The Joint Optimization of Spectro-Temporal Features and Neural Net Classifiers

György Kovács[1] and László Tóth[2,⋆]

[1] Department of Informatics, University of Szeged, Szeged, Hungary
[2] Research Group on Artificial Intelligence, Hungarian Academy of Sciences, Szeged, Hungary
{gykovacs,tothl}@inf.u-szeged.hu

Abstract. In speech recognition, spectro-temporal feature extraction and the training of the acoustical model are usually performed separately. To improve recognition performance, we present a combined model which allows the training of the feature extraction filters along with a neural net classifier. Besides expecting that this joint training will result in a better recognition performance, we also expect that such a neural net can generate coefficients for spectro-temporal filters and also enhance preexisting ones, such as those obtained with the two-dimensional Discrete Cosine Transform (2D DCT) and Gabor filters. We tested these assumptions on the TIMIT phone recognition task. The results show that while the initialization based on the 2D DCT or Gabor coefficients is better in some cases than with simple random initialization, the joint model in practice always outperforms the standard two-step method. Furthermore, the results can be significantly improved by using a convolutional version of the network.

Keywords: spectro-temporal features, Neural Net, phone recognition, TIMIT.

1 Introduction

Neurophysiological and biological studies (e.g. [1]) suggest that filters responsive to spectro-temporal modulations can be used for feature extraction in automatic speech recognition. Standard techniques for the extraction of modulation features like these include the application of the 2D DCT [2–4] or a set of Gabor filters [5–7] on the spectro-temporal representation of the speech signal. These features then form the input for some statistical modelling technique such as a hidden Markov model (HMM) or an artificial neural net (ANN). The feature extraction and the statistical modelling steps are usually separate, which is convenient, but suboptimal. Here, we propose to combine these steps in a specially designed neural net, and use this net not just to optimize new feature extraction filter sets, but also to improve the standard 2D DCT and Gabor filters. We will compare the 2D DCT, Gabor and the ANN-based optimized filter sets by evaluating their performance on the TIMIT speech database.

In Section 2, we describe the standard spectro-temporal feature extraction methods. Then in Section 3 we present the concept behind the joint handling of both the feature

[⋆] This publication is supported by the European Union and co-funded by the European Social Fund. Project title: Telemedicine-focused research activities in the fields of mathematics, informatics and medical sciences. Project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073.

I. Habernal and V. Matousek (Eds.): TSD 2013, LNAI 8082, pp. 552–559, 2013.

extraction filters and the neural net classifier. A further refinement – the application of convolutional neural nets – is also elucidated. After, in sections 4 and 5 we present the experiments and discuss the results. Lastly, in Section 6, we draw some brief conclusions about our study, and make a suggestion about future work.

2 Spectro-Temporal Filters

Localized spectro-temporal analysis is a neurophysiologically motivated feature extraction method for speech recognition [6] that has received much attention over the past few years. In this approach we extract spectro-temporally localized patches from the spectrogram of the speech signal, and create features for ASR purposes by processing them using standard filtering methods. Formally, a spectro-temporal feature can be described by the formula

$$o = \sum_{f=0}^{N} \sum_{t=0}^{M} P(f,t)F(f,t), \tag{1}$$

where N and M are the height and width of patch P and filter F, which have to be the same size. There are many different methods for getting the proper coefficients for the filter $F(f,t)$. Below, we describe two well-known methods, then in Section 3 we present a new method.

2.1 2D DCT

A common approach is to process the patches using a 2D DCT, which works with the following filter coefficients:

$$F_{pq}(f,t) = \cos \frac{\pi \cdot (f+0.5) \cdot p}{N} \cos \frac{\pi \cdot (t+0.5) \cdot q}{M}, \quad \begin{array}{l} 0 \leq q \leq N-1 \\ 0 \leq p \leq M-1 \end{array} \tag{2}$$

where N and M are the respective height and width of the filters for f and t, while p and q specify the modulation frequencies of the filter along the frequency and time axis. Using all possible values of p and q would result in as many features as the number of inputs. However, it is common practice [2] to retain just the output of the filters corresponding to the lowest-order coefficients. This is motivated by research suggesting that *"the auditory system may extract [...] relational information through computation of the low-frequency modulation spectrum in the auditory cortex"* [8]. For example, by keeping only 9 coefficients we achieved a performance competitive with the widely used MFCC features [3]. It should be mentioned, however, that though this approach works well in practice, the filters defined by the 2D DCT coefficients are not necessarily the optimal choice.

