Batch optimization in VW via LBFGS

Miroslav Dudík

12/16/2011

Outline

- gradient descent and Newton method
- LBFGS
- LBFGS in VW

Smooth convex unconstrained optimization

Goal: $\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w})$

where f is strongly convex
and twice continuously differentiable

Smooth convex unconstrained optimization

Goal: $\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w})$

where f is strongly convex
and twice continuously differentiable

Our objective:

$$f(\mathbf{w}) = \sum_{i=1}^{n} loss(\mathbf{w}; x_i, y_i) + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

- possibly weighted loss
- regularization can have coordinate-specific scaling (specified by user)

Warm-up: Gradient descent

- initialize $\mathbf{w_0}$
- for t=1,2,...:
  move in the direction of the steepest descent
  $$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f(\mathbf{w}_t)$$

Warm-up: Gradient descent

Gradient descent update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f(\mathbf{w}_t)$$

Warm-up: Gradient descent

gradient

$$\mathbf{g}_t = \nabla f(\mathbf{w}_t)$$

Gradient descent update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f(\mathbf{w}_t)$$

Equivalently:

- approximate

$$f(\mathbf{w}) \approx f(\mathbf{w}_t) + \mathbf{g}_t^\top (\mathbf{w}_t - \mathbf{w}) + \frac{1}{2\eta} \|\mathbf{w}_t - \mathbf{w}\|^2$$

Warm-up: Gradient descent

gradient

$$\mathbf{g}_t = \nabla f(\mathbf{w}_t)$$

Gradient descent update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f(\mathbf{w}_t)$$

Equivalently:

- approximate

$$f(\mathbf{w}) \approx f(\mathbf{w}_t) + \mathbf{g}_t^\top(\mathbf{w}_t - \mathbf{w}) + \frac{1}{2\eta}\|\mathbf{w}_t - \mathbf{w}\|^2$$

- optimize approximation:

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}} \left(f(\mathbf{w}_t) + \mathbf{g}_t^\top(\mathbf{w}_t - \mathbf{w}) + \frac{1}{2\eta}\|\mathbf{w}_t - \mathbf{w}\|^2 \right)$$

Warm-up: Gradient descent

gradient

$$\mathbf{g}_t = \nabla f(\mathbf{w}_t)$$

Gradient descent update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f(\mathbf{w}_t)$$

Equivalently:

- approximate

$$f(\mathbf{w}) \approx f(\mathbf{w}_t) + \mathbf{g}_t^\top(\mathbf{w}_t - \mathbf{w}) + \frac{1}{2\eta}\|\mathbf{w}_t - \mathbf{w}\|^2$$

- optimize approximation:

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}} \left(f(\mathbf{w}_t) + \mathbf{g}_t^\top(\mathbf{w}_t - \mathbf{w}) + \frac{1}{2\eta}\|\mathbf{w}_t - \mathbf{w}\|^2 \right)$$

Can we replace quadratic term by a tighter approximation?

Newton method

Hessian

$$\mathbf{H}_t = \nabla^2 f(\mathbf{w}_t)$$

Better approximation

$$f(\mathbf{w}) \approx f(\mathbf{w}_t) + \mathbf{g}_t^\top (\mathbf{w}_t - \mathbf{w}) + \frac{1}{2}(\mathbf{w}_t - \mathbf{w})^\top \mathbf{H}_t (\mathbf{w}_t - \mathbf{w})$$

Update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{H}_t^{-1} \mathbf{g}_t$$

Newton method

Hessian

$$\mathbf{H}_t = \nabla^2 f(\mathbf{w}_t)$$

Better approximation

$$f(\mathbf{w}) \approx f(\mathbf{w}_t) + \mathbf{g}_t^\top(\mathbf{w}_t - \mathbf{w}) + \frac{1}{2}(\mathbf{w}_t - \mathbf{w})^\top \mathbf{H}_t(\mathbf{w}_t - \mathbf{w})$$

Update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{H}_t^{-1}\mathbf{g}_t$$

