pISSN 2384-1095
eISSN 2384-1109

iMRI 2021;25:10-22

https://doi.org/10.13104/imri.2021.25.1.10

iMRI
Investigative Magnetic Resonance Imaging

Blended-Transfer Learning for Compressed-Sensing Cardiac CINE MRI

Seong Jae Park, Chang-Beom Ahn

Department of Electrical Engineering, Kwangwoon University, Seoul, Korea

Purpose: To overcome the difficulty in building a large data set with a high-quality in medical imaging, a concept of 'blended-transfer learning' (BTL) using a combination of both source data and target data is proposed for the target task.

Materials and Methods: Source and target tasks were defined as training of the source and target networks to reconstruct cardiac CINE images from undersampled data, respectively. In transfer learning (TL), the entire neural network (NN) or some parts of the NN after conducting a source task using an open data set was adopted in the target network as the initial network to improve the learning speed and the performance of the target task. Using BTL, an NN effectively learned the target data while preserving knowledge from the source data to the maximum extent possible. The ratio of the source data to the target data was reduced stepwise from 1 in the initial stage to 0 in the final stage.

Results: NN that performed BTL showed an improved performance compared to those that performed TL or standalone learning (SL). Generalization of NN was also better achieved. The learning curve was evaluated using normalized mean square error (NMSE) of reconstructed images for both target data and source data. BTL reduced the learning time by 1.25 to 100 times and provided better image quality. Its NMSE was 3% to 8% lower than with SL.

Conclusion: The NN that performed the proposed BTL showed the best performance in terms of learning speed and learning curve. It also showed the highest reconstructed-image quality with the lowest NMSE for the test data set. Thus, BTL is an effective way of learning for NNs in the medical-imaging domain where both quality and quantity of data are always limited.

Keywords: Deep neural network; Blended-transfer learning (BTL); Transfer learning (TL); Standalone learning (SL); Compressed sensing; Cardiac CINE MRI

Original Article

Received: July 17, 2020
Revised: September 14, 2020
Accepted: November 2, 2020

Correspondence to:
Chang-Beom Ahn, Ph.D.
Department of Electrical Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 01897, Korea.
Tel. +82-2-940-5148
E-mail: cbahn@kw.ac.kr

INTRODUCTION

Cardiac CINE magnetic resonance imaging (MRI) is a useful imaging method for diagnosing various heart diseases (1, 2). To suppress cardiac and respiratory motions, signals are generally measured in synchronization with ECG signals of breath-holds. Performing compressed sensing (CS) using sampling below the Nyquist rate is useful for measuring multi-slice cardiac CINE images within a few breath-holds (3, 4). Minimization of the L1-norm has been used to reconstruct undersampled data. However, reconstruction of undersampled data requires considerable time by iteratively applying a nonlinear solution (5).

Recently, deep learning has been applied to reduce the reconstruction time or improve image quality (6). Wang et al. (7) have proposed a framework for reconstructing high-quality images using a fully convolutional network (FCN). Although the proposed framework has a flexible structure, as FCN could be applied to a fraction of an image, it requires a post-processing after applying FCN and considerable time for learning as well. Yang et al. (8) have proposed ADMM-net which implements a classical algorithm called the Alternating Direction Method of Multipliers (ADMM) using a deep neural network (NN). Although ADMM-net has successfully reduced the reconstruction time, its performance is limited compared to the original algorithm.

Yang et al. (9) have proposed a De-Aliasing Generative Adversarial Network (GAN) that can remove artifacts due to undersampling. For the generator to generate an image similar to the real one, various terms are added to the loss function and a method for achieving stable learning is presented. Compared with various CS MRI methods, GAN shows improved performance with image-reconstruction time considerably reduced. However, as with other GAN-based studies (10, 11), the risk of NNs creating virtual artifacts is increased, especially at high compression ratios (CRs). In addition, its network performance is not satisfactory when the similarity between clinical data and learning data is low. Recently, some variants of GAN have been proposed to solve these problems. For example, Yu et al. (12) have proposed a conditional GAN. Zhu et al. (13) have reported a lesion focused super-resolution approach. Wang et al. (14) have shown an improved cyclic GAN. Schlemper et al. (15) and Zhu et al. (16) have applied deep networks for CS diffusion tensor cardiac magnetic resonance (DT-CMR). Hyun et al. (17) have applied deep learning for undersampled MRI reconstruction of head images using U-net (18). They combined deep learning and k-space correction with measured data and achieved a remarkable performance, far exceeding existing CS algorithms. Kofler et al. (19) have also shown that a modified U-net outperforms the existing iterative reconstruction method in image quality and computation time. The U-net is widely used in many applications because of its excellent performance. In this study, we also used U-net to reconstruct dynamic CS cardiac MRI, in contrast to the reconstruction of static head images in previous studies. The main contribution of this study, however, is the improved learning method of NNs. This will be discussed later.

Due to difficulty in building a large data set with high-quality in medical imaging, transfer learning (TL) has been tried (20, 21). In TL, two domains or tasks are defined, namely the source task and the target task. The source task typically performs pre-learning for NNs using a large amount of open data (source data) and the target task refines pre-learned NNs for target purposes. For example, Oquab et al. (22) have proposed a framework for TL between two data sets with different categories of labeling. Ciresan et al. (23) have modified the number of nodes in the fully connected layer to reduce the cost of learning classifier and improve performance. Meng et al. (24) have used TL to diagnose liver fibrosis using ImageNet as source learning. Because many TL-based studies have different applications and purposes, the target task should determine the manner to use some or entire NNs in the source task. For MRI, Chen et al. (25) have used TL for automatic segmentation of left ventricle myocardium in porcine cardiac cine MRI. Dar et al. (26) have used TL for CS brain study.

In this study, the open data set released by York University (27) was used for the source task. This open data set is called 'Y-data set.' Subsequently, the NN conducts fine-tuning using a data set measured for the target task. The measured data set is called 'K-data set.' Both Y-data and K-data sets are k-space data for cardiac CINE images with short-axis view. The Y-data set comprises clinical (patient) data obtained from 1.5T MRI. The K-data set comprises volunteer images obtained from 3.0T MRI. Both the data sets have different scan parameters.

NNs that conducted the source task showed improved learning speed, performance, and generalization for the target task. However, as the target task progressed, knowledge from the source task quickly decayed as filter kernels or weights of the NN changed. Therefore, knowledge of source task was not maintained correctly using TL and knowledge transfer did not work properly. The motivation for this study was to find a better learning method for compressed sensing (CS) cardiac CINE MRI. Scanning multi-slice CINE data in a few breath-holds and real-time reconstruction with diagnostic quality are critical for clinical applications. Machine learning seems to be the first choice for the task.

We proposed a 'blended-transfer learning' (BTL) to mitigate data problems by effectively utilizing open data sets for both pre-training and target task. We quantified both source knowledge and the newly acquired target knowledge in terms of normalized mean square error (NMSE). The learning method can optimize the network for both source and target tasks. To the best of our knowledge, no similar approach has been attempted for CS cardiac CINE

MRI. By using a combination of both source data and target data for the target task, network learns the target task well while preserving the source knowledge to the maximum extent possible. BTL contributes to the generalization of the NN by accumulating knowledge from both source and target tasks. It can reduce the conflict of knowledge in different domains.

