Performance Comparison of CatGAN using different Facial Expression Recognition Datasets

Junyeob Kim*, Marshall**, Jang-Sik Park**
*Dept of Engineering, University of Cambridge
**Dept of Electronics Engineering, Kyungsung University

Abstract

It is well known that the performance of Deep Neural Networks changes with different hyperparameters. However, there is not much information about the relationship between the dataset and the performance of the network. In this paper, we characterize two Facial Expression Recognition Datasets and show that there is difference in the performance of the Categorial Generative Adversarial Network between two datasets.

1. Introduction

With the development of deep learning networks, various applications, including object detection [1] and speech recognition [2], have been suggested. In order to train different networks to fit into their objectives, large amounts of datasets have been created. Since each dataset is generated in different environment, probability distributions of each dataset must be different. Therefore, the performance of deep learning models should change depending on the input datasets. In this paper, Facial Expression Recognition (FER) [3] datasets are used, as there are various datasets available. In this paper, using FER datasets, we show that the probability distributions of each FER dataset are different.

Moreover, within a network, there are several hyperparameters, such as activation function and optimizer. Changing these hyperparameters do not change the architecture of the networks, but it is possible to improve the performance of the network or make the training and validation process more effective. This paper makes use of Categorical Generative Adversarial Network (CatGAN) [4], a variation of Generative Adversarial Network (GAN) [5] for unsupervised and semi-supervised learning, to prove that the performance of the network differs depending on the input datasets. We also demonstrate which combination of hyperparameters are most suitable for the CatGAN, for the purpose of FER.

2. Hyperparameters

We first show that there is change in the learning process of the deep learning network with different hyperparameters. Glorot et al. [6] proved that using Rectified Linear Unit (ReLU) as activation functions [7] can improve the accuracy of Deep Neural Network (DNN) significantly compared to traditional non-linear functions. In addition to ReLU, Leaky ReLU [8] has been suggested to avoid potential problems caused with hard saturation of ReLU, and Clipped ReLU [9] is introduced to avoid activations from explosion. Equations of three activation functions are shown below.

$$ReLU(x) = \max(x,0) = \begin{cases} x & x > 0 \\ 0 & else \end{cases}$$

$$LReLU(x) = \max(x,0) = \begin{cases} x & x > 0 \\ kx & else \end{cases}$$

$$CReLU = \min[\max(x,0), y] = \begin{cases} x & x > 0 \\ 0 & else \end{cases}$$

where k is gradient, and y is a number to be clipped.

Other than activation functions, optimizers also affect the learning process of DNN significantly. The principle of the optimizer is to reach the local minimum of the objective function. Each optimizer approaches this problem in different way, resulting with the difference in the performance. This can be easily visualized using Rosenbrock [10] and Himmelblau [11] function.

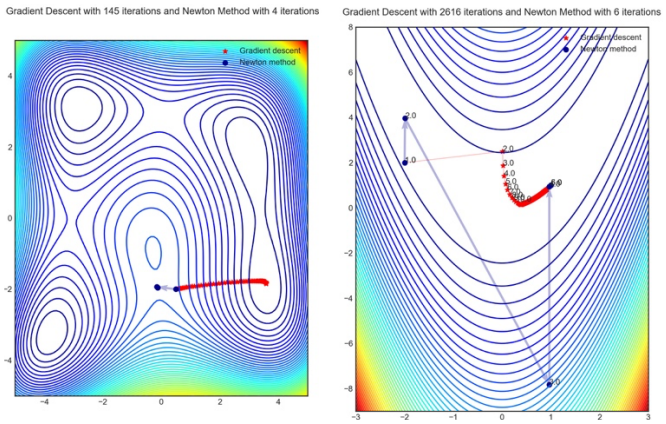

Figure 1. Gradient Descent (Red) and Newton Method (Blue) on Rosenbrock (Left) and Himmelblau (Right) Functions.

Figure 1 shows that for simple functions, Newton Method reaches to minimum point much faster than Gradient Descent but for complex functions with multiple saddle points, only gradient descent method reaches the local minimum point. Therefore, changing optimizer will affect the performance of DNN significantly.

