Han YAN, Chongquan ZHONG, Yuhu WU, Liyong ZHANG, Wei LU, 2023. A hybrid-model optimization algorithm based on the Gaussian process and particle swarm optimization for mixed-variable CNN hyperparameter automatic search. *Frontiers of Information Technology & Electronic Engineering*, 24(11):1557-1573. https://doi.org/10.1631/FITEE.2200515

A hybrid-model optimization algorithm based on the Gaussian process and particle swarm optimization for mixed-variable CNN hyperparameter automatic search

Key words: Convolutional neural network; Gaussian process; Hybrid model; Hyperparameter optimization; Mixed-variable; Particle swarm optimization

Corresponding author: Wei LU
E-mail: luwei@dlut.edu.cn
ORCID: https://orcid.org/0000-0002-5775-1222

Motivation

1. The convolutional neural network (CNN) hyperparameter types are different, and such mixed-variable characteristics are proved difficult in efficient search space encoding.
2. For traditional optimization algorithms (OAs), CNNs are evaluated by the fitness function through assessment criteria based on training, which increases the cost of fitness evaluation (FE) and damages the efficiency of OAs.
3. Considering the large number of CNN hyperparameters, it is still necessary to research how to accelerate the convergence for FE and ensure model performance after search.

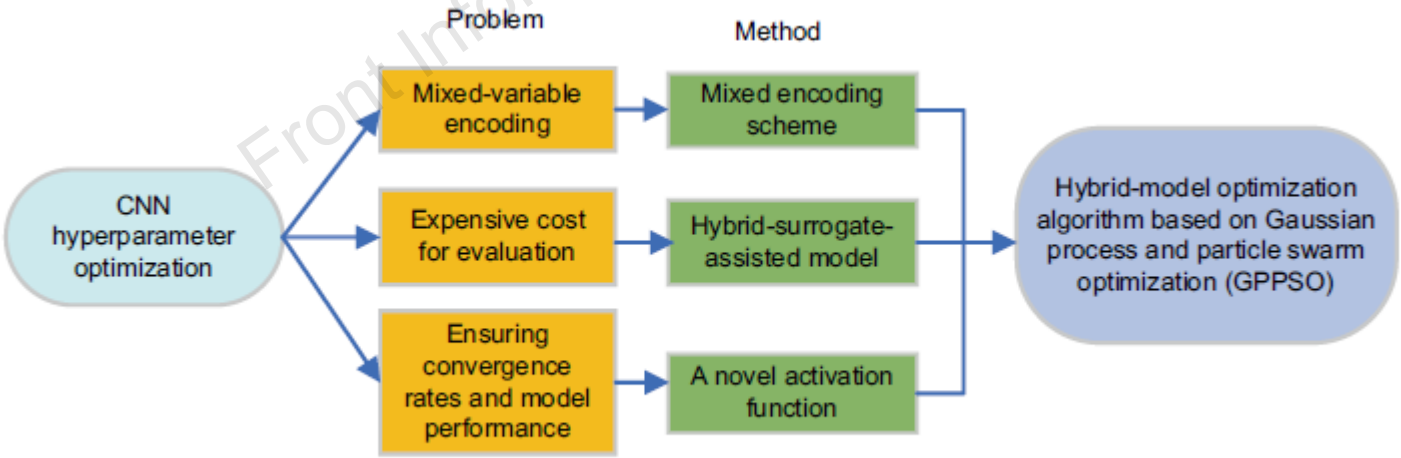

Fig. 1 Major challenges and contributions of GPPSO

Method

A novel encoding strategy can efficiently deal with the mixed-variable difficulty of CNN hyperparameters.

Table 1 Settings of mixed-variable encoding for CNN hyperparameters

Type	Name	Available choice	Search space	Initialization space
Continuous variable	Number of kernels in Conv layers	$\{1, 2, \cdots, +\infty\}$	$[1, +\infty)$	$[8, 128]$
	Number of neurons in FC layers	$\{1, 2, \cdots, +\infty\}$	$[1, +\infty)$	$[64, 512]$
	Value of the initial learning rate		$(0, 1)$	$(0, 1)$
	Value of the dropout rate		$[0, 1)$	$[0, 1)$
Discrete variable	Kernel size of Conv layers	$\{3 \times 3, 5 \times 5, 7 \times 7\}$	$\{0, 1, 2\}$	$\{0, 1, 2\}$
	Type of activation function	{ReLU, Sigmoid, Tanh, Ta-ReLU}	$\{0, 1, 2, 3\}$	$\{0, 1, 2, 3\}$
	Type of pooling layer	{Max pooling, average pooling}	$\{0, 1\}$	$\{0, 1\}$

CNN: convolutional neural network; Conv: convolutional; FC: fully connected

For the integer variable in continuous variables, the encoding strategy is shown in Eq. (9):

$$N_{\mathrm{k}} = \lfloor X_{\mathrm{nk}} \rfloor, \quad X_{\mathrm{nk}} \in [1, +\infty), \tag{9}$$

Taking the example of discrete variable encoding for activation function, the specific encoding strategy is shown in Eq. (10):

$$\mathrm{af} = \begin{cases} \mathrm{ReLU}, & \lceil X_{\mathrm{af}} \rceil \in [0,\ 1), \\ \mathrm{Sigmoid}, & \lceil X_{\mathrm{af}} \rceil \in [1,\ 2), \\ \mathrm{Tanh}, & \lceil X_{\mathrm{af}} \rceil \in [2,\ 3), \\ \text{Ta-ReLU}, & \lceil X_{\mathrm{af}} \rceil \in [3,\ 4), \end{cases} \tag{10}$$

Method

A hybrid-surrogate-assisted (HSA) mode can deal with the expensive computational cost problem in the search process.

