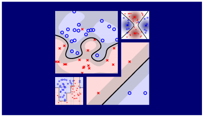
Machine Learning Soundings
(機器學習深測)



Lecture 3: Optimization in Deep Learning

Hsuan-Tien Lin (林軒田)

htlin@csie.ntu.edu.tw

Department of Computer Science
& Information Engineering

National Taiwan University
(國立台灣大學資訊工程系)

Roadmap

1. Deep Learning Foundations

Lecture 1: Neural Network

automatic **pattern feature extraction** from **layers of neurons** with **backprop** for GD/SGD

Lecture 3: Optimization in Deep Learning

- Difficulty of Deep Learning Optimization

2. Deep Learning Models

Difficulty of Deep Learning Optimization

error surface complicated

- local minima: not as bad as imagined
- saddle points/local maxima: easily escapable (especially with SGD)
- plateau: need larger learning rate η
- ravines: need to avoid oscillation

stability <> computation trade-off

slow computation of gradient (backprop)
$\Rightarrow$ SGD on minibatch
$\Rightarrow$ 'instable' estimate of gradient

getting more stable estimate? **averaging**

Running Average Estimate of Gradient

gradient descent: $\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \eta \cdot \mathbf{v}_t$

original minibatch SG

gradient estimate $\mathbf{v}_t = \Delta_t$ from one minibatch SG

averaging by multiple SG

if minibatch SG for M times at t-th iteration, each getting $\Delta_t^{(m)}$, more stable gradient estimate by uniform averaging $\mathbf{v}_t = \frac{1}{M}\sum_{m=1}^{M} \Delta_t^{(m)}$ —needing M times more computation than original minibatch SGD

speedup by reusing each $\Delta_t = \Delta_t^{(1)}$

$\mathbf{v}_t = \frac{1}{M}\sum_{m=1}^{M} \Delta_{t-m+1}$ —'moving window' average of SG

issue with 'moving window' average: **uniformly weighted**

Averaging SG Non-uniformly

Running Average

$$\mathbf{v}_t = \beta \mathbf{v}_{t-1} + (1-\beta)\Delta_t$$

with $0 \leq \beta < 1$ to control how much history to take $\beta = 0$: original SGD

$$\mathbf{v}_t = \sum_{m=1}^{t} \beta^{t-m}(1-\beta)\Delta_t$$

—size-t window, exponentially-decreasing aeveraging

SGD **with momentum**: optimization direction
= current SG (Δ_t) + historical inertia ($\mathbf{v}_{t-1}$)

Benefits of SGD with Momentum

$$
\begin{aligned}
\mathbf{v}_t &= \beta \mathbf{v}_{t-1} + (1-\beta)\Delta_t \\
\mathbf{w}_t &= \mathbf{w}_{t-1} - \eta \mathbf{v}_t
\end{aligned}
$$

- some variance in SG canceled out
- oscilliation across ravine dampened
- shallow local optima/saddle points escaped

SGD with momentum: ‘stablize’ SG with running average

Per-Component Learning Rate

fixed learning rate : $\mathbf{w}_t = \mathbf{w}_{t-1} - \eta \mathbf{v}_t$

per-component learning rate : $\mathbf{w}_t = \mathbf{w}_{t-1} - \boldsymbol{\eta}_t \odot \mathbf{v}_t$

intuition: scales error surface

want: smaller step for larger gradient component

Running Average of Gradient Magnitude

want: smaller step for larger gradient component, say

$$\eta_t = \frac{1}{\sqrt{\nabla E(\mathbf{w}_t) \odot \nabla E(\mathbf{w}_t)}}$$

- full gradient ∇E not available, SG only
- using $\|\Delta\|$ not very stable

idea: running average of $\Delta_t \odot \Delta_t$

RMSProp

$$
\begin{aligned}
\mathbf{u}_t &= \beta \mathbf{u}_{t-1} + (1-\beta)\Delta_t \odot \Delta_t \\
\boldsymbol{\eta}_t &= \eta \cdot (\mathbf{u}_t + \epsilon)^{-1/2} \\
\mathbf{w}_t &= \mathbf{w}_{t-1} - \boldsymbol{\eta}_t \odot \Delta_t
\end{aligned}
$$

RMSProp: SGD + per-component learning rate using running average of magnitude

Adam: Adaptive Moment Estimation

Adam ≈ momentum + RMSProp + global decay

$$\begin{aligned}
\mathbf{v}_t &= \beta_1 \mathbf{v}_{t-1} + (1-\beta_1)\Delta_t \\
\mathbf{u}_t &= \beta_2 \mathbf{u}_{t-1} + (1-\beta_2)\Delta_t \odot \Delta_t \\
\boldsymbol{\eta}_t &= \eta \cdot \sqrt{N/t} \cdot (\mathbf{u}_t \oplus \epsilon)^{\text{-1/2}} \\
\mathbf{w}_t &= \mathbf{w}_{t-1} - \boldsymbol{\eta}_t \odot \mathbf{v}_t
\end{aligned}$$

- momentum in $\mathbf{v}_t$
- RMSProp in $\mathbf{u}_t$
- global decay by $\sqrt{t/N}$
- (some minor correction of estimation)

Adam usually more aggressive than original SGD (but can also overfit faster)