A BAYESIAN INTERPRETATION OF THE LIGHT GATED RECURRENT UNIT

Alexandre Bittar, Philip N. Garner

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 layerwise 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.

Index Terms— speech recognition, deep learning, recurrent neural networks, Bayesian inference, Li-GRU

1. INTRODUCTION

In the wider deep-learning field, modelling of context is important. A current general trend is to use convolutional architectures, analogous to finite impulse response filters. Nevertheless, in signal processing, it is natural to prefer recurrence for processes that are understood to be autoregressive, the analogy being to infinite impulse response filters. A pertinent example is in automatic speech recognition (ASR) where the state of the art involves recurrent architectures.

The most successful architectures are based on the long short-term memory (LSTM), defined by Hochreiter and Schmidhuber in 1997 [1], with a memory cell and input and output gates to filter out irrelevant information and tackle the vanishing/exploding gradient problem. An additional forget gate and peephole connections were subsequently added by Gers et al. [2], [3]. A simplification of the unit resulted into the gated recurrent unit (GRU) of Cho et al. [4] in 2014, where the input and forget gate of the LSTM are combined into a single update gate, and the output gate is now called a reset gate and acts on the feedback inside the cell state. Further efforts to reduce the size of recurrent units were pursued by Zhou et al. [5] in 2016 with the minimally gated unit (MGU), where a single gate is used twice as the update and reset gates of the GRU. In the same spirit of getting rid of redundancies, Ravanelli et al. [6], [7] recently proposed an alternative simplification of the GRU called light GRU (Li-GRU) by removing the reset gate altogether. The Li-GRU outperformed the GRU and LSTM in different fields, notably on ASR tasks, where it represents the current state of the art.

More generally, multi-layer perceptrons (MLPs) have been shown to have probabilistic interpretations. Very recently, Garner and Tong [8] have been able to derive a recurrent unit architecture similar to the GRU using a Bayesian approach. Their work shed some light on the seemingly ad-hoc concepts of gates and memory cells inside the commonly used recurrent units. The input is treated as a sequence of observations, and the unit outputs are interpreted as the probabilities of hidden features being present at each timestep. Recurrence emerges naturally from Bayes's theorem which updates a prior probability into a posterior given new observational data.

The present paper stems from our attempts to remove some of the approximations in [8], in particular regarding the layerwise feedback. We show that

1. A probabilistic layerwise feedback can be introduced via a sigmoid unit. Without a forget mechanism, it reduces to the common fully-connected approach.

2. The natural feedback domain is log-probability, leading naturally to the softplus activation.

The resulting architecture is very close to the Li-GRU described above, but with a valid probabilistic formulation. We hence add an update gate, yielding a *light Bayesian recurrent unit* (Li-BRU) which forms a basis for evaluation in terms of architecture and number of trainable parameters. Whilst we intend the main contribution to be theoretical, augmenting the evolving toolkit of Bayesian components in deep learning, a modest evaluation shows that the resulting collection of modifications outperforms the Li-GRU (as well as the GRU and LSTM) on ASR tasks.

2. BAYESIAN INTERPRETATION OF RECURRENCE

Consider an input sequence $\boldsymbol{X}_T = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T] \in \mathbb{R}^{F \times T}$ of length T, where each observation $\boldsymbol{x}_t$ is a vector with F input dimensions. We assume that there are H hidden features $\{\phi_i \,|i = 1, \ldots, H\}$ 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. Using a Bayesian approach, 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 $t = 1, \ldots, T$. Let us start with the Bayesian update formula

$$P(\phi_{t,i}|\boldsymbol{X}_t) = \frac{p(\boldsymbol{x}_t|\phi_{t,i})P(\phi_{t,i}|\boldsymbol{X}_{t-1})}{\sum_{\phi'_{t,i}} p(\boldsymbol{x}_t|\phi'_{t,i})P(\phi'_{t,i}|\boldsymbol{X}_{t-1})}\,, \tag{1}$$

that we can rewrite as,

$$h_{t,i} = \frac{1}{1 + \dfrac{p(\boldsymbol{x}_t|\neg\phi_{t,i})}{p(\boldsymbol{x}_t|\phi_{t,i})} \cdot \dfrac{P(\neg\phi_{t,i}|\boldsymbol{X}_{t-1})}{P(\phi_{t,i}|\boldsymbol{X}_{t-1})}} \tag{2}$$

for the two-class case, where the posterior representing the desired unit output $h_{t,i}$ is expressed as a function of the ratio of likelihood $r_{t,i}$ and prior $p_{t,i}$, that we define in vectorized form for the whole layer as

$$\boldsymbol{r}_t := \frac{p(\boldsymbol{x}_t|\boldsymbol{\phi}_t)}{p(\boldsymbol{x}_t|\neg\boldsymbol{\phi}_t)} \quad \text{and} \quad \boldsymbol{p}_t := P(\boldsymbol{\phi}_t|\boldsymbol{X}_{t-1})\,. \tag{3}$$

