A COMPARISON BETWEEN DEEP NEURAL NETS AND KERNEL ACOUSTIC MODELS FOR SPEECH RECOGNITION

Zhiyun Lu[1†] *Dong Guo*[2†] *Alireza Bagheri Garakani*[2†] *Kuan Liu*[2†]

Avner May[3‡] *Aurélien Bellet*[4‡] *Linxi Fan*[2]

Michael Collins[3*] *Brian Kingsbury*[5] *Michael Picheny*[5] *Fei Sha*[1]

[1]U. of California (Los Angeles) [2] U. of Southern California [3]Columbia U.
[4]Team Magnet, INRIA Lille - Nord Europe [5] IBM T. J. Watson Research Center (USA)

[† ‡]: *contributed equally as the first and second co-authors, respectively*

ABSTRACT

We study large-scale kernel methods for acoustic modeling and compare to DNNs on performance metrics related to both acoustic modeling and recognition. Measuring perplexity and frame-level classification accuracy, kernel-based acoustic models are as effective as their DNN counterparts. However, on token-error-rates DNN models can be significantly better. We have discovered that this might be attributed to DNN's unique strength in reducing both the perplexity and the entropy of the predicted posterior probabilities. Motivated by our findings, we propose a new technique, entropy regularized perplexity, for model selection. This technique can noticeably improve the recognition performance of both types of models, and reduces the gap between them. While effective on Broadcast News, this technique could be also applicable to other tasks.

Index Terms— deep neural networks, kernel methods, acoustic models, automatic speech recognition

1. INTRODUCTION

Deep neural networks (DNNs) have significantly advanced the state-of-the-art in automatic speech recognition (ASR) [1, 2, 3, 4]. In stark contrast, kernel methods, which had once been extensively studied due to their powerful modeling of highly nonlinear data [5], have not been competitive on large-scale ASR tasks. There have been very few successful applications of kernel methods to ASR, let alone any "head-on" comparison to DNNs, except for a few efforts which were limited in scope [6, 7, 8]. The most crucial challenge is that kernel methods scale poorly with the size of the training dataset, and thus are perceived as being impractical for ASR.

In this paper, we investigate empirically how kernel methods can be scaled up to tackle typical ASR tasks. We also study how they are similar to and different from DNNs. We focus on using kernel methods for frame-level acoustic modeling, but also evaluate them and contrast to DNNs on recognition performance.

We have studied datasets for 3 languages and have made several interesting discoveries. First, we show that kernel methods can tackle large-scale ASR tasks equally efficiently. To this end, we build on the random feature approximation technique, well-known in the machine learning community [9]. Our contribution is to demonstrate its practical utility in constructing large-scale classifiers for acoustic modeling. Second, we have found that kernel-based acoustic models are as good as DNN-based ones, *if their performance is measured in terms of perplexity or frame-level classification accuracy*. However, when measuring word error rate (WER) performance, we have found that kernel-based acoustic models can lag significantly behind their DNN counterparts. For instance, on Broadcast News, IBM's DNN attains a WER of 16.7% while the kernel-based model has 18.6%. There is a sharp difference despite the two having nearly identical frame-level perplexities. Third, in the process of unraveling this mystery, we have discovered a new technique for selecting the best DNN acoustic model for decoding. Specifically, the new technique does *not* stop training when the perplexity on the heldout data starts to worsen. Instead, it looks at the trade-off between the perplexity and the entropy of the *predicted posterior probabilities* and favors models of lower entropy in exchange for a small sacrifice in perplexity.

Balancing these two factors leads to a new model selection criterion which we call *entropy-regularized perplexity*. Acoustic models selected with it have better decoding results: on the Broadcast News dataset the DNN WER improved to 16.1% and the kernel model to 17.5%. We believe this criterion (and possible other variants) could be widely applicable for training DNNs for other ASR tasks.

The rest of the paper is organized as follows. We review related work in §2. Our empirical work focuses on scaling kernel methods up to large-scale problems – we describe how in §3. In §4, we report extensive experiments comparing DNNs and kernel methods, followed by conclusions and discussion in §5.

