Projective Latent Space Decluttering

Andreas Hinterreiter[1,2], Marc Streit[1], and Bernhard Kainz[1]

[1] Biomedical Image Analysis Group, Imperial College, UK
{a.hinterreiter, g.meng16, b.kainz}@imperial.ac.uk
[2] Institute of Computer Graphics, Johannes Kepler University Linz, Austria
{andreas.hinterreiter}@jku.at

Abstract. High-dimensional latent representations learned by neural network classifiers are notoriously hard to interpret. Especially in medical applications, model developers and domain experts desire a better understanding of how these latent representations relate to the resulting classification performance. We present a framework for retraining classifiers by backpropagating manual changes made to low-dimensional embeddings of the latent space. This means that our technique allows the practitioner to control the latent decision space in an intuitive way. Our approach is based on parametric approximations of non-linear embedding techniques such as t-distributed stochastic neighbourhood embedding. Using this approach, it is possible to manually shape and declutter the latent space of image classifiers in order to better match the expectations of domain experts or to fulfil specific requirements of classification tasks. For instance, the performance for specific class pairs can be enhanced by manually separating the class clusters in the embedding, without significantly affecting the overall performance of the other classes. We evaluate our technique on a real-world scenario in fetal ultrasound imaging.

Keywords: Image classification · Non-linear embedding · Latent space.

1 Introduction

Interpretation of classification models is often difficult due to a high number of parameters and high-dimensional latent spaces. Dimensionality reduction techniques are commonly used to visualise and explain latent representations via low-dimensional embeddings. These embeddings are useful to identify problematic classes, to visualise the impact of architectural changes, and to compare new approaches to previous work. However, there is a lot of debate about how well such mappings represent the actual decision boundaries and the resulting model performance.

In this work, we aim to change the paradigm of passive observation of mappings to active intervention into the training process. This is challenging because embeddings are often randomly initialised and spatially inconsistent during training and across networks. There are many situations where a practitioner would like to alter embeddings to prioritise the decision boundary for certain classes over the others. One way to achieve this is to weight the training loss

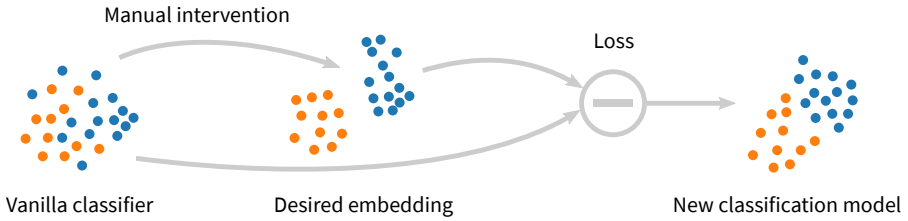

Fig. 1. The basic idea is to define a desired embedding, which is subsequently used to inform the training or fine-tuning process of a classification model. Our method generates embeddings in an end-to-end way which ensures 2D spatial consistency across training epochs and allows to directly back-propagate gradients through a classification network.

for these samples differently, similar to addressing a class imbalance problem, but this often leads to labour-intensive hyper-parameter tuning and does not always yield improved performance for the desired class [6]. We show a way to provide a desired embedding that informs the training process for the entire representation while maintaining the overall goal of manipulating individual class performance. This also leads to an improved understanding of how observations in the embedded latent space relate to actual performance changes. The overall idea of our work is outlined in Fig. 1.

Related Work: Low dimensional representations of high dimensional latent spaces have been subject to scientific research for many decades. The most popular approaches, in chronological order, are principal component analysis [21], multidimensional scaling [13], Isomap [19], t-distributed stochastic neighbourhood embedding (t-SNE) [11], and Uniform Manifold Approximation [12]. Commonly these methods are treated as independent modules and applied to a selected part of the representation, e. g., the penultimate layer of a discriminator network. The problem of this approach is that these mappings may be spatially inconsistent during training from epoch to epoch and that changes to the representation cannot inform the training process through back-propagation. Van der Maaten et al. [10,14] proposed to learn mappings through a neural network. This approach has the advantage that it can be directly integrated into an existing network architecture enabling end-to-end forward and backward updates. While unsupervised dimensionality reduction techniques have been used as part of deep learning workflows [20,9,3,18], we are not aware of any previous work that exploited parametric embeddings for a direct manipulation of learned representations.