As pointed out by Bridle [9] and more recently reiterated by Garner and Tong [8], Bayes's theorem has an explicit relationship with the sigmoid function $\sigma(x) = 1/(1+e^{-x})$, so that equation (2) can be rewritten as,

$$\boldsymbol{h}_t = \sigma\Big[\,\log(\boldsymbol{r}_t) + \text{logit}(\boldsymbol{p}_t)\Big]\,, \tag{4}$$

where $\text{logit}(x) = \log\big[x/(1-x)\big]$. As demonstrated in [8], 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 that share the same covariance matrix $\boldsymbol{\Sigma}$, i.e. $p(\boldsymbol{x}_t|\boldsymbol{\phi}_t) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $p(\boldsymbol{x}_t|\,\neg\boldsymbol{\phi}_t\,) \sim \mathcal{N}(\boldsymbol{\nu}, \boldsymbol{\Sigma})$, then the ratio of likelihood can be expressed as

$$\boldsymbol{r}_t = \exp\Big[\boldsymbol{W}_r^T\,\boldsymbol{x}_t + \boldsymbol{b}_r\Big]\,. \tag{5}$$

where $\boldsymbol{W}_r \in \mathbb{R}^{F\times H}$ and $\boldsymbol{b}_r \in \mathbb{R}^H$ are defined as,

$$\boldsymbol{W}_r = \Big(\boldsymbol{\nu}^T - \boldsymbol{\mu}^T\Big)\,\boldsymbol{\Sigma}^{-1} \tag{6a}$$

$$\boldsymbol{b}_r = -\frac{1}{2}\Big(\boldsymbol{\nu}^T\,\boldsymbol{\Sigma}^{-1}\,\boldsymbol{\nu} + \boldsymbol{\mu}^T\,\boldsymbol{\Sigma}^{-1}\,\boldsymbol{\mu}\Big)\,, \tag{6b}$$

In the next subsection, we will derive a novel way of estimating the prior $\boldsymbol{p}_t$, leading to a layer-wise recurrence without any approximation.

2.1. Prior

In the simplest case, where the features are time-independent, i.e. a feature is either present in the entirety of the sequence or not there at all, the prior probability is simply given by the one from the previous timestep $P(\phi_{t,i}|\boldsymbol{X}_{t-1}) = P(\phi_{t-1,i}|\boldsymbol{X}_{t-1}) = h_{t-1,i}$.

Now let us assume that the features can occur and vanish arbitrarily throughout the sequence. We can first assume they are independent, which results in a unit-wise feedback, where the prior probability $P(\phi_{t,i}|\boldsymbol{X}_{t-1})$ only depends on $P(\phi_{t-1,i}|\boldsymbol{X}_{t-1}) = h_{t-1,i}$.

In the more realistic scenario, where we further assume that there is an interdependence between the different features, i.e. that some will more naturally occur together than others, the prior now needs to depend on all $P(\phi_{t-1,j}|\boldsymbol{X}_{t-1})$ with $j = 1, \ldots, H$. Here we need to combine all H feature probabilities $h_{t-1,j}$ of the layer into a single prior probability. Dropping the $t-1$ in the index for simplicity, the h_i are between 0 and 1; it is reasonable to assume that they are each independently beta distributed:

$$\begin{aligned} p(h_i|\alpha_i, \beta_i) &= \frac{1}{B(\alpha_i, \beta_i)} h_i^{\alpha_i - 1}(1-h_i)^{\beta_i - 1} \\ &= \exp\Big[\,-\log B + (\alpha_i - 1)\log(h_i) \\ &\qquad + (\beta_i - 1)\log(1-h_i)\Big]\,. \end{aligned} \tag{7}$$

The joint distribution is then,

$$p(\boldsymbol{h}|\boldsymbol{\alpha}) = \prod_{i=1}^{H} p(h_i|\alpha_i, \beta_i)\,, \tag{8}$$

where $\boldsymbol{\alpha} = \big[\alpha_1, \ldots, \alpha_H, \beta_1, \ldots, \beta_H\big]^T$.