2. RELATED WORK

The computational complexity of exact kernel methods depends quadratically on the number of training examples at training time and linearly at testing time. Hence, scaling up kernel methods has been a long-standing and actively studied problem [10, 11, 12, 13, 14, 15]. Exploiting structures of the kernel matrix can scale kernel methods to 2 million to 50 million training samples [16].

In theory, kernel methods provide a feature mapping to an infinite dimensional space. But, for any practical problem the dimensionality is bounded above by the number of training samples. Approximating kernels with finite-dimensional features has been recognized as a promising way of scaling up kernel methods. The most relevant approach for our paper is the observation [9] that inner prod-

*Currently on leave at Google Inc. New York.

ucts between features derived from random projections can be used to approximate translation-invariant kernels [17, 5, 9]. Follow-up work on using those random features ("weighted random kitchen sinks" [18]) is a major inspiration for our work. There has been a growing interest in using random projections to approximate different kernels [19, 20, 21, 22].

Despite this progress, there have been only a few reported large-scale empirical studies of those techniques on challenging tasks from speech recognition [6, 7, 8]. However, the tasks were fairly small-scale (for instance, on the TIMIT dataset). By large, a thorough comparison to DNNs on ASR tasks is lacking. Our work not only fills this gap, but also reveals details of the similarities and differences between those two popular learning paradigms.

3. KERNEL-BASED ACOUSTIC MODELING

3.1. Kernels and random features approximation

Given a pair of data points $\boldsymbol{x}$ and $\boldsymbol{z}$, a positive definite kernel function $k(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ defines an inner product between the images of the two data points under a (nonlinear) mapping $\boldsymbol{\phi}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^M$,

$$k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\phi}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{z}) \tag{1}$$

where the dimensionality M of the resulting mapping $\boldsymbol{\phi}(\boldsymbol{x})$ can be infinite. Kernel methods avoid inference in $\mathbb{R}^M$. Instead, they rely on the kernel matrix over the training samples. When M is far greater than N, the number of training samples, this trick provides a nice computational advantage. However, when N is exceedingly large, this quadratic complexity in N becomes impractical.

[9] leverage a classical result in harmonic analysis and provide a fast way to approximate $k(\cdot,\cdot)$ with *finite*-dimensional features:

Theorem 1. *(Bochner's theorem, adapted from [9]) A continuous kernel $k(\boldsymbol{x}, \boldsymbol{z}) = k(\boldsymbol{x} - \boldsymbol{z})$ is positive definite if and only if $k(\boldsymbol{\delta})$ is the Fourier transform of a non-negative measure.*

More specifically, for shift-invariant kernels such as Gaussian RBF and Laplacian kernels,

$$k^{\mathsf{rbf}} = e^{-\|\boldsymbol{x}-\boldsymbol{z}\|_2^2/2\sigma^2}, \quad k^{\mathsf{lap}} = e^{-\|\boldsymbol{x}-\boldsymbol{z}\|_1/\sigma} \tag{2}$$

the theorem implies that the kernel function can be expanded with harmonic basis, namely

$$k(\boldsymbol{x} - \boldsymbol{z}) = \int_{R^d} p(\boldsymbol{\omega}) e^{j\boldsymbol{\omega}^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{z})}\, d\boldsymbol{\omega} = \mathbb{E}_{\boldsymbol{\omega}} \left[e^{j\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}} e^{-j\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{z}} \right] \tag{3}$$

where $p(\boldsymbol{\omega})$ is the density of a d-dimensional probability distribution. The expectation is computed on complex-valued functions of $\boldsymbol{x}$ and $\boldsymbol{z}$. For real-valued kernel functions, however, they can be simplified to the cosine and sine functions, see below.

