CSE – 426 Homework 9

Griffin Kent

(The class skipped Homework 8)

1) Given the training data $\boldsymbol{x}^{(1)} = [1,1]^T, y^{(1)} = 1, \boldsymbol{x}^{(2)} = [1,0]^T, y^{(1)} = 1, \boldsymbol{x}^{(3)} = [-1,0]^T, y^{(3)} = 0$, and $\boldsymbol{x}^{(4)} = [-1,-1]^T, y^{(1)} = 0$, find the within-class and between-class scatter matrices S_W and S_B for linear discriminant analysis. Don't just give the final matrices but show step-by-step going from $\boldsymbol{x}$ to S_0, S_1, S_W, $\boldsymbol{m}_0$, $\boldsymbol{m}_1$, and S_B.

(Solution): Separating the training examples into groups corresponding to their labels, we have

$$C_1 = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} : \boldsymbol{y} = 1\}$$
$$C_0 = \{\boldsymbol{x}^{(3)}, \boldsymbol{x}^{(4)} : \boldsymbol{y} = 0\}.$$

We can then calculate the mean vectors of the two groups by using the equation

$$\boldsymbol{m}_k = \frac{1}{N_k} \sum_{\boldsymbol{x} \in C_k} \boldsymbol{x}, \quad k = 0,1.$$

Then,

$$\boldsymbol{m}_1 = \frac{1}{2}\{[1,1]^T + [1,0]^T\} = \left[1, \frac{1}{2}\right]^T.$$
$$\boldsymbol{m}_0 = \frac{1}{2}\{[-1,0]^T + [-1,-1]^T\} = \left[-1, -\frac{1}{2}\right]^T.$$

We can then calculate the between-class scatter matrix S_B according to the following:

$$S_B = (\boldsymbol{m}_0 - \boldsymbol{m}_1)(\boldsymbol{m}_0 - \boldsymbol{m}_1)^T$$
$$= \left(\begin{bmatrix} -1 \\ -\frac{1}{2} \end{bmatrix} - \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}\right)\left(\begin{bmatrix} -1 \\ -\frac{1}{2} \end{bmatrix} - \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}\right)^T$$
$$= \begin{bmatrix} -2 \\ -1 \end{bmatrix}[-2,-1] = \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}.$$

The within-class scatter matrices can be found by the following equation:

$$S_k = \sum_{\boldsymbol{x} \in C_k} \boldsymbol{x}\boldsymbol{x}^T.$$

Then,

$$S_0 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}[-1,0] + \begin{bmatrix} -1 \\ -1 \end{bmatrix}[-1,-1]$$
$$= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

And

$$S_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}[1,1] + \begin{bmatrix} 1 \\ 0 \end{bmatrix}[1,0]$$
$$= \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

Thus, the within-class matrix is

$$S_W = S_0 + S_1 = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}.$$

■

2) Let the eigen-decomposition $A = PDP^{-1}$. Using the properties of the matrices P and D described in the lecture note of "Dimension Reduction", prove that

$$A^n = PD^nP^{-1}$$

Where A^n and D^n mean the n-th power of the matrices A and D, respectively.

(Proof): We will use a proof by induction.

Let $n = 1$ be the base case: $A = PDP^{-1}$.

Then, assume that the relationship $A^n = PD^nP^{-1}$ is true $\forall n > 1$.

Then our goal is to now show that $A^{n+1} = PD^{n+1}P^{-1}$; to that end:

$$A^{n+1} = A^n \cdot A = PD^nP^{-1} \cdot PDP^{-1}.$$

Since a matrix multiplied by its inverse is the identity matrix, we have

$$= PD^nDP^{-1} = PD^{n+1}P^{-1}.$$

Therefore, since we have shown that $A^{n+1} = PD^{n+1}P^{-1}$, $\forall n > 1$, we have proven that $A^n = PD^nP^{-1}$, completing the proof.

■

3) Given m training examples, feeding them through an MLP will generate m activation vectors at the output layer, collected in the matrix $A \in \mathbb{R}^{n[L] \times m}$. Let $Y \in \{0,1\}^{n[L] \times m}$ be the label matrix, with the i-th column being the one-hot label vector for the i-th training example. Prove that

$$\frac{\partial J}{\partial W^{[L]}} = \frac{1}{B}\left(A^{[L]} - Y\right)\left(A^{[L-1]}\right)^T \in \mathbb{R}^{n[L] \times n[L-1]}$$

Is indeed the average of the gradients of the loss function J with respect to $W^{[L]}$ over B training examples in a mini-batch. Your answer should contain the gradients of each of the B training examples.

