Explore the impact of using regularization methods on Overfitting CNN models

Hengyi Ma
a1875198
The University Of Adelaide

Abstract

CNN model is one of the popular models nowadays and is widely used in various fields of deep learning, especially in the field of CV. In recent years, with the explosion of research on CNN, more and more new structures and new models have been proposed, and these new models have shown excellent performance in many fields. Such as Inception and Mobilenet proposed by Google, etc. In this report, we choose a novel CNN structure proposed in recent years– the ghostnet, and apply it to the specified dataset for image classification prediction task. The model used shows overfitting during training. Based on this, we will introduce the commonly used regularization techniques dropout and L2, and explore the impact of using these regularization methods on the overfitting CNN models.

The code of this report can be accessed at: `https://github.com/HengyiMa/deeplearning-demo/tree/main/%E5%B9%BD%E7%81%B5%E7%BD%91%E7%BB%9C_%E6%AD%A3%E5%88%99%E5%8C%96%E6%96%B9%E6%B3%95%E7%A0%94%E7%A9%B6`

Keywords: image classification task, regularization technology, CNN, overfitting

1. Introduction

CNN: Neural network is a major achievement of mankind in the field of bionics. In the field of artificial intelligence, by simulating the neuron-synapse architecture in biological neural networks, Rosenblatt [8] proposed a neural network in 1957, and Rumelhart proposed a backpropagation algorithm [9] in 1986. These are the cornerstones of neural networks. With the continuous improvement of related work, Kunihiko Fukushima and Yann LeCun [7] formally proposed the concept of convolutional neural network.

What distinguishes convolutional neural networks from other neural networks is their superior performance on image, speech, or audio signal inputs. Before the advent of CNNs, manual, time-consuming feature extraction methods were used to identify objects in images. However, convolutional neural networks now offer a more scalable approach to image classification and object recognition tasks, leveraging the principles of linear algebra (in particular matrix multiplication) to identify patterns in images. The basic structure of a typical CNN is shown in the Figure 1:

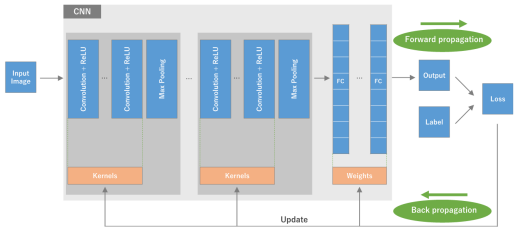

Figure 1: A typical structure of CNN

Ghostnet: The ghostnet is an improved CNN structure published by Kai Han, Yunhe Wang, Qi Tian[1] and others at CVPR in 2020. This paper designs a new structure to accelerate CNN training based on concepts such as residual connections. This structure divides the traditional convolution process into two parts, one of which uses a more concise calculation method to improve the utilization of features. Rate, this module can be used to replace traditional modules in CNN to design lightweight structures, while its effect is no worse than the best result at the time. They further designed a targeted structure G-ghostnet [2] in 2022. G-ghostnet can be used for multi-classification tasks of images. The test of this article will also be based on G-ghostnet. The structural comparison of traditional CNN structure, ghostnet and G-ghostnet is shown in the Figure 2:

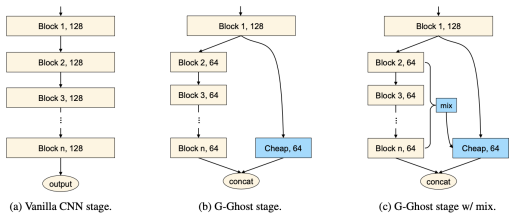

Figure 2: A comparison between CNN, ghostnet and G-ghostnet

2. Related work

Overfitting: Overfitting is a common problem in machine learning and deep learning, which refers to a situation where a model performs very well on training data but performs poorly on new, unseen data. Overfitting occurs when the model learns the noise and subtle features of the training data without capturing the true underlying patterns of the data.

