Layer-wise relevance propagation for explainable deep learning based speech recognition

Homanga Bharadhwaj
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
Kanpur, India
homangab@cse.iitk.ac.in

***Abstract*—We develop a framework for incorporating explanations in a deep learning based speech recognition model. The most cited criticism against deep learning based methods across domains is the non-interpretability of the model. This means that the model in itself provides very less or no insight into which features of the input are most responsible for the model's predictions, Layer-wise relevance propagation is an emerging technique for explaining the predictions of deep neural networks. It has shown great success in computer vision applications, but to the best of our knowledge there has been no application of its use in a speech-recognition setup. In this paper we develop a bi-directional GRU based speech recognition model in such a way that layer-wise relevance propagation can be suitably applied to explain the recognition task. We show through simulation results that the benefit of explainability does not compromise on the model accuracy of speech recognition.**

***Index Terms*—Speech Recognition, Explainable Deep Learning, Bi-directional GRU, Layer-wise relevance propagation**

I. Introduction

Neural Networks in conjunction with Hidden Markov Models have been used for a long time in speech recognition [1]. This interest has been renewed with the remarkable prowess demonstrated by deep neural networks [2]–[4]. Deep Neural Networks with their many hidden layers and multiple levels of non-linearities are able to infer higher abstraction level concepts compared to their single hidden layer counterparts. The fact that speech is an inherently dynamic process merits the use of a recurrent architecture (RNN) for temporal modeling [5], [6]. There have been works involving HMM and RNN combined and also involving direct end-to-end training of RNN [3], [7]. The latter has shown to be more successful of late and is the focus of our study in this paper.

Although Deep Neural Networks have been shown to perform very well in most domains, there is a broad consensus about their primary demerit - lack of intuition [8], [9]. As a result, DNN models are non-interpretable i.e. it is very difficult to explain how the model predicts an outcome and what factors influence it the most. It is also important from the point of view of transparency that the model behaves as intended. Some of the prevalent techniques in explaining DNNs exploit the information of local gradients while other methods aim to redistribute the model's final prediction output (relevance) onto the input variables via propagation backwards from the output layer to the input layer [10]–[12]. Sensitivity Analysis (SA) is gradient-based technique that measures relevancy of input variables by computing their partial derivatives with respect to the model output [13], [14]. Another, more popular technique for computing relevances is called Layer-wise Relevance Propagation (LRP) and is based on the relevance conservation principle. It backpropagates the total relevance of the output to each of the inputs through local relevance-propagation laws at each layer of the network [12].

In this work, we extend the LRP framework for a bi-directional GRU based speech recognition model and demonstrate the efficacy of the generated explanations through the method of perturbations [12], [15]. Our approach uses a bi-directional RNN (bi-directional GRU to be more specific) so that future context is also taken into account [3], [16] . We design this model such that LRP can be easily applied to it via modified local relevance propagation rules. LRP has been in use for explaining the predictions of deep architectures in computer vision tasks [12], [17] and natural language processing tasks [18] but not for speech recognition tasks. We believe that with the increase in use of robust deep learning models for speech recognition, the need for explainability i.e. creating an interpretable machine learning approach is crucial for transparency and this paper is an attempt towards bridging that gap.

The rest of the paper is organized as follows: Section 2 describes the primary concepts used in this paper, namely LRP and bi-directional GRU. Section 3 describes the specific LRP propgagation rules used in the model and also the design of the model using bi-directional GRUs. Section 4 presents a detailed analysis of the quality of explanations and the type of relevance distribution generated on a standard phoneme recognition task.

II. Preliminaries

A. Layer-wise relevance propagation

The basic principle underlying layer-wise relevance propagation (LRP) is the layer-wise conservation principle, whereby the prediction of the model $f(\mathbf{x})$ (called relevance) given input $\mathbf{x}$ is redistributed to each intermediate node via backpropagation until the input layer [12]. To formalize this notion, we first

