AN AUTO-ENCODER BASED APPROACH TO UNSUPERVISED LEARNING OF SUBWORD UNITS

Leonardo Badino, Claudia Canevari, Luciano Fadiga, Giorgio Metta

Istituto Italiano di Tecnologia
Genova, Italy

ABSTRACT

In this paper we propose an autoencoder-based method for the unsupervised identification of subword units. We experiment with different types and architectures of autoencoders to asses what autoencoder properties are most important for this task. We first show that the encoded representation of speech produced by standard autencoders is more effective than Gaussian posteriorgrams in a spoken query classification task. Finally we evaluate the subword inventories produced by the proposed method both in terms of classification accuracy in a word classification task (with lexicon size up to 263 words) and in terms of consistency between subword transcription of different word examples of a same word type. The evaluation is carried out on Italian and American English datasets.

Index Terms— unsupervised acoustic modeling, auto-encoders, deep learning

1. INTRODUCTION

The minimal linguistic resources needed to learn the acoustic models of a standard ASR system consist of a pronunciation lexicon and the phonetic transcription of the speech training data. A desirable additional resource is labeled bootstrap data. If we had a system able to identify the phonetic structure of the training speech data and extract an inventory of subword units in a fully unsupervised fashion we could efficiently recognize words or perform other kinds of task (e.g., keyword spotting) on speech of any language or accent where pronunciation lexicons and transcribed speech data are limited or do not exist. The impact of such a system is even more evident if we consider that ASR systems often fail to achieve a reasonable recognition accuracy when training and testing conditions are mismatched, even when adaptation techniques are applied [1]. Most of the previous work in automatic generation of subword units is based on Gaussian Mixture Models (GMMs). In this paper we propose an alternative strategy based on autoencoder neural networks (AE) [2].

If we are able to generate an inventory of subword units and want to convert speech into orthographic words, including words that are not in the training speech dataset, we need to learn a mapping between graphemes and subwords. That can be accomplished if a part of the training speech data is orthographically transcribed. For this mapping to be successfully learned (from few transcriptions) the subword units must produce pronunciation models that are "consistent", i.e., subword sequences representing instances of a same word type should be identical (if there is no pronunciation variation) or mostly identical, and discriminative(i.e., a one-subword inventory is 100% consistent but cannot discriminate words). Finally subwords must be well identifiable given the acoustic observations, i.e. they should allow good acoustic models. The latter property may affect subword consistency, with poor identifiability resulting in low consistency.

For unsupervised learning of subword units AEs are an interesting alternative to the more popular GMMs in that (i) AEs are generally better than GMMs in discovering the hidden structure of data [3]; (ii) it is straightforward to insert some weak prior knowledge in their training algorithm (see our segmental AE in section 3) or to modify it to make them robust to noise [4];(iii) by piling up AEs one can easily build deep architectures to capture increasingly complex features. Finally, through AEs we can describe phonetic states as bundles of binary features, a distributed description that interestingly recalls the phonological theory of Distinctive Features [5] and allows to explore novel strategies for unsupervised acoustic modeling.

2. RELATED WORK

Different approaches to subword unit generation based on Hidden Markov Models and GMMs have been proposed for various tasks ranging from keyword spotting to large vocabulary speech recognition [6, 7, 8, 9, 10, 11].

In [8] speech fragments are clustered into unspecified word/phrase types through an unsupervised spoken term discovery procedure [8]. The resulting word types are then modeled using HMMs where the number of states depends on the average length of their instances. The final inventory of subwords is created by clustering HMM states across word models. This approach depends on the accuracy of the spoken term discovery algorithm while our method does not need

The authors acknowledge the support of the European Commission project POETICON++ (grant agreement 288382).

any preliminary spoken term discovery and no assumption is made about the number of subwords per word.

In [9, 10] the unsupervised acoustic modeling problem is divided into three sub-tasks: segmentation, segment clustering and acoustic modeling of the clusters. In [10] the three sub-tasks are carried out at once using a Dirichlet Process mixture model. While an indication of the consistency of the obtained clusters/subwords is given by observing co-occurrence of subword units and phones, here we propose an explicit measure of consistency where no assumption is made on what type of subword (e.g., phoneme, syllable) our subwords should resemble to. Contrary to [8, 10] we test the discriminative power of the generated subwords by modeling words as subword transcriptions rather than as subword posteriorgrams.