For Gaussian RBF and Laplacian kernels, the corresponding densities are Gaussian and Cauchy distributions:

$$p^{\mathsf{rbf}}(\boldsymbol{\omega}) = N\left(0, \frac{1}{\sigma}\boldsymbol{I}\right), \quad p^{\mathsf{lap}}(\boldsymbol{\omega}) = \prod_d \frac{1}{\pi(1+\sigma^2\omega_d^2)} \tag{4}$$

This motivates a sampling-based approach of approximating the kernel function. Concretely, we draw $\{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_D\}$ from the distribution $p(\boldsymbol{\omega})$ and use the sample mean to approximate

$$k(\boldsymbol{x}, \boldsymbol{z}) \approx 1/D \sum_{i=1}^{D} \phi_i(\boldsymbol{x})\phi_i(\boldsymbol{z}) = \hat{\boldsymbol{\phi}}(\boldsymbol{x})^{\mathrm{T}} \hat{\boldsymbol{\phi}}(\boldsymbol{z}) \tag{5}$$

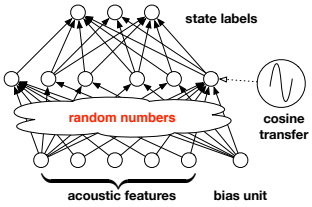

Fig. 1. Kernel-acoustic model seen as a shallow neural network

The ***random feature vector*** $\hat{\boldsymbol{\phi}}$ is thus composed of scaled cosines of random projections

$$\hat{\phi}_i(\boldsymbol{x}) = \sqrt{2/D}\, \cos(\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{x} + b_i) \tag{6}$$

where b_i is a random variable, uniformly sampled from $[0,\ 2\pi]$.

A key advantage of using approximate features over standard kernel methods is its scalability to large datasets. Learning with a representation $\hat{\boldsymbol{\phi}}(\cdot) \in \mathbb{R}^D$ is relatively efficient provided that D is far less than the number of training samples. For example, in our experiments (cf. section 4), we have 7 million to 16 million training samples, while $D \approx 25,000$ often leads to good performance.

3.2. Use random features for acoustic modeling

For acoustic modeling, we can plug the random feature vector $\hat{\boldsymbol{\phi}}(\boldsymbol{x})$ (converted from frame-level acoustic features) into a multinomial logistic regression model. Specifically, our model is a special instance of the *weighted sum of random kitchen sinks* [18]

$$p(y = c|\boldsymbol{x}) = \frac{e^{\boldsymbol{\theta}_c^{\mathrm{T}}\hat{\boldsymbol{\phi}}(\boldsymbol{x})}}{\sum_c e^{\boldsymbol{\theta}_c^{\mathrm{T}}\hat{\boldsymbol{\phi}}(\boldsymbol{x})}} \tag{7}$$

where the label y can take any value from $\{1, 2, \ldots, C\}$, each corresponding to a phonetic state label. $\boldsymbol{\theta}_c$ are learnable parameters.

3.3. View kernel-acoustic model as a shallow neural network

The model eq. (7) can be seen as a shallow neural network, shown in Fig. 1, with the following properties: (1) the parameters from the inputs (ie, acoustic feature vectors) to the hidden units are randomly chosen and not adapted; (2) the hidden units have $\cos(\cdot)$ as transfer functions; (3) the parameters from the hidden units to the output units are adapted (and can be optimized with convex optimization).

3.4. Extensions

The kernel acoustic model can also be extended to use the combination of multiple kernels – graphically, they correspond to juxtaposing several shallow neural networks together [23].

The number of phonetic state labels can be very large. This will significantly increase the number of parameters in $\{\boldsymbol{\theta}_c\}$. We can reduce it with a bottleneck layer (of 250 or 500 units) between the hidden units and the output layer. We experimented with two settings: a sigmoid bottleneck layer which corresponds to learning error-output-correct-code (ECOC) and a linear bottleneck layer which corresponds to low-rank factorization of the $\{\boldsymbol{\theta}_c\}$ [24].

4. EXPERIMENTAL RESULTS

We conduct extensive empirical studies comparing kernel methods to deep neural networks (DNNs) on typical ASR tasks.

