**IBM-Recurrent Neural Networks** (https://www.ibm.com/cloud/learn/recurrent-neural-networks)

A type of artificial neural network which uses sequential data or time series data. Utilize training data to learn. They are distinguished by their memory as they take information from prior inputs to influence the current input and output, heavily depends on the prior elements of a given sequence. This is divergent of traditional deep neural networks as they see inputs and outputs as independent factors.

Icon

Description automatically generated

RNN share parameters across each layer of the network. RNN leverages backpropagation through time (BPTT) to determine the gradients, model trains itself by calculation errors from its output layer to its input layers. Used a reduced number of hidden layers with in the neural network. Too little, it becomes insignificant. Too large, the will become NaN.

Text

Description automatically generated

**Bidirectional recurrent neural networks (BRNN)**: pulls in future data to improve the accuracy of it. EX: “feeling under the weather”, knowing the last word would be “weather”, the model will better predict “under”

**Long short-term memory (LSTM)**: A solution to vanishing gradient problem. Vanishing gradient issue: multiply many small numbers together, errors due to further back time steps have smaller gradients, bias parameters to capture short-term dependencies. Works to address the problem of long-term dependencies. If the previous state that is influencing the current prediction is not in the recent past, the RNN model may not be able to accurately predict the current state. Solution: LSTM have cells in the hidden layers of the neural network, which have three gates (input, output, and forget).

1. Activation functions: use ReLU-prevents the shrinkage of the gradient
2. Parameters (weights): Initialize weights to identity matrix, initialize biases to zero, this helps prevent weight from shrinking to zero.
3. Gated Cells: use a more complex recurrent unit with gates to control information passed through.

Diagram, schematic

Description automatically generated

1. Forget irrelevant parts of the previous state
2. Store relevant new information into the new cell state
3. Selectively updates cell state values
4. Output gate controls what information is sent to the next time step

Together-uninterrupted gradient flow

**Other Materials:**

Cryptocurrency Price Forecasting (<https://www.kaggle.com/taniaj/cryptocurrency-price-forecasting>)

RNN Slides: (<https://udrc.eng.ed.ac.uk/sites/udrc.eng.ed.ac.uk/files/attachments/UDRC_RNN_LSTM_LibrariesTutorial.pdf>)

Stock Prediction using RNN: (<https://towardsdatascience.com/stock-prediction-using-recurrent-neural-networks-c03637437578>)

RNN Science Direct: (<https://www.sciencedirect.com/science/article/pii/S1877050920304865>)