Yaowen Mei
1177855

Homework 1

ENGG 6500
October 16, 2021

Theory 1

- **Q1: Define Machine Learning and name four types of problems where it excels**.
  Machine learning is a subset of artificial intelligent. Where training data and computing algorithms are used to train computer models to mimic the way how human learns. These models are build on "training data", and after trained, these models will be able to make classification or prediction decision for input tasks which were not explicitly seen before.

  Machine learning excels in many scenarios, such as 1) medical image analysis (cancer detection), 2) financial risk management (credit card fraud detection), 3) bioinformatic research (protein folding prediction), 4) computer vision (service-robots path planing)

- **Q2: What is happening if your model performs great on the training data but generalizes poorly to new instances? Name three possible solutions to this problem.**
  The reason why a model performs excellent on training data while poor on new instance is that the model over fits the training data. The possible solutions includes: 1) filter the training data to remove outliers and noise points, 2) optimize/simplify the structure of the model (reduce hidden layers and neurons on hidden layers), and 3) use early-stopping techniques to provide guidance to the model so that it get trained before over-fitting happens.

- **Q3: What is a test set and a validation set and what are their usage and purpose.**
  A test set is a subset of your data that is **ONLY** used to test/evaluate the performance of the fully trained machine learning model. One should **NEVER** expose his test data to the machine learning model during the training process.

  A validation set is another subset of your data that is used to compare between different training models, so that the model's parameters can be tuned during the training process to obtain a better performance.

- **Q4: Let us say we are building an OCR and for each character, we store the bitmap of that character as a template that we match with the read character pixel by pixel. Explain when such a system would fail. Why do you think are bar code readers still used?**
  This system will have performance issue with even slightly change on the input (i.e. different font, rotation, shift of the input image). This system can storing only ONE template for each of the characters that it intends to recognize (like a hashmap), but it will be easily confused by different fonts. The reason why people are still using bar code is that 1) bar code reader is cheaper than an optical OCR device, 2) generally,

bar code reader is much faster and more accurate than OCR due to the reason stated above.

- **Q5: If a face image is a 100x100 pixel image, written in row-major, this is a 10,000- dimensional vector. If we shift the image one pixel to the right, this will be a very different vector in the 10,000-dimensional space. How can we build face recognizes robust to such distortions?**
  We can use convolutional neural network to build the face recongnization model, as the CNN can automatically detects the most important features (such as the position of eyes, and the relative postion between ears and eyes) without any supervision from human.
  Or inorder to overcome this distortion, we can prepossessing the images before feed into the model, such as getting the ratio between the distance between two eyes to the size of mouth, or the relative position between the center of ears to the center of two eyes, and then use this processed feature to train the model. In this way, the model will be more robust to the size change of the image, the position change of the object.

- **Q6: In estimating the price of a used car, it makes more sense to estimate the percent depreciation over the original price than to estimate the absolute price. Why?**

  The estimations of the price of used car is a regression problem, and the inputs are car attributes, such as year, color, brand, engine type, and mileage. Instead of estimating the absolute value, people prefer to estimate the percentage depreciation. The reason is that some input attributes are affecting the absolute price a lot, but does not significant affecting the percentage price depreciation. Used car customers are selling or buying based on percentage depreciation, but not on absolute value.

  For example, a used car dealer tell a customer that after 5 years, a V8 engine Audit S5's price depreciated by \$30K, while, after 5 years, a V6 engine Audit S5's price depreciated by \$25K, is not as efficient/direct as told them in general, Audit S5 price depreciate 50% over 5 years.

- **Q7: In light of the training dataset size, when is a simpler model better and when is a complex model better?**
  As a general rule, we use small size dataset for simpler model, and use big size dataset for complex model.

  This is to say, as we get more data, we build more complex models, but up to a point:
  1) If the training dataset size is small, we can use some technical skills, such as cross validate, to increase the accuracy of the training process, and we prefer a simple model.
  2) Then, as dataset size increase, we will be able to handle more complex models.
  3) However, if the dataset gets extremely big, then we can not have the complexity of

the model continue increase together with the dataset size, as in that case, the model will be overfitting, and will have low performance on real-world data input.

- **Q8: What is the reason to have different train and test accuracies? What should you do if while experimenting in the real world and you find a substantial difference between the train and test accuracies?**
  The accuracies of a model is defined as the number of correct classifications over the total number of classifications. The training accuracy is the accuracy of the model on data it was constructed on, while, the test accuracy is the accuracy of the model on data it never seen before. The reason for testing accuracy different from training accuracy is that the model is overfitting, which means the model only works well specifically on the training set, but not generalized well beyond the training set.

