On the Inductive Bias Transfer with Knowledge Distillation for Real-World Data

Byeong Tak Lee, Yong-Yeon Jo and Joon-myoung Kwon

MedicalAI, Inc., 163, Yangjaecheon-ro, Seoul, South Korea

Abstract

In the lack of data, an appropriate inductive bias is one of the key factors for the successful training of a model. One approach to transfer inductive bias between the different structures of networks is to utilize knowledge distillation. Several studies have achieved promising results in computer vision datasets using response-based knowledge distillation. However, we observe that the previous method fails to transfer inductive bias when the dataset contains fewer data points or classes. To solve the problem, we propose to use feature-based knowledge distillation instead of response-based knowledge distillation for effective inductive bias transfer. Through extensive experimentation and analysis, we demonstrate that the suggested method can transfer inductive bias and outperform previous methods.

Keywords

Inductive bias, Knowledge distillation, Electrocardiogram, Electronic health record

1. Introduction

Inductive biases are constraints enforcing the model to have specific properties [1, 2]. For example, convolution layer enforces the model to have properties of translational invariance and translational equivalence, and recurrent layer enforces the model to have properties of temporal invariance. The effect of an appropriate inductive bias is comparable to the effect of additional data; in other words, one can compensate for the lack of data by exploiting strong inductive biases [1, 2]. Nevertheless, such constraints are not always advantageous. If the inductive bias is too restrictive, the model can only learn limited representations [1]. One approach for encoding inductive bias in balance is knowledge distillation. For example, Data-efficient image Transformers (DeiT) use the convolution neural network to inherit its inductive bias to the Transformer network. It uses the distillation token to predict the output of the pre-trained convolutional neural network, achieving performance on par with the model already trained with a strong inductive bias [3]. By adjusting the hyperparameters of knowledge distillation, the level of inductive bias can be controlled.

We wonder about the applicability of transferring the inductive bias via knowledge distillation in various real-world datasets. To verify this, we evaluated the technique transferring the inductive bias in used DeiT on two types of medical datasets: the inductive biases (1) in convolutional neural networks (CNN) on the electrocardiograms (ECG) and (2) in recurrent neural networks (RNN) on the electronic health records (EHR). As shown in Figure 1, we observed that the performance of the transformer trained with DeiT is significantly inferior to that of the teacher networks. The result is a completely different result from DeiT [3]. This is the beginning point of our study. In this study, we first identify the reasons for the previous method's failure. After then, we propose a method for resolving it.

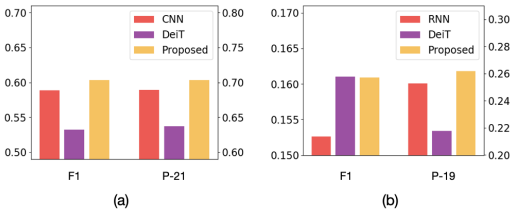

Figure 1: Performance in (a) ECG from Physionet 2021 and (b) EHR from Physionet 2019. F1 refers to the f-1 score (the left side of the y-axis), and P-19/P-21 indicate the physionet 19 and physionet 21 scores (the right side of the y-axis).

Our contributions to this study are the following: First, we analyze the limitation of the previous methods of transferring the inductive bias through knowledge distillation. Second, we examine the reason for the failure of the previous methods via rigorous experiments. Third, based on the findings from the experimental results, we propose an effective way to transfer inductive biases through knowledge distillation.

AMLTS'22: Workshop on Applied Machine Learning Methods for Time Series Forecasting, co-located with the 31st ACM International Conference on Information and Knowledge Management (CIKM), October 17-21, 2022, Atlanta, USA

bytaklee@medicalai.com (B. T. Lee); yy.jo@medicalai. (Y. Jo); cto@medicalai.com (J. Kwon)

2. Demystifying inductive bias encoded in the student network

There are two possible reasons for the failure of the previous method. First, if the teacher's inductive bias is weak, the signal from the teacher can be insufficient to provide valid information to the student (Figure 2(a)). The second possible explanation is that, even if the teacher has a sufficient inductive bias, the force pushing the student network to encode the teacher's inductive bias may be insufficient (Figure 2(b)). In this section, we explore the teacher's and student's representations and filters to identify the reason for the limitation of the previous approach.

