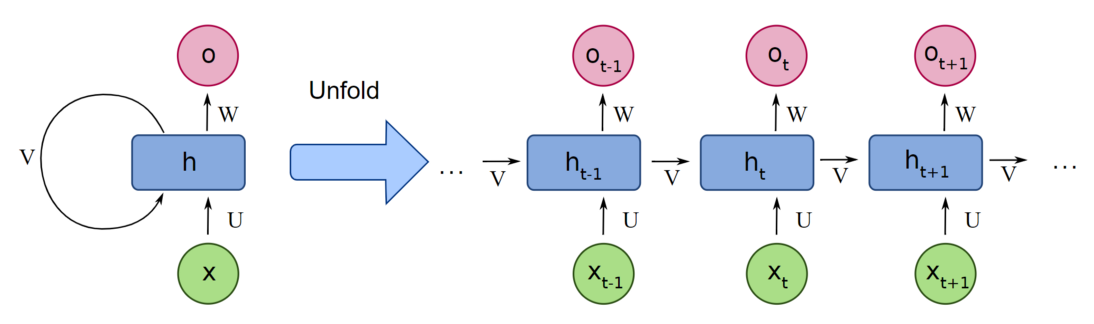
**Recurrent Neural Networks**

A Deep Learning approach for modelling sequential data is **Recurrent Neural Networks (RNN)**. RNNs were the standard suggestion for working with sequential data before the advent of attention models. Specific parameters for each element of the sequence may be required by a deep feedforward model. It may also be unable to generalize to variable-length sequences.



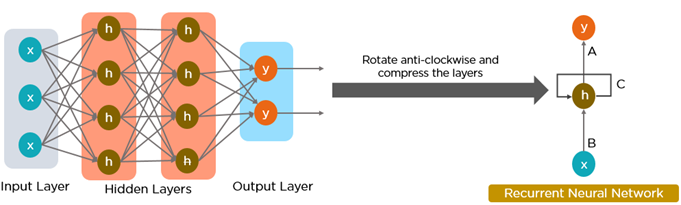
Recurrent Neural Networks use the same weights for each element of the sequence, decreasing the number of parameters and allowing the model to generalize to sequences of varying lengths. RNNs generalize to structured data other than sequential data, such as geographical or graphical data, because of its design.

Recurrent neural networks, like many other deep learning techniques, are relatively old. They were first developed in the 1980s, but we didn’t appreciate their full potential until lately. The advent of long short-term memory (LSTM) in the 1990s, combined with an increase in computational power and the vast amounts of data that we now have to deal with, has really pushed RNNs to the forefront.

**What is a Recurrent Neural Network (RNN)?**

Neural networks imitate the function of the human brain in the fields of AI, machine learning, and deep learning, allowing computer programs to recognize patterns and solve common issues.

RNNs are a type of neural network that can be used to model sequence data. RNNs, which are formed from feedforward networks, are similar to human brains in their behaviour. Simply said, recurrent neural networks can anticipate sequential data in a way that other algorithms can’t.