MATERIALS AND METHODS

Data

Two types of data sets, 'K-data set' and 'Y-data set', were used. They comprised k-space data of cardiac CINE images with short-axis view. Multi-slice cardiac CINE MRI scans three-dimensional (3-D) heart as a function of time, which may be presented using four variables, kx, ky, z, t with sizes of Nx, Ny, Nz, and Nt, respectively. Terms Nx and Ny denote sizes of the cross-sectional image along the x-axis and y-axis. Nz is the number of slices and Nt is the number of frames for a cycle of cardiac motion. Although cardiac CINE MRI data are 4-D data, the input to the NN is generally 2-D sectional images to reduce complexity.

The k-space data measured for all phase-encoding gradient values are defined as 'full data' which generate 'ground truth image' via inverse 2-D Fourier transform. These full data were undersampled via computer simulation along the phase-encoding axis to generate undersampled data. Sampling locations or mask were generated in the phase encoding direction as a function of cardiac phase to accommodate CINE imaging. Since the sampling period along the frequency encoding direction was relatively short, undersampling was not necessary. Sampling followed a modified Gaussian density function centered at the dc of the phase-encoding axis in the spatial frequency domain (3). Considering various CS applications of cardiac CINE MRI, the CR was chosen as 2, 3, 4, and 8.

The K-data set comprised full k-space data corresponding to a total of 2016 full 2-D sectional images obtained with a 3.0T MRI system (Siemens Healthcare, Erlangen, Germany) using a balanced-steady-state free precession (SSFP) sequence from eight healthy volunteers. A 5-channel cardiac array coil was used and the sensitivity map was acquired by a low-resolution scan separately from the main scan. The sensitivity map was used to reconstruct a combined image from channel images. The combined image was assumed to be ground truth image. For simplicity, k-space data were generated by 2-D Fourier transform of the combined image. Training data set was organized from Volunteers 1-4 (total 1104 images) and the test data set was organized from Volunteers 5-8 (total 912 images) to ensure that both data sets were disjoint from separate subjects. All studies were approved by the Institutional Review Board. Written informed consents were obtained from all volunteers participated in this study.

For the training data set, full k-space data of Volunteers 1-4 were undersampled using CRs of 4 and 8. They were subsequently augmented thrice to obtain 6624 images. Data augmentation was performed using vertical flip, horizontal flip, and 90° rotation with random probabilities. For the test data set, full k-space data of Volunteers 5-8 were undersampled using CRs of 2, 3, 4, and 8. Consequently, the test data set comprised 912 images for each CR to assess the performance of the NN in accordance with the CR.

To supplement the K-data set of fewer applicants, an open data set disclosed by York University was used for the source task (27). This York University data set comprised 33 subjects with 7980 sectional images, among which data from one outrageous subject were excluded. Sectional images out of the heart region were also excluded. Consequently, 5220 clinical sectional images from 32 subjects were selected as ground truth images and full k-space data was generated via the 2-D Fourier transform of ground truth images. The training data set was organized from Subjects 1-22 (total 3440) and the test data set was organized from Subjects 23-32 (total 1780). For the training data set, full k-space data from 22 subjects were undersampled using CRs of 4 and 8 without augmentation to obtain 6880 images. Undersampling was performed similarly as that for the K-data set. Images were arbitrarily readjusted to either 'as is,' 'vertical flip,' 'horizontal flip,' or '90° rotations.' For the test data set, full k-space data from 10 subjects were undersampled using CRs of 2, 3, 4, and 8. Consequently, the test data set comprised 1780 images for each CR.

Three sample ground truth images of K-data (left) and Y-data (right) sets are depicted in Figure 1. Image and scan parameters for K-data and Y-data sets are summarized in Table 1.

Reconstruction of CS CINE MRI Using Deep NN

Figure 2 depicts learning procedures of the NN (upper) to reconstruct CS CINE MRI. The process began with interpolation of missing data due to undersampling using data in adjacent frames. The last frame was assumed to be adjacent to the first one because of cyclic motion of the

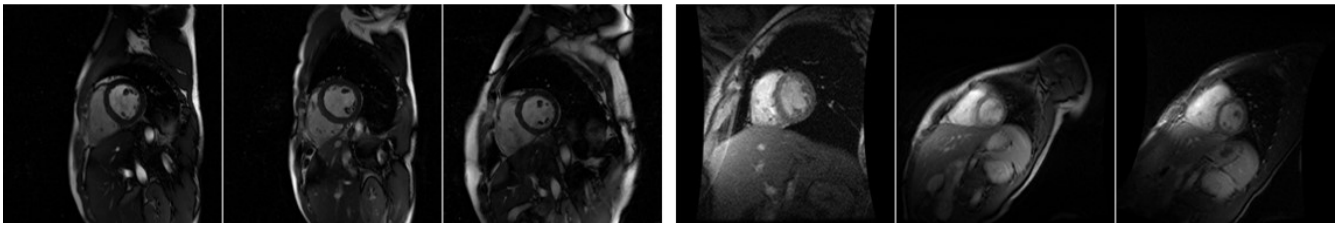

Fig. 1. Three sample images for the K-data set (left) and Y-data set (right). Each image is from a different subject. The K-data set is used for the target task and the Y-data set is used for the source task.

Table 1. Scan and Image Parameters for K-data and Y-data Sets

	K-data set	Y-data set
Main field strength	3.0T	1.5T
Sequence	Balanced SSFP	Balanced SSFP
TR/TE	3.88/1.94 ms	N/A
Transverse-matrix size	256 × 256	256 × 256
Transverse resolution	1.37 mm	0.93-1.64 mm
Field-of-view (FOV)	350 × 350 mm^2	240-420 × 240-420 mm^2
Number of slices	12	5-11
Slice thickness	8 mm	6-13 mm
Number of frames	16-24	20
Temporal resolution	31.04 ms	N/A
Application	Target learning (fine tuning)	Source learning

SSFP = steady-state free precession; TE = echo time; TR = repetition time

heart. In addition, if no data was measured in all frames, missing data were assumed to be zero. If all missing data were interpolated, an inverse 2-D Fourier transform was applied to perform initial image reconstruction. Images achieved via the initial reconstruction had aliasing and relatively poor spatial and temporal resolutions compared with those of ground truth images. The deep NN aims to reduce aliasing and improve the quality of the initial reconstructed images. The image was subtracted from the ground truth image to make a 'difference image.' The reconstructed image was normalized using Eq. [1] and used as input to the NN. The difference image was normalized and shifted using Eq. [2]. It was used as the label to the NN. The input and the label pair is used as a unit element for the training data set.

$$i\,(x, y) = \frac{I\,(x, y)}{MAX} \qquad [1]$$

$$d\,(x, y) = \frac{D\,(x, y)}{MAX} + 0.5 \qquad [2]$$

In Eq. [1], $I(x,y)$ denotes the initial reconstructed image and MAX denotes a value given for each participant. MAX could be either the maximum value or any representative value of the entire multi-slice, and multi-frame images included in each examination of a participant. The normalized image $i(x,y)$ was truncated to 1 or 0, respectively. If $i(x,y)$ is greater than 1 or below 0; $i(x,y)$ will lie between 0 and 1. In Eq. [2], $D(x,y)$ denotes the difference image and MAX denotes the value defined in Eq. [1]. Although the difference image $D(x,y)$ would be approximately distributed from $-MAX$ to MAX, most values were near zero. The normalized and shifted difference image $d(x,y)$ would be close to 0.5 for most values. If $d(x,y)$ is greater than 1 or less than 0, it will be set to be 1 or 0, respectively.