Problem: Hessian can be too big (matrix of size *dxd*)

LBFGS = a quasi-Newton method

[Nocedal 1980, Liu-Nocedal 1989]

Instead of the Newton update

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{H}_t^{-1}\mathbf{g}_t$$

Perform a ***quasi-Newton*** update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{K}_t \mathbf{g}_t$$

where: $\mathbf{K}_t$ is a low-rank approximation of $\mathbf{H}_t^{-1}$

η_t is obtained by line search

LBFGS = a quasi-Newton method

[Nocedal 1980, Liu-Nocedal 1989]

Instead of the Newton update

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{H}_t^{-1}\mathbf{g}_t$$

Perform a ***quasi-Newton*** update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{K}_t \mathbf{g}_t$$

where: $\mathbf{K}_t$ is a low-rank approximation of $\mathbf{H}_t^{-1}$

η_t is obtained by line search

- rank m specified by user (default m=15)
- instead of storage d^2, only storage $2dm$ required (update of $\mathbf{K}_t$ also has running time $O(dm)$ per iteration)

Line search in LBFGS

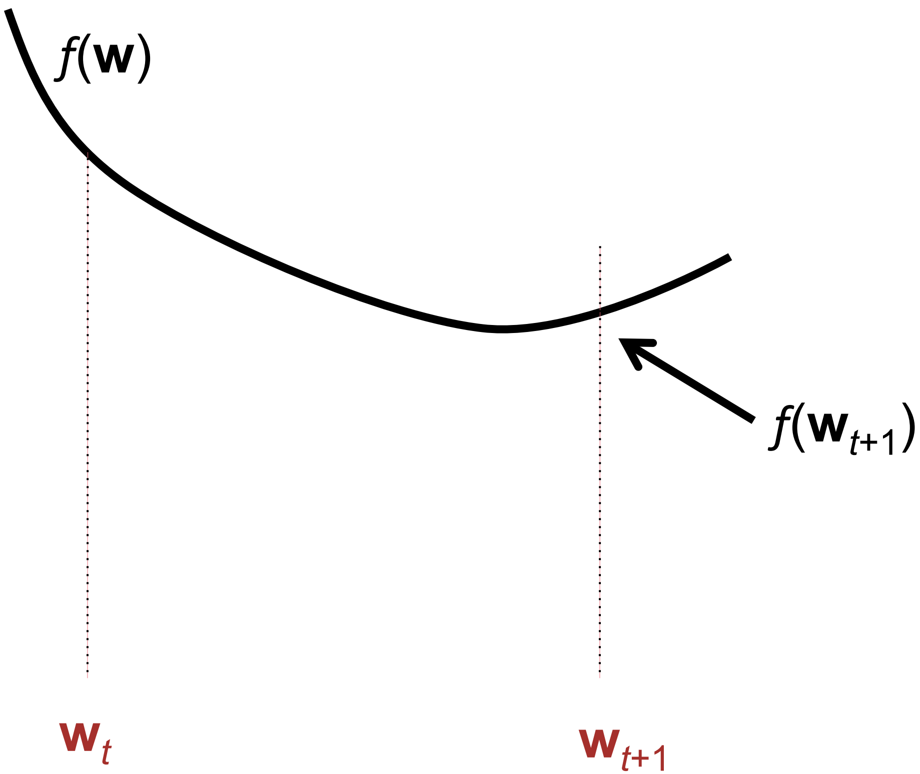
[Nocedal 1980, Liu-Nocedal 1989]

Update:

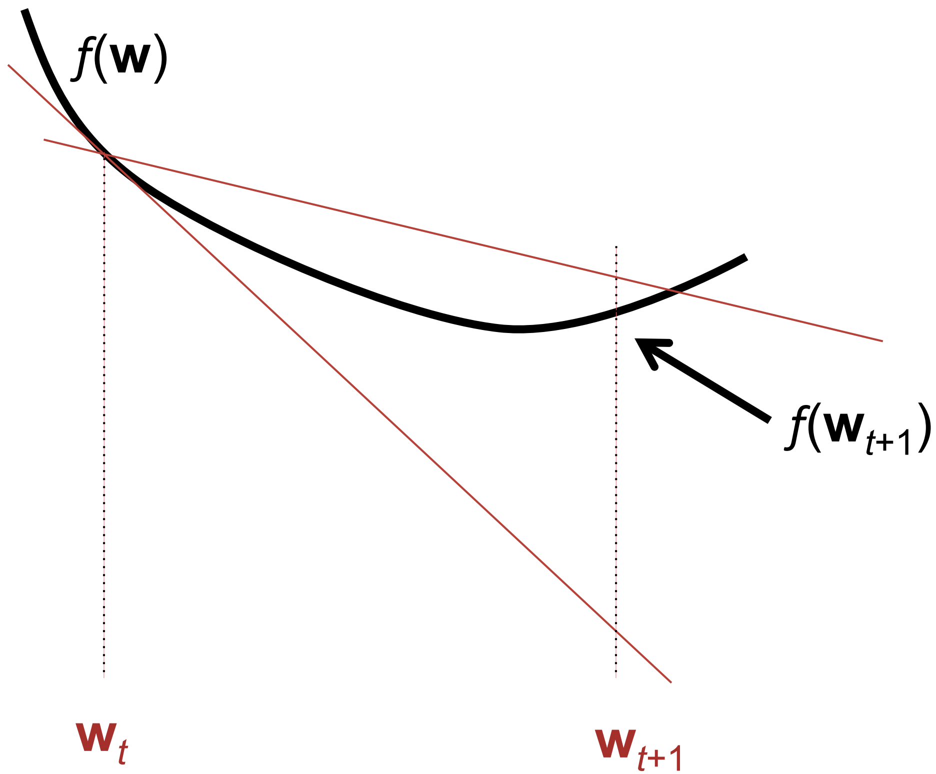
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{K}_t \mathbf{g}_t$$

- direction determined by $\mathbf{K}_t \mathbf{g}_t$
- step size η_t must satisfy ***Wolfe conditions***

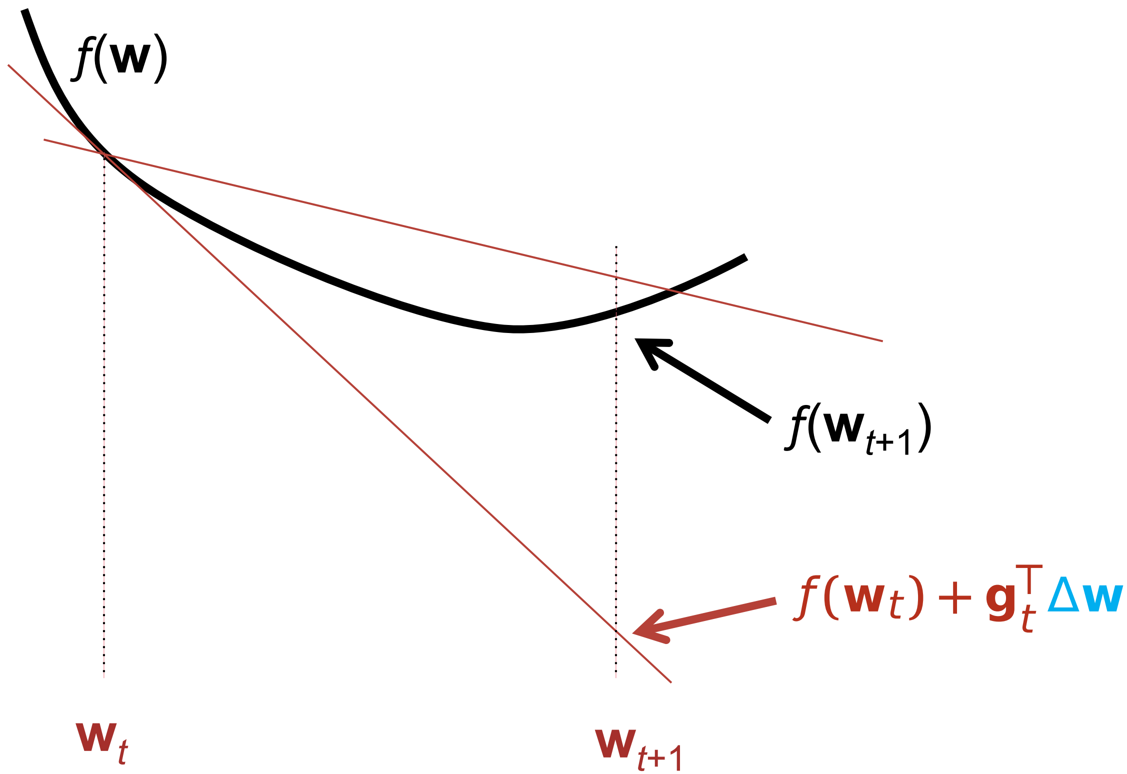
1st Wolfe condition:



1st Wolfe condition:



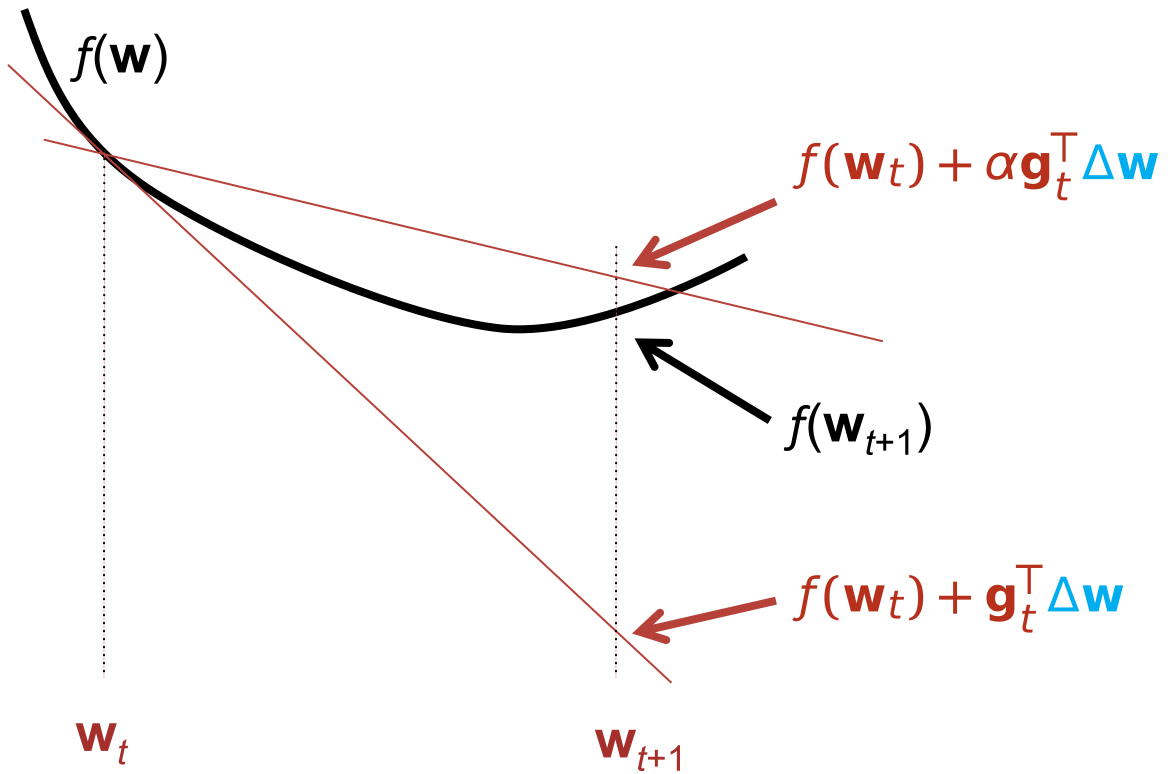
1st Wolfe condition:



change in **w**

$$\Delta \mathbf{w} = \mathbf{w}_{t+1} - \mathbf{w}_t$$

1st Wolfe condition:

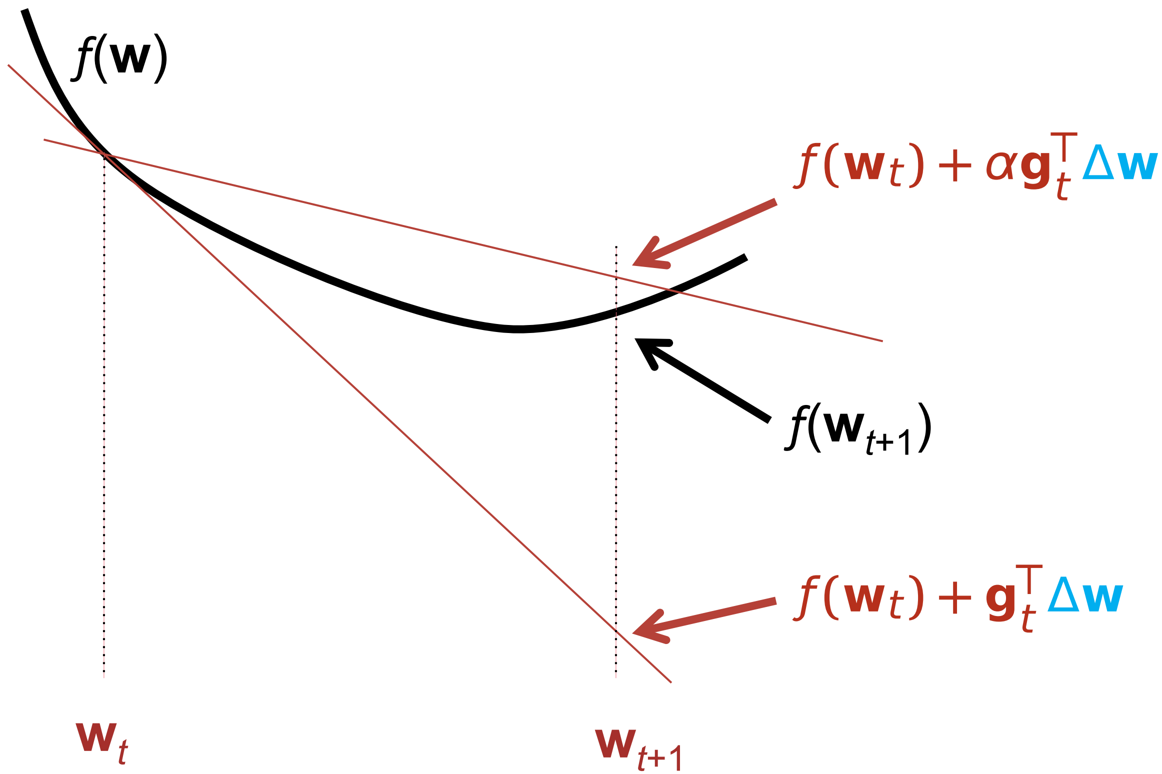


change in **w**

$$\Delta\mathbf{w} = \mathbf{w}_{t+1} - \mathbf{w}_t$$

1^{st} Wolfe condition:

$$f(\mathbf{w}_{t+1}) \le f(\mathbf{w}_t) + \alpha \mathbf{g}_t^\top \Delta\mathbf{w}$$ for some α in (0,0.5)



change in **w**

$$\Delta\mathbf{w} = \mathbf{w}_{t+1} - \mathbf{w}_t$$

1st Wolfe condition:

$f(\mathbf{w}_{t+1}) \leq f(\mathbf{w}_t) + \alpha \mathbf{g}_t^\top \Delta \mathbf{w}$ for some α in (0,0.5)

