**RISHIKA SWARNKAR**

*5 Sugar Maple, Apt 10, St. Charles, MO 63303* (573) 200-2495

[rishikaswarnkar@gmail.com](mailto:rishikaswarnkar@gmail.com)

Linkedin.com/in/rishikaswarnkar

Github.com/rishikaswarnkar

**Projects**

**Neural Network**

**An Artificial Neural Network is computational model inspired by biological working of neurons. TensorFlow and Keras are largest deep learning libraries that power industries like Airbnb, Google etc. for Artificial Intelligence.**

* **Neural Network Classification Model from Scratch**
* The network performs supervsied learning tasks using labelled training data (X- Feature Array, Y- labels). The model has dense layers and activation functions as parameters. It supports arbitary number of inputs, hidden layers, nodes per layer, and classes. The model returns an estimated probabilities of the observation being in the classes, predicted class for each observation and a score function that returns model’s loss and accuracy.
* **Training the Neural Network**
* Training the weights in the model by back propagating from output layer using gradient decent algorithm. The loss function is decreased as we propagate through the dense layers in each epoch (one epoch is one complete cycle of training entire data set while calculating loss and accuracy on each step).
* **Convolution Neural Network,** Dataset: CIFAR – 10 Training set: 50,000 images belonging to 100 classes; Validation Set: 10,000 images belonging to 200 classes.

The convolution neural network model is compiled with learning rate of 0.001. After 120 epochs the model reaches an accuracy of 97 % on training and 39 % on validation data on version one. The model reports the *confusion matrix and classification report* of predicted validation set. A running graph of training and validation accuracy and loss is displayed with each epoch. The trained model is saved as cifar100\_v1\_model.h5

* **Vehicle Classification,** Dataset: 600 vehicle images into 12 classes
* A ConvNet Model with **i**mage preprocessing to be training on vehicle data set to classify the data into 12 vehicle classes like ‘convertible’, ‘go-cart’, etc. mentioned in classes.txt. The model reaches an accuracy of 100% on training and 55.5% on validation sets. The graph of model’s loss and accuracy per epoch is displayed. The model reports the *confusion matrix and classification report* of predicted validation set.
* **Transfer Learning**, Dataset - CIFAR 100: Training set: 100,000 images belonging to 200 classes; Validation Set: 10,000 images belonging to 200 classes.

Designing a ConvNet Architecture by transferring learning from VGG16 model on ImagNet. 15 million parameters are trained for this model. A Bar Graph displays top 5 classes that are correctly classified and the probabilities of being correctly classified by the model. The model also displays loss and accuracy graph per epoch.

* **Breast Cancer Dataset**: Designing a neural network to train on Breast cancer dataset with 30 features as input to output the probability of tumor being ‘Benign’ or ‘malignant’. 93.4 % validation accuracy is achieved on this model.
* **Convolution Filters**: Designing a convolution filter to detect various features in the images like straight lines, curves, red channels, blue channels, greens channels etc.

**Machine learning:** Designing models that performs a specific task effectively without using explicit instructions, relying on patterns and inference instead.

* **Building Logistic Regression Model from Scratch in Python** It supports an arbitrary number of features inputs and will accept un-encoded class labels. The predict method has a threshold for classification, which decides the predicted class, based on estimated probabilities of each observation. The log- likelihood and accuracy of this model is calculated. We also calculate the precision and recall value for each class.
* **Tuning Model Hyper-parameters:** The model has 6 feature array (4 numerical variables and 2 categorical variables) and 4 labelled classes. The features are scaled and encoded. The data is modelled for Linear Regression, K nearest neighbor, Decision tress and Random Forest algorithms with a validation accuracy of 29 %, 89.6 %, 86% and 93.7 % respectively for tuned parameters.
* **Decision Tree Classifier from Scratch:** Gini score is used as impurity matrix**.**
* **Random Forest Classifier:** An ensemble of decision trees.

**Data Structures (C++):**

* Mailbox List Update: C++, **Linked List**
* Infix to Postfix and Evaluation: C++, **Stacks using Pointers**
* Customer Processing Line: C++, **Queues**
* Sorting Inventory: C++, **Exchange Sort, Shell Sort, Quick Sort**
* Store and Print Inventory: C++, **Binary Search tree [In order Traversals]**

**Android applications:**

* **Ski Trip Reservation**: Android Studio: Java + XML
* **Financial Calculator**: Android Studio: Java + XML
* **Autumn Playlist**: Android Studio: Java + XML

**Full Stack Software – Inventory Management Solution:**

* Inventory Management Solution: **PHP, HTML, MySQL**
* TO- DO List Web Application : **Node Package Manager & MEAN stack**

**SQL Server Integration Service**

Designed and implemented a data warehouse to Extract, Transform data and Load data to configure and deploy SSIS solutions to build data quality solutions. Implemented a package with Flat File Connection manager, OLE DB Connection Manager, Data Flow Task, configured the flat file source and the lookup transformations. Looping, Logging, Error Flow Redirection for the package. Used parameters with the Project Deployment Model. Prepared and deployed the Bundle.

**SQL Server Reporting Service**

Learned how to create a drillthrough report with parameters and a filter using the ReportViewer control.

**Major Classes:**

* Neural Network - Special Topics |DSCI -39001-11 - Spring 2019
* Machine Learning | DSCI – 35600-11 -Spring 2019
* Statistics | MTH - 24100-22 - Spring 2019
* Data Structures and Algorithm | CS360
* System Analysis and Design | CS200
* Android Application Development| CS160
* Computer science II: Data Structure
* Data structures | CS260
* Social Media Marketing CPM 230
* Calculus II