**Tensorflow Deep Learning Solutions for Images**

**Section 1: Machine Learning Toolkit**

**1.1 The Course Overview**

This video provides an overview of the entire course.

**1.2 Installing Docker**

How to download and install Docker on Linux and Windows?

* + Learn how to download the Docker installer
  + Explore the installation process of Docker on Windows
  + Learn the script Docker installation process on Linux

**1.3 The Machine Learning Dockerfile**

How to write a Dockerfile that configures the packages to get Keras running in a Docker container?

* + Clone the video source code
  + Code the base image and needed Linux packages
  + Expose ports and volumes to allow you container to talk to the world

**1.4 Sharing Data**

How to configure Docker security settings on Windows to allow access to Jupyter notebooks?

* + Configure needed security settings in Docker
  + Test security to validate volume sharing
  + Run the machine learning Dockerfile

**1.5 Machine Learning REST Service**

How to expose Jupyter as a machine learning REST service from your Docker container?

* + Run the Docker container
  + Use the exposed service endpoint URL
  + Verify Keras installation and loading

**Section 2: Image Data**

**2.1 MNIST Digits**

How to encode the images for machine learning?

* + Download the MNIST digits
  + Treat images as numbers
  + Visualize the encoded images

**2.2 Tensors: Just Multidimensional Arrays**

In this video, we will understand what a tensor is and we will Learn how to create them.

* + Use NumPy multidimensional arrays to store tensors
  + Set the tensor shapes
  + Set the tensor datatypes

**2.3 Turning Images into Tensors**

This video explains how to encode images into tensors, a needed step for machine learning.

* + Encode images as floating point
  + Understand samples and their placement in tensors
  + Normalize data to facilitate learning algorithms

**2.4 Turning Categories into Tensors**

How to encode categories into tensors?

* + Understand classes and discreet predictions for categories
  + Use one-hot encoding to create tensors
  + Visualize one hot encoding as a bitmap

**Section 3: Classical Neural Network**

**3.1 Classical/Dense Neural Network**

What is the structure of a neural network?

* + Use code to draw a dense neural network
  + Use code to draw a neural network designed for MNIST digits
  + Understand the relationship between tensors and networks

**3.2 Activation and Non Linearity**

What is non-linearity and why does it matter for neural networks?

* + Use nonlinearity in networks
  + Use the sigmoid in networks
  + Use the ReLU in networks

**3.3 Softmax**

What is softmax, and where do we use it in networks?

* + Understand the softmax function
  + Use softmax to predict classes or categories
  + Use softmax to predict probabilities

**3.4 Training and Testing Data**

What are training and testing data sets?

* + Loading data into normalized data sets
  + Use one-hot encoding to create training and testing outputs
  + Understand the use of training versus testing data sets

**3.5 Dropout and Flatten**

What are dropout and flatten layers and when do we use them in networks?

* + Use Keras functional model and input layers
  + Use the dropout layer to improve learning
  + Use the flatten layer to make layer and tensor shapes compatible

**3.6 Solvers**

What are solvers and what is their role in networks?

* + Understand learning and optimization
  + Compile and fitting a model to data with a solver
  + Understand loss and accuracy to evaluate a solver’s fitting of data

**3.7 Hyperparameters**

In this video, we will learn What are hyperparameters and parameters and we will also see a difference between them.

* + Understand trainable parameters in networks
  + Understand non- trainable parameters in networks
  + Understand hyperparameters in network layers

**3.8 Grid Search**

What is grid search and when do we use it in training networks?

* + Understand the difference between optimization with a solver and grid search
  + Configure a hyperparameter grid
  + Perform grid search and report outcomes

**Section 4: A Convolutional Neural Network**

**4.1 Convolutions**

What are convolutions and why would we use them in networks?

* + Use code to draw convolutions
  + Understand convolutions in two dimensions
  + Understand the benefits of convolutions for performance

**4.2 Pooling**

What is pooling, and why do we use it work convolutions?

* + Understand one dimensional pooling
  + Understand two dimensional pooling
  + Understand pooling across image channels

**4.3 Convolutional Neural Network**

How do we create a convolutional network to learn to recognize images?

* + Understand MNIST digits and image channels
  + Construct a convolutional network
  + Train and predict results with a convolutional network

**4.4 Deep Neural Network**

What is a deep neural network compared to a classical or convolutional network?

* + Understand the role of GPU in deep network performance
  + Build a multiple block, deep network
  + Train and predict results with a deep network

**Section 5: An Image Classification Server**

**5.1 REST API Definition**

How do we define an API for use with machine learning?

* + Clone the source code for the REST service
  + Configure a declarative API with Swagger/ OpenAPI yaml
  + Code an API handler that can receive an image file and classify it with keras

**5.2 Trained Models in Docker Containers**

How do we create a deployable REST service with a trained Keras model?

* + Understand the need for a trained model file along with the API service code
  + Code a Dockerfile that will build a Python runtime along with a pre-trained Keras model
  + Build the Docker container from the command line

**5.3 Making Predictions**

How do we use our API to make digit predictions?

* + Run the Docker container mapping ports to create a service
  + Use the Swagger/ Connexion built in UI to test the API
  + Use the API from the command line with curl