(Proof): We know that the gradient of J with respect to $W^{[L](i)}$ for one training example was derived (Eq. (27) from lecture notes) to be the following:

$$\frac{dJ}{dW^{[L](i)}} = \left(\boldsymbol{a}^{[L](i)} - y^{(i)}\right)\left(\boldsymbol{a}^{[L-1](i)}\right)^T$$

Where $\boldsymbol{a}^{[L](i)} \in \mathbb{R}^{n[L] \times 1}$ and $y^{(i)} \in \mathbb{R}^{n[L] \times 1}$ are column vectors of the i-th training example and $\left(\boldsymbol{a}^{[L-1](i)}\right)^T \in \mathbb{R}^{1 \times n[L-1]}$ is a row vector of the i-th training example. We can see that $\frac{dJ}{dW^{[L](i)}} \in \mathbb{R}^{n[L] \times n[L-1]}$. Then, taking the average of this over the examples in a mini-batch of size B, we have

$$\frac{1}{B}\sum_{i=1}^{B} \frac{dJ}{dW^{[L](i)}}$$

$$= \frac{1}{B}\sum_{i=1}^{B}\left(\boldsymbol{a}^{[L](i)} - y^{(i)}\right)\left(\boldsymbol{a}^{[L-1](i)}\right)^T \in \mathbb{R}^{n[L] \times n[L-1]}.$$

If we let $A \in \mathbb{R}^{n[L]\times m}$ and $Y \in \{0,1\}^{n[L]\times B}$ be matrices where each column represents the i-th training example and $\left(A^{[L-1]}\right)^T \in \mathbb{R}^{B\times n[L-1]}$ be a matrix with rows that correspond to the training examples, then we can represent the summations in terms of the matrix-vector product

$$= \frac{1}{B}\left(A^{[L]} - Y\right)\left(A^{[L-1]}\right)^T \in \mathbb{R}^{n[L]\times n[L-1]}.$$

Therefore, we can see that this expression is indeed the average of the gradients of the loss function J with respect to $W^{[L]}$ over B training examples in a mini-batch; completing the proof.

■

4) MLP can also be used for regression on multiple target values. Given training examples $\left\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right\}_{i=1}^{m}$ where $\boldsymbol{y}^{(i)} \in \mathbb{R}^k$ and $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$. MSE loss will be appropriate for training an MLP to predict $\hat{\boldsymbol{y}}^{(i)}$, so that $\hat{\boldsymbol{y}}^{(i)}$ is as close as possible to $\boldsymbol{y}^{(i)}$ for the input $\boldsymbol{x}^{(i)}$. The MSE loss function will then be

$$J = \frac{1}{m}\sum_{i=1}^{m}\left\|\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}\right\|_2^2,$$

Where $\|\boldsymbol{w}\|_2$ is the 2-norm of the vector $\boldsymbol{w}$. Find the partial derivative of J with respect to $\hat{\boldsymbol{y}}^{(i)}$. Is the softmax activation function appropriate for regression MLP? Justify your answer.

(Solution): Taking the partial derivative of J with respect to $\hat{\boldsymbol{y}}^{(i)}$:

$$\frac{\partial J}{\partial \hat{\boldsymbol{y}}^{(i)}} = \frac{1}{m}\frac{\partial}{\partial \hat{\boldsymbol{y}}^{(i)}}\sum_{i=1}^{m}\left\|\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}\right\|_2^2$$

$$= \frac{1}{m}\frac{\partial}{\partial \hat{\boldsymbol{y}}^{(i)}}\left\|\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}\right\|_2^2$$

$$= \frac{1}{m}\frac{\partial}{\partial \hat{\boldsymbol{y}}^{(i)}}\left(\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}\right)^2$$

$$= \frac{1}{m}2\left(\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}\right)\frac{\partial}{\partial \hat{\boldsymbol{y}}^{(i)}}\left(\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}\right)$$

$$= \frac{2}{m}\left(\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}\right).$$

The softmax function is only appropriate for regression tasks where the target variables are discrete. That is because the softmax function calculates the respective probabilities of classifying an example as one of K possible classes, where K is finite. However, when the class labels are continuous, there is an infinite number of classes. Therefore, the softmax function is not appropriate for classifying multiple *continuous* variables; however, it will be appropriate in classifying multiple *discrete* variables.

■

5) (Graduate Only) This question is related to Project 3. In the function *explain* of the *NN* class, you will be asked to find $\frac{\partial z_c^{[L]}}{\partial \boldsymbol{a}^{[0]}}$, the gradient of $z_c^{[L]}$ with respect to the input data $\boldsymbol{a}^{[0]}$, where $z_c^{[L]} = W_c^{[L]}\boldsymbol{a}^{[L-1]}$ and $W_c^{[L]}$ is the c-th row of the parameter matrix $W^{[L]}$, and $\boldsymbol{a}^{[0]} \in \mathbb{R}^n$ is the input vector. Use the following hints to find $\frac{\partial z_c^{[L]}}{\partial \boldsymbol{a}^{[0]}}$.

