Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

ISO 9001:2015 Certified
Level I Institutionally Accredited

LSPU Self-Paced Learning Module (SLM)

Course	Capstone Project 1
Sem/AY	Second Semester/2020-2021
Module No.	Module 1 – Topic 2
Lesson Title	**Selection and Formulation of Research Topic**
Week Duration	**Week 3**
Date	**March 29 - April 2**
Description of the Lesson	This topic focuses selection and formulation of the research topic providing the students research topics through orientation of various topic sources from global to institutional perspective. It also discusses research process, steps in identifying research project topics, strategies in refining topics that would lead to formulating thesis statement, research questions and research objectives

Learning Outcomes

Intended Learning Outcomes	Students should be able to meet the following intended learning outcomes: Conceptualize and develop a good research topic.
Targets/ Objectives	At the end of the lesson, students should be able to: 1. relate on the topics presented from various sources 2. identify viable research topics from the various sources 3. select a good research topic 4. enhance topic selected

Student Learning Strategies

Online Activities (Synchronous/ Asynchronous)	A. Online Discussion via Google Meet For this module you will be directed to engage in a one-hour synchronous discussion and the rest will be asynchronous and offline activities. To access to the online course materials please check your Google Classroom Account. These are the list of course materials provided on the LMS:

01 SLM-02- Selection and Formulation of Research Topic
02 Topic 2 Presentation
03 Topic 2 Course Video
04 For further study: watch this video link
05 Reading Supplements:

- 01 SDGs_Booklet_Web_En
- 02 Agenda for Sustainable Development web
- 03 Localization of SDG
- 04 PDP-and-Ambisyon-2040_NDC_v1
- 05 Abridged-PDP-2017-2022_Final
- 06 Approved Harmonized National RD Agenda 2017-2022
- 07 Harmonized DOST
- 08 RA01 - Exploring the research in information technology implementation

The one-hour synchronous discussion will be on the schedule reflected on your certificate of registration and will be done in Google Meet. Please be reminded to prepare and be ready 15 minutes prior to the said schedule to lessen connection issues. For those who cannot attend the session recordings will be available after and will be posted with 24 hours. In case you may not be able to attend the session, ensure to notify your instructor. Please be reminded of the web conference etiquettes and reminders uploaded on you LMS.

You will be given time to complete all performance tasks and activity provided on the LMS as listed below:

1. Watch the video lecture
2. Read the SLM
3. Accomplish performance tasks using work sheet provided available in Google Classroom and submit at the submission link provided
4. For further study- Watch this video with this url: How to Develop a Good Research Topic
https://youtu.be/nXNztCLYgxc
5. Attend the synchronous class - Google Meet
6. Participate in the online discussion: Activity No. 6: Mapping Research Agenda
7. Please read this “Exploring the research in information technology implementation for Activity 8
8. Participate in the online discussion: Activity No. 7: Topic Reflection
9. Do this Activity No. 8: Paper Review - Exploring the research in information technology implementation.

	Note: The insight that you will post on online discussion forum using Learning Management System (LMS) will receive additional scores in class participation. You will be given time to complete all assessment tasks and activity provided on the LMS. 1. Watch the video lecture 2. Read the SLM 3. Accomplish the performance tasks: ● Activity No. 6: Mapping Research Agenda [Group Activity] ● Activity No. 7: Topic Reflection ● Activity No. 8: Topic Reflection and Paper Review - Exploring the research in information technology implementation (For further instructions, refer to your Google Classroom and see the schedule of activities for this module) ***Note:*** *The insight that you will post on online discussion forum using Learning Management System (LMS) will receive additional scores in class participation.*
Offline Activities (e-Learning/Self-Paced)	For offline classes, please refer to the following learning guide questions: 1. Watch the video lecture 2. Read the SLM 3. Accomplish performance tasks using work sheet provided for the following activities: •Activity No. 6: Mapping Research Agenda [Group Activity] •Activity No. 7: Topic Reflection •Activity No. 8: Topic Reflection and Paper Review - Exploring the research in information technology implementation

Module Content

Topics Covered:

Topic 1. Introduction to Neural Networks
- 1.1 Linear Regression
- 1.2 Linear Classification
- 1.3 Overfitting problem and model validation
- 1.4 Model regularization
- 1.5 Stochastic gradient descent
- 1.6 Gradient descent extensions
- 1.7 Multilayer perceptron (MLP)
- 1.8 Chain rule
- 1.9 Backpropagation
- 1.10 Efficient MLP implementation
- 1.11 What is Tensorflow?
- 1.12 What Deep Learning is and is not?
- 1.13 Deep Learning as Language

This topic focuses on the foundation of deep learning algorithm which is the regression and neural network. The students are expected to understand the structure of linear regression and neural networks.

I. Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

Before attempting to fit a linear model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable causes the other (for example, higher SAT scores do not cause higher college grades), but that there is some significant association between the two variables. A scatterplot can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two variables is the correlation coefficient, which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

A linear regression line has an equation of the form **Y = a + bX**, where **X** is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when x = 0).

Least-Squares Regression

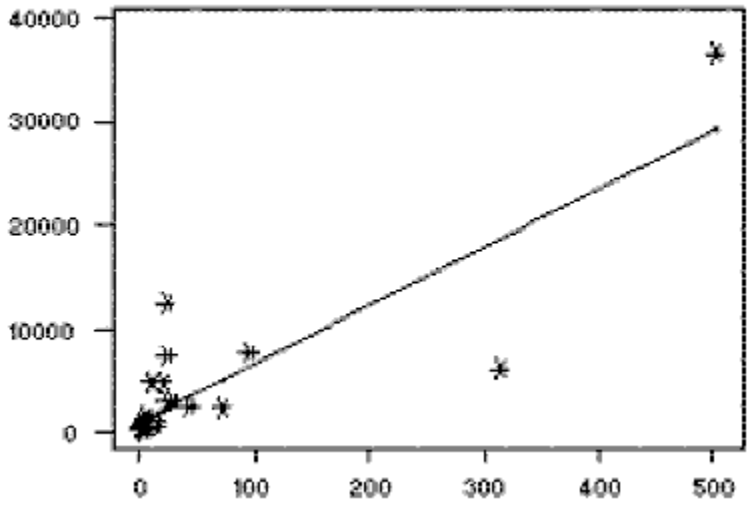
The most common method for fitting a regression line is the method of least-squares. This method calculates the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line (if a point lies on the fitted line exactly, then its vertical deviation is 0). Because the deviations are first squared, then summed, there are no cancellations between positive and negative values.

Example

The dataset "Televisions, Physicians, and Life Expectancy" contains, among other variables, the number of people per television set and the number of people per physician for 40 countries. Since both variables probably reflect the level of wealth in each country, it is reasonable to assume that there is some positive association between them. After removing 8 countries with missing values from the dataset, the remaining 32 countries have a correlation coefficient of 0.852 for number of people per television set and number of people per physician. The r^2 value is 0.726 (the square of the correlation coefficient), indicating that 72.6% of the variation in one variable may be explained by the other. (Note: see correlation for more detail.) Suppose we choose to consider number of people per television set as the explanatory variable, and number of people per physician as the dependent variable. Using the MINITAB "REGRESS" command gives the following results:

The regression equation is People.Phys. = 1019 + 56.2 People.Tel.

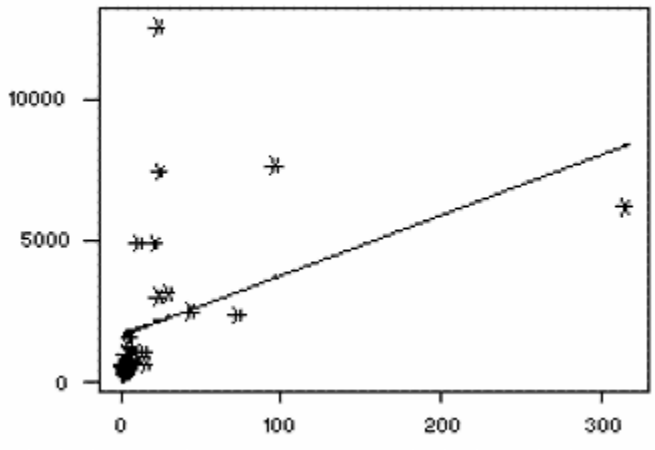
To view the fit of the model to the observed data, one may plot the computed regression line over the actual data points to evaluate the results. For this example, the plot appears to the right, with number of individuals per television set (the explanatory variable) on the x-axis and number of individuals per physician (the dependent variable) on the y-axis. While most of the data points are clustered towards the lower left corner of the plot (indicating relatively few individuals per television set and per physician), there are a few points which lie far away from the main cluster of the data. These points are known as outliers, and depending on their location may have a major impact on the regression line (see below).



