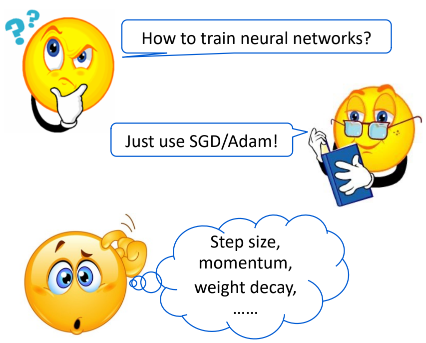
Guarantees for Tuning the Step Size using a Learning-to-Learn Approach

Xiang Wang, Shuai Yuan, Chenwei Wu, Rong Ge

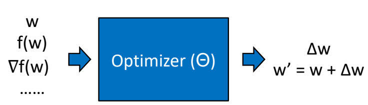
Duke University

How to train a neural net?

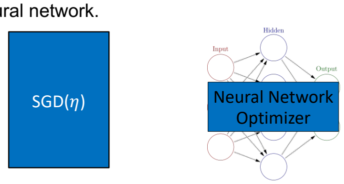


- For SGD/Adam, tuning the hyper-parameters can be very time-consuming.

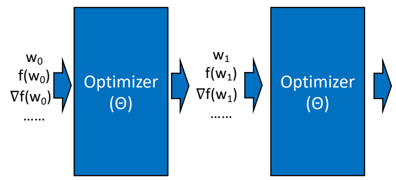
Learning to learn



- Idea: use a meta-learning approach to tune hyper-parameters or learn a new optimizer! [Andrychowicz et al. 2016] [Wichrowska et al. 2017] [Metz et al. 2019]
- Goal: find a good optimizer for a distribution of tasks.
- Idea: Abstract the optimization algorithm as a mapping from the current state to the next state with parameter Θ. Optimize the parameter Θ for the distribution of task.
- Optimizer can be as simple as SGD with tunable step size, can also be as complicated as a deep neural network.



How to train an optimizer?



- Unroll the optimizer for T steps.
- Define a meta-objective over the trajectory.
- Do (meta-)gradient descent on optimizer parameter Θ.
- No theoretical guarantees on training process or the learned optimizer

This work: Analyze step size tuning in GD/SGD for simple quadratic objectives.

Meta-gradient Explosion/Vanishing

- Objective: min $f(w) = \frac{1}{2}w^\top H w$
- Algorithm: gradient descent with constant step
  $w_{t+1} = w_t - \eta\nabla f(w_t) = (I - \eta H)w_t$
- Naïve meta-objective: loss at last step
  $\mathrm{F}(\eta) = f(w_{\eta,T})$

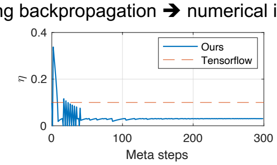
Point w at T-th iteration with step size η

Theorem: For almost all values of η, the meta-gradient $\mathrm{F}'(\eta)$ is either exponentially large or exponentially small in T.

- Idea: meta-gradient is exponentially large (small) because the meta-objective is exponentially large (small) in T.
- New objective: $\mathrm{G}(\eta) = \frac{1}{T}\log f(w_{\eta,T}) = \frac{1}{T}\log \mathrm{F}(\eta)$

Theorem: For the new objective, the meta-gradient $\mathrm{G}'(\eta)$ is always polynomial in all relevant parameters.

- $\mathrm{G}'(\eta) = \frac{dG}{dF}\cdot \mathrm{F}'(\eta)$, both terms are exponentially large or small, but they cancel each other.
- This exactly how one would compute $\mathrm{G}'(\eta)$ using backpropagation ➔ numerical issues!



