

DeepView: Visualizing Classification Boundaries of Deep Neural Networks as Scatter Plots Using Discriminative Dimensionality Reduction

Alexander Schulz*, Fabian Hinder and Barbara Hammer

Machine Learning Group, Bielefeld University, D-33619 Bielefeld, Germany
{aschulz, fhinder, bhammer}@techfak.uni-bielefeld.de

Abstract

Machine learning algorithms using deep architectures have been able to implement increasingly powerful and successful models. However, they also become increasingly more complex, more difficult to comprehend and easier to fool. So far, most methods in the literature investigate the decision of the model for a single given input datum. In this paper, we propose to visualize a part of the decision function of a deep neural network together with a part of the data set in two dimensions with discriminative dimensionality reduction. This enables us to inspect how different properties of the data are treated by the model, such as outliers, adversaries or poisoned data. Further, the presented approach is complementary to the mentioned interpretation methods from the literature and hence might be even more useful in combination with those. Code is available at <https://github.com/LucaHermes/DeepView>.

1 Introduction

The increasing relevance of methods of AI in diverse areas such as autonomous driving, algorithmic trading, medical diagnoses, or recommender systems is accompanied by a high potential of vulnerability of these technologies: their use in every-day life in possibly non-stationary environments violates basic assumptions of learning theory such as samples being i.i.d. [Montiel *et al.*, 2018; Losing *et al.*, 2018]; adversarial attacks or poisoning can lead to unpredicted behavior of a single decision or the whole model behavior [Elsayed *et al.*, 2018; Jagielski *et al.*, 2018]; and skewed sampling of training data can lead to severely biased or unfair machine learning models if no filtering takes place [Leavy, 2018]. In combination with legal requirements such as the European Union’s general data protection regulation and right of explanation, these issues have led to a recent boost of explainable AI [Mueller *et al.*, 2019], including sparse local explanations [Ribeiro *et al.*, 2018], causal modeling [Alvarez-Melis and Jaakkola, 2017], counterfactual reasoning [Sokol and Flach, 2018], feature relevance determination, or saliency maps [Samek *et al.*, 2017], to name just a few approaches. These methods are accompanied

by first approaches aiming to quantify what interpretability by humans means [Lage *et al.*, 2019].

Yet, many techniques focus on single decisions rather than displaying parts of the decision boundary and the network’s generalization behavior in the input space. Up to now, there has been comparably little effort to build on human’s astonishing visual perception abilities and to display the behavior of deep networks in a visual plot, which generalizes an extremely natural and intuitive visualization of classification prescriptions: a scatter plot. This is always used in standard textbooks of pattern recognition to explain a classification prescription, it even made it to the cover image in [Scholkopf and Smola, 2001]. Yet a method to compute scatter plots, displaying training data, enriched by the decision boundary and network confidence in the plane, does not yet exist for deep networks in high-dimensional input space. Indeed, high dimensionality constitutes the major obstacle, since there do not exist homeomorphic mappings in between topological spaces of different dimensionality. A key challenge is how to efficiently and effectively determine a regularization and which aspects of the data space need to be displayed. In this contribution, we will build on the rich work of non-linear dimensionality reduction techniques, in particular the recent methodology Uniform Manifold Approximation and Projection (UMAP) [McInnes *et al.*, 2018], which is mathematically substantiated by a clear topological motivation, and we propose a pipeline *DeepView*, enabling to display the decision functions of trained deep networks together with benchmark data. To do so, we introduce two central ingredients: (i) we propose a novel discriminative variant of UMAP, which takes into account the information relevant for a priorly trained deep network, and we propose a mathematically sound method to compute this information efficiently. (ii) Further, we propose a novel way how UMAP can be enriched to also provide an "inverse" mapping, which abstracts from information, which is not relevant for the deep network (an exact inverse cannot exist, obviously). (iii) We demonstrate the effectiveness of the new visualization pipeline *DeepView* for popular deep learning models and data sets.

2 Related Work

While many approaches in explainable AI aim to explain single decisions of a model, only few try to provide a large scale view of a trained model or to visualize its decision boundaries. Notable exceptions here constitute [Lapuschkin *et al.*, 2019];

*Contact Author

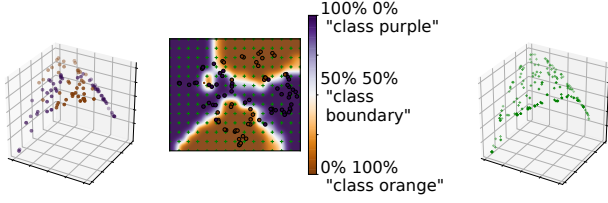


Figure 1: Illustration of the proposed method DeepView. A classifier trained on toy data (left) is visualized with DeepView (middle) utilizing the steps 1-4: data are projected (middle, step 1), a regular grid is created (middle, step 2) and projected with π^{-1} (right). Then the classifier is applied to the latter (step 3) and the resulting labels and entropy of the predictive distribution are displayed (middle, step 4). The orange island indicates overfitting due to a single orange point.

