CSCI 5521 Denis Rybkin
Section 2 rybki001@umn.edu

Homework 4
Due Friday, December 11

Problem 1. (30 points) Let $\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ with $\mathbf{x}^t \in \mathbb{R}^D, t = 1, \ldots, N$ be a given training set. Assume that the dataset is centered, i.e., $\sum_{t=1}^N \mathbf{x}^t = 0 \in \mathbb{R}^D$. We focus on performing linear dimensionality reduction on the dataset using PCA (principal component analysis). With PCA, for each $\mathbf{x}^t \in \mathbb{R}^D$, we get $\mathbf{z}^t = W\mathbf{x}^t$, where $\mathbf{z}^t \in \mathbb{R}^d$, $d < D$, is the low dimensional projection, and $W \in \mathbb{R}^{d\times D}$ is the PCA projection matrix. Let $\Sigma = \frac{1}{N}\sum_{t=1}^N \mathbf{x}^t(\mathbf{x}^t)^T$ be the sample covariance matrix. Further, let $\mathbf{v}^t = W^T\mathbf{z}^t$ so that $\mathbf{v}^t \in \mathbb{R}^D$.

(a) Professor HighLowHigh's statement is NOT correct.

Consider

$$\mathbf{z}^t = W\mathbf{x}^t \qquad \text{and} \qquad \mathbf{v}^t = W^T\mathbf{z}^t$$

We are given that the matrix W is the projection matrix such that $W \in \mathbb{R}^{d\times D},\ d < D$.

The equation

$$\mathbf{z}^t = W\mathbf{x}^t$$

is the projection from the original space to a new subspace.

The equation

$$\mathbf{v}^t = W^T\mathbf{z}^t$$

is the projection from the subspace back to the original space.

Hence, we can write it as

$$\mathbf{v}^t = W^TW\mathbf{x}^t$$

This equation implies that we can have $\mathbf{v}^t = \mathbf{x}^t$ only if W^TW is an identity matrix, i.e. $W^TW = I$. However, from the definition of identity matrices, for W^TW to be an identity matrix, W has to be an orthogonal square matrix, so we would have $W^T = W^{-1}$. But the definition of the problem states what $W \in \mathbb{R}^{d \times D},\ d < D$. Therefore, W can't be orthogonal, since it isn't a square matrix. Then it follows that $W^T W$ is not an identity matrix, i.e. $W^T W \neq I$. Hence, we have that

$$\mathbf{v}^t \neq \mathbf{x}^t$$

Thus, as we stated above, Professor HighLowHigh's statement is false.

(b) This statement is correct.

Let's consider a couple of different cases:

• If the projection matrix W is formed from all covariance matrix eigenvalues and all the eigenvalues are distinct, then the transformation $W^T W$ is an identity matrix I and hence

$$\mathbf{v}^t = \mathbf{x}^t$$

Therefore, we can rewrite

$$\sum_{t=1}^{N} ||\mathbf{x}^t||_2^2 - \sum_{t=1}^{N} ||\mathbf{v}^t||_2^2 = \sum_{t=1}^{N} ||\mathbf{x}^t - \mathbf{v}^t||_2^2$$

as

$$\sum_{t=1}^{N} ||\mathbf{x}^t||_2^2 - \sum_{t=1}^{N} ||\mathbf{x}^t||_2^2 = \sum_{t=1}^{N} ||\mathbf{x}^t - \mathbf{x}^t||_2^2 = 0$$

which is trivially true.

• If $d < D$, then we know that transformation to a lower dimensional subspace is implicit and hence some information is lost. The transformation $\mathbf{z} = W\mathbf{x}^t$ yields an orthogonal projection to the lower dimensional subspace.

The transformation back to the original space, $\mathbf{v} = W^T \mathbf{z}^t$, loses no more information from the lower dimensional subspace and therefore $||\mathbf{z}^t|| = ||\mathbf{v}^t||$.

