Recurrent Neural Networks (RNNs)

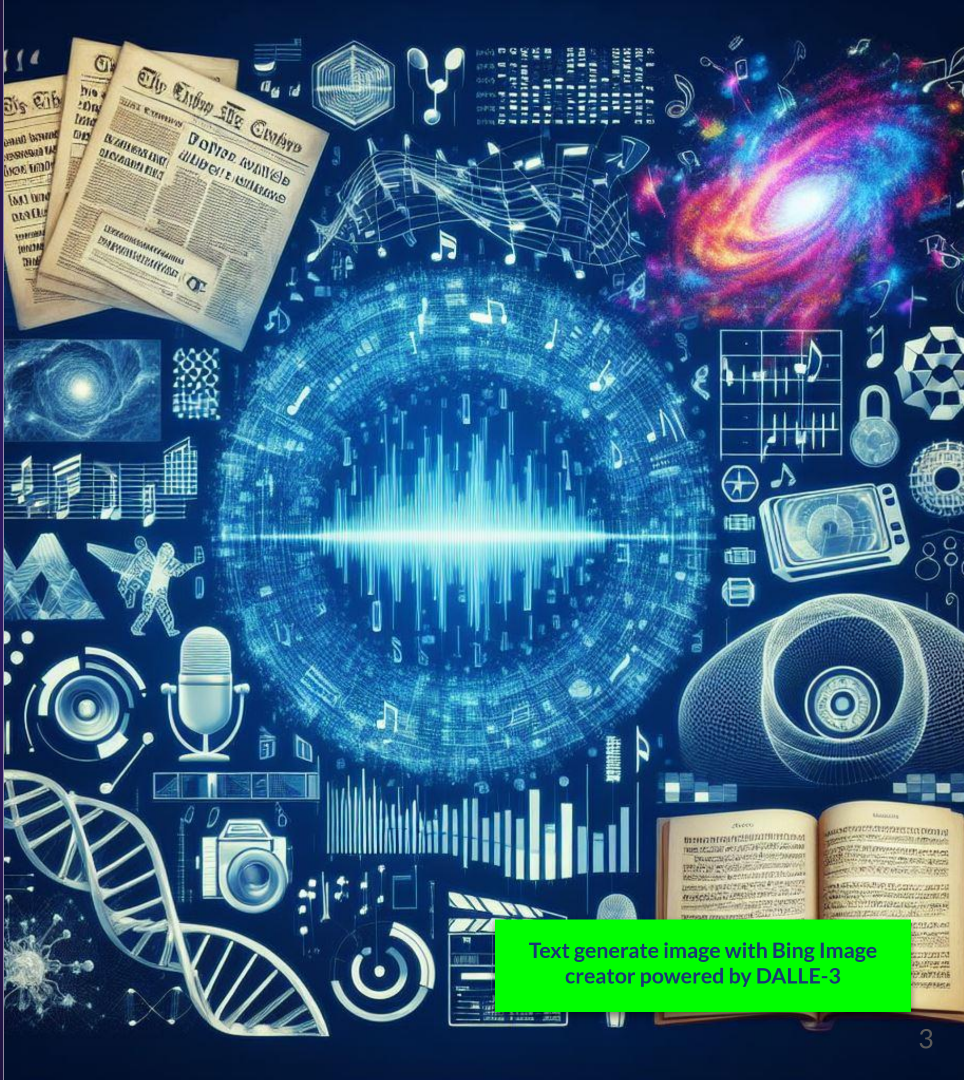
EE6350: Artificial Intelligence

Mr. M.W.G.C.K Moremada
Lecturer, Department of Electrical and Information Engineering, Faculty of Engineering, University of Ruhuna

Lecture Outline

1. RNNs: An Overview
   a. Examples
   b. RNN vs Feed-Forward NN
2. Sequential Data
   a. Examples
   b. Characteristic
3. RNNs
   a. RNN in Pytorch
   b. Simplified RNN Example
4. RNN Working
   a. Forward Propagation
   b. Common Activation Functions
   c. Backpropagation Through Time (BPTT)
   d. Example: Chatbot the Classify User Intentions
5. Types & Application of RNNs
   a. One-to-One
   b. One-to-Many
   c. Many-to-One
   d. Many-to-Many
   e. Advantages and Drawbacks
6. Problems with RNNs
   a. Vanishing Gradient Problem
   b. Exploding Gradient Problem
7. Variants of RNNs
   a. Deep RNNs
   b. Bidirectional RNNs
   c. LSTM
   d. GRU

RNNs: An Overview



Text generate image with Bing Image creator powered by DALLE-3

Predicting Sequence of Characters: E.g., Writing a Wikipedia Page

Input: Corpus of Wikipedia Text

Output Wikipedia Articles with Structured Markdown

```
Naturalism and decision for the m
by the Irish language by [[John Cl
with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal
in the [[Protestant Immineners]], which could be said to be directly in Cantonese
Communication, which followed a ceremony and set inspired prison, training. The
emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom
of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known
in western [[Scotland]], near Italy to the conquest of India with the conflict.
Copyright was the succession of independence in the slop of Syrian influence that
was a famous German movement based on a more popular servicious, non-doctrinal
and sexual power post. Many governments recognize the military housing of the
[[Civil Liberalization and Infantry Resolution 265 National Party in Hungary]],
that is sympathetic to be to the [[Punjab Resolution]]
(PJS)[http://www.humah.yahoo.com/guardian.
cfm/7754800786d17551963s89.htm Official economics Adjoint for the Nazism, Montgomery
was swear to advance to the resources for those Socialism's rule,
was starting to signing a major tripad of aid exile.]]
```

Ref: https://karpathy.github.io/2015/05/21/rnn-effectiveness/

Predicting Sequence of Characters: E.g., Write Like Shakespeare

Input: Works from Shakespeare

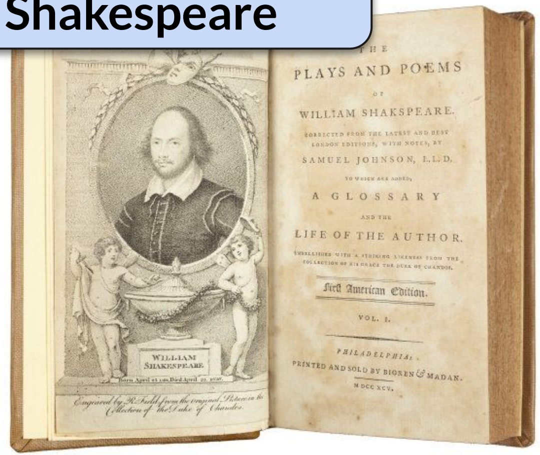


Image Source: https://www.shakespeare.org.uk/explore-shakespeare/blogs/first-american-edition-shakespeares-works/

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fe
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Ref: https://karpathy.github.io/2015/05/21/rnn-effectiveness/ |

Output: Text Following the Shakespeare's Writing Style

Predicting Sequence of Characters: E.g., Writing Code

Input: Code Files

Output: Predicted Code

Source: https://github.com/torvalds/linux/blob/master/kernel/gcov/base.c

```
SPDX-License-Identifier: GPL-2.0

 This code maintains a list of active profiling data structures.

   Copyright IBM Corp. 2009
 *   Author(s): Peter Oberparleiter <oberpar@linux.vnet.ibm.com>
 *
 *   Uses gcc-internal data definitions.
 *   Based on the gcov-kernel patch by:
 *              Hubertus Franke <frankeh@us.ibm.com>
 *              Nigel Hinds <nhinds@us.ibm.com>
 *              Rajan Ravindran <rajancr@us.ibm.com>
 *              Peter Oberparleiter <oberpar@linux.vnet.ibm.com>
 *              Paul Larson
 */

#define pr_fmt(fmt)     "gcov: " fmt

#include <linux/init.h>
#include <linux/module.h>
#include <linux/mutex.h>
#include <linux/sched.h>
#include "gcov.h"

int gcov_events_enabled;
DEFINE_MUTEX(gcov_lock);
```

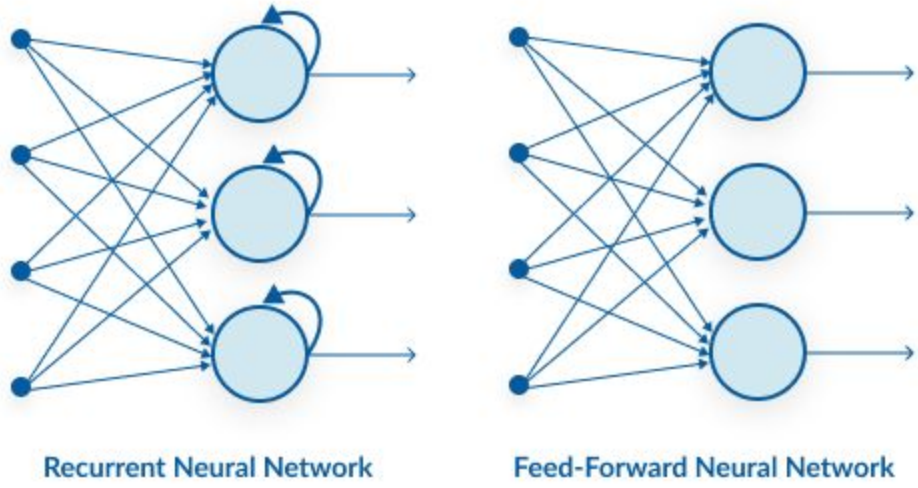
```
/*
 * Increment the size file of the new incorrect
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
```

Ref: https://karpathy.github.io/2015/05/21/rnn-effectiveness/

Recurrent Neural Networks (RNNs): An Overview

- Recurrent Neural Networks (RNNs) are a family of neural networks that **specialized in processing sequential data** (E.g., $x^{(1)}, \ldots, x^{(\tau)}$)
- RNNs can **scale to much longer sequences.**
- Most RNNs **can process sequences of variable length**.
- RNNS have an inherent "**memory**" as they take information from prior inputs to influence the current input and output.
- This can be indicated as a, **hidden layer** that remembers information through the passage of time.

RNN vs Feed-Forward NN



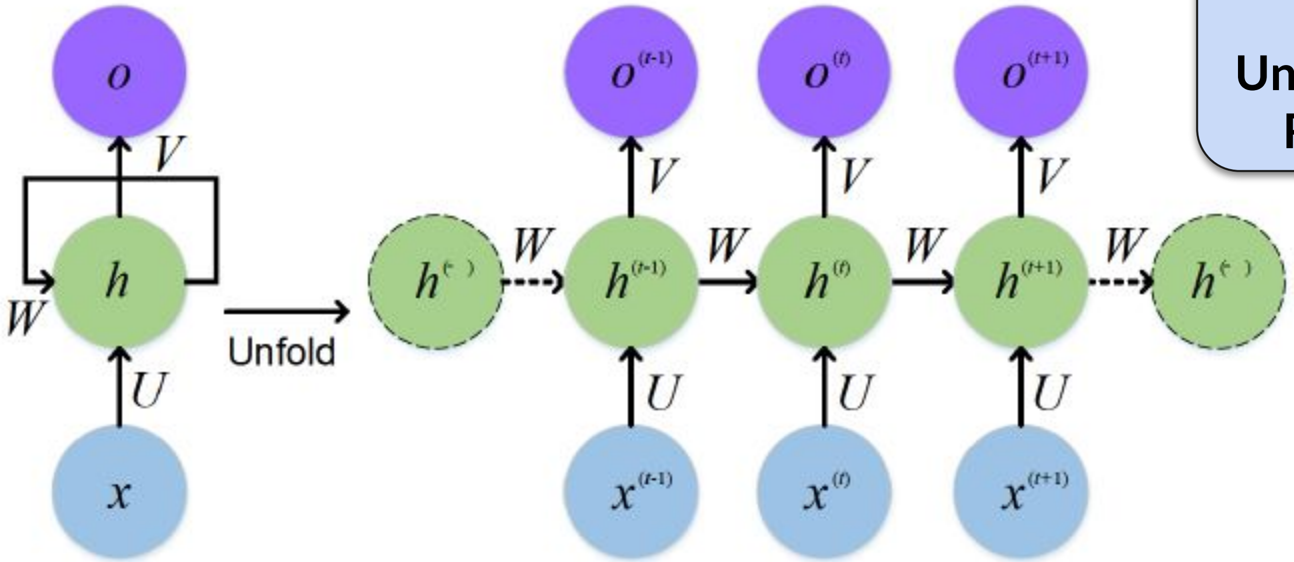
https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/

RNN vs Feed-Forward NN

- In conventional Feed-Forward NNs allows **data to flow in one direction only**; no loops, output of any layer will not affect same layer.
- In these networks' **outputs are depend on the current inputs** and does **not memorize** past data and no future scope.
- Also Feed-Forward NN **cannot handle sequential data**.
- **RNNs have signals travelling in both directions through feedback loops.**
- Features derived from earlier layers fed back into the network which gives them ability to **memorize**.

