No Learning Rates Needed

Introducing SaLSa – Stable Armijo Line Search

Philip Kenneweg
13 May 2024
University Bielefeld

01 Theory

We presented the Idea

02 Experiments

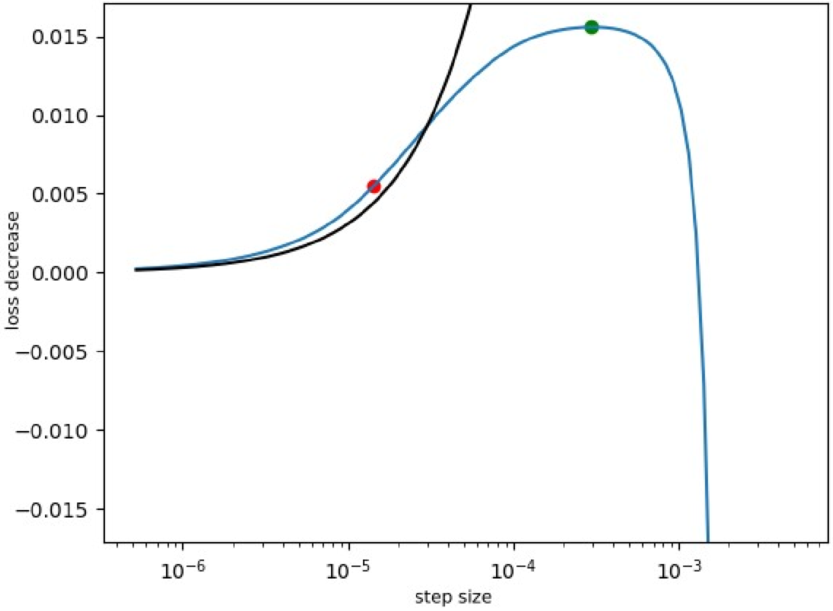
We show you some Empirical proof

03 Application

We make our Line Search easy to use

What is a Line Search?

- Learning rate is a hyperparameter
- We want to find its optimum
- Dynamic schedules possible



Key Idea

Algorithm 1 Basic Line Search

```
for every step k do
    for all η in range(η_min, η_max) do
        f_{k,η} = f_k(w_k − ∇f_k(w) · η)
    end for
    η_k ← arg min_η f_{k,η}
    w_k ← w_k − ∇f_k(w) · η_k
end for
```

Advantages

No learning rate tuning needed

Faster convergence

Better generalization

Disadvantages

High computational cost

Designed for classical optimization

Can not incorporate other optimizers (ADAM)

Existing Solutions

Algorithm 2 Armijo Line Search

$\eta_k = \eta_{k-1} \cdot 2^{1/b}$
while not $f_k(w_k + \eta_k d_k) \leq f_k(w_k) - c \cdot \eta_k ||\nabla f_k(w_k)||^2$ **do**
    $\eta_k = \eta_k \cdot \delta$
end while
$w_k = w_k + d_k \cdot \eta_k$

