Exercises 2.1 Practice derivative calculations I

Compute the first and second derivatives of the following functions (remember to use the product/chain rules where necessary). *Hint: see appendix for more information on how to compute first and second derivatives if these concepts are unfamiliar.*

a) $g(w) = \frac{1}{2}qw^2 + rw + d$ where q, r, and d are constants;

b) $g(w) = -\cos\left(2\pi w^2\right) + w^2$;

c) $g(w) = \sum_{p=1}^{P} \log\left(1 + e^{-a_p w}\right)$ where $a_1 \ldots a_p$ are constants.

Exercises 2.2 Practice derivative calculations II

Compute the gradient and Hessian matrix of the following (remember to use the product/chain rules where necessary). Note that here $\mathbf{w}$ is an $N \times 1$ dimensional vector in all three cases. *Hint: see appendix for more information on how to compute gradients and Hessians if these concepts are unfamiliar.*

a) $g(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{Q}\mathbf{w} + \mathbf{r}^T\mathbf{w} + d$, here $\mathbf{Q}$ is an $N \times N$ symmetric matrix (i.e., $\mathbf{Q} = \mathbf{Q}^T$), $\mathbf{r}$ is an $N \times 1$ vector, and d is a scalar;

b) $g(\mathbf{w}) = -\cos\left(2\pi \mathbf{w}^T\mathbf{w}\right) + \mathbf{w}^T\mathbf{w}$;

c) $g(\mathbf{w}) = \sum_{p=1}^{P} \log\left(1 + e^{-\mathbf{a}_p^T\mathbf{w}}\right)$ where $\mathbf{a}_1 \ldots \mathbf{a}_p$ are $N \times 1$ vectors.

Exercises 2.5 First order Taylor series geometry

Verify that the normal vector to the tangent hyperplane generated by the first order Taylor series approximation centered at a point $\mathbf{v}$ shown in Equation (2.3) takes the form $\mathbf{n} = \begin{bmatrix} 1 \\ -\nabla g(\mathbf{v}) \end{bmatrix}$.

$$h(\mathbf{w}) = g(\mathbf{v}) + \nabla g(\mathbf{v})^T(\mathbf{w} - \mathbf{v}), \tag{2.3}$$

Exercises 2.7 Second order convexity calculations

In Fig. 2.14 we show several one-dimensional examples of convex functions. Confirm using the second order definition of convexity that each is indeed convex.

Exercises 2.8 A non-convex function whose only stationary point is a global minimum

a) Use the first order condition to determine the stationary point of $g(w) = w\tanh(w)$ where $\tanh(w)$ is the hyperbolic tangent function. To do this you might find it helpful to graph the first derivative $\frac{\partial}{\partial w}g(w)$ and see where it crosses the w axis. Plot the function to verify that the stationary point you find is the global minimum of the function.

b) Use the second order definition of convexity to show that g is non-convex. *Hint: you can plot the second derivative* $\frac{\partial^2}{\partial w^2}g(w)$.

Exercises 2.13 Code up gradient descent

In this exercise you will reproduce Fig. 2.10 by using gradient descent in order to minimize the function $g(\mathbf{w}) = -\cos\left(2\pi\mathbf{w}^T\mathbf{w}\right) + \mathbf{w}^T\mathbf{w}$. Use the wrapper ***two_d_grad_wrapper_hw*** to perform gradient descent, filling in the form of the gradient in the subfunction

[in,out] = gradient_descent(alpha,w0).

This subfunction performs gradient descent and is complete with exception of the gradient. Here *alpha* is a fixed step length and w0 is the initial point (both provided in the wrapper), the *in* variable contains each gradient step taken i.e., $in = \left\{\mathbf{w}^k = \mathbf{w}^{k-1} - \alpha\nabla g\left(\mathbf{w}^{k-1}\right)\right\}_{k=0}^{K}$ and corresponding *out* is a vector of the objective function evaluated at these steps, i.e., $out = \left\{g\left(\mathbf{w}^k\right)\right\}_{k=0}^{K}$ where K is the total number of steps taken. These are collected so they may be printed on the cost function surface for viewing.

Exercises 2.17 Code up Newton's method

a) Use the first order condition to determine the unique stationary point of the function $g(\mathbf{w}) = \log\left(1 + e^{\mathbf{w}^T\mathbf{w}}\right)$ where $N = 2$ i.e., $\mathbf{w} = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T$.

b) Make a surface plot of the function $g(\mathbf{w})$ or use the second order definition of convexity to verify that $g(\mathbf{w})$ is convex, implying that the stationary point found in part **a)** is a global minimum. *Hint: to check the second order definition use Exercise 2.10.*

c) Perform Newton's method to find the minimum of the function $g(\mathbf{w})$ determined in part **a)**. Initialize your algorithm at $\mathbf{w}^0 = \mathbf{1}_{N\times 1}$ and make a plot of the function value $g\left(\mathbf{w}^k\right)$ for ten iterations of Newton's method, as was done in Exercise 2.14 with gradient descent, in order to verify that your algorithm works properly and is converging.

Make sure to follow the Newton's method algorithm as described in Algorithm 2.2 with only the maximum iteration stopping condition, and use the pseudo-inverse solution to each Newton system in your implementation as given in (2.21).

d) Now run your Newton's method code from part **c)** again, this time initializing at the point $\mathbf{w}^0 = 4 \cdot \mathbf{1}_{N\times 1}$. While this initialization is further away from the unique minimum of $g(\mathbf{w})$ than the one used in part **c)**, your Newton's method algorithm should converge *faster* starting at this point. At first glance this result seems very counter-intuitive, as we (rightly) expect that an initial point closer to a minimum will provoke more rapid convergence of Newton's method!

Can you explain why this result actually makes sense for the particular function $g(\mathbf{w})$ we are minimizing here? Or, in other words, why the minimum of the second order Taylor series approximation of $g(\mathbf{w})$ centered at $\mathbf{w}^0 = 4 \cdot \mathbf{1}_{N\times 1}$ is essentially the minimum of $g(\mathbf{w})$ itself? *Hint: use the fact for large values of t that $\log\left(1 + e^t\right) \approx t$, and that the second order Taylor series approximation of a quadratic function (like the one given in part **b)** of Exercise 2.4) is just the quadratic function itself.*