University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

A Review of the Overfitting Problem in Convolution Neural Network and Remedy Approaches

Hiba Kahdum Dishar[a], Lamia AbedNoor Muhammed[b]

[a]AL-Qadisiyah University, College of Computer Science and Information Technology, Computer Science Department, Diwaniyah, Iraq. com21.post13@qu.edu.iq

[b]Al-Qadisiyah University, College of Computer Science and Information Technology, Computer Science Department, Diwaniyah, Iraq. lamia.abed@qu.edu.iq

ARTICLE INFO

Article history:
Received: 25 /04/2023
Revised form: 10 /06/2023
Accepted : 18 /06/2023
Available online: 30 /06/2023

Keywords:

Convolutional Neural Network,

Overfitting Problem,

Deep learning,

Regularization strategy,

Data Expansion

ABSTRACT

Deep learning methods have attracted much attention over the past few years after their breakthroughs in speech recognition and computer vision. However, Convolution Neural Network (CNN) is one of the most important networks that have been used especially in image classification, these networks are facing an essential problem which is the "overfitting problem". This problem means that the difficulty implies that the model would just "memorize" previously observed patterns, rather than "learn" the important patterns. As a result, it will down the classification performance and become a huge problem. Different remedy approaches have been suggested, each one can exhibit different behavior in the reduction of the overfitting problem according to the nature of the training data. So, this variety of remedy approaches needs to be examined. This paper would be focused on the factors that can cause the overfitting problem, then continue into the approaches to solving this problem.

https://doi.org/10.29304/jqcm.2023.15.2.1240

1. Introduction

With the rapidly growing demand for Artificial Neural Networks (ANNs) to solve many complex problems, CNN as a class of deep learning techniques has evolved as an area of interest to researchers in the past few years. CNN has evolved in recent years into a method for producing encouraging results in a variety of machine-learning applications, including image classification, object detection, face detection, speech recognition, vehicle recognition, facial expression recognition, and text recognition, among others. Deep learning neural networks are a field that can be used for a variety of classification and recognition applications, CNN faces a fundamental problem which is an overfitting problem. When there is an excessive gap between the error rate during training and the error rate during testing, this

∗Corresponding author: Hiba Kahdum Dishar

Email addresses: com21.post13@qu.edu.iq

Communicated by 'sub etitor'

is an indication of overfitting. Although the model does well in the training dataset, it does poorly in the test dataset since the model's complexity is larger than the complexity of the actual situation [1]. Overfitting happens when a model is trained in a manner that is excessively complex, with the goal of producing an estimation that has a large variance but a low bias [2]. It is evident that overfitting is not a unique event and only signifies that the weights are being updated and that the gradient is not zero [3]. Next-generation networks are designed to be data-driven, however, they experience unpredictability because of changing user group habits and the disparate infrastructures that support these systems. In the meantime, the volume of data collected by the computer system continues to increase. A highly significant problem is how to recognize huge data and handle it to reduce network data transmission. Deep learning operations must be used to provide solutions to these and similar difficulties. However, deep learning confronts issues such as overfitting, which can reduce the effectiveness of its applications for tackling various network challenges [1].

Our main contributions are as follows:

1. A brief introduction to convolutional neural networks (CNNs) and their role in machine learning.

2. An overview of the overfitting problem in CNNs, including its causes and consequences. This could involve a discussion of the bias-variance tradeoff, which is a key concept in understanding overfitting.

3. A summary of the various approaches that have been proposed to address the overfitting problem in CNNs. This could include regularization techniques such as dropout and weight decay, as well as data augmentation methods like random cropping and flipping.

4. A comparison of the different approaches, highlighting their strengths and weaknesses, and discussing the scenarios in which they are most effective.

5. An exploration of recent developments and emerging trends in the field, such as the use of adversarial training to combat overfitting, or the application of transfer learning to reduce the need for large amounts of training data.

This paper is organized as a following: section I introductions of CNN and the overfitting problem in this network.

The detail of related works of overfitting problem in CNN that it's presented in section II. Section III explains CNN. Finally, section IV remedy approaches to the overfitting problem.

