Fine-tuning Deep Neural Networks by Interactively Refining the 2D Latent Space of Ambiguous Images

Jiafu Wei[1] , **Haoran Xie**[2] , **Chia-Ming Chang**[3] , **Xi Yang**[1]
[1] Jilin University
[2] Japan Advanced Institute of Science and Technology
[3] The University of Tokyo
weijf21@mails.jlu.edu.cn

Abstract

Deep neural networks (DNNs) have achieved excellent results currently in the classification, while they may still suffer from ambiguous images which are similar across classes. By contrast, humans have a relatively good ability to distinguish these categories of images. Therefore, we propose a human-in-the-loop solution to assist the network in better classifying the images by leveraging human knowledge. To achieve this, we project the high-dimensional latent space trained by the network onto a two-dimensional workspace. The users can interactively modify the projected coordinates of inputs on the workspace using our designed tools, then the modified information will be fed back to the network to fine-tune it, which in turn, affects the network's classification results, thereby improving the accuracy of network classification.

1 Introduction

At present, Deep neural networks (DNNs) maintain excellent results in classification. However, they are still difficult to extract enough features to distinguish the ambiguous data that are very similar but belong to different classes. Compared with the networks, humans may have better classification capabilities for a small number of pictures or a group of abstract pictures. For example, as shown in Figure 1, in a large sketch dataset called Quick Draw [Jongejan *et al.*, 2016], many abstract hand-drawn pictures are difficult for machines to distinguish, while humans can distinguish them well.

Human-in-the-loop is a promising direction to leverage human knowledge to help machines solve intractability problems in the field of artificial intelligence [Nashed and Biswas, 2018], many works apply this idea [Igarashi *et al.*, 2016; Huang and Canny, 2019]. Moreover, a lot of works now revolve around interactive interfaces [Larsson *et al.*, 2020; Wang *et al.*, 2014], where users make changes to the data on the workspace to help machines do tasks that are difficult for them to do. [Yang *et al.*, 2019] proposed an approach that eliminates the problem of co-occurrence bias in the identification process through human participation. [Sakata *et al.*, 2019] introduced a new network called CROWNN that engages people in the classification process, learns how humans classify, and then better implements the classification task. [Asai *et al.*, 2020] combine a code editor with a scatterplot editor so that users can visually observe the data and modify the data interactively. [Chang *et al.*, 2021] focus on labeling tasks, aiming to improve the efficiency and accuracy of human labeling.

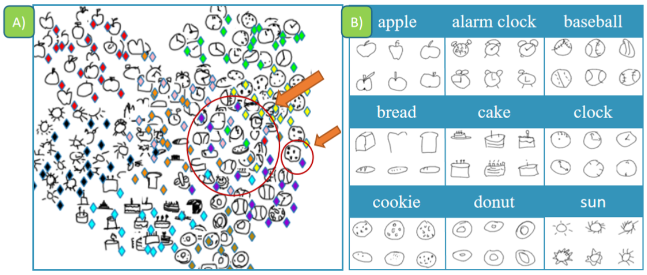

Figure 1: A) Visualized distribution of the features extracted by a DNN. The data in the big circle represent data of different categories mixed and there is no clear inter-kind boundary, and the data in the small circle represent the data with a wrong predicted result. We define these types of data as ambiguous images. B) The nine categories of images that are somewhat similar in the Quick Draw dataset.

In this paper, we describe a novel interactive approach to improving network accuracy by refining the 2D latent space of the inputs[1]. We first give a general overview of our work, introduce the rationale of our method, and then describe our experimental procedures and illustrate our conclusions with examples. At the same time, we designed an interactive user interface to validate our idea. We project high-dimensional features in a neural network onto a two-dimensional workspace, the position information of the projected point reflects the classification result of the point to a certain extent. Users can move the inputs they think are classified as wrong to the positions they think are correctly classified. Then, through the interactive user interface, we designed, the results of the movement can be fed back to the neural network for re-training, to achieve the purpose of fine-tuning the neural network. We conduct user studies to evaluate the performance of our proposed method, the feedback from users shows our method obtained the expected results.

