# Recurrent Poisson Process Unit for Speech Recognition

**Hengguan Huang[1], Hao Wang[2], Brian Mak[1]**
[1]The Hong Kong University of Science and Technology
[2]Massachusetts Institute of Technology
{*hhuangaj,mak*}@*cse.ust.hk*, hwang87@mit.edu

## Abstract

Over the past few years, there has been a resurgence of interest in using recurrent neural network-hidden Markov model (RNN-HMM) for automatic speech recognition (ASR). Some modern recurrent network models, such as long short-term memory (LSTM) and simple recurrent unit (SRU), have demonstrated promising results on this task. Recently, several scientific perspectives in the fields of neuroethology and speech production suggest that human speech signals may be represented in discrete point patterns involving acoustic events in the speech signal. Based on this hypothesis, it may pose some challenges for RNN-HMM acoustic modeling: firstly, it arbitrarily discretizes the continuous input into the interval features at a fixed frame rate, which may introduce discretization errors; secondly, the occurrences of such acoustic events are unknown. Furthermore, the training targets of RNN-HMM are obtained from other (inferior) models, giving rise to misalignments. In this paper, we propose a recurrent Poisson process (RPP) which can be seen as a collection of Poisson processes at a series of time intervals in which the intensity evolves according to the RNN hidden states that encode the history of the acoustic signal. It aims at allocating the latent acoustic events in continuous time. Such events are efficiently drawn from the RPP using a sampling-free solution in an analytic form. The speech signal containing latent acoustic events is reconstructed/sampled dynamically from the discretized acoustic features using linear interpolation, in which the weight parameters are estimated from the onset of these events. The above processes are further integrated into an SRU, forming our final model, called recurrent Poisson process unit (RPPU). Experimental evaluations on ASR tasks including ChiME-2, WSJ0 and WSJ0&1 demonstrate the effectiveness and benefits of the RPPU. For example, it achieves a relative WER reduction of 10.7% over state-of-the-art models on WSJ0.

## Introduction

Recently, recurrent neural network (RNN) has been successfully applied to diverse machine learning problems such as automatic speech recognition (ASR) (Graves, r. Mohamed, and Hinton 2013), machine translation (Bahdanau, Cho, and Bengio 2015) and computer vision (Vinyals et al. 2015). Although end-to-end encoder-decoder approaches are getting more popular in ASR, the hybrid long short-term memory recurrent neural network hidden Markov model (LSTM RNN-HMM) is still the dominant model used in industry (Yu and Li 2018). It is beneficial for an acoustic model to capture long-term dependencies of the observations at different times. However, the sequential gates computation of LSTM limits its parallelization potential. Simple recurrent unit (Lei and Zhang 2017) and quasi-RNN (Bradbury et al. 2016), simplify the implementation of LSTM-RNN, and increase the speed of computation for each processing step by dropping the connections between the hidden states and the LSTM gates, allowing them to be computed in parallel.

The hybrid RNN-HMM in acoustic modeling is essentially a generalized version of a dynamic Bayesian network (DBN), which is usually characterized by discretizing the time series data and capturing the dependency of those discretized items. According to the research in speech production and neuroethology, human speech signals may be encoded in point patterns involving acoustic events in the speech signal and neural spikes in the brain (Stevens 2000; Esser et al. 1997). Such points in time are referred to as acoustic event landmarks in (Stevens 2002). Based on this hypothesis, it may pose some challenges for RNN-HMM acoustic modeling: firstly, it arbitrarily discretizes the continuous input into the interval features at a fixed frame rate, which may introduce discretization errors and have a negative impact on the model performance accordingly; secondly, the occurrences of such acoustic events are unknown and such data are unavailable. Further more, the training targets of RNN-HMM are usually obtained from the recognition results of other (usually inferior) DBN models and misalignments are inevitable.

On the other hand, the temporal point process is a powerful mathematical tool to describe the latent mechanisms governing the occurrences of observed random events. It is a random process whose realization consists of a sequence of isolated events with their time-stamps. Due to their generality, point processes have been widely used for modeling phenomena such as earthquakes (Hawkes 1971a), human activities (Malmgren et al. 2008), financial data (Bacry et al. 2015), context-aware recommendations (Du et al. 2015) , etc. A common property of the problems above is that the precise event time intervals can carry important information about the underlying dynamics, which otherwise are not available from the sequence of interval features that are evenly

sampled from the continuous signal. Major research in this area focuses on exploring the observed event data to model the underlying dynamics of the system, while our work attempts to deal with the situation where acoustic events are not available/observed even during training.

In this paper, we develop a deep probabilistic model called recurrent Poisson process unit (RPPU) to deal with the aforementioned problems. The hybrid ASR system under the above hypothesis can be factored into three steps:

- Allocate the training acoustic events localized in time at the HMM state level to better align with the training targets.
- Reconstruct/sample a series of acoustic features from the interval features originally sampled at a fixed frame rate from the allocated acoustic events.
- Follow the traditional ASR processing procedure using the newly reconstructed acoustic features as additional inputs.

The first step is achieved by constructing a recurrent Poisson process (RPP), which consists of a collection of homogeneous Poisson processes (Kingman 1992) at a series of time intervals. In the proposed point process, the intensity function is determined by an RNN hidden state encoding the past history of the acoustic signal. Sampling from intensity-based models is usually performed via a thinning algorithm(Ogata 1981), which is computationally expensive. Our method is sampling-free and it provides a solution in an analytical form which ensures computational efficiency. In the second step, the better aligned acoustic features are dynamically reconstructed through a linear interpolation in which the weight parameters are estimated from the acoustic events drawn from the RPP. Finally, those estimated acoustic features are provided to an RNN as additional input to perform the HMM state prediction in a traditional way.