We define one set of parameters $\boldsymbol{\alpha}_1$ that represents the beta-distribution when the features are present at the next timestep: $p(\boldsymbol{h}_{t-1}|\boldsymbol{\alpha}_1) = p(\boldsymbol{h}_{t-1}|\boldsymbol{\phi}_t)$, and a second one, $\boldsymbol{\alpha}_2$ that corresponds to the distribution when they are absent: $p(\boldsymbol{h}_{t-1}|\boldsymbol{\alpha}_2) = p(\boldsymbol{h}_{t-1}|\neg\boldsymbol{\phi}_t)$. We then write

$$\begin{aligned} P(\boldsymbol{\phi}_t|\boldsymbol{h}_{t-1}) &= \frac{p(\boldsymbol{h}_{t-1}|\boldsymbol{\phi}_t)P(\boldsymbol{\phi}_t)}{P(\boldsymbol{h}_{t-1})} \\ &= \frac{1}{1 + \dfrac{p(\boldsymbol{h}_{t-1}|\neg\boldsymbol{\phi}_t)}{p(\boldsymbol{h}_{t-1}|\boldsymbol{\phi}_t)}\,\dfrac{1 - P(\boldsymbol{\phi}_t)}{P(\boldsymbol{\phi}_t)}} \end{aligned} \tag{9}$$

Using equations (7) and (8), the ratio of likelihood in the denominator of equation (9) can be computed as,

$$\frac{p(\boldsymbol{h}_{t-1}|\boldsymbol{\alpha}_2)}{p(\boldsymbol{h}_{t-1}|\boldsymbol{\alpha}_1)} = \exp\Big[\,-\boldsymbol{V}_p\log(\boldsymbol{h}_{t-1}) - \boldsymbol{b}_p\Big] \tag{10}$$

where $\boldsymbol{V}_p := \boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2$ and $\boldsymbol{b}_p := \log B(\boldsymbol{\alpha}_2) - \log B(\boldsymbol{\alpha}_1)$. Notice that we have ignored the $\log(1-h_i)$ terms. To do this, we need to make the assumption that all the β_i are equal to 1, or that they are the same for each class (feature presence). The second of these is similar to the identical covariance in the Gaussian assumption case for the ratio of likelihood.

The prior $P(\boldsymbol{\phi}_t)$ in equation (9) represents the probability of having the feature present at time t even before seeing the previous observation $\boldsymbol{x}_{t-1}$ or knowing $\boldsymbol{h}_{t-1}$. This can be

assumed to be some unconditional prior $P(\boldsymbol{\phi}_0)$. By putting equation (9) in sigmoid form like we did with equation (4), the constant prior term $\text{logit}\big[P(\boldsymbol{\phi}_0)\big]$ can be integrated inside $\boldsymbol{b}_p$, and we get this final expression for the layer-wise prior

$$\boldsymbol{p}_t = \sigma\Big[\boldsymbol{V}_p \log(\boldsymbol{h}_{t-1}) + \boldsymbol{b}_p\Big]\,, \tag{11}$$

where we assumed that $P(\boldsymbol{\phi}_t|\boldsymbol{X}_{t-1}) = P(\boldsymbol{\phi}_t|\boldsymbol{h}_{t-1})$.

2.2. Resulting BRU

With equations (5) and (11), we now have a probabilistically plausible way of computing the ratio of likelihood $\boldsymbol{r}_t$ and prior $\boldsymbol{p}_t$. By plugging them into equation (4), we can use the relationship between the sigmoid and logit functions,

$$\text{logit}\big[\sigma(x)\big] = x \tag{12}$$

to get the following update equation,

$$\boldsymbol{h}_t = \sigma\Big[\boldsymbol{W}_h \boldsymbol{x}_t + \boldsymbol{V}_h \log(\boldsymbol{h}_{t-1}) + \boldsymbol{b}_h\Big]\,, \tag{13}$$

where we redefined the parameters as $\boldsymbol{W}_h = \boldsymbol{W}_r$, $\boldsymbol{V}_h = \boldsymbol{V}_p$ and $\boldsymbol{b}_h = \boldsymbol{b}_r + \boldsymbol{b}_p$. These parameters 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.