Data source: The World Almanac and Book of Facts 1993 (1993), New York: Pharos Books. Dataset available through the JSE Dataset Archive.

Outliers and Influential Observations

After a regression line has been computed for a group of data, a point which lies far from the line (and thus has a large residual value) is known as an ***outlier***. Such points may represent erroneous data, or may indicate a poorly fitting regression line. If a point lies far from the other data in the horizontal direction, it is known as an ***influential observation***. The reason for this distinction is that these points have may have a significant impact on the slope of the regression line. Notice, in the above example, the effect of removing the observation in the upper right corner of the plot:



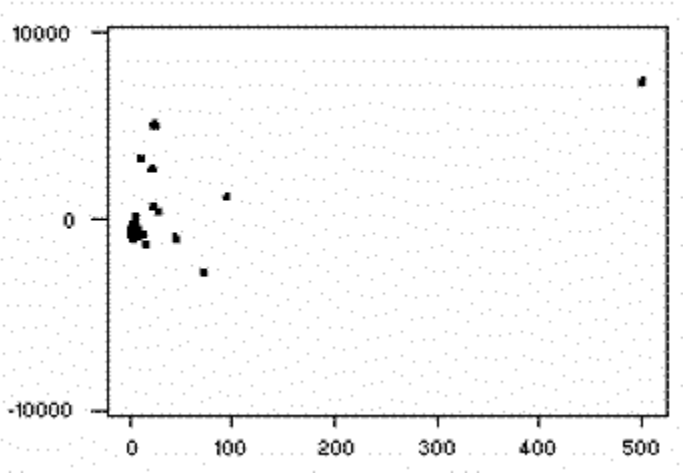
With this influential observation removed, the regression equation is now

People.Phys = 1650 + 21.3 People.Tel.

The correlation between the two variables has dropped to 0.427, which reduces the r^2 value to 0.182. With this influential observation removed, less that 20% of the variation in number of people per physician may be explained by the number of people per television. Influential observations are also visible in the new model, and their impact should also be investigated.

Residuals

Once a regression model has been fit to a group of data, examination of the residuals (the deviations from the fitted line to the observed values) allows the modeler to investigate the validity of his or her assumption that a linear relationship exists. Plotting the residuals on the y-axis against the explanatory variable on the x-axis reveals any possible non-linear relationship among the variables, or might alert the modeler to investigate lurking variables. In our example, the residual plot amplifies the presence of outliers.



Lurking Variables

If non-linear trends are visible in the relationship between an explanatory and dependent variable, there may be other influential variables to consider. A lurking variable exists when the relationship between two variables is significantly affected by the presence of a third variable which has not been included in the modeling effort. Since such a variable might be a factor of time (for example, the effect of political or economic cycles), a time series plot of the data is often a useful tool in identifying the presence of lurking variables.

Extrapolation

Whenever a linear regression model is fit to a group of data, the range of the data should be carefully observed. Attempting to use a regression equation to predict values outside of this range is often inappropriate, and may yield incredible answers. This practice is known as extrapolation. Consider, for example, a linear model which relates weight gain to age for young children. Applying such a model to adults, or even teenagers, would be absurd, since the relationship between age and weight gain is not consistent for all age groups.

