MATH 565: Lecture 10 (02/12/2026)

Today:
* AdaGrad
* RMSProp
* Adam
* Newton method

Recall Momentum-based learning: $\bar{v} \leftarrow \beta \bar{v} - \alpha \nabla J$, $\beta \in (0,1)$
$\bar{w} \leftarrow \bar{w} + \bar{v}$. $\beta = 0.9$ (typically)

AdaGrad (Adaptive subgradient method; Duchi, Hazan, Singer, 2011)

- Keep track of aggregated squared magnitude of $\frac{\partial J}{\partial w_i}$ $\forall i$.

$$A_i \leftarrow A_i + \left(\frac{\partial J}{\partial w_i}\right)^2 \quad \forall i$$

$$w_i \leftarrow w_i - \frac{\alpha}{\sqrt{A_i}}\left(\frac{\partial J}{\partial w_i}\right) \quad \forall i$$

can use $\sqrt{A_i + \epsilon}$ for $\epsilon > 0$, but small, to avoid ill-conditioning

* penalizes dimension (i) along which $\frac{\partial J}{\partial w_i}$ fluctuates wildly
* prefers movement along directions where the gradient is consistent for many steps.
  → same sign, $\approx$ same magnitude

But there are some potential drawbacks as well.

X absolute movement along each component slows down over time

X may become too slow quickly; stops making progress.

RMSProp (Root mean square propagation)

Hinton, 2012 (in a lecture!)

* use exponential averaging (or decay)
  - decay factor $\rho \in (0,1)$
  - weigh the squared aggregate from t steps ago by ρ^t. becomes much smaller for large t values

$$A_i \leftarrow \rho A_i + (1-\rho)\left(\frac{\partial J}{\partial w_i}\right)^2 \quad \forall i$$

influence of old gradients decrease exponentially with time.

$$w_i \leftarrow w_i - \frac{\alpha}{\sqrt{A_i}}\left(\frac{\partial J}{\partial w_i}\right) \quad \forall i.$$

✗ A_i values can be quite small at start. (we usually set $A_i = 0$ at start for initialization).

AdaM (Adam)

Adaptive momentum estimation (Kingma & Ba, 2014)

- combines ideas of RMSProp and momentum update

* $A_i \leftarrow \rho A_i + (1-\rho)\left(\frac{\partial J}{\partial w_i}\right)^2 \quad \forall i \qquad \rho \in (0,1)$

* also maintain exponentially smoothed gradient

$$F_i \leftarrow \rho_f F_i + (1-\rho_f)\left(\frac{\partial J}{\partial w_i}\right) \quad \forall i \qquad \rho_f \in (0,1)$$

like β (momentum parameter)

* $w_i \leftarrow w_i - \frac{\alpha_t}{\sqrt{A_i}} F_i$ where $\alpha_t = \alpha\left(\frac{\sqrt{1-\rho^t}}{1-\rho_f^t}\right)$

can help to overcome initialization issues.

Newton Method

* uses a tradeoff between first and second order derivatives.

* HJ : Hessian of $J(\bar{w})$

$$H_{ij} = \frac{\partial^2 J(\bar{w})}{\partial w_i \partial w_j}$$

can also consider it as the Jacobian of ∇ (gradient)

H is symmetric.

Taylor expansion:

quadratic approximation of $J(\bar{w})$

$$J(\bar{w}) \approx \underline{J(\bar{w}_0) + [\bar{w}-\bar{w}_0]^T \nabla J(\bar{w}_0) + \frac{1}{2}(\bar{w}-\bar{w}_0)^T HJ(\bar{w}_0)(\bar{w}-\bar{w}_0)}$$

first order optimality condition: $\nabla J(\bar{w}) = \bar{0}$

Equivalently, applying this condition to the quadratic approximation to get

$$\nabla J(\bar{w}_0) + HJ(\bar{w}_0)(\bar{w}-\bar{w}_0) = \bar{0}$$

Rearranging terms here gives us the Newton method update:

$$\bar{w} \leftarrow \bar{w}_0 - [HJ(\bar{w}_0)]^{-1} \nabla J(\bar{w}_0)$$

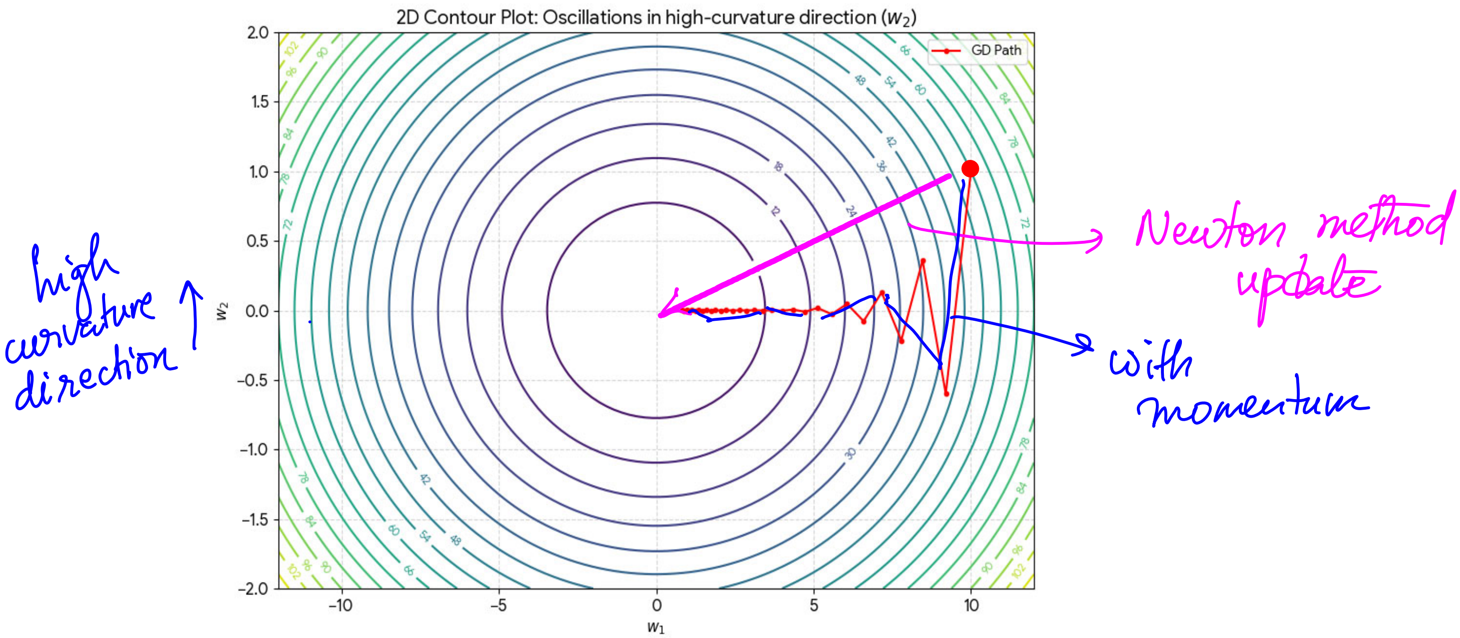
$\bar{w} \leftarrow \bar{w} - \alpha \nabla J$ gradient descent update

— there is no learning rate (α)!

— update is derived directly from the optimality condition.

— uses quadratic approximation of J (general loss function) and "goes directly to the bottom".

For $J = \frac{1}{2}\bar{w}_1^2 + 10\bar{w}_2^2$, the Newton method gets to the minimum in one step!



But for general loss functions, several Newton steps may be needed.

set $\bar{w}^0 = \bar{w}_0$ (initialization)

step k: compute $H = HJ(\bar{w}^k)$

$\nabla J = \nabla J(\bar{w}^k)$

set $\bar{w}^{k+1} = \bar{w}^k - H^{-1}\nabla J$.

continue until convergence. $(\|\bar{w}^{k+1} - \bar{w}^k\| < \epsilon)$ for small $\epsilon > 0$.

Can guarantee convergence in one step for quadratic loss functions J.