The objective function of RPPU is designed to strike a balance between the generation of arrival times of the latent acoustic events for clean training data and encoding sufficient uncertainty to capture the variability caused by the discretization errors and misalignments. Notably, RPPU can be trained with the standard backpropagation through time (BPTT) (Werbos 1990). The experiments on CHiME-2, WSJ0 and WSJ0&1 show that our new model consistently outperforms the conventional LSTM, SRU and quasi-RNNs.

## Background and Related Work

### Modeling acoustic HMM states with RNN

A hybrid RNN-HMM ASR system (Graves, Jaitly, and Mohamed 2013) consists of an RNN estimating posterior probabilities for HMM states of context-dependent phones conditioned on the acoustic input. Typically, for a sequence of training examples $[(\mathbf{x}_{t_1}, \mathbf{y}_{t_1}), (\mathbf{x}_{t_2}, \mathbf{y}_{t_2}), ..., (\mathbf{x}_{t_M}, \mathbf{y}_{t_M})]$ with $\mathbf{x}_{t_i} \in \mathbb{R}^n$, $\mathbf{y}_{t_i} \in \mathbb{R}^k$, for $1 \leq i \leq M$, the acoustic feature $\mathbf{x}_{t_i}$ is given as inputs to the network, while the vector $\mathbf{y}_{t_i}$ denotes the ground truth and is represented by a one-hot vector of $K$ context-dependent HMM states. It can be represented by the following two equations:

$$\mathbf{h}_{t_i} = G_\theta(\mathbf{x}_{t_i}, \mathbf{h}_{t_{i-1}}) \tag{1}$$

$$\hat{\mathbf{y}}_{t_i} = softmax(\mathbf{W}_y \mathbf{h}_{t_i} + \mathbf{b}_y) \tag{2}$$

where $\mathbf{h}_{t_i} \in \mathbb{R}^r$ describes the hidden state. The first equation defines the state transition mapping, in which the hidden state $\mathbf{h}_{t_i}$ is a nonlinear function of the current input $\mathbf{x}_{t_i}$ and the previous hidden state $\mathbf{h}_{t_{i-1}}$ and $\theta$ is the parameter set of $G$. $\mathbf{W}_y$ and $\mathbf{b}_y$ are the weight matrix and the bias of the output layer respectively. The output mapping usually adopts the softmax function to calculate the predictions $\hat{\mathbf{y}}_{t_i}$. In this sense, the RNN can be viewed as the "state classifier" optimized by minimizing the negative log-likelihood or cross-entropy:

$$-\log P(\mathbf{Y}|\mathbf{X}) = -\sum_{i=1}^{M}\sum_{k=1}^{K} \mathbf{y}_{t_i,k} \log \hat{\mathbf{y}}_{t_i,k} \,, \tag{3}$$

where both the lengths of the input sequence $\mathbf{X}$ and the target context-dependent HMM state sequence $\mathbf{Y}$ are $M$; the total number of context-dependent HMM states is $K$ and they are usually generated by forcefully aligning the training utterance with its transcription using an inferior acoustic model such as a GMM-HMM. The ultimate goal of the hybrid ASR system is to generate the most likely words or phoneme sequence. This is done by running the Viterbi algorithm (Forney 1973) within the HMM framework.

One major limitation of hybrid acoustic modeling is that training targets are generated from a family of dynamic Bayesian network models, e.g. GMM-HMM and RNN-HMM, and the arbitrary discretization of the continuous acoustic signal could result in mis-alignments. Nonetheless, the capability of handling such uncertainty only comes from the conditional output probability density given the deterministic transition function of a standard RNN. To effectively deal with this issue, the acoustic RNN model must be capable of approximating the arrival time of each training target and reconstructing/sampling the acoustic features dynamically based on the estimated arrival time, and this is the main focus of our model.

### Poisson Process

A Poisson process is a temporal point process defined in continuous time, in which the inter-arrival times are drawn i.i.d from an exponential distribution. It has a strong renewal property that the process can probabilistically restart at each arrival time, independently of the past. This enables us to describe the probabilistic behavior of the process via the intensity function $\lambda(t)$, which is a non-negative function. Within a small interval $[t, t+dt]$, the probability of an arrival is $\lambda(t)dt$.

By considering a sequence of arrival times of acoustic landmarks $L = \{t_1, t_2, \ldots, t_N\}$ sampled from a Poisson process $\mathfrak{P}$ over an interval $[0, T]$, we have:

$$L \sim \mathfrak{P}(g(\lambda(t)) \tag{4}$$

$$\Delta t_i = t_i - t_{i-1} \sim g(\lambda(t)) \tag{5}$$

where g is the exponential density function; $\Delta t_i$ is the inter-arrival time. We place the first landmark at time 0 for simplicity, and thus $t_0 = 0$. Given the observation of the previous landmark at time $t_{i-1}$, the probability that no landmark occurs up to time t since $t_{i-1}$ is $\mathbb{P}(t_i > t) = e^{\int_{t_{i-1}}^{t} -\lambda(t)dt}$.