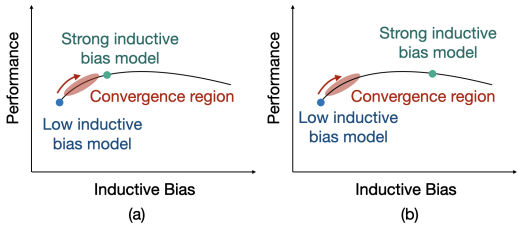

Figure 2: Inductive bias transfer depending on teacher model. (a) The inductive bias of teacher is not proper for the task. (b) Driving force to encode inductive bias to student is too weak.

2.1. Experiment setting

2.1.1. Dataset

We used the following datasets with different properties: Physionet 2021 for CNN and Physionet 2019 for RNN. Physionet 2021 is a public ECG datasets [4], which contains approximately 88,000 ECGs. Each ECG is assigned one or more arrhythmia labels for 26 classes of arrhythmia [4]. PhysioNet 2019 [5] is an EHR consisting of hourly clinical variables collected from the intensive care unit (ICU) of two hospital systems with 40,336 patients. The task is to predict sepsis within 12 hours, and the onset of sepsis is given to each patient.

We additionally used two external datasets to see if the DeiT preserves the inductive biases of CNN/RNN regardless of the data distribution. The Hangzhou dataset [6] contains 20,036 ECG recordings, and the eICU Collaborative Research Database [7] is a multi-center database containing over 200,000 admissions to ICU.

2.1.2. Architecture

We develop two teacher networks: (1) the ResNet-based network for ECG datasets [8]. Each block of ResNet contains two layers of convolution, and there are eight blocks in total. The architectural detail is identical to Hannun et al. [8]. (2) the long short time memory (LSTM) network for EHR datasets [9]. LSTM is stacked with the 3-layer, and each layer has 256 hidden units with a residual connection between each layer.

As a student network, we adopt a transformer [10]. There are two student networks, each of which has eight blocks for the ECG dataset and three blocks for the EHR dataset. Training a transformer on ECG datasets, we split a signal into patches following Dosovitskiy et al. [11]. Each patch consists of 100ms (20 timestamps) without overlapping and is used as the input of a transformer. In EHR datasets, a patient has multiple rows, each of which consists of a medical record at a time. A single row is used as a token of the input.

2.1.3. The other details of experiments

We set a batch size of 512 for the ECG dataset and a batch size of 256 for the EHR dataset. We use an Adam optimizer with the weight decay and the cosine warmup scheduler that peaks at ten epochs. In the experiment with ECGs, the rand augment policy [12] is adopted with six data augmentation methods, including the gaussian smoothing, time resampling with cut, gaussian noise, baseline wander, time mask, and channel mask. In the case of EHRs, data augmentation is not applied. Hyperparameters, such as the learning rate, weight decay, dropout, and parameters for the augment policy, are randomly selected from predefined search space, tuned by the asynchronous successive halving algorithm [13] using the ray framework [14]. The search space and selected hyperparameters are provided in Appendix A. Train, validation, and test set are divided into a ratio of 0.7:0.15:0.15.

2.2. Representation analysis

In order to analyze the inductive bias caused by the structure of networks, we first compare the representations of the teacher and the student networks. If the student(Transformer) successfully encodes the inductive bias of the teacher(CNN/RNN), there are high similarities in the representations between them (Figure 2(a)). On the other hand, if the similarities between the teacher's and the student's representations are low, the teacher's representation is not effectively transferred to the student (Figure 2(b)).

