**Gated Recurrent Units (GRU)**

The article introduces Gated Recurrent Units (GRUs) as a simplified version of LSTM that retains the key idea of incorporating an internal state and multiplicative gating mechanisms, but with faster computation. GRUs replace the three gates of LSTM with two gates: the reset gate and the update gate. These gates are given sigmoid activations, determining the amount of previous state to remember and controlling the amount of new state that is a copy of the old one, respectively.

The article then provides implementations of GRUs from scratch using different deep learning frameworks: PyTorch, MXNet, JAX, and TensorFlow. The implementations include initializing model parameters and defining the forward computation of the GRU, which updates the hidden state based on the input and the previous hidden state using the reset and update gates.

The article also shows how to train a language model using the GRU implementation and The Time Machine dataset. The training process and hyperparameters remain the same as in previous sections. After training, the article demonstrates how to predict the next sequence based on a given prefix using the trained GRU model.

In the concise implementation section, the article introduces a high-level API for instantiating a GRU model, which encapsulates the configuration details and provides faster computation using compiled operators.

The article concludes by mentioning exercises for further exploration, such as determining the best values for the reset and update gates for different time steps, adjusting hyperparameters and analyzing their influence, comparing runtime and performance between different GRU implementations, and experimenting with partial GRU implementations.

Overall, the article provides a comprehensive overview of GRUs, including their architecture, implementation from scratch, training, and usage in a language modeling task. It also encourages further experimentation to deepen understanding and explore different aspects of GRUs.

**Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction**

Summary

This document explores the use of LSTM and GRU neural network methods for traffic flow prediction. It discusses the limitations of existing models and the need for more accurate and real-time prediction methods. Experimental results show that LSTM and GRU outperform traditional ARIMA models. The section also includes an overview of parameter and non-parameter models, the structure and computation process of LSTM and GRU cells, and the experimental design and data used. The following section discusses the experimental results, which show that LSTM and GRU outperform ARIMA models, with GRU demonstrating better performance than LSTM. Future work includes testing RNNs with more hidden states and optimal time lags.

Neural networks for traffic flow prediction

This section of the document discusses the use of LSTM (Long Short Term Memory) and GRU (Gated Recurrent Units) neural network methods for traffic flow prediction. It highlights the limitations of existing linear models and the need for more accurate and real-time prediction methods in Intelligent Transportation Systems (ITS). The paper explores the application of deep learning methods, specifically LSTM and GRU, for short-term traffic flow prediction. Experimental results show that LSTM and GRU perform better than traditional ARIMA models. It also mentions that GRU has not been previously used for traffic flow prediction. The section provides a brief overview of parameter and non-parameter models used in traffic flow prediction and highlights the advantages and challenges associated with each. It explains the structure and computation process of LSTM and GRU cells in detail. The section concludes with an overview of the experimental design and data used, which is collected from the Performance Management System (PeMS) dataset in California.

Traffic flow models

This section of the document discusses the experimental results of comparing different models for traffic flow prediction. The models used in the experiment are ARIMA, LSTM NN, and GRU NN. The results show that LSTM NN and GRU NN outperform the ARIMA model. The MSEs and MAEs of the models are plotted in figures 3, 4, 5, and 6. The distribution of MSEs for the RNN models tend to be smaller, indicating better performance. The distribution of MAEs is more evenly distributed. The conclusion is that GRU NNs have better performance than LSTM NNs and ARIMA models. Future work includes testing RNNs with more hidden states and the optimal time lags.

**Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling**

Summary

The document discusses the empirical evaluation of gated recurrent neural networks (RNNs) for sequence modeling. The authors compare long short-term memory (LSTM) units and gated recurrent units (GRU) to traditional RNN units, finding that the advanced units, especially GRU, outperform the traditional units in terms of convergence and generalization. The document also explores the advantages and differences between LSTM and GRU units, highlighting the need for more comprehensive comparisons to determine which unit performs better in different tasks.

Comparing RNN units

This section of the document introduces the topic of empirical evaluation of gated recurrent neural networks (RNNs) on sequence modeling. The authors compare different types of recurrent units, specifically long short-term memory (LSTM) units and gated recurrent units (GRU), to more traditional RNN units such as tanh units. They evaluate these recurrent units on tasks of polyphonic music modeling and speech signal modeling. The experiments reveal that the advanced recurrent units, particularly GRU, outperform the traditional units in terms of convergence, parameter updates, and generalization. The section also provides background information on RNNs and explains the functioning of LSTM and GRU units.

RNN unit differences

This section of the document discusses the advantages and differences between two types of recurrent neural network units: the Long Short-Term Memory (LSTM) unit and the Gated Recurrent Unit (GRU). The section explains that the additive nature of these units allows for long-term memory retention and the creation of shortcut paths, which helps prevent the vanishing gradient problem. The section further describes the differences between the two units, such as the controlled exposure of memory content and the location of the input gate. The section concludes by stating that more thorough empirical comparisons are needed to determine which unit performs better in different tasks. The section then moves on to the experimental settings, tasks, and datasets used for the evaluation of these units. The section concludes by presenting the results of the experiments, which demonstrate the superiority of the gated units over the traditional unit in terms of convergence and final performance. However, the experiments do not provide conclusive evidence on which gated unit, LSTM or GRU, is better.

**Improving speech recognition by revising gated recurrent units**

Summary

The document discusses improvements to speech recognition using a revised version of Gated Recurrent Units (GRUs), with the removal of the reset gate and the use of Rectified Linear Units (ReLU) activations. This revised architecture reduces training time by over 30% and consistently improves recognition performance across different tasks, input features, and noisy conditions. The concept of batch normalization is also explored, with comparable performance observed when applied to RNNs. The experimental setup and results are presented, showing the proposed architecture achieving the best performance in terms of error rates. The document concludes with a list of references related to deep learning, speech recognition, and neural networks.

Improved speech recognition

This section of the document discusses improvements to speech recognition using a revised version of Gated Recurrent Units (GRUs). The authors propose to remove the reset gate in the GRU design and replace the hyperbolic tangent activation function with Rectified Linear Units (ReLU) activations in the state update equations. The revised architecture reduces training time by more than 30% and consistently improves recognition performance across different tasks, input features, and noisy conditions compared to a standard GRU.

Batch normalization in RNNs; experimental setup and results

This section of the document discusses the concept of batch normalization and its application to RNNs. The authors tried two different approaches to applying batch normalization to RNNs and found comparable performance between them. They also observed that using batch normalization helped in avoiding numerical issues with RNNs applied to long time sequences. The section then goes on to discuss the experimental setup, including the training datasets and tasks used. The authors conducted experiments with TIMIT and a Wall Street Journal (WSJ) task in a distant-talking scenario. The section also provides details on the system architecture and hyperparameters used in the experiments. Results of the experiments on TIMIT and DIRHA English WSJ datasets are presented, showing the performance of various RNN architectures. The proposed architecture, which removed the reset gate and used ReLU activations, achieved the best performance in terms of phone error rate and word error rate across the different datasets and features tested. The proposed architecture also significantly reduced training time compared to a standard GRU. The section concludes by discussing future efforts to extend this work to different tasks and datasets.