In [11] consistency and discriminative power of subword units are implicitly evaluated in a speaker-dependent phone recognition task where phone sequences are computed through a subword-to-phone transducer and subwords are generated by iteratively splitting an initial single-state HMM.

To the best of our knowledge there are no previous studies on the use of AE neural networks for the automatic identification of subword units. Multi-layer AE have been typically used to extract new features (typically referred to as bottleneck features) for supervised ASR (e.g., [12]).

Single-layer AEs and Restricted Boltzmann Machines (RBMs) are typically the basic components used to build Deep Learning architectures [2]. Deep Learning architectures have been claimed to be able to identify the phonetic structure of speech. In [13, 14] nodes of the topmost layers of deep networks (stacked sparse AEs with a particular kind of activation function and Deep Boltzmann Machines respectively) exhibit the main acoustic properties of some phones suggesting that these architectures can be effective in the unsupervised generation of a complete inventory of subword units.

3. AUTOENCODERS

An AE consists of an encoder and a decoder part. The encoder maps an input vector $\mathbf{x}$ into an hidden/encoding representation $\mathbf{z}$:

$$\mathbf{z} = f_\theta(\mathbf{x}) = s(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (1)$$

where $\mathbf{W}$ is a weight matrix, $\mathbf{b}$ a bias vector and s is typically the logistic sigmoid function.

The decoder maps back the hidden vector $\mathbf{h}$ to a "reconstructed" input $\mathbf{y}$:

$$\mathbf{y} = g_{\theta'}(\mathbf{h}) = l(\mathbf{W}'\mathbf{h} + \mathbf{b}') \quad (2)$$

If the input data is assumed to be Gaussian distributed, as in the present work, l is typically an identity function and the AE is trained (through backpropagation) to minimize the squared error $||\mathbf{x} - \mathbf{y}||^2$.

A simple variant of the standard AE is the denoising AE [4] where the training input to the AE is transformed in $\tilde{\mathbf{x}}$ by corrupting the input while the training objective is kept unaltered ($||\mathbf{x}-\mathbf{y}||^2$). For Gaussian distributed input the input is corrupted by adding Gaussian noise. The expectation is that the corruption of the input will force the AE to capture the most stable and relevant dependencies between input features and make the AE more robust to noise.

Here we propose a second variant (*s*egmental AE) where a randomly selected subset of training input vectors are substituted by the vectors that immediately follow them (i.e., $\mathbf{x_t}$ is transformed into $\tilde{\mathbf{x}}_\mathbf{t} = \mathbf{x_{t+1}}$). The working assumption is that two consecutive vectors in most cases fall within the same phonetic unit so that the substitution we apply will force the AE to learn dependencies that most characterize that phonetic unit and remove the phonetically irrelevant differences.

Usually deep AEs are built by 'stacking' single-layer AEs. However we discovered that the deep AEs described in [15] produced better results in our experiments. We directly create multi-layered AEs (e.g. a 60-10-60 neural net, where numbers indicate the number of nodes of each of the 3 hidden layers and the middle layer with 10 nodes is the encoding layer) and train them.

Before backpropagation training, all AEs were pre-trained by applying RMB pre-training [15].

4. SUBWORD UNIT GENERATION

Our approach to subword unit generation consists of the following steps:

1. **'Encoder posteriors'**. After AE training, each speech frame f_t is represented as a vector of values of encoding nodes $\mathbf{e}^{(t)} = [e_1^{(t)}, ..., e_i^{(t)}, ..., e_E^{(t)}]$ where E is the number of encoding nodes. Such values can be seen as the probabilities of the encoding nodes to activate (i.e., to have value = 1). We call them "encoder posteriors".

2. **Encoding node binarization**. We then binarize the e_i values. The binarization produces $C = 2^E$ possible configurations/states. Each speech frame can now be assigned an encoding state $\mathbf{b}^{(t)} = [b_1^{(t)}, ..., b_i^{(t)}, ..., b_E^{(t)}]$ (where $b_i^{(t)}$ is the binarization of $e_i^{(t)}$). The binarization requires activation thresholds in order to set to 1 all values above the threshold and to 0 all the lower values. We set the threshold of each encoding node to its mean value.

3. **State modeling**. Subsequently we group together all the frames belonging to the same encoding state and compute the average (i.e., the average encoder posteriors of each state) $\mathbf{m}_k = \frac{1}{|C_k|}\sum_{t \in C_k} \mathbf{e}_t$ where C_k is the set of frames associated to the k_{th} state.

