Due: 2021/01/03

Course Project

Computer Science Department in SJTU

1 Brief Introduction

1. 3 people for each group.

2. This project is group work. Each group is required to choose one topic from the given topics as your course project and submit a report by the end of this term.

3. Finally, each group should submit source code and report, accounting for 20% and 80% of the total score respectively.

4. **Submission deadline: 8:00 pm, 2021/01/03**
   Address: ftp://public.sjtu.edu.cn
   Username: huangwenjing
   Password: public

2 Description for topics

2.1 Topic A

Studies on bidirectional neural networks can be traced back to auto-association in the 1980s. One typical example was Least Mean Square Error Reconstruction (Lmser) self-organizing network that was first proposed in 1991 [24, 25]. Proceeded beyond AE, Lmser is featured by a bidirectional architecture with several built-in natures, for which readers can refer to Table I in Ref. [26]. One major nature of Lmser is Duality in Connection Weights (DCW). DCW refers to using the symmetric weights in corresponding layers in encoder and decoder such that the direct cascading by AE is further enhanced. The purpose of designing such a symmetrical structure in Lmser is to approximate identity mapping per two consecutive layers simply through $W^TW \approx I$, where $W^TW = I$ holds only if the weight matrix W is orthogonal. However, when W is not orthogonal even not a square matrix, what should we do to minimize the discrepancy between $\mathbf{x}$ and $W^dW^e\mathbf{x}$? One way is to constraint W^d to be the pseudo inverse of W^e. You are required to

1) Build a reconstruction network with the constraint that the weight matrix in the decoder to be the pseudo inverse of the encoder in the corresponding layers

2) Compare the reconstruction performance of the models with and without the constraint in 1) on the given data sets.

3) You can test their performance in different scenarios (for example, small training sample size) to explore what the constraint in 1) can bring to your model.

In convolutional networks, implementing the constraint in 1) need you to rewrite the deconvolution operation in some deep learning framework (like PyTorch, TensorFlow), thus only the CNN version is encouraged but not required. We would also like to see some other alternative constraints for CNN as long as they are designed to approximate the inverse mapping.

Data sets:

For fully connected networks: MNIST[1] [12], F-MNIST[2] [23]

For CNN: STL10 [2]

Note: The dataset for this topic requires < 100M of space and auto-encoder trains in < 1h on each data set.

2.2 Topic B

The Lmser net works in two phases, i.e., perception phase and learning phase. (For more information, please refer to [25]). In the perception phase, the input triggers the dynamic process by passing the signals up from the bottom layer, while simultaneously the signals in the upper layers will be passed down to the lower layers. It has been proved that the process will converge into an equilibrium state [25]. After the process reached its stable state, the top-down signal to the input layer u_0 is regarded as a reconstruction of the input x. In the perception phase, it is hard to implement the exact dynamic process in multiple layers because neurons are highly coupled and it may be time-consuming to reach convergence. Some works made some approximations for the dynamic process [7]. You can refer to the previous works or make new implementation. We believe that the perception phase can help to refine the reconstruction u_0 and help to complete corrupted images. You can use some popular networks as your backbone (for example, partial convolution[14], gated convolution [29] or contextual attention [28], etc.) and implement the perception phase to complete the corrupted image with rectangle masks and free-form masks [29].

Data sets: CelebAHQ[8], a high-quality version of the human face dataset generated from CelebA[15]. You can download the dataset (data256$\times$256.zip) from the link[3] in the footnote.

Note: The dataset for this topic requires < 500M of space and PConv[14] trains in 3 days on CelebA-HQ.

[1] `http://yann.lecun.com/exdb/mnist/`

[2] `https://github.com/zalandoresearch/fashion-mnist`

[3] `https://drive.google.com/file/d/1O89DVCoWsMhrIF3G8-wMOJ0h7LukmMdP/view?usp=sharing`

2.3 Topic C

Deep predictive coding networks (PCN) [22], a bi-directional and recurrent neural net, is proposed for object recognition. It has feed-forward, feedback, and recurrent connections, which is similar with Lmser [25, 26]. You are required to

1) Build a model based on Lmser for object recognition and try to raise the performance.

2) Compare the performance of the model you build with PCN.

Data set: CIFAR10 [4][11], MNIST[5] [12], SVHN[6][17]

2.4 Topic D

2.4.1 Topic D1

The project is inspired by *ISBI Challenge*[7] and designed for students to learn to build a model for a specific task. You are required to build a model for medical image segmentation. You are encouraged to add some feedback/recurrent link [26] from decoder to encoder to get better performance.

