Recurrent Neural Network

08 December 2023 20:30

Sequential data / time-series

(historical information)

↓

forecast future values

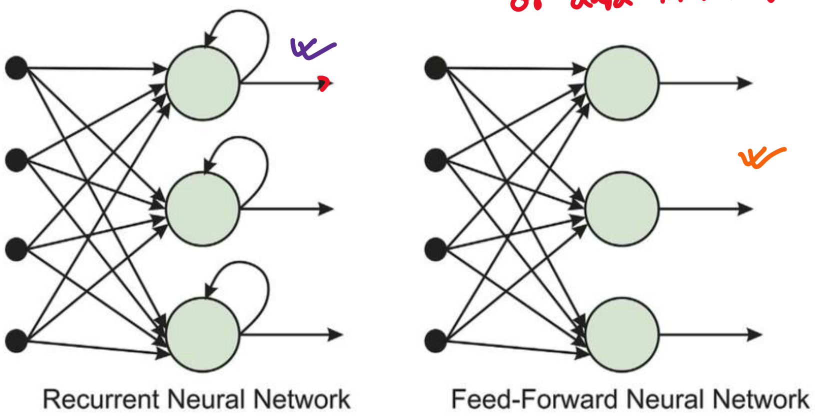
Recurrent Neural Network (RNN)

Recurrent neural networks are deep learning models that are typically used to solve time series problems.

The fundamental feature of a ***Recurrent Neural Network (RNN)*** is that the network contains at least one ***feed-back connection***, so the activations can flow round in a loop.

A RNN is a special type of ANN adapted to work for time series data or data that involves sequences.

'memory'



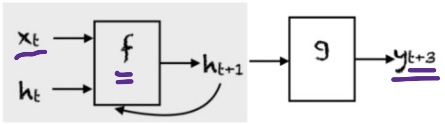
*Speech recognition

That enables the networks to do ***temporal processing*** and ***learn sequences***, e.g., perform sequence recognition/reproduction or temporal association/prediction.

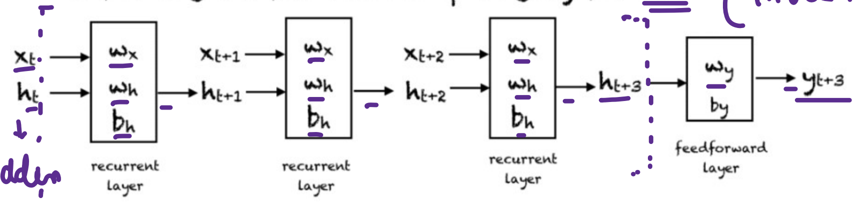
RNN

time step i/p → o/p

$t, t+1, t+2 \rightarrow t+3$



Unfolding the feedback loop in gray for k=3 (three time steps)



single feature (scalar)

hidden units

m = no of hidden units

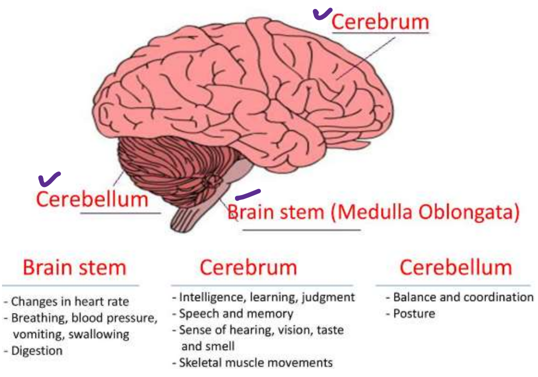
- $x_t \in R$ is the input at time step t.
- $y_t \in R$ is the output of the network at time step t.
- $h_t \in R^m$ vector stores the values of the hidden units/states at time t.
- $w_x \in R^m$ are weights associated with inputs in the recurrent layer
- $w_h \in R^{mxm}$ are weights associated with hidden units in the recurrent layer
- $w_y \in R^m$ are weights associated with hidden units to output units
- $b_h \in R^m$ is the bias associated with the recurrent layer
- $b_y \in R$ is the bias associated with the feedforward layer

The three main parts of the brain are:

NN - mimic the human brain
↓
weights (wi)

epoch → $\underline{\underline{W}}$ ↓
optimised

- The cerebrum
- The brainstem
- The cerebellum



Brain stem	Cerebrum	Cerebellum
- Changes in heart rate - Breathing, blood pressure, vomiting, swallowing - Digestion	- Intelligence, learning, judgment - Speech and memory - Sense of hearing, vision, taste and smell - Skeletal muscle movements	- Balance and coordination - Posture

Arguably the most important part of the brain is the cerebrum. It contains four lobes:

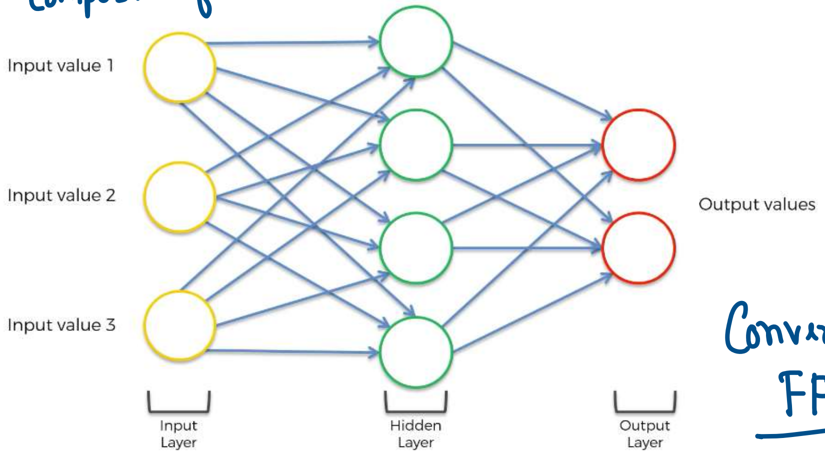
- ✓ The frontal lobe → Short term memory ← (RNN)
- ✓ The parietal lobe
- ✓ The temporal lobe → long term memory ← Artificial Neural Network (ANN)
- ✓ The occipital lobe → vision ← (CNN) (computer vision problems)

The main innovation that neural networks contain is the idea of weights.

The ability of a neural network to change its weights through each epoch of its training stage is similar to the long-term memory that is seen in humans

As mentioned, recurrent neural networks are used to solve time series problems. They can learn from events that have happened in recent previous iterations of their training stage. In this way, they are often compared to the frontal lobe of the brain – which powers our short-term memory.

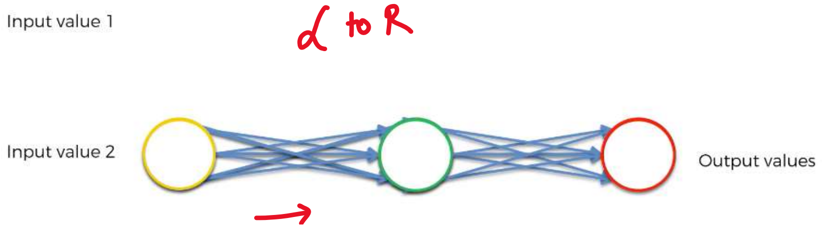
Composition of a Recurrent Neural Network (RNN)