4. **State clustering**. If the number of encoding states is > 64 (i.e., if $E > 6$) we collapse states having similar average encoder posteriors using a simple k-means with number of clusters set to $K = 64$. We then recompute the state means

$\mathbf{m}_k$. The final set of states is our (1-state) subword inventory.

5. **Word/utterance modeling**. We could now represent each training and testing utterance using the $\mathbf{b}_t$ representation and then transform the frame-level subword sequences into atemporal sequences, i.e., transcriptions. However such approach would ignore the sequential nature of speech and the risk of having (long) transcriptions of a same word/phrase type that are largely inconsistent because of 'quick' state transitions at the frame level (e.g., $12-12-3-12-12$) due to poor subword acoustic modeling or noise. Here we propose a strategy to smooth those state transitions.

We build HMM models for each training utterance by first computing the 'subword posteriors' $\mathbf{q^{(t)}} = [q_1^{(t)}, ..., q_i^{(t)}, ..., q_K^{(t)}]$ of each frame :

$$q_i^{(t)} = 1 - \frac{\sqrt{\sum_{j=1}^{E}(\mathbf{e}_j^{(t)} - \mathbf{m}_{i,j})^2/E}}{\sum_{k=1}^{K}\sqrt{\sum_{j=1}^{E}(\mathbf{e}_j^{(t)} - \mathbf{m}_{k,j})^2/E}} \quad (3)$$

Each $q_i^{(t)}$ is in the [0 1] range and $\sum_{i=1}^{K}(q_i^{(t)}) = 1$.

Once we have the 'subword posteriors' we create the matrix of subword transition probabilities T_u for each training utterance. Only self-transitions ($T_u(i,i)$) and transitions to the next state ($T_u(i,i+1)$) are allowed. $T_u(i,i) = T_u(j,j)$ and $T_u(i,i+1) = T_u(j,j+1)$ for any i and j and $T_u(i,i) + T_u(i,i+1) = 1$. That means that the transition matrix is regulated by one single transition penalty parameter, $tP = T_u(i,i)/T_u(i,i+1)$, which affects the expected dwell time.

If we divide the $q^{(t)}$'s by the subword priors we can then model an utterance as a linear HMM. Finally the scaled $q^{(t)}$'s and the transition matrix (weighted by tP/K) are fed into a Viterbi decoder that outputs the best frame-level subword sequence that we then map into the atemporal sequence, i.e. transcription. Finally the HMM model is refined by using the new subword sequence.

6. **Subword acoustic re-modeling**. The previous step reassigns a subword label to each training data frame. We could then recompute the $\mathbf{m}_k$ of each subword and then transcribe again each utterance and repeat until a convergence criterion is reached. Instead we refine the e_i estimation by training a classifier to directly assign subword probabilities to each speech frame. This classifier (in this study a 3 hidden-layer 60-60-60 Deep Neural Network (DNN) classifier which takes as input 1 MFSCs vector) was used to improve the posterior estimation on the test data, but could be used to improve the training utterance models in an iterative procedure.

Note that by increasing E the number of subwords (2^E) can quickly become intractable. This problem could be easily solved if we consider that the number of subwords cannot be larger than the number of training data frames. Alternatively we could train AEs with much larger E and then directly cluster frames with similar state posterior vectors into a subword. Such approach produces worse results (see Results).

7. **Adding weak supervision (optional)**. So far the algorithm is entirely unsupervised. We also experimented with a weakly supervised version of it where the training word examples are given a word label (type). Such information is used to build one single model per word type. We run force alignment on all word examples using the model of a word example and compute the average probability. We repeat using the models of all the remaining word examples. The model with the largest average probability is selected as the model of the word type.

5. DATASETS AND EXPERIMENTAL SETUP

Datasets. We used 3 datasets, one extracted from the Italian Lecce dataset [16] and 2 from the TIMIT corpus [17].

All the utterances of the Lecce corpus are single-word utterances. The vocabulary consists of 73 words + 65 pseudowords. Several word pairs are minimal pairs. The training set contains at most 4 examples of the same word type (with either affirmative or interrogative intonation) for an overall 832 single-word utterances spoken by 4 different female speakers. The test data contains 1 or 2 (when there is an interrogative instance) examples of each word type for an overall 208 single-word utterances spoken by a fifth speaker.

