Model ensembles and mixtures of experts

MIPT

2024

Model ensembling

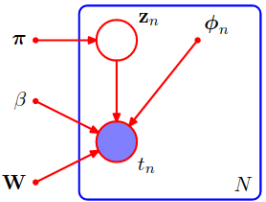
Definition (Wiki)

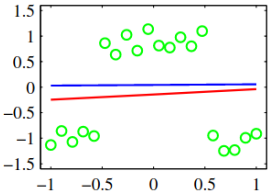
In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. A machine learning ensemble consists of only a concrete finite set of alternative models, but typically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

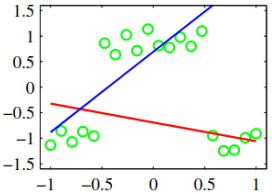
Naive ensembling

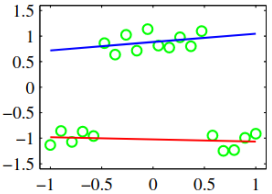
Mixture model

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









Model selection: coherent Bayesian inference

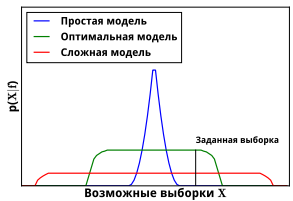
First level: find optimal parameters:

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

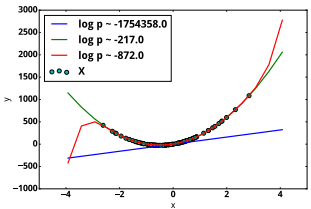
Second level: find optimal model:

Evidence:

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



Model selection scheme



Polynomial regression example

Mixutre vs Bayesian model averaging

Mxiture:

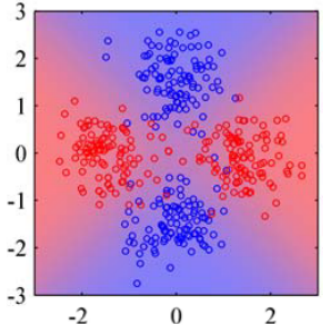
$$\mathsf{f} = \sum \gamma_i \mathsf{f}_i(\mathsf{X}) = \sum \prod_{\mathsf{x}} p(\mathsf{x}, \gamma_i)$$

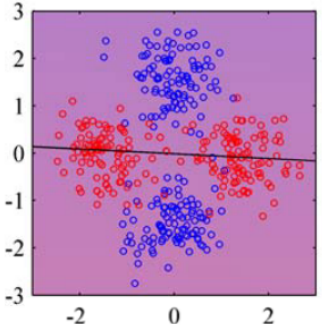
Averaging:

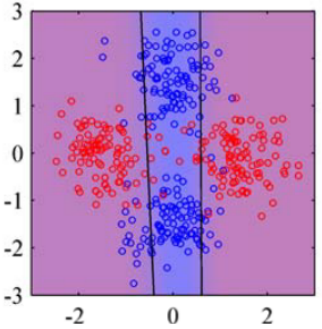
$$\mathsf{f} = \sum p(\mathsf{f}_i)\mathsf{f}_i(\mathsf{X}).$$

