Version	Significant Changes (Marked with a Symbol)	Modified by	Modification Date
1.0	First Version of the course	Rojesh Shikhrakar, Kshitiz Mandal	18 December, 2019
1.1	Reorder the chapters in module 5 (RNN)	Kshitiz Mandal, Miran Ghimire	09 January, 2020
1.2	Removed graphical DL, policy gradient algorithm	Kshitiz Mandal	
1.3	Change Module 8 (RL)	Kshitiz Mandal	06 February, 2020

Contents

Introduction

All the syllabuses are reviewed regularly so that it will be able to reflect the latest thinking and current best practices employed in industries and take into account different national and international contexts in which these courses will be taught.

Syllabus Aims

The Syllabus aims to:

- Provide a worthwhile learning experience for all learners and enable them to acquire sufficient knowledge and skills to get started in the domain of AI
- Facilitate and Standardise Course Content Development and Delivery

Introduction to the Course

This is the second part of the Microdegree course covering different topics in Deep Learning.

Course Objectives:

- Select and implement appropriate libraries, framework, techniques for different deep learning architectures
- Explain the math and code behind some algorithms and build and make changes to the code
- Assess the performance, evaluate and compare different architectures to design and deploy an end-to-end solution
- Run experiment to change some details in code to improve the algorithm

Guided Hours

12 weeks course maximum of 4 hours per week.

Modules	**Class Time(hrs)**	**Self Study (hrs)**
M1 Introduction to the Course	2	
M2 Introduction to Deep Learning	2	
M3 Components of Deep	6	

Learning		
M4 Convolutional Neural Network	4	
M5 Recurrent Neural Networks	4	
M6 Attention and Neural Computers	4	
M7 Generative Models	6	
M8 Deep Reinforcement Learning	12	
M9 DL in Production	6	
Total	**48**	

Pre-requisites

Students must have completed the Machine Learning course.

Distinct Features Used in the Syllabus

Bold Outcomes refers to **Must Have** learning outcomes,

Normal Text refers to Should Have learning outcomes,

Italic Outcomes refer to *Good to Have or Could Have learning outcomes*.

Red Outcomes refers to Won't have Outcomes

Colour based on Categories

Bold Dark Blue Teach **Fundamental Concepts**

Bold Dark Green Covers **State-of-the-Art Research Topics**

Bold Dark Orange Focuses on **Application in different fields** the topic is being used.

Syllabus Outline:

Deep Learning course outline and planner

Assessment

The Assessment will be based on the cognitive domain of Bloom's Taxonomy to classify learning objectives into different levels of complexity and specificity, viz. Remember, Understand, Apply, Analyze, Evaluate, Create

Assessment Objectives	Categories	Objective	Action Words
AO1	**Remember**	Recall Facts and basic concepts	Define, list, memorize, repeat, state, recognize, recall, reproduce
AO2	**Understand**	Explain Ideas or Concepts	Classify, describe, discuss, explain, identify, locate, recognize, report, select, translate, interpret, exemplify, paraphrase,
AO3	**Apply**	Use Information in a new situation, mathematical modeling	Execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch, illustrate, write
AO4	**Analyze**	Draw connections among ideas, distinguish between ideas	Differentiate, organize, relate, compare, contrast, distinguish, discriminate, test, examine, criticize, experiment, question, show
AO5	**Evaluate**	Justify a stand or decision	Check, Appraise, argue, defend, judge, select, support, value, critique, evaluate
AO6	**Create**	Produce new or original Work/product, point of view	Design, assemble, construct, conjecture, develop, generate, plan, produce, design, formulate, investigate, write

Source: https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/

Weight Distribution on

Components	AO1	AO2	AO3	AO4	AO5	AO6	Weights
Quizzes	25%	25%	20%	20%	10%	-	
Programming Assignment	10%	20%	20%	20%	10%	10%	
Projects	-	-	10%	20%	20%	50%	
Classroom Assessment							

Course Contents

Module 1 Introduction to the Course

Unit 1.1 Introduction to the Course

Students should be able to :

1. Introduction to the Course	A. list what course covers and what it does not cover B. describe the course structure and flow of the course through the course overview
2. Course Logistics	A. explain the assessment and evaluation criteria B. Recall and implement the honor code and violation issues

Module 2 Introduction to Deep Learning [hrs]

This module discusses the importance of deep learning compared to traditional machine learning. It talks about the brief history of deep learning and how it has impacted different fields. This module also discusses the building block of Deep learning and implement a simple ANN module using a deep learning framework.

