i-REVNET: DEEP INVERTIBLE NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

It is widely believed that the success of deep convolutional networks is based on progressively discarding uninformative variability about the input with respect to the problem at hand. This is supported empirically by the difficulty of recovering images from their hidden representations, in most commonly used network architectures. In this paper we show that this loss of information is *not* a necessary condition to learn representations that generalize well on complicated problems, such as ImageNet. Via a cascade of homeomorphic layers, we build the i-RevNet, a network that can be fully inverted up to the final projection onto the classes, i.e. no information is discarded. Building an invertible architecture is difficult, for example, because the local inversion is ill-conditioned, we overcome this by providing an explicit inverse. An analysis of i-RevNets learned representations suggests an explanation of the good accuracy by a progressive contraction and linear separation with depth. To shed light on the nature of the model learned by the i-RevNet we reconstruct linear interpolations between natural images representations.

1 INTRODUCTION

A CNN may be very effective in classifying images of all sorts (He et al., 2016; Krizhevsky et al., 2012), but the cascade of linear and nonlinear operators reveals little about the contribution of the internal representation to the classification. The learning process is characterized by a steady reduction of large amounts of uninformative variability in the images while simultaneously revealing the essence of the visual class. It is widely believed that this process is based on progressively discarding uninformative variability about the input with respect to the problem at hand (Dosovitskiy & Brox, 2016; Mahendran & Vedaldi, 2016; Shwartz-Ziv & Tishby, 2017; Achille & Soatto, 2017). However, the extent to which information about the input signal is discarded is lost somewhere in the intermediate non-linear processing steps. In this paper, we aim to provide insight in the variability reduction process by proposing an invertible convolutional network, that does not discard any information about the input.

The difficulty to recover images from their hidden representations is found in many commonly used network architectures (Dosovitskiy & Brox, 2016; Mahendran & Vedaldi, 2016). This poses the question if a substantial loss of information is necessary for successful classification. We show no information needs to be discarded. By using homeomorphic layers, the invariance can be built only at the very last layer via a projection.

In Shwartz-Ziv & Tishby (2017), minimal sufficient statistics are proposed as a candidate to explain the reduction of variability. Tishby & Zaslavsky (2015) introduces the information bottleneck principle which states that an optimal representation must reduce the mutual information between an input and its representation to reduce as much uninformative variability as possible. At the same time, the network should maximize the mutual information between the desired output and its representation to effectively preserve each class from collapsing onto other classes. The effect of the information bottleneck was demonstrated on small datasets in Shwartz-Ziv & Tishby (2017); Achille & Soatto (2017). However, in this work, we show it is not a necessary condition and we build a cascade of homeomorphic layers, which preserves the mutual information between input and hidden representation and shows that the loss of information can only occur at the final layer. This way we demonstrate that a loss of information can be avoided even for large-scale problems like ImageNet.

One way to reduce variability is to progressively contract them with respect to a meaningful ℓ^2 metric in the intermediate representations.

Several works (Oyallon, 2017; Zeiler & Fergus, 2014) observed a phenomenon of progressive separation and contraction in non-invertible networks on limited datasets. Those progressive improvements can be interpreted as the creation of progressively stronger invariants for classification. Ideally, the contraction should not be too brutal to avoid removing important information from the intermediate signal. This shows that a good trade-off between discriminability and invariance has to be progressively built. In this paper, we extend some findings of Zeiler & Fergus (2014); Oyallon (2017) to ImageNet and, most importantly, show that a loss of information is not necessary for observing a progressive contraction.

The duality between invariance and separation of the classes is discussed in Mallat (2016). Here, intra-class variabilities are modeled as Lie groups that are processed by performing a parallel transport along those symmetries. Filters are adapted through learning to the specific bias of the dataset and avoid to contract along discriminative directions. However, using groups beyond the Euclidean case for image classification is hard. Mainly because groups associated with abstract variabilities are difficult to estimate due to their high-dimensional nature, as well as the appropriate degree of invariance required. An illustration of this framework on the Euclidean group is given by the scattering transform (Mallat, 2012), which builds invariance to small translations while being recoverable to a certain extent. In this work, we introduce a network that cannot discard any information except at the last layer, while we demonstrate numerically progressively contraction and separating the signal classes.