For further understanding: *https://www.youtube.com/watch?v=CtKeHnfK5uA*

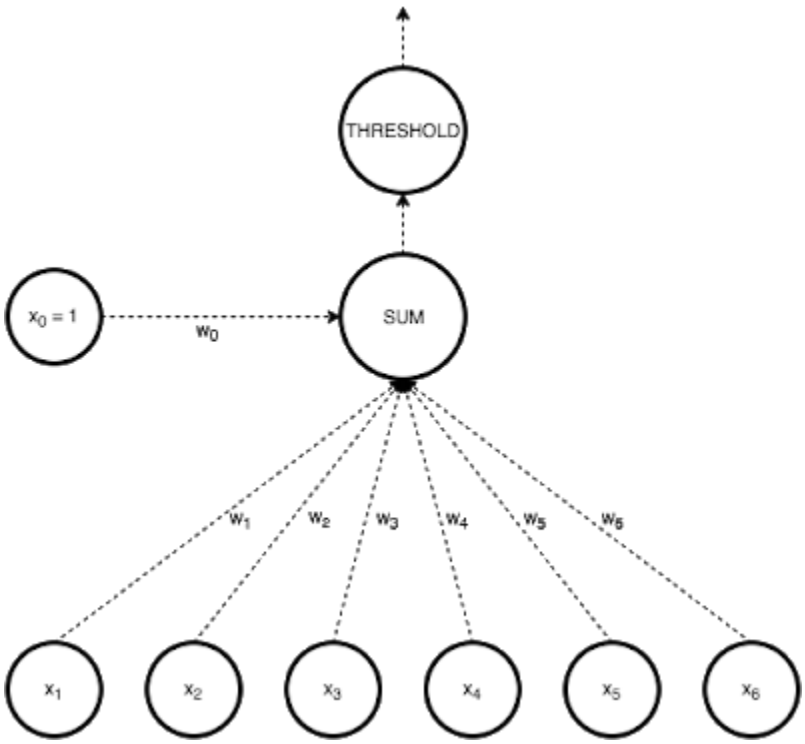
II. Linear Classifier

Linear classifiers classify data into labels based on a linear combination of input features. Therefore, these classifiers separate data using a line or plane or a hyperplane (a plane in more than 2 dimensions). They can only be used to classify data that is linearly separable. They can be modified to classify non-linearly separable data.

Perceptron

In Perceptron, we take weighted linear combination of input features and pass it through a thresholding function which outputs 1 or 0. The sign of w^Tx tells us which side of the plane $w^Tx=0$, the point x lies on. Thus, by taking threshold as 0, perceptron classifies data based on which side of the plane the new point lies on.

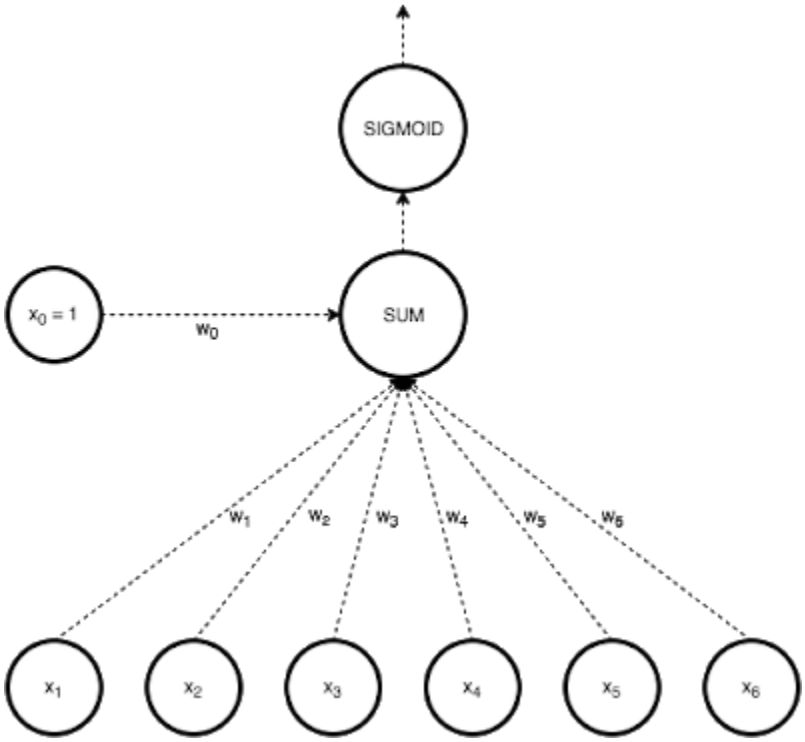
The task during training is to arrive at the plane (defined by w) that accurately classifies the training data. If the data is linearly separable, perceptron training always converges.