2. Literature survey

Recently, research on the overfitting issues have been conducted by numerous specialists and researchers. Several previous works have pointed out this overfitting problem issue in CNN. There are several methods to avoid this problem such as regularization which is critical for neural networks, especially for models with a massive number of parameters. Dropout is one approach to solving this issue. Throughout education, it is crucial to randomly remove units from the neural network (along with their connections). Units are prevented from coadapting too much as a result. samples of dropout rates from an increasingly large number of various "thinned" networks [4],

Nitish Srivastava et al. [5], dropout is a technique for addressing overfitting problem. The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different "thinned" networks. At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights. This significantly reduces overfitting and gives major improvements over other regularization methods. they show that dropout improves the performance of neural networks on supervised learning tasks in vision, speech recognition, document classification and computational biology, obtaining state-of-the-art results on many benchmark data sets. they applied dropout to a speech recognition task. they use the TIMIT data set which consists of recordings from 680 speakers covering 8 major dialects of American English reading ten phonetically-rich sentences in a controlled noise-free environment. Dropout neural networks were trained on windows of 21 log-filter bank frames to predict the label of the central frame.

Li Wan et al. [6], propose DropConnect which generalizes Dropout by randomly dropping the weights rather than the activations. Like Dropout, the technique is suitable for fully connected layers only. they compare and contrast the two methods on four different image datasets, we trained a model with 12 networks with DropConnect and achieved a state-of-the-art result of 9.32%, indicating the power of their approach.

Hui Zhu et al. [7], suggest the use of the attention mechanism to drop the discriminative feature units in a targeted regularization technique called TargetDrop. In particular, it conceals the feature maps' target channels' matching target regions. The regularization effect of their strategy is demonstrated by experimental results compared to those from methods used for various networks or other networks. Two primary components make up the TargetDrop experiments: comparing the regularization effect with other cutting-edge dropout-based approaches for ResNet-18 and demonstrating the performance for various topologies.

Minghui Liu et al. [3], utilize a straightforward yet efficient way to look for features linked to the objective, keeping those traits while discarding others, in contrast to the current methods. They discover that by sharpening the network's attention on its aim, this unique technique can enhance network performance. Additionally, Focused Dropout’s ability to prevent overfitting and improve accuracy can be enhanced by boosting weight decay. The results of the experiments demonstrate that, for a small additional cost, 10% of batches using FocusedDropout can significantly improve performance compared to baselines on a variety of classification datasets, including CIFAR10, CIFAR100, and Tiny ImageNet, and is adaptable to a variety of CNN models.

Zhihao Ouyang et al. [8], suggest AttentionDrop, a unique feature-dropping dropout variant based on attention data. According to the values of the activation units, it specifically localizes irregularly shaped masks. Additionally, using soft values in adaptive masks reduces the possibility of a total loss of crucial information. Experimental results demonstrate the effectiveness of their AttentionDrop on public datasets for image classification, CIFAR-10 and CIFAR-100.

Salman et al. [9], proposed a consensus-based classification model that avoids overfitting and significantly improves classification accuracy, especially when the amount of training data is restricted. The suggested approach achieved 95% accuracy with 90% of the test samples categorized, which is significantly higher than any of the individual models.

Assiri et al. [10] introduced a consensus-based classification technique that prevents overfitting and considerably enhances classification accuracy, particularly once the number of training data is limited. With 90% of the test samples categorized, the suggested algorithm obtained 95% accuracy, much higher than any of the other specific models. The highest performance was reached by testing and evaluating these strategies by applying models to five well-known datasets: MNIST, CIFAR10, CIFAR100, SVHN, and STL10.

Wei et al. [11] developed two similar neural networks using similar content and attempted to establish a correlation between two distinct labels. They used the JoCoR technique to achieve +1.28 enhancements over the best baseline approach and a remarkable +5.11 accuracy has increased compared to the norm.

