

CHANNEL REDUNDANCY AND OVERLAP IN CONVOLUTIONAL NEURAL NETWORKS WITH CHANNEL-WISE NNK GRAPHS

David Bonet^{*} Antonio Ortega[†] Javier Ruiz-Hidalgo^{*} Sarath Shekizhar[†]

^{*} Universitat Politècnica de Catalunya, Barcelona, Spain

[†] University of Southern California, Los Angeles, USA

ABSTRACT

Feature spaces in the deep layers of convolutional neural networks (CNNs) are often very high-dimensional and difficult to interpret. However, convolutional layers consist of multiple channels that are activated by different types of inputs, which suggests that more insights may be gained by studying the channels and how they relate to each other. In this paper, we first analyze theoretically channel-wise non-negative kernel (CW-NNK) regression graphs, which allow us to quantify the overlap between channels and, indirectly, the intrinsic dimension of the data representation manifold. We find that redundancy between channels is significant and varies with the layer depth and the level of regularization during training. Additionally, we observe that there is a correlation between channel overlap in the last convolutional layer and generalization performance. Our experimental results demonstrate that these techniques can lead to a better understanding of deep representations.

Index Terms— Convolutional neural networks, channel redundancy, graph construction, intrinsic dimension, interpretability

1. INTRODUCTION

Convolutional neural networks (CNNs) are among the most successful models for supervised classification [1, 2]. CNNs sequentially process inputs through blocks of convolutional layers followed by non-linearity, normalization and pooling layers, and fully connected softmax layers for classification. Each convolutional layer consists of multiple filters that are applied in parallel to a common input, resulting in very high dimensional data representations, formed by the aggregation of multiple *channels*, where each channel is the part of the output that corresponds to a single convolutional filter. It is generally accepted that initial layers encode basic features [3], e.g., edges and corners, while the features learned in deeper layers are harder to interpret. Hence, the gap in the application of deep learning models in more sensitive areas remains significant and our understanding of how these models generalize is still limited [4].

Interest in the geometric properties of high dimensional datasets has led to various attempts to define the *intrinsic dimension* (ID) of the data. While multiple definitions are possible [5], the general idea is that, for a given type of data representation, ID is related to the number of parameters needed to describe a dataset accurately [6]. Measuring ID helps to understand how neural networks learn to transform data effectively [7], even with significant overparameterization. Further, specific local ID characteristics are associated with adversarial robustness [8] or generalization for noisy labels [9].

This work was partially supported by the Spanish Research Agency (AEI) under project PID2020-117142GB-I00 of the call MCIN/AEI/10.13039/501100011033, and by a DARPA grant (FA8750-19-2-1005) in Learning with Less Labels (LwLL) program and by NSF (CCF-2009032).

Graph properties have also been used to interpret neural network performance [10, 11], latent space geometry [12, 13], and to improve model robustness [14]. However, none of these approaches used channel-wise information to study how the feature vectors in different channels relate to each other.

In this paper, we study channel redundancy using graphs constructed from training data and we propose a novel way to estimate the ID of data representations based on the level of information overlap across channels. To this aim, we construct channel-wise non-negative kernel (NNK) regression graphs [15] using data representations at the output of each convolutional layer. NNK graphs take advantage of the local geometry to sparsify a K -nearest neighbor (KNN) graph. Further, NNK neighborhoods are stable for large enough K , and their size is indicative of the ID of the manifold the data belongs to [15]. This method has been shown to provide advantages for semi-supervised learning, image representation [16], and generalization estimation [10]. We first proposed a channel-wise approach for NNK graphs in [17], where information from multiple channels was used to estimate generalization and perform early stopping without requiring a validation set.

We analyze the conditions that guarantee that two nodes that are NNK graph neighbors in two channels, i.e., overlap, will also be neighbors in an NNK graph constructed by aggregating the two channels. These results allow us to quantify the level of similarity between data points (across multiple channels) without a combinatorial explosion in the number of graphs to be constructed. From our experimental results, we find that channel redundancy in CNNs is significant and varies according to the layer depth and the level of training regularization. Our method for assessing the redundancy in CNNs can be leveraged to improve or create new channel pruning techniques. We also observe that CW-NNK overlap in the last convolutional layer is strongly correlated with generalization performance. Our measure of overlap can be combined with [17] to improve generalization estimation and efficiency for large scale problems.

