Solution 1: Taylor Approximation

Given a scalar function $f : \mathbb{R}^n \to \mathbb{R}$, its first-order Taylor approximation about a point $\mathbf{z} \in \mathbb{R}^n$ is given by $\widehat{f}(\mathbf{x}) = f(\mathbf{z}) + \nabla f(\mathbf{z})^\top (\mathbf{x} - \mathbf{z})$. We will consider the function,

$$E(u, v) = e^u + e^{2v} + e^{uv} + u^2 - 3uv + 4v^2 - 3u - 5v,$$

where u and v are scalars.

a) The first-order Taylor approximation of $E(u + \Delta u, v + \Delta v)$ about $\mathbf{z} = \begin{bmatrix} u & v \end{bmatrix}^\top = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$ is

$$\widehat{E}(\Delta u, \Delta v) = E(0, 0) + \nabla E(0, 0)^\top \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}.$$

We have $E(0, 0) = 3$ and $\nabla E(u, v) = \begin{bmatrix} \frac{\partial E}{\partial u} \\ \frac{\partial E}{\partial v} \end{bmatrix} = \begin{bmatrix} e^u v e^{uv} + 2u - 3v - 3 \\ 2e^{2v} + u e^{uv} + 2u + 8v - 5 \end{bmatrix}$. Substituting in above, we get

$$\widehat{E}(\Delta u, \Delta v) = 3 + \begin{bmatrix} -2 & -3 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = 3 - 2\Delta u - 3\Delta v.$$

Comparing the expression above, we can infer $a = 3, a_u = -2, a_v = -3$.

b) To minimize $\widehat{E}$ over all possible steps of Δu and Δv, such that the stepsize $||(\Delta u, \Delta v)|| = 0.5$, we use our knowledge of gradient descent. Particularly, the step should be in the direction of $-\nabla \widehat{E}$. Therefore,

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = -c \nabla \widehat{E}(0, 0) = c \begin{bmatrix} 2 \\ 3 \end{bmatrix},$$

for some $c > 0$. Now, applying the stepsize constraint, we have $0.5 = c\sqrt{2^2 + 3^2} \iff c = \frac{1}{2\sqrt{13}}$. Hence, the minimizer at $(0, 0)$ is

$$\begin{bmatrix} \Delta u^\star \\ \Delta v^\star \end{bmatrix} = \frac{1}{2\sqrt{13}} \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

Solution 2: Logistic and Multi-class Softmax Regression (Midterm 2019, Problem 4)

We consider a multi-class softmax regression model to classify datavectors $\mathbf{x}_i \in \mathbb{R}^{d+1}$ into class labels $y_i \in \{1, 2\}$. The weightvectors $\mathbf{w}(1), \mathbf{w}(2)$ correspond to classes 1 and 2 respectively. Given any datavector $\mathbf{x}_n \in \mathbb{R}^{d+1}$, the model yields the probability estimate for true class $y_n \in \{1, 2\}$ as

$$\hat{p}_\Omega^{\mathrm{SM}}(y_n \mid \mathbf{x}_n) = \frac{e^{\mathbf{w}(y_n)^\top \mathbf{x}_n}}{\sum_{i=1}^{2} e^{\mathbf{w}(i)^\top \mathbf{x}_n}}, \tag{1}$$

where $\Omega = \{\mathbf{w}(1), \mathbf{w}(2)\}$. The corresponding loss function for the training sample $(\mathbf{x}_n, y_n)$ is

$$e_n^{\mathrm{SM}}(\Omega) = -\log\left(\hat{p}_\Omega^{\mathrm{SM}}(y_n \mid \mathbf{x}_n)\right). \tag{2}$$

Similarly, we consider the binary logistic regression model from Homework 2, that produces the probability estimate for the datavector belonging to class 1 as

$$\hat{p}_{\mathbf{w}}^{\mathrm{LR}}(y_n = 1 \mid \mathbf{x}_n) = \frac{e^{\mathbf{w}^\top \mathbf{x}_n}}{1 + e^{\mathbf{w}^\top \mathbf{x}_n}}. \tag{3}$$