Contribution: To the best of our knowledge, our work is the first that allows manual interventions in learned representation spaces in an intuitive way. We show that our method can improve latent space clustering and facilitates targeted performance gains. Manual interventions are desired by ML practitioners for

model fine-tuning as well as by clinical target users who benefit from increased interpretability. We show evidence for the usefulness of our approach in the context of anatomical standard plane classification during fetal ultrasound imaging.

2 Method

Projective Latent Space Decluttering (PLSD) can be applied to any neural network classifier. Considering a dataset $X = (x_1, \ldots, x_N)$ of N instances belonging to M classes and a neural network C that was trained to predict the ground truth labels g_i of x_i, where $g_i \in \{\gamma_1, \ldots, \gamma_K\}$. Let $C_m(x_i)$ be the activations of the network's mth layer, and let the network have M layers in total.

Given such a trained network, PLSD consists of three steps: (1) training of a secondary network $\tilde{E}$ that approximates a given non-linear embedding $E = (y_1, \ldots, y_N)$ for the outputs $C_m(x_i)$ of layer m; (2) modifying the positions y_i of embedded points, yielding new positions y'_i; and (3) retraining C, such that $\tilde{E}(C_m(x_i)) \approx y'_i$. In the following sections, we will discuss these three steps in detail.

2.1 Parametric Embeddings

Non-linear embedding techniques based on neighbourhood graphs ae routinely applied for creating low-dimensional representations of high-dimensional latent spaces [5,17]. The two most widely used non-linear embedding techniques in this context is t-SNE [11].

For t-SNE, distances between high-dimensional points z_i and z_j are converted to probabilities of the neighbourhood p_{ij} by placing Gaussian kernels of different variance on the high-dimensional points. The variance of each kernel is then adjusted such that the perplexity of each distribution equals a given value. This perplexity value can be understood as a smooth measure for how many nearest neighbours are covered by the high-dimensional distributions.

To obtain the embedding, a set of low-dimensional points is initialised, and low-dimensional pair-wise neighbourhood probabilities q_{ij} are calculated based on a heavy-tailed t-distribution. The low-dimensional positions are then adjusted by minimising the Kullback–Leibler divergence $\mathrm{KL}(p_{ij}||q_{ij})$ between the high- and low-dimensional probability distributions.

Given a set of d-dimensional points $z_i \in \mathbb{R}^d$, t-SNE yields a set of d'-dimensional points $z' \in \mathbb{R}^{d'}$. However, it does not yield a general function $E : \mathbb{R}^d \to \mathbb{R}^{d'}$ that is defined for all $z \in \mathbb{R}^d$. This means that standard t-SNE is not extensible to out-of-sample input, i. e., it is not possible to add new points to existing embeddings. This lack of out-of-sample extensibility prevents implementations that allow backpropagation of losses through the embedding.

In order to allow out-of-sample extension, van der Maaten introduced the idea of approximating t-SNE with neural networks [10]. We adapt van der Maaten's approach and introduce two important extensions, based on recent advancements related to t-SNE [16]:

- *PCA initialisation:* In standard frameworks, low-dimensional points are initialised with the first two principal components. This leads to an improved reproducibility across multiple runs, as well as a better reservation of global structure.
- *Approximate nearest neighbours:* one performance bottleneck for classical t-SNE is the calculation of the pairwise distance matrix, which grows with the square of the number of input values. To reduce this computational overhead, calculating only the distances to a given number of approximate nearest neighbours [4] can circumvent this computational overhead without noticeable effects on the embedding quality.

Our approach is an unsupervised learning workflow for obtaining a network that approximates t-SNE for a set of input vectors $(z_1, \ldots, z_N)$ given a perplexity value Perp. The individual steps can be summarised as:

1. Calculate a matrix of pairwise distances, only taking into account the k approximate nearest neighbours, where $k = \min(3 \times \text{Perp}, N - 1)$.
2. Find correct variances for the kernels. In contrast to the simple binary search used by van der Maaten [10], we use Brent's method [2].
3. Calculate the high-dimensional pair-wise probabilities p_{ij}.
4. Optionally pretrain the network such that its 2D output matches the first two principal components of z_i.
5. Train the embedding network. For each input batch the low-dimensional pairwise probabilities q_{ij} are calculated, and the KL-divergence $\text{KL}(p_{ij}||q_{ij})$ is used as a loss function.

