Authors
Yashan Guo, Yuchen Li, Thomas Czartoryski

Introduction

Image classification plays a significant role in various fields, such as visual search and stock photography. With the rapid development of technologies, many well-performed architectures and methods have been proposed. The paper focuses on creating a model to complete image processing tasks with small model size and high accuracy.

This paper investigates the Fashion-MNIST dataset, which is a widely used benchmark dataset for image classification tasks.Fashion-MNIST contains 70,000 grayscale images of clothing items in 10 categories. Its main task is to do supervised learning to predict the correct category label for each 28x28 pixel image. By training on labeled samples, the goal is to develop a highly accurate classifier that generalizes well to unseen test images. Investigating model optimization strategies on this dataset can provide valuable insights and references for other related tasks.

However, there are several challenges in analyzing the Fashion-MNIST dataset. First, the low resolution of pixels limits the number and quality of discriminative features that can be extracted, making it more difficult to capture subtle differences between similar categories such as shirts and t-shirts. Second, some categories contain items that are visually similar, share features and differ only in details, which may be difficult for the model to learn and distinguish. Furthermore, the imbalance in sample distribution among categories may cause the model to be biased towards learning and predicting the majority of the categories. Addressing these challenges requires careful design of feature extractors, classifiers, and sampling strategies or cost-sensitive learning. By developing a CNN model that achieves high accuracy on Fashion-MNIST while maintaining a small number of parameters.

Method

Convolutional Neural Network (CNN) is a deep learning model that is mainly used to process data with a grid structure, such as images. It automatically extracts important features from images through structures such as convolutional, pooling, and fully connected layers, and is used to perform visual tasks such as image classification and object detection.

In this study, we used an improved AlexNet model combined with the VGG architecture, which ultimately achieves a high image classification accuracy of 94% by tuning the parameters and optimizing the structure. This hybrid approach leverages the strengths of each model: AlexNet's simplicity and efficiency and VGG's small convolutional filters for capturing details.

Dataset

The Fashion-MNIST dataset was created by researchers from Zalando, a European e-commerce company, introduced by Xiao et al. in their 2017 paper[4]. The authors demonstrate that Fashion-MNIST poses a more challenging classification task compared to MNIST, with a basic SVM classifier achieving an accuracy of around 89-90%.

Prior to Fashion-MNIST, the MNIST dataset (LeCun et al., 1998) had been the most widely used benchmark for image classification algorithms. However, MNIST was considered too easy for modern machine learning methods. Other similar datasets that have been studied include EMNIST (Cohen et al., 2017), an extension of MNIST with handwritten letters, and KMNIST (Clanuwat et al., 2018), a dataset of handwritten Japanese characters.

In the domain of fashion product images, the DeepFashion dataset (Liu et al., 2016) has been used for tasks such as clothing detection, pose estimation, and attribute prediction. However, DeepFashion is much larger and more complex than Fashion-MNIST, making it less suitable for benchmarking basic classification algorithms.

Data Analysis and Results

AlexNet

For the traditional AlexNet model, we have made some adjustments to enhance efficiency and accuracy. Therefore, this study will not discuss all parameter settings, but will only introduce four adjusted parameters or methods.

A. *Optimizer: Adam*

- Adaptive Moment Estimation (Adam) and Stochastic Gradient Descent (SGD) are two popular optimizers that are commonly used for minimizing the error rate. Adam uses adaptive learning rate and momentum to increase the training speed. Comparing the speed of convergence and model's performance on new data, Adam is better than SGD as an optimizer for the model.
- Filter size: Through different filter size trials, spatial size(5) shows a higher test accuracy than spatial size(3) of traditional AlexNet models. Bigger the spatial size is, the more nonlocal view the model has.

B. Dropout rate: 0 → 0.7

To avoid overfitting, a dropout layer is needed to drop some neurons randomly in an epoch. Through adjusting different numbers, setting the rate into 0.7 is the best.

C. Applying higher fully connected layers : 1st FC layer = 128 → 256, 2nd FC layer = 64 → 128

Fully connected layer, also called the dense layer, is another crucial factor that influences accuracy. Each node in the fully connected layer has a full connection to all the nodes in the previous layer. A smaller fully connected layer containing low capacity will significantly decrease the accuracy on the test dataset. Thus, the fully connected layer number is creased in the AlexNet model built in the research.

