Advanced Optimization

Optimization

All variables are using the same learning rate in each iteration, e.g.,

$$w_1 \leftarrow w_1 - \eta \frac{\partial f}{\partial w_1}, \quad w_2 \leftarrow w_2 - \eta \frac{\partial f}{\partial w_2}$$

While their gradients maybe different.

Advanced Optimization

Adagrad

RMSProp

Adadelta

Adam

Optimization - Adagrad

Adagrad adjusts the learning rate according to the gradient value of the independent variable in each dimension, to eliminate problems caused when a unified learning rate has to adapt to all dimensions

It uses the **cumulative variable** s_t , obtained from a square by element operation on the mini-batch stochastic gradient g_t .

Optimization - Adagrad

At time step 0, we initialize $\boldsymbol{s}_0$ to 0.

At time step t, we first sum up the results of the square by element operation for the mini-batch gradient $\boldsymbol{g}_t$ to get the variable $\boldsymbol{s}_t$.

$$\boldsymbol{s}_t \leftarrow \boldsymbol{s}_{t-1} + \boldsymbol{g}_t \odot \boldsymbol{g}_t,$$

Here, $\odot$ is the symbol for multiplication by element.

Optimization - Adagrad

Next, we re-adjust the learning rate of each element in the independent variable of the objective function using element operations:

$$\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t-1} - \frac{\eta}{\sqrt{\boldsymbol{s}_t + \epsilon}} \odot \boldsymbol{g}_t,$$

Here, η is the learning rate while ϵ is a constant added to maintain numerical stability, such as 10^{-6}.

Advanced Optimization - Summary

Adagrad

$$s_t \leftarrow s_{t-1} + g_t \odot g_t$$

$$x_t \leftarrow x_{t-1} - \frac{\eta}{\sqrt{s_t + \epsilon}} \odot g_t$$

RMSProp

Adadelta

Adam

Optimization - RMSProp

RMSProp, uses the **exponentially weighted moving average** (EWMA) on the square by element of all the mini-batch stochastic gradients g_t up to the time step t.

Specifically, given the hyperparameter $0 \leq \gamma < 1$, RMSProp is computed at time step $t > 0$.

$$\mathbf{s}_t \leftarrow \gamma \mathbf{s}_{t-1} + (1 - \gamma) \mathbf{g}_t \odot \mathbf{g}_t$$

Optimization - RMSProp

RMSProp, uses the **exponentially weighted moving average** (EWMA) on the square by element of all the mini-batch stochastic gradients g_t up to the time step t.

$$
\begin{aligned}
\mathbf{s}_t &= (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t + \gamma \mathbf{s}_{t-1} \\
&= (1-\gamma)\left(\mathbf{g}_t \odot \mathbf{g}_t + \gamma \mathbf{g}_{t-1} \odot \mathbf{g}_{t-1}\right) + \gamma^2 \mathbf{s}_{t-2} \\
&\cdots \\
&= (1-\gamma)\left(\mathbf{g}_t \odot \mathbf{g}_t + \gamma \mathbf{g}_{t-1} \odot \mathbf{g}_{t-1} + \cdots + \gamma^{t-1} \mathbf{g}_1 \odot \mathbf{g}_1\right)
\end{aligned}
$$

$$\frac{1}{1-\gamma} = 1 + \gamma + \gamma^2 + \cdots$$

Optimization - RMSProp

Like Adagrad, RMSProp re-adjusts the learning rate of each element in the independent variable of the objective function with element operations and then updates the independent variable.

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \frac{\eta}{\sqrt{\mathbf{s}_t + \epsilon}} \odot \mathbf{g}_t$$

Here, η is the learning rate while ϵ is a constant added to maintain numerical stability, such as 10^{-6}.

Advanced Optimization - Summary

Adagrad

$$\boldsymbol{s}_t \leftarrow \boldsymbol{s}_{t-1} + \boldsymbol{g}_t \odot \boldsymbol{g}_t \qquad \boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t-1} - \frac{\eta}{\sqrt{\boldsymbol{s}_t + \epsilon}} \odot \boldsymbol{g}_t$$

RMSProp

$$\mathbf{s}_t \leftarrow \gamma \mathbf{s}_{t-1} + (1-\gamma)\mathbf{g}_t \odot \mathbf{g}_t \qquad \boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t-1} - \frac{\eta}{\sqrt{\boldsymbol{s}_t + \epsilon}} \odot \boldsymbol{g}_t$$

Adadelta

Adam

Optimization - Adadelta

Adadelta differs from RMSProp in its replacement of the hyperparameter η.

Firstly, like RMSProp, the Adadelta algorithm uses the variable s_t, which is an EWMA on the squares of elements in mini-batch stochastic gradient g_t.

At time step 0, all the elements are initialized to 0.

