Guided Learning of Nonconvex Models through Successive Functional Gradient Optimization

Rie Johnson* and **Tong Zhang**†

RJ Research Consulting*
Hong Kong University of Science and Technology†

Training Deep Neural Networks

Challenge: nonconvex optimization problem

- converge to local minimum with sub-optimal generalization

Training Deep Neural Networks

Challenge: nonconvex optimization problem

- converge to local minimum with sub-optimal generalization

This work:

- how to find a local minimum with better generalization

Training Deep Neural Networks

Challenge: nonconvex optimization problem

- converge to local minimum with sub-optimal generalization

This work:

- how to find a local minimum with better generalization

Idea:

- restricting search space leads to better generalization

Method:

- guided functional gradient training (guide restricts search space)

Problem Formulation

Supervised learning:

$$\hat{\theta} = \arg\min_{\theta} \left[\frac{1}{|S|} \sum_{(x,y)\in S} L(f(\theta; x), y) + R(\theta) \right].$$

- x: input
- y: output
- $f(\theta; x)$: vector function to predict y from x.
- θ: model parameter.
- S: training data
- L: loss function
- $R(\theta)$: regularizer such as weight-decay $\lambda\|\theta\|_2^2$

Example:

- K-class classification where $y \in \{1, 2, \ldots, K\}$
- $f(\theta; x)$ is K-dimensional, linked to conditional probabilities

GULF: GUided Learning through Functional gradient

General GULF Procedure (f: model we are training):

- (Step 1) **Generate a guide function** f^*
  - apply functional gradient to reduce the loss of the current model f,
  - f^* is an improvement over f in terms of loss but not too far from f.
- (Step 2) **Move the model** f **towards the guide function** f^*
  - using SGD according to some distance measure.
  - guide serves as a restriction of model parameter search space

GULF: GUided Learning through Functional gradient

General GULF Procedure (f: model we are training):

- (Step 1) **Generate a guide function** f^*
  - apply functional gradient to reduce the loss of the current model f,
  - f^* is an improvement over f in terms of loss but not too far from f.
- (Step 2) **Move the model** f **towards the guide function** f^*
  - using SGD according to some distance measure.
  - guide serves as a restriction of model parameter search space

Motivation:

- functional gradient learning of additive models in gradient boosting (Friedman, 2001), known to have good generalization
- natural idea: use functional gradient learning to guide SGD

Result:

- worse training error but better test error

Step 1: Move Guide Ahead

We formulate Step 1 as

$$f^*(x,y) := \operatorname*{argmin}_q \left[\underbrace{D_h(q, f(x))}_{\text{guide near previous model}} + \alpha \underbrace{\nabla L_y(f(x))^\top q}_{\text{functional gradient}} \right], \qquad (1)$$

where α is a meta-parameter, and the Bregman divergence D_h is defined by

$$D_h(u, v) = h(u) - h(v) - \nabla h(v)^\top (u - v).$$

Step 1: Move Guide Ahead

We formulate Step 1 as

$$f^*(x,y) := \operatorname*{argmin}_q \left[\underbrace{D_h(q, f(x))}_{\text{guide near previous model}} + \alpha \underbrace{\nabla L_y(f(x))^\top q}_{\text{functional gradient}} \right], \qquad (1)$$

where α is a meta-parameter, and the Bregman divergence D_h is defined by

$$D_h(u, v) = h(u) - h(v) - \nabla h(v)^\top (u - v).$$

(1) is equivalent to mirror descent in function space.

$$\nabla h(\underbrace{f^*(x,y)}_{\text{new guide}}) = \nabla h(\underbrace{f(x)}_{\text{previous model}}) - \alpha \underbrace{\nabla L_y(f(x))}_{\text{functional gradient}}. \qquad (2)$$

Step 2: Following the Guide

Update network parameter θ to reduce

$$\underbrace{\left\langle D_h(f(\theta;x), f^*(x,y))\right\rangle_{(x,y)\in S}}_{\text{next model near guide}} + \underbrace{R(f)}_{\text{regularizer}} \tag{3}$$

with SGD repeatedly to improve model $f(\theta;\cdot)$:

$$\theta \leftarrow \theta - \eta \nabla_\theta \left[\left\langle D_h(f(\theta;x), f^*(x,y))\right\rangle_{(x,y)\in B} + R(\theta)\right], \tag{4}$$

where B is a mini-batch sampled from a training set S.

Step 2: Following the Guide

Update network parameter θ to reduce

$$\underbrace{\left\langle D_h(f(\theta;x), f^*(x,y))\right\rangle_{(x,y)\in S}}_{\text{next model near guide}} + \underbrace{R(f)}_{\text{regularizer}} \tag{3}$$

with SGD repeatedly to improve model $f(\theta;\cdot)$:

$$\theta \leftarrow \theta - \eta\nabla_\theta\left[\left\langle D_h(f(\theta;x), f^*(x,y))\right\rangle_{(x,y)\in B} + R(\theta)\right], \tag{4}$$

where B is a mini-batch sampled from a training set S. Remarks:

- $f(\theta;\cdot)$: move towards guide function f^* in Bregman divergence
- $R(\theta)$: regularization term
- $f^*(x,y)$: guide to restrict SGD search space $\rightarrow$ better generalization

