Greedy InfoMax for Self-Supervised Representation Learning

Sindy Löwe* [1] **Peter O'Connor** [1] **Bastiaan S. Veeling*** [1]

Abstract

We propose a novel deep learning method for local self-supervised representation learning that does not require labels nor end-to-end backpropagation but exploits the natural order in data instead. Inspired by the observation that biological neural networks appear to learn without backpropagating a global error signal, we split a deep neural network into a stack of gradient-isolated modules. Each module is trained to maximize the mutual information between its consecutive outputs using the InfoNCE bound from Oord et al. (2018). Despite this greedy training, we demonstrate that each module improves upon the output of its predecessor, and that these representation are competitive with end-to-end optimized models on downstream classification tasks in the audio domain. The proposal enables optimizing modules asynchronously, allowing large-scale distributed training of very deep neural networks on unlabelled datasets.

1. Introduction

Modern deep learning models are typically optimized using end-to-end backpropagation and a global, supervised loss function. Although empirically proven to be very successful (Krizhevsky et al., 2012; Szegedy et al., 2015; He et al., 2016), this approach is considered biologically implausible for a number of reasons. Most importantly, despite some evidence for top-down connections in the brain, there does not appear to be a global objective that is optimized by propagating error signals backwards (Crick, 1989).

In addition to this lack of a natural counterpart, the supervised training of neural networks with end-to-end backpropagation suffers from practical disadvantages as well. Supervised learning requires labeled inputs, which are expensive to obtain. As a result, it is not applicable to the majority of available data and suffers from a higher risk of overfitting, as the number of parameters required for a deep model often exceeds the number of labeled datapoints at hand. At the same time, end-to-end backpropagation creates a substantial memory overhead in a naïve implementation, as the entire computational graph, including all parameters, activations and gradients, needs to fit in a processing unit's working memory. This prevents the application of deep learning models to large input data that surpasses current memory constraints and inhibits the efficiency of hardware accelerator design due to a lack of locality.

In this paper, we introduce a novel learning approach, *Greedy InfoMax* (GIM), that eliminates these problems by dividing a deep architecture into consecutive modules that we train greedily using a local, self-supervised loss per module. Given unlabeled high-dimensional sequential data, we encode it iteratively, module by module. By using a loss that enforces the individual modules to maximally preserve the information between consecutive inputs, we exploit the natural order of the data and enable the stacked model to collectively create compact representations that can be used for downstream tasks. Our contributions are as follows:[1]

- The proposed Greedy InfoMax algorithm achieves strong performance on audio classification tasks despite greedy self-supervised training.
- This enables asynchronous, decoupled training of neural networks, allowing for training arbitrarily deep networks on higher-dimensional input data.

2. Background

Recent work (Oord et al., 2018; Hjelm et al., 2019) has proposed how we can exploit slow features in natural data to learn representations by maximizing the mutual information shared among neighbors. In this work, we focus specifically on Contrastive Predictive Coding (CPC) (Oord et al., 2018). This self-supervised end-to-end learning approach extracts useful representations from sequences by maximizing the mutual information between the extracted representations of temporally nearby inputs.

[1]AMLab, University of Amsterdam, Netherlands (*equal contribution). Correspondence to: Sindy Löwe <loewe.sindy@gmail.com>.

Published at the ICML workshop on Self-Supervised Learning, Long Beach, USA, 2019.

[1]Our code is available at https://github.com/loeweX/Greedy_InfoMax.