Then, the probability that the first landmark lies in the interval $[t_i, t_i + dt]$ since $t_{i-1}$ is computed as the product of $\mathbb{P}(t_i > t)$ and $\lambda(t_i)dt$, leading to the corresponding density function:

$$f_i(t) = \lambda(t) e^{\int_{t_{i-1}}^{t} -\lambda(t)dt} \tag{6}$$

By the strong renewal property, the likelihood of the whole arrivals $L$ over an interval $[0, T]$ takes the form:

$$\begin{aligned} P(L|\lambda(t)) &= \mathbb{P}(t_{N+1} > T) \times \prod_{i=1}^{N} f_i(t_i) \\ &= e^{\int_0^T -\lambda(t)dt} \prod_{n=1}^{N} \lambda(t_n) \end{aligned} \tag{7}$$

where $\mathbb{P}(t_{N+1} > T)$ is the probability that no landmark is observed in the interval $(t_N, T]$. It may not be tractable as an integral over the intensity function does not always have an analytic expression. But it is not the case for a homogeneous Poisson process with a constant intensity.

## Related Work

Temporal point processes have been a principled framework for modeling phenomena on an event-by-event basis across a wide range of domains. It has originally been used for modeling earthquakes (Hawkes 1971b; 1971a) in seismology. More recently, in social network, a hawkes process has been used to model timing and rich features of social interactions (Zhou, Zha, and Song 2013); in human activity modeling, Poisson Processes have been applied to model the inter-arrival time of human activities (Malmgren et al. 2008).

A major limitation of these existing work is that they often make strong assumptions about the generative processes of the event data, which may not be well-suited for real world problem. Therefore, most of existing work focus on enhancing the flexibility of point process models, e.g. , a nonparametric Bayesian approach of point processes have been explored in (Teh and Rao 2011); (Mei and Eisner 2017) extended the multivariate Hawkes process (Hawkes 1971a) to a neurally self-modulating multivariate point process using a continuous-time LSTM. Similarly, (Du et al. 2016) proposed a model based on marked temporal point process that model the event timings and the markers with the help of a LSTM. However, these methods focus only on modeling the observable (not latent) events, our proposed work try to develop a framework which explicitly models acoustic events as latent variables, consequently producing better phoneme-level alignment and leading to better ASR performance.

The closest work in ASR is (Jansen and Niyogi 2009), in which an acoustic model based on marked Poisson process has been proposed for a sub-task of event-based ASR . This subtask requires the speech signal to be segmented prior to acoustic modeling. Therefore, the timings of the acoustic events are provided during training and the model parameters of intensity function is learned by simply using maximum likelihood estimation (MLE). In contrast, the annotations of such acoustic events are not available in our setting; hence direct supervised learning via MLE is not applicable for our task.

# Recurrent Poisson Process Unit

## Problem Formulation

Let us consider a time interval $[0, t_N]$, where time is discretized into $N$ frames of duration 10ms. Given a sequence of acoustic features $\mathbf{X} = \{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, ..., \mathbf{x}_{t_N}\}$, our goal is to approximate a sequence of arrival times $\widetilde{L} = \{\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_N\}$ so that a new sequence of acoustic features $\widetilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_{\tilde{t}_1}, \tilde{\mathbf{x}}_{\tilde{t}_2}, ..., \tilde{\mathbf{x}}_{\tilde{t}_N}\}$ can be estimated which should align better with the given targets $Y = \{\mathbf{y}_{t_1}, \mathbf{y}_{t_2}, ..., \mathbf{y}_{t_N}\}$ in terms of resolution and precision, and a more robust acoustic model can be learned from these newly estimated training samples.

## Recurrent Poisson Process

A recurrent Poisson process (RPP), consisting of a collection of homogeneous Poisson processes (Kingman 1992) for a series of time intervals, is a special type of temporal point process, in which the intensity function is determined by an RNN hidden state encoding the history of an acoustic signal. One may be tempted to learn the temporal point process simply using maximum likelihood estimation (MLE). Unfortunately, the annotation of the latent acoustic events in the acoustic speech signal is not available; hence direct supervised learning via MLE is not possible. Our RPP addresses this challenge by modeling these latent acoustic events as latent variables, which are then used as part of the generative process that is linked to the training targets.

**Generate Timings for a Recurrent Poisson Process** Assume that we are given N intensities $\{\lambda_{t_1}, \lambda_{t_2}, \ldots, \lambda_{t_N}\}$, and a sequence of input features $\{\mathbf{x}_{t_{1-d}}, \ldots, \mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_N}\}$ , in which $d$ context frames are padded to the left. Suppose the starting acoustic landmark is at time $t_{1-d}$ and it follows a homogeneous Poisson process with intensity $\lambda_{t_1}$ at the interval $[t_{1-d}, 2t_1 - t_{1-d}]$ which starts at time $t_{1-d}$ and is centered at $t_1$. We will try to obtain the time estimate $\tilde{t}_1$ of the first acoustic landmark, and then repeat the procedure to obtain the whole $\widetilde{L} = \{\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_N\}$. To be more specific, given the $(i-1)$-th acoustic landmark at the estimated time $\tilde{t}_{i-1}$ and an interval $[\tilde{t}_{i-1}, 2t_i - \tilde{t}_{i-1}]$, the probability density of the next landmark being in this interval can be written as:

$$f_i^*(t) = \frac{f_i(t)}{\int_{\tilde{t}_{i-1}}^{2t_i - \tilde{t}_{i-1}} f_i(t)dt} = \frac{\lambda_{t_i} e^{-\lambda_{t_i}(t - \tilde{t}_{i-1})}}{1 - e^{-2\lambda_{t_i}(t_i - \tilde{t}_{i-1})}} \,. \tag{8}$$