The NN adjusts filter kernels and weights in such a manner that the output for a given input matches the label. Initial values of both filter kernels and weights affect the performance and learning speed. The NN was implemented using Keras based on TensorFlow. The learning for NN takes considerable time. However, once the learning is completed, reconstruction using NN is significantly faster than that using traditional CS algorithms.

From Figure 2, it is evident that for reconstruction (lower part in Fig. 2), the input has similar process as that for learning (upper part in Fig. 2). The NN estimates the normalized difference image as output. The following difference image is obtained via shift and denormalization:

$$O\,(x, y) = MAX \cdot [o\,(x, y) - 0.5] \qquad [3]$$

where MAX is the value defined in Eq. [1], $o(x,y)$ is the output of the NN, and $O(x,y)$ is the estimated difference image. Refined reconstruction is performed by adding the estimated difference image to the initial reconstructed image. The refined reconstructed image has less aliasing

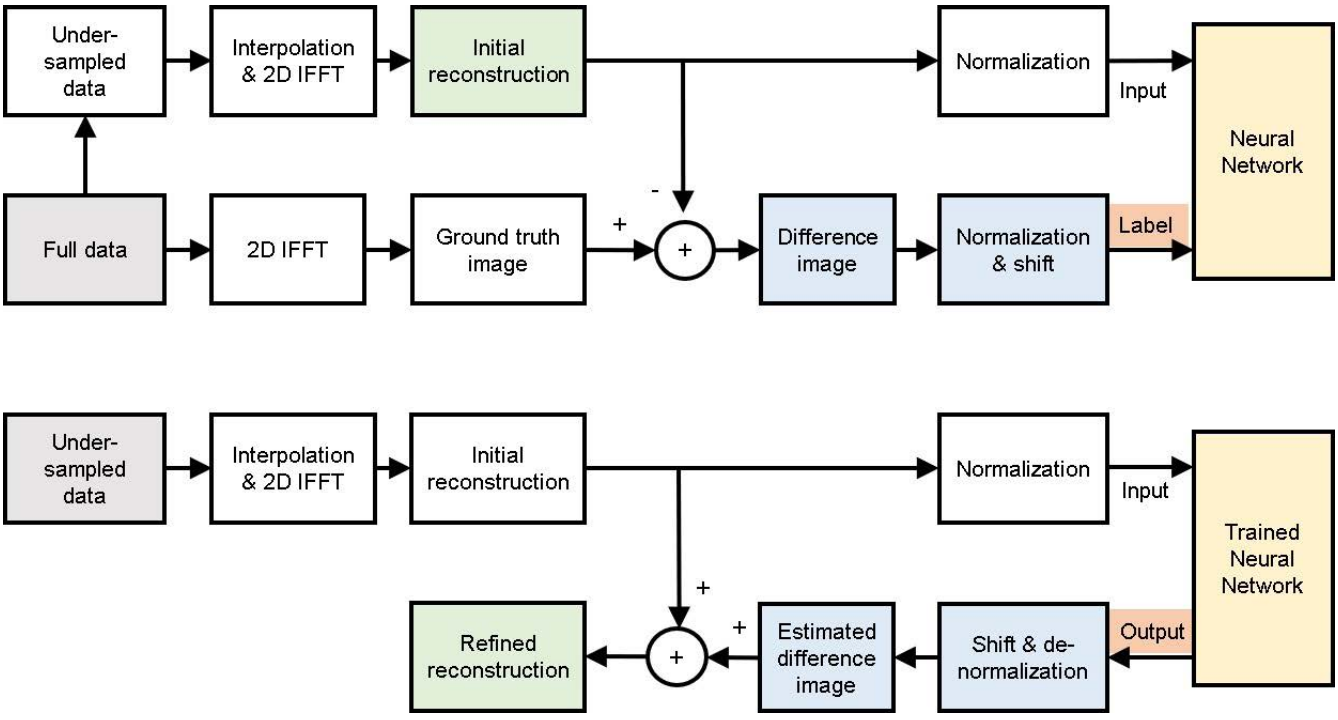

Fig. 2. Learning (upper) and reconstruction (lower) of the neural network.

and better quality than the initial reconstructed image.

Architecture of the Deep NN

The designed deep NN comprises multi-layer, multi-stage convolutional NNs (CNNs) (18). It contains encoding and decoding processes connected in a hierarchical structure. It has an overall shape of 'U,' as depicted in Figure 3. In the encoding process, multi-stage convolutions (3 × 3) and max-pooling (2 × 2) were applied for each layer. Convolutions were used to extract features from the input. Features extracted were then stored in the feature map. Kernels were adjusted such that the output of the NN matched the label. Max-pooling was performed to reduce the input size. Reduced input was then convolved with a kernel of the same size. The aforementioned step was equivalent to performing a convolution with a kernel twice larger than that used in the previous layer. Therefore, the max-pooling enables extraction of features of various sizes. Notably, the output of each layer in the encoding process was the input of the next layer. It was also concatenated to the input in the decoding process (skip connection) of the same level of the layer. As the number of layers increased, the size of the input decreased, while the number of channels increased. In the last layer of the encoding process, the input size was 16 × 16 and the channel size was 1024.

In the decoding process, transpose-convolution (2 × 2) and multi-stage convolutions (3 × 3) were repeatedly performed to increase the size of the feature map and decrease the channel size. In the final layer of decoding, the output size was 256 × 256 and the channel size was 1 (same size as that of the input image). A Rectified Linear Unit (ReLU) was used as an activation function for all convolutions of size (3 × 3) in both encoding and decoding layers. The convolution at the final stage of the output layer used a kernel of size (1 × 1) and the sigmoid function as the activation function to make the output range from 0 to 1.

Learning Methods

Filter kernels of the NN were adjusted using the training data set. Initial values of filter kernels considerably affected both learning speed and performance. The initialization was considered in terms of the following learning methods.

(1) Standalone Learning (SL)

In SL, filter kernels are randomly initialized. In this study, we used the method proposed previously (28). Because no prior knowledge was included, the method was named SL.

