1. Design [15%] (Wang, Zhou) o neural network description and justification: introduce adopted customised convolutional net and u-net, their structures, properties, expected performance and why they are chosen (justification) o experiment factors: first introduce the basic pipeline of the experiment (check mindmap), then choose important/ malleable factors in the steps to experiment with
2. Implementation [20%] (Liu, Zhou) o how to implement in detail: data processing and coding structure o performance analysis mechanisms: training, testing methods, metrics (SSIM) o comments in the code: finished code to-date already commented
3. Experiments [45%] (Xu) /with other two if cannot be finished in time o description of experiments to optimise generalization performance o results presentation, presented in a statistical rigourous manner o how the training and test data sets are used
4. Conclusions [10%] (Xia) o key findings: check references' structure and expression

1. Introduction

1) fast MRI

Magnetic resonance imaging (MRI) plays an important role in advanced medical test. However, the long acquisition time in MRI often exceeds half an hour which results in low patient throughput and high patient anxiety. Hence, a faster MRI process is required to reduce the examination cost and to provide a more comfortable experience for patients. People gradually pay more attention on safety problems caused by its long duration in MRI machines as well. Some acute diseases intrinsically requires the treatment measures to provide fast and valid judgement on the state of illness since long diagnosis time will deteriorate the patients' condition and lead to secondary injury.

MRI is an indirect process which produces cross-sectional images of the subject’s anatomy by their frequency and phase measurements, whereas typical direct process involves spatially-related measurements. The measurements are regularly correlated to a multidimensional Fourier-space representation of an imaged body. This is known as k-space.

Faster MRI process can be achieved by scanning a undersampled part of the k-space. A 4-fold acceleration can reduce the scanning time significantly from over half an hour to only a few minutes. Comparing with classical methods, image reconstruction with neural network can yield better imaging results and handle image data more efficiently when larger image with higher resolution are processed, whereas the cost of classical inverse optimisation approach increases non-linearly. The spatially-resolved image m is then estimated from the undersampled k-space by performing an inverse multidimensional Fourier transform, where m̂ is a noise-corrupted estimate of the true image m, P() is a mask matrix selection over full k-space y:

$$\hat{m} = F^{-1}(P(y))$$

2) datasets

The datasets used in the fast MRI coursework are provided in prior with baseline functions, which is extracted from the open fastMRI Dataset(http://fastmri.med.nyu.edu/ (http://fastmri.med.nyu.edu/)), including a training dataset and a test dataset. By inspecting the key values of the HDF5 files, we find out that the training data is the raw k-space image generated from thousands of MRI examination, which is used for both training and validation. Each data in the test set consists of a 4-fold and 8-fold undersampled k-space data with the masks that generate them, respectively.

3) task

Essentially, our learning task in the coursework is to develop a method to reconstruct accurate images from the masked k-space data with both undersampling rates in the test set. The results are to be measured by the similarity to the ground truth images generated by the original test data.

The main goal we want to achieve in the coursework is to propose a deep learning approach which is capable of reconstructing high accuracy images even from undersampled scan images with large acceleration rates. By training a fine-tuned U-net neural network, we obtain over 40% relative improvement of reconstructed image quality using 4-fold and 8-fold accelerated scan than using the masked data directly.

2. Design

1) neural network description and justification

The coursework is considered as an image reconstruction problem. The neural network is fed with samples of input images and their better constructed peer. It is expected that, by learning a deep strcuture of layers, the trained model learns to find the intrinsic phases and magnitude among local neighbourhood between the pairs of images so that it can be applied to the reconstruction of other undersampled input.

In addition, the fast MRI problem is defined by, albeit the structure similarity stands, a deeper connection that the k-space data that generates the input image is a spatial subset of the k-space data that generates the ground truth image. Due to the Fourier Transformation process, it gives them spatio-temporal dependencies to each other. In another word, an image reconstruction method should be able to 'peek' through fractions of waves on certain frequencies to generate the full image.

To develop a specific methodology on the problem, we first consider Convolutional Neural Network(CNN) for its capability of learning spatial properties and reducing imaging noise. U-net is later introduced to the team because its unique structure is good at partitioning different pixels. It is smartly built with blocks of convolutional layers in an autoencoder style. The team decide to build baseline models on CNN first for a better understanding of the problem, and go for U-net architecture later if it yields better performance.