Unit 2.1 Introduction to the Module

Students should be able to :

1. Introduction to the Module	A. Describe the motivation to learn the content in the module B. Describe the flow of the contents in the module (overview) and their learning outcomes
2. Introduction to Deep Learning	A. Show an understanding of the problem of traditional Machine Learning, i.e the need for hand-crafted extractor B. Describe that deep learning with its hierarchical layers learns abstract representations of features. C. Differentiate Deep Learning with Machine Learning D. Explain why deep learning has become more and more successful and popular recently E. Describe key events that happened that has affected deep learning development
3. Current Applications of Deep Learning	A. Exemplify different applications of deep learning in different domains B. Understand different scenarios where deep learning could or could not be used.

Unit 2.2 Neuron

Students should be able to :

1. Neurons and the Brain	A. Describe how billions of neurons forming the brain is collectively responsible for intelligence B. Describe how a biological neuron transmits information and its associated terminologies: synapse, action potential, excitation and inhibition C. Describe different ways in which researchers are trying to replicate the function of the neuron D. Describe how the brain works in a modular fashion.
2. The perceptron	A. describe Rosenblatt's perceptron as a linear unit with a step activation function B. Explain how Perceptron Learning Algorithm helps to learn the parameter for the perceptron C. Understand why perceptron fails in XOR Problem and other limitations of the perceptron D. Implement Perceptron from scratch [Assignment]

Unit 2.3 Artificial Neural Network

Students should be able to :

1. Introduction to Artificial Neural Network	A. Recall logistic regression as a classifier B. Appreciate perceptron and logistic regression as a type of artificial neuron C. Explain how neural networks can represent the nonlinear functions by stacking the layers of artificial neurons with non-linear activation functions
2. Universal approximation theorem	A. State universal approximation theorem that a feed-forward network containing a finite number of neurons with activation function can approximate continuous functions B. Understand the implications of universal approximation theorem

Unit 2.4 Learning Parameters in ANN

Students should be able to :

1. Forward propagation and Computation Graph	A. Sketch and explain the concept of a computational graph to explain and implement the process of forward propagation B. Implement Forward Propagation from Scratch
2. Error Metrics	A. Describe the purpose of the error function B. Distinguish cost vs loss C. Explain the cross-entropy loss as negative of log loss used for binary as well as multi-class classification problems D. Explain MAE and MSE as a loss function for regression problems E. Recognize that the choice of error function depends on the problem and can be customized
3. Backward propagation	A. describe how errors propagate back through the computation graph using the chain rule. (auto differentiation) B. Sketch and estimate the error propagated to each node. C. Implement Back Propagation from Scratch
4. Gradient descent	A. Understand the intuition of gradient descent to reduce the error function by changing the parameters (moving down the loss landscape) B. Understand the update step of gradient descent C. Understand the terminologies associated with the learning the network: epoch, batch size, learning rate. D. Understand stochastic vs mini-batch vs batch gradient descent E. Implement Gradient Descent from Scratch

Unit 2.5 Implementing ANN

Students should be able to :

1. Deep Learning Frameworks	A. Know different tools available for deep learning B. Understand autograd as one of the main value propositions of DL frameworks.

2. Intro to Tensorflow	A. Understand different components of Tensorflow
3. ANN in Tensorflow	A. Recall Training, Testing, validation, hyperparameter tuning, and other relevant component required to implement ANN algorithm. B. Implement ANN in Tensorflow explaining each of the components listed above. a. Use sigmoid as the activation function b. Use the provided functions in tf.nn/tf (not keras)

Unit 2.6 Module Summary

Students should be able to :

1. Module Summary	A. Summarise the overall concepts covered in the module B. Relate the contents covered in this module with other topics covered in the course

Module 3 Components of Deep Learning

Unit 3.1 Introduction to the Module

Students should be able to :

1. Introduction to the Module	A. Understand the motivation to learn the content in the module in relation to other modules B. Understand the flow of the contents in the module (overview) and their learning outcomes