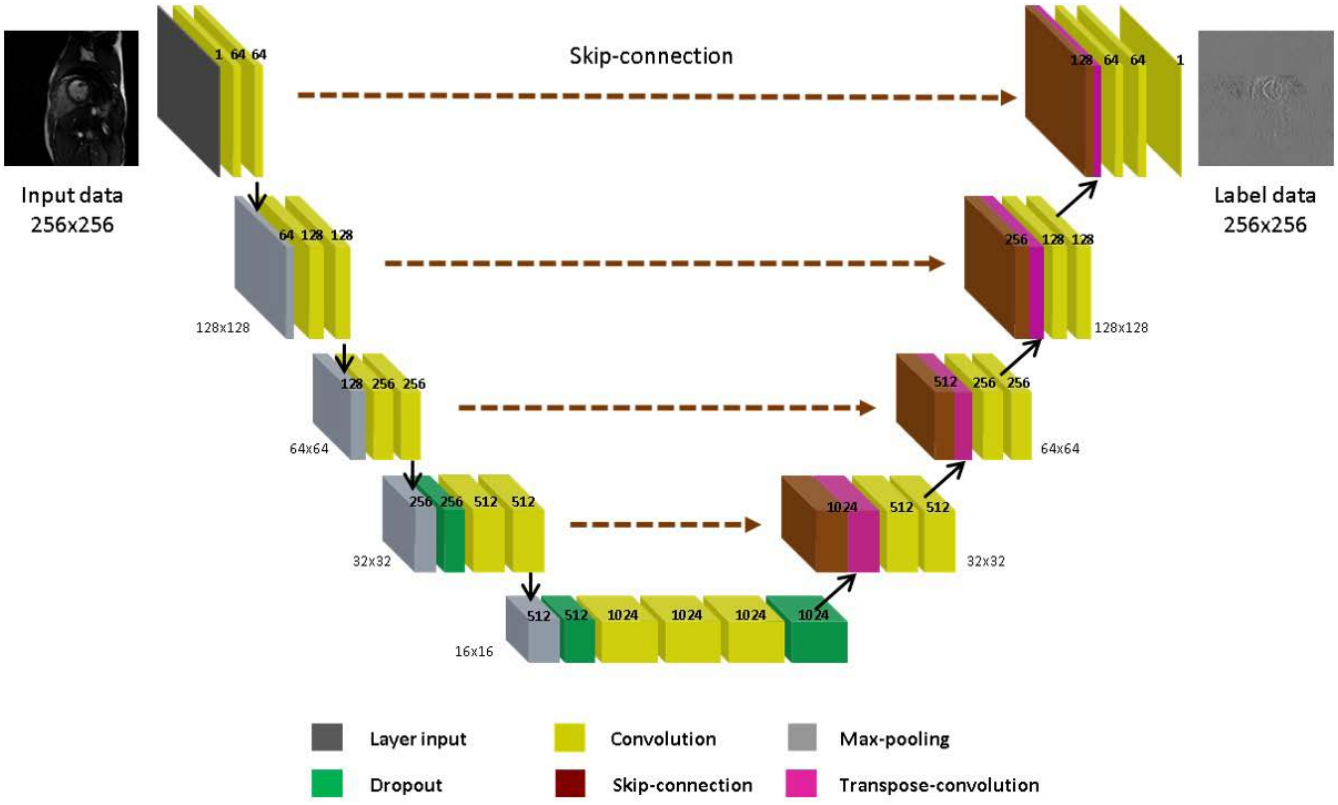

Fig. 3. Architecture of the neural network to reconstruct compressed-sensing cardiovascular CINE MRI. The network is shaped like a 'U.' Main components of the neural network are CNNs. Max-pooling and transpose convolution are applied for downsampling and upsampling of the input, respectively.

(2) Transfer Learning (TL)

Sometimes, it is desirable to transfer some or the entire NN (source network) that has undergone pre-learning using similar kinds of data (source data) to another network (target network) to improve the learning speed and network performance. Hence, it is called 'TL.' In this study, filter kernels of the NN that had performed the source task were used as initial values of the NN for the target task.

(3) Blended-Transfer Learning (BTL)

Filter kernels obtained from the source task can considerably improve the learning speed and network performance for the target task. However, initial values from the source task significantly changed in a few epochs of the target task. Therefore, we proposed 'BTL' by appropriately blending both source (Y-data set) and target (K-data set) data sets for the target task. Using a mixed data set, the target network could learn the target data while preserving the knowledge from the source task to the maximum extent possible.

RESULTS

NNs that can perform learning using various methods are investigated. The SL-based NN performed the target task using the K-data set via random initialization of filter kernels. Both TL- and BTL-based NNs performed pre-learning first using the Y-data set. For pre-learning, filter kernels were randomly initialized as in the case of SL. These filter kernels obtained using the source task were set to the target NN as initial values. The NN further conducted learning using the K-data set in the case of TL. The BTL-based NN performed learning using both the K-data and Y-data sets. The learning rate was optimally chosen for each data set as follows: 1.0×10^{-5} for the K-data set and 8.0×10^{-5} for the Y-data set. For the combination of K-data and Y-data sets, hyper parameters for the target data (K-data) set were chosen. The batch size was 32 and the binary cross-entropy was selected as the loss function.

In Figure 4, learning curves of NNs after performing SL and TL are depicted. The horizontal axis represents the number of epochs and the vertical axis represents NMSE.

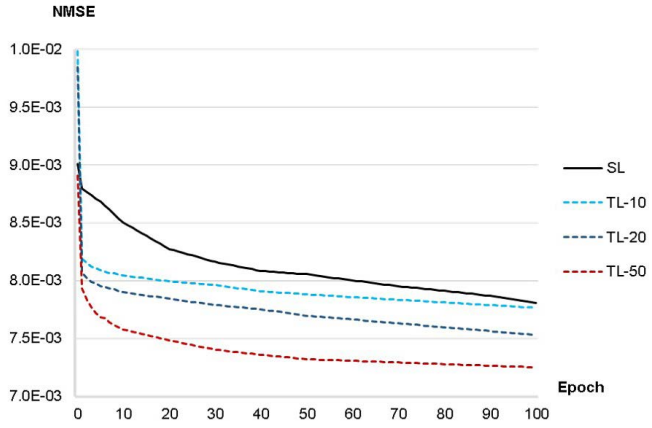

Fig. 4. Learning curves of neural networks that have conducted SL and TL. The number after the hyphen is the number of epochs for the source task. The horizontal axis represents the number of epochs for the target task and the vertical axis represents the NMSE for the target data.

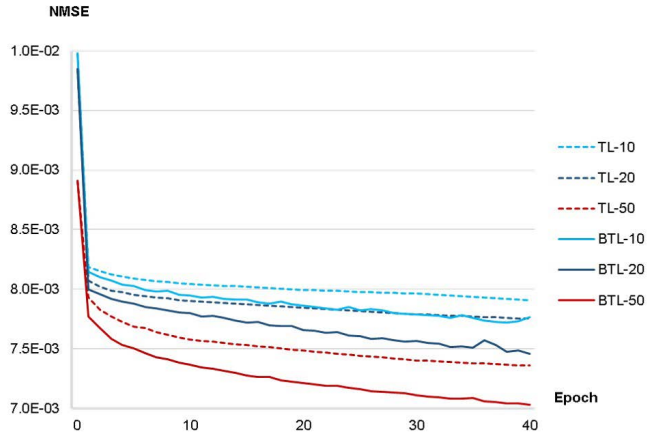

Fig. 5. Learning curves of neural networks that have conducted TL and BTL. The number after hyphen is the number of epochs for the source task. The horizontal axis represents the number of epochs for the target task and the vertical axis represents the NMSE for the target data.