2.2 Gabor Filters

Another family of filters that has been used for feature extraction in speech recognition is Gabor filters [7]. Their application is motivated by their similarity with the spatio-temporal receptive fields of the auditory cortex. These filters are defined [9] as a product of a two-dimensional Gaussian (3)

$$W(f,t) = \frac{1}{2\pi\sigma_f\sigma_t} e^{-\frac{1}{2}\left(\frac{(f-f_0)^2}{\sigma_f^2} + \frac{(t-t_0)^2}{\sigma_t^2}\right)}, \tag{3}$$

and an oriented sinusoid (4)

$$S_{p,q}(f,t) = e^{j\left(\frac{\pi \cdot f \cdot p}{N} + \frac{\pi \cdot t \cdot q}{M}\right)}, \tag{4}$$

where we iterate f and t over the frequency and time intervals of the patch, and σ_f and σ_t specify the respective bandwiths of the filters. Again, N and M specify the transform size, while p and q specify the slanting of the sinusoid as well as its periodicity. These parameters allow many different filters, and unlike in the case of 2D DCT (where there is an assumption about which filters should be kept), the selection of the right Gabor filters for ASR is a question yet to be answered [9, 10].

3 Joint Optimization of Neural Net Classifiers and Spectro-Temporal Filters

The spectro-temporal features extracted by the filters form the input of a machine learning algorithm, which is usually a hidden Markov model (HMM), though the artficial neural net (ANN) algorithm is also a feasible alternative. The feature extraction and the classification steps are conventionally performed in two distinct steps. Our proposal here is to treat the feature extraction filters as the lowest layer of a neural net, and let the training algorithm tune the filter coefficients as well. To explain how our approach works, let us examine the operation of a simple perceptron model. In general, its output can be obtained using the formula.

$$o = a\left(\sum_{i=1}^{L} x_i \cdot w_i + b\right), \tag{5}$$

where $\mathbf{x}$ is the input of the neuron, L is the length of the input, $\mathbf{w}$ is the weight vector, and b is a bias corresponding to that neuron. For the activation function a we usually apply the sigmoid function, but it is also possible to create a linear neuron by setting a to the identity function. In that case, setting $b = 0$ and $L = N \cdot M$, and representing filter F and patch P in (1) in vector form (which is actually just a notational change), we see that (1) is just a special case of (5). This means that the spectro-temporal filters can be integrated into an ANN classifier system as special neurons, with the filter coefficients corresponding to the weights of the given neuron.

3.1 Structure of the ANN for Combined Feature Extraction and Posterior Estimation

Fig. 1 shows the proposed structure of the ANN that can perform spectro-temporal feature extraction and classification (phone posterior estimation) in one step. When compared to a conventional neural net, the main difference is the introduction of what we