Unit 3.2 Vanishing and Exploding Gradient

Students should be able to :

1. Challenge with Gradient Descent	A. Recall gradient descent B. Understand the challenges associated with Gradient Descent
2. Vanishing Gradient	A. Understand the problem of Vanishing gradient B. Understand how the choice of activation function improve performance in the above problem C. List the different other techniques that has been useful to improve upon this problem
3. Exploding Gradients	A. Understand the problem of Exploding gradient B. Implement gradient clipping to solve the problem of exploding gradient C. List the different other techniques that has been useful to improve upon this problem

Unit 3.3 Activation Function and Weight Initialisation

Students should be able to :

1. Activation Function	A. Describe different activation in use with motivation: a. Variants of ReLU: LReLU, PReLU,RReLU,SReLU b. Exponential linear units (ELU) c. Maxout B. Implement tf.nn.relu instead of Sigmoid in ANN Architecture a. Write code for sigmoid using tf from scratch b. Write code for relu using tf from scratch c. Use tf.nn.relu in the ANN code created in 2.5.3 (B)
2. Weight Initialisation	A. Relate the Importance of Initializing weights to vanishing and exploding gradients and other benefits (convergence speed, ...) B. Understand the implication of uniform (zeros,ones,constant) Initialisation and random (Normal,Uniform) Initialisation C. Implement weight initialization using a. Tf.random.normal and tf.random_normal_initializer() b. Tf.random.uniform and tf.random_uniform_initializer()
3. Xavier Initialisation	A. Understand the poor performance in training deep feedforward NN B. using random initialization

	C. Understand Xavier improves over the above problem
4. Other Initialisation	A. Discuss about other Initialisation techniques such as a. he initialisation, b. lecun initialization

Unit 3.4 Optimization methods

Students should be able to :

1. SGD with Momentum	A. Understand the challenges associated with SGD B. Describe exponentially weighted average C. Understand and visualize concept of momentum D. Understand and visualize the concept of Nesterov momentum. E. Implement SGD with momentum or Nestrov Momentum [tf] a. Implement the maths behind the optimization using tensorflow.
2. Adagrad, RMSprop, Adam	A. Understand about Adagrad, RMSProp, Adam B. Implement Adagrad, RMSProp, Adam using tf C. Compare adagrad, RMSprop and Adam [comparisons: accuracy, rate, chances of getting stuck at local minima]

Unit 3.5 Normalization

Students should be able to:

1. Introduction	A. Recognize the need for the need of normalization
2. Standardizing and Min max normalization	A. Understand the concept of and implement a. Standardization i. Use tf.math.reduce_mean/tf.math.reduce_std b. min-max normalization. i. Use tf.math.reuduce_max/tf.math.reduce_min
3. Batch normalization	A. Understand the motivation behind batch normalization B. Implement the batch normalization using tf a. Use tf.nn.batch_normalization() b. Use tf.keras.layers.BatchNormalization()
4. Layer normalization	A. Understand the motivation behind layer normalization0 B. Implement the layer normalization using tf a. Use tf.keras.layers.LayerNormalization()

Unit 3.6 Regularization

Students should be able to :

1. Introduction to Regularization	A. Appreciate the importance of regularization and how it reduces overfitting B. Recall the concept of Bias-Variance Tradeoff and appreciate the idea that variance caused by non-convexity of loss landscape in NN doesn't follow the traditional Tradeoff C. Recall L1 and L2 Regularisation in the context of ANN

2. Early Stopping	A. Understand what is early stopping and when is it used B. Implement early stopping using tf
3. Weight Decay	A. Understand the what is weight decay and when it used B. Implement weight decay using tf
4. Dropout, dropconnect	A. Understand dropout and how it helps in regularization B. Understand the concept of fast dropout C. Understand the concept and distinguish the concept of Dropout and dropconnect D. Implement dropout and dropconnect in tf
5. Data Augmentation	A. Understand the motivation behind data augmentation technique. B. Show some examples of data augmentation technique in a. image, b. sound, c. nlp.