Zahavy *et al.*, 2016] where a projection of data is interpreted with regard to different strategies for solving the task at hand. Although they depict different groups of data, they do not show decision boundaries of a classifier. Further relevant related work is [Schulz *et al.*, 2015], where decision boundaries of classifiers are depicted, but this approach is based on a density estimation in the input space, rendering it infeasible for typical domains of deep networks.

As concerns discriminative dimensionality reduction (DiDi), [Venna *et al.*, 2010] demonstrated that DiDi based on the Fisher metric produces better or at least comparable visualizations in relation to other formalizations. Hence, we focus our discussion on DiDi implemented with the Fisher metric.

3 DeepView: Visualizing the Decision Function of a Deep Network

In the trivial case of two-dimensional data, the visualization of a trained classification model is a straight-forward and useful thing: we can apply the classifier to each position in an area around our data and encode the predicted label and certainty in the background of a scatter plot. This illustrates very clearly how the trained model behaves in each area of the data space.

For high-dimensional data, however, this is not directly possible for several reasons: (i) While we can apply dimensionality reduction (DR) to visualize data, regular DR techniques will try to preserve all the structure in the data and, such, make critical compromises, for instance preserving brightness differences instead of object relevant properties, in the case of images. (ii) The classifier is usually trained in a high-dimensional space. Hence, applying it to every position in the data space is not feasible because of an exponentially growing number of positions. (iii) Finally, visualizing a full high-dimensional classifier in two dimensions is not possible because the decision function is high-dimensional and an unguided projection of it (if possible at all) would yield drastic information loss.

3.1 DeepView: Proposed Visualization Scheme

In this contribution, we develop a scheme that enables us to circumvent these problems and propose feasible approximations for the case of deep neural network classifiers. In particular, we generalize the approach presented in [Schulz *et al.*, 2015] which is restricted to shallow classifiers and (intrinsically)

low-dimensional data. More specifically, we propose to use DiDi based on the trained classification model (section 4.1), enabling the DR to focus on the aspects of the data which are relevant for the classifier, alleviating problem (i). Further, instead of trying to obtain predictions for the full data space, we develop a scheme to obtain predictions only for a relevant subspace which we then use to visualize the decision function in two dimensions, such, solving problem (ii) and tackling problem (iii) by a reasonable approximation. This is based on inverse dimensionality reduction and is modeled in such a way that it matches the way the DR is constructed (section 4.2).

We propose to apply the following steps:

1. Apply the DiDi technique Fisher UMAP (developed in section 4.1) which is based on the underlying deep network to project a data set consisting of points \mathbf{x}_i to two dimensions, yielding $\mathbf{y}_i = \pi(\mathbf{x}_i)$.
2. Create a tight regular grid of samples \mathbf{r}_i in the two-dimensional space and map it to the high-dimensional space using the approach presented in section 4.2, yielding points $\mathbf{s}_i = \pi^{-1}(\mathbf{r}_i)$.
3. Apply the neural network f to \mathbf{s}_i in order to obtain predictions and certainties.
4. Visualize the label together with the entropy of the certainty for each position \mathbf{r}_i in the background of the projection space in order to obtain an approximation of the decision function.

These steps are demonstrated on a toy example in Figure 1. We propose novel implementations of steps 1 and 2, enabling the resulting approach to visualize deep neural networks.

4 Dimensionality Reduction

Dimensionality reduction techniques for visualization aim to find mappings $\pi : (S, d_S) \rightarrow \mathbb{R}^d$, $d = 2, 3$, where (S, d_S) is some metric space, such that π preserves the information encoded in a set of data points $\mathbf{x}_1, \dots, \mathbf{x}_n \in S$ as good as possible. The main aspect of a DR method is therefore to find a measure, and hence a cost function, to compare the information contained in two sets of points, allowing us to find a set of points $\mathbf{y}_1, \dots, \mathbf{y}_n \in \mathbb{R}^d$, $d = 2, 3$ encoding approximately the same information. While the state of the art approach for performing nonlinear DR is t-Distributed Stochastic Neighbor Embedding (t-SNE) [van der Maaten and Hinton, 2008], recently a novel technique has been developed called UMAP [McInnes *et al.*, 2018] which produces at least comparable results to t-SNE and is formulated in a way that we make use of to develop our inverse projection π^{-1} . Hence, we focus on UMAP and introduce some formal details in the following.