Mixture of experts

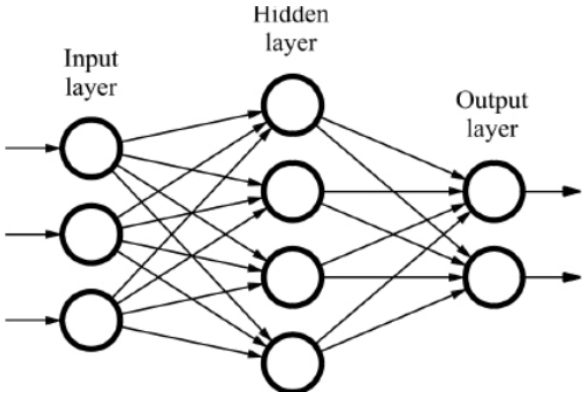
$$f = \sum \gamma_i(x) f_i(x)$$



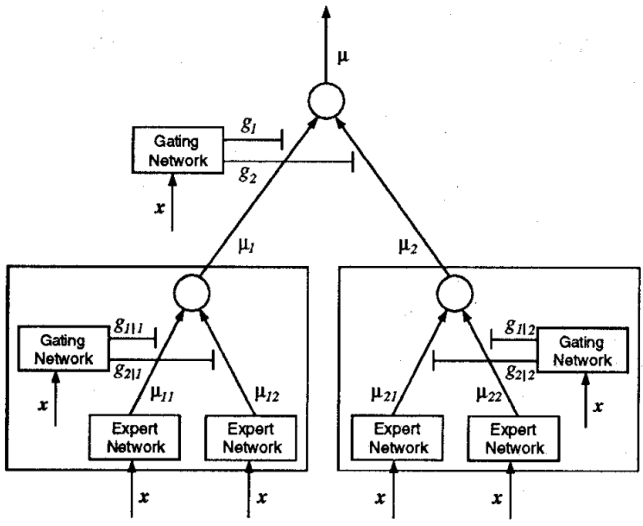




MLP: mixture?



Mixutre of experts with hierarchy



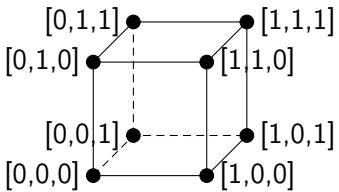
Multimodels

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

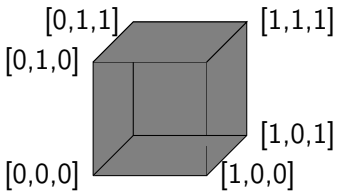
$$\sum \gamma = 1, \quad \gamma_i \in 0, 1.$$

Structure restrictions

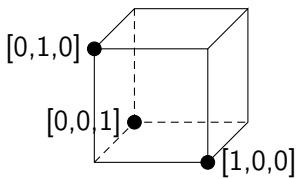
An example of restrictions for structure parameter γ, $|\gamma| = 3$.



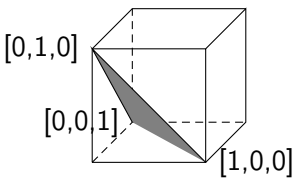
Cube vertices



Cube interior



Simplex vertices

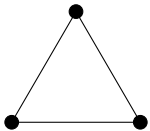


Simplex interior

Prior distribution for the model structure

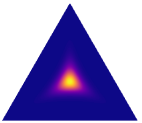
Every point in a simplex defines a model.

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



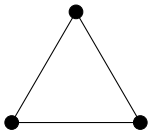
$\lambda_{\text{temp}} \to 0$

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



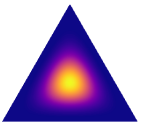
$\lambda_{\text{temp}} = 5.0$

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



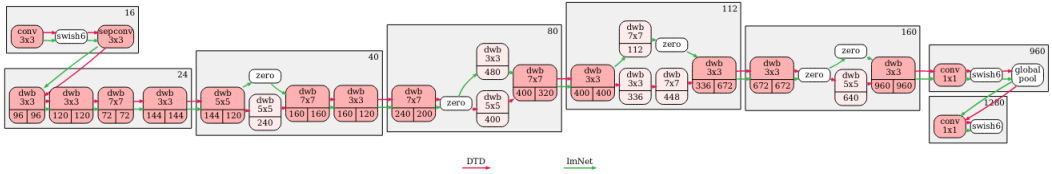
$\lambda_{\text{temp}} \to 0$

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



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

Multi-domain tasks



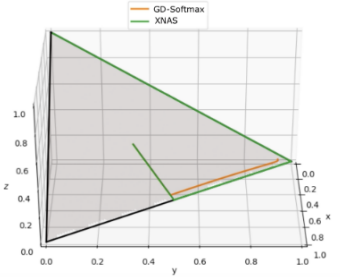
XNAS

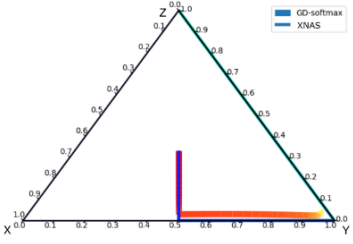
Algorithm 1 XNAS for a single forecaster