There are many reasons that may lead to overfitting: an overly complex model with a large number of parameters or capacity is prone to overfitting; when the amount of training data is small, the model is more likely to remember the characteristics of each sample instead of learning the data general pattern. This can lead to model overfitting; too many training cycles or iterations can lead to overfitting. The model continues to optimize on the training data, but the generalization performance decreases.

Regularization: Regularization is a means to deal with overfitting of the model. The goal of regularization is to limit the complexity of the model to prevent it from overfitting the training data. There are currently many regularization ideas to prevent and alleviate overfitting, such as by adding additional terms (regularization terms) to the loss function of the model. To achieve regularization, these terms penalize the complexity of the model and encourage the parameters of the model to maintain small values, such as L1 regularization and L2 regularization.

In this article, we mainly study L2 regularization [4]. L2 regularization is achieved by adding the sum of squares of the model parameters to the loss function, that is, the regularization term is the L2 norm of the parameters. The effect of L2 regularization is to encourage the parameters of the model to keep small values, but not to compress them to zero. This helps reduce the differences between parameters, improves the smoothness of the model, and reduces overreliance on a single feature. The mathematical expression of L2 regularization is as follows:

$$LossFunction = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y} - Y)^2 + \lambda \sum_{i=1}^{N} \theta_i^2$$

Dropout:Dropout [5], like L2 regularization, is a regularization method used to alleviate model overfitting, but the working principles of the two are different. The core idea of Dropout is to randomly turn off a part of neurons during the training process, thereby preventing the neurons from being too dependent on certain specific inputs, making the network more robust and improving generalization capabilities. The impact of dropout on the model is as follows in Figure 3:

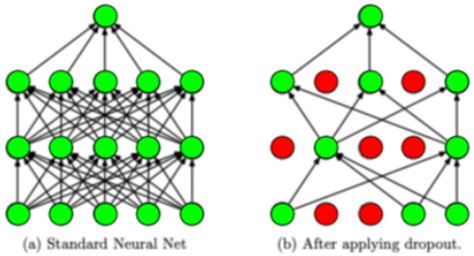

Figure 3: The way dropout works in a neural network

3. Proposed method

For this experiment, we chose the dataset CIFAR-10, which is a dataset often used in CV image multi-classification problems. The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. [6]

The code given by the author of G-ghostnet uses an improved regnet model and provides models at different complexity to adapt to different datasets. Here, different complexity means the number of modules of the model/the number of channels in the convolution process, each module contains a complete input-convolution-output combination, which can be regarded as a CNN module. We divide the experiment into two parts: pre-experiment and main experiment.

In the pre-experiment, we tested three groups of ghostnet structures with different complexities.

The four parts of the first group include 1, 3, 7, and 5 basic modules respectively, and the number of calculated channels in each part is 64, 128, 288, and 672.

The four parts of the second group include 2, 6, 15, and 2 basic modules respectively, and the number of calculation channels in each part is 96, 192, 432, and 1008.

The four parts of the third group include 2, 4, 10, and 1 respectively, and the calculated number of channels in each part is 168, 392, 784, and 1624.

We will select a more efficient and severely overfitting model from these three groups of models as our subsequent modified baseline model in order to study the impact of regularization on model performance. In order to ensure fairness, all models use the same data partitioning and processing methods, use the same learning rate of 0.01, and run for 30 epochs.

On top of this, we will group experiments on the selected models: The first group will use dropout as an optimization method, using four different parameters of 0.65, 0.75, 0.85, and 0.95 to longitudinally compare the impact of dropout on the model.

The second group will use L2 regularization as an opti-

mization method, and use four different parameters of 0.01, 0.1, 0.15, and 0.2 to longitudinally compare the impact of L2 regularization strength on the model.

The third group will combine the two regularization methods of dropout and L2, and compare the combination of the two methods with any single method, as well as the comparison between baselines.

In order to ensure the fairness of the three groups of experiments and effective comparison with the baseline, we will continue the experiment using the parameters inherited from the pre-experiment, such as the learning rate, data processing method, and epoch are all set to 20. At the same time, we will also design an epoch of 20 baseline model for fair comparison.

