Math6380P Project2
Classification of Nexperia Image Dataset: An Averaging Ensemble Approach

ABUDLLAH,Murad(20673712)
mabdullah@connect.ust.hk

December 13, 2020

Abstract

This project aims to analyze the ensemble model on the Nexperia image dataset. By using the ensemble of DenseNet121 & ResNet34 we were able to secure 0.99866 *area under the ROC curve* on the test set. First, the networks were trained with different training parameters. Then their ensemble was used to generate pseudo labels for the test set. Overall performance was improved due to the network's training on test data set with pseudo labels combined with training data.

Remark on Contribution

Everything in the project was done by me.

1 Introduction

As the neural networks get deeper and deeper, it is challenging to train them due to the vanishing, and exploding gradient problem [1, 2]. The *Deep Residual Network* was proposed in [3] to address this problem. The ResNets were able to converge using *Stochastic Gradient Descent* with backpropagation on very deep layers. Fig. 1 shows the basic building block of the ResNet. *Dense Convolution Network (DenseNet)* was proposed in [4] to address the better information flow between different layers. Fig. 2 shows the 5-layer dense block. We can see the connectivity pattern in this block all the preceding layers are added to subsequent layers, because of this dense connectivity patterns these are called *DenseNet*.

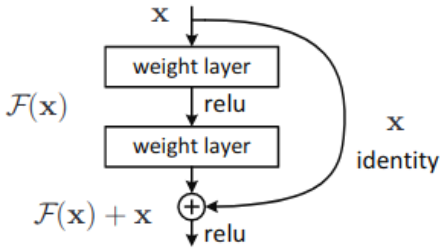

Figure 1: Basic building block of the ResNets [3]

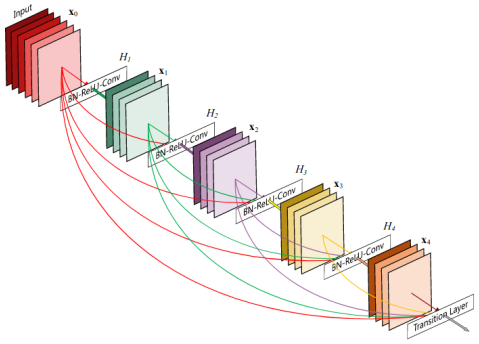

Figure 2: 5 layer DenseNet [4]

2 Dataset

I participated in the in-class competition on Kaggle. So, I used the Nexperia image dataset. The *left* image in Fig. 3 shows some example images from the dataset. The green rectangle shows the defective part of the semiconductor. The *right* image shows the training data is highly imbalance. The defective chips are approximately 1/4 of the overall training dataset.

3 Problem Description

As we have analyzed some images from the dataset in Fig. 3. Our problem here is to predict the defect score for the images. My goal is to analyze how one can use the ensemble of different models to create pseudo labels for test data to improve the ensemble's overall performance. The effectiveness of the pseudo labels technique is to be analyzed.

4 Our Ensemble Strategy

We used the pre-trained ResNet34 and DenseNet121 for our training pipeline. Following are the steps of our training Strategy

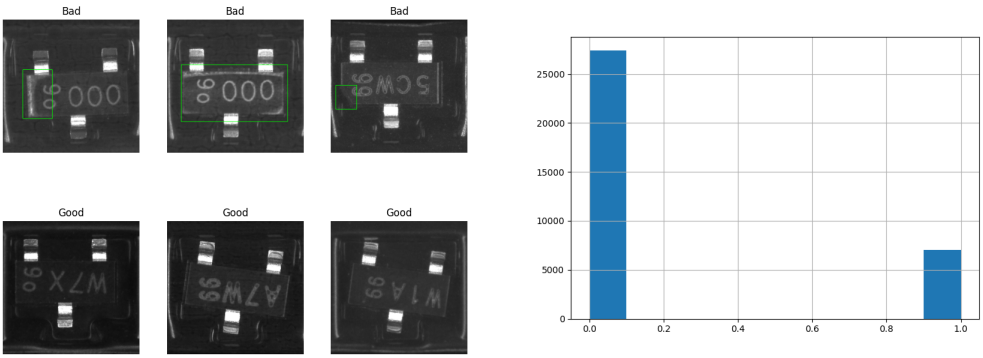

Figure 3: Visualization of the dataset (*left*) Some images from data set. (*right*) The imbalance of training data

- We fine-tuned the last layer of both networks DenseNet121 & ResNet34.
- Then we created the pseudo label for the test data set by the ensemble of the ResNet34 & DenseNet121.
- The ResNet34 was finally fine-tuned with data by combing the train data and test data with pseudo labels created in the previous step.
- The submission was created with the ensemble of ResNet34 of the previous step and the ResNet34 trained in first step.

Fig. 4 shows the above steps graphically, P1, P2 & P3 are pseudo labels for test set and Avg is averaging operation.

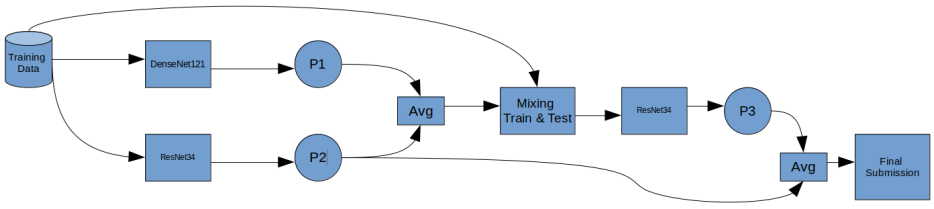

Figure 4: Illustration of training technique.