The two TIMIT subsets consists of all content words uttered by female speakers that have at least 4 or 3 examples in the training TIMIT set respectively and 1 in the testing set. The first subset (cv4-1) consists of 162 word types and 810 word examples, the second dataset (cv3-1) of 263 word types and 1052 word examples.

AE training. The acoustic input to the AEs was a 60 (20 + Δ + $\Delta\Delta$) mel-scaled filterbank coefficients (MSFCs) vector extracted from speech signal previously segmented into 25 ms Hamming windows sampled every 10 ms. The input variables were normalized to have 0 mean and 1 standard deviation.

Evaluation measures. Our method was evaluated in a word classification task. When the word label was provided during training (weak supervision) each word type was represented by a single model, whereas in the fully unsupervised setting a word type was represented by the model of one of its word examples (drawn from the set of its examples) . Each testing example was assigned the word type whose model scored highest when force-aligning it. In the fully unsupervised setting the final word classification accuracy is the average of four different accuracies (to take into account the random selection of examples per word type).

The consistency of the transcriptions was assessed by computing the average Levenshtein distance between subword transcriptions of all training word examples of the same word type.

6. RESULTS

GMM vs. AE. Before presenting the main results we provide an empirical motivation to use AEs for unsupervised acoustic

modeling. Table 1 shows that encoder posteriorgrams outperform GMM posteriorgrams in a spoken query classification task similar to that described in [8]. Training and testing words are represented as either GMM or encoder posteriorgrams and word pair similarity is computed using Dynamic Time Warping (with cosine distance). Each test word example is assigned the word type of the training word example most similar to it. Both GMMs and AEs have 32 hidden variables. AEs are trained on MFSC vectors while GMMs are trained on MFCCs.

	Lecce	Timit cv4-1	Timit cv3-1
GMMs	64.9	56.2	57.0
Standard autoencoder	72.1	61.7	63.9

Table 1. *GMM Posteriograms vs. AE Posteriorgrams. Classification accuracies in a spoken query classification task.*

Subword Generation. Table 2 shows the transcription consistency error (CErr) and classification accuracies on the Timit subsets produced by different subword generation systems (with $tP = 1$). UAcc is the accuracy of the entirely unsupervised approach whereas SAcc is the accuracy obtained when training word labels are given.

Standard, denoising and segmental AEs produced similar results with the latter two often producing higher accuracy at close CErr values (see Figure 1). A larger number of encoding nodes produces significantly larger classification accuracy and lower consistency error (e.g., compare 6-node and 16-node encoding layers) while deeper architectures do not seem to produce significant improvements.

Only in the "Segmental + DNN class" case we applied the subword posterior remodeling (step 6 in the subword generation) which always produced higher classification accuracy and lower transcription consistency error.

Comparing results on Timit cv4-1 and cv3-1 we see that although the classification task in cv3-1 is more difficult (as the number of word classes is larger) the classification accuracies are almost identical (although CErr is slightly larger in cv3-1). That is most probably due to the fact that in cv3-1 the AEs are trained on a larger dataset. This is an encouraging result as all our AEs were trained on small datasets (compared to previous work), much larger datasets may produced much better results.

Finally we compared our systems with a fully supervised hybrid DNN-HMM method where the phonetic label of each frame was given (through Expectation Maximization in the Lecce dataset, or manual annotation in TIMIT) and the phone posteriors were computed by the same kind of DNN classifier used for subword acoustic remodeling. The CErr of the supervised method was computed by running the DNN-based phone classifier on all training speech frames and then transforming the frame-level transcription into (atemporal) phone transcriptions. The SAcc difference between the unsupervised and the supervised approach is larger in the TIMIT datasets (see Table 2) than in the Lecce dataset (not shown, SAcc of best unsupervised system = 63.9, SAcc of supervised system = 77.4). The CErr of the best unsupervised systems may seem large when considered as an absolute value but notable when compared to that of the supervised system.

