

T81-558: Applications of Deep Neural Networks

Module 1: Python Preliminaries

- Instructor: [Jeff Heaton](#), McKelvey School of Engineering, [Washington University in St. Louis](#)
- For more information visit the [class website](#).

Module 1 Material

- **Part 1.1: Course Overview** [\[Video\]](#) [\[Notebook\]](#)
- Part 1.2: Introduction to Python [\[Video\]](#) [\[Notebook\]](#)
- Part 1.3: Python Lists, Dictionaries, Sets and JSON [\[Video\]](#) [\[Notebook\]](#)
- Part 1.4: File Handling [\[Video\]](#) [\[Notebook\]](#)
- Part 1.5: Functions, Lambdas, and Map/Reduce [\[Video\]](#) [\[Notebook\]](#)

Watch one (or more) of these depending on how you want to setup your Python TensorFlow environment:

- [How to Submit a Module Assignment locally](#)
- [How to Use Google CoLab and Submit Assignment](#)
- [Installing TensorFlow, Keras, and Python in Windows CPU or GPU](#)
- [Installing TensorFlow, Keras, and Python for an Intel Mac](#)
- [Installing TensorFlow, Keras, and Python for an M1 Mac](#)

Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

```
In [1]: try:
        from google.colab import drive
        %tensorflow_version 2.x
        COLAB = True
        print("Note: using Google CoLab")
    except:
        print("Note: not using Google CoLab")
        COLAB = False
```

Part 1.1: Overview

Deep learning is a group of exciting new technologies for neural networks. [\[Cite:lecun2015deep\]](#) By using a combination of advanced training techniques neural network architectural components, it is now possible to train neural networks of much greater complexity. This book introduces the reader to deep neural networks, regularization units (ReLU), convolution neural networks, and recurrent neural networks. High-performance computing (HPC) aspects demonstrate how deep learning can be leveraged both on graphical processing units (GPUs), as well as grids. Deep learning allows a model to learn hierarchies of information in a way that is similar to the function of the human brain. The focus is primarily upon the application of deep learning, with some introduction to the mathematical foundations of deep learning. Readers will make use of the Python programming language to architect a deep learning model for several real-world data sets and interpret the results of these networks.

[\[Cite:goodfellow2016deep\]](#)

Origins of Deep Learning

Neural networks are one of the earliest examples of a machine learning model. Neural networks were initially introduced in the 1940s and have risen and fallen several times in popularity. The current generation of deep learning began in 2006 with an improved training algorithm by Geoffrey Hinton.

[\[Cite:hinton2006fast\]](#) This technique finally allowed neural networks with many layers (deep neural networks) to be efficiently trained. Four researchers have contributed significantly to the development of neural networks. They have consistently pushed neural network research, both through the ups and downs. These four luminaries are shown in Figure 1.LUM.

Figure 1.LUM: Neural Network Luminaries



The current luminaries of artificial neural network (ANN) research and ultimately deep learning, in order as appearing in the figure:

- [Yann LeCun](#), Facebook and New York University - Optical character recognition and computer vision using convolutional neural networks (CNN). The founding father of convolutional nets.
- [Geoffrey Hinton](#), Google and University of Toronto. Extensive work on neural networks. Creator of deep learning and early adapter/creator of backpropagation for neural networks.
- [Yoshua Bengio](#), University of Montreal and Botler AI. Extensive research into deep learning, neural networks, and machine learning.
- [Andrew Ng](#), Baidu and Stanford University. Extensive research into deep learning, neural networks, and application to robotics.

Geoffrey Hinton, Yann LeCun, and Yoshua Bengio won the [Turing Award](#) for their contributions to deep learning.