These learning curves were evaluated using the K-data set (training data). Generally, NMSE decreased as the number of epochs increased. Because filter kernels obtained from the source task were transferred to the target NN in the case of TL, the amount of learning in the source task might affect the learning performance in the case of TL. We used the number of epochs in the source task as the amount of learning. The number after the hyphen represents the number of epochs in the source task. As depicted in Figure 4, NNs with large epochs in the source task showed

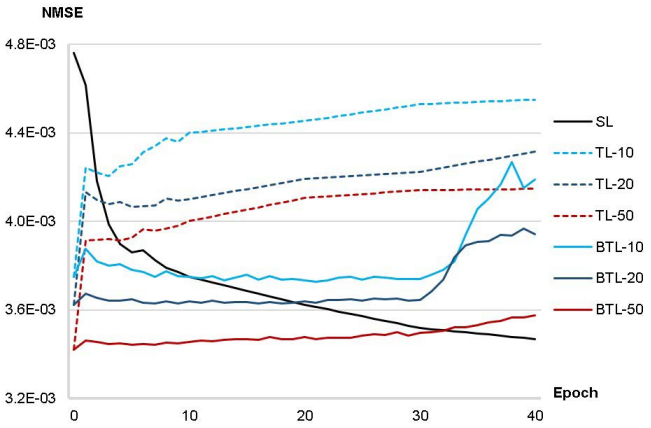

Fig. 6. Loss of source knowledge by the target neural network is evaluated using the NMSE of the source data. NMSE rapidly increases for a few epochs in the case of TL, whereas it remains almost constant in the case of BTL until the last stage wherein the source data is completely excluded for the target task.

satisfactory performance. For example, TL-50 showed a lower NMSE than that TL-10 for a given epoch. Comparing learning curves of TL with those of SL, TL-10, -20, and -50 showed lower NMSEs than SL. Particularly, at early epochs, NMSEs for TL decreased more rapidly than those for SL. For example, the NMSE of TL-20 (26) or TL-50 (3) was lower than that of SL (100). The number in parenthesis represents the number of epochs in the target task. This implies that TL can improve the learning speed and performance of NN in the target task.

BTL comprises 4 stages, each of which consists of 10 epochs. Ratios of the source (Y-data) to target (K-data) data sets are 1, 0.5, 0.25, and 0 as stages progress. In each stage, K-data set was used while Y-data set was randomly sampled according to the ratio. Sampled data were restored for the next stage. For example, the data set in stage 2 comprised 9936 images, of which 6624 were from the K-data set and 3312 were from the Y-data set.

Learning curves for both TL and BTL are depicted in Figure 5. BTL used the same initial filter kernels as those used by TL. BTL with larger pre-learning showed a better performance as observed in TL. In addition, BTL achieved a lower NMSE than TL. For example, TL-10 (80), TL-20 (26), and TL-50 (3) showed similar NMSEs as SL (100), while BTL-10 (28), BTL-20 (9), and BTL-50 (1) achieved similar NMSEs as SL (100).

The loss of knowledge from the source task was evaluated using the NMSE of the source data for the target network as

a function of epoch as depicted in Figure 6. The horizontal axis represents the number of epochs for the target task and the vertical axis represents the NMSE of the source data using the target NN. The learning curve of the source NN is also depicted using a solid black line. Its horizontal and vertical axes represent the number of epochs for the source task and the NMSE of source data by the source NN, respectively. The source learning curve represents the initial NMSE before the target task begins (epoch = 0). For example, the initial NMSEs for TL-10 and BTL-10 were obtained from the source learning curve with an epoch of 10 (= 3.8×10^{-3}).

As the number of epochs increased in the target task, NMSE also rapidly increased in the case of TL as depicted in Figure 6. However, the initial NMSE was almost maintained for BTL. For example, irrespective of the amount of pre-learning for BTL, NMSE remained almost intact up to stage 3 (Y-data/K-data = 0.25). However, for BTL-10 and BTL-20, the NMSE was somewhat increasing in the final stage (stage 4, K-data only). For BTL-50, the NMSE was almost retained. Therefore, BTL with a considerable amount of pre-learning successfully created a network that was optimized for both target and source tasks, thereby contributing to the generalization of NN via knowledge accumulation.

Feature maps of NNs after performing learning using various methods are depicted in Figure 7. The output of the final CNN at each layer is visualized using an input image in the target (K-data) data set. Because the number of channels varied from layer to layer, the feature map was averaged over the entire channel values as shown in Eq. [4]. In addition, because the ReLU was used as the activation function after performing convolution, all feature values

