Chapter – 0 (coursera DL)

**Introduction to**

**Neural Network & Deep Learning**

Course Phases

**1.1 About the Course**

* The *Neural Network & Deep Learning* is a foundational program that will help you
* Understand the *capabilities*, *challenges*, and *consequences* of *deep learning* and
* Prepare you to *participate in the development* of leading-edge AI technology.
* Objectives:
* Build and train neural network architectures such as:
* Convolutional Neural Networks (CNN),
* Recurrent Neural Networks (RNN),
* LSTMs,
* Transformers

* learn how to make them better with 'STRATEGIES' such as
* Dropout,
* BatchNorm,
* Xavier/He initialization, and more.

* master theoretical concepts and their industry applications using Python and TensorFlow and
* tackle real-world cases such as:
* Speech recognition,
* Music synthesis,
* Chatbots,
* Machine Translation,
* Natural Language Processing (NLP), and more.

This course provides a pathway for you to take the definitive step in the world of AI by helping you gain the knowledge and skills to level up your career.

Along the way, you will also get 'career advice' from "deep learning experts" from industry and academia.

* Applied Learning Projects: By the end you’ll be able to:

1. Build and train **deep neural networks**,

* Implement vectorized neural networks,
* Identify architecture parameters, and
* Apply DL to your applications

1. Use best practices to 'train and develop test sets' and 'analyze bias/variance' for building DL applications,

* Use standard NN techniques,
* Apply optimization algorithms, and
* Implement a neural network in 'TensorFlow'

1. Use strategies for reducing errors in ML systems,

* understand complex ML settings, and apply
  + end-to-end learning,
  + transfer learning,
  + multi-task learning

1. Build a **Convolutional Neural Network (CNN**), apply it to

* Visual detection and recognition tasks,
* Use neural style transfer to generate art, and
* Apply these algorithms to image, video, and other 2D/3D data

1. Build and train **Recurrent Neural Networks (RNN)** and its variants (**GRUs, LSTMs**),

* Apply RNNs to character-level language modeling,
* Work with NLP and Word Embeddings, and
* Use HuggingFace tokenizers and transformers to perform
  + Named Entity Recognition and
  + Question Answering

**PHASES**

Phase A: Neural Networks and Deep Learning

Phase B: Improving Deep Neural Networks: Hyperparameter Tuning, Regularization and Optimization

Phase C: Structuring Machine Learning Projects

Phase D: Convolutional Neural Networks

Phase E: Sequence Models

**1.2 Phase A: Neural Networks and Deep Learning**

* Phase 'A' Objectives:

1. Study the foundational concept of 'neural networks' and 'deep learning'.
2. Be familiar with the significant technological trends driving the rise of deep learning;
3. Build, train, and apply fully connected deep neural networks;

* Implement efficient (vectorized) neural networks;
* Identify key parameters in a neural network’s architecture; and
* Apply deep learning to your own applications.

1. Understand the capabilities, challenges, and consequences of deep learning
2. Prepare you to participate in the development of leading-edge AI technology.

* Skills you'll gain:
* Tensorflow
* Deep Learning
* Hyperparameter tuning
* Mathematical Optimization
* Phase modules: There are 4 modules in this PHASE

A1. **Introduction to Deep Learning**

A2. **Neural Networks Basics**

A3. **Shallow Neural Networks**

A4. **Deep Neural Networks**

* **A1. Introduction to Deep Learning**
* Objectives:
* Analyze the major trends driving the rise of deep learning, and
* Give examples of where and how it is applied today.
* Topics: 6 Topics [74min], 2 readings, 1 quiz, 2 app items, 1 plugin

A1.01 Welcome [5]

A1.02 What is a Neural Network? [7]

A1.03 Supervised Learning with Neural Networks [8]

A1.04 Why is Deep Learning taking off? [10]

A1.05 About this Course [2]

A1.06 Geoffrey Hinton Interview [40]

Introduction to Deep Learning

* **A2. Neural Networks Basics**
* Objectives:
* Set up a machine learning problem with a neural network mindset
* Use vectorization to speed up your models.
* Topics: 19 Topics [160min], 5 readings, 1 quiz, 2 programming assignments