3. Dataset Distribution

In order to prove that there are differences in probability distributions of datasets, T-SNE [12] has been applied to the datasets. Two FER datasets are used; MMI [13]; and KDEF [14]

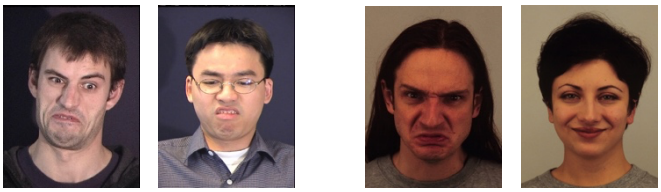

Figure 2. Sample images from MMI (left) and KDEF (right) datasets.

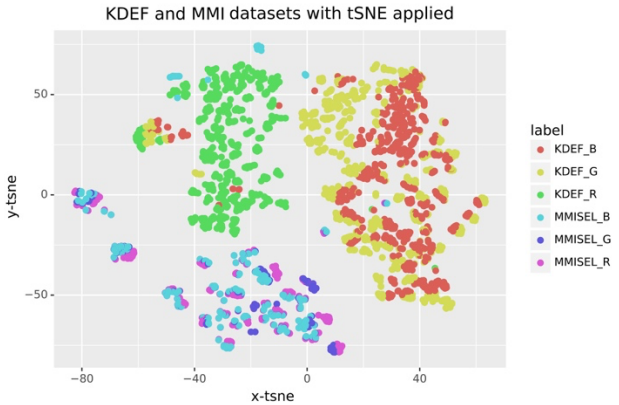

Figure 3. Result of T-SNE applied to MMI and KDEF datasets.

As shown in fig 2, two datasets are located in separate sections of the T-SNE map. This shows that two datasets have distinguishable features,

To show that these features represent the difference in the probability distribution of the datasets, distributions are plotted, with three channels of colors.

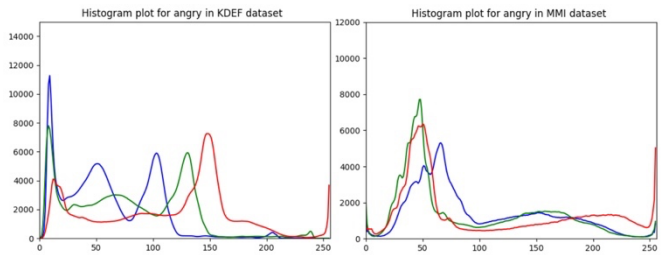

Figure 4. RGB distributions of KDEF (Left) and MMI (Right) for angry facial expression.

Fig 3 represents RGB distributions of two datasets, for same facial expression. It is possible to deduce that two datasets have different Mixture of Gaussian (MoG) distributions.

4. FER with CatGAN

The Network we use to compare the performance of each hyperparameter and datasets, is CatGAN [4]. As mentioned previously, this is a variation of GAN [5], which is a generative network. With the adversarial network, GAN improves the performance of the generative model by the minimax two player game, of Generator against Discriminator. On the other hand, the purpose of CatGAN is different. This network aims to improve the performance of the Discriminator, which is modified in order to perform classification tasks. In this paper, we used the exact architecture used in the paper [4] for the CIFAR-10 dataset, except that we applied batch normalization [15] for every convolution in the Generator as Radford et al. [16] proposed - Deep Convolution GAN (DCGAN).

For the experiment, we use MMI and KDEF as the input FER dataset. Activation functions and optimizers used are listed in table 1.

Table 1. Showing different hyperparameters used for the experiment

Optimizers	Adam [17]	RMSProp [18]	SGDM [19]
Activation Functions	ReLU	Leaky ReLU	Clipped ReLU

5. Simulation results and consideration

The settings for the experiment are shown in the table below. Google Colaboratory (CoLab) is used to run train and test the created network.

Table 2. Experiment settings

	MMI	KDEF
Train set	187	588
Test set	80	251
Input image dimension	60×45×3	
Batch size	45	
Epoch	50	
Input noise dimension	128	
Learning rate	0.0001	

Figure 4 shows the results of the experiment, comparing accuracy of the trained network with different hyperparameters. Clipped ReLU resulted with 100% accuracy, regardless of the optimizer and input dataset.