We introduce the i-RevNet, an invertible deep network. i-RevNets retain all information about the input signal in any of their intermediate representations up until the last layer. Our architecture builds upon the recently introduced RevNet (Gomez et al., 2017), where we replace the non-invertible components of the original RevNets by invertible ones. i-RevNets achieve the same performance on Imagenet compared to similar non-invertible RevNet and ResNet architectures (Gomez et al., 2017; He et al., 2016). In this framework, we show that loss of information is not a necessary condition to learn representations that generalize well to unseen data.
To shed light on the mechanism underlying the generalization-ability of the learned representation, we show that i-RevNets progressively separate and contract signals with depth. Our results are evidence for an effective reduction of variability through a contraction with a recoverable input.

2 Related Work

Several recent works show that significant information about the input images is lost with depth in successful Imagenet classification CNNs (Dosovitskiy & Brox, 2016; Mahendran & Vedaldi, 2016). To understand the loss of information, the references propose to invert the representations by means of learned or hand-engineered priors. The approximate inversions indicate increased geometric and photometric invariance with depth. Multiple other works report progressive properties of deep networks that may be linked to discarded information in the representations as well, such as linearization (Radford et al., 2015), linear separability (Zeiler & Fergus, 2014), contraction (Oyallon, 2017) and low-dimensional embeddings (Aubry & Russell, 2015). However, it is not clear from above observations if the loss of information is a necessity for the observed progressive phenomena. In this work, we show that progressive separation and contraction can be obtained while at the same time allowing an exact reconstruction of the signal.

Multiple frameworks have been introduce that permit to learn invertible representations under certain conditions. Parseval networks (Cisse et al., 2017) have been introduced to increase the robustness of learned representations with respect to adversarial attacks. In this framework, the spectrum of convolutional operators is constrained to norm 1 during learning. The linear operator is thus injective. As a consequence, the input of Parseval networks can be recovered if but only if the built-in non-linearities are invertible as well, which is typically not the case. Bruna et al. (2013) derive conditions under which pooling representations are, but our method directly overcomes this issue. The Scattering transform (Mallat, 2012) is an example of predefined deep representation, approximately invariant to translations, that can be reconstructed when the degree of invariance specified is small. Yet, it requires a gradient descent optimization and no guarantee of convergences are known. In summary, the references make clear that invertibility requires special care in designing the architecture or special care in designing the optimization procedure. In this paper, we introduce a network, that overcomes this issue and has an exact inverse by construction.

Our main inspiration in this work are the recent reversible residual network (RevNet), introduced in Gomez et al. (2017). The reference builds a network from invertible residual network blocks. This ResNet-block architecture is similar to the lifting scheme (Sweldens, 1998) and similar to Feistel cipher diagrams (Menezes et al., 1996), as we will show. RevNets illustrate how to build invertible ResNet-type blocks for the goal of removing the need to store intermediate activations to be used in the backward pass. However, we note that RevNets still employ multiple non-invertible operators like max-pooling and downsampling operators as part of the network. As such, RevNets are not invertible by construction. In this paper, we show how to build an invertible type of RevNet architecture that performs competitively with RevNets on Imagenet, which we call i-RevNet.

3 THE i-REVNET

This section introduces the general framework of the i-RevNet architecture and explains how to explicitly build a pseudo-inverse to an i-RevNet. Its practical implementation is discussed, and we demonstrate competitive numerical results.

3.1 AN INVERTIBLE ARCHITECTURE

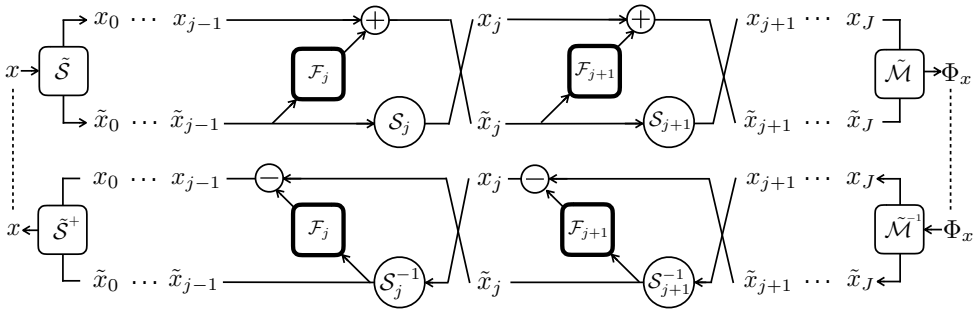

Figure 1: The main component of the i-RevNet and its pseudo-inverse. RevNet blocks are interleaved with convolutional bottlenecks $\mathcal{F}_j$ and reshuffling operations $\mathcal{S}_j$ to ensure invertibility of the architecture and computational efficiency. The input is processed through a splitting operator $\tilde{\mathcal{S}}$, and output is merged through $\tilde{\mathcal{M}}$. Observe that the inverse network is obtained with minimal adaptations.

