A Survey on Predicting Generalization in Deep Learning (PGDL) Competition

Jiaxi Zhao NUS

Sep 2021

Overview of the competition design

data $\rightarrow F(\Phi) \longleftrightarrow E(L(D_{test}) - L(D_{valid}))$

- **Data point**: trained network with unknown algorithm
  - Sequential (no skip-connection)
  - Trained until they reach the interpolation regime
  - Architecture and trained weight provided
  - Two classed: VGG-like models and Network in network   0~20%
  - Good performance in reality, interpolate fast while has large generalization error range, reasonable number of hyperparameters 7
- All the 8 tasks we considered in the competition are image recognition tasks with different variations of convolutional neural networks.

- Neurips 2020 competition: Predicting generalization in deep learning. arXiv preprint arXiv:2012.07976, 2020.

Overview of the competition design

- **Phase 0**: Public data was given to the competitors: test on Task 1 & 2.
- **Phase 1**: First online leaderboard, accessible at the beginning of the competion (also called *public leaderboard*). This leaderboard is composed of Task 4 and Task 5 and was used to compute the scores displayed on the leaderboard for the first phase of the competition.
- **Phase 2**: Private leaderboard, only accessible in the last phase of the competition, where competitors can upload their very best metrics. Winners are determined only on their score on this leaderboard (to prevent overfitting of the public leaderboard, as usual). This phase is composed of Task 6, Task 7, Task 8, and Task 9.

Examples of task (Task 1)

- **Model**: VGG-like models, with 2 or 6 convolutional layers [conv-relu-conv-relu-maxpool] x 1 or x 3. One or two dense layers of 128 units on top of the model.When dropout is used, it is added after each dense layer.
- **Dataset**: CIFAR-10 [15](10 classes, 3 channels).
- **Training**: Trained for at most 1200 epochs, learning rate is multiplied by 0.2 after 300, 600 and 900 epochs. Cross entropy and SGD with momentum 0.9. Initial learning rate of 0.001
- **Hparams**: Number of lters of the last convolutional layer in [256, 512]. Dropout probability in [0, 0.5]. Number of convolutional blocks in [1, 3]. Number of dense layers (excluding the output layer) in [1, 2]. Weight decay in [0.0, 0.001]. Batch size in [8, 32, 512].

Examples of task (Task 9)

- **Model**: Network in Network.
- **Dataset**: CIFAR-10, with the standard data augmentation (random horizontal and random crops after padding by 4 pixels.
- **Training**: Trained for at most 1200 epochs, learning rate is multiplied by 0.2 after 300, 600 and 900 epochs. Cross entropy and SGD with momentum 0.9. Initial learning rate of 0.01.
- **Hparams**: Number of lters in the convolutional layers in [256, 512], Number of convolutional layers in [9, 12], dropout probability in [0.0, 0.25], weight decay in [0.0, 0.001], batch size in [32, 512].

Three categories of generalization estimators

- Principled complexity measures
- Data augmentation
- Intermediate Representation Analysis

Winner of the competition

- *Interpex*
- *Always Generalize*
- *BrAIn*

$1-g$

$M(NN, Task) \to g$

calculate → (

vc

Rank	User / Team name	Score
1	**interpex**	**22.92**
2	**Always Generalize**	**10.16**
3	**BrAIn**	**9.99**
4	spn	7.99
5	Vashisht (user name)	6.51
6	Tuebingen	6.39
7	samiul	5.98
8	smeznar (user name)	5.94
9	FZL	5.60
10	IBM-NTUST	4.92

- Methods and Analysis of The First Competition in Predicting Generalization of Deep Learning, Proceedings of Machine Learning Research 133:170{190, 2021 NeurIPS 2020 Competition and Demonstration Track