Unit 3.7 Hyperparameter Tuning

Students should be able to :

1. Introduction to Hyperparameter Tuning	A. Appreciate the need of hyperparameter tuning B. Highlight the importance of hyper parameter tuning the learning rate first in DL. C. Suggest different techniques and best practises of hyperparameter tuning
2. Hyperparameter Search in DL	A. Recall the idea of Grid Search and problem of using grid search in Deep Learning B. Recall Random Search and Demonstrate an example of Random Search in ANN C. Recall Genetic Grid Search and Demonstrate an example of Genetic Grid Search in ANN
3. Learning rate scheduling	A. Understand the concepts of learning rate scheduling and different scheduling techniques such as: a. Constant learning rate b. Time based decay, 1/t decay c. Step decay d. Exponential decay e. Cyclic B. Implement learning rate scheduling in tf
4. Bayesian Optimisation	A. Understand the concept of Gaussian Process. B. Understand the concept behind Bayesian Optimisation for hyperparameter Tuning
5. Architecture Search	A. Appreciate the importance of architecture search B. Understand the concept of Neural Architecture Search (NAS) C. Appreciate about the concept of AutoML

Unit 3.8 Module Summary

Students should be able to :

1. Module Summary	A. Summarise the overall concepts covered in the module B. Relate the contents covered in this module with other topics covered in the course

Module 4 Convolutional Neural Network

Students will be able to understand and implement CNN network and its associated terminologies.

They should be able to understand different seminal architectures using CNN

Unit 4.1 Introduction to the Module

Students should be able to :

1. Introduction to the Module	A. Appreciate the motivation behind convolution and CNN B. Describe the different computer vision problems we are trying to solve using CNN C. Exemplify different application of CNN D. Understand the flow of the contents in the module (overview) and their learning outcomes

Unit 4.2 Building Block of CNN

Students should be able to :

1. Convolution	A. Describe 2d convolution operation with example of edge detection B. Implement 2d convolution using numpy/tensorflow C. Explain the concept of filter along with the impact of the following on convolution a. Filters size b. Stride [reduce spatial dimension, make model invariant to small translation of input] c. Padding D. Determine the size of output image given the filter size, stride and padding E. Explain the idea of convolutions over volume F. Implement Convolution Module in tf a. tf.nn.conv2d() b. tf.keras.layers.Conv2D()
2. Pooling and FC	A. Understand the concept and importance of Pooling B. Understand the effect of a. Average Pooling b. Max pooling C. Implement using tf: a. Average pooling i. tf.nn.avg_pool2d() ii. tf.keras.layers.AveragePooling2D() b. Max pooling i. tf.nn.max_pool2d() ii. tf.keras.layers.MaxPool2D() D. Understand the need of Fully Connected Layer
3. Backpropagation in CNN	A. Understand how backpropagation works for a single convolution layer B. Understand how backpropagation works for pooling layers a. Average Pooling b. Max pooling

	C. Implement Backpropagation using TF core. D. *Visualization of feature map*
4. LeNet	A. Understand the Motivation and working behind LeNet B. Implement LeNet using TF a. Use only the tf.nn functions and not tf.keras

5. Features of CNN	A. Understand the idea and limitations of
	    a. Local Connectivity: Receptive field
	    b. Shift Invariance
	    c. Parameter sharing in contrast to Dense Network

6. Regularisation in CNN	A. Explain how regularisation studied in previous chapter applies and impact CNN B. Explain how Batch normalization is used to address the issues related to internal covariance shift within feature maps C. Discuss special regularisation trick used in CNN

7. Special CNN Techniques	A. Discuss Special Cases of CNN such as
	    a. Cascade CNN
	    b. Dilated CNN
	        i. Use tf.nn.conv2d with dilation parameter to show the difference
	        ii. Use tf.nn.atrous_conv2d()

Unit 4.3 Seminal Architecture

Talk about the main key points of that architecture, and its implementation if possible.

