Записки научных
семинаров ПОМИ
Том 529, 2023 г.

**D. Grebenkin, I. Bondarenko**

# WAV2VEC2 WITHOUT ATTENTION: DO YOU NEED HOPFIELD NETWORKS FOR SELF-SUPERVISED LEARNING OF SPEECH REPRESENTATIONS?

Abstract. In this work, we consider the possibility of replacing multi-head attention with dense associative memory (DAM) layers in the wav2vec2 automatic speech recognition algorithm. We examine the hypothesis that the concept of modern Hopfield networks is more suitable for restoration of missing fragments of the audio signal task and speech-to-text task than multi-head attention. Our experiments indicate that the model with the new architecture allows to improve the quality of speech recognition and can be used for pretraining the models on a large amount of audio data.

## §1. Introduction

Automatic speech recognition (ASR) is one the most popular tasks in modern computer linguistics; these algorithms are used in a plethora of different electronic devices. There are two main approaches to ASR: classic (component) and end-to-end. Although the second one is more popular nowadays, component systems were the most used for a long period of time because some classic systems like Kaldi [20] could be adapted to the special speech domain only by training the language models involved on texts from that domain. It made these systems competitive in some special tasks such as speech processing for low-resource languages. Nevertheless, modern end-to-end systems have some advantages that make them more popular. Although the performance quality of these systems depends only on the training dataset, the transfer learning [21] approach can still be applied; it considers the use of knowledge and rules of the base model, which solves a specific problem, to solve another similar task within a particular language. What is more, neural network algorithms have better performance, they require less computing resources, and their size can be reduced by using quantization or pruning techniques.

*Key words and phrases*: speech recognition and self-attention and associative memory.

Most modern ASR end-to-end systems have Transformer-based architectures; the Transformer uses a multi-head attention [24] approach to find dependencies among input tokens. However, some studies show its disadvantages: Transformers have limitations on a big class of regular languages [6], self-attention can perform worse because of the rather long input tokens [23], the construction of abstract representations is always made from lower layers, although "higher layers are available" [10]. Our goal in this work was to test the possibility of replacing multi-head attention with associative memories for self-supervised learning of speech representations from unlabeled data and for transforming these representations to the words of a natural language. We used a small amount of audio data to compare different architectures and confirm our hypothesis.

## §2. Related work

Transformer-based architectures made a breakthrough for natural language processing tasks, and they have been applied in speech recognition since 2018. The authors of the "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition" [9] modified the standard architecture with convolutional neural network (CNN) [17] layers to process data before passing it to the encoder as inputs. The authors tested the models based on this architecture with the Wall Street Journal dataset and they used word error rate (WER) as the evaluation metric. The results showed that Transformer-based models can compete with other automatic speech recognition approaches.

A more global change in architecture was proposed in the work "Transformer Transducer: A Streamable Speech Recognition Model with Transformer Encoders and RNN-T Loss" [27]. The only stack of audio and label encoders was used. The researchers used a Recurrent Neural Network Transducer (RNN-T) instead of CTC (connectionist temporal classification) as a loss function, making it possible to use probabilities based not only on the inputs but also based on predicted labels. The training of the Transformer-Transducer took less time in comparison with the original recurrent neural network (RNN) and it had better accuracy on the LibriSpeech benchmark.

The paper "Conformer: Convolution-Augmented Transformer for Speech Recognition" [12] introduced an approach that combined the advantages of Transformers and CNNs. The Conformer architecture had a convolution module after the multi-head self-attention module. It made it possible to

take both global and local dependencies of the input audio into account. This model outperformed the previous Transformer and CNN in terms of WER results on LibriSpeech dataset. The Conformer-based model for the Russian language was published in 2022 by NVIDIA, and it had low WER values on the Russian LibriSpeech dataset [2].

Researchers from *Facebook* proposed the *wav2vec 2.0* framework for self-supervised learning of speech recognition models in 2020 [5]. The training of this model meant processing some unlabeled audiodata to learn speech representations. The second step was to teach a model to predict phonemes of speech by using fine-tuning techniques with the CTC loss. This model achieved state of the art on LibriSpeech that year. The developers also trained a single *wav2vec2* model to predict cross-lingual speech representations by training it on speech taken from several languages [8]. The model had low WER and phoneme error rate values, and it was fine-tuned to lots of different domains.