Perceptron

Logistic Regression

In Logistic regression, we take weighted linear combination of input features and pass it through a sigmoid function which outputs a number between 1 and 0. Unlike perceptron, which just tells us which side of the plane the point lies on, logistic regression gives a probability of a point lying on a particular side of the plane. The probability of classification will be very close to 1 or 0 as the point goes far away from the plane. The probability of classification of points very close to the plane is close to 0.5.



Logistic Regression

For further learning: *https://www.youtube.com/watch?v=yIYKR4sgzI8*

III. Model Validation

Model validation is the process of evaluating whether the hypothesis function is an acceptable description of data. Model validation techniques check whether predictive accuracy of model deteriorates when presented with previously unseen data (data not used during training). Model validation is an useful technique to avoid overfitting.

Cross validation is a popular model validation technique which evaluates how well a hypothesis function generalizes over an independent dataset.

Cross Validation

In machine learning problems, we are given a training set on which the hypothesis function is trained and a test set on which it is evaluated. In cross validation, the training set is split into training set and validation set. The purpose of validation set is to evaluate performance of hypothesis function on unseen data during the training phase. Validation set is a small sample of training data.

A typical flow of cross-validation is as follows -

1. Sample a small subset from training data as validation set
2. Train a hypothesis on remaining data
3. Evaluate its accuracy on validation set

Multiple such hypothesis functions are trained by performing above steps multiple times. The resulting hypothesis function is then taken as the weighted average of predictions of all the hypothesis functions. The weights are proportional to the accuracies of the hypotheses on their corresponding validation sets

K-fold Cross Validation

In k-fold validation, the training data is split into k equal sized sets. Out of this 1 set is kept as validation set and remaining k-1 sets are used for training. The process is repeated for all k sets to obtain k hypothesis functions each having an accuracy evaluated over its corresponding validation set. The k results are then averaged to produce a single prediction.
Advantage of this method is that all examples are used for training and validation and all of them are used for validation only once. k=n, where n is the number of samples in training set, is a special case called one-fold cross validation.

Further Reading and References

Wikipedia contributors. "Cross-validation (statistics)." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 6 Sep. 2016. Web. 6 Sep. 2016.

[Chapter 5] Mitchell, Tom M. "Machine learning." (1997).

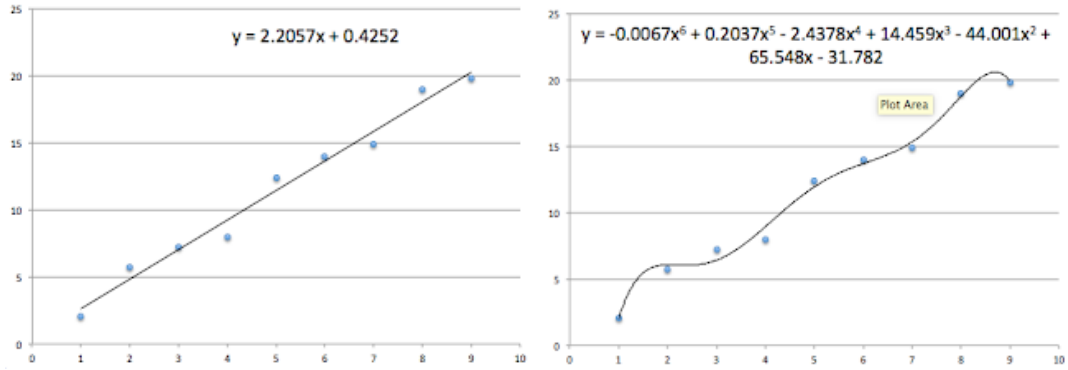
Overfitting and Regularization

Regularization is the process of modifying the cost function by adding a 'penalty' which prevents overfitting.

Occam's Razor

Principal of Occam's Razor states that, given everything else is the same, a shorter/simpler explanation for the observed data should be preferred over a longer/complex one.

