Name: Amirmohammad Pirhosseinloo **99131006**
machine learning course **HW1**

Problem 1

definitions:

(1) supervised learning: refers to a class of algorithms that determine a predictive model using data points with known outcomes.

(2) semi-supervised learning: Semi-supervised learning stands somewhere between the supervised learning and unsupervised. It solves classification problems, which means you'll ultimately need a supervised learning algorithm for the task. But at the same time, you want to train your model without labeling every single training example, for which you'll get help from unsupervised machine learning techniques.

(3) unsupervised learning: Unsupervised learning refers to the use of algorithms to identify patterns in data sets containing data points that are not labeled.

(4) reinforcement learning: it is learning by interacting with an environment. An RL agent learns from the consequences of its actions, rather than from being explicitly taught and it selects its actions on basis of its past experiences (exploitation) and also by new choices (exploration).

(5) transfer learning: Transfer learning is the reuse of a pre-trained model on a new problem.

(6) classification: Classification in machine learning is when using an algorithm to draw conclusions from data that it already has, and then uses these conclusions to categorise new data it receives.

(7) regression: Regression consists of a set of machine learning methods that allow us to predict a continuous outcome variable (y) based on the value of one or multiple predictor variables (x).

(8) online learning: Online machine learning includes methods to create machine learning models using data which becomes available sequentially in time.

(9) overfitting: Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model.

(10) active learning: Active learning is a special case of machine learning in which a learning algorithm can interactively query a user (or some other information source) to label new data points with the desired outputs.

(11) correlation: Correlation is a bivariate analysis that measures the strength of association between two variables and the direction of the relationship. In terms of the strength of relationship, the value of the correlation coefficient varies between +1 and -1. A value of

± 1 indicates a perfect degree of association between the two variables. As the correlation coefficient value goes towards 0, the relationship between the two variables will be weaker. The direction of the relationship is indicated by the sign of the coefficient; a + sign indicates a positive relationship and a – sign indicates a negative relationship.

(12) independence:Two events A and B are statistical independent if and only if their joint probability can be factorized into their marginal probabilities, i.e., P($A \cap B$) = $P(A)P(B)$

Problem 2

(a) Whats the effect of increasing the size of training dataset on model's bias and variance?
answer: It doesn't have any effect on lowering the bias. To achieve such a goal, its better to change the model (algorithm). However, it'll help in decreasing the variance in a high variance model since it helps in generalization of model.

(b) Describe 4 was to prevent overfitting.
answer:
Use more data to train as described in the previous section.
Reduce model complexity by removing some features (parameters).
Use regularization techniques to eliminate the effect of higher parameters in the model.
Use Early stopping method to choose the best value for #iterations.

Problem 3

Consider two datasets both sampled from the same distribution. There are 2000 samples in the first one and 100000 in the second one. Two models are trained from the datasets with 70 percent of dataset as the the training data and the rest as the test one. Compare training and testing costs of two models to each other.
answer:
This is the figure comparing the costs

Problem 4

Define MAE, MSE, and RMSE then explain their use-cases.

1. MAE (Mean Absolute Error): represents the average of the absolute difference between the actual and predicted values in the dataset.
$$\frac{1}{M}\sum|\hat{y} - y_i|$$
its better to use MAE when you have very few or no outliers in the dataset or in a better way when you want to ignore the outliers while fitting your model to your data.

2. MSE (Mean Square Error): represents the squared difference between the actual and predicted values in the dataset.
$$\frac{1}{M}\sum(\hat{y} - y_i)^2$$

3. RMSE (Root Mean Square Error): the square root of MSE.
$$\sqrt{\frac{1}{M}\sum(\hat{y} - y_i)^2}$$

you can use MSE or RMSE when you have a large number of outliers in your data and want to accommodate them while fitting your model.

Problem 5

What is the momentum effect in gradient descent algorithm? What are its benefits? What is the impact of lower momentum and the higher momentum?
answer

Momentum is a method which helps accelerate gradients vectors in the right directions, thus leading to faster converging.

A problem with the gradient descent algorithm is that the progression of the search can bounce around the search space based on the gradient. For example, the search may progress downhill towards the minima, but during this progression, it may move in another direction, even uphill, depending on the gradient of specific points (sets of parameters) encountered during the search.

This can slow down the progress of the search, especially for those optimization problems where the broader trend or shape of the search space is more useful than specific gradients along the way.

One approach to this problem is to add history to the parameter update equation based on the gradient encountered in the previous updates.

In gradient descent, the weights are updated by:

$$change(x) = step\_size * f'(x)$$
$$x = x - change(x)$$

While in momentum gradient descent:

$$change(x,t) = step\_size * f'(x(t-1)) + momentum * change(x,t-1)$$
$$x(t) = x(t-1) - change(x,t)$$

For example, a large momentum (e.g. 0.9) will mean that the update is strongly influenced by the previous update, whereas a modest momentum (0.2) will mean very little influence.