4.1. Tasks, datasets and evaluation metrics

We train both DNNs and kernel-based multinomial logistic regression models, as described in § 3, to predict context-dependent HMM state labels from acoustic feature vectors. The acoustic features are 360-dimensional real-valued dense vectors, and are a standard speaker-adapted representation used by IBM [25]. The state labels are obtained via forced alignment using a GMM/HMM system.

We tested these models on three datasets. The first two are the IARPA Babel Program Cantonese (IARPA-babel101-v0.4c) and Bengali (IARPA-babel103b-v0.4b) limited language packs. Each pack contains a 20-hour training and a 20-hour test set. We designate about 10% of the training data as a held-out set to be used for model selection and tuning. The training, held-out, and test sets contain different speakers. Babel data is challenging because it is two-person conversations between people who know each other well (family and friends) recorded over telephone channels (in most cases with mobile telephones) from speakers in a wide variety of acoustic environments, including moving vehicles and public places. As a result, it contains many natural phenomena such as mispronunciations, disfluencies, laughter, rapid speech, background noise, and channel variability. An additional challenge in Babel is that the only data available for training language models is the acoustic transcripts, which are comparatively small. The third dataset is a 50-hour subset of Broadcast News (BN-50) [26, 27]. 45 hours of audio are used for training, 5 hours are a held-out set, and the test set is 2 hours. This is well-studied benchmark task in the ASR community due to both its convenience and relevance to developing core ASR technology.

We use three metrics to evaluate the acoustic models:

Perplexity Given examples, $\{(\boldsymbol{x}_i, y_i), i = 1 \dots m\}$, the perplexity is defined as $\mathsf{ppx} = \exp\left\{-\frac{1}{m}\sum_{i=1}^{m} \log p(y_i|\boldsymbol{x}_i)\right\}$. Perplexity is usually correlated with the next two performance measures, so we use perplexity on the held-out for model selection and tuning.

Accuracy The classification accuracy is defined as

$$\mathsf{acc} = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\left[y_i = \arg\max_{y\in 1,2,\dots,C} p(y|\boldsymbol{x}_i)\right]$$

Token Error Rate (TER) We feed the predictions of acoustic models, which are real-valued probabilities, to the rest of the ASR pipeline and calculate the misalignment between the decoder's outputs and the ground-truth transcriptions. For Bengali and BN-50, the error is the word-error-rate (WER), while for Cantonese it is character error rate (CER).

4.2. Details of Acoustic Models

For all kernel-based models, we use either Gaussian or Laplacian kernels and we also studied combinations of kernels. For more details, please see [23]. Kernel models have 3 hyperparameters: the bandwidths for Gaussian or Laplacian kernels, the number of random projections, and the step size of the (convex) optimization procedure (as adjusting it has a similar effect to early-stopping). As a rule of thumb, the kernel bandwidth ranges from 0.3–5 times the median of the pairwise distances in the data (with 1 times the median working well). We typically use 2,000 to 400,000 random features, though stable performance is often observed at 25,000 or above.

For all DNNs, we tune hyperparameters related to both the architecture and the optimization. This includes the number of layers, the number of hidden units in each layer, the learning rate, the rate decay, the momentum, regularization, etc. We differentiate two types of DNNs: ibm where the DNNs are first layer-wise discriminatively trained [28, 25] and rbm where the DNNs are first trained unsupervisedly and then discriminatively trained [29]. ibm is part of IBM's Attila package. For rbm, we also tune hyperparameters for the unsupervised learning phase.

ibm acoustic model contains five hidden-layers, each of which contains 1024 units with logistic nonlinearities. The best rbm has 4 hidden years, with 2000 hidden units per layer. The outputs of either types of models have either 1000 or 5000 softmax units, corresponding to the quinphone context-dependent HMM states clustered using decision trees. All layers in the DNN are fully connected. For discriminative training, stochastic gradient descent with a mini-batch size of 250 samples, with tuned momentum, the learning rate annealing and early stopping on the held-outs.