2.2.1. Output similarity

We first examine the output similarity as shown in Table 1. The number in the table is the r-square value. The similarity between DeiT and its teacher (CNN/RNN) is slightly higher than the similarity between the naive transformer and CNN/RNN; however, the discrepancy between the teacher and the student is still large. This

Table 1
Probability similarity between the CNN/RNN and the transformers trained using different methods.

	ECG		EHR	
	P21	Hanzhou	P19	eICU
Transformer	0.530	0.174	0.439	0.308
DeiT	0.601	0.206	0.464	0.351
BBM	0.796	0.592	0.776	0.728

implies the possibility of failure of DeiT encoding the inductive bias of its teacher architecture.

2.2.2. Internal representation similarity

To examine internal representation similarity driven by the architecture, we exploit central kernel analysis [15]. Figure 3 illustrates the representational similarity between CNN/RNN in comparison to DeiT and Transformer. We observe that CNN/RNN's feature extraction process differs from that of Transformer. DeiT has higher similarity to CNN/RNN compared to Transformer, but there is still a substantial difference to its teacher. Specifically, in the case of DeiT, only the early layers exhibit a significant dissimilarity between the representations, indicating that the early layers of DeiT failed to learn the CNN/RNN representation.

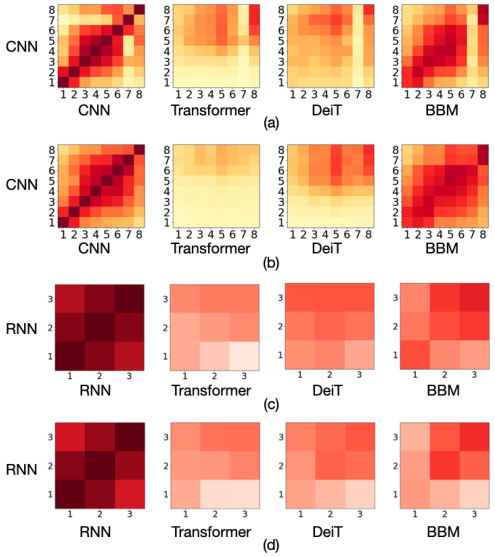

Figure 3: The representational similarity between the CNN/RNN and the transformers. Axes of each matrix represent the order of blocks. (a), (b), (c), and (d) are similarity in Physionet 2021, Hangzhou, Physionet 2019, and eICU dataset.

2.3. Self-attention analysis

Suppose the inductive bias of the teacher(CNN/RNN) is appropriately transferred to the student(Transformer). In that case, the student's self-attention should display the pattern of the teacher, i.e., spatial/temporal invariance and locality (Figure 2(a)). However, the student's self-attention would not exhibit the pattern of the teacher if the inductive bias of the teacher is not appropriately transferred to the student (Figure 2(b)). Figure 4 depicts the averaged self-attention matrices in each block across all samples and heads. It is difficult to distinguish the pattern of DeiT distinct from Transformer. To elaborate, DeiT does not exhibit the characteristics that demonstrate the inductive bias of CNN/RNN, such as translational/temporal invariance or locality.