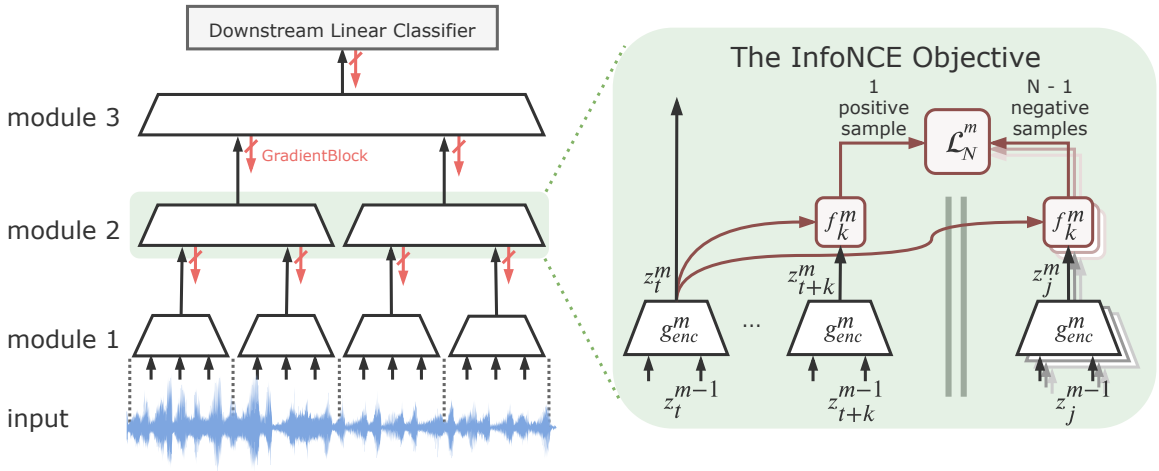

Figure 1. The Greedy InfoMax Learning Approach. **(Left)** For the self-supervised learning of representations, we stack a number of modules through which the input is forward-propagated in the usual way, but gradients do not propagate backwards. Instead, every module is trained greedily using a local loss. **(Right)** Every encoding module maps its inputs z_t^{m-1} at time-step t to $g_{enc}^m(\text{GradientBlock}(z_t^{m-1})) = z_t^m$, which is used as the input for the following module. The InfoNCE objective is used for its greedy optimization. This loss is calculated by contrasting the predictions of a module for its future representations z_{t+k}^m against negative samples z_j^m, which enforces each module to maximally preserve the information of its inputs. We employ an additional autoregressive module g_{ar}, which is not depicted here.

In order to achieve this, CPC first processes the sequential input signal x using an encoding model $g_{enc}(x_t) = z_t$, and additionally produces a representation c_t that aggregates the information of all inputs up to time-step t using an autoregressive model $g_{ar}(z_{0:t}) = c_t$. Then, the mutual information between the extracted representations z_{t+k} and c_t of temporally nearby patches is maximized by following the principles of Noise Contrastive Estimation (NCE) (Gutmann & Hyvärinen, 2010): CPC takes a bag $X = \{z_{t+k}, z_{j_1}, z_{j_2}, ...z_{j_{N-1}}\}$ for each delay k, with one "positive sample" z_{t+k} which is the encoding of the input that follows k time-steps after c_t, and $N-1$ "negative samples" z_{j_n} which are uniformly drawn from all available encoded input sequences.

Each pair of encodings (z_j, c_t) is scored using a function $f(\cdot)$ to predict how likely it is that the given z_j is the positive sample z_{t+k}. In practice, Oord et al. (2018) use a log-bilinear model $f_k(z_j, c_t) = \exp\left(z_j^T W_k c_t\right)$ with a unique weight-matrix W_k for each k-steps-ahead prediction. The scores from $f(\cdot)$ are used to predict which sample in the bag X is correct, leading to the InfoNCE loss:

$$\mathcal{L}_N = -\sum_k \mathop{\mathbb{E}}_X \left[\log \frac{f_k(z_{t+k}, c_t)}{\sum_{z_j \in X} f_k(z_j, c_t)}\right]. \qquad (1)$$

As shown by Oord et al. (2018), we can reformulate $-\mathcal{L}_N$ as a lower bound on the mutual information $I(z_{t+k}, c_t)$. Minimizing $\mathcal{L}_N$, thus optimizes $I(z_{t+k}, c_t)$, which in itself lower bounds the mutual information $I(x_{t+k}, c_t)$ between the future input x_{t+k} and the current representation c_t.

Layer-wise Information Preservation in Neuroscience Linsker (1988) developed the InfoMax principle in 1988. It theorizes that the brain learns to process its perceptions by maximally preserving the information of the input activities in each layer. Additionally, neuroscience suggests that the brain predicts its future inputs and learns by minimizing this prediction error, i.e. its "surprise" (Friston, 2010). Rao & Ballard (1999) indicate that this process may happen at each layer within the brain. Our proposal draws motivation from these theories, resulting in a method that learns to preserve the input information of each module by learning representations that are predictive of future inputs.

3. Greedy InfoMax

In this paper, we pose the question whether we can effectively optimize the mutual information between representations at each layer of a model in isolation, enjoying the many practical benefits that greedy training (decoupled, isolated training of parts of a model) provides. In doing so, we introduce a novel approach for self-supervised representation learning: Greedy InfoMax (GIM). As depicted on the left side of Fig. 1, we take a conventional deep learning architecture and divide it by depth into a stack of M modules. In our experiments, each module corresponds to an individual layer of the architecture. Rather than training this model end-to-end, we prevent gradients from flowing between modules and employ a local self-supervised loss instead.

