RinQ Fingerprinting: Recurrence-Informed Quantile Networks for Magnetic Resonance Fingerprinting

Elisabeth Hoppe[1]([✉]), Florian Thamm[1], Gregor Körzdörfer[2], Christopher Syben[1], Franziska Schirrmacher[1], Mathias Nittka[2], Josef Pfeuffer[2], Heiko Meyer[2], and Andreas Maier[1]

[1] Pattern Recognition Lab, Department of Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
{elisabeth.hoppe,florian.thamm}@fau.de
[2] MR Application Development, Siemens Healthcare GmbH, Erlangen, Germany

Abstract. Recently, Magnetic Resonance Fingerprinting (MRF) was proposed as a quantitative imaging technique for the simultaneous acquisition of tissue parameters such as relaxation times T_1 and T_2. Although the acquisition is highly accelerated, the state-of-the-art reconstruction suffers from long computation times: Template matching methods are used to find the most similar signal to the measured one by comparing it to pre-simulated signals of possible parameter combinations in a discretized dictionary. Deep learning approaches can overcome this limitation, by providing the direct mapping from the measured signal to the underlying parameters by one forward pass through a network. In this work, we propose a Recurrent Neural Network (RNN) architecture in combination with a novel quantile layer. RNNs are well suited for the processing of time-dependent signals and the quantile layer helps to overcome the noisy outliers by considering the spatial neighbors of the signal. We evaluate our approach using in-vivo data from multiple brain slices and several volunteers, running various experiments. We show that the RNN approach with small patches of complex-valued input signals in combination with a quantile layer outperforms other architectures, e.g. previously proposed Convolutional Neural Networks for the MRF reconstruction reducing the error in T_1 and T_2 by more than 80%.

Keywords: Deep learning · Recurrent Neural Networks · Magnetic resonance fingerprinting reconstruction

1   Introduction

One disadvantage of the most currently used Magnetic Resonance Imaging (MRI) techniques is the qualitative nature of the images, thus in most cases no absolute values of the underlying physical tissue parameters, e.g. T_1 and T_2 relaxations, are obtained. Magnetic Resonance Fingerprinting (MRF) was recently proposed to overcome this limitation: It provides an accelerated acquisition of time signals which differ with the various tissue types by using randomly modified parameters during the acquisition (e.g. Flip Angle (FA) or Repetition Time (TR)) and strong undersampling with spiral readouts. These signals are compared to simulated signals of possible parameter combinations of T_1 and T_2 and quantitative maps are reconstructed [7,8]. However, this state-of-the-art approach suffers from high computational effort: Every measured signal is compared to every simulated signal using template matching algorithms. Due to storage and computational limitations, this dictionary can only have a finite amount of possibilities and thus the maps are limited to these parameter combinations and can be erroneous [13]. The more combinations the dictionary contains, the more expensive is the reconstruction in terms of time and storage. In order to provide continuous predictions, to accelerate this process and to eliminate the burden of high storage requirements during the reconstruction, deep learning (DL) can be used: Reconstruction is now performed by forward passing the signal (or signals) through a (regression) network, which is able to predict the T_1 and T_2 relaxation times for the input. Proposed approaches vary from Fully Connected Neural Networks (FCNs) [1], Convolutional Neural Networks (CNNs) [2,5,6] and other architectures, e.g. incorporating an U-Net [3]. However, also state-of-the-art DL solutions have their drawbacks: While FCNs are known to tend to overfit because of the huge number of parameters, CNNs are not optimally suited for time-resolved tasks. To overcome these limitations, we propose Recurrent Neural Networks (RNNs) for this reconstruction task due to their capabilities to capture the time dependency in the signal better than e.g. CNNs. We evaluate our approach using in-vivo data from multiple brain slices and several volunteers and investigate with an extensive evaluation following aspects: (1) the superior performance of RNNs over CNNs, (2) complex-valued input signal data instead of magnitude data as in some previous approaches (e.g. [1,5]) and (3) spatially connected signal patches instead of one signal for the input layer in combination with a novel quantile filtering layer prior to the output layer. We expect small, spatially connected patches to have the same type of tissue and therefore the same quantitative parameters. The knowledge of spatial neighbors was shown to help the reconstruction accuracy by e.g. [3], but they used the whole image as input. To be able to train their network, all signals have to be compressed and possibly important information may be lost in the signals. Our approach uses smaller, not compressed patches of spatially connected signals (cf. Fig. 1). To the best of our knowledge, RNNs for MRF were only investigated using signals from a synthetic dataset and without the consideration of spatial neighbor signals [10].
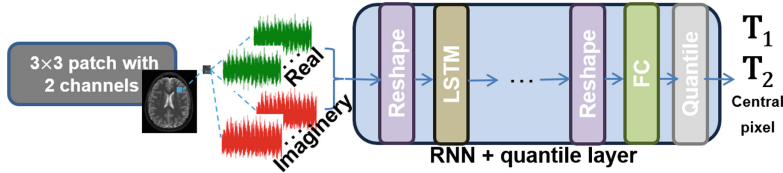
Fig. 1. Overview over the MRF reconstruction process using deep learning. We map the reconstruction process using a Recurrent Neural Network with complex-valued input signals in combination with a quantile layer. LSTM: Long Short-Term Memory layer, FC: Fully Connected layer.

