Cloud Based Machine Learning Portable Inference Models Using NeoPulse™ AI Studio

Griffin Wu & James Guo, Department of Electrical Engineering, University of Washington

Industry Mentor: Rajeev Dutt(CEO) & Dominique Davis(AI Researcher)

Sponsored by UW EE department & DimensionalMecahnics Inc.

Faculty Adviser: Prof. Payman Arabshahi, Prof. Blake Hannaford, Shruti Misra (TA) & Douglas G.Smith (TA)

DimensionalMechanics, Inc

Introduction

What is NeoPulse AI Studio

Neopulse AI Studio is a complete framework for training, querying, and managing AI models.

- Runs on AWS EC2 Instance with Linux environment.
- Uses the NeoPulse Modeling Language (NML) designed for machine learning.
- Features 'auto' keyword to different levels' automatic model training processes.

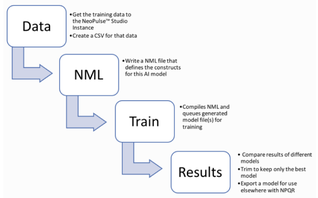

Figure 1. High Level Training Workflow.

Figure 2. Different Levels of Auto Modes and Training Process.

Full Auto Example: IMDB Movie Review Sentiment Classification

Develop an algorithm to classify movie reviews on IMDB as positive or negative

- Data set used: ~25,000 labeled movie Reviews on IMDB
- Uses Neopulse Studio's full auto architecture function.

```
oracle("mode") = "classification"

source:
    bind = "/DM-Dash/examples/sentiment/data.csv" ;
    input:
        x ~ from "Review"
            -> text: [200]
            -> TextDataGenerator: [no_words = 20000] ;
    output:
        y ~ from "Label"
            -> flat: [2]
            -> FlatDataGenerator: [] ;
    params:
        validation_split = 0.5 ;

architecture:
    input:  x ~ text: [200] ;
    output: y ~ flat: [2] ;

    x -> auto -> y ;

train:
    compile:
        optimizer = auto,
        loss = auto,
        metrics = ['accuracy'] ;
    run:
        epochs = 4 ;
        dashboard: ;
```

Figure 3. NML Training Script.

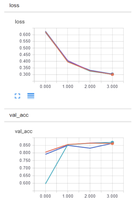

Figure 4. Result Visualization.

```
ubuntu@ip-172-31-57-215:~$ neopulse list sentiment_auto
MODEL            ITER        STATUS                METRICS                2_0_POOL_LENGTH 2_0_LSTM_SIZE   2_0_FILTER_LENGTH 2_0_NB_FILTERS_CN 2_0_BI_LSTM
_SIZE OUT_ACTIVATION  COMPILE_10_0        COMPILE_9_0     INP_SIZE_E      EMBED_DIMENSION
SyRqdjqym-0      4          TRAINING_COMPLETED val_loss,val_acc,loss,acc 8              128             7                 331               104
        'softmax'       'binary_crossentropy' 'adadelta'      361T            77
SyRqdjqym-1      4          TRAINING_COMPLETED val_loss,val_acc,loss,acc 8              128             7                 331               104
        'softmax'       'binary_crossentropy' 'adadelta'      361T            77
SyRqdjqym-2      4          TRAINING_COMPLETED val_loss,val_acc,loss,acc 4              128             7                 331               104
        'softmax'       'binary_crossentropy' 'adadelta'      361T            77
SyRqdjqym-3      4          TRAINING_COMPLETED val_loss,val_acc,loss,acc 8              128             7                 331               104
        'softmax'       'binary_crossentropy' 'adadelta'      361T            77
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:247] PoolAllocator: After 4224 get requests, put_count=10078 evicted_count=5000 eviction_rate=0.49613
and unsatisfied allocation rate=0
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:247] PoolAllocator: After 1407 get requests, put_count=3543 evicted_count=1000 eviction_rate=0.282247
and unsatisfied allocation rate=0
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:247] PoolAllocator: After 4545424 get requests, put_count=4546781 evicted_count=7000 eviction_rate=0.
00153955 and unsatisfied allocation rate=0.00154397
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:247] PoolAllocator: After 10986930 get requests, put_count=10980311 evicted_count=17000 eviction_rate
=0.0015471 and unsatisfied allocation rate=0.00154675
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:247] PoolAllocator: After 18410091 get requests, put_count=18411446 evicted_count=27000 eviction_rate
=0.001466446 and unsatisfied allocation rate=0.00146767
I tensorflow/core/common_runtime/gpu/pool_allocator.cc:247] PoolAllocator: After 24557315 get requests, put_count=24558668 evicted_count=37000 eviction_rate
=0.0015066 and unsatisfied allocation rate=0.00150758
```