UMAP assumes that data is distributed according to a uniform distribution on a Riemannian manifold which may be approximated by a simplicial complex. To find the low-dimensional embedding, the problem is restricted to the complex's 1-skeleton, i.e. the probability that two points are connected by an edge. It is assumed that the probability of an edge is induced by the distance between the end points and the local density. In the embedding space $(\mathbb{R}^d, d = 2, 3)$ this is modeled by a student-t-distribution $w_{ij} = (1 + a\|\mathbf{y}_i - \mathbf{y}_j\|^{2b})^{-1}$, where a and b are hyper parameters, in the original space (S, d_S) one

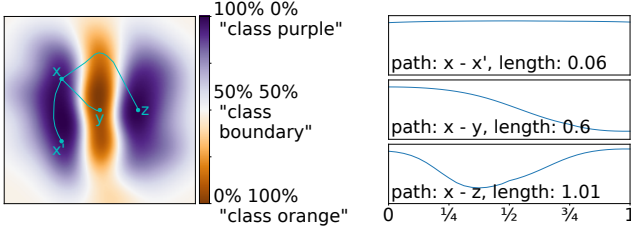


Figure 2: Measuring the distance between points. Left: f and paths. Right: class probability of class purple along the respective paths.

uses $v_{ij} = v_{i|j} \perp v_{j|i}$, where $x \perp y = x + y - xy$ is the sum T-conorm and $v_{i|j} = \exp(-\max(d(\mathbf{x}_i, \mathbf{x}_j)^2 - \rho_i, 0)/\sigma_i)$, with σ_i the k -perplexity at \mathbf{x}_i and $\rho_i = \min_{i \neq j} d(\mathbf{x}_i, \mathbf{x}_j)^2$ the distance to the nearest neighbor.

One then finds $\mathbf{y}_1, \dots, \mathbf{y}_n \in \mathbb{R}^d$, $d = 2, 3$ by minimizing the Kullback-Leibler divergence of the embedding given the data:

$$\operatorname{argmin}_{\mathbf{y}_1, \dots, \mathbf{y}_n} \sum_{i \neq j} D_{\text{KL}}(v_{ij} || w_{ij})$$

4.1 Discriminative Dimensionality Reduction

DR usually only takes the observed data points into account. However, since we would like to visualize a classifier f together with data points, it is reasonable to incorporate f as well; such methods are referred to as DiDi methods. These particularly focus on the structure important for a given classification task [Venna *et al.*, 2010; Goldberger *et al.*, 2004; Gisbrecht *et al.*, 2015] and such specify which structure in the data is more important than other. A very promising approach for this purpose is to integrate class information in the distance computation using the Fisher metric and consequently apply a regular DR method on top. Thereby, the Fisher metric is not defined in the parameter space of a model as in [Amari, 1985; Amari and Nagaoka, 2007], but in the data space [Kaski *et al.*, 2001; Peltonen *et al.*, 2004; Ruiz *et al.*, 2013]. For this case, [Venna *et al.*, 2010] demonstrated suitability of DiDi mappings based on the Fisher metric and [Schulz *et al.*, 2015] illustrated its benefit for visualization of shallow classifiers.

Here, we augment the literature by proposing (i) a new derivation of the Fisher metric in the context of DiDi, (ii) a new approach to estimate the Fisher metric without needing to compute gradients and (iii) to use f for the probabilistic model instead of a new non-parametric estimate. Together, these enable us to compute a DiDi visualization for high-dimensional data and deep networks.

We will now construct a metric d , suitable for our considerations. As it will turn out this is a generalization of the Fisher metric. A significant benefit of our approach is that it is capable of handling non-differentiable, even non-continuous, classifiers and allows a natural approximation that yields a much faster computation, in particular for complicated f .

Let S be our source space equipped with a metric d_S , \mathcal{C} be the collection of all class labels and $f : S \rightarrow \mathbf{P}(\mathcal{C})$ be our classifier, where $\mathbf{P}(\mathcal{C})$ denotes the space of all possible class probabilities. In contrast to d_S we would like d to also capture the topological features induced

by the decision boundaries of f : A first, natural step is to consider $f^*d_{\text{JS}}(s, t) := d_{\text{JS}}(f(s), f(t))$ the so called pullback-metric along f , here $d_{\text{JS}}(p, q) = \sqrt{D_{\text{JS}}(p||q)} := \sqrt{(D_{\text{KL}}(p||m) + D_{\text{KL}}(q||m))/2}$, $m = (p + q)/2$ denotes the Jensen-Shannon-metric, a metric on $\mathbf{P}(\mathcal{C})$. This pullback is an indicator for the boundary features in the sense that it tells us whenever two points are mapped to the same class or not, however it has two issues: (a) it is no proper metric on S , i.e. all points with the same assignment are collapsed to a single point, (b) it cannot decide whenever two points belong to the same connected component induced by the decision boundaries, e.g. Figure 2 x, x' vs. x, z .

To overcome (a) we regularize with d_S using parameter $\lambda \in [0, 1]$, i.e. we consider $(1 - \lambda)f^*d_{\text{JS}}(s, t) + \lambda d_S(s, t)$.