We describe i-RevNets in their general setting. Their foundations are largely grounded in the recent RevNet architectures (Gomez et al., 2017). In a RevNet, an initial input is split into two sublayers $(x_0, \tilde{x}_0)$ of equal size, thanks to a splitting operator $\tilde{\mathcal{S}}x \triangleq (x_0, \tilde{x}_0)$. The operator $\tilde{\mathcal{S}}$ is linear, injective, reduces the spatial resolution of the coefficients and can potentially increase the layer size. We can thus build a pseudo inverse $\tilde{\mathcal{S}}^+$ that will be used for the inversion. The number of coefficients of the next layers is maintained, and at each depth j, the representation $\Phi_j x$ is again decoupled into two variables $\Phi_j x \triangleq (x_j, \tilde{x}_j)$ that play interlaced roles. The strategy implemented by a RevNet consists in an alternation between additions, and non-linear operators $\mathcal{F}_j$, while progressively downsampling the signal thanks to the operators $\mathcal{S}_j$. Here, $\mathcal{F}_j$ consists of convolutions and non-linearity on $\tilde{x}_j$. The pair of the final layer is concatenated through a merging operator $\tilde{\mathcal{M}}$. We will omit $\tilde{\mathcal{M}}, \tilde{\mathcal{M}}^{-1}, \tilde{\mathcal{S}}^+$ and $\tilde{\mathcal{S}}$ for the sake of simplicity, when not necessary.

Figure 1 describes the blocks of an i-RevNet. The design is similar to the Feistel cipher diagrams (Menezes et al., 1996) or a lifting scheme (Sweldens, 1998), which are invertible and efficient implementations of complex transforms like second generation wavelets. In this way we avoid the non-invertible modules of a RevNet (e.g. max-pooling or strides) which are necessary to train them in reasonable time and are supposed to help to build invariance w.r.t. translation variability. Our method shows we can replace them by linear and invertible modules $\mathcal{S}_j$, that potentially reduce the spatial resolution (we refer to it as a spatial down-sampling for the sake of simplicity), while maintaining the layer's size by increasing the number of channels.

We keep the computational cost manageable by tightly coupling downsampling and increase in width of the network. Reducing the spatial resolution can be undesirable, so $\mathcal{S}_j$ can potentially be the identity. We refer to such networks as i-RevNets. This leads to the following equations:

$$\begin{cases} x_{j+1} = \mathcal{S}_{j+1}\tilde{x}_j \\ \tilde{x}_{j+1} = x_j + \mathcal{F}_{j+1}\tilde{x}_j \end{cases} \iff \begin{cases} \tilde{x}_j = \mathcal{S}_{j+1}^{-1}x_{j+1} \\ x_j = \tilde{x}_{j+1} - \mathcal{F}_{j+1}\tilde{x}_j \end{cases} \tag{1}$$

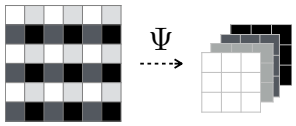

Figure 2: Illustration of the invertible down-sampling

Our downsampling layer can be written for u the spatial variable and λ the channel index:

$$\mathcal{S}_j x(u, \lambda) = x(\Psi(u, \lambda))$$

where Ψ is some invertible mapping. In principle any invertible downsampling operation like e.g. dilated convolutions (Yu & Koltun, 2015) can be considered here. We use the inverse of the operation described in Shi et al. (2016) as illustrated in Figure 2, since it preserves roughly the spatial ordering, and thus permits to avoid mixing different neighborhoods via the next convolution. $\tilde{\mathcal{S}}$ is similar, but also linearly increases the channel dimensionality, for example by concatenating 0.