Table 1. Comparison in perplexity (ppx) and accuracy (acc: %)

	Bengali		Cantonese		BN-50	
Model	ppx	acc	ppx	acc	ppx	acc
DNN-ibm	3.4	71.5	6.8	56.8	7.4	50.8
DNN-rbm	3.3	72.1	6.2	58.3	6.7	52.7
kernel	3.5	71.0	6.5	57.3	7.3	51.2

Table 2. Best token error rates on test set (%)

Model	Bengali	Cantonese	BN-50
DNN-ibm	70.4	67.3	16.7
DNN-rbm	69.5	66.3	16.6
kernel	70.0	65.7	18.6

4.3. Main Results

Table 1 contrasts the best perplexity and accuracy attained by various acoustic models on held-outs. Note that cross-entropy errors (ie, the logarithm of the perplexity) are the training criteria of those models. Thus, ppx correlates with classification accuracies well. Moreover, the performances by those 3 models are close to each other. Kernel models have somewhat better performance than IBM's DNN.

Table 2 reports the best TERs. On Bengali, rbm performs marginally better than the kernel model, while kernel performs noticeably better than the two DNN models on Cantonese. However, the most surprising result is that kernel performs significantly worse on BN-50 than either rbm or ibm. Note that in Table 1, the kernel model has similar perplexity and accuracy as rbm and better ones than ibm. In what follows, we analyze the cause for this mismatch.

4.4. Tradeoff between perplexity and entropy

One possible explanation is that the perplexity might be an inadequate proxy for TER. As the predictions are probabilities to be combined with language models, we can capture the characteristics of the predictions using the entropy as it considers posterior probabilities assigned to all labels while the perplexity (as a training criteria) focuses only on the posterior probability assigned to the correct state label. The entropy measures the degree of confusions in the predictions, which could have interplayed with the language models.

Fig. 2 plots the progression of several DNN models in perplexity and entropy (each cyan colored line corresponds to a model's training course). We also plot with colored markers the WERs evaluated at the end of every four epochs. Clearly, in the beginning of the training, both the entropy and the perplexity decrease, which also

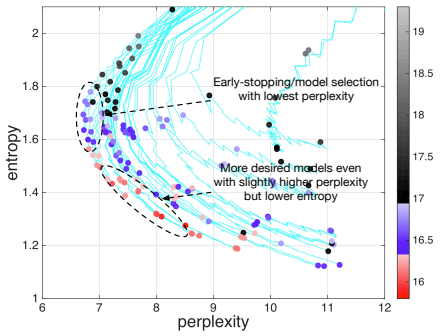

Fig. 2. Training DNN acoustic models to minimize perplexity is not sufficient to arrive at the best WER – after the typical early-stopping point where the perplexity is lowest, continuing to train to increase the perplexity but decrease the entropy leads to the best WER.

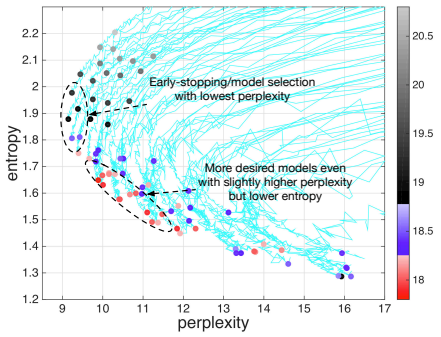

Fig. 3. Similar to training DNN acoustic models, as in Fig. 2, training kernel models also has tradeoff between perplexity and entropy where the lowest perplexity does not correspond to the lowest WER.

corresponds to an improving WER. Note that, using perplexity for early-stopping — a common practice in training multinomial logistic regression model — will result in models that are designated by the blue colored points on the leftmost of the plot. However, those models have sub-optimal WERs as continuing the training to have an increased perplexity but in exchange for a decreased entropy results in models with better WERs (the red colored points).

