INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Frame and Segment Level Recurrent Neural Networks for Phone Classification

Martin Ratajczak[1], *Sebastian Tschiatschek*[2], *Franz Pernkopf*[1]

[1]Graz University of Technology, Signal Processing and Speech Communication Laboratory
[2]ETH Zurich, Learning & Adaptive Systems Group

martin.ratajczak@gmail.com, sebastian.tschiatschek@inf.ethz.ch, pernkopf@tugraz.at

Abstract

We introduce a simple and efficient frame and segment level RNN model (FS-RNN) for phone classification. It processes the input at *frame level* and *segment level* by bidirectional gated RNNs. This type of processing is important to exploit the (temporal) information more effectively compared to *(i)* models which solely process the input at frame level and *(ii)* models which process the input on segment level using features obtained by heuristic aggregation of frame level features. Furthermore, we incorporated the activations of the last hidden layer of the FS-RNN as an additional feature type in a neural higher-order CRF (NHO-CRF). In experiments, we demonstrated excellent performance on the TIMIT phone classification task, reporting a performance of 13.8% phone error rate for the FS-RNN model and 11.9% when combined with the NHO-CRF. In both cases we significantly exceeded the state-of-the-art performance.

Index Terms: Frame and segment level recurrent neural network, neural higher-order conditional random field, phone classification

1. Introduction

We consider phone classification, i.e. the problem of classifying phones from an audio recording given the correct boundaries between different phonetic units. This is important for comparing different classification strategies while at the same time avoiding to align the phones. Many approaches to phone classification [1, 2, 3] are based on computing fixed length feature vectors for each (variable length) phonetic segment, which in turn are used for applying standard classification techniques like logistic regression, support vector machines or neural networks. These fixed length feature vectors are commonly computed from frame level features by heuristic aggregation [1, 2, 3].

This aggregation can result in a loss of relevant information for the phone classification problem as substantiated in our experiments. As a remedy, we propose the FS-RNN model in this paper. Our model first processes the acoustic input at frame level by several layers of bidirectional RNNs spanning the whole input utterance. This way, the RNNs can *propagate* the relevant information for phone classification across segment boundaries—an important property as neighboring phones are known to influence each other [4, 3]. Using the (given) segment boundaries, we extract a fixed number of hidden activations for every phone segment from the last layer of the RNNs and use them as input for several more layers of bidirectional RNNs processing the acoustic input at segment level. Finally, the hidden activations of the last layers of these segment level RNNs are used for classification of the phones. By jointly training the whole model, the RNNs are optimized to extract the *right* information for the classification task while minimizing the loss of relevant information. The FS-RNN achieves an impressive performance of 13.8% phone error rate on the TIMIT phone classification task. Note that the Neural Transducer [5] also uses RNNs across segment boundaries but employs a difficult encoder-transducer-architecture for its operation.

By using features derived from the FS-RNN within linear-chain conditional random fields (CRFs) [6]—well established models for sequence labeling tasks such as speech recognition [7] or phone classification [8]—we are able to achieve even better performance. In particular, we consider *higher-order* (HO) CRFs which facilitate input-independent (such factors depend on the output labels $\mathbf{y}$ only) [9] and input-dependent higher-order factors (such factors depend on both the input $\mathbf{x}$ and output variables $\mathbf{y}$) [10, 11]. These HO factors increase the expressiveness of the CRFs compared to first-order models. *Neural models* amongst other models have been used to parametrize *first-order* factors [12, 13, 14, 15, 16] and *higher-order* factors [3, 17]. By additionally using the FS-RNN features, we augment the model by local factors depending on a single label and on the whole input sequence. In this way we are able to achieve a performance of 11.9% phone error rate on the TIMIT phone classification task.