The final layer $\Phi x \triangleq \Phi_J x = (x_J, \tilde{x}_J)$ is then averaged along the spatial dimension, followed by a ReLU non-linearity and finally a linear projection on the class probes, which are fed to a supervised training algorithm. From a given i-RevNet, it is possible to define a pseudo inverse Φ^+, i.e. $\Phi^+\Phi x = x$, whose convolutional section is as well an i-RevNet. This i-RevNet is dual, in the sense that it requires to replace $(\mathcal{S}_j, \mathcal{F}_j)$ by $(\mathcal{S}_j^{-1}, -\mathcal{F}_j)$ at each depth j, and to apply the pseudo-inverse $\tilde{\mathcal{S}}$ on the output. In consequence, its implementation is simple and specified by Equation (1). In subsection 4.2, we discuss that this pseudo-inverse Φ^+ has the same numerical stability as Φ, while however being very sensitive to small variations of an input on a large subspace.

3.2 Architecture, training and performances

In this subsection, we describe the particular model that we used during all our numerical experiments and its training procedure. The hyper-parameters were selected in order to optimize the computational cost of the constituting layers while keeping performance competitive. For the same reasons as in Gomez et al. (2017), our scheme also allows avoiding storing any intermediate activations at training time, making memory consumption for very deep i-RevNets not an issue in practice. We compare our implementation with RevNet-56, as provided in the open source release of Gomez et al. (2017).

Our model consists in a cascade of 18 blocks $\mathcal{F}_j$. Each of the $\mathcal{F}_j$ is a bottleneck block, which consists of a succession of 3 convolutional operators, each preceded by Batchnormalization and ReLU non-linearity. The second layer has two times fewer channels than the other two, while their corresponding kernel sizes are respectively $1 \times 1, 3 \times 3, 1 \times 1$. Consequently, similar to a RevNet-56 or a ResNet-50 (He et al., 2016), the input is propagated through 55 non-linearities until the layer $\Phi_J x$, thus we name our architecture i-RevNet-55. The final representation is spatially averaged and projected through the 1000 classes after a ReLU non-linearity.

We now discuss how we progressively decrease the spatial resolution, while increasing the number of channels per layer by use of the operators $\mathcal{S}_j$. The splitting operator $\tilde{\mathcal{S}}$ consists in a linear and injective embedding that downsamples by a factor 4^2 the spatial resolution by applying Ψ 2 times and increasing the number of output channels from 48 to 96 by simply adding 0. It permits to increase the initial layer size, and consequently, the size of the next layers. $\mathcal{S}_j$ allows us to reduce the number of computations while maintaining good classification performance. $\mathcal{S}_j$ will correspond to a downsampling operator respectively at the depth $j = 5, 9, 15$, just as in a normal RevNet. The spatial resolution of these layers is then reduced by a factor 2^2 while increasing the number of channels by a factor of 4 respectively to 96, 384, 1536 and 6144. Furthermore, it means that the corresponding spatial resolution for an input of size 224^2 is respectively $56^2, 28^2, 14^2, 7^2$. The total number of coefficients at each layer is then about $0.3M$. All the remaining layers S_j are kept fix to the identity as explained in the section above.

The training on Imagenet follows the same setup as Gomez et al. (2017), however, we observed one third longer wall-clock times for i-RevNets compared to plain RevNets. We train the model with momentum SGD with momentum 0.9. We regularized the model with a ℓ^2 weight decay of 10^{-4} and batch normalization. The dataset is processed for 600k iterations on a batch size of 256, distributed on 4GPUs. The initial learning rate is 0.1, dropped by a factor of ten every 160k iterations. The dataset was augmented according to Gomez et al. (2017). The images values are mapped to $[0, 1]$, while following geometric transformations were applied: random scaling, random horizontal flipping, random cropping of size 224^2, and finally color distortions. No other regularizations were incorporated into the classification pipeline. At test time, we rescale the image size to 256^2 and perform a center crop of size 224^2.

The Top-1 error rates on the validation set of ILSVRC-2012 of ResNet-50, RevNet-56 and i-RevNet-55 are 25.2%, 24.7% and 25.2% respectively. The number of parameters is respectively 26M, 28M, and 181M. This is caused by the fact, that we maintain the size of the intermediate layers, whose number of channels becomes quite large and thus the kernel sizes are too. The classification accuracy shows that despite its invertibility, this architecture leads to a comparable classification accuracy.

4 Analysis of the pseudo-inverse

We now analyse the representation Φ built by our neural network and its pseudo-inverse Φ^+. We first explain why obtaining Φ^+ is challenging, even locally. We then discuss the reconstruction, while displaying in the image space linear interpolations between representations.