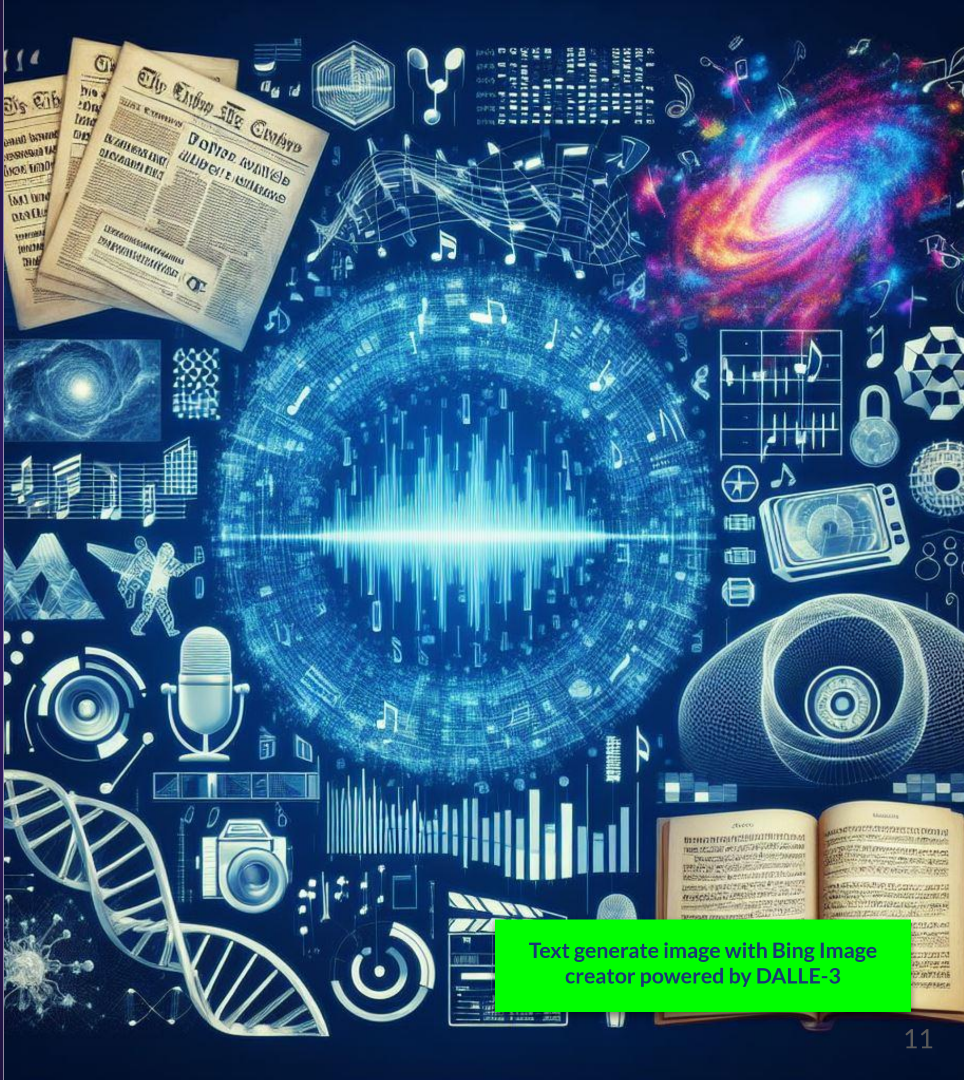
Recurrent Neural Networks (RNNs): An Overview

Standard and Unfolded RNN



W. Feng, N. Guan, Y. Li, X. Zhang, and Z. Luo, “Audio visual speech recognition with multimodal recurrent neural networks,” May 2017, doi: https://doi.org/10.1109/ijcnn.2017.7965918.

Sequential Data



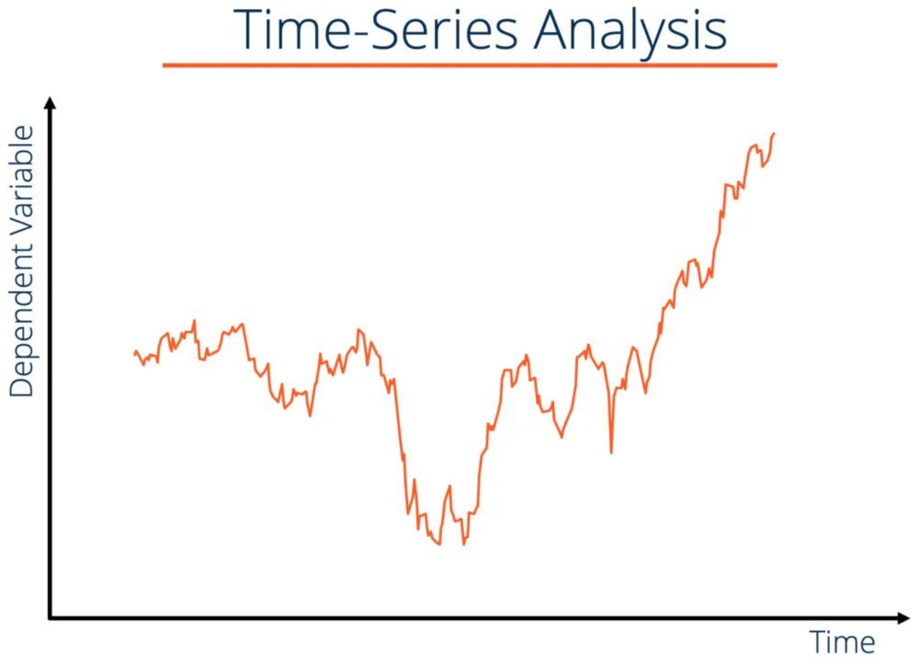
Text generate image with Bing Image creator powered by DALLE-3

What are The Sequence Data?

- Sequence data refers to a type of data where the **order of elements is significant,** and the **elements are typically arranged in a specific sequence or chronological order.**
- In sequence data, each element in the sequence is **related to the ones that come before and after it,** and the **way these elements are ordered** provides context, meaning, and structure to the data.

Sequential Data: Examples

- **Time Series Data:** Measurements are taken in regular intervals over time, E.g., stock prices, temperature readings, sensor data.



Sequential Data: Examples

- Natural Language Text

- Speech Signals

Sequential Data: Examples

- Music

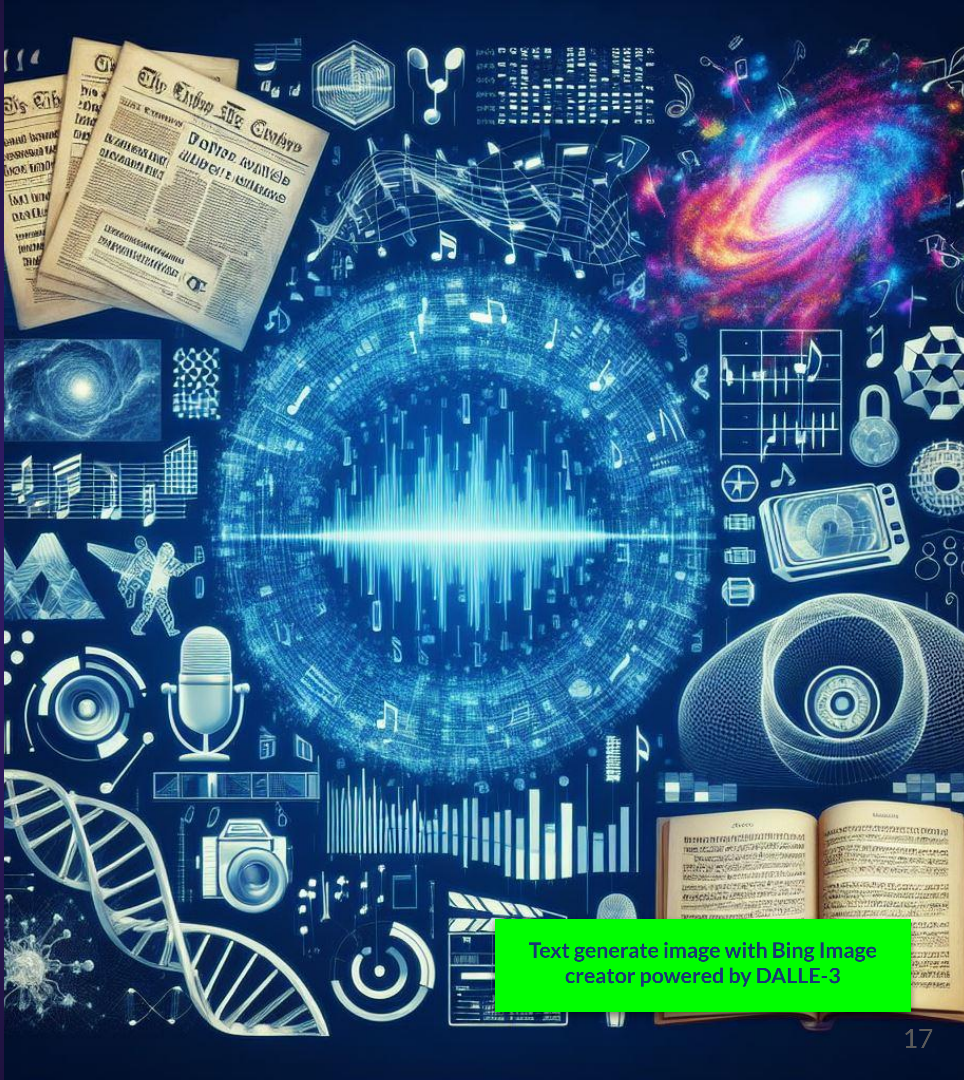
- DNA Sequences

- Video Frames

Sequence Data Characteristics

- **Order Matters:** Arrangement of data important and changing of oder may leads to different meanings/interpretations.
- **Temporal or Sequential Relationships:** Elements are typically collected or observed in a specific order over time.
- **Variable Length:** Sequences can have variable lengths, E.g., sentences of different length, DNA sequences of different length.