While van der Maaten used a network architecture with three hidden layers of sizes 500, 500, and 2000 [10], we found that much smaller networks can be trained much more effectively while yielding more reliable results. For all embeddings we used a perplexity of 50 and a dense network architecture with two hidden layers of sizes 300 and 100. The resulting simple network can be connected to any complex neural network, such as convolutional neural networks for medical image classification.

2.2 Projective Latent Constraints

Once the network $\tilde{E}$ for approximating t-SNE has been trained, new constraints on the embedded latent space can be defined. This is most easily done by visualising the embedded points, $y_i = E(C_m(x_i))$, in a scatter plot with points coloured categorically by their ground truth labels g_i. For our applications, we chose only simple modifications of the embedding space: shifting of entire class clusters, and contraction of class clusters towards their centres of mass.

For shifting the cluster corresponding to class γ_j, we can define

$$y_i' = \begin{cases} y_i + \delta, & \text{if } g_i = \gamma_j \\ y_i & \text{else} \end{cases} . \tag{1}$$

For contraction of a class cluster for class γ_j by a factor of κ towards its centre of mass, we can define

$$y'_i = \begin{cases} (1-\kappa)\ y_i + \kappa\ \bar{y}_j, & \text{if } g_i = \gamma_j \\ y_i & \text{else} \end{cases}, \text{ where } \bar{y}_j = \text{mean}(\{y_i \,|\, g_i = \gamma_j\}). \quad (2)$$

The modified embedding positions y'_i are used as target values for the subsequent regression learning task.

2.3 Retraining the Classifier

In the final step, the original classifier is retrained with an adapted loss function $\mathcal{L}_{\text{PLSD}}$ based on the modified embedding,

$$\mathcal{L}_{\text{PLSD}}(x_i, g_i, y'_i) = (1-\lambda)\ \mathcal{L}_{\text{class}}(C_M(x_i), g_i) + \lambda\ \mathcal{L}_{\text{emb}}(\tilde{E}(C_m(x_i)), y'_i). \quad (3)$$

The new loss function combines the original classification loss function $\mathcal{L}_{\text{class}}$, typically a cross entropy term, with an additional term $\mathcal{L}_{\text{emb}}$. Minimisation of $\mathcal{L}_{\text{emb}}$ causes the classifier to learn new activations that yield an embedding similar to y'. As $\tilde{E}$ is simply a neural network, backpropagation of the loss is straightforward. In our experiments, we use the squared euclidean distance for $\mathcal{L}_{\text{emb}}$ and test different values for the weighting coefficient λ. We also experiment with only counting the embedding loss for instances of classes that were altered in the embedding.

3 Experiments: Decluttering Standard Plane Detection in Ultrasound Images

We test our approach on a challenging diagnostic view plane classification task in fetal ultrasound screening. The dataset consists of about 12,000 2D fetal ultrasound images sampled from 2,694 patient examinations with gestational ages between 18 and 22 weeks. Eight different ultrasound systems of identical make and model (GE Voluson E8) were used for the acquisitions to eliminate as many unknown image acquisition parameters as possible. Anatomical standard plane image frames were labelled by expert sonographers as defined in the UK FASP handbook [15]. We selected a subset of images that tend to be confused by established models [1]: Four Chamber View (4CH), Abdominal, Femur, Spine, Left Ventricular Outflow Tract (LVOT) and Right Ventricular Outflow Tract (RVOT) / Three Vessel View (3VV). RVOT and 3VV were combined into a single class after clinical radiologists confirmed that they are identical. We split the resulting dataset into 4,777 training and 1,024 test images.

The architecture of our baseline classifier is SonoNet-64 [1]. The network was trained for 5 epochs with pure classification loss, i. e., $\mathcal{L} = \mathcal{L}_{\text{class}}$. We used a batch size of 100, a learning rate of 0.1, and 0.9 Nesterov momentum. During these first five training epochs, we used random affine transformations for data augmentation ($\pm 15°$ rotation, ± 0.1 shift, 0.7 to 1.3 zoom).