2. RELATED WORK

Recently, [7, 18] study ID for several CNN architectures and show that the IDs of deep representations are notably smaller than the nominal dimension of the features. More generally, these works observe that the ID increases in the initial layers and then decreases closer to the final layers. [7, 18] attribute the initial increase in ID to the fact that initial layers perform low-level pre-processing and feature extraction, many of which are task-independent. However, in the later layers, only task-relevant features are learned, and the ability of the model to compress the dimensionality of data in the last layers is indicative of the model's generalization.

Given the large number of channels that comprise each layer of state-of-the-art CNNs, and the computation and memory they re-

quire, there has been a growing interest in understanding how these channels are related to each other. In particular, redundancy between channels has been studied based on detecting pairs of filters with opposite phase [19], using guided back-propagation patterns [20], and analyzing filter correlation [21]. Channel similarity estimates can be used to compress models by pruning channels that are similar in a layer [22, 23] or by leveraging the structural redundancy of the channels in a layer [24]. Orthogonality constraints have also been proposed to de-correlate channels during training [20, 21, 25].

3. PRELIMINARIES

3.1. Notation

We denote scalars, vectors, and matrices using lowercase (e.g., x and θ), lowercase bold (e.g., \mathbf{x} and $\boldsymbol{\theta}$), and uppercase bold (e.g., \mathbf{K} and Φ) letters, respectively. Subscript i denotes the i^{th} data point associated with the feature vector \mathbf{x}_i . We use a double subscript to index and denote a channel and related measures. For example, a feature vector $\mathbf{x}_i \in \mathbb{R}^D$ obtained as the concatenation of C channel subvectors $\mathbf{x}_{i_c} \in \mathbb{R}^{D_c}$ where $\sum_{c=1}^C D_c = D$. The indices of the KNN and NNK neighbors of a data point \mathbf{x}_i in the aggregate space are denoted by sets $\mathcal{S}(\mathbf{x}_i)$ and $\mathcal{N}(\mathbf{x}_i)$, while $\mathcal{S}(\mathbf{x}_{i_c})$ and $\mathcal{N}(\mathbf{x}_{i_c})$ are the neighborhood sets for the channel subvector \mathbf{x}_{i_c} , respectively.

3.2. Non-Negative Kernel (NNK) regression graphs

Given a set of N data points represented by feature vectors $\mathbf{x} \in \mathbb{R}^D$, a graph is constructed by connecting each data point (node) to similar data points, so that the weight of an edge between two nodes is based on the similarity of the data points, with the absence of an edge (a zero weight) denoting least similarity. Conventionally, one defines similarity between data points using positive definite kernels [26] such as the Gaussian kernel with bandwidth σ :

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2) \quad (1)$$

Unlike weighted KNN and ϵ -neighborhood graphs [27] that are sensitive to the choice of hyperparameters K/ϵ , NNK graphs [15] are suggested as a principled approach to graph construction based on a signal representation view.

An advantage of NNK over KNN, which can be viewed as a thresholding-based representation, is its robustness to the choice of parameter K . While KNN is still used as an initialization, NNK performs a further optimization akin to orthogonal matching pursuits [28] in kernel space, resulting in a *stable* representation (even as K chosen for initialization grows, the number of NNK neighbors remains nearly constant) with the added advantage of having a *geometric* interpretation based on the Kernel Ratio Interval (KRI) theorem [15]:

$$\mathbf{K}_{j,k} < \frac{\mathbf{K}_{i,j}}{\mathbf{K}_{i,k}} < \frac{1}{\mathbf{K}_{j,k}} \quad (2)$$

where $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. In the case of the Gaussian kernel (1), considering the edge θ_{ij} connecting node i and node j , we can define a hyperplane normal to the edge direction. The hyperplane divides the space in two, a region R_{ij} that contains \mathbf{x}_i , and its complement \bar{R}_{ij} . Then, a third node k will be connected to i only if $\mathbf{x}_k \in R_{ij}$. If $\mathbf{x}_k \in \bar{R}_{ij}$, $\theta_{ik} = 0$ and we say that k has been eliminated by the hyperplane created by j . The inductive application of the KRI theorem to \mathbf{x}_i leads to a convex polytope around node i disconnecting all the other points outside the polytope. Intuitively, NNK ignores data that are further away along a *similar* direction as an already chosen point and looks for neighbors in an *orthogonal* direction.