A2.01 Binary Classification [8]

A2.02 Logistic Regression [5]

A2.03 Logistic Regression Cost Function [8]

A2.04 Gradient Descent [11]

A2.05 Derivatives [7]

A2.06 More Derivative Examples [10]

A2.07 Computation Graph [3]

A2.08 Derivatives with a Computation Graph [14]

A2.09 Logistic Regression Gradient Descent [6]

A2.10 Gradient Descent on m Examples [8]

A2.11 Vectorization [8]

A2.12 More Vectorization Examples [6]

A2.13 Vectorizing Logistic Regression [7]

A2.14 Vectorizing Logistic Regression's Gradient Output [9]

A2.15 Broadcasting in Python [11]

A2.16 A Note on Python/Numpy Vectors [6]

A2.17 Quick tour of Jupyter/iPython Notebooks [3]

A2.18 Explanation of Logistic Regression Cost Function (Optional) [7]

A2.19 Pieter Abbeel Interview [16]

**prj:**

Downloading your Notebook, Downloading your Workspace and

Refreshing your Workspace

Neural Network Basics [50]

Python Basics with Numpy [60]

Logistic Regression with a Neural Network Mindset [180]

* **A3. Shallow Neural Networks**
* Objectives:
* Build a neural network with one hidden layer, using forward propagation and backpropagation.
* Topics: 12 Topics [109min], 1 reading, 1 quiz, 1 programming assignment

A3.01 Neural Networks Overview [4]

A3.02 Neural Network Representation [5]

A3.03 Computing a Neural Network's Output [9]

A3.04 Vectorizing Across Multiple Examples [9]

A3.05 Explanation for Vectorized Implementation [7]

A3.06 Activation Functions [10]

A3.07 Why do you need Non-Linear Activation Functions? [5]

A3.08 Derivatives of Activation Functions [7]

A3.09 Gradient Descent for Neural Networks [9]

A3.10 Backpropagation Intuition (Optional) [15]

A3.11 Random Initialization [7]

A3.12 Ian Goodfellow Interview [14]

Shallow Neural Networks [50]

**prj:**

Planar Data Classification with One Hidden Layer [180]

* **A4. Deep Neural Networks**
* Objectives:
* Analyze the key computations underlying deep learning,
* Then use them to build and train deep neural networks for Computer Vision tasks.
* Topics: 8 Topics [64min], 7 readings, 1 quiz, 2 programming assignments

A4.01 Deep L-layer Neural Network [5]

A4.02 Forward Propagation in a Deep Network [7]

A4.03 Getting your Matrix Dimensions Right [11]

A4.04 Why Deep Representations? [10]

A4.05 Building Blocks of Deep Neural Networks [8]

A4.06 Forward and Backward Propagation [10]

A4.07 Parameters vs Hyperparameters [7]

A4.08 What does this have to do with the brain? [3]

**Reading:**

Feedforward Neural Networks in Depth [10]

Confusing Output from the AutoGrader [10]

Key Concepts on Deep Neural Networks [50]

**prj:**

Building your Deep Neural Network: Step by Step [180]

Deep Neural Network - Application [180]

**1.3 Phase B: Improving Deep Neural Networks: Hyperparameter Tuning, Regularization and Optimization**

* Phase 'B' Objectives:

1. Understand the deep learning processes that drive performance and

* Generate good results systematically.