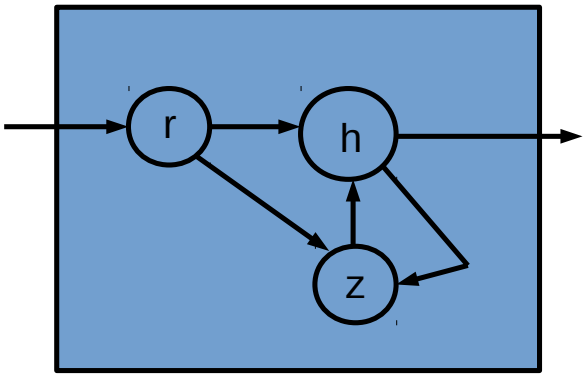


Fig. 1: A GRU cell showing the update gate z, the reset gate r and the current memory content h

note that a DNN consists of multiple elementary computational units of the following form:

$$x_i^{(l)} = h\left(0, \sum_j x_j^{(l-1)} w_{ji}^{(l-1,l)} + a_i^{(l+1)}\right) \tag{1}$$

Here, $h(z)$ is a non-linearity like sigmoid or ReLU, i indexes a neuron for layer l, j runs over all neurons joined to neuron i and $w_{ji}^{(l-1,l)}, a_i^{(l+1)}$ are parameters of the network learned from supervisory data. Several such computational units join to form the entire network. The output $f(\mathbf{x})$ is evaluated in a forward-pass and the parameters are updated by back-propagating the model error. As shown in [12], using the same network-graph architecture, we can redistribute the total relevance $f(\mathbf{x})$ at the output to input-layer relevance using local redistribution rules:

$$R_j^{(l-1)} = \sum_i \frac{x_j^{(l-1)} w_{ji}^{(l-1,l)}}{\sum_k x_k^{(l-1)} w_{ki}^{(l-1,l)}} R_j^{(l)} \tag{2}$$

Here, i indexes a neuron for layer l, j runs over all neurons joined to neuron i. This rule is applied in a back-ward pass through the network starting at the output layer to produce a heatmap which is called the relevance map. It satisfies the relevance conservation property i.e. $\sum_k R_k^{(l)} = f(\mathbf{x})$.

B. Bi-directional RNN (GRU)

Here, we discuss in brief the architecture of the bi-directional RNN, in particular our focus is on bi-directional GRU. A standard RNN computes an output vector sequence $\mathbf{y} = (y_1, .., y_T)$, given an input vector sequence $\mathbf{x} = (x_1, .., x_T)$ by first computing a hidden vector sequence $\mathbf{h} = (h_1, .., h_T)$ iteratively:

$$h_t = G\left(U_{xh} x_t + U_{hh} h_{t-1} + a_h\right) \tag{3}$$

$$y_t = U_{hy} h_t + a_y \tag{4}$$

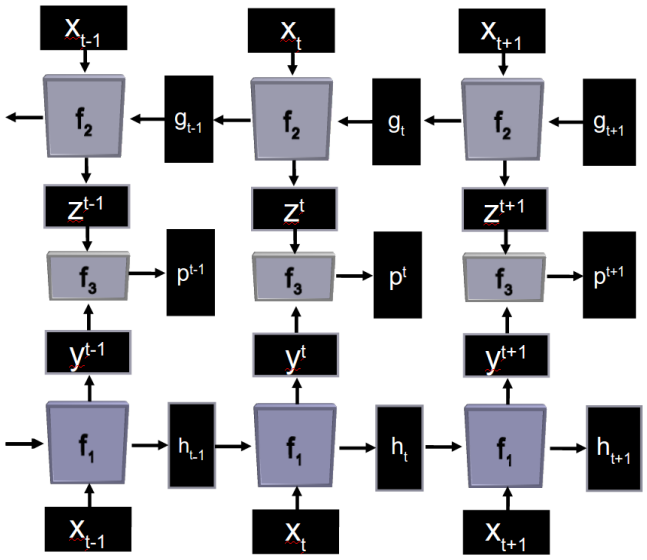


Fig. 2: A schematic diagram of a bi-directional RNN showing all the moving parts of the model. Here, h and g denote the forward and backward hidden states respectively and p denotes the final output and each time-step. For a bi-directional GRU, the functions f_1 and f_2 are replaced by GRU cells. f_3 is typically a simple function, such as a weighted average of h and g as shown in equation 12

Here, U terms denote the weight matrices, a terms denote the bias vectors and G is the function for the hidden layer. It can simply be an element-wise application of a ReLU or signmoid function or in GRUs and LSTMs, a complex combination of many functions. In this paper, we use GRUs, where G is a composite function described as follows:

$$z_t = \sigma(U_{xz} x_t + U_{hz} h_{t-1} + a_z) \tag{5}$$

$$r_t = \sigma(U_{xr} x_t + U_{hr} h_{t-1} + a_r) \tag{6}$$

$$g_t = tanh(U_{xg} x_t + U_{hg}(r_t \circledast h_{t-1}) + a_g) \tag{7}$$

$$h_t = z_t \circledast h_{t-1} + (1 - z_t) \circledast g_t \tag{8}$$

Here, r, z, h are respectively the vectors for reset gate, update gate and hidden state, while σ is the standard sigmoid activation function. Here, $\circledast$ denotes the Hadamard product.