Our main contributions are: (i) We introduce the FS-RNN model which uses bidirectional RNNs at frame and segment level for phone classification. Therefore, we are able to capture short and long range temporal information more effectively. (ii) In experiments, we demonstrate that FS-RNNs achieve better performance on phone classification than bidirectional RNNs on frame level or segment level only. (iii) In experiments, we evaluate FS-RNN factors integrated in the HO-CRF model and obtain 11.9% phone error rate.

This paper is structured as follows: In Section 2 we briefly describe the phone classification problem and introduce our notation. We propose the FS-RNN model in Section 3. In Section 4 we recap the neural HO-CRFs and show how to equip them with features derived from FS-RNNs. In Section 5 we evaluate our model on the TIMIT phone classification task. Section 6 concludes the paper.

2. Background and Notation

In phone recognition, the task is to map an input utterance given in the form of an audio recording in time domain, to a sequence of phones $\mathbf{y} = (y_1, \ldots, y_T)$ of length T. Typically, in a preprocessing step, frequency domain features, e.g. MFCC and Mel filterbank outputs, are computed from the input utterance by

This work was supported by the Austrian Science Fund (FWF) under the project number P25244-N15. Furthermore, we acknowledge NVIDIA for providing GPU computing resources.

http://dx.doi.org/10.21437/Interspeech.2017-1064

considering short snippets (*frames*) of the time domain data. We denote these features as $\mathbf{x}^{\text{f}} = (\mathbf{x}^{\text{f}}_1, \ldots, \mathbf{x}^{\text{f}}_L)$, where $\mathbf{x}^{\text{f}}_i$ are the features of the i^{th} frame and L is the total number of frames in the processed utterance. Each y_t *corresponds* to a variable length phonetic *segment* spanning multiple frames, whose first frame is a_t and whose last frame is $a_{t+1} - 1$ (we have a_1=1 and for ease of notation we define $a_{T+1} = L + 1$), i.e. the t^{th} segment spans the frame level features $\mathbf{x}^{\text{f}}_{a_t}, \ldots, \mathbf{x}^{\text{f}}_{a_{t+1}-1}$. The frames and phonetic segments are strictly monotonically aligned, i.e. $a_t < a_{t+1}$. In this paper we consider *phone classification*, a special case of phone recognition, in which the alignments $(a_1, \ldots, a_{T+1})$ are known during training and testing.

Heuristic segment level features. Many approaches to phone classification [1, 2, 3] are based on computing fixed length features for each (variable length) phonetic segment. On these fixed length features, many standard machine learning techniques for supervised learning, e.g. logistic regression or support vector machines, can be applied. A common approach for computing these fixed length segment level features is to average frame level features approximately over segment thirds [1] and to stack the results into a single feature vector per segment. We refer to this features as $\mathbf{x}^{\text{s}}$. While this approach of aggregating frame level features is appealing because of its simplicity, it results in a loss of information relevant for the task.

3. The FS-RNN Model

The FS-RNN model processes the input at frame level *and* segment level by separate bidirectional RNNs [18]. As substantiated by the experiments in Section 5.3, this type of processing is important to exploit the (temporal) information of the input more effectively compared to *(i)* models which solely process the input at the frame level using frame level features and *(ii)* models which process the input on segment level using features obtained by heuristic aggregation of frame level features.