Layer description - Zhou

2) naive CNN model

The convolution layer introduces a concept of local perception, which tells that in the process of the human brain recognizing a picture, the whole picture is not recognized at the same time, but each feature in the picture is first perceived locally, and then the local operation is comprehensively performed at a higher level to obtain global information.

To fit the image into the model, the input is expected to have 4 dimensions. The raw file data is first transformed and loaded as input in iteration by a data loader. The implementation process before training will be discussed later.

We then develop our CNN model with three 2D convolutional layers, the first two of them with a kernel size of 5. The number of output channels starts from 16, which is doubled in the second layer. The last convolutional layer is applied with a kernel size of 1 to reduce the channels of feature maps in a single perceptron.

Each convolutional layer is followed by an activation layer. The activation function makes a non-linear mapping of the output of the convolution layer. In this model, we choose the ReLU function to make every iteration faster. After activation, feature space is reduced by a pooling layer. The pooling layer is mainly used for feature dimension reduction, by compressing the number of data and parameters. It helps to reduce model verfitting, and improves the fault tolerance of the model. Here we choose a max pooling over a (2, 2) window.

Finally, the feature maps are converted to a 1-dimensional vector by up-sampling and the third convolution layer. Since the features are in 1 dimension now, we feed them to 3 fully connected layers to produce the final result. After several previous convolutions, activations, and pooling, the model will fully learn a high-quality feature picture fully connected layer.

CNN can directly process grayscale images and can be used directly to process image-based classification. CNN has unique advantages in image processing due to its special structure of local weight sharing. The layout is closer to the real biological neural network. Weight sharing reduces the complexity of the network, especially for multidimensional input vectors. The feature that allows images to be directly entered on the network avoids the complexity of rebuilding data when retrieving and classifying features.Maybe the performance of CNN used for MRI processing is not well.

3) U-Net

Then, we will train the neural network model by U-net based on CNN. When it comes to classification, the information provided by the pixels is always taken into account. However, this information generally includes two types: one is environmental field information, and the other is detailed information. The pixel-based approach has a great deal of uncertainty about the choice of the form. Choosing a size that is too large not only requires more pooling layers to make the environmental information appear, but also loses the local detailed information. But U-net uses a network structure that includes both down sampling and up sampling. Down sampling is used to gradually display the environmental information, and the process of up sampling is to combine the down sampling information of each layer and up sampling input information to restore the detailed information, and gradually restore the image accurately. Besides, U-Net combines the location information from the down sampling path to finally obtain a general information combining localization and context, which is necessary to predict a good segmentation map.

U-NET model are mainly composed of two parts :

- Down sampling part is also called the feature extraction. U-NET starts with one channel. Firstly, each convolutional layer did stride-2 convolution operations, and then we encapsulate these two convolution layers into a class ConvBlock. Specifically, each convolution layer followed by kernel, instance normalization, ReLU activation and drop out, kernel size is 3 * 3 and the number of channels doubles after the first convolution operation, and remains the same the second time.They lose two pixels for every stride-2 convolution because of a lack of padding.After the first 3X3 convolution operation after each pooling layer, the number of 3X3 convolution kernels doubles. Dropout is often used to randomly delete some neurons in a neural network to prevent overfitting and optimize the model by using normalization at the same time. Then, We choose a max pooling over a (2, 2) window used to half the size of feature map. Then again, the convolution and max-pooling steps are followed until the last pool layer. The number of pool layers depends on experimenters.
- Up sampling, which is also known as deconvolution, or transpose convolution, is a critical part of U-net. There is a way, know as bilinear interpolation which means the feature map merged at the same scale as the number of channels corresponding to the feature extraction part. Therefore, before merging, the model will copy and crop the feature map in the shallow layers, which results in the final output is a region in the center of the input.After the first 3x3 convolution operation after each merging, the number of 3X3 convolution kernels becomes half. Unlike down sampling, up sampling double the size of image. And then passing through ConvBlock. Finally, in the last layer, we use the kernel 1 * 1 to accept the output image. U-NET is derived from CNN. U-shaped structure is the biggest feature of U-NET. U-NET performs several down sampling and use the contact connection on the corresponding layer, which ensured that the finally recovered feature maps incorporated more low-level features. Multiple up sampling also makes the information such as edge restoration of the image more detailed.