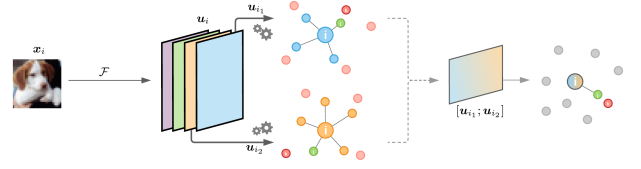


Fig. 1: CW-NNK graphs: Given an input \mathbf{x}_i , we forward it to a layer of interest of a CNN, with transformation \mathcal{F} mapping \mathbf{x}_i to the layer feature maps \mathbf{u}_i . We construct CW-NNK graphs by comparing the intermediate representations of \mathbf{x}_i and a set of training images.

4. CHANNEL-WISE NNK (CW-NNK) GRAPHS

In intermediate representations of CNNs, very high-dimensional feature vectors can be seen as the aggregation of multiple subvectors, i.e., channels. We propose to construct channel-wise NNK (CW-NNK) graphs, i.e., use the data representations of the dataset in each individual channel of a specific layer to construct C independent graphs, as described in Figure 1. The geometric properties of CW-NNK graphs, derived in Section 4.1, allow us to analyze *efficiently* the amount of overlap between channels. Independence between the sets of channel-wise NNK neighbors can serve as an indication that each channel specializes to different features, whereas an overlap between the sets corresponds to redundancy in the features obtained. Furthermore, studying the size and the similarity between CW-NNK neighborhoods we can develop a better understanding of why the ID can be relatively low even in a high dimensional space. In this context, the average number of NNK neighbors can be viewed as a characteristic parameter of the data, and following the ideas in [5], we could say that a smaller number of neighbors (more overlap between channels) is indicative of lower ID.

4.1. CW-NNK graphs analysis

A theoretical analysis that builds on the graphs obtained from the NNK optimization in individual channels allows us to infer relevant properties of the graph that we would obtain in the aggregate high dimensional feature space of multiple channels. For simplicity, we consider a scenario with two channels and their aggregate:

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{x}_{i_1} \\ \mathbf{x}_{i_2} \end{bmatrix} \in \mathbb{R}^D$$

where $\mathbf{x}_{i_1} \in \mathbb{R}^{D_1}$ and $\mathbf{x}_{i_2} \in \mathbb{R}^{D_2}$ are the two channels forming \mathbf{x}_i and $D_1 + D_2 = D$, but all the results presented in this section can be extended to the multiple channel case. We analyze the properties of the set of NNK neighbors $\mathcal{N}(\mathbf{x}_i)$ in the aggregate space given the neighbor sets of each channel, $\mathcal{N}(\mathbf{x}_{i_1})$ and $\mathcal{N}(\mathbf{x}_{i_2})$. An extended analysis of this method and its applications is carried out in [29].

Theorem 4.1. *If $j \in \mathcal{N}(\mathbf{x}_{i_1}) \cap \mathcal{N}(\mathbf{x}_{i_2})$, and $j \in \mathcal{S}(\mathbf{x}_i)$, then $j \in \mathcal{N}(\mathbf{x}_i)$.*

Proof. Consider a three-node scenario with j , k , and query i . We know that $\theta_{i_1,j_1} > 0$ and $\theta_{i_2,j_2} > 0$. Based on the KRI theorem (2):

$$\theta_{i_1,j_1} > 0 \iff \mathbf{K}_{j_1,k_1} < \frac{\mathbf{K}_{i_1,j_1}}{\mathbf{K}_{i_1,k_1}} \quad (3)$$

$$\theta_{i_2,j_2} > 0 \iff \mathbf{K}_{j_2,k_2} < \frac{\mathbf{K}_{i_2,j_2}}{\mathbf{K}_{i_2,k_2}} \quad (4)$$

This implies that j is not eliminated by any other hyperplane created by a third point k in any of the two channels (and thus it is an NNK neighbor in both channels).

