

[Home](#) [Teaching](#) [Presentations](#) [Publications](#) [Graduate Students](#) [Curriculum](#)
[Vitae](#) [Problems](#) [Contact](#)

Deep Learning (Fall 2019)

Course: Deep Learning

[View on GitHub](#)

Course (Deep Learning): [Deep Learning Tutorials](#) [TensorFlow Tutorials](#) [Graph Neural Networks](#) [Projects](#) [Data Handling](#)

Lecturer: [Hossein Hajiabolhassan](#)

The Webpage of the Course: [Deep Learning](#)

[Data Science Center](#), [Shahid Beheshti University](#)

Index:

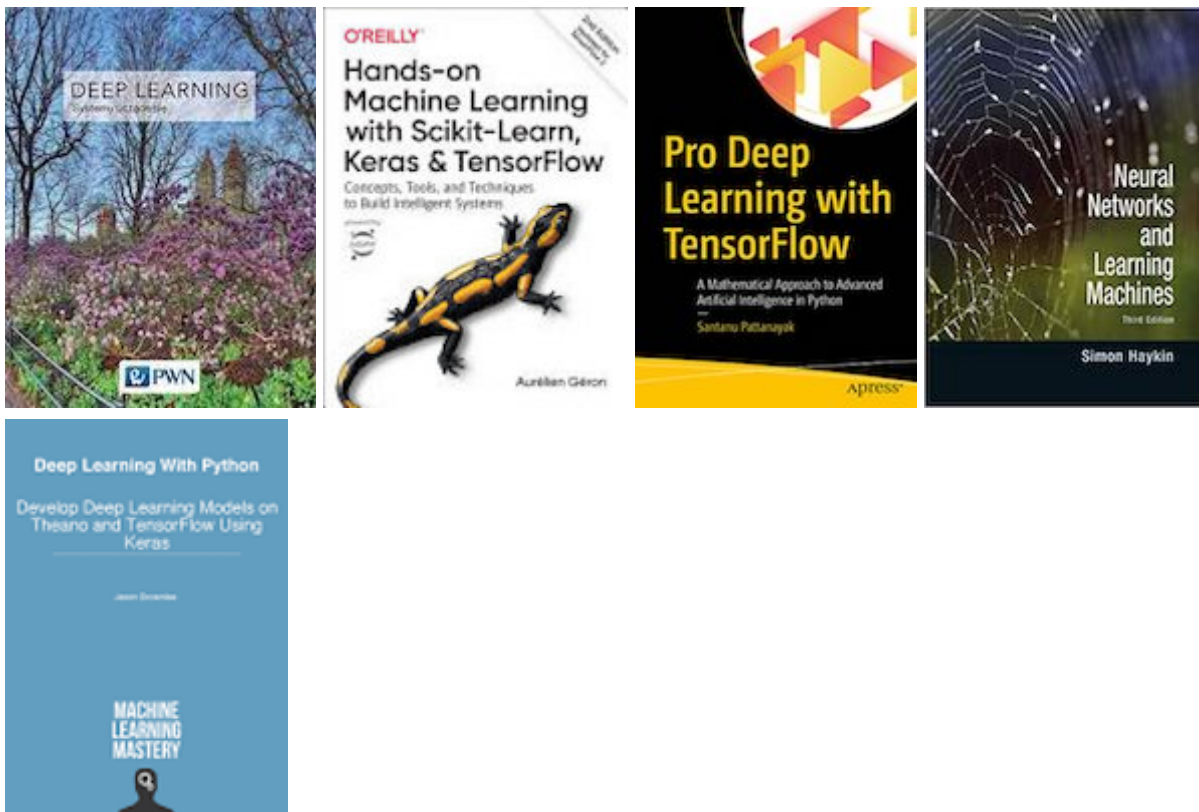
- [Course Overview](#)
- [Main TextBooks](#)
- [Slides and Papers](#)
 1. [Lecture 1: Introduction](#)
 2. [Lecture 2: Toolkit Lab 1: Google Colab and Anaconda](#)
 3. [Lecture 3: Toolkit Lab 2: Image Preprocessing by Keras](#)
 4. [Lecture 4: Deep Feedforward Networks](#)
 5. [Lecture 5: Toolkit Lab 3: Introduction to Artificial Neural Networks with Keras](#)
 6. [Lecture 6: Regularization for Deep Learning](#)
 7. [Lecture 7: Optimization for Training Deep Models](#)
 8. [Lecture 8: Toolkit Lab 4: Training Deep Neural Networks](#)
 9. [Lecture 9: Toolkit Lab 5: Custom Models and Training with TensorFlow 2.0](#)
 10. [Lecture 10: Convolutional Networks](#)
 11. [Lecture 11: Toolkit Lab 6: TensorBoard](#)
 12. [Lecture 12: Sequence Modeling: Recurrent and Recursive Networks](#)
 13. [Lecture 13: Practical Methodology](#)
 14. [Lecture 14: Applications](#)
 15. [Lecture 15: Autoencoders](#)
 16. [Lecture 16: Generative Adversarial Networks](#)
 17. [Lecture 17: Graph Neural Networks](#)
- [Additional Resources](#)
- [Class Time and Location](#)