Given that no information is lost in transformation from the lower dimensional subspace to the original space, we can imagine our problem geometrically, with $||\mathbf{x}||$ and $||\mathbf{v}||$ being two vectors (imagine that $||\mathbf{v}||$ spans along x-axis, for convenience) and $||\mathbf{x - v}||$ being a distance between their endpoints.

Observe that we can use the Pythagoras' Theorem, if we prove that $\mathbf{v}^t \perp (\mathbf{x}^t - \mathbf{v}^t)$.

Remember that if any two vectors are orthogonal, then their inner product is 0. That is,

$$(\mathbf{x}^t - \mathbf{v}^t) \cdot \mathbf{v}^t = 0$$

We know that $\mathbf{v}^t = W^T W \mathbf{x}^t$. Let's try to show that the inner product is 0 by using this fact:

$$\begin{aligned}
&(\mathbf{x}^t - \mathbf{v}^t) \cdot \mathbf{v}^t = \\
&(\mathbf{x}^t - W^T W \mathbf{x}^t) \cdot W^T W \mathbf{x}^t = \\
&\mathbf{x}^t W^T W \mathbf{x}^t - (W^T W \mathbf{x}^t)(W^T W \mathbf{x}^t) = \\
&\mathbf{x}^t W^T W \mathbf{x}^t - (\mathbf{x}^t W^T W)(W^T W \mathbf{x}^t) = \\
&\mathbf{x}^t W^T W \mathbf{x}^t - (\mathbf{x}^t W^T W W^T W \mathbf{x}^t)
\end{aligned}$$

Note that we assumed that the projection matrix W contains the eigenvectors of Σ associated with unique eigenvalues. It means that the columns of W are orthonormal and hence $WW^T = I$. Then we continue:

$$\begin{aligned}
&\mathbf{x}^t W^T W \mathbf{x}^t - (\mathbf{x}^t W^T W W^T W \mathbf{x}^t) = \\
&\mathbf{x}^t W^T W \mathbf{x}^t - (\mathbf{x}^t W^T I \; W \mathbf{x}^t) = \\
&\mathbf{x}^t W^T W \mathbf{x}^t - \mathbf{x}^t W^T W \mathbf{x}^t = 0
\end{aligned}$$

Hence, it follows that $(\mathbf{x}^t - \mathbf{v}^t) \cdot \mathbf{v}^t = 0$ and therefore $\mathbf{v}^t \perp (\mathbf{x}^t - \mathbf{v}^t)$.

Then, by the Pythagoras' Theorem, we get:

$$||\mathbf{x}^t||_2^2 = ||\mathbf{v}^t||_2^2 + ||\mathbf{x}^t - \mathbf{v}^t||_2^2$$

which we can rearrange to get:

$$||\mathbf{x}^t||_2^2 - ||\mathbf{v}^t||_2^2 = ||\mathbf{x}^t - \mathbf{v}^t||_2^2$$

Therefore, for our case, we get

$$\sum_{t=1}^{N} ||\mathbf{x}^t||_2^2 - \sum_{t=1}^{N} ||\mathbf{v}^t||_2^2 = \sum_{t=1}^{N} ||\mathbf{x}^t - \mathbf{v}^t||_2^2$$

(trivially true, since if it's true for a vector, it is also true for sum of vectors)

Therefore, Professor HighLowHigh's statement is indeed correct.

Problem 2. (30 points) Let $\mathcal{Z} = \{(\mathbf{x}^1, \mathbf{r}^1), \ldots, (\mathbf{x}^N, \mathbf{r}^N)\}, \mathbf{x}^t \in \mathbb{R}^d, \mathbf{r}^t \in \mathbb{R}^k$ be a set of N training samples. We consider training a multilayer perceptron as shown in Figure 1 of Homework 4. We consider a general setting where the transfer functions at each stage are denoted by g, i.e.,

$$z_h^t = g(a_h^t) = g\left(\sum_{j=1}^{d} w_{h,j} x_j^t + w_0\right) \quad \text{and} \quad y_i^t = g(a_i^t) = g\left(\sum_{h=1}^{H} v_{i,h} z_h^t + v_{i0}\right)$$

