A Bayesian interpretation of the light gated recurrent unit

Alexandre Bittar [1] Philip N. Garner [1]

[1]Idiap Research Institute, Martigny, Switzerland

Abstract

We summarise previous work showing that the basic sigmoid activation function arises as an instance of Bayes's theorem, and that recurrence follows from the prior. We derive a layer-wise recurrence without the assumptions of previous work, and show that it leads to a standard recurrence with modest modifications to reflect use of log-probabilities. The resulting architecture closely resembles the Li-GRU which is the current state of the art for ASR. Although the contribution is mainly theoretical, we show that it is able to outperform the state of the art on the TIMIT and AMI datasets.

Context and motivation

In signal processing, modelling of context can be achieved via recurrence. The most successful architectures are based on the long short-term memory (LSTM), which uses a memory cell and different gates to filter out irrelevant information. Recent efforts have been pursued to reduce the size of recurrent units and avoid redundancies. This notably lead to the gated recurrent unit (GRU) [1] in 2014 and light GRU [3] in 2018. Following the recent work of Garner and Tong [2], our motivation is to provide a probabilistic interpretation of recurrent units and shed some light on the seemingly ad-hoc concepts of gates and memory cells.

Bayesian interpretation of recurrence

Consider an input sequence $\boldsymbol{X}_T = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_T] \in \mathbb{R}^{F \times T}$ of length T, where each observation $\boldsymbol{x}_t$ is a vector of size F. We assume that there are H hidden features $\{\phi_i \,|i = 1, \dots, H\}$ that we wish to detect along the sequence. At each timestep t, a feature has two possible states: present or absent, that we write as $\phi_{t,i}$ and $\neg\phi_{t,i}$ respectively. We want to build a layer of H recurrent units that will output the stacked probabilities $\boldsymbol{h}_t := P(\boldsymbol{\phi}_t|\boldsymbol{X}_t) \in [0,1]^H$ of the different features being present at each timestep $t = 1, \dots, T$.

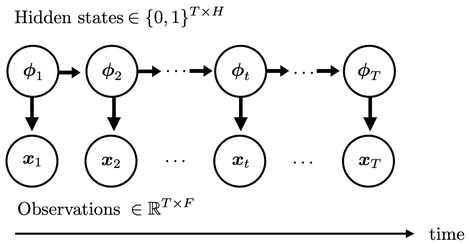

Figure 1. Diagram of the approach

Bayes's theorem

For the two-class case, Bayes's formula can be rewritten in a sigmoid form for the whole layer,

$$\boldsymbol{h}_t = \sigma\Big[\log(\boldsymbol{r}_t) + \text{logit}(\boldsymbol{p}_t)\Big]\,. \tag{1}$$

The posterior $\boldsymbol{h}_t$, representing the desired unit outputs, is expressed as a function of the ratio of likelihood $\boldsymbol{r}_t$ and prior $\boldsymbol{p}_t$, defined as

$$\boldsymbol{r}_t := \frac{p(\boldsymbol{x}_t|\boldsymbol{\phi}_t)}{p(\boldsymbol{x}_t|\neg\boldsymbol{\phi}_t)} \quad \text{and} \quad \boldsymbol{p}_t := P(\boldsymbol{\phi}_t|\boldsymbol{X}_{t-1})\,. \tag{2}$$

Ratio of likelihood and Prior

If we assume that the likelihood of observing $\boldsymbol{x}_t$ given the current state of the features $\boldsymbol{\phi}_t$ can be represented with multivariate normal distributions, the ratio of likelihood can be expressed as

$$\boldsymbol{r}_t = \exp\Big[\boldsymbol{W}_r^T\,\boldsymbol{x}_t + \boldsymbol{b}_r\Big]\,. \tag{3}$$