Problem 6

Assume that there is a dataset with n samples, each sample is d dimensional, and SSE equals to: $J(w) = \sum (y_{\mathrm{i}} - w^{\mathrm{T}} x_{\mathrm{i}})^2$

1. Prove that $\hat{w}$ which results in minimum SSE in linear regression equals to: $(X^{\mathrm{T}}X)^{\text{-}1}X^{\mathrm{T}}Y$
   answer:
   common matrix derivative formulas:

$$\frac{\partial(B^{\mathrm{T}}AB)}{\partial B} = AB + A^{\mathrm{T}}B$$
$$\frac{\partial(AB)}{\partial B} = A^{\mathrm{T}}$$

$J(w) = (Xw - Y)^{\mathrm{T}}(Xw - Y)$
$= ((Xw)^{\mathrm{T}} - Y^{\mathrm{T}})(Xw - Y)$
$= (Xw)^{\mathrm{T}}Xw - (Xw)^{\mathrm{T}}Y - Y^{\mathrm{T}}Xw + Y^{\mathrm{T}}Y$
$= w^{\mathrm{T}}X^{\mathrm{T}}Xw - (Y^{\mathrm{T}}Xw)^{\mathrm{T}} - Y^{\mathrm{T}}Xw + Y^{\mathrm{T}}Y$
$= w^{\mathrm{T}}X^{\mathrm{T}}Xw - 2Y^{\mathrm{T}}Xw + Y^{\mathrm{T}}Y$
$\frac{\partial J(w)}{\partial w} = X^{\mathrm{T}}Xw + X^{\mathrm{T}}Xw - 2X^{\mathrm{T}}Y = 2X^{\mathrm{T}}Xw - 2X^{\mathrm{T}}Y$
$\frac{\partial J(w)}{\partial w} = 0 \Rightarrow X^{\mathrm{T}}Xw = X^{\mathrm{T}}Y \Rightarrow w = (X^{\mathrm{T}}X)^{\text{-}1}X^{\mathrm{T}}Y$

2. Explain two problems of using the above formula and suggest the solutions to fix them.
   a) $X^{\mathrm{T}}X$ might be non invertible. Since all of the cells of $X^{\mathrm{T}}X$ are positive, the matrix is positive semi definite, so we can always obtain a non singular (invertible) matrix from $X^{\mathrm{T}}X$ and use it to find $\hat{w}$.
   b) Its too computationally expensive to compute inverse of a matrix ($o(n^2 log_2 n)$).

3. What happens if one of the columns of X is a linear combination of the others?
   answer:
   Display matrix X as:
   $$\begin{bmatrix} x_1 & x_2 & \dots & x_{\mathrm{n}} \end{bmatrix}$$
   Assume that column x_{n} is a linear combination of the other columns (It doesn't matter whether to select x_{n} or any other column as the linear combination of the others, its just for simplicity). We can compute $X^{\mathrm{T}}X$ as follows:
   $x_n = a_1 x_1 + a_2 x_2 + \dots + a_{n-1} x_{n-1}$
   $$X^T = \begin{bmatrix} x_1^T \\ x_2^T \\ \dots \\ x_n^T \end{bmatrix}$$
   $$X^T X = \begin{bmatrix} x_1^T x_1 & x_1^T x_2 & \dots & x_1^T x_n \\ x_2^T x_1 & x_2^T x_2 & \dots & x_2^T x_n \\ \dots & \dots & \dots & \dots \\ x_n^T x_1 & x_n^T x_2 & \dots & x_n^T x_n \end{bmatrix}$$
   extending last column of $X^T X$: $\begin{bmatrix} a_1 x_1^T x_1 & a_2 x_1^T x_2 & \dots & a_{n-1} x_1^T x_{n-1} \\ a_1 x_2^T x_1 & a_2 x_2^T x_2 & \dots & a_{n-1} x_2^T x_{n-1} \\ \dots & & & \\ a_1 x_n^T x_1 & a_2 x_n^T x_2 & \dots & a_{n-1} x_n^T x_{n-1} \end{bmatrix}$
   We can conclude that the last column of $X^T X$ is the linear combination of other columns of $X^T X$. Also a matrix with linear dependent columns is non invertible, so $X^T X$ is non invertible, therefore, its not possible to compute $\hat{w}$ in this case. To solve the problem, we can simply eliminate the column x_n since it doesn't provide any special information to our model. Such columns can be eliminated in feature extraction phase.

