Embedding Complexity of Learned Representations in Neural Networks*

Tomáš Kuzma and Igor Farkaš

Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava
{kuzma,farkas}@fmph.uniba.sk

Abstract

In classification tasks, the set of training examples for each class can be viewed as a limited sampling from an ideal infinite manifold of all sensible representants of this class. A layered artificial neural network model trained for such a task can then be interpreted as a stack of continuous transformations which gradually mold these complex manifolds from the original input space to simpler dissimilar internal representations on successive hidden layers – the so-called *manifold disentaglement hypothesis* This, in turn, enables the final classification to be made in a linear fashion. We propose to assess the extent of this separation effect by introducing a class of measures based on the *embedding complexity* of the internal representations, with evaluation of the KL-divergence of t-distributed stochastic neighbour embedding (t-SNE) appearing as the most suitable method. Finally, we demonstrate the validity of the disentanglement hypothesis by measuring embedding complexity, classification accuracy and their relation on a sample of image classification datasets.

Keywords: neural networks, manifold disentanglement, embedding complexity

Introduction

As an analogue to biological neural networks found in nature, artificial neural networks are constructed as graphical models of directionally connected units – neurons. While biological neurons can have complex and time dependent behaviours, artificial neurons are usually (but not always, for example *spiking neural networks*) modelled in an extremely simplified fashion:

- the output of an artificial neuron is represented by a single scalar real value (basically a frequency, or more precisely a time-average of virtual spike trains)
- each input synapse is given a single real-valued coefficient, *weight*
- output of a neuron is determined by an *activation function* on the weighted-sum of the inputs

* This work was supported by grants VEGA 1/0796/18 and KEGA 042UK-4/2019.

This simplified model is then trained by a variety of mostly similarly simple, often biologically-implausible methods (usually variants of error back-propagation).

Parts of biological neural networks, for example in the mammalian visual cortex [3, 4], are structured in a simple layered fashion – each layer of neurons takes its inputs directly from the previous layer. This leads to a simple formulation of a neural network, called the *feedforward neural network*, which is easy to conceptualize and implement and also leads to increased computational efficiency due to the apparent potential for parallelization. Overwhelming proportion of neural models used in practice are either directly of this type, contain only small modifications (e.g. in residual networks each layer also connects to some of the indirectly preceding layers), or are constructed of blocks of this type (e.g. RNNs).

This simple layered construction also lends itself to a reinterpretation of the mechanics of a neural network – instead of viewing individual connected units, we can view the activations of entire layers of neurons at once as vectors in a n-dimensional space, where n is the number of units of a particular layer. The transition between each pair of successive layers then consists of two portions: a simple linear transformation (by the complete matrix of individual neuron weights) and an activation function (assuming that it's shared across the layer), which is almost always monotonous and continuous. The transition between two layers can then be viewed as a smooth non-linear transformation.

This interpretation allows us to examine the process of classifying an input from a *manifold perspective*. Each sample from the dataset represents a point in a high-dimensional space, belonging to a certain class. While the number of samples of a class in a practical dataset is usually limited, we may consider any such set a sampling from an infinite set of potential inputs (e.g. all images of dogs). This ideal infinite set S is then assumed to be continuous (a smoothness prior), i.e. for any input x and real positive ϵ there is an x' also in the set S, which is close to x (i.e. $\|x - x'\| \leq \epsilon$). This ideal set then forms a *low-dimensional manifold* in the input space and the classification problem can be viewed as partitioning the input space such that no partition contains parts of more than one manifold.

Complex classifiers usually construct this partition in multiple stages, with the interim stages transforming and simplifying the input and the final stage performing the partition in a less complex way. In the case of conventional artificial neural network classifiers, the final layer has one output neuron for each of the classes, and the classification is determined by which neuron has the largest activation.[1] This in turn produces a Voronoi-esque partition of the output space, whereas the preceding layers disentangle [1] the class manifolds from their complex structure in the input space into separable regions in the output space.