Now, for conventional RNNs (vanilla, GRU or LSTM based) the context is only the past and never the future. This is a reasonable restriction when the index t corresponds to real time and the task is to infer a future item given the history of all previous items. However, in speech recognition, typically transcription of whole utterances occur at once and so there is no reason to not exploit future context. To this end, we use a Bi-directional RNN that processes data in both directions by leveraging two separate hidden layers. If we denote the forward hidden sequence as $\mathbf{h^1}$ and the backward hidden sequence as $\mathbf{h^2}$, then the output sequence $\mathbf{y}$ is iteratively computed as:

$$h_t^1 = G\left(U_{xh^1}x_t + U_{h^1h^1}h_{t-1}^1 + a_{h^1}\right) \quad (9)$$

$$h_t^2 = G\left(U_{xh^2}x_t + U_{h^2h^2}h_{t-1}^2 + a_{h^2}\right) \quad (10)$$

$$y_t = U_{h^1y}h_t^1 + U_{h^2y}h_t^2 + a_y \quad (11)$$

III. Description of the Method

A. LRP for bi-directional GRU

Similar to [3], we focus on end-to-end training of the bi-directional GRU, wherein our model learns to map directly from acoustic to phonetic sequences. In applying LRP, we need to first define the local relevance propagation rules for the model. We need to take care of multiplicative interactions between two lower layer neurons that yield a neuron in the next layer i.e. say $z_1^{(l)} = z_2^{(l-1)} \cdot z_3^{(l-1)}$, where z_1, z_2, z_3 denote computational neurons in the network and l denotes the layer number. This typically occurs in an GRU based RNN, wherein exactly one of the two neurons in the lower layer (say $z_2^{(l-1)}$) is a gate with value in the range [0,1] as output of a sigmoid activation function.

For this particular configuration, to derive the local relevance propagation rule, we note that in the forward propagation, the neuron $z_2^{(l-1)}$ involved in the gate already determines what fraction of the information content in the other neuron ($z_3^{(l-1)}$) must be withheld for the final model prediction. Hence, if $R_1^{(l)}$ denotes the relevance of the neuron in the upper layer, then we re-distribute it to the previous layer neurons as $R_3^{(l-1)} = R_1^{(l)}$ and $R_2^{(l)} = 0$. We apply this rule to all the neuron-connections of the form $z_1^{(l)} = z_2^{(l-1)} \cdot z_3^{(l-1)}$ in both the forward sequence and backward sequence of the bi-directional GRU. For the fully connected layers (or weighted connections of the form described in equation 1), we use two local relevance propagation rules shown below, which are variants of that described in equation 2:

$$R_j^{(l-1)} = \sum_i \frac{x_j^{(l-1)} w_{ji}^{(l-1,l)}}{\sum_k x_k^{(l-1)} w_{ki}^{(l-1,l)} + \epsilon S(\sum_k x_k^{(l-1)} w_{ki}^{(l-1,l)})} R_j^{(l)} \quad (12)$$

$$R_j^{(l-1)} = \sum_i \left(\alpha \cdot \frac{z_{ji}^+}{\sum_k z_{ki}^+} + \beta \cdot \frac{z_{ji}^-}{\sum_k z_{ki}^-} \right) R_j^{(l)} \quad (13)$$

Here,ϵ is a real number in the range (0,1) and $S(\cdot)$ denotes the sign of the quantity within the parenthesis. Also, $z_{ji} = x_j^{(l-1)} w_{ji}^{(l-1,l)}$ and z_{ji}^+ and z_{ji}^- respectively denote the positive and negative parts of z_{ji}, maintaining $z_{ji}^+ + z_{ji}^- z_{ji}$. Further, the constraint $\alpha + \beta$ is enforced in order for the local propagation equations to be conservative layer-wise [12], [15].