If we instead choose our units to output log-probabilities, i.e. $\boldsymbol{h}_t := \log\big[P(\boldsymbol{\phi}_t|\boldsymbol{X}_t)\big]$, this actually corresponds to a softplus activation function, $\text{softplus}(x) = -\log\big[\sigma(-x)\big]$, as described by Dugas et al. [10], where the sign differences can be integrated inside the trainable parameters and we write the resulting forward pass as,

$$\boldsymbol{h}_t = \text{softplus}\Big[\boldsymbol{W}_h\, x_t + \boldsymbol{V}_h\, \boldsymbol{h}_{t-1} + \boldsymbol{b}_h\Big]\,. \tag{14}$$

3. COMPARISON WITH RNN

The equation (13) resulting from this Bayesian approach resembles the following forward pass of a standard RNN unit,

$$\boldsymbol{h}_t = \tanh\Big[\boldsymbol{W}_h\, x_t + \boldsymbol{V}_h\, \boldsymbol{h}_{t-1} + \boldsymbol{b}_h\Big]\,, \tag{15}$$

which is also the basis of LSTMs and GRUs. In these standard recurrent units, the feedback is not taken on the logarithm of probabilities but on $\boldsymbol{h}_{t-1}$ directly. The activation function is a hyperbolic tangent, which is a rescaled version of a sigmoid with $\tanh(x) = 2\sigma(2x) - 1$. If $\sigma(x)$ describes a probability in [0,1], then $\tanh(x)$ is simply a rescaled representation of that probability in the [-1,1] range. The main difference is therefore on the absence of the log in the feedback, but since here $\boldsymbol{h}_{t-1} \in [-1, 1]^H$ it does not make sense to take its logarithm, as $\log(x \leq 0)$ is not defined.

Coming back to equation (14) with the softplus activation, we notice that there is no log in the feedback as $\boldsymbol{h}_t$ already describes log-probabilities. As shown by Glorot et al. [11], the rectified linear unit function is a linear approximation of the softplus. Using the ReLU activation turns out to be exactly the forward pass of a light GRU [7] without the update gate,

$$\boldsymbol{h}_t \approx \text{ReLU}\Big[\boldsymbol{W}_h\, \boldsymbol{x}_t + \boldsymbol{V}_h\, \boldsymbol{h}_{t-1} + \boldsymbol{b}_h\Big]\,. \tag{16}$$

In the following section, we derive an alternative to the Li-GRU by adding an update gate through a Bayesian approach.

4. DEFINING THE LI-BRU

Let us start with the BRU described by equation (13). In a slight simplification of the approach of [8], let us define a binary state variable $\rho_{t,i}$ that is indicative of the relevance of the current observation $\boldsymbol{x}_t$ for the occurrence of the i-th hidden feature. The associated probabilities $z_{t,i} = P(\rho_{t,i}|\boldsymbol{X}_t)$ can be computed as a layer of BRUs with equation (13),

$$\boldsymbol{z}_t = \sigma\Big[\boldsymbol{W}_z\, \boldsymbol{x}_t + \boldsymbol{V}_z\, \log(\boldsymbol{h}_{t-1}) + \boldsymbol{b}_z\Big]\,. \tag{17}$$

We chose the recurrence to be on $\boldsymbol{h}_{t-1}$ instead of $\boldsymbol{z}_{t-1}$, simply because we observed better performance. Both are valid choices as they represent probabilities and can be considered to be beta-distributed (see subsection 2.1).

The idea is to apply $\boldsymbol{z}_t$ as a gate on probabilities, which is why we choose equation (13) and not (14), that describes log-probabilities instead. The desired output probability $h_{t,i}$ can then be expressed by marginalizing as,

$$\begin{aligned} h_{t,i} &= P(\phi_{t,i}|\boldsymbol{X}_t) \\ &= \textstyle\sum_{\rho'_{t,i}} P(\phi_{t,i}|\boldsymbol{X}_t, \rho'_{t,i})\, p(\rho'_{t,i}|\boldsymbol{X}_t) \\ &= (1 - z_{t,i}) P(\phi_{t,i}|\boldsymbol{X}_{t-1}) + z_{t,i}\, P(\phi_{t,i}|\boldsymbol{X}_t) \\ &= (1 - z_{t,i})\, h_{t-1,i} + z_{t,i}\, h_{t,i}\,, \end{aligned} \tag{18}$$

which represents taking an arithmetic weighted mean of two probabilities p_1, p_2 as $p = z_t \cdot p_1 + (1 - z_t) \cdot p_2$. When context is not relevant, we write $P(\phi_{t,i}|\boldsymbol{X}_t, \neg\rho_{t,i}) = P(\phi_{t,i}|\boldsymbol{X}_{t-1})$.