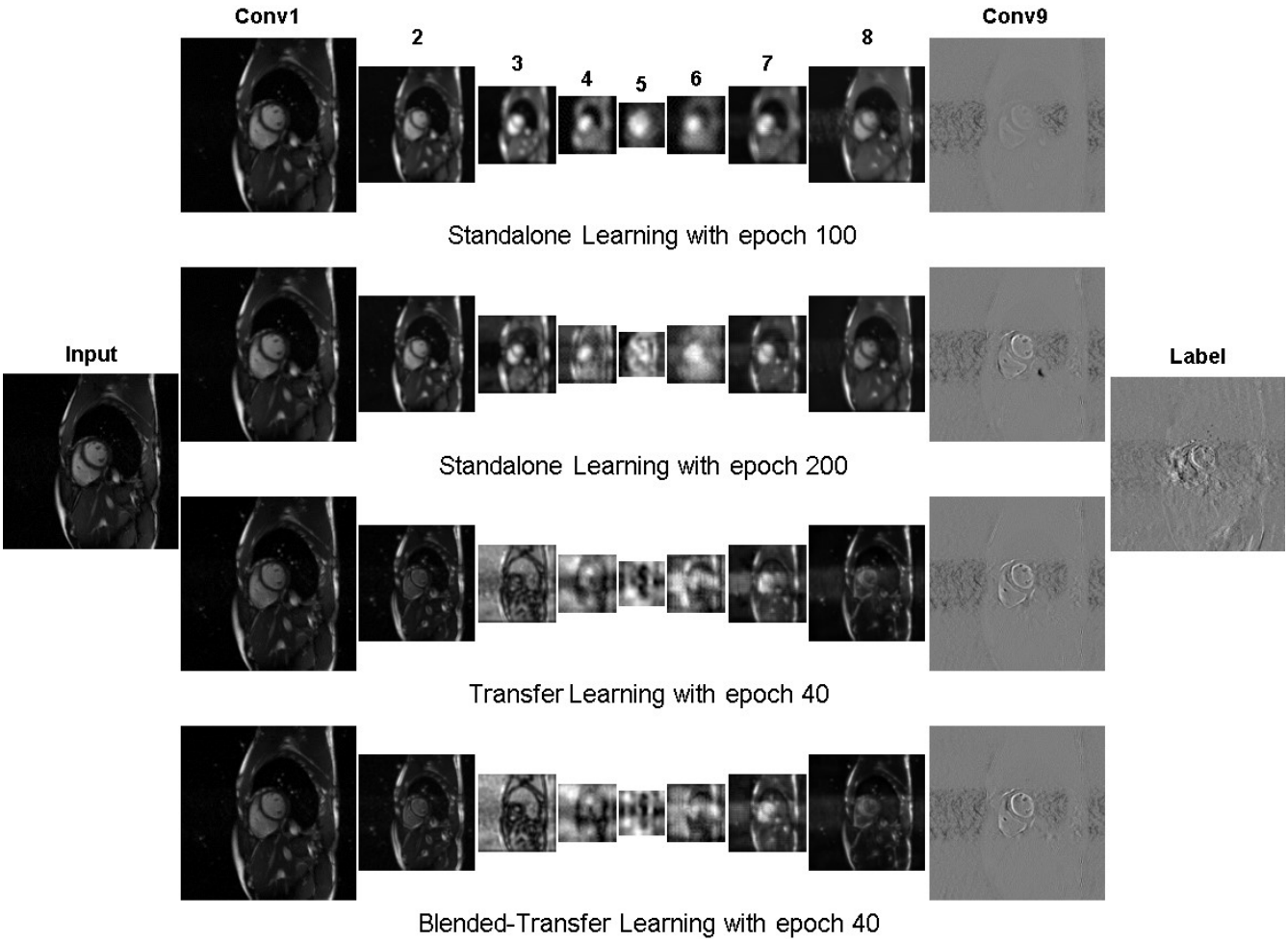

Fig. 7. Visualization of feature maps of neural networks that have performed learnings using various methods. From top to bottom: SL with epoch 100, SL with epoch 200, TL with epoch 40, and BTL with epoch 40. Source tasks were performed with epochs of 50 for both TL and BTL.

were non-negative. The final CNN at the final layer had only one channel. Its values ranged from 0 to 1 by applying the sigmoid function as the activation function.

$$F(x, y) = \frac{1}{N}\sum_{j=1}^{N} f(x, y, c_j) \quad [4]$$

Sizes of feature maps decreased up to the fifth layer and then increased up to the final layer. For comparison, input and label data are depicted on the left and right sides in Figure 7. Feature maps for the three learning methods distinctly differed from one another. These feature maps built using SL(100) mainly showed large structures, whereas those built using SL(200) showed more detailed structures (e.g., see the fifth layer). For TL-50(40) or BTL-50(40), high frequency shapes and detailed structures are satisfactorily presented, even for a small number of epochs (i.e., 40).

In Figure 8, reconstructed images (upper) and line profiles as a function of cardiac phase (lower) using NNs that have

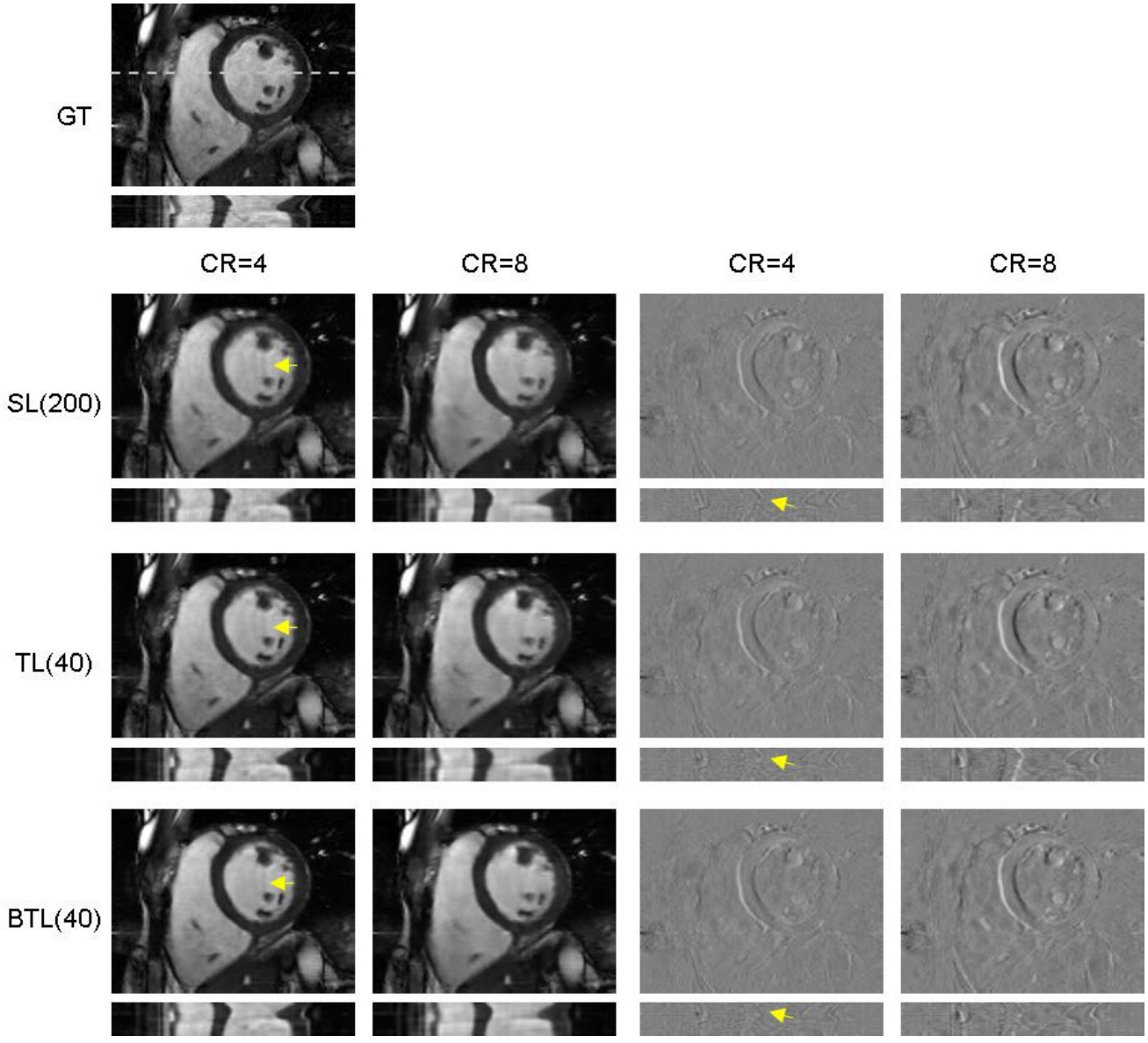

Fig. 8. Reconstructed images (upper) and line profiles as a function of cardiac phase (lower) using neural networks that have performed learning using various methods. The CRs are 4 and 8. From top to bottom: ground truth image, images reconstructed using SL with epoch 200, TL with epoch 40, and BTL with epoch 40. Error images are shown in the right to ensure better visualization.

performed learnings using various methods are depicted. Test data with CRs of 4 and 8 are depicted. From top to bottom, the ground truth image and images reconstructed using SL(200), TL-50(40), and BTL-50(40) are depicted. Line profiles passing through the myocardium that is shown in the ground truth image are vertically stacked according to the cardiac phase. Error images between the reconstructed images and the ground truth image are depicted in right. They were amplified by a factor 4 to ensure better visualization. Improvements achieved using the BTL-based NN were compared with those achieved using the SL-based NN. For example, low frequency oscillation in the ventricle was observed in images reconstructed using SL, which was reduced using BTL (indicated by arrows). An aliasing error existed along the longitudinal direction near the edge in profiles in the case of SL. However, the aliasing error was reduced in the case of BTL (see arrows in the error images of profiles).