4.1 An ill-conditioned inversion

In the previous section, we have described the i-RevNet architecture, that permits defining an inverse network. We explain now why this is normally difficult, by studying its local inversion. We study the local stability of a network Φ and its pseudo-inverse Φ^+ w.r.t. to its input, which means that we will quantify locally the variations of the network and its inverse w.r.t. to small variations of an input. As Φ is differentiable (and its inverse as well), an equivalent way to perform this analysis is to analyze the singular spectrum of the differential $\partial\Phi$ at some point, as for (x, y) close the following holds:

$$\Phi x \approx \Phi y + \partial\Phi_y(x - y).$$

Ideally, a well-conditioned operator has a spectrum constant equal to 1, for instance as achieved in Cisse et al. (2017).

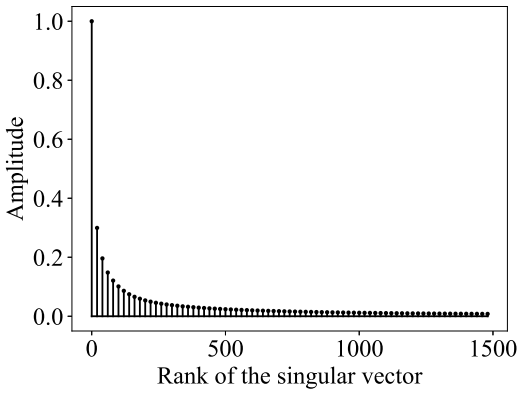

Figure 3: Normalized sorted spectrum of $\partial\Phi_x$.

In our numerical applications to an image x, $\partial\Phi_x$ corresponds to a very large matrix (square of the number of coefficients of the image at least) whose computations are thus expensive. In order to reduce the computational effort, we only consider variations of the image in a window of size 100^2. If π is the projection that permits to extract this window, we then consider $\partial\Phi_x \circ \pi$. Figure 3 corresponds to the spectrum of the differential, in increasing order, for a given natural image from ImageNet. The example we plot is typical of the behavior of Φ. Observe there is a fast decay, and that roughly the first 800 singular values dominate the remaining. This indicates Φ linearizes locally the space in a considerably smaller space in comparison to the original input dimension. However, the dimensionality is still quite large (i.e. > 10) and thus we can not infer that Φ lays locally in a low-dimensional manifold. It also proves that inversing Φ is difficult and is an ill-conditioned problem. Thus obtaining implicitly this inverse would be a challenging task that we avoided, thanks to the formal reconstruction algorithm provided by Subsection 3.1.

4.2 LINEAR INTERPOLATION AND RECONSTRUCTION

Visualizing or understanding the important direction in the representation of inner layers of a CNN, and in particular, the final layer is complex because typically the cascade is either not invertible or unstable. One approach to reconstruct from an output layer consists in finding the input images that matches the activation through a gradient descent, and to observe the resulting image. However, this technique leads only to a partial or informal reconstruction (Mahendran & Vedaldi, 2015). Another method consists in embedding the representation in a lower dimensional space and comparing the common attributes of nearest neighbors (Szegedy et al., 2013). It is also possible to train a CNN to reconstruct the representation (Dosovitskiy & Brox, 2016). Yet these methods require a priori knowledge in order to find the appropriate embeddings or training sets. We now discuss the improvements achieved by the i-RevNet.

Our main claim is that while the local inversion is ill-conditioned, the pseudo-inverse Φ^+ computations are as numerically stable as the forward network Φ. Indeed, observe that Φ^+ uses the same blocks $\mathcal{F}_j$ (Lipschitz bounded) as Φ, and thus it leads to the same numerical stability bounds. In particular, as the forward pass of the network does not seem to suffer from significant instabilities (e.g. adversarial examples), which are rare events (Lu et al., 2017), we will assume it is the case for Φ^+ as well. All our experiments with images from ImageNet and the Basel Faces dataset, indicate empirically that one can reconstruct perfectly any signal x from its representation Φx with a negligible error.

Figure 4: This graphic displays several reconstructed sequences $\{x^t\}_t$. The left image corresponds to x^0 and the right image to x^1.

