Model ensembles and mixtures of experts

MIPT

2022

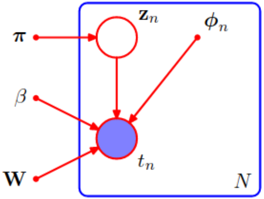
Model ensembling

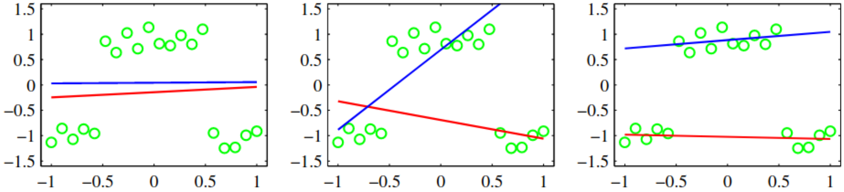
Definition (Wiki)

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. A machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

Mixture model

$$f = \sum \gamma_i f_i(x)$$





Model selection: coherent Bayesian inference

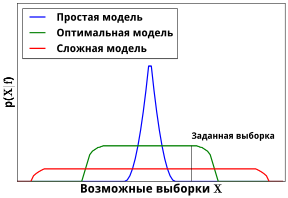
First level: find optimal parameters:

$$\mathbf{w} = \arg\max \frac{p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})}{p(\mathfrak{D}|\mathbf{h})},$$

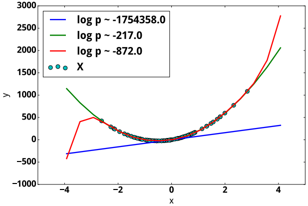
Second level: find optimal model:

Evidence:

$$p(\mathfrak{D}|\mathbf{h}) = \int_{\mathbf{w}} p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})d\mathbf{w}.$$



Model selection scheme



Polynomial regression example

Mixutre vs Bayesian model averaging

Mxiture:

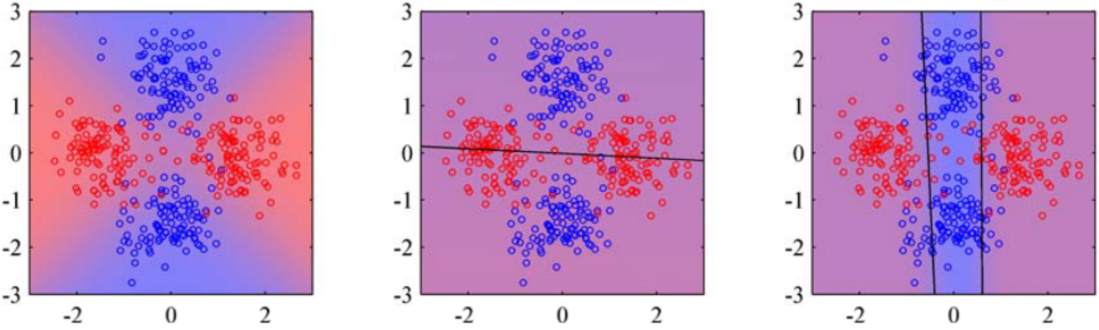
$$\mathsf{f} = \sum \gamma_i \mathsf{f}_i(\mathsf{X}) = \sum \prod_{\mathsf{x}} p(\mathsf{x}, \gamma_i)$$

Averaging:

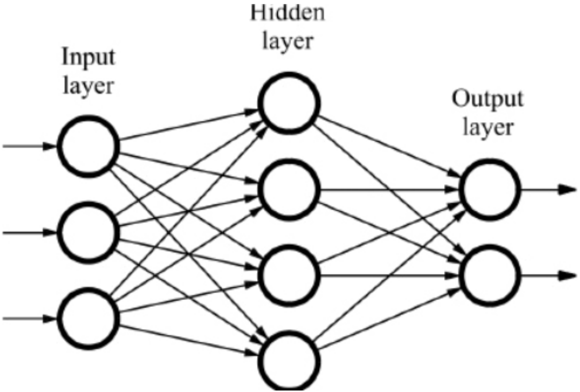
$$\mathsf{f} = \sum p(\mathsf{f}_i)\mathsf{f}_i(\mathsf{X}).$$