First Place: *interpex*

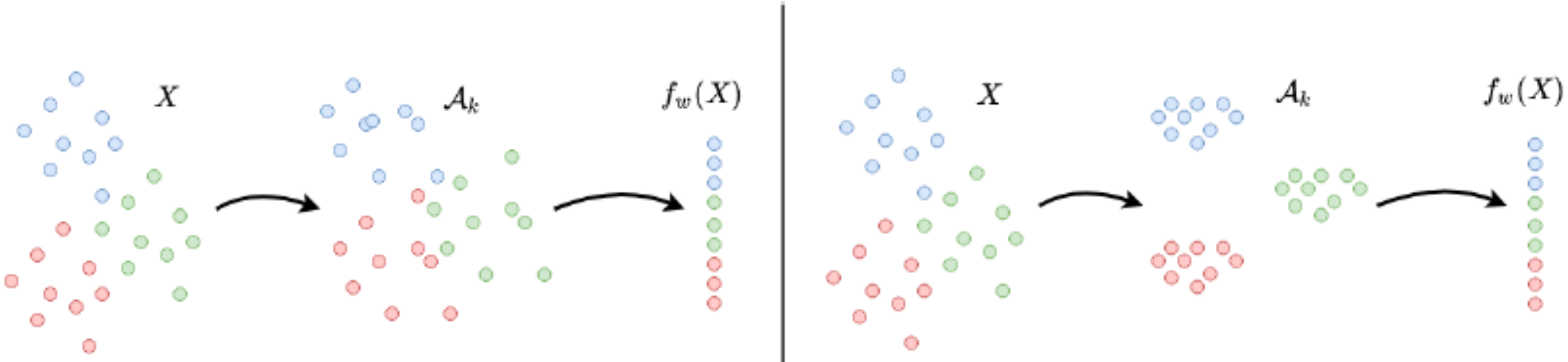
- Consistency

$S_i \downarrow$ In.

i: index
k: layer.

$$\tilde{\mathcal{A}}_k = \Phi(\mathcal{A}_k) \qquad \mathcal{S}_i = \left(\frac{1}{n_i}\sum_{i=1}^{n_i} |\tilde{\mathcal{A}}_k^i - \mu_{\tilde{\mathcal{A}}_k^i}|^p\right)^{1/p} \qquad \mathcal{M}_{i,j} = ||\mu_{\tilde{\mathcal{A}}_k^i} - \mu_{\tilde{\mathcal{A}}_k^j}||_p$$

$$\mathcal{C}_k = \frac{1}{\kappa}\sum_{i=1}^{\kappa} \max_{i \neq j} \frac{\mathcal{S}_i + \mathcal{S}_i}{\mathcal{M}_{i,j}}$$



(a) (b)

First Place: *interpex*

- Robustness:
  - Use Mixup

$$\tilde{A}^i = \lambda A_1^i + (1-\lambda) A_2^i$$

$$\tilde{y}^i = y^i$$

$$\mathcal{C}_k = \sum_{i=1}^{\kappa} \frac{1}{N_i} \sum_{n=1}^{N_i} I\left(f_{w,k}(\tilde{\mathcal{A}}_{k,n}^i)[y_i] \leq \max_{j \neq \tilde{y}_n} f_{w,k}(\tilde{\mathcal{A}}_{k,n}^i)[j]\right) = \sum_{i=1}^{\kappa} \hat{L}\left(f_{w,k}(\tilde{\mathcal{A}}_k^i)\right)$$

First Place: *interpex*

i j

- Separability

$$D_{(i,j),k} = \{\mathcal{A}'_k \mid f_{w,k}(\mathcal{A}'_k)[i] = f_{w,k}(A'_k)[j]\}$$

$$d_{f_{w,k},(i,j)}(\mathcal{A}'_k) = \frac{f_{w,k}(\mathcal{A}'_k)[i] - f_{w,k}(\mathcal{A}'_k)[j]}{||\nabla_{\mathcal{A}'_k} f_{w,k}(\mathcal{A}'_k)[i] - \nabla_{\mathcal{A}'_k} f_{w,k}(\mathcal{A}'_k)[j]||_2}$$

$$\mathcal{C}_k = -\theta(d_{f_{w,k},(i,j)}(\mathcal{A}'_k))$$

Second Place: *Always Generalize*

- For every sample in a randomly sampled subset of the training set, the input is augmented with a collection of augmentations and the class prediction of each output is compared to that of the original image.
- Sophisticated augmentation procedure provides better results.

