In: Proc. of the 26th Int. Conference on Artificial Neural Networks (ICANN), Alghero, Italy, September 11-14, 2017
The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-68600-4_1

Semi-Supervised Phoneme Recognition with Recurrent Ladder Networks

Marian Tietz, Tayfun Alpay, Johannes Twiefel, and Stefan Wermter

Knowledge Technology Institute, Department of Informatics, Universität Hamburg,
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany
{tietz,alpay,twiefel,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM/

Abstract. Ladder networks are a notable new concept in the field of semi-supervised learning by showing state-of-the-art results in image recognition tasks while being compatible with many existing neural architectures. We present the recurrent ladder network, a novel modification of the ladder network, for semi-supervised learning of recurrent neural networks which we evaluate with a phoneme recognition task on the TIMIT corpus. Our results show that the model is able to consistently outperform the baseline and achieve fully-supervised baseline performance with only 75% of all labels which demonstrates that the model is capable of using unsupervised data as an effective regulariser.

Keywords: semi-supervised learning, recurrent neural networks, ladder networks, phoneme recognition

1 Introduction

There is no doubt that the recent success of deep learning is tied to the rising availability of labelled data. While tasks such as image or text classification have greatly benefited from this availability, there are still a number of domains, e.g. speech recognition, where the majority of the research community has no free access to large amounts of labelled data. One promising approach towards this problem is semi-supervised learning where models trained with *labelled* data can be further improved by training with *unlabelled* data.

Recent methods, such as graph-supported training [10], sparse autoencoders ([4]; SSSAE) and especially the Ladder Network (LN) [11], a stacked Denoising Autoencoder (DAE) with shortcut connections, show promising results for semi-supervised training of feed-forward neural networks. The LN has been shown to deliver state-of-the-art results in semi-supervised image classification while still being compatible with many existing feed-forward neural networks [11].

However, this novel architecture has not yet been explored on more complex sequential tasks, such as speech recognition, where Recurrent Neural Network (RNN) architectures, like Gated Recurrent Units (GRU; [1]), are the current state of the art. We therefore propose a novel Recurrent Ladder Network

(RLN) architecture and evaluate it on the TIMIT phoneme recognition benchmark [5]. We introduce a novel recurrent layer for the LN decoder in order to find better-suited abstractions for semi-supervised learning and test two noise injection schemes tailored to support recurrent dynamics to increase the regularising nature of the RLN. Our results show that after hyper-parameter optimization the model is able to significantly outperform the baseline in all experiments using unsupervised data as a regulariser and achieves fully-supervised baseline performance while training only on 75% of the labelled data.

2 The Ladder Network Architecture

The basic idea of the LN architecture [11], depicted in Fig. 1, is to make autoencoders more expressive by adding shortcut connections from the encoder to the decoder. Each decoder layer is then able to combine the preactivation of the encoder layer with the reconstruction of the previous decoder layer by means of a combinator function $g(\cdot,\cdot)$. Therefore, the encoder does not have to carry all reconstruction information since the shortcuts can compensate for it. Since the shortcuts allow perfect reconstruction by simply copying the encoder input to the decoder output, Gaussian noise $\mathcal{N}(0,\sigma^2)$ is added to prevent the direct usage of these short-circuits and enforce learning in the intermediate layers, i.e. we use a denoising autoencoder. To ensure that the noise can be removed, the decoder's (noisy) reconstruction $\hat{\mathbf{z}}^{(l)}$ is compared to the encoder's (clean) preactivation $\mathbf{z}^{(l)}$ and added to the unsupervised objective function:

$$C_{\mathrm{DAE}} = \sum_{l}^{n} \lambda_l C_d^{(l)} \quad \text{with} \quad C_d^{(l)} = \| \, \mathbf{z}^{(l)} - \hat{\mathbf{z}}^{(l)} \|^2 \ , \tag{1}$$

where n is the total amount of layers, $\mathbf{z}^{(l)}$ is the preactivation vector of the l-th encoder layer without noise and $\hat{\mathbf{z}}^{(l)}$ the l-th decoder layer reconstruction from noisy input. The hyper-parameter λ_i controls the targeted similarity between the encoder and decoder layers and prevents short-circuits by punishing direct copies of the noisy data by weighting the difference between the layers. For semi-supervised learning the encoder path is also used for the supervised task, i.e. its output is evaluated with a supervised objective function C_{sup} and combined with the unsupervised objective function C_{DAE}: $C_{\mathrm{semsup}} = C_{\mathrm{sup}} + C_{\mathrm{DAE}}$. When using the encoder in a supervised task the shortcuts help with reconstruction as the needed information may also be retrieved over the shortcuts [11].