Data set: The ISBI 2012 EM Segmentation Dataset [1] can be downloaded from here[8]. The data description is same with *ISBI Challenge* except that we split the raw train data set (consist of 30 samples) into two parts: the new train set and new test set. The downloaded data set consists of 30 samples, 25 for train and 5 for test. You are required to train and test your model on the newly split data sets.

Note: The dataset for this topic requires $<$ 100M of space and U-net[20] trains less than 3 hours on the data set.

2.4.2 Topic D2

The project is inspired by ISBI 2019 SegTHOR Challenge [19] and designed for students to learn to build a model for a specific task. You are required to build a model for multiple organ segmentation from CT slices. You are encouraged to use bi-directional network proposed in LMSER[26] to get better performance.

[4] `https://www.cs.toronto.edu/~kriz/cifar.html`
[5] `http://yann.lecun.com/exdb/mnist/`
[6] `http://ufldl.stanford.edu/housenumbers/`
[7] `http://brainiac2.mit.edu/isbi_challenge/`
[8] `https://jbox.sjtu.edu.cn/l/Ou6oan` (password: hfnu)

Data set: The ISBI 2019 SegTHOR Dataset [19] can be downloaded from here[9]. The data description is same with *ISBI Challenge*. Every scan has a size of 512x512x(150x284) voxels. And the whole dataset includes 40 cases, 1-30 is train set and 31-40 is test set, you are required train you model on train set and test your model on test set.You may need some pre-processing for those data.

Note: Maybe the dataset for this topic a little bit large for you and training may take about 2 days.

2.5 Topic E

Recently progress on the development of deep learning has advanced the state-of-the-art performance on face recognition. For the need for security, face recognition has become a research problem that we all pay more attention to. At present, many state-of-the-art methods have achieved good performance in high-resolution(HR) face images. However, when they process low-resolution(LR) and small-size face images, such as 16×16, 20×20, especially images taken under very bad conditions, the results are usually not very ideal. Low-resolution face recognition(LRFR) is still a challenging task due to the margin between LR probe face images and HR gallery images. In this task, we uniformly require everyone to use 16×16 as the size of low-resolution images.

2.5.1 Topic E1

Here are one paper reference [16], we hope you can repeat and have new ideas for improvement. You can use some pictures and tables to do some analysis and comparison to show your experimental results.

- **Training set:** CASIA-WebFace [27] dataset is provided to train your model. Because the raw images take up too large space, we have done face alignment and cropped images to 120×120 size in the gray channel. Later, you can continue processing according to your needs based on the data set we provide. You can get the dataset from here[10]. You can also preprocess dataset by yourself according to insightface[11] to do face detection and face alignment, and finally crop to the target size.

- **Testing set**: SCface [5] dataset is provided to test your model performance. Here is an introduction to the datasets[12]. It contains 130 subjects and taken by different quality cameras at different distances to simulate real-world situations, supporting robust face recognition algorithm testing. We also have prepossessed images to 120×120 like the

[9] `https://jbox.sjtu.edu.cn/l/noXQob` (password: eidr)
[10] `https://jbox.sjtu.edu.cn/l/fJ6wsU` (password: agtj)
[11] `https://github.com/deepinsight/insightface`
[12] `https://www.scface.org/`

training set above. Here are references on how to do the test, [13, 16]. You can get the dataset from here[13].

For further improvement or innovations, here are some references [3, 4, 10, 18, 21]. Of course, the higher the result of the model on the test dataset, the better. You can also add other test datasets such as LFW[14] [6] and MegaFace [9] to fully demonstrate your model performance.

Note: The dataset for this task requires approximately 10G of space and takes about 20 hours on two XP GPUs based on Resnet18.

2.5.2 Topic E2

Here are some other references about face recognition and bidirectional network [3, 24], we hope that you can apply LMSER to the face recognition network to study whether reconstruction is helpful for face recognition. You are required to do two experiments on HR images (120×120) and on LR images (16×16) respectively. We recommend that use Resnet18 as the backbone of your model to make a fair comparison with others.

- **Training set:** Refer to the training set in 2.5.1.
- **Testing set**: LFW is provided to test your model. Remind that training and testing images should be the same resolution. You can download LFW from here[14].