Network architecture. The architecture of our FS-RNN model is illustrated in Figure 1. One or more layers of bidirectional RNNs process the input at frame level, i.e. the input to the first RNN layer are the features $\mathbf{x}^{\text{f}}$ computed frame-wise from the input $\mathbf{x}$. This is the *frame RNN* in Figure 1. Note that this frame RNN spans the whole input sequence and is not limited to an individual segment. This enables to propagate information across segments—an important property as neighboring phones are known to strongly influence each other [4, 3]. For each (variable length) phonetic segment of the utterance, we extract the hidden activations from the last layer of the frame RNN at the first, center and last frame using the alignment information (segment boundaries) and stack them, resulting in a segment feature vector. By concatenating these extracted feature vectors for every segment, we obtain segment level features $\mathbf{x}^{\text{s,RNN}} = (\mathbf{x}^{\text{s,RNN}}_1, \ldots, \mathbf{x}^{\text{s,RNN}}_T)$. These segment level features could be used for predicting the labels in each segment but we noticed that further processing on the segment level by additional RNNs improves performance, i.e. the features $\mathbf{x}^{\text{s,RNN}}$ are processed by one or more additional layers of bidirectional RNNs. This is the *segment RNN* in Figure 1. By extracting the hidden activations of the last layer of the segment RNN, we obtain another type of segment level features $\mathbf{x}^{\text{s,RNN2}}$ (using these features for phone classification results in impressive performance gains, cf. Section 5). Finally, for every segment t, we add an output layer with input $\mathbf{x}^{\text{s,RNN2}}_t$ and a softmax activation corresponding to the posterior probability $p(y_t \mid \mathbf{x}^{\text{s,RNN2}}_t)$.

Training. The FS-RNN model is parameterized by the parameters of the frame RNNs $\mathbf{w}^{\text{f}}$, the parameters of the segment RNNs $\mathbf{w}^{\text{s}}$ and the parameters of the output layer $\mathbf{w}^{\text{out}}$. For ease of notation we refer to all these parameters by $\mathbf{w}$. Given the training-data $((\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(J)}, \mathbf{y}^{(J)}))$, which is a collection of J input-output sequence pairs drawn i.i.d. from some unknown data distribution, we optimize the parameters of our model to maximize the conditional log-likelihood:

$$\max_{\mathbf{w}} \quad \sum_{j=1}^{J} \sum_{t=1}^{T_j} \log p(y_t^{(j)} \mid \mathbf{x}^{\text{f}(j)}, \mathbf{a}^{(j)}), \tag{1}$$

where T_j is the length of the j^{th} utterance, $y_t^{(j)}$ is the label of the t^{th} segment in the j^{th} utterance, $\mathbf{x}^{\text{f}(j)}$ are the frame level features of the j^{th} utterance and $\mathbf{a}^{(j)}$ the corresponding alignments. As common in neural networks, we use a mathematical expression compiler, in particular Theano [19], which is supporting symbolic differentiation, to compute the gradients and update the weights $\mathbf{w}$.

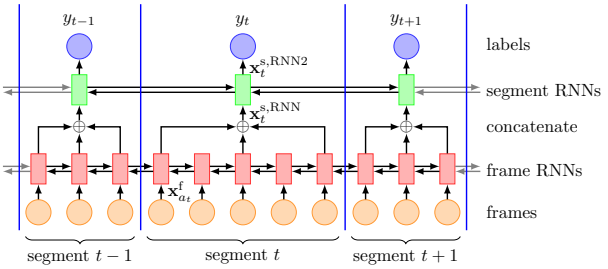

Figure 1: *Proposed* FS-RNN *model. The frame-level input features* $\mathbf{x}^{\text{f}}$ *are processed by multiple layers of RNNs (red). Note that the frame-level RNNs span the whole input sequence and are not limited to individual segments. Separate layers of RNNs (green) process segment level features* $\mathbf{x}^{\text{s,RNN}}$ *extracted from the frame-level RNNs and produce new segment level features* $\mathbf{x}^{\text{s,RNN2}}$*. We use these features to predict one phone for each segment.*

4. Neural Higher-Order CRFs

In this section we show how the FS-RNN model can be used as part of neural higher-order CRFs (NHO-CRFs), which were introduced in [3, 17]. Combining FS-RNNs and NHO-CRFs yields improved performance compared to using either of these models separately, cf. Section 5. We first briefly recap NHO-CRFs for phone classification and then show how features can be extracted from the FS-RNN and used in the NHO-CRF.