Subsequently, the probability estimate for class 2 is $\hat{p}_{\mathbf{w}}^{\mathrm{LR}}(y_n = 2 \mid \mathbf{x}_n) = 1 - \hat{p}_{\mathbf{w}}^{\mathrm{LR}}(y_n = 1 \mid \mathbf{x}_n)$. The single-sample loss function remains identical to softmax and is given by

$$e_n^{\mathrm{LR}}(\mathbf{w}) = -\log\left(\hat{p}_{\mathbf{w}}^{\mathrm{LR}}(y_n \mid \mathbf{x}_n)\right). \tag{4}$$

a) The gradient of e_n^{SM} will have to be calculated for both cases of y_n. We first expand the single-sample loss function in (2) as

$$e_n^{\text{SM}}(\Omega) = -\mathbf{w}(y_n)^\top \mathbf{x}_n + \log\left(\sum_{i=1}^{2} e^{\mathbf{w}(i)^\top \mathbf{x}_n}\right). \tag{5}$$

Now, evaluating the gradient about $\mathbf{w}(j)$ some $j \in \{1, 2\}$ we have

$$\nabla_{\mathbf{w}(j)} e_n^{\text{SM}}(\Omega) = \underbrace{\nabla_{\mathbf{w}(j)}\big[-\mathbf{w}(y_n)^\top \mathbf{x}_n\big]}_{\nabla_{\mathbf{w}(j)} e_{n,1}^{\text{SM}}} + \underbrace{\nabla_{\mathbf{w}(j)}\left[\log\left(\sum_{i=1}^{2} e^{\mathbf{w}(i)^\top \mathbf{x}_n}\right)\right]}_{\nabla_{\mathbf{w}(j)} e_{n,2}^{\text{SM}}}. \tag{6}$$

Further evaluating the two subparts in (6), we get

$$\nabla_{\mathbf{w}(j)} e_{n,1}^{\text{SM}} = \begin{cases} 0 & y_n \neq j \\ -\mathbf{x}_n & y_n = j \end{cases}, \tag{7}$$

$$\nabla_{\mathbf{w}(j)} e_{n,2}^{\text{SM}} = \frac{e^{\mathbf{w}(j)^\top \mathbf{x}_n} \nabla_{\mathbf{w}(j)} (\mathbf{w}(j)^\top \mathbf{x}_n)}{\sum_{i=1}^{2} e^{\mathbf{w}(i)^\top \mathbf{x}_n}} = \mathbf{x}_n \frac{e^{\mathbf{w}(j)^\top \mathbf{x}_n}}{\sum_{i=1}^{2} e^{\mathbf{w}(i)^\top \mathbf{x}_n}}. \tag{8}$$

Finally, after combining (7) and (8), we get the gradient of the loss function over $y_n, j \in \{1, 2\}$ as

$$\nabla_{\mathbf{w}(j)} e_n^{\text{SM}}(\Omega) = \begin{cases} \mathbf{x}_n \frac{e^{\mathbf{w}(j)^\top \mathbf{x}_n}}{\sum_{i=1}^{2} e^{\mathbf{w}(i)^\top \mathbf{x}_n}} & y_n \neq j \\ \mathbf{x}_n \left(\frac{e^{\mathbf{w}(j)^\top \mathbf{x}_n}}{\sum_{i=1}^{2} e^{\mathbf{w}(i)^\top \mathbf{x}_n}} - 1\right) & y_n = j \end{cases}. \tag{9}$$

b) For the classification probability estimates to be identical across logistic and softmax regression, we need to have the following equality,