Then, in the aggregate space we have $K_{i,j} = K_{i_1,j_1} K_{i_2,j_2}$ since $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{c=1}^C \|\mathbf{x}_{i_s} - \mathbf{x}_{j_s}\|^2 / 2\sigma^2\right)$ and therefore $j \in \mathcal{N}(\mathbf{x}_i)$ if

$$\theta_{i,j} > 0 \iff K_{j_1,k_1} K_{j_2,k_2} < \frac{K_{i_1,j_1} K_{i_2,j_2}}{K_{i_1,k_1} K_{i_2,k_2}}. \quad (5)$$

Considering (3) and (4), let $K_{j_1,k_1} = a$, $K_{j_2,k_2} = b$, $\frac{K_{i_1,j_1}}{K_{i_1,k_1}} = a + \gamma$ and $\frac{K_{i_2,j_2}}{K_{i_2,k_2}} = b + \epsilon$. Then, we can substitute terms in (5):

$$ab < (a + \gamma)(b + \epsilon), \text{ so that}$$

$$a\epsilon + b\gamma + \gamma\epsilon > 0 \iff \theta_{i,j} > 0,$$

where $0 \leq a, b \leq 1$ and $\gamma, \epsilon > 0$. Therefore, $\theta_{i,j} > 0$ and

$$j \in \mathcal{N}(\mathbf{x}_i) \iff j \in \mathcal{N}(\mathbf{x}_{i_1}), j \in \mathcal{N}(\mathbf{x}_{i_2}), j \in \mathcal{S}(\mathbf{x}_i) \quad (6)$$

There are no extra conditions for the initial sets of neighbors $\mathcal{S}(\mathbf{x}_{i_1})$ and $\mathcal{S}(\mathbf{x}_{i_2})$. First, we know that $j \in \mathcal{S}(\mathbf{x}_{i_1}) \cap \mathcal{S}(\mathbf{x}_{i_2})$. However, if there exists a third point $k \in \mathcal{S}(\mathbf{x}_i)$ that has not been selected in the initialization of one of the channels, e.g., $k \in \mathcal{S}(\mathbf{x}_{i_1})$ and $k \notin \mathcal{S}(\mathbf{x}_{i_2})$, we also want to verify (5) for this third point, but K_{j_2,k_2} is unknown. We do know, however, that $0 \leq K_{i,j} \leq 1$, and $K_{i_2,k_2} < K_{i_2,j_2}$ because $j \in \mathcal{S}(\mathbf{x}_{i_2})$ and $k \notin \mathcal{S}(\mathbf{x}_{i_2})$. Then, $K_{j_2,k_2} < \frac{K_{i_2,j_2}}{K_{i_2,k_2}}$ so (4) is fulfilled and (3) is also fulfilled since $k \in \mathcal{S}(\mathbf{x}_{i_1})$, therefore (5) is also fulfilled. Also, condition $j \in \mathcal{S}(\mathbf{x}_i)$ can be easily met by selecting $\mathcal{S}(\mathbf{x}_i) = \mathcal{S}(\mathbf{x}_{i_1}) \cup \mathcal{S}(\mathbf{x}_{i_2})$. \square

Corollary 4.1.1. $\mathcal{N}(\mathbf{x}_{i_1}) \cap \mathcal{N}(\mathbf{x}_{i_2}) \subseteq \mathcal{N}(\mathbf{x}_i)$ if $\mathcal{N}(\mathbf{x}_{i_1}) \cap \mathcal{N}(\mathbf{x}_{i_2}) \subseteq \mathcal{S}(\mathbf{x}_i)$.

Any point j such that $j \in \mathcal{N}(\mathbf{x}_{i_1}) \cap \mathcal{N}(\mathbf{x}_{i_2})$ verifies both (3) and (4). Thus, (5) is verified for j as well. By constructing only C graphs, i.e., one per channel, Theorem 4.1 and Corollary 4.1.1 allow us to determine, from the intersections of neighborhoods in different channels ($j \in \mathcal{N}(\mathbf{x}_{i_1}) \cap \mathcal{N}(\mathbf{x}_{i_2})$) properties of the neighborhood in the aggregate ($\mathcal{N}(\mathbf{x}_i)$) without having to construct the aggregate NNK graph, which involves additional complexity. Thus, C graph constructions provide a lower bound on the overall overlap, without the need to construct $\sum_{n=2}^C \binom{C}{n}$ graphs to find the neighbors in all possible aggregations of channels, which would be impractical.