Mixture of experts

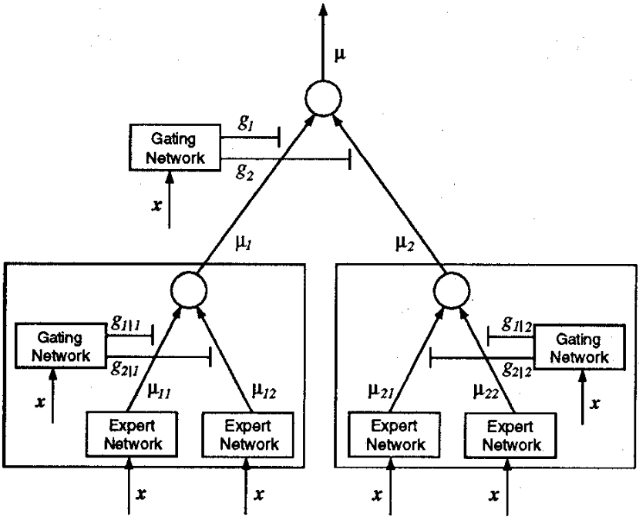
$$f = \sum \gamma_i(x) f_i(x)$$



MLP: mixture?



Mixutre of experts with hierarchy



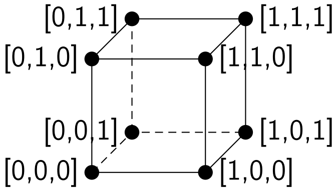
Multimodels

$$\mathsf{f} = \sum \gamma_i(\mathsf{x})\mathsf{f}_i(\mathsf{x}),$$

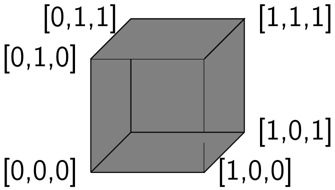
$$\sum \boldsymbol{\gamma} = 1, \quad \boldsymbol{\gamma}_i \in 0, 1.$$

Structure restrictions

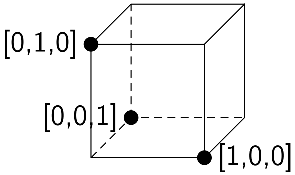
An example of restrictions for structure parameter γ, $|\gamma| = 3$.



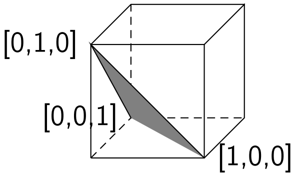
Cube vertices



Cube interior



Simplex vertices

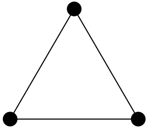


Simplex interior

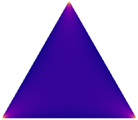
Prior distribution for the model structure

Every point in a simplex defines a model.

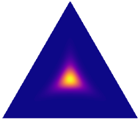
Gumbel-Softmax distribution: $\boldsymbol{\Gamma} \sim \mathrm{GS}(s, \lambda_{\text{temp}})$



$\lambda_{\text{temp}} \to 0$

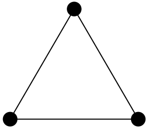


$\lambda_{\text{temp}} = 0.995$

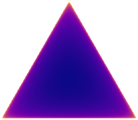


$\lambda_{\text{temp}} = 5.0$

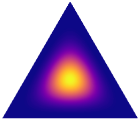
Dirichlet distribution: $\boldsymbol{\Gamma} \sim \mathrm{Dir}(s, \lambda_{\text{temp}})$