Given training data, there could be multiple hypothesis functions that explain it with same/similar accuracies. For example, refer figure below. Occam's Razor states that the simpler hypothesis, which is a first-degree polynomial, describes the actual data better than the sixth order polynomial (Even though the accuracy of sixth order polynomial is better on the training data). Even our intuition says that the straight line represents the data better and the higher order polynomial would not be able to accurately represent unseen data.



Two hypothesis functions of different complexities explaining same data

Overfitting

Training data often contains random noise. The noise could come from various sources like measurement errors, etc. When an optimization algorithm tries to reduce cost, it tries to fit the hypothesis function exactly for all points in the training data. Thu the algorithm ends up overfitting. Such a hypothesis function performs poorly on the test data.

Multilayer Perceptron

The perceptron is very useful for classifying data sets that are linearly separable. They encounter serious limitations with data sets that do not conform to this pattern as discovered with the XOR problem. The XOR problem shows that for any classification of four points that there exists a set that are not linearly separable.

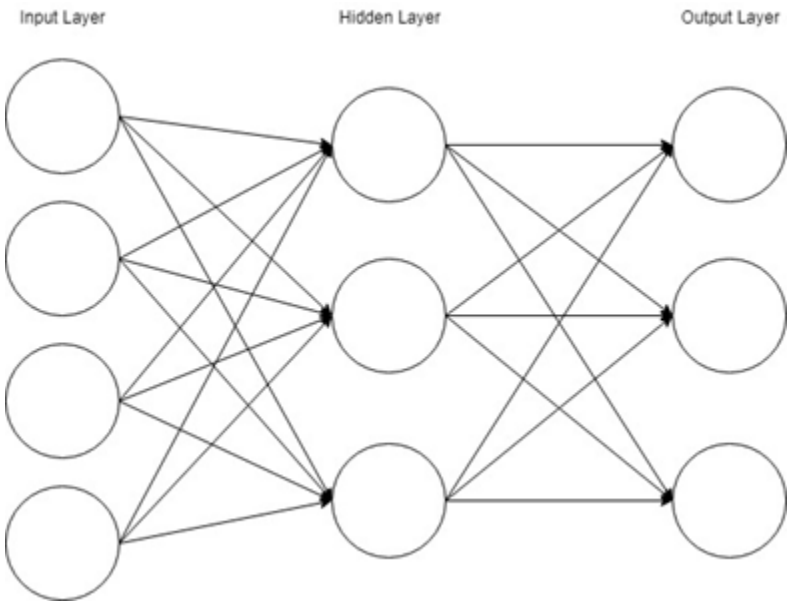
The MultiLayer Perceptron (MLPs) breaks this restriction and classifies datasets which are not linearly separable. They do this by using a more robust and complex architecture to learn regression and classification models for difficult datasets.

How does a multilayer perceptron work?

Multilayer Perceptron (MLP): used to apply in computer vision, now succeeded by Convolutional Neural Network (CNN). MLP is now deemed insufficient for modern advanced computer vision tasks. Has the characteristic of fully connected layers, where each perceptron is connected with every other perceptron. Disadvantage is that the number of total parameters can grow to very high (number of perceptron in layer 1 multiplied by # of p in layer 2 multiplied by # of p in layer 3…). This is inefficient because there is redundancy in such high dimensions. Another disadvantage is that it

disregards spatial information. It takes flattened vectors as inputs. A light weight MLP (2–3 layers) can easily achieve high accuracy with MNIST dataset.

The Perceptron consists of an input layer and an output layer which are fully connected. MLPs have the same input and output layers but may have multiple hidden layers in between the aforementioned layers, as seen below.



The algorithm for the MLP is as follows:

1. Just as with the perceptron, the inputs are pushed forward through the MLP by taking the dot product of the input with the weights that exist between the input layer and the hidden layer (WH). This dot product yields a value at the hidden layer. We do not push this value forward as we would with a perceptron though.
2. MLPs utilize activation functions at each of their calculated layers. There are many activation functions to discuss: rectified linear units (ReLU), sigmoid function, tanh. Push the calculated output at the current layer through any of these activation functions.
3. Once the calculated output at the hidden layer has been pushed through the activation function, push it to the next layer in the MLP by taking the dot product with the corresponding weights.
4. Repeat steps two and three until the output layer is reached.
5. At the output layer, the calculations will either be used for a backpropagation algorithm that corresponds to the activation function that was selected for the MLP (in the case of training) or

a decision will be made based on the output (in the case of testing).