Theorem 4.2. In a three-node scenario, if $j \in \mathcal{N}(\mathbf{x}_{i_1}) \cap \mathcal{N}(\mathbf{x}_{i_2})$ and we have that $k \notin \mathcal{N}(\mathbf{x}_{i_1})$ and $k \notin \mathcal{N}(\mathbf{x}_{i_2})$, then j eliminates k in both channels and $k \notin \mathcal{N}(\mathbf{x}_i)$.

Proof. If $\theta_{i_1,k_1} = 0$ because of the hyperplane created by $\theta_{i_1,j_1} > 0$, and $\theta_{i_2,k_2} = 0$ because of the hyperplane created by $\theta_{i_2,j_2} > 0$, from the KRI theorem (2):

$$\theta_{i_1,k_1} = 0 \iff \frac{1}{K_{j_1,k_1}} < \frac{K_{i_1,j_1}}{K_{i_1,k_1}} \quad (7)$$

$$\theta_{i_2,k_2} = 0 \iff \frac{1}{K_{j_2,k_2}} < \frac{K_{i_2,j_2}}{K_{i_2,k_2}} \quad (8)$$

Then, in the aggregate, $k \in \mathcal{N}(\mathbf{x}_i)$ if

$$\theta_{i,k} > 0 \iff \frac{1}{K_{j_1,k_1} K_{j_2,k_2}} > \frac{K_{i_1,j_1} K_{i_2,j_2}}{K_{i_1,k_1} K_{i_2,k_2}}. \quad (9)$$

Considering (7) and (8), let $\frac{1}{K_{j_1,k_1}} = a$, $\frac{1}{K_{j_2,k_2}} = b$, $\frac{K_{i_1,j_1}}{K_{i_1,k_1}} = a + \gamma$ and $\frac{K_{i_2,j_2}}{K_{i_2,k_2}} = b + \epsilon$:

$$ab > (a + \gamma)(b + \epsilon)$$

$$a\epsilon + b\gamma + \gamma\epsilon < 0$$

where $0 \leq a, b \leq 1$ and $\gamma, \epsilon > 0$. Thus, $\theta_{i,k} = 0$ and $k \notin \mathcal{N}(\mathbf{x}_i)$.

In words, if a point k is eliminated by a hyperplane created by the same point j in all subvectors, k will not be connected as an NNK neighbor in the aggregate space. \square

Lemma 4.3. If $k \notin \mathcal{N}(\mathbf{x}_{i_1})$ and $k \notin \mathcal{N}(\mathbf{x}_{i_2})$, but in a three-node scenario with i and j , j only eliminates k in one channel, $\theta_{i_1,k_1} = 0$ and $\theta_{i_2,k_2} > 0$, then it is possible that $k \in \mathcal{N}(\mathbf{x}_i)$.

Proof. Consider node k which is not selected as an NNK neighbor in any of the two channels. In a three-node scenario with points i and j in channel 1, $\theta_{i_1,k_1} = 0$ because of the hyperplane created by θ_{i_1,j_1} . But in a three-node scenario with the same points in channel 2, $\theta_{i_2,k_2} > 0$, so k is eliminated from $\mathcal{N}(\mathbf{x}_{i_2})$, not because of the hyperplane created by θ_{i_2,j_2} , but by the hyperplane created by a fourth point q , denoted by its normal $\theta_{i_2,q_2} > 0$.