To overcome (b) notice that $f^*d_{\text{JS}}(s, t)$ only captures information of f at s and t but not "between them". A natural way to extend such a local metric, which is also widely used for example in Riemannian geometry and hence information geometry, is by using an arc-length metric, i.e. we use the length of the shortest path connecting two points according to the regularized pullback described above as their distance; its behavior illustrated in Figure 2.

We therefore arrive at the following formalization:

Definition 1. For a metric space (S, d_S) together with a classifier f define the DiDi-metric d with mixture $\lambda \in [0, 1]$ as the arc-length metric induced by the pullback of the Jensen-Shannon-metric along f regularized with d_S , i.e.

$$d(x, y) = \inf_{\substack{\gamma: [0, 1] \rightarrow S \text{ continuous,} \\ \gamma(0) = x, \gamma(1) = y}} L(\gamma)$$

$$L(\gamma) = \sup_{0 = t_0 < \dots < t_n = 1} \sum_{i=1}^n (1 - \lambda) f^*d_{\text{JS}}(\gamma(t_{i-1}), \gamma(t_i)) + \lambda d_S(\gamma(t_{i-1}), \gamma(t_i)).$$

Theorem 1. Let (S, d_S) be a metric space and f a smooth classifier on S . Denote by d the DiDi-metric with mixture $\lambda = 0$ and by d_{Fisher} the Fisher metric (as defined in [Kaski *et al.*, 2001]) induced by f . Then it holds

$$d_{\text{Fisher}}(x, y) = \sqrt{8}d(x, y)$$

for all $x, y \in S$.

Proof. Follows from [Crooks, 2007]; details omitted due to space restrictions \square

Note that definition 1 does not need gradient computations unlike [Kaski *et al.*, 2001]. Further unlike [Schulz *et al.*, 2015] it no longer requires density estimation of the data space, which is infeasible for typical domains of deep networks.

To implement this approach, we follow [Peltonen *et al.*, 2004] and assume that $d(x, y)$ can be approximated by n equidistant points $p_i = (1 - \frac{i}{n})x + \frac{i}{n}y$ on a straight line, i.e.

$$d(x, y) \approx \sum_{i=1}^n (1 - \lambda) d_{\text{JS}}(f(p_{i-1}), f(p_i)) + \lambda d_S(p_{i-1}, p_i).$$

These approximations are evaluated in [Peltonen *et al.*, 2004] with the result that they constitute a good compromise between speed and accuracy for the application of DR.

In a final step, a nonlinear DR technique is used to project the resulting distances $d(x, y)$ for visualization. For this purpose, the papers [Venna *et al.*, 2010; Gisbrecht *et al.*, 2015] have demonstrated that neighbor embedding methods such as NeRV and t-SNE are particularly well suited. Here, we utilize the novel method UMAP [McInnes *et al.*, 2018] which belongs to this class of methods and has beneficial properties that we will exploit in the following.

4.2 Similarity Coordinate Embedding for Inverse Dimensionality Reduction

So far we have discussed the task of finding a mapping $\pi : (S, d_S) \rightarrow \mathbb{R}^d, d = 2, 3$. Now we are interested in finding a reverse map $\pi^{-1} : \mathbb{R}^d \rightarrow (S, d_S)$ that acts as a pseudo-inverse of π . In particular we would like to obtain π^{-1} in a natural way, under the assumption that π is given by UMAP.

In some sense π^{-1} performs an out of sample extension in the opposite direction; therefore let us first consider a "usual" out of sample extension: Suppose we are given some new sample $\mathbf{x} \in S$ and let $v_i(\mathbf{x})$ represent the probability that \mathbf{x}_i and \mathbf{x} are close or similar (i.e. $v_i(\mathbf{x}_j) = v_{ij}$), then UMAP aims to find $\mathbf{y} \in \mathbb{R}^d$ with $w_i(\mathbf{y})$ representing the probability that \mathbf{y}_i and \mathbf{y} are close or similar, by minimize the Kullback-Leibler divergence of $v_i(\mathbf{x})$ and $w_i(\mathbf{y})$. Following a Bayesian perspective, to determine π^{-1} we interchange the role of \mathbf{x} and \mathbf{y} and arrive at

$$\pi^{-1}(\mathbf{y}) := \operatorname{argmin}_{\mathbf{x} \in S} \sum_{i=1}^n D_{\text{KL}}(w_i(\mathbf{y}) \| v_i(\mathbf{x}))$$

where we use $v_i(\mathbf{x}) = \exp(-d_S(\theta_i, \mathbf{x})^2 / \sigma_i)$ and $w_i(\mathbf{y}) = (1 + a \|\rho_i - \mathbf{y}\|^{2b})^{-1}$ as in the case of UMAP, where we have to find $\theta_1, \dots, \theta_n$ and ρ_1, \dots, ρ_n such that π^{-1} fits π on our observations, i.e. $d_S(\pi^{-1}(\pi(\mathbf{x}_i)), \mathbf{x}_i) \rightarrow \min$.