$\lambda_{\text{temp}} \to 0$

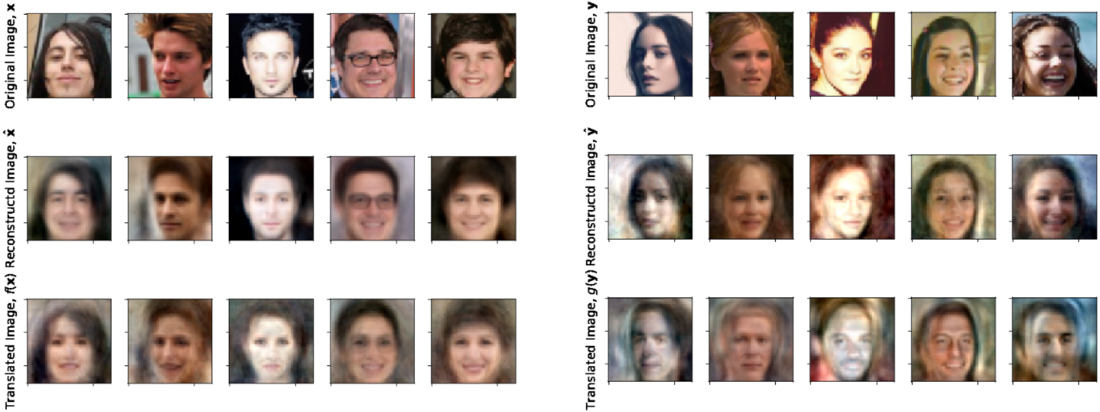


$\lambda_{\text{temp}} = 0.995$

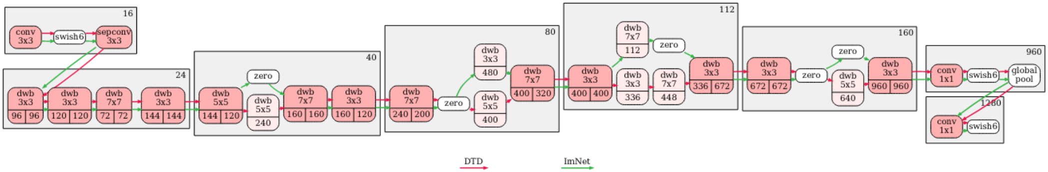


$\lambda_{\text{temp}} = 5.0$

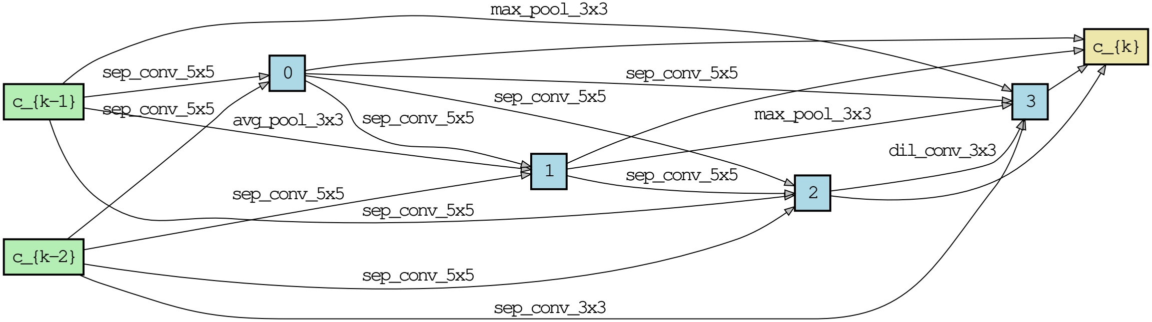
Multi-domain tasks



Multi-domain tasks



Neural architecture search example



Structure selection: neural architecture search space

The model f is defined by the **structure** $\boldsymbol{\Gamma} = [\gamma^{0,1}, \gamma^{1,2}]$.

$$\text{Model: } \mathrm{f}(\mathrm{x}) = \textbf{softmax}\left((\mathrm{w}_0^{1,2})^\mathsf{T} \mathrm{f}_1(\mathrm{x})\right), \quad \mathrm{f}(\mathrm{x}) : \mathbb{R}^n \to [0,1]^{|\mathbb{Y}|}, \quad \mathrm{x} \in \mathbb{R}^n.$$

$$\mathrm{f}_1(\mathrm{x}) = \gamma_0^{0,1} \mathrm{g}_0^{0,1}(\mathrm{x}) + \gamma_1^{0,1} \mathrm{g}_1^{0,1}(\mathrm{x}),$$

where $\mathrm{w} = [\mathrm{w}_0^{0,1}, \mathrm{w}_0^{1,2}]^\mathsf{T}$ — parameter matrices, $\mathrm{g}_{0,1}^0$ is a convolution,$\mathrm{g}_{0,1}^1$ is a pooling operation, $\mathrm{g}_{1,2}^0$ is a generalized-linear function.