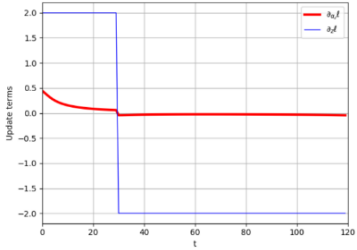
Input: The learning rate η,
Loss-gradient bound $\mathcal{L}$,
Experts predictions $\{f_{t,i}\}_{i=1}^{N}$ $\forall t = 1, \ldots, T$
Init: $I_0 = \{1, \ldots, N\}$, $v_{0,i} \leftarrow 1$, $\forall i \in I_0$
for rounds $t = 1, \ldots, T$ **do**
  Update $\boldsymbol{\omega}$ by descending $\nabla_{\boldsymbol{\omega}} \ell_{\text{train}}(\boldsymbol{\omega}, \boldsymbol{v})$
  $p_t \leftarrow \frac{\sum_{i \in I_{t-1}} v_{t-1,i} \cdot f_{t-1,i}}{\sum_{i \in I_{t-1}} v_{t-1,i}}$ #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

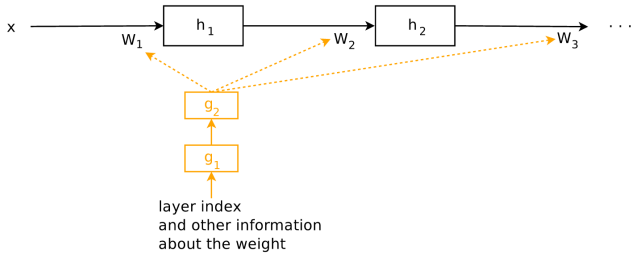
Definition

Given a set Λ.

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

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

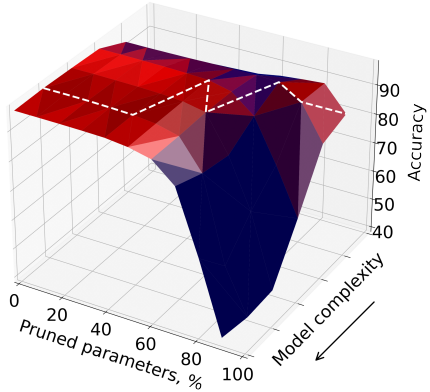
where $\mathbb{R}^u$ is a set of hypernetwork parameters.



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:

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

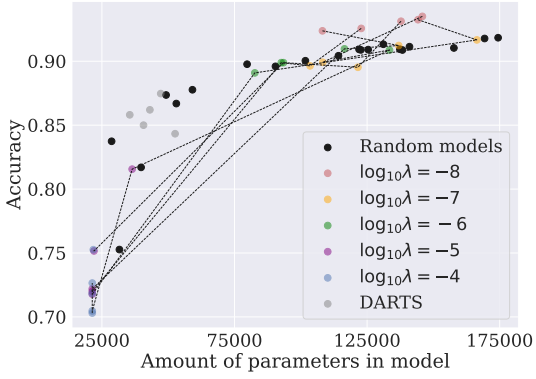
Our proposal

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



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

Uncertainity estiamtion, Depeweg et al., 2017[1]

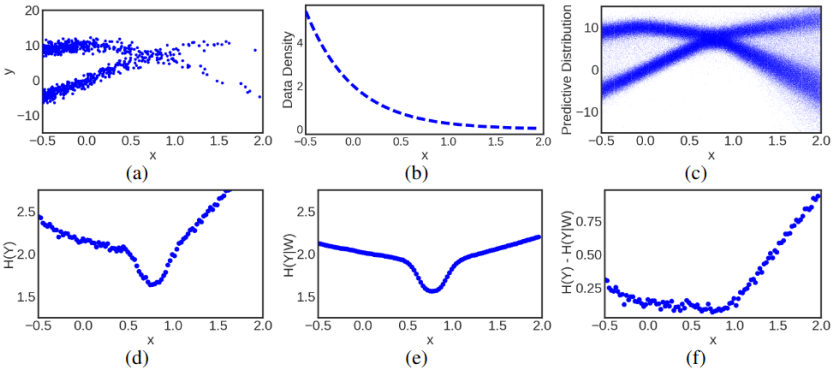


Figure 2. Uncertainty decomposition on bimodal data. (a): Raw data. (b): Density of x in data. (c): Predictive distribution: $p(y_\star|x_\star)$. (d): Estimate of $\mathrm{H}(y_\star|x_\star)$. (e): Estimate of $\mathbf{E}_{q(\mathcal{W})}\left[\mathrm{H}(y_\star|x_\star,\mathcal{W})\right]$. (f): Estimate of entropy reduction $\mathrm{H}(y_\star|x_\star) - \mathbf{E}_{q(\mathcal{W})}\left[\mathrm{H}(y_\star|x_\star,\mathcal{W})\right]$.

[1] See the talk of Maxim Panov: https://intsystems.github.io/materials/seminars/.

Uncertainity estimation using ensembles

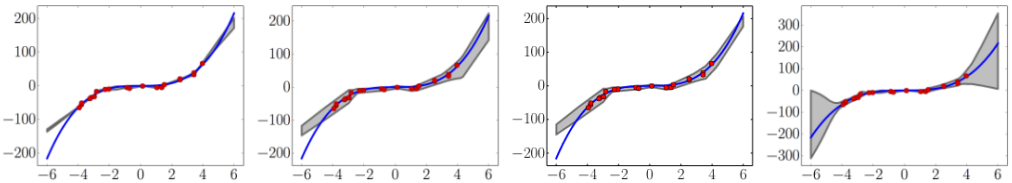
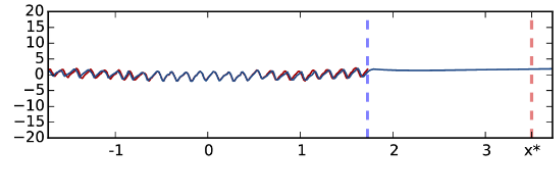


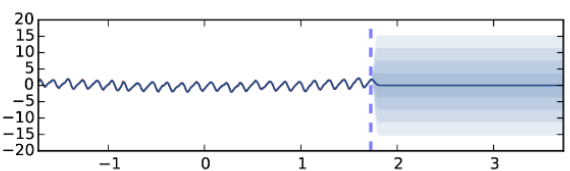
Figure 1: Results on a toy regression task: x-axis denotes x. On the y-axis, the blue line is the *ground truth* curve, the red dots are observed noisy training data points and the gray lines correspond to the predicted mean along with three standard deviations. Left most plot corresponds to empirical 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.

Uncertainity estimation: MC Dropout

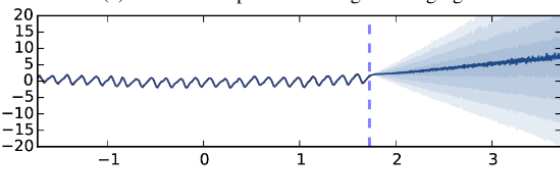
- Variational dropout: dropout can be considered as an ensemble of models[2]
- MC Dropout: variance of prediction model is an approximation of uncertainity
  - **Which type of uncertainity?**



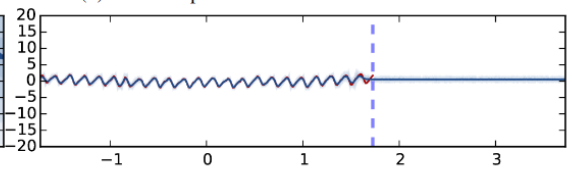
(a) Standard dropout with weight averaging



(b) Gaussian process with SE covariance function



(c) MC dropout with ReLU non-linearities



(d) MC dropout with TanH non-linearities

[2] See the talk of Eduard Vladimirov

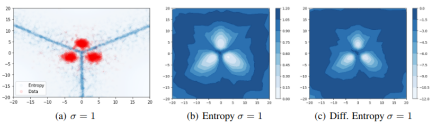
Prior network[3]