5 Challenge from Project 1

5.1 Visualization of Features by t-SNE

t-distributed stochastic neighbor embedding (t-SNE) is a tool to visualize the high-dimensional data [5]. It converts the similarities between data points and tries to minimize the KL divergence between the joint probabilities of low dimensional embeddings and high dimensional data. Fig. 5 shows the t-SNE visualization of the features extracted by pre-trained ResNet34 as we can see that both the classes are highly overlapped. It is tough for a linear classifier to classify both classes using these features.

5.2 Comparison of ResNet34 & Scatter Net

Table 1 shows the NC1, equal-norms of class-mean, equal angularity and maximal-angle equiangularity computed from features extracted from ResNet34 and Scatter Net, respectively. The neural collapse is only different in both the networks; everything is nearly equal. Class-means, total covariance matrix, and within-class covariance, etc., can be found in the provided jupyter notebook.

Figure 5: t-SNE visualization of features extracted by pre-trained ResNet34.

Table 1: Comparison of ResNet34 & Scatter Net

Statistics	ResNet34	Scatter Net
NC1	11.32641	2.361179e-05
Equal-Norms of class-mean	0.3274342	0.32743422
Equal-angularity	0.9999999	0.9999999
Closeness to maximal-angle equiangularity	0.9999999	1.0

6 Experiments

I demonstrated the effectiveness of the ensemble of the ResNet34 & DenseNet121 for generating the pseudo labels for the test data to improve the overall ensemble's performance on the test dataset. The 80% of the training data was used for training and 20% for validation data.

6.1 Training

Both networks are trained using stochastic gradient descent (SGD). The DenseNet121 was trained on the batch size of 48 for 40 epochs. Fig. 7 shows the loss and AUC curve for the training of the DenseNet121. The ResNet34 was trained on the batch size of 32 for 40 epochs. The learning rate for both the networks was initially set to 1e-3 and lowered by 10 every 3rd epoch. I used the weight decay of 5e-4 with a Nesterov momentum of 0.9. Fig. 1 shows the training and validation loss curves for fine-tuning of the ResNet34. The final ResNet34 trained on the combined data set was trained for 10 epochs with a batch size of 32. The SGD was also used for the training of this network with the same learning rate and weight decay.

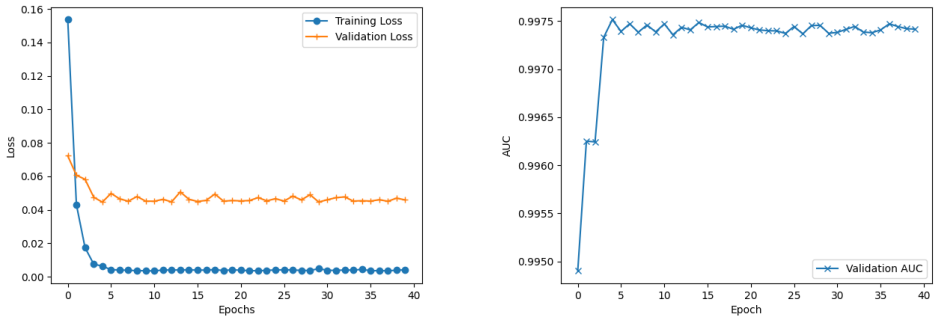

Figure 6: Loss and AUC curves (*left*) Loss curve. (*right*) AUC curve

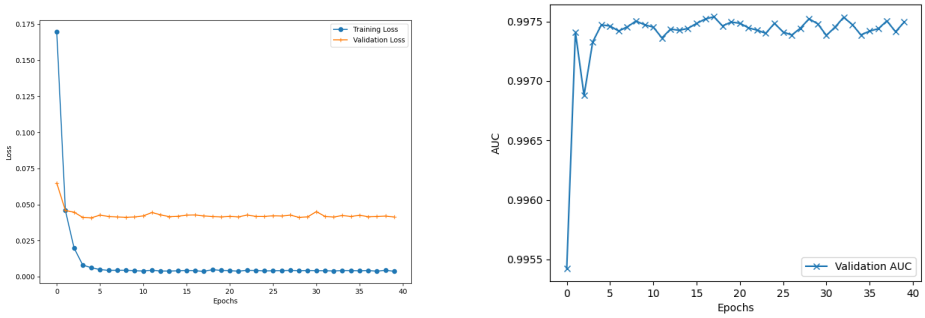

Figure 7: Loss and AUC curves (*left*) Loss curve. (*right*) AUC curve

7 Discussion

In this project, I used the pre-trained networks and fine-tuned them on Nexperia image data set. It was observed that the pseudo labels generated for the test data set could help to improve the *area under ROC curve* on the test data by mixing the test data set and training data set. As the labels generated for the test set are noisy, now the network has seen the test set with noisy labels, which may not accurately predict the test data set. The loss curves of the DenseNet121 in Fig. 6 are very smooth as compared to the ResNet34 loss curves in Fig. 7. This smoothness of DenseNet curves can be explained by the greater batch size and connectivity pattern of DenseNets, responsible for better information flow between the layers.

References

[1] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[2] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[4] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[5] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 3, pp. 1249–1268, 2016.