- ORIGINAL ARTICLE -

Towards more efficient initialization methods for Convolutional Neural Networks via K-Means and Principal Components

Hacia métodos más eficientes de inicialización para Redes Neuronales Convolucionales por medio de K-Medias y Componentes Principales

Federico Rabinovich[1], Franco Ronchetti[1,2], and Facundo Quiroga[1,2]

[1]*Instituto de Investigación en Informática LIDI - Universidad Nacional de La Plata, La Plata, Argentina*
federicorabinovich@gmail.com
[2]*Comisión de Investigaciones Científicas de la Pcia. de Bs. As. (CIC-PBA), La Plata, Argentina*

Abstract

This paper presents an exploration of unsupervised methods for initializing and training filters in convolutional layers, aiming to reduce the dependency on labeled data and computational resources. We propose two unsupervised methods based on the distribution of input data and evaluate their performance against traditional Glorot Uniform initialization. By initializing solely the initial layer of a basic CNN network with one of our proposed methods, we attained a 0.78% enhancement in final accuracy compared to traditional Glorot Uniform initialization. Our findings suggest that these unsupervised methods could serve as effective alternatives for filter initialization, potentially leading to more efficient training processes and a better understanding of CNNs.

Keywords: Clustering, Convolutional Neural Networks, Initialization, K-Means, Principal Components.

Resumen

Este artículo presenta una exploración de métodos no supervisados para la inicialización y el entrenamiento de filtros en capas convolucionales, con el objetivo de reducir la dependencia de datos etiquetados y recursos computacionales. Proponemos dos métodos no supervisados basados en la distribución de los datos de entrada y evaluamos su rendimiento en comparación con la inicialización Glorot Uniform tradicional. Al inicializar únicamente la capa inicial de una red CNN básica con uno de nuestros métodos propuestos, logramos una mejora del 0.78% en la precisión final en comparación con la inicialización Glorot Uniform tradicional. Nuestros hallazgos sugieren que estos métodos no supervisados podrían servir como alternativas efectivas para la inicialización de filtros, lo que podría llevar a procesos de entrenamiento más eficientes y a una mejor comprensión de las redes convolucionales.

Palabras claves: Agrupamiento no supervisado, Componentes principales, Inicialización, K-Medias, Redes Neuronales Convolucionales.

1 Introduction

Convolutional Neural Networks have revolutionized the field of image recognition and classification. The traditional approach to training these networks involves weight initialization by sampling from uniform or normal distribution followed by supervised learning with backpropagation and gradient descent. This method usually requires extensive computational resources and labeled datasets that poses significant challenges, limiting access to these technologies and contributing to ecological concerns.

The quest for more efficient training methods has led us to the exploration of unsupervised alternatives that leverage the inherent structure of the input data.

This research focuses on developing and evaluating unsupervised methods for initializing convolutional filters, with the ultimate goal of reducing the reliance on labeled data and computational power at training time. We propose to use Principal Components and K-Means applied to image patches to initialize convolutional layers instead of traditional methods based on sampling from normal or uniform distributions.

The use of Principal Components and K-Means for initializing convolutional filters is justified by their simplicity, computational efficiency, and interpretability. K-Means, a centroid-based clustering algorithm, scales well with large datasets and offers stable, reproducible results, making it practical for initializing CNNs. Principal Components, as a dimensionality reduction technique, captures significant features of the data, facilitating effective clustering and reducing data complexity. Both methods are widely implemented in machine learning libraries, ensuring ease of use and reproducibility. Their extensive study and proven effectiveness in various domains provide a solid foundation for their application for initializing convolutional filters. Alternative clustering techniques, such as DBSCAN or hierarchical clustering, may introduce sensitivity to hyperparameters and less consistent outcomes, complicating the initialization process. Thus, Principal Components and K-Means strike a balance of robustness, efficiency, and clarity, making them a reliable baseline for further experimentation and comparison with more complex methods.

Preliminary results indicate that the Principal Components algorithm could serve as a potential candidate for generating CNN kernels closer to a local minima compared to those obtained by random distribution sampling (i.e., Glorot or He). We argue that initialization methods grounded in data distribution, such as K-Means (KM) and Principal Components (PC), represent a more intuitive choice than random sampling. These methods harness information from the data to establish a more reasonable starting point for training.

In order to initialize the parameters of the first layer of a CNN, we commence by randomly sampling a reduced subset of images from the CIFAR-10 dataset (refer to Fig. 1).

