AutoML: Neural Architecture Search Project

Main Project V

2021-2022

AutoML Project Summary

- After developing DeepECG project, I realized that, in the field of ECG, task-specific design requires high levels of expertise and experience, leading to performance variation per different designs.
- In my experience, task-specific design processes are cumbersome and their meticulous configurations rarely translate to other data or tasks. In other words, I should always perform an iterative trial-and-error process to build the whole training process.
- I thought that even for other expert researchers, experimenting with numerous combinations every time is an inconvenient and time-consuming task.
- Therefore, I began this project to reduce the specialized knowledge required for designing neural networks.
- I proposed an end-to-end automated ML framework that automatically found optimal frameworks from preprocessing to neural networks at a single training, which showed similar results tuned by experts.
- This experience enabled me to discover the necessity of efficiency when designing ML models.

AutoML Project Summary

- **Contribution**
  - Building AutoML system based on reinforcement learning for time-series data
- **Development**
  - Python, pytorch
- **Difficulty of Project**
  - Search space design that suited for time-series data
  - Controller development according to search spaces expansion
- **Achievement**
  - Automatically construct the task-specific pipeline from preprocessing to architecture
  - Submitted to ICASSP 2023 conference

Input & Output definition

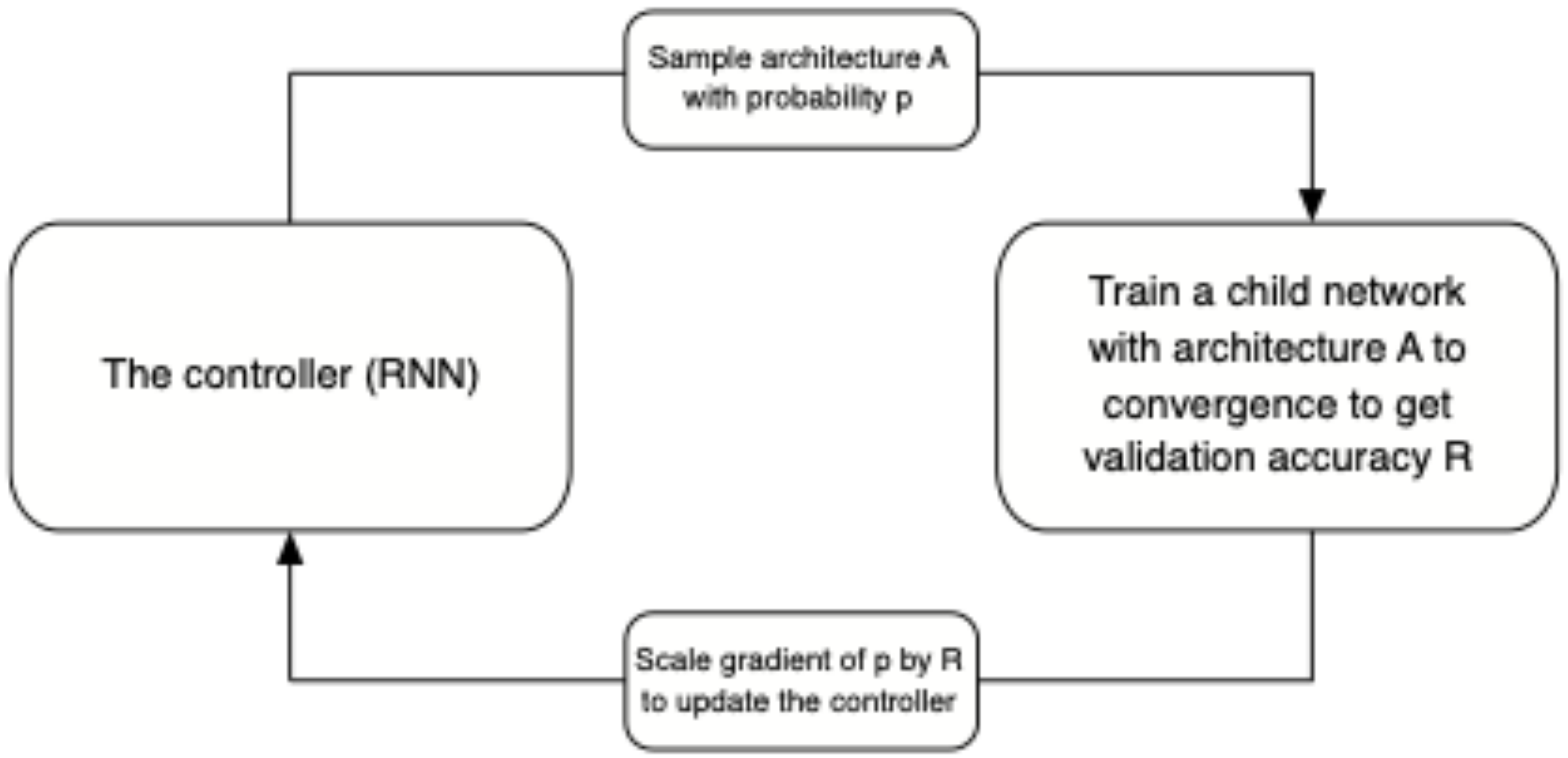
- I: Time-series multi-dimensional data (ECG data)
- O: Heart diseases (Multi-label classification)