B. The Architecture

For defining the end-to-end training method, we need to parametrize a distribution $P(Y|X)$, where Y denotes sequences of phonetic output given input sequence X. To optimize the model end-to-end through gradient descent, we can back-propagate the log-probability of the target sequence of outputs Z, $\log P(Z|X)$ with respect to the parameters of the network. Let the total number of phonemes possible be P, length of Z be Q and the length of X be R.

One of the very first methods proposed is Connectionist Temporal Classification (CTC) [7], [19] which essentially quantifies a distribution over phonemes for every input timestep $P(p|t)$ through a soft-max layer. Eventually, a distribution over alignments between input and target sequences is computed based on the probability distribution of phonemes, which is used by a forward-backward algorithm to compute the posterior probability distribution $P(Z|X)$ [7]. There have been techniques combining RNNs trained via CTC [20] and using RNNs for only prediction of each phoneme give the previous ones (RNN transducer which computes a distribution $P(p|t_1, t_2)$, where t_1 is input timestep and t_2 is output timestep) [21]. In this work, we use a formulation similar to RNN transducer that is convenient for application of LRP.

$$P(p|t_1, t_2) = \frac{e^{y_{t_1,t_2}(p)}}{\sum_{l=1}^{P} e^{y_{t_1,t_2}(l)}} \quad (14)$$

Here $y_{t_1,t_2}(p)$ is obtained at the output of a bi-directional GRU computation (as shown below), which is finally fed to a softmax layer to obtain $P(p|t_1, t_2)$.

$$h_{t_1,t_2}^f = GRU\left(q_{t_1,t_2}^f, r_{t_2}, a_{h^f}\right) \quad (15)$$

$$h_{t_1,t_2}^b = GRU\left(q_{t_1,t_2}^b, r_{t_2}, a_{h^b}\right) \quad (16)$$

$$y_{t_1,t_2} = U_{h^fy}h_{t_1,t_2}^f + U_{h^by}h_{t_1,t_2}^b + a_y \quad (17)$$

Here, q_{t_1,t_2}^f and q_{t_1,t_2}^b are respectively the forward and backward hidden sequences obtained from the CTC network [7], r denotes the hidden sequence of the next-step prediction network, t_1 and t_2 denote time indices for the input and output sequence respectively. $GRU(\cdot)$ denotes the the sequence of computation steps as shown in equations (5) to (9). h_{t_1,t_2}^f and h_{t_1,t_2}^b are the forward and backward hidden states obtained (for a given (t_1, t_2)) after the bi-directional GRU computation.

C. Training

We found that initializing weights of the proposed model with that of a pre-trained CTC network and a pre-trained prediction network yields better results than random-initalization. The prediction network is pre-trained on the phonetic transcriptions of the training data and the output layer weights used while pre-training the networks are removed during re-training. We quantitatively describe the difference in performance of both these variants in Section 4 and also compare the effectiveness of LRP based explanations via the method of perturbations. We compare results of decoding based on both beam search [7] and prefix search [21]. For effective training, we employ early stopping [22], [23] and gradient clipping [23], [24] as regularizers. In addition, we use an adaptive learning rate (ADAM) [25] during optimization.

D. Method of Perturbations for LRP evaluation

We apply LRP to the model by application of local relevance propagation rules for each node in the network as described previously in this section. Now, we need a scheme for validating how effective the generated explanations are. To make things clear, the explanations assign relevance scores to each feature (for speech, samples of the audio sequence; for image, pixels of the input image) of the input sequence with regard to how relevant that feature is towards informing the model's final classification outcome. This in effect generates a heatmap of relevance over the input space.

The perturbation method for evaluating the visualization of what a DNN has learned was introduced in [15]. This method was originally introduced for models where the input is an image and LRP assigns relevance score to each pixel in the input image, but it is straightforward to extend it to our case of a speech-recognition model. The central idea is that if the value of highly important input variables are perturbed as predicted by the model,the decline in prediction score should be steeper than if other less important variables are perturbed. Employing an iterative scheme to perturb input variables, we have an objective measure of explanation quality - steeper decline in classification accuracy is indicative of a more successful explanation scheme.