For the K-Means based approach, we compute k centroids which are directly applied as initialization values for the filters after a simple normalization transformation. Analogously, for the PC based method, the eigenvectors of the tensor are employed as the kernels for the CNN's first layer.

In this paper, we present preliminary results for the application of these initialization methods. We evaluate these methods on the first layer of a simple CNN and then compare the results obtained with traditional Glorot Uniform initialization. PC specifically shows significant improvements in convergence time and final accuracy. Considering that the initialization method was applied to the first layer only, following this technique for the complete network promises immense benefits in terms of reducing computational power, training time, and labeled dataset size. This path is being followed in current works and we hope to share their results in future publications.

This document is organized as follows: in section 2 we discuss previous works related to the proposed methods, in section 3 we explain these methods in more depth, in section 4 we present the experiments undertaken and their results and in section 5 we discuss the conclusions of this work.

2 Related work

We briefly review previous works where PC and K-Means were explored as initialization or training strategies for CNNs.

2.1 Strategies based on Principal Component Analysis (PCA) for Training and Initialization of Convolutional Filters

PCA decomposition of datasets has garnered attention among researchers for training convolutional filters within Convolutional Neural Networks (CNNs). PCA identifies orthogonal vectors, termed eigenvectors, ordered by variance explained, making them efficient for capturing input space variability. Chan et al. (2014) [1] pioneered PCA-based filter training, proposing strategies for cascaded convolutional layer training. They introduced PCANet, constructing filters via PCA on input patches. Ren et al. (2016) [2] employed PCA to initialize filter parameters, enhancing subsequent Backpropagation training convergence and gradient degradation mitigation. Soon et al. (2020) [3] extended PCA training to all convolutional layers, omitting GPU dependence for efficiency. Zhang et al. (2022) [4] combined PCA with Linear Discriminant Analysis (LDA) in PLDANet for computational cost reduction and parameter interpretability. Sun et al. (2019) [5] applied PCA for early disease detection in pigs, optimizing filter size for classifier accuracy. Tian et al. (2015) [6] introduced Stacked PCA Network for facial recognition, emphasizing linear activation functions between convolutional layers. Khaw et al. (2017) [7] reduced training time by training only the fully connected layer after PCA-based filter training for image noise classification.

2.2 Strategies based on K-Means for Training and Initialization

K-Means, a widely-used unsupervised clustering algorithm, serves as a strategy for convolutional filter training. Coates et al. (2012) [8] utilized K-Means for unsupervised patch clustering to preprocess images before neural network training, alleviating computational costs associated with hierarchical feature layer training. However, literature on K-Means-based unsupervised convolutional filter training is scarce.

3 Proposed methods

This section outlines various unsupervised methods proposed for initializing filters for convolutional layers without the reliance on labeled image datasets. Each method follows a distinct implementation procedure but is grounded in common basic criteria and considerations.

In all initializing methods, the dataset undergoes preprocessing to obtain patches from the original images for the initial layer. These patches, with dimensions identical to the filters being constructed, are utilized for filter initialization across various methods.

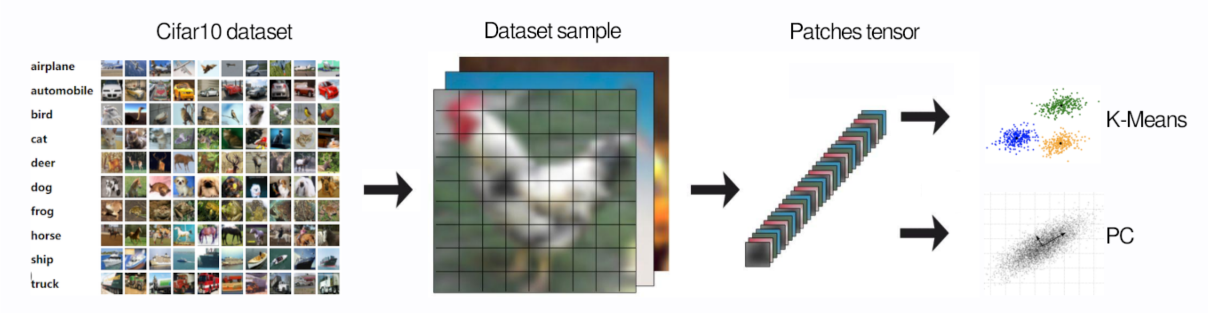

Figure 1: General overview of the process: A sample of images is drawn from CIFAR-10. Patches are extracted to form a single tensor, onto which either the PC or K-Means method is applied.