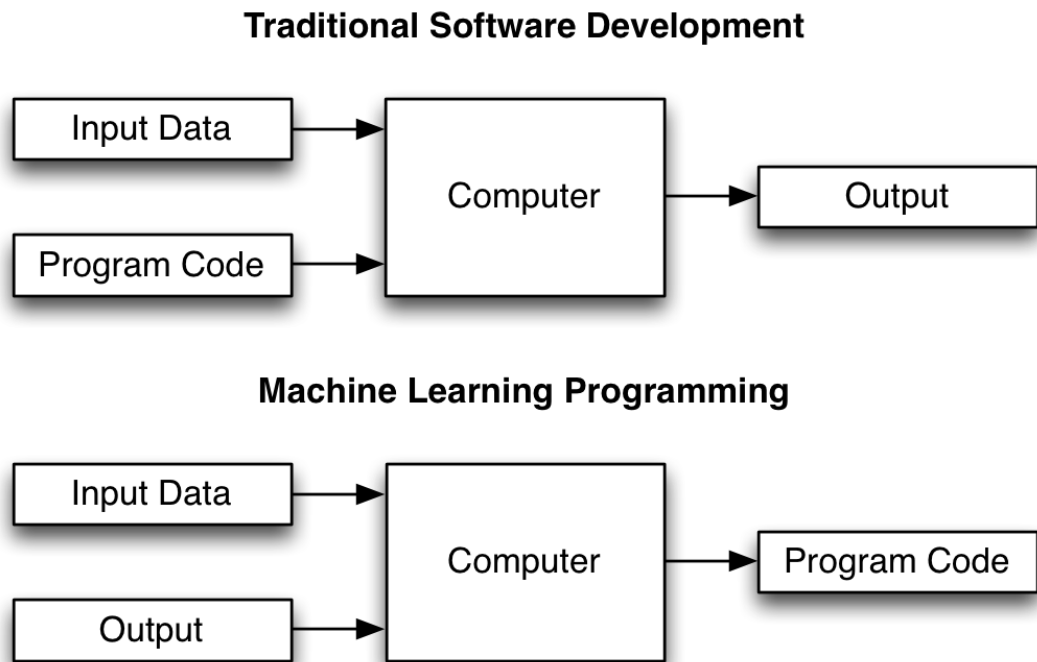
What is Deep Learning

The focus of this book is deep learning, which is a prevalent type of machine learning that builds upon the original neural networks popularized in the 1980s. There is very little difference between how a deep neural network is calculated compared with the first neural network. We've always been able to create and calculate deep neural networks. A deep neural network is nothing more than a neural network with many layers. While we've always been able to

create/calculate deep neural networks, we've lacked an effective means of training them. Deep learning provides an efficient means to train deep neural networks.

If deep learning is a type of machine learning, this begs the question, "What is machine learning?" Figure 1.ML-DEV illustrates how machine learning differs from traditional software development.

Figure 1.ML-DEV: ML vs Traditional Software Development



- **Traditional Software Development** - Programmers create programs that specify how to transform input into the desired output.
- **Machine Learning** - Programmers create models that can learn to produce the desired output for given input. This learning fills the traditional role of the computer program.

Researchers have applied machine learning to many different areas. This class explores three specific domains for the application of deep neural networks, as illustrated in Figure 1.ML-DOM.

Figure 1. ML-DOM: Application of Machine Learning



- **Computer Vision** - The use of machine learning to detect patterns in visual data. For example, is an image a picture of a cat or a dog.
- **Tabular Data** - Several named input values allow the neural network to predict another named value that becomes the output. For example, we are using four measurements of iris flowers to predict the species. This type of data is often called tabular data.
- **Natural Language Processing (NLP)** - Deep learning transformers have revolutionized NLP, allowing text sequences to generate more text, images, or classifications.
- **Reinforcement Learning** - Reinforcement learning trains a neural network to choose ongoing actions so that the algorithm rewards the neural network for optimally completing a task.
- **Time Series** - The use of machine learning to detect patterns in time. Typical time series applications are financial applications, speech recognition, and even natural language processing (NLP).
- **Generative Models** - Neural networks can learn to produce new original synthetic data from input. We will examine StyleGAN, which learns to create new images similar to those it saw during training.