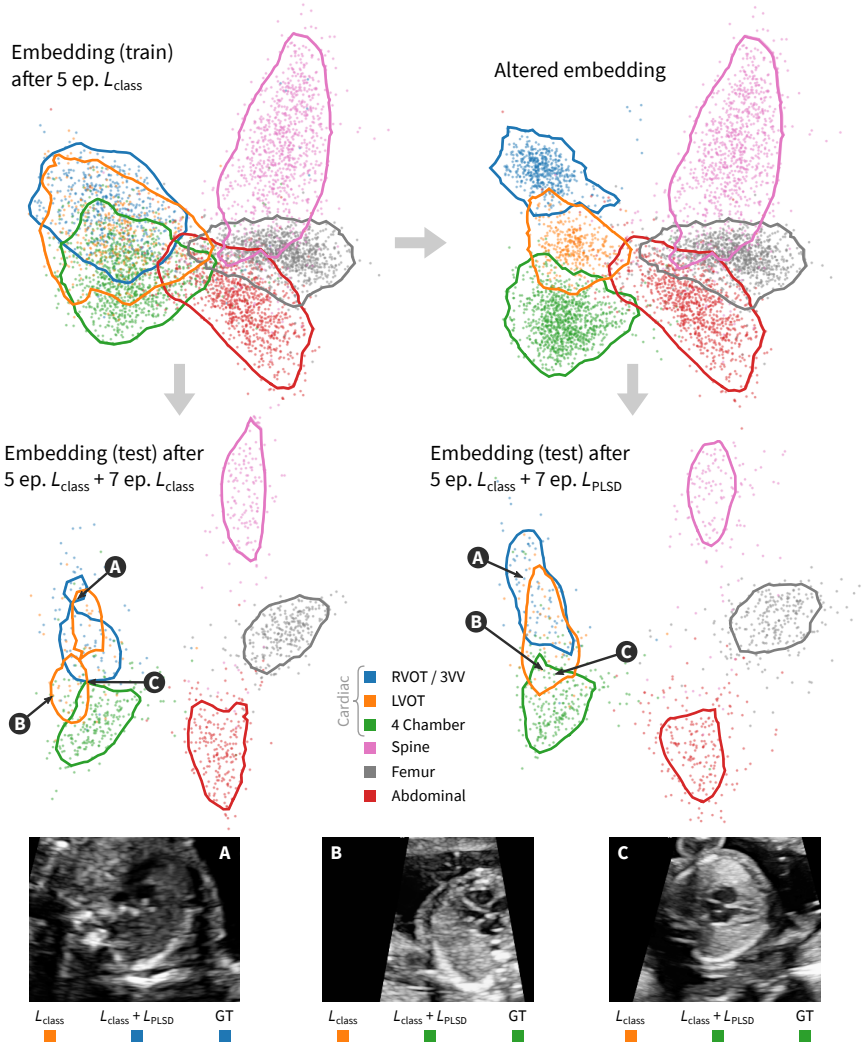

Fig. 2. Projective Latent Space Decluttering for standard plane classification in fetal ultrasound images. Top left: embedding of the baseline network's output (train) after 5 epochs of classification training ($\mathcal{L} = \mathcal{L}_{\text{class}}$). Top right: altered output embedding (train set) with manually separated cardiac classes. Centre left: Output embedding (test) after resuming standard classification training for 7 epochs ($\mathcal{L} = \mathcal{L}_{\text{class}}$), starting from the baseline classifier (top left). Centre right: Decluttered embedding (train set) after resuming training with an updated loss function ($\mathcal{L} = \mathcal{L}_{\text{PLSD}} = 0.9\ \mathcal{L}_{\text{class}} + 0.1\ \mathcal{L}_{\text{emb}}$), starting again from the baseline classifier (top left). For easier comparability, class-specific contours lines at a density threshold of $1/N$ are shown, where N is the total number of train or test images, respectively. Performance measures for the classifiers are given in Table 1. Bottom: Three example images that were successfully classified after applying PLSD. For each image, the positions in both embeddings are indicated.

The 6-dimensional final-layer logits for the non-transformed training images were used as inputs for the training of the parametric t-SNE network. We used a fully connected network with two hidden layers of sizes 300 and 100. The t-SNE network was trained for 10 epochs with a learning rate of 0.01 and a batch size of 500. The perplexity was set to 50. We pretrained the network for 5 epochs to approximate a PCA initialisation.

The ultrasound dataset is imbalanced, with 1,866 images in the three cardiac classes, and 2,911 images in the three non-cardiac classes. There are about twice as many 4CH images as RVOT/3VV, and three times as many 4CH images as LVOT. As a result, after five epochs of classification learning, our vanilla classifier could not properly distinguish between the three cardiac classes. This is apparent in the baseline embedding shown in Fig. 2 (top left).