3.1 Relationship between filters, patches and convolutions

One relevant consideration for convolutional filter initialization is that a specific filter attains maximum activation when faced with an identical patch under the condition that both are normalized by their Euclidean norm. Under these circumstances, maximum activation of a convolutional filter is achieved solely with an image patch identical to the filter. Ultimately, once both filter and patch are normalized, convolution reduces to a mere cosine similarity calculation, since the convolution operation at the patch-filter interaction level is just a dot product.

This last point becomes evident when vectorizing both the patch and the filter. Once transformed into vectors, the dot product operation, i.e., the sum of the element-wise product of both vectors, is entirely equivalent to convolution. Moreover, considering that the cosine of the angle formed by both vectors is equivalent to the normalized dot product, the previously exposed point becomes clear: a filter will attain its maximum activation against an identical patch under the condition that both are normalized beforehand.

To further elucidate the behavior of the convolution operation and the relationship between filter-patch similarity and the obtained activation value, outputs for the same filter and different image patches are presented (refer to Fig. 2). While the filters generated by the various methods presented here are normalized in one way or another, image patches from the input set undergo no patch-level normalization. Consequently, the condition necessary to achieve maximum activation under maximum cosine similarity between filter and patch is not fulfilled to reduce convolution operation to a dot product (at the patch level). Under this circumstance, maximum activations for a specific filter are obtained with patches containing maximum scalars for positive dimensions of the filter and minimum scalars for negative dimensions of the filter. Thus, from the perspective of filter activation for the entire patch set, tonal gradations of the filter are irrelevant to maximize its activation.

Another circumstance arises when taking the inverse perspective, i.e., against a specific image patch, its convolution with different filters will produce differentiated activations according to the tonal gradation - the scalar value - present in each dimension of each filter. That is, two filters with the same dimensions with positive scalars and the same dimensions with negative scalars, but different, will not produce the same activation scalar when convolved with the same image patch. Translated to the level of the entire image, this means that these two filters will produce different feature maps for the same input image, even with identical signs for the scalars of all their dimensions.

Applied to the hypothesis posed earlier regarding the relationship between filter-patch similarity and activation level, generating initial weights for filters from the input patch set distribution aims to produce a representative set of feature maps for input images. For a specific image patch and considering filters normalized with a zero mean, the filter that will give the highest activation to said image patch will be the one with which it has the highest similarity.

One fundamental criterion involves initializing filters based on the distribution of input data. Here, input data refers not only to the set of images used to train the models but also to the feature maps produced by each convolutional layer. Even though all experiments where conducted for the first layer, this could potentially be extended to a all layers in a CNN. Each layer is independently initialized, utilizing the output produced by the immediate preceding layer as the training set through the inference process (forward pass). For the initial layer, the input data comprises the image set directly.

Following sections detail the methodologies employed for initializing convolutional filters, outlining the preprocessing, training, and post-processing stages for each method: Principal Components (CP), K-Means (KM), and random patches.

3.2 Principal Components (PC) Method

In this section, the process of initializing convolutional filters through the Principal Components (PC) method is addressed. The method involves utilizing input im-