Regression, Classification and Beyond

Machine learning research looks at problems in broad terms of supervised and unsupervised learning. Supervised learning occurs when you know the correct outcome for each item in the training set. On the other hand, unsupervised learning utilizes training sets where no correct outcome is known. Deep learning supports both supervised and unsupervised learning; however, it also adds reinforcement and adversarial learning. Reinforcement learning teaches the neural network to carry out actions based on an environment. Adversarial learning pits two neural networks against each other to learn when the data provides no correct outcomes. Researchers continue to add new deep learning training techniques.

Machine learning practitioners usually divide supervised learning into classification and regression. Classification networks might accept financial data and classify the investment risk as risk or safe. Similarly, a regression neural network outputs a number and might take the same data and return a risk score. Additionally, neural networks can output multiple regression and classification scores simultaneously.

One of the most powerful aspects of neural networks is that the input and output of a neural network can be of many different types, such as:

- An image
- A series of numbers that could represent text, audio, or another time series
- A regression number
- A classification class

Why Deep Learning?

For tabular data, neural networks often do not perform significantly better than different than other models, such as:

- [Support Vector Machines](#)
- [Random Forests](#)
- [Gradient Boosted Machines](#)

Like these other models, neural networks can perform both **classification** and **regression**. When applied to relatively low-dimensional tabular data tasks, deep neural networks do not necessarily add significant accuracy over other model types. However, most state-of-the-art solutions depend on deep neural networks for images, video, text, and audio data.

Python for Deep Learning

We will utilize the Python 3.x programming language for this book. Python has some of the widest support for deep learning as a programming language. The two most popular frameworks for deep learning in Python are:

- [TensorFlow/Keras](#) (Google)
- [PyTorch](#) (Facebook)

This book focuses primarily upon Keras, with some applications in PyTorch. For many tasks, we will utilize Keras directly. We will utilize third-party libraries for higher-level tasks, such as reinforcement learning, generative adversarial neural networks, and others. These third-party libraries may internally make use of

either PyTorch or Keras. I chose these libraries based on popularity and application, not whether they used PyTorch or Keras.

To successfully use this book, you must be able to compile and execute Python code that makes use of TensorFlow for deep learning. There are two options for you to accomplish this:

- Install Python, TensorFlow and some IDE (Jupyter, TensorFlow, and others).
- Use [Google CoLab](#) in the cloud, with free GPU access.

If you look at this notebook on Github, near the top of the document, there are links to videos that describe how to use Google CoLab. There are also videos explaining how to install Python on your local computer. The following sections take you through the process of installing Python on your local computer. This process is essentially the same on Windows, Linux, or Mac. For specific OS instructions, refer to one of the tutorial YouTube videos earlier in this document.

To install Python on your computer, complete the following instructions:

- [Installing Python and TensorFlow - Windows/Linux](#)
- [Installing Python and TensorFlow - Mac Intel](#)
- [Installing Python and TensorFlow - Mac M1](#)

Check your Python Installation

Once you've installed Python, you can utilize the following code to check your Python and library versions. If you have a GPU, you can also check to see that Keras recognize it.

```
In [3]: # What version of Python do you have?
import sys

import tensorflow.keras
import pandas as pd
import sklearn as sk
import tensorflow as tf

check_gpu = len(tf.config.list_physical_devices('GPU'))>0

print(f"Tensor Flow Version: {tf.__version__}")
print(f"Keras Version: {tensorflow.keras.__version__}")
print()
print(f"Python {sys.version}")
print(f"Pandas {pd.__version__}")
print(f"Scikit-Learn {sk.__version__}")
print("GPU is", "available" if check_gpu \
      else "NOT AVAILABLE")
```

Tensor Flow Version: 2.1.0

Keras Version: 2.2.4-tf

Python 3.7.16 (default, Jan 17 2023, 22:20:44)

[GCC 11.2.0]

Pandas 1.3.5

Scikit-Learn 1.0.2

GPU is NOT AVAILABLE

Module 1 Assignment

You can find the first assignment here: [assignment 1](#)