SNS COLLEGE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

Approved by AICTE & Affiliated to Anna University
Accredited by NBA & Accrediated by NAAC with 'A+' Grade,
Recognized by UGC saravanampatti (post), Coimbatore-641035.

Department of Biomedical Engineering

Course Name: 19ECT303 & Artificial Intelligence and machine learning

III Year : V Semester

Unit V–DEEP LEARNING

Topic : Bi directional Recurrent Neural Networks

BRNN Vs RNN

- An architecture of a neural network called a bidirectional recurrent neural network (BRNN) is made to process sequential data.
- In order for the network to use information from both the past and future context in its predictions, BRNNs process input sequences in both the forward and backward directions. This is the main distinction between BRNNs and conventional recurrent neural networks.

2 Distinct hidden layers

A BRNN has two distinct recurrent hidden layers, one of which processes the input sequence forward and the other of which processes it backward. After that, the results from these hidden layers are collected and input into a prediction-making final layer. Any recurrent neural network cell, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit, can be used to create the recurrent hidden layers.

Forward and reverse direction

The BRNN functions similarly to conventional recurrent neural networks in the forward direction, updating the hidden state depending on the current input and the prior hidden state at each time step. The backward hidden layer, on the other hand, analyses the input sequence in the opposite manner, updating the hidden state based on the current input and the hidden state of the next time step.