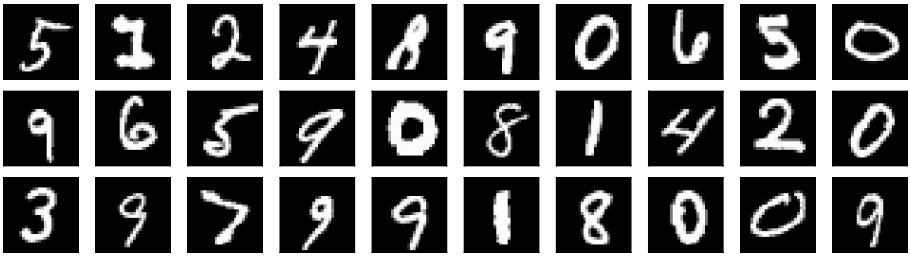
[1] Output neurons usually have softmax activation, but this is immaterial for the argmax selection, wherein any strictly-increasing function produces the same results.

Datasets

To study the process of untangling class manifolds, datasets of medium complexity are required. The complexity should be high enough so that the problem cannot be solved in the input space by a simple classifier, but a complex transformation, such as by an artificial neural network, is necessary. However, at the same time, the untransformed or partially transformed inputs cannot be inscrutable to available embedding methods as to remain interpretable. These restrictions led us to select two suitable datasets, both inadvertently being visual tasks.

MNIST

MNIST [6] is the quintessential basic dataset for optical character recognition, consisting of 50 000 training and 10 000 testing images in ten classes (digits). Each input is a grayscale 28×28 pixel bitmap of a hand-written digit, pre-processed to be centered and upright. Few examples:



SVHN

The *StreetView House Numbers dataset* [8] (further referred to only as SVHN) is a more challenging task for digit recognition, which adds color, distracting surroundings, blurring and oblique perspectives. Each input image is a full-color 32×32 cutout from a StreetView photo, including the following random samples:

Models

To evaluate the manifold disentanglement process, we will employ simple (deep) feed-forward networks that are minimally powerful enough to satisfactorily classify

the selected datasets. For simplicity, we will only use fully-connected layers of 100 neurons and use the same activation function at each hidden layer, one of:

– logistic sigmoid:

$$\operatorname{logsig}(x) := \frac{1}{1 + \exp(-x)}$$

– hyperbolic tangent:

$$\tanh(x) := \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

– softsign function (introduced in [2]):

$$\operatorname{softsign}(x) := \frac{x}{1 + |x|}$$

– rectified linear units (ReLU):

$$\operatorname{relu}(x) := \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

With final classification layer having a neuron for each class with a *softmax* activation. All of these networks can be satisfactorily trained within a 100 epochs using simple stochastic gradient descent with momentum.

Methods

To assess the progress of the manifold disentanglement process we propose to measure the *embedding complexity*, i.e. how difficult is to embed the activation vectors for a balanced sample of training inputs to a lower dimensional space. To utilize both numeric and visual examination of the resulting quality of embeddings, we chose to realize the embedding into an output space of two dimensions. We examined several popular embedding methods (in order of increasing sophistication):

– PCA – Principal Component Analysis
– LLE – Locally-Linear Embedding [9] (not pictured)
– MDS – Multi-Dimensional Scaling [5]
– Isomap [10]
– t-SNE – t-distributed Stochastic Neighbour Embedding [7]

Figure 1 shows the differences in the resulting embedding. As the t-SNE embedding proves to be qualitatively superior, we will resort to only using this method. The method is also powerful enough that in the case of the MNIST dataset it manages to mostly separate the clusters even directly on the input data, therefore further qualitative comparisons will be restricted to the SVHN dataset.