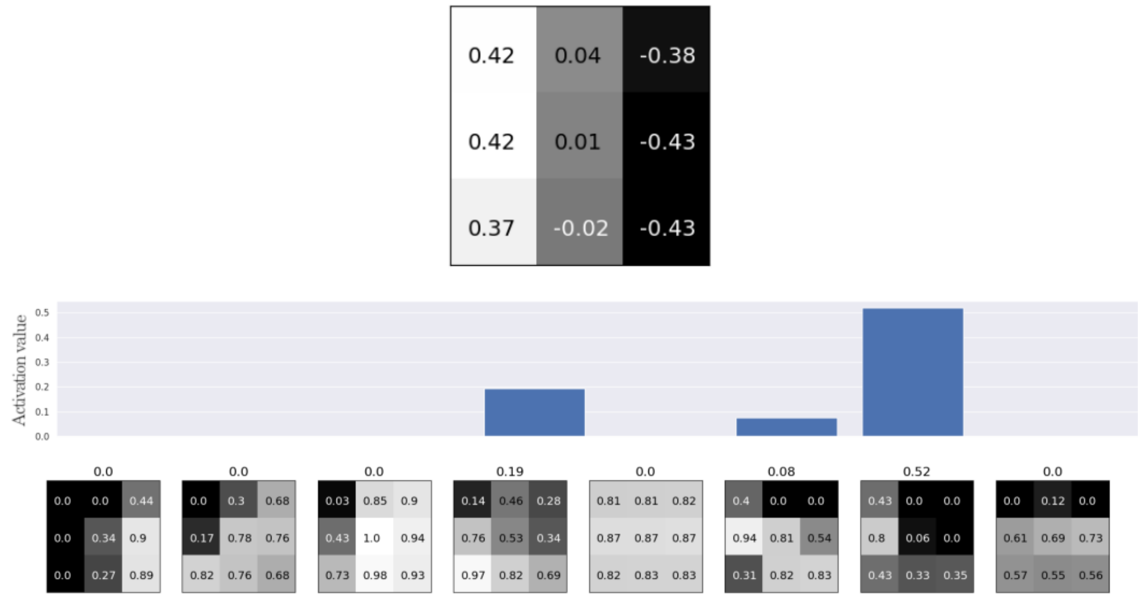

Figure 2: Top: Normalized 3x3 convolutional filter. Middle row: Activation values for each of the patches in the bottom row when convoluted with the top filter. Bottom row: Unnormalized patches.

ages to obtain patches and determine the directions in pixel space where these patches exhibit maximum variability. Each patch, obtained by scanning the input image left to right and top to bottom, is considered as a set of values that can be vectorized. For example, a spatial dimension of 5x5 pixels with an input image that has 3 channels results in a 5x5x3 patch. These patches are then used to create a dataset, from which directions of maximum variability are calculated to construct convolutional filters.

The underlying hypothesis of this approach is based on the hypothesis that dimensions of maximum variability in input data facilitate condensing information for subsequent layers of the network.

The methodology entails preprocessing, training, and post-processing steps for generating a set of filters through PC. Preprocessing involves reading images, obtaining a representative sample, and generating a tensor of image patches. These patches are standardized at the patch level to ensure uniform activation capacities and balanced output ranges for the ensuing filters.

Training involves standardizing patches at the feature level, computing the covariance matrix, and obtaining eigenvectors. The top eigenvectors, corresponding to the desired number of convolutional filters, are retained for further processing.

Post-processing involves resizing and concatenating the eigenvectors to obtain a tensor applicable as weights for a convolutional layer. Notably, the biases of the layer are set to zero as the set of weights is already zero-mean normalized. Fig. 3 shows the result of using the PC method to compute the 27 filters for a 3x3 convolutional layer of a network using CIFAR10 images. In this case, since the convolution has a kernel of 3x3 and the images have 3 channels, the PC method can compute at most 3x3x3=27 filters, since that is the dimension of the samples.

3.3 K-Means (KM) Method

The process of initializing convolutional filters through the K-Means-based method is founded on two premises. Firstly, there exist regions in the patch space that are densely populated compared to others, including regions completely devoid of patches. In other words, it is presumed that there are clusters or modes of patches, as opposed to a relatively uniform distribution in the patch space. Secondly, the centroids of these patch clusters, when used as convolutional filters, efficiently capture the majority of the contained information.

The proposition with K-Means is to identify n regions and their centroids, where n represents the number of filters to be generated, which can best cover the distribution of input patches in the d-dimensional space. Ideally, these centroids capture the highest number of patches through their activations, thereby generating a comprehensive representation of the distribution of input patches in the output feature maps.

Fig. 4 also shows the result of using the K-Means method to compute filters for a 3x3 convolutional layer of a network using CIFAR10 images. In this case, the number of filters was selected so as to match those computed by the PC method, but other values of K are possible.

Additionally, the use of K-Medoids is proposed as an alternative, employing the median instead of the mean of the cluster as a filter. This algorithm is pre-

Figure 3: Top row: 27 3x3 filters obtained through PC with a sample of 100 images from the CIFAR10 dataset. Bottom row: the first 32 filters (with higher eigenvalues) of 7x7 obtained through PC with a sample of 100 images from the CIFAR10 dataset.

sented as a more robust alternative against outliers and ensures that the proposed filters are actual images patches from the dataset improving the interpretability of the resulting weights.

3.4 Random Patches Method

This method serves as a baseline for comparison with the other two methods. Filters are generated from a random sample of patches, aiming to establish the validity of the hypotheses underlying the PC and KM methods.

The procedure for training filters using random patches involves a preprocessing stage identical to that described for KM method. From the tensor generated in this preprocessing stage, random elements are sampled, as many as the desired number of filters, and a new tensor identical to the previous one is constructed containing only the randomly selected elements. The postptocessing stage is again identical to that described for KM method. Fig. 5 shows sample random patches from CIFAR10, after using this method.

