EDITABLE NEURAL NETWORKS

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, Artem Babenko

Problem

- Correcting model's behavior is important.
- Given a mistake of the model, how can we update the model parameters to fix it without changing the current performance?

Edit operation

- Model: $f(x, \theta)$
- Example to edit:(x , y_{ref})
- Definition:

$$Edit_{\alpha}^{k}(\theta, l_e, k) = \begin{cases} \theta, & \text{if } l_e(\theta) \leq 0 \text{ or } k = 0 \\ Edit_{\alpha}^{k-1}(\theta - \alpha \cdot \nabla_{\theta} l_e(\theta), l_e), & \text{otherwise} \end{cases}$$

Where $l_e(\hat{\theta}) = \max_{y_i} \log p(y_i|x, \hat{\theta}) - \log p(y_{ref}|x, \hat{\theta})$

and α is a parameter of a certain editor (GD, RMSprop, Adam, …)

Edit operation

- How to evaluate an editor?

  + **Drawdown**: mean absolute difference of classification error before and after performing an edit. Smaller drawdown indicates better editor locality.

  + **Success Rate**: a rate of edits, for which editor succeeds in under k=10 gradient steps

  + **Num Steps**: an average number of gradient steps needed to perform a single edit

Effect of different editors

- Baseline model: Resnet-18.
- Data: CIFAR-10.
- Examples to edit:
  + Sample 1000 images from CIFAR-10.
  + For each image, assign a random label.

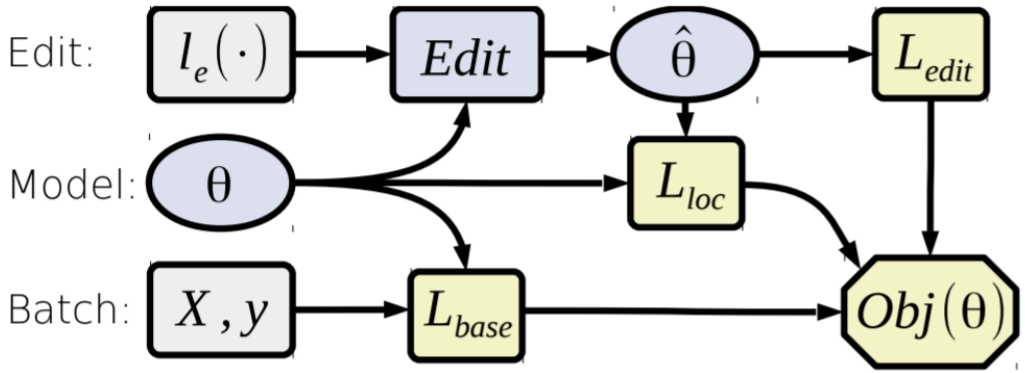
Edit operation

- Pretrain Resnet-18 and execute edits:

Editor Function	GD	Scaled GD	RProp	RMSProp	Momentum	Adam
Drawdown	3.8%	2.81%	1.99%	**1.77%**	2.42%	19.4%
Success Rate	98.8%	99.1%	100%	100%	96.0%	100%
Num Steps	3.54	3.91	2.99	3.11	5.60	3.86

Editable training

- Goal: Make the model get "preprared" for future edits.



$$Obj(\theta, l_e) = \mathcal{L}_{base}(\theta) + c_{edit} \cdot \mathcal{L}_{edit}(\theta) + c_{loc} \cdot \mathcal{L}_{loc}(\theta)$$

$$\mathcal{L}_{edit}(\theta) = max(0, l_e(Edit^k_\alpha(\theta, l_e))$$

$$\mathcal{L}_{loc}(\theta) = \underset{x \sim p(x)}{E} D_{KL}(p(y|x, \theta)||p(y|x, Edit^k_\alpha(\theta, l_e)))$$

Editable training

- Results:

Training Procedure	Editor Function	Editable Layers	Test Error Rate	Test Error Drawdown	Success Rate	Num Steps
Baseline Training	GD	All	6.3%	3.8%	98.8%	3.54
	RMSProp	Chain 3	6.3%	1.77%	100%	3.11
Editable $c_{loc} = 0.01$	GD	All	6.34%	1.42%	100%	3.39
	GD	Chain 3	6.28%	1.44%	100%	2.82
	RMSProp	Chain 3	**6.31%**	**0.86%**	100%	4.13

Editable fine-tuning

- Editable training takes more time than normal training.
- How to avoid training from scratch?
- Solution:

+ Preserve the original "achievement" by choosing $\mathcal{L}_{base}(\theta)$ to be the KL divergence between the original predictions and finetuned models' predictions.

+ Add some extra layers to better deal with edits.

Editable fine-tuning

- Results:

Model Architecture	Training Procedure	Editable Layers	Test Error Rate	Mean Drawdown	Success Rate	Num Steps
ResNet18	Pre-trained	Chain 3	30.95%	3.89%	99.8%	3.582
	Pre-trained	Extra layer	30.95%	9.18%	100%	4.272
	Distillation	Extra layer	30.75%	2.80%	100%	2.63
	Editable	Chain 3	30.53%	3.78%	99.8%	3.616
	Editable	Extra layer	**30.61%**	**0.57%**	100%	3.388
DenseNet169	Pre-trained	Chain 3	25.49%	5.20%	100%	2.551
	Pre-trained	Extra layer	25.47%	9.05%	100%	3.874
	Distillation	Extra layer	24.33%	1.67%	100%	2.822
	Editable	Chain 3	24.32%	4.47%	100%	2.556
	Editable	Extra layer	**24.38%**	**0.96%**	100%	2.970