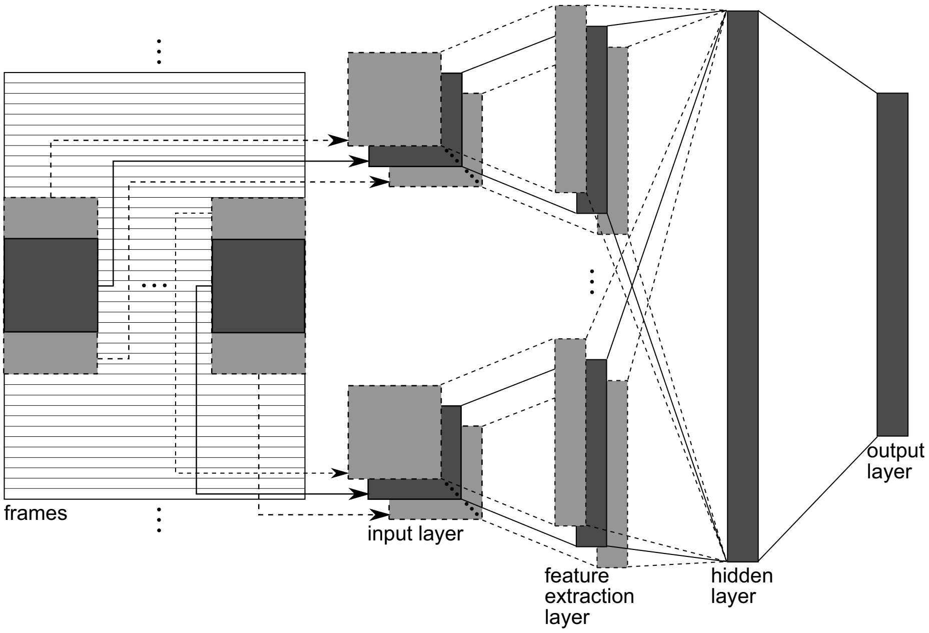

Fig. 1. Structure of the ANN for joint feature extraction and classification. The boxes in light grey correspond to additional units used by the convolutional version of the network.

call the feature extraction layer. Here, the dark grey areas in Fig. 1 mean that the spectro-temporal patches of the speech signal are concatenated to form the input data for the input layer. Then the linear neurons in the feature extraction layer perform the spectro-temporal filtering of (1). The output of this layer is channelled into the hidden layer, and from this point on the system behaves just like a conventional neural net. Hence, if the weights of the feature extraction layer were initialized with 2D DCT or Gabor filter coefficients, and only the weights of the hidden and output layers were tuned during training, then the model would be equivalent to a more traditional system, and incorporating the feature extraction step into the system would be just an implementational detail.

3.2 Fine-Tuning the Spectro-Temporal Filters

The structure in Fig. 1 allows the algorithm to evaluate the spectro-temporal features and the ANN in one step. However, our main goal here was to extend the scope of the backpropagation algorithm to the feature extraction layer as well. This way, we could also train the weights associated with the spectro-temporal filters, and hence fine-tune the initial coefficients. Of course, we had the option to initialize these coefficents randomly (just as we do with all the other weights of the network), but it was also possible to initialize them with the 2D DCT or Gabor coefficients. Usually, as the backpropagation algorithm guarantees only a locally optimal solution, initializing the model with weights that already provide a good solution may help the backpropagation algorithm find a better local optimum than the one found using random initial values.

3.3 Convolutional Neural Nets

It is well known that integrating a longer temporal context into the acoustic features can significantly improve recognition performance. In HMM-based recognition the Δ and $\Delta\Delta$ features are used for this purpose, while in ANN/HMM hybrids a common technique is to use several neighbouring acoustic vectors [11]. Although spectro-temporal features process longer time intervals than tradional techniques (such as MFCC), we observed that adding the delta features to the feature set improves the results [4]. Unfortunately, incorporating the delta features into the joint model would be technically challenging. However, training the network on several neighbouring feature vectors instead of just one is possible by modifying the proposed structure and creating a convolutional neural net [12, 13]. This modification is shown in Fig. 1 by the boxes drawn in light grey. As can be seen, in convolutional networks the feature extraction layer performs its operation on several input patches instead of just one. We should add that the same weights are applied on each input block, so the number of weights will not change in this layer. Obviously, the number of feature vectors processed by the hidden layer increases, but in other respects the hidden and output layers work just as before. Note also that the patches used do not necessarily have to be immediate neighbours, but here we chose this simplest scenario.

4 Experimental Setup

All the experiments reported here were conducted on the TIMIT speech corpus. In the train-test partitioning, we followed the widely accepted standard of having 3696 train sentences and a core test set of 192 sentences. The phonetic labels of the database were “fused” into 39 categories, as is standard practice [14]. To create a phone recognizer from the frame-level phone posterior estimates of the neural net, we utilized a modified version of the Hidden Markov Model toolkit (HTK) [15] with a simple bigram language model.