Advantages

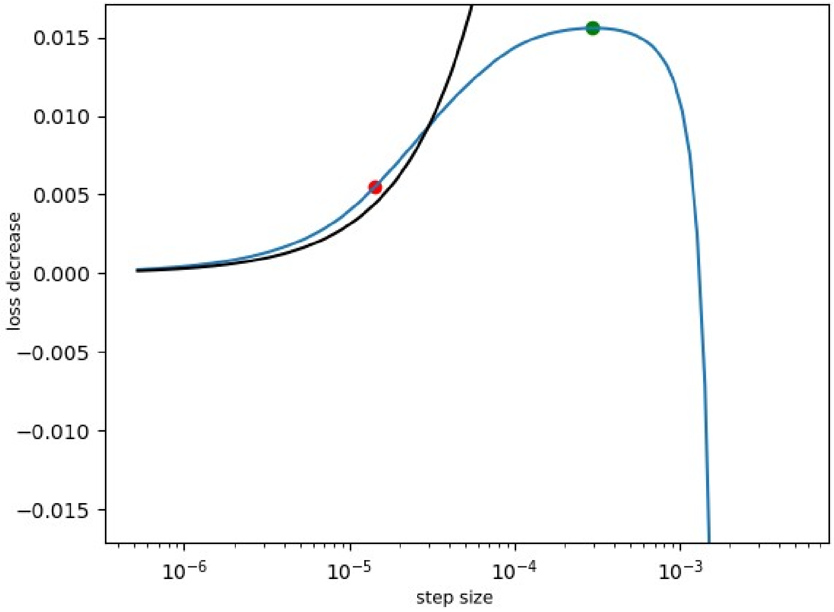
Lower computational cost

Can work with other optimizers

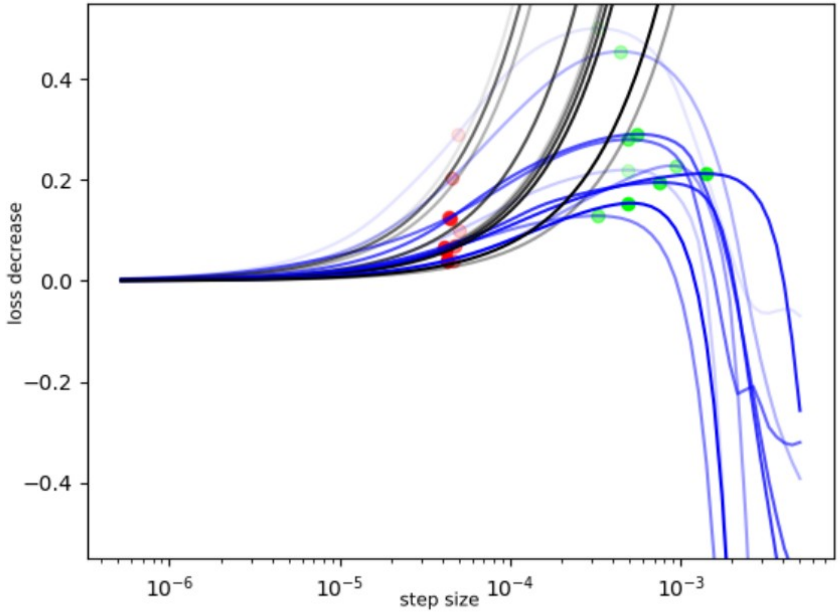
Disadvantages

Has problems for more complex NN

Not computationally stable



0.015
0.010
0.005
0.000
−0.005
−0.010
−0.015
loss decrease
10^{-6} 10^{-5} 10^{-4} 10^{-3}
step size



0.4
0.2
0.0
−0.2
−0.4
loss decrease
10^{-6} 10^{-5} 10^{-4} 10^{-3}
step size

Our Solution

$$f_k(w_k) - f_k(w_k + \eta_k d_k) \geq c \cdot \eta_k ||\nabla f_k(w_k)||^2 \tag{3.6}$$

$f_k(w_k) - f_k(w_k + \eta_k d_k)$ denotes the decrease in loss and $||\nabla f_k(w_k)||^2$ denotes the gradient norm. In order to apply exponential smoothing to both terms we define h_k and s_k as follows:

$$\begin{aligned} h_k = h_{k-1} \cdot \beta_3 + (f_k(w_k) - f_k(w_k + \eta_k d_k)) \cdot (1 - \beta_3) \\ s_k = s_{k-1} \cdot \beta_3 + ||\nabla f_k(w_k)||^2 \cdot (1 - \beta_3) \end{aligned} \tag{3.7}$$

$$h_k \geq c \cdot \eta_k \cdot s_k \tag{3.8}$$

Combining SaLSa and the Adam optimizer is done by computing s_k as follows:

$$s_k = s_{k-1} \cdot \beta_3 + \frac{||\nabla f_k(w_k)||^2}{\sqrt{\hat{v}_k} + \epsilon} \cdot (1 - \beta_3) \tag{3.9}$$

Our Solution

$$\bar{\eta}_k(\beta) = \beta\bar{\eta}_{k-1} + (1-\beta)\cdot\eta_{k-1}$$

We calculate the average rate of change as follows:

$$r_k = \frac{\bar{\eta}_k(0.9)}{\bar{\eta}_k(0.99)}$$

and invert it if $r_k \leq 1$:

$$\bar{r_k} = \begin{cases} r_k & \text{if } r_k \geq 1 \\ r_k^{-1} & \text{otherwise} \end{cases}$$

we set the line search frequency L_k to the closest integer of:

$$L_k = \frac{1}{\bar{r_k} - 1} \tag{3.13}$$

and clamp it to the range $L_{k+1} \in [1, 10]$. We perform the line search every L_{k+1} steps. This reduces the extra compute needed from roughly 30% to approximately 3% for longer runs. In practice, we did not notice any performance degradation.

