Research and Applications of DIVERSITY *in Ensemble Classification*

Yijun BIAN

Department of Computer Science and Technology
University of Science and Technology of China (USTC)
Hefei, Anhui 230027, P.R. China

26 August 2020

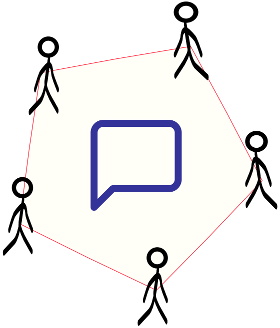
Overview

1. **Introduction**
   - Ensemble learning
   - Diversity
   - Ensemble pruning
2. **Relationship between diversity and ensemble performance in classification**
   - Error decomposition in ensemble classification
   - Relationship between it and ensemble performance
   - Utilising diversity to construct better ensembles
3. **Ensemble pruning based on objection maximisation with a general distributed framework**
   - Objection maximisation for ensemble pruning
   - Pruning algorithms
4. **Sub-architecture ensemble pruning in neural architecture search**
   - Sub-architecture ensemble pruning in NAS (*SAEP*)

Overview

1. Introduction
   - Research significance
   - Research contents
2. Relationship between diversity and ensemble performance in classification
3. Ensemble pruning based on objection maximisation with a general distributed framework
4. Sub-architecture ensemble pruning in neural architecture search

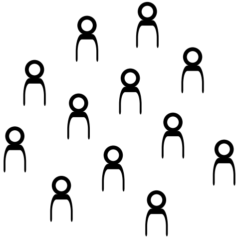
Example



brainstorm
benefit by mutual discussion

drift apart
where to go next? no idea

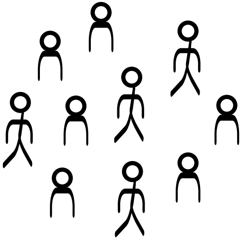
Ensemble learning



Ensemble learning

- Applications
  object recognition, object detection, object tracking
  fault diagnosis, malware detection, depression detection etc.

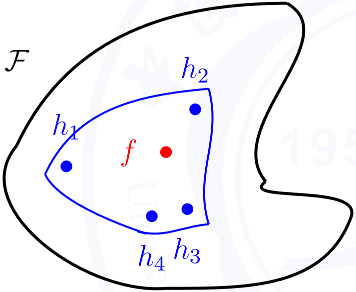
Ensemble learning



Ensemble learning

- Applications
- Catogories
  - Homogeneous ensembles
  - Heterogeneous ensembles

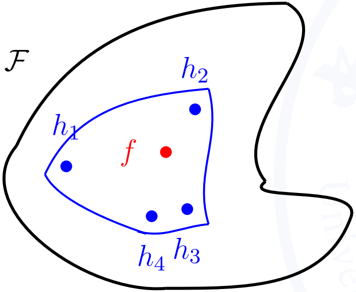
Ensemble learning



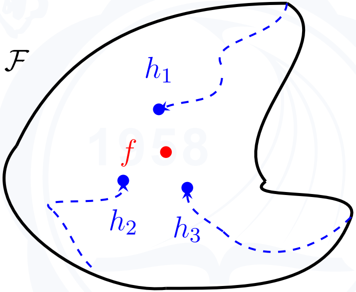
(a) Statistical

Figure 1.1. Three fundamental reasons why constructing good ensembles is often possible [1]

Ensemble learning



(a) Statistical



(b) Computational

Figure 1.1. Three fundamental reasons why constructing good ensembles is often possible [1]

Ensemble learning

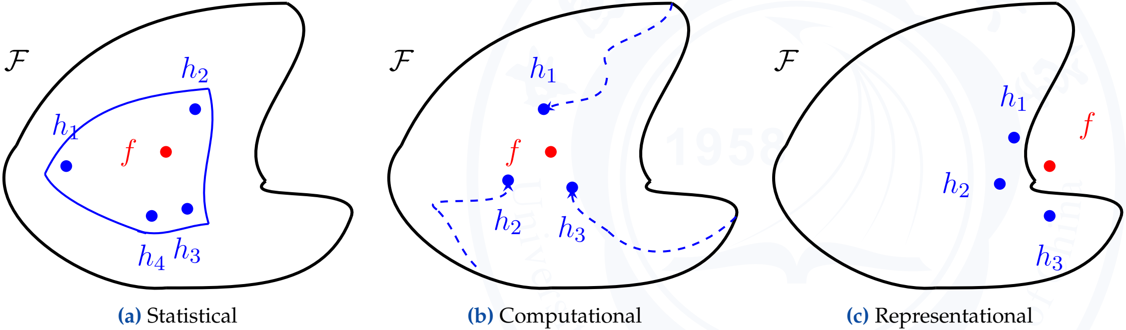
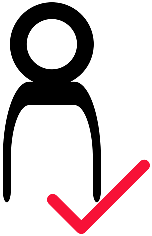


Figure 1.1. Three fundamental reasons why constructing good ensembles is often possible [1]

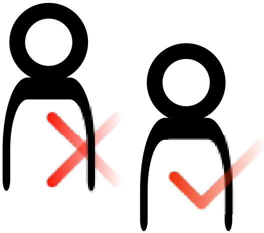
Ensemble learning



Ensemble learning

- Crucial elements
  - Accurate

Ensemble learning



Ensemble learning

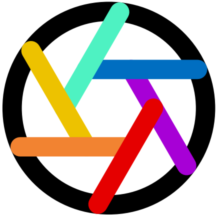
- Crucial elements
  - Accurate
  - Diverse

Ensemble learning

Ensemble learning

- Crucial elements
  - Accurate
  - Diverse
- **How to balance them?**
  Understanding diversity

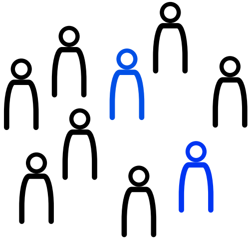
Diversity



Diversity

- Constructing ensembles
  - Diverse individual classifiers
  - Creating them implicitly or heuristically

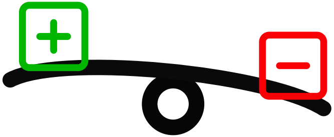
Diversity



Diversity

- Constructing ensembles
- Originated from
  - Error decomposition of regression ensembles

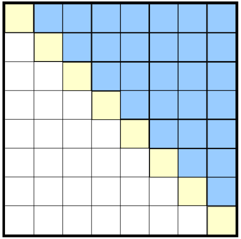
Diversity



Diversity

- Constructing ensembles
- Originated from
- Existing measures

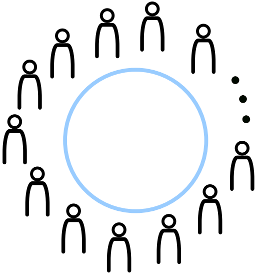
Diversity



Diversity

- Constructing ensembles
- Originated from
- Existing measures
  - Pairwise measures

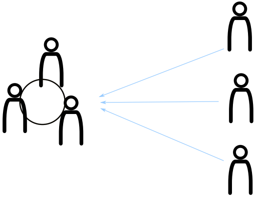
Diversity



Diversity

- Constructing ensembles
- Originated from
- Existing measures
  - Pairwise measures
  - Non-pairwise measures

Diversity



Diversity

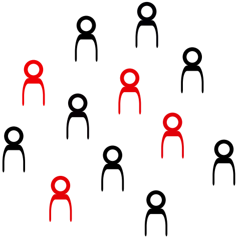
- Constructing ensembles
- Originated from
- Existing measures
  - Pairwise measures
  - Non-pairwise measures
  - Correlation penalty function, ambiguity

Diversity

Diversity

- Constructing ensembles
- Originated from
- Existing measures
- Relationship
- Utilisation