## §3. Dense associative memory and self-attention

**3.1. Self-attention analogues.** The disadvantages of self-attention algorithms motivate researchers to find new ways to learn the dependencies among input data. The authors of the study "Addressing Some Limitations of Transformers with Feedback Memory" [10] proposed a different Transformer architecture which forms the inputs for the current timestep with some abstract representations from previous steps, called "memory vectors". This feedback feature allowed the proposed architecture to improve the quality and speed parameters in comparison with other Transformer architectures, and the amount of parameters was reduced by a factor of two. In the work "Hopfield Networks is All You Need" [22] the researchers underline the fact that some attention heads perform averaging over a very large number of patterns in lower layers which leads to lack of attention to some positions on this level. They propose to use Gaussian averaging heads with fewer parameters than self-attention heads.

**3.2. Hopfield networks evolution.** In the same work the authors proposed to use associative memory layers in Transformers. Their work was based on the research of Hopfield networks.

A Hopfield network is a fully connected recurrent neural network; it is basically an auto-association mechanism: the network can recover whole images by using one or several of their fragments as an input. Information

about the images is stored in the weights of the network after its training, and it can be described with the following formula [25]:

$$w_{ij} = \sum_{d=1..m} (OUT_{i,d} OUT_{j,d})$$

where $m$ is the number of memorized input vectors, $d$ is the number or memorized output vector, $OUT_{i,j}$ is the $i$th component of the memorized output vector.

However, Hopfield networks have a memory restriction: the amount of trainable images can be estimated [13] as

$$K^{\max} \approx 0.14N$$

where $N$ is the number of neurons.

In the work "Dense Associative Memory for Pattern Recognition" [15], D. Krotov and J.J. Hopfield presented a modification of the Hopfield energy function that defines the associative memory mechanism:

$$E = -\sum_{\mu=1}^{K} F(\sum \xi_i^\mu \sigma_i)$$

where $\sigma_i$ are dynamical variables and $\xi_i^\mu$ are memorized patterns. They called it **"dense associative memory" (DAM)** because it allowed to add more image information to the memory of this network:

$$K^{\max} \approx \alpha_n N^{n-1}$$

**3.3. DAM for speech recognition.** The authors of "Hopfield Networks is All You Need" developed a generalization of the Hopfield energy function that allowed them to use the Hopfield energy function in place of the attention mechanism. They developed an implementation of DAM layers for using it in deep learning architectures for pooling, memory, prototype learning, and attention tasks. We decided to use this implementation in our experiments. Our theory is that DAM is better than self-attention in speech recognition for the following reason: the ASR model with trained DAM layers contains images of phoneme representations from a natural language, and they can be recovered from any type of input audio. We think that using DAM can make this model stable to background noises, to reverberation effects, to speech features such as dialects or pronunciation

Table 1. Character error rate values (percents) on test data.

| **Dataset** | **w2v2-classic-tiny** | **w2v2-hopfield-tiny** |
|---|---|---|
| voxforge-ru | 47.6 | **35.9** |
| sberdevices-golos-crowd (test) | 48.9 | **35.5** |
| sberdevices-golos-farfield (test) | 57.4 | **45.6** |

disorders, and it will not be necessary to train this model on a large amount of augmented data.

We used modern Hopfield network layers for creating a new ASR model with a wav2vec2-based architecture to test our ideas. We pretrained two Russian ASR models: a version with the classic wav2vec2 architecture (**w2v2-classic-tiny**) and a modified version with Hopfield layers (**w2v2-hopfield-tiny**). The pretraining was a self-supervised learning task, the models learned speech representations by using a part of unlabeled inputs as labels. We used the implementation of Hopfield layers from the Python library [1] for the second model: the original Wav2Vec2EncoderLayer was replaced by HopfieldEncoderLayer for Transformers with the following parameters: *input size = 768*, *dropout = 0.1*, *dim feedforward = 2048*, *activation = 'relu'*.

The second step was to fine-tune both models on correctly labeled data to make them convert phonemes into words. The hyperparameters were as follows: *batch size = 32*, *epochs = 8*.

## §4. Experiments and discussion

**4.1. Data evaluation.** A part of the SOVA Dataset *RuYoutube* [3] was used as a dataset for pretraining both models; it contains 200 hours of Russian audio records. We used the SOVA Dataset *RuDevices* [3] which contained 100 hours of Russian speech for fine-tuning.