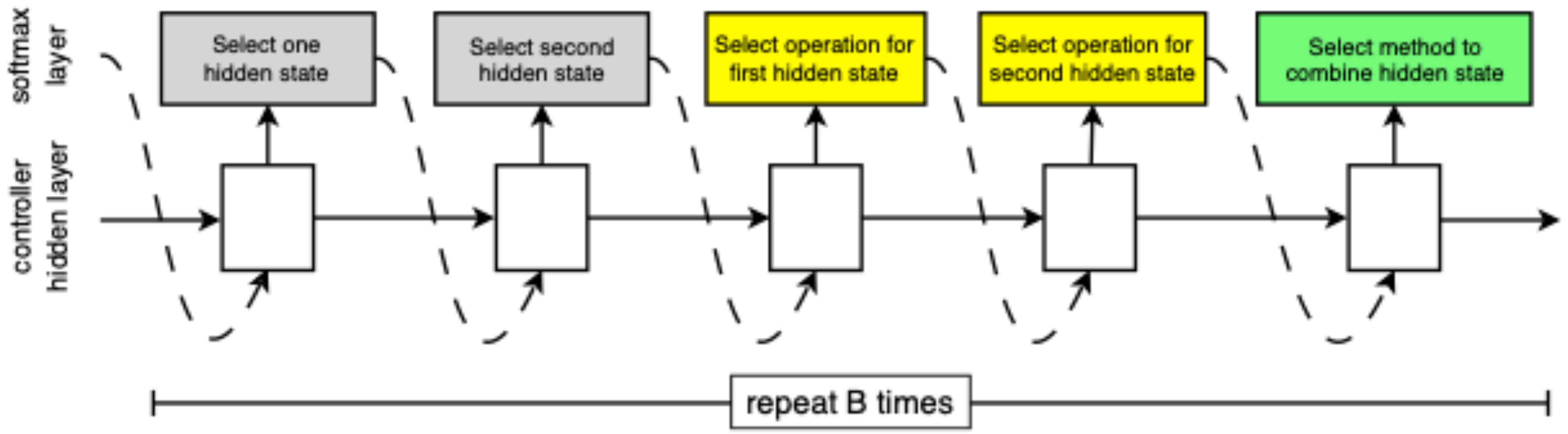
Motivation

- Requiring an iterative trial-and-error process to build the whole training process (ranging from preprocessing to architecture), which is an inconvenient and time-consuming task

AutoML Project
Reference Papar: ENAS 2018, google

- I borrowed the ENAS concept (google 2018) which utilized reinforcement learning concept
- ENAS optimized the whole pipeline based on two-step optimization
  - **Child models** (ω) update through Monte Carlo Policy Gradient: $\nabla_\omega \mathbb{E}_{\mathbf{m}\sim\pi(\mathbf{m};\theta)}\left[\mathcal{L}(\mathbf{m};\omega)\right] \approx \frac{1}{M}\sum_{i=1}^{M}\nabla_\omega \mathcal{L}(\mathbf{m}_i,\omega)$
  - **Controller** (θ) updates through policy gradient (Maximizing the expected reward $\mathrm{R}(\mathrm{m},\omega)$) : $\mathbb{E}_{\mathrm{m}\sim\pi(\mathrm{m};\theta)}[\mathcal{R}(\mathrm{m},\omega)]$





NasNet (google)