IV. Simulation Results

A. Setup

We use Tensorflow r1.4 [26] on Python 3 for all relevant programming. All the experiments of phoneme recognition were performed on the TIMIT dataset, which is a well-established standard. We used 18 sets that comprise of 2830 sentences by 357 speakers for training, and one set, namely TID7 (160 sentences by 20 speakers), for testing the model [27]. We encode the audio data with Discrete Fourier Transform Filter Bank with 41 coefficients on mel-scale. The primary and secondary derivatives were also encoded. The input vectors (of size 126) were normalized to have zero mean and unit variance in the training data. There are a total of 61 phoneme labels, all of which were used for training and mapped to 39 classes. We run each experiment 3 times and report the average value so as to reduce variations of random initializations.

For comparison, we consider three schemes of explanation, namely Sensitivity Analysis (SA) [14], ϵ LRP (based on equation 13) and $\alpha\beta$ LRP (based on equation 14). The models on which we apply the three explanation schemes include Bi-directional GRU (Bi-GRU), Bi-directional LSTM (Bi-LSTM), Uni-directional GRU (Uni-GRU), Uni-directional LSTM (Uni-LSTM) and CTC. Since SA and LRP are techniques for neural networks, we limit ourselves to comparison against NN based models for speech recognition.

B. Relevance map comparison by perturbation

We quantitatively validate the generated phoneme-level relevance map obtained by SA, ϵLRP and $\alpha\beta LRP$ by adopting a scheme of random perturbation. We first train all the models and apply the SA and LRP explanation schemes. This gives us a relevance map over the samples for each audio sequence. Now, for perturbation, we consider the test audio data and in decreasing order of relevance of samples, we replace the corresponding sample by a random noise sample from a uniform distribution. After applying random perturbation, we re-predict on the audio data (test again on the phoneme recognition task but with the new audio data that has been perturbed) and note the accuracy averaged over the entire test audio data. As expected, we observe a decrease in accuracy with more number of perturbation steps, where in subsequent steps, less relevant samples are being perturbed.

Figure 3 shows the decrease in accuracy for each model on the three explanation schemes. We observe that irrepsective of the explanation scheme, the proposed Bi-GRU approach has the steepest decrease in accuracy with perturbation indicating that it is the most explainable model. This is important because explainability is a major challenge in deep learning based models and it is preferable to use models that are most explainable without compromising on the accuracy of predictions. To analyze if the gains in explainability for Bi-GRU come at the cost of lower prediction accuracy, we compare all the five models on various metrics in Table 2. We observe that Bi-GRU has comparable precision and recall values (slightly higher infact) as Bi-LSTM. Uni-GRU and Uni-LSTM have lower accuracy on the test dataset, which is in line with previous research on phoneme recognition [cite].

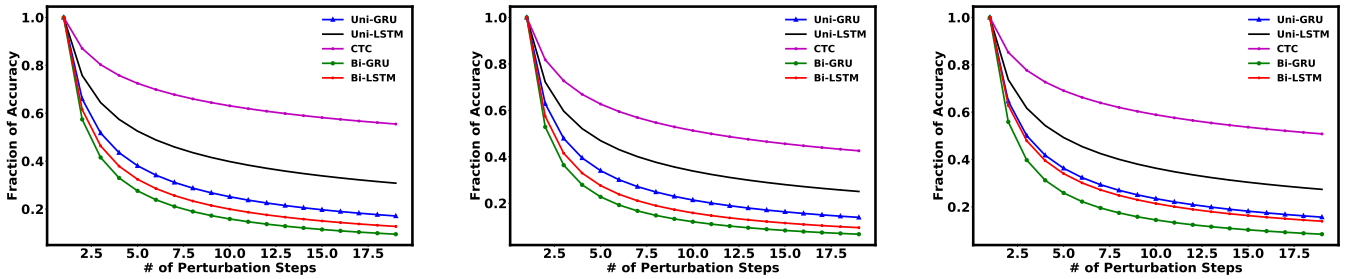
From Figure 3, we can also compare how good the three explanation schemes themselves are. It is evident that for $\alpha\beta$ LRP, the decrease in accuracy with perturbation steps is the steepest across all models, indicating that the relevance predicted by $\alpha\beta$ LRP is the most pertinent for inferring the samples which most contribute to the task of phoneme recognition. Similar results on comparison between SA and LRP approaches on CNN architectures for document classification were reported in [12]. It is interesting to note that this holds even for phoneme recognition.