Table1: summery of literature survey

References	Years	Method	Dataset	Accuracy	Error
Nitish Srivastava et al. [5]	2014	dropout	MNIST, SVHN, CIFAR-10 and CIFAR-100	_	0.79, 2.47, 11.68 and 37.20
Li Wan et al. [6]	2013	dropconnect	MNIST dataset and CIFAR-10 dataset	_	1.35 and 18.7
Hui Zhu et al. [7]	2022	Cutout +TargetDrop	CIFAR-10 and CIFAR-100	_	3.67 and 21.25

Minghui Liu et al. [3]	2022	Focuseddropout	CIFAR10 , CIFAR100 Tiny ImageNet	81.90 ± 0.11, 74.97 ± 0.16 64.58 ± 0.16	—
Zhihao Ouyang et al. [8]	2019	Attentiondrop	CIFAR10, CIFAR100 dataset	96.37±0.09 78.60±0.16	—
Salman et al. [9]	2019	consensus-based overfitting avoidance algorithm	MNIST dataset	95%	—
Assiri et al. [10]	2020	Stochastic Optimization of Plain Convolutional Neural Networks	MNIST, IFAR10, CIFAR100, SVHN and STL10	99.83, 94.29, 72.96, 98.50 and 88.08	0.17, 5.71, 27.04, 1.50 and 11.92
Wei et al. [11]	2020	(JoCoR) Method withCo-Regularization	MNIST, CIFAR-10 and CIFAR-100	98.06 ± 0.04, 85.73 ± 0.19 53.01 ± 0.04	—

3. Convolutional neural network

Convolutional Neural Network (CNN), as a specialized form of Artificial Neural Network (ANN), consists of a stack of trained convolution filters that extract hierarchical contextual picture characteristics. CNN is a common type of deep learning networks [12]. It already outperforms competing algorithms in a variety of fields, including digit recognition and natural image categorization [13] Convolutional layer, pooling layer, and fully connected layer are some of the various building parts (sometimes referred to as layers) that make up CNN. The most crucial element of any CNN architecture is the convolutional layer. It has a number of convolutional kernels, also known as filters, that are convolved with the N-dimensional metrics of the input image to produce an output feature map. One way to conceptualize a kernel is as a matrix of numbers or separate values, where each value or number indicates the kernel's weight. All the weights of a kernel are given random numbers when the CNN model's training phase begins [14]. The feature maps are sub-sampled using pooling layers (generated after convolution operation). By establishing the scope of the pooled zone and the operation's stride, the pooling approach is carried out. The added pooling layer to CNN is used instead of multiplying the filter with image pixels [15]. Every CNN architecture uses its final layer (or layers) for categorization. An output layer with complete connectivity is part of the CNN design. This shows a connection between each neuron in this layer and every neuron in the outputs of the layer under it. Additionally, the fully connected layer that came after it is utilized as the classifier can be seen in Figure 1 [16].

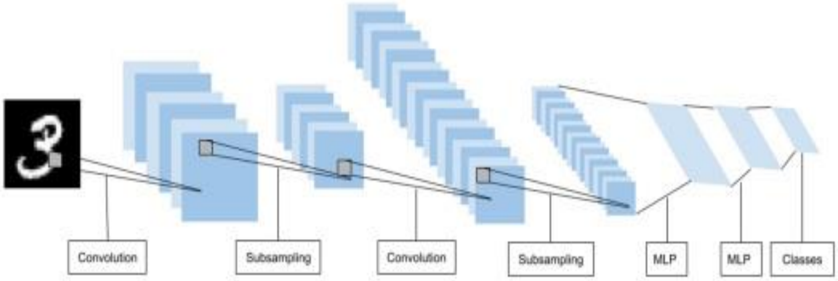

Fig .1 CNN architecture model[16].

Nowadays, the abundance of photos and videos on the internet is stimulating the creation of semantic analysis-focused search tools and algorithms[15]. CNN's introduction of a useful class of models for understanding the contents of an image has enabled better image classification, segmentation, detection, and retrieval[16]. A unique variety of multi-layer neural networks, known as CNN or ConvNet is inspired by the functioning of the visual system of living organisms [17]. CNNs are employed successfully in a variety of pattern and image recognition applications. Machine learning technique is a key component of object recognition strategies [17].