To make our approach feasible we have to find a way to compute π^{-1} in reasonable time. Since this heavily depends on d_S we may only consider two examples in more detail: d_S is the Euclidean metric or a Riemannian metric.

Supposing that $S = \mathbb{R}^D$ with $d \ll D$ and $d_S = \|\cdot\|_2$ is the Euclidean metric.

Theorem 2. *Let $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^D$ be source points and $\mathbf{y}_1, \dots, \mathbf{y}_n \in \mathbb{R}^d$ their corresponding projections. Denote by $f(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n D_{\text{KL}}(w_i(\mathbf{y}) \| v_i(\mathbf{x}))$ the cost function of π^{-1} and by $\hat{f}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n w_i(\mathbf{y}) \|\theta_i - \mathbf{x}\|^2 / \sigma_i$. Then it holds $\hat{f}(\mathbf{x}, \mathbf{y}) < f(\mathbf{x}, \mathbf{y})$. Furthermore under the assumption of Gaussian noise in the input space in mean it holds*

$$f(\mathbf{x}, \mathbf{y}) - \hat{f}(\mathbf{x}, \mathbf{y}) \in \mathcal{O}(\exp(-D/2)),$$

i.e. $\hat{f}(\mathbf{x}, \mathbf{y})$ converges in mean exponentially fast to $f(\mathbf{x}, \mathbf{y})$ as the number of dimensions increases.

Furthermore it holds

$$\operatorname{argmin}_{\mathbf{x} \in S} \hat{f}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n \frac{w_i(\mathbf{y}) / \sigma_i}{\sum_{j=1}^n w_j(\mathbf{y}) / \sigma_j} \cdot \theta_i.$$

Proof. All proofs are omitted due to space restrictions \square

Using this theorem we see that approximation π^{-1} by a radial basis-function network is well suited.

To generalize this approach to arbitrary Riemannian manifolds recall that those are locally, approximately given by inner product spaces. It is therefore enough to generalize our approach to arbitrary, finite dimensional, inner product spaces:

Lemma 1. *Let S be a finite dimensional real vector space. Let $d : S \times S \rightarrow \mathbb{R}$ be a metric induced by an inner product and X be a S -valued random variable. Then it holds*

$$\operatorname{argmin}_{\mathbf{x} \in S} \mathbb{E} [d(X, \mathbf{x})^2] = \operatorname{argmin}_{\mathbf{x} \in S} \mathbb{E} [\|X - \mathbf{x}\|_2^2].$$

So if we consider a Riemannian manifold (M, d_M) and we approximate d_M at \mathbf{x}^* , the point we are searching for, we obtain the same formula as in the euclidean case.

Furthermore when training π^{-1} using Euclidean metric, the result was comparable to the case where we trained with an local approximation of d_S using Fisher-matrices.

5 Experiments

In this section we apply the proposed method DeepView to visualize classifiers trained on the datasets CIFAR-10 and Fashion-MNIST and demonstrate exemplary findings in the presence of adversarial and backdoor attacks. Before it can be applied, however, it is important to investigate how accurate this visualization is.

5.1 Evaluating the Proposed Visualization

Here, two questions need to be answered: (i) how well does π show the view of the classifier on the data and (ii) how well is π^{-1} depicting the decisions of the classifier in these areas?

Addressing question (i), we pursue the following intuition: If the projection π takes the decision function of the classifier properly into account, then the classification decisions should be partially represented in the structure of the projected data points. I.e. points that are close to each other should be points that are classified similarly by the classification model. We can verify the degree to how much this is true by evaluating the accuracy of a simple classifier trained in the projection space using the labels of the deep network. For this purpose, we utilize the leave-one-out error of a k-nearest neighbor (kNN) classifier with $k = 5$ being a standard choice and refer to this measure as Q_{kNN} . When we consider UMAP based on the Euclidean metric, we denote this measure as $Q_{\text{kNN-E}}$.

