

Advanced Machine Learning Subsidiary Notes

Lecture 9: Optimisation

Adam Prügel-Bennett

April 21, 2020

1 Keywords

- Gradient descent, quadratic minima, differing length scales

2 Main Points

2.1 Optimisation

- Once you've designed your learning machine and chosen your loss function the rest is optimisation
- A very general method is to iteratively reduce your loss function
- In high dimensions the gradient of the loss function points in the direction of maximum increasing loss
- We still have the problem of determining the step size
- **Newton's Method**
 - Uses the Hessian, \mathbf{H} , with elements

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

- Assuming we are in a quadratic minimum the optima will be given by

$$\mathbf{w}^* = \mathbf{w} - \mathbf{H}^{-1} \nabla L(\mathbf{w})$$

- If we are not in a quadratic minimum, but sufficiently close we will converge to the minima *quadratically*
 - * quadratically means that if we start with an error ϵ the error will be ϵ^2 after one iteration ϵ^4 after two iterations, etc.
- If we are a long way from the minimum we might go anywhere
- In very high dimensions it is not practical
- **Quasi-Newton Methods**
 - There exists a host of methods that approximate the Hessian
 - By ensuring the approximation is positive definite it means we move in directions that are positively correlated with the gradient
 - Methods include *conjugate gradient* and *Levenberg-Marquardt*
 - Levenberg-Marquardt is used for least squares problems
 - These are usually preferred over Newton's methods as they are computationally cheaper

- **Gradient Descent**

- The alternative to (quasi-)Newton methods is to just follow the negative gradient

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - r \nabla L(\boldsymbol{w}^{(t)})$$

- If the step size, r , is too large we can diverge from quadratic minimum
- Need to tune the step size to the curvature of the problem
- In high dimension our maximum step size is limited by the direction with the greatest curvature (the largest eigenvalue of the Hessian)

3 Exercises

3.1 Divergence

- Assume a loss $L(w) = \frac{1}{2}w^2$ (we are in 1-dimension)
- If we update $w^{(t+1)} = w^{(t)} - r \nabla L(w^{(t)})$
 1. Compute the optimum step size
 2. What is r_{\max} such that we no longer converge if $r \geq r_{\max}$
 3. If $r > r_{\max}$ calculate how $w^{(t)}$ grows with time
- Answer given below

3.2 Quadratic Optima

- Consider a 2-d loss function $L(\boldsymbol{w}) = w_1^2/2 + w_2^2 - w_1 w_2$
 1. Compute the gradient
 2. Compute the Hessian
 3. Compute the eigenvalues of the Hessian
 4. Plot the contour lines of $L(\boldsymbol{w})$

4 Answers

4.1 Divergence

- The update equation is

$$w^{(t+1)} = (1 - r) w^{(t)}$$

1. The optimum step size is $r = 1$ (the minimum is at $w = 0$)
2. $r_{\max} = 2$
3. $w^{(t)} = (1 - r)^t w^{(0)}$

4.2 Quadratic Optima

1.

$$\nabla L(\boldsymbol{w}) = \begin{pmatrix} w_1 - w_2 \\ 2w_2 - w_1 \end{pmatrix}$$

2.

$$\mathbf{H} = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}$$

3. Let $T = \text{tr} \mathbf{H} = 3$ and $D = \det(\mathbf{H}) = 1$ then $\lambda = \frac{1}{2} \left(T \pm \sqrt{T^2 - 4D} \right) = (3 \pm \sqrt{5})/2 = \{0.382, 2.618\}$