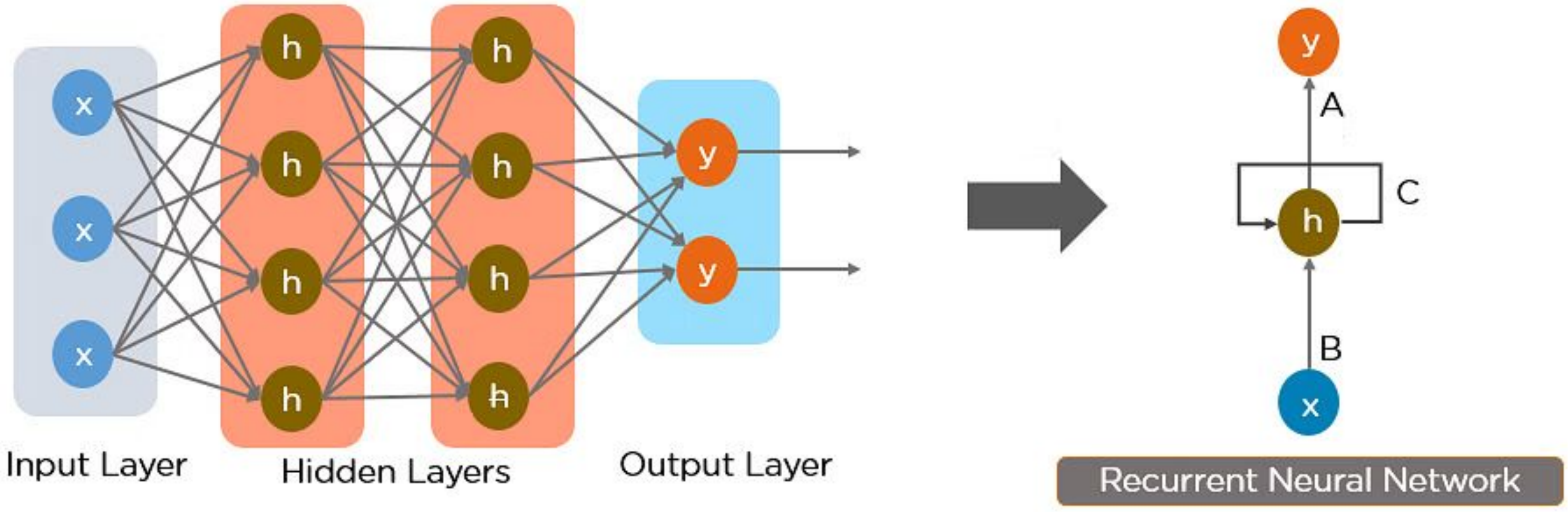
RNNs


Text generate image with Bing Image creator powered by DALLE-3

RNNs

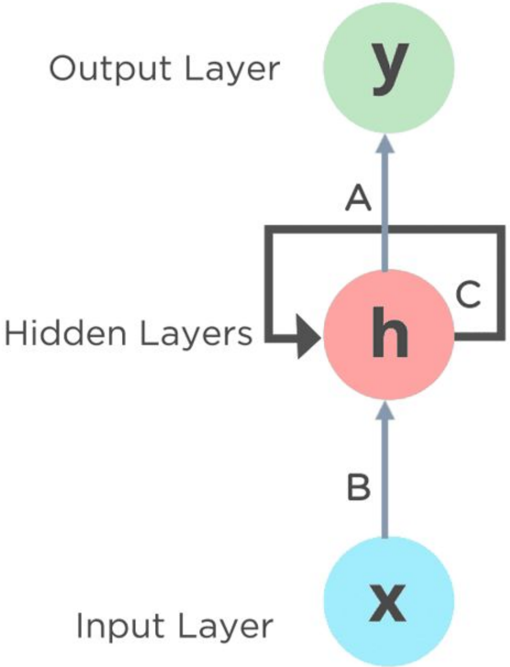
- The conventional feedforward ANNs consider each inputs and outputs are independent of each other.
- However, this assumption would be a disadvantage under the tasks such as next word prediction etc.
- RNNs perform same task for the all the elements in a sequence and **output depends on the previous computations**.
- So, RNN got a **memory**, which captures information about has been calculated so far.

RNNs


x
h
y
Input Layer
Hidden Layers
Output Layer
A
B
C
Recurrent Neural Network

RNNs

- Nodes in different layers of the ANN are **compressed** to form form a single layer of RNN.

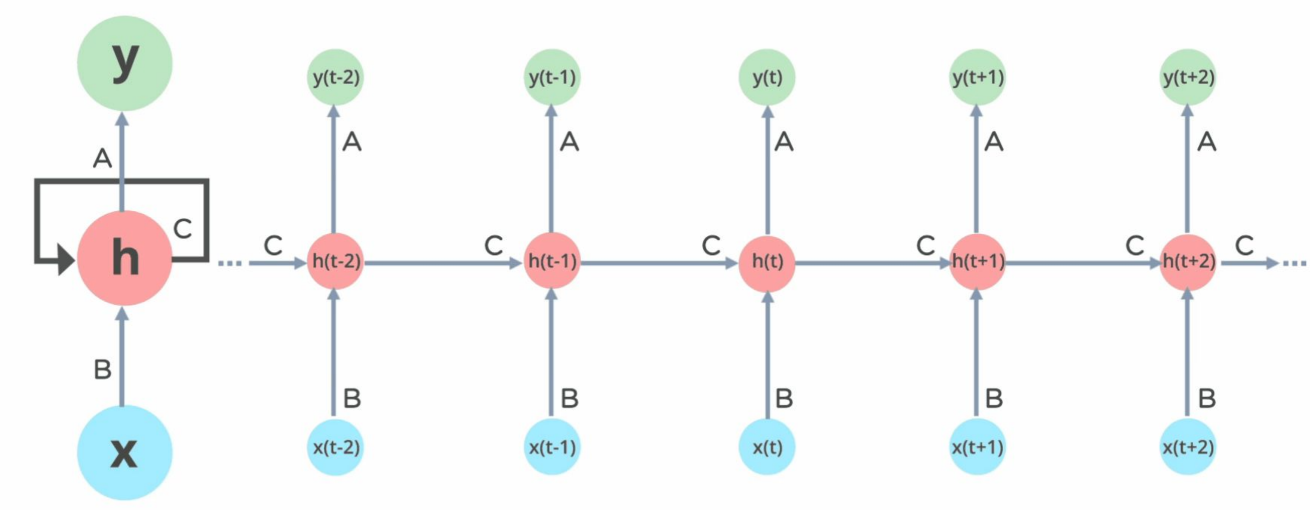


A, B and C are the parameters

A. Biswal, "Recurrent Neural Network Tutorial," Simplilearn.com, Apr. 10, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

RNNs

- In RNNs, the information cycles through a loop to the middle hidden layer.



A. Biswal, "Recurrent Neural Network Tutorial," Simplilearn.com, Apr. 10, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

RNN in Pytorch

CLASS torch.nn.RNN(*self*, *input_size*, *hidden_size*, *num_layers=1*, *nonlinearity='tanh'*, *bias=True*, *batch_first=False*, *dropout=0.0*, *bidirectional=False*, *device=None*, *dtype=None*) [SOURCE]

Apply a multi-layer Elman RNN with $\tanh$ or ReLU non-linearity to an input sequence. For each element in the input sequence, each layer computes the following function:

$$h_t = \tanh(x_t W_{ih}^T + b_{ih} + h_{t-1} W_{hh}^T + b_{hh})$$

where h_t is the hidden state at time *t*, x_t is the input at time *t*, and $h_{(t-1)}$ is the hidden state of the previous layer at time *t-1* or the initial hidden state at time *0*. If `nonlinearity` is `'relu'`, then ReLU is used instead of $\tanh$.

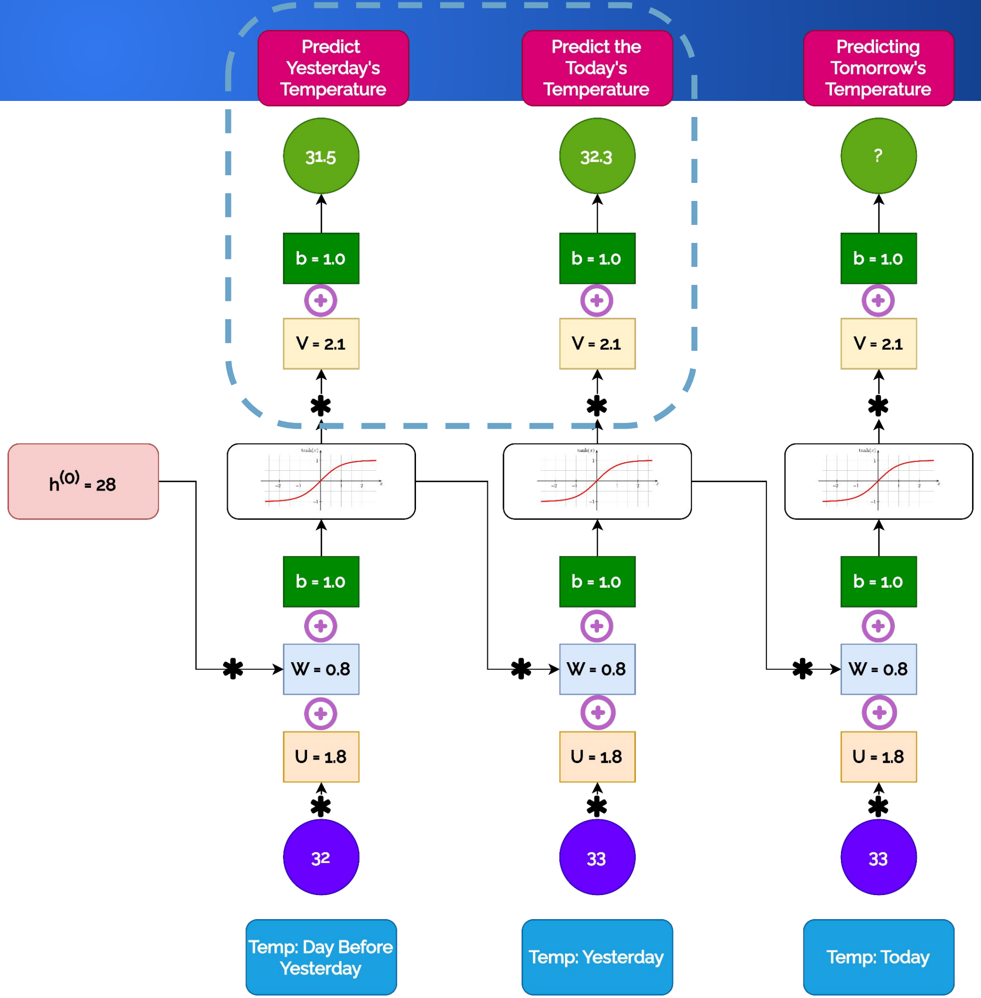
RNN in Pytorch

- **input_size** – The number of expected features in the input *x*
- **hidden_size** – The number of features in the hidden state *h*
- **num_layers** – Number of recurrent layers. E.g., setting `num_layers=2` would mean stacking two RNNs together to form a *stacked RNN*, with the second RNN taking in outputs of the first RNN and computing the final results. Default: 1
- **nonlinearity** – The non-linearity to use. Can be either `'tanh'` or `'relu'`. Default: `'tanh'`
- **bias** – If `False`, then the layer does not use bias weights *b_ih* and *b_hh*. Default: `True`
- **batch_first** – If `True`, then the input and output tensors are provided as (*batch, seq, feature*) instead of (*seq, batch, feature*). Note that this does not apply to hidden or cell states. See the Inputs/Outputs sections below for details. Default: `False`
- **dropout** – If non-zero, introduces a *Dropout* layer on the outputs of each RNN layer except the last layer, with dropout probability equal to `dropout`. Default: 0
- **bidirectional** – If `True`, becomes a bidirectional RNN. Default: `False`

Simplified RNN Example

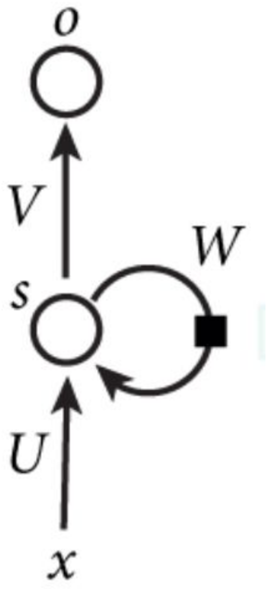
- **Task:** We are going to create a temperature forecasting model for Galle.
- **We are interested in predicting tomorrow's temperature.**

Note: Mentioned are just indicative values. Furthermore, in the core of RNNs computations are happening as matrix multiplications. This is a simplified example for concept illustration.



RNNs: A More Formal Definition

RNNs use hidden state to capture information about the past.

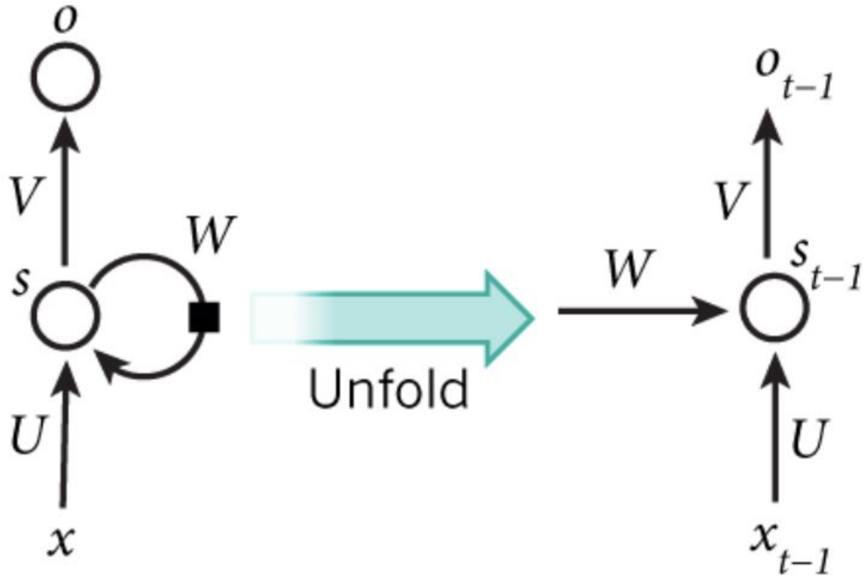


s = *h* which is refers to the hidden state. This notation being used in this set of slides interchangeably.

https://dennybritz.com/posts/wildml/recurrent-neural-networks-tutorial-part-1/

RNNs: A More Formal Definition

RNNs use hidden state to capture information about the past.