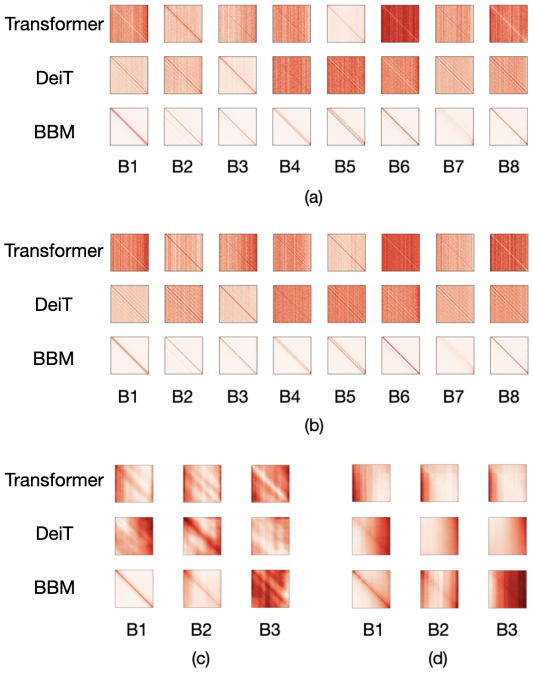

Figure 4: The self-attention of the transformers. Each (a), (b), (c), and (d) is self-attention matrix in Physionet 2021, Hangzhou, Physionet 2019, and eICU dataset. BN indicates the self-attention matrix at N th layer.

2.4. Discussion

The examination reveals that using DeiT, the teacher's inductive bias is not well transferred to the student, which is the case of Figure 2(b). There could be several reasons why DeiT works with ImageNet but not with our dataset. The first possibility is the size of the dataset. In the case of ImageNet, large data of 1M is sufficient to transfer inductive bias via KD. However, the size of

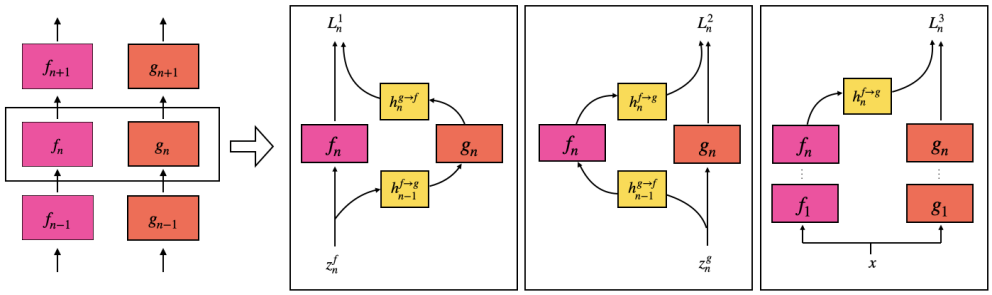

Figure 5: Pink and orange boxes are blocks of the student and the teacher, respectively. The yellow box is a dimension transformation layer. From the left to right, the panels present Equation 2, 3, and 4, respectively.

the data we utilized is only 8 percent of ImageNet, so it may be challenging to transfer inductive bias via KD. The second possibility is the number of classes. ImageNet consists of one thousand classes, whereas the dataset we utilized consists of twenty-six for Physionet 2021 and two classes for Physionet 2019. With this respect, DeiT may not work with our dataset because distributions obtained from our dataset contain less information than distributions obtained from ImageNet. Based on this, we believe the problem can be alleviated if the student is provided with more information to encode inductive bias.

3. Better solution for transferring inductive bias

3.1. Feature-based knowledge distillation

The knowledge distillation utilized in the DeiT is a type of response-based knowledge distillation that distills knowledge using the model's output. In contrast to the previous works, we impose a stronger signal by using feature-based knowledge distillation to enforce the student network to learn the teacher's inductive bias. Additionally, knowledge distillation is performed on feature maps in order to transfer spatial information from the teacher to the student effectively.

First, we divide the teacher (g) and student (f) into the same number of blocks and then perform the knowledge distillation between corresponding blocks (f_n, g_n) of the teacher and the student. Since features transverse multiple layers, the dimension of it varies. For example, in the case of CNN, the pooling operation and convolution with stride change the dimensions with temporal direction, and the convolution operation change also increases the dimension of the feature. Because of this, the dimension of features used for knowledge distillation can vary. To solve the problem, we introduce a transformation function (h) that transforms each dimension to be identical. This function resizes the feature's dimensions along the temporal axis and projects them along the depth axis.

$$h^{f\to g}(z) = I(z)\, W \tag{1}$$

where $h(\cdot) := \mathbb{R}^{t\times d} \to \mathbb{R}^{t'\times d'}$ consist of two-layer: $I(\cdot) := \mathbb{R}^{t\times d} \to \mathbb{R}^{t'\times d}$ represents the resize along the temporal axis, and $W \in \mathbb{R}^{d\times d'}$ is linear transformation along the depth axis.