D. Applying higher numbers of filters : 1st layer filter num = 16 → 32, 2nd layer filter num = 32 → 64

As one of the most important components of the AlexNet model, the number of it significantly impacts the accuracy of the model. Fewer filters will extract fewer features resulting in the model unable to make satisfied classification.

E. Loss Function

In order to make the adjustment more logically and statistically, the cross-entropy loss function is applied in the model:

$$C = -\sum_{i=1}^{M} y_i log(a_i)$$

y_i is the ground truth

a_i is the output of the neural network

F. Weights update and backpropagation: chain rule

i. To update the weights

$$w_{new} = w - \eta\frac{\partial C}{\partial w}$$

$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial s}\frac{\partial s}{\partial w} = (a - y)\frac{\partial s}{\partial w} = (a - y)a_{pre}$$

ii. To update the biases

$$b_{new} = b - \eta\frac{\partial C}{\partial b}$$

$$\frac{\partial C}{\partial b} = \frac{\partial C}{\partial s}\frac{\partial s}{\partial b} = (a - y)\frac{\partial s}{\partial b} = (a - y)$$

where η is the learning rate, s is the value without activation function and a is the output of the activation function

G. Final AlexNet model

According to the mathematical method and different adjusted processes, a more appropriate AlexNet model is finally built and this model's accuracy is more reasonable and logical(shown in Fig1).

Optimizer: Adam	Spatial size: 5	Dropout rate: 0.7
1st FC layer: 256	2nd FC layer: 128	
1st layer filter num: 32	2nd layer filter num: 64	

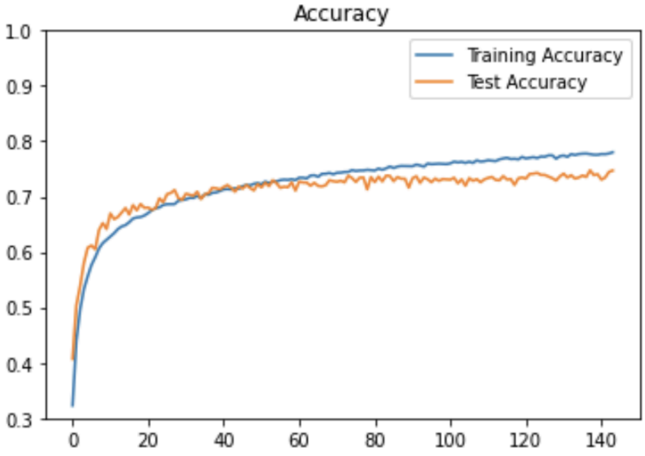

Fig. 1. Training Accuracy: 0.7810 Test Accuracy: 0.7446

OPTIMIZED VGG MODEL

The accuracy rate obtained from the previous AlexNet model is not satisfactory. Thus, other popular state-of-the- art CNN architectures such as AlexNet, VGG, ResNet and DenseNet are taken into consideration. As the computed sources are limited, the ResNet-like architecture is not suitable since it contains deeper neural networks which need weeks for training. Therefore, the basic idea of later modeling is to learn new architectures from VGG Network and try to modify the AlexNet model. Also, some modern optimization algorithms would be utilized to improve accuracy.

A. VGG Model

VGG networks were first proposed by Karen Simonyan and Andrew Zisserman of Oxford Robotics Institute in the year 2014. The VGG architecture is similar to the AlexNet, which is considerably larger than AlexNet but having the same general structure. However, in the VGG network, the number of the feature maps or convolutions is increased as the depth of the network increases, but AlexNet is not. Also, it uses multiple 3*3 kernel-sized filters rather than large kernel-sized filters applied in AlexNet. Since the multiple stacked smaller-sized kernels help the model to learn more complex features at a lower cost, VGG works better than AlexNet. Learning from the VGG networks, several aspects of the current AlexNet model need to be adjusted.

a. Setting multiple convolutional layers at the same level.

b. Window size should be reduced. Instead of using 5 or 7 of window size, 3 is the most efficient for all convolution layers.

c. The Global Average Pooling and Softmax layers are useful to reduce the dimension of feature maps that are output by convolutional layers. Thus, one or two fully connected layers at the end of the network in the current model could be replaced by Global Average Pooling and Softmax layers to achieve better performance.