2) experiment factors

3. Implementation

Data loading - Wang

Firstly, we load the data by using data_loader.ipynb, and then we show slices in the sample as k-space data and transform k-space data to real images and ground truth. K-space is an abstract space (three-dimensional space) or a plane (two-dimensional space). As MR imaging data is arranged at a specific K-space position according to different spatial frequencies, and finally transformed into an image, so all we need to do is to transform k-space data to real images and compare the original image and the output image by SSIM. Explicitly, K-space uses spatial frequency as the unit (Hz / cm), the spatial frequency K is described by the three mutually perpendicular components Kx, Ky, and Kz. These three vectors correspond to a three-dimensional frequency space, so this abstract space is called K-space.

Since the unavailablity of the original data, we can validate our reconstruction method only on the training set.

First of all, we try to construct a CNN network by ourselves. There are four steps of the CNN part, which are data processing, the construction of network, training model, and testing model.

Firstly, for data processing, we need to change the dimension of input data. The initial input and target data shape is (1, 640, 372, 2). So in data processing, firstly we compute absolute value to get a real image and the shape is (1, 640, 372) now. Secondly use unsqueeze function to raise dimension to (1, 1, 640, 372) because the upsample function in model require a 4 dimensions input, Finally use T.center_crop to crop the images to the central 320x320 pixel region(1, 1, 320, 320).

Then for network model, it contains two convolution layers, two pooling layers, upsample and a 1*1 convolution layer and ReLU activation function. The kernel of first and second convolution layer are both 5*5， the stride and padding are both 1 and 2 separately, and the out_channels is 16 and 32 respectively. And the two pooling layers are both 2*2 of maxpooling. For upsample, the scale is 4 and mode is bilinear and each layer contains a ReLU. First of all, the input data size is (1, 1, 320, 320),after the first convolution, I extracted 16 features, the size change to (1, 16, 320, 320), because (320+2*2-5)/1+1=320. Then passing the maxpooling, the size reduce to (1, 16, 160, 160), the same with last layer, the size continue to change to (1, 32, 80, 80) after the second convolution and maxpooling. Then we need to use upsample to inprove size to (1, 32, 320, 320) and at last change to (1, 1, 320, 320) by 1*1 convolution layer.

Turn to training model, I used the dataloader function from pytorch. Because the size of input and target data is same, so the loss function we can choose could be l1 or l2(Mse), firstly i choose l1 and sdg for optimizer. I set learning rate 0.001, after data processing like before and begin to train. because the limit of time and the gpu of my laptop, I just trained 2 epochs and save the model. However, I observe the loss has a remarkable fluctuation and keep unchanged in the end.

Fig.1. The reduction of loss curve

By testing model, the performance of my model is not good. I input the testing data, set AF = 8 and use ssim function test, the average ssim is just 0.45 and the image is below. So i try to modify the network and adjust hyperparameter to optimize the model. Firstly, i change the loss function to mse and reduce learning rate to 0.0001. Besides them, I also add a convolution layer to extract 64 features, and add 2 1*1 convolution to reduce size progressively. After training 2 epochs and testing, I find the performance is still not good, but I do not have enough to continue to adjust model, and we decide to use unet model. Fig.2. the left is the image with undersampling rate 8, the center is the target image and the right is the image after inputting in model.

For dataloader, firstly use load_data_path function to load all file names, paths and slices. Next getting the dataset by MRIDataset function, finally use DataLoader function to get a data iterator which can iterate each set of data. I would mainly introduce the get_epoch_batch function in MRIDataset. It could random select a few slices from each volume. Firstly it loads the data from file and transform to tensor, the apply random mask by MaskFunc function. At last after undersampling and normalizing data, we got the input data.

To experiments part

1) how to implement in detail

2) performance analysis mechanism

After gaining the output images, we can transform them to ground truth to make a comparison. And the value range of SSIM is [0,1], which means the larger the value is, the better the performance is. Structural similarity (SSIM) is also a full-reference image quality evaluation index. It measures the similarity of two images from the aspects of brightness, contrast, and structure. The SSIM algorithm is designed to take into account the visual characteristics of the human eye and it is more in line with the human eye's visual perception than traditional methods. MSE or PSNR algorithms are both evaluations of absolute errors. For the fuzzy changes in the human's perception of the structural information of the image, the model also introduces some perception phenomena related to the changes in perception, including the brightness mask and the contrast mask. The structural information refers to the internal dependency between pixels, especially amongst pixels that are close in space. These dependencies carry important information on the target's visual perception.