With a transformation function, we match and train each block of the teacher and the student (f_n, g_n) to be similar as illustrated in Figure 5. In addition to matching between blocks of the teacher and the student, we also perform knowledge distillation between the output of the successive composition of blocks of the teacher and the student $(f_n \circ \cdots \circ f_1, g_n \circ \cdots \circ g_1)$. Each loss function term is formulated as follows.

$$\mathcal{L}_n^1 = \sum_{t,d} \left|\left| f_n(z_{n-1}^f) - (h_n^{g\to f} \circ g_n \circ h_{n-1}^{f\to g})(z_{n-1}^f) \right|\right|_2^2 \tag{2}$$

$$\mathcal{L}_n^2 = \sum_{t,d} \left|\left| g_n(z_{n-1}^g) - (h_n^{f\to g} \circ f_n \circ h_{n-1}^{g\to f})(z_{n-1}^g) \right|\right|_2^2 \tag{3}$$

$$\mathcal{L}_n^3 = \sum_{t,d} \left|\left| (h_n^{f\to g} \circ f_n \circ \cdots \circ f_1)(x) - (g_n \circ \cdots \circ g_1)(x) \right|\right|_2^2 \tag{4}$$

Incorporating all, the loss function used in transferring the inductive bias is $\mathcal{L}_{\mathcal{BBM}} = \sum_n \left(\mathcal{L}_n^1 + \mathcal{L}_n^2 + \mathcal{L}_n^3\right)$. We refer to the proposed method as block-by-block matching (BBM) because it performs knowledge distillation by matching each block of the teacher and the student. Using BBM method, the final loss function used for training is as follows: $\mathcal{L} = \mathcal{L}_{\mathcal{CLS}} + \lambda \mathcal{L}_{\mathcal{BBM}}$, where L_{CLS} indicates the loss function for classification with cross entropy and λ is weight term for BBM.

3.2. Results

3.2.1. Details of experiments

We divide Transformer and CNN into four blocks in the ECG experiment, respectively. Each network's blocks

are divided equally, so each ResNet block contains four sub-blocks, and each transformer block contains two sub-blocks. Transformer and RNN are divided into three blocks for the EHR experiment, with each block containing one block of Transformer and one layer of LSTM, respectively.

3.2.2. Result

Table 2 shows the performance of the proposed method against other methods in Physionet 2021 and 2019. In Physionet 2021, there is a significant gap between Transformer and CNN. DeiT is unable to close this gap, but our approach not only closes the gap but also outperforms CNN. A similar trend is observed in Physionet 2019.

Table 2
Generalization performance of existing methods and proposed method. The teacher network stands for CNN in ECG dataset and RNN in EHR dataset.

	ECG		EHR	
	F-1	P-21	F-1	P-19
Transformer	0.4959	0.6070	0.1579	0.1876
Teacher	0.5890	0.6895	0.1526	0.2528
DeiT	0.5322	0.6571	0.1609	0.2177
BBM	**0.6037**	**0.7026**	**0.1610**	**0.2619**

3.2.3. Evaluation on inductive bias transfer

As shown in Figure 1 and 3, BBM demonstrates higher similarity in representation with its teacher. In addition, as demonstrated in Figure 4, we observe that the self-attention matrix of BBM successfully encodes its teacher's inductive bias, such as spatial/temporal invariance or locality in the self-attention analysis. These prove that the proposed method encodes the inductive bias of its teacher successfully.

3.3. Ablation study

We viewed the network as composite functions, performing knowledge distillation on each function. Here, the question of the optimal number of blocks naturally arises. We perform experiments with varying the number of blocks to answer this question. As shown in Table 3, the performance increases as the number of blocks increases.

