Homework 1

January 13, 2022

Please submit your HW on Canvas; include a PDF printout of any code and results, clearly labeled, e.g. from a Jupyter notebook. It is due Wednesday January 19th by 11:59pm EST.

Problem 1

Find the derivatives of the following functions. **Check your answers numerically** by computing $f(x+\delta x) - f(x)$ for a small random δx and compare it to your linear operator $f'(x)[\delta x]$. (Use Julia, Matlab, Python/Numpy, or any language/library of your choice that has matrix–vector operations.)

a) $f(x) = (x^T x)^4$ a scalar function of the vector $x \in \mathbb{R}^n$

b) $f(x) = \cos(x^T A x)$ a scalar function of the vector $x \in \mathbb{R}^n$ (where $A \in \mathbb{R}^{n,n}$)

c) $f(A) = \mathrm{trace}(A^4)$, a scalar function of $A \in \mathbb{R}^{n,n}$ (Hint: use the cyclic property of trace.)

d) $f(A) = A^4$ where $A \in \mathbb{R}^{n,n}$. Express your answer as a linear operator.

e) $f(A) = \theta^T A$, where $\theta \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n,m}$. Express your answer as a linear operator.

f) $f(x) = \sin.(x)$, meaning the *element-wise* application of the sine function to each entry of a vector $x \in \mathbb{R}^n$, whose result is another n-component vector. Express your answer as a Jacobian matrix.

Problem 2

As discussed in class, a typical neural network (NN) is a sequence of N "layers": you start with a vector of inputs x_0, pass it through a function f_1, then to a function f_2, and so on. This can be written as a recurrence relation:

$$x_k = f_k(p, x_{k-1})$$

where x_k is the vector of values in layer k and $p \in \mathbb{R}^n$ is the vector of all the free parameters of the NN (the weights, biases, etcetera). That is, the final output layer x_N, after N steps of the recurrence, is the computation $x_N = f_N(p, f_{N-1}(p, f_{N-2}(\cdots f_1(p, x_0))))$ One then computes a *scalar* loss function $L(p) = (x_N(p) - y_0)^T (x_N(p) - y_0)$ measuring the accuracy of the neural network against the correct answer y_0 (in practice averaged over many "training" pairs (x_0, y_0), but here with just one for simplicity). We want the derivative L', i.e. the gradient, in order to minimize the loss by moving (more-or-less) "downhill" in parameter space.

a) Evaluate L' left-to-right ("back-propagation"), as in class for $N = 2$. Write down a recurrence relation, involving no matrix–matrix products (only vector–matrix/matrix–vector products and additions), which yields the gradient L' after $\approx N$ steps.

b) Suppose that there are n parameters $p_k \in \mathbb{R}^n$ per layer, and $p = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{pmatrix} \in \mathbb{R}^{nN}$ is a stack of the parameters for each layer (i.e., f_k only depends explicitly on p_k). Explain how this leads to a "sparse" (mostly zero) Jacobian $\frac{\partial f_k}{\partial p}$, sketch the pattern of nonzero entries, and explain how this could be exploited to evaluate your recurrence in the previous part more efficiently.

Problem 3

Consider a vector space V of differentiable real-valued *functions* $v(x)$ on $x \in [0,1]$, which vanish at the endpoints: $v(0) = v(1) = 0$.[1] (Be sure you understand why this is a vector space!)

a) Let $f(v) = \int_0^1 \sin(v(x))dx$. What is $f'(v)$ as a linear operator? That is, similar to what we did in class, $f'(v)[\delta v] \approx f(v + \delta v) - f(v)$, to first order in δv, for any small perturbation *function* $\delta v \in V$.

b) Let $g(v) = \int_0^1 \sqrt{1 + v'(x)^2}dx$, where $v'(x)$ is the derivative. What is $g'(v)$ as a linear operator $g'(v)[\delta v]$ acting on the perturbation $\delta v \in V$, as in the previous part? Express your operator in terms of δv, *not* the derivative $\delta v'$. (Hint: integrate by parts.)

c) As in 18.01, an extremum occurs when $g' = 0$, i.e. when $g'(v)[\delta v] = 0$ for *any* δv. With the $g(v)$ from part (b), for what functions v is $g' = 0$? Is it a maximum or a minimum of g?

d) Geometrically, $g(v)$ is the ______ of the curve $v(x)$, and so its minimum/maximum (choose 1) occurs when $v(x)$ is a ______.

Problem 4

Write down the Jacobian for $Y = A^T SA$, where A is a fixed 2x2 matrix, and S and Y are symmetric 2×2 input matrices and output matrices. Write the Jacobian explicitly as a 3×3 matrix involving elements of A, in terms of the 3 degrees of freedom of the inputs S and outputs Y.

[1] Technically, we need to restrict ourselves to functions $v(x)$ where the integrals $f(v)$ and $g(v)$ in the problem exist; this is related to a special kind of vector space called a "Sobolev space." It's not worth worrying about this here; just assume the integrals don't blow up.