4. Overfitting problem

Machine learning systems that use deep neural networks with many parameters are especially useful. Overfitting is a significant issue in these networks, though. Big networks are very slow to utilize, merging the forecasts of numerous different large neural nets during testing, overfitting is made more challenging to handle [4]. Overfitting as shown in Figure 2. is a significant issue in supervised machine learning that inhibits models from correctly generalizing to accurately match observed data on training data. Due to the presence of noise, the small size of the training set, the complexity of the classifiers, as well as unknown data on the test set, overfitting arises[14]. Despite the fact that there are numerous picture recognition algorithms based on CNNs, the recognition effectiveness is not excellent. Today, Several recognition algorithms use a particular database to determine the depth and scale of the network. The most effective parameters and optimization techniques are discovered through persistent exploration. The initial state parameters and optimization strategy used for the convolutional neural network will have a significant effect on how the network learns. If the selection is not good, the network will not operate, or it may fall below the local minimum overfit, or have many other issues [18], [19]. It's possible that this phenomenon has multiple root causes. In general, it may be stated that as Deep Neural Networks (DNNs) are easily able to overfit on corrupted labels, training with too many noisy labels decreases the generalization performance of DNNs [19], [20], [21]. Due to overfitting, the system learns the noise patterns contained in the training data, resulting in a substantial difference between the training and test error [9]. Continuous gradient updating and the scale sensitivity of cross-entropy loss are the root causes of overfitting, one of the core problems with deep neural networks. By splitting the samples into correctly and mistakenly categorized ones, it can be seen that they behave quite differently, with the loss decreasing incorrectly classified samples while increasing in incorrectly classified ones[9]. Hypothesis complexity: a crucial topic in statistics and machine learning, the trade-off in complexity is a balance between Variance and Bias. It refers to striking a balance between precision and consistency. Because of this, the models can be radically different on different datasets [22]. A high number of learnable parameters makes DNNs susceptible to overfitting. The model identification efficiency will be very poor if it is trained on more and more characteristics, which introduces more variations into the mix[23]. Induction algorithms frequently use several comparison techniques. During these operations, they consistently evaluate the item with the greatest score after comparing the scores of many objects. However, it is possible that some items will be chosen by this process that will not increase classification accuracy, or may even decrease it [22]. By examining the training dynamics, identifying overfitting emerges as a major challenge in deep learning. Almost all applications emphasize preventing overfitting despite the fact that it is well understood, either by increasing data or by changing the topologies or hyperparameters of neural networks[7]

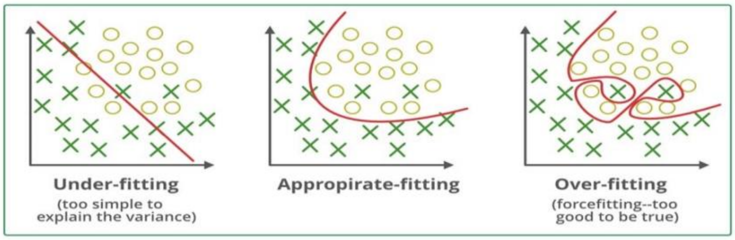
Under-fitting
(too simple to explain the variance)
Appropirate-fitting
Over-fitting
(forcefitting--too good to be true)

Fig. 2 Underfitting and overfitting cases in ML [23]

5. Remedy approaches to the overfitting problem

In order to address the issue of overfitting, several methods have been proposed:

5.1 Data expansion strategy is suggested as a method to update hyper-parameter sets for complex models. The algorithm isn't the sole factor affecting machine learning accuracy. In many circumstances, the numbers and quality of the training dataset can affect its performance. Model training is hyper-parameter tuning. Well-tuned parameters balance accuracy and regularity to prevent overfitting. The model needs enough examples to tweak these parameters [22]. The total classification accuracy in machine learning is not solely dependent on the algorithm. Well-tuned parameters achieve a balanced trade-off between training accuracy and regularity, and they also mitigate the disadvantages of overfitting and underfitting. The model needs enough examples for learning in order to fine-tune these parameters. The number of parameters is directly correlated with the number of samples. And the more parameters that need to be tweaked, the more sophisticated model. In other words, especially for complex models, an extended dataset can significantly enhance forecast accuracy. Because of this, data augmentation is frequently employed and has been proven to be successful as a general method to enhance the performance of model generalization in diverse application domains, comprising image processing and pattern recognition [22]. CutMix was presented to implement the augmentation strategy: patches (as illustrated in Figure 3) are cut and pasted among training images, and ground truth labels are blended according to the area of the patches [24]. Mixup employs convex combinations of pairs of instances and labels to train neural networks. This causes mixup regularization of the neural network, which favors simple linear behavior between training samples [25].