Rewrite as

$\Delta f \leq \alpha \mathbf{g}_t^\top \Delta \mathbf{w}$

where $\Delta f = f(\mathbf{w}_{t+1}) - f(\mathbf{w}_t)$

1st Wolfe condition:

$f(\mathbf{w}_{t+1}) \leq f(\mathbf{w}_t) + \alpha \mathbf{g}_t^\top \Delta\mathbf{w}$ for some α in (0,0.5)

Rewrite as

$\Delta f \leq \alpha \mathbf{g}_t^\top \Delta\mathbf{w}$

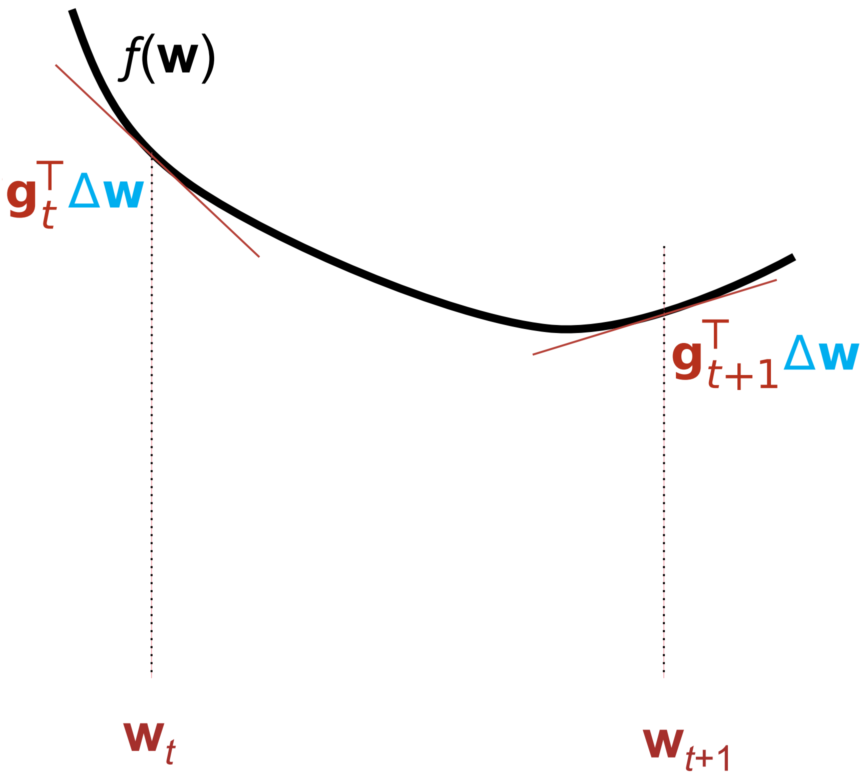
where $\Delta f = f(\mathbf{w}_{t+1}) - f(\mathbf{w}_t)$

Equivalent to: $\alpha \leq \frac{\Delta f}{\mathbf{g}_t^\top \Delta\mathbf{w}}$

(because $\mathbf{g}_t^\top \Delta\mathbf{w}$ is negative)

