**RNN on Time Series Data**

**Introduction**

The forecasting of time series represents an essential procedure suitable for multiple disciplines that need to predict temperature conditions and environmental elements. The following document investigates several deep learning models that predict temperatures through time series data analysis. The developed forecasting models consist of:

1. Simple RNN (Recurrent Neural Network)
2. Stacked RNN
3. GRU (Gated Recurrent Unit)
4. LSTM (Long Short-Term Memory)
5. A system which combines 1D Convolutional Neural Networks (CNN) with RNNs exists for this analysis.

The evaluation yielded performance assessments through three metrics including training curve monitoring alongside validation loss assessment and testing prediction analysis.

**Dataset Overview**

The research uses Jena Climate Dataset spanning from 2009 to 2016 to evaluate temperature data collected by the Jena station every ten minutes along with pressure and humidity values and additional weather components. Prediction of temperature serves as the target variable yet accepts input from different weather-related parameters.

The data collection occurs at an interval of one point every ten minutes with a sampling rate of 6.

Our model considers a sequence consisting of 120 past time steps to analyze future temperature outcomes.

The data distribution divides into three sections as training and validation and testing while using batch size 256.

Model Architectures

**1. Simple RNN**

A Simple RNN constitutes the basic design among recurrent neural networks because its recurrent layer conducts feedback from each neuron to rest. The architecture is as follows:

**Layer 1**: The system contains as a simple RNN with 64 units which offers sequence feedback.

**Layer 2:** Simple RNN with 16 units (no return sequences).

**Output Layer:** Dense layer with a single unit for temperature prediction.

The training process lasted for 30 epochs using MSE as the loss function along with the Adam optimization method. Since Simple RNN shows limited potential in capturing long-term relationships between sequence elements it functions as a foundation to understand more advanced architecture types.

**2. Stacked RNN**

The Stacked RNN consists of the Simple RNN model with multiple layers placed consecutively on top of one another. Through such architecture the model gains ability to recognize intricate patterns along with advanced temporal relationships. The architecture is as follows:

**Layer 1**: The first RNN layer contains 64 units along with sequence return capability.

**Layer 2:** The second level consists of a Simple RNN with 32 processing units that generates sequence output.

**Layer 3:** Simple RNN with 16 units (no return sequences).

**Output Layer:** Dense layer with a single unit for temperature prediction.

The model trains and validates similarly to the Simple RNN structure with multiple layers that enable the detection of deep temporal dependencies.

**3. GRU (Gated Recurrent Unit)**

GRU represents an enhanced RNN design which fixes the gradient vanishing issue during processing. The gating system in GRU controls information transmission while operating with fewer parameters than LSTMs. The architecture is as follows:

**Layer 1:** The first layer includes a GRU with 64 units that generates sequence information.

**Layer 2:** The second layer contains a GRU network with 32 units but it enables sequence outputs.

**Layer 3:** GRU layer with 16 units (no return sequences).

**Output Layer:** Dense layer with a single unit for temperature prediction.

The time series forecasting performance of GRU surpasses that of Simple RNN because GRU demonstrates more efficiency and superior capability in extracting long-term dependencies from the data.

**4. LSTM (Long Short-Term Memory)**

LSTM functions as an enhancement of RNN because it has been created to solve the long-term dependency limitations in sequential data. Every LSTM unit contains a memory cell that enables storage of vital information across extended periods. The architecture is as follows:

**Layer 1:** The first LSTM layer contains 64 units that process sequential data through its returning sequence structure.

**Layer 2:** The second layer contains an LSTM unit arrangement with 32 units which generates sequence output.

**Layer 3:** LSTM layer with 16 units (no return sequences).

**Output Layer:** Dense layer with a single unit for temperature prediction.

The analysis phase consists of LSTM capturing effective long-term dependency patterns which makes this architecture ideal for temperature forecasting operations.

**5. 1D Convolutional Neural Network (CNN) Combined with RNN (CNN+RNN)**

RNN and CNN components of the CNN+RNN model work together to exploit both Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) advantages. The technical capacity of CNNs to detect local data patterns matches the ability of RNNs to handle temporal patterns. The architecture is as follows:

**Layer 1:** 1D Convolutional layer with 32 filters and kernel size 3.

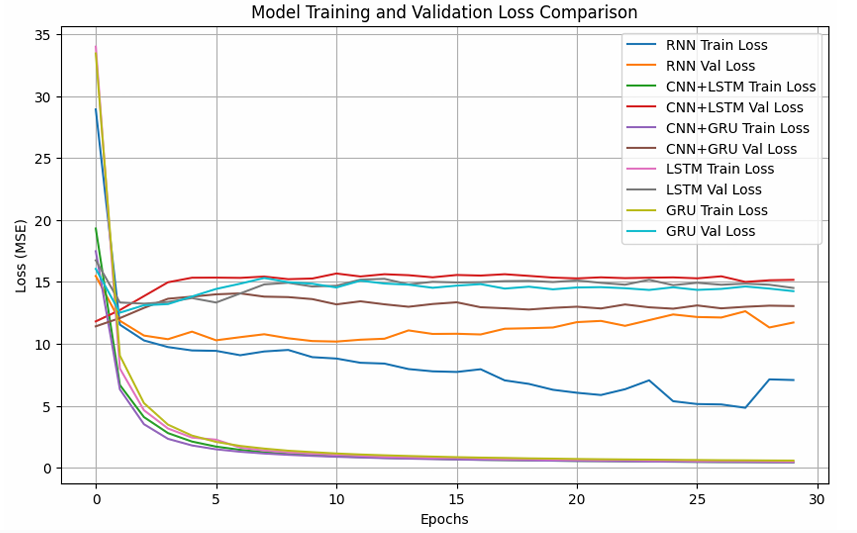
**Layer 2:** MaxPooling layer with pool size 2.

**Layer 3:** The model contains a Layer 3 with 64 units in an LSTM or GRU layer that produces sequence outputs.

**Layer 4:** Dense layer with a single unit for temperature prediction.

This model architecture which combines convolutional features with sequential learning is anticipated to achieve superior performance during training and validation steps because it adopts features from both approaches. Through CNN layers the model can obtain spatial features from time series data while RNN layers detect temporal patterns.

**Models Performance:**



The **CNN+GRU model** demonstrates the best training ability according to the displayed graphical data. The training process of CNN+GRU demonstrates the lowest training loss (MSE) measurements at every point. The model keeps its position as the lowest point of the graph from beginning to end. The **CNN+GRU model** demonstrates quick learning through its quick reduction of training loss during the first training epochs.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Training Loss Trend (Lower is Better)** | **Validation Loss Trend (Lower is Better)** | **Notes** |
| **RNN** | Moderate, plateaus at higher loss | Moderate, plateaus at higher loss | Higher loss overall compared to other models. Shows signs of underfitting. |
| **CNN+LSTM** | Low, but not the lowest | Moderate, plateaus higher than training | Decent training, but validation loss is significantly higher, suggesting potential overfitting. |
| **CNN+GRU** | Lowest, rapid convergence | Low, plateaus slightly higher than train | Best training performance. Validation loss is slightly higher than training but still relatively low, indicating a good balance between fit and generalization. |
| **LSTM** | Low, converges quickly | Low, converges quickly | Performs well in both training and validation, but not as well as CNN+GRU. |
| **GRU** | Low, converges quickly | Low, converges quickly | Similar to LSTM, performs well but not as well as CNN+GRU. |

**Evaluation of the Model:**