Students should be able to :

1. VGGNet	A. Understand the contribution of Alexnet B. Discuss on Effective Receptive Field and effect of small size Filters C. Describe VGGNet and its specific contribution D. Implement VGGNet architecture a. Implement the VGG block by using tf.keras api E. Demonstrate how the blocks can be combined to form a model.
2. Network-in-Network (NIN)	A. Explain how Network-in-Network enhance model discriminability for local patches within the receptive field B. Discuss the impact and contribution of NIN and 1X1 Convolution
3. InceptionNet	A. Describe GoogleLeNet and its contribution B. Understand the motivation behind InceptionBlock that uses multiple filter sizes C. Understand the ideas of Spatial Exploitation, Parallelism and Depth Exploitation as exploited by Inception Family D. Discuss improvements over Inception V2, V3, V4 E. Implement the Inception Block a. Using tf.keras to implement the Inception v4 block.
4. ResNet	A. Appreciate the problem that Resnet aims to solve B. Elaborate the impact of multipath connectivity and Skip connections C. Describe ResNet and its specific contribution

	D. Implement Residual Block a. Using tf.keras
5. DenseNet	A. Describe DenseNet and its specific contribution
6. RestNet	*A.* Describe RestNet and its specific contribution
7. Transfer Learning	**A.** Appreciate the importance of pre-training and freezing the layers **B.** Use pre-trained models for different use-cases

Unit 4.4 Module Summary

Students should be able to :

1. Module Summary	A. Summarise the overall concepts covered in the module B. Relate the contents covered in this module with other topics covered in the course

Module 5 Recurrent Neural Networks

Unit 5.1 Introduction to the Module

Students should be able to :

1. Introduction to the Module	A. Exemplify and describe the problem involved in modeling sequential data and limitations of traditional autoregressive models like HMM B. Understand the motivation for RNN for overcoming the limitations of traditional autoregressive models C. Understand the flow of the contents in the module (overview) and their learning outcomes

Unit 5.2 Fundamentals of RNN

Students should be able to :

1. vanilla RNNs	A. Understand the structure of a recurrent unit and how they form connections to build a RNN B. Analyze the relation between hidden unit of ANN and recurrent unit of RNN. C. Implement forward pass in an RNN Block from tf a. Use tf.keras.layers.SimpleRNNCell and tf.keras.layers.RNN
2. Computation Graph and Backpropagation in RNN	A. Understand the computational graph of an RNN B. Workout the calculations for backpropagation. (Backpropagation Through Time) C. Implement Backward pass in RNN Block from tf. a. Use tf.keras.layers.SimpleRNNCell and tf.keras.layers.RNN
3. Vanishing &	A. Understand about long term dependencies and the problem RNNs

Exploding gradients in RNN	face(realize how vanishing and exploding gradient appears in RNN) B. Understand about gradient clipping a. How it solves exploding gradient. b. Why it cannot be used to solve vanishing gradients. C. Solution for vanishing gradient: LSTM, GRU, Bidirectional RNNs
4. Deep Recurrent Networks	A. Describe the concept of Deep Recurrent Networks and why stacking is useful. B. Implement stacked RNN cells using tf a. Use tf.keras.layers.StackedRNNCells
5. Strategies for Multiple Time Scales	A. Adding skip connection through Time B. Understand Leaky (Hidden) units with linear self-connections behavior as moving average C. Removing connections
6. Bidirectional RNNs	A. Describe Bidirectional RNN and it's contribution

Unit 5.3 Seminal Architectures

Students should be able to :

1. Long Short-Term Memory networks (LSTMs)	A. Describe the components of cell state in LSTM. B. Discuss the impact of the cell states C. Understand how LSTM addresses the long term dependency problem D. Discuss about bi-directional LSTM E. Implement LSTM a. Using tf.keras.layers.LSTMCell b. Using tf.keras.layers.LSTM	
2. Gated-Recurrent Units (GRUs)	A. Describe the components of the gates in GRU B. Analyze the structure of GRU in contrast to LSTM C. explain when to use GRU instead of LSTM D. Implement GRU a. Using tf.keras.layers.GRUCell b. Using tf.keras.layers.GRU	
3. CNN LSTM	A. Recall about CNN B. Recognize that different architectures C. describe the flow of the contents in the can be combined focusing on LSTM and CNN	

Add:

- Variations of LSTM
    - LSTM with Peephole connections
    -

Unit 5.4 Module Summary

Students should be able to :

1. Module Summary	A. Summarize the overall concepts covered in the module B. Discuss Area of use and Practical Examples of RNN and LSTM C. Challenges in using RNN & LSTM D. Relate the contents covered in this module with other topics covered in the course