$$\hat{p}_\Omega^{\text{SM}}(y_n = 1 \mid \mathbf{x}_n) = \hat{p}_{\mathbf{w}}^{\text{LR}}(y_n = 1 \mid \mathbf{x}_n) \iff \frac{e^{\mathbf{w}(1)^\top \mathbf{x}_n}}{e^{\mathbf{w}(1)^\top \mathbf{x}_n} + e^{\mathbf{w}(2)^\top \mathbf{x}_n}} = \frac{e^{(\mathbf{w}(1)-\mathbf{w}(2))^\top \mathbf{x}_n}}{1 + e^{(\mathbf{w}(1)-\mathbf{w}(2))^\top \mathbf{x}_n}} = \frac{e^{\mathbf{w}^\top \mathbf{x}_n}}{1 + e^{\mathbf{w}^\top \mathbf{x}_n}} \tag{10}$$

$$\iff \mathbf{w} = \mathbf{w}(1) - \mathbf{w}(2). \tag{11}$$

Thus, both models are identical if and only if (11) holds.

c) The stochastic gradient descent (SGD) involves the following weight update rule for a randomly selected datavector index n in the dataset

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \epsilon \nabla_{\mathbf{w}_k} e_n, \tag{12}$$

where $k \in \mathbb{N}$ is the iteration or epoch number and $\epsilon > 0$ is the learning rate. Having already calculated the gradient of single-sample loss in softmax regression, we invoke the same for logistic regression below from our work in Homework 2.

$$\nabla_{\mathbf{w}} e_n^{\text{LR}}(\mathbf{w}) = \begin{cases} \mathbf{x}_n \frac{-e^{-\mathbf{w}^\top \mathbf{x}_n}}{1+e^{-\mathbf{w}^\top \mathbf{x}_n}} & y_n = 1 \\ \mathbf{x}_n \frac{e^{\mathbf{w}^\top \mathbf{x}_n}}{1+e^{\mathbf{w}^\top \mathbf{x}_n}} & y_n = 2 \end{cases}. \tag{13}$$

Therefore, the SGD weight updates for logistic regression for $y_n = 1$ will be

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \epsilon^{\text{LR}} \frac{\mathbf{x}_n e^{-\mathbf{w}_k^\top \mathbf{x}_n}}{1 + e^{-\mathbf{w}_k^\top \mathbf{x}_n}}. \tag{14}$$

Similarly, using (9), the the SGD weight updates for softmax regression for $y_n = 1$ will be

$$\mathbf{w}_{k+1}(1) = \mathbf{w}_k(1) + \epsilon^{\text{SM}} \frac{\mathbf{x}_n e^{\mathbf{w}_k(2)^\top \mathbf{x}_n}}{e^{\mathbf{w}_k(1)^\top \mathbf{x}_n} + e^{\mathbf{w}_k(2)^\top \mathbf{x}_n}} \tag{15}$$

$$\mathbf{w}_{k+1}(2) = \mathbf{w}_k(2) - \epsilon^{\text{SM}} \frac{\mathbf{x}_n e^{\mathbf{w}_k(2)^\top \mathbf{x}_n}}{e^{\mathbf{w}_k(1)^\top \mathbf{x}_n} + e^{\mathbf{w}_k(2)^\top \mathbf{x}_n}}. \tag{16}$$

Now, comparing (14) to (15) − (16) and utilizing (11) to simplify the gradient terms in softmax regression, we can infer that

$$\mathbf{w}_k + \epsilon^{\text{LR}} \frac{\mathbf{x}_n}{1 + e^{\mathbf{w}_k^\top \mathbf{x}_n}} = \underbrace{(\mathbf{w}_k(1) - \mathbf{w}_k(2))}_{\mathbf{w}_k} + 2\epsilon^{\text{SM}} \frac{\mathbf{x}_n}{1 + e^{\mathbf{w}_k^\top \mathbf{x}_n}} \iff \epsilon^{\text{LR}} = 2\epsilon^{\text{SM}}, \tag{17}$$

can only make the evolution of SGD weight updates identical across logistic and softmax regression.