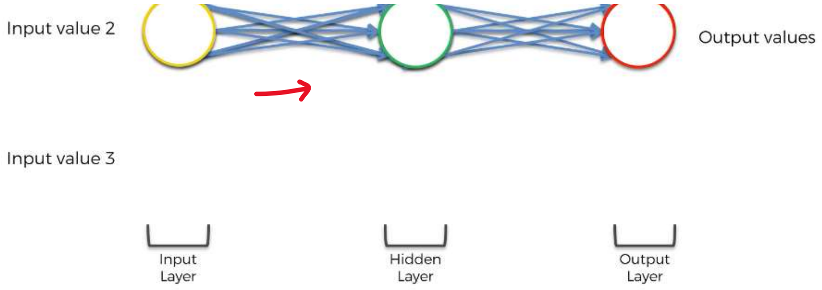


Conventional FFNN

✓ each layer of the network are squashed together

L to R

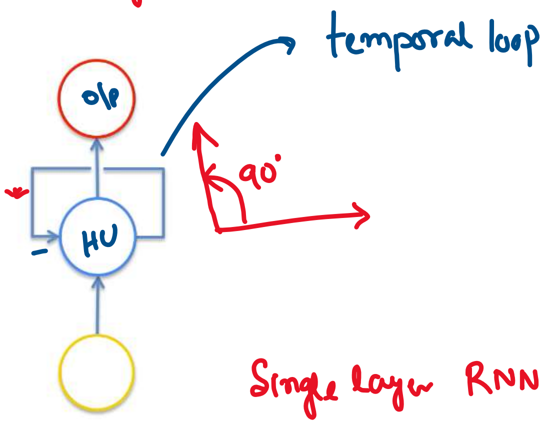




2nd :- Simplify neural network's neuron Synapse into a single line

3rd :- Entire NN → rotated 90°

4th :- Loop around the hidden layer

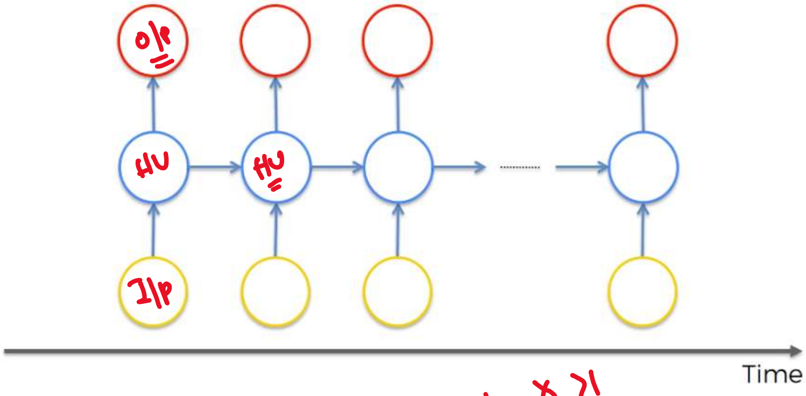


Single Layer RNN

Gradient Descent

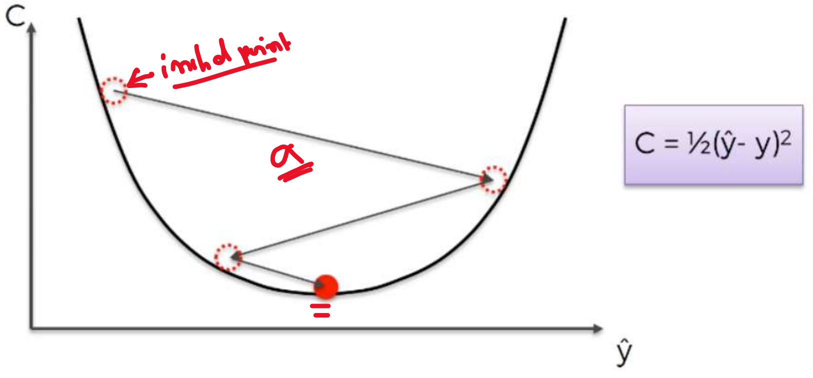
Optimization technique

↳ global minimum

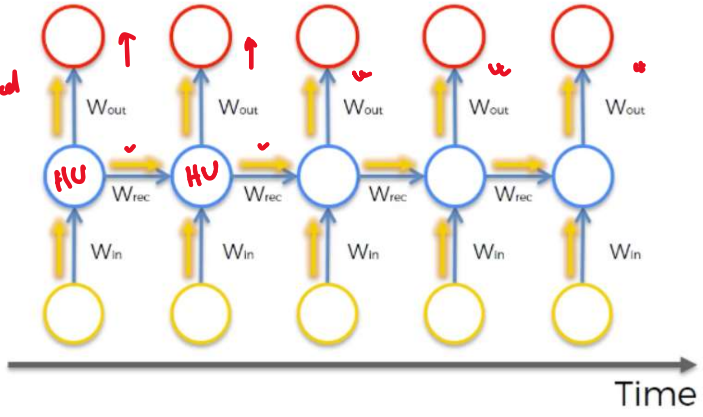


10 × 0.9 - 9 × 0.9 = 0.81 × 0.9

any number × <1 ↔ Vanishing / Exploding any number × >1



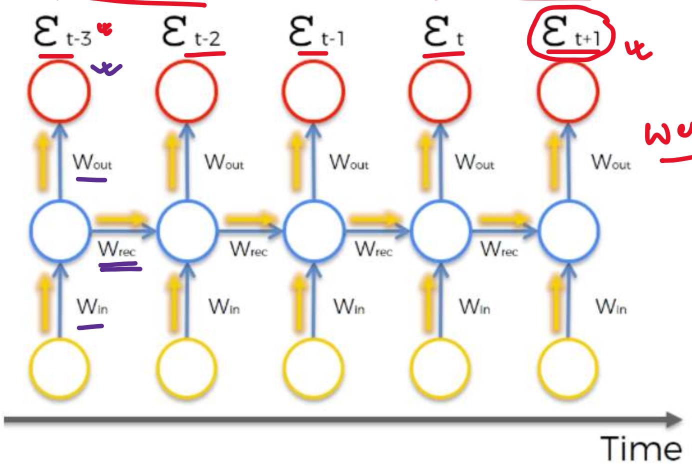
Cost function = Calculated for each observation



Gradient descent + feedback (backpropagation)
↓
updating the neuron synapse (weights)

deep level

Shallow level → O/P



weight t+1

Cost function computed at a deep layer of the NN will be used to change the weights of neuron at Shallower layers

- When `Wrec` is small, you experience a vanishing gradient problem (qualitative analysis)
- When `Wrec` is large, you experience an exploding gradient problem

To summarize, the vanishing gradient problem is caused by the multiplicative nature of the backpropagation algorithm. It means that gradients calculated at a deep stage of the recurrent neural network either have too small of an impact (in a vanishing gradient problem) or too large of an impact (in an exploding gradient problem) on the weights of neurons that are shallower in the neural net.

Solving the Vanishing Gradient Problem

Weight =?