Figure 5. Generated parameters from full auto ('AI building AI') mode.

Application Cases

Input Data Types

Neopulse AI Studio supports a wide array of different input data types, including:

- Vector, for data and statistics applications.
- Text, for natural language processing applications.
- Image and video, for computer vision applications.
- Audio, for audio signal processing applications.

Problem Types: Classification, Regression

Classification

Output of algorithm being discrete, integer values.

Example:

Dog Breed Classification

- Input Source: Dog breed image data sets from ImageNet, ~120 dog breeds
- Can use full auto function to achieve >99% accuracy.

Figure 6. Dog Breed Classification [1]

Figure 7. Face Localization [2]

Regression

Output of algorithm being continuous data values.

Examples:

Stock Market Volatility Predction

- Input Source: Stock Market information from 449 companies since 2000.
- Inapplicable to Black Swan Events problem.
- Volatility(Financial): An indicator of how much stock value fluctuates every day, using (Highest point – Lowest point)/Highest point or standard deveation to approximate.
- Used values of single/several days as combinations of data.
- Implemented linear regression, time-series LSTM network and full auto mode.
- Applied the best result method to real volatility used by Clients in the end.

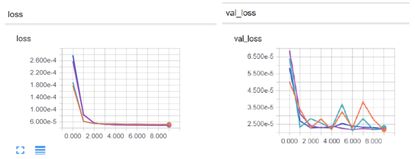

Figure 8. Results for Stock Volatility Prediction

Face Localization

- Input Source: UCI DriFace: 606 samples from 4 drivers with several facial features and pictures taken from different angles.
- Object detection algorithms: Edge Detection, Sliding window approach, Regional Proposal.
- Localized and Found bounding boxes around face, left/right eye, nose and mouth positions.
- Implemented Recurrent Neural Network algorithms and full auto mode.

Case Study: VGG Net

What is VGG Net?

2014 ImageNet Large Scale Visual Recognition Challenge(ILSVRC) Winner, Computer Vision Milestone

ILSVRC: Image Classification Task

Develop an algorithm that classifies photographs into different categories (e.g. Dog, Cat, Car, Person).

- Data set given by competition has ~150,000 photographs and 1000 image categories.
- Due to time and budget restrictions, we randomly chose 20 categories from the data set to train our algorithms.

Technical Approach

Recreate VGG Net Architecture and use Neopulse Studio's full auto mode to solve the ILSVRC classification task, compare the process with other popular machine learning(ML) platforms.

VGG Net architecture defined in Neopulse and custom functions to formulate architecture:

Define Architecture of VGG in Neopulse NML script and formulate parts of the architecture into functions for modular use.

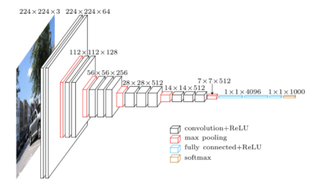

Figure 9. Macro architecture of VGGNet [3].

```
architecture:
    input: x ~ image: [shape = [224, 224], channels=3];
    output: y ~ flat: [20];

    x -> ZeroPadding2D:[padding=[1,1]]
      -> Conv2D: [64, [3,3]]
      -> Activation: ['relu']
      -> ZeroPadding2D:[padding=[1,1]]
      -> Conv2D: [64, [3,3]]
      -> Activation: ['relu']
      -> MaxPooling2D: [pool_size=[2,2], strides=[2,2]]


      -> ZeroPadding2D:[padding=[1,1]]
      -> Conv2D: [128, [3,3]]
      -> Activation: ['relu']
      -> ZeroPadding2D:[padding=[1,1]]
      -> Conv2D: [128, [3,3]]
      -> Activation: ['relu']
      -> MaxPooling2D: [pool_size=[2,2], strides=[2,2]]
```

Figure 10. Architecture of VGG in Neopulse script.