NHO-CRF. The NHO-CRF specifies a conditional distribution

$$p^{\text{CRF}}(\mathbf{y} \mid \mathbf{x}, \mathbf{a}) = \frac{1}{Z(\mathbf{x}, \mathbf{a})} \prod_{t=1}^{T} \prod_{n=1}^{N} \Phi_t(\mathbf{y}_{t-n+1:t}; \mathbf{x}, \mathbf{a}), \tag{2}$$

for an output sequence $\mathbf{y}$ of length T given an input sequence $\mathbf{x}$, where $\Phi_t(\mathbf{y}_{t-n+1:t}; \mathbf{x}, \mathbf{a})$ are non-negative factors that can depend on the label sub-sequence $\mathbf{y}_{t-n+1:t}$, the whole input sequence $\mathbf{x}$ and the alignments $\mathbf{a}$, and where $Z(\mathbf{x}, \mathbf{a})$ is the normalization. The factors in $(\text{N} - 1)^{\text{th}}$-order CRFs can depend on label sub-sequences of maximal span N. All factors

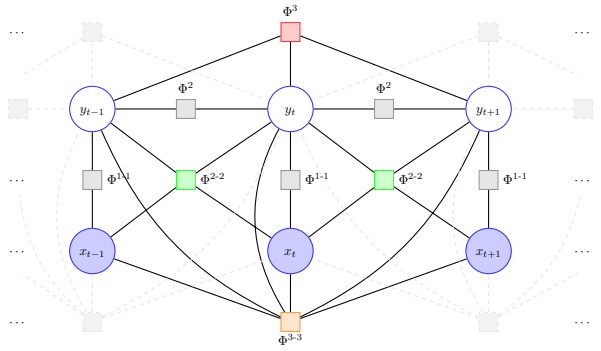

Figure 2: *Factor graph of a HO-CRF of order* 2 *using input-dependent uni-gram factors* $\Phi^{1\text{-}1}$ *and bi-gram transition factors* Φ^2 *(typical) and additionally* 3*-gram factors* Φ^3 *as well as input-dependent factors* $\Phi^{2\text{-}2}$ *and* $\Phi^{3\text{-}3}$*. The* FS-RNN *factors are omitted for legibility.*

$\Phi_t(\mathbf{y}_{t-n+1:t}; \mathbf{x}, \mathbf{a})$ are given in log-linear form, i.e.

$$\Phi_t(\mathbf{y}_{t-n+1:t}; \mathbf{x}, \mathbf{a}) = \exp\left(\sum_k \mathbf{w}_k^n \mathbf{f}_k(\mathbf{y}_{t-n+1:t}; \mathbf{x}, \mathbf{a}, t)\right),$$

where $\mathbf{f}_k(\mathbf{y}_{t-n+1:t}; \mathbf{x}, \mathbf{a}, t)$ are arbitrary vector-valued feature functions and $\mathbf{w}_k^n$ are the weight parameters. These feature functions can for example be indicator functions, linear functions, or functions computed using neural networks. We distinguish *(i)* n-gram input-independent factors which are denoted as Φ^n and depend on an output subsequence of length n, *(ii)* n-m-gram input-dependent factors which are denoted as $\Phi^{m\text{-}n}$ and depend on an output subsequence of length n and an (aligned) input subsequence of length m (on segment level), and *(iii)* FS-RNN factors denoted as $\Phi^{\text{FS-RNN}}$. An illustration of the HO-CRF including all but the FS-RNN factors is shown in Figure 2. For details on the n-gram input-independent and n-m-gram input-dependent factors we refer the interested reader to [3, 17]. In the following we describe the FS-RNN factors that can be derived from our proposed model in Section 3.