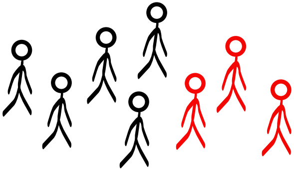
Ensemble pruning



Ensemble pruning

- Categories

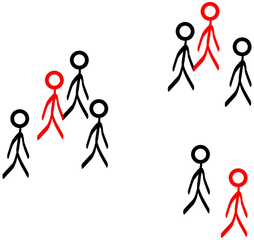
Ensemble pruning



Ensemble pruning

- Categories
  - Ranking-based

Ensemble pruning



Ensemble pruning

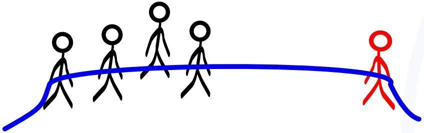
- Categories
  - Ranking-based
  - Clustering-based

Ensemble pruning

Ensemble pruning

- Categories
  - Ranking-based
  - Clustering-based
  - Optimisation-based

Ensemble pruning



Ensemble pruning

- Categories
  - Ranking-based
  - Clustering-based
  - Optimisation-based
- Centralised

Research outline

Research and Application of Diversity in Ensemble Classification

Ensemble Learning Area | **Other Areas**

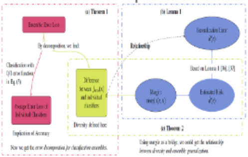
Main Challenge

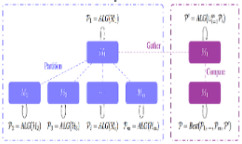
- The role that diversity plays in ensemble classifiers is not quite clear yet
- It is hard to balance diversity and accuracy because they conflict with each other
- They overlooked diversity, a key element in that area, while using ensemble methods

Motivation

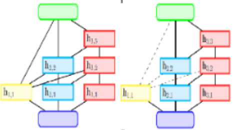
- To investigate when diversity helps in ensemble classification
- To balance them properly and accelerate the pruning process
- To bridge the gap of diversity in neural architecture search with ensemble methods

Technical Methodology





(a) The First Phase (b) The Second Phase

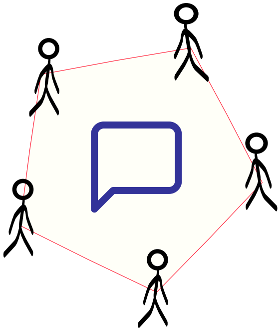


Research Content

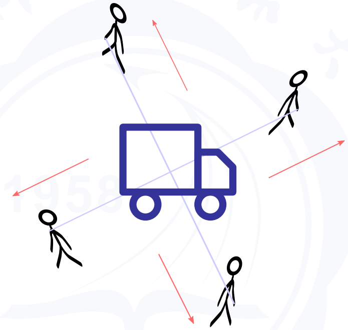
- Relationship between diversity and ensemble performance in classification ensembles
- Ensemble pruning based information entropy and a general distributed framework
- Sub-architecture ensemble pruning in neural architecture search

Overview

Background



brainstorm
benefit by mutual discussion



drift apart
where to go next? no idea

Methodology

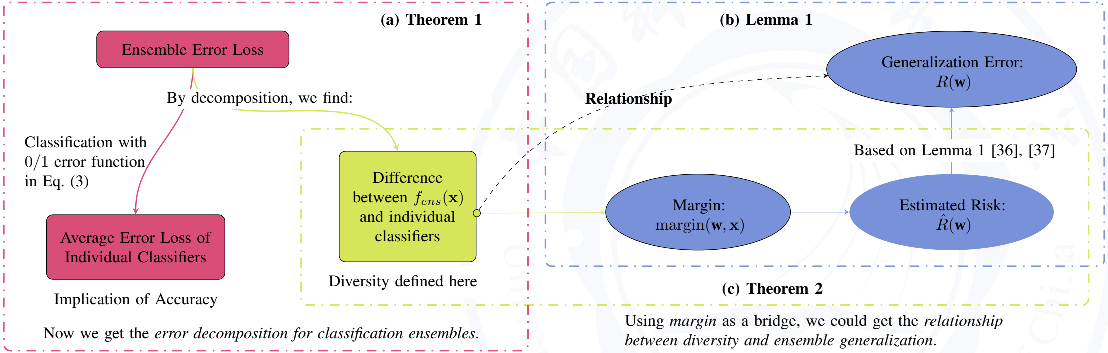


Figure 2.1. Illustration for the proposed methodology. (a) Illustration for the error decomposition for classification ensembles. (b) Illustration of Lemma 1 [2, 3]. (c) Illustration for the relationship between the proposed diversity and ensemble performance.

Error decomposition in ensemble classification

- Loss of the ensemble[1]

$$
\begin{aligned}
&\mathrm{Err}\left(h_{ens}(\boldsymbol{x})y\right) - \sum_{i=1}^{n} w_i \,\mathrm{Err}\left(h_i(\boldsymbol{x})y\right) \\
&= - \boxed{\frac{1}{2}\left(h_{ens}(\boldsymbol{x}) - \sum_{i=1}^{n} w_i h_i(\boldsymbol{x})\right) y}
\end{aligned}
\tag{2.1}
$$

[1] Ensemble classifier with weighted voting: $h_{ens}(\boldsymbol{x}) = \mathrm{sign}\left(\sum_{i=1}^{n} w_i h_i(\boldsymbol{x})\right)$
Margin of a classifier $h(\cdot)$ on one instance: $\mathrm{margin}(h, \boldsymbol{x}) = h(\boldsymbol{x})y$
Error function of a classifier $h(\cdot)$ on one instance $\boldsymbol{x}$: $\mathrm{Err}(h, \boldsymbol{x}) = -\frac{1}{2}(\mathrm{margin}(h, \boldsymbol{x}) - 1)$

Error decomposition in ensemble classification

- Loss of the ensemble
- Weighted loss of individual classifiers [1]

$$\mathrm{Err}\left(h_{ens}(\boldsymbol{x})y\right) - \sum_{i=1}^{n} w_i \,\mathrm{Err}\left(h_i(\boldsymbol{x})y\right) = -\boxed{\frac{1}{2}\left(h_{ens}(\boldsymbol{x}) - \sum_{i=1}^{n} w_i h_i(\boldsymbol{x})\right)y} \tag{2.1}$$

[1] Employed 0/1 error function of a classifier: If it classifies the instance correctly, $\mathrm{Err}(h, \boldsymbol{x}) = 0$; If it classifies the instance incorrectly, $\mathrm{Err}(h, \boldsymbol{x}) = -1$; Ties (i.e., $h(\boldsymbol{x})y = 0$) lead to $\mathrm{Err}(h, \boldsymbol{x}) = 0.5$.

Error decomposition in ensemble classification

- Loss of the ensemble
- Weighted loss of individual classifiers

$$\mathrm{Err}\left(h_{ens}(\boldsymbol{x})y\right) - \sum_{i=1}^{n} w_i \,\mathrm{Err}\left(h_i(\boldsymbol{x})y\right) = -\boxed{\frac{1}{2}\left(h_{ens}(\boldsymbol{x}) - \sum_{i=1}^{n} w_i h_i(\boldsymbol{x})\right) y} \tag{2.1}$$

- Difference between them

Error decomposition in ensemble classification

- Loss of the ensemble
- Weighted loss of individual classifiers

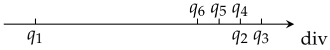
$$\mathrm{Err}\,(h_{ens}(\boldsymbol{x})y) - \sum_{i=1}^{n} w_i\,\mathrm{Err}\,(h_i(\boldsymbol{x})y)$$
$$= - \boxed{\frac{1}{2}\left(h_{ens}(\boldsymbol{x}) - \sum_{i=1}^{n} w_i h_i(\boldsymbol{x})\right)y} \tag{2.1}$$