In a Li-GRU, due to the ReLU activation, the update gate acts on log-probabilities and thus corresponds to taking a geometric weighted mean of two probabilities: $p = p_1^{z_t} \cdot p_2^{1-z_t}$, since $\log(p) = z_t \cdot \log(p_1) + (1 - z_t) \cdot \log(p_2)$. The proper approach would be to first exponentiate the log-probabilities before applying the gate. In practice, we found no significant difference in doing so, suggesting that taking the geometric mean of the probabilities is an appropriate approximation.

We can now define the Bayesian equivalent of a Li-GRU, that we call Li-BRU, where the z-gate acts on probabilities,

$$\boldsymbol{z}_t = \sigma\Big[\boldsymbol{W}_z\, \boldsymbol{x}_t + \boldsymbol{V}_z\, \log(\boldsymbol{h}_{t-1}) + \boldsymbol{b}_z\Big] \tag{19a}$$

$$\tilde{\boldsymbol{h}}_t = \sigma\Big[\boldsymbol{W}_h\, \boldsymbol{x}_t + \boldsymbol{V}_h\, \log(\boldsymbol{h}_{t-1}) + \boldsymbol{b}_h\Big] \tag{19b}$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t * \tilde{\boldsymbol{h}}_t + (1 - \boldsymbol{z}_t) * \boldsymbol{h}_{t-1}\,. \tag{19c}$$

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

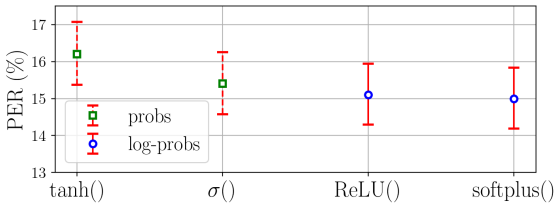

Fig. 1. PER on TIMIT testset for various RNN architectures.

5. EXPERIMENTS

Following the pytorch-kaldi implementation of [7] and [12], 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 [13] recipe. Experiments on the TIMIT [14] and AMI [15] corpora are performed with Adam [16] and RMSprop [17] optimizers respectively, both during 24 epochs with drop-out regularisation (p=0.2). Batch-normalization [18] is used on feed-forward connections, as suggested in [7]. Our aim is not to surpass the best reported phone error rate (PER) on TIMIT or AMI (8.30% [19] and 17.84% [20] respectively), but to perform a self-consistent comparison between recurrent units.

5.1. Without the update gate

We first test the simple BRU resulting from section 2 on the TIMIT dataset and compare it to Vanilla RNN units (see Figure 1). We make a distinction between the units that have a feedback on (rescaled) probabilities, like the standard RNN of equation (15), and the ones that use log-probabilities (as justified in subsection 2.1). Note that all units have the same number of trainable parameters (i.e. F+H+1 per unit). We make the hypothesis that the units with a feedback on log-probabilities perform better than the ones with probabilities. The error-bars on Figure 1 show the 95% equal-tailed confidence interval for a beta-assumption for the error-rate. As they are relatively large on TIMIT due to the small size of the test set (7215 utterances versus 90002 for AMI), we perform a matched-pairs test, as described by Gillick and Cox [21], on the different speakers and obtain a p-value of $4.96 \cdot 10^{-5}$. Alternatively using a Wilcoxon signed-rank test [22] gives us a p-value of $5.09 \cdot 10^{-4}$. We consider both results to be small enough to validate our hypothesis.

5.2. Testing the complete Li-BRU

Let us now add the update gate and compare the resulting Li-BRU from equations (19) to state of the art recurrent units. As illustrated in Figure 2, 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 also tested the Li-GRU architecture with a softplus activation instead of a ReLU. As mentioned at the end of section 4, the unit gave the same results as the Li-BRU, both on TIMIT and AMI, signifying that applying the gate to probabilities or log-probabilities is practically equivalent. The root of the improvement therefore seems to lie in the choice of the activation function. The importance of using the softplus function instead of its approximation by the ReLU is especially visible on AMI.