FS-RNN factors. These factors depend on a single output label y_t and the *whole* input sequence at *frame level*. These factors use feature functions of the form $\mathbf{f}^{\text{FS-RNN}}(y_t; \mathbf{x}, \mathbf{a}, t) = \mathbf{r}(\mathbf{x}^{\text{f}}, \mathbf{a}, t)\,[\mathbf{1}(y_t = b_1), \mathbf{1}(y_t = b_2), \ldots]^T$, where we use the FS-RNN architecture to model $\mathbf{r}(\mathbf{x}^{\text{f}}, \mathbf{a}, t)$ and $b_1, b_2, \ldots$ are the different phones. In particular, $\mathbf{r}(\mathbf{x}^{\text{f}}, \mathbf{a}, t)$ corresponds to the hidden activations of the last layer of the segment RNNs of the FS-RNN (cf. Section 3 and Figure 1) extracted for the t^{th} segment, i.e. $\mathbf{x}_t^{\text{s,RNN2}}$. It is important to note that the function $\mathbf{r}(\mathbf{x}^{\text{f}}, \mathbf{a}, t)$ maps the *whole* input sequence into a new feature representation at segment level using the alignments.

Parameter Learning. The parameters $\mathbf{w}^{\text{CRF}} = \{\mathbf{w}_k^n \mid \forall k, n\}$ are optimized to maximize the conditional log-likelihood of the training-data $((\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(J)}, \mathbf{y}^{(J)}))$ given the alignment. We compute the conditional log-likelihood by computing the forward recursion as described in [8]. Then, as in the case of the FS-RNN model, we use Theano [19] to compute the gradients and update the weights $\mathbf{w}^{\text{CRF}}$. Note that while we could optimize the parameters $\mathbf{w}^{\text{CRF}}$ of the NHO-CRF and the parameters $\mathbf{w}$ of the feature functions jointly as in [17], we rather kept the parameters $\mathbf{w}$ fixed while learning the parameters of the NHO-CRF for simplicity.

5. Experiments

We evaluated the performance of the proposed FS-RNN model and the NHO-CRF with FS-RNN factors on the TIMIT phone classification task.

5.1. TIMIT Data Set

The TIMIT data set [20] contains recordings of 5.4 hours of English speech from 8 major dialect regions of the United States. The recordings were manually segmented at phone level. We use this segmentation for phone classification. We collapsed the original 61 phones into 39 phones. We computed frame level features $\mathbf{x}^{\text{f}}$ using MFCCs for NHO-CRFs and Mel filterbank outputs for RNNs and in both cases the respective delta and double-delta coefficients. Additionally, for every phonetic segment, we combined the features $\mathbf{x}^{\text{f}}$ of the corresponding frames into segment level feature vectors $\mathbf{x}^{\text{s}}$ by averaging them approximately over segment thirds—details on the pre-processing and data set are presented in [1]. The task is, given an utterance and a corresponding segmentation, to infer the phone within every segment. The validation set is used for parameter tuning. The performance measure is the phone error rate (PER) in [%].

5.2. Experimental Setup

For pre-training the MLPs we used a batch size of 100 samples, an initial learning rate of 0.002 and no ℓ_2-norm regularizer. For training RNNs we used a batch size of 1, an initial learning rate of 0.001 and an ℓ_2-norm regularizer weighted by 0.00025. We used the ADAM optimizer [21] for 100 epochs with a learning rate decay of 0.9. Optimization of the HO-CRF weights was performed using stochastic gradient ascent using a batch size of one sample, an initial learning rate of 0.001, learning rate decay of 0.998, a momentum of 0.0001 and a maximum of 500 of epochs. An ℓ_2-norm regularizer on the model weights was used with a fixed regularization factor of 0.001. We utilized early stopping based on the development data set. Further, the features of the factors of the CRF were normalized to zero mean and unit standard deviation.

5.3. Labeling Results for the FS-RNN model

In a first set of explorative experiments, we evaluate the performance of RNNs that process the input utterances only based on *(a)* frame level features or only on *(b)* segment level features computed by heuristic aggregation. We then show that a processing on frame level *and* segment level by FS-RNNs is advantageous and results in improved performance.