Basic idea: instead of training a probabilistic model that approximates $\int_{\mathrm{w}} p(\mathrm{y}|\mathrm{X},\mathrm{w})p(\mathrm{w})d\mathrm{w}$ we extned it in the following way:

$$\int_{\mathrm{w},\boldsymbol{\mu}} p(\mathrm{y}|\boldsymbol{\mu})p(\boldsymbol{\mu}|\mathrm{w},\mathrm{X})p(\mathrm{w})d\mathrm{w}d\boldsymbol{\mu}.$$

The model is trained with prior:

- $\boldsymbol{\mu} \sim \mathrm{Dir}(\boldsymbol{\alpha}, t), t \approx 0$ for ordinary dataset objects;
- $\boldsymbol{\mu} \sim \mathcal{U}$ for OOD objects.



(a) $\sigma = 1$ (b) Entropy $\sigma = 1$ (c) Diff. Entropy $\sigma = 1$

(d) $\sigma = 4$ (e) Entropy $\sigma = 4$ (f) Diff. Entropy $\sigma = 4$

[3] See the talk of Gregory Ksenofontov

Neural ensemble search

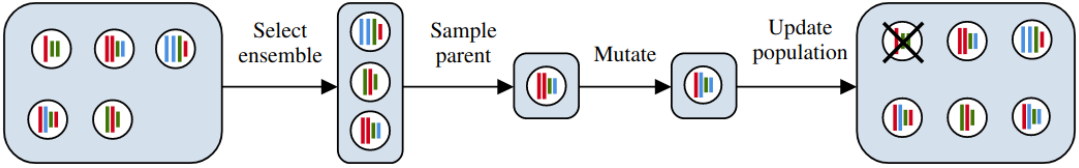


Figure 3: Illustration of one iteration of NES-RE. Network architectures are represented as colored bars of different lengths illustrating different layers and widths. Starting with the current population, ensemble selection is applied to select parent candidates, among which one is sampled as the parent.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

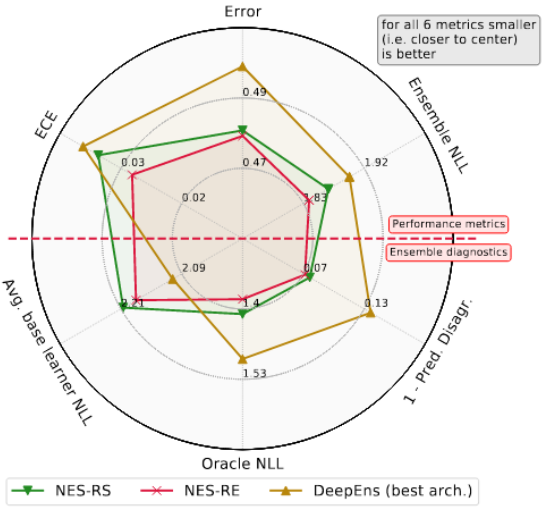
Neural ensemble search

Algorithm 1: NES with Regularized Evolution

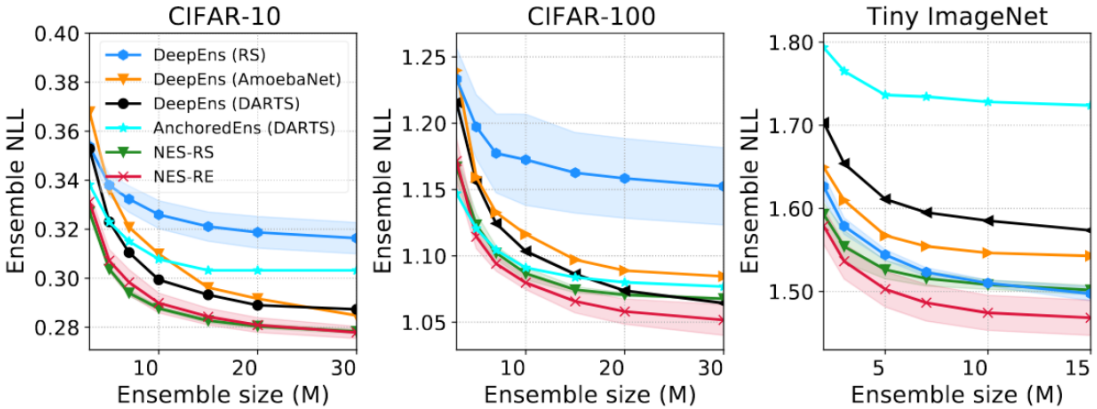
Data: Search space $\mathcal{A}$; ensemble size M; comp. budget K; $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}$; population size P; number of parent candidates m.

