Benchmark for Recurrent vs. Deep Feedforward Neural Networks

Fabio Schwinger

Motivation

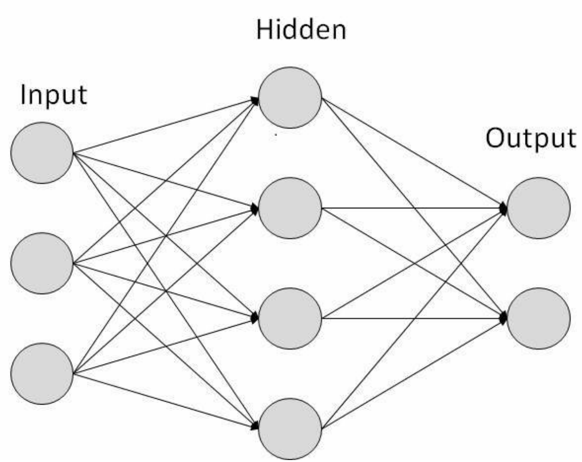
- Compare FFNN's and RNN's
- Preprocess the datasets based on scientific reasoning
- Narrow down hyperparameters with the help of literature research
- Implement hyperparameter tuning

Problems

- Imbalance of the datasets
- Sequence Length
- Preprocessing and cleaning
- Add missing prev_prev_content_id and prev_content_id

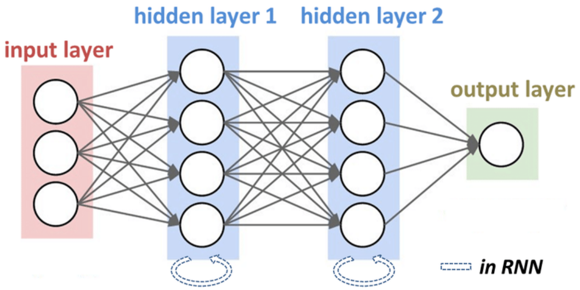
The structure of a feed forward neural

- Simplest neural network architecture
- Input feature vectors are 1-dimensional (single vector per instance)
- Supports only forward propagation
- Consists of at least one input and one output layer
- Can contain an arbitrary amount of hidden layers
    - Rarely more than 2 (Heaton, 2015)
- No cycles or loops



© Xun/Initial Margin Simulation with Deep Learning (2019)

The structure of a recurrent neural network



© Shukai et al.Classification and prediction … with machine learning techniques (2019)

- Remembers multiple inputs over time (Schmidt, 2019)
- Input feature vectors are 2-dimensional (sequence of vectors)
- Contains at least one input layer and one output layer, similar to FFNN
- Recurrence through backpropagation through time in hidden layers
  - Hidden layers utilize all timestamps of an input
- Capable of capturing temporal dependencies through feedback loops

universität wien

Applied preprocessing steps

- Embedding layers for categorical id values (Heaton, 2019)
- Feature Selection
- Feature Encoding
    - One-hot encoding
    - Ordinal encoding (Potdar et al., 2017)
    - Cyclical encoding (Van Wyk, 2022)
    - Scaling
- Adding prev_prev and prev content id's
- Train-, test- and validation split (Roshan, 2022)
- Handling of missing values
- Removal of low appearing content id's
- Removal of the last record of every session
- Deciding on a sequence length
- Add padding to the recurrent neural network (Dwarampudi & Subba Reddy, 2019)
    - Pre- and post sequence padding
    - Pre- and post sequence truncation

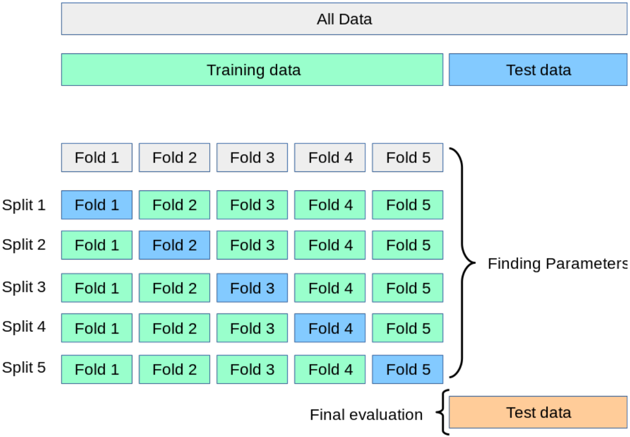
universität wien

Different hyperparameters

- Number of hidden layers (Heaton, 2015)
- Number of neurons (Heaton, 2015)
- Activation function (Sharma et al., 2017)
- Loss function (Chollet et al., 2015)
- Learning rate (Bengio, 2017)
- Optimizers (Choi et al., 2020)
- Batch Normalization (Bjorck et al., 2018)
- Dropout and Dropout rate (Srivastava et al., 2014)
- Batch size (Bengio, 2018)
- Epochs (Brownlee, 2018)