where a_h^t, a_i^t respectively denote the input activation for hidden node h and output node i. Further, let $L(\cdot, \cdot)$ be the loss function, so that the learning focuses on minimizing:

$$E(W, V|\mathcal{Z}) = \sum_{t=1}^{N} \sum_{i=1}^{k} L(r_i^t y_i^t).$$

(a) We know that the gradient descent update rule for $\Delta v_{i,h}$ can be expressed as

$$\Delta v_{i,h} = -\eta \frac{\partial E}{\partial v_{i,h}}$$

Since we only need to update one weight, $v_{i,h}$, and we are using stochastic gradient descent, we can remove the summations over k and N, and rewrite the loss function that we need to evaluate in the following form:

$$E(W, V|Z) = L(r_i^t, y_i^t) = L(r_i^t, g(a_i^t)) = L(r_i^t, g(\sum_{h=1}^{H} v_{i,h} z_h^t + v_{i0}))$$

According to our book, we need to evaluate the following chain rule:

$$\frac{\partial E}{\partial v_{i,h}} = \frac{\partial E}{\partial y_i^t} \frac{\partial y_i^t}{\partial v_{i,h}}$$

However, we have an activation function, $g(a_i^t)$. Then we can account for it with:

$$\frac{\partial E}{\partial v_{i,h}} = \frac{\partial E}{\partial y_i^t} \frac{\partial y_i^t}{\partial a_i^t} \frac{\partial a_i^t}{\partial v_{i,h}}$$

Now, let's apply the chain rule:

$$\frac{\partial E}{\partial y_i^t} = \frac{\partial L(r_i^t, y_i^t)}{\partial y_i^t}$$

$$\frac{\partial y_i^t}{\partial a_i^t} = g^{'}(a_i^t)$$

$$\frac{\partial a_i^t}{\partial v_{i,h}} = \frac{\partial}{\partial v_{i,h}} (\sum_{h=1}^{H} v_{i,h} z_h^t + v_{i0}) = z_h^t$$

Then we know that the update rule is a product of learning rate η, error Δ_i^t and input z_h^t:

$$\Delta v_{i,h} = \eta \Delta_i^t z_h^t$$

Where our error is

$$\Delta_i^t = g^{'}(a_i^t)\bigg(- \frac{\partial L(r_i^t, y_i^t)}{\partial y_i^t}\bigg)$$

Hence, we showed that the stochastic gradient descent update for $v_{i,h}$ is of the form $v_{i,h}^{new} = v_{i,h}^{old} + \Delta v_{i,h}$ with the update

$$\Delta v_{i,h} = \eta \Delta_i^t z_h^t, \qquad \text{where} \qquad \Delta_i^t = g^{'}(a_i^t)\bigg(- \frac{\partial L(r_i^t, y_i^t)}{\partial y_i^t}\bigg)$$

(b) In the book, we are given that the update rule for $w_{h,j}$ is:

$$\frac{\partial E}{\partial w_{h,j}} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial z_h} \frac{\partial z_h}{\partial w_{h,j}}$$

However, we have to account for two activation functions, $g(a_i^t)$ and $g(a_h^t)$. Then, using the information provided to us, we get:

$$\frac{\partial E}{\partial w_{h,j}} = \frac{\partial E}{\partial y_i^t} \frac{\partial y_i^t}{\partial a_i^t} \frac{\partial a_i^t}{\partial z_h} \frac{\partial z_h}{\partial a_h^t} \frac{\partial a_h^t}{\partial w_{h,j}}$$

Now we need to take the partial derivative of a_i^t with respect to z_h. Therefore, we get:

$$\frac{\partial}{z_h}(\sum_{h=1}^{H} v_{i,h} z_h^t + v_{i0}) = v_{i,h}$$

Also observe that the error from the hidden layer to the output layer is propagated to that layer. It can be observed in the following terms:

$$\Delta_i^t = \frac{\partial E}{\partial y_i^t}\frac{\partial y_i^t}{\partial a_i^t}$$