This method provides a way to run control experiments to ensure that the PC and KM methods actually contribute to the initialization beyond simply using a patch sampling strategy.

4 Experiments

In this section, we present the experiments conducted with each of the filter initialization methods for convolutional layers. The objective of these experiments is to validate the hypotheses proposed throughout this work and to gain understanding regarding the operation of convolutional networks. Specifically, the aim is to estimate the utility of the proposed alternative initialization methods and validate the underlying hypothesis.

4.1 Objectives

The primary objective of this series of experiments is to establish a comparison point between filters generated based on a certain inductive bias from input data (images) and those generated completely randomly (Glorot Uniform). This allows clarifying the impact of specific initialization policies on filter training and comparing such policies with the results obtained through purely probabilistic generation. Additionally, it enables determining how much these policies influence the final performance outcome of the network compared to generation based on simple random sampling.

Using the sets of filters generated by each method, training runs are conducted using the standard method of gradient descent and backpropagation. Comparing both the convergence speed of training and the final performance of the network in the image classification task with different initialization methods not only enhances understanding of how convolutional networks operate but also provides alternative initialization methods with the potential for positive impact on both the subsequent performance obtained from the network and the potential reduction of computational resources required for training.

Furthermore, comparisons are made between the initial (initialization) and final (after training) states of the same set of filters. Filters that change very little during training can be indicative of the usefulness of the initialization. In other words, these filters have probably been initialized closer to a local minimum.

4.2 Criteria for Experimentation

In order to ensure the comparability of results among the different filter initialization strategies, a set of criteria for conducting the experiments is proposed.

4.2.1 Standardized Convolutional Network:

All experiments utilize the same simple convolutional network, denoted as SimpleConv, constructed ad-hoc. This network is based on the repetition of 11 simple blocks, each comprising a convolutional layer, a batch normalization layer, and a ReLU activation. The filters used in all cases are 3x3 pixels, without padding, and with a stride of 1.

4.2.2 Datasets:

All experiments are conducted on two widely used datasets in the literature: CIFAR-10 and Fashion MNIST.

4.2.3 Replication:

Each experiment is repeated three times, and the results are averaged. This approach aims to minimize the risks of inconclusive results due to the inherent stochasticity of the training procedure.

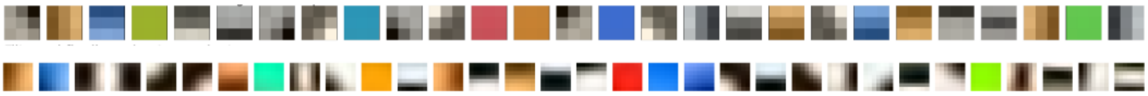

Figure 4: Top row: 27 filters of 3x3 obtained through K-Means (KM) with a sample of 100 images from the CIFAR10 dataset. Bottom row: 32 filters of 7x7 obtained via K-Means (KM) with a sample of 100 images from the CIFAR10 dataset.

Figure 5: Top row: 27 filters of size 3x3 obtained through random patch sampling from a sample of 100 images from the CIFAR10 dataset. Bottom row: 32 filters of size 7x7 obtained through random patch sampling from a sample of 100 images from the CIFAR10 dataset.

4.2.4 Initialization Methods:

Each experiment involves the initialization of the filters of the first layer using one of the following initialization procedures:

1. Glorot Uniform.
2. Principal Components.
3. K-Means.
4. Random Patch Selection.

The rest of the layers are initialized with Glorot Uniform in all setups. Following initialization, standard training procedures (Gradient Descent and Backpropagation) are applied for 250 epochs. The learning rate is set to 1e-3 for the first 200 epochs and reduced to 1e-4 for the remaining 50 epochs. As shown in Fig. 6, while all initializations achieve similar accuracies, the PC method has a slighty higher performance.

4.3 Results

The results of the trainings conducted subsequent to initialization through each of the proposed methods and Glorot Uniform are visualized below. For each method, the network's performance on the validation dataset was measured at the end of each of the 250 epochs.

Table 1: Final accuracy

Init method	CIFAR-10	F MNIST
PC	**89.52 %**	**92.69 %**
Glorot Uniform	89.43 %	91.91 %
KM	89.09 %	91.61 %
Random patches	88.58 %	N/A

As a summary of the results obtained in the various training sessions and to facilitate subsequent analysis, a graph presenting the final accuracy values is provided. Table 1 compares the final performances achieved by the different initialization methods. The results (averaged over 3 runs) are presented independently for each dataset.