The average NMSE for the test data set is summarized in Table 2. It was obtained by taking the arithmetic mean of the NMSE of each applicant. In Table 2, one SL-based NN was added (top) with 40 epochs to match the number of epochs for target task with TL or BTL. From Table 2, it was evident that the NN that performed TL or BTL showed lower NMSE than the NN that performed SL. The NN that performed BTL showed the lowest NMSE for all CRs. Although differences in NMSE according to the learning method were not considerable, i.e., approximately 3% to 8%, they were consistent for all CRs.

Using Keras (version 2.2.4) in Tensorflow (version 13.1.1) on an Intel(R) Xeon(R) CPU e5-2620 v4 @ 2.1GHz and NVIDIA TITAN RTX system, the elapsed times for the learnings of the NNs in Table 2 were measured to be approximately 83 min for SL(40), 409 min for SL(200), 82 min for TL(40), and 119 min for BTL(40). The elapsed time of the source task was 107 min with 50 epochs. For reconstructing the test data set, the elapsed time was approximately 8 s for each CR, equivalent to 8.73 ms per image.

Table 2. Average NMSE of Reconstructed Images by NNs Using Various Learning Methods is Shown in 10^{-3} Unit

Learning method	CR = 2	CR = 3	CR = 4	CR = 8
SL(40)[a]	2.08	3.37	4.16	7.95
SL(200)[b]	2.03	3.26	4.03	7.67
TL-50(40)[c]	2.02	3.21	3.97	7.48
BTL-50(40)[d]	2.01	3.20	3.96	7.35

[a]SL with 40 epochs for the target task.
[b]SL with 200 epochs for the target task.
[c]TL with 50 and 40 epochs for source and target tasks, respectively.
[d]BTL with 50 and 40 epochs for source and target tasks, respectively.
BTL = blended-transfer learning; NMSE = normalized mean square error; NN = neural network; SL = standalone learning; TL = transfer learning

DISCUSSION

In preparation of the data set, assumptions were made to simplify training of the NN. The ground truth image in Figure 2 was assumed to be a magnitude image, so were the initial reconstructed image (input) and the label to the NN. If complex value reconstruction is needed, NNs might be constructed separately for real and imaginary parts of the input and label. For multi-channel data acquired using a cardiac phased array coil, combined image and corresponding k-space data are used for training and test of the NN. Undersampling of k-space data corresponding to the combined image instead of individual channel data reduced training time considerably. In real applications, each channel data should be reconstructed using the trained NN. The coil sensitivity mentioned in the data section is needed to reconstruct a combined image from these channel images.

The designed NN is based on CNN which assumes 2-D input and 2-D output. The filtered image obtained by convolution of the input image with a 2-D kernel of size (3 × 3) reflects characteristics of the input image. For example, a high-pass filter kernel would generate an edge image and a low-pass filter kernel would generate an image with contrast. If many filter kernels are used, we can obtain various image characteristics that can be stored in channels. Feature map is a 3-D matrix in which 2-D filtered images are stacked along the channel. The 3-D feature map is an input to the next convolution with a 3-D filter kernel of size (3 × 3 × number of channels). The 2-D image obtained by the convolution of 3-D feature map with 3-D filter kernel would be stacked to form a new 3-D feature map. These kernels are adjusted to make the output of the NN match the label data.

Because the output of the convolution is a weighted sum of the input within the kernel and because the ReLU truncates output that is less than zero, the output of the CNN depends on the input and its variation restricted to a certain bound. Although the max-pooling used to reduce the spatial resolution of the input may change the shape of the input, the maximum variation is limited to 1 pixel.

Because the transpose convolution that is used to increase the spatial resolution is based on the convolution process, a linear relationship between the input and output is maintained. Furthermore, each kernel is convolved with the entire input (3-D feature map). Therefore, update of the kernel is affected by the entire input rather than local structures of the input.

The function of the CNN is different from that of fully connected network or GAN. A fully connected network can move a node to any location. The number of nodes involved in determining its weight is also relatively small, thereby simplifying the creation of a local structure. The GAN can also add local structures from the discriminator or generator. Due to the nature of medical imaging, creating or erasing local structures unrelated to the input can result in misdiagnosis.

As depicted in Figure 3, a fully connected network and GAN are not involved in the NN. Therefore, the NN may learn from the data in the manner of converting a coarse image to a refined one. However, it cannot extract or store local structures from the data and insert them into an unrelated input image. For example, the source task is performed using clinical data. However, disease-related structures are not stored. Such structures cannot appear during the target task. Therefore, a CNN-based network with BTL or TL can be safely used for clinical purposes.

In clinical applications, building an optimal NN for each CR is time consuming and expensive. In addition, it is difficult to define the exact CR because field-of-view (FOV) and organ (heart) size vary from subject to subject. Therefore, a universal NN was designed with excellent performance using CRs of 2, 3, 4, and 8. To this end, data with CRs of 4 and 8 were mixed and used for training the NN.

Using TL, the NN learns the target task faster by using the source network as initial values of the target network. However, these initial values rapidly changed as the target task progressed. To accommodate the generalization of the NN, knowledge in one domain should not overwrite that in another domain. TL transfers the source knowledge to the target network as initial values, which may be easily changed. Similarly, BTL transfers the source knowledge to the target network as initial values. However, it is involved in subsequent network learning by providing a combination of source data and target data for the target task. Therefore, learning from the target data is performed and source knowledge is simultaneously refreshed using the source data included. Because the target task is the ultimate goal of forming the NN, the amount of source data is reduced as the stage of the target task increases.

Learning curves with BTL showed lower NMSE by 0.1 to 0.2 dB than those with TL as depicted in Figure 5. Interestingly, learning using a combination of target data and source data provided a lower NMSE than that provided upon learning using only the target data, although the evaluation was performed using only target data. The kernels obtained using the gradient descent algorithm with the target data might still need to reach the global optimum. Because many filter kernels are adjustable and because data can contradict each other sometimes, a unique solution might not exist. Therefore, both initial values of the network and the path to the final configuration are important. Using a combination of source data and target data for a target task, the NN may learn a smooth transition from the source knowledge to target knowledge, consequently finding a solution to satisfy both source and target data with minimal adjustments. To avoid excessive data growth, the amount of the source data is set to be equal to the amount of the target data in the initial stage. The amount reduced as the stage increased, finally becoming zero at the final stage (i.e., stage 4). Because learning at each stage is a transition to the final network configuration, learning at each stage is not necessarily high. It was limited to 10 epochs in our application.

The extent of the source knowledge lost as the target task progressed was assessed using the NMSE of the source data with the target network. As depicted in Figure 6, NMSE rapidly increased from the beginning of target task (epochs 1 and 2) with TL, meaning that initial values adopted from the source network had considerably changed at the early stage of the target task. Meanwhile, the NMSE of the source data was almost maintained with BTL while the target task was effectively achieved as depicted in Figure 5. Therefore, BTL formed a network configuration that satisfied the target task without significant loss of the source knowledge.