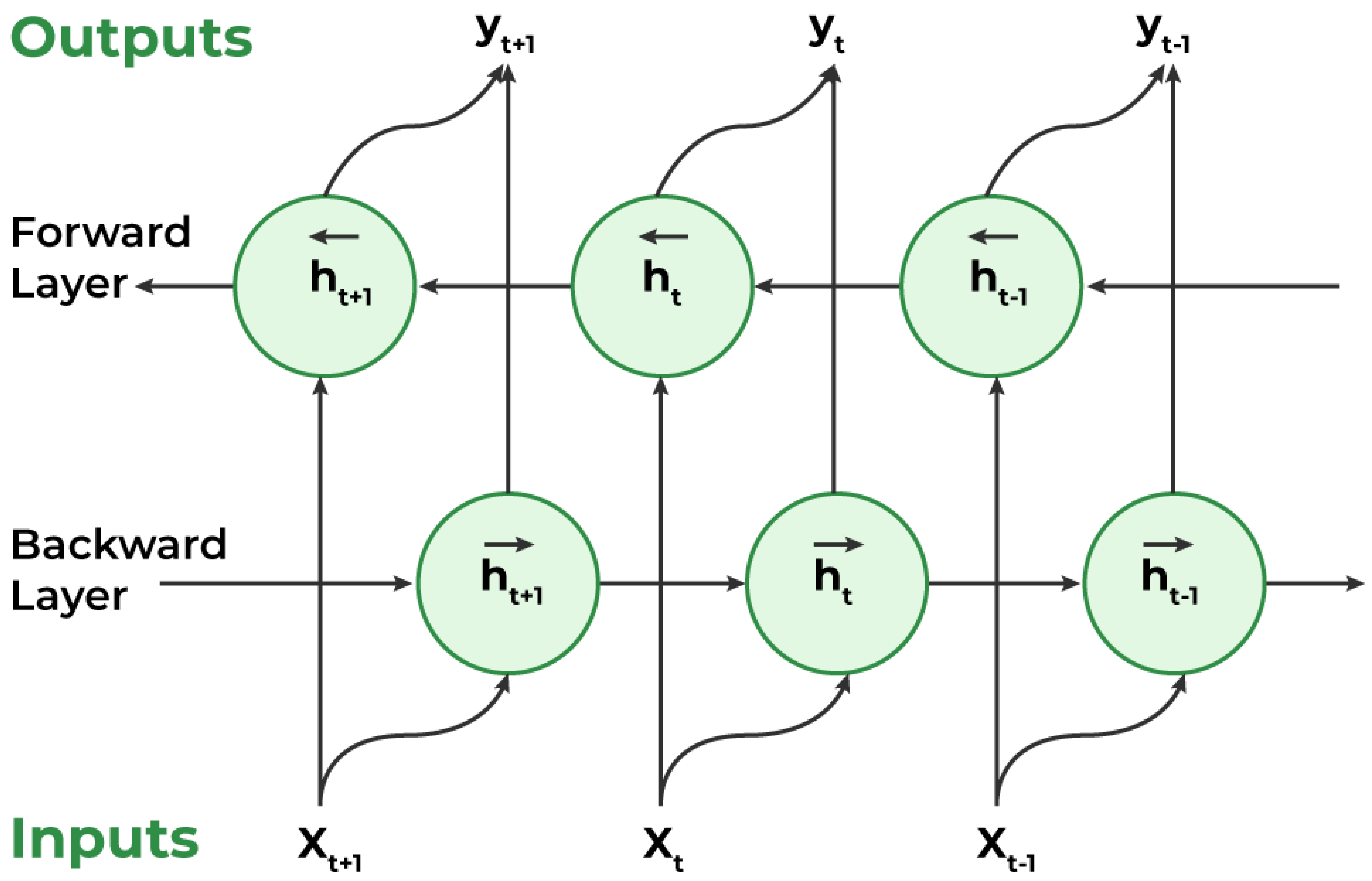
Bidirectional analysis

Compared to conventional unidirectional recurrent neural networks, the accuracy of the BRNN is improved since it can process information in both directions and account for both past and future contexts. Because the two hidden layers can complement one another and give the final prediction layer more data, using two distinct hidden layers also offers a type of model regularisation.

Forward and Backward gradient

In order to update the model parameters, the gradients are computed for both the forward and backward passes of the backpropagation through the time technique that is typically used to train BRNNs. The input sequence is processed by the BRNN in a single forward pass at inference time, and predictions are made based on the combined outputs of the two hidden layers. layers.

BRNN Architecture

- 

Working of BRNN

.

1. **Inputting a sequence:** A sequence of data points, each represented as a vector with the same dimensionality, are fed into a BRNN. The sequence might have different lengths.
2. **Dual Processing:** Both the forward and backward directions are used to process the data. On the basis of the input at that step and the hidden state at step t-1, the hidden state at time step t is determined in the forward direction. The input at step t and the hidden state at step t+1 are used to calculate the hidden state at step t in a reverse way.

CONTD . . .

3. **Computing the hidden state:** A non-linear activation function on the weighted sum of the input and previous hidden state is used to calculate the hidden state at each step. This creates a memory mechanism that enables the network to remember data from earlier steps in the process.
4. **Determining the output:** A non-linear activation function is used to determine the output at each step from the weighted sum of the hidden state and a number of output weights. This output has two options: it can be the final output or input for another layer in the network.

CONTD . . .

5. **Training:** The network is trained through a supervised learning approach where the goal is to minimize the discrepancy between the predicted output and the actual output. The network adjusts its weights in the input-to-hidden and hidden-to-output connections during training through backpropagation.

Output calculation in BRNN

To calculate the output from an RNN unit, we use the following formula:

```
Ht (Forward) = A(Xt * WXH (forward) + Ht-1 (Forward) * WHH (Forward) +
bH (Forward)
Ht (Backward) = A(Xt * WXH (Backward) + Ht+1 (Backward) * WHH
(Backward) + bH (Backward)

where,
A = activation function,
W = weight matrix
b = bias
```

CONTD . . .

-

The hidden state at time t is given by a combination of H_t (Forward) and H_t (Backward). The output at any given hidden state is :

$$Y_t = H_t * W_{AY} + b_y$$

Applications of BRNN

- Sentiment analysis
- Named entity Recognition
- Part of speech Tagging
- Machine Translation
- Speech recognition

Advantages of BRNN

- Context from both past and future
- Enhanced accuracy
- Efficient handling of variable-length sequences
- Resilience to noise and irrelevant information
- Ability to handle sequential dependencies

Disadvantages of BRNN

- Computational complexity
- Long training time
- Difficulty in parallelization
- Overfitting
- Interpretability

THANK YOU !!!
HAPPINESS ISN'T OUTSIDE, ITS WITHIN