Explorative Experiments. First, we trained bidirectional RNNs either on frames (F-RNN) or on segments (S-RNN). The input to the F-RNNs is $\mathbf{x}^{\text{f}}$, and using the given alignment information, the hidden activations $\mathbf{x}^{\text{s,RNN}}$ (cf. Section 3) are extracted and used as inputs to an output layer with a softmax activation for each segment. The input to the S-RNNs are the segment level features $\mathbf{x}^{\text{s}}$ computed by heuristic aggregation (cf. Section 5.1). In Table 1, we compared the performance of F-RNNs and S-RNNs. F-RNNs outperform S-RNNs with a performance of 15.1% compared to 17.6% PER, respectively.

Improved Performance of FS-RNN. Furthermore, we evaluated the FS-RNN model introduced in Section 3 which processes the input by RNNs on frame and segment level. Surprisingly, we achieved an impressive performance of 13.8% for the best architecture. In Table 1, we compare the performance of different architectures of the FS-RNN to the performance of

F-RNNs and S-RNNs. Note that for various configurations the FS-RNN outperforms the other two models.

Table 1: TIMIT Phone Classification: *Labeling results for F-RNNs, S-RNNs, and* FS-RNN*s for varying numbers of hidden units in the frame RNNs and the segment RNNs. Performance measure: Phone error rate (PER) in [%].*

	# Hidden units: Frame Layers	# Hidden units: Seg. Layers	PER valid	PER test
F-RNN	150	-	15.69	16.67
	150-150	-	14.50	15.35
	150-150-150	-	**14.09**	**15.07**
	150-150-150-150	-	14.23	**14.72**
S-RNN	-	150	18.50	18.97
	-	150-150	17.24	17.64
	-	150-150-150	**16.90**	**17.55**
	-	150-150-150-150	17.69	17.96
FS-RNN	150	150	13.92	14.40
	150	150-150	13.93	15.18
	150	150-150-150	14.71	15.97
	150-150	150	13.25	14.22
	150-150	150-150	13.26	13.94
	150-150	150-150-150	13.97	15.09
	150-150-150	150	**13.22**	**13.81**
	150-150-150	150-150	13.68	14.87
	150-150-150	150-150-150	13.93	15.01
	150-150-150-150	150	13.25	14.06
	150-150-150-150	150-150	13.70	14.08

5.4. Labeling Results for NHO-CRF

In this section, we present results for the NHO-CRF [3] using FS-RNN factors.

Factors and pre-training. Before optimizing the weights of the CRF, we pre-trained the feature functions in the $\Phi^{\text{m-n}}$ factors represented by MLP networks [3] and the $\Phi^{\text{FS-RNN}}$ factors. For pre-training the MLP networks, we defined new datasets $\mathcal{D}_{m\text{-}n}$. The dataset $\mathcal{D}_{m\text{-}n}$ contains all possible sub-sequences of the training data of length n (on the output side) and the corresponding input sub-sequences of length m from the (heuristically aggregated) segment level features $\mathbf{x}^s$. For pre-training the m-n gram MLP networks, we maximized the log conditional likelihood $\sum_{(\mathbf{x}^{s,r},\mathbf{y}^r)\in\mathcal{D}_{m\text{-}n}} \log p^{\text{MLP}}(\mathbf{y}^r|\mathbf{x}^{s,r})$, where r is the index of the r^{th} training example in $\mathcal{D}_{m\text{-}n}$. Note that we train one network for each factor $\Phi^{m\text{-}n}$. For the MLP networks we used the best hyper-parameter settings and the special convolution over segments architecture (MLPCS) from [8]. Furthermore, we used virtual adversarial training (VAT) [22] for training the MLP networks to improve their generalization properties [8]. For pre-training the FS-RNN factors we used the best network architecture from Section 5.3 and maximized the log conditional likelihood in Equation (1).

