Training Outline

Part 1: Introduction and Background (1 hr)

Background and Intro

Lab #1

Big Data

* Sign in: <https://community.cloud.databricks.com/login.html>
* Clusters
  + Display, Create, Edit, Start/Stop
* Quickstart
  + Click on Explore the Quickstart Tutorial (center left)
  + Launch Cluster
  + Step through each cell
* Datasets
  + <https://docs.azuredatabricks.net/getting-started/databricks-datasets.html>
* Word Count
  + <https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/8599738367597028/2067276070342403/3601578643761083/latest.html>
  + Create a new cell
  + Type in: %fs ls “dbfs:/databricks-datasets/cs100/lab1/data-001”
  + Use the Shakespeare text file and re-run

IoT

* Demonstration of Remote Monitoring
* Go To Azure
  + Find Applications/Remote Monitoring
  + Click on the URL
* Explore the dashboard, notice critical warnings
* Notice the telemetry for Chiller-02.0
* Check the alarm status, and acknowledge
* Go to the failing chiller and execute a job to relieve pressure
* Check status of the chillers

Blockchain

* Navigate to Hyperledger Composer Playground
* Create a network and select perishable goods
* Connect to the network
* Examine the network including model, script, access control
* Go to Test
* Execute the setup transaction
* Add a temperature reading of < 1 to the shipment
* Execute a shipment received transaction
* Examine the payments

AI/ML

* Deep Traffic - <https://selfdrivingcars.mit.edu/deeptraffic/>
* Familiarize yourself with the layout
* Scan through the neural network configuration code
* Modify the network to include 2 new layers
* Modify the training parameters – keep exploration vs exploitation in mind
  + Learning Rate
  + Discount
* Train your network
* Test your network
* How did you do?

Part 2: Tools and Techniques (1 hr)

Background Slides

Lab #2

* Using Databricks
* SQL – Population vs. Price
  + - <https://docs.azuredatabricks.net/getting-started/spark/dataframes.html#load-sample-data>
  + Streaming Data
    - <https://docs.azuredatabricks.net/getting-started/spark/streaming.html#notebook-stream>
  + Machine Learning
    - <https://docs.azuredatabricks.net/getting-started/spark/machine-learning.html#ml-notebook>
    - <https://docs.azuredatabricks.net/_static/notebooks/gbt-regression.html>
  + Graph Analysis – SF Bike Trips
    - <https://docs.azuredatabricks.net/spark/latest/graph-analysis/graphframes/graph-analysis-tutorial.html>
* ArcGIS/Python
  + Open ArcGIS Pro
  + Load the Deep Objects project
  + Walk through the demo
* Azure IoT
  + Create an IoT Central Application Resource in Azure
  + Launch the deployed application
  + Create a directory for your project
  + Install necessary packages
  + Bring up the application code (javascript) in Visual Studio Code
  + Modify the sensor parameters
  + Start the code with: “node <filename>.js”
  + Check out the analytics display in the web application
  + Stream data through event hub
  + Connect Databricks to your data stream
* Ethereum Blockchain
  + Demonstration of Remix
  + Open <https://remix.ethereum.org/#optimize=false&version=soljson-v0.4.21+commit.dfe3193c.js&evmVersion=null>
  + Click on the icon “Deploy and Run Transactions” on left hand side
  + Start Ganache, set environment to Web3 provider, localhost:7545
  + Provide string name and Deploy referendum contract
  + Open up the deployed contract by clicking left arrow (should view contract methods)

Part 3: Applying ML to Object Detection (1 hr) –

Background Slides

Lab #3

* Single Node Object Detection Using OpenCV, Tensorflow, Keras, and Python
  + Open up the CNN file and the identify\_parking\_spots file in Visual Code
  + Open a command line window
  + Set your python env
    - Train the model
    - Run the model
* Multi Node Object Detection w/ Databricks/Spark

Summary

Pulling it all together

What are the near-term opportunities?

How can you leverage this technology in your own work?

Resources

* Databricks/Spark Tutorials
* Azure Deployed Resource Tutorials
* OpenCV tutorials
* Jason Brownlee/Machine Learning Mastery
* Randall Sanderson, 3 Blue, 1 Brown
* Visualgo.net
* R2d3.us
* Playground.tensorflow.org
* <http://arogozhnikov.github.io/2016/04/28/demonstrations-for-ml-courses.html>
* <https://ethernaut.openzeppelin.com/>
* <https://blog.valohai.com/reinforcement-learning-tutorial-part-1-q-learning>