Given a pair of images $\{x^0, x^1\}$, we propose to study linear interpolations between the pair of representations $\{\Phi x^0, \Phi x^1\}$, in the feature domain. Those interpolations do not necessarily correspond to existing images as Φ^+ is only a pseudo-inverse. We reconstruct a convex path between two input points; it means that if:

$$\phi^t = t\Phi x^0 + (1-t)\Phi x^1,$$

then:

$$x^t = \Phi^+ \phi^t$$

is a signal that corresponds to an image. We discretized $[0, 1]$ into $\{t_1, ..., t_k\}$, adpating manually the step and reconstruct the sequence of $\{x^{t_1}, ..., x^{t_k}\}$. Results are displayed in the Figure 4.

The pair of images were selected randomly from ImageNet and the Basel face dataset (Paysan et al., 2009). We fine-tuned the final layer with some dropout because we empirically observe a nicer visualization, this did not change the model performance significantly.

First, observe that a linear interpolation in the representation is not a linear interpolation in image space and that intermediary images seem noisy. However, one can visually determine morphing visual attributes of the images and the class seems always recognizable. Second, one can observe a clear out-of-plane rotation of the bottom images as suggested in Aubry & Russell (2015). It might suggest a phenomenon of linearization, and we thus investigate how the linear separation progresses with depth in the next section.

5 A CONTRACTION

We first show that a localized or linear classifier progressively improves with depth. Then, we describe the linear subspace spanned by Φ, namely the *feature space*, showing that the classification can be performed on a quite smaller subspace, which can be built via a PCA.

5.1 PROGRESSIVE LINEAR SEPARATION AND CONTRACTION

We show that while each layer of our deep representation being invertible, each layer still builds a progressively more linearly separable and contracted representation. Similarly to Oyallon (2017), we investigate those properties in each layer, with the following experimental protocol. As the computations are quite heavy because ImageNet has $N = 1.2M$ images, we had to use a random subset of the classes. We fix its size to 100 randomly selected classes, that consists of $N = 120k$ images, and keep the same subset during all our next experiments. At each depth j, we extract the features $\{\Phi_j x^n\}_{n \leq N}$ of the training set, we average them along the spatial variable and standardize them in order to avoid any ill-conditioning effects. We used both a nearest neighbor classifier and a linear SVM. The former is a localized classifier that indicates that the ℓ^2 metric is progressively more important for classification, while a linear SVM measure the linear separation of the different classes. The parameters of the linear SVM are cross-validated on a small subset of the training set, and then we train it on the 100 classes. We evaluate both classifiers on the validation set of ImageNet and report the Top1 accuracy in figure 5.

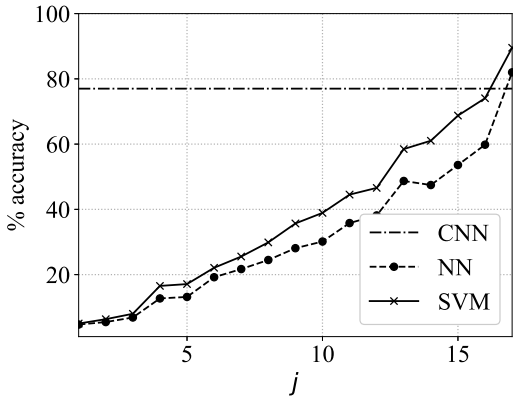

Figure 5: Accuracy at depth j for a linear SVM and a 1-nearest neighbor classifier applied to Φ_j.

We observe that both classifiers progressively improve with depth. They evolve similarly, the linear SVM performing slightly better than the nearest neighbor classifier because it is a more robust and discriminative classifier than the nearest neighbor. The classification performed by the CNN leads to 77.0%, and the linear SVM performs slightly better because we did not refine CNN's training to 100 classes. Observe that there is a jump of performances more intense on the 3 last layers, which seems to indicate that the former layers have prepared the representation to be more contracted and linearly separated into those final layers. In particular, those results might suggest that the data are embedded in a lower dimensional space, but this would be difficult to validate as the data are cursed by the dimensionality: estimating the local dimension is difficult. In the next section, we, however try to compute the dimension of the discriminative part of the representation.

5.2 DIMENSIONALITY ANALYSIS OF THE FEATURE SPACE

In this section, we investigate if we can refine the dimensionality of informative variabilities in the final representation. Indeed, the cascade of convolutional operators has been trained on the training set to separate the 1000 different classes while being a homeomorphism on its feature space. Thus, the dimensionality of the feature space is potentially large.