$$\begin{tikzcd}[column sep=huge]
\mathrm{f}_0(\mathrm{x}) = \mathrm{x} \arrow[r, bend left, "\gamma_0^{0,1} \mathrm{g}_0^{0,1}(\mathrm{x}) = \gamma_0^{0,1} \textbf{Conv}(\mathrm{x}, \mathrm{w}_0^{0,1})"] \arrow[r, bend right, "\gamma_1^{0,1} \mathrm{g}_1^{0,1}(\mathrm{x}) = \gamma_1^{0,1} \textbf{MaxPool}(\mathrm{x})"'] & \mathrm{f}_1(\mathrm{x}) \arrow[r, "\gamma_0^{1,2} \mathrm{g}_0^{1,2}(\mathrm{x}) = \gamma_0^{1,2} \textbf{softmax}\left((\mathrm{w}_0^{1,2})^\mathsf{T} \mathrm{x}\right)"] & \mathrm{f}_2(\mathrm{x})
\end{tikzcd}$$

Deep learning model structure as a graph

Define:

1. acyclic graph (V, E);
2. for each edge $(j,k) \in E$: a vector primitive differentiable functions $\mathrm{g}^{j,k} = [\mathrm{g}_0^{j,k}, \dots, \mathrm{g}_{K^{j,k}}^{j,k}]$ with length of $K^{j,k}$;
3. for each vertex $v \in V$: a differentiable aggregation function $\mathbf{agg}_v$.
4. a function $\mathrm{f} = \mathrm{f}_{|V|-1}$:

$$\mathrm{f}_v(\mathrm{w}, \mathrm{x}) = \mathbf{agg}_v \left(\{ \langle \boldsymbol{\gamma}^{j,k}, \mathrm{g}^{j,k} \rangle \circ \mathrm{f}_j(\mathrm{x}) | j \in \mathrm{Adj}(v_k) \} \right), v \in \{1, \dots, |V|-1\}, \quad \mathrm{f}_0(\mathrm{x}) = \mathrm{x} \qquad (1)$$

that is a function from $\mathbb{X}$ into a set of labels $\mathbb{Y}$ for any value of $\boldsymbol{\gamma}^{j,k} \in [0,1]^{K^{j,k}}$.

Definition

A *parametric set of models* $\mathfrak{F}$ is a graph (V, E) with a set of primitive functions $\{\mathrm{g}^{j,k}, (j,k) \in E\}$ and aggregation functions $\{\mathbf{agg}_v, v \in V\}$.

Statement

A function $\mathrm{f} \in \mathfrak{F}$ is a model for each $\boldsymbol{\gamma}^{j,k} \in [0,1]^{K^{j,k}}$.

Neural Architecture Search: problem statement

w are model parameters.
$\mathbf{\Gamma}$ is a structure.

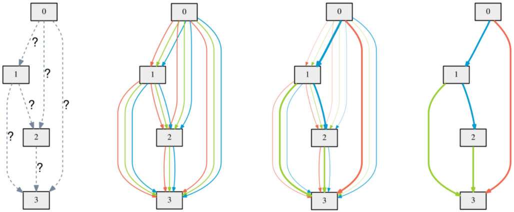
$$\mathbf{\Gamma}^* = \arg\max Q(\mathrm{w}^*, \mathbf{\Gamma}),$$

$$\mathrm{w}^* = \arg\max L(\mathrm{w}, \mathbf{\Gamma}).$$

DARTS

The model is a multigraph, where edges $[g^e]$ correspond to submodels, vertices $f_v(x)$ are the results of submodels:

$$f_v = \langle \gamma, \mathbf{softmax}([g^e(x)]) \rangle.$$



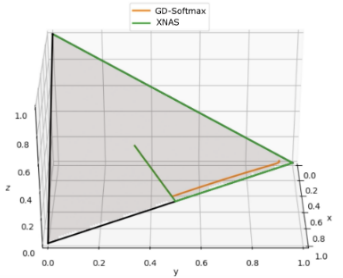
XNAS

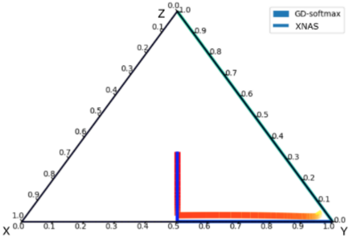
Algorithm 1 XNAS for a single forecaster