Then we can estimate the time for the $i$-th landmark as its expected value in following closed- form solution:

$$\begin{aligned} \tilde{t}_i &= \int_{\tilde{t}_{i-1}}^{2t_i - \tilde{t}_{i-1}} t f_i^*(t)dt \\ &= 2t_i - \tilde{t}_{i-1} + \frac{1}{\lambda_{t_i}} - \frac{2(t_i - \tilde{t}_{i-1})}{1 - e^{-2\lambda_{t_i}(t_i - \tilde{t}_{i-1})}} \end{aligned} \tag{9}$$

Generally, the aforementioned point process can be factored into N independent homogeneous Poisson processes. For the $i$-th sub-process with intensity $\lambda_{t_i}$, the arrival sub-sequence $L_i$ is drawn:

$$L_i \sim \mathfrak{P}_i(g(\lambda_{t_i})) \,. \tag{10}$$

It has only one single arrival and $L_i = \{t_i\}$. Thus, for a sequence of observations $\mathbf{L} = \{L_1, L_2, \ldots, L_N\}$, its likelihood is the following joint probability density:

$$P(\mathbf{L}|\lambda_{t_1}, \lambda_{t_2}, \ldots, \lambda_{t_N}) = \prod_{n=1}^{N} P(L_i|\lambda_{t_1}) \tag{11}$$

where

$$P(L_i|\lambda_{t_1}) = \lambda_{t_i} e^{-\lambda_{t_i}(t_i - t_{i-1})} \, . \tag{12}$$

**Conditional Intensity Function for a Recurrent Poisson Process** In the neural spiking modelling (Snoek, Zemel, and Adams 2013), the intensity function of the neural spikes is usually conditioned on the external covariate. In a similar spirit, we determine our intensity function using the hidden state $\tilde{h}_{t_i}$ of an RNN, which encodes the temporal dependencies among the past history of the acoustic signal $X$. To avoid the explosion of $\frac{1}{\lambda_{t_i}}$, we define the inverse of the intensity function as:

$$\frac{1}{\lambda_{t_i}} = c\sigma(\phi(\tilde{h}_{t_i})) + \epsilon \tag{13}$$

such that the inverse of the intensity function is upper bounded by $c$ and the intensity function is upper bounded by $\frac{1}{\epsilon}$. This is important as it limits the search space during optimization when $\phi(\cdot)$ are neural networks, which transform the hidden states into a scalar.

## Recurrent Poisson Process Unit: Integrate Recurrent Poisson Process into RNN

The arrival time sequence of the acoustic landmarks generated from a recurrent Poisson process is on the real line. However, we are only given the discretized input sequence. The missing input vectors are reconstructed by linear interpolation as follows:

$$\widetilde{\mathbf{x}}_{\tilde{t}_i} = \sum_{n=1}^{N} \mathbf{x}_{t_n} \max(0, 1 - |\tilde{t}_i - n|) \, . \tag{14}$$

This enables the loss gradients to reach both the inputs and the estimated arrival times from the recurrent Poisson process.

In this paper, we use simple recurrent unit (Lei and Zhang 2017) to implement RNN. SRU simplifies the architecture of LSTM and dramatically reduces the computational time by dropping the connections between its hidden states and gates so that computation at the gates can be done in parallel.

Below are the updating formulas of recurrent Poisson process unit.

$$\left[\hat{\mathbf{r}}_{t_i}, \hat{\mathbf{f}}_{t_i}, \widehat{\mathbf{c}}_{t_i}\right] = \mathbf{W}_x \left[\mathbf{x}_{t_i}, \tilde{\mathbf{x}}_{\tilde{t}_i}\right] + \mathbf{b} \tag{15}$$

$$\mathbf{r}_{t_i} = \sigma(\hat{\mathbf{r}_{t_i}}) \tag{16}$$

$$\mathbf{f}_{t_i} = \sigma(\hat{\mathbf{f}_{t_i}}) \tag{17}$$

$$\mathbf{c}_{t_i} = \mathbf{f}_{t_i} \odot \mathbf{c}_{t_i - 1} + (1 - \mathbf{f}_{t_i}) \odot \hat{\mathbf{c}}_{t_i} \tag{18}$$

$$\mathbf{h}_{t_i} = \mathbf{r}_{t_i} \odot tanh(\mathbf{c}_{t_i}) + (1 - \mathbf{r}_{t_i}) \odot \mathbf{W}_h \left[\mathbf{x}_{t_i}, \tilde{\mathbf{x}}_{\tilde{t}_i}\right] \tag{19}$$

where $\mathbf{r}_{t_i}$ are the reset gate outputs; $\mathbf{f}_{t_i}$ are the forget gate outputs; $\mathbf{c}_{t_i}$ are the memory cell outputs; $\mathbf{W}_x$ and $\mathbf{W}_h$ are the weight matrices; $\mathbf{b}$ are the gate bias vectors; $\mathbf{h}_{t_i}$ are the hidden state outputs; any quantity with a 'hat' (e.g. $\widehat{\mathbf{c}}_{t_i}$) is the activation value of the quantity before an activation function is applied; $\odot$ is the element-wise multiplication operation; $\sigma$ is the sigmoid function.

## Learning

Our design of the loss function aims at striking a balance between the generation of arrival times of the latent acoustic events for clean training data and encoding sufficient uncertainty to capture the variability caused by the discretization errors and misalignments.