The test subsets of the Russian *voxforge* dataset (voxforge-ru) [4] and Russian Golos [14] (sberdevices-golos-crowd (test), sberdevices-golos-farfield (test)) were used for model evaluation. We used the character error rate (CER) as the comparison metric, and the resulting values are shown in Table 1.

We have also tested the models on augmented data to confirm our assumption about model resistansce to noise artefacts. Audio from the *voxforge* dataset was augmented with sounds from the Freesound Audio

Table 2. Character error rate values (percents) on augmented voxforge-ru.

| Noise type | SNR (dB) | w2v2-classic-tiny | w2v2-hopfield-tiny |
|---|---|---|---|
| pets noises | 5.0 | **54.9** | 55.0 |
| pets noises | 10.0 | 48.4 | 48.4 |
| pets noises | 15.0 | 43.8 | 43.8 |
| speech noises | 5.0 | 63.9 | **63.8** |
| speech noises | 10.0 | **54.8** | 54.9 |
| speech noises | 15.0 | 47.8 | **47.6** |

Tagging 2019 research code competition corpora [11] and with different signal-to-noise ratio (SNR) values. The results are presented in Table 2.

Our experimental results indicate that **w2v2-hopfield-tiny** has a significantly better error rate value than the classic model: it was reduced by 12.3 percent on average. The model also had better decoding time on test data: 4 percent faster on Russian Golos and 5 percent faster on voxforge than **w2v2-classic-tiny**. The test on augmented data shows that the models have approximately equal CER values for various types of noises and SNR values.

**4.2. Discussion.** We did the analysis of predicted labels from the *voxforge* dataset (100 files) to explain these results. The embeddings of **stressed vowels** (IPA: [a], [æ], [ɛ], [e], [ɨ], [i], [o], [ɵ], [u], [ʉ]), **unstressed vowels** (IPA: [ɐ], [ə], [ o̞], [ɪ], [ɨ̞], [ʊ], [ʉ̞]), **affricates** (IPA: [ʦ], [ʨ]), **occlusive consonants** (IPA: [b], [bʲ], [p], [pʲ], [d], [dʲ], [t], [tʲ], [g], [gʲ], [k], [kʲ]), **fricative consonants** (IPA: [f], [fʲ], [v], [vʲ], [s], [sʲ], [z], [zʲ], [x], [xʲ], [ʂ], [ʐ], [ɕː]), and **sonorants** (IPA: [r], [rʲ], [l], [lʲ], [m], [mʲ], [n], [nʲ]) were collected with the forced alignment method [16]:

(1) the last hidden layer outputs (an array of embeddings) of a model (**w2v2-hopfield-tiny** or **w2v2-classic-tiny**) were extracted after processing an input audio;
(2) the big Russian wav2vec2 model [7] was used to get the logits and to find the time segments of every sound in the input audio while using the letters of audio transcription as written representations (graphemes) of the phonemes;