As shown on the right side of Fig. 1, each encoding module g_{enc}^m within our architecture maps the output

from the previous module z_t^{m-1} to an encoding $z_t^m = g_{enc}^m(\text{GradientBlock}(z_t^{m-1}))$. No gradients are flowing between modules, which is enforced using a gradient blocking operator defined as $\text{GradientBlock}(x) \triangleq x, \nabla \text{GradientBlock}(x) \triangleq 0$. The encoding layers g_{enc}^m within our model are trained using $f_k^m(z_{t+k}^m, z_t^m) = \exp\left(z_{t+k}^{m\,T} W_k^m z_t^m\right)$ as a log-bilinear model and the following module-local InfoNCE loss:

$$\mathcal{L}_N^m = -\sum_k \mathop{\mathbb{E}}_X \left[\log \frac{f_k^m(z_{t+k}^m, z_t^m)}{\sum_{z_j^m \in X} f_k^m(z_j^m, z_t^m)}\right]. \quad (2)$$

In contrast to Oord et al. (2018), we do not employ an autoregressive model for the calculation of our module-local InfoNCE loss. Instead, we use the autoregressor as a separate module on top of the encoding stack, and train it greedily as described in Appendix A.

Iterative Mutual Information Maximization From Oord et al. (2018) it follows that the module-local InfoNCE loss maximizes the lower bound of the mutual information $I(z_{t+k}^{m-1}, z_t^m)$ between the future input to a module and its current representation. This can be seen as a maximization of the mutual information between the input and the output of a module, subject to the constraint of temporal disparity. Thus, the InfoNCE loss can successfully enforce each module to preserve the information of its inputs, while providing the necessary regularization (Krause et al., 2010; Hu et al., 2017) for circumventing degenerate solutions. This enables the greedily optimized modules to provide meaningful inputs to their successors and the network as a whole to provide useful features for downstream tasks without the use of a global error signal.

Practical benefit Applying GIM to high-dimensional inputs, we can optimize each module in sequence to decrease the memory costs during training. In the most memory constrained scenario, individual modules can be trained, frozen, and their outputs stored as a dataset for the next module, which effectively removes the depth of the network as a factor of the memory complexity.

4. Experiments

We evaluate GIM in the audio domain on the sequence-global task of *speaker* classification and the local task of *phone* classification. These two tasks are interesting for self-supervised representation learning as the former requires representations that discriminate speakers but are invariant to content, while the latter requires the opposite.

Experimental Details We follow the setup of Oord et al. (2018) unless specified otherwise, and use a 100-hour subset of the publicly available LibriSpeech dataset (Panayotov et al., 2015), which contains the utterances of 251 different speakers with aligned phone labels provided by Oord et al. (2018) divided into 41 classes. We first train the self-supervised model consisting of five convolutional layers and one autoregressive module, a single-layer gated recurrent unit (GRU). After convergence, a linear multi-class classifier is trained on top of the context-aggregate representation c^M without fine-tuning the representations. Remaining implementation details are presented in Appendix B.

Table 1. Results for classifying speaker identity and phone labels in the LibriSpeech dataset. All models use the same audio input sizes and the same architecture. GIM creates representations that are useful for audio classification tasks despite its greedy training and lack of a global objective.

Method	Phone Classification Accuracy	Speaker Classification Accuracy
MFCC features	39.7%	17.6%
Randomly initialized	27.6%	1.9%
Supervised	74.6%	98.5%
Greedy Supervised	71.1%	84.5%
CPC (Oord et al., 2018)[a]	64.6%	97.4%
Greedy InfoMax (GIM)	60.0%	97.5%

[a] In our reimplementation, we achieved 62.2% for the phone and 98.8% for the speaker classification task.

Results Following Table 1, we analyse the performance of models on phone and speaker classification accuracy. *Randomly initialized* features perform poorly, demonstrating that both tasks require complex representations. The traditional, hand-engineered *MFCC features* improve over the random features, but provide limited linear separability on both tasks. Both *CPC* and *GIM* get close to supervised performance on speaker classification, despite their feature models having been trained without labels, and GIM without end-to-end backpropagation. *Greedy supervised* on this task performs poorly, suggesting that the InfoMax principle suits this task especially well. On phone classification, *CPC* does not reach the supervised performance (64.6% versus 74.6%). *GIM* achieves 60%, which still improves upon the hand-engineered *MFCC features*. This discrepancy between near-supervised performance on the speaker task and less-than-optimal performance on the phone task suggests that the features extracted by GIM and CPC are biased towards sequence-global tasks.