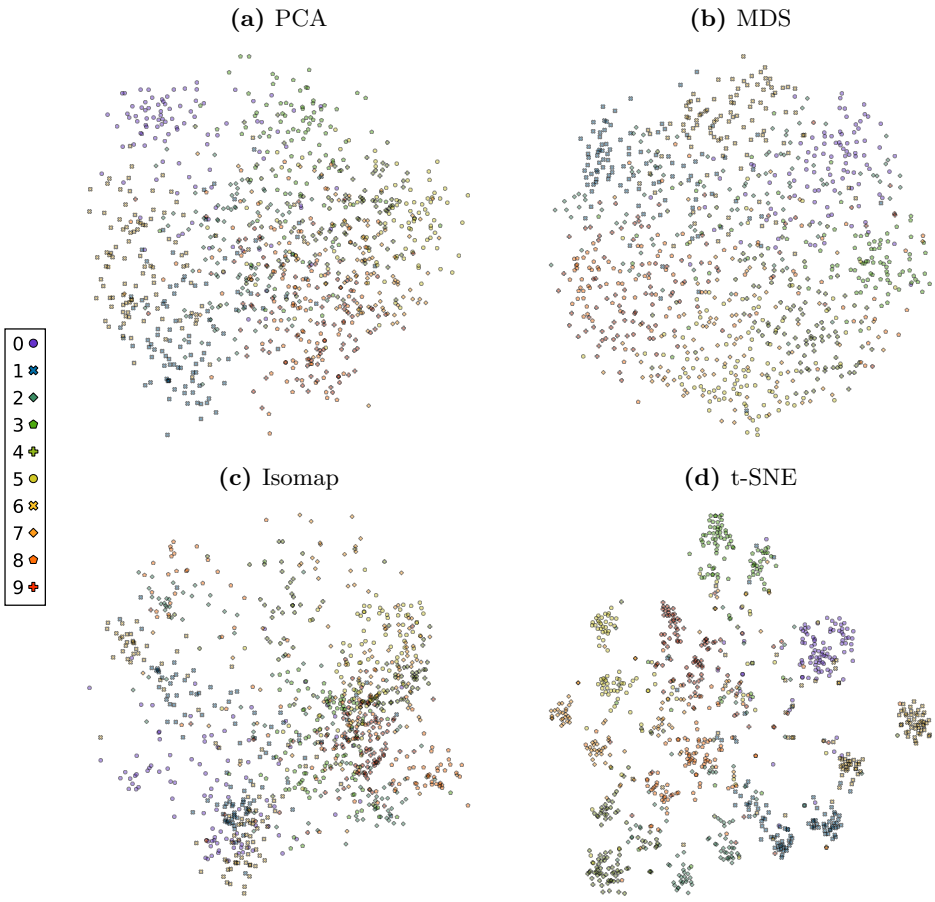


Fig. 1: A comparison of embedding methods applied to the final 7th hidden layer with a softsign activation, in a network classifying images of the StreetView House Numbers dataset. A thousand input images, one hundred from each of the classes, were provided as inputs for this network. Each of the inputs gradually transforms to an activation vector of one hundred real numbers, represented in this plot by a single point of the final embedding. The t-SNE method showcases its clearly superior clustering ability.

t-SNE

The *t-distributed Stochastic Neighbour Embedding* by [7], or *t-SNE*, is a popular non-linear embedding method, which is based on preserving the *stochastic neighbourhood* of elements, i.e. for every (oriented) pair of datapoints, we assign a probability (hence *stochastic*) that the two datapoints are close. This is in contrast to more conventional methods which usually use a fixed neighbourhood, either an adjustable parameter of the algorithm (e.g. k nearest neighbours in

Locally-Linear Embedding or Isomap), or optimized to satisfy an internal condition, or factor all data points into consideration (e.g. Multi-dimensional Scaling or Principal Component Analysis). This probability is modelled using Gaussian distributions in the *input* space (with x_i being the i-th input point):

$$p_{j|i} := \frac{1}{\sqrt{2\pi\sigma_i^2}} \cdot \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma_i^2}\right)$$

$$\hat{p}_{j|i} := \frac{p_{j|i}}{\sum_{k \neq m} p_{k|m}}$$

$$\hat{p}_{i,j} := \frac{\hat{p}_{i|j} + \hat{p}_{j|i}}{2}$$

where the $p_{j|i}$ represents the (directed) probability that j is a neighbour i, the $\hat{p}_{j,i}$ is the normalized neighbourhood score and the $\hat{p}_{i,j}$ is its symmetric (undirected) version. (The σ_i parameter is programmatically tuned for a desired *perplexity*.)