B. Optimization Methods

a. **Batch Normalization**: As a technique of standardizing the output of the previous neurons, Batch normalization method has the effect of preventing overfitting, stabilizing the network and reducing the training time.

b. **Initialization**: Three different methods have been taken for a trial, involving He_Norm, LSUV_init and Glorot_Uniform. Through comparing the result, the He_Norm shows the best performance since it improves the validation accuracy and shortens the training time

c. **Optimizer**: A right optimization algorithm is helpful to reduce training time exponentially and provide more accurate results. It directly determines how to change weights or to learn rates of the neural network. In this optimized network, the Adam optimizer is used since the model training with Adam usually converges faster than with regular SGD

d. **Data augmentation**: The data augmentation benefits performance of the model. However, it would lower the accuracy if adding too many different augmentation methods at one time. Through several times of trying, this research figures out that the performance degrades when flipping the image vertically or rotating the image at a large angle. Thus, only horizontal flipping and random cropping are applied to the final version of the optimized model.

e. **Activation functions**: The choice of activation functions is related to the output of a neural network. The functions connected with each neuron and decided whether they need to be activated based on the relevance of the input to the model's prediction. There are many widely used activation functions—for instance, ReLU, PReLU, ELU and other variants. However, in the optimized model built, results from ReLU are better than PReLU, ELU and other variations. The reason could be that the Batch Normalization is set in all layers that replace effect ELU.

f. **Dropout**: Dropout is also a useful method for increasing validation accuracy significantly. In the new model, a dropout layer with a rate of 0.1 to 0.3 is added for each convolutional layer and pooling layer, forces the layer to learn more from the input data

g. **Batch size**: The decision of Batch size is essential for the model. Large batch size will accelerate the training process but will also reduce accuracy. A large batch will cause the network to fall into a local minimum. However, a small batch size will improve the performance, but will also significantly slow down the training speed. In this assignment, since the network is relatively small, a small batch size of 32 is suitable.

C. *Final Results*

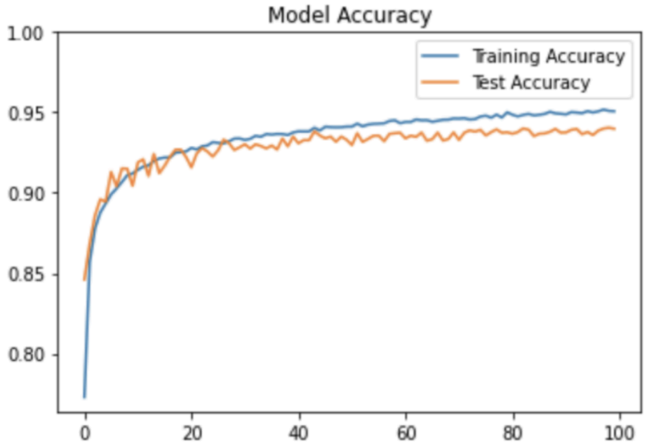

Fig. 2. VGG Model Accuracy

The best accuracy is 0.9405
The number of training epochs is 100.
The training time is 100 * 24sec = 40 min
The inference time is 0.584sec
model size is 198,927
Total params is 198,351
Trainable params is 197,251
Non-trainable params is 1,100

Results Analysis

The optimized AlexNet model demonstrated an accuracy rate of approximately 74%, which was not satisfactory given the project goals. To enhance performance, the model underwent adjustments, incorporating elements from VGG and ResNet architectures, leading to a final model that achieved around 94% accuracy with fewer parameters.

While the achieved improvements are commendable, there are always areas for further enhancement in machine learning projects. For example, exploring deeper or different composite architectures like DenseNet, which might offer better feature integration, could potentially push accuracy higher. Also, expanding the dataset or using more complex data augmentation strategies could provide the models with a richer variety of training examples, possibly leading to better generalization on unseen data.

Additionally, employing more advanced techniques in optimization algorithms and layer configurations, or exploring recent advances in activation functions and loss functions, might also yield improvements. However, these changes often come with increased computational costs and complexity, requiring more resources and potentially longer development times.

Discussion - Related Work

The latest methods for clothing image classification mainly focus on four aspects. First, researchers have built large-scale, high-quality clothing image datasets, such as Fashion-MNIST [4] and DeepFashion [6], which includes over 800,000 images with rich annotations. These datasets provide diverse samples for training and evaluating classification algorithms.