- Difference between them
-

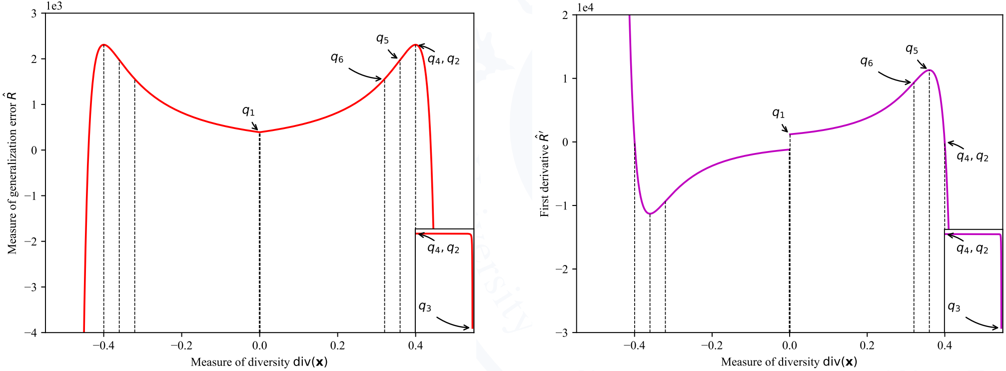
$$\mathrm{div}(h_{ens},\boldsymbol{x}) = \frac{1}{2}\,\mathrm{margin}(h_{ens},\boldsymbol{x}) - \frac{1}{2}\sum_{i=1}^{n} w_i \cdot \mathrm{margin}(h_i,\boldsymbol{x}) \tag{2.2}$$

NB. Diversity on one single instance

Relationship between it and ensemble performance



$$\hat{R}(\mathrm{div}(h_{ens}, \boldsymbol{x}))$$
$$\hat{R}'(q_2) = \hat{R}'(q_4) = 0$$
$$\hat{R}''(q_5) = 0$$
$$\hat{R}'''(q_6) = 0$$



(a) (b)

Figure 2.2. Illustration of the estimator of generalisation error and its first derivative, impacted by the proposed measure of diversity.

Relationship between it and ensemble performance

Estimated risk $\hat{R}(\boldsymbol{w})$ to reflect the upper bound of generalisation error $R(\boldsymbol{w})$, with the same variation tendency

$$R(\boldsymbol{w}) \leqslant \frac{2}{m}\left(\kappa(\boldsymbol{w}) \log_2\left(\frac{8em}{\kappa(\boldsymbol{w})}\right) \log_2(32m) + \log_2\left(\frac{2m}{\xi}\right)\right) , \tag{2.3}$$

and

$$\hat{R}(\boldsymbol{w}) = \left(\frac{8\delta}{\gamma(\boldsymbol{w})}\right)^2 \log_2\left(8em\left(\frac{\gamma(\boldsymbol{w})}{8\delta}\right)^2\right) , \tag{2.4}$$

$$\gamma(\boldsymbol{w}) = \min_{(\boldsymbol{x},y)\in\mathcal{D}} (1-2\varepsilon)(\lambda - 2\,\mathrm{div}(h_{ens},\boldsymbol{x})) , \tag{2.5}$$

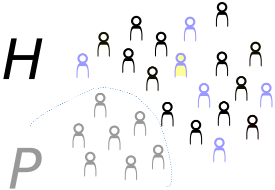
where

$$\lambda = \begin{cases} 1 , & \text{if } \mathrm{div}(h_{ens},\boldsymbol{x}) \in (0,\frac{1}{2}) ; \\ 0 , & \text{if } \mathrm{div}(h_{ens},\boldsymbol{x}) = 0 ; \\ -1 , & \text{if } \mathrm{div}(h_{ens},\boldsymbol{x}) \in (-\frac{1}{2},0) \end{cases} \tag{2.6}$$

Table 2.1. Monotone intervals. The first column is diversity $\mathrm{div}(h_{ens},\boldsymbol{x}^*)$. The second and the third columns are the estimated risk $\hat{R}(\boldsymbol{w})$ and its first derivative, respectively.

$\mathrm{div}(h_{ens},\boldsymbol{x}^*)$	$\hat{R}(\boldsymbol{w})$	$\hat{R}'(\boldsymbol{w})$	$\Delta\hat{R}$	$\Delta\hat{R}'$
$(-q_3,-q_2)$	$\nearrow$ convex	$\searrow$ concave	smaller	larger
$(-q_2,-q_5)$	$\searrow$ convex	$\searrow$ concave	smaller	larger
$(-q_5,-q_6)$	$\searrow$ concave	$\nearrow$ concave	larger	larger
$(-q_6,-q_1)$	$\searrow$ concave	$\nearrow$ convex	larger	smaller
(q_1,q_6)	$\nearrow$ concave	$\nearrow$ concave	larger	larger
(q_6,q_5)	$\nearrow$ concave	$\nearrow$ convex	larger	smaller
(q_5,q_2)	$\nearrow$ convex	$\searrow$ convex	smaller	smaller
(q_2,q_3)	$\searrow$ convex	$\searrow$ convex	smaller	smaller

Utilising diversity to construct better ensembles



Minimal margin

High accuracy

Algorithm 1. Ensemble pruning based on diversity (*EPBD*)

Input: Training set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_m, y_m)\}$, original ensemble $\mathcal{H} = \{h_1(\cdot), ..., h_n(\cdot)\}$

Output: Pruned sub-ensemble $\mathcal{P}$, meeting that $\mathcal{P} \subset \mathcal{H}$

$\mathcal{H} = \emptyset$;

repeat

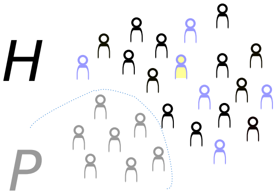
  Search for the specific data instance $(\boldsymbol{x}, y)$ which satisfies the search criterion (i.e., Eq. (2.5)) ;

  Sort the classifiers in $\mathcal{H}$ that classify this instance correctly in ascending order according to the accuracy performance.

  Move the top one $h(\cdot)$ in the previous step from $\mathcal{H}$ to $\mathcal{P}$

until The termination condition is satisfied.

Utilising diversity to construct better ensembles



High diversity

High accuracy

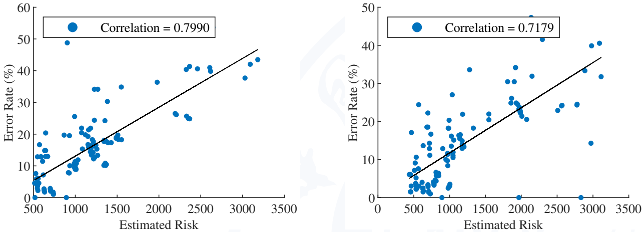
Algorithm 2. Ensemble pruning framework utilising the trade-off between accuracy and diversity (*FTAD*)

Input: Training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, original ensemble $\mathcal{H} = \{h_j(\cdot)\}_{j=1}^n$, arbitrary diversity measure $DIV(\cdot)$

Output: Pruned sub-ensemble $\mathcal{P}$, meeting that $\mathcal{P} \subset \mathcal{H}$

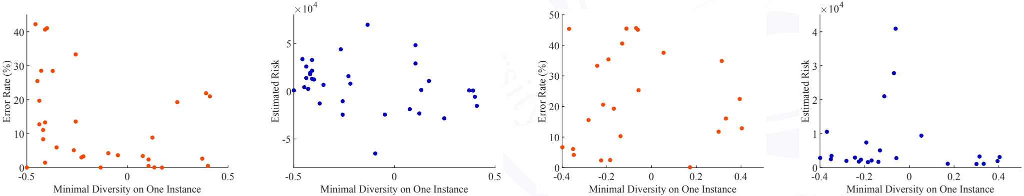
1: $\mathcal{H} = \emptyset$;
2: **repeat**
3: Compute the ensemble diversity on each data instance using the specified diversity measure $DIV(\cdot)$, and choose the one with the highest diversity .
4: Sort classifiers in $\mathcal{F}$ that classify this instance correctly in ascending order according to the accuracy performance.
5: Move the top one $h(\cdot)$ in the previous step from $\mathcal{H}$ to $\mathcal{P}$.
6: **until** The termination condition is satisfied.