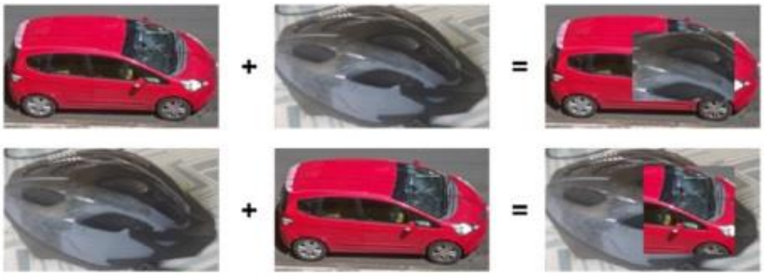

Fig .3 Explain cutmix [16]

5.2 Early-stopping strategy is utilized to avoid overfitting by discontinuing training before performance stops enhancement. This technique is used to prevent the "learning speed slowing" effect. Once a certain point is achieved, the accuracy of algorithms either stops increasing altogether or even begins to decrease as a result of noise learning. The Landweber version of the idea goes back to the 1970s [22], [26]. It has been used in iterative algorithms since the 1990s, particularly neural networks. The goal is to obtain the optimal fit between under-fitting and over-fitting to end training. ANNs learn to find optimal weights and biases.

5.3 Regularization strategy has been proposed to ensure the performance of models to a large extent while also addressing problems that arise in the actual world. This is achieved by the selection of features and the distinction between features that are more and less beneficial. The results of a model may be influenced by a number of different aspects. When there are more features to consider, the model gets more difficult to understand. A model that is overfitting has a tendency to take into consideration all of the features, despite the fact that some of the features have extremely limited effects on the result. Or, what's even worse, half of them are merely noises that don't contribute anything meaningful to the output [11]. In other words, those ineffective elements must be limited in their impact. However, it is not always possible to determine which features are unnecessary, therefore aim to reduce them by reducing the cost function in the model [22]. To do so, the cost function should be given a "penalty term" called regularizer. One theory of linear regression used in L1regularization is the Lasso Regression. The punishment term in this method is the so-called cab distance, which is the total of the absolute values of all the weights. The "Ridge Regression" theory is applied during L2 regularization. Use the Euclidean distance as the punishment term in this strategy[22]. When it is discovered that the testing mistake is substantially heavier than the training error, L1 and L2 regularization procedures as well as the activation function are used to lessen overfitting issues[23].

5.4 Dropout is a method that helps prevent overfitting and offers a way of roughly merging an exponentially large number of distinct neural network topologies in an effective manner. The process of removing units from a neural network, including hidden and visible ones, is referred to as a "dropout." [15]. By "dropping" a unit, the entire set of connections it has coming in and going out of the network can be momentarily deleted. Since Dropout [5] was developed for enhancing the effectiveness of networks by preventing an excessive amount of data from fitting into the training set. Among deep learning practitioners, the dropout method has gained popularity as a way to regularize models from other groups [27]. Although though dropout regularization is frequently used in deep learning, convolutional neural networks perform less well since dropped information can still pass through the networks due to their spatially linked features. In order to solve this, some structured types of dropout have been proposed; however, because features are lost at random, these forms of dropout are prone to over- or under-regularization[7].

5.5 Batchnormalization is employed to diminish a change in the activation layers' "internal covariance shift." Internal covariance shift may be attributed to the alteration in activation spread within each layer. Due to the constant weight increase when teaching, the "internal covariance shift" can be extremely high (it can occur while training data samples are selected from various distributions, such as day-light and night-vision images), and with this high "internal covariance shift," the model converges more slowly and training time increases. The Batch Normalization technique is developed as a CNN hierarchical layer in order to overcome this issue. Batch Normalization minimizes the likelihood of overfitting issues as a result of its small regularization effect [28]. BatchNorm is a method that, to a good degree, aims to improve the training of neural networks by standardizing the distributions of layer inputs. Batching is the method that is utilized to successfully complete this task. This is accomplished by stretching the network layers that regulate the mean and variance, the initial two moments of this dispersion. The disadvantages of Batch Normalization: Increased computational cost: Batch Normalization requires additional computations during both the forward and backward passes. The normalization step introduces some overhead, which can slow down the training process, especially for smaller batch sizes. However, with the availability of modern hardware and optimized implementations, this disadvantage is often mitigated. Batch size dependence: The effectiveness of Batch Normalization can be affected by the choice of batch size. In some cases, very small batch sizes, such as 1 or 2, can lead to unstable and inaccurate estimates of the batch statistics, reducing the effectiveness of Batch Normalization. It is generally recommended to use larger batch sizes for better performance. Inference-time concerns: During inference or deployment, the statistics used for normalization are typically computed from the training data. This means that Batch Normalization introduces an extra step and requires additional memory to store the running means and variances. It can increase the inference time and memory requirements, especially in scenarios where memory is constrained. Dependency on batch order: The order of the samples within a mini-batch can impact the batch normalization statistics since they are calculated based on the samples within the batch. This dependency on batch order can introduce some instability during training and might require additional techniques, such as using batch-wise or group-wise normalization, to mitigate its effects [29].