Our method has the following benefits: 1. Help the net-

[1] https://youtu.be/5EvDa3XiTiY

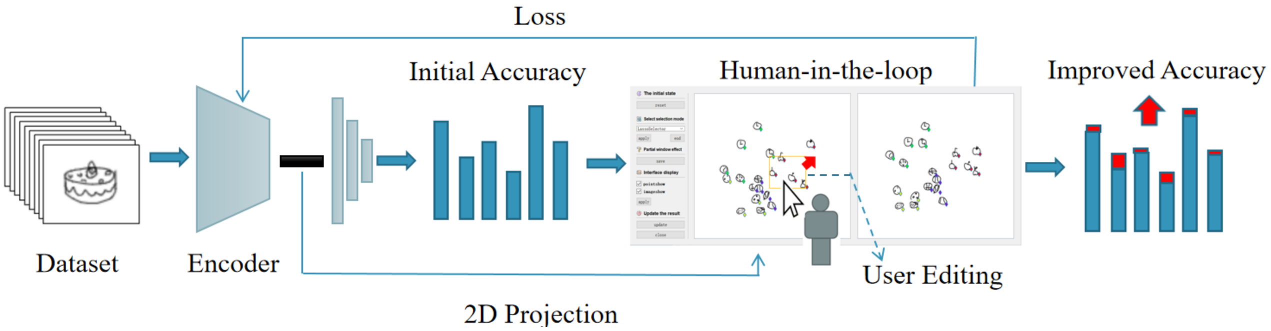

Figure 2: An overview of how our network works.

work to distinguish ambiguous data and achieve the purpose of improving classification accuracy. 2. Get a more understandable latent space for the user. Through the user interface, we designed, the positional relationship and distance between classes can be adjusted. 3. Guide the learning process of networks. Not only does it allow the network to learn features faster, but also allows small networks to achieve the same level of performance as large networks.

2 Method

We add a projection branch to the pre-trained network to allow direct and convenient use and modification of the latent space on the screen, and we use dimensionality reduction methods to project higher dimensions into 2D. Here we use principal component analysis (PCA) as a dimensionality reduction method. First, the high-dimensional vectors are converted into two-dimensional vectors through PCA, and then the result obtained by PCA is used as the supervision value to train the projection branch. After the network training is completed, the data can be projected onto our two-dimensional workspace through the projection branch, realizing data visualization. At the same time, the user can modify the coordinate values of the projected points in the workspace to fine-tune the network, as shown in Figure 2.

Fine-tuning Loss Function. The result of DNN classification is controlled by the loss function. The loss function is weighted by the classification loss and the distance difference loss before and after adjustment (see Formula 1). The classification loss loss$_{cls}$ is obtained by performing cross-entropy calculation (CE) between the predicted label and the true label. The distance difference loss loss$_{dis}$ can only be obtained from these points whose positions have been adjusted. The specific method is to calculate the difference between the coordinates before and after the movement through the L2 norm. The parameters (w$_{cls}$, w$_{dis}$) are used to adjust the balance between the two losses. Through these losses, the network can be fine-tuned with the points the user moves. And then, the output result will be closer to the result changed by the user, thereby achieving the purpose that the user can assist the network in better classification.

$$Loss = w_{cls} \times loss_{cls} + w_{dis} \times loss_{dis} \tag{1}$$

User interface. We design a user interface to help users modify the 2D latent space, as shown in Figure 3. Our UI is divided into three parts from left to right, namely the function bar, the modification window, and the result window. The reset button in the function bar is used to restore the two windows to the initial state, and the update button is used to transfer the modified content in the modified window to the network for retraining and updating the coordinate values in the result window. The coordinate values in the modification window can be changed, and the coordinates before and after modification are used to fine-tune the network. The result window is used to visualize the updated 2D coordinate values. We provide three methods for moving coordinate values, namely, dragging with the mouse to move a single coordinate, using the lasso tool or the selection box tool to move multiple coordinates.