2   Methods

2.1   Recurrent Neural Networks

General Architectures: We devise a regression RNN to solve the MRF reconstruction task: From the input (one or more time signals), the network predicts the quantitative relaxation parameters for this signal. For the development of the networks, we use the well-known Long Short-Term-Memory (LSTM) layers [4]. In order to keep the sequence in a moderate size, we reshaped the signals of length $n = 3,000$ data points into 30 even sized parts. Thus, every sequence element consists of 100 complex-valued (flatterned to 200 values from the real and imaginary parts, respectively) or magnitude data points and is used in front of the LSTM layer as the first layer of our RNNs. This reshaping reduces the risk of vanishing or exploding gradient problems during the training [11]. One LSTM layer is followed by the Rectified Linear Unit (ReLU) activation and a batch normalization (BN). Afterwards, we use 4 fully connected layers, each followed by a ReLU activation and a BN layer (each operating on either the magnitude or on the real and imaginery data points separately), to execute the regression.

Quantile Layer: To cope with signal outliers due to undersampling or noise during the acquisition, we propose to combine the RNN architecture with a quantile layer as the last layer prior to the output. Inspired by work from Schirrmacher et al. in [12], we use small 3×3 patches of signals, which are locally connected for the input layer. Thus, the input for one regression is increased by a factor of 9 compared to networks with one signal as input. For the output, we compute the 0.5 quantile of all predictions from this neighborhood. The quantile operation $q()$ can be reformulated as $q(f) = \boldsymbol{Q}f$, where $\boldsymbol{Q}$ denotes a sparse matrix which stores the position of the quantile. In the backward pass, the gradient w.r.t. the input is simply the transposed matrix $\boldsymbol{Q}^T$. We expect the signals from small patches to belong to similar or same parameters as they originate from same or similiar tissue type. The quantile layer enables a pooling operation that is more robust to noise compared to common pooling operations such as maximum or average pooling. To the best of our knowledge, we are the first to incorporate this operation as a network layer.

2.2 Training and Evaluation

All our models are trained based on the mean squared error (MSE) loss and optimized using ADAM. We evaluate all models by measuring the difference between the predicted and the ground-truth T_1 and T_2 relaxation times, computed as the relative mean error and the appropriate standard deviation. Data is split into disjunct training, validation and test sets. The validation set is used to select the best model from all training epochs, the test set for testing a model on unknown data afterwards.

3 Experiments and Results

3.1 Data Sets

Data Acquisition: All data sets for our experiments were measured as axial brain slices in 8 volunteers (4 male, 4 female, 43 ± 15 years) on a MAGNETOM Skyra 3T MR scanner (Siemens Healthcare, Erlangen, Germany) using a prototype sequence based on Fast Imaging with Steady State Precession with spiral readouts [7] and following sequence parameters: Field-of-View: 300 mm, resolution: $1.17 \times 1.17 \times 5.0 \, \text{mm}^3$, variable TR (12–15 ms), FA (5–74°), number of repetitions: 3,000, undersampling factor: 48. From 2 volunteers, 2 different slices were available, from 6 volunteers, 4 slices were available each. All slices were measured at different positions and points in time to reduce possible correlations between slices from one volunteer.