The combinator function $g(\cdot,\cdot)$ models $p(\mathbf{z}^{(l)} \mid \mathbf{z}^{(l+1)})$ and is responsible for creating the reconstruction of the l-th layer $\hat{\mathbf{z}}^{(l)}$ with the help of the reconstruction of the previous layer $\hat{\mathbf{z}}^{(l+1)}$ and the shortcut value of the l-th layer $\tilde{\mathbf{z}}^{(l)}$, i.e., $\hat{\mathbf{z}}^{(l)} = g(\tilde{\mathbf{z}}^{(l)}, \hat{\mathbf{z}}^{(l+1)})$. The function may attempt to remove the noise from $\tilde{\mathbf{z}}^{(l)}$ with the help of the previous reconstruction, infer the inverse mapping $\hat{\mathbf{z}}^{(l+1)} \rightarrow \hat{\mathbf{z}}^{(l)}$ or do a combination of both.

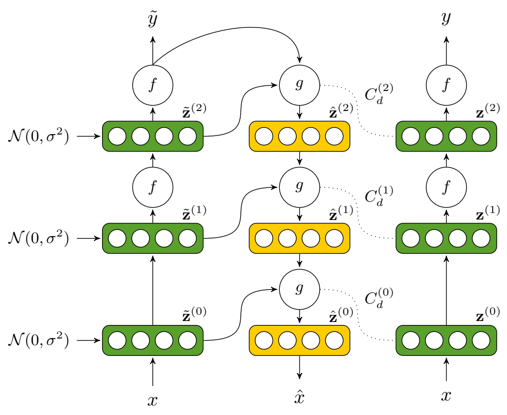


Fig. 1. Illustration of the non-recurrent LN architecture with one hidden and one output layer. The encoder and decoder paths are highlighted in **green** and yellow, respectively.

3 Recurrent Ladder Networks

In this section, we will elaborate our modelling choices for the RLN. In order to extend the original LN to support recurrence in the encoder, both the noise injection scheme and the decoder have to be adapted since recurrent layers use additional context layers. Overall, we are proposing two noise injection methods and two decoder variants (see Fig. 2). Our supervised baseline model will be the encoder of the RLN since it encodes the task closely to the full RLN but has no means of using unsupervised data. The resulting six model combinations are No-Decoder with Feed-Forward Noise (ND-FFN), No-Decoder with Recurrent Noise (ND-RN), Recurrent Decoder with Feed-Forward Noise (RD-FFN) and Recurrent Noise (RD-RN) as well as a Feed-Forward Decoder with Feed-Forward Noise (FFD-FFN) and Recurrent Noise (FFD-RN).

3.1 Noise Injection

In the *feed-forward* case, noise is applied directly to the preactivations so that the output of the layer and the shortcut are affected, i.e. $\tilde{\mathbf{z}} = W\tilde{\mathbf{x}} + \mathbf{n}$ with $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$. This would, however, introduce noise into the context memory of recurrent layers even *after* receiving the noisy output from the previous layer, effectively amplifying the noise even further. Therefore, we apply noise only to the preactivation and the shortcut without direct perturbation of the context memory. A hidden layer $\mathbf{h}_t$ and its noisy counterpart $\tilde{\mathbf{h}}_t$ are therefore updated as follows:

$$\mathbf{h}_t = f(\mathbf{z}_t) = f(W\tilde{\mathbf{x}}_t + U\mathbf{h}_{t-1}), \tag{2}$$

$$\tilde{\mathbf{h}}_t = f(\tilde{\mathbf{z}}_t) = f(\mathbf{z}_t + \mathbf{n}), \tag{3}$$

where $f(\cdot)$ is the activation function, $\mathbf{x}_t$ the input, W the input weight matrix, and U the hidden-to-hidden weights, updated at each time step t.

This noise injection method will be referred to as *recurrent noise* from here on. Another method of noise injection that we tested, referred to as *feed-forward noise*, is to not inject additional noise at the recurrent layer, i.e. feed-forward layers will be injected with noise but recurrent layers will not.