At time step $t>0$, given the hyperparameter $0 \leq \rho < 1$ (counterpart of γ in RMSProp), compute using the same method as RMSProp:

$$\boldsymbol{s}_t \leftarrow \rho \boldsymbol{s}_{t-1} + (1-\rho)\boldsymbol{g}_t \odot \boldsymbol{g}_t .$$

Optimization - Adadelta

Adadelta maintains an additional state variable, Δx_t the elements of which are also initialized to 0 at time step 0. We use Δx_{t-1} to compute the variation of the independent variable:

$$\boldsymbol{g}_t' \leftarrow \sqrt{\frac{\Delta \boldsymbol{x}_{t-1} + \epsilon}{\boldsymbol{s}_t + \epsilon}} \odot \boldsymbol{g}_t,$$

Here, ϵ is a constant added to maintain the numerical stability, such as 10^{-6}.

Optimization - Adadelta

Next, we update the independent variable:

$$\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t-1} - \boldsymbol{g}'_t$$

Finally, we use Δx to record the EWMA on the squares of elements in g', which is the variation of the independent variable.

$$\Delta \boldsymbol{x}_t \leftarrow \rho \Delta \boldsymbol{x}_{t-1} + (1 - \rho) \boldsymbol{g}'_t \odot \boldsymbol{g}'_t$$

As we can see, if the impact of ϵ is not considered here, Adadelta differs from RMSProp in its replacement of the hyperparameter η with $\sqrt{\Delta \boldsymbol{x}_{t-1}}$.

Advanced Optimization - Summary

Adagrad

$$\mathbf{s}_t \leftarrow \mathbf{s}_{t-1} + \mathbf{g}_t \odot \mathbf{g}_t \qquad \mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \frac{\eta}{\sqrt{\mathbf{s}_t + \epsilon}} \odot \mathbf{g}_t$$

RMSProp

$$\mathbf{s}_t \leftarrow \gamma \mathbf{s}_{t-1} + (1 - \gamma)\mathbf{g}_t \odot \mathbf{g}_t \qquad \mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \frac{\eta}{\sqrt{\mathbf{s}_t + \epsilon}} \odot \mathbf{g}_t$$

Adadelta

$$\mathbf{s}_t \leftarrow \rho \mathbf{s}_{t-1} + (1 - \rho)\mathbf{g}_t \odot \mathbf{g}_t . \qquad \mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \mathbf{g}'_t$$

$$\mathbf{g}'_t \leftarrow \sqrt{\frac{\Delta \mathbf{x}_{t-1} + \epsilon}{\mathbf{s}_t + \epsilon}} \odot \mathbf{g}_t, \qquad \Delta \mathbf{x}_t \leftarrow \rho \Delta \mathbf{x}_{t-1} + (1 - \rho)\mathbf{g}'_t \odot \mathbf{g}'_t$$

Adam

Optimization - Adam

Adam combines RMSProp with Momentum. So, in addition to using the decaying average of past squared gradients for parameter-specific learning rate, it uses a decaying average of past gradients in place of the current gradient.

The momentum variable v_t is the EWMA of the mini-batch stochastic gradient

$$\boldsymbol{v}_t \leftarrow \beta_1 \boldsymbol{v}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$$

Just as in RMSProp,

$$\boldsymbol{s}_t \leftarrow \beta_2 \boldsymbol{s}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t \odot \boldsymbol{g}_t$$

Optimization - Adam

Notice that when t is small, the sum of the mini-batch stochastic gradient weights from each previous time step will be small. For example, when β^1 =0.9, $\boldsymbol{v}_1 = 0.1\boldsymbol{g}_1$. To eliminate this effect, for any time step t, we can divide v_t by $1 - \beta_1^t$, so that the sum of the mini-batch stochastic gradient weights from each previous time step is 1.

So, we perform **bias corrections** for variables v_t and s_t:

$$\hat{\boldsymbol{v}}_t \leftarrow \frac{\boldsymbol{v}_t}{1 - \beta_1^t} \qquad \hat{\boldsymbol{s}}_t \leftarrow \frac{\boldsymbol{s}_t}{1 - \beta_2^t}$$

Optimization - Adam

Adam

$$\boldsymbol{v}_t \leftarrow \beta_1 \boldsymbol{v}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$$

$$\boldsymbol{s}_t \leftarrow \beta_2 \boldsymbol{s}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t \odot \boldsymbol{g}_t$$

$$\hat{\boldsymbol{v}}_t \leftarrow \frac{\boldsymbol{v}_t}{1 - \beta_1^t}$$

$$\hat{\boldsymbol{s}}_t \leftarrow \frac{\boldsymbol{s}_t}{1 - \beta_2^t}$$

$$\boldsymbol{g}_t' \leftarrow \frac{\eta \hat{\boldsymbol{v}}_t}{\sqrt{\hat{\boldsymbol{s}}_t} + \epsilon},$$

$$\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t-1} - \boldsymbol{g}_t'$$