5.6 Network reduction strategy excludes noises from the training set. Noise learning causes overfitting. Noise reduction is a logical overfitting inhibition research direction. Pruning reduces the final classifiers' size in relationship learning, especially with decision tree learning. Pruning reduces classification complexity by removing unnecessary or less useful data, preventing overfitting, and improving classification accuracy [22]. In the study [11], a new approach is put out for preventing the negative effects of noisy labeling on neural networks. On the same dataset, it entails training two neural networks that are similar to one another and attempting to correlate two different labels. Only the smallest losses on the batch are utilized to update the parameter of the designs after the method calculates the loss by summing the cross-entropy losses of the two networks as well as the contrastive loss between them.

5.7 Cross-validation: DNNs can be trained on a subset and tested on the full dataset in order to contrast the theoretical and empirical generalization performance findings [15]. The data collection D was partitioned into two partitions for cross-validation, namely the training set symbolized by T and the test set represented by R, where the union of these two subsets is the entire dataset and the intersection is the empty set:

$$T \cup \mathrm{R} = D, \tag{1}$$
$$T \cap \mathrm{R} = \emptyset. \tag{2}$$

The T is employed for model training. The R is used to test the model's performance once it has been trained. They use various approaches for cross-validation [2]. In machine learning, the complete historical data is analyzed and split into training and testing sets. Testing data from the past will be used for analysis in the future. In most circumstances, 80% of the data are regarded as training data, and 20% are regarded as testing data. Cross-validation is a key machine learning technique that separates the data into k blocks, with the kth block serving as testing data and the remaining blocks serving as training data. There are K iterations in the K-folds cross-validation method [12]. Testing data will be replaced with new data at the end of each loop. Redoing or iterating here might occur at random. There are other types of cross-validation methods that can be used with CNNs, there are: Stratified K-fold cross-validation is Similar to K-fold, but with the added guarantee that each fold's class distribution reflects the dataset as a whole. When working with datasets that are unbalanced where certain classes may have a disproportionately large number of samples—this is especially helpful. This technique avoids bias and delivers more accurate performance estimates by keeping the class proportions in each fold. LOOCV, or Leave-One-Out Cross-Validation The model is trained using the remaining N-1 data points in LOOCV, and each data point is utilized as a separate validation set. N times are used to complete this operation, where N is the total amount of data points. Among all cross-validation techniques, LOOCV offers the lowest variance and least amount of bias because it trains on practically all of the data that is available. But it might be computationally expensive. The advantages of cross-validation in general, regardless of the specific type, include the ability to estimate a model's performance on unseen data, detect overfitting, and compare different models or hyperparameters. It provides a more reliable evaluation than using a single train-test split, especially when the dataset is limited. Additionally, cross-validation allows for a more thorough understanding of the model's behavior across different subsets of the data, aiding in the detection of potential issues or biases [23].

6. Conclusion

Overfitting is a crucial problem in deep neural networks, and its effect on model generalization, therefore, must find an optimal solution to reduce this problem and model become a well performance in the test set. The difficulty of neural network structures has an impact on overfitting, the quantity and the kind of samples used in training sets, and the number of training epochs. Therefore, to reduce this problem must be fine-tuning the hyper-parameters, which were established using a lot of data. overfitting is a significant issue that can negatively impact the performance of CNNs. However, by utilizing appropriate techniques such as regularization, early stopping, data augmentation, and transfer learning, it is possible to mitigate overfitting and improve the generalization of the model. In this paper, we try to understand this problem through analyses of the overfitting phenomenon. Each task needs an appropriate approach to avoid overfitting and sometimes it needs more than one remedy approach. In a conclusion, the solutions that exist recently do not completely solve the overfitting problem but reduce it.