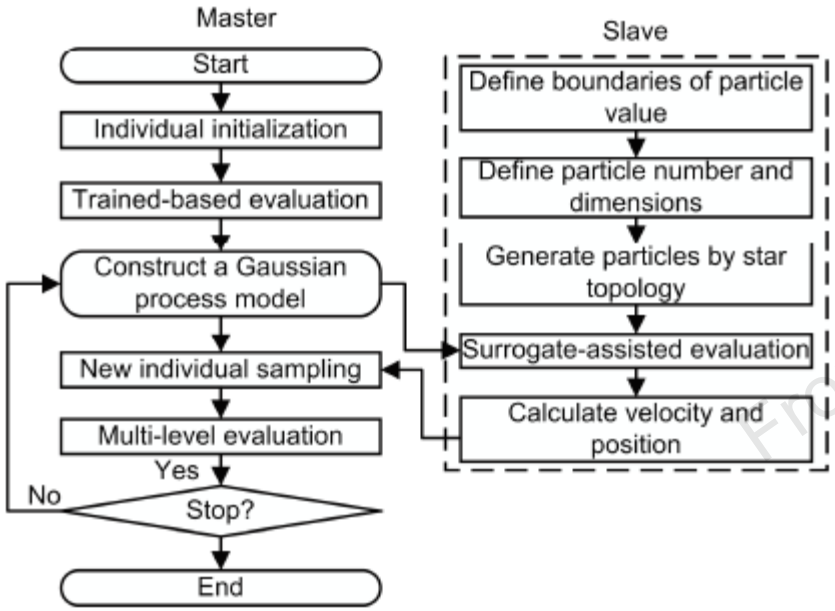

Fig. 4 General flowchart of the hybrid surrogate-assisted model

Algorithm 1 Multi-level evaluation strategy

Input: Dataset of training CNNs, $\mathcal{D}_{\text{train}}$;
dataset of evaluating CNNs, $\mathcal{D}_{\text{eval}}$;
architecture constructed by individuals, $\mathcal{P}_{\text{arch}}$;
fitness of individuals in the architecture, fit_{arch};
set of individuals to be evaluated, $\mathcal{P}$;
number of initial individuals, N_{I};
number of all individuals, N_{A};
number of training epochs, T
Output: Set of individuals evaluated, $\mathcal{P}_{\text{eval}}$;
the fitness set of individuals evaluated, fit_{eval}

begin
/* construct the initial GP model */
Initialize N_{I} individuals;
Obtain $\mathcal{P}_{\text{arch}}$ and fit_{arch} of the initial individuals by training;
$\text{GP}_{\text{I}} \leftarrow$ build an initial GP model with $\mathcal{P}_{\text{arch}}$ and fit_{arch};
/* first level evaluation by the GP model */
gfitness $\leftarrow$ fitness of $\mathcal{P}$ predicted by GP_{I};
afitness $\leftarrow$ the average fitness of fit_{arch};
$\mathcal{P}_{\text{eval}} \leftarrow$ empty set;
$\text{fit}_{\text{eval}} \leftarrow$ empty set;
ri $\leftarrow$ a random integer in $\{1, 2, \cdots, N_{\text{A}}\}$;
/* second level evaluation by training */
for each individual $\mathcal{P}_i$ in $\mathcal{P}$ **do**
  if gfitness > afitness or ri $== i$ **then**
    Net $\leftarrow$ build a candidate architecture according to the hyperparameters in $\mathcal{P}_i$;
    Net $\leftarrow$ train Net with T epochs on $\mathcal{D}_{\text{train}}$;
    accuracy $\leftarrow$ validate Net on $\mathcal{D}_{\text{eval}}$;
    $\text{fit}_{\text{eval}} \leftarrow \text{fit}_{\text{eval}} \cup \{\text{accuracy}\}$;
    $\mathcal{P}_{\text{eval}} \leftarrow \mathcal{P}_{\text{eval}} \cup P_i$;
    $\text{GP}_{\text{I}} \leftarrow$ evolve GP_{I} through fit_{eval} and $\mathcal{P}_{\text{eval}}$;
  end if
end for
end