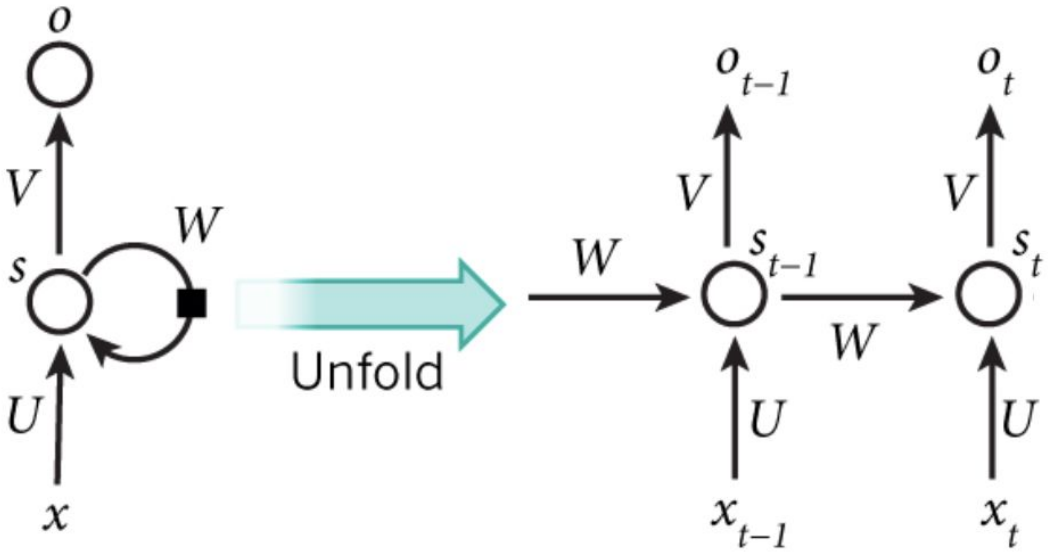
First Time Step

https://dennybritz.com/posts/wildml/recurrent-neural-networks-tutorial-part-1/

RNNs: A More Formal Definition

RNNs use hidden state to capture information about the past.

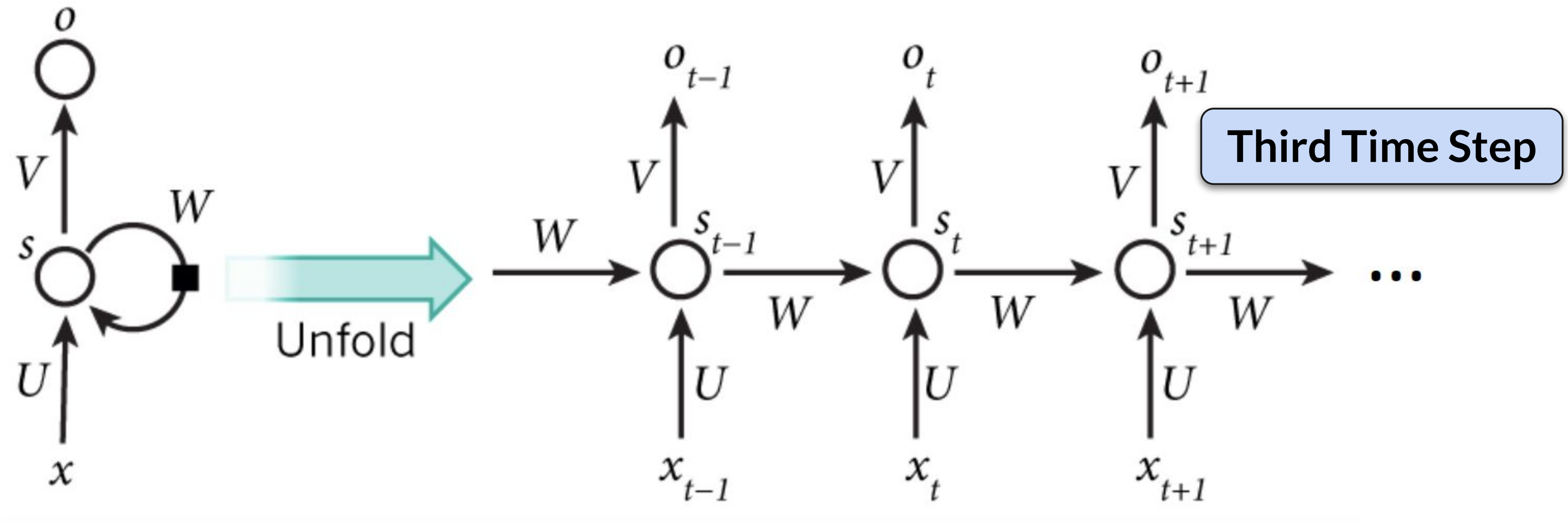
Second Time Step



https://dennybritz.com/posts/wildml/recurrent-neural-networks-tutorial-part-1/

RNNs: A More Formal Definition

RNNs use hidden state to capture information about the past.



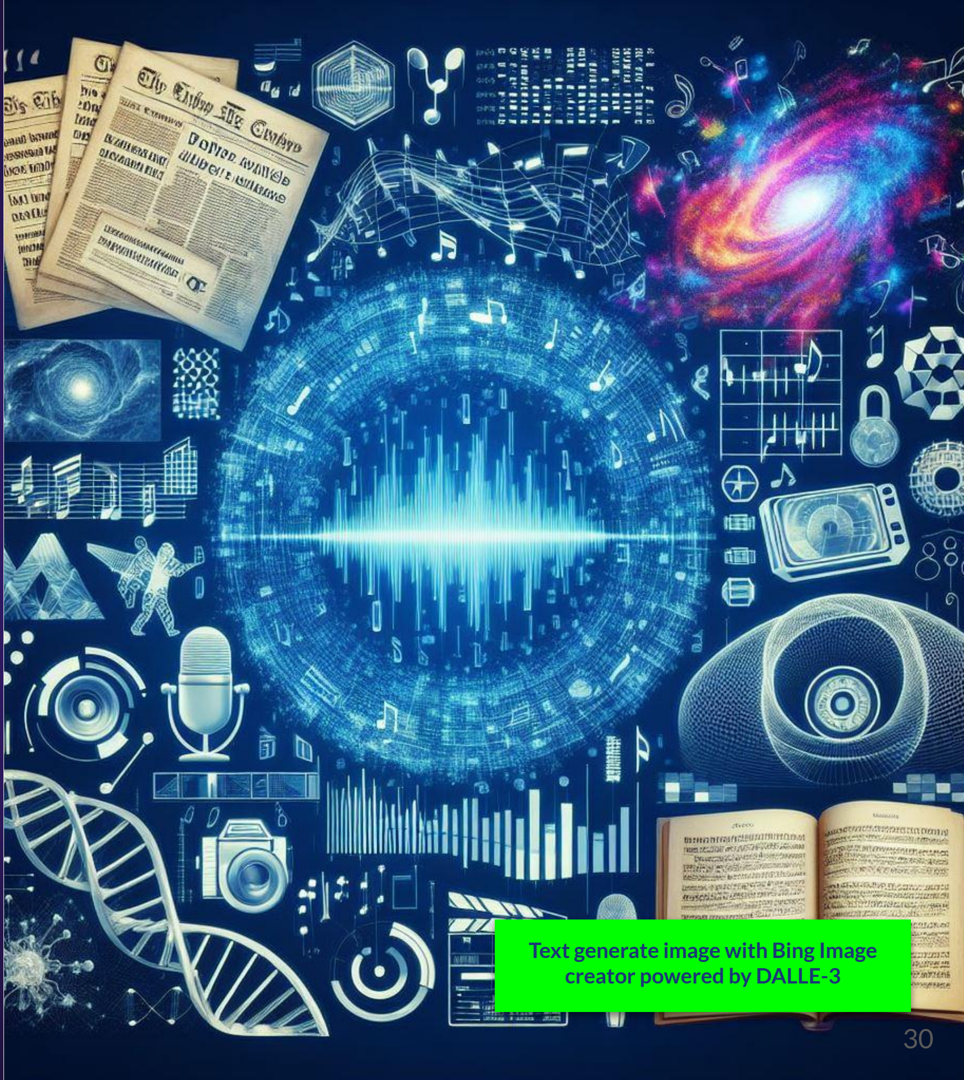
https://dennybritz.com/posts/wildml/recurrent-neural-networks-tutorial-part-1/

RNNs: A More Formal Definition

- In the above diagram:
    - $\boldsymbol{x_t}$ is the input at time step $\boldsymbol{t}$.
    - $\boldsymbol{s_t}$ (or $\boldsymbol{h_t}$ in previous diagrams) is the hidden state which is the "memory" of the network.
    - $\boldsymbol{o_t}$ is the output at step $\boldsymbol{t}$.
    - ***U, V, W*** are the weight matrices between input-to-hidden connection, hidden to-output connection, and hidden-to-hidden connection, respectively.

The U, V, W parameters are shared among all the time steps!

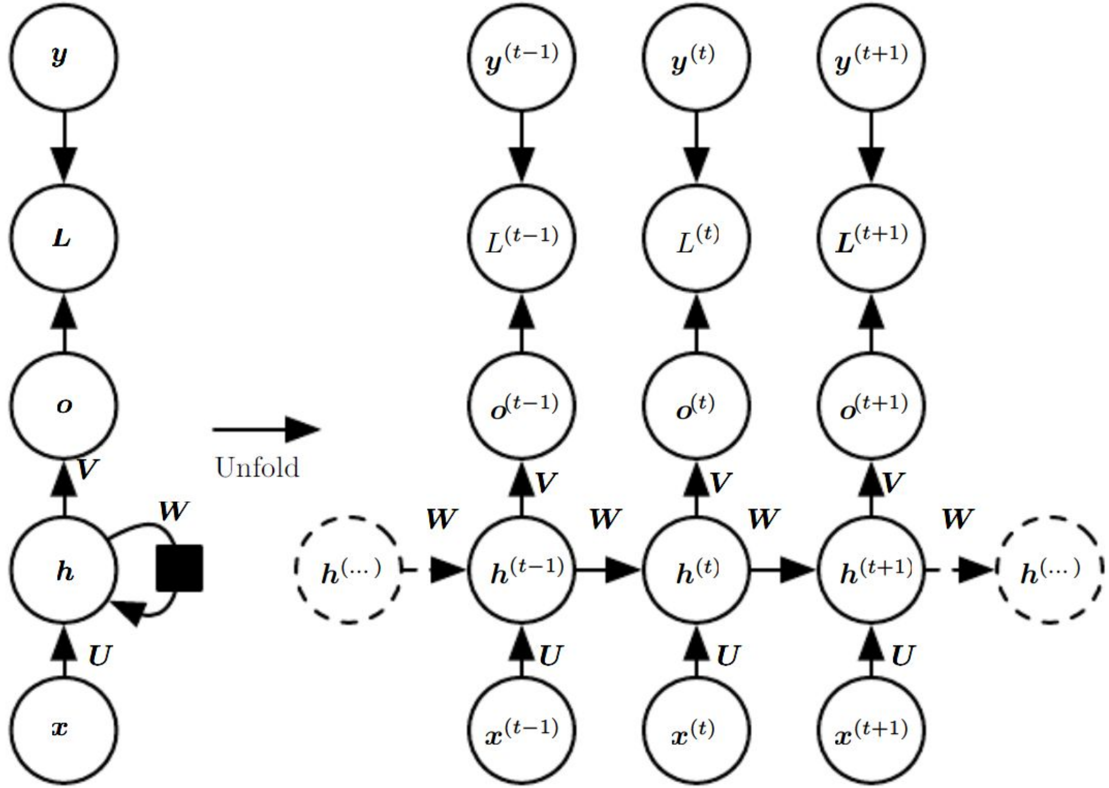
RNN Working



Text generate image with Bing Image creator powered by DALLE-3

How RNNs Work?