4. Compute the closed form of $\hat{w}$ if $J(w) = \sum (y_{\mathrm{i}} - w^{\mathrm{T}} x_{\mathrm{i}})^2 + ||w||^2$. Also explain the benefits of adding $||w||^2$.
   answer:
   common matrix derivative formulas:
   $$\frac{\partial (B^{\mathrm{T}} B)}{\partial B} = 2B$$
   $J(w) = \sum (y_{\mathrm{i}} - w^{\mathrm{T}} x_{\mathrm{i}})^2 + w^{\mathrm{T}} w$
   $\frac{\partial J(w)}{\partial w} = 2X^{\mathrm{T}} X w - 2X^{\mathrm{T}} Y + 2w$
   $\frac{\partial J(w)}{\partial w} = 0 \Rightarrow w = (X^{\mathrm{T}} X + I)^{-1} X^{\mathrm{T}} Y$
   Adding such a term to the formula makes the $X^T X + I$ invertible and we able to compute $\hat{w}$.

5. Compute closed form of $\hat{w}$ if $J(w) = \sum F_{\mathrm{i}} (y_{\mathrm{i}} - w^{\mathrm{T}} x_{\mathrm{i}})^2$. (weighted linear regression)
   answer:
   common matrix derivative formulas:
   $$\frac{\partial (B^{\mathrm{T}} A B)}{\partial B} = AB + A^{\mathrm{T}} B$$

$$\frac{\partial(AB)}{\partial B} = A^{\mathrm{T}}$$

F_{i} coefficients are demonstrated in a matrix such as below:

$$F = \begin{bmatrix} F_1 & 0 & 0 & ... & 0 & 0 \\ 0 & F_2 & 0 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & F_{n-1} & 0 \\ 0 & 0 & 0 & ... & 0 & F_n \end{bmatrix}$$

Its obvious that $F^T = F$

Now follow the steps:

$J(w) = (Xw - Y)^{\mathrm{T}}F(Xw - Y)$

$= ((Xw)^{\mathrm{T}}F - Y^{\mathrm{T}}F)(Xw - Y)$

$= (Xw)^{\mathrm{T}}FXw - (Xw)F^{\mathrm{T}}Y - Y^{\mathrm{T}}FXw + Y^{\mathrm{T}}FY$

$= w^{\mathrm{T}}X^{\mathrm{T}}FXw - Y^{\mathrm{T}}FXw + Y^{\mathrm{T}}FY$

$\frac{\partial J(w)}{\partial w} = X^{\mathrm{T}}FXw + X^{\mathrm{T}}FXw - 2X^{\mathrm{T}}FY = 2X^{\mathrm{T}}FXw - 2X^{\mathrm{T}}FY$

$\frac{\partial J(w)}{\partial w} = 0 \Rightarrow X^{\mathrm{T}}FXw = X^{\mathrm{T}}FY \Rightarrow w = (X^{\mathrm{T}}FX)^{\text{-}1}X^{\mathrm{T}}FY$

Problem 7

Assume that $y = w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$.

(a) Find an equation for $p(y|x_1, x_2)$.
answer:
$\prod p(y_i|x_1^i, x_2^i; w_0, w_1, w_2, w_3, s^2) = \prod \frac{1}{\sqrt{2\pi s^2}} e^{\frac{(y_i - (w_0 + w_1x_1^i + w_2x_2^i + w_3(x_1^i)^2))^2}{2s^2}}$

(b) Find the log likelihood for training dataset.
answer:
$L(w_0, w_1, w_2, w_3, s^2) = log \prod p(y_i|x_1^i, x_2^i; w_0, w_1, w_2, w_3, s^2)$
$= \sum log(p(y_i|x_1^i, x_2^i; w_0, w_1, w_2, w_3, s^2))$
$= -\frac{n}{2}log(2\pi) - nlog(s) - \frac{1}{2s^2}\sum (y_i - (w_0 + w_1x_1^i + w_2x_2^i + w_3(x_1^i)^2))^2$

(c) Find $f(w_0, w_1, w_2, w_3)$ such that the best fit is calculated by minimizing f.
answer:
$\sum (y_i - (w_0 + w_1x_1^i + w_2x_2^i + w_3(x_1^i)^2))^2$

(d) Calculate $\frac{\partial f}{\partial w}$.
answer:
$\frac{\partial f}{\partial w_0} = 2\sum (y_i - (w_0 + w_1x_1^i + w_2x_2^i + w_3(x_1^i)^2))$
$\frac{\partial f}{\partial w_1} = 2\sum (y_i - (w_0 + w_1x_1^i + w_2x_2^i + w_3(x_1^i)^2))x_1^i$
$\frac{\partial f}{\partial w_2} = 2\sum (y_i - (w_0 + w_1x_1^i + w_2x_2^i + w_3(x_1^i)^2))x_2^i$
$\frac{\partial f}{\partial w_3} = 2\sum (y_i - (w_0 + w_1x_1^i + w_2x_2^i + w_3(x_1^i)^2))(x_1^i)^2$