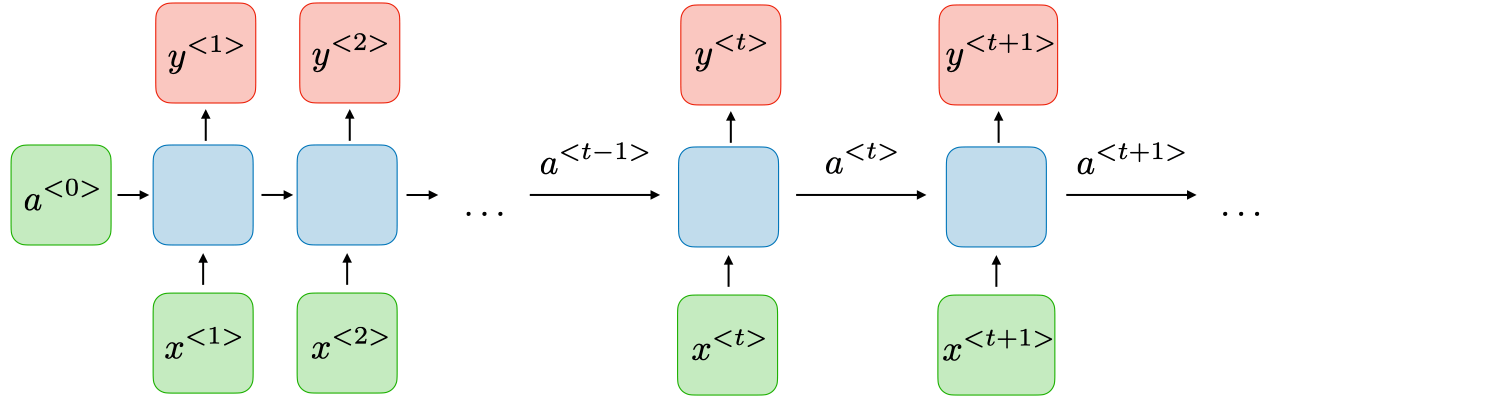


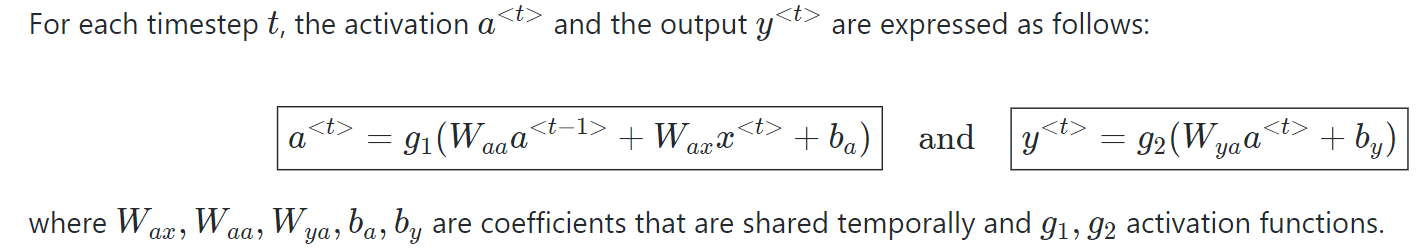
 All of the inputs and outputs in standard neural networks are independent of one another, however in some circumstances, such as when predicting the next word of a phrase, the prior words are necessary, and so the previous words must be remembered. As a result, RNN was created, which used a Hidden Layer to overcome the problem. The most important component of RNN is the Hidden state, which remembers specific information about a sequence.

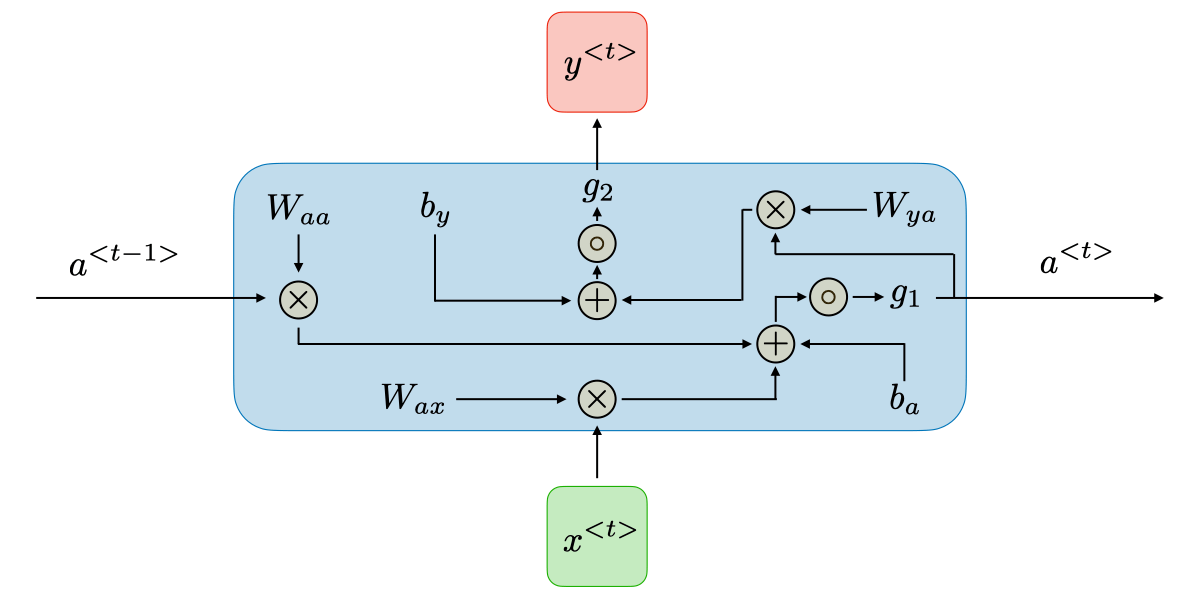
RNNs have a Memory that stores all information about the calculations. It employs the same settings for each input since it produces the same outcome by performing the same task on all inputs or hidden layers.

