Homework 4

Deadline: Wednesday, Feb. 14, at 11:59pm.

Submission: You must submit your solutions as a PDF file through MarkUs[1]. You can produce the file however you like (e.g. LaTeX, Microsoft Word, scanner), as long as it is readable.

Late Submission: MarkUs will remain open until 2 days after the deadline; until that time, you should submit through MarkUs. If you want to submit the assignment more than 2 days late, please e-mail it to `csc321staff@cs.toronto.edu`. The reason for this is that MarkUs won't let us collect the homeworks until the late period has ended, and we want to be able to return them to you in a timely manner.

Weekly homeworks are individual work. See the Course Information handout[2] for detailed policies.

1. **Gradient descent. [5pts]** We can get quite a bit of insight into the behavior of gradient descent by looking at how it behaves on quadratic functions. Suppose we are trying to optimize a quadratic function

$$\mathcal{C}(\boldsymbol{\theta}) = \frac{a_1}{2}(\theta_1 - r_1)^2 + \cdots + \frac{a_N}{2}(\theta_N - r_N)^2,$$

with each $a_i > 0$. We can exactly solve for the dynamics of gradient descent. In other words, we can find an exact formula for $\theta_i^{(t)}$, where t is the number of gradient descent updates.

(a) **[1pt]** Derive the gradient descent update rule for each θ_i with learning rate α. It should have the form

$$\theta_i^{(t+1)} = \cdots,$$

where the right-hand side is some function of the previous value $\theta_i^{(t)}$, as well as r_i, a_i, and α. (It's an interesting and useful fact that the different θ_i's evolve independently, so we can analyze a single coordinate at a time.)

(b) **[2pts]** Now let's introduce the *error* $e_i^{(t)} = \theta_i^{(t)} - r_i$. Take your update rule from the previous part, and write a recurrence for the errors. It should have the form

$$e_i^{(t+1)} = \cdots,$$

where the right-hand side is some function of $e_i^{(t)}$, a_i, and α.

(c) **[1pt]** Solve this recurrence to obtain an explicit formula for $e_i^{(t)}$ in terms of the initial error $e_i^{(0)}$. For what values of α is the procedure stable (the errors decay over time), and for what values is it unstable (the errors grow over time)?

Aside: your answer will indicate that large learning rates are more unstable than small ones, and that high curvature dimensions are more unstable than low curvature ones.

(d) **[1pt]** Using your answer for the previous part, write an explicit formula for the cost $\mathcal{C}(\boldsymbol{\theta}^{(t)})$ as a function of the initial values $\boldsymbol{\theta}^{(0)}$. (You can write it as a summation over indices, i.e., you don't need to vectorize it.) As $t \to \infty$, which term comes to dominate?

[1] `https://markus.teach.cs.toronto.edu/csc321-2018-01`

[2] `http://www.cs.toronto.edu/~rgrosse/courses/csc321_2018/syllabus.pdf`

Aside: you'll find that if you use the optimal α, the asymptotic behavior roughly depends on the condition number

$$\kappa = \frac{\max_i a_i}{\min_i a_i}.$$

This supports the claim that narrow ravines are problematic for gradient descent.

(e) **[Optional (not marked), and advanced]** This part is optional, but you may find it interesting. We'll make use of eigendecompositions of symmetric matrices; see MIT OpenCourseware for a refresher:

`https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/positive-definite-matrices-and-applications/`

`symmetric-matrices-and-positive-definiteness/`

It turns out we've actually just analyzed the fully general quadratic case. I.e., suppose we try to minimize a cost function of the form

$$\mathcal{C}(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} - \mathbf{r})^T \mathbf{A}(\boldsymbol{\theta} - \mathbf{r}),$$

where $\mathbf{A}$ is a symmetric positive definite matrix, i.e. a symmetric matrix with all positive eigenvalues. (This is the general form for a quadratic function which curves upwards.) Determine the gradient descent update for $\boldsymbol{\theta}$ in vectorized form. Then write a recurrence for the error vector $\mathbf{e} = \boldsymbol{\theta} - \mathbf{r}$, similarly to Part (1b). It will have the form

$$\mathbf{e}^{(t+1)} = \mathbf{B}\mathbf{e}^{(t)},$$

where $\mathbf{B}$ is a symmetric matrix. Determine the eigenvectors and eigenvalues of $\mathbf{B}$ in terms of the eigenvectors and eigenvalues of $\mathbf{A}$, and use this to find an explicit form for $\mathbf{e}^{(t)}$ and for $\mathcal{C}(\boldsymbol{\theta}^{(t)})$ in terms of $\boldsymbol{\theta}^{(0)}$. The result will be closely related to your answer from Part (1d).

2. **Dropout. [5pts]** For this question, you may wish to review the properties of expectation and variance: `https://metacademy.org/graphs/concepts/expectation_and_variance`

Dropout has an interesting interpretation in the case of linear regression. Recall that the predictions are made stochastically as:

$$y = \sum_j m_j w_j x_j,$$

where the m_j's are all i.i.d. (independnet and identically distributed) Bernoulli random variables with expectation 1/2. (I.e., they are indepdendent for every input dimension and every data point.) We would like to minimize the cost

$$\mathcal{E} = \frac{1}{2N} \sum_{i=1}^{N} \mathbb{E}[(y^{(i)} - t^{(i)})^2], \tag{1}$$

where the expectation is with respect to the $m_j^{(i)}$'s.

Now we show that this is equivalent to a regularized linear regression problem:

(a) **[2pts]** Find expressions for $\mathbb{E}[y]$ and $\mathrm{Var}[y]$ for a given $\mathbf{x}$ and $\mathbf{w}$.

(b) **[1pt]** Determine $\tilde{w}_j$ as a function of w_j such that

$$\mathbb{E}[y] = \tilde{y} = \sum_j \tilde{w}_j x_j.$$

Here, $\tilde{y}$ can be thought of as (deterministic) predictions made by a different model.

(c) **[2pts]** Using the model from the previous section, show that the cost $\mathcal{E}$ (Eqn. 1) can be written as

$$\mathcal{E} = \frac{1}{2N} \sum_{i=1}^{N} (\tilde{y}^{(i)} - t^{(i)})^2 \ + \ \mathcal{R}(\tilde{w}_1, \ldots, \tilde{w}_D),$$

where $\mathcal{R}$ is a function of the $\tilde{w}_D$'s which does not involve an expectation. I.e., give an expression for $\mathcal{R}$. (Note that $\mathcal{R}$ will depend on the data, so we call it a "data-dependent regularizer.")

Hint: write the cost in terms of the mean and variance formulas from part (a). For inspiration, you may wish to refer to the derivation of the bias/variance decomposition from Lecture 9.