4. Experiment analysis

First, we select an object suitable for subsequent experiments from models of different complexity. We will analyze the changes in three sets of experimental loss curves and accuracy curves.

The first set of experimental models, the experimental conditions of g_ghost_regnetx_008 are shown in the Figure 4: Test Accuracy: 40.25%

Figure 4: Result for g_ghost_regnetx_008 model

The second set of experimental models, the experimental results of g_ghost_regnetx_032 are as shown in the Figure 5: Test Accuracy: 39.75%

Figure 5: Result for g_ghost_regnetx_032 model

The third set of experimental models, the experimental results of g_ghost_regnetx_064 are as shown in the Figure 6: Test Accuracy: 42.75%

Figure 6: Result for g_ghost_regnetx_064 model

From the analysis of the three sets of loss curves and accuracy curves, it can be seen that: a. the three sets of models have serious overfitting; b. as the complexity of the model deepens, overfitting occurs earlier, the second and third sets Overfitting appears approximately after the 5th epochs, but overfitting in the first group appears around the 10th epochs; c. Overfitting in the second and third groups is more serious, and at the end of the experiment, the validation loss of both Larger than the first group; d. Although all three models have serious overfitting and the validation accuracy is not very good, judging from the training set accuracy and training set loss, the more complex the model is, the better it is to learn the training set data The faster the speed, the better the learning effect. The third group can even achieve an accuracy of nearly 100% of the training set data, which is higher than the data of the second group of nearly 90% and the first group of nearly 80%. From the loss curve, The third group of models has the lowest training set loss at the end.

Combining the analysis of the above three models, we finally selected the third group of experimental models with the highest complexity, g_ghost_regnetx_064, as our model because it reflects the best learning ability and severe overfitting, which helps us explore and alleviate the impact of relieving overfitting problem on model performance.

In view of the fact that the third group of models has caused serious overfitting and equipment limitations, our next experiments are unified to 20 epochs. Based on this, first we need to retrain a baseline model of 20 epochs. The effect is as shown in the Figure 7: Test Accuracy: 42.00

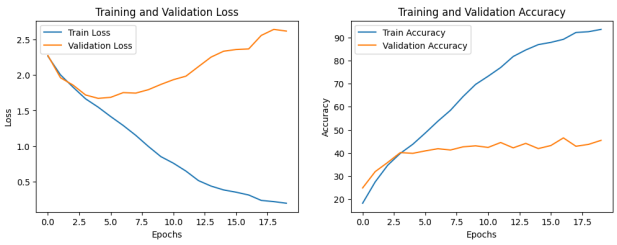

Figure 7: Result for baseline model

Next, we studied the results obtained by using dropout as

an optimization method and adjusting the model using four different parameters of 0.65, 0.75, 0.85, and 0.95.

The first set of experiments uses dropout=0.65, which means there is a 65% probability of discarding neurons in training, the effect is as shown in the Figure 8: Test Accuracy: 41.00%

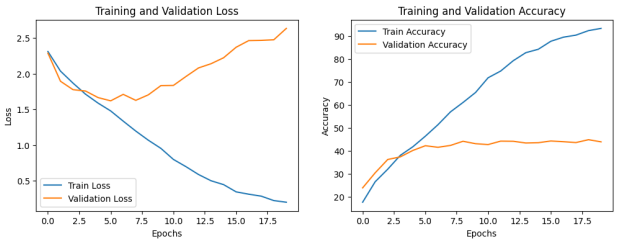

Figure 8: Result for dropout=0.65 model

The second set of experiments uses dropout=0.75, which means there is a 75% probability of discarding neurons in training, the effect is as shown in the Figure 9: Test Accuracy: 43.75%

Figure 9: Result for dropout=0.75 model

The third set of experiments uses dropout=0.85, which means there is an 85% probability of discarding neurons in training, the effect is as shown in the Figure 10: Test Accuracy: 45.50%