Module 6 Attention and Neural Computation

Unit 6.1 Introduction to the Module

Students should be able to :

1. Introduction to the Module	A. describe the problem involved in modeling sequential data and the motivation for Attention module (overview) and their learning outcomes

Unit 6.2 Neural Attention

Students should be able to :

1. Sequence-to-Sequence Architectures	A. Describe encoder and decoders. B. Describe the sequence to sequence architecture.
2. Attention	A. describe the drawbacks of vanilla seq2seq architecture B. describe how attention mechanism addresses the drawbacks of seq2seq
3. Attention Types	A. describe the different types of attention a. Self Attention b. Soft vs Hard Attention c. Global vs local attention B. Implement soft attention using tf.keras
4. Attention is all you need	A. Discuss about the Attention is all you need paper

Unit 6.3 Neural Computers

Students should be able to :

1. Neural Computers	A. Understand the motivation behind neural computers B. Describe the types of neural computers
2. Neural Turing Machines	A. Understand the motivation behind NTM B. Describe the architecture of NTM
3. Differentiable Neural Computers	A. Understand how it extends on NTM B. Describe the architecture of differentiable Neural Computers

Unit 6.4 Module Summary

Students should be able to :

1. Module Summary	A. Summarize the overall concepts covered in the module B. Relate the contents covered in this module with other topics covered in the course.

Module 7 Deep Unsupervised Learning

Unit 7.1 Introduction to the Module

Students should be able to :

1. Introduction to the Module	A. Understand the importance of unsupervised learning B. Compare and state examples of Discriminative and Generative Architecture C. Understand the flow of the contents in the module (overview) and their learning outcomes

Unit 7.2 AutoEncoders

Students should be able to :

1. Autoencoders	A. Recall about encoder and decoder B. Explain about auto encoders using a single layer in both encoder and decoder C. Implement denoising encoder decoder using tf.keras
2. Variational AutoEncoders (VAE)	A. Discuss about Variational AutoEncoders B. Recall about KL divergence C. Relate on how VAE is a generative network
3. the “reparameterization trick”	A. Explain the reparameterization trick B. Implement the reparameterization trick for calculating loss in tf

Unit 7.3 Generative Adversarial Networks

Students should be able to :

1. GAN’s	A. Define about generators and discriminators B. Explain about the minmax/adversarial game the discriminators and generators are trying to optimize and the loss function used. C. Problems with training GANs a. Mode collapse b. Discriminators behaviour i. Poor performance: no accurate feedback ii. Great performance: vanishing gradient c. No proper evaluation metric D. Implement a generator block and a discriminator model E. Train a GAN
2. Wasserstein GANs	A. Explain about Wasserstein Distance B. Describe why the Wasserstein distance improves over the traditional GAN C. Train a WGAN
3. Adversarially Learned Inference (ALI)	A. Recognize the inference problem with GAN’s B. Discuss how ALI uses encoder-decoder architecture for the generators and discriminators.

4. Deep Convolutional GAN (DCGAN)	A. Appreciate how CNN can be used for unsupervised learning task as well in DCGAN B. Understand the structure of the CNNs for generators and discriminators. C. Train a DCGAN
5. Conditional GAN	A. Appreciate how GAN's can be extended to a conditional model by conditioning the generator and discriminator. B. Realize how CGANs are not strictly unsupervised. C. Train a CGAN

Unit 7.4 Module Summary

Students should be able to :

1. Module Summary	A. Summarise the overall concepts covered in the module B. Discuss current research in deep unsupervised learning methods C. Discuss challenges of applying unsupervised learning techniques to real-world problems D. Relate the contents covered in this module with other topics covered in the course

Module 8 Deep Reinforcement Learning

Unit 8.1 Introduction to the Module

Students should be able to :

1. Introduction to the Module	A. Understand the motivation to learn the content in the module B. Understand the type of problem to be solved using Reinforcement Learning. C. Comparison of RL with supervised and unsupervised learning D. Understand the flow of the contents in the module (overview) and their learning outcomes

2. Overview RL basic Concept.	A. Recall Distributions and sampling
	B. Components in the RL
	    a. Agent
	    b. State
	    c. Action
	    d. Reward
	    e. Environment
	C. Block Diagram of RL stating the components and their interaction