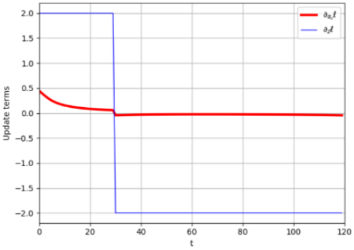
Input: The learning rate η,
Loss-gradient bound $\mathcal{L}$,
Experts predictions $\{f_{t,i}\}_{i=1}^{N}$ $\forall t = 1, \dots, T$
Init: $I_0 = \{1, \dots, N\}$, $v_{0,i} \leftarrow 1$, $\forall i \in I_0$
for rounds $t = 1, \dots, T$ **do**
  Update $\boldsymbol{\omega}$ by descending $\nabla_{\boldsymbol{\omega}} \ell_{\text{train}}(\boldsymbol{\omega}, \boldsymbol{v})$
  $p_t \leftarrow \frac{\sum_{i \in I_{t-1}} v_{t-1,i} \cdot f_{t-1,i}}{\sum_{i \in I_{t-1}} v_{t-1,i}}$ #Predict
  {loss gradient revealed: $\nabla_{p_t} \ell_{\text{val}}(p_t)$}
  for $i \in I_{t-1}$ **do**
    $R_{t,i} = -\nabla_{p_t} \ell_{\text{val}}(p_t) \cdot f_{t,i}$ #Rewards
    $v_{t,i} \leftarrow v_{t-1,i} \cdot \exp\{\eta R_{t,i}\}$ #EG step
  end for
  $\theta_t \leftarrow \max_{i \in I_{t-1}} \{v_{t,i}\} \cdot \exp\{-2\eta \mathcal{L}(T-t)\}$
  $I_t \leftarrow I_{t-1} \setminus \{i \mid v_{t,i} < \theta_t\}$ #Wipeout
end for

ImageNet Architecture	Test error	Params (M)	Search cost
SNAS [50]	27.3	4.3	1.5
ASAP [29]	26.7	5.1	0.2
DARTS [25]	26.7	4.9	1
NASNet-A [56]	26.0	5.3	1800
PNAS [24]	25.8	5.1	150
Amoeba-A [33]	25.5	5.1	3150
RandWire [48]	25.3	5.6	0
SharpDarts [17]	25.1	4.9	0.8
Amoeba-C [33]	24.3	6.4	3150
XNAS	**24.0**	5.2	0.3

XNAS







Model selection: hybrid approach

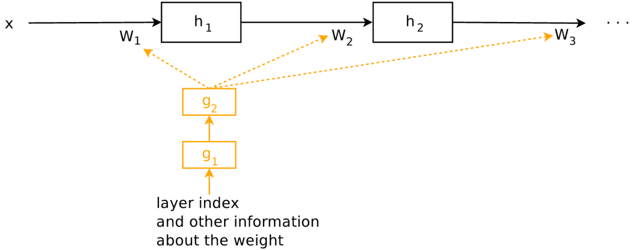
Definition

Given a set Λ.

Hypernetwork is a parametric mapping from Λ to set $\mathbb{R}^n$ of the model f parameters:

$$G : \Lambda \times \mathbb{R}^u \to \mathbb{R}^n,$$

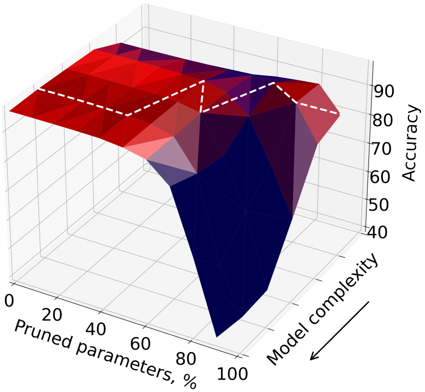
where $\mathbb{R}^u$ is a set of hypernetwork parameters.



Ha et al., 2016

Hypernetworks for the optimal model selection

Hypernetworks allow to select the optimal model on the inference step.



Architecture complexity control

The hypernetworks can approximate not only the model parameter w, but also structural parameters γ.