Table 3
Ablation study on the effect of the number of blocks. Each row indicate the number of blocks used for function matching. Block 1 uses the entire encoder as the single block.

	ECG		EHR	
	F-1	P-21	F-1	P-19
BBM-block1	0.5938	0.7001	0.1617	0.2066
BBM-block2	0.6027	0.7015	-	-
BBM-block3	-	-	0.1610	0.2619
BBM-block4	0.6037	0.7026	-	-

4. Conclusion

We show the limitation of DeiT on the transfer of inductive bias and demonstrate that this issue can be resolved using feature-based knowledge distillation. Through experimental studies in medical data, we demonstrate that our method consistently outperforms existing methods as well as the strong inductive bias models. Additionally, an extensive analysis verifies that the proposed method transfers meaningful inductive bias to transformers. Many studies focus on transferring the inductive bias into Transformer on ImageNet. However, there is insufficient analysis of other real-world data with different properties to ImageNet. We expect our study will help bridge the gap between research on ImageNet and real-world data.

References

[1] A. Goyal, Y. Bengio, Inductive biases for deep learning of higher-level cognition, arXiv preprint arXiv:2011.15091 (2020).

[2] S. Abnar, M. Dehghani, W. Zuidema, Transferring inductive biases through knowledge distillation, arXiv preprint arXiv:2006.00555 (2020).

[3] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International Conference on Machine Learning, PMLR, 2021, pp. 10347–10357.

[4] M. A. Reyna, N. Sadr, E. A. P. Alday, A. Gu, A. J. Shah, C. Robichaux, A. B. Rad, A. Elola, S. Seyedi, S. Ansari, et al., Will two do? varying dimensions in electrocardiography: The physionet/computing in cardiology challenge 2021, Computing in Cardiology 48 (2021) 1–4.

[5] M. A. Reyna, C. Josef, S. Seyedi, R. Jeter, S. P. Shashikumar, M. B. Westover, A. Sharma, S. Nemati, G. D. Clifford, Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019, in: 2019 Computing in Cardiology (CinC), IEEE, 2019, pp. Page–1.

[6] Alibaba-Cloud, Hefei high-tech cup, ecg human-machine intelligence competition-prediction of abnormal ecg events, 2019. URL: https://tianchi.aliyun.com/competition/entrance/231754/introduction.

[7] T. J. Pollard, A. E. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, O. Badawi, The eicu collaborative research

database, a freely available multi-center database for critical care research, Scientific data 5 (2018) 1–13.
[8] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, A. Y. Ng, Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network, Nature medicine 25 (2019) 65–69.
[9] J. Wang, B. Peng, X. Zhang, Using a stacked residual lstm model for sentiment intensity prediction, Neurocomputing 322 (2018) 93–101.
[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).
[12] E. D. Cubuk, B. Zoph, J. Shlens, Q. V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 702–703.
[13] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, A. Talwalkar, Massively parallel hyperparameter tuning (2018).
[14] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, et al., Ray: A distributed framework for emerging {AI} applications, in: 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18), 2018, pp. 561–577.
[15] S. Kornblith, M. Norouzi, H. Lee, G. Hinton, Similarity of neural network representations revisited, in: International Conference on Machine Learning, PMLR, 2019, pp. 3519–3529.

A. Implementation details

All the hyperparamters in experiments are chosen based on extensive hyperparameter search, which is performed using asynchronous successive halving algorithm. The search space and selected hyperparameters are described in the following.

A.1. Experiments on Physionet2021

A.1.1. Convolution network

The total of 100 search space is explored for maximum epoch of 100 with early-stopping rate of 0.5 every 20 epochs. Learning rate $\in [0.00001, 0.01]$ and weight decay $\in [0.00001, 0.1]$ are sampled from log-uniform distribution. And dropout $\in [0, 0.3]$, rand-augment number $\in [1, 5]$, rand-augment intensity $\in [0, 1]$ are sampled from quantified uniform distribution with the interval of 0.05, 1, and 0.1, respectively [12]. The chosen set of hyperparameters are 0.0003552 for learning rate, 0.00002430 for weight decay, 0.1 for dropout, and $3/0.7$ for rang-augment number/intensity.