Ground-Truth Data: In order to create accurate ground-truth data for our DL experiments, we used a fine resolved dictionary containing overall 691,497 possible parameter combinations with T_1 in the range of 10 to 4,500 ms and T_2 of 2 to 3,000 ms, respectively. To be able to reconstruct the relaxation maps in a reasonable time and to reduce the memory requirements, the dictionary and measured signals were compressed to 50 main components in the time domain using SVD prior to the template matching.

3.2 Experiments for Finding Architectural Settings

Experimental Setup: We ran three specific types of experiments to investigate following issues:

1. Performance of networks using magnitude input signals $S_m \in \mathbb{R}$ vs. complex-valued input signals $S_c \in \mathbb{C}$. For this, we compared the CNN (architectural details see Sect. 3.3) and RNN models with 1×1 S_m and S_c.
2. Performance of networks using CNN vs. RNN models (both with a comparable number of learnable parameters). For this, we compared the CNN and RNN models with 1×1 input signals S_c.
3. Performance of networks using 1×1 input signals S_c vs. 3×3 input signals S_c in combination with a 0.5 quantile layer prior to the output. For this, we compared RNN models with and without a quantile layer.

Data Splitting: As only a limited amount of data sets (overall 12 slices from 4 volunteers) was available for our extensive experiments, we first used all slices from these 4 volunteers randomly separated into training, validation and test sets (8, 2 and 2 slices, respectively). We then used additional 16 slices from another 4 volunteers (again randomly separated) for experiments with our best fitted model (19 slices for training, 7 for validation, 2 for testing).

3.3   Comparison with Other DL Architecture

We used the CNN model with overall 4 convolutional and 4 fully connected layers with ReLU activations and average pooling in [5] to compare our approach with another DL based MRF reconstruction framework. We extended this baseline model with BN layers after each convolutional and fully connected layer.

3.4   Results

Results can be found in Table 1 (validation loss from the best epoch) and in Fig. 2 (parameter maps on the same test set from all models).

Table 1. Validation losses across different experiments. Best results are marked in bold. The validation loss is measured as $\sqrt{\text{MSE}}$ over T_1 and T_2 values. CNN_1: CNN model with 1×1 input signals, RNN_1: RNN model with 1×1 input signals, RNN_3: RNN model with 3×3 signal patch as input and quantile layer, RNN_3^*: the same as RNN_3, trained with the larger data set. Detailed information about the models see Sects. 2.1 and 3.3.

	Validation loss [ms]			
Input signals	Model			
	CNN_1	RNN_1	RNN_3	RNN_3^*
$S_m \in \mathbb{R}$	636.96	424.96	–	–
$S_c \in \mathbb{C}$	470.26	269.20	**221.52**	**195.34**

4   Discussion

In summary, the main observation from our results is the clear improvement of the performance using our proposed RNN model in combination with complex-valued input signals and the quantile layer in comparison to all other tested models.