- Figure indicates the **complete diagram of RNN operation.**
- Here ***L*** is the loss and which measures how far each output ***(o)*** from the target ***(y)***.



I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, Massachusetts: The Mit Press, 2016.

How RNNs Work? - Forward Propagation

At the beginning of the forward propagation, the initial stage, $h^{(0)}$ is specified and after that, under each time step from t = 1 to t = **τ**, we apply the following updates accordingly,

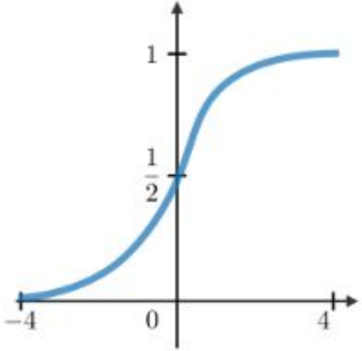
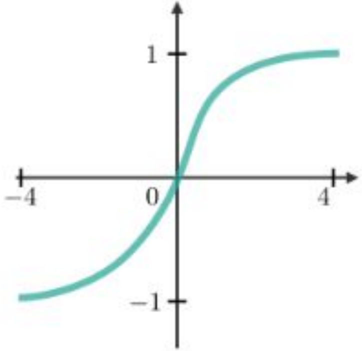
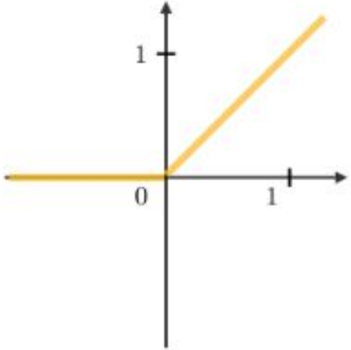
$$\begin{aligned} \boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \\ \boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}) \\ \boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \\ \hat{\boldsymbol{y}}^{(t)} &= \operatorname{softmax}(\boldsymbol{o}^{(t)}) \end{aligned}$$

Assume that activation function as the **hyperbolic tangent** and outputs are **discrete** (e.g., RNN being used for predict words or characters).

I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, Massachusetts: The Mit Press, 2016.

Recap

How RNNs Work? : Common Activation Functions

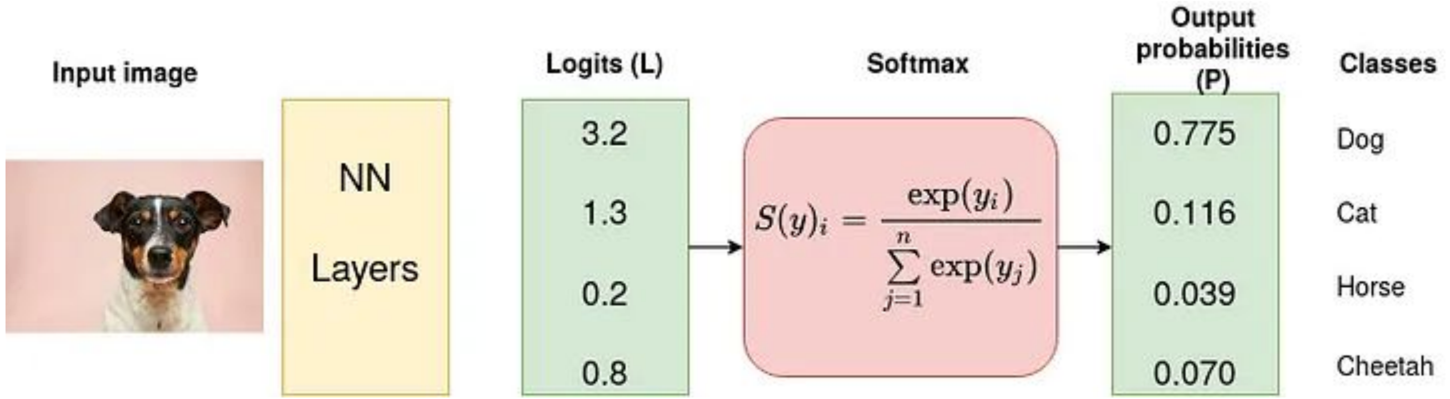
Sigmoid	Tanh	RELU
$g(z) = \dfrac{1}{1+e^{-z}}$	$g(z) = \dfrac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$
		

A. Amidi and S. Amidi, “CS 230 - Recurrent Neural Networks Cheatsheet,” Stanford.edu, 2019. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Info Slide

Softmax Activation

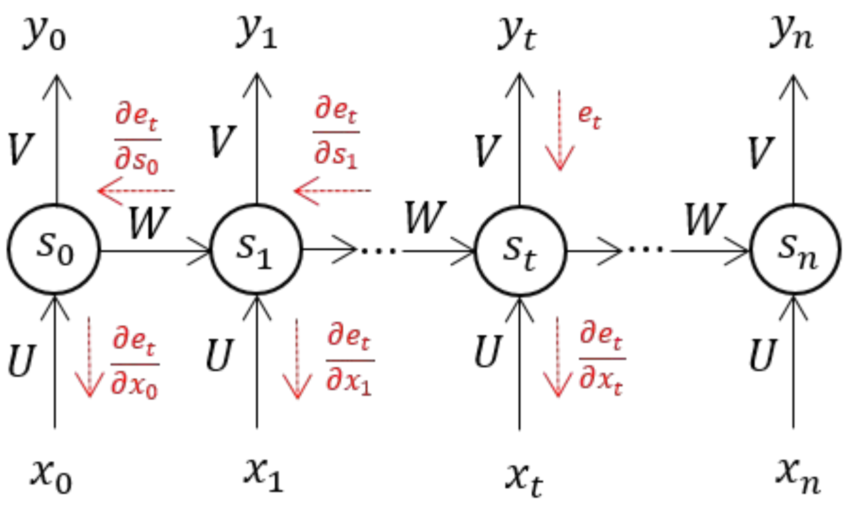
- Softmax activation function **normalize the output of a network to a probability distribution over predicted output classes.**
- The probabilities in vector ***v*** sums to one for all possible outcomes or classes.



https://towardsdatascience.com/softmax-activation-function-how-it-actually-works-d292d335bd78

How RNNs Work? - Backpropagation Through Time (BPTT)

- RNN predicts outputs using not only the current inputs but also taking the past inputs.
- Backpropagation Through Time, or BPTT, is the application of the Backpropagation training algorithm to recurrent neural network applied to sequence data like a time series.



How RNNs Work? - Backpropagation Through Time (BPTT)

- **Steps:**
  a. Forward propagate the data and compute the output at each time step.
  b. Error/loss calculation under each time step.
  c. Backpropagate the errors through the time across the unenrolled network.
  d. Weights update using some optimization algorithm such as gradient descent.
  e. Repeat.

Ref: M. Nguyen, “Illustrated Guide to Recurrent Neural Networks,” Medium, Sep. 20, 2018. https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

Example: Chatbot the Classify User Intentions

- In this case,
  - Encode sequence of text using RNN.
  - Then feed the RNN output into a feed-forward neural network to perform the classification.
  - User input: “**What time is it?**”

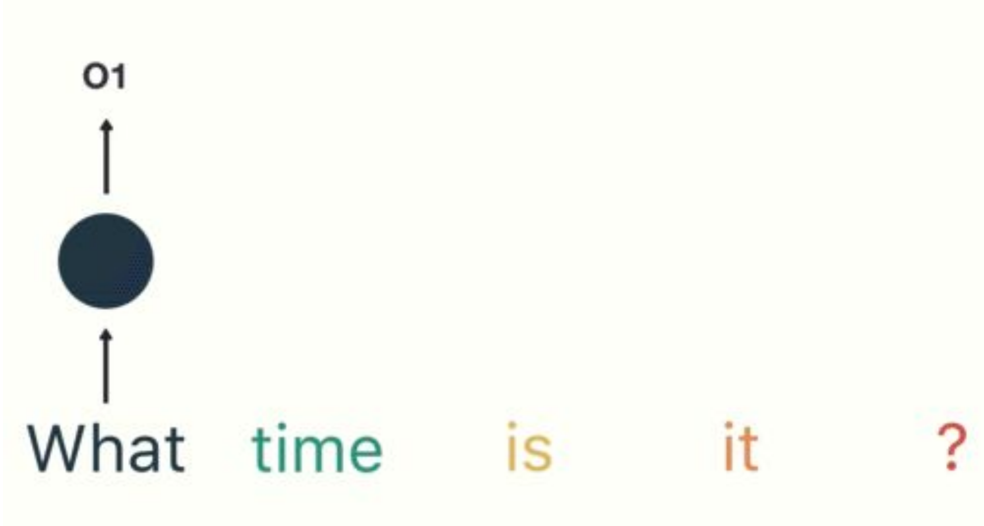
Break the sentence into words and feed it one word at a time to the RNN.

Ref: M. Nguyen, “Illustrated Guide to Recurrent Neural Networks,” Medium, Sep. 20, 2018. https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

Example: Chatbot the Classify User Intentions

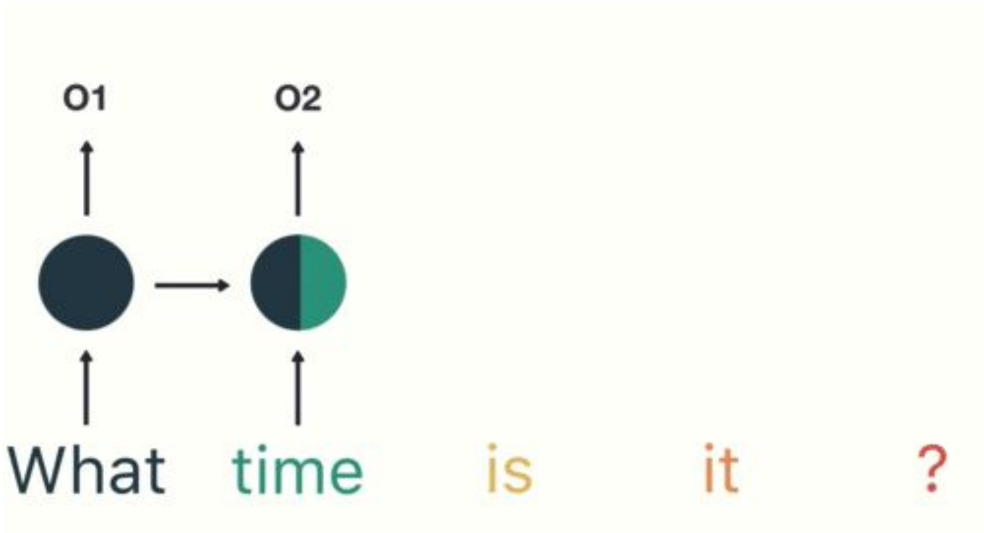
Ref: M. Nguyen, “Illustrated Guide to Recurrent Neural Networks,” Medium, Sep. 20, 2018. https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

Example: Chatbot the Classify User Intentions



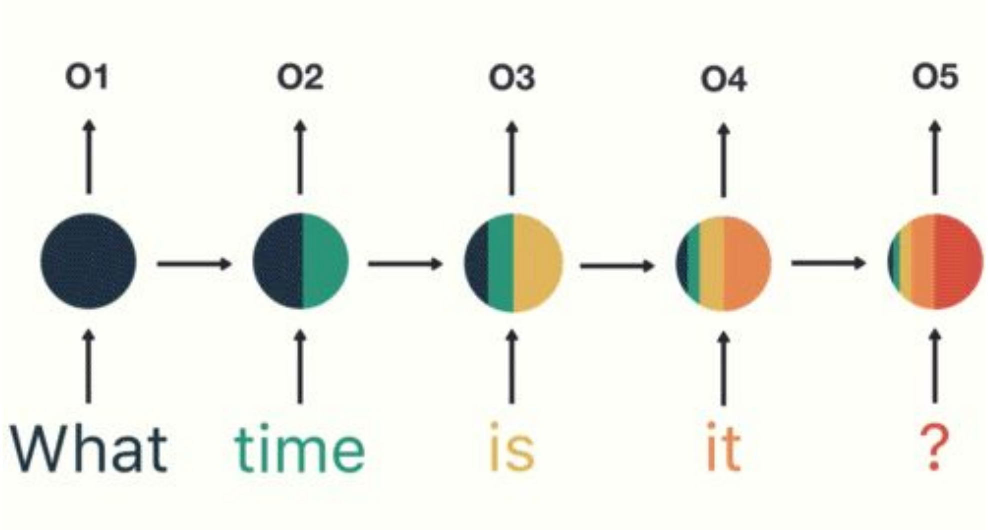
Ref: M. Nguyen, “Illustrated Guide to Recurrent Neural Networks,” Medium, Sep. 20, 2018. https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