Ablation study The layer-local training enabled by GIM provides a step towards biologically plausible optimization and improves memory efficiency. However, the autoregressive module g_{ar} aggregates multiple inputs and employs Backpropagation Through Time (BPTT), which puts a damper on both benefits. In Table 2, we present the per-

Table 2. Ablation studies for removing the biologically implausible and memory-heavy backpropagation through time.

Method	Accuracy
Speaker Classification	
Greedy InfoMax (GIM)	97.5%
GIM without BPTT	95.2%
GIM without g_{ar}	98.5%
Phone Classification	
Greedy InfoMax (GIM)	60.0%
GIM without BPTT	55.4%
GIM without g_{ar}	50.8%

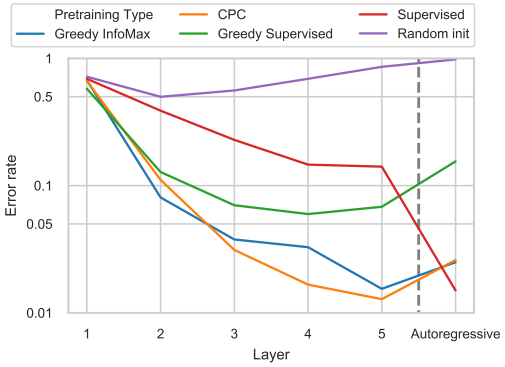

Figure 2. Speaker Classification error rates on log scale (lower is better) for intermediate representations (layers 1 to 5), as well as for the final representation created by the autoregressive layer (corresponding to the results in Table 1).

formance of ablated models that block gradients flowing between time-steps (*GIM without BPTT)* or remove the autoregressive module altogether (*GIM without* g_{ar}).

Together, these two ablations indicate a crucial difference between the tested downstream tasks. For the phone classification task, we see a steady decline of performance when we reduce the modelling of temporal dependencies, indicating their importance for solving this task. When classifying the speaker identity, on the other hand, the GIM Encoder, which models temporal dependencies the least, performs the best of all GIM models. This indicates that the GIM approach performs best on downstream tasks where temporal dependencies do not need to be modeled by an autoregressive module. In this setting, GIM's performance is on par with the CPC model which makes use of end-to-end backpropagation, a global objective, and BPTT.

Intermediate module representations The greedy layer-wise training of GIM allows us to train arbitrarily deep models without ever running into a memory constraint. We investigate how the created representations develop in each individual module by training a linear classifier on top of each layer and measuring their performance on the speaker identification task. With results presented in Fig. 2, we first observe that each GIM encoding module improves upon the representation of its predecessor. Interestingly, CPC exhibits similar performance in intermediate modules despite these modules relying solely on the error signal from the global loss function on the last module. This is in stark contrast with the supervised end-to-end model, whose intermediate layers lag behind their greedily trained counterparts. This suggests that, in contrast to the supervised loss, the InfoMax principle "stacks well", such that the greedy, iterative application of the InfoNCE loss performs similar to its global application.

5. Related Work

We have studied the effectiveness of the self-supervised CPC approach (Oord et al., 2018; Hénaff et al., 2019) when applied to gradient-isolated modules, freeing the method from end-to-end backpropagation. There are a number of optimization algorithms that eliminate the need for backpropagation altogether (Scellier & Bengio, 2017; Lillicrap et al., 2016; Kohan et al., 2018; Balduzzi et al., 2015; Lee et al., 2015). In contrast to our method, these methods employ a global supervised loss function and focus on finding more biologically plausible ways to assign credit to neurons.

A recently published work by Nøkland & Eidnes (2019) demonstrates that backpropagation-free biologically plausible training is possible, with a focus on supervised local signals. In an attempt to validate information bottleneck theory, Elad et al. (2018) develop a supervised, layer-wise training method. In contrast to our proposal, these methods rely on labeled data.

Jaderberg et al. (2017) develop decoupled neural interfaces, which enjoy the same asynchronous training benefits as Greedy InfoMax (GIM), but achieve this by taking an end-to-end supervised loss and locally predicting its gradients. Hinton et al. (2006) and Bengio et al. (2007) focus on deep belief networks and propose a greedy layer-wise unsupervised pretraining method based on Restricted Boltzmann Machine principles, followed by optimizing globally using a supervised loss.

6. Conclusion

We presented Greedy InfoMax, a novel self-supervised greedy learning approach. The relatively strong performance demonstrates that deep neural networks do not necessarily require end-to-end backpropagation of a supervised loss on perceptual tasks. Our proposal makes the model less vulnerable to overfitting, reduces the vanishing gradient problem and enables memory-efficient asynchronous distributed training.