1. Train and develop test sets and analyze bias/variance for building deep learning applications;
2. Use standard neural network techniques such as

* initialization,
* L2 and dropout regularization,
* hyperparameter tuning,
* batch normalization, and
* gradient checking;

1. Implement and apply a variety of optimization algorithms, such as

* mini-batch gradient descent,
* Momentum,
* RMSprop and Adam,
* check for their convergence;

1. Implement a neural network in TensorFlow.

* Skills you'll gain:
* Gated Recurrent Unit (GRU)
* Recurrent Neural Network (RNN)
* Natural Language Processing (NLP)
* Long Short Term Memory (LSTM)
* Attention Models
* Phase modules: There are 3 modules in this PHASE

B1. **Practical Aspects of Deep Learning**

B2. **Optimization Algorithms**

B3. **Hyperparameter Tuning, Batch Normalization and Programming Frameworks**

* **B1. Practical Aspects of Deep Learning**
* objectives:
* Discover and experiment with a variety of different initialization methods,
* apply L2 regularization and dropout to avoid model 'overfitting', then
* apply gradient checking to identify errors in a fraud detection model.
* Topics: [15 Topics [130min], 4 readings, 1 quiz, 3 programming assignments, 1 app item]

B1.01 Train / Dev / Test sets [12]

B1.02 Bias / Variance [8]

B1.03 Basic Recipe for Machine Learning [6]

B1.04 Regularization [9]

B1.05 Why Regularization Reduces Overfitting? [7]

B1.06 Dropout Regularization [9]

B1.07 Understanding Dropout [7]

B1.08 Other Regularization Methods [8]

B1.09 Normalizing Inputs [5]

B1.10 Vanishing / Exploding Gradients [6]

B1.11 Weight Initialization for Deep Networks [6]

B1.12 Numerical Approximation of Gradients [6]

B1.13 Gradient Checking [6]

B1.14 Gradient Checking Implementation Notes [5]

B1.15 Yoshua Bengio Interview [25]

**readings:**

Practical aspects of Deep Learning [50]

**prj:**

Initialization [180]

Regularization [180]

Gradient Checking [180]

* **B2. Optimization Algorithms**
* objectives:
* more advanced optimizations,
* random minibatching,
* learning rate decay scheduling to speed up your models.
* Topics: [11 Topics [92min], 3 readings, 1 quiz, 1 programming assignment]

B2.01 Mini-batch Gradient Descent [11]

B2.02 Understanding Mini-batch Gradient Descent [11]

B2.03 Exponentially Weighted Averages [5]

B2.04 Understanding Exponentially Weighted Averages [9]

B2.05 Bias Correction in Exponentially Weighted Averages [4]

B2.06 Gradient Descent with Momentum [9]

B2.07 RMSprop [7]

B2.08 Adam Optimization Algorithm [7]

B2.09 Learning Rate Decay [6]

B2.10 The Problem of Local Optima [5]

B2.11 Yuanqing Lin Interview [13]

**readings:**

Optimization Algorithms [50]

**prj:**

Optimization Method

* **B3. Hyperparameter Tuning, Batch Normalization and Programming Frameworks**
* objectives:
* Explore TensorFlow to build neural networks quickly and easily,
* Train a neural network on a TensorFlow dataset.
* Topics: [11 Topics [103min], 7 readings, 1 quiz, 1 programming assignment]

B3.01 Tuning Process [7]

B3.02 Using an Appropriate Scale to pick Hyperparameters [8]

B3.03 Hyperparameters Tuning in Practice: Pandas vs. Caviar [6]

B3.04 Normalizing Activations in a Network [8]

B3.05 Fitting Batch Norm into a Neural Network [12]

B3.06 Why does Batch Norm work? [11]

B3.07 Batch Norm at Test Time [5]

B3.08 Softmax Regression [11]

B3.09 Training a Softmax Classifier [10]

B3.10 Deep Learning Frameworks [4]

B3.11 TensorFlow [15]

**readings:**

Hyperparameter tuning, Batch Normalization, Programming Frameworks [50]

**prj:**

TensorFlow Introduction [180]

**1.4 Phase C: Structuring Machine Learning Projects**

* Phase 'C' Objectives:

1. Learn how to build a successful machine learning project and

* Get to practice decision-making as a machine learning project leader.

1. Diagnose errors in a machine learning system;
2. Prioritize strategies for reducing errors;
3. understand complex ML settings, such as:

* mismatched training/test sets, and
* comparing to and/or surpassing human-level performance;

1. Apply different ML techniques, such as:

* end-to-end learning,
* transfer learning, and
* multi-task learning.