Figure 10: Result for dropout=0.85 model

The fourth set of experiments uses dropout=0.95, which means there is a 95% probability of discarding neurons in training, the effect is as shown in the Figure 11: Test Accuracy: 31.25%

Judging from the test accuracy, the accuracy of the first and fourth groups is not as good as the baseline, and the

Figure 11: Result for dropout=0.95 model

accuracy of the second and third groups is higher than the baseline. Comparing the first three groups of experiments, it can be found that as dropout increases, the curve of the validation loss becomes smoother and smoother, and the inflection point also moves to the right, indicating that appropriate enhancement of dropout can help alleviate and delay the over-fitting problem. However, the training loss curve also gradually slows down, indicating that the learning ability of the model has declined. Comparing the training accuracy of the first three groups of models, it was found that the slope of the training accuracy curve also gradually slowed down, further confirming that dropout will inhibit the model's powerful ability to learn data. However, from the fourth set of experiments, it can be found that excessive dropout will lead to catastrophic results in model learning. Even if the model has been highly overfitting, we can see that the loss curve and accuracy curve have no rules at all, and the model cannot learn any data. , Accordingly, based on the occurrence of over-fitting opportunities and the accuracy of the model, we selected the third set of experiments with dropout=0.85 and test accuracy: 45.50% as the best set of results.

Then, we study the results obtained by using L2 as an optimization method and adjusting the model with four different parameters of L2=0.01, 0.1, 0.15, and 0.2: The first set of experiments uses L2=0.01, the effect is as shown in the Figure 12: Test Accuracy: 39.25%

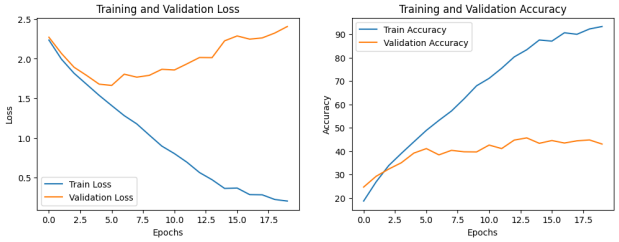

Figure 12: Result for L2=0.01 model

The second set of experiments uses L2=0.1, the effect is as shown in the Figure 13: Test Accuracy: 42.75%

The third set of experiments uses L2=0.15, the effect is as shown in the Figure 14: Test Accuracy: 47.50%

Figure 13: Result for L2=0.1 model

Figure 14: Result for L2=0.15 model

The fourth set of experiments uses L2=0.2,the effect is as shown in the Figure 15: Test Accuracy: 36.00%

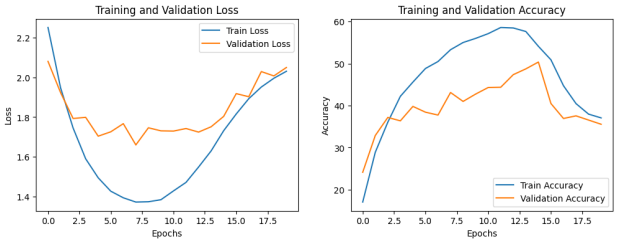

Figure 15: Result for L2=0.2 model

First, we compare the baseline model and four experimental results. On the loss curve, as the intensity of L2 regularization increases, it can be found that the slope of the training set loss becomes more and more gentle, and even first decreases and then increases, indicating that L2 regularization has a great impact on the model's ability to learn data. . At the same time, it can be seen that the validation loss curve in the second and third sets of experiments is easily suppressed in a gentle interval, indicating that L2 regularization has a strong constraint on validation loss. Observing the accuracy curve, we can see that with the strengthening of L2 regularization, the accuracy growth of the training set will be quickly suppressed, falling from 90% to 70%, and finally to 60%, but in the intermediate stage, the accuracy of the verification set and the test set There will be a significant improvement, indicating that the effect of L2 regularization is significant. We choose the experiment with L2=0.15 as the best experimental result, then Test Accuracy: 47.50%.