Advantages

Even lower computational cost

Can work with other optimizers

computationally stable

Can now train Transformer and other modern architectures

Disadvantages

No general proofs of convergence possible due to momentum term.

01 Theory

We presented the Idea

02 Experiments

We show you some Empirical proof

03 Application

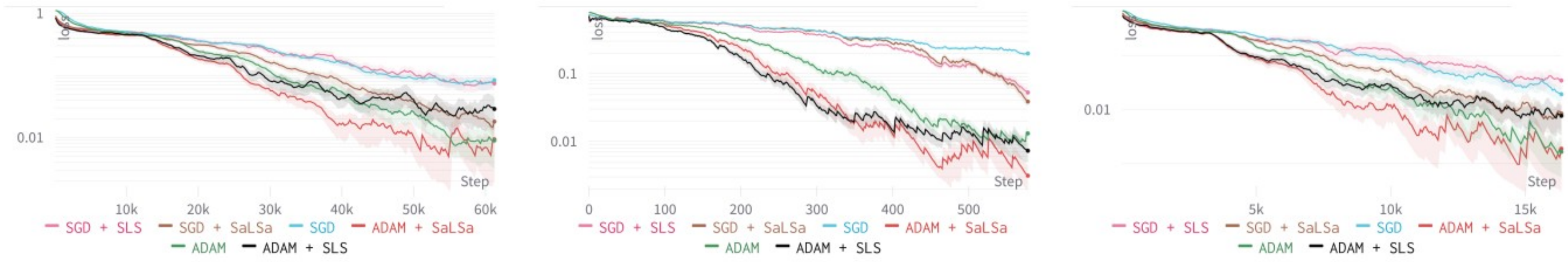
We make our Line Search easy to use

Experiments – more than 50% reduction on final loss

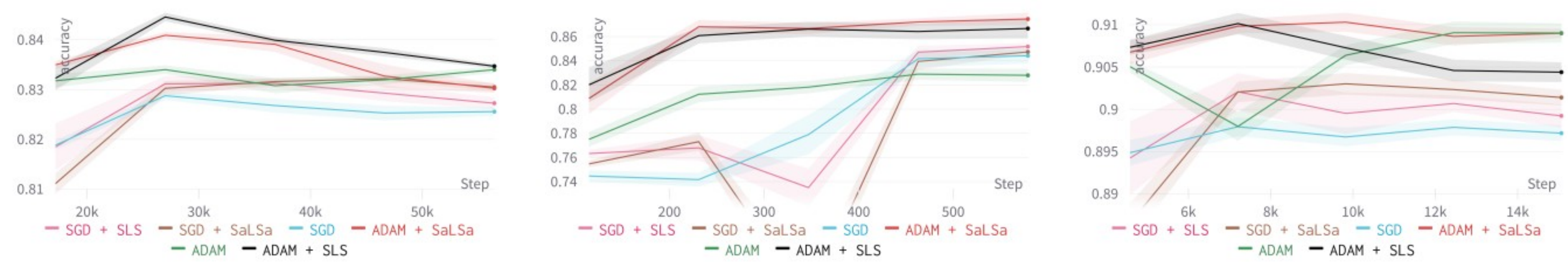
	ADAM -	SGD -	ADAM SLS	SGD SLS	ADAM SaLSa	SGD SaLSa
MNLI	0.009567	0.08613	0.03713	0.06901	**0.005867**	0.02174
QNLI	0.00258	0.02079	0.00504	0.03667	**0.000628**	0.0091627
MRPC	0.01312	0.1978	0.007298	0.05262	**0.003126**	0.03862
SST2	**0.005857**	0.02561	0.009457	0.0412	0.006991	0.01837
GPT-2	2.86	3.572	2.917	3.566	**2.772**	3.559
ResNet34						
CIFAR10	0.01394	0.00982	**0.0009508**	0.05646	0.003314	0.003773
CIFAR100	0.03739	0.01143	0.01337	0.08245	**0.003774**	0.01453
ResNet50						
ImageNet	0.9122	1.547	2.036	1.144	**0.8339**	0.9788
log average	0.0355	0.0930	0.0315	0.134	**0.0148**	0.0477
average rank	2.75	4.625	3.125	5.5	**1.25**	3.75