Sequence Labeling. After the discriminative pre-training of the MLP networks and the FS-RNN for the corresponding factors as described above, we normalized the outputs of the respective feature functions to zero mean and unit variance and used them as features to train NHO-CRFs. Table 2 shows phone classification results for NHO-CRFs including only MLPCS factors versus MLPCS+FS-RNN factors. We incrementally included the m-n-gram MLPCS factors to the NHO-CRFs of the first row in the table containing $\Phi^1, \Phi^2, \Phi^{1\text{-}1}$. Additional higher-order factors ("+" indicates additional factors to the model of previous line) improved consistently the PER. We achieved our best PER result of 11.9% including FS-RNN and all MLPCS factors.

Summary. Finally, we compared our best result to other state-of-the-art methods based on MFCC features as shown in Table 3 and to deep scattering spectrum [23], a method based on more advanced preprocessing which in combination with support vector machines achieves a PER of 15.9%. Recently, Mel-based convolutional neural network ensembles plus MFCC features [24] outperformed previous results on this task with a performance of 15.0%. Our recently published CRF+MLPCS-BN+VAT using standard MFCC features achieves a performance of 13.0% [8]. We obtain a PER of 13.8% using FS-RNN. Furthermore, by using FS-RNN as additional factors in the NHO-CRF we achieve 11.9% PER — the best reported performance on this task.

Table 2: TIMIT Phone Classification: *Labeling results for NHO-CRFs including only MLPCS factors versus MLPCS+*FS-RNN *factors. Performance measure: Phone error rate in [%].*

NHO-CRF Factors	MLPCS valid	MLPCS test	MLPCS+FS-RNN valid	MLPCS+FS-RNN test
$\Phi^1, \Phi^2, \Phi^{1\text{-}1}$	19.66	20.32	13.43	13.87
$+\Phi^{3\text{-}1}$	15.47	16.13	12.61	12.85
$+\Phi^{2\text{-}2}$	14.62	14.91	12.29	12.89
$+\Phi^{4\text{-}2}$	14.00	14.30	12.11	12.44
$+\Phi^3, \Phi^{3\text{-}3}$	13.50	13.21	11.92	11.85
$+\Phi^{5\text{-}3}$	**13.22**	**13.04**	**11.74**	**11.92**

Table 3: TIMIT Phone Classification: *Summary of labeling results. Performance measure: Phone error rate (PER) in [%].*

Model	PER [%]
GMMs ML [25]	25.9
Large-Margin GMM [25]	21.1
Heterogeneous Measurements [1]	21.0
NHO-CRF [17]	17.7
MLPCS-BN+VAT (Isolated) [8]	16.8
Deep Scattering Spectrum [23]	15.9
NHO-CRF (disc. pre-training) [3]	15.8
CNN Ensemble (Mel)+MFCC [24]	15.0
NHO-CRF + MLPCS-BN+VAT [8]	13.0
Proposed models in this paper:	
F-RNN	**15.1**
S-RNN	**17.6**
FS-RNN	**13.8**
FS-RNN+NHO-CRF + MLPCS-BN+VAT	**11.9**

6. Conclusion

We introduced the FS-RNN model which uses frame and segment level bidirectional RNNs for phone classification. Furthermore, we utilized the FS-RNN as a new feature type for the neural HO-CRFs. This improved the performance of neural HO-CRFs significantly. In experiments, we demonstrated excellent performance of our approach on the TIMIT phone classification task, reporting a new state-of-the-art performance of 11.9% phone error rate. In future work we will extend our model to phone recognition [26, 27, 28].

7. References

[1] Andrew K. Halberstadt and James R. Glass, "Heterogeneous acoustic measurements for phonetic classification," in *EUROSPEECH*, 1997, pp. 401–404.

[2] Hung-An Chang and James R. Glass, "Hierarchical large-margin Gaussian mixture models for phonetic classification," in *Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2007, pp. 272–277.

[3] Martin Ratajczak, Sebastian Tschiatschek, and Franz Pernkopf, "Neural higher-order factors in conditional random fields for phoneme classification," in *Interspeech*, 2015, pp. 2137–2141.

[4] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, A. Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] Navdeep Jaitly, David Sussillo, Quoc V Le, Oriol Vinyals, Ilya Sutskever, and Samy Bengio, "A neural transducer," *arXiv preprint arXiv:1511.04868*, 2015.