Validating the proposed relationship



(a) **(b)**

Figure 2.3. Relationship of error rate and estimated risk calculated based on diversity for binary classification. Note that the bagging was used with NBs and LMs as individual classifiers, respectively.



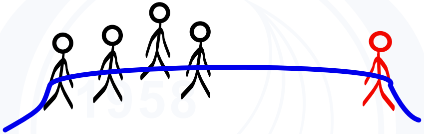
(a) **(b)** **(c)** **(d)**

Figure 2.4. Relationship of diversity and ensemble performance for binary classification. (a–b) Using bagging with DTs as individual classifiers; (c–d) Using AdaBoost with LMs as individual classifiers.

Overview

Background

Accurate vs. Diverse



Centralised

Objection maximisation for ensemble pruning

- *Trade-off between diversity and accuracy of two individual classifiers*[1,2]
  - Redundancy between these two individual classifiers

$$\mathrm{TDAC}(h_i, h_j) = \begin{cases} \lambda\, \mathrm{VI}(\boldsymbol{h}_i, \boldsymbol{h}_j) + (1-\lambda)\dfrac{\mathrm{MI}(\boldsymbol{h}_i, \boldsymbol{y}) + \mathrm{MI}(\boldsymbol{h}_j, \boldsymbol{y})}{2}\,, & \text{if } h_i \neq h_j\,; \\ 0\,, & \text{otherwise} \end{cases} \tag{3.1}$$

[1] Given two discrete random variables X and Y, the mutual information between them is defined as $\mathrm{I}(X;Y) = H(X) - H(X|Y) = \sum_{x\in X, y\in Y} p(x,y) \log {p(x,y)}/{p(x)p(y)}$, where $p(\cdot,\cdot)$, $\mathrm{H}(\cdot)$, and $\mathrm{H}(\cdot,\cdot)$ are the joint probabilty, the entropy function, and the joint entropy function, respectively.

[2] The normalised mutual information and the normalised variation of information of them are $\mathrm{MI}(X,Y) = {I(X;Y)}/{\sqrt{H(X)H(Y)}}$, and $\mathrm{VI}(X,Y) = 1 - {I(X;Y)}/{H(X,Y)}$, respectively.

Objection maximisation for ensemble pruning

- *Trade-off between diversity and accuracy of two individual classifiers*[1,2]
  - Redundancy between these two individual classifiers
  - Relevance between this individual classifier and the class vector

$$\boxed{\mathrm{TDAC}(h_i,h_j)} = \begin{cases} \lambda\, \boxed{\mathrm{VI}(\boldsymbol{h}_i,\boldsymbol{h}_j)} + (1-\lambda)\dfrac{\boxed{\mathrm{MI}(\boldsymbol{h}_i,\boldsymbol{y})} + \boxed{\mathrm{MI}(\boldsymbol{h}_j,\boldsymbol{y})}}{2}\,, & \text{if } \; h_i \neq h_j\,; \\ 0\,, & \text{otherwise} \end{cases} \tag{3.1}$$

[1]Given two discrete random variables X and Y, the mutual information between them is defined as $\mathrm{I}(X;Y) = H(X) - H(X|Y) = \sum_{x\in X, y\in Y} p(x,y) \log {}^{p(x,y)}/_{p(x)p(y)}$, where $p(\cdot,\cdot)$, $\mathrm{H}(\cdot)$, and $\mathrm{H}(\cdot,\cdot)$ are the joint probabilty, the entropy function, and the joint entropy function, respectively.

[2]The normalised mutual information and the normalised variation of information of them are $\mathrm{MI}(X,Y) = {}^{I(X;Y)}/_{\sqrt{H(X)H(Y)}}$, and $\mathrm{VI}(X,Y) = 1 - {}^{I(X;Y)}/_{H(X,Y)}$, respectively.

OMEP based on information entropy

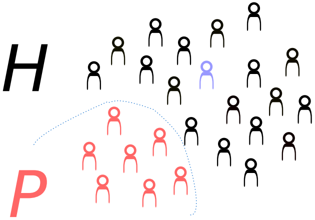
- *Trade-off between diversity and accuracy of two individual classifiers*
  - Redundancy between these two individual classifiers
  - Relevance between this individual classifier and the class vector

$$\mathrm{TDAC}(h_i,h_j) = \begin{cases} \lambda\, \mathrm{VI}(\boldsymbol{h}_i,\boldsymbol{h}_j) + (1-\lambda)\dfrac{\mathrm{MI}(\boldsymbol{h}_i,\boldsymbol{y}) + \mathrm{MI}(\boldsymbol{h}_j,\boldsymbol{y})}{2}\,, & \text{if } h_i \neq h_j\,; \\ 0\,, & \text{otherwise} \end{cases} \tag{3.1}$$

- *Trade-off between diversity and accuracy of an ensemble*

$$\mathrm{TDAS}(\mathcal{H}) = \frac{1}{2}\sum_{h_i\in\mathcal{H}}\sum_{h_j\in\mathcal{H}} \mathrm{TDAC}(h_i,h_j) \tag{3.2}$$

OMEP based on information entropy



Pick one of them randomly at first
Then pick multiple h^* iteratively by

$$\operatorname{argmax}_{h_i \in \mathcal{H} \setminus \mathcal{P}} \sum_{h_j \in \mathcal{P}} \mathrm{TDAC}(h_i, h_j)$$

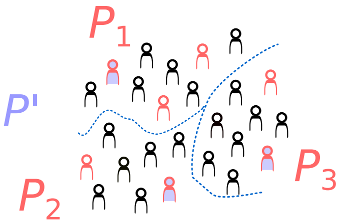
- Ensemble pruning $\Leftrightarrow$ objection maximisation

$$\max_{\mathcal{P} \subset \mathcal{H}, |\mathcal{P}|=k} \boxed{\mathrm{TDAS}(\mathcal{P})} \tag{3.3}$$

- Objective: to find a $\mathcal{P}$, that is,

$$\operatorname*{argmax}_{\mathcal{P} \subset \mathcal{H}, |\mathcal{P}|=k} \boxed{\mathrm{TDAS}(\mathcal{P})} \tag{3.4}$$

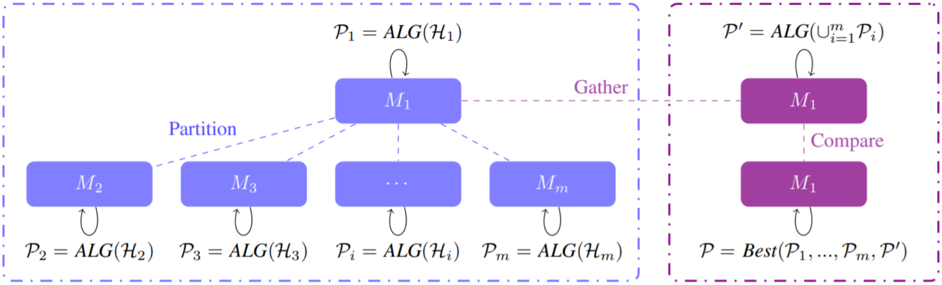
Pruning framework in a distributed setting



e.g., Gets DOMEP if COMEP is used,

$$\mathcal{P}' \leftarrow COMEP\ (\cup_{i=1}^{m} \mathcal{P}_i,\ k)$$

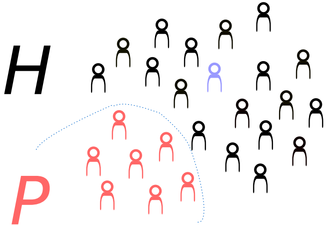
$$\mathcal{P} \leftarrow \text{argmax}_{\mathcal{T} \in \mathcal{P}_1,\ldots,\mathcal{P}_m,\mathcal{P}'} \text{TDAS}(\mathcal{T})$$



(a) The First Phase **(b) The Second Phase**

Figure 3.1. Ensemble pruning framework in a distributed setting (*EPFD*)

Centralised OMEP



Algorithm 3. Centralised objection maximisation for ensemble pruning (*COMEP*)

Input: Set of an original ensemble $\mathcal{H}$, threshold k as the size of the pruned sub-ensemble