It is important to emphasize and recall that all results presented in this work correspond to variations of weight initialization performed always and solely on the first convolutional layer. The remaining layers were initialized using Glorot Uniform. Hence, the scale of impact on performance between one training method and another is reduced.

As observed, both the random patch selection method and the initialization via K-Means do not offer better performance than Glorot Uniform. Conversely, the initialization method via Principal Components seems to offer better performance on both datasets and even achieve slightly faster convergence than Glorot Uniform. The differences are, of course, relatively minor because the different methods are applied only to the first layer of the network. It is expected that consistently applying these initialization methods across all layers will yield more significant performance variations among the different methods.

The consistency of results obtained across both datasets is noteworthy, which lends greater confidence to the conclusions drawn.

4.4 Resulting filters after initialization and training

In addition to increase the final accuracy of the method, the initialization procedures can improve the interpretability and reduce the number of training iterations of the network. Therefore, we explore the filters generated by each initialization method across each dataset, along with their final state post-training. Additionally, we analyze the cosine distance between the initial and final states for each filter to quantify the changes produced on the filters. Fig. 7 and Fig. 8 show this comparison for a model trained on the CIFAR10 and FashionMNIST datasets.

For all experiments conducted on CIFAR-10, a total

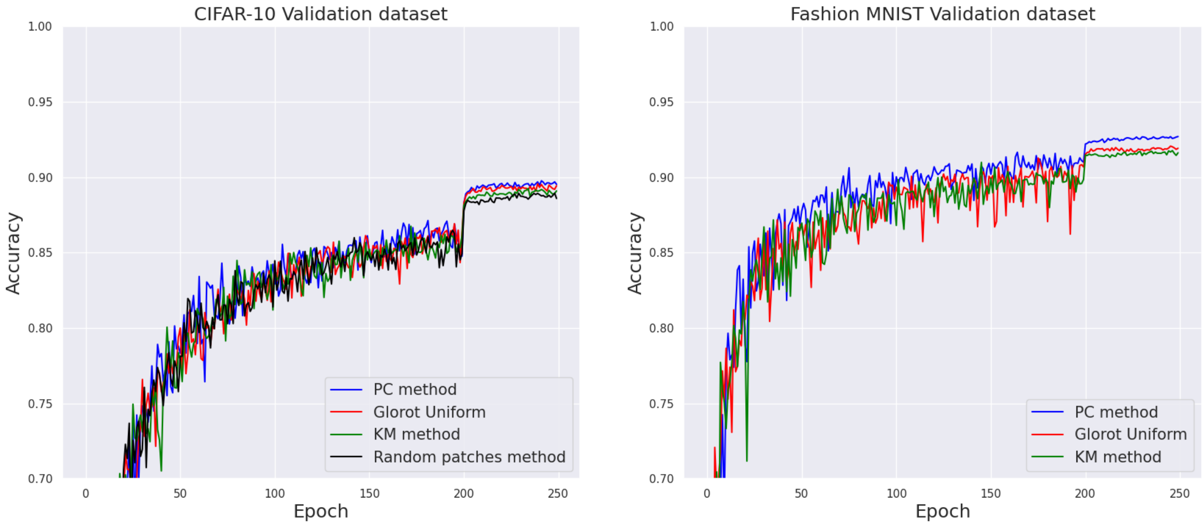

Figure 6: Accuracy and convergence time of the training procedure for the different initialization methods over validation dataset. Left: CIFAR-10. Right: Fashion MNIST.

of 27 filters were generated. The rationale behind this selection lies in the inherent constraints of the filter training method using Principal Component (PC). Specifically, this method can generate a maximum number of filters equal to the dimensions of the image patches used as input. For RGB color images with 3 channels and 3x3 pixel patches, this results in a dimensionality of 27 (3 channels × 3 pixels wide × 3 pixels high). To ensure comparability across results obtained with different methods and architectures, the number of filters was fixed at 27, representing the maximum feasible for Principal Components.

Similarly, in all experiments conducted on Fashion MNIST, 9 filters were employed to enable comparability across different architectures and methods. In the case of Fashion MNIST, with a single channel and 3x3 patches, 9 filters represent the maximum number feasible through Principal Components.

Across both datasets, filters initialized with Glorot Uniform experienced the most significant modifications during training. Conversely, filters initialized through Principal Components (PC) and K-Means (KM) underwent fewer modifications during training. This observation suggests that initialization via either of these two methods provides some approximation to a local minimum for the first convolutional layer.