Sample P architectures $\alpha_1, \ldots, \alpha_P$ independently and uniformly from $\mathcal{A}$.
Train each architecture α_i using $\mathcal{D}_{\text{train}}$, and initialize $\mathfrak{p} = \mathcal{P} = \{f_{\theta_1,\alpha_1}, \ldots, f_{\theta_P,\alpha_P}\}$.
while $|\mathcal{P}| < K$ **do**
    Select m parent candidates $\{f_{\widetilde{\theta}_1,\widetilde{\alpha}_1}, \ldots, f_{\widetilde{\theta}_m,\widetilde{\alpha}_m}\} = \texttt{ForwardSelect}(\mathfrak{p}, \mathcal{D}_{\text{val}}, m)$.
    Sample uniformly a parent architecture α from $\{\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_m\}$. `// α stays in 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 $\{f_{\theta_1^*,\alpha_1^*}, \ldots, f_{\theta_M^*,\alpha_M^*}\} = \texttt{ForwardSelect}(\mathcal{P}, \mathcal{D}_{\text{val}}, M)$ by forward step-wise selection without replacement.
return ensemble $\texttt{Ensemble}(f_{\theta_1^*,\alpha_1^*}, \ldots, f_{\theta_M^*,\alpha_M^*})$

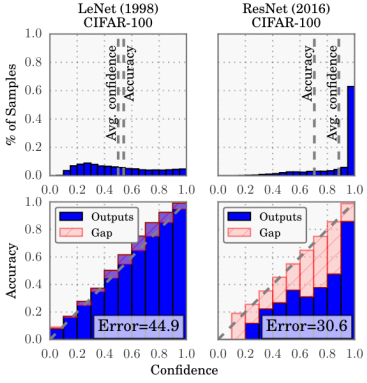
Neural ensemble search



Neural ensemble search



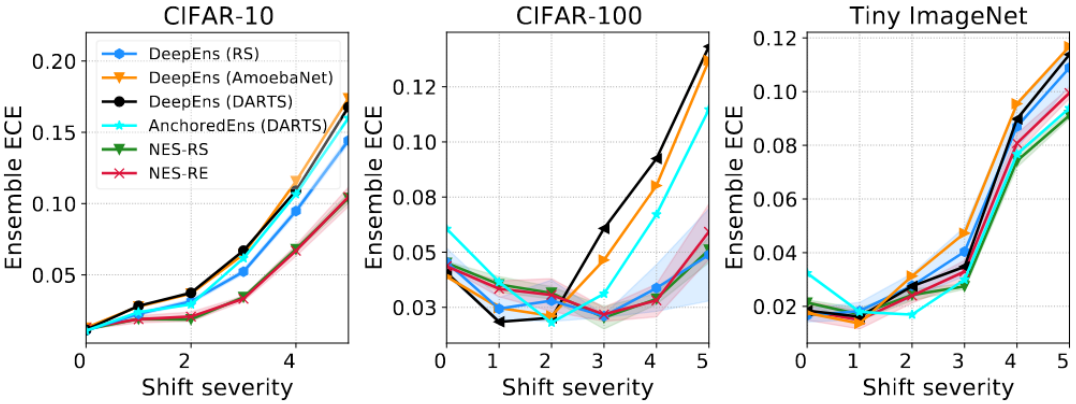
Neural ensemble search



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

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

Neural ensemble 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.
- Depeweg, Stefan, et al. "Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning." International conference on machine learning. PMLR, 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
- Kingma, Durk P., Tim Salimans, and Max Welling. "Variational dropout and the local reparameterization trick." Advances in neural information processing systems 28 (2015).
- Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." international conference on machine learning. PMLR, 2016.
- Malinin, Andrey, and Mark Gales. "Predictive uncertainty estimation via prior networks." Advances in neural information processing systems 31 (2018).