Assuming that the H feature probabilities $h_{t-1,i}$ of the layer are independently beta distributed, they can be combined into a single prior probability,

$$\boldsymbol{p}_t = \sigma\Big[\boldsymbol{V}_p \log(\boldsymbol{h}_{t-1}) + \boldsymbol{b}_p\Big]\,. \tag{4}$$

Resulting forward pass

By plugging equations (3) and (4) into (1), we get the following update equation,

$$\boldsymbol{h}_t = \sigma\Big[\boldsymbol{W}_h\boldsymbol{x}_t + \boldsymbol{V}_h \log(\boldsymbol{h}_{t-1}) + \boldsymbol{b}_h\Big]\,, \tag{5}$$

or in log-probabilities form, i.e. $\boldsymbol{h}_t := \log\big[P(\boldsymbol{\phi}_t|\boldsymbol{X}_t)\big]$,

$$\boldsymbol{h}_t = \text{softplus}\Big[\boldsymbol{W}_h\,x_t + \boldsymbol{V}_h\,\boldsymbol{h}_{t-1} + \boldsymbol{b}_h\Big]\,, \tag{6}$$

where $\boldsymbol{W}_h$, $\boldsymbol{V}_h$ and $\boldsymbol{b}_h$ are representative of the distributions of $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$ when the features are present or absent, and can be treated as trainable parameters of the model.

Adding a context relevance gate

We define a binary state variable $\rho_{t,i}$, which indicates if $\boldsymbol{x}_t$ is relevant for the occurrence of $\phi_{t,i}$. The associated probabilities $z_{t,i} = P(\rho_{t,i}|\boldsymbol{X}_t)$ can be computed with equation (5). The desired output probabilities $h_{t,i}$ can then be expressed by marginalizing over $\rho_{t,i}$ so that we get the forward pass of a Li-GRU with minor modifications,

$$\boldsymbol{z}_t = \sigma\Big[\boldsymbol{W}_z\,\boldsymbol{x}_t + \boldsymbol{V}_z\,\log(\boldsymbol{h}_{t-1}) + \boldsymbol{b}_z\Big] \tag{7a}$$

$$\tilde{\boldsymbol{h}}_t = \sigma\Big[\boldsymbol{W}_h\,\boldsymbol{x}_t + \boldsymbol{V}_h\,\log(\boldsymbol{h}_{t-1}) + \boldsymbol{b}_h\Big] \tag{7b}$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t * \tilde{\boldsymbol{h}}_t + (1 - \boldsymbol{z}_t) * \boldsymbol{h}_{t-1}\,. \tag{7c}$$

Additionally, the input of the i-th layer corresponds to the log-probabilities of the previous layer $\boldsymbol{x}_t^{[i]} = \log\big(\boldsymbol{h}_t^{[i-1]}\big)$.

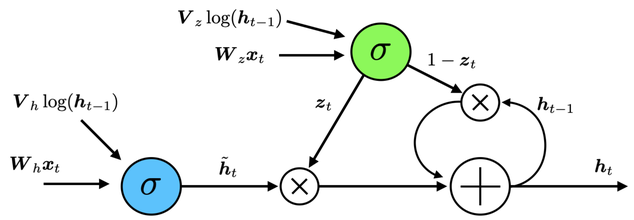

Figure 2. Diagram of the Li-BRU defined by equations (7)

Comparison with Li-GRU

The forward pass of a Li-GRU is the following,

$$\boldsymbol{z}_t = \sigma\Big[\boldsymbol{W}_z\,\boldsymbol{x}_t + \boldsymbol{V}_z\,\boldsymbol{h}_{t-1} + \boldsymbol{b}_z\Big] \tag{8a}$$

$$\tilde{\boldsymbol{h}}_t = \text{ReLU}\Big[\boldsymbol{W}_h\,\boldsymbol{x}_t + \boldsymbol{V}_h\,\boldsymbol{h}_{t-1} + \boldsymbol{b}_h\Big] \tag{8b}$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t * \tilde{\boldsymbol{h}}_t + (1 - \boldsymbol{z}_t) * \boldsymbol{h}_{t-1}\,. \tag{8c}$$

The differences with our unit are

- the ReLU activation, which is an approximation of the softplus
- the gate applied to log-probabilities.