1. Basics of building and shipping many deep learning products.

* Skills you'll gain:
* Artificial Neural Network (ANN)
* Backpropagation
* Python Programming
* Deep Learning
* Neural Network Architecture
* Phase modules: There are 2 modules in this course

C1. **ML Strategy part I**

C2. **ML Strategy part II**

* **C1. ML Strategy part I**
* objectives:
* Streamline and optimize your ML production workflow by implementing:
* 'strategic guidelines' for goal-setting and
* applying 'human-level performance' to help define key priorities.
* Topics: [13 Topics [99min], 2 readings, 1 quiz, 1 app item]

C1.01 Why ML Strategy [2]

C1.02 Orthogonalization [10]

C1.03 Single Number Evaluation Metric [7]

C1.04 Satisficing and Optimizing Metric [5]

C1.05 Train/Dev/Test Distributions [6]

C1.06 Size of the Dev and Test Sets [5]

C1.07 When to Change Dev/Test Sets and Metrics? [11]

C1.08 Why Human-level Performance? [5]

C1.09 Avoidable Bias [6]

C1.10 Understanding Human-level Performance [11]

C1.11 Surpassing Human-level Performance [6]

C1.12 Improving your Model Performance [4]

C1.13 Andrej Karpathy Interview [15]

**case study:**

Bird Recognition in the City of Peacetopia [75]

* **C2. ML Strategy part II**
* objectives:
* Develop time-saving 'Error Analysis Procedures'
* Using that analysis- evaluate the most worthwhile options to pursue and gain intuition for:
* How to 'split your data' and When to use:
* multi-task,
* transfer,
* end-to-end deep learning.
* Topics: [11 Topics [131min], 2 readings, 1 quiz]

C2.01 Carrying Out Error Analysis [10]

C2.02 Cleaning Up Incorrectly Labeled Data [13]

C2.03 Build your First System Quickly, then Iterate [5]

C2.04 Training and Testing on Different Distributions [10]

C2.05 Bias and Variance with Mismatched Data Distributions [18]

C2.06 Addressing Data Mismatch [10]

C2.07 Transfer Learning [11]

C2.08 Multi-task Learning [12]

C2.09 What is End-to-end Deep Learning? [11]

C2.10 Whether to use End-to-end Deep Learning [10]

C2.11 Ruslan Salakhutdinov Interview [17]

**Case Study:**

Autonomous Driving [75]

**1.5 Phase D: Convolutional Neural Networks (CNN)**

* Phase 'D' Objectives:

1. Understand how computer vision has evolved
2. Become familiar with computer vision applications such as:

* Autonomous driving,
* Face recognition,
* Reading radiology images, and more.

1. Build a CNN, including recent variations such as: residual networks;
2. Apply Convolutional Networks to visual detection and recognition tasks;
3. Use 'neural style transfer' to generate 'art' and

* Apply these algorithms to a variety of image, video, and other 2D or 3D data.
* Skills you'll gain:
* Decision-Making
* Machine Learning
* Deep Learning
* Inductive Transfer
* Multi-Task Learning
* Phase modules: There are 4 in this phase

D1. **Foundations of Convolutional Neural Networks**

D2. **Deep Convolutional Models: Case Studies**

D3. **Object Detection**

D4. **Special Applications: Face recognition & Neural Style Transfer**

* **D1. Foundations of Convolutional Neural Networks**
* objectives:
* Implement the foundational layers of CNNs (pooling, convolutions) and
* Stack those layers properly in a deep network to solve 'multi-class image classification' problems.
* Topics: [12 Topics [139min], 5 readings, 1 quiz, 2 programming assignments, 1 app item]

D1.01 Computer Vision [5]

D1.02 Edge Detection Example [11]

D1.03 More Edge Detection [7]

D1.04 Padding [9]

D1.05 Strided Convolutions [8]

D1.06 Convolutions Over Volume [10]

D1.07 One Layer of a Convolutional Network [16]

D1.08 Simple Convolutional Network Example [8]

D1.09 Pooling Layers [10]