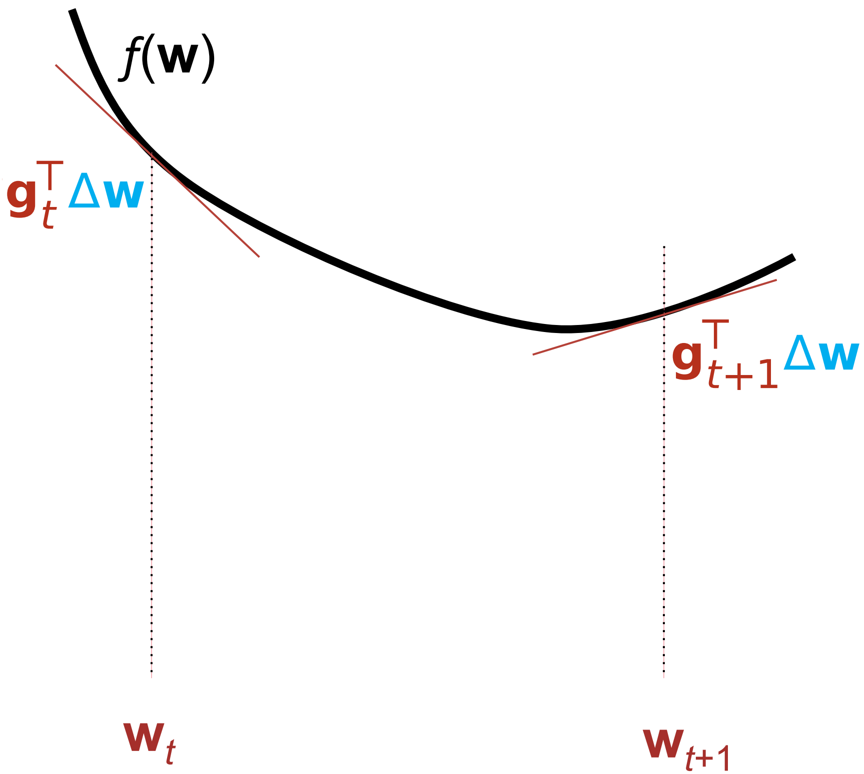
We use notation $\text{wolfe1} = \frac{\Delta f}{\mathbf{g}_t^\top \Delta\mathbf{w}}$ for the ratio on the rhs.

2nd Wolfe condition (strengthened):



2nd Wolfe condition (strengthened):

$$\left|\mathbf{g}_{t+1}^\top \Delta\mathbf{w}\right| \le \beta \mathbf{g}_t^\top \Delta\mathbf{w} \qquad \text{for some } \beta \text{ in } (\alpha,1)$$



2^{nd} Wolfe condition (strengthened):

$$\left|\mathbf{g}_{t+1}^{\top}\Delta\mathbf{w}\right| \le \beta\mathbf{g}_{t}^{\top}\Delta\mathbf{w}$$ for some β in $(\alpha,1)$

Rewrite as $\beta \ge \left|\frac{\mathbf{g}_{t+1}^{\top}\Delta\mathbf{w}}{\mathbf{g}_{t}^{\top}\Delta\mathbf{w}}\right|$.

We use notation $\text{wolfe2} = \frac{\mathbf{g}_{t+1}^{\top}\Delta\mathbf{w}}{\mathbf{g}_{t}^{\top}\Delta\mathbf{w}}$ for the ratio on the rhs.

Summarizing Wolfe conditions

Let $\text{wolfe1} = \frac{\Delta f}{\mathbf{g}_t^\top \Delta \mathbf{w}}$ and $\text{wolfe2} = \frac{\mathbf{g}_{t+1}^\top \Delta \mathbf{w}}{\mathbf{g}_t^\top \Delta \mathbf{w}}$.

Let $0<\alpha<0.5$, $\alpha<\beta<1$.

i) wolfe1 $\geq \alpha$

ii) |wolfe2| $\leq \beta$

In VW, the Wolfe conditions are not enforced

- ratios wolfe1 and wolfe2 are logged
- it is always possible to choose α and β in the hindsight as long as:

  wolfe1>0 and -1<wolfe2<1

Line search and termination in VW

- in the first iteration:
  - evaluate directional 2nd derivative and initialize step size according to the one-dimensional Newton step
  - if the loss does not decrease (i.e., wolfe1<0), shrink the step
- in the subsequent iterations:
  - set step size to 1.0
  - if the loss does not decrease (i.e., wolfe1<0), shrink the step
- terminate if

  either: the specified number of passes over the data is reached

  or: the relative decrease in the objective $f(\mathbf{w})$ falls below a threshold

LBFGS switches

--bfgs

turn on LBFGS optimization

--l2 0.0

L2 regularization coefficient

--mem 15

rank of the inverse Hessian approximation

--termination 0.001

termination threshold for the relative loss decrease