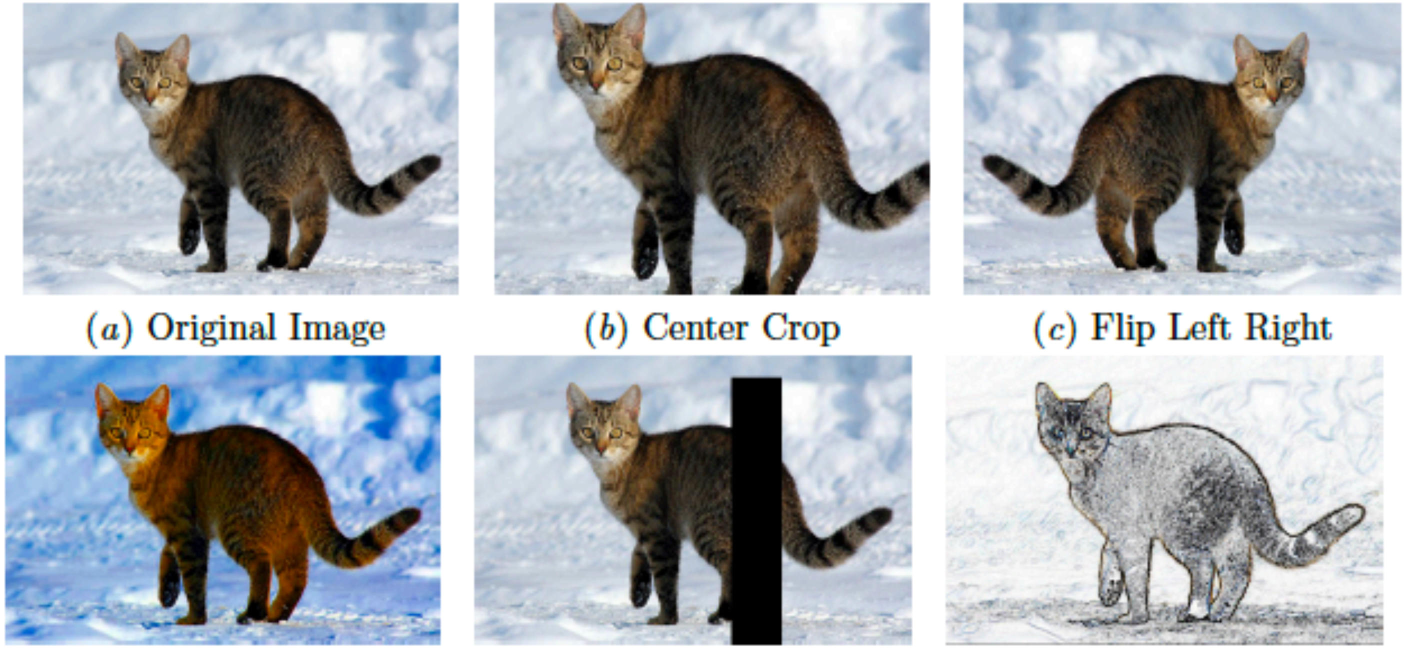
Second Place: *Always Generalize*

- For every sample in a randomly sampled subset of the training set, the input is augmented with a collection of augmentations and the class prediction of each output is compared to that of the original image.
- Sophisticated augmentation procedure provides better results.
- The penalty is determined based on the strength of the augmentation. The strength of augmentation is determined by the ability of the augmentation to change the texture in the input. Augmentations that do not alter the texture of the image, but to tend to alter the shape in the image, are weak.

λ_{flip}	$\lambda_{saturation}$	λ_{crop_resize}	λ_{sobel}	$\lambda_{brightness}$	$\lambda_{flip+saturation}$	λ_{cutout}	λ_v	Public Score	Private Score
6	1	3	2	1	12	0	3	33.67	9.16
6	1	2	3	1	9	0	0	40.9	9.25
6	1	2	3	1	12	2	0	**41.8**	**10.6**

Second Place: *Always Generalize*

- The list of augmentations used were Flip, Random Saturation, Crop and Resize, Brightness, Random Erasing (Zhong et al.), Sobel lter (Kanopoulos et al., 1988) and Virtual Adversarial Perturbation (Miyato et al., 2018).