**The Architecture of a Traditional RNN**

RNNs are a type of neural network that has hidden states and allows past outputs to be used as inputs. They usually go like this:







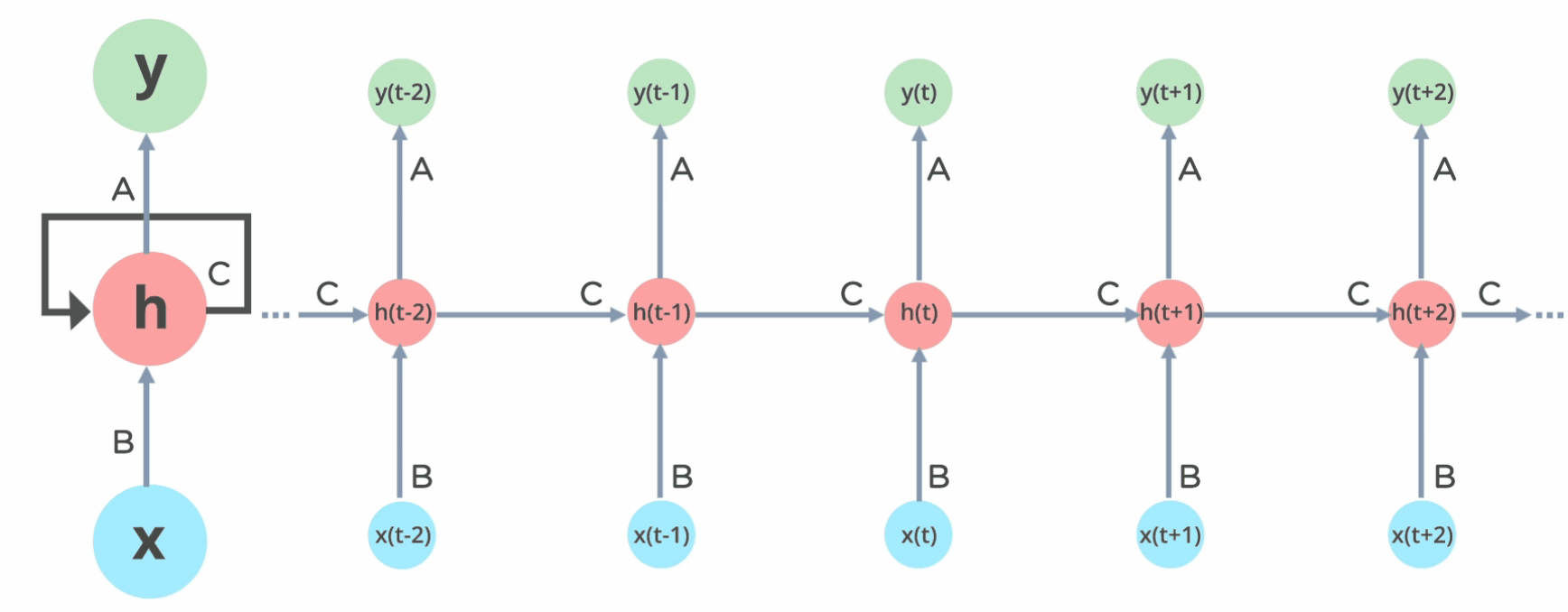
RNN architecture can vary depending on the problem you’re trying to solve. From those with a single input and output to those with many (with variations between).