Output: Set of the pruned sub-ensemble $\mathcal{P}$, meeting that $\mathcal{P} \subset \mathcal{H}$ and $|\mathcal{P}| \leqslant k$

1: $\mathcal{P} \leftarrow$ an arbitrary individual classifier $h_i \in \mathcal{H}$

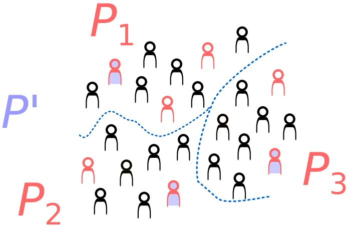
2: **for** $2 \leqslant i \leqslant k$ **do**

3: $\quad h^* \leftarrow \operatorname*{argmax}_{h_i \in \mathcal{H} \setminus \mathcal{P}} \sum_{h_j \in \mathcal{P}} \mathrm{TDAC}(h_i, h_j)$

4: $\quad$ Move h^* from $\mathcal{H}$ to $\mathcal{P}$

5: **end for**

Distributed OMEP



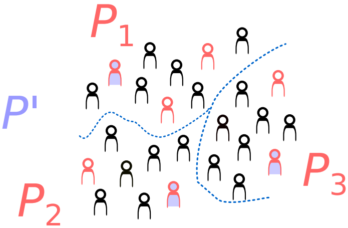
Algorithm 4. Distributed objection maximisation for ensemble pruning (*DOMEP*)

Input: Set of an original ensemble $\mathcal{H}$, threshold k as the size of the pruned sub-ensemble, number of machines m

Output: Set of the pruned sub-ensemble $\mathcal{P}$, meeting that $\mathcal{P} \subset \mathcal{H}$ and $|\mathcal{P}| \leqslant k$

1: Partition $\mathcal{H}$ randomly into m groups as equally as possible, i.e., $\mathcal{H}_1, ..., \mathcal{H}_m$
2: **for** $1 \leqslant i \leqslant m$ **do**
3: $\quad \mathcal{P}_i \leftarrow COMEP(\mathcal{H}_i, k)$
4: **end for**
5: $\mathcal{P}' \leftarrow COMEP(\bigcup_{i=1}^{m} \mathcal{P}_i, k)$
6: $\mathcal{P} \leftarrow \underset{\mathcal{T} \in \{\mathcal{P}_i, ..., \mathcal{P}_m, \mathcal{P}'\}}{\operatorname{argmax}} \mathrm{TDAS}(\mathcal{T})$

EP framework in a distributed setting



Algorithm 5. Ensemble pruning framework in a distributed setting (*EPFD*)

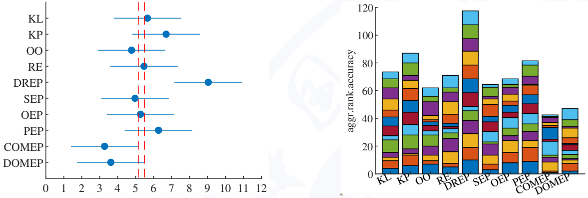
Input: Set of an original ensemble $\mathcal{H}$, number of machines m, a pruning method *ALG*

Output: Set of the pruned sub-ensemble $\mathcal{P}$, meeting that $\mathcal{P} \subset \mathcal{H}$

Partition $\mathcal{H}$ into $\{\mathcal{H}_i\}_{i=1}^{m}$ randomly
for $1 \leqslant i \leqslant m$ **do**
    $\mathcal{P}_i \leftarrow$ output from any pruning method *ALG* on $\mathcal{H}_i$
end for
$\mathcal{P}' \leftarrow$ output from *ALG* on $\bigcup_{i=1}^{m} \mathcal{P}_i$
$\mathcal{P} \leftarrow$ the best one among $\mathcal{P}_1, ..., \mathcal{P}_m$, and $\mathcal{P}'$ according to some certain criteria such as accuracy

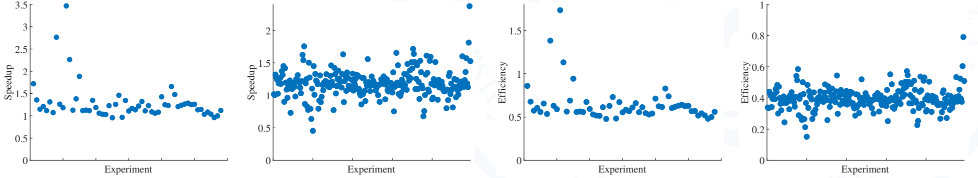
---

Comparison between *COMEP* and baselines



(a) **(b)**

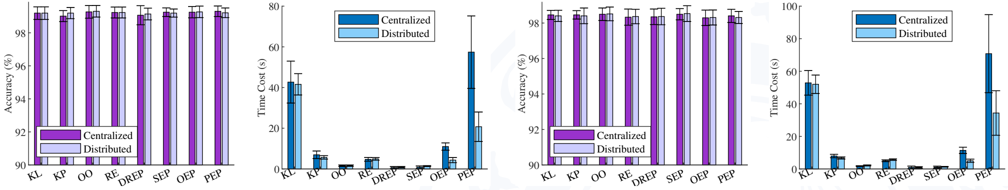
Figure 3.2. Comparison of the state-of-the-art methods with *COMEP* and *DOMEP* on the test accuracy. (a) Friedman test chart (non-overlapping means significant difference) [4]. (b) The aggregated rank for each method (the smaller the better) [5].



(a) **(b)** **(c)** **(d)**

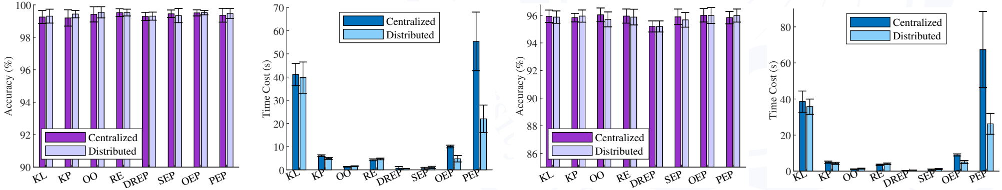
Figure 3.3. Comparison of speedup and efficiency between *COMEP* and *DOMEP*.
(a–b) Speedup with 2 or 3 machines, respectively. (c–d) Efficiency with 2 or 3 machines, respectively.

Comparison between *EPFD* and baselines



(a) (b) (c) (d)

Figure 3.4. Comparison of test accuracy and time cost between SOTA pruning methods and their corresponding distributed versions in binary classification.



(a) (b) (c) (d)

Figure 3.5. Comparison of test accuracy and time cost between SOTA pruning methods and their corresponding distributed versions in multi-class classification.

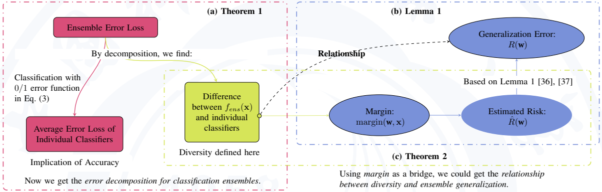
Brief summary

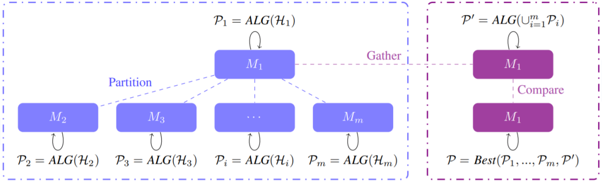
Binary classification

- Error decomposition in ensemble classificaiton
- Quantitative relationship between diversity and ensemble performance
- Pruning based on diversity to construct better ensembles

Multi-class classification

- Trade-off between diversity and accuracy based on information entropy
- Objection maximisation for ensemble pruning
- Ensemble pruning framework in a distributed setting