3.2 Recurrent Decoder

The decoder path in an autoencoder models the inverse information flow of the encoder path. We propose two modelling options for the decoder path in an RLN. The first (Fig. 2c) is a recurrent layer with $g(\cdot,\cdot)$ as activation function:

$$\mathbf{u}_t^{(l)} = V\hat{\mathbf{z}}_t^{(l+1)} + O\hat{\mathbf{z}}_{t-1}^{(l)}, \tag{4}$$

$$\hat{\mathbf{z}}_t^{(l)} = g(\tilde{\mathbf{z}}_t^{(l)}, \mathbf{u}_t^{(l)}), \tag{5}$$

where V are the input weights, O the hidden-to-hidden weights, $\mathbf{u}_t^{(l)}$ the pre-activation of the recurrent decoder and $\tilde{\mathbf{z}}_t^{(l)}$ the noisy preactivation of the l-th encoder layer at time-step t from the shortcut. The second modelling option is to simply use a feed-forward network (Fig. 2d) in the decoder [11].

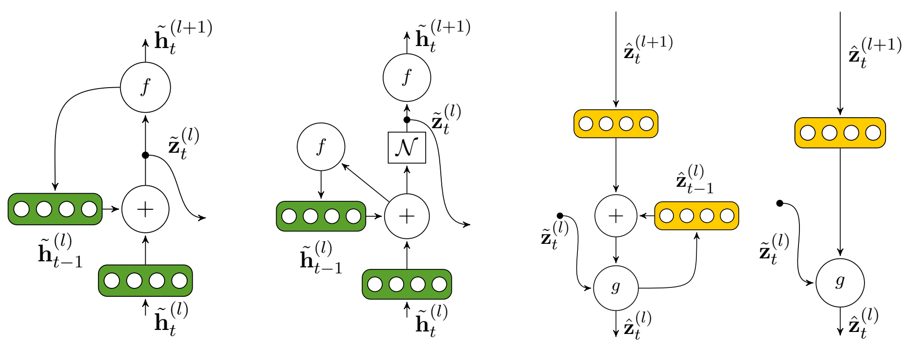
Batch normalisation is heavily used in the LN both for normalisation of the layer-wise reconstruction cost and for normalisation of layer activations. It was considered problematic with recurrent networks until the introduction of recurrent batch normalisation [2]. Since it potentially requires tuning of another hyper-parameter we decided to model the RLN without batch normalisation with the exception of the layer-wise reconstruction cost function $C_d^{(l)}$ which is computed exactly as described by Rasmus et al. [11].

4 Experiments

We evaluate the RLN on the TIMIT phoneme recognition benchmark [5], a widely used test corpus which allows comparing our architecture to previous approaches. The audio samples of the corpus are reduced in dimensionality by using libROSA[1] to compute 13 Mel Frequency Cepstral Components (MFCC) [3] and their first and second derivative with 20ms frames and 10ms frame skip, similar to related work [4]. The 39-dimensional feature vectors are normalised to have zero mean and unit variance. We grouped easily confused phonemes of the English phoneme alphabet as described by Halberstadt [8] resulting in 39 phoneme classes to predict.

We use Connectionist Temporal Classification (CTC) [6] for the supervised cost C_{sup} to solve the problem of label alignment. Phoneme Error Rate (PER) is used for evaluation and computed using the Levenshtein distance of all label sequences to the predictions, normalised to the total length of all label sequences.

[1] https://librosa.github.io


$\tilde{\mathbf{h}}_t^{(l+1)}$
f
$\tilde{\mathbf{z}}_t^{(l)}$
$+$
$\tilde{\mathbf{h}}_{t-1}^{(l)}$
$\tilde{\mathbf{h}}_t^{(l)}$
$\mathcal{N}$
$\hat{\mathbf{z}}_t^{(l+1)}$
$\hat{\mathbf{z}}_{t-1}^{(l)}$
g
$\hat{\mathbf{z}}_t^{(l)}$

(a) FFN (b) RN (c) RD (d) FFD

Fig. 2. Overview of (a) feed-forward noise (FFN) and (b) recurrent noise (RN) injection schemes for the encoders (**green**) introduced in subsection 3.1 as well as (c) recurrent decoder (RD) and (d) feed-forward decoder (FFD) layouts (yellow) introduced in subsection 3.2. Combining all encoder and decoder layouts gives a total of six model variants including the two no-decoder (ND) baselines ND-RN and ND-FFN.

The predictions are obtained by using best path decoding [6], i.e. choosing the phoneme class with the highest probability at each time step.

