras.layers.LSTM(hidden_size)

h_out = lstm(inputs)
```

```
[7] inputs.shape

(3, 2, 10)

[8] print('Output size(batch, hidden_size) = ', h_out.shape)

Output size(batch, hidden_size) = (3, 20)
```


COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.

COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG