# UNSUPERVISED LEARNING OF ACOUSTIC FEATURES VIA DEEP CANONICAL CORRELATION ANALYSIS

*Weiran Wang*[1] *Raman Arora*[2] *Karen Livescu*[1] *Jeff A. Bilmes*[3]

[1]TTI-Chicago [2]Johns Hopkins University [3]University of Washington

**ABSTRACT**

It has been previously shown that, when both acoustic and articulatory training data are available, it is possible to improve phonetic recognition accuracy by learning acoustic features from this multi-view data with canonical correlation analysis (CCA). In contrast with previous work based on linear or kernel CCA, we use the recently proposed deep CCA, where the functional form of the feature mapping is a deep neural network. We apply the approach on a speaker-independent phonetic recognition task using data from the University of Wisconsin X-ray Microbeam Database. Using a tandem-style recognizer on this task, deep CCA features improve over earlier multi-view approaches as well as over articulatory inversion and typical neural network-based tandem features. We also present a new stochastic training approach for deep CCA, which produces both faster training and better-performing features.

***Index Terms*—** multi-view learning, neural networks, deep canonical correlation analysis, XRMB, articulatory measurements

## 1. INTRODUCTION

Modern speech recognizers often use deep neural networks (DNNs) trained to predict the posterior probabilities of phonetic states [1]. In the two most common approaches, either (1) the DNN outputs are scaled by the state priors and used as an observation model in a hidden Markov model (HMM)-based recognizer (the hybrid approach [2]) or (2) the outputs of some layer of the network (possibly a narrow "bottleneck" layer or the final layer) are post-processed and used as acoustic features in an HMM system with a Gaussian mixture model (GMM) observation distribution (the tandem approach [3]). Working within the tandem approach, we investigate whether we can learn better DNN-based acoustic features via *unsupervised* learning using an external set of unlabeled *multi-view* data, in our case simultaneously recorded audio and articulatory measurements.

The idea of feature learning using multi-view data has been explored previously using canonical correlation analysis (CCA) [4] and its nonlinear extension kernel CCA (KCCA) [5, 6]. Here we propose to use the recently devloped deep CCA (DCCA) approach, which differs from linear/kernel CCA in that the feature mapping is implemented with a DNN rather than a linear/kernel function. Considering the earlier successes of CCA/KCCA, and the general success of DNNs for speech tasks, it is a natural question whether multi-view feature learning can benefit from the more flexible functional form of a DNN. We investigate this question, using data from the University of Wisconsin X-ray Microbeam Database (XRMB) [7], on speaker-independent phonetic recognition in a setting where no articulatory data is available for the recognizer training speakers. We find that DCCA indeed improves over previous CCA-based features, as well as over DNN-based articulatory inversion and over standard DNN-based features trained on the recognizer's training data. We extend previous DCCA work by proposing and testing a new faster stochastic training method and by comparing various network architectures.

In the following sections, we give a unifying review of several CCA variants (Section 2) and present deep CCA along with its training procedure (Section 3); describe related work (Section 4); present experimental results comparing DCCA to various alternatives (Section 5); and discuss ongoing and future work (Section 6).

This research was supported by NSF grant IIS-1321015. The opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funding agency.

## 2. A UNIFYING OVERVIEW OF CCA VARIANTS

We first review canonical correlation analysis (CCA), unifying the formulation of linear and nonlinear (kernel and deep) CCA, so as to clarify their relationships and put deep CCA in context. In a multi-view learning scenario, we have access to different types of measurements of the same underlying signal, such as audio+articulation, audio+video, images+text, or text in two languages [8, 9, 10, 11]. In our setting, the training data consist of pairs of observations $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^{D_x}$ and $\mathbf{y}_i \in \mathbb{R}^{D_y}$ represent input features computed over one frame of simultaneously recorded acoustics and articulation. We also denote $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$, $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$.

Suppose we have some (possibly nonlinear) feature mappings $\mathbf{f} : \mathbb{R}^{D_x} \to \mathbb{R}^{d_x}$ and $\mathbf{g} : \mathbb{R}^{D_y} \to \mathbb{R}^{d_y}$ for view 1 and view 2 respectively. The dimensionalities $d_x$ and $d_y$ are arbitrary and could be infinite, e.g. if we use feature mappings induced by kernels in a Reproducing Kernel Hilbert Space (RKHS). One popular way of learning a compact representation from multi-view data is via CCA [12]. The objective of CCA is to find $L \le \min(d_x, d_y)$ pairs of linear projection vectors $\mathbf{U} \in \mathbb{R}^{d_x \times L}$ and $\mathbf{V} \in \mathbb{R}^{d_y \times L}$ such that the projections of each view are maximally correlated with their counterparts in the other view, constrained such that the dimensions in the representation are uncorrelated with each other. There are a number of equivalent ways of writing the objective, one of which is

$$\max_{\mathbf{U},\mathbf{V}} \quad \frac{1}{N} \operatorname{tr}\left(\mathbf{U}^\top \mathbf{F}\mathbf{G}^\top \mathbf{V}\right) \tag{1}$$
$$\text{s.t.} \quad \mathbf{U}^\top \left(\frac{\mathbf{F}\mathbf{F}^\top}{N} + r_x I\right)\mathbf{U} = \mathbf{V}^\top \left(\frac{\mathbf{G}\mathbf{G}^\top}{N} + r_y I\right)\mathbf{V} = I,$$

where $\mathbf{F} = \mathbf{f}(\mathbf{X}) = [\mathbf{f}(\mathbf{x}_1), \dots, \mathbf{f}(\mathbf{x}_N)] \in \mathbb{R}^{d_x \times N}$, $\mathbf{G} = \mathbf{g}(\mathbf{Y}) = [\mathbf{g}(\mathbf{y}_1), \dots, \mathbf{g}(\mathbf{y}_N)] \in \mathbb{R}^{d_y \times N}$, and $(r_x, r_y) \ge 0$ are regularization parameters (we assume that $\mathbf{F}$ and $\mathbf{G}$ are centered at the origin for notational simplicity; if they are not, we can center them as a preprocessing operation). If we use the original input data without further feature extraction, i.e. $\mathbf{F} = \mathbf{X}$ and $\mathbf{G} = \mathbf{Y}$, then we recover the linear CCA objective. Let $\mathbf{\Sigma}_{12} = \frac{1}{N}\mathbf{F}\mathbf{G}^\top$, $\mathbf{\Sigma}_{11} = \frac{1}{N}\mathbf{F}\mathbf{F}^\top + r_x I$ and $\mathbf{\Sigma}_{22} = \frac{1}{N}\mathbf{G}\mathbf{G}^\top + r_y I$ be the cross- and (regularized) auto-covariance matrices of the feature-mapped data in the two views. It can be shown that the optimal value of (1) is the sum of the top $L$ singular values of the matrix $\mathbf{T} = \mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2}$. Letting $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ be the matrices of the first $L$ left and right singular vectors of $\mathbf{T}$,

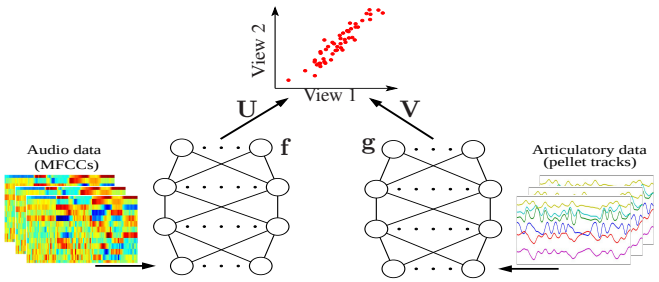


**Fig. 1**. Schematic diagram of deep canonical correlation analysis.

the optimum of (1) is achieved by $(\mathbf{U}, \mathbf{V}) = (\mathbf{\Sigma}_{11}^{-1/2}\tilde{\mathbf{U}}, \mathbf{\Sigma}_{22}^{-1/2}\tilde{\mathbf{V}})$. As a result, when the feature mappings $\mathbf{f}$ and $\mathbf{g}$ are fixed, the projection matrices $\mathbf{U}$ and $\mathbf{V}$ can be computed in closed form via singular value decomposition (SVD) of the $d_x \times d_y$ matrix $\mathbf{T}$. The final CCA features (projections) are $\tilde{\mathbf{f}}(\mathbf{x}) = \mathbf{U}^\top \mathbf{f}(\mathbf{x})$ for view 1 and $\tilde{\mathbf{g}}(\mathbf{y}) = \mathbf{V}^\top \mathbf{g}(\mathbf{y})$ for view 2.

One can show that (1) is equivalent to the following (by switching $\max(\cdot)$ with $\min -(\cdot)$, and adding $1/2$ times the constraints):

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2(N)} \left\| \mathbf{U}^\top \mathbf{F} - \mathbf{V}^\top \mathbf{G} \right\|_F^2 + \frac{r_x}{2} \|\mathbf{U}\|_F^2 + \frac{r_y}{2} \|\mathbf{V}\|_F^2 \quad (2)$$

$$\text{s.t.} \quad \text{the same constraints in (1).}$$

In other words, CCA minimizes the squared difference between the projections of the two views, subject to whitening constraints. Under certain assumptions on the input distributions, CCA maximizes mutual information [13] and has a probabilistic interpretation [14].

The above formulation encompasses several variants. First, we have already mentioned that the formulation is exactly the original (linear) CCA when $\mathbf{f}$ and $\mathbf{g}$ are identity mappings. In order to learn richer features, one may use nonlinear mappings. One nonlinear approach is kernel CCA (KCCA), corresponding to choosing $\mathbf{f}(\mathbf{x}) = k_x(\mathbf{x}, \cdot)$ and $\mathbf{g}(\mathbf{y}) = k_y(\mathbf{y}, \cdot)$ where $k_x$ and $k_y$ are positive-definite kernel functions (e.g., Gaussian RBF kernel $k(\mathbf{a}, \mathbf{b}) = e^{-\|\mathbf{a}-\mathbf{b}\|^2/2s^2}$ where $s$ is the kernel width) [15, 16]. From the representer theorem of reproducing kernel Hilbert spaces (RKHS), we know that the solution of (1) has the form $\mathbf{U} = \sum_{i=1}^N \boldsymbol{\alpha}_i k_x(\mathbf{x}_i, \cdot)$ and $\mathbf{V} = \sum_{i=1}^N \boldsymbol{\beta}_i k_y(\mathbf{y}_i, \cdot)$ where $\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i \in \mathbb{R}^L$, $i = 1, \ldots, N$, and the final CCA mapping can be written as $\tilde{\mathbf{f}}(\mathbf{x}) = \sum_{i=1}^N \boldsymbol{\alpha}_i k_x(\mathbf{x}, \mathbf{x}_i)$ and $\tilde{\mathbf{g}}(\mathbf{y}) = \sum_{i=1}^N \boldsymbol{\beta}_i k_y(\mathbf{y}, \mathbf{y}_i)$. Therefore one can conveniently work with Gram matrices instead of infinite dimensional RKHS space and optimize over the coefficients $\{\boldsymbol{\alpha}_i\}_{i=1}^N$ and $\{\boldsymbol{\beta}_i\}_{i=1}^N$ by solving an eigenvalue problem of size $N \times N$. Note that $(r_x, r_y) > 0$ are needed to avoid trivial solutions. Kernel CCA is computationally challenging for large data sets due to the $N \times N$ eigenvalue problem, for which approximate or iterative solutions are often needed [5, 17].

## 3. DEEP CCA

If we implement the feature mappings $\mathbf{f}$ and $\mathbf{g}$ of Section 2 with neural networks, this results in deep CCA (DCCA) [18], illustrated in Figure 1. A $K$-layer neural network implements the nested mapping $\mathbf{f}(\mathbf{x}) = \mathbf{f}_K(\cdots \mathbf{f}_1(\mathbf{x}; \mathbf{W}_1) \cdots); \mathbf{W}_K)$, where $\mathbf{W}_j$ are the weight parameters (biases at each layer can be absorbed in $\mathbf{W}_j$ by appending an extra 1 to its input) of layer $j$, $j = 1, \ldots, K$, and $\mathbf{f}_j$ is the mapping of layer $j$ which takes the form of a linear mapping followed by a (typically nonlinear) element-wise activation: $\mathbf{f}_j(\mathbf{t}) = \sigma(\mathbf{W}_j^\top \mathbf{t})$. In DCCA, we learn weights $\mathbf{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_K\}$ that optimize the canonical correlations at the output layers:

$$\min_{\mathbf{W_f}, \mathbf{W_g}, \mathbf{U}, \mathbf{V}} -\frac{1}{N} \operatorname{tr}\left( \mathbf{U}^\top \mathbf{F}(\mathbf{X}; \mathbf{W_f}) \mathbf{G}(\mathbf{Y}; \mathbf{W_g})^\top \mathbf{V} \right) \quad (3)$$

$$\text{s.t.} \quad \text{the same constraints in (1),}$$

where we have made explicit the dependence of $\mathbf{F}$ and $\mathbf{G}$ on their inputs and weight parameters $\mathbf{W_f}, \mathbf{W_g}$. Another $\ell_2$ regularization term $\lambda(\|\mathbf{W_f}\|^2 + \|\mathbf{W_g}\|^2)$ may be added to the objective. The projections $(\mathbf{U}, \mathbf{V})$ can be regarded as adding an extra linear layer on top of $(\mathbf{f}, \mathbf{g})$ respectively. The final DCCA features (projections) are $\tilde{\mathbf{f}}(\mathbf{x}) = \mathbf{U}^\top \mathbf{f}(\mathbf{x})$ for view 1 and $\tilde{\mathbf{g}}(\mathbf{y}) = \mathbf{V}^\top \mathbf{g}(\mathbf{y})$ for view 2.

Although both KCCA and DCCA provide nonlinear feature transformations, they differ in their functional forms. KCCA is nonparametric and linearly combines "similarities" between the test sample and each training sample (through the kernel). In contrast, DCCA is parametric and transforms a test sample through layers of linear mapping and nonlinear activations. The parametric form of DCCA makes it typically faster to train and test for data sets of reasonable sizes for speech tasks.

### 3.1. Stochastic optimization of deep CCA

The DCCA objective (3) differs from typical DNN regression or classification training objectives. Typically, the objectives are unconstrained and can be written as the expectation (or sum) of error functions (e.g., squared loss or cross entropy) incurred at each training example. This property naturally suggests stochastic gradient descent (SGD) for optimization, where one iteratively estimates the gradient based on one or a few training examples (a minibatch) and takes a small step in the opposite direction. However, in (3) there are two networks, and the objective can not be written as an unconstrained sum of errors. The difficulty lies in the fact that the training examples are coupled through the covariance matrices, which can not be reliably estimated with only a small amount of data. When introducing deep CCA, Andrew *et al.* [18] used a full batch algorithm (L-BFGS) for optimization. This is undesirable for applications with large training sets, as each gradient step computed on the entire training set can be very expensive in both memory and time. In contrast, here we use a more efficient SGD and show that it works well even for this type of objective if larger minibatches are used, presumably because a large minibatch contains enough information to estimate the covariances and therefore the gradient accurately enough. We first give a brief derivation of the gradient.

Note that $\mathbf{U}$ and $\mathbf{V}$ have a closed-form solution for fixed $\mathbf{f}$ and $\mathbf{g}$ as discussed in Section 2. Substituting that solution into our objective, we obtain $\operatorname{tr}\left(\mathbf{U}^\top \mathbf{F} \mathbf{G}^\top \mathbf{V}\right) = \sum_{j=1}^L \sigma_j(\mathbf{T})$, where $\sigma_j(\mathbf{T})$, $j = 1, \ldots, L$ are the $L$ largest singular values of $\mathbf{T} = \mathbf{\Sigma}_{11}^{-1/2} \mathbf{\Sigma}_{12} \mathbf{\Sigma}_{22}^{-1/2}$. Let $\mathbf{T} = \tilde{\mathbf{U}} \mathbf{D} \tilde{\mathbf{V}}^\top$ be the rank-$L$ SVD of $\mathbf{T}$. Then the gradient of the total correlation with respect to the feature matrix is [18]

$$\frac{\partial \sum_{j=1}^L \sigma_j(\mathbf{T})}{\partial \mathbf{F}} = \frac{1}{N}(2\Delta_{11}\mathbf{F} + \Delta_{12}\mathbf{G}),$$

with $\Delta_{11} = -\frac{1}{2}\mathbf{\Sigma}_{11}^{-1/2} \tilde{\mathbf{U}} \mathbf{D} \tilde{\mathbf{U}}^\top \mathbf{\Sigma}_{11}^{-1/2}$, $\Delta_{12} = \mathbf{\Sigma}_{11}^{-1/2} \tilde{\mathbf{U}} \tilde{\mathbf{V}}^\top \mathbf{\Sigma}_{22}^{-1/2}$, and $\partial \sum_{j=1}^L \sigma_j(\mathbf{T}) / \partial \mathbf{G}$ has an analogous expression. We then compute the gradient with respect to $\mathbf{W_f}$ and $\mathbf{W_g}$ through standard backpropagation. Given the gradient $\nabla_{\mathbf{W}}$ of our objective (3) with respect to all weight parameters $\mathbf{W} = [\mathbf{W_f}; \mathbf{W_g}]$ evaluated on minibatches, our weight update takes the following form:

$$\Delta \mathbf{W}^t = \mu^t \Delta \mathbf{W}^{t-1} - \epsilon^t \nabla_{\mathbf{W}} \quad \text{and} \quad \mathbf{W}^t = \mathbf{W}^{t-1} + \Delta \mathbf{W}^t$$

where $\mu^t \in [0, 1)$ and $\epsilon^t$ are the momentum parameter and learning rate at step $t$ (although we use a fixed learning rate and momentum in our experiments). We run SGD until the total correlation stops improving on a held-out validation set.

## 4. RELATED WORK

If we use the identity mapping for $\tilde{\mathbf{g}}$ without feature extraction for view 2, and remove the whitening constraints in (2), our objective

becomes $\min_{\tilde{\mathbf{f}}} ||\tilde{\mathbf{f}}(\mathbf{X}) - \mathbf{Y}||^2$, which learns a network $\tilde{\mathbf{f}}$ to predict articulatory measurements $\mathbf{Y}$ from acoustics $\mathbf{X}$ with a least-squares loss. This corresponds to articulatory inversion with a DNN, which has been used for speech recognition using different types of articulatory features [19, 20, 21]. A natural baseline against which to compare our CCA-based approaches is to use the outputs of such an articulatory inversion network as additional features for recognition.

CCA and KCCA have been successfully used in various non-speech domains [13, 9] as well as for speech recognition [4, 6]. While KCCA features work better than linear CCA for speech recognition, the bottleneck is a solver that scales to reasonably large data sets. Arora and Livescu use an incremental low-rank SVD approximation of the kernel matrices to reduce complexity [5], which helps to scale up KCCA to larger but not very large data sets. Recently, Lopez-Paz *et al.* have proposed a potentially more scalable approach [17], which approximates kernel operations with $\ell_2$ metric operations on transformed inputs using random Fourier features [22]. Huang *et al.* also show that random Fourier features perform well for phonetic frame classification, matching the performance of DNNs on TIMIT [23]. A key problem with this approach, however, is the need to use a very large number of random features, leading to non-trivial optimization of large $\ell_2$ problems. Though this difficulty can be somewhat alleviated by ensemble models in the classification setting [23], it is less clear how to apply the ensemble idea to KCCA. In our experiments (presented in the following section), we indeed find that computation remains a limitation for randomized KCCA.

## 5. EXPERIMENTS

We experiment with data from the XRMB corpus [7] of simultaneously recorded speech and articulatory measurements from 47 American English speakers (22 male, 25 female). Each speaker's recordings comprise ∼20 minutes of read speech including multi-sentence recordings, individual sentences, isolated word sequences, and number sequences, as well as non-speech oral motor tasks. We exclude isolated words and motor tasks, leaving up to 53 utterances per speaker. The articulatory measurements are horizontal and vertical displacements of 8 pellets on the tongue, lips, and jaw.

Our baseline acoustic features are 13-dimensional mel-frequency cepstral coefficients (MFCCs) computed every 10ms over a 25ms window, along with their first and second derivatives, resulting in 39-dimensional frames. We downsample the articulatory data from an original rate of $\sim 145.7$Hz to 100Hz to match the frame rate of our acoustic features, with missing entries (mistracked pellets) reconstructed by a smoothed low-rank matrix completion approach [24] to obtain more training frames (which indeed improves the results). Similarly to previous related work [6], the inputs to multi-view feature learning are acoustic and articulatory features concatenated over a 7-frame window around each frame, giving 273D acoustic inputs and 112D articulatory inputs for the CCA models.

We extend previous work with speaker-independent experiments [6] to a larger number of speakers and speaker-independence in both feature learning and recognizer training.[1] We split the XRMB speakers into disjoint sets of 35/8/2/2 speakers for CCA training/recognizer training/tuning/testing. The 35 speakers for CCA training are fixed; the remaining 12 are used in a 6-fold experiment (recognizer training on 4 2-speaker folds, tuning on 1 fold, and testing on the last fold). Each split/fold is gender-balanced. Each speaker has roughly $50,000$ frames, giving 1.43M multi-view training frames excluding silence. We remove the per-speaker mean and variance of the articulatory measurements for each training speaker.

[1] We thank Louis Goldstein for providing alignments for all 47 speakers.

We compare the following acoustic feature transformations:

**Standard discriminative DNN features (DNN)**. We train a 3-layer neural network to predict the monophone label from the 273D acoustic input on the *recognizer* training set (for each fold separately), which contains 8 speakers with ground truth alignment. We use the last layer hidden activations with dimensionality reduced to $L$ by principal components analysis (PCA) as tandem features. This baseline shows what can be done without using the unlabeled external multi-view data.

**DNN-based articulatory inversion (AI)**. We train a 3-layer neural network to map from the 273D acoustic input to the 112D articulatory measurements in each frame, and then reduce the dimensionality to $L$ via PCA. We also considered deeper architectures, as well as articulatory inversion to the 16-dimensional single-frame features, but these did not improve on the reported AI model and are not detailed further here.

**Linear CCA (CCA)**. Standard CCA, $(\tilde{\mathbf{f}}, \tilde{\mathbf{g}})$ a linear transformation.

**Kernel CCA (KCCA)**. Exact KCCA is intractable for our data due to its memory needs. We instead use the random Fourier feature approach of Lopez-Paz *et al.* [17]: For each view, we randomly sample $M$ $D_x$/$D_y$-dimensional vectors from a Gaussian distribution and map the input to $\mathbb{R}^M$ by computing the dot-product with the random samples followed by an elementwise cosine. We then apply linear CCA on the transformed features, as linear operations on these features approximate those in the RKHS corresponding to the Gaussian kernel. We solve this linear CCA step exactly via SVD. The total computational complexity for training is $\mathcal{O}(M^3 + M^2N)$, and for testing it is linear in the number of model parameters (random Gaussian samples+CCA projection matrix) and of order $\mathcal{O}(MD_x + ML)$ per test sample. We tune the kernel width in each view by grid search using $M = 5,000$ random samples, and then test the selected model using $M = 30,000$ random samples (the largest $M$ for which we could store and compute an exact SVD of an $M \times M$ matrix on a workstation with 64G main memory).

**Deep CCA (DCCA)**. We investigate different neural network architectures for each view. For $\mathbf{f}$ (and similarly for $\mathbf{g}$), we use a $K_\mathbf{f}$-layer network where the first $K_\mathbf{f} - 1$ layers are of the same width and the last layer is of width $L$, the desired final dimensionality. We denote such a DCCA architecture DCCA-$[K_\mathbf{f}, K_\mathbf{g}]$. We use rectified linear units (ReLU) rather than the inverse cubic activation of [18], as these are faster and performed similarly in preliminary experiments.

All of the learned feature types are used in a tandem approach [3], i.e., they are appended to the original 39D MFCC features and used in a HMM/GMM recognizer. The recognizer is a basic 3-state left-to-right monophone HMM-based model with a TIMIT bigram language model (as pointed out previously [6], an XRMB bigram model is too biased). We tune the language model weight/penalty on one fold and fix them on others, and tune the number of diagonal Gaussian components (up to 32) for each fold separately.

We use hidden layers of $1,500$ ReLUs for the DNN, AI, and DCCA features; using a narrower hidden layer in the middle ("bottleneck") gives worse recognition performance. The networks are all trained via SGD with the minibatch size, learning rate and momentum parameter tuned by grid search. For the DNN features we use dropout [25], with the dropout probability chosen from $\{0, 0.2, 0.5\}$. For all CCA-based features, we also tune for regularization parameters $(r_x, r_y)$. The dimensionality $L$ is tuned over $\{30, 50, 70\}$. In each fold, we select the best hyperparameters based on recognition accuracy on the tuning speakers, and use the corresponding learned

model for the test speakers.

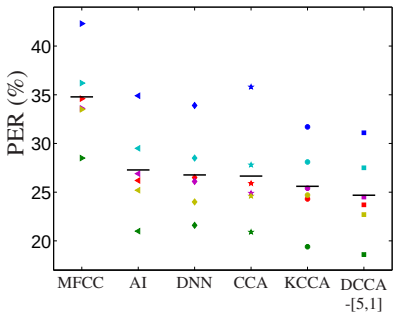


**Fig. 2**. Phone error rate using different feature transformations. Each marker denotes a feature type and each color denotes a fold. Horizontal bars give the average PER for each feature type.

The phone error rates (PER) obtained by each feature transformation on all folds are given in Figure 2. All of the CCA-based feature transformations, as well as articulatory inversion, produce large improvements over the baseline MFCCs, confirming that the articulatory measurements contain valuable information for learning better acoustic features. For DCCA, we give the final result for the architecture that performed best on tuning data, but several architectures perform similarly. Asymmetric architectures where the acoustic view uses a highly nonlinear (deep) network and the articulatory view uses a linear mapping (DCCA-[$K_f$,1]) tend to achieve better performance than using nonlinear networks for both views, and the performance improves as the acoustic view network gets deeper (i.e., larger $K_f$). Such asymmetric networks are "close" to articulatory inversion, but are still quite different in that the articulatory view is transformed and the features are optimized for correlation and not squared error.

There is a wide range of hyper-parameters that lead to similarly competitive results for DCCA. For KCCA, we find it important to use a large number of random features $M$ to get a competitive result, consistent with [23]. With $M = 5,000$, KCCA is slightly worse than linear CCA. With $M = 30,000$, KCCA has about 14.6 million parameters (random Gaussian samples + projection matrices), which is 1.6 times the number of parameters in a DCCA-[5,1] architecture (and is slower for testing as the cost is linear in the number of weight parameters for both algorithms), yet it is outperformed by deep CCA by a large margin. It is conceivable that randomized KCCA could improve over DCCA with even more parameters, but DCCA is easier and faster to train.

Our AI network gets a root mean squared error (RMSE) of 1.96mm per dimension (averaged over dimensions) for reconstructing tuning speakers' articulatory measurements, and an RMSE of 1.17mm for the 35 training speakers. It is possible to reduce training RMSE with further training or larger networks, but over-fitting happens early during training (we have trained a 6 hidden layer network which failed to improve the RMSE of test speakers or recognition accuracy). We believe this relatively weak generalization performance of AI is due to the remaining speaker variation in articulatory measurements that is not accounted for by our per-speaker mean and variance normalization. The supervised DNN features are learned on a smaller set of (labeled) frames than are the CCA-based and AI features (8 speakers vs. 35 speakers).

The differences in PER between DCCA and other feature types are significant at levels of $< 0.05$ according to paired-sample t-tests. In informal experiments varying the number of speakers in the unlabeled multi-view training data, we observe that linear CCA retains its performance with many fewer speakers, while DCCA requires more speakers; again this may be due to unaccounted for speaker variation, and deserves further analysis in future work.

### 5.1. Analysis of DCCA optimization

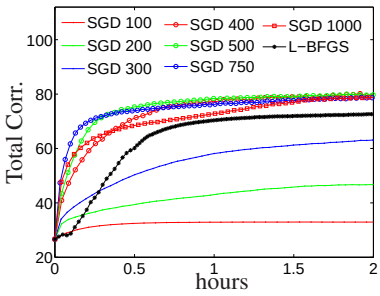


**Fig. 3**. Learning curves (total correlation vs. training time) of DCCA on the 'JW11' set of [18]. The maximum correlation is the dimensionality (112). Each marker corresponds to one epoch (one pass over the training data) for SGD, or one iteration for L-BFGS. "SGD $n$" = SGD with minibatch size $n$.

We now consider the effect of our SGD optimization and demonstrate the importance of minibatch size. We use the two-view data and train/tune/test splits of speaker JW11 used in [18] in order to compare directly with their results. The acoustic/articulatory network has two hidden layers of 1800/1200 ReLUs respectively, and the output dimensionality (and therefore the maximum possible total correlation over dimensions) is $L = 112$. We use a small weight decay $\lambda = 10^{-4}$ and do not pre-train. We do grid search for several hyper-parameters: $r_x, r_y \in \{10^{-6}, 10^{-4}, 10^{-2}, 1, 10^2\}$, constant learning rate in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, fixed momentum in $\{0, 0.5, 0.9, 0.95, 0.99\}$, and minibatch size in $\{100, 200, 300, 400, 500, 750, 1000\}$. Figure 3 shows the learning curves on the tuning set for different minibatch sizes, each using the optimal values for the other hyper-parameters. It is clear that for small minibatch sizes (100, 200), the correlation quickly plateaus at a low value, whereas for large enough minibatch size, there is always a steep increase at the beginning which is a known advantage of stochastic first-order algorithms [26], and a wide range of learning rate/momentum give very similar results.

For comparison, we also train the same model using L-BFGS (using the implementation of Mark Schmidt, which includes a good line-search procedure [27]) in full-batch mode (although it could also be used with minibatches) with the same random initial weight parameters and tune $(r_x, r_y)$ on the same grid. While L-BFGS does well on the training set, its performance on tuning/test is usually worse than that of SGD with reasonable hyper-parameters. In fact, with this shallow architecture, L-BFGS achieves a total tuning correlation of 73.7, while stochastic training achieves a tuning correlation of 80.5, higher than the best correlation obtained by [18].

## 6. CONCLUSION

We have shown that deep CCA can be optimized well by SGD, and thus that we can use it on large-scale data sets; and that using unlabeled multi-view acoustic-articulatory data external to the recognizer's labeled training set, we can improve phonetic recognition and do so better with DCCA than with previous CCA-based approaches or with DNN-based articulatory inversion. The improvement over articulatory inversion suggests that predicting the details of articulation is neither important nor useful, perhaps because inversion requires learning details that are more speaker-specific than the hidden subspace learned by CCA-based techniques. Compared to current implementations of kernel CCA, DCCA scales better to large data. Future directions include incorporating multi-view feature learning in a hybrid model; incorporating supervision in the case where we have labels for the multi-view data (i.e., extending [28] to jointly training a deep and highly correlated representation); and further analysis of stochastic training and network types for DCCA.

## 7. REFERENCES


[1] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[2] Hervé Bourlard and Nelson Morgan, *Connectionist Speech Recognition. A Hybrid Approach*, Kluwer Academic Publishers Group, Dordrecht, The Netherlands, 1994.

[3] Hynek Hermansky, Daniel P. W. Ellis, and Sangita Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. of the IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP)*, Istanbul, Turkey, 2000.

[4] Sujeeth Bharadwaj, Raman Arora, Karen Livescu, and Mark Hasegawa-Johnson, "Multi-view acoustic feature learning using articulatory measurements," in *International Workshop on Statistical Machine Learning for Speech Processing (IWSML)*, 2012.

[5] Raman Arora and Karen Livescu, "Kernel CCA for multi-view learning of acoustic features using articulatory measurements," in *Symposium on Machine Learning in Speech and Language Processing (MLSLP)*, 2012.

[6] Raman Arora and Karen Livescu, "Multi-view CCA-based acoustic features for phonetic recognition across speakers and domains," in *Proc. of the IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP)*, Vancouver, Canada, 2013.

[7] John R. Westbury, *X-Ray Microbeam Speech Production Database User's Handbook Version 1.0*, Waisman Center on Mental Retardation & Human Development, University of Wisconsin, Madison, WI, June 1994.

[8] David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004.

[9] Richard Socher and Fei-Fei Li, "Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora," in *Proc. of the 2010 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, 2010, pp. 966–973.

[10] Alexei Vinokourov, Nello Cristianini, and John S. Shawe-Taylor, "Inferring a semantic representation of text via cross-language correlation analysis," in *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[11] C. Mario Christoudias, Raquel Urtasun, and Trevor Darrell, "Multi-view learning in the presence of view disagreement," 2008.

[12] Harold Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, Dec. 1936.

[13] Magnus Borga, "Canonical correlation: A tutorial," Available at `http://www.imt.liu.se/people/magnus/cca/`, Jan. 12 2001.

[14] Francis R. Bach and Michael I. Jordan, "A probabilistic interpretation of canonical correlation analysis," Tech. Rep. 688, Dept. of Statistics, University of California, Berkeley, Apr. 21 2005.

[15] Pei Ling Lai and Colin Fyfe, "Kernel and nonlinear canonical correlation analysis," *Int. J. Neural Syst.*, vol. 10, no. 5, pp. 365–377, Oct. 2000.

[16] Shotaro Akaho, "A kernel method for canonical correlation analysis," in *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*, Osaka, Japan, 2001, Springer-Verlag.

[17] David Lopez-Paz, Suvrit Sra, Alex Smola, Zoubin Ghahramani, and Bernhard Schoelkopf, "Randomized nonlinear component analysis," in *Proc. of the 31st Int. Conf. Machine Learning (ICML)*, Eric Xing and Tony Jebara, Eds., Beijing, China, 2014, pp. 1359–1367.

[18] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu, "Deep canonical correlation analysis," in *Proc. of the 30th Int. Conf. Machine Learning (ICML)*, Sanjoy Dasgupta and David McAllester, Eds., Atlanta, GA, 2013, pp. 1247–1255.

[19] Benigno Uría, Steve Renals, and Korin Richmond, "A deep neural network for acoustic-articulatory speech inversion," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[20] Leonardo Badino, Claudia Canevari, Luciano Fadiga, and Giorgio Metta, "Deep-level acoustic-to-articulatory mapping for DBN-HMM based phone recognition," in *2012 IEEE Spoken Language Technology Workshop (SLT)*, Miami, FL, Dec. 2–5 2012, pp. 370–375.

[21] Vikramjit Mitra, Ganesh Sivaraman, Hosung Nam, Carol Espy-Wilson, and Elliot Saltzman, "Articulatory features from deep neural networks and their role in speech recognition," in *Proc. of the IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP)*, Florence, Italy, 2014, pp. 3041–3045.

[22] Ali Rahimi and Ben Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems (NIPS)*, John C. Platt, Daphne Koller, Yoram Singer, and Sam Roweis, Eds. 2008, vol. 20, pp. 1177–1184, MIT Press, Cambridge, MA.

[23] Po-Sen Huang, Haim Avron, Tara Sainath, Vikas Sindhwani, and Bhuvana Ramabhadran, "Kernel methods match deep neural networks on TIMIT: Scalable learning in high-dimensional random Fourier spaces," in *Proc. of the IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP)*, Florence, Italy, 2014.

[24] Weiran Wang, Raman Arora, and Karen Livescu, "Reconstruction of articulatory measurements with smoothed low-rank matrix completion," in *IEEE Spoken Language Technology Workshop (SLT)*, South Lake Tahoe, CA, 2014.

[25] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," July 3 2012.

[26] Leon Bottou and Olivier Bousquet, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems (NIPS)*, John C. Platt, Daphne Koller, Yoram Singer, and Sam Roweis, Eds. 2008, vol. 20, pp. 161–168, MIT Press, Cambridge, MA.

[27] Mark Schmidt, "minFunc," 2012, Code available at `http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html`.

[28] Raman Arora and Karen Livescu, "Multi-view learning with supervision for transformed bottleneck features," in *Proc. of the IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP)*, Florence, Italy, 2014.