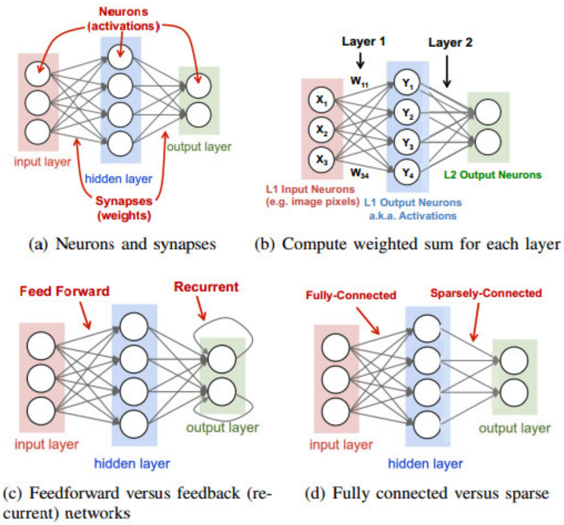




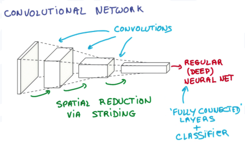
(a) The First Phase **(b) The Second Phase**

Overview

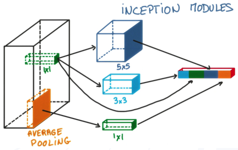
Deep neural networks



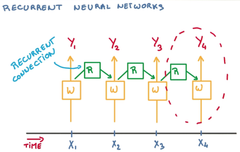
(a)



(b)



(c)



(d)

Figure 4.1. Deep neural netowrks, DNNs.

Neural architecture search

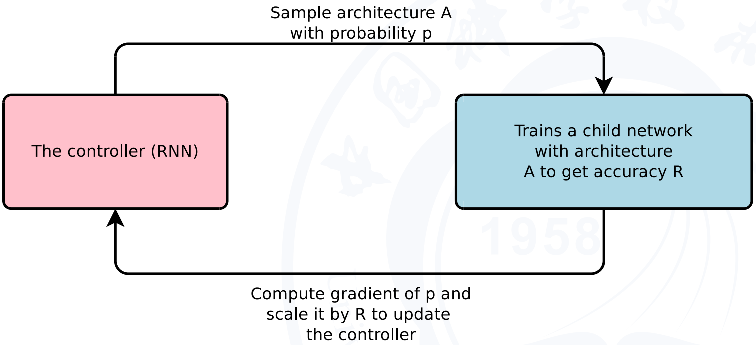


Figure 4.2. Neural architecture search, NAS[1] [6].

- *CIFAR-10*: 800 networks being trained on 800 GPUs concurrently at any time
- *Penn Treebank (PTB)*: 400 networks being trained on 400 GPUs concurrently at any time
- *WMT14 English → German translation*: 12 workers and each one uses 8 GPUs

[1] Zoph et al. [6] "Neural architecture search with reinforcement learning," *ICLR*, 2017.

NAS+ ensemble learning

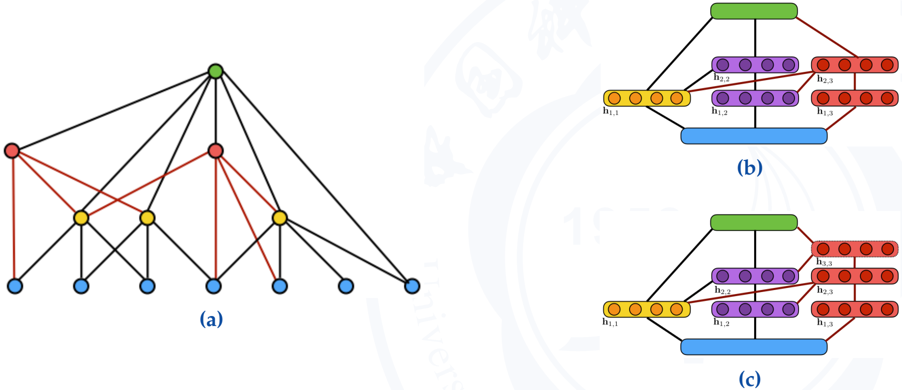


Figure 4.3. Examples AdaNet [7]; BoostResNet [8]; AdaNAS [9].
NB.[2] **AdaNet** (a) A general network architecture; (b) Illustration of the algorithm's incremental construction of a neural network.

[2] Cortes et al. [7] "Adanet: Adaptive structural learning of artificial neural networks," *ICML*, 2017: 874–883.

NAS+ ensemble learning

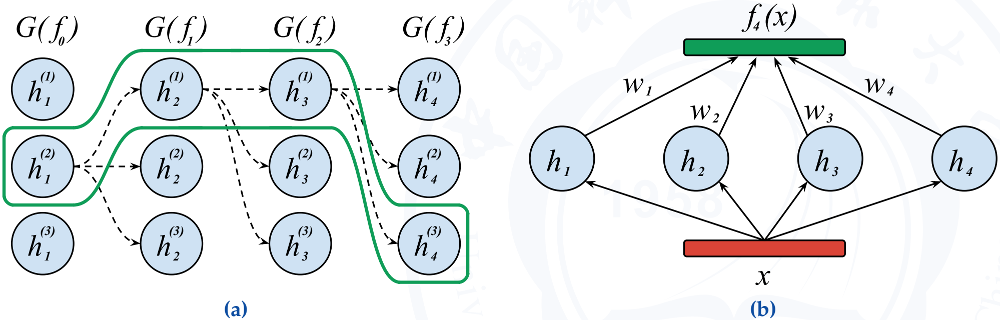


Figure 4.3. Examples AdaNet [7]; BoostResNet [8]; AdaNAS [9].
NB.[2,3] **AdaNAS** (a) Illustration of the search process over four iterations; (b) Illustration of the final ensemble.

[2]Huang et al. [8] "Learning deep resnet blocks sequentially using boosting theory," *ICML*, 2018.
[3]Macko et al. [9] "Improving neural architecture search image classifiers via ensemble learning," *arXiv preprint arXiv:1903.06236*, 2019.

Baseline algo and problem statement

For each input $x \in \mathcal{X}$, its output connects to all intermediate units, that is,

$$f(x) = \sum_{k=1}^{\ell} \mathbf{w}_k \cdot \mathbf{h}_k(x) \tag{4.1}$$

- where $\sum_{k=1}^{\ell} \|\mathbf{w}_k\| = 1$

Baseline algo and problem statement

For each input $x \in \mathcal{X}$, its output connects to all intermediate units, that is,

$$f(x) = \sum_{k=1}^{\ell} \mathbf{w}_k \cdot \mathbf{h}_k(x) \tag{4.1}$$

- where $\sum_{k=1}^{\ell} \|\mathbf{w}_k\| = 1$
-

and $\mathbf{h}_k = [\, h_{k,1}, \ldots, h_{k,n_k} \,]^\top$

Baseline algo and problem statement

For each input $x \in \mathcal{X}$, its output connects to all intermediate units, that is,

$$f(x) = \sum_{k=1}^{\ell} \boxed{\mathbf{w}_k} \cdot \boxed{\mathbf{h}_k(x)} \tag{4.1}$$

- where $\sum_{k=1}^{\ell} \|\mathbf{w}_k\| = 1$
- and $\mathbf{h}_k = [\, h_{k,1}, \ldots, h_{k,n_k} \,]^\top$
- let $h_{k,j}$ be the function of a unit in the k^{th} layer

$$h_{k,j}(x) = \sum_{s=0}^{k-1} \mathbf{u}_s \cdot \phi_s(\mathbf{h}_s(x)) \tag{4.2}$$

note that $\phi_s(\mathbf{h}_s) = (\phi_s(h_{s,1}), \ldots, \phi_s(h_{s,n_s}))$

Baseline algo and problem statement (cont.)