We experimented with PLSD to improve the performance for the cardiac classes, in particular for RVOT/3VV and LVOT. Figure 2 (top right) shows the case of contracting the class clusters of RVOT/3VV and LVOT by a factor of $\kappa = 1/2$, and that of 4CH by a factor of $\kappa = 1/4$, according to Eq. 2. Additionally, the three clusters were shifted by $\delta_{\mathrm{RVOT}} = (-7, +10)$, $\delta_{\mathrm{4CH}} = (-5, -10)$, and $\delta_{\mathrm{LVOT}} = (+2, -2)$, respectively, according to Eq. 1.

Table 1. Global and class-specific performance measures for standard plane classification in fetal ultrasound images with and without PLSD, evaluated on the test set. The last two columns are weighted averages of the values for the three cardiac and the three non-cardiac classes, respectively. (* The class labelled as RVOT also includes 3VV.)

		RVOT*	4CH	LVOT	Abd.	Femur	Spine	Cardiac	Other
Precision	Class. only	**0.82**	0.82	0.42	0.93	0.98	0.97	0.77	0.96
	PLSD	0.78	**0.85**	**0.61**	0.91	0.97	0.96	**0.80**	0.95
Recall	Class. only	0.38	0.94	**0.46**	0.96	0.97	0.94	0.76	0.96
	PLSD	**0.73**	0.94	0.28	0.96	0.97	0.94	**0.81**	0.96
F_1-score	Class. only	0.56	0.88	**0.44**	0.95	0.97	0.95	0.75	0.96
	PLSD	**0.76**	**0.89**	0.41	0.94	0.97	0.95	**0.80**	0.95

For the PLSD results, training was resumed for 7 epochs with a mixed loss function as defined in Eq. 3. We experimented with different values for λ; all results given in this section are for $\lambda = 0.1$, which was found to be a suitable value in this application scenario. For a fair comparison, training of the baseline network was also resumed for 7 epochs without altering the loss function, i. e., continuing to learn only classification. In both cases, the remaining training epochs were performed without data augmentation, but with all other hyperparameters kept the same as for the vanilla classifier.

The outputs were then embedded with the parametric t-SNE mapping learned on the baseline outputs. These embeddings are shown in Fig. 2 (centre). By resuming the training with included embedding loss, the clusters for the three

cardiac classes retain the ordering that the initial baseline embedding suggested. The contraction constraint also led to more convex clusters for the test outputs. Figure 2 (bottom) also shows three exemplary images that were misclassified in case of the pure classification loss model, but correctly classified after applying PLSD.

Table 1 lists the class-specific precision, recall, and F_1-scores for the two different networks. By applying PLSD the average quality for the cardiac classes could be improved without negatively affecting the performance for the remaining classes. In some experiments, we observe much larger quality improvements for individual classes. For example, in one case the F_1-score for LVOT improved by a factor of two. In these extreme cases, however, local improvements were often accompanied by significant performance drops for other classes.

4 Discussion

As PLSD is based on manual intervention, we would like to list some guidelines for these transformations, which we found during our experiments. First, the embedding space needs to have empty regions that are close to the regions that should be decluttered. Large alterations cannot be preserved well due to the rather localised information on which the approximated embedding is based. Second, in order to learn the alterations correctly for some classes, other classes need to be allowed to move more freely. In the ultrasound experiments, for example, we found that restricting the embedding loss only to the cardiac classes—while leaving the remaining classes free to move—led to the best results. Finally, the embedding itself is non-linear and, despite the PCA initialisation, might only represent a local minimum of the Kullback–Leibler divergence. This means that seemingly obvious changes made in the embedding often completely contradict the original classification task. Thus, both components of the PLSD loss need to be observed carefully during training.

In case of easier classification tasks, such as MNIST [8] and CIFAR-10 [7], we found that the process was much more forgiving towards more extreme changes of the embedding. We also found that relatively high values for λ still led to improved classification performance.

In future work, we would like to experiment with parameterised versions of different dimensionality reduction techniques. We are also planning to create an interactive, visual prototype to focus on the manual aspect of PLSD.