Then, consider (9) and let $\frac{1}{K_{j_1,k_1}} = a$, $\frac{1}{K_{j_2,k_2}} = b + \epsilon$, $\frac{K_{i_1,j_1}}{K_{i_1,k_1}} = a + \gamma$ and $\frac{K_{i_2,j_2}}{K_{i_2,k_2}} = b$. Then,

$$a(b + \epsilon) > (a + \gamma)b$$

$$a\epsilon > b\gamma$$

where $0 \leq a, b \leq 1$ and $\gamma, \epsilon > 0$ considering (7) and (8). Therefore, a node k that is not an NNK neighbor of i in any channel, but has not been eliminated by the same node j in all channels can still be selected as NNK neighbor in the aggregate space. \square

5. EXPERIMENTS

We use CW-NNK graphs to study the overlap between channels within a CNN layer and how this overlap affects the dimensionality of the layer. We consider a 7 layer CNN model composed of 6 convolutional layers of 16 depth channels with ReLU activations, interleaving max-pooling every 2 layers and 1 fully connected softmax layer. A dropout layer [30] is added after the ReLU activations for regularization. Models are trained on the well-known CIFAR-10 dataset [31] for 200 epochs with the Adam optimizer with a learning rate of 0.001, and a batch size of 50. A class-balanced subset of 1000 randomly sampled data points from the original train set is used to construct CW-NNK graphs. Given a query data point, the rest of the subset is used as train data for graph construction. The intermediate representations of the input data (feature maps) at the output of a convolutional block are used as feature vectors.

5.1. Channel redundancy across CNN layers

Section 4.1 developed properties showing that when an NNK neighbor is common across two channels it will also be an NNK neighbor in the aggregation of those two channels, without having to construct the graph for the aggregation. We next show empirically that these overlaps of neighborhoods in two channels for a given instance do happen in practice. For each convolutional layer, we compute the ratio between the number of pairwise channel NNK intersections and the average number of NNK neighbors per channel (CW-NNK overlap), and we average it for all the data points in the subset. Figure 2 shows that there is indeed a significant number of neighbors overlapping in different channels in the shallower layers. The CW-NNK overlap is reduced in the deeper layers, reaching a minimal point in the last convolutional layer, denoting more independence between channels. This confirms what has been observed in other works that study channel redundancy by different means [19, 22].

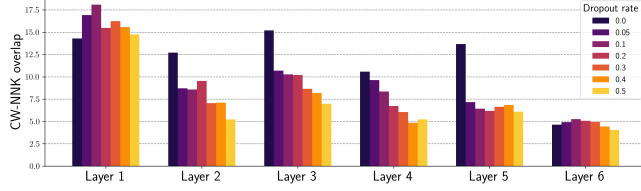


Fig. 2: Number of pairwise channel NNK intersections to average number of NNK neighbors per channel ratio (CW-NNK overlap) in each layer for different dropout rates.

We also observe that channel redundancy in the intermediate layers (second to penultimate layer) depends on the level of model regularization. Higher dropout rates lead to channels of intermediate layers having less overlap, while lack of regularization results in higher overlap. These observations are promising for further improvement of channel pruning techniques and redundancy constrains to improve training. However, less overlap between channels does not necessarily imply higher generalization. As shown in [21], over-specialized channels can lead to worse performance [21], and a controlled level of redundancy is preferable to maintain performance. We discuss the CW-NNK overlap in the last convolution layer and its relation to ID and generalization next.

5.2. CW-NNK overlap and ID as generalization indicators

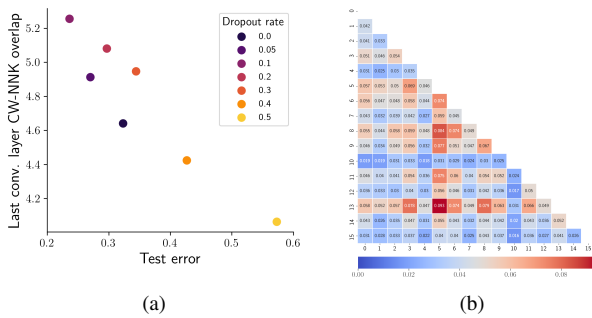


Fig. 3: (a) Last convolutional layer CW-NNK overlap as a function of model test error. (b) CW-NNK overlap for all pairs of channels in the last convolutional layer.