Note that if there are k output nodes, then we also have k weights $(v_{i,h})$ and we need to account for error for each of them. We can simply do it with:

$$\sum_{i=1}^{k} \Delta_i^t v_{i,h}$$

Similarly to what we did above, we will also get the rest of the terms in the chain rule as follows:

$$\frac{\partial z_h}{\partial a_h^t} = g^{'}(a_h^t)$$

and also

$$\frac{\partial a_h^t}{\partial w_{h,j}} = \frac{\partial}{w_{h,j}}(\sum_{j=1}^{d} w_{h,j} x_j^t + w_0) = x_j^t$$

Then we have the update rule as the product of learning rate η, error Δ_h^t and input x_j^t:

$$\Delta w_{h,j} = \eta \Delta_h^t x_j^t$$

Where we have the error Δ_h^t:

$$\Delta_h^t = g^{'}(a_h^t)\left(\sum_{i=1}^{k} \Delta_i^t v_{i,h}\right)$$

Hence, we showed that the stochastic gradient descent update for $w_{h,j}$ is of the form $w_{h,j}^{new} = w_{h,j}^{old} + \Delta w_{h,j}$ with the update

$$\Delta w_{h,j} = \eta \Delta_h^t x_j^t, \qquad \text{where} \qquad \Delta_h^t = g^{'}(a_h^t)\left(\sum_{i=1}^{k} \Delta_i^t v_{i,h}\right)$$

Problem 3. (40 points)

Description: For this problem, we are using 2-class linear SVMs with parameters $(\mathbf{w}, w_0)$ where $\mathbf{w} \in \mathbb{R}^d, w_0 \in \mathbb{R}$. The implementation is quite similar to Linear Regression from previous homework, except we are using the new objective function given in the homework write-up and this time we implemented the mini-batch gradient descent. We are using Boston50 and Boston25 datasets, similarly to previous homeworks. One major change is that now we are using 1 and -1 for labels (instead of 0, 1).

Results:

Error rates for MySVM2 with m = 40 for Boston50						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.2255	0.1188	0.1089	0.1584	0.1584	0.1540	0.0410

Error rates for MySVM2 with m = 200 for Boston50						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.1863	0.1386	0.1782	0.1485	0.1485	0.1600	0.0187

Error rates for MySVM2 with m = n for Boston50						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.1471	0.1584	0.1386	0.1386	0.1683	0.1502	0.0116

Error rates for LogisticRegression for Boston50						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.1863	0.1089	0.1980	0.1089	0.1485	0.1501	0.0374

Error rates for MySVM2 with m = 40 for Boston25						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.1765	0.1683	0.1188	0.0891	0.0792	0.1264	0.0398

Error rates for MySVM2 with m = 200 for Boston25						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.1275	0.0990	0.0693	0.1386	0.1287	0.1126	0.0254

Error rates for MySVM2 with m = n for Boston25						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.0980	0.0792	0.1386	0.1485	0.0792	0.1087	0.0294

Error rates for LogisticRegression for Boston25						
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	SD
0.1275	0.0891	0.0990	0.1287	0.0990	0.1087	0.0163

Extra Credit Problem: Consider Problem 2 with specific choices of the activation function $g(a)$. We will assume $L(r_i^t, y_i^t) = (r_i^t - y_i^t)^2$.