To build the supervised and unsupervised training sets we keep all input data for unsupervised training and reduce the supervised set by drawing samples from the full dataset until the least represented phonemes are drawn a minimum number of times to prevent under-representing a class while keeping the distribution intact. We cycle the supervised dataset to match sizes with the unsupervised set, similar to the implementation by Rasmus et al. [11].

4.1 Training Procedure

All networks have been trained using Adam [9] with a learning rate of 0.002 for at least 100 epochs until the validation error stopped improving. The models are four-layer networks consisting of one GRU layer with 192 units with $\tanh(\cdot)$ activation and one feed-forward output layer with softmax activation, as well as the inverse layers in the decoder. The noisy softmax output is used to classify phonemes during training for additional regularisation. Since the performance of the encoder is likely to correlate with RLN performance, hyper-parameters, including layer sizes and learning rate, were determined empirically by a grid search using the encoder described in section 3, i.e. an RLN with $\lambda_i = 0$, which also serves as the baseline. DAE cost weights $(\lambda_0, \lambda_1, \lambda_2) = (1000, 10, 0.1)$ and the MLP combinator $g(\cdot, \cdot)$ were both adopted from Rasmus et al. [11].

We test the semi-supervised learning capabilities of the six RLN variants from section 3, by varying the labels for the supervised part of the architecture in steps of 25% (940), 50% (1856), 75% (2754), and 100% (3696) of labelled sequences while the unsupervised part of the model always receives all available unlabelled data.

5 Results & Discussion

An overview of our results can be seen in Fig. 3 where the different modelling choices are directly compared against each other. The overall best results after hyper-parameter optimization for each supervised data split, as well as results of other approaches, are shown in Table 1.

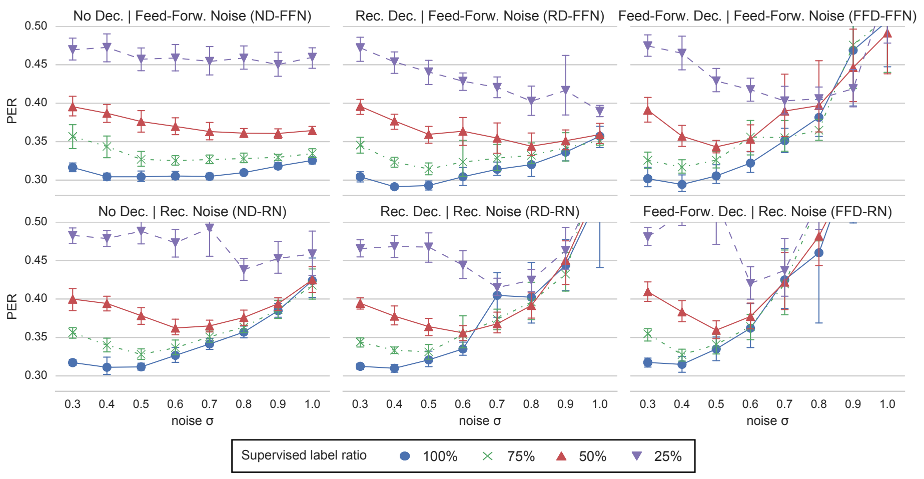


Fig. 3. Comparison of PER achieved by the RLN variants with varying amount of labelled data (25%-100%) and noise standard deviation σ. Each data point represents the mean, the whiskers cover a 95% confidence interval. Higher σ are needed for fewer labels to prevent overfitting.

As can be seen in Table 1, the RLN consistently outperforms the baseline configuration, even in fully-supervised training and is able to achieve the same performance as the baseline with 25% less labelled data which shows that the RLN complements the encoder well and demonstrates the compatibility of the LN with existing models. On average, the RD models perform better than the FFD models for most σ, more so with fewer labels, suggesting that the recurrent decoder is better at filtering noise. This also explains why the RD models work better with higher σ compared to FFD.

The noise injection method and the chosen σ greatly impact the overall performance. The performance curves are roughly concave and shift towards stronger noise with less available labels because the network overfits easily with fewer labels which is prevented by the higher noise. Performance degrades for higher σ because the network needs to be trained significantly longer to remove the noise which the chosen training parameters do not allow.