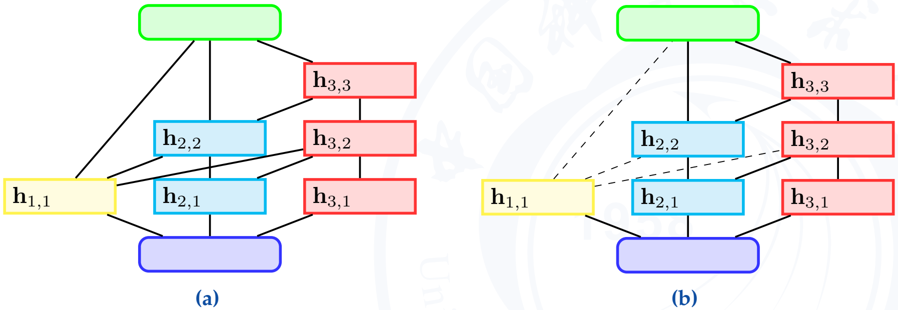
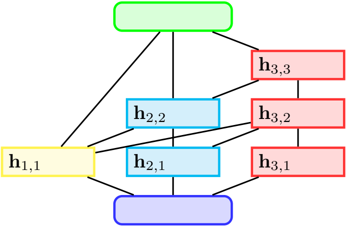


Figure 4.4. This figure is used to illustrate the difference between *SAEP* and AdaNet during the incremental construction of neural architectures. Layers in blue and green indicate the input and output layers, respectively. Units in yellow, cyan, and red are added at the first, second, and third iteration, respectively.

(a) AdaNet [7]: A line between two blocks of units indicates that these blocks are fully-connected. (b) *SAEP*: Only some valuable blocks are kept (those that will be pruned are denoted by black dashed lines), which is the key difference from AdaNet. The criteria used to decide which sub-architectures will be pruned have three proposed solutions in our *SAEP*, i.e., *PRS*, *PAP*, and *PIE*.

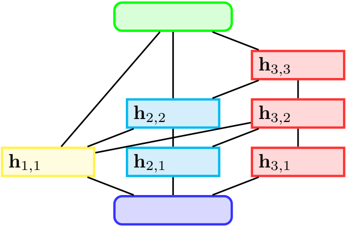
Sub-architecture ensemble pruning in NAS (*SAEP*)



Objective function to generate new candidate sub-architectures

$$\mathcal{L}_g(\mathbf{w}) = \hat{R}_{S,\rho}(f) + \Gamma \tag{4.3}$$

Sub-architecture ensemble pruning in NAS (*SAEP*)



Objective function to generate new candidate sub-architectures

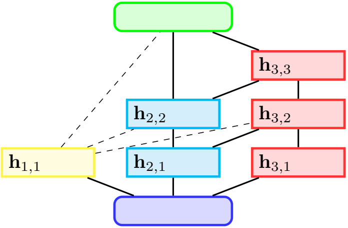
$$\mathcal{L}_g(\mathbf{w}) = \hat{R}_{S,\rho}(f) + \Gamma \tag{4.3}$$

To extend the objective to multi-class classification problems

$$g(x, y, f) = 2\mathbb{I}(f(x) = y) - 1 \tag{4.4}$$

$$\hat{R}_{S,\rho}(f) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left(g(x_i, y_i, f) \leqslant \rho\right) \tag{4.5}$$

Sub-architecture ensemble pruning in NAS (*SAEP*)



Algorithm 6. Sub-architecture ensemble pruning in neural architecture search (*SAEP*)

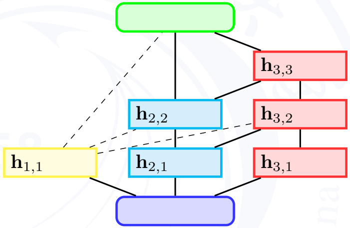
Input: Dataset $S = (x_i, y_i)_{i=1}^m$, number of iteration T
Output: Final function $f^{(T)}$

Initialize $f^{(0)} = \mathbf{0}$, and $l^{(0)} = 1$.
for $t = 1$ **to** T **do**
  $\mathbf{w}', \mathbf{h}' = \operatorname{argmin}_{\mathbf{w},\mathbf{h}} \mathcal{L}_g(f^{(t-1)} + \mathbf{w} \cdot \mathbf{h})$ s.t. $\mathbf{h} \in \mathcal{H}_{l^{(t-1)}}$.
  $\mathbf{w}'', \mathbf{h}'' = \operatorname{argmin}_{\mathbf{w},\mathbf{h}} \mathcal{L}_g(f^{(t-1)} + \mathbf{w} \cdot \mathbf{h})$ s.t. $\mathbf{h} \in \mathcal{H}_{l^{(t-1)}+1}$.
  if $\mathcal{L}_g(f^{(t-1)} + \mathbf{w}' \cdot \mathbf{h}') \leqslant \mathcal{L}_g(f^{(t-1)} + \mathbf{w}'' \cdot \mathbf{h}'')$ **then**
    $f^{(t)} = f^{(t-1)} + \mathbf{w}' \cdot \mathbf{h}'$.
  else
    $f^{(t)} = f^{(t-1)} + \mathbf{w}'' \cdot \mathbf{h}''$.
  end if
  Choose $\mathbf{w}_p$ based on one certain strategy, i.e., picking randomly in PRS, $\mathcal{L}_d(\mathbf{w})$ of Eq. (4.6) in PAP, or $\mathcal{L}_e(\mathbf{w}_i)$ of Eq. (4.7) in PIE.
  Set $\mathbf{w}_p$ to be zero.
end for

Sub-architecture ensemble pruning in NAS (*SAEP*)

There are three strategies to decide which sub-architectures are less valuable to be pruned

- Pruning by random selection (*PRS*)

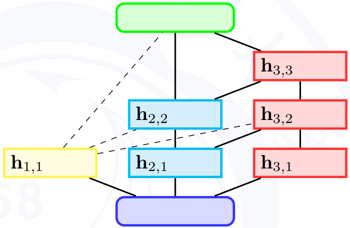


1. Whether or not to pick one of them to be pruned
2. If so, which one of the sub-architectures to prune

Sub-architecture ensemble pruning in NAS (*SAEP*)

There are three strategies to decide which sub-architectures are less valuable to be pruned

- Pruning by random selection (*PRS*)
- Pruning by *accuracy performance* (*PAP*)



$$\mathcal{L}_d(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m}\left[g(\boldsymbol{x}_i, y_i, f) - g(\boldsymbol{x}_i, y_i, f - \mathbf{w}\cdot\mathbf{h})\right] \tag{4.6}$$

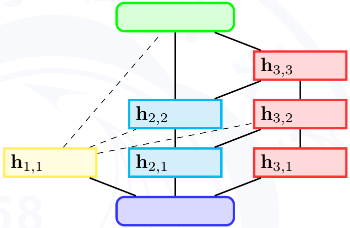
and the reason is

$$\mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[g(\boldsymbol{x}, y, f) - g(\boldsymbol{x}, y, f - \mathbf{w}_j\cdot\mathbf{h}_j)\right] \leqslant 0$$

Sub-architecture ensemble pruning in NAS (*SAEP*)

There are three strategies to decide which sub-architectures are less valuable to be pruned

- Pruning by random selection (*PRS*)
- Pruning by *accuracy performance* (*PAP*)
- Pruning by *information entropy* (*PIE*)



$$\mathcal{L}_e(\mathbf{w}_i) = \sum_{\mathbf{w}_j \cdot \mathbf{h}_j \in f \setminus \{\mathbf{w}_i \cdot \mathbf{h}_i\}} \mathcal{L}_p(\mathbf{w}_i, \mathbf{w}_j) \tag{4.7}$$

where

$$\mathcal{L}_p(\mathbf{w}_i, \mathbf{w}_j) = (1-\alpha)\,\mathrm{VI}(\boldsymbol{w}_i, \boldsymbol{w}_j) + \alpha \frac{\mathrm{MI}(\boldsymbol{w}_i, \mathbf{y}) + \mathrm{MI}(\boldsymbol{w}_j, \mathbf{y})}{2}$$