3. K-armed Bandit Problem	A. Solve k-armed bandit problem B. Understand the Exploration vs Exploitation tradeoff

4. Markov Decision Process (MDP)	A. Understand the Markov property
	B. Understand the additional elements of MDP
	    a. Transition Probability
	    b. Reward Probability
	    c. Discount factor
	C. Understand the relation between rewards and returns
	D. Bellman Equation
	E. Understand:
	    a. Policy functions
	    b. Value functions
	    c. Action value functions
	F. Use a 4x4 gridworld as an example to calculate the state values solving the Bellman equations

5. Dynamic Programming	A. Understand Policy Evaluation and Policy Improvement B. Understand about Policy Iteration for solving MDP C. Understand about Value Iteration for solving MDP D. Identify the problem with DP and how asynchronous DP can address the issue.

6. Monte Carlo Methods	A. Understand about the Monte Carlo Methods and Model free
	B. Discuss on types of Value approximation in Monte Carlo (Prediction)
	    a. First visit
	    b. Every visit
	C. Use policy evaluation and policy improvement to define Generalized Policy Iteration

7. Temporal Difference Learning	A. Understand how Temporal Difference (TD) combines Monte Carlo (MC) method and Dynamic Programming (DP) B. Understand about TD controls: a. SARSA: On-policy TD Control b. Q-learning: Off-policy TD Control C. Differentiate between Q-learning and SARSA

Unit 8.2 Deep Q-learning

https://www.freecodecamp.org/news/improvements-in-deep-q-learning-dueling-double-dqn-prioritized-experience-replay-and-fixed-58b130cc5682/

Students should be able to :

Introduction	A. Discuss about deep learning in Reinforcement Learning B. Briefly explain why the vanilla deep neural networks failed to perform for RL tasks
1. Deep Q Network	A. Understand the mechanisms in DQN a. Target Network b. Experience Relay B. Implement DQN in tensorflow

Unit 8.3 Policy Gradient

https://www.freecodecamp.org/news/improvements-in-deep-q-learning-dueling-double-dqn-prioritized-experience-replay-and-fixed-58b130cc5682/

Students should be able to :

2. Policy Gradient	A. Discuss about Policy Gradient B. Understand Policy Gradient Theorem C. Differentiate between the value based and policy based approaches.
3. DPG	A. Discuss about the Deterministic Policy Gradient (DPG) (No implementation needed)

Unit 8.4 Echo Module Summary

Students should be able to :

1. Module Summary	A. Summarize the overall concepts covered in the module B. Relate the contents covered in this module with other topics covered in the course

Module 9 DL in Production

Student's should be able to deploy a research level machine learning model to a fully integrated API that can be utilized to make predictions

Unit 9.1 Introduction to the Module

Students should be able to :

1. Introduction to the Module	A. Recall Machine Learning Pipeline and describe how model deployment fits into Deep Learning Pipeline, along with ideas about production environments B. Describe overall Deployment Pipelines C. Describe different possible scenarios and challenges in model deployment such as pluggability, reusability, scalability, fault tolerance,	A.

Unit 9.2 Deployment

Students should be able to :

1. Deep Learning Hardware	A. Evaluate the hardware options viz. CPU, GPU and TPU for different use cases and decide what to use, understand constraints of each. B. Evaluate cases where GPU is suitable and recommend GPU resources (Memory/speed) in different scenarios C. Evaluate and choose appropriate Cloud based Services to suit the need of the project. D. Describe and discuss possibilities and opportunities with Neuromorphic Chips	A.
2. Serving Model Via Rest API	A. Create Model Serving API endpoints for prediction in response to requests	A. Flask-RESTful — Flask-RESTful 0.3.8 documentation
3. Deployment Ecosystem	A. Understand the pipeline for Continuous integration B. Understand how to use SaaS for CI (Travis, Jenkins, etc.) C. Describe and use Containerization (via Docker) D. Describe uses of Kubernetes E. Deployment in different Production environment a. Local Machine Deployment b. Within a Docker container F. Interoperability using ONNX	A. https://www.docker.com/products/kubernetes B. https://travis-ci.org/ C. https://jenkins.io/ D. ONNX | Home
4. DL Deployment	A. Develop pipeline using TensorFlow Extended (TFX) a. TF Serving b. TF data pipeline B. Understand concepts of Deploying to Cloud a. GCP Deep learning Containers b. AWS Lambda/Serverless C. Deploy using On-Device Inference and Edge Computing toolkits	A. TensorFlow Extended (TFX) B. C. TensorFlow Lite D. TensorFlow.js for ML using JavaScript