References

[1] M. Xiao *et al.*, "Addressing Overfitting Problem in Deep Learning-Based Solutions for Next Generation Data-Driven Networks," *Wirel Commun Mob Comput*, vol. 2021, 2021.

[2] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial," *arXiv preprint arXiv:1905.12787*, 2019.

[3] M. Liu *et al.*, "Focuseddropout for convolutional neural network," *Applied Sciences*, vol. 12, no. 15, p. 7682, 2022.

[4] M. R. NarasingaRao, V. Venkatesh Prasad, P. Sai Teja, M. Zindavali, and O. Phanindra Reddy, "A survey on prevention of overfitting in convolution neural networks using machine learning techniques," *International Journal of Engineering and Technology (UAE)*, vol. 7, no. 2.32, pp. 177–180, 2018.

[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[6] L. Wan, M. Zeiler, S. Zhang, Y. le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *International conference on machine learning*, 2013, pp. 1058–1066.

[7] H. Zhu and X. Zhao, "TargetDrop: a targeted regularization method for convolutional neural networks," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 3283–3287.

[8] Z. Ouyang, Y. Feng, Z. He, T. Hao, T. Dai, and S.-T. Xia, "Attentiondrop for convolutional neural networks," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019, pp. 1342–1347.

[9] S. Salman and X. Liu, "Overfitting mechanism and avoidance in deep neural networks," *arXiv preprint arXiv:1901.06566*, 2019.

[10] Y. Assiri, "Stochastic optimization of plain convolutional neural networks with simple methods," *arXiv preprint arXiv:2001.08856*, 2020.

[11] H. Wei, L. Feng, X. Chen, and B. An, "Combating noisy labels by agreement: A joint training method with co-regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13726–13735.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[13] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Transactions on geoscience and remote sensing*, vol. 55, no. 2, pp. 645–657, 2016.

[14] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[15] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental concepts of convolutional neural network," in *Recent trends and advances in artificial intelligence and Internet of Things*, Springer, 2020, pp. 519–567. doi: 10.1007/978-3-030-32644-9_36.

[16] C. F. G. Dos Santos and J. P. Papa, "Avoiding overfitting: A survey on regularization methods for convolutional neural networks," *ACM Computing Surveys (CSUR)*, vol. 54, no. 10s, pp. 1–25, 2022.

[17] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," in *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, 2018, pp. 122–129.

[18] A. Kamilaris and F. X. Prenafeta-Boldú, "A review of the use of convolutional neural networks in agriculture," *J Agric Sci*, vol. 156, no. 3, pp. 312–322, 2018.

[19] F. Samadi, G. Akbarizadeh, and H. Kaabi, "Change detection in SAR images using deep belief network: a new training approach based on morphological images," *IET Image Process*, vol. 13, no. 12, pp. 2255–2264, 2019.

[20] D. Arpit *et al.*, "A closer look at memorization in deep networks," in *International conference on machine learning*, 2017, pp. 233–242.

[21] P. Chen, B. ben Liao, G. Chen, and S. Zhang, "Understanding and utilizing deep neural networks trained with noisy labels," in *International Conference on Machine Learning*, 2019, pp. 1062–1070.

[22] X. Ying, "An overview of overfitting and its solutions," in *Journal of physics: Conference series*, 2019, vol. 1168, no. 2, p. 022022.

[23] J. Kolluri, V. K. Kotte, M. S. B. Phridviraj, and S. Razia, "Reducing overfitting problem in machine learning using novel L1/4 regularization method," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 2020, pp. 934–938.

[24] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[25] M. J. Dinneen, "Improved mixed-example data augmentation," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1262–1270.

[26] G. Raskutti, M. J. Wainwright, and B. Yu, "Early stopping and non-parametric regression: an optimal data-dependent stopping rule," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 335–366, 2014.

[27] Z. Lu, C. Xu, B. Du, T. Ishida, L. Zhang, and M. Sugiyama, "LocalDrop: A hybrid regularization for deep neural networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 44, no. 7, pp. 3590–3601, 2021.

[28] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?," *Adv Neural Inf Process Syst*, vol. 31, 2018.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.