We use the standard Poisson process as the prior for the recurrent Poisson process to restrict the complexity of the approximated recurrent Poisson process. We measure the distance between the recurrent poisson process and the standard Poisson process by Kullback–Leibler divergence in terms of the inter-arrival time distribution. The inter-arrival time distribution for the $i$-th sub-process is defined as an exponential distribution $g(\lambda_{t_i})$. Since all these distributions are independent, we can enjoy the additive property of KL divergence of these two processes:

$$\sum_{i=1}^{N} KL(g_s(\lambda = 1)||g(\lambda_{t_i})) = \sum_{i=1}^{N} (\lambda_{t_i} - \log(\lambda_{t_i}) - 1) \tag{20}$$

where $g_s(\lambda = 1)$ is the inter-arrival time distribution for a stand Poisson process.

Although we assume the original arrival times of landmarks, $\{1, 2, \ldots, N\}$, are noisy, the negative likelihood of this "incorrec" time sequence can be a desirable regularizer to avoid overfitting in noisy conditions.

As such, the total loss is the sum of the cross-entropy loss, negative log likelihood of noisy arrival time and the KL divergence between the underlying recurrent Poisson process and the standard process:

$$\begin{aligned} &- \log P(\mathbf{Y}|\mathbf{X}) - \alpha \log(P(\mathbf{L}|\lambda_{t_1}, \lambda_{t_2}, \ldots, \lambda_{t_N})) \\ &+ \beta \sum_{i=1}^{N} KL(g_s(\lambda = 1)||g(\lambda_{t_i}))) \end{aligned} \tag{21}$$

Since the negative log likelihood terms and the KL terms has exactly the same form in term of optimization. The final objective can be written as:

$$- \log P(\mathbf{Y}|\mathbf{X}) + \gamma \sum_{i=1}^{N} (\lambda_{t_i} - \log(\lambda_{t_i})) \tag{22}$$

where $\gamma$ is the weight for the regularization term. We adopt the backpropagation through time (BPTT) for joinly training both recurrent Poisson process and recurrent Poisson process Unit.

# Experiments

## Datasets

We evaluated the proposed RPPU on three ASR corpora: ChiME-2 (Vincent et al. 2013), WSJ0 (Garofolo et al. 1993) and WSJ0&1 (Garofolo et al. 1993; Consortium and others 1994).

Table 1: Model configuraions for all datasets and the training time for CHiME-2. L: number of layers; N: number of hidden states per layer; P: number of model parameters; T: Training time per epoch (hr).

| Model | L | N | P | T |
|---|---|---|---|---|
| LSTM | 3 | 2048 | 130M | 0.71 |
| SRU | 12 | 2048 | 156M | 0.32 |
| Quasi-RNN | 12 | 1024 | 117M | 0.22 |
| RPPU | 12 | 1024 | 142M | 0.37 |

**CHiME-2** CHiME-2 corpus is a medium-large vocabulary corpus, which was generated by convolving clean Wall Street Journal (WSJ0) utterances with binaural room impulse responses (BRIRs) and real background noises at signal-to-noise ratios (SNRs) in the range [-6,9] dB. The training set contains about 15 hours of speech with 7138 simulated noisy utterances. The transcriptions are based on those of the WSJ0 training set. The development and test sets contain 2460 and 1980 simulated noisy utterances, respectively. The WSJ0 text corpus, consisting of 37M words from 1.6M sentences, is used to train a trigram language model with a vocabulary size of 5k.

**WSJ0** WSJ0 is a clean speech corpus recorded in a clean environment using close microphones. The standard WSJ0 si-84 training set with 7138 clean utterances was used for acoustic modeling. The evaluation was performed on eval92-5k which is a 5k-vocabulary non-verbalized test set, and the si-dt-05 dataset was used as the development set. The 5k trigram language model used for evaluation was trained from the WSJ0 text corpus.

**WSJ0&1** WSJ0&1 is a complete Wall Street Journal speech corpus, which involves speech data from both WSJ0 and WSJ1. The training set WSJ0&1 si-284 with 36515 utterances contains approximately 80 hours of speech, 95% of which was used for training. The rest was used as the development set. The evaluation of WSJ0&1 was performed on the dev93-20k and eval93-20k test sets, both of which are 20k open-vocabulary non-verbalized test sets. The evaluation was performed with a 20k trigram language model trained from the transcription of WSJ0&1 si-284. We report the speech recognition performance in terms of word error rate (WER).

## Feature Extraction and Preprocessing

Acoustic hidden Markov models (HMM) based on Gaussian-mixture model (GMM), LSTM, SRU and quasi-RNN were built. GMM-HMM models employed fMLLR-adapted (Gales and others 1998) 39-dimensional MFCC features. All neural-network-based models used 40-dimensional Mel-filterbank coefficients (Biem et al. 2001) without their derivatives. Inputs of all neural networks consisted of the current frame together with its 4 right contextual frames. We performed per-speaker mean and variance normalization for the input to all the neural network models.

Table 2: WER (%) on test set of CHiME-2.

| Model | WER |
|---|---|
| DNN Kaldi s5 | 29.1 |
| LSTM | 26.1 |
| SRU | 26.2 |
| Quasi-RNN | 26.1 |
| RPPU | **24.4** |

## Training Procedure

GMM-HMM employed fMLLR-adapted 39-dimensional MFCC features and was trained using the standard Kaldi s5 recipe (Povey and others 2011). They were then used to derive the state targets for subsequent RNN training through forced alignment for ChiME-2, WSJ0 and WSJ0&1. Specifically, the state targets were obtained by aligning the training data with the DNN acoustic model through the iterative procedure outlined in (Dahl et al. 2012).