The original, less successful SNE variant also uses Gaussian distribution in the output space y_i, which doesn't take into account the difference in the number of dimensions between the spaces (i.e. embedding). The improved t-SNE method uses a t-Student distribution with a single degree of freedom (also called the Cauchy distribution), with a heavy-tail which alleviates this quantitative difference:

$$q_{i,j} = q_{j|i} = q_{i|j} := \frac{1}{1 + \|y_i - y_j\|^2}$$

$$\hat{q}_{i,j} := \frac{q_{i,j}}{\sum_{k \neq m} q_{k,m}}$$

The disparity between the distributions of probabilities in the input and output spaces is quantified by the *Kullback-Leibner* or *KL-divergence*:

$$KL := -\sum_{i \neq j} p_{i,j} \log \frac{q_{i,j}}{p_{i,j}}$$

The desired embedding is then produced by minimizing this divergence with respect to the placement of points in the output space. This turns out to be a convex optimization problem well-suited for a range of gradient-based methods.

Results

For each combination of dataset (MNIST or SVHN) and model (one to seven layers, one of the four activation functions), we train five independent networks (or *runs*). We then sample activations of each hidden neuron for 100 randomly selected input samples for each of the 10 input classes (MNIST and SVHN both), totalling 1000 activation vectors for each layer. In each independent run, we

then embed those activation into two dimensions using t-SNE and measure the resulting KL-divergence as the hardness score. For a quantitative overview, we plot the result of all runs into a single bar chart, with the averaged value shown as the bold line and individual runs as translucent overlapping rectangles (this is an alternate version of a boxplot, which puts the greatest emphasis on the mean value). To better visualize the qualitative differences between the embeddings, we also plot the actual embeddings (for a single run; see figures 5, 7 and 8).

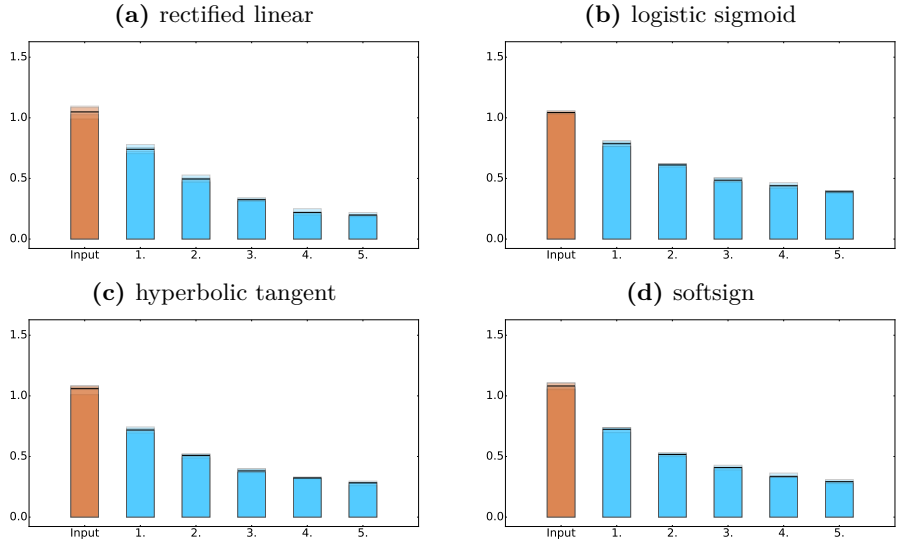


Fig. 2: The hardness score (KL-divergence) of the embedding of the activations of successive layers of a five-layer network classifying MNIST digits with four choices of the activation function. Transforming the inputs through the layers of the network makes subsequent embedding much easier.

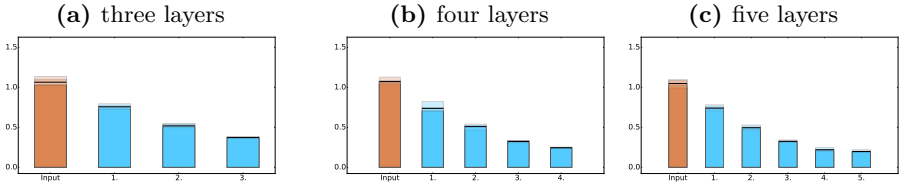


Fig. 3: The same dataset, MNIST, transformed by networks having three to five hidden layers of rectified linear units (ReLU) in total. Fewer layers lead to a quick decrease in hardness, but the final embedding is easier in larger networks.