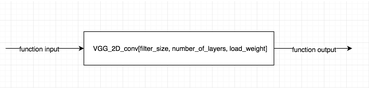

Figure 11. Creating custom architecture similar to VGG Net utilizing Neopulse fuction construct.

Comparison With Other Popular Platforms

Detailed Codes are available at https://github.com/GriffinTheOne/2018CapstoneFair.git

- Platforms used: Neopulse Studio, Caffe, TensorFlow

Platform Installation

Neopulse: Pre-installed on AWS instance, no local installation required, built-in multi-GPU support.

Caffe: Directly built from source. Has many dependencies. Overall difficult.

TensorFlow: Hard to install with early Ubuntu enviornments. Need to install dependencies for GPU support.

Training Scripts

Neopulse: Uses call_auto mode and build architecture with Keras library API.

Caffe: Architecture available since it was originally developed on Caffe. Intuitive CLI commands.

TensorFlow: Requires python and TensorFlow basic undenstanding before training.

Model Query (Application)

Neopulse: Uses Portable Inference Model (PIM) which is portable with easiness.

Caffe & TensorFlow: Required to reload all the weights into a potentially huge architecture before use.

Data Preparation

Neopulse: Describe input shape/type and validation split ratio. Can specify pre-process parameters for auto processing. ~ 10 lines of code.

Caffe & TensorFlow: Manual Selection of training/validation data, file conversion and pre-processing of images required. ~70-185 lines of code.

Hyperparameter Fine-tuing

Neopulse: Fully automated and optimized process with no extra requirement from user. Utilizes spectral optimization combined with reinforcement learning.

Caffe: Full manual adjustment required. Time consuming and requires deeper understanding of ML before use.

TensorFlow: Has somedefault parameters to start the process.

Training Results

Results of training in Neopulse, Caffe and TensorFlow are shown in the graphs below.

Neopulse

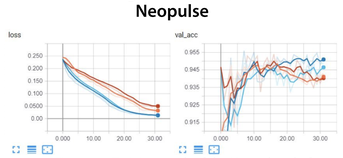

Figure 12. Training result of the Neopulse platform. ~95% Accuracy after 30 epochs.

TensorFlow

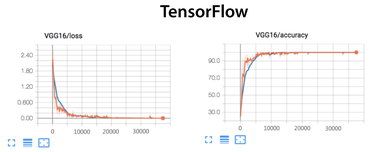

Figure 13. Training result of the Neopulse platform. ~95% Accuracy after 20,000 iterations.

Caffe

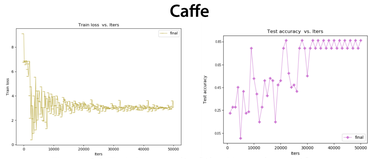

Figure 14. Training result of the Neopulse platform. ~85% Accuracy after 30,000 iterations.

Conclusions and Results

- The NeoPulse™ framework makes machine learning widely accessable and easy to use.
- Recreated VGGNet with our own dataset using several platforms and approaches, established a better understanding of Machine Learning and Neural Networks.
- Trained several machine learning models with decent results using self-designed architecture with the Neopulse Studio, demonstrating the Platform's capabilities.
- Demonstrated the capabilities of Neopulse Studio's full auto mode that makes machine learning simpler, and optimizes the algorithms beyond what can be done manually.

References:

[1] ImageNet Repository

[2] (Deborah Smith, 2018)

[3] (Leonardblier, 2016)

ELECTRICAL ENGINEERING
UNIVERSITY of WASHINGTON