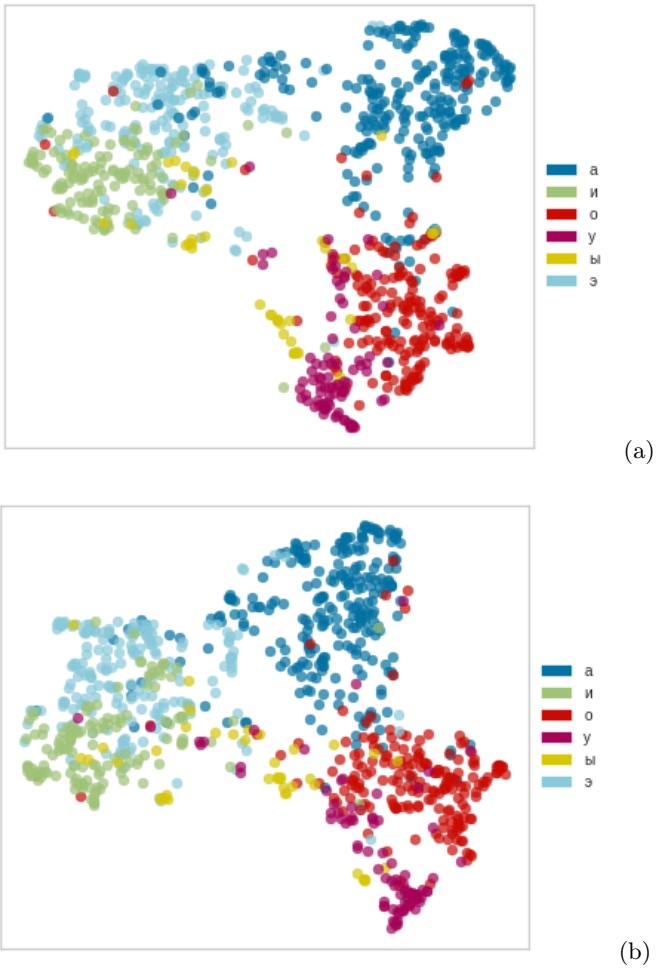


Figure 1. UMAP projections of 768-dimensional contextual embeddings of predicted letters (representations of stressed vowels) from the last hidden layer of (a) **w2v2-classic-tiny**, (b) **w2v2-hopfield-tiny**.

Table 3. Silhouette Coefficient scores for some types of sound embeddings.

| **Letter representations** | **w2v2-classic-tiny** | **w2v2-hopfield-tiny** |
|---|---|---|
| stressed vowels | 0.0613 | **0.0615** |
| unstressed vowels | 0.0048 | **0.0050** |
| sonorants ('ð', 'ë', 'ǐ', 'ì') | **0.1469** | 0.1365 |
| occlusive consonants ('á', 'ï', 'ä', 'ó', '$\tilde{a}$', '$\hat{e}$') | 0.1059 | **0.1073** |
| affricates ('œ', 'ö') | 0.2458 | **0.2763** |
| fricative consonants ('$\hat{o}$', '$\hat{a}$', '$\tilde{n}$', 'ç', 'ø', 'ú', 'æ', '$\tilde{o}$') | **0.0842** | 0.0788 |

(3) these time segments were used to collect the arrays of the corresponding embeddings of sounds for every letter from the tiny model outputs;
(4) the arrays of embeddings were averaged.

A grapheme (letter) can correspond to several phonemes and their allophones, and that is why we can use it as a label for our analysis. Forced alignment of stressed vowels was performed with accents from the russian-g2p tool [26]. The labels 'øa', 'þ', 'ÿ' of stressed vowels represent other phonemes in different positions in writing, so they were transformed to 'ý', 'ó', 'á' respectively.

The UMAP projections [19] of stressed vowel embeddings are shown in Figure 1. The long stressed vowel durations made it possible to show the bounds of close, mid, open lines and front, central, back columns of the real Russian vowel chart [18]; it can be seen most clearly in Figure 1(b). The both models make the similar phonemes representations because the stressed vowel sounds have the constant frequency sound ranges (two first formants). However, the **w2v2-classic-tiny** representation clusters are more scattered, some of them are tinier like '$\hat{u}$' in comparison with Figure 1(b).

We compared the quality of sound representations with the Silhouette Coefficient (Table 3) and we used the Wilcoxon signed-rank test with all types of embeddings to find out the statistical significance of the advantage of **w2v2-hopfield-tiny**'s performance over **w2v2-classic-tiny**. The test statistic was 9.0 with a $p$-value of 0.84375, which means that the models have obtained relatively equivalent images of speech representations. The

**w2v2-hopfield-tiny** had worse Silhouette Coefficient scores than **w2v2-hopfield-tiny** on some groups of consonants but it has better values on all groups of vowels.

## §5. Conclusion

We have confirmed the hypothesis that using the dense associative memory for self-supervised learning for speech recognition is more efficient when pretraining tiny models with different architectures on a small audio dataset. We have successfully changed the original Transformer-based ASR model, and it achieved better error rates on test data than the classic model. Nevertheless, the reason for this efficiency is not fully explained. Our assumption about the better compactness of the speech representation from w2v2-hopfield has been only partially confirmed (for phonetic classes of longer duration, such as stressed vowels). Since we do not have a solid foundation, we cannot deny that for larger neural networks the observed picture may change.

Further work will focus on several goals. The first is pretraining wav2vec2-hopfield on a larger amount of data to develop a competitive solution for many speech recognition tasks. Another direction of study will be devoted to modeling specific associative dependencies among the phonemes of many languages to create a multilingual ASR model.

