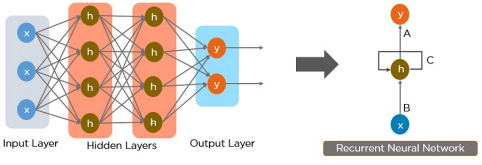
Recurrent Neural Nets

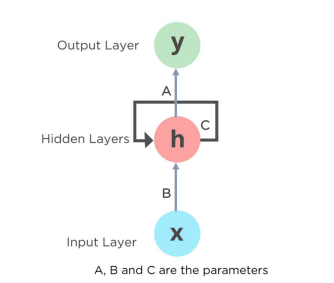
• RNNs are a type of neural network that can be used to model sequence data. • RNN works on the principle of saving the output of a particular layer and feeding this back to the input in orderto predict the output of the layer. • All of the inputs and outputs in standard neural networks are independent of one another. • However in some circumstances, such as when predicting the next word of a phrase, the prior words arenecessary, and so the previous words must be remembered. 

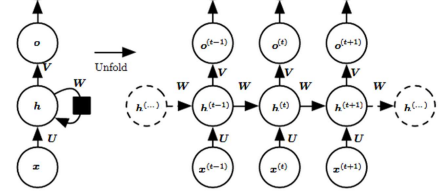
What is a RNN?

• 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 asequence.

Architecture



• Here, “x” is the input layer, “h” is

the hidden layer, and “o” is theoutput layer.

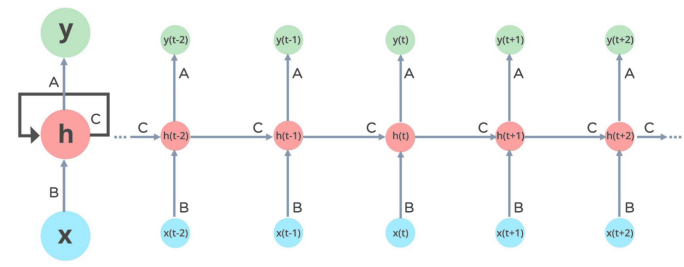
• W, V, and U are the networkparameters used to improve theoutput of the model.

• At any given time t, the currentinput is a combination of input atx(t) and h(t-1).

• The output at any given time isfetched back to the network toimprove on the output.

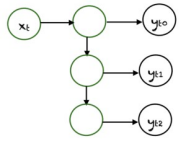
Unfolding a RNN

How does it work?

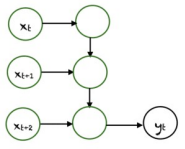


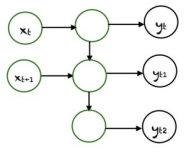
Types of RNNs

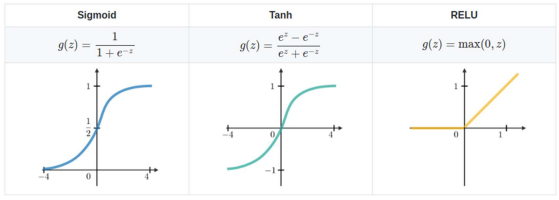
• One To One: There is only one pair here. A one to-one architecture is used in traditional neural networks. 

• 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.

Types of RNNs

• 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.

Common Activation Functions

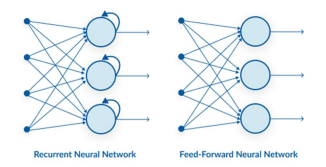
• Bidirectional Recurrent Neural Networks (BRNN) • In BRNN, inputs from future time steps are used to improve the accuracy ofthe network. It is like knowing the first and last words of a sentence to predictthe middle words. • Gated Recurrent Units (GRU)

• These networks are designed to handle the vanishing gradient problem. Theyhave a reset and update gate. These gates determine which information is tobe retained for future predictions. • Long Short Term Memory (LSTM)

• LSTMs were also designed to address the vanishing gradient problem inRNNs. LSTMs use three gates called input, output, and forget gate. Similar toGRU, these gates determine which information to retain.Various Architectures

A feed-forward neural network has only one route of information flow: from the input layer to theoutput layer, passing through the hidden layers. Feed-forward neural networks are poor predictions of what will happen next because they have nomemory of the information they receive.