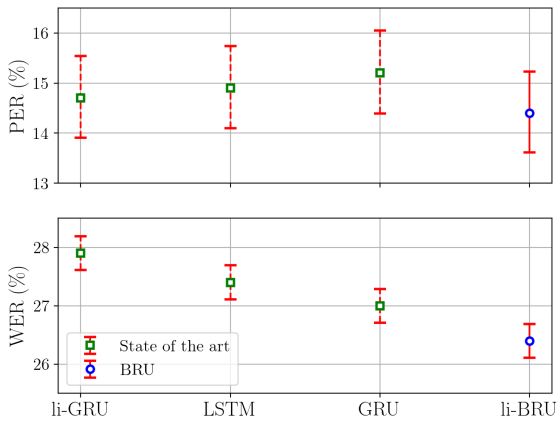

Fig. 2. PER on TIMIT (top) and WER on AMI (bottom) for various RNN architectures.

6. 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.

7. ACKNOWLEDGEMENTS

This project received funding under NAST: Neural Architectures for Speech Technology, Swiss National Science Foundation grant 200021_185010.

8. REFERENCES

[1] S. Hochreither and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. DOI: 10.1162/neco.1997.9.8.1735.

[2] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, pp. 2451–2471, 2000. DOI: 10.1049/cp:19991218.

[3] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, Oct. 2002.

[4] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Begio, "Learning phrase representations using RNN encoder- decoder for statistical machine translation," in *Proceedings of the 2014 EMNLP Conference*, Association for Computational Linguistics, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[5] G.-B. Zhou, J. Wu, C.-L. Zhang, and Z.-H. Zhou, "Minimal gated unit for recurrent neural networks," *International Journal of Automation and Computing*, vol. 13, no. 3, pp. 226–234, Jun. 2016. DOI: 10.1007/s11633-016-1006-2.

[6] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Improving speech recognition by revising gated recurrent units," in *Proceedings of Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 1308–1312. DOI: 10.21437/Interspeech.2017-775.

[7] M. Ravanelli, A. Bordes, and Y. Bengio, "Light gated recurrent units for speech recognition," *Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, 2018. DOI: 10.1109/TETCI.2017.2762739.

[8] P. N. Garner and S. Tong, "A Bayesian approach to recurrence in neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. DOI: 10.1109/TPAMI.2020.2976978. Early Access; preprint available as arXiv: 1910.11247.

[9] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*, ser. NATO ASI Series F: Computer and Systems Sciences, F. Fogelman Soulié and J. Hérault, Eds., vol. 68, Berlin Heidelberg: Springer-Verlag, 1990, pp. 227–236. DOI: 10.1007/978-3-642-76153-9_28.

[10] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, "Incorporating second-order functional knowledge for better option pricing," in *Advances in Neural Information Processing Systems*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., vol. 13, MIT Press, 2001, pp. 472–478.

[11] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, 2011, pp. 315–323.

[12] M. Ravanelli, T. Parcollet, and Y. Begio, "The pytorch-kaldi speech recognition toolkit," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, UK, May 2019, pp. 6465–6469. DOI: 10.1109/ICASSP.2019.8683713.

[13] D. Povey, A. Ghosal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Hawaii, USA, Dec. 2011, pp. 1–4.

[14] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," NIST, Gaithersburg, MD, USA, NISTIR 4930, Feb. 1993.

[15] J. Carletta, S. Ashby, S. Bourban, M. Guillemot, M. Kronenthal, G. Lathoud, M. Lincoln, I. McCowan, T. Hain, W. Kraaij, W. Post, J. Kadlec, P. Wellner, M. Flynn, and D. Reidsma, "The AMI meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, Edinburgh, 2005, p. 100.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations*, San Diego, Dec. 2015. arXiv: 1412.6980.

[17] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, 2012.

[18] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, Feb. 2015. arXiv: 1502.03167.

[19] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," Jul. 2020. arXiv: 2006.11477.

[20] N. Kanda, Y. Fujita, and K. Nagamatsu, "Lattice-free state-level minimum bayes risk training of acoustic models," in *Proceedings of Interspeech*, Hyderabad, India, Sep. 2018, pp. 2923–2927. DOI: 10.21437/Interspeech.2018-79.

[21] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Glasgow, UK, May 1989, pp. 532–535. DOI: 10.1109/ICASSP.1989.266481.

[22] F. Wilcoxon, "Individual comparisons by ranking methods," in *Biometrics Bulletin*, vol. 1, Dec. 1945, pp. 80–83. DOI: 10.1007/978-1-4612-4380-9_16.