Baseline: DARTS

A model architecture is a directed graph with non-linear operations $\mathrm{f}^{(i,j)}$ that are induced by basic functions $\mathrm{g}^{(i,j)}$ with weights obtained by softmax function application:

$$\mathrm{f}^{(i,j)}(\mathrm{x}) = \langle \mathbf{softmax}(\boldsymbol{\gamma}^{(i,j)}), \mathrm{g}^{(i,j)}(\mathrm{x})\rangle$$

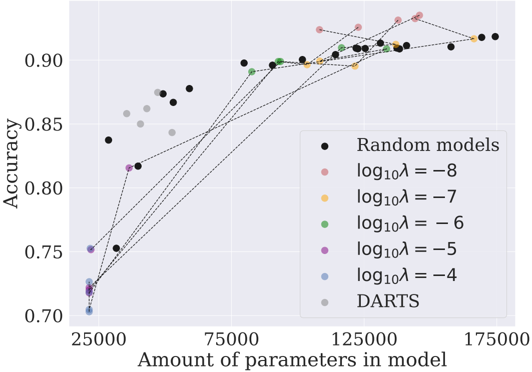
Our proposal

To use a mapping $\boldsymbol{\gamma}(\lambda)$ instead of constant structural parameters $\boldsymbol{\gamma}(\lambda)$, where λ is a regularization term for the loss function:

$$\mathrm{E}_{\lambda}\left(\log\ p(\mathrm{y}|\mathrm{X}, \mathrm{w}, \boldsymbol{\Gamma}(\lambda)) + \lambda \sum_{(i,j)} \langle \mathbf{softmax}\left(\boldsymbol{\gamma}(\boldsymbol{\lambda})^{(i,j)}\right), \mathrm{n}(\mathrm{g}^{(i,j)})\rangle\right),$$

where $\mathrm{n}(\mathrm{g}^{(i,j)})$ is a vector of amount of parameters for all the basic functions g.

Example: CIFAR-10



$$\mathsf{E}_\lambda \left(\log\ p(\mathsf{y}|\mathsf{X}, \mathsf{w}, \mathbf{\Gamma}(\lambda)) + \lambda \sum_{(i,j)} \langle \mathbf{softmax} \left(\boldsymbol{\gamma}(\boldsymbol{\lambda})^{(i,j)} \right), \mathsf{n}(\mathsf{g}^{(i,j)}) \rangle \right).$$

Uncertainity estimation using ensembles

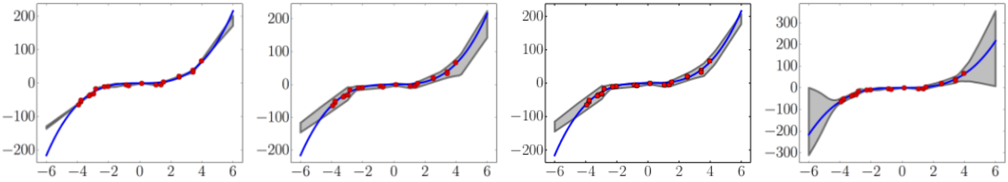


Figure 1: Results on a toy regression task: x-axis denotes x. On the y-axis, the blue line is the *ground truth* curve, the red dots are observed noisy training data points and the gray lines correspond to the predicted mean along with three standard deviations. Left most plot corresponds to empirical variance of 5 networks trained using MSE, second plot shows the effect of training using NLL using a single net, third plot shows the additional effect of adversarial training, and final plot shows the effect of using an ensemble of 5 networks respectively.

Neural ensebmle search

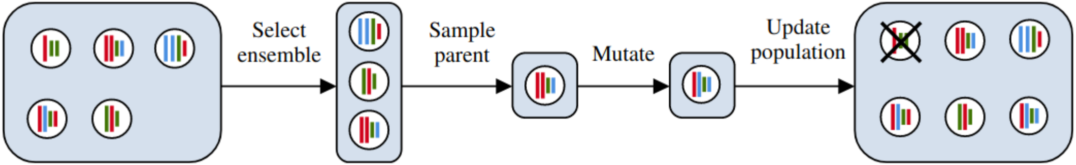


Figure 3: Illustration of one iteration of NES-RE. Network architectures are represented as colored bars of different lengths illustrating different layers and widths. Starting with the current population, ensemble selection is applied to select parent candidates, among which one is sampled as the parent. A mutated copy of the parent is added to the population, and the oldest member is removed.