(a) Original Image (b) Center Crop (c) Flip Left Right
(d) Random Saturation (e) Random Erasing (f) Sobel Filter

Second Place: *Always Generalize*

Algorithm 1: Proposed metric calculation

Input: Consider a model θ; x is the input; λ is the penalty for an augmentation.
Result: Generalization metric ϕ

```
φ = 0 ;
forall samples x do
    x' = Augment(x) ;
    if arg max_ŷ P_θ(ŷ|x) = arg max_ŷ P_θ(ŷ|x') then
        φ = φ − | max_ŷ P_θ(ŷ|x) − P_θ(ŷ|x') |
    else
        φ = φ − λ
    end
end
```

Performance of theoretic bounds

- Failure of Norm & Margin-based measure
- Success of Sharpness-based measure

- Fantastic generalization measures and where to nd them. In International Conference on Learning Representations, 2020.

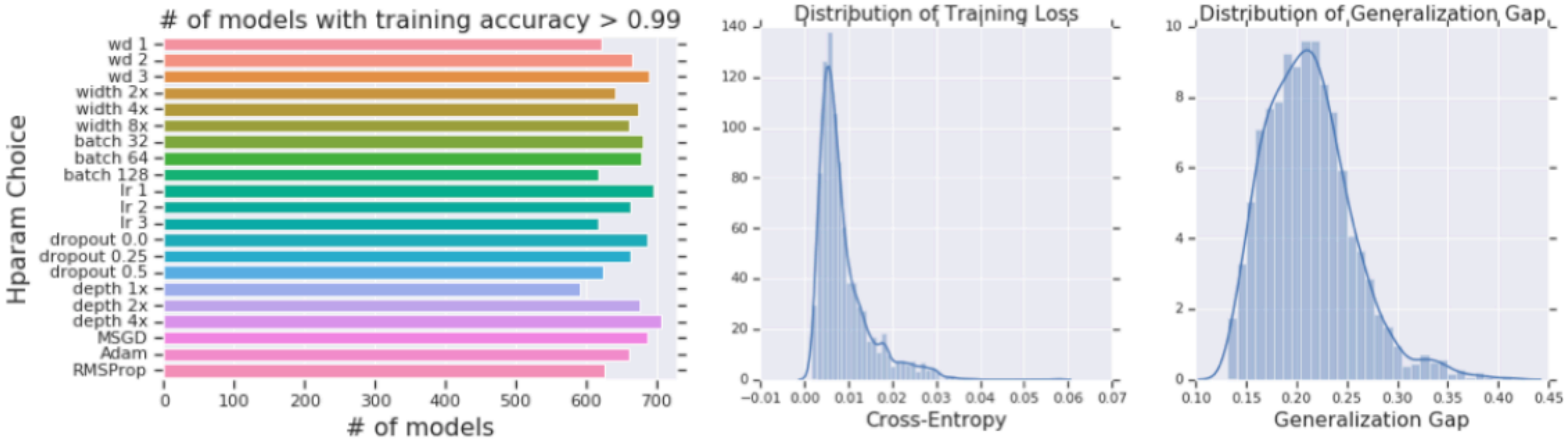


Figure 1: **Left**: Number of models with training accuracy$>99\%$ for each hyperparameter type. **Mid.**: Distribution of training cross-entropy (that of training error in Fig. 4). **Right**: Distribution of generalization gap.