(ii) Evaluating π^{-1} is less straight-forward. Here we suggest a scheme to evaluate the quality of the depicted decision function on the positions of the data points. For every point \mathbf{y}_i , compare the classification label of its original counterpart and the classification label of its inverse projection. More formally, we calculate the accordance of $f(\pi^{-1}(\mathbf{y}_i))$ and $f(\mathbf{x}_i)$. Depending on the selected points, such an evaluation will have different implications: Using pairs $(\mathbf{x}_i, \mathbf{y}_i)$ that have been employed for training π^{-1} will result in an assessment of the quality at the positions of the data points. Using point pairs that have not been utilized for training π^{-1} will rate the quality of the mapping at positions not seen before, i.e. of areas without data. Both are useful since they tell us how accurate the visualization is in areas where data points are available and

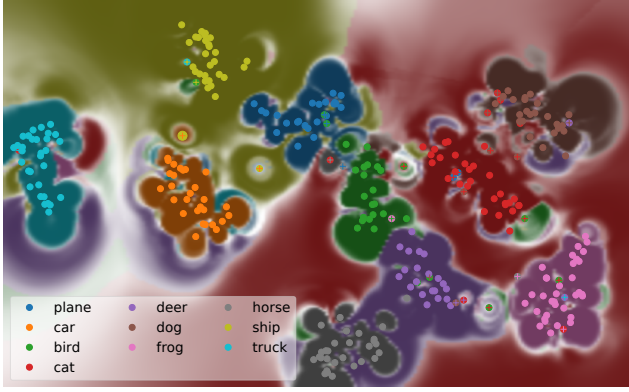


Figure 3: Visualization of a ResNet-20 model trained on the CIFAR-10 data set together with a subset of the test data. The big yellow circle depicts an adversarial example.

in those where this is not the case. We will refer with Q_{data} to the former and with $Q_{-\text{data}}$ to the latter.

We evaluate the resulting visualizations with these scores, where we use 70% of the data for training π .

5.2 Hyperparameter Selection

We choose the amount of Euclidean regularization λ in the following way: we evaluate π for Q_{kNN} with $\lambda \in [0.2, 0.8]$ and choose the largest one that does not degrade Q_{kNN} significantly. As a result we set $\lambda = 0.65$ for both data sets. As concerns UMAP, we set $n_neighbors = 30$ in all cases and min_dist to 1 for Fashion-MNIST. For π^{-1} , we set a to the smallest value that does not lead to a large drop in Q_{data} and $b = 1$.

5.3 Visualizing a ResNet-20 Network Trained on the CIFAR-10 Dataset

The *CIFAR-10 data set* consists of 32x32 color images with 10 classes (see Figure 3). The training set contains 50.000 examples and the present implementation¹ has an accuracy of 91.7% on the 10.000 test images, using a pre-trained residual network (ResNet) with 20 layers [He *et al.*, 2016].

The result of DeepView applied to a subset of 300 points selected randomly from the test set and the ResNet-20 network is shown in Figure 3. Each point corresponds to one image and the color indicates its original label. If classified differently by the deep net, a cross in the according color indicates that label. The color of the background depicts the classification of the net in that area and the intensity encodes the certainty.

First of all, we consider the evaluation of the resulting visualization and, second, discuss the information than can be drawn from it. The former is summarized in Table 1. For the sake of completeness, we also evaluate the UMAP projection based on the Euclidean metric ($Q_{\text{kNN-E}}$). The accuracy in this case amounts to 18.3%, which makes clear that such an embedding is not useful in this case. With the Fisher metric, however, the accuracy is 96.3% indicating that the projection space very well resembles the classification behavior of our

¹We employ the implementation from https://github.com/akamaster/pytorch_resnet_cifar10.



Figure 4: A zoomed in view on Figure 3 together with markers of specified positions around the adversarial example (left). The resulting images of the inverse mapping π^{-1} for the previously specified positions, together with the assigned label and according certainty of the classifier (right).

model. Concerning the visualization of the decision function, the close to perfect Q_{data} demonstrates that the visualization is very accurate at the given data points. For the vicinity of the data points, $Q_{-\text{data}}$ assess an accuracy 83.3%.

Regarding the visualization, the decision boundaries for the class 'orange' seem to have a complex shape, indicating that this class might be particularly difficult. Indeed, when inspecting the confusion matrix, this class has the smallest true positive rate with 83.2%. Furthermore, we can identify several points that are treated specially. One example is the yellow point that we have depicted with a larger symbol. This constitutes an adversarial example that we have created using the sign of the model gradient [Goodfellow *et al.*, 2014] and have added to the 300 images. The visualization shows how this point is classified as orange by the classifier, but this area is surrounded by the yellow class. Since this seems worth a closer look, we zoom into the according area (see Figure 4, left). Because this visualization is constructed by an inverse mapping, we can inspect images according to arbitrary positions in the visualization. In order to do so, we specify potentially interesting positions in the vicinity of the adversarial example (see 4, left) and depict their projections with π^{-1} on the right side. The markers '0' and '2' are in the area of the yellow class ('ship') and the according images can be clearly identified as such. Although the images of the markers '2', '3', '4' and '5' look fairly similar, their classifications vary heavily, going from 'ship' to 'car' and back to 'ship'. These images show, that there seems to be a kind of 'pocket' of the 'car' class inside the 'ship' class while the images in this pocket still look like natural ships to a human. This concept of 'pockets' has been addressed in the literature before [Feinman *et al.*, 2017]. Marker '1', being close to an area classified as 'bird', looks like a bird but is classified as 'ship'. Here, the decision boundaries of the model seem not to be well tuned. A last

Data Set	$Q_{\text{kNN-E}}$	Q_{kNN}	Q_{data}	$Q_{-\text{data}}$
CIFAR-10	18.3%	96.3%	99.5%	83.3%
Fashion-MNIST	66.1%	94.8%	98.6%	95.0%

Table 1: Results of the evaluation criteria defined in section 5.1, characterizing the quality of the embeddings. Except for $Q_{\text{kNN-E}}$, Fisher UMAP is utilized.