Apart from its training, it has no memory of what transpired in the past.

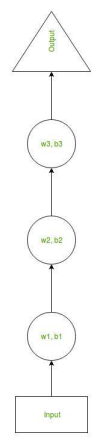
Feed-Forward Neural Networks vs The information is in an RNN cycle via a loop. Before making a judgment, it evaluates the current input as well as what it has learned from past inputs. A recurrentneural network, on the other hand, may recall due to internal memory. It produces output, copies it, and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

Recurrent Neural Networks

• RNN have a “memory” which remembers all information about whathas been calculated. • It uses the same parameters for each input as it performs the sametask on all the inputs or hidden layers to produce the output. • This reduces the complexity of parameters, unlike other neural

Introduction

networks.

• Suppose there is a deeper network with one input layer, three hidden layers and one output layer. 

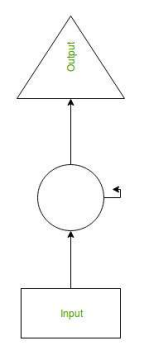
• Then like other neural networks, each hidden layer will have its own set of weights and biases, let’s say, • for hidden layer 1 the weights and biases are (w1, b1), • (w2, b2) for second hidden layer and

• (w3, b3) for third hidden layer.

• This means that each of these layers are independent of each other, i.e. they do not memorize the previous

How RNN works

outputs.

• Now the RNN will do the following: 

• RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous outputs by giving each output as input to the next hidden layer.

• Hence these three layers can be joined together such that the weights and bias of all the hidden layers is the same, into a single recurrent layer.

How RNN works

How RNN works

• Formula for calculating current state: 

• where:

• ht-> current state

• ht-1 -> previous state

• xt-> input state

How RNN works

• Formula for applying Activation function(tanh): 

• where:

• whh -> weight at recurrent neuron

• wxh -> weight at input neuron

How RNN works

• Formula for calculating output: 

• where,

• Yt-> output

• Why -> weight at output layer

How RNN works



How RNN works



How RNN works



How RNN works



• A single time step of the input is provided to the network. • Then calculate its current state using set of current input and the previousstate. • The current ht becomes ht-1 for the next time step. • One can go as many time steps according to the problem and join theinformation from all the previous states. • Once all the time steps are completed the final current state is used tocalculate the output. • The output is then compared to the actual output i.e., the target outputand the error is generated. • The error is then back-propagated to the network to update the weightsand hence the network (RNN) is trained.Training through RNN

• With regard to its inputs, a gradient is a partial derivative. If you’re not surewhat that implies, consider this: a gradient quantifies how much the outputof a function varies when the inputs are changed slightly. • A function’s slope is also known as its gradient. The steeper the slope, thefaster a model can learn, the higher the gradient. The model, on the otherhand, will stop learning if the slope is zero. A gradient is used to measurethe change in all weights in relation to the change in error. • Exploding Gradients: Exploding gradients occur when the algorithm givesthe 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 valuesare too small, causing the model to stop learning or take far too long. Thiswas a big issue in the 1990s, and it was far more difficult to address thanthe exploding gradients. Fortunately, Sepp Hochreiter and JuergenSchmidhuber’s LSTM concept solved the problem.Issues



• Advantages of Recurrent Neural Network • An RNN remembers each and every information through time. It is useful intime series prediction only because of the feature to remember previousinputs as well. This is called Long Short Term Memory. • Recurrent neural network are even used with convolutional layers to extendthe effective pixel neighborhood. • Disadvantages of Recurrent Neural Network • Gradient vanishing and exploding problems.

• Training an RNN is a very difficult task.

• It cannot process very long sequences if using tanh or relu as an activation

Advantages and Disadvantages function.