As observed in [7], the ID in the last layer on the training set is a strong predictor for generalization performance. Between multiple models, lower IDs strongly correlate with higher classification accuracy. In general, the information overlap between channels plays a key role in the estimation of the overall ID of the data manifold. The ID of an aggregate of multiple channels may be additive if there is no overlap between them, while if high overlap is observed, the ID of the aggregate could be similar to the ID of a single channel, related to the average number of CW-NNK neighbors. We compare the CW-NNK overlap in the last convolutional layer with the test error of our model using different dropout rates. Since the models are trained for 200 epochs without early stopping, lack of regularization leads to overfitting to the training data, i.e., high test error. However, an excessive regularization, i.e., high dropout rates, can cause underfitting [30], leading to both train and test errors to be high. Figure 3a shows how the CW-NNK overlap strongly correlates with test error, demonstrating that a high CW-NNK overlap in the last convolutional layer is indicative of a lower ID of the aggregate feature vector, and thus, of higher generalization performance. The last convolutional layer results do not contradict the observations in Section 5.1, since

less redundancy in intermediate layers, owing to regularization, does not necessarily affect generalization performance.

5.3. CW-NNK overlap distribution and examples

Given that there is a considerable overlap of neighbors between different channels in a layer, we want to observe whether these intersections are randomly distributed among all channels or follow a pattern. Figure 3b represents the CW-NNK overlap for all pairs of channels in the last convolutional layer of the trained model that achieves highest test accuracy, with a dropout rate of 0.1. It is important to note that with only C graphs we can observe similarities beyond pairs of channels (i.e., neighbors that intersect in trios, quartets, etc., of channels) and in this matrix we represent all pairwise combinations of channels that occur. We observe that, in both layers, most of the intersections are concentrated in a few channels, which capture similar features. On the other hand, other channels are very specialized and independent, and since the CW-NNK overlap is very low, the captured features differ from those of other channels.

A major advantage of this method is that the feature overlap can be interpreted directly from the input images. Figure 4 shows the sets of neighbors of three channels of the last convolutional layer for a given query. We can observe how three images appear in the neighborhood of the first two channels. All the other neighbors selected in these channels share similar general features (vehicles, light or blue backgrounds, etc.). However, the bottom channel does not overlap with the other two, and we can confirm that the channel captures very different features from the input data, selecting data points from other non-vehicle classes as neighbors with very different features.

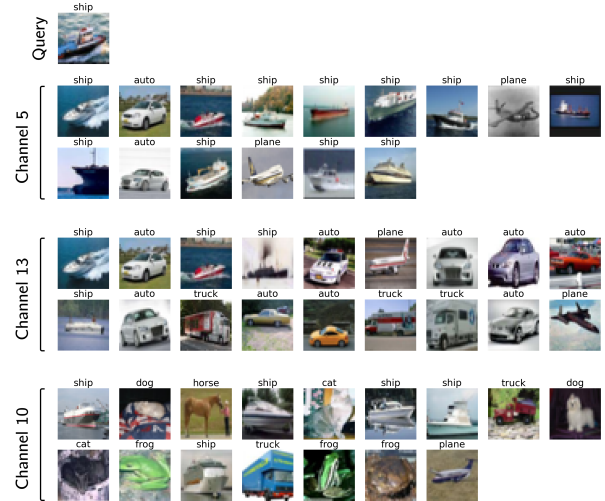


Fig. 4: Sets of neighbors for three channels of the last convolutional layer for a given query. Channels 5 and 13 overlap and share similar features while channel 10 captures very different features.

6. CONCLUSION

We study the conditions that allow computing the overlap between channels efficiently through CW-NNK graphs, and which effect it has on the overall ID of the data. We discuss the impact of regularization and layer depth on channel redundancy. Finally, we observe that CW-NNK overlap in the last convolutional layer is strongly correlated with generalization performance. Future work will consider using the level of overlap for channel pruning and to improve generalization estimation for channel-wise early stopping [17].