All RNNs were trained by optimizing the categorical cross entropy using BPTT and SGD. Prior to optimization, all the weight matrices were initialized following a LeCun Normal distribution introduced in (Klambauer et al. 2017). We applied a dropout rate of 0.1 to the connections between recurrent layers. The learning rate for LSTM/SRU, Quasi-RNN and RPPU models was initially set to 0.25, 0.2, and 0.07 respectively. We decayed the learning rate until it went below $1 \times 10^{-6}$.

**Models** We adopted SRU as the building block to construct the proposed RPPU and compare our proposed model with the following baselines: (i) The LSTM with three stacked layers; (ii) SRU with 12 stacked layers; (iii) quasi-RNN with 12 stacked layers and the highway connection (Lei and Zhang 2017).

The LSTM has only three stacked layers because we did not observe WER reduction by stacking more layers. To ensure similar numbers of model parameters for different models, we set the number of hidden states per layer to 2048 for both LSTM and SRU, and 1024 for both quasi-RNN and RPPU. The filter width of quasi-RNN was 3, which ensured a similar number of parameters for the RPPU. Our RPPU had 2 context frames padded to the left of the input and two previous hidden states padded to the left of the input of each hidden RPPU layer. For simplicity, in the intensity function of RPPU, $c$ was set to 100 and $\epsilon$ was set to 0.01 (these two hyperparameters can be tuned to further improve performance).

## Results and Analysis

**Results on CHiME-2** Table 1 shows the model configurations of the baseline models and the new RPPU model for all datasets. The training time per epoch for CHiME-2 is also provided. The timing experiments used the Theano package and were performed on a machine running the Ubuntu operating system with a single Intel Core i7-7700 CPU and a GTX 1080Ti GPU. Each model took around 25 iterations, and their average running time is reported. We can see that

Table 3: Detailed WER (%) on the CHiME-2 test set.

| Model | -6 Db | -3 Db | 0 Db | 3 Db | 6 Db | 9 Db |
|---|---|---|---|---|---|---|
| LSTM | 42.4 | 33.5 | 26.7 | 21.1 | 17.3 | 15.3 |
| SRU | 42.5 | 34.0 | 26.2 | 22.2 | 17.4 | 15.1 |
| Quasi-RNN | 42.1 | 32.8 | 27.8 | 20.8 | 17.5 | 15.6 |
| RPPU | **39.9** | **31.1** | **24.9** | **20.3** | **16.0** | **13.2** |

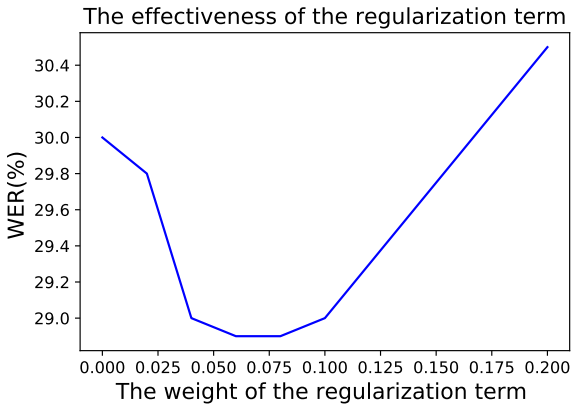


Figure 1: WER on Development set of CHiME-2 by varying the weight of the regularization term

SRU is much faster than LSTM and our RPPU runs almost as fast as SRU while having a similar number of parameters.

Table 2 shows the word recognition performance of the baseline models and the new RPPU model for CHiME-2. Firstly, we can see that all of our RNN baselines achieve a similar WER. These baselines perform much better than the DNN baseline from Kaldi s5. Our proposed RPPU performs the best among all the candidates in terms of WER, outperforming the RNN baselines by about 1.7% absolute. We also report the detailed WERs as a function of the SNR in CHiME-2 shown in Table 3. For all SNRs, the RPPU outperforms other models by a large margin. This suggests that incorporating the recurrent Poisson process into RNN structures lends itself to the model's robustness.

To validate the effectiveness of the regularization term in RPPU for CHiME-2, we varied its weight $\gamma$ to find the best configuration, as can be seen in Figure 1. We obtained the best performance in the development set when the weight $\gamma$ is around 0.08. We hence set $\gamma$ to 0.08 as our final configuration based on this observation. These results indicate the effectiveness of our proposed objective function.

**Analyze the Property of RPP** Here, we took the generated time points from the recurrent Poisson process (RPP) of the 5-th layer of RPPU to perform both qualitative and quantitative analyses.

The standard Poisson process is the prior of the RPP in RPPU; hence we used the distance between the estimated value and its mean to approximately measure RPP's flexibility. To better understand how RPP works, we randomly took two utterances "423c02162" and "423c02166" at 9DB and -3DB SNR respectively, from the development set and generated the associated arrival times from the RPP. We display the estimated time points associated with acoustic events at two different SNRs in Figure 2. We can see that as the noise level increases, the estimated time points go towards the mean of a standard Poisson process. This suggests that RPPU can produce the time points based on the noise level: less flexibility is allowed for RPP's point generation when the data is too noisy.