A.1.2. Transformer trained from the scratch

Experiment setting is identical to convolution network, and the selected set of hyperparameters are 0.0002331 for learning rate, 0.00001312 for weight decay, 0.15 for dropout, and $3/0.8$ for rang-augment number/intensity.

A.1.3. DeiT

For DeiT, We performed hard-label distillation as described in equation (3) of [3]. In the search space, we added loss ratio between classification token and knowledge distillation token λ. Asλ is closer to 0, the ratio of knowledge distillation in loss increases. The other setting is identical to the setting in the convolution network. The selected set of hyperparameters are 0.0002517 for learning rate, 0.00002253 for weight decay, 0.15 for dropout, $3/0.5$ for rang-augment number/intensity, and 0.6 for knowledge distillation loss ratio.

A.1.4. BBM: Knowledge distillation

The total of 30 search space is explored for maximum epoch of 500 with early-stopping rate of 0.5 every 30 epochs. Learning rate $\in [0.0001, 0.1]$ and weight decay $\in [0.00001, 0.1]$ are sampled from log-uniform distribution. And dropout $\in [0, 0.3]$, rand-augment number $\in [3, 5]$, rand-augment intensity $\in [0.5, 1]$ are sampled from quantified uniform distribution with the interval of 0.05, 1, and 0.1, respectively. The chosen set of hyperparameters are 0.0007713 for learning rate, 0.03116 for weight decay, 0.1 for dropout, and $3/0.9$ for rang-augment number/intensity.

A.2. Experiments on Physionet2019

A.2.1. Recurrent network

The total of 100 search space is explored for maximum epoch of 100 with early-stopping rate of 0.5 every 20 epochs. Learning rate $\in [0.0001, 0.01]$ and weight decay

$\in [0.00001, 0.1]$ are sampled from log-uniform distribution. And dropout $\in [0, 0.3]$ is sampled from quantified uniform distribution with the interval of 0.05. Hidden dimension of hidden unit is sampled from $\in \{128, 256, 512\}$. The chosen set of hyperparameters are 0.0007194 for learning rate, 0.00001238 for weight decay, 0.1 for dropout, and 256 for hidden units.

A.2.2. Transformer trained from the scratch

For the transformer, the number of head $\in \{4, 8\}$, the dimension of the model $\in \{128, 256, 512\}$, the dimension of transformed network in the feed forward layer $\in \{128, 256, 512\}$ are randomly sampled. Other settings are identical to the recurrent network. The chosen set of hyperparameters are 0.0008136 for learning rate, 0.00001523 for weight decay, 0.25 for dropout. For hyperparameters of transformer architecture, each hidden unit, model, and head is 128, 512, and 8.

A.2.3. DeiT

For the hyperparameters related to transformer's architecture, We used the hyperparameter set selected in the transformer trained from the scratch. We performed hard-label distillation as the experiment in Physionet2021. In the search space, we added loss ratio between classification token and knowledge distillation token. The chosen set of hyperparameters are 0.0001554 for learning rate, 0.0001728 for weight decay, 0.05 for dropout, and 0.2 for the the ratio of knowledge distillation.

A.2.4. BBM: Knowledge distillation

The architecture selected in transformer trained from the scratch is used. The total of 30 search space is explored for maximum epoch of 500 with early-stopping rate of 0.5, every 30 epochs. Learning rate $\in [0.0001, 0.1]$ and weight decay $\in [0.00001, 0.1]$ are chosen with log-uniform distribution. And drop out $\in [0, 0.3]$ is sampled from quantified uniform distribution with the interval of 0.05. The chosen set of hyperparameters are 0.002099 for learning rate, 0.0001718 for weight decay, and 0.05 for dropout.