D1.10 CNN Example [12]

D1.11 Why Convolutions? [9]

D1.12 Yann LeCun Interview [27]

**readings:**

The Basics of ConvNets [50]

**prj:**

Convolutional Model, Step by Step [180]

Convolution Model Application [180]

* **D2. Deep Convolutional Models: Case Studies**
* objectives:
* Discover some powerful practical tricks and methods used in deep CNNs, straight from the research papers, then
* Apply transfer learning to your own deep CNN.
* Topics: [14 Topics [127min], 3 readings, 1 quiz, 2 programming assignments]

D2.01 Why look at case studies? [2]

D2.02 Classic Networks [18]

D2.03 ResNets [7]

D2.04 Why ResNets Work? [9]

D2.05 Networks in Networks and 1x1 Convolutions [6]

D2.06 Inception Network Motivation [10]

D2.07 Inception Network [8]

D2.08 MobileNet [16]

D2.09 MobileNet Architecture [8]

D2.10 EfficientNet [3]

D2.11 Using Open-Source Implementation [4]

D2.12 Transfer Learning [8]

D2.13 Data Augmentation [9]

D2.14 State of Computer Vision [12]

**readings:**

Deep Convolutional Models [50]

**prj:**

Residual Network [180]

Transfer Learning with MobileNet [180]

* **D3. Object Detection**
* objectives:
* Apply your new knowledge of CNNs to Object Detection: one of the hottest (and most challenging!) fields in computer vision.
* Topics: [14 Topics [110min], 4 readings, 1 quiz, 2 programming assignments]

D3.01 Object Localization [11]

D3.02 Landmark Detection [5]

D3.03 Object Detection [5]

D3.04 Convolutional Implementation of Sliding Windows [11]

D3.05 Bounding Box Predictions [14]

D3.06 Intersection Over Union [4]

D3.07 Non-max Suppression [8]

D3.08 Anchor Boxes [9]

D3.09 YOLO Algorithm [6]

D3.10 Region Proposals (Optional) [6]

D3.11 Semantic Segmentation with U-Net [7]

D3.12 Transpose Convolutions [7]

D3.13 U-Net Architecture Intuition [3]

D3.14 U-Net Architecture [7]

**readings:**

Detection Algorithms [50]

**prj:**

Car detection with YOLO [180]

Image Segmentation with U-Net [180]

* **D4. Special Applications: Face recognition & Neural Style Transfer**
* objectives:
* Explore how CNNs can be applied to multiple fields, including
* Art generation and
* Face recognition.
* Implement your own algorithm to generate art and recognize faces!
* Topics: [11 Topics [75min], 6 readings, 1 quiz, 2 programming assignments]

D4.01 What is Face Recognition? [4]

D4.02 One Shot Learning [4]

D4.03 Siamese Network [4]

D4.04 Triplet Loss [15]

D4.05 Face Verification and Binary Classification [6]

D4.06 What is Neural Style Transfer? [2]

D4.07 What are deep ConvNets learning? [7]

D4.08 Cost Function [3]

D4.09 Content Cost Function [3]

D4.10 Style Cost Function [13]

D4.11 1D and 3D Generalizations [9]

**readings:**

Special Applications: Face Recognition & Neural Style Transfer [50]

**prj:**

Face Recognition [180]

Art Generation with Neural Style Transfer [180]

**1.6 Phase E: Sequence Models**

* Phase 'E' Objectives:

1. Become familiar with "SEQUENCE MODELS" and their exciting applications such as:

* speech recognition,
* music synthesis,
* chatbots,
* machine translation,
* natural language processing (NLP)

1. Build and train Recurrent Neural Networks (RNNs) and commonly-used variants such as GRUs and LSTMs;

* Apply RNNs to 'Character-level' Language Modeling;

1. Gain experience with NLP and Word Embeddings;
2. Use HuggingFace tokenizers and transformer models to solve different NLP tasks such as:

* NER and
* Question Answering.
* Skills you'll gain:
* Facial Recognition System
* Tensorflow
* Convolutional Neural Network
* Deep Learning
* Object Detection and Segmentation
* Phase modules: There are 4 modules in this phase