Experiments

Natural Language Processing - Transformer Experiments

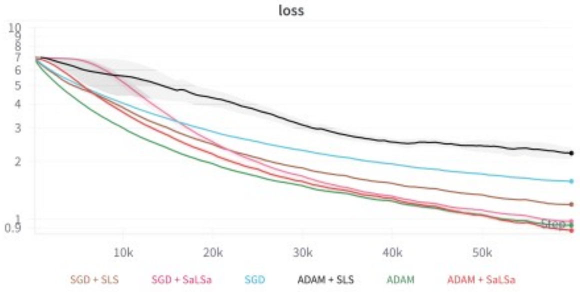


(A) MNLI　(B) MRPC　(C) QNLI

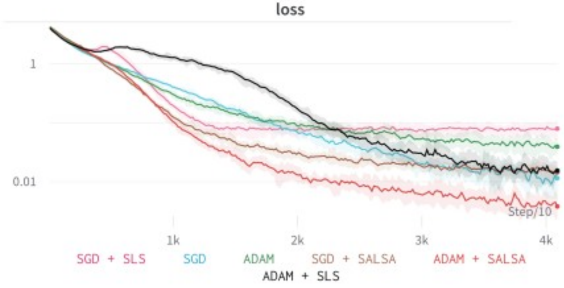


(D) MNLI　(E) MRPC　(F) QNLI

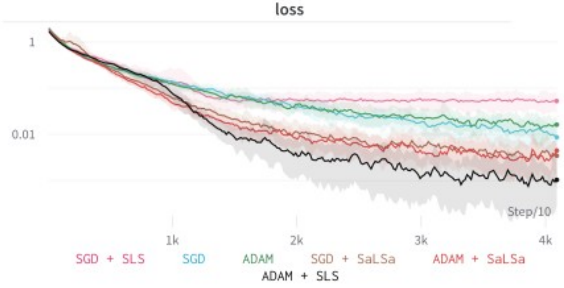
Experiments



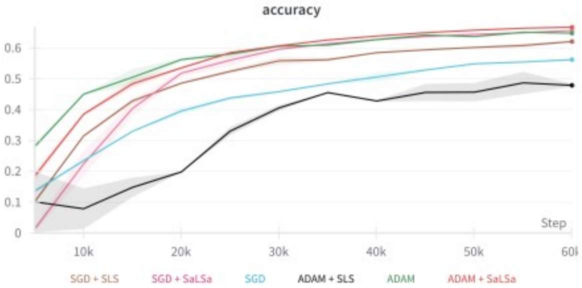
(A) Ima-geNet



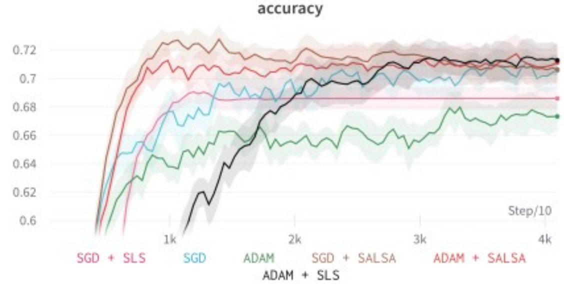
(B) Cifar 100



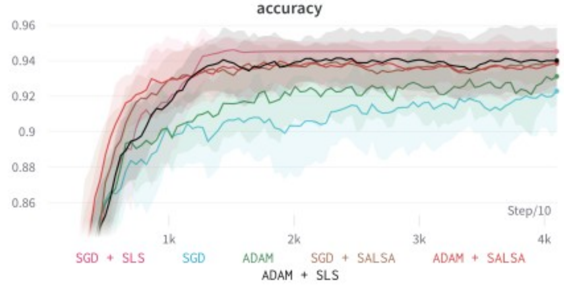
(C) Cifar 10



(D) Ima-geNet



(E) Cifar 100



(F) Cifar 10

Summary

We examined the state of the art

We iterated on the ideas and found improvements

We provide an easy to use framework

Questions & Answers

Thanks for listening!