	a. Reducing network size/ quantization b. On Mobile & IOT using Tensorflow Lite c. On Browser via Tensorflow.js	
5. DataOps & ModelOps	A. Differentiate DevOps, DataOps and ModelOps B. Use tools for Data and Model Versioning C. Evaluate different methods for Model Updating	A. http://hummer.io/docs/2019-ic2e-modelops.pdf B. C. https://dvc.org/
6. Post-Production	A. Debugging on Production model B. Online Evaluation of Models C. Maintenance of DL Models	A.

Unit 9.3 Scalability and Data Privacy

Students should be able to :

1. Scalability	A. Describe how vertical and Horizontal Scalability is performed for Deep Learning B. Describe different Parallelization Methods a. Data Parallelization b. Model Parallelization c. Pipeline Parallelism d. Hybrid Parallelism C. Run Distributed DL model training in multiple servers and/or GPUs D. Discuss challenges and issues with scaling	A. [1903.11314] Scalable Deep Learning on Distributed Infrastructures: Challenges, Techniques and Tools B. [1802.09941] Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis C. Custom training with tf.distribute.Strategy
2. Data Privacy and Learning	A. Justify Data Privacy Concerns with relevant examples, and current trends in Data Privacy B. Describe Differential Privacy and discuss on Global and Local Differential Privacy C. Describe how privacy is protected in Federated Learning a. Horizontal Federated learning b. Vertical Federated Learning c. Federated Transfer Learning D. Use Tensorflow Privacy tool / keras tf-federated	A. https://www.udacity.com/course/secure-and-private-ai--ud185 B. C. https://github.com/tensorflow/privacy
3. Trustable AI	A. Justify the importance of Fairness in AI and methods to prevent Bias B. Exemplify different Adversarial Attack and different methods to improve Adversarial Robustness C. Define with explainability/interpretability is important and Use tools to improve interpretability (eg LIME)	A. https://arxiv.org/abs/1908.09635 B. C. https://christophm.github.io/interpretable-ml-book/

Unit 9.4 Module Summary

Students should be able to :

1. Module Summary		

	A. Summarise the overall concepts covered in the module B. Relate the contents covered in this module with other topics covered in the course	

Glossary of command words

This glossary table below is a guide to develop learning outcomes for each chapter that would help to develop, teach and learn content as well as for evaluation.

Command words	**AO**	**Indication (What it means)**	**Use Cases**
Define,	1	Formal statement/phrases of definition	equations, terms
What is meant by	1	Definition + Significance or context of the term	
Describe	2	State the main points, with diagrams, examples or so on	Phenomena, experiment, observation
Explain	2	Reasoning with reference of theory applied	Relationship, cause-effects,
State	1	Express in concise words, without supporting arguments	Theorem, law, fact, value without calculation
List	1	List down number of points without elaboration	
Exemplify	1,2	Provide examples	
Discuss	5	Critical account on the topic	critique,
Deduce, solve, Predict	4	Produce the answer through logical connections than just by recall	
Suggest, propose	4	applying knowledge to a new situation or when there is no unique idea	
Calculate, find out, workout, carry out	3	get a numerical from given data, some value, through some work	Workout to calculate a value
Determine	3	Quantity calculated with certainty	Magnitude, scale ,
Show	3	Derive result through a structured explicit evidence	
Justify, support	4	Support a case with evidence/arguments	
Verify, prove	4	Confirm that a given statement/result is true	
Estimate	4,5	Reasoned order of magnitude for the quantity	
Sketch	3,4	Make freehand sketch drawing/curve with key features	Diagrams, graphs, figures
Compare	4	Provide similarities and differences	
Recognise, identify, name, select	1	identify from having encountered them before	

Implement, code	3,6		
Create, design, construct,	6		
understand			
appreciate			