Understanding the Representation and Computation of Multilayer Perceptrons: A Case Study in Speech Recognition

Tasha Nagamine [1] **Nima Mesgarani** [1]

Abstract

Despite the recent success of deep learning, the nature of the transformations they apply to the input features remains poorly understood. This study provides an empirical framework to study the encoding properties of node activations in various layers of the network, and to construct the exact function applied to each data point in the form of a linear transform. These methods are used to discern and quantify properties of feed-forward neural networks trained to map acoustic features to phoneme labels. We show a selective and nonlinear warping of the feature space, achieved by forming prototypical functions to account for the possible variation of each class. This study provides a joint framework where the properties of node activations and the functions implemented by the network can be linked together.

1. Introduction

In recent years, deep learning has achieved remarkable performance on a variety of tasks in machine learning (D. Andor & Collins, 2016; D. Silver & Hassabis, 2016; K. He & Sun, 2016), including automatic speech recognition (G. E. Dahl & Acero, 2012; G. Hinton & Kingsbury, 2012; A.-R. Mohamed & Hinton, 2010; W. Xiong & Zweig, 2016). Despite these successes, our understanding of deep neural networks (DNNs) and the nature of their computation and representations lags far behind these performance gains. This has motivated a number of recent studies aimed at better understanding the computational principles of deep learning in the hope of gaining intuitions that may lead to improved models.

It has been established that networks with at least one hidden layer are universal approximators (Cybenko, 1989; K. Hornik & White, 1989). Several recent theoretical studies have proven that deeper architectures are able to more efficiently solve problems than shallow models, given a limited number of parameters (Eldan & Shamir, 2015; Lin & Tegmark, 2016). Other studies have focused on networks with rectified linear units, and have shown that deeper networks are able to learn more complex functions by splitting the input space into exponentially more linear response regions than equivalent shallow models (G. Montufar & Bengio, 2014; R. Pascanu & Bengio, 2013). Finally, recent successes using deep residual networks and subsequent analyses show the effectiveness of extremely deep representations in supervised learning tasks (K. He & Sun, 2016; A. Veit & Belongie, 2016).

At the same time, there is a growing body of empirical studies that aim to understand the behavior of neural networks by developing mathematical methods and techniques for visualization (Zeiler & Fergus, 2013; J. Yosinski & Lipson, 2016), as well as studies of instability in the face of adversarial examples (C. Szegedy & Fergus, 2013; A. Nguyen & Clune, 2015) and contraction and separation properties of these models (Mallat, 2016). In the field of speech recognition, several studies explored the representations of speech learned by feed-forward networks used in acoustic modeling (A.-R. Mohamed & Penn, 2012; T. Nagamine & Mesgarani, 2015; 2016). Finally, an architecture-independent method was proposed to summarize the complexity of the parameters learned in feed-forward networks (S. Wang & Aslan, 2016).

In this study, we aim to bridge the gap between these theoretical and empirical studies by providing methods for analyzing both the properties of activations in each layer of the network and estimating the exact linear function that is applied to each data point. We discern and quantify properties of the network representation and computation to determine 1) what aspects of the feature space are encoded in different layers (internal representation) and 2) how the network transforms the feature space to achieve categorization (network function). Our method thus provides a joint framework in which the properties of node activations are directly linked to the properties of the function that is

[1]Columbia University, New York, NY, USA. Correspondence to: Nima Mesgarani <nima@ee.columbia.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

learned, which stands in contrast to previous studies that focus on a specific property of the network. These methods are easily extensible to other applications for MLPs, as well as other feed-forward network architectures.

2. Methods

2.1. Deep neural network acoustic models

Neural network acoustic models for phone recognition were trained on the Wall Street Journal speech corpus (J. Garofolo & Pallett, 1993; 1994). The corpus includes approximately 80 hours of read speech training data. We report test accuracies using the eval93 set. Input features to all models were extracted using the Kaldi speech recognition toolkit (D. Povey & Vesely, 2011) and consisted of 23-dimensional log-Mel filter bank coefficients (25 ms window, 10 ms frame rate) with applied mean and variance normalization, spliced over 11 context frames. Models were trained on HMM state targets consisting of monophones with one state per phone (40 output states, corresponding to 39 phonemes and one silence state). All alignments for training were obtained using the WSJ s5 recipe in Kaldi.

To ensure the generality of our findings, we explored two network architectures using 256 and 2048 nodes per hidden layer with five hidden layers. In this work, we analyze models using rectified linear units (ReLUs) because it has been shown that ReLU networks are faster to train, are less sensitive to initialization than sigmoid units, and improve performance in speech recognition tasks (G. E. Dahl & Hinton, 2013; A. L. Maas & Ng, 2013; M. D. Zeiler & Hinton, 2013). For the networks with 2048 nodes per hidden layer, we adopted batch normalization (Ioffe & Szegedy, 2015) and dropout (N. Srivastava & Salakhutdinov, 2014) as regularization methods, with a 20% dropout rate on input and hidden layers. All models were trained using Theano (J. Bergstra & Bengio, 2010). Hyperparameters were determined using grid search. Networks with 256 and 2048 nodes per hidden layer were trained with 25 and 100 epochs of backpropagation, respectively. Because the goal of this study is to examine the transformation from features to phone posteriors of MLP acoustic models (not language models), we report performance using frame-wise classification accuracy rather than word error rate.

2.2. Analyzing the learned network functions

To analyze and compare neural networks, we extended the methods used in (S. Wang & Aslan, 2016) to construct an extended data Jacobian matrix (EDJM) for the nodes in a neural network. The basic concept underlying the EDJM is that a neural network with layers of subsequent nonlinear transformations can be mathematically reproduced by finding a per-data point linear system mapping input to output. More formally, consider a dataset D consisting of inputs $X = [x_1, ..., x_N]$ and outputs $Y = [y_1, ..., y_N]$. For a deep neural network model with λ layers and arbitrary nonlinear function ϕ mapping inputs x_n to predicted output $\hat{y}_n = \phi^\lambda(W^\lambda\phi^{\lambda-1}(W^{\lambda-1}\phi^{\lambda-2}(\dots W^1 x_n)))$, the data Jacobian matrix (DJM) of any output unit can be constructed by finding the gradient of the node with respect to each of the inputs using the following equation:

$$\begin{aligned} DJM_\theta(x_i) &= \frac{\partial \hat{y}_i}{\partial x_i} = \frac{\partial \hat{y}_i}{\partial h_i^{\lambda-1}} \frac{\partial h_i^{\lambda-1}}{\partial h_i^{\lambda-2}} \dots \frac{\partial h_i^1}{\partial x_i} \\ &= \frac{\partial \phi^\lambda\left(h_i^{\lambda-1}\right)}{\partial h_i^{l-1}} \frac{\partial h_i^{\lambda-1}}{\partial \phi^{\lambda-1}\left(h_i^{\lambda-2}\right)} \dots \frac{\partial \phi^1(x_i)}{\partial x_i}. \end{aligned} \tag{1}$$

Here, h_i^ℓ represents the activation of layer ℓ for data sample i, while θ indicates dependence on network parameters. Now assuming a linear output function and rectified linear units (ReLU) with a nonlinearity of the form $\phi_{ReLU}(z) = \max(0, z)$, where z is the weighted input to the node, this equation can be written:

$$DJM_\theta(x_i) = W_{\lambda-1}^\lambda \frac{\partial h_i^{\lambda-1}}{\partial W_{\lambda-2}^{\lambda-1} h_i^{\lambda-2}} W_{\lambda-2}^{\lambda-1} \frac{\partial h_i^{\lambda-2}}{\partial W_{\lambda-3}^{\lambda-2} h_i^{\lambda-3}} \dots W_{\text{IN}}^1. \tag{2}$$

This method can be easily extended to any node in any hidden layer $\ell < \lambda$ by replacing λ with ℓ in Eq. 2. Intuitively, this is equivalent to finding the unique linear mapping function $\hat{W}$ for a network node for each individual data point. Using the piece-wise linear property of the rectified linear function, the previous equation can be easily simplified because taking the gradient with respect to any node activation is equivalent to setting rows of the weight matrix to 0 for zero-activation data points:

$$\hat{W}_{\ell-1}^\ell(x_i)[m,n] = \begin{cases} W_{\ell-1}^\ell[m,n], & \text{if } h_i^\ell[m] > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Thus, for a feed-forward ReLU network, finding the DJM for point i for a given node is mathematically equivalent to setting selected rows of each weight matrix to 0 and doing a simple matrix multiplication:

$$\begin{aligned} \hat{y}_i &= W_{\lambda-1}^\lambda \phi\left(W_{\lambda-2}^{\lambda-1} \phi\left(\dots W_{\text{IN}}^1 x_i\right)\right) \\ &= \hat{W}_{\lambda-1}^\lambda \left(\hat{W}_{\lambda-2}^{\lambda-1}\left(\dots \hat{W}_{\text{IN}}^1 x_i\right)\right) \\ &= DJM_\theta(x_i). \end{aligned} \tag{4}$$

Calculating the EDJM for the whole dataset over every node in a layer results in a tensor of dimension $[N \times d_{in} \times d_{out}]$, where for each data point $i \in [1, ..., N]$, we have constructed a linear map from input to output. This method, outlined in Eq. 1, is applicable to any network

Table 1. Frame-wise accuracy of acoustic models on WSJ dataset.

LAYERS	NODES	TR284	DEV93	EVAL93
5	256%	81.43%	80.45%	80.56%
5	2048%	84.53%	81.70%	81.91%

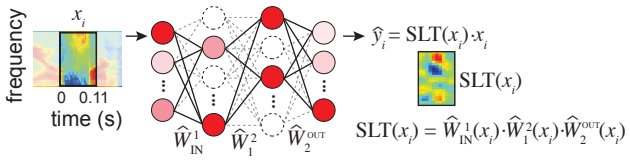


Figure 1. For a single data point x_i, given a network with ReLUs and a linear output, the sample-dependent linear transform $\mathrm{SLT}(x_i)$ can be found by setting rows of the weight matrices to zero according to Eq. 3.

with a nonlinearity that is differentiable with respect to its inputs (e.g., sigmoid and hyperbolic tangent), in which the gradients can be interpreted as linear approximations to the network function for a given data point. For clarity, for the rest of this paper we will refer to the EDJM of individual nodes as the *sample-dependent linear transform* (SLT). The SLT for a given dataset has a fixed dimensionality independent of network architecture, providing a useful tool for comparing different networks.

2.3. Quantifying complexity with SVD

In a neural network, both activations and the SLT of nodes can be written as a two-dimensional matrix of the form $\mathbf{M} = (m_{i,j}) \in \mathbb{R}^{N \times F}$. In the case of activations $\mathbf{H}^{\ell}$ in layer ℓ, N is the number of data points in dataset D, and F is dimensionality of the layer. In the case of the SLT, we perform decompositions on the SLT of individual nodes, where F is the dimension of the inputs. Given a 2D matrix, we can then perform matrix factorization using singular value decomposition (SVD) of the form $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{*}$.

The singular values (sorted by decreasing order) of the diagonal of $\mathbf{\Sigma}$ define the weights given to the orthonormal basis functions defined by $\mathbf{U}$ and $\mathbf{V}$. Because the first dimension of the activation matrix is defined over data points, the singular values of $\mathbf{SLT}_m^{\ell}$ can serve thus as a metric of the complexity (diversity) of the learned network function. Consider the case where $\mathrm{rank}(\mathbf{SLT}_m^{\ell}) = 1$. This means that for node m in layer ℓ, the linear system from input to output is the same for all data points and the function for this node is linear. At the other extreme, a uniform distribution of singular values suggests that the function learned for each data point is drastically different. Thus, the relative values of the SVD spectra for SLT matrices can serve as a metric of nonlinearity for the system. Similarly, the distribution of singular values of $\mathbf{H}^{\ell}$ indicate how uniformly the nodes within a layer respond to different data points.

In general, the matrices $\mathbf{H}^{\ell}$ and $\mathbf{SLT}_n^{\ell}$ are full-rank. Thus, we quantify the shape of the distribution of the SVD spectrum by normalizing by its maximum value, where $\max(\mathbf{\Sigma}) = \sigma_1$, and computing the area under the curve (AUC). We can compute this score for the first D singular values σ of the spectrum using the following equation:

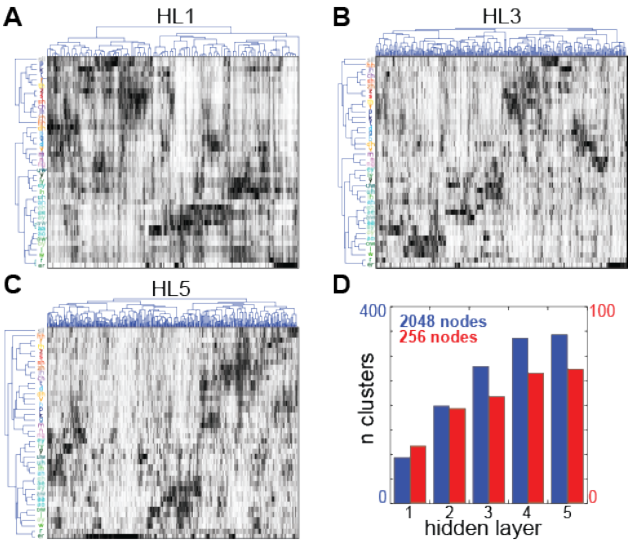


Figure 2. Unsupervised hierarchical clustering of activations in each hidden layer across classes (rows) and nodes (columns, 2048 nodes/layer) for hidden layers 1 (A), 3 (B), and 5 (C). (D) Number of clusters of nodes (columns) in each hidden layer based on a distance cutoff criterion ($d < 0.5$, correlation distance) for networks with 256 and 2048 nodes/layer.

$$\mathrm{AUC} = \frac{1}{\max(\mathbf{\Sigma})} \frac{\sigma_D - \sigma_1}{2D} \sum_{i=1}^{D} (\sigma_{i+1} + \sigma_i) \qquad (5)$$

Because the AUC is computed after normalizing the spectrum so that its maximum value is 1, intuitively, this metric quantifies the non-uniformity of the singular values of the spectrum. Larger values signify greater uniformity in the distribution of singular values and thus a higher degree of complexity.

3. Results: Analysis of node activations

We begin by considering the network activations $\mathbf{H}$, which can be interpreted as nonlinearly transformed feature representations of the network inputs. To characterize this representation, we studied both the properties of the activations of individual nodes (local encoding) and the population of nodes in each layer (global encoding).

3.1. Visualizing the global feature encoding

In a supervised learning task, hidden layers of a neural network extract representations with meaningful, class-based distinctions. To study these distinctions, for each hidden layer, we grouped node activations to the WSJ eval93 set by their corresponding label. To quantify the overall pattern of selectivity in hidden layer ℓ with M nodes, we compute the average activation h_m for every node m for each distinct label $k \in \{1, ..., K\}$. This results in a vector of dimension $[K \times 1]$ for each node that characterizes its aver-

Table 2. Classification accuracy using linear discriminant analysis (LDA) trained on the WSJ dev93 set.

	$N_{\text{NODES}} = 256$		$N_{\text{NODES}} = 2048$	
LAYER	DEV93	EVAL93	DEV93	EVAL93
FEATS	60.18%	56.04%	60.18%	56.04%
HL1	67.38%	64.40%	74.58%	68.75%
HL2	70.15%	67.83%	77.34%	72.08%
HL3	72.68%	70.59%	79.88%	75.65%
HL4	75.06%	73.63%	81.81%	78.63%
HL5	76.78%	75.65%	82.70%	80.04%

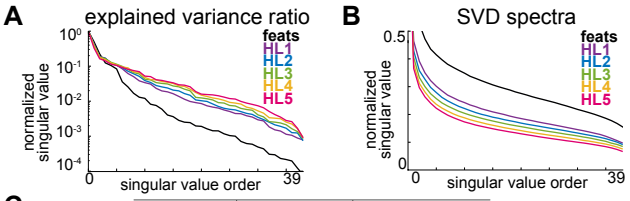
age response to each class, which we call a class selectivity vector (CSV), where each element is calculated as follows:

$$CSV_m[k] = \frac{1}{N_k} \sum_{i=1}^{N_k} h_m(x_i) \, \forall (y_i = k). \quad (6)$$

Concatenating the CSVs into a matrix of dimension $[K \times M]$ summarizes the overall selectivity of a hidden layer, where rows correspond to classes and columns correspond to individual nodes. Examining this matrix reveals patterns in both the individual node and population coding of classes within a layer. To visualize these selectivity patterns, we performed an unsupervised hierarchical clustering on columns (nodes) based on the similarity of their CSVs, and rows (phonemes) based on the similarity of their overall activation pattern over nodes in a layer (UPGMA, correlation distance) in hidden layers 1, 3, and 5 of the 2048 node/layer network (Fig. 2A-C). For visualization purposes, the dendrogram showing clusters of nodes was truncated at a set cutoff distance of 0.5.

Examining these vectors reveals nodes with various types of response profile, including selectivity to both individual classes and groups of classes with shared features. For example, each layer has a group of nodes dedicated to encoding only silence (by far the most common label in the training set). The remaining classes, corresponding to speech sounds, tend to be encoded jointly, with groups of nodes encoding phoneme classes are predominantly organized by phonetic features such as manner and place of articulation. However, other nodes exhibit selectivity patterns not easily explained by acoustic or phonetic feature distinctions.

We observe that in deeper layers, individual nodes tend to become less broadly selective to classes, evidenced by sparser selectivity patterns in their CSVs. The result is that in deeper representations, the average activations of nodes to individual classes tend to become differentiated from one another. This is quantified in the top dendrograms showing the clustering of nodes in Fig. 2A-C, where we observe that the use of a distance cutoff criterion (correlation distance, $d > 0.5$) results in a larger number of branches in each subsequent hidden layer. This holds true for networks with 256 nodes and 2048 nodes per layer (Fig. 2D). This shows that neural networks are successful at using progressive hidden layer representations to decorrelate their inputs in order to extract useful features for classification.



NETWORK	LDA EVR		SVD SPECTRA	
N_{NODES}	256	2048	256	2048
FEATS	1.69		11.73	
HL1	2.00	2.07	8.28	6.55
HL2	2.07	2.17	7.64	5.75
HL3	2.18	2.49	7.09	5.07
HL4	2.22	2.73	6.54	4.44
HL5	2.18	2.63	6.08	4.20

Figure 3. (A) Explained variance ratio (EVR) for LDA model trained on features and activations from hidden layers. (B) SVD spectra for LDA-transformed activations, separated by class and averaged. (C) AUC for spectra in (A) and (B) for networks with 256 and 2048 nodes/layer.

3.2. Deep representations are more class-discriminable

Next, we wanted to directly examine the discriminability of classes in each layer of the network. Measuring the distance between classes directly from the activations, however, may produced biased results. Between networks and even within networks, activations in different layers suffer from the problem that they 1) may have different dimensionalities, or 2) are drawn from different distributions (e.g., layers may exhibit differing amounts of sparsity, or contain "dead" nodes).

To remedy this, we applied linear discriminant analysis (LDA) as a supervised dimensionality-reduction technique (SVD solver) on input features and hidden layer activations. Doing so allows us to project hidden representations of a layer $\mathbf{H}^\ell$ onto a set of $K-1$ orthonormal bases $\widetilde{\mathbf{H}}^\ell$ that maximally retain class discriminability and are more easily comparable between layers and networks. Due to the large size of the training set, we used the WSJ dev93 to train the LDA models, noting that the development set was not used in the training of any of the neural network models. Table 2 shows the classification accuracy using LDA on input features and hidden layer activations for networks with 256 and 2048 nodes/layer. We observe two main patterns. First, for both networks, representations of classes are more separable in each subsequent hidden layer. Second, given layers of equal depth, class discriminability is greater for the larger network.

By examining the ratio of explained variance of each LDA model, we can see that more dimensions are needed to explain a fixed percentage of the variance in deeper layers (Fig. 3A), meaning that in deeper layers, the representation contains more meaningfully discriminant dimensions. However, we also wanted to investigate properties of the

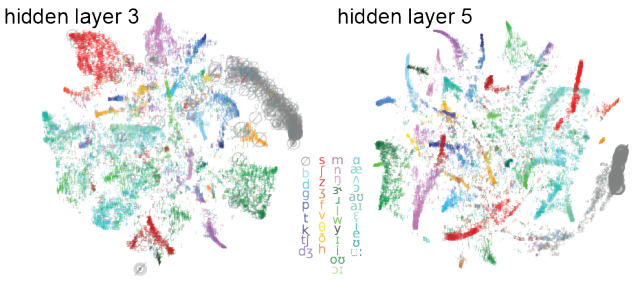


Figure 4. Visualization of activations $\widetilde{\mathbf{H}}$ on the WSJ eval93 set in hidden layers 3 and 5 (2048 nodes/layer) using t-SNE. For visualization, data points were aggregated using K-means clustering.

representation within a class. We did so by separating the transformed activations $\widetilde{\mathbf{H}}^\ell$ into separate matrices for each class k, then performing SVD on each matrix individually in each layer. Figure 3B shows the resultant SVD spectra, averaged over all classes. In deeper layers, on average the spectra require fewer singular values to reach the same ratio of explained variance, indicating that class separability is accompanied by a normalization of activations within classes (to represent a single class, fewer orthogonal bases are required). These results are quantified in Figure 3C using the AUC score from Eq. 5. For the LDA models' explained variance ratio (EVR), we observe that the AUC score tends to increase with layer depth, while the LDA-transformed activations of individual classes exhibit smaller AUC scores in deeper layers. Taken together, these results suggest that representations learned in hidden layers of a neural network become increasingly discriminative, in part through normalization of within-class variability.

This normalization is visualized in Fig. 4 for a network with 2048 nodes/layer, where t-SNE (van der Maaten & Hinton, 2008) is used to project $\widetilde{\mathbf{H}}^\ell$ for $\ell = 3, 5$ into 2D (Euclidean distance). To reduce the dimensionality of the problem, we used K-means clustering to cluster data points of the same class, choosing the number of clusters for each class as $\frac{N_k}{10}$, where N_k is the number of data points of class k in the WSJ eval93 set. The color and symbol (international phonetic alphabet) of each cluster indicate class membership; the size of each point is proportional to the number of data points in the cluster as determined by K-means. We observe a tighter clustering of classes in deeper layers, echoing the quantification in Fig. 3.

3.3. Representations are transformed non-uniformly

In this section, we show that the normalization shown in the previous section is greater for data points that are "difficult" to classify. We accomplish this by first dividing the WSJ eval93 set into two groups based on whether the data points were classified correctly or incorrectly using the LDA model trained on the input features. This is based on the intuition that examples that are classified correctly in a simple linear classifier present a less difficult problem to the neural network than those that are incorrect and thus linearly inseparable. We denote the LDA-transformed activations of hidden layers ℓ to correct and incorrect groups as $\widetilde{\mathbf{H}}^\ell_{\text{COR}}$ (separable) and $\widetilde{\mathbf{H}}^\ell_{\text{INC}}$ (inseparable), respectively. We also consider the whole dataset, which we simply denote as $\widetilde{\mathbf{H}}^\ell$. Furthermore, we filter examples in each of these groups by considering only data points that are classified correctly at the output layer.

To determine if the feature transformation that takes place between hidden layers is applied non-uniformly across examples, we examined the relative location in the representational space of the hidden layers for three example classes when they are easily confusable (/p,t,k/) and when they are seldom confused (/t,n,s/) in the input features (confusion matrix shown in Fig. 5D). Fig. 5A-B visualizes the Euclidean distance between data points in $\widetilde{\mathbf{H}}^\ell$ using multidimensional scaling (MDS) (Kruskal & Wish, 1978) for classes /t,n,s/ and /p,t,k/. Individual class label is shown by color, while labeled points shown in black represent the centroid of each class. The distances between classes for each hidden layer are overlaid at the right. Comparison of the relative distances between the classes in Fig. 5A versus B shows that the network nonlinearities expand the space between the more confusable classes (/p,t,k/) more than for classes that are more linearly separable (/t,s,n/).

While this analysis shows that the hidden layer representations $\widetilde{\mathbf{H}}^\ell$ between more overlapping categories are warped more nonlinearly, we also investigated whether this non-uniform and selective nonlinear transformation occurs within each class as well. Fig. 5C tracks the relative location of data points contained in $\widetilde{\mathbf{H}}^\ell_{\text{INC}}$ as they propagate through the network. Compared to Fig. 5B, we see that there is a rapid nonlinear increase in the relative distance between these more difficult examples of the same class, which occurs more strongly than for the examples in $\widetilde{\mathbf{H}}^\ell$. This observation is quantified in Fig. 5E, where we show the relative expansion (increase in distance as compared to the distance between examples in the LDA-transformed input features $\widetilde{\mathbf{X}}$) of $\widetilde{\mathbf{H}}^\ell_{\text{COR}}$ and $\widetilde{\mathbf{H}}^\ell_{\text{INC}}$ for both /p,t,k/ and /t,n,s/.

With the intuition gained from considering a subset of classes, we next quantify this finding for all classes in Fig. 5F for networks with both 256 and 2048 nodes/layer. However, instead of focusing on the relative expansion between three classes, we consider the expansion of each class to its most frequently confused class at the output layer. Our results show that over all classes, linearly inseparable examples undergo a greater degree of expansion than linearly separable ones. The observed non-uniform, nonlinear, and focused stretching of the feature space illustrates the power of a multilayer neural network to nonlinearly expand specific parts of the feature space that are critical for discrim-

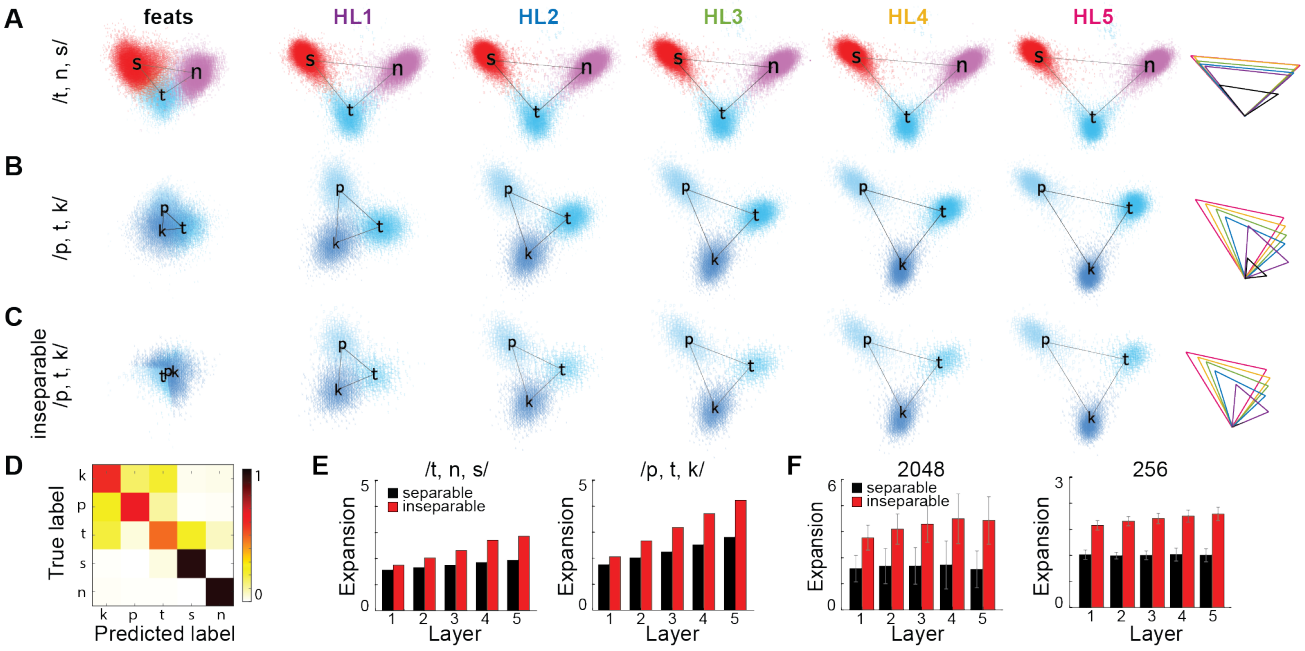


Figure 5. First two MDS dimensions of LDA-transformed features and hidden layer activations for (A) three dissimilar classes, /t/, /n/, and /s/, (B) three similar classes, /p/, /t/, and /k/, and (C) inseparable examples of /p,t,k/ (in $\widetilde{\mathbf{H}}^{\ell}_{\text{INC}}$). For visualization purposes, data points were aggregated using K-means clustering; the size of each point is proportional to the number of examples in that cluster. Centroids for each class are shown in black text. (D) Confusion matrix for LDA classifier trained on input features for classes in (A-C). (E-F) Average relative expansion of centroids of linearly separable and inseparable (LDA, input features) data points, relative to input features for (E) classes /t,n,s/ (left) and /p,t,k/, (F) all classes for a neural network with 2048 nodes/layer (left) and 256 nodes/layer (right).

ination of overlapping categories, while at the same time applying more linear transformations to the parts of the feature space which are less overlapping.

4. Results: Analysis of the network function

The function of a ReLU network can be interpreted as a collection of sample-dependent linear transformations (SLTs) of the input features. In this section, we directly characterize the properties of this function and examine the contribution of different layers to the resulting computation of the network. Because the size of the SLT for a layer of a neural network is very large, we restrict the analyses in this section to the network with 256 nodes/layer.

4.1. Networks learn clusters of functions

We start by looking at the sample-dependent linear transforms (SLTs) mapping inputs to hidden and output layer representations of the neural networks. Fig. 6A visualizes the SLT for the output node for class /r/ ($\mathbf{SLT}^{out}_{/r/}$) for all correctly classified data points with label /r/ in the WSJ eval93 set, sorted by clustering (UPGMA, Euclidean distance). Here, columns represent data points, and rows are the linear weights (templates) applied to each sample to map it to the output node. We see that while the network can potentially learn a different template for each sample, the SLTs tend to cluster in groups (with several broad clusters separated by dashed black lines), revealing similarities between linear transforms that are applied to subsets of data points. The distance between each SLT for class /r/ is visualized using MDS in Fig. 6B, while the average linear mapping function for each cluster is shown in Fig. 6C.

4.2. Deeper layers encode more diverse functions

The number of distinct SLTs that are applied to data points from the same class shows the diversity of templates learned for that class. This diversity of templates can also be interpreted as the complexity of the function that is learned for a given class. With this intuition, to quantitatively analyze a network we perform SVD on the SLT of node n in layer ℓ and keep the singular value spectrum contained in the diagonal of the matrix $\mathbf{\Sigma}^{\ell}_{n}$ (normalized by the maximum value).

Because we are interested in the transformations occurring in deep networks at each layer, we applied this analysis on each of the individual nodes of the output and hidden layers of the network. For each node, if we retain the vector of singular values and take their average, we see that given the same dataset, deeper layers also encode more complex functions (7A, top left). In this analysis we neglect the first

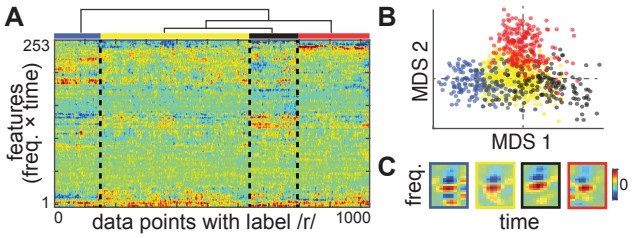


Figure 6. (A) SLTs of correctly classified data points with the label /r/, clustered by similarity. (B) Pairwise distances between individual SLTs in (A) projected into 2D using MDS. Color indicates cluster assignment. (C) Average SLT for clusters defined in (A).

hidden layer because the SLT of each node is bimodal (contains only values of W_i^1 or 0 for a data point, meaning that $\text{rank}(\mathbf{SLT}_n^1) = 1$). In deeper layers of the network, the AUC score from Eq. 5 is larger, indicating that more orthogonal bases are needed to explain a fixed percentage of the variance. For a comparison of deep vs. shallow networks with the same number of parameters, see (S. Wang & Aslan, 2016).

While this analysis shows that on average individual nodes encode more diverse SLTs in deep layers, we also wanted to confirm that this resulted in a more complex population coding of classes. To do this, we performed a similar analysis, this time doing one SVD for each class on the matrix containing the SLT of all nodes in a layer to instances of that class (reshaped to size $[N_k \times (F \times D)]$). In this way, we perform SVD on all the nodes in a layer in response to each classes. Fig. 7B shows the resultant score calculated from SVD on the SLT, averaged over classes, plotted against the scores from Fig. 3C calculated similarly from the activations. We observe a strong negative correlation, indicating that the increased normalization that occurs within classes in the activations of deeper layers is due to an increase in the diversity of SLTs that create more linear response regions in the network.

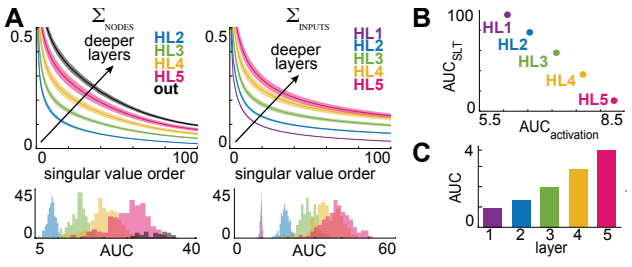


Figure 7. (A) SVD spectra of SLTs for hidden layers computed over individual nodes ($\mathbf{\Sigma}_{\text{NODES}}$) and input dimensions ($\mathbf{\Sigma}_{\text{INPUTS}}$), averaged. Below are histograms of AUC scores over nodes/input dimensions for the average spectra. (B) AUC score from SVD spectra calculated from activations vs. SLT. (C) Area under the curve (AUC) for $\mathbf{\Sigma}^\ell_{n,\text{INC}} - \mathbf{\Sigma}^\ell_{n,\text{COR}}$ calculated from SLT of nodes.

Finally, we examined the SLTs for more difficult examples in the dataset. Recall from section 3.3 that the representational space was transformed more for linearly inseparable data points than for separable ones. We replicated this result in the SLT space by performing an SVD for individual nodes by splitting the data points into the analogous linearly separable and inseparable groups $\mathbf{SLT}^\ell_{\text{COR}}$ and $\mathbf{SLT}^\ell_{\text{INC}}$ based on classification accuracy from the LDA model trained on input features for each hidden layer ℓ. We then calculated the score for the difference between the spectra $\mathbf{\Sigma}^\ell_{n,\text{INC}} - \mathbf{\Sigma}^\ell_{n,\text{COR}}$, averaged over nodes n, using area under the curve (AUC). The resulting positive values shown in Fig. 7D show that the diversity of SLTs learned for the difficult examples is consistently greater that the easier ones. Therefore, the nonuniform warping of the feature space is achieved by an increase in the number of SLTs that are learned, resulting in division of the feature space into a greater number of linear sub-regions to accommodate these more subtle class distinctions.

4.3. Visualizing the SLT from input to output

We showed in the previous section that the functions learned in subsequent layers of deep neural networks are more complex. Here, we seek to visualize the learned functions that define the separate classes. Because we are interested in the features important for class distinctions, for the remainder of this section we consider the subset of data points in the eval93 set that are classified correctly at the output layer of the network. For each output node, we calculate the SLT which is a matrix of dimension $[N_k \times F]$, where N_k is the number of data points belonging to class k. To visualize the learned linear templates at the output layer, we find the centroid over data points N_k and compute the mean of the SLT and corresponding input features over the 50 samples closest to the centroid. Results for selected classes are shown in Fig. 8A, where the average input features (left, outlined in red) and corresponding mean SLT (right, outlined in blue) are presented side-by-side.

A standard MLP trained for supervised learning must normalize sources of variability in the feature space in order to perform a successful classification. In a phoneme recognition task, there are two main sources of variability within a class. The first source comes from differences in pronunciations of phones, or allophonic variations of the same phoneme (Ladefoged & Johson, 2010). The second source of variability comes from the fact that the network must learn that time-shifted variants of the same inputs belong to the same class. In the previous section, we showed that the network equivalent sample-dependent linear systems are quite diverse at the output layer, evidenced by the large AUC score of the SVD spectra of output nodes. To visualize this diversity of templates, which uncover the possible variations of features for each class, we used unsupervised

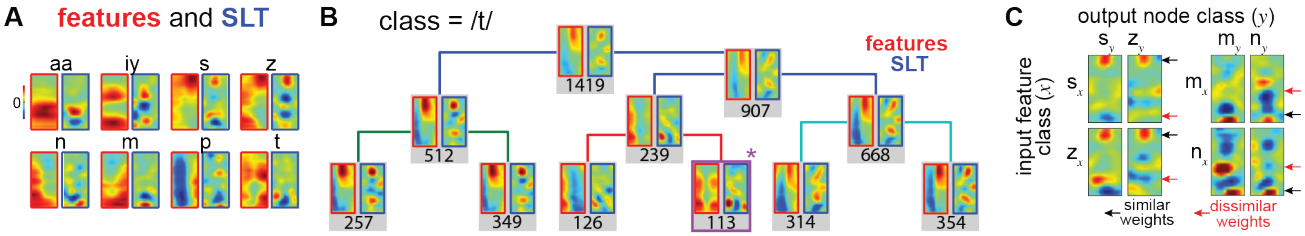


Figure 8. (A) Features (red) and corresponding SLT (blue), averaged over the WSJ eval93 set for selected classes. (B) Unsupervised hierarchical clustering for SLT (red), shown with corresponding features (blue). Number of data points in each cluster is shown below the features and SLT. (C) Comparison of average SLT for selected output nodes (columns) to data points of selected classes (rows).

hierarchical clustering (UPGMA, Euclidean distance). The clustering of SLTs for one example class (phonemes = /t/) is shown in Figure 8B. In this figure, the mean SLT of each group is outlined in blue and the corresponding average features are outlined in red; below each feature/SLT pair are the number of data points in that cluster.

This clustering analysis shows that the network learns to emphasize different feature dimensions depending on what variation of each class is presented to the network. For example, consider the cluster of /t/ highlighted and starred in purple in Fig. 8B. This cluster represents a flapped allophone of /t/ (such as in the American English pronunciation of "water"). The average SLT for this variation of /t/ shows absence of power in high frequency bands (compare to the average features at the top). This cluster has a vastly different spectral profile than the cluster of /t/ shown above, and we can see that the corresponding mean SLT for this cluster emphasizes different parts of the feature space. This analysis thus provides a data driven method to discovering the variations present in phonemes, which has been the subject of linguistic debate for decades (Stevens, 2000).

In the previous two analyses, we considered the mapping of data points of each class to the corresponding nodes in the output. However, it is equally important to form functions that provide evidence against a data point belonging to other classes. Because one can find the SLTs for each sample to all output nodes, this method makes it possible to also investigate the resolution of confusable features of similar classes. Fig. 8C examines these properties through the visualization of the output nodes corresponding to two frequently confused class pairs (/s/ and /z/; /m/ and /n/) and the average SLT of these nodes to both classes. What we observe is that for easily confused pairs, there are shared features (shown in black), but also distinguishing features between the classes (shown in red). For example, the main difference between phonemes /s/ and /z/ are the low frequencies (red arrow), which are negatively weighted for the /s/ node and positively weighted for the /z/ node. This is consisten with the input features for /s/ and /z/ 8A, where we see the voiced characteristics of /z/ give the inputs energy in low frequencies.

5. Discussion

In this study, we introduce novel techniques for jointly analyzing the internal representation and the transformations of features that are used by a multilayer perceptron. Using networks trained to map acoustic features of speech to phoneme categories, we determined the encoding properties of each phoneme by the individual and population of nodes in hidden layers of the network. We found a progressive non-uniform and nonlinear transformation of the feature space which increases the discriminant dimensions of overlapping instances of different classes.

To study how the network achieves categorization of its inputs, we proposed a method by which the exact function that is applied to every data point can be constructed in the form of a linear transformation of the input features (sample-dependent linear transform, or SLT). We found that while the network can potentially learn unique functions for every data point, instead the network applies similar functions to clusters of data points. These clusters of shared functions exemplify the prototypical variations of each class, suggesting a computational strategy to explicitly model variability. More generally, analyzing the properties of these functions provides a data-driven feature interpretation method, which can be used for feature selection, discovering dependencies between features, or analyzing the variability of each class.

Overall, our study provides a novel and intuitive account of how deep neural networks perform classification, and provides a qualitative and quantitative method for comparison of networks with various sizes and architectures. Although the results here are presented in a phoneme recognition task, these methods are easily extensible to other applications trained on alternative datasets or architectures (e.g., very deep networks, bottlenecks, convolutional networks, or any other network where the activation function is differentiable with respect to the inputs). Moreover, analyzing the SLTs allows one to investigate the encoding properties of misclassified data points, which can provide intuitions that aid in devising improved feature extraction and classification techniques.

References

A. L. Maas, A. Y. Hannun and Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning*, Atlanta, GA, 2013.

A. Nguyen, J. Yosinkski and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, Boston, MA, 2015.

A.-R. Mohamed, G. E. Dahl and Hinton, G. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14–22, 2010.

A.-R. Mohamed, G. Hinton and Penn, G. Understanding how deep belief networks perform acoustic modelling. In *ICASSP*, Kyoto, Japan, 2012.

A. Veit, M. J. Wilber and Belongie, S. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems*, Barcelona, Spain, 2016.

C. Szegedy, W. Zaremba, I. Sutskever J. Bruna D. Erhan I. Goodfellow and Fergus, R. Intriguing properties of neural networks. *arXiv:1312.6199*, 2013.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

D. Andor, C. Alberti, D. Weiss A. Severyn A. Presta K. Ganchev S. Petrov and Collins, M. Globally normalized transition-based neural networks. *arXiv:1603.06042*, 2016.

D. Povey, A. Ghoshal, G. Boulianne L. Burget O. Glembek N. Goel M. Hannemann P. Motlicek Y. Qian P. Schwarz J. Silovsky G. Stemmer and Vesely, K. The kaldi speech recognition toolkit. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Waikoloa, HI, 2011.

D. Silver, A. Huang, C. J. Maddison A. Guez L. Sifre G. van den Driessche J. Schrittwieser I. Antonoglou V. Panneershelvam M. Lanctot S. Dieleman D. Grewe J. Nham N. Kalchbrenner I. Sutskever T. Lillicrap M. Leach K. Kavukcuoglu T. Graepel and Hassabis, D. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Eldan, R. and Shamir, O. The power of depth for feedforward neural networks. *arXiv:1512.03965*, 2015.

G. E. Dahl, D. Yu, L. Deng and Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

G. E. Dahl, T. N. Sainath and Hinton, G. E. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *ICASSP*, pp. 8609–8613, Vancouver, Canada, 2013.

G. Hinton, L. Deng, D. Yu G. E. Dahl A. Mohamed N. Jaitly A. Senior V. Vanhoucke P. Nguyen T. N. Sainath and Kingsbury, B. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Processing Magazine*, pp. 82–97, 2012.

G. Montufar, R. Pascanu, K. Cho and Bengio, Y. On the number of linear regions of deep neural networks. In *Neural Information Processing Systems*, pp. 1–9, Montreal, Canada, 2014.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*, 2015.

J. Bergstra, O. Breuleux, F. F. Bastien P. Lamblin R. Pascanu G. Desjardins J. Turian D. Warde-Farley and Bengio, Y. Theano: a cpu and gpu math compiler in python. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Austin, TX, 2010.

J. Garofolo, D. Graff, D. Paul and Pallett, D. *CSR-I (WSJ0) Complete*. Linguistic Data Consortium, Philadelphia, 1993.

J. Garofolo, D. Graff, D. Paul and Pallett, D. *CSR-II (WSJ1) Complete*. Linguistic Data Consortium, Philadelphia, 1994.

J. Yosinski, J. Clune, A. Nguyen T. Fuchs and Lipson, H. Understanding neural networks through deep visualization. *arXiv:1506.06579*, 2016.

K. He, X. Zhang, S. Ren and Sun, J. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, USA, 2016.

K. Hornik, M. Stinchcombe and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Kruskal, Joseph B. and Wish, Myron. *Multidimensional Scaling*. Sage Publications, Newbury Park, 1978.

Ladefoged, P. and Johson, K. *A Course in Phonetics*. Wadsworth Publishing, Boston, MA, 2010.

Lin, H. W. and Tegmark, M. Why does deep and cheap learning work so well? *arXiv:1608.08225*, 2016.

M. D. Zeiler, M. Ranzato, R. Monga M. Mao K. Yang Q. V Le P. Nguyen A. Senior V. Vanhoucke J. Dean and Hinton, G. E. On rectified linear units for speech processing. In *ICASSP*, pp. 3517–3521, Vancouver, Canada, 2013.

Mallat, S. Understanding deep convolutional networks. *Phil. Trans. R. Soc*, 374(2065), 2016.

N. Srivastava, G. E. Hinton, A. Krizhevsky I. Sutskever and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Marchine Learning Research*, 15:1929–1958, 2014.

R. Pascanu, G. Montufar and Bengio, Y. On the number of response regions of deep feedforward networks with piecewise linear activations. *arXiv:1312.6098v5*, 2013.

S. Wang, A. Mohamed, R. Caruana J. Bilmes M. Plilipose M. Richardson K. Geras G. Urban and Aslan, O. Analysis of deep neural networks with the extended data jacobian matrix. In *International Conference on Machine Learning*, New York, NY, 2016.

Stevens, K. N. *Acoustic Phonetics*. The MIT Press, 2000.

T. Nagamine, M. L. Seltzer and Mesgarani, N. Exploring how deep neural networks form phonemic categories. In *INTERSPEECH*, pp. 1912–1916, Dresden, Germany, 2015.

T. Nagamine, M. L. Seltzer and Mesgarani, N. On the role of nonlinear transformations in deep neural network acoustic models. In *INTERSPEECH*, pp. 803–807, San Francisco, CA, 2016.

van der Maaten, L.J.P. and Hinton, G.E. Visualizing high-dimensional data using t-sne. *Journal of Marchine Learning Research*, 9:2579–2605, 2008.

W. Xiong, J. Droppo, X. Huang F. Seide M. Seltzer A. Stolcke D. Yu and Zweig, G. The microsoft 2016 conversational speech recognition system. *arXiv:1609.03528*, 2016.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. *arXiv:1311.2901*, 2013.