AutoML Project
Reference Papar: ENAS 2018, google

VUNO®

- **Child models** (ω) through Monte Carlo Policy Gradient Update: $$\nabla_\omega \mathbb{E}_{\mathbf{m}\sim\pi(\mathbf{m};\theta)}\left[\mathcal{L}(\mathbf{m};\omega)\right] \approx \frac{1}{M}\sum_{i=1}^{M}\nabla_\omega \mathcal{L}(\mathbf{m}_i,\omega)$$
  - Optimizing child models by using sum of cross entropy loss at given m sampled models (based on weight sharing)
- **Controller** (θ) through policy gradient (Maximizing the expected reward $\mathrm{R}(\mathrm{m},\omega)$) : $$\mathbb{E}_{\mathrm{m}\sim\pi(\mathrm{m};\theta)}[\mathcal{R}(\mathrm{m},\omega)]$$
  - Maximizing the expectation of the reward given the policy of agent
  - In fact, the purpose of reinforcement learning is to let the agent find the best behavior strategy that have the optimal reward
  - The agent's policy designs behavior strategies, and optimal reward (optimal trajectories) was derived based on policy gradient
  - Here, policy gradient is based on REINFORCE approach to obtain best behavior strategy
    - REINFORCE (Action-Reward): Calculate each state's output while finishing episodes and memorizing given rewards
      - On current policy (i.e., m models) -> Trajectory probabilities (encoder-decoder output each step) * Sum of all rewards
      - If reward is high, probability increases to get good trajectory

AutoML Project Contribution

- Despite AutoML success in several fields, these methods are difficult to directly apply to the electrocardiography (ECG) fields that morphological and temporal characteristics of the signal should be essentially considered.

- Furthermore, until now, there are scarce automated learning methods that take into consideration the entire framework ranging from proper search spaces to suitable automated learning processes in the ECG field.

- Here, I newly designed "Macro" search spaces for considering time-series characteristics
  - Preprocessing search space designs
    - Input sizes, Normalization, data augmentation
  - Architecture search space designs
    - CNN kernels, strides, etc.
  - Other strategies for search space designs
    - Layer skip connection, Layer activation functions

- Designing our modified controller which samples ranging from preprocessing to architecture (Next page)

AutoML Project Contribution

- I designed our modified controller which samples ranging from preprocessing to architecture while providing stable training strategies
  - Prior to this section, I determined many search spaces by considering ECG characteristics. Thus, the number of search spaces is expanded. The issue of this is that the construction of the controller should be redesigned according to increasing the length of search spaces.
  - I hypothesized it is important to make the controller understandable which the search space is designed at current step. This is because I believed that the property between preprocessing and architecture sampling is slightly different. Thus, I distinguish each step by adding additional clues
  - Plus, when it comes to the metric in the field of biosignals, the ROC value is mainly used as the main metric. However, unfortunately, calculating the ROC from a mini-batch will be unstable, and this unstable performance may even badly affect the learning of the controller. This indicates that it is difficult to use the ROC metric as a reward function to train the controller, so I firstly utilized another metric for training.

AutoML Project
Results

- Finally, I found the entire pipeline from preprocessing to architecture structure like the right figure.
- I observed the robust classification results of the proposed method on three public datasets compared to the classification performance tuned by experts.

ENAS(google)