(a) We know that the gradient descent update rule for $\Delta v_{i,h}$ can be expressed as

$$\Delta v_{i,h} = -\eta \frac{\partial E}{\partial v_{i,h}}$$

Since we only need to update one weight, $v_{i,h}$, and we are using stochastic gradient descent, we can remove the summations over k and N, and rewrite the loss function that we need to evaluate in the following form:

$$E(W, V|Z) = L(r_i^t, y_i^t) = (r_i^t - y_i^t)^2 =$$

$$(r_i^t - g(a_i^t))^2 = (r_i^t - \max(0, a_i^t))^2$$

According to our book, we need to evaluate the following chain rule:

$$\frac{\partial E}{\partial v_{i,h}} = \frac{\partial E}{\partial y_i^t} \frac{\partial y_i^t}{\partial v_{i,h}}$$

However, we have an activation function, $g(a_i^t)$. Then we can account for it with:

$$\frac{\partial E}{\partial v_{i,h}} = \frac{\partial E}{\partial y_i^t}\frac{\partial y_i^t}{\partial a_i^t}\frac{\partial a_i^t}{\partial v_{i,h}}$$

Now, let's apply the chain rule:

$$\frac{\partial E}{\partial y_i^t} = \frac{\partial L(r_i^t, y_i^t)}{\partial y_i^t} = \frac{\partial (r_i^t - y_i^t)^2}{\partial y_i^t} = 2y_i^t - 2r_i^t$$
$$\frac{\partial y_i^t}{\partial a_i^t} = g^{'}(a_i^t) = a_i^{t'} \text{ if } a_i^t > 0, \text{ otherwise } 0$$
$$\frac{\partial a_i^t}{\partial v_{i,h}} = \frac{\partial}{\partial v_{i,h}}(\sum_{h=1}^{H} v_{i,h}z_h^t + v_{i0}) = z_h^t$$

Then we know that the update rule is a product of learning rate η, error Δ_i^t and input z_h^t:

$$\Delta v_{i,h} = \eta \Delta_i^t z_h^t$$

Where our error is

$$\Delta_i^t = g^{'}(a_i^t)(2r_i^t - 2y_i^t)$$

Note that both our error rate and our update rule is 0 if $a_i^t \leq 0$.

Otherwise, we have the error rate

$$\Delta_i^t = z_h^t(2r_i^t - 2y_i^t)$$

And our update rule is

$$\Delta v_{i,h} = 2\eta(z_h^t)^2(r_i^t - y_i^t)$$

Hence, we showed that the stochastic gradient descent update with specific $g(\cdot)$ and $L(\cdot,\cdot)$ for $v_{i,h}$ is of the form $v_{i,h}^{new} = v_{i,h}^{old} + \Delta v_{i,h}$ with the update

$$\Delta v_{i,h} = \eta \Delta_i^t z_h^t, \qquad \text{where} \qquad \Delta_i^t = g^{'}(a_i^t)(2r_i^t - 2y_i^t)$$

where both of them are zero when $a_i^t \leq 0$ (otherwise, it has an equation specified before the conclusion).

(b) Consider

$$g(a) = \max(0, a) + \alpha \min(0, a)$$

Then we have

$$g(a) = \begin{cases} a \text{ if } a > 0 \\ \alpha \cdot a \text{ if } a < 0 \\ 0 \text{ if } a = 0 \end{cases}$$

Now we take the derivative of each of them and get:

$$g(a) = \begin{cases} a^{'} \text{ if } a > 0 \\ \alpha \cdot a^{'} \text{ if } a < 0 \\ 0 \text{ if } a = 0 \end{cases}$$

As we can see, first and third case is not dependent on α. For the second case, however, we get:

$$\frac{\partial a_i^t}{\partial v_{i,h}} = \frac{\partial}{\partial v_{i,h}} \alpha (\sum_{h=1}^{H} v_{i,h} z_h^t + v_{i0}) = \alpha z_h^t$$

(c) Consider

$$g(a) = \max(0, a) + \alpha \min(0, a)$$

We are given that $\alpha \in [0, 1]$. If we choose $\alpha = 1$, then the function above can be simplified to

$$g(a) = \max(0, a) + \min(0, a) = a$$

Then, from the definition of Problem 2, we know that two layer perceptron is linear, since a denote the input activation.

Hence, there indeed exists a specific choice of $\alpha \in [0, 1]$ which makes the two layer perceptron a linear model.