7. REFERENCES

- [1] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.
- [2] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, “Designing network design spaces,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10428–10436.
- [3] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [4] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.
- [5] P. Campadelli, E. Casiraghi, C. Ceruti, and A. Rozza, “Intrinsic dimension estimation: Relevant techniques and a benchmark framework,” *Mathematical Problems in Engineering*, vol. 2015, 2015.
- [6] R. Bennett, “The intrinsic dimensionality of signal collections,” *IEEE Transactions on Information Theory*, vol. 15, no. 5, pp. 517–525, 1969.
- [7] A. Ansuini, A. Laio, J. H. Macke, and D. Zoccolan, “Intrinsic dimension of data representations in deep neural networks,” *Advances in Neural Information Processing Systems*, vol. 32, pp. 6111–6122, 2019.
- [8] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, and J. Bailey, “Characterizing adversarial subspaces using local intrinsic dimensionality,” in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018, pp. 1–15.
- [9] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. Erfani, S. Xia, S. Wijewickrema, and J. Bailey, “Dimensionality-driven learning with noisy labels,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 3355–3364.
- [10] S. Shekizhar and A. Ortega, “DeepNNK: Explaining deep models and their generalization using polytope interpolation,” *arXiv preprint arXiv:2007.10505*, 2020.
- [11] V. Gripon, A. Ortega, and B. Girault, “An inside look at deep neural networks using graph signal processing,” in *Proceedings of ITA*, February 2018, pp. 1–9.
- [12] C. Lassance, V. Gripon, and A. Ortega, “Representing deep neural networks latent space geometries with graphs,” *Algorithms*, vol. 14, no. 2, 2021.
- [13] C. Lassance, M. Bontonou, G. B. Hacene, V. Gripon, J. Tang, and A. Ortega, “Deep geometric knowledge distillation with graphs,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8484–8488.
- [14] C. Lassance, V. Gripon, and A. Ortega, “Laplacian networks: Bounding indicator function smoothness for neural networks robustness,” *APSIPA Transactions on Signal and Information Processing*, vol. 10, 2021.
- [15] S. Shekizhar and A. Ortega, “Graph construction from data by non-negative kernel regression,” in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3892–3896.
- [16] S. Shekizhar and A. Ortega, “Efficient graph construction for image representation,” in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1956–1960.
- [17] D. Bonet, A. Ortega, J. Ruiz-Hidalgo, and S. Shekizhar, “Channel-wise early stopping without a validation set via NNK polytope interpolation,” *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (AP-SIPA ASC)*, 2021.
- [18] S. Recanatesi, M. Farrell, M. Advani, T. Moore, G. Lajoie, and E. Shea-Brown, “Dimensionality compression and expansion in deep neural networks,” *arXiv preprint arXiv:1906.00443*, 2019.
- [19] W. Shang, K. Sohn, D. Almeida, and H. Lee, “Understanding and improving convolutional neural networks via concatenated rectified linear units,” in *international conference on machine learning*. PMLR, 2016, pp. 2217–2225.
- [20] J. Wang, Y. Chen, R. Chakraborty, and S. X. Yu, “Orthogonal convolutional neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11505–11515.
- [21] K. Kahatapitiya and R. Rodrigo, “Exploiting the redundancy in convolutional filters for parameter reduction,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1410–1420.
- [22] Y. He, X. Zhang, and J. Sun, “Channel pruning for accelerating very deep neural networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1389–1397.
- [23] M. Jaderberg, A. Vedaldi, and A. Zisserman, “Speeding up convolutional neural networks with low rank expansions,” in *Proceedings of the British Machine Vision Conference. BMVA Press*, 2014.
- [24] Z. Wang, C. Li, and X. Wang, “Convolutional neural network pruning with structural redundancy reduction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14913–14922.
- [25] P. Rodríguez, J. Gonzalez, G. Cucurull, J. M. Gonfaus, and X. Roca, “Regularizing cnns with locally constrained decorrelations,” *International Conference on Learning Representations (ICLR)*, 2017.
- [26] N. Aronszajn, “Theory of reproducing kernels,” *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.
- [27] V. Chvátal and P. Hammer, “Aggregations of inequalities,” *Studies in Integer Programming, Annals of Discrete Mathematics*, vol. 1, pp. 145–162, 1977.
- [28] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [29] D. Bonet, “Improved neural network generalization using channel-wise nnk graph constructions,” B.S. Thesis, Universitat Politècnica de Catalunya, Spain, 2021.
- [30] 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.
- [31] A. Krizhevsky, G. Hinton, et al., “Learning multiple layers of features from tiny images,” *Technical report, University of Toronto*, 2009.