References

Balduzzi, D., Vanchinathan, H., and Buhmann, J. Kickback cuts backprop's red-tape: biologically plausible credit assignment in neural networks. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pp. 153–160, 2007.

Crick, F. The recent excitement about neural networks. *Nature*, 337(6203):129–132, 1989.

Elad, A., Haviv, D., Blau, Y., and Michaeli, T. The effectiveness of layer-by-layer training using the information bottleneck principle. *OpenReview*, 2018.

Friston, K. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127, 2010.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hénaff, O. J., Razavi, A., Doersch, C., Ali Eslami, S. M., and van den Oord, A. Data-Efficient image recognition with contrastive predictive coding. May 2019.

Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, 18 (7):1527–1554, 2006.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Hu, W., Miyato, T., Tokui, S., Matsumoto, E., and Sugiyama, M. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1558–1567. JMLR. org, 2017.

Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., Silver, D., and Kavukcuoglu, K. Decoupled neural interfaces using synthetic gradients. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1627–1635, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kohan, A. A., Rietman, E. A., and Siegelmann, H. T. Error forward-propagation: Reusing feedforward connections to propagate errors in deep learning. *arXiv preprint arXiv:1808.03357*, 2018.

Krause, A., Perona, P., and Gomes, R. G. Discriminative clustering by regularized information maximization. In *Advances in neural information processing systems*, pp. 775–783, 2010.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Lee, D.-H., Zhang, S., Fischer, A., and Bengio, Y. Difference target propagation. In *Joint european conference on machine learning and knowledge discovery in databases*, pp. 498–515. Springer, 2015.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7:13276, 2016.

Linsker, R. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Nøkland, A. and Eidnes, L. H. Training neural networks with local error signals. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Rao, R. P. and Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2 (1):79, 1999.

Scellier, B. and Bengio, Y. Equilibrium propagation: bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

A. Training of the autoregressive module

For some downstream tasks, a broad context is essential. In speech recognition, for example, the receptive field of z_t^M might not carry the full information required to distinguish phonetic structures. To provide this context, we use the autoregressive model g_{ar} as an independent module that we append to the stack of encoding modules, resulting in a context-aggregate representation $c_t^M = g_{ar}^M\left(\text{GradientBlock}\left(z_{0:t}^{M-1}\right)\right)$. We train this module independently using the module-local InfoNCE loss with the following adjusted scoring function:

$$f_k^M(z_{t+k}^{M-1}, c_t^M) = \exp\left(\text{GradientBlock}\left(z_{t+k}^{M-1}\right)^T W_k^M c_t^M\right).$$

B. Experimental Setup

We use PyTorch (Paszke et al., 2017) for all our experiments. The detailed description of our employed architecture is given in Table 3. We train our model on 4 GPUs (GeForce 1080 Ti) each with a minibatch of 8 examples. Our model is optimized with Adam (Kingma & Ba, 2014) and a learning rate of 2e-4 for 300 epochs. We use the same random seed for all our experiments. Overall, our hyperparameters were chosen to be consistent with Oord et al. (2018).

Table 3. General outline of our architecture

Layer	Output Size (Sequence Length × Channels)	Parameters Kernel	Stride	Padding
Input	20480×1			
Conv1	$4095^2 \times 512$	10	5	2
Conv2	$1023^2 \times 512$	8	4	2
Conv3	$512^2 \times 512$	4	2	2
Conv4	$257^2 \times 512$	4	2	2
Conv5	128×512	1	2	1
GRU	128×256	-	-	-

[2]For applying the InfoNCE objective on these layers, we randomly sample a time-window of size 128 to decrease the dimensionality.

For the self-supervised training using the InfoNCE objective, we need to contrast the predictions of the model for its future representations against negative samples. We draw these samples uniformly at random from across the input batch that is being evaluated. Thus, the negative samples can contain samples from the same audio file at different timings, as well as from different audio files. We found that including the positive sample (i.e. the future representation that is currently to be predicted) in the negative samples did not have a negative effect on the final performance. For each evaluation of the InfoNCE loss, we use a total of 10 negative samples and predict up to $k = 12$ time-steps into the future.

We train the linear logistic regression classifier using the representations of the top, autoregressive module without pooling. Again, we employ the Adam optimizer but select different learning rates than before. For this hyperparameter search, we split the training set provided by Oord et al. (2018) into two random subsets using 25% of the samples as a validation set. In the speaker classification experiment, we use a learning rate of 1e-3, while we set it to 1e-4 for the phone classification experiment.