Weight initialization is one technique that can be used to solve the vanishing gradient problem. It involves artificially creating an initial value for weights in a neural network to prevent the backpropagation algorithm from assigning weights that are unrealistically small.

(Long Short term Memory)

LSTMs are used in problems primarily related to speech recognition, with

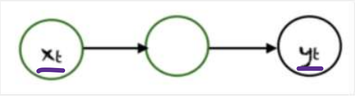
LSTMs are used in problems primarily related to speech recognition, with one of the most notable examples being Google using an LSTM for speech recognition in 2015 and experiencing a 49% decrease in transcription errors.

- When `Wrec` is small, you experience a vanishing gradient problem
- When `Wrec` is large, you experience an exploding gradient problem

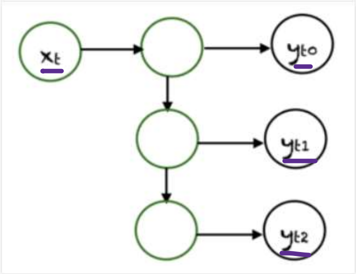
Quantify

- When `Wrec < 1`, you experience a vanishing gradient problem
- When `Wrec > 1`, you experience an exploding gradient problem

One to One

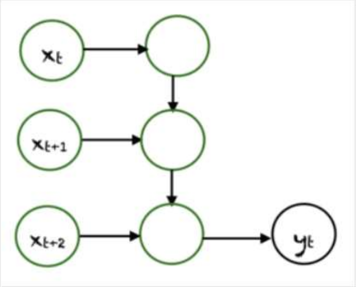


One to Many



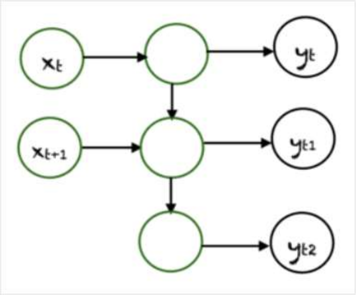
Eg :- Music Generation

Many to One



Sentiment Analysis, emotion detection

Many to Many



Machine Translation

RNNs have various advantages, such as:

- Ability to handle sequence data
- Ability to handle inputs of varying lengths
- Ability to store or "memorize" historical information

The disadvantages are:

- The computation can be very slow.
- The network does not take into account future inputs to make decisions.
- Vanishing gradient problem, where the gradients used to compute the weight update may get very close to zero, preventing the network from learning new weights. The deeper the network, the more pronounced this problem is. → LSTM