Algorithm 2 Individual generation strategy

Input: The initial Gaussian process model, GP_{I};
number of particles, N;
fitness of individual i, fit_i;
dimension of particles, d;
number of generations, n;
number of iterations, T
Output: The generated individual to be evaluated, $\mathcal{P}_{\text{new}}$

begin:
/* initialize a group of particles with d dimensions corresponding to hyperparameters in CNNs */
Initialize a set of N particles as $\mathcal{P}$;
for each particle $\mathcal{P}_i$ in $\mathcal{P}$ **do**
  for each dimension d of $\mathcal{P}_i$ **do**
    Randomly initialize the particle position x_{id} within the given range;
    Randomly initialize the particle velocity v_{id} within the given range;
  end for
end for
/* search individuals for multi-level evaluation by PSO */
for $k = 1$ to T **do**
  for each individual $\mathcal{P}_i$ in $\mathcal{P}$ **do**
    $\text{fit}_i \leftarrow$calculated fitness by GP_{I} in Algorithm 1;
    if the fitness value is larger than pbest_{id} in history **then**
      $\text{pbest}_{id} \leftarrow \text{fit}_i$;
    end if
    $\text{gbest}_d \leftarrow$ the particle with the best fitness value;
    Calculate velocity v_{id} through Eq. (7);
    Calculate position x_{id} through Eq. (8);
  end for
end for
$\mathcal{P}_{\text{new}} \leftarrow x_{id}$ of the last n generations;
end

Method

A novel activation function (Ta-ReLU) can accelerate the convergence in the process of population evolution and improve the performance of the model after training.

$$\text{Ta-ReLU} = \begin{cases} x, & x < 0, \\ \alpha \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}, & x \geq 0. \end{cases} \quad (11)$$

Table 3 Comparisons with the basic algorithms of GPPSO on CIFAR-10 and CIFAR-100 datasets

Method	CIFAR-10					CIFAR-100				
	Test accuracy (%)	Training accuracy (%)	Number of parameters ($\times 10^6$)	Search time (min)	Training time (min)	Test accuracy (%)	Training accuracy (%)	Number of parameters ($\times 10^6$)	Search time (min)	Training time (min)
Manual (ResNet20)	92.25	98.71	0.38		116	68.96	90.78	0.40		111
BO	92.91	99.92	4.70	20	274	66.47	90.62	4.70	20	279
BO_ac	92.26	99.96	5.26	17	367	65.64	75.60	3.57	23	326
PSO	84.17	86.09	4.65	50	258	54.25	28.20	4.70	192	265
PSO_ac	74.57	74.75	6.17	150	361	51.77	54.77	4.74	212	296
GPPSO	91.85	99.95	4.77	45	270	65.91	94.00	4.74	33	330
GPPSO_ac	95.26	99.96	5.26	39	261	76.36	97.65	4.44	39	304

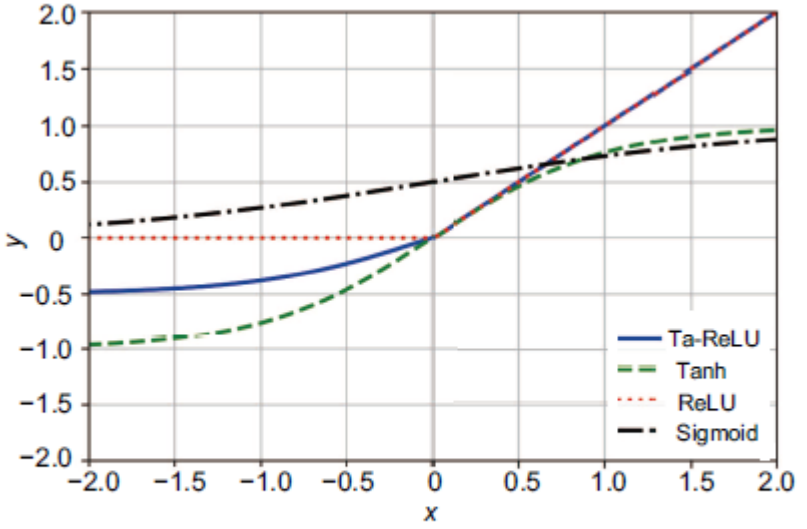

Fig. 5 Comparison of different activation functions

Conclusions

1. We designed a novel encoding strategy to deal with the mixed-variable problem of CNN hyperparameters.
2. We proposed a hybrid-surrogate-assisted (HSA) based on GP and PSO to save computational cost.
3. We suggested a novel activation function (Ta-ReLU) to improve the model performance and ensure convergence rate.

Han YAN received his MS degree in control theory and engineering from Dalian Jiaotong University, Dalian, China, in 2020. He is currently pursuing his PhD degree in control science and engineering with Dalian University of Technology, Dalian, China, under the supervision of Prof. Chongquan ZHONG. His research interests include image processing and pattern recognition.

Wei LU received his MS and PhD degrees in control theory and control engineering from Dalian University of Technology, Dalian, China, in 2004 and 2015, respectively.

In 2004, he joined Dalian University of Technology, where he is currently a professor with the School of Control Science and Engineering. His current research interests include computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data mining, and fuzzy intelligent systems. In the above areas, he has published more than 60 papers.