SAEP leads to ensemble architectures with smaller size

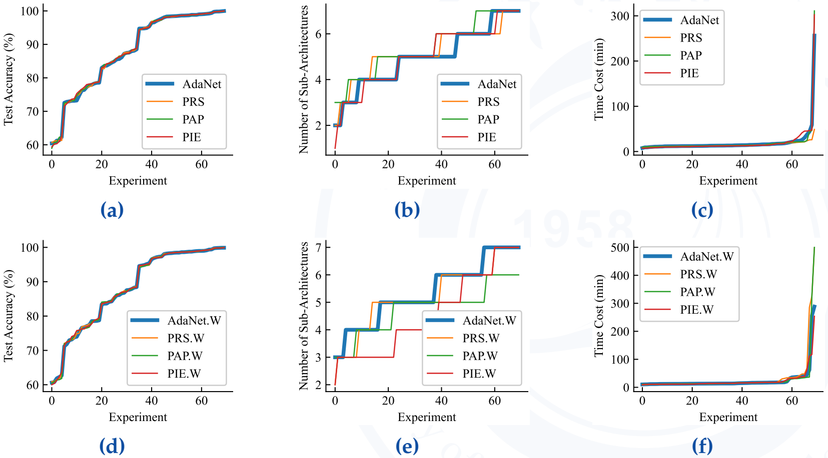


Figure 4.5. Comparison of the baseline AdaNet and the proposed *SAEP* including their corresponding variants, using MLPs as sub-architectures for image classification. (a–c) Comparison of performance of AdaNet and *SAEP*. (d–f) Comparison of performance of their corresponding variants.

PIE generates sub-ensemble architectures with more diversity

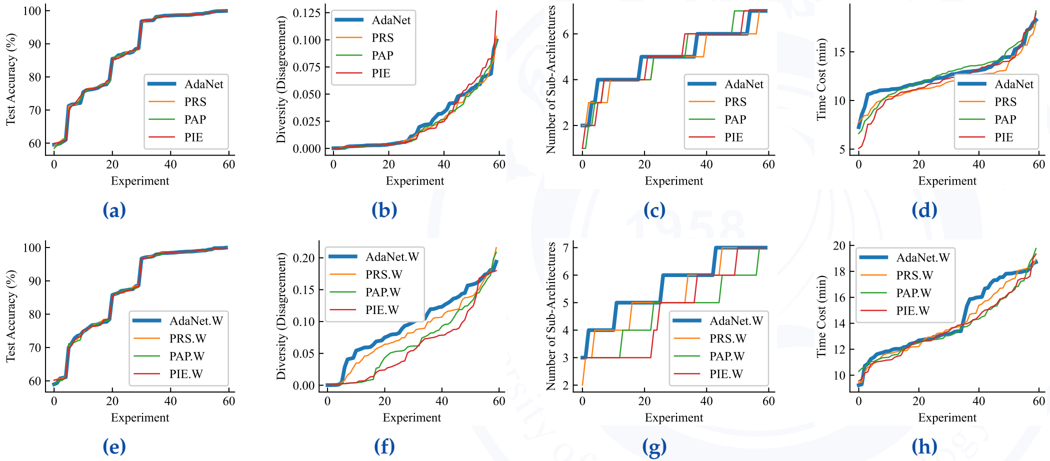


Figure 4.6. Comparison of the baseline AdaNet and the proposed *SAEP* including their corresponding variants, using MLPs as sub-architectures for binary classification. (a–c) Comparison of performance of AdaNet and *SAEP*. (d–f) Comparison of performance of their corresponding variants.

PIE generates sub-ensemble architectures with more diversity

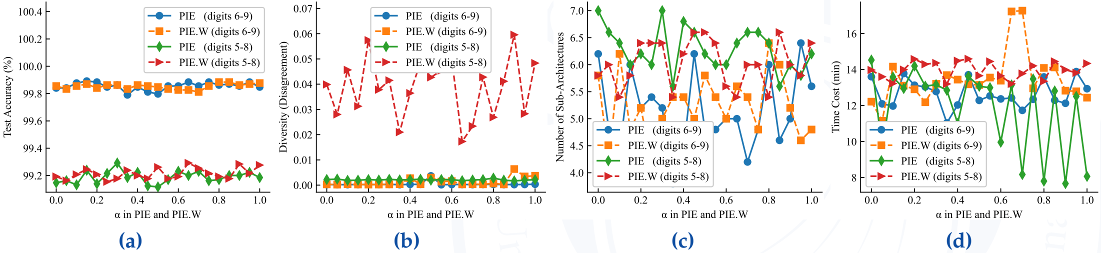
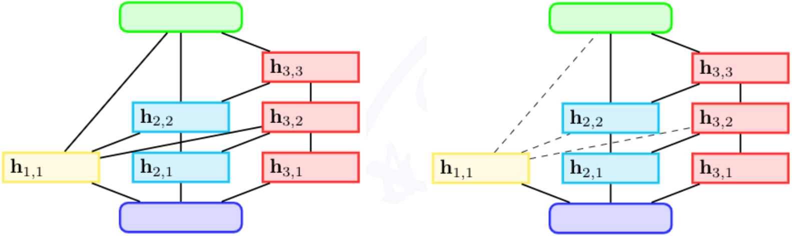


Figure 4.6. The effect of different α values in *PIE* and *PIE.W* for binary classification. (a) The effect of the α value on the test accuracy performance of sub-ensemble architectures. (b) The effect of the α value on the diversity of sub-ensemble architectures, measured by the disagreement measure.[4] (c) The effect of the α value on the size of sub-ensemble architectures. (d) The effect of the α value on the time cost.

[4] The disagreement between two sub-architectures $\mathbf{w}_i$ and $\mathbf{w}_j$ is $\mathrm{dis}(\mathbf{w}_i, \mathbf{w}_j) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(\,\mathbf{h}_i(x) \neq \mathbf{h}_j(x)\,)$, and the diversity of the ensemble architecture f using the disagreement measure is $\mathbf{dis}(f) = \frac{2}{\ell(\ell-1)} \sum_{\mathbf{w}_i \cdot \mathbf{h}_i \in f} \sum_{\mathbf{w}_j \cdot \mathbf{h}_j \in f, \mathbf{h}_j \neq \mathbf{h}_i} \mathrm{dis}(\mathbf{w}_i, \mathbf{w}_j)$

Brief summary (*SAEP*)



(a) (b)

- Sub-architecture ensemble pruning in neural architecture search (*SAEP*)
  - Pruning by Random Selection (*PRS*)
  - Pruning by Accuracy Performance (*PAP*)
  - Pruning by Information Entropy (*PIE*)

Application in other areas (e.g., NAS)

- Obtaining smaller sub-architecture ensembles via diversity without much accuracy decline
- Exploring distinct deeper sub-architectures if diversity is not sufficient enough

References I

[1] DIETTERICH T G. Ensemble methods in machine learning[C]//MCS. Springer, 2000: 1-15.

[2] HERBRICH R, GRAEPEL T. A pac-bayesian margin bound for linear classifiers: Why svms work[C]//NIPS. 2001: 224-230.

[3] HERBRICH R, GRAEPEL T. A pac-bayesian margin bound for linear classifiers[J]. IEEE Trans Inf Theory, 2002, 48(12): 3140-3150.

[4] DEMŠAR J. Statistical comparisons of classifiers over multiple data sets[J]. J Mach Learn Res, 2006, 7: 1-30.

[5] QIAN C, YU Y, ZHOU Z H. Pareto ensemble pruning[C]//AAAI. 2015.

[6] ZOPH B, LE Q V. Neural architecture search with reinforcement learning[C]//ICLR. 2017.

[7] CORTES C, GONZALVO X, KUZNETSOV V, et al. Adanet: Adaptive structural learning of artificial neural networks[C]//ICML. 2017: 874-883.

[8] HUANG F, ASH J, LANGFORD J, et al. Learning deep resnet blocks sequentially using boosting theory[C]//ICML. 2018.

[9] MACKO V, WEILL C, MAZZAWI H, et al. Improving neural architecture search image classifiers via ensemble learning[J]. arXiv preprint arXiv:1903.06236, 2019.