Convergence Result

Define α-*regularized loss*

$$\ell_\alpha(\theta) := \left\langle L\left(f(\theta; x), y\right)\right\rangle_{(x,y)\in S} + \frac{1}{\alpha}R(\theta). \tag{5}$$

Theorem

Under apporiate assumptions, consider the GULF algorithm with a sufficiently small α and η.
Assume that θ_{t+1} is an improvement of θ_t with respect to minimizing

$$Q_t(\theta) := \left\langle D_h(f(\theta; x), f^*(x, y))\right\rangle_{(x,y)\in S} + R(\theta)$$

so that $Q_t(\theta_{t+1}) \leq Q_t(\theta_t - \eta\nabla Q_t(\theta_t))$, then
GULF finds a local minimum of $\ell_\alpha(\cdot)$:

$$\nabla\ell_\alpha(\theta_t) \to 0.$$

Remarks

GULF is very different from standard training of α-regularized loss.

- better generalization from guide to restrict the search space

Remarks

GULF is very different from standard training of α-regularized loss.

- better generalization from guide to restrict the search space

For $h = L_y(f)$ with cross-entropy loss for classification, Step 2 becomes self-distillation parameter update:

$$\theta \leftarrow \theta - \eta \nabla_\theta \Big\langle (1-\alpha) \underbrace{L(f_\theta, \mathrm{prob}(f_{\theta_t}))}_{\text{distillation with old model}} + \alpha \underbrace{L_y(f_\theta)}_{\text{training loss}} \Big\rangle_{(x,y)\in S}$$

Remarks

GULF is very different from standard training of α-regularized loss.

- better generalization from guide to restrict the search space

For $h = L_y(f)$ with cross-entropy loss for classification, Step 2 becomes self-distillation parameter update:

$$\theta \leftarrow \theta - \eta \nabla_\theta \left\langle (1-\alpha) \underbrace{L(f_\theta, \mathrm{prob}(f_{\theta_t}))}_{\text{distillation with old model}} + \alpha \underbrace{L_y(f_\theta)}_{\text{training loss}} \right\rangle_{(x,y)\in S}$$

Our result gives a convergence proof of self-distillation, and

generalizes it to other loss functions.

Empirical Results

Methods compared:

- (ini:random) GULF starting with random initialization
- (ini:base) GULF starting with initialization by regular training
- (base-λ/α) standard training with α-regularized loss
- (base-loop) standard training with learning rate resets
- label-smoothing: use noisy label

Empirical Results

Methods compared:

- (ini:random) GULF starting with random initialization
- (ini:base) GULF starting with initialization by regular training
- (base-λ/α) standard training with α-regularized loss
- (base-loop) standard training with learning rate resets
- label-smoothing: use noisy label

First three converge to local minimum solutions of α-regularized loss.

Result

			C10	C100	SVHN	
1	baselines	base model	6.42	30.90	1.86	1.64
2		base-λ/α	6.60	30.24	1.78	1.67
3		base-loop	6.20	30.09	1.93	**1.53**
4		label smooth	6.66	30.52	1.71	1.60
5	GULF2	ini:random	5.91	**28.83**	1.71	**1.53**
6		ini:base	**5.75**	29.12	**1.65**	1.56

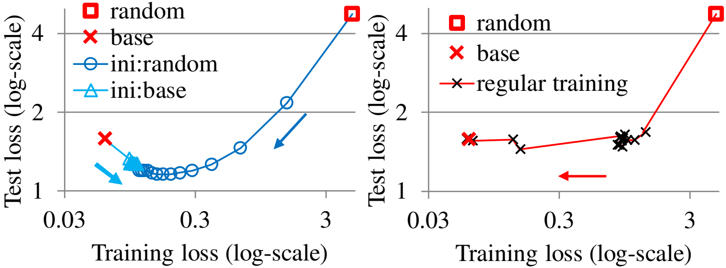
Table: Test error (%). Median of 3 runs. Resnet-28 (0.4M parameters) for CIFAR10/100, and WRN-16-4 (2.7M parameters) for SVHN. Two numbers for SVHN are without and with dropout.

Result

			C10	C100	SVHN	
1	baselines	base model	6.42	30.90	1.86	1.64
2		base-λ/α	6.60	30.24	1.78	1.67
3		base-loop	6.20	30.09	1.93	**1.53**
4		label smooth	6.66	30.52	1.71	1.60
5	GULF2	ini:random	5.91	**28.83**	1.71	**1.53**
6		ini:base	**5.75**	29.12	**1.65**	1.56

Table: Test error (%). Median of 3 runs. Resnet-28 (0.4M parameters) for CIFAR10/100, and WRN-16-4 (2.7M parameters) for SVHN. Two numbers for SVHN are without and with dropout.

Similar results with larger models and on imagenet.



(a) GULF2 (b) Regular training

Figure: Test loss in relation to training loss. The arrows indicate the direction of time flow. CIFAR100. ResNet-28.

GULF solution properties:

- worse training loss but better test loss (better generalization)
- different weight-decay behavior in regularizer

Summary

Background:

- Nonconvex optimization stuck in local minimum
- Want to find a local minimum with better generalization

Method:

- Guided learning through successive functional gradient optimization
- Find local solution with worse training loss but better generalization

Why:

- Restricted search space $\rightarrow$ better generalization

Our method generalizes self-distillation.