Example: Chatbot the Classify User Intentions

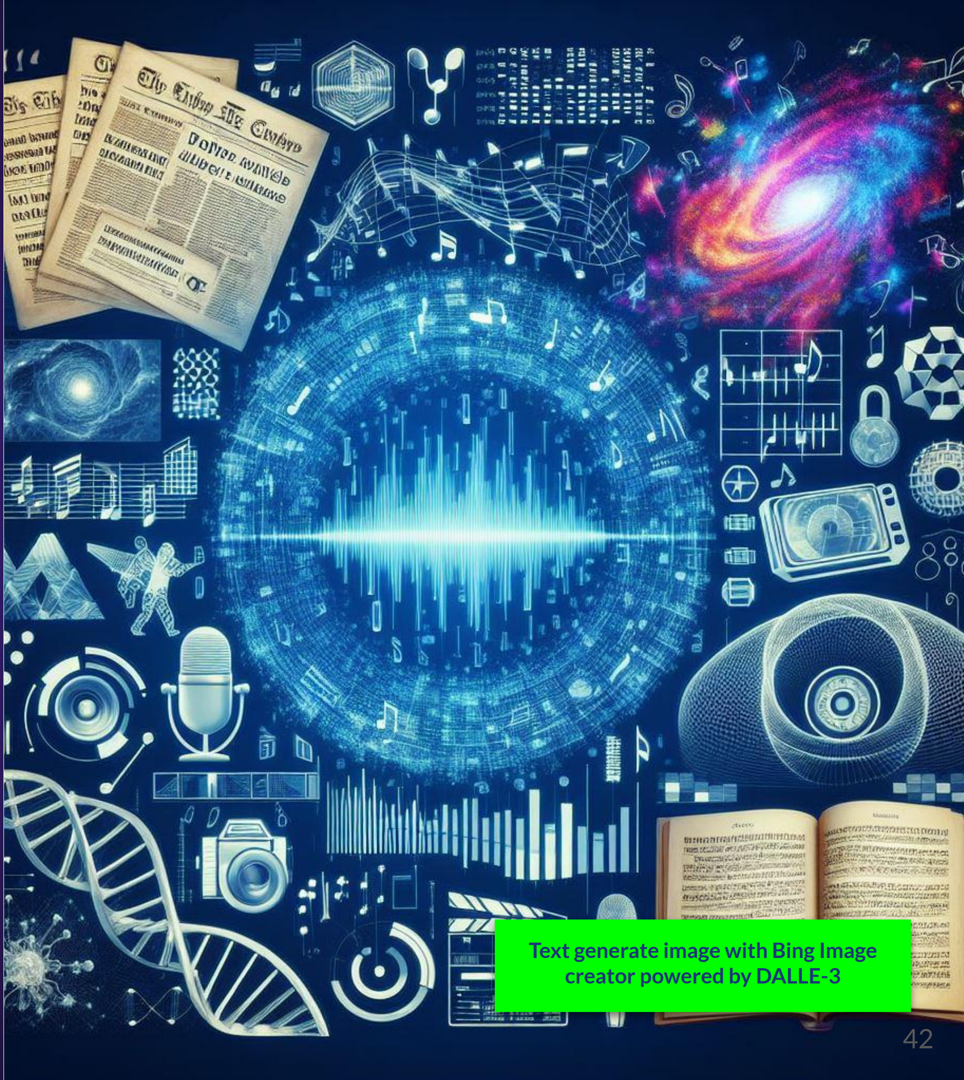


Ref: M. Nguyen, "Illustrated Guide to Recurrent Neural Networks," Medium, Sep. 20, 2018. https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

Example: Chatbot the Classify User Intentions

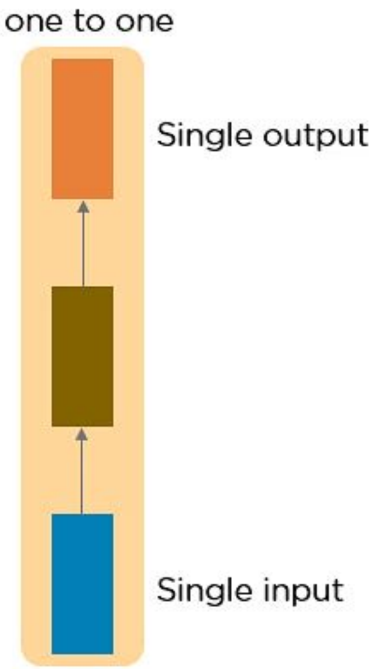


Types & Applications of RNNs



Text generate image with Bing Image creator powered by DALLE-3

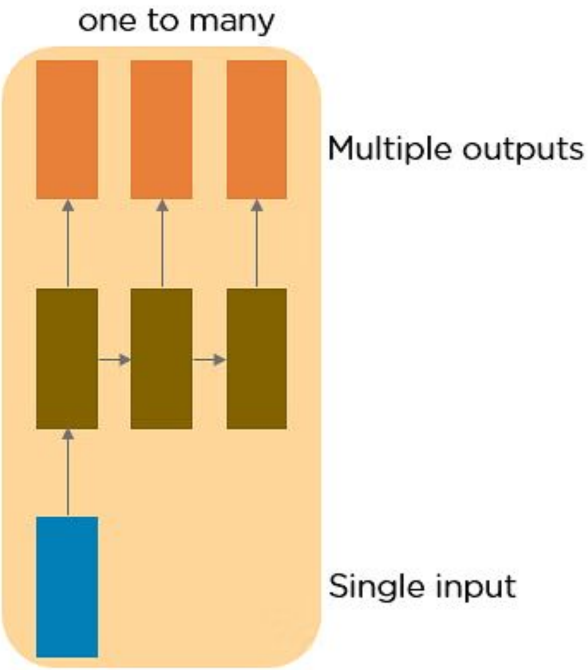
RNN Types: One-to-One



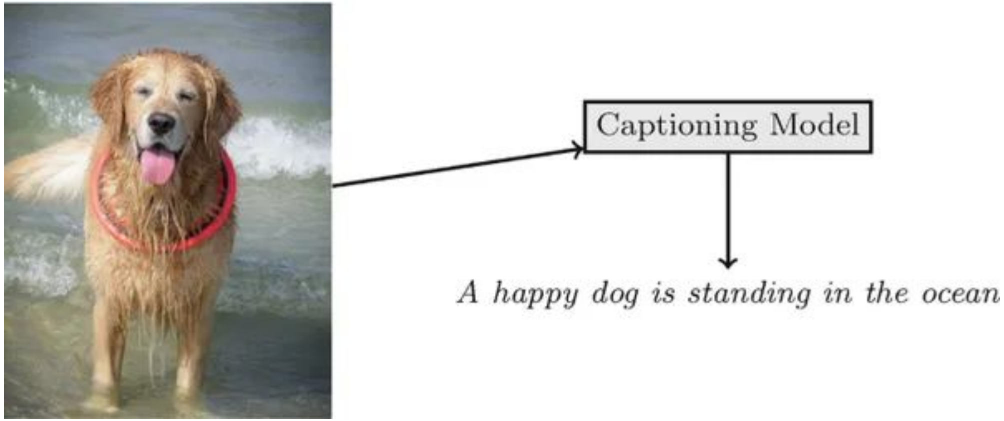
Examples for these types are the vanilla/traditional neural networks that maps single output to a single output.

A. Biswal, "Recurrent Neural Network Tutorial," Simplilearn.com, Apr. 10, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

RNN Types: One-to-Many

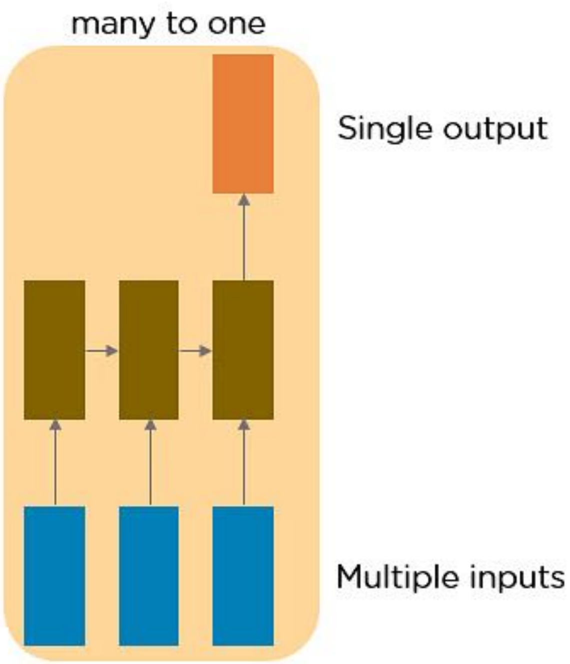


Example: Image Captioning

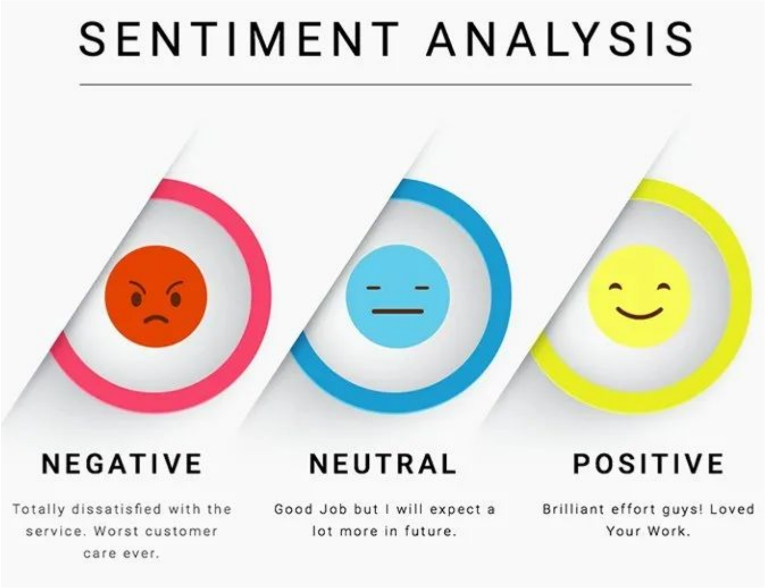


A. Biswal, "Recurrent Neural Network Tutorial," Simplilearn.com, Apr. 10, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