Note: The dataset for this task requires approximately 10G of space and takes about 20 hours on two XP GPUs based on Resnet18.

3 Basic Assignments

3.1 Codes

Each group is required to submit your codes by providing a link to your GitHub repo (attached at the end of your report). If you do not know how to use GitHub, please visit its tutorial[15] for some advice. Codes will be judged by cleanness and readability, so remember to comment on your codes.

3.2 Report

Each group is required to turn in a report that contains your main ideas, methods, algorithms, experimental settings, and results. You can write your report in word or latex (encouraged), English (encouraged) or Chinese. The details of the four parts are:

[13] `https://jbox.sjtu.edu.cn/l/b1kNW8` (password: lixh)
[14] `https://jbox.sjtu.edu.cn/l/pn3mat`(password: buvv)
[15] `https://guides.github.com/activities/hello-world/`

Main Ideas: A brief introduction of your report, including your proposed methods and the performance.

Methods: The proposed method for the project, including a detailed description of your methods.

Algorithms: The algorithms you choose to optimize your model (how to train your model or update the parameters), including the motivation, detailed description and pseudo-code if necessary.

Experimental Settings: The experimental settings, including the structure of your network (if you have), learning rate, batch size, etc.

Results: The performance of your model on the test set.

Code link: The link to your github code.

Contribution percentage: At the end of report, you are needed to list the contribution percentage of each team member. A table template is shown below: .

Table 1: contribution percentage

student_id	name	%
02003391xxxx	Li Mei	50
02003391xxxx	Wang Mei	30
02003391xxxx	Liu Mei	20

References

[1] Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireşan, Alessandro Giusti, Luca M Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, 9:142, 2015.

[2] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[4] Shiming Ge, Shengwei Zhao, Chenyu Li, and Jia Li. Low-resolution face recognition in the wild via selective knowledge distillation. *IEEE Transactions on Image Processing*, PP(4):1–1, 2018.

[5] Mislav Grgic, Kresimir Delac, and Sonja Grgic. Scface–surveillance cameras face database. *Multimedia tools and applications*, 51(3):863–879, 2011.

[6] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008.

[7] Wenjing Huang, Shikui Tu, and Lei Xu. Deep cnn based lmser and strengths of two built-in dualities. *Neural Processing Letters*, pages 1–17, 2020.

[8] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[9] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4873–4882, 2016.

[10] Hanyang Kong, Jian Zhao, Xiaoguang Tu, Junliang Xing, Shengmei Shen, and Jiashi Feng. Cross-resolution face recognition via prior-aided face hallucination and residual knowledge distillation. 2019.

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] Pei Li, Loreto Prieto, Domingo Mery, and Patrick J Flynn. On low-resolution face recognition in the wild: Comparisons and new techniques. *IEEE Transactions on Information Forensics and Security*, 14(8):2000–2012, 2019.

[14] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. *CoRR*, abs/1804.07723, 2018.

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[16] Ze Lu, Xudong Jiang, and Chi Chung Kot. Deep coupled resnet for low-resolution face recognition. *IEEE Signal Processing Letters*, pages 526–530, 2018.

[17] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[18] Xingchao Peng, Judy Hoffman, Stella X Yu, and Kate Saenko. Fine-to-coarse knowledge transfer for low-res image classification. 2016.

[19] Su Ruan Bernard Dubray Roger Trullo, Caroline Petitjean. Segmentation of organs at risks in thoracic ct images using a sharp mask architecture and conditional random fields. *international Symposium on Biomedical Imaging*, 13(4):600–612, 2004.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[21] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[22] Haiguang Wen, Kuan Han, Junxing Shi, Yizhen Zhang, Eugenio Culurciello, and Zhongming Liu. Deep predictive coding network for object recognition. *arXiv preprint arXiv:1802.04762*, 2018.

[23] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[24] Lei Xu. Least mse reconstruction for self-organization: (i)&(ii). In *Proc. of 1991 International Joint Conference on Neural Networks*, pages 2363–2373, 1991.

[25] Lei Xu. Least mean square error reconstruction principle for self-organizing neural-nets. *Neural networks*, 6(5):627–648, 1993.

[26] Lei Xu. An overview and perspectives on bidirectional intelligence: Lmser duality, double ia harmony, and causal computation. *IEEE/CAA Journal of Automatica Sinica*, 6(4):865–893, 2019.

[27] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[28] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[29] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.