We replace the final non-linear projection described in Subsection 3.2 by a linear projection, which decreases the performances on the entire validation set from 75.4% to 73.2%, which is roughly the same. This indicates that the non-informative variabilities for classification can be removed via a linear projection on the final layer Φ, which lay on a space of dimension 1000, at most. However, this projection has been built via supervision, which can still retain directions that have been contracted and thus will not be selected by an algorithm such as PCA. We show in fact a PCA retains the necessary information for classification in a small subspace.

To do so, we build the linear projectors π_d on the subspace of the d first principal components, and we propose to measure the classification power of the projected representation with a supervised classifier, e.g. nearest neighbor or a linear SVM, on the previous 100 class task. Again, the feature representation $\{\Phi x^n\}_{n \leq N}$ are spatially averaged to remove the translation variability, and standardized on the training set. We apply both classifiers, and we report the classification accuracy of $\{\pi_d \Phi x^n\}_{n \leq N}$ w.r.t. to d on the Figure 6. A linear projection removes some information that can not be recovered by a linear classifier, therefore we observe that the classification accuracy only decreases significantly for $d \leq 200$. This shows that the signal indeed lies in a subspace much lower dimensional than the original feature dimensions that can be extracted simply with a PCA that extracts the direction of largest variances, illustrating a successful contraction of the representation.

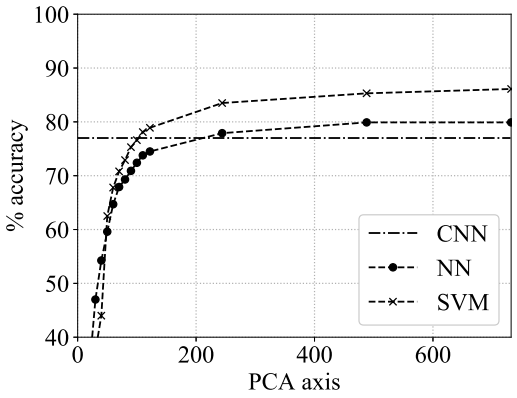

Figure 6: This plot corresponds to the accuracy of a linear SVM and nearest neighbor against the number of principal components retained.

6 Conclusion

Invertible representations and their relationship to loss of information are on the agenda of deep learning for some time. Understanding how transformations in feature space are related to the corresponding input is an important step towards interpretable deep networks, invertible deep networks may play an important role in such analysis since for example, one could potentially back-track a property from the feature space to the input space. To the best of our knowledge, this work provides the first empirical evidence that learning invertible representations that do not discard any information about their input on large-scale supervised problems is possible.

To achieve this we introduce the i-RevNet class of CNN which is fully invertible and permits to exactly recover the input from its last convolutional layer. i-RevNets achieve the same classification accuracy in the classification of complex datasets as illustrated on ILSVRC-2012, when compared to the RevNet (Gomez et al., 2017) and ResNet (He et al., 2016) architectures with a similar number of layers. Furthermore, the inverse network is obtained for free when training an i-RevNet, requiring only minimal adaption to recover inputs from the hidden representations.

The absence of loss of information is surprising, given the wide believe, that discarding information is essential for learning representations that generalize well to unseen data. We show that this is not the case and propose to explain the generalization property with empirical evidence of progressive separation and contraction with depth.

References

Alessandro Achille and Stefano Soatto. On the emergence of invariance and disentangling in deep representations. *arXiv preprint arXiv:1706.01350*, 2017.

Mathieu Aubry and Bryan C Russell. Understanding deep features with computer-generated imagery. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2875–2883,

2015.

Joan Bruna, Arthur Szlam, and Yann LeCun. Signal recovery from pooling representations. *arXiv preprint arXiv:1311.4025*, 2013.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pp. 854–863, 2017.

Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4829–4837, 2016.

Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. *arXiv preprint arXiv:1707.04585*, 2017.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196, 2015.

Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255, 2016.

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Stéphane Mallat. Understanding deep convolutional networks. *Phil. Trans. R. Soc. A*, 374(2065): 20150203, 2016.

Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

Edouard Oyallon. Building a regular decision boundary with deep networks. *arXiv preprint arXiv:1703.01775*, 2017.

Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pp. 296–301. Ieee, 2009.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, 2016.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM journal on mathematical analysis*, 29(2):511–546, 1998.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *Information Theory Workshop (ITW), 2015 IEEE*, pp. 1–5. IEEE, 2015.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.