Cross validation

- Each fold gets used as test fold a single time
- Each fold gets used $n-1$ times as training fold
- Data Resampling method to evaluate the ability to classify previously unseen data (Berrar, 2018)
- Higher number of folds can lead to a low bias with a high variance (overfitting) (Kohavi, 1995)
- Lower number of folds can lead to a high bias with a low variance (underfitting) (Kohavi, 1995)



© Pedregosa et al./Scikit-learn: Machine Learning in Python (2011)

Implementation

- Implemented in python 3.12 and its well known libraries
  - pandas, numpy, scikit-learn, category_encoders, scipy, matplotlib, keras, tensorflow, optuna
- A package with 12 different subclasses
  - Each responsible for a specific subtask
- Mutliple feed forward and recurrent neural networks
  - Different types of Models
    - Preprocessed Datasets
    - No weather data
    - Only prev_prev_content_id, prev_content_id, content_id and session_id

universität wien

Results with preprocessed datasets

	# of content id's	# of sessions id's	# of samples	FFNN Accuracy	RNN Accuracy
small_original	24	79	173	51%	50%
large_original	49	74	284	25%	0%
large	53	118	284	26%	0%
large no_search	5	121	337	79%	38%
medium	215	955	8280	49%	16%
medium no_search	13	955	8331	82%	83%
small	24	36	127	37%	44%
small no_search	3	36	135	87%	100%
hotel s	86	142	947	51%	50%
hotel s no_search	8	142	995	85%	80%
hotel w	66	129	912	33%	45%
hotel w no_search	6	132	952	93%	91%

universität wien

Results with special sub datasets

	FFNN Accuracy	RNN Accuracy
small_original	49%	82%
large_original	25%	4%

Table: Results of the neural networks without weather data

	FFNN Accuracy	RNN Accuracy
small_original	24%	75%
large_original	37%	25%

Table: Results of the neural networks with only content_id, prev_content_id, prev_prev_content_id and session_id as features

	Accuracy
small_original	82%
large_original	0%

Table: Results of the recurrent neural networks without the prev and the prev_prev_content_id

Interpretation of the results

- No_search datasets significantly outperform the other variant of the dataset
  - No_search datasets have far less unique content id's
  - Otherwise these datasets are the same
- $9/10$ no_search dataset result in good model (accuracy $>= 70\%$)
  - None of other datasets exceed 51% with weather data
- FFNN's and RNN's have very similar results for most of the datasets
- Special subdatasets impact the RNN's in a more positive way

Lessons learned

- Improved knowledge about FFNN’s and RNN’s
- Hyperparameter tuning using libraries like Keras Tuner and Optuna
- Cleaning and preprocessing of datasets
  - The importance of these steps to achieve good neural networks

Techdemo

`https://github.com/RogueRefiner/BachelorThesis`

References

Kohavi, R. 1995. A Study of Cross-Validation and Bootstrap. for Accuracy Estimation and Model Selection.
Pedregosa, F. et al. 2011. Scikit-learn: Machine Learning in Python.
Srivastava, N. 2014. Dropout: a simple way to prevent neural networks from overfitting.
Heaton, J. 2015. Artificial Intelligence for Humans: Deep learning and neural networks.
Chollet, F. et al. 2015. Keras.
Sharma, S. et al. 2017. Activation functions in neural networks.
Bengio, Y. et al. 2017. Deep learning. Potdar, K. et al. 2017. A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers.
Berrar, D. 2018. Cross-Validation.
Bjorck, N. et al. 2018. Understanding batch normalization.
Bengio, Y. 2018. Practical recommendations for gradient-based training of deep architectures.

References

Brownlee, J. 2018. What is the Difference Between a Batch and an Epoch in a Neural Network.

Dwarampudi, M & Subba Reddy N. V. 2019. Effects of padding on LSTMs and CNNs.

Heaton, J. 2019. What are Embedding Layers in Keras (11.5).

Shukai, M. et al. 2019. Classification and prediction of wave chaotic systems with machine learning techniques.

Schmidt, R. 2019. Recurrent Neural Networks (RNNs): A gentle Introduction and Overview.

Xun, M. 2019. Initial Margin Simulation with Deep Learning.

Choi, D. et al. 2020. On Empirical Comparisons of Optimizers for Deep Learning.

Roshan, J. 2022. Optimal Ratio for data splitting.

Van Wyk, J. 2022. Encoding cyclical features for deep learning.