5 Conclusion

We introduced Projective Latent Space Decluttering, a promising technique to inject additional information into neural network classifiers by means of constraints derived from manual interventions in the embedded latent space. We applied PLSD successfully to obtain targeted improvement for a subset of classes in standard plane classification for ultrasound images without negatively affecting the overall performance. This example particularly shows the potential

of Projective Latent Space Decluttering in medical image classification scenarios with imbalanced datasets.

The source code of PLSD will be publicly available by the time of the conference.

References

1. Baumgartner, C.F., Kamnitsas, K., Matthew, J., Fletcher, T.P., Smith, S., Koch, L.M., Kainz, B., Rueckert, D.: Sononet: real-time detection and localisation of fetal standard scan planes in freehand ultrasound. IEEE transactions on medical imaging **36**(11), 2204–2215 (2017). https://doi.org/10.1109/TMI.2017.2712367
2. Brent, R.P.: Algorithms for minimization without derivatives. Courier Corporation (2013)
3. Chen, X., Weng, J., Lu, W., Xu, J., Weng, J.: Deep manifold learning combined with convolutional neural networks for action recognition. IEEE transactions on neural networks and learning systems **29**(9), 3938–3952 (2017), 10.1109/TNNLS.2017.2740318
4. Dong, W., Moses, C., Li, K.: Efficient k-nearest neighbor graph construction for generic similarity measures. In: Proceedings of the 20th international conference on World wide web. pp. 577–586 (2011), https://www.cs.princeton.edu/cass/papers/www11.pdf
5. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why Does Unsupervised Pre-training Help Deep Learning? Journal of Machine Learning Research **11**, 625–660 (2010), http://jmlr.org/papers/volume11/erhan10a/erhan10a.pdf
6. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. Journal of Big Data **6**(1), 27 (2019). https://doi.org/10.1186/s40537-019-0192-5
7. Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 (canadian institute for advanced research), http://www.cs.toronto.edu/~kriz/cifar.html, accessed: 2020-03-16
8. LeCun, Y., Cortes, C.: The MNIST database of handwritten digits (2005), http://yann.lecun.com/exdb/mnist/, acessed: 2020-03-16
9. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: Artificial intelligence and statistics. pp. 562–570 (2015), proceedings.mlr.press/v38/lee15a.pdf
10. van der Maaten, L.: Learning a parametric embedding by preserving local structure. In: Artificial Intelligence and Statistics. pp. 384–391 (2009), http://proceedings.mlr.press/v5/maaten09a.html
11. van der Maaten, L., Hinton, G.: Visualizing Data using t-SNE. Journal of Machine Learning Research **9**(Nov), 2579–2605 (2008), https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf
12. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 (Dec 2018), http://arxiv.org/abs/1802.03426
13. Mead, A.: Review of the development of multidimensional scaling methods. Journal of the Royal Statistical Society: Series D (The Statistician) **41**(1), 27–39 (1992). https://doi.org/10.2307/2348634
14. Min, M.R., van der Maaten, L., Yuan, Z., Bonner, A.J., Zhang, Z.: Deep supervised t-distributed embedding. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10) (2010), https://www.cs.toronto.edu/~cuty/DSTEM.pdf

15. NHS: Fetal anomaly screening programme: programme handbook June 2015. Public Health England (2015)
16. Poliar, P.G., Straar, M., Zupan, B.: openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding. bioRxiv (Aug 2019). https://doi.org/10.1101/731877, http://biorxiv.org/lookup/doi/10.1101/731877
17. Rauber, P.E., Fadel, S.G., Falco, A.X., Telea, A.C.: Visualizing the Hidden Activity of Artificial Neural Networks. IEEE Transactions on Visualization and Computer Graphics **23**(1), 101–110 (Jan 2017). https://doi.org/10.1109/TVCG.2016.2598838
18. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-learning with latent embedding optimization. arXiv:1807.05960 (2018), https://arxiv.org/abs/1807.05960
19. Tenenbaum, J.B.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science **290**(5500), 2319–2323 (Dec 2000). https://doi.org/10.1126/science.290.5500.2319, http://www.sciencemag.org/cgi/doi/10.1126/science.290.5500.2319
20. Tomar, V.S., Rose, R.C.: Manifold regularized deep neural networks. In: Fifteenth Annual Conference of the International Speech Communication Association (2014)
21. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometrics and intelligent laboratory systems **2**(1–3), 37–52 (1987). https://doi.org/10.1016/0169-7439(87)80084-9