Below are some examples of RNN architectures that can help you better understand this.

* **One To One:**There is only one pair here. A one-to-one architecture is used in traditional neural networks.
* **One To Many:**A single input in a one-to-many network might result in numerous outputs. One too many networks are used in the production of music, for example.
* **Many To One:**In this scenario, a single output is produced by combining many inputs from distinct time steps. Sentiment analysis and emotion identification use such networks, in which the class label is determined by a sequence of words.
* **Many To Many:**For many to many, there are numerous options. Two inputs yield three outputs. Machine translation systems, such as English to French or vice versa translation systems, use many to many networks.

**How does Recurrent Neural Networks work?**

The information in recurrent neural networks cycles through a loop to the middle hidden layer.



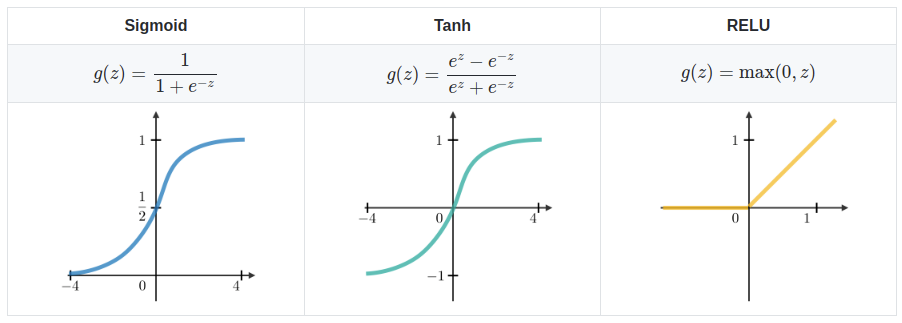
The input layer **x**receives and processes the neural network’s input before passing it on to the middle layer.

Multiple hidden layers can be found in the middle layer **h**, each with its own activation functions, weights, and biases. You can utilize a recurrent neural network if the various parameters of different hidden layers are not impacted by the preceding layer, i.e. There is no memory in the neural network.

The different activation functions, weights, and biases will be standardized by the Recurrent Neural Network, ensuring that each hidden layer has the same characteristics. Rather than constructing numerous hidden layers, it will create only one and loop over it as many times as necessary.

**Common Activation Functions**

A neuron’s activation function dictates whether it should be turned on or off. Nonlinear functions usually transform a neuron’s output to a number between 0 and 1 or -1 and 1.



The following are some of the most commonly utilized functions:

* **Sigmoid:**The formula **g(z) = 1/(1 + e^-z)** is used to express this.
* **Tanh:**The formula **g(z) = (e^-z – e^-z)/(e^-z + e^-z)** is used to express this.
* **Relu:**The formula **g(z) = max(0 , z)** is used to express this.

Advantages and disadvantages of RNN

**Advantages of RNNs:**

* Handle sequential data effectively, including text, speech, and time series.
* Process inputs of any length, unlike feedforward neural networks.
* Share weights across time steps, enhancing training efficiency.