C. Distribution of relevance over the sentence length

Here we perform an interesting analysis of the relevance heatmap obtained by averaging over all sentences (spoken by all speakers) in the TIMIT dataset for each of the five models. We divide the audio stream of each sentence into 12 equal intervals and sum up the relevances of samples in each interval for a particular phoneme. We then average over all possible phonemes in the corpus and finally normalize to 1, since we are computing a distribution. As observed in Figure 4, the distribution is not exactly symmetric for most models and has a peak before the center of the sentence, indicating higher relevance of samples in the first half of spoken sentences on an average for the task of phoneme recognition.

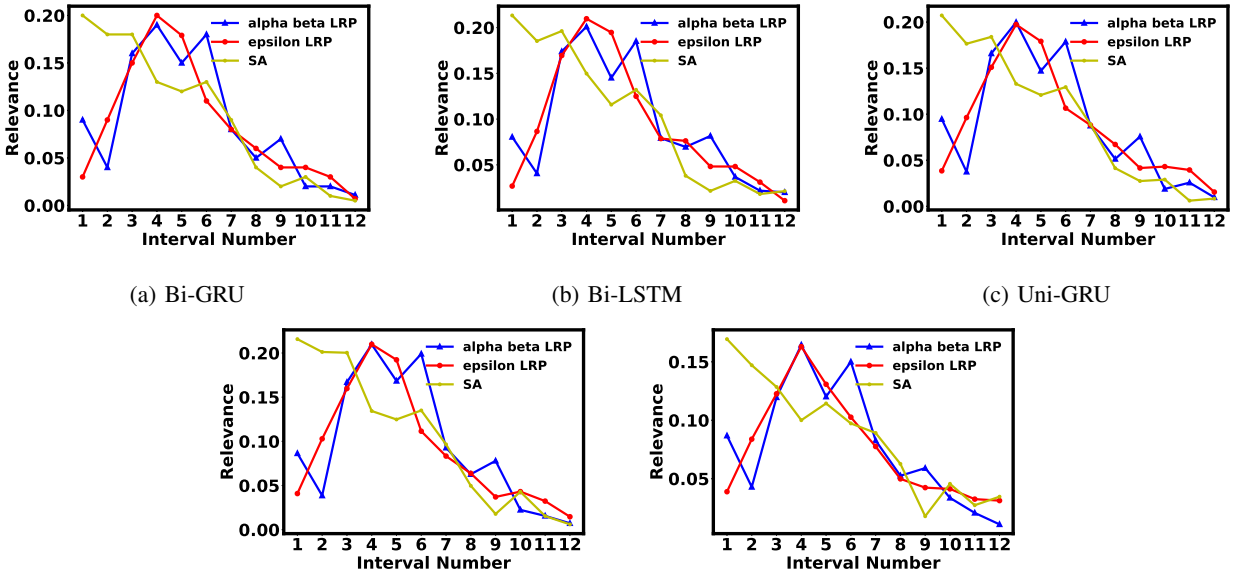
D. Analysis of the model accuracy

Table 1 draws a comparison between the different baseline models and the proposed approach (Bi-GRU), with regard to the test-set prediction accuracy of phoneme recognition on the



(a) Sensitivity Analysis (SA) (b) ϵ Layer-wise relevance propagation (c) $\alpha\beta$ Layer-wise relevance propagation

Fig. 3: Sensitivity Analysis (SA), ϵ Layer-wise relevance propagation and $\alpha\beta$ Layer-wise relevance propagation for different models. The decrease in accuracy averaged over the entire test-set for all categories is shown as a function of the number of perturbations. The greater the decrease in accuracy the better, because it implies that the explanations indicating relevancy of input variables is more accurate.



(a) Bi-GRU (b) Bi-LSTM (c) Uni-GRU

(d) Uni-LSTM (e) CTC

Fig. 4: Relevance distribution over the audio sequence of sentences divided into 12 equal intervals and averaged over all phonemes. Each figure correponds to a different model and shows the relevance distribution with respect to the three explanation schemes, namely $\alpha\beta$ LRP, ϵ LRP and SA

TIMIT dataset. From Figure 3, it is evidence that Bi-GRU is the most explainable in the sense that all the three explanation schemes evaluated via the method of perturbations yield the most-suited input space relevance map for this model. From Table 1, we see that even for the task of phoneme recognition, Bi-GRU outperforms the other models (albeit performs only slightly better than Bi-LSTM). The primary inference to draw from this observation is that explainability does not come at the cost of prediction accuracy. This result should motivate the development of speech-recognition models like the one described in this paper, which are explainable in addition to performing a high accuracy on specific speech recognition tasks.