Additionally, it can be observed that the filters that undergo the most significant modification in the case of initialization by Principal Components (PC) are the last ones, which coincidentally have the smallest eigenvalues. There even seems to be a certain increase in the cosine distance between the initial and final filters as one progressively moves through the set of filters. The results are consistent for both datasets, as shown in Fig. 9.

The initialization of filters by both PC and KM provides an initial parameter configuration that is much more efficient in terms of subsequent optimization needs through training. This could well suggest that the filters are already very close to a local minimum. The implications of these results are of significance within the framework of convolutional neural network training.

Considering the results obtained regarding convergence speed and accuracy, as well as the minimal alteration experienced by the filters during training, the initialization method via PC appears to offer significant advantages over Glorot Uniform. This would naturally result in the optimization of computational resources during training, as it potentially implies the requirement for fewer iterations to achieve network convergence.

5 Conclusions

This study investigated alternative methods for initializing convolutional neural network (CNN) filters, with a primary focus on enhancing training efficiency.

Our experimental approach involved constructing filters through K-Means (KM) and Principal Components (PC) algorithms, juxtaposed against traditional Glorot Uniform initialization. Notably, filters initialized via Glorot Uniform exhibited substantial modifications during training, indicative of a greater distance from local minima and consequently higher computational resource requirements. In contrast, PC and KM methods yielded filters closer to local minima, suggesting potential reductions in training efforts and computational costs.

Furthermore, our evaluation of network performance post-initialization and subsequent training revealed a slight superiority of PC initialization in terms of performance and convergence speed compared to Glorot Uniform. This finding underscores the effi-

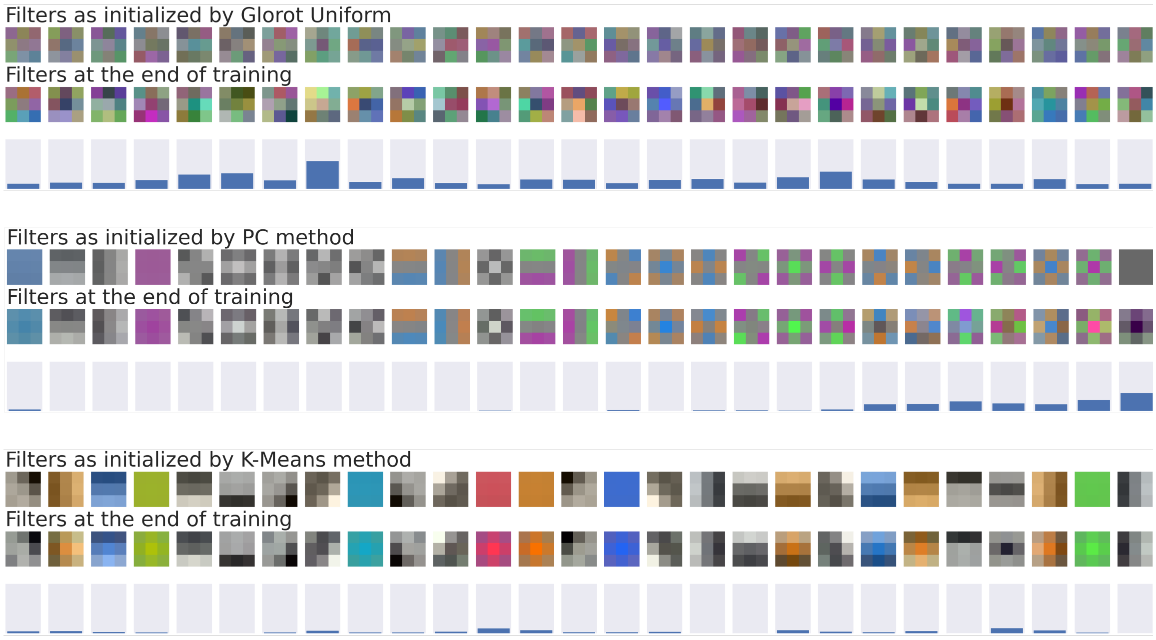

Figure 7: Filters resulting from the different initialization methods, before and after training, on the CIFAR-10 dataset. For each set, the cosine distance before and after training is also visualized. First group: Glorot Uniform. Second group: Principal Components (PC). Third group: K-Means (KM).