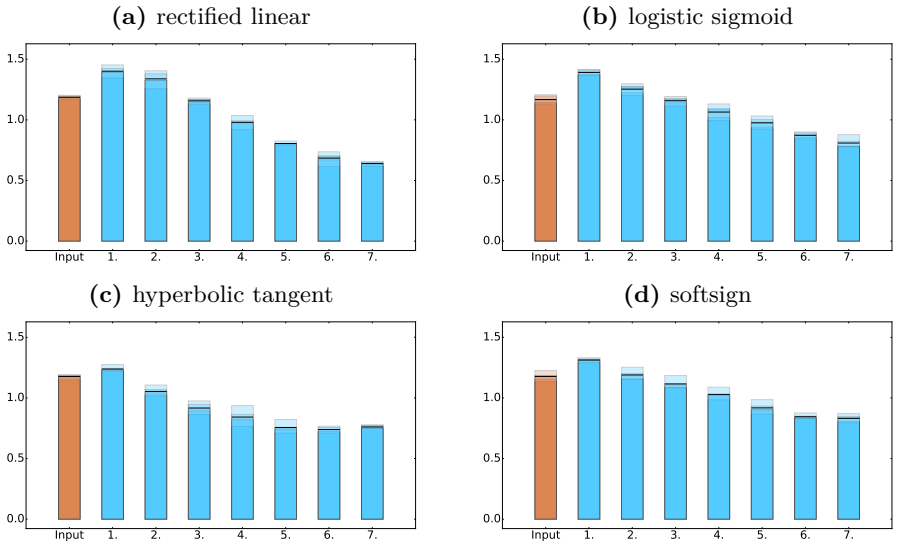


Fig. 4: Hardness scores for seven layer networks with inputs from the SVHN dataset. While the embedding is much harder, the decreasing trend still persists.

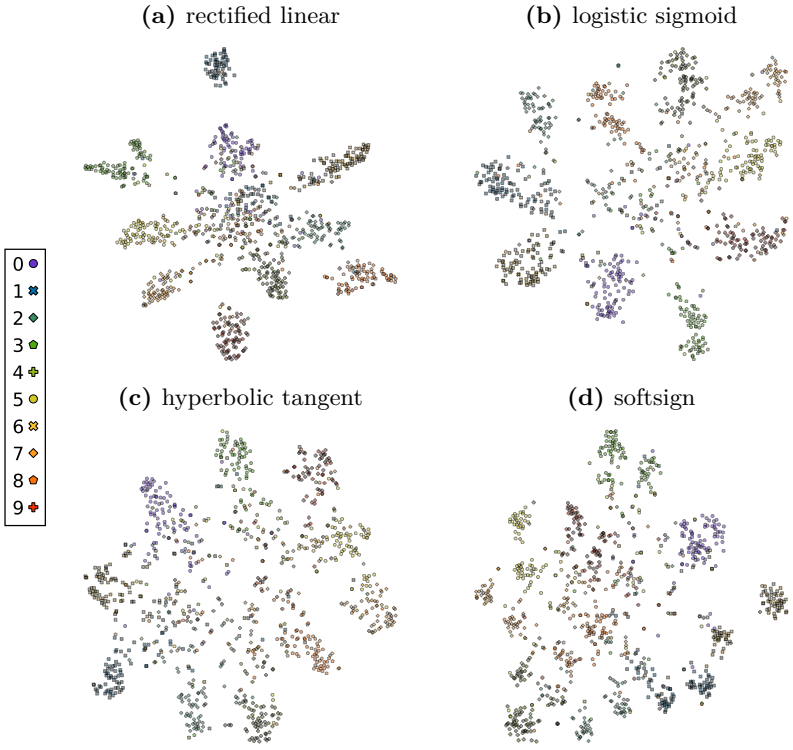


Fig. 5: Embeddings of a single run of SVHN in seven layer networks.

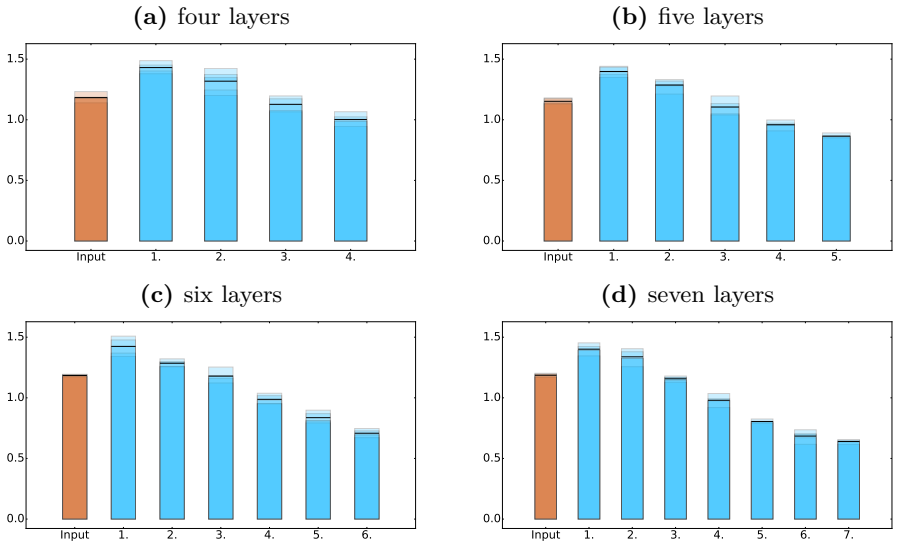


Fig. 6: Four to seven layers of rectified linear units on the SVHN dataset. Smaller networks on this complex task lead to embeddings that are much worse.

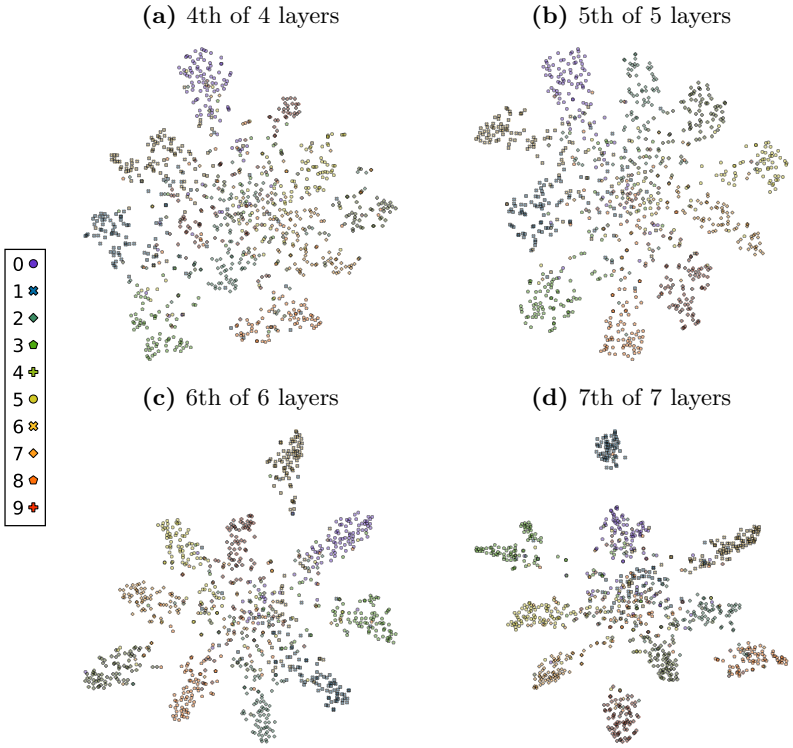


Fig. 7: Actual embeddings of the final layers of networks evaluated above.

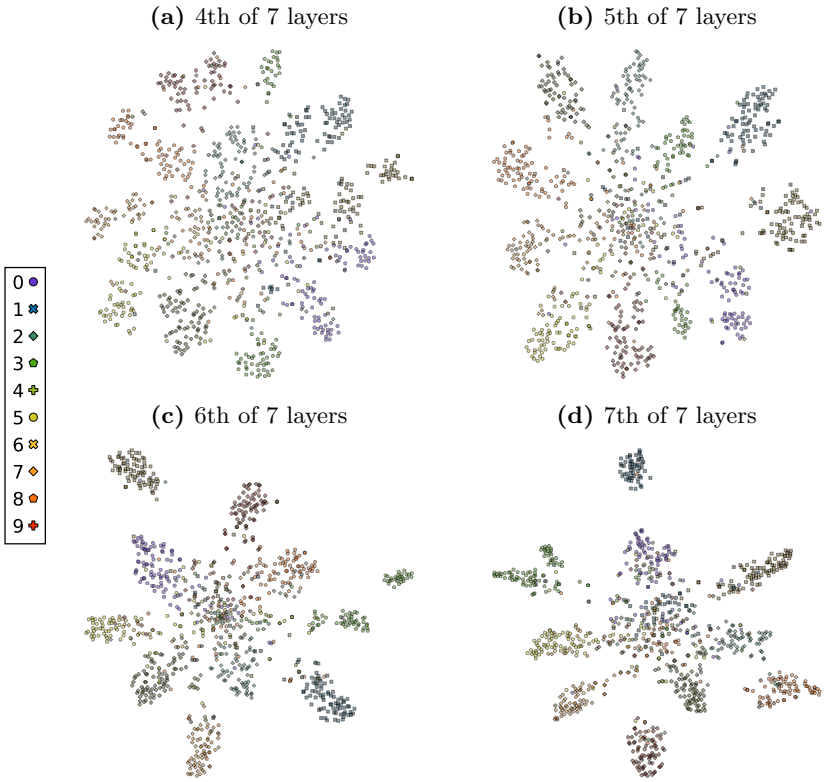


Fig. 8: Embeddings of the activations of the last four layers of a seven layer network, with inputs from the SVHN dataset. Clusterings computed on the higher layers are not only better scoring, but also visually more focused.

The embedding complexity scores, measured by the *KL-divergence* of the *t-SNE* embeddings, are plotted in figures 2 and 3 for the MNIST dataset (all runs). For the more challenging SVHN dataset, the selected scores for multiple runs and selected resulting embeddings for a single run are depicted in figures 4, 5, 6, 7, 8.

Conclusion

In this paper, we propose a novel method for gaining both quantitative and qualitative insight into the inner workings of deep neural networks, by examining the complexity of embedding their *learned representations* – the activations of their hidden layers. The inputs of modern machine learning tasks, especially of the visual variety, are highly complex and cannot be easily embedded to a low-dimensional space. In classification tasks, however, the activations of the final hidden layer are classified in a linear fashion (linear weighting usually followed by softmax and winner-takes-all classification), and therefore must be easily

embeddable. We proceed to quantify this process in which intermediate layers of the network gradually transform the activation manifolds from complex to simple ones. This was previously postulated as the *manifold disentanglement hypothesis*.

Examining several popular methods of embedding, we find that only *t-distributed stochastic neighbour embedding (t-SNE)* is sufficiently capable of dealing with the complex activations encountered. We measure the criterion explicitly optimized in t-SNE, the *KL-divergence* between the pairwise "closeness" distributions of the input and output datapoints, as our embedding hardness score across two datasets, MNIST and StreetView House numbers (SVHN). For every dataset, we perform measurements across several different network architectures (defined by the number of hidden layers and the choice of the activation function) and with multiple independently initialized and trained instances. Our experimental results robustly show that the complexity of internal representations in the network decreases towards the output layer.

Bibliography

[1] Brahma, P.P., Wu, D., She, Y.: Why deep learning works: A manifold disentanglement perspective. IEEE Transactions on Neural Networks and Learning Systems **10**(27), 1997–2008 (October 2016)

[2] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the International conference on artificial intelligence and statistics. pp. 249–256 (2010)

[3] Hubel, D.H., Wiesel, T.N.: Receptive fields of single neurons in the cat's striate cortex. The Journal of Physiology **148**, 574–591 (1959)

[4] Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology **60**, 106–154 (1962)

[5] Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika **29**(1), 1–27 (Mar 1964)

[6] LeCun, Y., Cortes, C.: MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/ (2010)

[7] van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008)

[8] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011 (2011)

[9] Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (December 2000)

[10] Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (December 2000)