AE Type / System	Hlayers	Timit cv4-1 (cv3-1)		
		CErr	SAcc	UAcc
Standard	6	13.0	29	25.3
Denoising	6	13.1	31.4	27.2
Segmental	6	13.4	34.6	22.8
Denoising	16	12.0	43.8	37.7
Segmental	16	12.1	42.0	36.4
Denoising*	32	12.0	41.4	33.3
Segmental	60-16	12 (13.4)	45.7 (44.9)	37.0 (38.7)
Segmental	60-60-16	12.5 (13.5)	43.8 (44.5)	36.9 (36.5)
Segmental + DNN class	60-60-16	10.7 (12.3)	49.4 (47.5)	-
Phonemes	-	6.7 (7.2)	70.9 (70.7)	-

Table 2. *Transcription consistency error (CErr) and word classification accuracies (UAcc and SAcc) produced by different subword generation systems in the 2 TIMIT subsets. Hlayer indicates the structure of the AE, e.g., a 60-16 AE has 2 hidden layers in the encoding part and 16 is the number of encoding nodes. Denoising* AE is the alternative method commented at the end of step 6 of section 4.*

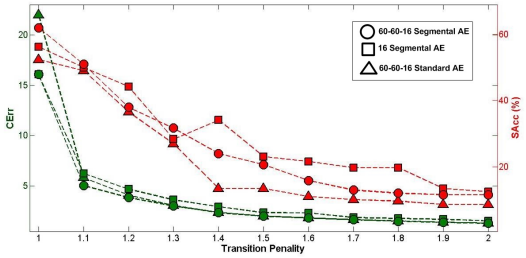

Fig. 1. *Consistency error (green line) and word classification accuracy (red line) (in the Lecce dataset) of three systems as a function of the transition penalty value, the parameter affecting the expected dwell time.*

Figure 1 shows that the transition penalty value tP can be used to trade-off the transcription consistency error and the classification accuracy.

7. CONCLUSIONS

We have presented a strategy based on autoencoders to identify subword units in an unsupervised setting. The subword inventory generated by our approach may produce consistent word transcriptions and good acoustic models. Results are extremely encouraging and we expect to improve them by refining some partly unexplored aspects of our method and by experimenting with larger unsupervised training datasets.

8. REFERENCES

[1] N. Morgan, J. Cohen, S.H. Krishnan Parthasarathi, S.Y. Chang, and S. Wegmann, "Outing unfortunate charactheristics of hmms," in *Final Report: OUCH Project*.

[2] Y. Bengio, "Learning deep architectures for AI," *Technical Report 1312*, 2007.

[3] A. Coates, H. Lee, and A.Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *AISTATS 14*, 2011.

[4] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[5] R. Jakobson, G. Fant, Halle, and Morris, *Preliminaries to Speech Analysis: the Distinctive Features and their Correlates*, Cambridge, Ma.; MIT Press, 1952.

[6] M. Bacchiani and M. Ostendorf, "Joint lexicon, acoustic unit inventory and model design," *Speech Communication*, vol. 29, pp. 99–114, 1999.

[7] R. Singh, B. Raj, and M. Stern, "Automatic generation of subword units for speech recognition systems," *IEEE T-ASLP*, vol. 10, no. 2, pp. 89–99, 2002.

[8] A. Jansen and K. Church, "Towards unsupervised training of speaker independent acoustic models," in *Proceedings of INTERSPEECH*, 2011, pp. 1693–1696.

[9] A. Garcia and H. Gish, "Keyword spotting of arbitrary words using minimal speech resources," in *Proceedings of ICASSP*, 2006.

[10] C.Y. Lee and J. Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012.

[11] B. Varadarajan, S. Khudanpur, and E. Dupoux, "Unsupervised learning of acoustic sub-word units," in *Proceedings of ACL-08:HLT Short Papers*, 2008.

[12] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Proceedings of INTERSPEECH*, 2011.

[13] H. Lee, Y. Largman, P. Pham, and A.Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in Neural Information Processing Systems (NIPS)*, vol. 22, 2009.

[14] M.D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.V. Le, P. Nguyen, A. Senior, V. Vanhoucke, Dean. J., and G. Hinton, "On rectified linear units for speech processing," in *IEEE International Conference on Acoustic Speech and Signal Processing (ICASSP 2013) Vancouver*, 2013.

[15] G.E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[16] M. Grimaldi, B. Gili Fivela, F. Sigona, M. Tavella, Fitzpatrick P., L. Craighero, L. Fadiga, G. Sandini, and G. Metta, "New technologies for simultaneous acquisition of speech articulatory data: 3d articulograph, ultrasound and electroglottograph," in *Proceedings LangTech.*, Rome, Italy, 2008.

[17] J.S. Garofolo, L.F. Lamel, W.M. Fisher, G.J. Fiscus, Dahlgreen N.L. Pallett, D.S., and V. Zue, "Timit acoustic-phonetic continuous speech corpus," in *Linguistic Data Consortium, Philadelphia*, 1993.