RNN Types: Many-to-One

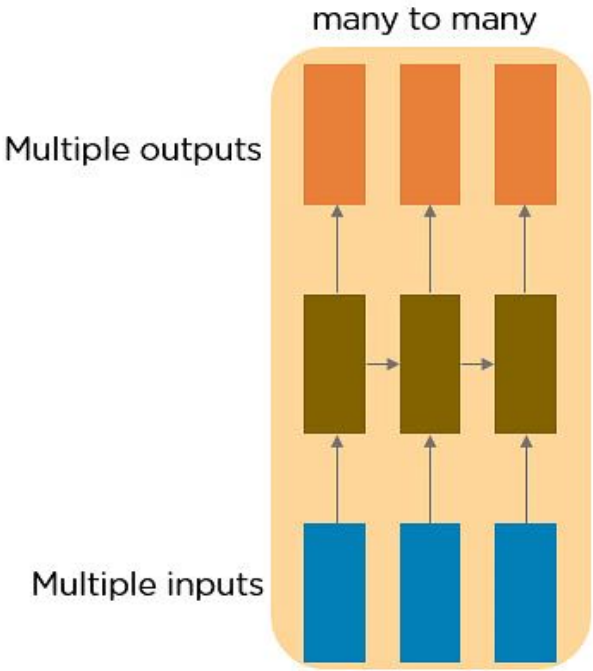


Example: Sentiment Analysis

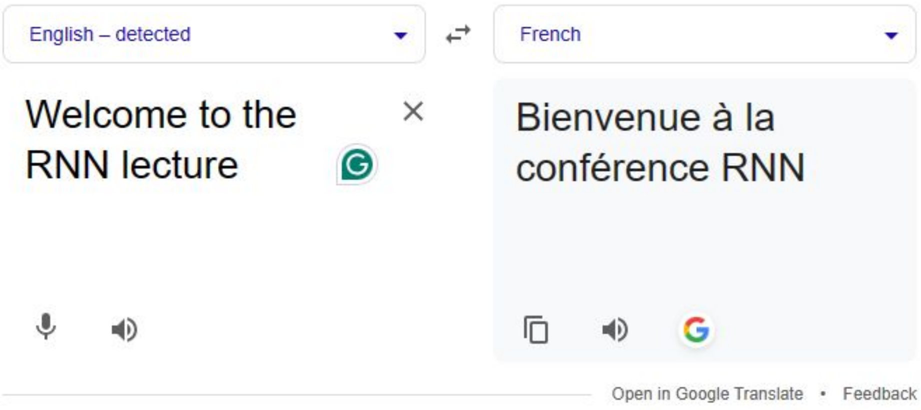


A. Biswal, "Recurrent Neural Network Tutorial," Simplilearn.com, Apr. 10, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

RNN Types: Many-to-Many



Example: Language Translation



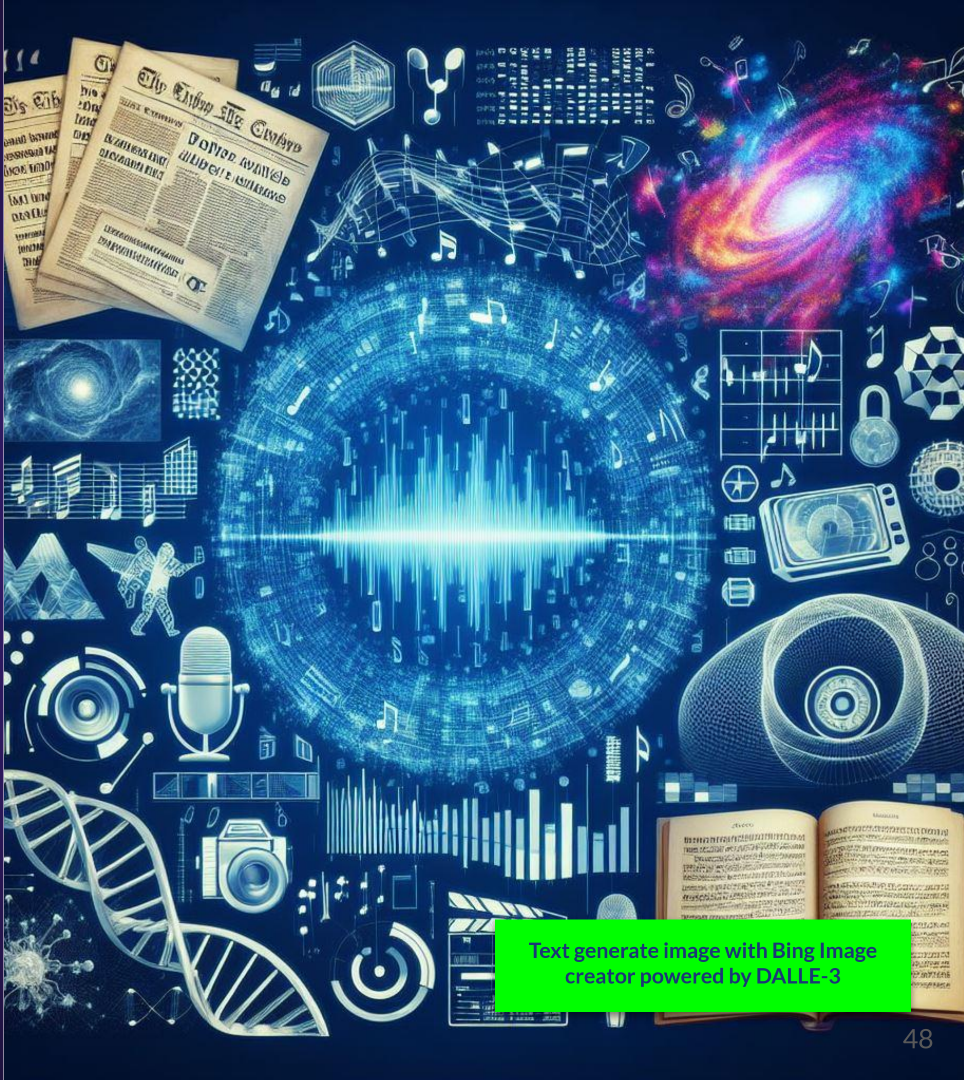
A. Biswal, "Recurrent Neural Network Tutorial," Simplilearn.com, Apr. 10, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

RNN: Advantages and Drawbacks

Advantages	Drawbacks
• Possibility of processing input of any length. • Model size not increasing with size of input. • Computation takes into account historical information. • Weights are shared across time.	• Computation being slow. • Difficulty of accessing information from a long time ago. • Cannot consider any future input for the current state.

A. Amidi and S. Amidi, “CS 230 - Recurrent Neural Networks Cheatsheet,” Stanford.edu, 2019. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

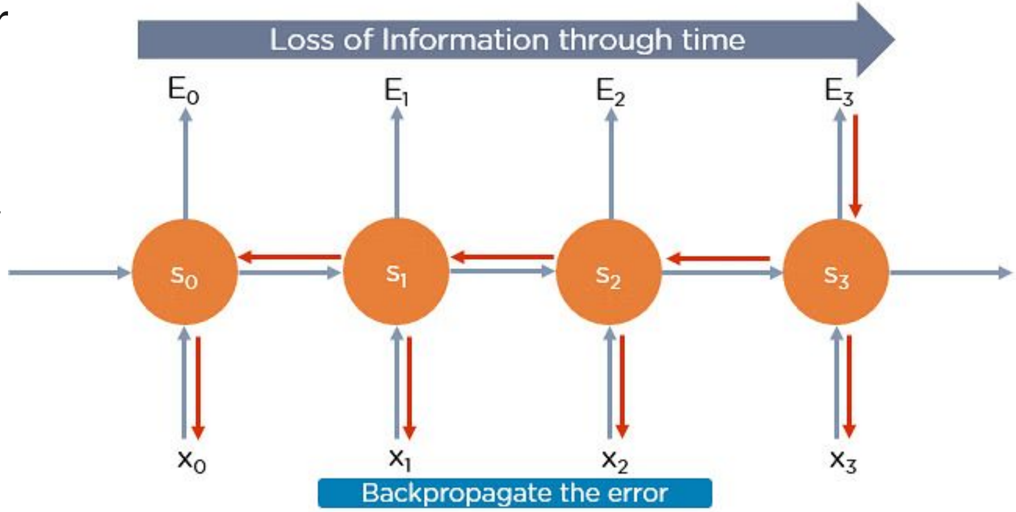
Problems with RNNs



Text generate image with Bing Image creator powered by DALLE-3

RNN Problems: Vanishing Gradient Problem

- When gradients become too small, the parameter updated become insignificant.
- This makes earning long sequences difficult.

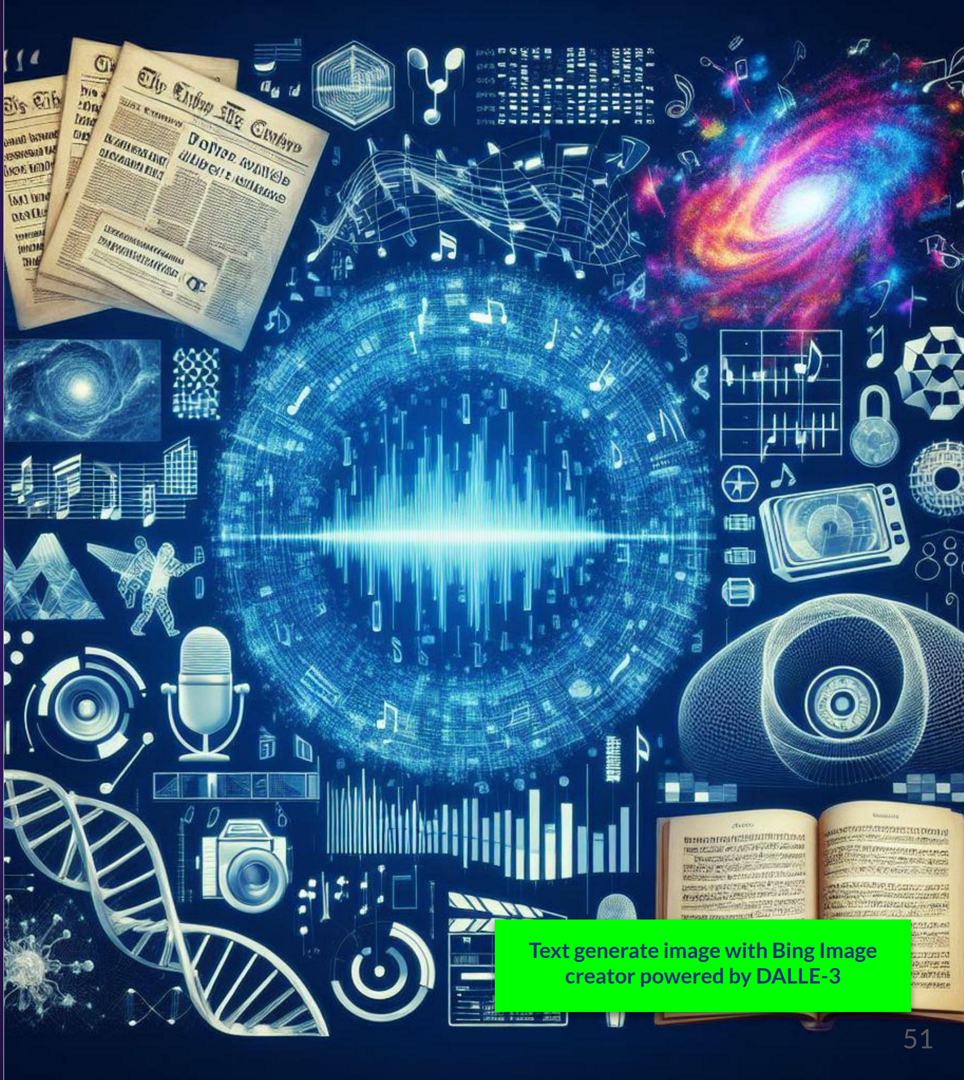


A. Biswal, "Recurrent Neural Network Tutorial," Simplilearn.com, Apr. 10, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

RNN Problems: Exploding Gradient Problem

- During the RNN training, the gradients tends to grow exponentially instead of decaying.
- This occurs when large error gradients accumulate which resulting in large updates to the weights.