3 Experiments

Experimental Data Selection. The reasons for the emergence of ambiguous data can be divided into two parts. One is that there are deviations in the sketch data itself, which is caused by human factors such as the sketches drawn by the painters not being clear enough or not being careful enough. Second, there are different degrees of similarity between sketches of different categories, which will hinder the effect of network classification. With the improvement of the similarity between the sketches, the difficulty of network classification will also increase. Our experiments revolve around ambiguous data. We use the Quick Draw dataset to test our method. It is a large sketch dataset with over a billion hand-drawn sketches. This dataset provides images of various kinds and formats. We selected nine kinds of sketch data for experiments and used the DNN to complete the image classification task. The nine types of images selected are shown in Figure 1 B), these images have a certain degree of similarity, and people will easily distinguish them.

Training Details. We performed all our experiments using PyTorch. The base network we use for our experiments consists of four convolutional layers and three fully connected layers. The training and fine-tuning process of the network is optimized using the stochastic gradient descent (SGD) (with momentum). We train the training set until the training result is stable, and obtain the initial pre-training model. During

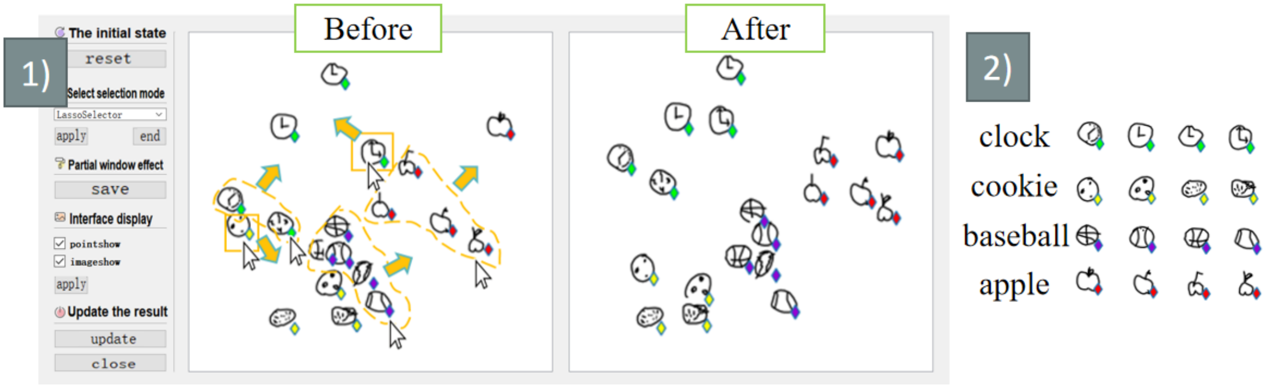

Figure 3: 1) The overall layout of the user interface. 2) The categories corresponding to the sketches in 1). The sketch data are all from the Quick Draw dataset, and these colored points on the lower right corner of the sketch represent the predicted value corresponding to the sketch data, the dots of the same color represent the same type of data.

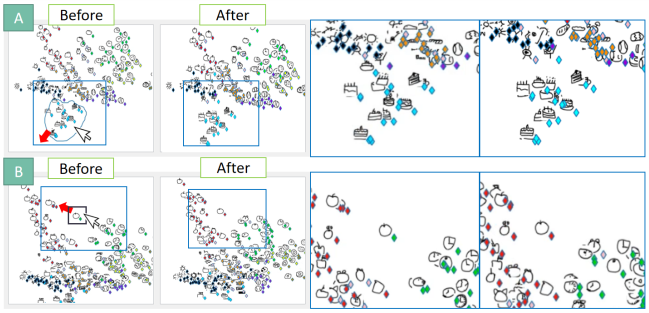

Figure 4: The third and fourth pictures in A are partially enlarged pictures of the first and second pictures, which are used to compare the changes before and after, and B is the same. After moving, the inter-class distance between the original adjacent classes in Figure A becomes larger, the inter-class classification is clearer, and the originally wrong predicted value in Figure B becomes the correct predicted value (green turns to red). And the accuracy rates of A and B after moving are 86.48% and 86.36%, improved by 0.15% and 0.03% respectively.

training and fine-tuning, the learning rate and momentum are set to 0.04 and 0.9, respectively. All fine-tuned results go through one to three epochs. We set (w_{cls}, w_{dis}) to be (0.89, 0.11) respectively. The number of each type of test data is 500, and the number of points that we move in the experiment ranges from 1 to 30.