We observe a similar tradeoff in training kernel-based acoustic models, as shown in Fig. 3. Similarly, WER depends jointly on the perplexity and the entropy and the best perplexity or entropy does not result in the best WER. Note that when decoding, we tune the scaling of acoustic scores. Thus, balancing perplexity and entropy cannot be trivially achieved by scaling the inputs to the softmax.

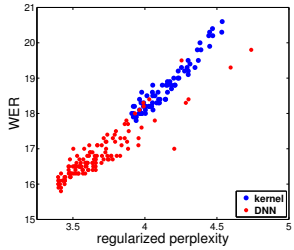

Fig. 4. WER is almost linear in the regularized perplexity

Table 3. Regularized perplexity is a better model selection criteria

Model	perplexity	regularized perplexity	Oracle
rbm	16.6	16.1	15.8
kernel	18.6	17.5	17.5

4.5. A new model selection criterion

Fig. 2 and 3 suggest that we should select the best acoustic model using both perplexity and entropy. Fig. 4 shows that it is possible to predict WER from *entropy-regularized perplexity*, defined as $\log(\text{perplexity}) + \text{entropy}$, namely,

$$-\frac{1}{m}\sum_i \sum_{k=1}^{K} [\mathbb{I}(k = y_i) + P(y = k|x_i)] \log P(y = k|x_i) \quad (8)$$

which has an almost linear relationship with the WER.

Table 3 illustrates the advantage of using this regularized perplexity on heldout to select models – for both kernel and DNN acoustic models, their WERs are improved, and the improvement on kernel models is substantial (1% WER reduction in absolute).

While this new technique reduces the gap between kernel and DNN models, kernel method still lags behind. Continuing to pry is left to future work.

5. CONCLUSION

As multiway classifiers, DNNs and kernel models do not seem to have significant differences when their performances are measured in terms of perplexity and accuracy. However, when integrated into the rest ASR pipeline, on Broadcast News (and possibly other) tasks, DNNs are able to attain much lower token error rates (TERs).

Our analysis shows that when the perplexity and the entropy of the predicted posterior probabilities are balanced, models have better TERs. Moreover, DNNs can achieve lower entropy when they have similar perplexity as kernel models. Motivated by these findings, we design a "regularized perplexity" model selection/early-stopping criteria that select better acoustic models which improve WERs over previous models that were selected using un-regularized perplexity.

To the best of our knowledge, this paper is the first to pinpoint the unique niche possessed by DNNs in better integration with decoders. In future, we will try to understand why DNN has this appealing property despite being optimized with objectives that do not take into consideration language models and structural loss [26].

6. ACKNOWLEDGEMENT

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

Additional supports include USCs Center for High-Performance Computing (http://hpc.usc.edu), USC Provost Graduate Fellowship (ABG), NSF#-1065243, 1451412, 1139148, a Google Research Award, an Alfred. P. Sloan Research Fellowship and ARO# W911NF-12-1-0241 and W911NF-15-1-0484.

7. REFERENCES

[1] Yoshua Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[3] Abdel-rahman Mohamed, George Dahl, , and Geoffrey Hinton, "Acoustic Modeling Using Deep Belief Networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[4] Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature Engineering in Context-dependent Deep Neural Networks for Conversational Speech Transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, 2011, pp. 24–29.

[5] B. Schölkopf and A. Smola, *Learning with kernels*, MIT Press, 2002.

[6] Li Deng, Gökhan Tür, Xiaodong He, and Dilek Z. Hakkani-Tür, "Use of Kernel Deep Convex Networks and End-to-end Learning for Spoken Language Understanding," in *2012 IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA, December 2-5, 2012*, 2012, pp. 210–215.

[7] C.-C. Cheng and B. Kingsbury, "Arccosine Kernels: Acoustic Modeling with Infinite Neural Networks," in *Proc. ICASSP*, 2011, pp. 5200–5203.

[8] Po-Sen Huang, Haim Avron, Tara N Sainath, Vikas Sindhwani, and Bhuvana Ramabhadran, "Kernel Methods Match Deep Neural Networks on TIMIT," in *Proc. ICASSP*, 2014, vol. 1, p. 6.