Magnitude vs. Complex-Valued Signal Inputs: We first compare our models trained with S_m and S_c inputs. The utilization of both components of the complex-valued signals, instead of only computing the magnitudes for the input layers of the networks, is an essential factor for the performance. A clear reduction of the errors is achieved using S_c for both approaches (CNN: more than
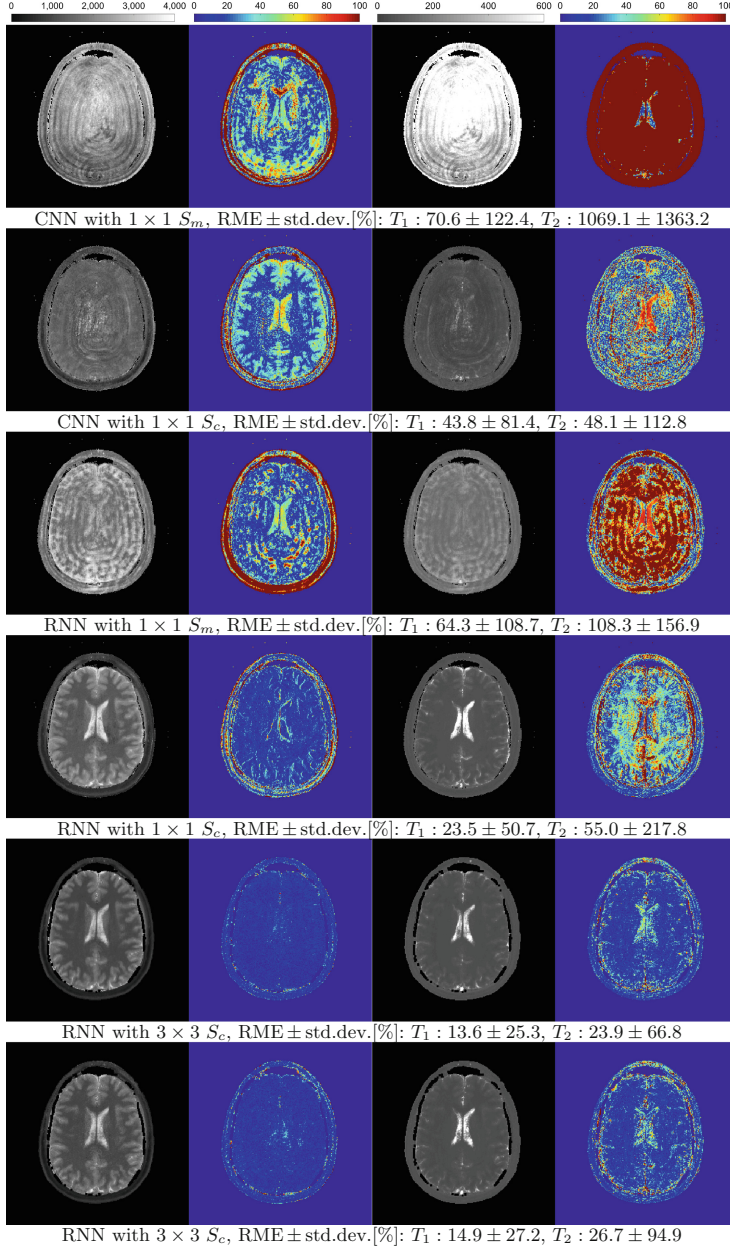
Fig. 2. Predicted maps of one test data set from models using small data set (rows 1–5), or large data set (row 6). First column: T_1 maps. Second column: T_1 relative mean errors to the ground-truth. Third column: T_2 maps. Fourth column: T_2 relative mean errors to the ground-truth. For better visibility, all relative error maps were clipped at 100%, the background of all T_1 and T_2 maps was set to -200 and they were windowed equally for fair comparison (0–4,000 ms and 0–600 ms, respectively). RME: Relative mean error, std.dev.: standard deviation.

62%, RNN: more than 50%). Comparing the visual results of e.g. the same RNN model using S_m and S_c (cf. rows 3, 4 in Fig. 2), the complex version clearly yields reduced relative mean errors and improved parameter maps without being corrupted by the heavy ringing artifacts which appear with the S_m inputs.

CNN vs. RNN: A clear improvement is also achieved using a RNN instead of a CNN model with a reduction of the errors up to 53%. Independent of the input signal types, the CNN model is not able to reconstruct meaningful parameter maps showing soft tissue contrast. In comparison, the RNN model is capable of reconstructing high detail parameter maps, showing the better capability of the RNN for processing time-dependent signals. Nevertheless, this holds only for the RNN using S_c, since the RNN using S_m is still corrupted by the ringing artifacts.

Quantile Layer: Our results show additionally, that a quantile layer furthermore improves the performance (cf. rows 4, 5 in Fig. 2), reducing the errors by 57% and 43% for T_1 and T_2, respectively, in comparison to a RNN without quantile layer. The influence of the quantile layer is particularly evident at transitions between different tissue types in the parameter map. With the help of the quantile layer, the errors at the edges can be enormously reduced, as the 0.5 quantile layer acts as an edge-preserving denoising filter (cf. the relative error maps in rows 4, 5 in Fig. 2).

