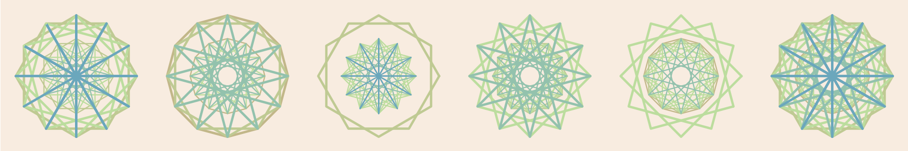
# Quiz 2

CSC316 Machine Learning
Professor Leon Tabak

25 February 2022



## Questions

1. Name the mathematical operations that are shown in these expressions:

   (a) $\vec{u} \cdot \vec{v}$

   (b) $\mathbf{M}^T$

   (c) $\frac{\partial f(x,y,z)}{\partial x}$

   (d) $\nabla f(x, y, z)$

---

(a) The **dot product** of two vectors is a number. It is the sum of the products of the corresponding elements of the two vectors.

(b) The **transpose** of a $m \times n$ matrix $\mathbf{M}$ is an $n \times m$ matrix. The rows of $\mathbf{M}$ are the columns of $\mathbf{M}^T$. The columns of $\mathbf{M}$ are the rows of $\mathbf{M}^T$.

(c) The **partial derivative** of a function is a measure of how much a small change in one of the arguments of a function changes the value of the function..

(d) The **gradient** of a function is a vector whose elements are partial derivatives of the function.

2. (a) Distinguish between utility functions (also called fitness functions) and cost functions.

   (b) Distinguish between cost functions and loss functions.

   Hint: Look up “cost functions” in the index of *Hands-On Machine Learning* for help in answering the first part of this question. Search on the Internet for the answer to the second part.

---

(a) A cost function measures how bad a model is. A utility function measures how good a model is.

We want to minimize costs. We want to maximize utilities.

(b) Many data scientists use the phrase “loss function” to mean the error associated with a single instance in the training set, and “cost function” to mean the sum or average of errors over the whole training set. However, not all make this distinction.

3. Identify several methods of the Pandas DataFrame class that you might use to explore a dataset. What do these methods show us?

---

Here are several methods of the DataFrame class that we used:

- info()
- describe()
- head()

- tail()

You can find others by searching on the Internet with phrases like "exploring data with Pandas."

4. Find documentation for the corr() method of the Pandas DataFrame class. What can we learn about our dataset through the use of this method?

---

The corr() method returns a correlation matrix.

The elements of this matrix have values in the interval $[-1.0, +1.0]$. These values are measures of linear correlation.

The values are measures of the degree to which the value of one variable in our dataset determines the value of another.

The rows and columns in the correlation matrix are both labeled with the names of the variables in the dataset.

The correlation of any variable $x$ with itself is $+1.0$. Therefore, the values on the main diagonal of a correlation matrix are all equal to $+1.0$.

The correlation of a variable $x$ with a variable $y$ is the same as the correlation of $y$ with $x$. Therefore, a correlation matrix is symmetrical—the upper right portion is a mirror image of the lower left part.

We can use a correlation matrix when constructing a linear regression model to see which variables will most strongly determine the output. We might discover that the values of some variables are only very weakly correlated with the labels. That discovery might justify a decision to drop those variables from the dataset.

5. Here is a statement that gives us another useful function:

```
from sklearn.metrics import confusion_matrix
```

In what kinds of projects might we use this function? What information will the function give us?

---

A confusion matrix contains a measure of the performance of a classifier.

We want to know (for example), when we present our model with an object that belongs to class A, how often doe the model correctly predict that the object belongs to class A? How often does it incorrectly predict that the object belongs to class B? class C? And so on.

The confusion matrix is square. Its rows and columns are labeled with the names of the classes. The labels on the rows signify actual values. The labels on the columns signify predicted values.

The confusion matrix for a perfect classifier will contain zeros everywhere except on the main diagonal.

6. What is one hot encoding?

---

If we have a categorical variable that has $n$ possible values, we can replace it with $n$ categorical variables that each have just 2 possible values (0 or 1).

For example, let's suppose that we have a categorical variable called *party* and values *Democrat*, *Republican*, *Green*, and *Libertarian*. We can drop the *party* column in our table and add new columns named *Democrat*, *Republican*, *Green*, and *Libertarian*.

In each row of the table, there will be a one in just one of the four new columns. There will be zeros in the other three columns. (Each voter belongs to just one party.)

7. Here are several classes that we saw in the first four chapters of *Hands-On Machine Learning*.

   - DecisionTreeRegressor
   - LinearRegression
   - LogisticRegression
   - RandomForestRegressor
   - SGDClassifier

   Identify two methods that we used and that are common to all of these classes.

---

Here are two methods that we used:

- fit()
- predict()

For other classes, we also used:

- transform()
- fit_transform()

8. What does this code do?

   Hint: Run the program.

```
import numpy as np
import pandas as pd

from sklearn.preprocessing import StandardScaler

# alternative to StandardScaler
from sklearn.preprocessing import MinMaxScaler

def main():

    data = np.array( [[3, 4], [5, 12], [7, 24]] )

    df = pd.DataFrame( data, columns = ['x', 'y'] )

    scaler = MinMaxScaler()
    scaler = scaler.fit( data )

    df = scaler.transform( df )

    print( df )
# end of main()

if __name__ == '__main__':
    main()
```

The program creates an array that looks like this:

$$\begin{bmatrix} 3 & 4 \\ 5 & 12 \\ 7 & 24 \end{bmatrix}$$

The program transforms the data in each column, setting the minimum value to zero, the maximum value to one, and scaling all other values proportionally.

The result is an array that looks like this:

$$\begin{bmatrix} 0.0 & 0.0 \\ 0.5 & 0.4 \\ 1.0 & 1.0 \end{bmatrix}$$

9. The invention of a better training algorithm in 1986 was a milestone in the development of neural networks.

   David Rumelhart, Geoffrey Hinton, and Ronald Williams proposed replacing the use of a step function for the activation function with this function:

   $$f(x) = \frac{1}{1 + e^{-x}}$$

   What key property did the new activation function have that the older step function did not have?

---

   The logistic function (also called the sigmoid function) is differentiable. Furthermore, it has a non-zero derivative.

   Gradient descent cannot find a minimum for the cost function without a non-zero derivative that tells it the direction in which to move.

10. Draw a picture of the ReLu activation function.

---

It is a horizontal line that bends upward at $x = 0$ to become a line with $slope = 1$.

The value of ReLu(x) is 0.0 for all $x < 0.0$ and $x$ for all $x \geq 0.0$.

11. List some of the choices that a data scientist has when designing a neural network.

    Hint: Look at the section in Chapter 10 of *Hands-On Machine Learning* that is titled *Fine-Tuning Neural Network Hyperparameters*.

    Hint: Look at TensorFlow Playground

---

- Choose the number of hidden layers in the network.
- Choose the number of neurons in each hidden layer.
- Choose the activation functions in each layer.
- Select a learning rate.
- Specify a batch size.
- Specify the number of epochs for training.
- Consider a different optimizer.

12. Keras gives us more than one API for building neural networks. Name two.

---

- sequential API
- functional API
- subclassing API

## Group Activity

In the afternoon hour we will together read and discuss What is Natural Language Processing?, by IBM Cloud Education.