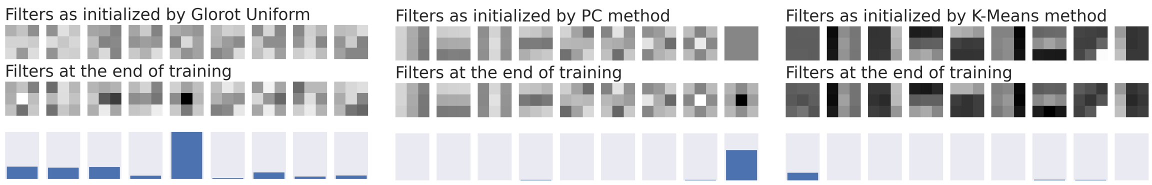

Figure 8: Filters resulting from the different initialization methods, before and after training, on the Fashion MNIST dataset. First group: Glorot Uniform. Second group: Principal Components (PC). Third group: K-Means (KM).

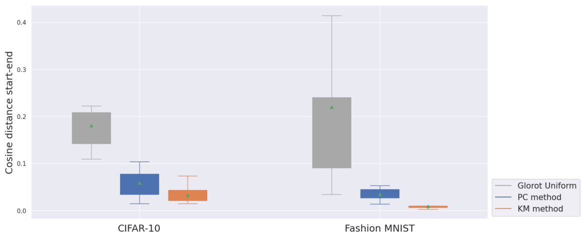

Figure 9: Average cosine distances for the set of filters between their initial configuration obtained by each of the initialization methods and their final configuration after training via Gradient Descent and backpropagation.

cacy of PC initialization in enhancing CNN training efficiency.

While our study offers valuable insights, it is important to acknowledge its limitations. Future research endeavors should seek to explore the generalizability of our findings across all convolutional layers and for different network architectures. It is anticipated that consistently applying Principal Components for initialization across all convolutional layers could result in more significant gains, both in terms of final performance of the network and convergence speed.

Competing interests

The authors have declared that no competing interests exist.

Authors' contribution

The authors confirm contribution to the paper as follows: FER: Conceptualization, Data curation, Investigation, Writing-Original draft preparation, Writing-Reviewing and Editing; FR: Investigation, Methodology, Supervision, Writing-Reviewing and Editing; FQ: Data curation, Investigation, Methodology, Supervision, Writing-Reviewing and Editing. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the authors used OpenAI's ChatGPT4 to refine text quality. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015. doi: 10.1109/TIP.2015.2475625.

[2] X.-D. Ren, H.-N. Guo, G.-C. He, X. Xu, C. Di, and S.-H. Li, "Convolutional neural network based on principal component analysis initialization for image classification," in *Proc. IEEE First Int. Conf. Data Science Cyberspace (DSC)*, 2016, pp. 329–334. doi: 10.1109/DSC.2016.18.

[3] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, "Semisupervised PCA convolutional network for vehicle type classification," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8267–8277, Aug. 2020. doi: 10.1109/TVT.2020.3000306.

[4] C. Zhang, M. Mei, M. Zhuolin, J. Zhang, A. Deng, et al., "PLDANet: Reasonable combination of PCA and LDA convolutional networks," *Int. J. Comput. Commun. Control*, vol. 17, no. 2, Apr. 2022. doi: 10.15837/ijccc.2022.2.4541.

[5] L. Sun, Y. Liu, S. Chen, B. Luo, Y. Li, and C. Liu, "Pig detection algorithm based on sliding windows and PCA convolution," *IEEE Access*, vol. 7, pp. 44229–44238, 2019. doi: 10.1109/ACCESS.2019.2907748.

[6] L. Tian, C. Fan, Y. Ming, and Y. Jin, "Stacked PCA network (SPCANet): An effective deep learning for face recognition," in *Proc. Int. Conf. DSP*, 2015, pp. 1039–1043. doi: 10.1109/ICDSP.2015.7252036.

[7] H. Y. Khaw, F. C. Soon, J. H. Chuah, and C. O. Chow, "Image noise types recognition using convolutional neural network with principal components analysis," *IET Image Process.*, vol. 11, no. 12, pp. 1238–1245, Dec. 2017. doi: 10.1049/iet-ipr.2017.0374.

[8] A. Coates and A. Y. Ng, "Learning feature representations with K-means," in *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, and K.-R. Muller, Eds., 2nd ed. Berlin, Germany: Springer, LNCS 7700, 2012, pp. 561–580.

Citation: F. Rabinovich, F. Ronchetti and F. Quiroga. *Towards more efficient initialization methods for Convolutional Neural Networks via K-Means and Principal Components*. Journal of Computer Science & Technology, vol. 25, no. 1, pp. 44–52, 2025.
DOI: 10.24215/16666038.25.e04
Received: April 15, 2024 **Accepted:** October 25, 2024.