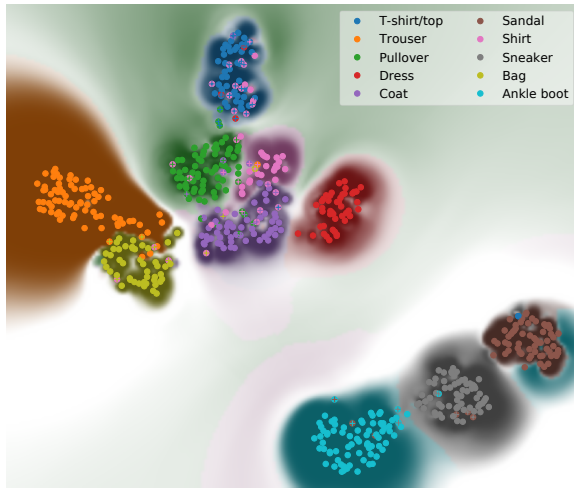


Figure 5: Visualization of a ConvNet trained on a poisoned Fashion-MNIST data set together with 600 test data points including 20 backdoored samples.

example marker '7' in a 'pocket-like' region shows a ship with an intense blue area above it, which might be the reason for the wrong classification. A further analysis could be performed to investigate this aspect with e.g. saliency maps.

5.4 Fashion-MNIST with Backdoors

We train a 4-layered convolutional network for the Fashion-MNIST data set [Xiao *et al.*, 2017], thereby applying data poisoning. We follow [Gu *et al.*, 2017] and introduce backdoors by adding small patterns to some points of one class and changing their label to a target class. The resulting model has 90% accuracy on the clean test set, so the poisoning is not easily detectable. Successively, we apply DeepView to a test set of 600 examples including 20 backdoored images and investigate whether we can detect the latter in the visualization.

The DeepView visualization is depicted in Figure 5, where a group in the lower right corner consisting of different types of shoes can be observed. In the top part, class overlap between blue points ('T-shirt/top') and pink ones ('Shirt') is visible. Less expected is the neighborhood of orange ('Trousers') and yellow points ('Bag') which is particularly the case for a subset of orange points. Hence we take a close look at this area in Figure 6. We use a similar approach as before and investigate the suspicious area with π^{-1} . Here we can observe that one area in the main orange cluster corresponds to regular trousers (marker 0), while the area at markers 2 and 3 corresponds to bags which are classified as trousers and have a specific pattern in the top right corner. Indeed, this reveals the backdoor attack.

For the according visualization, the most time demanding step, the calculation of Fisher distances, took around 90 seconds on a standard desktop computer with a consumer level graphics card. If computed directly, the computational complexity is squared in the number of data. Approximate schemes such as used by UMAP are in principle applicable here as well.

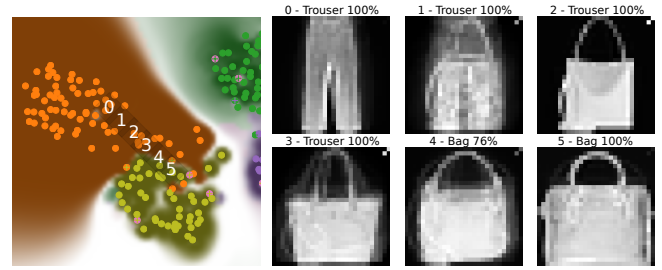


Figure 6: A zoomed in view on Figure 5 together with markers (left). The resulting images of π^{-1} for these positions, together with the assigned label and according certainty of the classifier (right).

6 Conclusion

In this work we propose *DeepView*, to the best of our knowledge the first algorithm that is able to visualize a smooth two-dimensional manifold of the decision function of a deep neural network which is trained on high-dimensional data such as natural images. For this purpose, we adopt a mathematically precise formulation of DiDi together with a matching choice of inverse DR. We apply DeepView to two deep networks, a Residual Network with 20 layers trained on CIFAR-10 and a 4-layered ConvNet trained on poisoned Fashion-MNIST, and illustrate how it provides insight into the model and data.

While we demonstrate DeepView only for image data, the method is in no way restricted to this domain and utilization for e.g. text analysis constitutes an interesting further application area. An exciting open question in this regard is how to extend the utilized Fisher metric in case of temporal data.