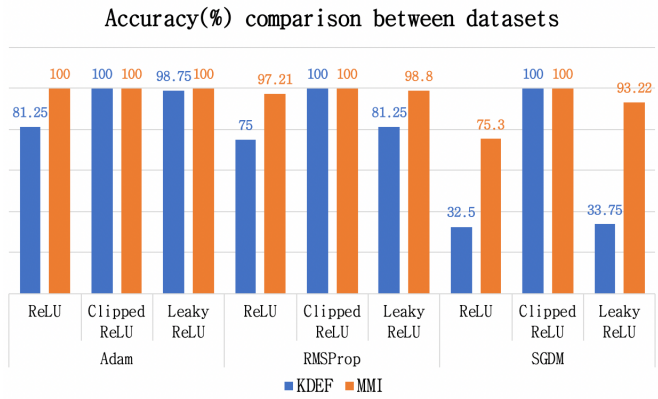

Figure 4. Accuracy comparison between differing hyperparameters

6. Conclusion

In this paper, we showed that there are some differences in the performance of CatGAN between MMI and KDEF dataset. We also demonstrated that two datasets have distinguishable features, that can be represented with MoGs.

The particular CatGAN used in this paper showed excellent performance with FER task. However, the generator did not perform as expected - generating image of people with facial expressions. This might be because the generated image is compressed with the combination of features from different faces with different facial expressions. Also, since the network used in this experiment is originally for CIFAR-10, it might not perform very well with the FER datasets.

Overall, understanding the differing nuances between datasets, and setting them as another hyperparameter for network training might improve the performance of the networks. As such, this topic is worth further studies.

References

[1] Z. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Trans Neural Netw Learn Syst*, Vol.30, no. 11, pp. 3212-3232, 2019.

[2] G. Hinton *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The. Shared Views of Four Research Groups," *IEEE Signal Process Mag*, Vol. 29, no. 6, pp. 82-97, 2012.

[3] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," *IEEE Transactions on Affective Comput.*

[4] J. T. Springenberg, "Unsupervised and Semi-Supervised Learning with Categorical Generative Adversarial Networks," *ICLR 2016*.

[5] I. J. Goodfellow et al, "Generative Adversarial Nets". *Adv Neural Inf Process Syst*, Vol. 2, pp. 2672-2680 2014.

[6] X. Glorot, A. Bordes, Y. Bengio, "Deep Sparse Rectifier Neural Networks," *AISTATS*, Vol. 15, pp. 315-323, 2011.

[7] V. Nair and G. E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines," *ICML*, pp. 807-814, 2010.

[8] A. L. Maas, A. Y. Hannun, A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *ICML*, Vol. 30, 2013.

[9] A. Hannun et al., "Deep Speech: Scaling up end-to-end speech recognition," *http://arxiv.org/abs/1412.5567*, 2014.

[10] H. H. Rosenbrock, "An Automatic Method for Finding the Greatest or Least Value of a Function," *The Computer Journal*, Vol. 3, no. 3, pp. 175-184, 1960.

[11] D. M. Himmelblau, "Applied Nonlinear Programming," *McGraw-Hill Book Company*, New York, New York, 1972.

[12] L. van der Maaten and G.E. Hinton, "Visualizing High-Dimensional Data Using t-SNE," *J. Machine Learning Research*, Vol. 9, pp. 2579-2605, 2008.

[13] M. Pantic, M. Valstar, R. Rademaker, L. Maat, "Web-based database for facial expression analysis," *Proc. 13th ACM Int'l Conf. Multimedia (Multimedia '05)*, pp. 317-321, 2005.

[14] M. G. Calvo, D. Lundqvist, "Facial expressions of emotion (KDEF): Identification under different display-duration conditions," *Behav Res*, Vol. 40, no. 1, pp. 109-115, 2008.

[15] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *arXiv preprint arXiv:1502.03167*, 2015.

[16] A. Radford, L. Metz, S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *ICLR 2016*.

[17] D. Kingma, J. Ba, "Adam: A method for stochastic optimization," *ICLR 2015*.

[18] T. Tieleman, G. Hinton, "Lecture 6.5-RmsProp: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, 2012.

[19] N. Qian, "On the momentum term in gradient descent learning algorithms," Neural Networks, Vol. 12, no. 1, pp. 145-151, 1999.