References

[1] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio.
Learning phrase representations using RNN encoder- decoder for statistical machine translation.
In Proceedings of the 2014 EMNLP Conference, page 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[2] P. N. Garner and S. Tong.
A Bayesian approach to recurrence in neural networks.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[3] M. Ravanelli, A. Bordes, and Y. Bengio.
Light gated recurrent units for speech recognition.
Transactions on Emerging Topics in Computational Intelligence, 2(2):92–102, 2018.

Experiments

We perform a self-consistent comparison between recurrent units on 2 speech recognition tasks, using the TIMIT and AMI corpora. Following the pytorch-kaldi implementation of [3], all presented experiments use a recurrent architecture with 4 layers of H=550 bidirectional units. The F=50 fMLLR input features are extracted via the Kaldi recipe.

Without the update gate

We first test the simple BRU from equation (6) on TIMIT and compare it to conventional units of the same size. Units that have a feedback on log-probabilities perform significantly better.

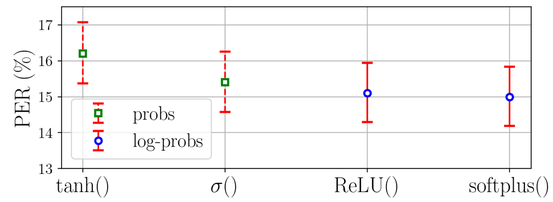

Figure 3. Error-rates on TIMIT testset for various RNN architectures.

With the update gate

We now test the complete Li-BRU from equations (7) to state of the art recurrent units.

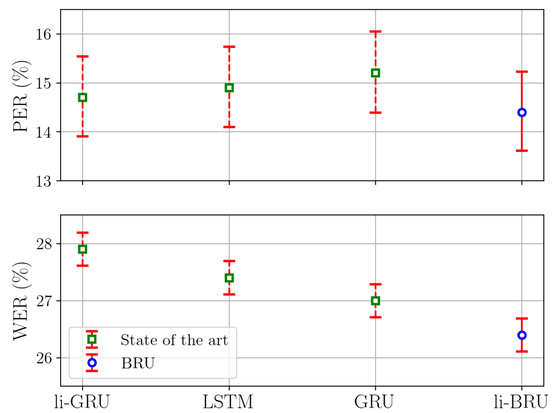

Figure 4. Error-rates on TIMIT (top) and AMI (bottom) for various RNN architectures.

The Li-BRU outperforms all other state of the art recurrent architectures on both TIMIT and AMI datasets with error-rates of 14.4 %, and 26.4% respectively. We found that applying the gate to probabilities or log-probabilities is practically equivalent. The root of improvement of the Li-BRU over the Li-GRU therefore seems to lie in the use of the softplus function instead of its approximation by the ReLU.

Conclusion

In previous work, it was shown that a Bayesian analysis of a sigmoid activation function led naturally to a unit-wise recurrence and, with approximations, to a layer-wise recurrence. In this paper, in a mainly theoretical contribution, we have shown that beta-distributed sigmoid outputs feeding into another sigmoid unit constitute a layer-wise recurrence without approximations. Without a forget gate, this reduces to a standard fully-connected recurrence, but with a softplus activation. Given that the update gate of a GRU can also be derived probabilistically, the approach led naturally to comparison with a Li-GRU. In an experimental evaluation, we confirmed that the resulting light Bayesian recurrent unit (Li-BRU) can modestly but significantly outperform the state of the art on two ASR tasks (TIMIT and AMI datasets), demonstrating the importance of the probabilistic derivation. More generally, the new techniques contribute to a growing toolkit of Bayesian approaches for neural architectures.