We believe that the presented approach can not only provide insights into trained models but also contribute to improve these, by e.g. providing insights into areas of lacking data.

Acknowledgements

Funding in the frame of the BMBF project MechML, 01IS18053E is gratefully acknowledged. We also thank Luca Hermes for providing a more efficient implementation.

References

- [Alvarez-Melis and Jaakkola, 2017] David Alvarez-Melis and Tommi S. Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models. *CoRR*, abs/1707.01943, 2017.
- [Amari and Nagaoka, 2007] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*. Translations of mathematical monographs. American Mathematical Society, 2007.
- [Amari, 1985] Shun-ichi Amari. Differential-geometrical methods in statistics. *Lecture Notes on Statistics*, 28:1, 1985.
- [Crooks, 2007] Gavin E. Crooks. Measuring thermodynamic length. *arXiv preprint arXiv:0706.0559v2*, 2007.
- [Elsayed *et al.*, 2018] Gamaleldin F. Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alex Kurakin, Ian J. Goodfellow, and Jascha Sohl-Dickstein. Adversarial

- examples that fool both human and computer vision. *CoRR*, abs/1802.08195, 2018.
- [Feinman *et al.*, 2017] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.
- [Gisbrecht *et al.*, 2015] Andrej Gisbrecht, Alexander Schulz, and Barbara Hammer. Parametric nonlinear dimensionality reduction using kernel t-sne. *Neurocomputing*, 147:71–82, January 2015.
- [Goldberger *et al.*, 2004] Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, 2004.
- [Goodfellow *et al.*, 2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [Gu *et al.*, 2017] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2017.
- [He *et al.*, 2016] 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*, pages 770–778, 2016.
- [Jagielski *et al.*, 2018] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *IEEE Symposium on Security and Privacy*, pages 19–35, May 2018.
- [Kaski *et al.*, 2001] Samuel Kaski, Janne Sinkkonen, and Jaakko Peltonen. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*, 12:936–947, 2001.
- [Lage *et al.*, 2019] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. *CoRR*, abs/1902.00006, 2019.
- [Lapuschkin *et al.*, 2019] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 2019.
- [Leavy, 2018] Susan Leavy. Gender bias in artificial intelligence: The need for diversity and gender theory in machine learning. In *2018 IEEE/ACM 1st International Workshop on Gender Equality in Software Engineering (GE)*, pages 14–16, May 2018.
- [Losing *et al.*, 2018] Viktor Losing, Barbara Hammer, and Heiko Wersing. Tackling heterogeneous concept drift with the self-adjusting memory (SAM). *Knowl. Inf. Syst.*, 54(1):171–201, 2018.
- [McInnes *et al.*, 2018] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- [Montiel *et al.*, 2018] Jacob Montiel, Jesse Read, Albert Bifet, and Talel Abdesslem. Scikit-multiflow: A multi-output streaming framework. *Journal of Machine Learning Research*, 19:72:1–72:5, 2018.
- [Mueller *et al.*, 2019] Shane T. Mueller, Robert R. Hoffman, William J. Clancey, Abigail Emrey, and Gary Klein. Explanation in human-ai systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable AI. *CoRR*, abs/1902.01876, 2019.
- [Peltonen *et al.*, 2004] Jaakko Peltonen, Arto Klami, and Samuel Kaski. Improved learning of riemannian metrics for exploratory analysis. *Neural Networks*, 17:1087–1100, 2004.
- [Ribeiro *et al.*, 2018] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference, (AAAI-18), (IAAI-18), and (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1527–1535, 2018.
- [Ruiz *et al.*, 2013] Héctor Ruiz, Terence A Etchells, Ian H Jarman, José D Martín, and Paulo JG Lisboa. A principled approach to network-based classification and data representation. *Neurocomputing*, 112:79–91, 2013.
- [Samek *et al.*, 2017] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *CoRR*, abs/1708.08296, 2017.
- [Scholkopf and Smola, 2001] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
- [Schulz *et al.*, 2015] Alexander Schulz, Andrej Gisbrecht, and Barbara Hammer. Using discriminative dimensionality reduction to visualize classifiers. *Neural Processing Letters*, 42(1):27–54, August 2015.
- [Sokol and Flach, 2018] Kacper Sokol and Peter Flach. Glass-box: Explaining ai decisions with counterfactual statements through conversation with a voice-enabled virtual assistant. In *IJCAI-18*, pages 5868–5870, 7 2018.
- [van der Maaten and Hinton, 2008] Laurens J.P. van der Maaten and Geoffrey E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.
- [Venna *et al.*, 2010] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11:451–490, 2010.
- [Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.
- [Zahavy *et al.*, 2016] Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. Graying the black box: Understanding dqns. In *Proceedings of ICML*, volume 48, pages 1899–1908, New York, New York, USA, 20–22 Jun 2016. PMLR.