Several researchers have explored various algorithms and models to address the problem of clothing image classification. Liu et al. [1] proposed a two-stage model for fashion landmark detection, which first performs global clothing category classification and then combines local features for landmark localization. Their approach achieved good localization performance on

clothing images. Zheng et al. [2] constructed a large-scale dataset called DeepFashion with clothing attribute annotations and introduced a weakly-supervised learning method for clothing attribute classification, enabling fine-grained understanding of clothing images. ModaNet introduces a multi-scale pyramid pooling structure to better capture local and global features of clothing items. Al-Halah et al. [3] developed a fashion style prediction model based on visual features, which analyzes the color, texture, and shape of clothing to predict its fashion style.
We also saw different modeling applications in different works. Hadi Kiapour et al. [5] propose a method for matching street clothing photos to online shop images, while Ak et al. [6] develop an efficient multi-attribute similarity learning approach for attribute-based fashion search.

The findings of this study share similarities with existing work but also present some different discoveries. This paper focuses more on the computational efficiency and parameter scale of the model, which is a significant application in the real world, like used in some embedded devices. Through network structure improvements and hyperparameter tuning, a more compact and efficient model is obtained while maintaining high classification accuracy. In addition, our work adopts optimization strategies targeted to the characteristics of the Fashion-MNIST dataset, such as data augmentation and batch normalization, achieving better results compared to the baseline models.

Discussion - Related implementations

Several developers on Kaggle have already attempted various machine learning and deep learning methods on the Fashion-MNIST dataset. In this section, we will discuss two notable implementations and compare them with our proposed approach.
First, we reviewed the CNN implementation for the Fashion-MNIST dataset provided by @HaneenHossam on Kaggle[13]. This implementation uses the Keras framework to build a CNN model consisting of two convolutional layers, two pooling layers, and two fully connected layers. The author uses ReLU activation function, Adam optimizer, and cross-entropy loss function to train the model, and applies Early Stopping and Dropout regularization to prevent overfitting. The model achieves approximately 91% classification accuracy on the test set. Compared to @HaneenHossam's implementation, we perform more fine-grained adjustments and optimizations on the classic architectures. We increase the number and depth of convolutional layers, use smaller convolutional kernels and strides, and introduce a global average pooling layer to obtain a deeper and more streamlined network structure. Second, we employ more modern optimization techniques to accelerate the model's training process and convergence speed.
https://www.kaggle.com/code/haneenhossam/fashion-mnist-cnn

Another notable implementation is the autoencoder and variational autoencoder (VAE) practice on the MNIST dataset by Gabriel Cabas [14]. In this work, Gabriel focuses on the following aspects: Building a simple autoencoder using Keras to compress MNIST images into a latent space and reconstruct them. By visualizing the encoder's output, he demonstrates the learned representations of digits by the autoencoder. Implementing a variational autoencoder (VAE) with KL divergence regularization, which makes the latent space smoother and facilitates interpolation and random sampling to generate new digits. He showcases some sample digit images generated by the VAE.
While Gabriel's practice shares some similarities with our work in exploring autoencoders and generative models on MNIST, our focus is different. In addition to VAE, we experiment with more generative models such as GAN and WGAN, and compare their image generation effects.We also systematically evaluate the quantitative metrics of each model, such as reconstruction error and generated sample quality, rather than relying solely on subjective visualization results. Furthermore, in the autoencoder part, we explore the application of overcomplete autoencoders in tasks like denoising and anomaly detection.
https://www.kaggle.com/code/gabrielcabas/autoencoders-and-vae-practice

Analysis

We believe that the proposed algorithms and features are highly effective for the image classification task on the Fashion-MNIST dataset. By adjusting key components such as network depth, convolutional kernel size, and fully connected layers according to the characteristics of the Fashion-MNIST dataset, such as image size and number of classes, we have obtained a more efficient and streamlined model architecture. This targeted optimization enables our model to better adapt to the features and challenges of the Fashion-MNIST dataset, resulting in superior classification performance.

Then, we have employed various modern optimization techniques. Batch normalization effectively reduces internal covariate shift by normalizing the input of each layer, accelerating training convergence and enhancing model stability. He_initialization and Adam optimizer further promote the model's training efficiency and performance from the perspectives of weight initialization and optimization algorithm, respectively. The comprehensive application of these optimization techniques allows our model to learn the key features and patterns of the Fashion-MNIST dataset more quickly and stably.

Moreover, for a small dataset like Fashion-MNIST, data augmentation and regularization are particularly important, as they can effectively compensate for the insufficient number of samples and enable the model to better capture the essential features of clothing images without overfitting the training set details.