4.1 Time-Frequency Processing

We chose the log mel-scaled spectrogram as the initial time-frequency representation of the signal. We computed the spectrograms using 400 samples (25 ms) per frame at 160 sample (10ms) hops, and applied a 1024-point FFT on the frames. They were then transformed to a log mel-scale with 26 channels, and each sentence was normalized so as to give zero mean and unit variance. After, a copy of the lower four channels were mirrored in order to avoid artificially down-weighting low frequency bins near the lower edge of the spectrogram.

Table 1. Phone recognition correctness/accuracy scores (the average of 20 independently trained neural nets)

Initial	filter weights	
filter weights	unaltered	trained
Random	73.95% / 67.04%	76.58% / 69.73%
2D DCT	75.29% / 68.81%	76.64% / 69.79%
Gabor	75.25% / 67.59%	76.56% / 69.71%

4.2 Initialization of Filter Coefficients

In an earlier paper, we performed an extensive search to get the optimal size of the time-frequency patches [3]. Based on these findings, the patches – and consequently the filters used here – had a size of 9x9, which corresponded to 9 mels in height and 90 ms in width (9 frames). The filters were applied with a step size of 4 mels (4 channels) in frequency. We tried out three different initialization schemes for the filter coefficients (i.e. the feature extraction layer of the network). In the first case, they were initialized with random numbers, as they usually are with neural nets. In the second case, they were initialized using the 2D DCT filter coefficients we utilized in our studies [3, 4]. And in the third case, the coefficients were initialized based on the Gabor filter coefficients that we found in earlier studies and had given us good results.

4.3 Neural Net Classifier

In the experiments, the classifier we applied was a multilayer neural net modified for this purpose. It consisted of a hidden feature extraction layer with a linear activation function, a hidden layer (with 1000 neurons) with the sigmoid activation function, and an output layer containing softmax units. The number of output neurons was set to the number of classes (39), while the number of neurons in the input and feature extraction layers varied, depending on how many neighbouring patches were actually used. The neural net was trained with random initial weights in the hidden and output layers, using standard backpropagation on 90% of the training data in semi-batch mode, while cross-validation on the remaining, randomly selected 10% of the training set was used as the stopping criterion.

5 Results and Discussion

The phone recognition results we got on the TIMIT corpus using a non-convolutional network are listed in Table 1. The rows of the table correspond to the various filter initialization schemes. The first column shows what we got when the filter coefficients were not trained, while in the second column they were also modified by backpropagation. The first thing we notice is that the joint training method always gives better scores than those obtained with fixed filter coefficients (with significance $p < 10^{-11}$). Second, the initialization techniques gave practically the same results, so starting from the 2D DCT or Gabor filters did not help the optimization process compared to the case with

Table 2. Phone recognition correctness/accuracy scores obtained with the convolutional network (taking the average score of 20 networks)

Initial	filter weights	
filter weights	unaltered	trained
Random	77.65% / 72.02%	78.24% / 72.61%
2D DCT	77.52% / 71.17%	78.14% / 72.52%
Gabor	78.32% / 71.74%	78.46% / 72.83%

random initialization. However, we also see that when there is no fine-tuning of filters involved, the 2D DCT and Gabor filter sets clearly outperform the randomly initialized ones ($p < 10^{-5}$). This sounds reasonable and, in fact, one might expect much worse results from random filters. Interestingly, there are studies which show that in many cases a large set of random base functions can give a representation that is just as good as a carefully selected function set. Recently, a similar study was published for the case of dictionary learning for speech feature extraction [16]. The 'extreme learning machine' of Huang et al. also exploits this suprising fact: this learning model is practically a two-layer network, where both layers are initialized randomly, and the lowest layer is not trained at all [17].