## References

1. *ml-jku/hopfield-layers: Hopfield networks is all you need*, `https://github.com/ml-jku/hopfield-layers`, Last access: 2023-02-15.
2. *Nvidia/stt-ru-conformer-transducer-large · hugging face*, `https://huggingface.co/nvidia`, Last access: 2023-01-30.
3. *sovaai/sova-dataset*, `https://github.com/sovaai/sova-dataset`, Last access: 2023-02-15.
4. *voxforge.org*, `http://www.voxforge.org/ru`, Last access: 2023-02-15.
5. A. Baevski, H. Zhou, A. Mohamed, M. Auli, *wav2vec 2.0: A framework for self-supervised learning of speech representations*, (2020).
6. S. Bhattamishra, K. Ahuja, N. Goyal, *On the ability and limitations of transformers to recognize formal languages*, 01 2020, pp. 7096–7116.
7. I. Bondarenko, *Xlsr wav2vec2 russian*, `https://huggingface.co/bond005/wav2vec2-large-ru-golos`, 2022.
8. A. Conneau, A. Baevski, R. Collobert, A. Mohamed, M. Auli, *Unsupervised cross-lingual representation learning for speech recognition*, (2020).
9. L. Dong, S. Xu, Bo Xu, *Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition*, 04 2018, pp. 5884–5888.

10. A. Fan, T. Lavril, E. Grave, A. Joulin, S. Sukhbaatar, *Addressing some limitations of transformers with feedback memory*, 2020.
11. E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, X. Serra, *Audio tagging with noisy labels and minimal supervision*, 2020.
12. A. Gulati, J. Qin, Chung-Cheng Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, R. Pang, *Conformer: Convolution-augmented transformer for speech recognition*, (2020).
13. J. Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, Proceedings of the National Academy of Sciences of the United States of America **79** (1982), 2554–8.
14. N. Karpov, A. Denisenko, F. Minkin, *Golos: Russian dataset for speech research*, (2021).
15. D. Krotov, J. Hopfield, *Dense associative memory for pattern recognition*, (2016).
16. L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, G. Rigoll, *Ctc-segmentation of large corpora for german end-to-end speech recognition*, Speech and Computer (Cham) (Alexey Karpov and Rodmonga Potapova, eds.), Springer International Publishing, 2020, pp. 267–278.
17. Y. Lecun, B. Boser, J. Denker, D. Henderson, R. Howard, W. E. Hubbard, L. Jackel, *Backpropagation applied to handwritten zip code recognition.* — Neural Computation **1** (1989), 541–551.
18. M. I. Matusevich, *Modern russian language. phonetics.* Prosveshchenie Publ., 1976.
19. L. McInnes, J. Healy, J. Melville, *Umap: Uniform manifold approximation and projection for dimension reduction*, 2018.
20. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, K. Vesel, *The kaldi speech recognition toolkit*, IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (2011).
21. L. Y. Pratt, *Discriminability-based transfer between neural networks*, Proceedings of the 5th International Conference on Neural Information Processing Systems (San Francisco, CA, USA), NIPS'92, Morgan Kaufmann Publishers Inc., 1992, p. 204–211.
22. H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, L. Gruber, M. Holzleitner, M. Pavlovic, G. Sandve, V. Greiff, D. Kreil, M. Kopp, G. Klambauer, J. Brandstetter, S. Hochreiter, *Hopfield networks is all you need*, (2020).
23. D. Varis, O. Bojar, *Sequence length is a domain: Length-based overfitting in transformer models*, (2021).
24. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, *Attention is all you need*, 2017.
25. P. D. Wasserman, *Neural computing: theory and practice*, Van Nostrand Reinhold Co., New York, NY, USA, 1989.

26. O. Yakovenko, I. Bondarenko, M. Borovikova, D. Vodolazsky, *Algorithms for automatic accentuation and transcription of russian texts in speech recognition systems*, Speech and Computer (Cham) (Alexey Karpov, Oliver Jokisch, and Rodmonga Potapova, eds.), Springer International Publishing, 2018, pp. 768–777.
27. Q. Zhang, Han Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, S. Kumar, *Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss*, (2020).

Laboratory of Applied Digital Technologies,
Novosibirsk State University
*E-mail*: d.grebenkin@g.nsu.ru

Laboratory of Applied Digital Technologies,
Novosibirsk State University
*E-mail*: i.bondarenko@g.nsu.ru

Поступило 6 сентября 2023 г.