[9] Ali Rahimi and Benjamin Recht, "Random Features for Large-scale Kernel Machines," in *Advances in Neural Information Processing Systems 20*, 2007, pp. 1177–1184.

[10] Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, Eds., *Large Scale Kernel Machines*, MIT Press, Cambridge, MA., 2007.

[11] Alex Smola, "Personal communication," 2014.

[12] Dennis DeCoste and Bernhard Schölkopf, "Training Invariant Support Vector Machines," *Mach. Learn.*, vol. 46, pp. 161–190, 2002.

[13] John C. Platt, "Fast Training of Support Vector Machines using Sequential Minimal Optimization," in *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[14] Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung, "Core Vector Machines: Fast SVM Training on Very Large Data Sets," *Journal of Machine Learning Research*, vol. 6, pp. 363–392, 2005.

[15] Kenneth L. Clarkson, "Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm," *ACM Trans. Algorithms*, vol. 6, no. 4, pp. 63:1–63:30, 2010.

[16] Sören Sonnenburg and Vojtech Franc, "COFFIN: A Computational Framework for Linear SVMs," in *Proc. of the 27th Intl. Conf. on Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 999–1006.

[17] Christian Berg, Jens Peter Reus Christensen, and Paul Ressel, *Harmonic Analysis on Semigroups*, Springer, 1984.

[18] Ali Rahimi and Benjamin Recht, "Weighted Sums of Random Kitchen Sinks: Replacing Minimization with Randomization in Learning," in *Advances in Neural Information Processing Systems 21*, 2008, pp. 1313–1320.

[19] Purushottam Kar and Harish Karnick, "Random Feature Maps for Dot Product Kernels," in *Proc. of the 29th Intl. Conf. on Mach. Learn. (ICML)*, 2012.

[20] Raffay Hamid, Ying Xiao, Alex Gittens, and Dennis DeCoste, "Compact Random Feature Maps," in *Proc. of the 31th Intl. Conf. on Mach. Learn. (ICML)*, 2014, pp. 19 – 27.

[21] Quoc Viet Le, Tamás Sarlós, and Alexander Johannes Smola, "Fastfood: Approximating Kernel Expansions in Loglinear Time," in *Proc. of the 30th Int. Conf. on Mach. Learn. (ICML)*, 2013.

[22] A. Vedaldi and A. Zisserman, "Efficient Additive Kernels via Explicit Feature Maps," *IEEE Trans. on Pattern Anal. & Mach. Intell.*, vol. 34, no. 3, pp. 480–492, 2012.

[23] Zhiyun Lu, Avner May, Kuan Liu, Alireza Bagheri Garakani, Dong Guo, Aurélien Bellet, Linxi Fan, Michael Collins, Brian Kingsbury, Michael Picheny, and Fei Sha, "How to Scale Up Kernel Methods to Be As Good As Deep Neural Nets," 2014, `http://arxiv.org/abs/1411.4000`.

[24] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran, "Low-rank Matrix Factorization for Deep Neural Network Training with High-dimensional Output Targets," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6655–6659.

[25] B. Kingsbury, J. Cui, X. Cui, M. J. F. Gales, K. Knill, J. Mamou, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schlüter, A. Sethy, and P. C. Woodland, "A High-performance Cantonese Keyword Search System," in *Proc. ICASSP*, 2013, pp. 8277–8281.

[26] Brian Kingsbury, "Lattice-based Optimization of Sequence Classification Criteria for Neural-network Acoustic Modeling," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3761–3764.

[27] Tara N Sainath, Brian Kingsbury, Bhuvana Ramabhadran, Petr Fousek, Petr Novak, and Abdel-rahman Mohamed, "Making Deep Belief Networks Effective for Large Vocabulary Continuous Speech Recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 30–35.

[28] Frank Seide, Gang Li, and Dong Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks," in *Proc. of Interspeech*, 2011, pp. 437–440.

[29] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neual Comp.*, vol. 18, no. 7, pp. 1527–1554, 2006.