Experimental Procedure and Results Analysis. We selected two kinds of ambiguous data for experiments. The data of these cases has the characteristics that it is difficult for machines to distinguish them correctly, but it is easy for humans to distinguish them. We test the data of these cases separately, as shown in Figures 4 A) and B). The test set accuracy rate before moving is 86.33%, and when the network is fine-tuned with proper movement, the accuracy can be achieved at 86.48% and 86.36%, respectively. It can be seen from the test results that the correct motion can improve the accuracy of network classification, and the location of the updated data also moves closer to the location where the user moves, making the location more reasonable. At the same time, the correct motion can also adjust the predicted value of the data with the wrong predicted value to the correct predicted value, as shown in Figure 4 B), the predicted value of the original data is green (representing the clock), and the predicted value of the shifted data turns red (representing the apple), completing the transition from the wrong predicted value to the correct predicted value.

We invited four users to evaluate our user interface. We give each user four to six minutes to understand and learn to use our user interface. Next, we assigned participants several tasks to determine the utility of our system. Tasks include validating whether our method improves accuracy and whether our designed user interface is convenient. All users agreed that our user interface is not only convenient to use but also the form of two-dimensional point coordinates that can intuitively express the network's classification of different sketch data. In addition, the method that can affect the effect of network classification by moving the coordinates of the sketch is very novel and has good development potential.

4 Limitations and Conclusion

The effect of our proposed method has been validated on small datasets, however, it should be further evaluated on the common large datasets. Besides, the projection method and distance matching between the 2D space and the original latent space requires further research.

In this paper, we propose an interactive user interface for improving the accuracy of network classification. By projecting the features obtained by neural network classification onto a two-dimensional workspace, users can manually change the positions of these projected two-dimensional points, eliminating possible deviations in the dataset, and then positively affect the results of network classification. Our work proves that the combination of neural networks and human-in-the-loop methods can transfer the advantages of humans to the network. Human participation can eliminate some of the deviations that may occur in the classification process of the machine and can improve the accuracy of neural networks in classification tasks. In future work, we will further work based on the above limitations.

References

[Asai *et al.*, 2020] Kentaro Asai, Tsukasa Fukusato, and Takeo Igarashi. Integrated development environment with interactive scatter plot for examining statistical modeling. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2020.

[Chang *et al.*, 2021] Chia-Ming Chang, Chia-Hsien Lee, and Takeo Igarashi. Spatial labeling: Leveraging spatial layout for improving label quality in non-expert image annotation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2021.

[Huang and Canny, 2019] Forrest Huang and John F Canny. Sketchforme: Composing sketched scenes from text descriptions for interactive applications. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*, pages 209–220, 2019.

[Igarashi *et al.*, 2016] Takeo Igarashi, Naoyuki Shono, Taichi Kin, and Toki Saito. Interactive volume segmentation with threshold field painting. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 403–413, 2016.

[Jongejan *et al.*, 2016] Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. The quick, draw!-ai experiment. *Mount View, CA, accessed Feb*, 17(2018):4, 2016.

[Larsson *et al.*, 2020] Maria Larsson, Hironori Yoshida, Nobuyuki Umetani, and Takeo Igarashi. Tsugite: Interactive design and fabrication of wood joints. In *UIST*, pages 317–327, 2020.

[Nashed and Biswas, 2018] Samer Nashed and Joydeep Biswas. Human-in-the-loop slam. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Sakata *et al.*, 2019] Yusuke Sakata, Yukino Baba, and Hisashi Kashima. Crownn: Human-in-the-loop network with crowd-generated inputs. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7555–7559. IEEE, 2019.

[Wang *et al.*, 2014] Fangzhou Wang, Yang Li, Daisuke Sakamoto, and Takeo Igarashi. Hierarchical route maps for efficient navigation. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pages 169–178, 2014.

[Yang *et al.*, 2019] Xi Yang, Bojian Wu, Issei Sato, and Takeo Igarashi. Directing dnns attention for facial attribution classification using gradient-weighted class activation mapping. In *CVPR Workshops*, pages 103–106, 2019.