  In the real world, if I found a significant difference between training and testing accuracies, 1) I will check the data distribution between my training data and testing data, sometimes we do get two significant different distribution between training and testing, in this case, we will need to do a "better" random shuffle, and re-separate the data into training, validation, and testing with proper percentage (i.e. 6:2:2, or 7:2:1), 2) I should also try to increase the total number of my sample points so that the sample is generalized enough to represent the whole population, 3) I can also try to optimize my model by reducing the number of hidden layers and reducing the number of neurons on each hidden layer to prevent overfitting.

- **Q9. In the KNN algorithm, explain how to select a K value that results a high performance of the classifier.**
  From previous years' Prof Ahmed's ENGG6500 Machine Learning Course Note: The K value must be selected by experiment. To choose a good K value, we need follow the following steps:

  1. start with k = 1, and use test set to evaluated the test rate
  2. repeat with k = k+2
  3. choose the k value so that the model has the minimum error rate
  4. keep in mind that k must be odd number to avoid tie

- **Q10. Does the Decision Tree Algorithm guaranteed to terminate? Explain your reasoning**
  Decision Tree does not always terminate.

  According to the course note, the Decision Tree splitting procedure stops when either of these following two conditions are met: 1) Stop splitting if all the records belong to one class 2) Sometimes, people conduct early termination by stop splitting when the percentage of one class at the node is greater than a threshold. This is also called

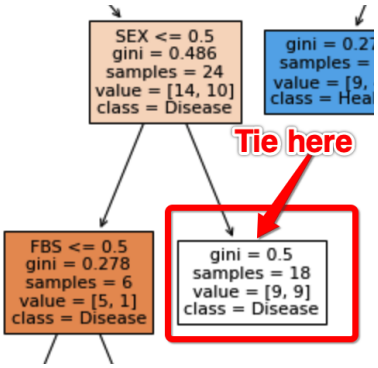

Figure 1: A sample picture of a not terminate leafnode of a decision tree. This example is a live example from the coding question-2 of this assignment. This node has 9 health people and 9 sick people, with gini index = 0.5, decision tree does not know how to classify this node so it is showing white color for this node.

pruning.

However, for a noise data, if there are only few features to split, but a lot of category (leaf nodes) need to be classified, such as only use Gender feature (Male or Female, one features) to classify a person's blood test (A, AB, B, O, 4 classes), this decision tree might not be able to terminate (Say, in the training data, if 50% of female are with blood type AB, and the other 50% of female are with blood type O, then the decision tree can not move any future, and therefore, can not terminate). I have also included another example from the coding part (heart dataset) of this assignment, as can be seen from Figure 1, in the training set, 9 healthy person and 9 sick person are classified into this leafnode, but because there are 50% health people and 50% sick people in this leafnode, decision tree can not be terminated and the model does not know what this leaf node should be classified to.

Math 2

Question-1 Gradient of a Linear-Quadratic Function: Let $f(x) = \frac{1}{2}x^T Ax + b^T x$, where A is symmetry and b is a n dimensional vector on $\mathbb{R}$, What is $\nabla f(x)$?

Solution-1: Step-1: Consider a linear function of the form:

$$f(x) = b^T x$$

We can derive the gradient in matrix notation as the following:

$$f(x) = b^T x = \sum_j^n b_j x_j$$

Then, take the partial derivative with respect to a generic element k:

$$\frac{\partial}{\partial x_k} f(x) = \frac{\partial}{\partial x_k}\left[\sum_j^n b_j x_j\right] = \sum_j^n b_j \delta_{jk} = b_k$$

Expend the generic partial derivative into a vector form

$$\nabla f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \dots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix} = \begin{bmatrix} b_1 \\ \dots \\ b_n \end{bmatrix} = b$$

So, our first conclusion is for a LINEAR multi-variable function:

$$\nabla f(x) = \nabla b^T x = b \tag{1}$$

Step-2: Now, we consider a quadratic function of the form:

$$f(x) = x^T A x$$

We can derive the gradient in matrix notation as the following:

$$f(x) = x^T \begin{bmatrix} \sum_j^n a_{1j} x_j \\ \dots \\ \sum_j^n a_{nj} x_j \end{bmatrix} = \sum_i^n \sum_j^n x_i a_{ij} x_j$$

Then, take the partial derivative with respect to a generic element k:

$$\frac{\partial}{\partial x_k} f(x) = \frac{\partial}{\partial x_k} \sum_i^n \sum_j^n x_i a_{ij} x_j = \sum_i^n \sum_j^n \frac{\partial (x_i a_{ij})}{\partial x_k} x_j + \sum_i^n \sum_j^n x_i \frac{\partial (a_{ij} x_j)}{\partial x_k} = \sum_j^n a_{kj} x_j + \sum_i^n x_i a_{ik}$$