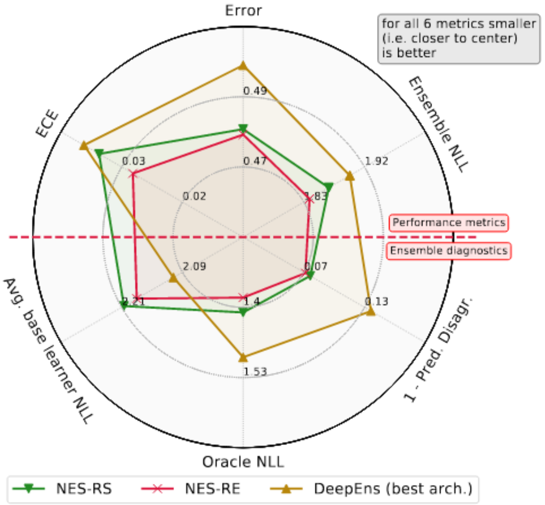
Neural ensebmle search

Algorithm 1: NES with Regularized Evolution

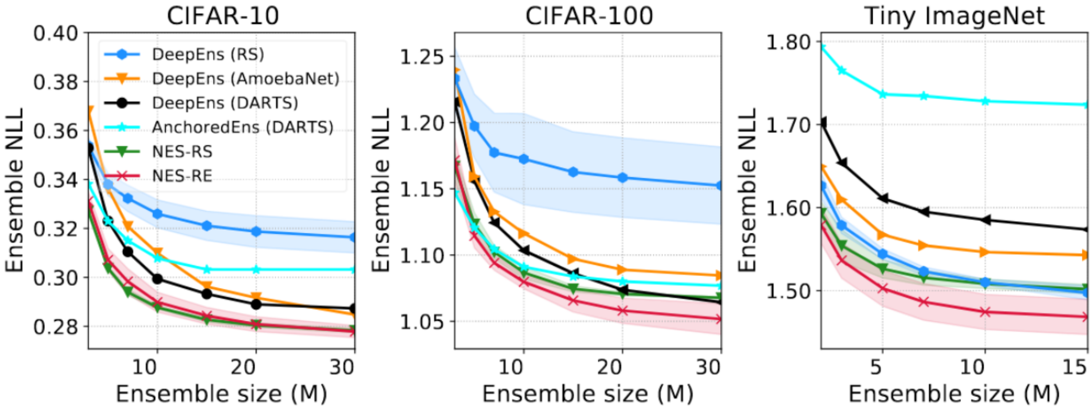
Data: Search space $\mathcal{A}$; ensemble size M; comp. budget K; $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}$; population size P; number of parent candidates m.

Sample P architectures $\alpha_1, \dots, \alpha_P$ independently and uniformly from $\mathcal{A}$.
Train each architecture α_i using $\mathcal{D}_{\text{train}}$, and initialize $\mathfrak{p} = \mathcal{P} = \{f_{\theta_1,\alpha_1}, \dots, f_{\theta_P,\alpha_P}\}$.
while $|\mathcal{P}| < K$ **do**
  Select m parent candidates $\{f_{\widetilde{\theta}_1,\widetilde{\alpha}_1}, \dots, f_{\widetilde{\theta}_m,\widetilde{\alpha}_m}\} = \texttt{ForwardSelect}(\mathfrak{p}, \mathcal{D}_{\text{val}}, m)$.
  Sample uniformly a parent architecture α from $\{\widetilde{\alpha}_1, \dots, \widetilde{\alpha}_m\}$. // α stays in $\mathfrak{p}$.
  Apply mutation to α, yielding child architecture β.
  Train β using $\mathcal{D}_{\text{train}}$ and add the trained network $f_{\theta,\beta}$ to $\mathfrak{p}$ and $\mathcal{P}$.
  Remove the oldest member in $\mathfrak{p}$. // as done in RE [49].
Select base learners $\{f_{\theta_1^*,\alpha_1^*}, \dots, f_{\theta_M^*,\alpha_M^*}\} = \texttt{ForwardSelect}(\mathcal{P}, \mathcal{D}_{\text{val}}, M)$ by forward step-wise selection without replacement.
return ensemble $\texttt{Ensemble}(f_{\theta_1^*,\alpha_1^*}, \dots, f_{\theta_M^*,\alpha_M^*})$