		batch size	dropout	learning rate	depth	optimizer	weight decay	width	overall τ	Ψ
Corr	vc dim 19	0.000	0.000	0.000	-0.909	0.000	0.000	-0.171	-0.251	-0.154
	# params 20	0.000	0.000	0.000	-0.909	0.000	0.000	-0.171	-0.175	-0.154
	$1/\gamma$ (22)	0.312	-0.593	0.234	0.758	0.223	-0.211	0.125	0.124	0.121
	entropy 23	0.346	-0.529	0.251	0.632	0.220	-0.157	0.104	0.148	0.124
	cross-entropy 21	0.440	-0.402	0.140	0.390	0.149	0.232	0.080	0.149	0.147
	oracle 0.02	0.380	0.657	0.536	0.717	0.374	0.388	0.360	0.714	0.487
	oracle 0.05	0.172	0.375	0.305	0.384	0.165	0.184	0.204	0.438	0.256
	canonical ordering	0.652	0.969	0.733	0.909	-0.055	0.735	0.171	N/A	N/A
									$\vert\mathcal{S}\vert = 2$	$\min \forall \vert\mathcal{S}\vert$
MI	vc dim	0.0422	0.0564	0.0518	0.0039	0.0422	0.0443	0.0627	0.00	0.00
	# param	0.0202	0.0278	0.0259	0.0044	0.0208	0.0216	0.0379	0.00	0.00
	$1/\gamma$	0.0108	0.0078	0.0133	0.0750	0.0105	0.0119	0.0183	0.0051	0.0051
	entropy	0.0120	0.0656	0.0113	0.0086	0.0120	0.0155	0.0125	0.0065	0.0065
	cross-entropy	0.0233	0.0850	0.0118	0.0075	0.0159	0.0119	0.0183	0.0040	0.0040
	oracle 0.02	0.4077	0.3557	0.3929	0.3612	0.4124	0.4057	0.4154	0.1637	0.1637
	oracle 0.05	0.1475	0.1167	0.1369	0.1241	0.1515	0.1469	0.1535	0.0503	0.0503
	random	0.0005	0.0002	0.0005	0.0002	0.0003	0.0006	0.0009	0.0004	0.0001

Table 1: Numerical Results for Baselines and Oracular Complexity Measures

	CIFAR VGG	SVHN NiN	CINIC FCN bn	CINIC FCN	Flowers NiN	Pets NiN	Fashion VGG	CIFAR NiN
Margin†	13.59	16.32	2.03	2.99	0.33	1.24	0.45	5.45
SN-Margin† [3]	5.28	3.11	0.24	2.89	0.10	1.00	0.49	6.15
GN-Margin 1st [22]	3.53	35.42	26.69	6.78	4.43	1.61	1.04	13.49
GN-Margin 8th [22]	0.39	31.81	7.17	1.70	0.17	0.79	2.12	1.16
TV-GN-Margin 1st [22]	19.22	36.90	31.70	16.56	4.67	4.20	0.16	**25.06**
TV-GN-Margin 8th [22]	38.18	41.52	6.59	16.70	0.43	5.65	**2.35**	10.11
kV-Margin† 1st	5.34	26.78	**37.00**	**16.93**	**6.26**	2.07	1.82	15.75
kV-Margin† 8th	30.42	26.75	6.05	15.19	0.78	1.76	0.33	2.26
kV-GN-Margin† 1st	17.95	44.57	30.61	16.02	4.48	3.92	0.61	21.20
kV-GN-Margin† 8th	**40.92**	**45.61**	6.54	15.80	1.13	**5.92**	0.29	8.07

Table 1: **Mutual information scores on PGDL tasks.** We compare different margins across tasks in PGDL. The first and second rows indicate the datasets and the architecture types used by tasks. The methods that are supported with theoretical bounds are marked with †. Our k-variance normalized margins outperform the baselines in 6 out of 8 tasks in PGDL dataset.

References

- Yiding Jiang, Pierre Foret, Scott Yak, Daniel M Roy, Hossein Mobahi, Gintare Karolina Dziugaite, Samy Bengio, Suriya Gunasekar, Isabelle Guyon, and Behnam Neyshabur. Neurips 2020 competition: Predicting generalization in deep learning. arXiv preprint arXiv:2012.07976, 2020.
- Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to nd them. In International Conference on Learning Representations, 2020.

References

- Parth Natekar and Manik Sharma. Representation based complexity measures for predicting generalization in deep learning. arXiv preprint arXiv:2012.02775, 2020.
- Sumukh Aithal K, Dhruva Kashyap, and Natarajan Subramanyam. Robustness to augmentations as a generalization metric, 2021.
- Yiding Jiang, Pierre Foret, Scott Yak, Daniel M Roy, Hossein Mobahi, Gintare Karolina Dziugaite, Samy Bengio, Suriya Gunasekar, Isabelle Guyon, and Behnam Neyshabur. Methods and Analysis of The First Competition in Predicting Generalization of Deep Learning, Proceedings of Machine Learning Research 133:170{190, 2021 NeurIPS 2020 Competition and Demonstration Track