[*Hints: First find the gradient* $\frac{\partial z_c^{[L]}}{\partial \boldsymbol{z}^{[L-1]}}$*, then follow the backpropagation algorithm to find the gradient* $\frac{\partial z_c^{[L]}}{\partial \boldsymbol{z}^{[\ell]}}$ *for all* $1 \le \ell \le L-1$*. Lastly, find* $\frac{\partial z_c^{[L]}}{\partial \boldsymbol{a}^{[0]}}$ *in terms of* $\frac{\partial z_c^{[L]}}{\partial \boldsymbol{z}^{[1]}}$*. Your final expression should be a chain of matrix multiplications and element-wise vector multiplications. To use your answer in project 3, you will need to vectorize your final expression for multiple input vectors.*]

(Solution): To begin, taking the partial of $z_c^{[L]} = W_c^{[L]}\boldsymbol{a}^{[L-1]}$ with respect to $\boldsymbol{a}^{[L-1]}$, we have

$$\frac{\partial z_c^{[L]}}{\partial \boldsymbol{a}^{[L-1]}} = \left(W_c^{[L]}\right)^T \frac{\partial z_c^{[L]}}{\partial z_c^{[L]}} = \left(W_c^{[L]}\right)^T \in \mathbb{R}^{n[L-1]\times 1}.$$

The next step is to calculate the partial of $z_c^{[L]}$ with respect to $\boldsymbol{z}^{[L-1]}$; using Eq. (30) from the lecture notes:

$$\frac{\partial z_c^{[L]}}{\partial \boldsymbol{z}^{[L-1]}} = \left[\frac{\partial z_c^{[L]}}{\partial \boldsymbol{a}^{[L-1]}}\right] \otimes \left(g^{[L-1]}\right)'\left(\boldsymbol{z}^{[L-1]}\right)$$
$$= \left(W_c^{[L]}\right)^T \otimes \left(g^{[L-1]}\right)'\left(\boldsymbol{z}^{[L-1]}\right) \in \mathbb{R}^{n[L-1]\times 1}.$$

Where $\otimes$ is the element-wise multiplication of two vectors.

Calculating the partial of $z_c^{[L]}$ with respect to $\boldsymbol{a}^{[L-2]}$:

$$\frac{\partial z_c^{[L]}}{\partial \boldsymbol{a}^{[L-2]}} = \left(W^{[L-1]}\right)^T \frac{\partial z_c^{[L]}}{\partial \boldsymbol{z}^{[L-1]}} \in \mathbb{R}^{n[L-2]\times 1}.$$

Calculating the partial of $z_c^{[L]}$ with respect to $\boldsymbol{z}^{[L-2]}$:

$$\frac{\partial z_c^{[L]}}{\partial \boldsymbol{z}^{[L-2]}} = \left[\frac{\partial z_c^{[L]}}{\partial \boldsymbol{a}^{[L-2]}}\right] \otimes \left(g^{[L-2]}\right)'\left(\boldsymbol{z}^{[L-2]}\right) \in \mathbb{R}^{n[L-2]\times 1}.$$

We can further continue this process for all $1 \le \ell \le L-1$ until we can calculate the partial of $z_c^{[L]}$ with respect to $\boldsymbol{z}^{[1]}$:

$$\frac{\partial z_c^{[L]}}{\partial \boldsymbol{z}^{[1]}} = \left[\frac{\partial z_c^{[L]}}{\partial \boldsymbol{a}^{[1]}}\right] \otimes \left(g^{[1]}\right)'\left(\boldsymbol{z}^{[1]}\right) \in \mathbb{R}^{n[1]\times 1}.$$

Finally, we can obtain an expression for the partial of $z_c^{[L]}$ with respect to $\boldsymbol{a}^{[0]}$

$$\frac{\partial z_c^{[L]}}{\partial \boldsymbol{a}^{[0]}} = \left(W^{[1]}\right)^T \frac{\partial z_c^{[L]}}{\partial \boldsymbol{z}^{[1]}} \in \mathbb{R}^{n[0]\times 1}.$$

In order to vectorize this expression, we can first see that we can vectorize the partial of $z_c^{[L]}$ with respect to any $\boldsymbol{z}^{[\ell]}$ by using Eq. (41) of the lecture notes:

$$\frac{\partial Z_c^{[L]}}{\partial Z^{[\ell]}} = \left[\frac{\partial Z_c^{[L]}}{\partial A^{[\ell]}}\right] \otimes \left(g^{[\ell]}\right)'\left(Z^{[\ell]}\right).$$

Likewise, we can vectorize the partial of $z_c^{[L]}$ with respect to $\boldsymbol{a}^{[0]}$ with the following:

$$\frac{\partial Z_c^{[L]}}{\partial A^{[0]}} = \left(W^{[1]}\right)^T \frac{\partial Z_c^{[L]}}{\partial Z^{[1]}}.$$

■