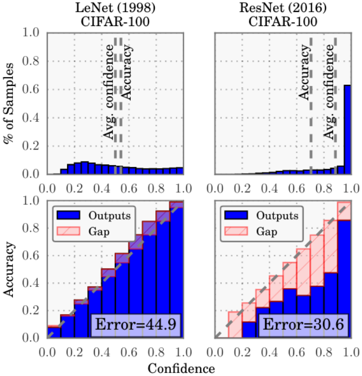
Neural ensebmle search



Neural ensebmle search



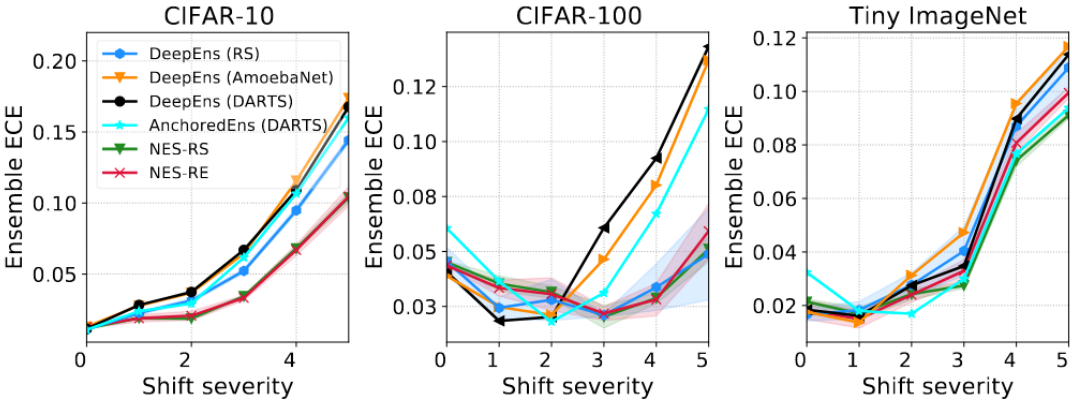
Neural ensebmle search



$$\mathop{\mathbb{E}}_{\hat{P}}\left[\left|\mathbb{P}\left(\hat{Y}=Y \mid \hat{P}=p\right)-p\right|\right]$$

$$\mathrm{ECE}=\sum_{m=1}^{M}\frac{|B_m|}{n}\left|\mathrm{acc}(B_m)-\mathrm{conf}(B_m)\right|,$$

Neural ensebmle search



References

- Bishop C. M., Nasrabadi N. M. Pattern recognition and machine learning. – New York : springer, 2006. – Т. 4. – №. 4. – С. 738.
- Адуенко А. А. 2017. Выбор мультимоделей в задачах классификации. Диссертация.
- Jordan M. I., Jacobs R. A. Hierarchical mixtures of experts and the EM algorithm //Neural computation. – 1994. – Т. 6. – №. 2. – С. 181-214.
- Бахтеев О. Ю. 2020. Байесовский выбор субоптимальной структуры модели глубокого обучения. Диссертация.
- Kuznetsova R., Bakhteev O., Ogaltsov A. Variational learning across domains with triplet information //arXiv preprint arXiv:1806.08672. – 2018.
- Wang Q. et al. Multi-path neural networks for on-device multi-domain visual classification //Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. – 2021. – С. 3019-3028.
- Liu H., Simonyan K., Yang Y. Darts: Differentiable architecture search //arXiv preprint arXiv:1806.09055. – 2018.
- Nayman N. et al. Xnas: Neural architecture search with expert advice //Advances in neural information processing systems. – 2019. – Т. 32.
- Ha D., Dai A., Le Q. V. Hypernetworks //arXiv preprint arXiv:1609.09106. – 2016.
- Grebenkova O., Bakhteev O. Y., Strijov V. Variational deep learning model optimization with complexity control //Informatika i Ee Primeneniya (Inform. Appl.). – 2021. – Т. 15. – №. 1. – С. 42-49.
- Yakovlev K. D. et al. Neural Architecture Search with Structure Complexity Control //International Conference on Analysis of Images, Social Networks and Texts. – Springer, Cham, 2022. – С. 207-219.
- Lakshminarayanan B., Pritzel A., Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles //Advances in neural information processing systems. – 2017. – Т. 30.
- Zaidi S. et al. Neural ensemble search for uncertainty estimation and dataset shift //Advances in Neural Information Processing Systems. – 2021. – Т. 34. – С. 7898-7911.
- Guo C. et al. On calibration of modern neural networks //International conference on machine learning. – PMLR, 2017. – С. 1321-1330