Comparing two different regularization methods, we can also observe some differences. Although the effects of L2 regularization and dropout regularization will first get better and then get worse as the parameters increase, the specific details are different. First of all, L2 regularization on the loss image can easily suppress the increase in validation loss when the regularization intensity is large, which is something dropout cannot do. Secondly, when the regularization intensity is too strong, dropout regularization will cause the model to be unable to learn new content, but L2 regularization will cause the model to learn the content first and then forget the content, causing the loss curve to increase instead. We can see that the regularization intensity of dropout will not affect the learning upper limit of the model, but will only affect the learning speed, unless it is extreme. However, the intensity of L2 regularization will directly affect the learning upper limit of the model.

Finally, we try to combine the best experimental results under two different regularization situations to see whether a better model will be obtained. We try to run baseline+dropout=0.85+L2=0.15 and select the best epoch=14 to stop. The experimental results are as follows in Figure 16: Test Accuracy: 49.75%

Figure 16: Result for baseline+dropout=0.85+L2=0.15 model

It can be found that the experimental combination of baseline+dropout=0.85+L2=0.15 is better than the baseline and any single best result, and the image trend is closer to the L2 regularized image.

5. Conclusion

In this article, I selected a lightweight model in the CV image classification task of CNN structure in recent years to explore the impact of dropout and L2 two different regularization methods on the performance of the overfitting model. By training the model under different strengths and combinations of regularization parameters, I analyzed the results of the experiment and made several interesting points based on the differences in principles between dropout and L2 regularization.

1. Dropout regularization simplifies the model by turning off neurons. In fact, it only slows down the speed of

model overfitting, forcing the model to only learn a part of the content at a time. This delay ability is positively related to the strength of dropout within a certain range. At the same time, dropout will not only slow down the speed of model overfitting, but also cause the model's learning speed to decrease.

2. L2 regularization simplifies the model by reducing the weights. This regularization method is more powerful than dropout and can effectively suppress the overfitting of the model. This can be observed through the experimental validation loss. But at the same time, L2 regularization may lead to poorer model performance, because multiple epoch runs during training will inherit the previous parameters, causing this regularization effect to continue to amplify, and even cause the model's learning ability to decline instead of just slowing down learning speed.

3. Based on theoretical analysis and experimental testing, I believe that these two regularization methods can be superimposed on each other because they involve using different methods to constrain model complexity. Experiments also support this conclusion. At the same time, there are many tools with similar effects in daily life, and superimposing each other may not necessarily produce better results.

4. Even if regularization can effectively improve the performance of the model, its effect is still limited. There is still a lot of room for optimization between the accuracy of the training set and the accuracy of the verification set. I think the following points may help to further improve the model performance, this will also become my subsequent research direction:

a. Use a larger data set: For an overly complex model, even if regularization can simplify the model, over-regularization will cause the model to be unable to learn the data. Therefore, on the premise of proper application of regularization, enlarging the data set can be used to guide the diversification of model parameters and learn more general features.

b. Simplified model: Observation of pre-experiments shows that although g_ghost_regnetx_008 does not show the best training set effect, overfitting is relatively slight. We can choose simpler models for training because their overfitting problem is not that serious.

c. Introducing residual connections to the overall scale: According to the idea of Resnet [3], we can introduce overall residual connections. Although the CNN architecture used in this article contains residual connections, these residual connections exist within the block. From the overall perspective of the model, it is still easy to be highly overfitting. We can introduce residual connections on a larger scale to alleviate the problem of overfitting.

References

[1] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1580–1589, 2020.

[2] Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chunjing Xu, Enhua Wu, and Qi Tian. Ghostnets on heterogeneous devices via cheap operations. *International Journal of Computer Vision*, 130(4):1050–1069, 2022.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Shubham Jain. An overview of regularization techniques in deep learning (with python code). *Analytics Vidhya*, 19, 2018.

[5] Salman H Khan, Munawar Hayat, and Fatih Porikli. Regularization of deep neural networks with spectral dropout. *Neural Networks*, 110:82–90, 2019.

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

[7] 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.

[8] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[9] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.