We believe that the model optimization and improvement strategies proposed in this paper have a certain level of generalizability and can be applied to other similar image classification tasks and datasets. For datasets with similar characteristics, such as low resolution, grayscale images, and class imbalance, our method can provide beneficial reference and inspiration. For example, in the fields of handwritten character recognition, traffic sign classification, and medical image diagnosis, there exist challenges similar to those in Fashion-MNIST. Transferring our model architecture adjustments and optimization methods to these tasks is likely to achieve equally outstanding performance. Of course, when applying to new relevantly small datasets, appropriate adjustments and optimizations should be made according to their specific characteristics and requirements. But on larger datasets, our model might not be powerful enough. In such cases, more powerful models with larger spatial footprints are required to process the data effectively.

We believe that these strategies can provide important reference value for solving other image classification datasets with similar characteristics. In the future, we plan to extend these methods to more image classification tasks and datasets to further verify their generalizability and effectiveness, and continuously improve and optimize them to achieve more robust and efficient image classification models.

Conclusion

For anyone seeking to use our solution for a similar image classification task, we recommend starting with our optimized model architecture and hyperparameter settings. The combination of a streamlined VGG-like architecture with effective optimization techniques has proven successful on the Fashion-MNIST dataset and may generalize well to other datasets with similar characteristics, such as grayscale images, small image sizes, and a moderate number of classes. However, it is important to keep in mind that some fine-tuning may be necessary to adapt the model to the specific characteristics and requirements of a new dataset.

If provided with more data, there are several potential avenues for further improving the model's performance and robustness. First, with a larger and more diverse training set, the model could learn more representative and generalizable features, potentially increasing its accuracy and reducing overfitting. Second, additional data would allow for more extensive experimentation with data augmentation techniques, helping the model to better handle variations in object appearance, lighting, and other factors.

For a company seeking to run this solution at a high scale, we recommend leveraging our optimized model architecture and hyperparameters as a starting point, but also investing in the computational resources and infrastructure necessary to train and deploy deep learning models efficiently. Additionally, the company should prioritize data quality and diversity when curating their training dataset, as this can have a significant impact on the model's performance and generalization ability. By following these recommendations and building upon our optimized solution, a company can develop a robust and scalable image classification system that delivers accurate and reliable results.

Reference

[1] Z. Liu, S. Yan, P. Luo, X. Wang, and X. Tang. Fashion Landmark Detection in the Wild. arXiv:1608.03049, 2016.

[2] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu. ModaNet: A Large-Scale Street Fashion Dataset with Polygon Annotations. arXiv:1807.01394, 2018.

[3] Z. Al-Halah and R. Stiefelhagen. Fashion Forward: Forecasting Visual Style in Fashion. arXiv:1705.06394, 2017.

[4] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747, 2017.

[5] M. Hadi Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg. Where to Buy It: Matching Street Clothing Photos in Online Shops. In Proceedings of the IEEE international conference on computer vision, pages 3343-3351, 2015.

[6] K. E. Ak, J. H. Lim, J. Y. Tham, and A. A. Kassim. Efficient Multi-attribute Similarity Learning towards Attribute-based Fashion Search. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2597-2606, 2019.

[7] K. Bortoletto. Predicting blastocyst formation of day 3 embryos using a convolutional neural network (CNN): a machine learning approach. Fertility and sterility, 112(3):e272-e273, 2019. doi:10.1016/j.fertnstert.2019.07.807

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278-2324, 1998.

[9] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. EMNIST: Extending MNIST to handwritten letters. In 2017 International Joint Conference on Neural Networks (IJCNN), pages 2921-2926. IEEE, 2017.

[10] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. Deep learning for classical Japanese literature. arXiv preprint arXiv:1812.01718, 2018.

[11] A. Sangaiah. Deep Learning and Parallel Computing Environment for Bioengineering Systems. In Deep Learning and Parallel Computing Environment for Bioengineering Systems. Elsevier Science Technology, 2019.

[12] R. Shin. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE transactions on medical imaging, 35(5):1285-1298, 2016. doi:10.1109/tmi.2016.2528162

[13] Kaggle:https://www.kaggle.com/code/haneenhossam/fashion-mnist-cnn

[14] Kaggle:https://www.kaggle.com/code/gabrielcabas/autoencoders-and-vae-practice

Statement of contributions: All group members contributed equally to this project. Yuchen Li Yashan Guo and Thomas Czartoryski collaborated closely on both the coding and writing aspects of the work.

Appendix: See the submitted zip file