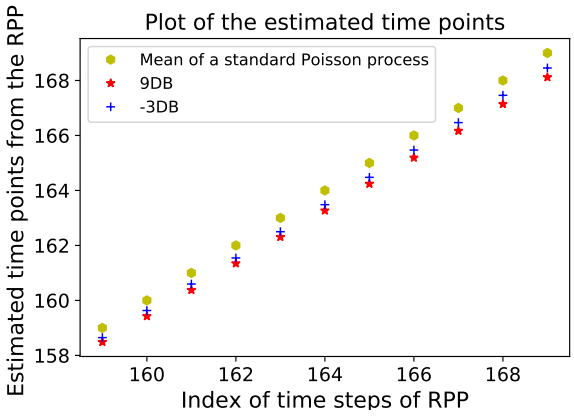


Figure 2: Arrival times produced by RPPU at each time steps. The yellow points represent the mean of a standard Poisson process. The blue pluses and red stars represent the generated time points estimated from the randomly chosen utterance at 9DB and -3DB SNR, respectively.

Table 4: Similarity with the ground-truth phoneme-level alignment for our baseline and estimated alignment using RPP on development set of CHiME-2

| Alignment type | Similarity(%) |
|---|---|
| Baseline alignment | 55.5 |
| Estimated alignment using RPP | **64.7** |

To evaluate how these generated time points can be helpful in better aligning the acoustic inputs with the acoustic HMM states, we conducted the analysis on the development set of CHiME-2. Notice that the noisy CHiME-2 data is generated from the clean WSJ0 data; thus the ground-truth alignments of the development set of WSJ0 serve also as the ground-truth alignments for the corresponding development set of CHiME-2 data. Here, the ground-truth phoneme-level alignment for development set of CHiME-2 is obtained by force aligning the development set si-dt-05 of WSJ0 using the WSJ0 fMLLR-based DNN acoustic model.

We firstly obtained the baseline frame-level alignment by force aligning the CHiME-2 noisy development set using CHiME-2 DNN, which is the model we used to derive the training labels of CHiME-2 RNNs. It is then transformed to phoneme-level alignment, serving as our baseline phoneme-level alignment. We then used the generated time points from RPP to obtain an estimated phoneme-level alignment by replacing the integer time indices of the baseline frame-level

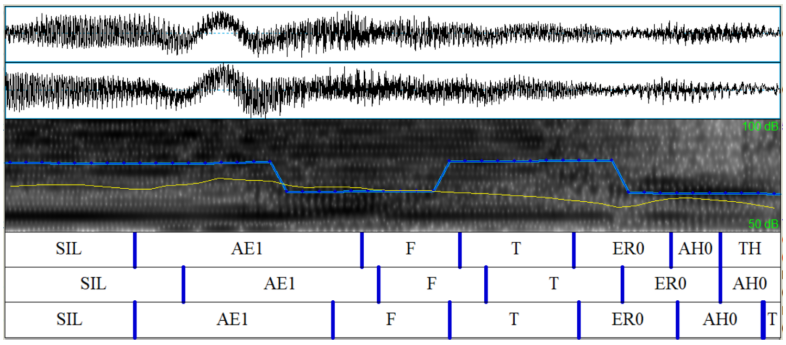


Figure 3: In the textgrids, the first tier represents the ground-truth alignment generated from the clean utterance of WSJ0, the second tier represents our baseline alignment generated from the noisy utterance of CHiME-2, this last tier denotes the corresponding estimated alignment using RPP. The blue line and the yellow line in the middle spectrogram represents pitch and intensity, respectively.

Table 5: WER (%) on evaluation set eval92-5k of WSJ0.

| Model | WER |
|---|---|
| DNN (Chen and Mak 2015) | 3.2 |
| LSTM | 2.8 |
| SRU | 2.8 |
| Quasi-RNN | 2.8 |
| RPPU | **2.5** |

Table 6: WER (%) on evaluation sets of WSJ0&1.

| Model | dev93-20k | eval93-20k |
|---|---|---|
| LSTM | 7.4 | 6.8 |
| SRU | 7.5 | 6.8 |
| Quasi-RNN | 7.4 | 7.0 |
| RPPU | **6.9** | **6.2** |

alignment with such time points and then transforming it to the phoneme-level alignment. We compared it with both the ground-truth and the baseline alignment on phoneme-level to see how RPP works.

We define the similarity between two alignments by calculating the percentage of their overlaps in time. The similarity with the ground-truth alignment for the baseline and the estimated alignment using RPP are shown in Table 4. We can see that the similarity of estimated alignment achieves 9.2 % absolute gains. This demonstrates RPPU's capability in automatically aligning the acoustic inputs with the HMM state targets.

Apart from the quantitative analysis, we show one example using Praat (Boersma and others 2002) to better understand how RPPU works. This example is the partial alignment of the randomly chosen utterance "050c01017" within the duration of the first 0.47 seconds. As shown in Figure 3, the third tier, which corresponds to the estimated alignment, is aligned much better with the ground-truth alignment shown in the first tier than the baseline alignment in the second tier. Interestingly, it seems that the first boundary in the textgrids can be determined by the intensity in the yellow line, and that the right boundary of 'F' can be determined by the rising of the pitch. The estimated alignment fits better with the clean alignments in terms of those two boundaries. This might suggest that RPPU is capable of predicting the arrival of some acoustic events from some traits of the audio signal.

**Results on WSJ0** To evaluate how RPPU behaves in a clean condition, we applied our method to WSJ0 which contains the clean utterances from which the CHiME-2 corpus was derived. We used the same model configurations of CHiME-2 for all RNN models. From Table 5, we can observe that all three RNN baseline systems using Mel-filterbank features achieve a WER of 2.8%. These results are comparable to the prior work using DNN (Chen and Mak 2015). Our RPPU achieves the best WER of 2.5%, yielding 10.7% relative performance gain over the other RNN baseline systems.