[6] John Lafferty, Andrew McCallum, and Fernando Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning (ICML)*, 2001, pp. 282–289.

[7] Asela Gunawardana, Milind Mahajan, Alex Acero, and John C. Platt, "Hidden conditional random fields for phone classification," in *Interspeech*, 2005, pp. 1117–1120.

[8] Martin Ratajczak, Sebastian Tschiatschek, and Franz Pernkopf, "Virtual adversarial training applied to neural higher-order factors for phone classification," in *Interspeech*, pp. 2756–2760. 2016.

[9] Nan Ye, Wee S. Lee, Hai L. Chieu, and Dan Wu, "Conditional random fields with high-order features for sequence labeling," in *Neural Information Processing Systems (NIPS)*, pp. 2196–2204. 2009.

[10] Xian Qian, Xiaoqian Jiang, Qi Zhang, Xuanjing Huang, and Lide Wu, "Sparse higher order conditional random fields for improved sequence labeling," in *International Conference on Machine Learning (ICML)*, 2009, pp. 849–856.

[11] Thomas Lavergne, Alexandre Allauzen, Josep M. Crego, and François Yvon, "From n-gram-based to CRF-based Translation Models," in *Workshop on Statistical Machine Translation*, 2011, pp. 542–553.

[12] Hugo Larochelle and Yoshua Bengio, "Classification using discriminative restricted Boltzmann machines," in *International Conference on Machine Learning (ICML)*, 2008, pp. 536–543.

[13] Jian Peng, Liefeng Bo, and Jinbo Xu, "Conditional neural fields," in *Neural Information Processing Systems (NIPS)*, 2009, pp. 1419–1427.

[14] Rohit Prabhavalkar and Eric Fosler-Lussier, "Backpropagation training for multilayer conditional random field based phone recognition.," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 5534–5537.

[15] Laurens van der Maaten, Max Welling, and Lawrence K. Saul, "Hidden-unit conditional random fields.," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 479–488.

[16] Martin Ratajczak, Sebastian Tschiatschek, and Franz Pernkopf, "Sum-product networks for structured prediction: Context-specific deep conditional random fields," in *International Conference on Machine Learning (ICML) Workshop on Learning Tractable Probabilistic Models Workshop*, 2014.

[17] Martin Ratajczak, Sebastian Tschiatschek, and Franz Pernkopf, "Structured regularizer for neural higher-order sequence models," in *European Conference on Machine Learning (ECML)*, 2015, pp. 168–183.

[18] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio, "Gated feedback recurrent neural networks," *CoRR*, vol. abs/1502.02367, 2015.

[19] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

[20] Victor Zue, Stephanie Seneff, and James R. Glass, "Speech database development at MIT: Timit and beyond," *Speech Communication*, vol. 9, no. 4, pp. 351–356, 1990.

[21] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[22] Takeru Miyato, Masanori Koyama, Ken Nakae, and Shin Ishii, "Distributional smoothing with virtual adversarial training," in *International Conference on Learning Representations (ICLR)*, 2016.

[23] Joakim Andén and Stéphane Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, 2014.

[24] Hao Tang, Weiran Wang, Kevin Gimpel, and Karen Livescu, "Discriminative segmental cascades for feature-rich phone recognition," *CoRR*, vol. abs/1507.06073, 2015.

[25] Fei Sha and L.K. Saul, "Large margin Gaussian mixture modeling for phonetic classification and recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006, pp. 265–268.

[26] Geoffrey Zweig and Patrick Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2009, pp. 152–157.

[27] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP)*, 2013, pp. 6645–6649.

[28] Liang Lu, Lingpeng Kong, Chris Dyer, Noah A. Smith, and Steve Renals, "Segmental recurrent neural networks for end-to-end speech recognition," *CoRR*, vol. abs/1603.00223, 2016.