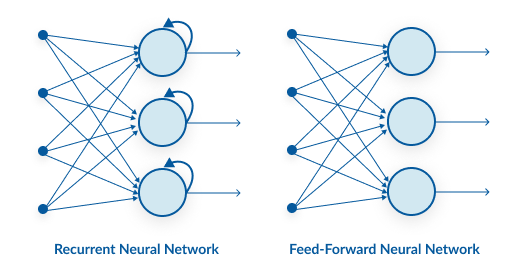
**Disadvantages of RNNs:**

* Prone to vanishing and exploding gradient problems, hindering learning.
* Training can be challenging, especially for long sequences.
* Computationally slower than other neural network architectures.

**Recurrent Neural Network Vs Feedforward Neural Network**

A feed-forward neural network has only one route of information flow: from the input layer to the output layer, passing through the hidden layers. The data flows across the network in a straight route, never going through the same node twice.

The information flow between an [RNN](https://www.analyticsvidhya.com/blog/2021/06/recurrent-neural-networks-introduction-for-beginners/) and a feed-forward neural network is depicted in the two figures below.

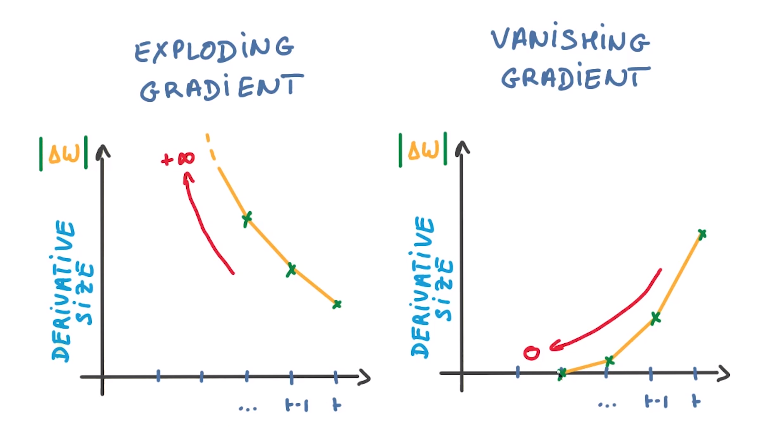


Feed-forward neural networks are poor predictions of what will happen next because they have no memory of the information they receive. Because it simply analyses the current input, a feed-forward network has no idea of temporal order. Apart from its training, it has no memory of what transpired in the past.

The information is in an RNN cycle via a loop. Before making a judgment, it evaluates the current input as well as what it has learned from past inputs. A recurrent neural network, on the other hand, may recall due to internal memory. It produces output, copies it, and then returns it to the network.

**Two issues of Standard RNNs**

There are two key challenges that RNNs have had to overcome, but in order to comprehend them, one must first grasp what a gradient is.



 With regard to its inputs, a gradient is a partial derivative. If you’re not sure what that implies, consider this: a gradient quantifies how much the output of a function varies when the inputs are changed slightly.

A function’s slope is also known as its gradient. The steeper the slope, the faster a model can learn, the higher the gradient. The model, on the other hand, will stop learning if the slope is zero. A gradient is used to measure the change in all weights in relation to the change in error.

* **Exploding Gradients:**Exploding gradients occur when the algorithm gives the weights an absurdly high priority for no apparent reason. Fortunately, truncating or squashing the gradients is a simple solution to this problem.
* **Vanishing Gradients:**Vanishing gradients occur when the gradient values are too small, causing the model to stop learning or take far too long. This was a big issue in the 1990s, and it was far more difficult to address than the exploding gradients. Fortunately, Sepp Hochreiter and Juergen Schmidhuber’s LSTM concept solved the problem.

**RNN Applications**

Recurrent Neural Networks are used to tackle a variety of problems involving sequence data. There are many different types of sequence data, but the following are the most common: Audio, Text, Video, Biological sequences.

Using RNN models and sequence datasets, you may tackle a variety of problems, including:

* Speech recognition
* Generation of music
* Automated Translations
* Analysis of video action
* Sequence study of the genome and DNA