Generalization of Trained Optimizer

- Setting: least squares problem
  $y = w_*^\top x + \xi, \|w_*\| = 1, x \sim N(0, I_d), \xi \sim N(0, \sigma^2)$
- Objective: squared loss on training data

$$f(w) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - w^\top x_i)^2$$

- Algorithm: gradient descent with constant step size (similar for SGD)
  $w_{t+1} = w_t - \eta\nabla f(w_t)$

Two ways to define meta-objective

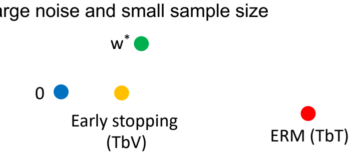
1. Train-by-train:
Define meta-objective on training set, e.g., simply choose $F(\eta) = f(w_{\eta,T})$
2. Train-by-validation [Metz et al. 2019]
Use a separate validation set $(x_1', y_1'), \ldots (x_{n_2}', y_{n_2}')$, define

$$G(\eta) = \frac{1}{2n_2}\sum_{i=1}^{n_2}\left(y'_i - w_{\eta,T}x'_i\right)^2$$

When do we need train-by-validation?

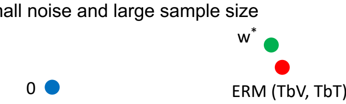
Theorem:
1. when noise σ is large, and n (#samples) is a constant fraction of d (#dimension), then train-by-validation is better.
2. When n (#samples) is much larger than d (#dimension), then train-by-train is close to optimal.

1. Large noise and small sample size



- Train-by-train chooses a large constant step size so that the optimizer quickly converges to the ERM solution. When the noise is large, the ERM solution overfits to the noise and is far from w*
- Train-by-validation chooses a smaller step size to leverage the signal in the training samples without overfitting to the noise.

2. Small noise and large sample size



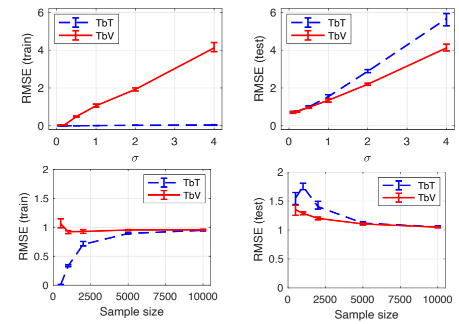
- The ERM solution is close to w*

[Andrychowicz et al. 2016] Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. In Advances in neural information processing systems, pp. 3981–3989, 2016.

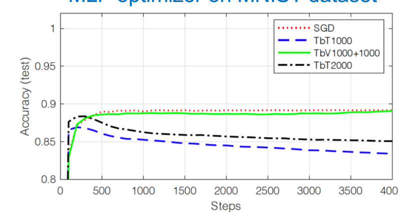
[Wichrowska et al. 2017] Wichrowska, O., Maheswaranathan, N., Hoffman, M. W., Colmenarejo, S. G., Denil, M., de Freitas, N., and Sohl-Dickstein, J. Learned optimizers that scale and generalize. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 3751–3760. JMLR. org, 2017.

Empirical Verification

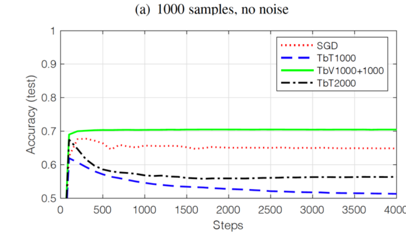
Step size tuning on least square problems



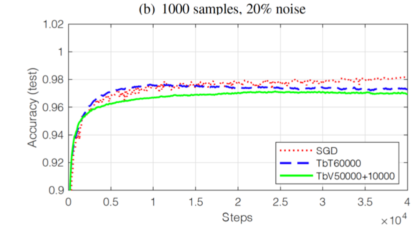
MLP optimizer on MNIST dataset



(a) 1000 samples, no noise



(b) 1000 samples, 20% noise



(c) All samples, no noise

[Metz et al. 2019] Metz, L.,Maheswaranathan, N., Nixon, J., Freeman, D., and Sohl-Dickstein, J. Understanding and correcting pathologies in the training of learned optimizers. In International Conference on Machine Learning, pp. 4556–4565, 2019.