Notably, the learning speed or performance of the NN varied depending on the learning method, even for the same target data and network structure. Feature maps were also different depending on the learning method (Fig. 7). Because there exist many adjustable filter kernels in deep NNs, different configurations are allowed for similar performance indicators. Accordingly, BTL is a means to construct a better NN configuration.

As depicted in Figure 8, the highest quality was obtained for reconstructed images and profiles when the NN performed BTL. As shown in Table 2, the lowest NMSE of

reconstructed images was obtained in the case of BTL. In this case, visual perception and evaluation based on NMSE agreed well, but not always. Seitzer et al. (29) have proposed a hybrid method in which a visual refinement component is learnt on an MSE based reconstruction network.

Both K-data and Y-data sets are cardiac short-axis view data obtained using a balanced SSFP sequence. The K-data set was obtained from 3.0T MRI for volunteers and Y-data set was obtained from 1.5T MRI for patients. The Y-data set had lower image quality than the K-data set. In addition, aliasing error was present along the direction of the phase-encoding gradient. Despite these differences, TL or BTL resulted in significant improvements and shorter learning times compared with SL, meaning that a high similarity between source data and target data might not be necessary for TL or BTL. Further application of BTL with two data sets that have low similarity to each other is a challenge to be addressed.

In conclusion, we developed BTL to improve the generality of a NN with a limited amount of data. BTL used a combination of source and target data sets for the target task. Consequently, the NN learned the target data while retaining source knowledge to the maximum extent possible. The NN that performed the proposed BTL showed the best performance in terms of learning speed and learning curves. It showed the highest reconstructed-image quality with the lowest NMSE for the test data set. Thus, BTL is an effective way of learning for NNs in the medical-imaging domain where both quality and quantity of data are always limited.

Acknowledgments

This work was supported by a grant (No. NRF-2019R1A2C2005660) of the National Research Foundation (NRF) funded by the Korea government (MSIT). The present research was also supported by a Research Grant of Kwangwoon University in 2019.

REFERENCES

1. Lee VS. Cardiovascular MRI: physical principles to practical protocols. Philadelphia: Lippincott Williams & Wilkins, 2006
2. Manning WJ, Pennell DJ. Cardiovascular magnetic resonance. 2nd ed. Philadelphia: Saunders, 2010
3. Park J, Hong HJ, Yang YJ, Ahn CB. Fast cardiac CINE MRI by iterative truncation of small transformed coefficients. Investig Magn Reson Imaging 2015;19:19-30
4. Yoon JH, Kim PK, Yang YJ, Park J, Choi BW, Ahn CB. Biases in the assessment of left bentricular dunction by compressed sensing cardiovascular CINE MRI. Investig Magn Reson Imaging 2019;23:114-124
5. Lustig M, Donoho D, Pauly JM. Sparse MRI: the application of compressed sensing for rapid MR imaging. Magn Reson Med 2007;58:1182-1195
6. Lee D, Lee J, Ko J, Yoon J, Ryu K, Nam Y. Deep learning in MR image processing. Investig Magn Reson Imaging 2019;23:81-99
7. Wang S, Su Z, Ying L, et al. Accelerating magnetic resonance imaging via deep learning. Proc IEEE Int Symp Biomed Imaging 2016;2016:514-517
8. Yang Y, Sun J, Li H, Xu Z. Deep ADMM-Net for compressive sensing MRI. Adv Neural Inf Process Syst (NIPS), 2016:10-18
9. Yang G, Yu S, Dong H, et al. DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction. IEEE Trans Med Imaging 2018;37:1310-1321
10. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. Adv Neural Inf Process Syst (NIPS), 2014;2672-2680
11. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015
12. Yu S, Dong H, Yang G, et al. Deep de-aliasing for fast compressive sensing MRI. arXiv preprint arXiv:1705.07137, 2017
13. Zhu J, Yang G, Lio P. How can we make GAN performs better in single medical image super-resolution? A lesion focused multi-scale approach. Proc IEEE Int Symp Biomed Imaging (ISBI) 2019;1669-1673
14. Wang C, Papanastasiou G, Tsaftaris S, et al. TPSDicyc: improved deformation invariant cross-domain medical image synthesis. Int Workshop on Mach Learn Med Image Reconstr (MLMIR) 2019;245-254
15. Schlemper J, Yang G, Ferreira P, et al. Stochastic deep compressive sensing for the reconstruction of diffusion tensor cardiac MRI. Int Conf Med Image Comput Comput Assist Interv (MICCAI) 2018;295-303
16. Zhu J, Yang G, Ferreira P, et al. A ROI focused multi-scale super-resolution method for the diffusion tensor cardiac magnetic resonance. Proc Int Soc Magn Reson Med (ISMRM) 2019;1
17. Hyun CM, Kim HP, Lee SM, Lee S, Seo JK. Deep learning for undersampled MRI reconstruction. Phys Med Biol 2018;63:135007

18. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. Int Conf Med Image Comput Comput Assist Interv (MICCAI) 2015;234-241
19. Kofler A, Dewey M, Schaeffter T, Wald C, Kolbitsch C. Spatio-temporal deep learning-based undersampling artefact reduction for 2D radial cine MRI with limited training data. IEEE Trans Med Imaging 2020;39:703-717
20. Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2010;22:1345-1359
21. Park SJ, Yoon JH, Ahn CB, Transfer learning for compressed-sensing cardiac CINE MRI. Proc Int Soc Magn Reason Med (ISMRM) 2020;2223
22. Oquab M, Bottou L, Laptev I, Sivic J. Learning and transferring mid-level image representations using convolutional neural networks. Proc IEEE Comput Vis Pattern Revognit (CVPR), 2014:1717-1724
23. Ciresan D C, Meier U, Schmidhuber J. Transfer learning for Latin and Chinese characters with deep neural networks. Proc Int Jt Conf Neural Netw (IJCNN) 2012;1-6
24. Meng D, Zhang L, Cao G, Cao W, Zhang G, Hu B. Liver fibrosis classification based on transfer learning and FCNet for ultrasound images. IEEE Access 2017;5:5804-5810
25. Chen A, Zhou T, Icke I, et al. Transfer learning for the fully automatic segmentation of left ventricle myocardium in porcine cardiac cine MR images. Int Workshop on Stat Atlases Comput Models Heart (STACOM) 2017;21-31
26. Dar SUH, Ozbey M, Catli AB, Cukur T. A transfer-learning approach for accelerated MRI using deep neural networks. Magn Reson Med 2020;84:663-685
27. Andreopoulos A, Tsotsos JK. Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI. Med Image Anal 2008;12:335-357
28. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Proc Int Conf Artif Intell Stat (AISTATS) 2010;249-256
29. Seitzer M, Yang G, Schlemper J, et al. Adversarial and perceptual refinement for compressed sensing MRI reconstruction. Int Conf Med Image Comput Comput Assist Interv (MICCAI) 2018;232-240