V. Conclusion

In this paper we described the design of an explainable speech recognition model. We extend the idea of layer-wise relevance propagation to a Bi-directional GRU based speech recognition model and demonstrate its efficacy in generation of explanations for the phoneme recognition task. Although Bi-directional RNN has been in use for speech recognition for a long time now, but we develop the Bi-directional GRU framework in such a way that layer-wise relevance propagation can be suitable applied to it. Layer-wise relevance propagation has been in use for explaining computer vision tasks and to some extent natural language tasks, but its application for deep-learning based speech recognition tasks is innovative and

TABLE I: A comparison of various models on the phoneme recognition task in TIMIT corpus. The data has been processed as Described in Section 4.1 and the results reported here are on the test set. Here, P denotes precision and R denotes recall. The value of Error is in percentage. 'Pre' in the column heading denotes initialization with weights from the pre-trained networks as described in Section 3.3. The rest of the models are initialized with random weights. It can be observed that the models initialized by pre-trained weights perform better than their random initialization counterparts on almost all the metrics.

	CTC	Uni-LSTM	Uni-GRU	Bi-LSTM	Bi-GRU	Pre Uni-LSTM	Pre Uni-GRU	Pre Bi-LSTM	Pre Bi-GRU
P@3	0.34	0.41	0.40	0.47	0.48	0.43	0.44	0.49	0.51
P@5	0.44	0.49	0.51	0.57	0.57	0.50	0.54	0.59	0.60
R@3	0.66	0.73	0.71	0.78	0.80	0.75	0.73	0.79	0.82
R@5	0.71	0.75	0.75	0.80	0.82	0.76	0.76	0.82	0.84
Error	32.6	25.4	24.2	20.5	17.2	25.0	23.8	18.7	16.3

in our opinion opens up interesting lines of research in explainable speech recognition models. We also demonstrate that the explainability introduced by applying layer-wise relevance propagation does not compromise the accuracy in the main task of phoneme recognition.

REFERENCES

[1] Herve A Bourlard and Nelson Morgan, *Connectionist speech recognition: a hybrid approach*, vol. 247, Springer Science & Business Media, 2012.

[2] Abdel-rahman Mohamed, George E Dahl, Geoffrey Hinton, et al., "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech & Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[3] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[4] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio, "Batch-normalized joint training for dnn-based distant speech recognition," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 28–34.

[5] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 167–174.

[6] Anthony J Robinson, "An application of recurrent nets to phone probability estimation," *IEEE transactions on Neural Networks*, vol. 5, no. 2, pp. 298–305, 1994.

[7] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[8] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[9] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[10] Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Muller, and Wojciech Samek, "Analyzing classifiers: Fisher vectors and deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2912–2920.

[11] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller, "How to explain individual classification decisions," *Journal of Machine Learning Research*, vol. 11, no. Jun, pp. 1803–1831, 2010.

[12] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, pp. e0130140, 2015.

[13] Muriel Gevrey, Ioannis Dimopoulos, and Sovan Lek, "Review and comparison of methods to study the contribution of variables in artificial neural network models," *Ecological modelling*, vol. 160, no. 3, pp. 249–264, 2003.

[14] Yannis Dimopoulos, Paul Bourret, and Sovan Lek, "Use of some sensitivity criteria for choosing networks with good generalization ability," *Neural Processing Letters*, vol. 2, no. 6, pp. 1–4, 1995.

[15] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2660–2673, 2017.

[16] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[17] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," in *International Conference on Artificial Neural Networks*. Springer, 2016, pp. 63–71.

[18] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek, """ what is relevant in a text document?": An interpretable machine learning approach," *PloS one*, vol. 12, no. 8, pp. e0181142, 2017.

[19] Alex Graves, "Supervised sequence labelling," in *Supervised sequence labelling with recurrent neural networks*, pp. 5–13. Springer, 2012.

[20] Kyuyeon Hwang and Wonyong Sung, "Sequence to sequence training of ctc-rnns with partial windowing," in *International Conference on Machine Learning*, 2016, pp. 2178–2187.

[21] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[22] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto, "On early stopping in gradient descent learning," *Constructive Approximation*, vol. 26, no. 2, pp. 289–315, 2007.

[23] Yoshua Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade*, pp. 437–478. Springer, 2012.

[24] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: a system for large-scale machine learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.

[27] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.