Challenges and Limitations: Our experiments show the improved performance step-by-step, that increases from (1) magnitude to complex-valued input signals, (2) from a CNN to a RNN model and (3) from a RNN without a quantile layer to a RNN with a quantile layer. Even though we use a limited amount of data, our results are a strong indication, that our model is able to generalize. Using our best RNN model and training it with slightly more data already decreased the error (cf. Table 1), which encourages this assumption. One further step, however, is the evaluation of our proposed approach using data splits with completely unseen volunteer data sets in the validation or test data when more data is available (preliminary experiments in this direction are attached in the Supplementary Material). Moreover, we used a very fine-resolved dictionary for the ground-truth data. While this is crucial for accurate ground-truth data, this further increases the amount of training data that is necessary to fully imprint the complex mapping into the network. In comparison to other MRF DL approaches (e.g. the MRF-EPI sequence in [1]), we used signals from a very strongly undersampled acquisition (undersampling factor: 48), which leads to very noisy and corrupted signals compared to simulated dictionary signals. As shown by Hoppe et al. in [5,6], fully sampled dictionary signals can be easily learned by simple CNN models. However, undersampled in-vivo data are more challenging to reconstruct with the MRF DL method, thus a more complex model is required.

5   Conclusion

We proposed a regression RNN for MRF reconstruction. Our architecture combines a model used to deal with time-dependent complex-valued input signals incorporated as a LSTM layer with a novel quantile layer to deal with signal outliers, which are very common due to the strong undersampling during the acquisition. We evaluated our approach in a proof-of-concept study with various experiments and showed, that our model outperforms other DL models like CNNs or RNNs without the additional quantile layer, reducing the errors by more than 80%. One limitation of our study is the restricted amount of training data, which will be addressed in future work. Furthermore, another future step will be a deeper comparison of the different architectures and their features which can help to improve the interpretability of the networks. In addition, the incorporation of known operations based on the imaging physics within the networks as described in [9] can help to reduce the complexity and improve the performance at the same time. This also will be investigated for our application.

References

1. Cohen, O., Zhu, B., Rosen, M.S.: Mr fingerprinting deep reconstruction network (drone). Magn. Reson. Med. **80**(3), 885–894 (2018)
2. Fang, Z., Chen, Y., Lin, W., Shen, D.: Quantification of relaxation times in MR fingerprinting using deep learning. In: Proceedings of the International Society for Magnetic Resonance in Medicine. Scientific Meeting and Exhibition, vol. 25. NIH Public Access (2017)
3. Fang, Z., Chen, Y., Liu, M., Zhan, Y., Lin, W., Shen, D.: Deep learning for fast and spatially-constrained tissue quantification from highly-undersampled data in magnetic resonance fingerprinting (MRF). In: Shi, Y., Suk, H.-I., Liu, M. (eds.) MLMI 2018. LNCS, vol. 11046, pp. 398–405. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00919-9_46
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
5. Hoppe, E., et al.: Deep learning for magnetic resonance fingerprinting: accelerating the reconstruction of quantitative relaxation maps. In: Proceedings of the Joint Annual Meeting ISMRM-ESMRMB (26th Annual Meeting and Exhibition), Paris, France, p. 2791 (2018)
6. Hoppe, E., et al.: Deep learning for magnetic resonance fingerprinting: a new approach for predicting quantitative parameter values from time series. Studies in Health Technology and Informatics, vol. 243, pp. 202–206 (2017)
7. Jiang, Y., Ma, D., Seiberlich, N., Gulani, V., Griswold, M.A.: MR fingerprinting using fast imaging with steady state precession (FISP) with spiral readout. Magn. Reson. Med. **74**(6), 1621–1631 (2015)
8. Ma, D., et al.: Magnetic resonance fingerprinting. Nature **495**(7440), 187–192 (2013)
9. Maier, A.K., et al.: Learning with known operators reduces maximum training error bounds. Nat. Mach. Intell. **1**, 373–380 (2019)
10. Oksuz, I., et al.: Magnetic resonance fingerprinting using recurrent neural networks. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), pp. 1537–1540. IEEE (2019)

11. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International conference on machine learning, pp. 1310–1318 (2013)
12. Schirrmacher, F., et al.: Temporal and volumetric denoising via quantile sparse image prior. Med. Image Anal. **48**, 131–146 (2018)
13. Wang, Z., Zhang, Q., Yuan, J., Wang, X.: MRF denoising with compressed sensing and adaptive filtering. In: 2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI 2014), pp. 870–873. IEEE (2014)