Additional reference: https://www.youtube.com/watch?v=u5GAVdLQyIg

Backpropagation

Back-propagation algorithm enables us to calculate the partial derivative of the cost function w.r.t each weight in the network. Since we know the output loss as a function of all weights, we can calculate all the partial derivatives analytically. Or we could perturb each weight at a time and observe the change in the output loss and thus calculate partial derivatives. But this process is cumbersome and when the number of weights increase, the calculation becomes computationally expensive. This was one of the reasons why interest in neural network research was lost in the early 70s. The invention of backpropagation algorithm made it easy to calculate partial derivatives w.r.t all the weights in the network easy. This advancement led to renewed interest in neural networks research.

For visual discussion: https://www.youtube.com/watch?v=Ilg3gGewQ5U
https://www.youtube.com/watch?v=iajq0xQZ2cQ

Tensorflow

TensorFlow's eager execution is an imperative programming environment that evaluates operations immediately, without building graphs: operations return concrete values instead of constructing a computational graph to run later. This makes it easy to get started with TensorFlow and debug models, and it reduces boilerplate as well. To follow along with this guide, run the code samples below in an interactive python interpreter.
Eager execution is a flexible machine learning platform for research and experimentation, providing:

- *An intuitive interface*—Structure your code naturally and use Python data structures. Quickly iterate on small models and small data.
- *Easier debugging*—Call ops directly to inspect running models and test changes. Use standard Python debugging tools for immediate error reporting.
- *Natural control flow*—Use Python control flow instead of graph control flow, simplifying the specification of dynamic models.

Eager execution supports most TensorFlow operations and GPU acceleration.

Tensorflow Variable

A TensorFlow **variable** is the recommended way to represent shared, persistent state your program manipulates. This guide covers how to create, update, and manage instances of tf.Variable in TensorFlow.

Variables are created and tracked via the tf.Variable class. A tf.Variable represents a tensor whose value can be changed by running ops on it. Specific ops allow you to read and modify the values of this tensor. Higher level libraries like tf.keras use tf.Variable to store model parameters.

Create a variable

To create a variable, provide an initial value. The tf.Variable will have the same dtype as the initialization value.

```
my_tensor = tf.constant([[1.0, 2.0], [3.0, 4.0]])
my_variable = tf.Variable(my_tensor)

# Variables can be all kinds of types, just like tensors
bool_variable = tf.Variable([False, False, False, True])
complex_variable = tf.Variable([5 + 4j, 6 + 1j])
```

A variable looks and acts like a tensor, and, in fact, is a data structure backed by a tf.Tensor. Like tensors, they have a dtype and a shape, and can be exported to NumPy.

```
print("Shape: ", my_variable.shape)
print("DType: ", my_variable.dtype)
print("As NumPy: ", my_variable.numpy())
```

```
Shape:  (2, 2)
DType:  <dtype: 'float32'>
As NumPy:  [[1. 2.]
 [3. 4.]]
```

Additional Reference: https://www.youtube.com/watch?v=tXVNS-V39A0

Performance Tasks

For the activities:

1. Watch the video lecture for Topic 1.
2. Read the SLM.
3. Accomplish performance tasks using work sheet provided available in Google Classroom and submit at the submission link provided.
4. For further study- Watch this video
5. Attend the synchronous class - Google Meet

Activity Guide:

The activities for this topic will be shared during synchronous class, however it is advisable to submit your activity or performance tasks prior to the synchronous class. Also, these activities will ensure you are processing your thought on the course content. This will enhance your learning and knowledge.

use the worksheet provided, for online classes please see visit the LMS for online discussion instructions.

Understanding Directed Assess

Learning Resources

- https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html
- https://sites.google.com/site/machinelearningnotebook2/classification/multi-class-classification/backpropagation
- https://www.coursera.org/learn/intro-to-deep-learning?specialization=aml