4. Experiments

```
In [ ]: noise corrupted
```

The first experiment is to determine which baseline model we should use in the following development. By implementing both customised CNN model and the constructed U-net model to the training dataset, two models are trained over 20 epochs with learning rate of 0.001. The optimizer and loss function are set by default as Stochastic Gradient Descent(SGD) and L1-loss, respectively. To evaluate the baseline over a comparable condition, the number of output channels of the first convolution layer in U-net is set to be 32. As shown in figure[?],

1) description of experiments to optimise generalization performance

2) results presentation in a statistical rigourous manner

3) how the training and test datasets are used

5. Conclusions

1) key findings

In this paper, we have implemented a CNN-based and U-Net-based technique on undersampled k-space data with high accuracy and recall. Various deep-learning techniques based on CNN have been proposed to address the problem of reconstructing MRI from undersampled k-space data. Therefore, we tried CNN as one of our solution to reconstruct the images. However, in comparison of U-Net single-coil baseline model, CNN obtain relatively low performance and quility of reconstructed images. For the results acquired from U-Net baseline model, we conclude that the models with very large capacities trained on large amounts of data can enable high acceleration factors and larger U-Net baseline model shows better performance.

Although our dataset was originally acquired for a focused task, the inclusion of raw k-space data allows methods to be developed for the imaging pipeline itself, in principle allowing them to be applied on any MRI scanner for any imaging task.

We hope that the availability of this dataset will accelerate research in MRI reconstruction, and will serve as a benchmark during training and validation of new algorithms.

6. Description of Contribution

1. Introduction [10%] (Xia)

2. Design [15%] (Zhou)

3. Implementation [20%] (Wang, Liu, Zhou)

4. Experiments [45%] (Xu, Liu)

5. Conclusions [10%] (Xia, Xu)

6. Coding (Xu, Liu)

7. Literature reference(Xia)

References

[1] A. H. Baykan, S. Sahin, I. Inan, S. Kafadar, and S. M. Erturk. RapidSequence MRI Analysis of Acute Abdominal Pain.ResearchGate, 92019.

[2] K. G. Hollingsworth. Reducing acquisition time in clinical MRI bydata undersampling and compressed sensing reconstruction.Physicsin Medicine and Biology, 60(21):R297–R322, oct 2015.

[3] B. J. Kim, H. G. Kang, H.-J. Kim, S.-H. Ahn, N. Y. Kim, S. Warach,and D.-W. Kang. Magnetic Resonance Imaging in Acute IschemicStroke Treatment.J. Stroke, 16(3):131, 9 2014.

[4] J. Liu and D. Saloner. Accelerated MRI with CIRcular CartesianUnderSampling (CIRCUS): a variable density Cartesian samplingstrategy for compressed sensing and parallel imaging.Quant. ImagingMed. Surg., 4(1):57–67, 2 2014.

[5] A. S. Lundervold and A. Lundervold. An overview of deep learningin medical imaging focusing on MRI.Z. Med. Phys., 29(2):102–127,5 2019.

[6] K. Nael and W. Kubal. Magnetic Resonance Imaging of Acute Stroke.Magn. Reson. Imaging Clin. N. Am., 24(2):293–304, 5 2016.

[7] I. Oksuz, B. Ruijsink, E. Puyol-Anton, A. Bustin, G. Cruz, C. Prieto,D. Rueckert, J. A. Schnabel, and A. P. King. Deep Learning using K-space Based Data Augmentation for Automated Cardiac MR MotionArtefact Detection.arXiv, 8 2018.

[8] P. Putzky and M. Welling. Invert to Learn to Invert.arXiv, 11 2019.

[9] M. E. Ryan, A. Jaju, J. D. Ciolino, and T. Alden.Rapid MRIevaluation of acute intracranial hemorrhage in pediatric head trauma.Neuroradiology, 58(8):793–799, 8 201