Recurrent noise injection was expected to achieve better regularisation due to the additional noise at the recurrent layer but does not. By observing the encoder layers we found that their outputs often differed significantly which causes unrecoverable perturbations in the recurrent layers when applying equally strong noise to all layers instead of noise relative to each layer's output. Employing batch normalisation might solve this, as hypothesised in related work [12]: normalising the preactivation of each layer to unit variance before adding noise makes the change in variance relative to the preactivation, therefore coupling noise and layer activation strength with the benefit of reducing the search space for σ significantly. We predict that this will lead to an increase in performance when using fewer labels.

Even though our best results for the RLN are slightly lower ranked when compared with related approaches, our model has significantly fewer parameters (e.g. differing by a factor of 160 when compared to SSSAE [4]). We therefore hypothesise that an increase of parameters and more complex layer architectures will result in even better performance. This is indicated by our best RLN achieving similar results (31.66% PER, 175k parameters) as the Bi-directional Long Short-Term Memory (BLSTM) ([6]; 31.25% PER, 114k parameters) while using only half of the labels.

Table 1. Best results in phoneme error rate (PER), achieved by the proposed RLN modelling options: No decoder (ND, baseline), recurrent decoder (RD), feedforward decoder (FFD), feedforward noise (FFN), and recurrent noise (RN). †: linear interpolation between 10% and 30% labels. ††: Graves et al. [7] have shown significantly improved results with more parameters (17.7% PER, 4.3m param.).

labels	ND-FFN	RD-FFN	FFD-FFN	ND-RN	RD-RN	FFD-RN	SSSAE [4]	BLSTM [6]
25%	40.65	**36.40**	37.13	39.90	38.82	**36.41**	31.0†	-
50%	34.22	**31.66**	32.06	34.07	**33.07**	33.39	-	-
75%	30.96	**29.16**	30.31	31.17	30.84	**30.42**	-	-
100%	29.11	**28.02**	28.08	29.26	29.67	**29.26**	-	31.25††
param	0.134m	0.177m	0.159m	0.134m	0.177m	0.159m	28.7m	0.114m

6 Conclusion

We have shown that the recurrent ladder network is able to perform as good as similarly parametrised BLSTM models while using only 50% of the labelled data, demonstrating the RLN's ability to effectively regularise itself using unsupervised training data. Current state-of-the-art methods performed better overall but this does not come as a surprise given that these models use up to 160 times more parameters. We argue that this gap could potentially be closed by scaling up our models, as demonstrated for BLSTM models by Graves et al. [7].

The proposed recurrent decoder proved to be better at denoising than the feed-forward decoder. Additionally, we found that recurrent noise injection does not perform as expected and we hypothesise that it needs the help of normalisation (e.g. batch normalisation) to work efficiently.

In the future, we would also like to take advantage of the semi-supervised learning abilities of the RLN in conjunction with more complex recurrent models such as bidirectional and attention-based RNNs to utilise unlabelled data even more effectively and explore how the learning framework scales with more complex temporal dynamics in more challenging tasks such as end-to-end speech recognition or question answering.

Acknowledgments. The authors gratefully acknowledge partial support from the German Research Foundation DFG under project CML (TRR 169), the European Union under project SECURE (No 642667), and the Hamburg Landesforschungsförderungsprojekt CROSS.

References

1. Cho, K., Van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
2. Cooijmans, T., Ballas, N., Laurent, C., Gülçehre, Ç., Courville, A.: Recurrent batch normalization. arXiv preprint arXiv:1603.09025 (2016)
3. Davis, S., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech, and Signal Processing 28(4), 357–366 (1980)
4. Dhaka, A. K., Salvi, G.: Semi-supervised learning with sparse autoencoders in phone classification. arXiv preprint arXiv:1610.00520 (2016)
5. Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S.: DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon Technical Report N 93 (1993)
6. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of ICML-2006, pp. 369–376 (2006)
7. Graves, A., Mohamed, A., Hinton, G. E.: Speech recognition with deep recurrent neural networks. In: Proceedings of ICASSP-2013, pp. 6645–6649 (2013)
8. Halberstadt, A. K.: Heterogeneous acoustic measurements and multiple classifiers for speech recognition. Ph.D. thesis, Massachusetts Institute of Technology (1998)
9. Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Liu, Y., Kirchhoff, K.: Graph-based semi-supervised learning for phone and segment classification. In: Proceedings of INTERSPEECH-2013, pp. 1840–1843 (2013)
11. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: Proceedings of NIPS-2015, pp. 3532–3540 (2015)
12. Zhang, Y., Lee, K., Lee, H.: Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In: Proceedings of ICML-2016, pp. 612–621 (2016)