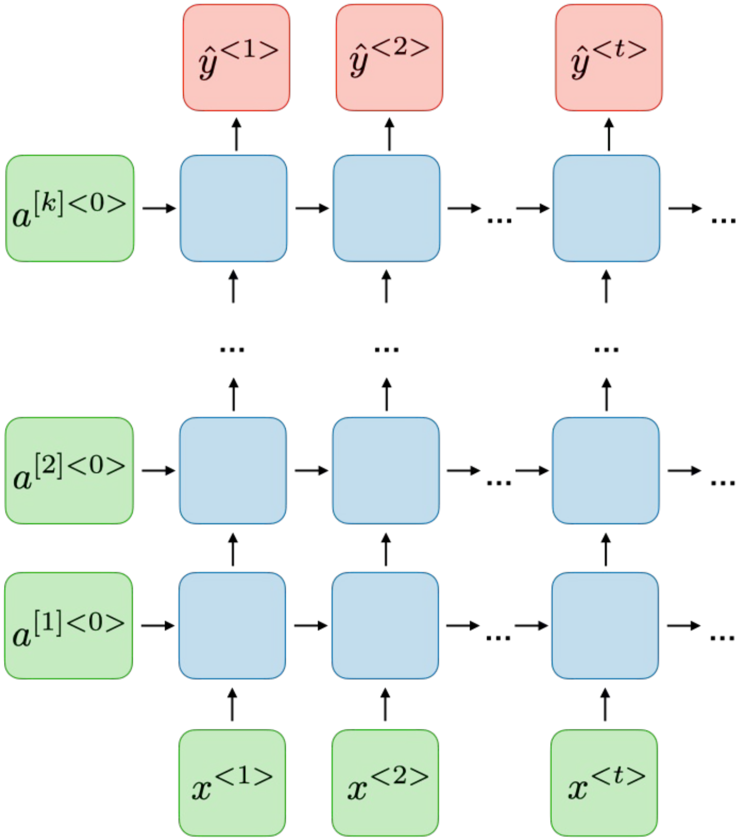
Variants of RNNs



Text generate image with Bing Image creator powered by DALLE-3

Deep RNNs

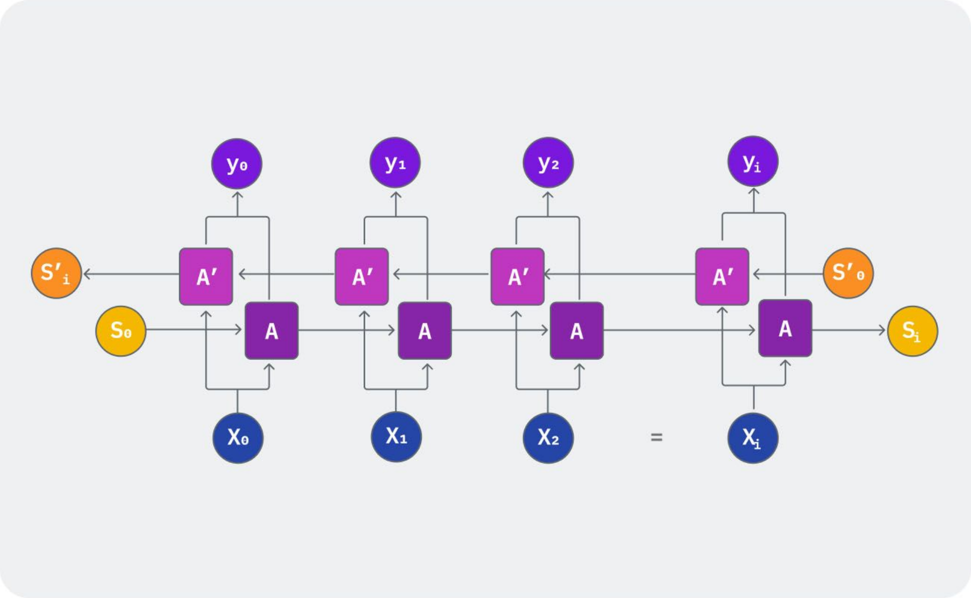
- Having multiple layer of RNN RNNs that stacked on top of each other.
- A Deep RNN takes the output from one layer of recurrent units and feeds it into the next layer, allowing the network to capture more complex relationships between the input and output sequences.



A. Amidi and S. Amidi, "CS 230 - Recurrent Neural Networks Cheatsheet," Stanford.edu, 2019. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Bidirectional RNNs (BRNNs)

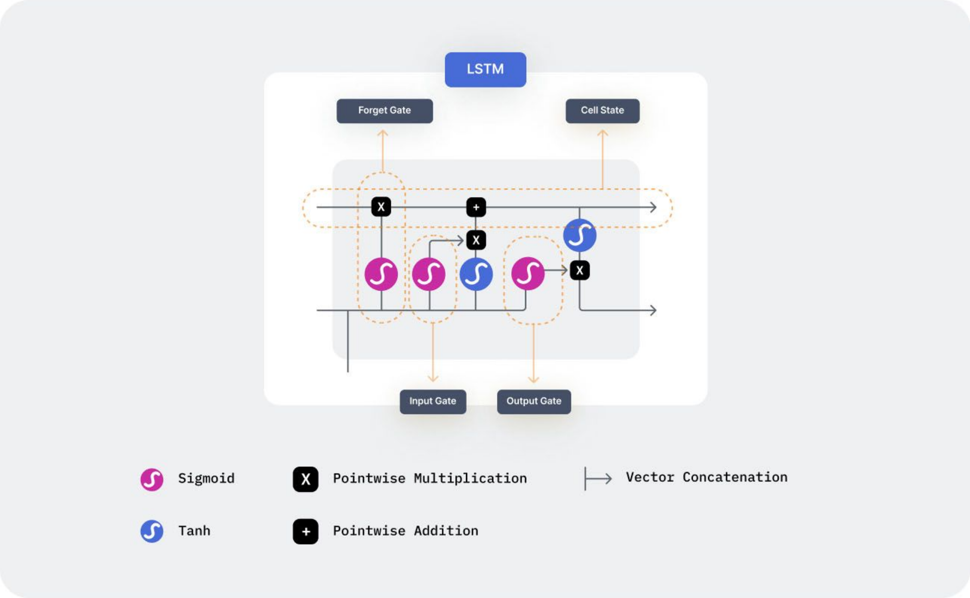
- In this case, the output at certain time “t” will not depend only on present and previous inputs but also **future inputs.**
- BRNNs are **combination of two RNNs.**



“The Complete Guide to Recurrent Neural Networks,” www.v7labs.com. https://www.v7labs.com/blog/recurrent-neural-networks-guide

Long Short Term Memory (LSTM)

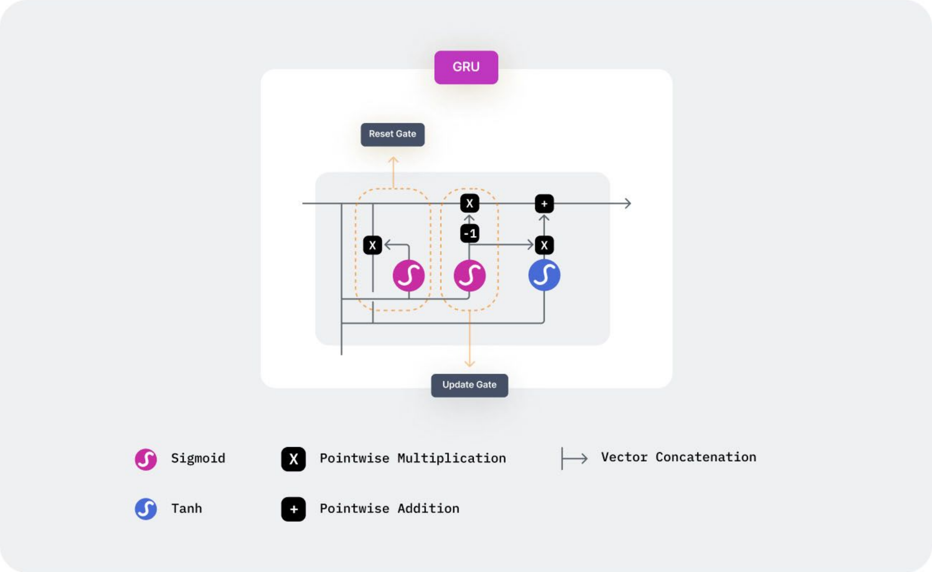
- Handles vanishing gradient problem.
- **Selectively remember or forget information from input sequence.**
- Employs three gates as, **input, output and forget** gates.



“The Complete Guide to Recurrent Neural Networks,” www.v7labs.com. https://www.v7labs.com/blog/recurrent-neural-networks-guide

Gated Recurrent Unit

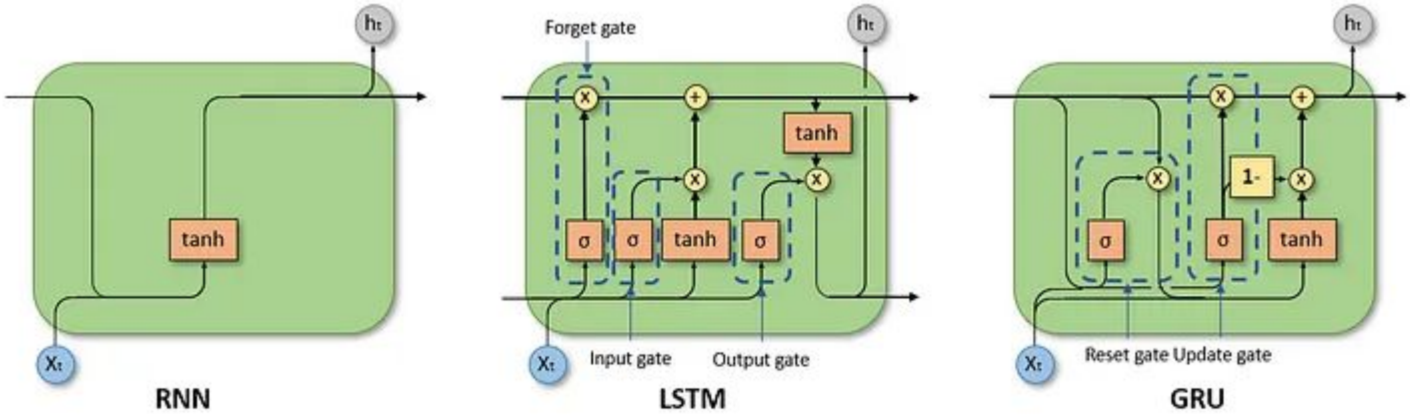
- Can be trained to keep long term information while removing whatever irrelevant to the prediction.
- **Two** types of gates as **update and reset** gates.



"The Complete Guide to Recurrent Neural Networks," www.v7labs.com. https://www.v7labs.com/blog/recurrent-neural-networks-guide

LSTM & GRU

- Both capable of learning long-term dependencies which mitigates the problems in vanilla RNNs related to short-term memory.



J. Dancker, "A Brief Introduction to Recurrent Neural Networks," Medium, Dec. 26, 2022. https://towardsdatascience.com/a-brief-introduction-to-recurrent-neural-networks-638f64a61ff4

References

1. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, Massachusetts: The Mit Press, 2016.
2. “Time Series Data Analysis,” Corporate Finance Institute. https://corporatefinanceinstitute.com/resources/data-science/time-series-data-analysis/
3. A. Biswal, “Recurrent Neural Network Tutorial,” Simplilearn.com, Apr. 10, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn
4. W. Feng, N. Guan, Y. Li, X. Zhang, and Z. Luo, “Audio visual speech recognition with multimodal recurrent neural networks,” May 2017, doi: https://doi.org/10.1109/ijcnn.2017.7965918.
5. D. Gurari, “Recurrent Neural Networks.” Accessed: Mar. 31, 2024. [Online]. Available: https://home.cs.colorado.edu/~DrG/Courses/IntroToMachineLearning/Lectures/10_RecurrentNeuralNetworks.pdf
6. Jason Brownlee, “A Gentle Introduction to Backpropagation Through Time,” Machine Learning Mastery, Jun. 22, 2017. https://machinelearningmastery.com/gentle-introduction-backpropagation-time/
7. M. Inuwa, “Vision Transformers (ViT) in Image Captioning Using Pretrained ViT Models,” Analytics Vidhya, Jun. 26, 2023. https://www.analyticsvidhya.com/blog/2023/06/vision-transformers/ (accessed Apr. 01, 2024).
8. P. Dixit, “Sentiment Analysis: Building from the Ground Up,” Analytics Vidhya, May 16, 2020. https://medium.com/analytics-vidhya/sentiment-analysis-building-from-the-ground-up-e12e9195fac4
9. “The Complete Guide to Recurrent Neural Networks,” www.v7labs.com. https://www.v7labs.com/blog/recurrent-neural-networks-guide
10. “Deep RNN,” www.linkedin.com. https://www.linkedin.com/pulse/deep-rnn-dhiraj-patra#:~:text=A%20Deep%20RNN%20takes%20the (accessed Apr. 01, 2024).
11. J. Dancker, “A Brief Introduction to Recurrent Neural Networks,” Medium, Dec. 26, 2022. https://towardsdatascience.com/a-brief-introduction-to-recurrent-neural-networks-638f64a61ff4
12. “Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs,” Denny’s Blog, Sep. 17, 2015. https://dennybritz.com/posts/wildml/recurrent-neural-networks-tutorial-part-1/
13. A. Amidi and S. Amidi, “CS 230 - Recurrent Neural Networks Cheatsheet,” Stanford.edu, 2019. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Thank You