Given that A is symetry, so $a_{ij} = a_{ji}$:

$$\frac{\partial}{\partial x_k} f(x) = \sum_j^n a_{kj} x_j + \sum_i^n x_i a_{ik} = \sum_i^n a_{ki} x_i + \sum_i^n x_i a_{ik} = \sum_i^n a_{ki} x_i + \sum_i^n x_i a_{ki} = 2\sum_i^n a_{ki} x_i$$

Expend the generic partial derivative into a vector form

$$\nabla f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \dots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix} = \begin{bmatrix} 2\sum_i^n a_{1i} x_i \\ \dots \\ 2\sum_i^n a_{ni} x_i \end{bmatrix} = 2Ax$$

So, our second conclusion is for a quadratic multi-variable function:

$$\nabla f(x) = \nabla\left(x^T A x\right) = 2Ax \tag{2}$$

Therefore, the solution for the first question is:

$$\begin{aligned}\nabla f(x) &= \nabla\left(\frac{1}{2}x^T A x + b^T x\right) \\ &= \nabla\left(\frac{1}{2}x^T A x\right) + \nabla\left(b^T x\right) \\ &= Ax + b\end{aligned} \tag{3}$$

□

Therefore, our final conclusion is for a linear-quadratic function, given that A is symmetry, we always have:

$$\boxed{\nabla f(x) = Ax + b} \tag{4}$$

Question-2 Gradient of a composition of functions: Let $f(x) = g(h(x))$, where g and h are differentiable, What is $\nabla f(x)$?

Definition 1 (Composition)**.** Consider $g : \mathbb{R} \to \mathbb{R}$, and $h : \mathbb{R}^n \to \mathbb{R}$. The **composition** of g and h is the function $g \circ h$ from $\mathbb{R}^n$ to $\mathbb{R}$ defined as:

$$f(x) = g \circ h(x) \equiv g(h(x))$$

The composition, $f : \mathbb{R}^n \to \mathbb{R}$ is a function of the form $w = f(x_1, x_2 \cdots x_n)$, where w is the only dependent variable (output) and x is the independent n dimensional variable (inputs)

Due to the fact that f has only one output and n inputs, its derivative (***Jacobian***) has n row and one columns:

$$J_f(x) = \nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

Similarly, g has only one output and one input, its derivative has one row and one column (scalar):

$$J_g(t) = \nabla g(t) = \frac{\partial g}{\partial t} = g' \tag{5}$$

h has one output and n inputs, its derivative has n row and one columns:

$$J_h(x) = \nabla h(x) = \begin{bmatrix} \frac{\partial h}{\partial x_1} \\ \vdots \\ \frac{\partial h}{\partial x_n} \end{bmatrix} = \frac{\partial h}{\partial x} \tag{6}$$

As a side note, when a function f has only one output, its derivative (Jacobian) is also called the gradient of f, which is denoted as ∇f

Applying the multi-variable Chain Rule:

$$J_f(x) = J_{g \circ h}(x) = (J_g \circ h(x)) J_h = J_g(h) J_h$$

Then, Applying equation 5 and equation 6, we have:

$$J_f(x) = \nabla (g(h(x))) = \nabla g(h(x)) \nabla h(x) = g'(h) \begin{bmatrix} \frac{\partial h}{\partial x_1} \\ \vdots \\ \frac{\partial h}{\partial x_n} \end{bmatrix} = g'(h) \frac{\partial h}{\partial x}$$

□

Therefore, our final conclusion is for this composition function, the gradient is:

$$\boxed{\nabla f(x) = \nabla g(h(x)) \nabla h(x) = g'(h) \frac{\partial h}{\partial x}} \tag{7}$$

I didn't see any necessity to include the definition of Hessian operator in the body of this question, unless, Professor is expecting a matrix re-presentation as the solution for question 1 and 2. Therefore, I am including the matrix representation for them below:

Question-1:

$$\nabla f(x) = Ax + b = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_2 \\ a_{12}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + b_2 \\ \vdots \\ a_{1n}x_n + a_{2n}x_n + \cdots + a_{nn}x_n + b_n \end{bmatrix}$$

Question-2:

$$\nabla f(x) = g'(h) \frac{\partial h}{\partial x} = \begin{bmatrix} g'(h) \frac{\partial h}{\partial x_1} \\ g'(h) \frac{\partial h}{\partial x_2} \\ \vdots \\ g'(h) \frac{\partial h}{\partial x_n} \end{bmatrix}$$