**Results on WSJ0&1** We also conducted experiments on a larger corpus, WSJ0&1. The same model configurations of CHiME-2 were applied on all RNN models in WSJ0&1. The recognition results are shown in Table 6. We can see that our best baseline LSTM achieves WER of 7.4% and 6.8% and our RPPU gives the lowest WER of 6.9% and 6.2% on dev93-20k and eval93-20k test sets, respectively. Overall, the RPPU achieves 6.8% and 9.1% relative WER reductions over the best LSTM baseline system on the two test sets.

## Conclusion

We propose a novel model that can address hybrid acoustic modeling by incorporating the proposed recurrent Poisson process (RPP) into a recurrent neural network (RNN). We show that our model can generate much better alignments while performing the HMM state modeling. Our experiments on CHiME-2, WSJ0 and WSJ0&1 show that our method achieves much better results than several RNN baselines in ASR.

## Acknowledgements

The work was partially supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. HKUST16215816 ).

## References


Bacry, E.; Iuga, A.; Lasnier, M.; and Lehalle, C.-A. 2015. Market impacts and the life cycle of investors orders. *Market Microstructure and Liquidity* 1(02):1550009.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.

Biem, A.; Katagiri, S.; McDermott, E.; and Juang, B.-H. 2001. An application of discriminative feature extraction to filter-bank-based speech recognition. *IEEE Transactions on Speech and Audio Processing* 9(2):96–110.

Boersma, P., et al. 2002. Praat, a system for doing phonetics by computer. *Glot international* 5.

Bradbury, J.; Merity, S.; Xiong, C.; and Socher, R. 2016. Quasi-recurrent neural networks. *arXiv preprint arXiv:1611.01576*.

Chen, D., and Mak, B. 2015. Distinct triphone acoustic modeling using deep neural networks. In *interspeech*.

Consortium, L. D., et al. 1994. CSR-II (WSJ1) complete. *Linguistic Data Consortium, Philadelphia, vol. LDC94S13A*.

Dahl, G. E.; Yu, D.; Deng, L.; and Acero, A. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing* 20(1):30–42.

Du, N.; Wang, Y.; He, N.; Sun, J.; and Song, L. 2015. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems*, 3492–3500.

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *SIGKDD*, 1555–1564.

Esser, K.-H.; Condon, C. J.; Suga, N.; and Kanwal, J. S. 1997. Syntax processing by auditory cortical neurons in the FM–FM area of the mustached bat Pteronotus parnellii. *Proceedings of the National Academy of Sciences* 94(25):14019–14024.

Forney, G. D. 1973. The Viterbi algorithm. *Proceedings of the IEEE* 61(3):268–278.

Gales, M. J., et al. 1998. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer speech & language* 12(2):75–98.

Garofolo, J.; Graff, D.; Paul, D.; and Pallett, D. 1993. CSR-I (WSJ0) complete. *Linguistic Data Consortium, Philadelphia*.

Graves, A.; Jaitly, N.; and Mohamed, A.-r. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 273–278.

Graves, A.; r. Mohamed, A.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 6645–6649.

Hawkes, A. G. 1971a. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58(1):83–90.

Hawkes, A. G. 1971b. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58(1):83–90.

Jansen, A., and Niyogi, P. 2009. Point process models for event-based speech recognition. *Speech Communication* 51(12):1155–1168.

Kingman, J. F. C. 1992. *Poisson processes*, volume 3. Clarendon Press.

Klambauer, G.; Unterthiner, T.; Mayr, A.; and Hochreiter, S. 2017. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, 971–980.

Lei, T., and Zhang, Y. 2017. Training RNNs as fast as CNNs. *arXiv preprint arXiv:1709.02755*.

Malmgren, R. D.; Stouffer, D. B.; Motter, A. E.; and Amaral, L. A. 2008. A Poissonian explanation for heavy tails in e-mail communication. *Proceedings of the National Academy of Sciences* pnas–0800332105.

Mei, H., and Eisner, J. 2017. The neural Hawkes process: A neurally self-modulating multivariate point process. In *NIPS*, 6757–6767.

Ogata, Y. 1981. On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory* 27(1):23–31.

Povey, D., et al. 2011. The Kaldi speech recognition toolkit. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*.

Snoek, J.; Zemel, R. S.; and Adams, R. P. 2013. A determinantal point process latent variable model for inhibition in neural spiking data. In *NIPS*, 1932–1940.

Stevens, K. N. 2000. *Acoustic phonetics*, volume 30. MIT press.

Stevens, K. N. 2002. Toward a model for lexical access based on acoustic landmarks and distinctive features. *The Journal of the Acoustical Society of America* 111(4):1872–1891.

Teh, Y. W., and Rao, V. 2011. Gaussian process modulated renewal processes. In *Advances in Neural Information Processing Systems*, 2474–2482.

Vincent, E.; Barker, J.; Watanabe, S.; Le Roux, J.; Nesta, F.; and Matassoni, M. 2013. The second CHiME speech separation and recognition challenge: Datasets, tasks and baselines. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 126–130.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Werbos, P. J. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 78(10):1550–1560.

Yu, D., and Li, J. 2018. Recent progresses in deep learning based acoustic models (updated). *arXiv preprint arXiv:1804.09298*.

Zhou, K.; Zha, H.; and Song, L. 2013. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*, 641–649.