E1. **Recurrent Neural Networks**

E2. **Natural Language Processing & Word Embeddings**

E3. **Sequence Models & Attention Mechanism**

E4. **Transformer Network**

* **E1. Recurrent Neural Networks**
* objectives:
* Discover recurrent neural networks, a type of model that performs extremely well on temporal data,
* Also get familiar with several of its variants, including LSTMs, GRUs and Bidirectional RNNs,
* Topics: [12 Topics [111min], 4 readings, 1 quiz, 3 programming assignments, 1 app item]

E1.01 Why Sequence Models? [2]

E1.02 Notation [8]

E1.03 Recurrent Neural Network Model [16]

E1.04 Backpropagation Through Time [6]

E1.05 Different Types of RNNs [9]

E1.06 Language Model and Sequence Generation [12]

E1.07 Sampling Novel Sequences [8]

E1.08 Vanishing Gradients with RNNs [6]

E1.09 Gated Recurrent Unit (GRU) [16]

E1.10 Long Short Term Memory (LSTM) [9]

E1.11 Bidirectional RNN [8]

E1.12 Deep RNNs [5]

**readings:**

Recurrent Neural Networks [50]

**prj:**

Building your Recurrent Neural Network - Step by Step [180]

Dinosaur Island-Character-Level Language Modeling [180]

Jazz Improvisation with LSTM [180]

* **E2. Natural Language Processing & Word Embeddings**
* objectives:
* NLP with deep learning is a powerful combination.
* Using "word vector" representations and embedding layers, train RNNs with outstanding performance across a wide variety of applications, including:
* sentiment analysis,
* named entity recognition and
* neural machine translation.
* Topics: [10 Topics [98min], 2 readings, 1 quiz, 2 programming assignments]

E2.01 Word Representation [10]

E2.02 Using Word Embeddings [9]

E2.03 Properties of Word Embeddings [11]

E2.04 Embedding Matrix [3]

E2.05 Learning Word Embeddings [10]

E2.06 Word2Vec [12]

E2.07 Negative Sampling [11]

E2.08 GloVe Word Vectors [11]

E2.09 Sentiment Classification [7]

E2.10 Debiasing Word Embeddings [11]

**readings:**

Natural Language Processing & Word Embeddings [50]

**prj:**

Operations on Word Vectors - Debiasing [180]

Emojify [180]

* **E3. Sequence Models & Attention Mechanism**
* objectives:
* Augment your sequence models using an attention mechanism,
* It's an algorithm that helps your model decide 'where to focus its attention' given a sequence of inputs.
* Then, explore speech recognition and how to deal with audio data.
* Topics: [10 Topics [98min], 3 readings, 1 quiz, 2 programming assignments]

E3.01 Basic Models [6]

E3.02 Picking the Most Likely Sentence [8]

E3.03 Beam Search [11]

E3.04 Refinements to Beam Search [10]

E3.05 Error Analysis in Beam Search [9]

E3.06 Bleu Score (Optional) [15]

E3.07 Attention Model Intuition [9]

E3.08 Attention Model [12]

E3.09 Speech Recognition [8]

E3.10 Trigger Word Detection [4]

**readings:**

Sequence Models & Attention Mechanism [50]

**prj:**

Neural Machine Translation [180]

Trigger Word Detection [180]

* **E4. Transformer Network**
* objectives:
* Combine Transformer Network with Attention mechanism
* Topics: [5 Topics [42min], 5 readings, 1 quiz, 1 programming assignment, 3 ungraded labs]

E4.01 Transformer Network Intuition [5]

E4.02 Self-Attention [11]

E4.03 Multi-Head Attention [8]

E4.04 Transformer Network [14]

E4.05 Conclusion and Thank You! [2]

**readings:**

Transformers [50]

**prj:**

Transformers Architecture with TensorFlow [180]

Transformer Pre-processing [60]

Transformer Network Application: Named-Entity Recognition [60]

Transformer Network Application: Question Answering [60]