Table 2 shows the phone recognition scores on the TIMIT speech corpus using the convolutive version of the network with 4 neighbouring patches. We see that the difference between the performance of the fine-tuned and the untrained filter sets is smaller than that for Table 1. As regard the initialization methods of the trained filters, Gabor filters gave slightly better results in this case ($p < 10^{-2}$). However, the convolutional network seems to work just as well with random filters as with 2D DCT coefficients. This is an interesting observation that needs to be examined further. But it is already quite clear that a convolutional structure brings about a large improvement to the network. The superior performance of a convolutional network is in accordance with findings in similar studies [12, 13].

6 Conclusions

Here, we presented a method for the joint training of spectro-temporal filters and acoustic models using a special neural network structure. The proposed algorithm was tested in a phone recognition task for the TIMIT speech database. Our results confirmed that joint optimization does indeed result in a better recognition performance than that got by the standard, separate feature extraction and acoustic modelling approach. We also found that further significant improvements could be attained with a convolutional neural network structure. However, starting the training using a filter coefficient set like the 2D DCT set or Gabor set did not always result in better recognition accuracy scores compared to those using simple random initialization. In the future, we would like to study the behaviour of the new network in more detail, so as to learn more about its properties and limitations.

References

1. Aertsen, A.M., Johannesma, P.I.: The spectro-temporal receptive field. A functional characteristic of auditory neurons. Biological Cybernetics 42(2), 133–143 (1981)
2. Bouvrie, J., Ezzat, T., Poggio, T.: Localized Spectro-Temporal Cepstral Analysis of Speech. In: Proc. ICASSP, pp. 4733–4736 (2008)
3. Kovács, G., Tóth, L.: Localized Spectro-Temporal Features for Noise-Robust Speech Recognition. In: Proc. ICCC-CONTI 2010, pp. 481–485 (2010)
4. Kovács, G., Tóth, L.: Phone Recognition Experiments with 2D-DCT Spectro-Temporal Features. In: Proc. SACI 2011, pp. 143–146 (2011)
5. Meyer, B.T., Kollmeier, B.: Optimization and evaluation of Gabor feature sets for ASR. In: Proc. Interspeech 2008, pp. 906–909 (2008)
6. Kleinschmidt, M.: Localized Spectro-Temporal Features for Automatic Speech Recognition. In: Proc. EuroSpeech 2003, pp. 2573–2576 (2003)
7. Kleinschmidt, M.: Methods for capturing spectrotemporal modulations in automatic speech recognition. Acta Acustica United With Acustica 88(3), 416–422 (2002)
8. Greenberg, S.: Understanding Speech Understanding: Towards A Unified Theory Of Speech Perception. In: Proceedings of the ESCA Tutorial and Advanced Research Workshop on the Auditory Basis of Speech Perception, pp. 1–8 (1996)
9. Ezzat, T., Bouvrie, J., Poggio, T.: Spectro-Temporal Analysis of Speech Using 2-D Gabor Filters. In: Proc. Interspeech 2007, pp. 506–509 (2007)
10. Kleinschmidt, M., Gelbart, D.: Improving Word Accuracy with Gabor Feature Extraction. In: Proc. ICSLP 2002, pp. 25–28 (2002)
11. Bourlard, H., Morgan, N.: Connectionist speech recognition: A hybrid approach. Kluwer Academic Pub. (1994)
12. Abdel-Hamid, O., Mohamed, A., Jiang, H., Penn, G.: Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition. In: Proc. ICASSP 2012, pp. 4277–4280 (2012)
13. Vesely, K., Karafiat, M., Grezl, F.: Convolutive Bottleneck Network features for LVCSR. In: Proc. ASRU 2011, pp. 42–47 (2011)
14. Lee, K.-F., Hon, H.-W.: Speaker-independent phone recognition using Hidden Markov models. IEEE Trans. Acoust., Speech Signal Processing 37, 1641–1648 (1989)
15. Young, S., et al.: PC The HTK book version 3.4. Cambridge University Engineering Department, Cambridge (2006)
16. Vinyals, O., Deng, L.: Are sparse representations enough for acoustic modeling? In: Proc. INTERSPEECH (2012)
17. Huang, G.-B., Wang, D.H., Lan, Y.: Extreme learning machines: A survey. International Journal of Machine Learning and Cybernetics 2(2), 107–122 (2011)