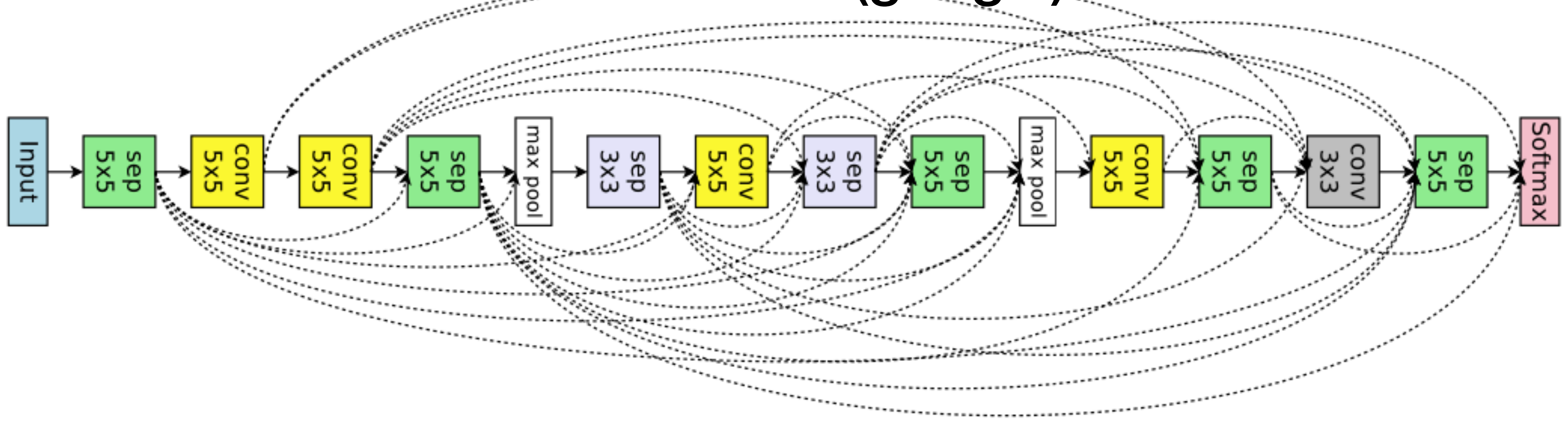


Figure 7. ENAS's discovered network from the macro search space for image classification.

Table 1. Performance Comparison of Our NAS Method and Other non-AutoML Methods

Dataset	Model	AUROC						
		Normal	AFIB	1AVB	LBBB	PAC	PVC	other labels
Shaoxing (5-fold)	[2] (Physionet 1st rank, tuned by experts)	98.42	99.25	94.60	99.29	89.01	96.01	96.51
	[3] (Physionet 2nd rank, tuned by experts)	98.74	99.39	94.79	98.62	92.97	95.97	97.38
	N2NAS (AutoML)	97.85	98.70	96.22	99.02	92.41	93.62	94.50
Physionet (5-fold)	[2] (Physionet 1st rank, tuned by experts)	93.43	91.90	94.62	96.55	84.78	91.95	75.38
	[3] (Physionet 2nd rank, tuned by experts)	92.30	94.62	96.26	96.57	85.56	93.93	78.33
	N2NAS (AutoML)	95.09	96.28	95.39	97.09	90.81	96.76	92.42
Lancet (Testset)	[2] (Physionet 1st rank, tuned by experts)	99.20	96.42	95.72	95.09	84.10	96.84	82.52
	[3] (Physionet 2nd rank, tuned by experts)	98.89	95.06	96.34	95.88	84.13	97.01	82.70
	N2NAS (AutoML)	99.04	96.23	95.94	96.17	91.06	97.14	78.05

Submitted in ICASSP 2023

AutoML Project Conclusion

- In this project, I proposed a novel AutoML framework for multi-label ECG classification.
    - In particular, I devised not only diverse search spaces based on the temporal characteristics of the ECG signal that can utilize inherent ECG properties but also integrate the development strategies ranging from preprocessing to architecture sampling through a modified controller
- I learned that unexpected CNN architecture designs were achieved from an automated framework.
- In addition, I observed that similar NAS structures were extracted in the same dataset, but different NAS structures were selected according to a different dataset.
- Also, unnecessary search spaces (e.g., 500 Hz sampling) were found.
- From this perspective, this framework will be one of the convenient and efficient tools to automatically design a deep learning framework for a future company project.

N2NAS: AN END-TO-END AUTOML FRAMEWORK FOR MULTI-LABEL ECG CLASSIFICATION

Oyeon Kwon and Yeha Lee

VUNO Inc.

ABSTRACT

An automated learning process to reduce specialized knowledge in designing neural networks has been actively proposed in the field of deep learning. Despite their success in several fields, these methods are difficult to directly apply to the electrocardiography (ECG) fields that morphological and temporal characteristics of the signal should be essentially considered. Furthermore, until now, there are scarce automated learning methods that take into consideration the entire framework ranging from proper search spaces to suitable automated learning processes in the ECG field. In this paper, we propose an end-to-end automated machine learning (N2NAS) framework for multi-label ECG classification. In this framework, we devised not only diverse search spaces

even the same task [2, 3]. However, these processes are cumbersome and their meticulous configurations rarely translate to another data or task. Therefore, researchers should always perform an iterative trial-and-error process to build the whole training process. Even for expertise researchers, experimenting with numerous combinations every time is an inconvenient and time-consuming task, and it will also be a big hurdle for beginners who enter this field for the first time.

In this paper, we propose an end-to-end AutoML (N2NAS) framework for multi-label ECG classification. The main contribution is to build the entire pipeline which can automatically sample search spaces suitable for the ECG's inherent property by combining the preprocessing and architecture.

Submitted in ICASSP 2023

First author