- Projects
 - Google Colab
 - Fascinating Guides For Machine Learning
 - Latex
- Grading
- Prerequisites
 - Linear Algebra
 - Probability and Statistics
- Topics
- Account
- Academic Honor Code
- Questions
- Miscellaneous:
 - Data Handling

Course Overview:

In **this** course, you will learn the foundations **of** Deep Learning, understand how to build neural networks, and learn how to lead successful machine learning projects. You will learn about Convolutional networks, RNNs, LSTM, Adam, Dropout, BatchNorm, and more.

Main TextBooks:



Main TextBooks:

- [Deep Learning](#) (available in online) by Bengio, Yoshua, Ian J. Goodfellow, and Aaron Courville
- [Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow \(2nd Edition\)](#) by Aurelien Geron

Additional TextBooks:

- [Pro Deep Learning with TensorFlow: A Mathematical Approach to Advanced Artificial Intelligence in Python](#) by Santanu Pattanayak
- [Neural Networks and Learning Machines \(3rd Edition\)](#) by Simon Haykin
- [Deep Learning with Python](#) by J. Brownlee

Slides and Papers:

Recommended Slides & Papers:

1. Introduction

Required Reading:

- [Chapter 1](#) of the [Deep Learning](#) textbook.
- Slide: [Introduction](#) by Ian Goodfellow

Suggested Reading:

- Demo: [3D Fully-Connected Network Visualization](#) by Adam W. Harley

Additional Resources:

- [Video](#) of lecture by Ian Goodfellow and discussion of Chapter 1 at a reading group in San Francisco organized by Alena Kruchkova
- Paper: [On the Origin of Deep Learning](#) by Haohan Wang and Bhiksha Raj

Applied Mathematics and Machine Learning Basics:

- Slide: [Mathematics for Machine Learning](#) by Avishkar Bhoopchand, Cynthia Mulenga, Daniela Massiceti, Kathleen Siminyu, and Kendi Muchungi
- Blog: [A Gentle Introduction to Maximum Likelihood Estimation and Maximum A Posteriori Estimation \(Getting Intuition of MLE and MAP with a Football Example\)](#) by Shota Horii

1. Toolkit Lab 1: Google Colab and Anaconda

Required Reading:

- Blog: [Google Colab Free GPU Tutorial](#) by Fuat
- Blog: [Managing Environments](#)
- Blog: [Kernels for Different Environments](#)
- Install: [TensorFlow 2.0 RC is Available](#)

Suggested Reading:

- Blog: [Stop Installing Tensorflow Using pip for Performance Sake!](#) by Michael Nguyen
- Blog: [Using Pip in a Conda Environment](#) by Jonathan Helmus
- Blog: [How to Import Dataset to Google Colab Notebook?](#)
- Blog: [How to Upload Large Files to Google Colab and Remote Jupyter Notebooks](#) (For Linux Operating System) by Bharath Raj

Additional Resources:

- PDF: [Conda Cheat Sheet](#)
- Blog: [Conda Commands \(Create Virtual Environments for Python with Conda\)](#) by LipingY
- Blog: [Colab Tricks](#) by Rohit Midha

1. Toolkit Lab 2: Image Preprocessing by Keras

Required Reading:

- Blog: [How to Load, Convert, and Save Images With the Keras API](#) by Jason Brownlee
- Blog: [Classify Butterfly Images with Deep Learning in Keras](#) by Bert Carremans
Read the part of Data augmentation of images
- Blog: [Keras ImageDataGenerator Methods: An Easy Guide](#) by Ashish Verma

Suggested Reading:

- Blog: [Keras ImageDataGenerator and Data Augmentation](#) by Adrian Rosebrock
- Blog: [How to Configure Image Data Augmentation in Keras](#) by Jason Brownlee
- Blog: [A Quick Guide To Python Generators and Yield Statements](#) by Jason Rigden
- NoteBook: [Iterable, Generator, and Iterator](#)
- Blog: [Vectorization in Python](#)
- Blog: [numpy.vectorize](#)

Additional Resources:

- Blog: [Learn about ImageDataGenerator](#) by Yumi
- Blog: [Images Augmentation for Deep Learning with Keras](#) by Jakub Skalecki
- Blog: [A Detailed Example of How to Use Data Generators with Keras](#) by Afshine Amidi and Shervine Amidi
- Blog: [Iterables vs. Iterators vs. Generators](#) by Vincent Driessen

1. Deep Feedforward Networks

Required Reading:

- [Chapter 6](#) of the [Deep Learning](#) textbook.
- Slide: [Feedforward Neural Networks \(Lecture 2\)](#) by Ali Harakeh
- Slides: [Deep Feedforward Networks 1](#) and [2](#) by U Kang
- Chapter 20 of [Understanding Machine Learning: From Theory to Algorithms](#)
- Slide: [Neural Networks](#) by Shai Shalev-Shwartz
- Slide: [Backpropagation and Neural Networks](#) by Fei-Fei Li, Justin Johnson, and Serena Yeung
- Blog: [7 Types of Neural Network Activation Functions: How to Choose?](#)
- Blog: [Back-Propagation, an Introduction](#) by Sanjeev Arora and Tengyu Ma

Interesting Questions:

- [Why are non Zero-Centered Activation Functions a Problem in Backpropagation?](#)

Suggested Reading:

- Blog: [The Gradient](#) by Khanacademy
- Blog: [Calculus on Computational Graphs: Backpropagation](#) by Christopher Olah

Additional Resources:

- Blog: [Activation Functions](#) by Sefik Ilkin Serengil
- Paper: [Mish: A Self Regularized Non-Monotonic Neural Activation Function](#) by Diganta Misra
- Blog: [Activation Functions](#)
- Blog: [Analytical vs Numerical Solutions in Machine Learning](#) by Jason Brownlee
- Blog: [Validating Analytic Gradient for a Neural Network](#) by Shiva Verma
- Blog: [Stochastic vs Batch Gradient Descent](#) by Divakar Kapil

- **Video:** (.flv) of a presentation by Ian Goodfellow and a group discussion at a reading group at Google organized by Chintan Kaur.
- **Extra Slide:**
 - Slide: [Deep Feedforward Networks](#) by Ian Goodfellow

1. Toolkit Lab 3: Introduction to Artificial Neural Networks with Keras

Required Reading:

- Notebook: [Chapter 10 – Introduction to Artificial Neural Networks with Keras](#) from [Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow \(2nd Edition\)](#) by Aurelien Geron

Suggested Reading:

- Blog: [Epoch vs Batch Size vs Iterations](#) by Sagar Sharma
- Blog: [How to Load Large Datasets From Directories for Deep Learning in Keras](#) by Jason Brownlee
- Blog: [A Thing You Should Know About Keras if You Plan to Train a Deep Learning Model on a Large Dataset](#) by Soumendra P
 - Question: [Keras2 ImageDataGenerator or TensorFlow tf.data?](#)
 - Blog: [Better Performance with tf.data](#) by the TensorFlow Team
- Blog: [Standardizing on Keras: Guidance on High-level APIs in TensorFlow 2.0](#) by the TensorFlow Team
- Blog & Notebook: [How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras](#) by Jason Brownlee

Additional Resources:

- PDF: [Keras Cheat Sheet](#)
- Blog: [Properly Setting the Random Seed in ML Experiments. Not as Simple as You Might Imagine](#) by Open Data Science
- Blog: [Technical Notes On Using Data Science & Artificial Intelligence: To Fight For Something That Matters](#) by Chris Albon (read the Keras section)
- Blog: [Keras Tutorial: Develop Your First Neural Network in Python Step-By-Step](#) by Jason Brownlee
- Blog: [How to Use the Keras Functional API for Deep Learning](#) by Jason Brownlee
- Blog: [Keras Tutorial for Beginners with Python: Deep Learning Example](#)
- Blog: [Learn Tensorflow 1: The Hello World of Machine Learning](#) by Google Codelabs
- Blog: [Learn Tensorflow 2: Introduction to Computer Vision \(Fashion MNIST\)](#) by Google Codelabs
- Blog: [Your first Keras Model, with Transfer Learning](#) by Google Codelabs

- Blog & Notebook: [How to Choose Loss Functions When Training Deep Learning Neural Networks](#) by Jason Brownlee
- Blog: [TensorFlow 2.0 Tutorial 02: Transfer Learning](#) by Chuan Li

Building Dynamic Models Using the Subclassing API:

- Object-Oriented Programming:
 - Blog: [Object-Oriented Programming \(OOP\) in Python 3](#) by the Real Python Team
 - Blog: [How to Explain Object-Oriented Programming Concepts to a 6-Year-Old](#)
 - Blog: [Understanding Object-Oriented Programming Through Machine Learning](#) by David Ziganto
 - Blog: [Object-Oriented Programming for Data Scientists: Build your ML Estimator](#) by Tirthajyoti Sarkar
 - Blog: [Python Callable Class Method](#) by Lalu Erfandi Maula Yusnu
- The Model Subclassing API:
 - Blog: [How Objects are Called in Keras](#) by Adaickalavan

1. Regularization for Deep Learning

Required Reading:

- [Chapter 7](#) of the [Deep Learning](#) textbook.
- Slide: [Regularization For Deep Models \(Lecture 3\)](#) by Ali Harakeh
- Slide: [Bagging and Random Forests](#) by David Rosenberg
- Slide: [Deep Learning Tutorial \(Read the Part of Dropout\)](#) by Hung-yi Lee

Suggested Reading:

- Blog & Notebook: [How to Build a Neural Network with Keras Using the IMDB Dataset](#) by Niklas Donges
- Blog & Notebook: [Neural Network Weight Regularization](#) by Chris Albon
- Blog: [Train Neural Networks With Noise to Reduce Overfitting](#) by Jason Brownlee
- Blog & Notebook: [How to Improve Deep Learning Model Robustness by Adding Noise](#) by Jason Brownlee
- Paper: [Ensemble Methods in Machine Learning](#) by Thomas G. Dietterich
- Paper: [Dropout: A Simple Way to Prevent Neural Networks from Overfitting](#) by Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov

Additional Reading:

- Blog: [TensorFlow 2.0 Tutorial 04: Early Stopping](#) by Chuan Li

- Blog: [Analysis of Dropout](#) by Paolo Galeone
- **Extra Slides:**
 - Slide: [Regularization for Deep Learning](#) by Ian Goodfellow
 - Slides: Regularization for Deep Learning [1](#) and [2](#) by U Kang
 - Slide: [Training Deep Neural Networks](#) by Aykut Erdem

1. Optimization for Training Deep Models

Required Reading:

- [Chapter 8](#) of the [Deep Learning](#) textbook.
- Slide: [Optimization for Training Deep Models \(Lecture 4\)](#) by Ali Harakeh
- Slide: [Optimization for Training Deep Models - Algorithms \(Lecture 4\)](#) by Ali Harakeh
- Blog: [Batch Normalization in Deep Networks](#) by Sunita Nayak

Suggested Reading:

- Lecture Note: [Matrix Norms and Condition Numbers](#) by Ralucca Gera
- Blog: [Initializing Neural Networks](#) by Katanforoosh & Kunin, [deeplearning.ai](#), 2019
- Blog: [How to Initialize Deep Neural Networks? Xavier and Kaiming Initialization](#) by Pierre Ouannes
- Blog: [What Is Covariate Shift?](#) by Saeed Izadi
- Blog: [Stay Hungry, Stay Foolish](#): This interesting blog contains the computation of back propagation of different layers of deep learning prepared by Aditya Agrawal

Additional Reading:

- Blog: [Why Momentum Really Works](#) by Gabriel Goh
- Blog: [Understanding the Backward Pass Through Batch Normalization Layer](#) by Frederik Kratzert
- [Video](#) of lecture / discussion: This video covers a presentation by Ian Goodfellow and group discussion on the end of Chapter 8 and entirety of Chapter 9 at a reading group in San Francisco organized by Taro-Shigenori Chiba.
- Blog: [Preconditioning the Network](#) by Nic Schraudolph and Fred Cummins
- Paper: [Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification](#) by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun
- Blog: [Neural Network Optimization](#) by Matthew Stewart
- Paper: [Understanding the Disharmony between Dropout and Batch Normalization by Variance Shift](#) by Xiang Li, Shuo Chen, Xiaolin Hu, and Jian Yang
- **Extra Slides:**
 - Slide: [Conjugate Gradient Descent](#) by Aarti Singh

- Slide: [Training Deep Neural Networks](#) by Aykut Erdem
- Slides: Optimization for Training Deep Models [1](#) and [2](#) by U Kang

1. Toolkit Lab 4: Training Deep Neural Networks

Required Reading:

- Notebook: [Chapter 11 – Training Deep Neural Networks](#) from [Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow \(2nd Edition\)](#) by Aurelien Geron

Suggested Reading:

- Blog: [How to Accelerate Learning of Deep Neural Networks With Batch Normalization](#) by Jason Brownlee
- Blog: [Why is my Validation Loss Lower than my Training Loss?](#) by Adrian Rosebrock

Additional Resources:

- PDF: [Self-Normalizing Neural Networks](#) by Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter

1. Toolkit Lab 5: Custom Models and Training with TensorFlow 2.0

Required Reading:

- Notebook: [Chapter 12 – Custom Models and Training with TensorFlow](#) from [Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow \(2nd Edition\)](#) by Aurelien Geron
- Slide: [Introducing tf.data](#): The `tf.data` module contains a collection of classes that allows you to easily load data, manipulate it, and pipe it into your model. The slides were prepared by Derek Murray, the creator of `tf.data` explaining the API (don't forget to read the speaker notes below the slides).
- Notebook: [Chapter 13 – Loading and Preprocessing Data with TensorFlow](#) from [Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow \(2nd Edition\)](#) by Aurelien Geron

Suggested Reading:

- Blog: [What's Coming in TensorFlow 2.0](#) by the TensorFlow Team
- Blog: [TF.data Reborn from the Ashes](#) by Prince Canuma
- Blog: [Introducing Ragged Tensors](#) by Laurence Moroney
- Blog & Notebook: [Load Images with tf.data \(A File from a URL, If It is not Already in the Cache\)](#)
- Blog: [How to use Dataset and Iterators in Tensorflow with Code Samples](#) by Prasad Pai

- Blog: Analyzing `tf.function` to Discover AutoGraph Strengths and Subtleties: [Part 1](#), [Part 2](#), and [Part 3](#) by Paolo Galeone
- Blog: [TPU-Speed Data Pipelines: `tf.data.Dataset` and `TFRecords`](#) by Google Codelabs

Additional Resources:

- Blog: [Building a Data Pipeline \(Using Tensorflow 1 and `tf.data` for Text and Images\)](#)
- Blog: [Swift](#) was announced in 2014. The Swift programming language has quickly become one of the fastest growing languages in history. Swift makes it easy to write software that is incredibly fast and safe by design.
- GitHub: [Swift for TensorFlow](#)

TensorFlow 1.0:

- To Learn TensorFlow 1.0, Check the Section of [TensorFlow-1](#).

1. Convolutional Networks

Required Reading:

- [Chapter 9](#) of the [Deep Learning](#) textbook.
- Slide: [Convolutional Neural Networks \(Lecture 6\)](#) by Ali Harakeh
- Slide: [Convolutional Networks](#) by Ian Goodfellow

Suggested Reading:

- Blog: [Convolutional Neural Networks CheatSheet](#) by Afshine Amidi and Shervine Amidi
- Notebook: [Chapter 14 – Deep Computer Vision Using Convolutional Neural Networks](#) from [Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow \(2nd Edition\)](#) by Aurelien Geron
- Blog: [Image Convolution Examples](#) by Utkarsh Sinha
- Blog: [Convolutions and Backpropagations](#) by Pavithra Solai
- Blog: [Understanding Convolutions](#) by Christopher Olah
- Blog: [A Comprehensive Guide to Convolutional Neural Networks — the ELI5 Way](#) by Sumit Saha
- Blog: [A Basic Introduction to Separable Convolutions](#) by Chi-Feng Wang
- Blog: [Depth wise Separable Convolutional Neural Networks](#) by Mayank Chaurasia
- Blog: [Type of convolutions: Deformable and Transformable Convolution](#) by Ali Raza
- Blog: [Review: DilatedNet — Dilated Convolution \(Semantic Segmentation\)](#) by Sik-Ho Tsang
- Blog: [Region of Interest Pooling Explained](#) by Tomasz Grel

Additional Reading:

- Blog: [A Convolutional Neural Network Tutorial in Keras and TensorFlow 2](#) by Isak Bosman
- Blog & Notebook: [Cats and Dogs Image Classification Using Keras](#) by Ashwin Joy
- Blog & Notebook: [Learn Tensorflow 3: Introduction to Convolutions](#) by Google Codelabs
- Blog & Notebook: [Learn Tensorflow 4: Convolutional Neural Networks \(CNNs\)](#) by Google Codelabs
- Blog & Notebook: [Learn Tensorflow 5: Complex Images](#) by Google Codelabs
- Blog & Notebook: [Learn Tensorflow 6: Use CNNs with Larger Datasets](#) by Google Codelabs
- Blog & Notebook: [Convolutional Neural Networks, with Keras and TPUs](#) by Google Codelabs
- Blog & Notebook: [Modern Convnets, Squeezenet, with Keras and TPUs](#) by Google Codelabs
- Blog & Notebook: [TensorFlow 2.0 Tutorial 01: Basic Image Classification](#) by Chuan Li

1. Toolkit Lab 6: TensorBoard

TensorBoard:

- Video: [Inside TensorFlow: Summaries and TensorBoard](#)
- Blog: [TensorBoard Overview](#)
- Notebook: [Get started with TensorBoard](#)
- Notebook: [Examining the TensorFlow Graph](#)
- Notebook: [Displaying Image Data in TensorBoard](#)
- Notebook: [Using TensorBoard in Notebooks](#)

Suggested Reading:

- Blog & Notebook: [TensorBoard: Graph Visualization](#)
- Blog & Notebook: [TensorBoard Histogram Dashboard](#)
- Blog & Notebook: [TensorBoard: Visualizing Learning](#)

Additional Reading:

- Notebook: [TensorBoard Scalars: Logging Training Metrics in Keras](#)
- Notebook: [Hyperparameter Tuning with the HParams Dashboard](#)
- Notebook: [TensorBoard Profile: Profiling basic training metrics in Keras](#)
- Blog: [TensorFlow 2.0 Tutorial 03: Saving Checkpoints](#) by Chuan Li

1. Sequence Modeling: Recurrent and Recursive Networks

Required Reading:

- [Chapter 10](#) of the [Deep Learning](#) textbook.

- Slide: [Sequence Modeling: Recurrent and Recursive Networks](#) by U Kang
- Slide: [Training Recurrent Nets](#) by Arvind Ramanathan
- Slide: [Long-Short Term Memory and Other Gated RNNs](#) by Sargur Srihari

Suggested Reading:

- Blog: [Understanding LSTM Networks](#) by Christopher Olah
- Blog: [Illustrated Guide to LSTM's and GRU's: A Step by Step Explanation](#) by Michael Nguyen

Additional Reading:

- [Video](#) of lecture / discussion. This video covers a presentation by Ian Goodfellow and a group discussion of Chapter 10 at a reading group in San Francisco organized by Alena Kruchkova.
- Blog: [Gentle introduction to Echo State Networks](#) by Madalina Ciortan
- Blog: [Understanding GRU Networks](#) by Simeon Kostadinov
- Blog: [Animated RNN, LSTM and GRU](#) by Raimi Karim
- Slide: [An Introduction to: Reservoir Computing and Echo State Networks](#) by Claudio Gallicchio

1. Practical Methodology

Required Reading:

- [Chapter 11](#) of the [Deep Learning](#) textbook.
- Slides: [Practical Methodology](#) by Sargur Srihari
 - Part 0: [Practical Design Process](#)
 - Part 1: [Performance Metrics](#)
 - Part 2: [Default Baseline Models](#)
 - Part 3: [Whether to Gather More Data](#)
 - Part 4: [Selecting Hyperparameters](#)
 - Part 5: [Debugging Strategies](#)

Suggested Reading:

- Metrics:
 - Blog: [Demystifying KL Divergence](#) by Naoki Shibuya
 - Blog: [Demystifying Cross-Entropy](#) by Naoki Shibuya
 - Blog: [Deep Quantile Regression](#) by Sachin Abeywardana
 - Blog: [An Illustrated Guide to the Poisson Regression Model](#) by Sachin Date
 - Blog: [Generalized Linear Models](#) by Semih Akbayrak
 - Blog: [ROC curves and Area Under the Curve Explained \(Video\)](#) by Data School

- Blog: [Introduction to the ROC \(Receiver Operating Characteristics\) Plot](#)
- Slide: [ROC Curves](#) by Maryam Shoaran
- Blog: [Precision-Recall Curves](#) by Andreas Beger

Additional Reading:

- Slide: [Practical Methodology](#) by Ian Goodfellow
- Slide: [Practical Methodology](#) by U Kang

1. Applications

Required Reading:

- [Chapter 12](#) of the [Deep Learning](#) textbook.
- Slide: [Applications](#) by U Kang

Suggested Reading:

- Blog: [How Neural Networks Learn Distributed Representations](#) By Garrett Hoffman

Additional Reading:

- Blog: [30 Amazin Applications of Deep Learning](#) by Yaron Hadad
- Slides: [Applications](#) by Sargur Srihari

1. Autoencoders

Required Reading:

- [Chapter 14](#) of the [Deep Learning](#) textbook.
- Slide: [Autoencoders](#) by Sargur Srihari

Suggested Reading:

- Blog: [Autoencoders vs PCA: When to Use?](#) by Urwa Muaz
- Blog: [Intuitively Understanding Variational Autoencoder: And Why They're so Useful in Creating Your Own Generative Text, Art and Even Music](#) by Irhum Shafkat
- Blog: [Understanding Variational Autoencoders \(VAEs\)](#) by Joseph Rocca
- Blog: [Generative Modeling: What is a Variational Autoencoder \(VAE\)?](#) by Peter Foy
- Blog: [Tutorial - What is a Variational Autoencoder?](#) by Jaan Altosaar
- Slide: [Generative Models](#) by Mina Rezaei
- Blog: [A High-Level Guide to Autoencoders](#) by Shreya Chaudhary

Additional Reading:

- Blog: [Tutorial - What is a Variational Autoencoder?](#) by Jaan Altosaar
- Slide: [Autoencoders](#) by U Kang

1. Generative Adversarial Networks

Required Reading:

Slide: [Generative Adversarial Networks \(GANs\)](#) by Binglin, Shashank, and Bhargav

Paper: [NIPS 2016 Tutorial: Generative Adversarial Networks](#) by Ian Goodfellow

Suggested Reading:

- Blog: [Generative Adversarial Networks \(GANs\), Some Open Questions](#) by Sanjeev Arora
- Paper: [Generative Adversarial Networks: An Overview](#) by Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath

Additional Reading:

- Blog: [GANs Comparison Without Cherry-Picking](#) by Junbum Cha
- Blog: [New Progress on GAN Theory and Practice](#) by Liping Liu
- Blog: [Play with Generative Adversarial Networks \(GANs\) in your browser!](#)
- Blog: [The GAN Zoo](#) by Avinash Hindupur
- [Generative Adversarial Networks \(GANs\), Some Open Questions](#) by Sanjeev Arora

1. Graph Neural Networks:

Required Reading:

- Slide: [Graph Neural Networks](#) by Xiachong Feng
- Paper: [A Comprehensive Survey on Graph Neural Networks](#) by Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, Philip S. Yu

Suggested Reading:

- Blog: [Deep Graph Library \(DGL\): A Python package that interfaces between existing tensor libraries and data being expressed as graphs.](#)

Additional Reading:

- GitHub: [Graph Neural Networks](#)

Additional Resources:

- Papers:
 - [Papers with Code](#): The mission of Papers With Code is to create a free and open resource with Machine Learning papers, code and evaluation tables.
 - [Deep Learning Papers Reading Roadmap](#) by Flood Sung
 - [Awesome - Most Cited Deep Learning Papers](#) by Terry Taewoong Um
- Deep Learning Courses:
 - [Tensorflow for Deep Learning Research](#) by Chip Huyen
 - [Deep Learning](#) by Aykut Erdem
- Program:
 - [Ludwig](#) is a toolbox built on top of TensorFlow that allows to train and test deep learning models without the need to write code. - [Installation](#)
 - [TensorFlow Playground](#): an interactive visualization of neural networks, written in typescript using d3.js by Daniel Smilkov and Shan Carter
- The blog of [Christopher Olah](#): Fascinating tutorials about neural networks
- The blog of [Adit Deshpande](#): The Last 5 Years In Deep Learning
- [Fascinating Tutorials on Deep Learning](#)
- [Deep Learning \(Faster Data Science Education by Kaggle\)](#) by Dan Becker

Class Time and Location:

Sunday and Tuesday 13:00-14:30 AM (Fall 2019)

Projects:

Projects are programming assignments that cover the topic of this course. Any project is written by [Jupyter Notebook](#). Projects will require the use of Python 3.7, as well as additional Python libraries.

- [Get Started with TensorFlow](#)

Google Colab:

[Google Colab](#) is a free cloud service and it supports free GPU!

- [How to Use Google Colab](#) by Souvik Mandal
- [Primer for Learning Google Colab](#)
- [Deep Learning Development with Google Colab, TensorFlow, Keras & PyTorch](#)

Fascinating Guides For Machine Learning:

- [Technical Notes On Using Data Science & Artificial Intelligence: To Fight For Something That Matters](#) by Chris Albon

Latex:

The students can include mathematical notation within markdown cells using LaTeX in their **Jupyter Notebooks**.

- A Brief Introduction to LaTeX [PDF](#)
- Math in LaTeX [PDF](#)
- Sample Document [PDF](#)

Grading:

- Projects and Midterm – 50%
- Endterm – 50%

Prerequisites:

General mathematical sophistication; and a solid understanding of Algorithms, Linear Algebra, and Probability Theory, at the advanced undergraduate or beginning graduate level, or equivalent.

Linear Algebra:

- Video: Professor Gilbert Strang's [Video Lectures](#) on linear algebra.

Probability and Statistics:

- [Learn Probability and Statistics Through Interactive Visualizations](#): Seeing Theory was created by Daniel Kunin while an undergraduate at Brown University. The goal of this website is to make statistics more accessible through interactive visualizations (designed using Mike Bostock's JavaScript library D3.js).
- [Statistics and Probability](#): This website provides training and tools to help you solve statistics problems quickly, easily, and accurately - without having to ask anyone for help.
- Jupyter NoteBooks: [Introduction to Statistics](#) by Bargava
- Video: Professor John Tsitsiklis's [Video Lectures](#) on Applied Probability.
- Video: Professor Krishna Jagannathan's [Video Lectures](#) on Probability Theory.

Topics:

Have a look at some reports of [Kaggle](#) or Stanford students ([CS224N](#), [CS224D](#)) to get some general inspiration.

Account:

It is necessary to have a [GitHub](#) account to share your projects. It offers plans for both private repositories and free accounts. Github is like the hammer in your toolbox, therefore, you need to have it!

Academic Honor Code:

Honesty and integrity are vital elements of the academic works. All your submitted assignments must be entirely your own (or your own group's).

We will follow the standard of Department of Mathematical Sciences approach:

- You can get help, but you MUST acknowledge the help on the work you hand in
- Failure to acknowledge your sources is a violation of the Honor Code
- You can talk to others about the algorithm(s) to be used to solve a homework problem; as long as you then mention their name(s) on the work you submit
- You should not use code of others or be looking at code of others when you write your own: You can talk to people but have to write your own solution/code

Questions?

I will be having office hours for this course on Sunday (09:00 AM–10:00 AM). If this is not convenient, email me at hhaji@sbu.ac.ir or talk to me after class.

Deep-Learning is maintained by [hhaji](#).

This page was generated by [GitHub Pages](#).