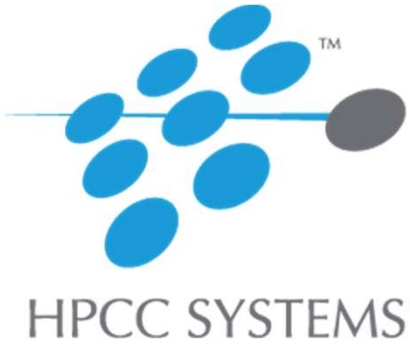


Machine Learning with HPCC

Lesson 5

Machine Learning – Generalized Neural Networks Tutorial

Part 1 of 2 – Introduction and Installation

What is a GNN?

- The GNN (Generalized Neural Network) is a bundle that provides an ECL interface to Keras and Tensorflow.
- Using GNN, an ECL developer can construct, train, and utilize arbitrarily complex Neural Networks such as Classical, Convolutional, and Recurrent networks.
- These networks can be utilized to analyze complex data such as images, video, and time-series.

What is Tensorflow?

- A Neural Networks framework that provides a single core technology that can incorporate current and future types of neural network layers.
- Allows a user to assemble these layers in arbitrary combinations to achieve their particular application goals.
- Tensors is used as the primary interface structure. Tensors can be thought of as N-dimensional arrays.
- Tensors are, therefore, an excellent generalized form for handling input and output of any dimensionality. Furthermore, tensorflow supports complex neural network layers that have higher dimensional sets of weights. Access to these weights is also provided via Tensors.

What is Keras?

- Keras is the most widely used interface for Tensorflow.
- It is an open-source neural network library written in Python.
- It defines a high-level interface for creating neural networks, layer by layer.
- Capable of running with other neural network frameworks, insulating a user from strict dependence on Tensorflow. The most commonly used combination, however, remains a Keras Interface into Tensorflow.

GNN Bundle Overview

- The Generalized Neural Network Bundle (GNN) allows the ECL programmer to combine the parallel processing power of HPCC Systems with the powerful Neural Network capabilities of Keras and Tensorflow.
- Each node in the HPCC cluster is attached to an independent Keras / Tensorflow environment.
- May use GPUs or TPUs (Tensor Processing Units)
- The bundle provides a Tensor MODULE, allowing users to efficiently encode, decode, and manipulate Tensors within ECL.
- Designed to be easy to use and familiar with Keras users as well.

Using GNN

- It is recommended you will need some experience with Keras or another NN (Neural Network) interface.
- Data is first converted to Tensors.
- GNN is straightforward from there.
- This tutorial will first illustrate how to use GNN once your data is available in Tensor form.
- We will then demonstrate how to use Tensors.

Installing the GNN Bundle

- The GNN Bundle installs like all standard HPCC Systems bundles.
- From the ClientTools BIN folder, run the following script in administrator's mode:

```
ecl bundle install https://github.com/hpcc-systems/GNN.git
```

- You must also have Python3 and Tensorflow installed on each node on your HPCC cluster.
- Remember, you must also have the ML_Core bundle installed as in previous lessons!

Installing Tensorflow and Python on HPCC

Your HPCC Systems administrator with administrator's rights can install Tensorflow and Python3, or do it yourself as shown here:

1. On Ubuntu, first refresh the APT (Advanced Package Tool) repository:

```
sudo apt update
```

2. Install Python3 if not already installed:

```
sudo apt install python3
```

3. Install pip3 (Python3 package installer) – this will take a few minutes

```
sudo apt install python3-pip
```

4. Install tensorflow for all users. This is the recommended approach, since it needs to be available to the *hpcc* user as well as the current user. The –H sudo option is necessary in order to have it installed globally:

```
sudo -H pip3 install tensorflow
```

Finally, the **setuptest.ecl** file found in the Test directory of the GNN bundle will verify that Python3 and Tensorflow are correctly installed on each Thor node.

Using Tensors

- The GNN uses a Tensor MODULE that provides a way to construct and manage N-dimensional datasets in ECL.
- These dimensions are also known as **shapes** in Tensors. A shape of [*n*,*n*] indicates a 2-dimensional Tensor. A shape of [*n*,*n*,*n*,*n*] represents a 4-dimensional Tensor. (*n* is any integer)
- In ECL, we express these shapes using a **RECORD-oriented** Tensor (*Tensdata*). This Tensor is used to represent our Training and Test data. The first term of this Tensor should always start at zero.
  For example, a 50 X 50 color image would be expressed as [0,50,50,3];
  Observation=0, Width=50, Height=50, 3 colors (RGB)
- There is also a **Rectangular Tensor** (*t_Tensor*) created in the Tensor MODULE that represents meta-data, like dimensional shape and type.
- **There is a two-step process for creating an ECL Tensor:**

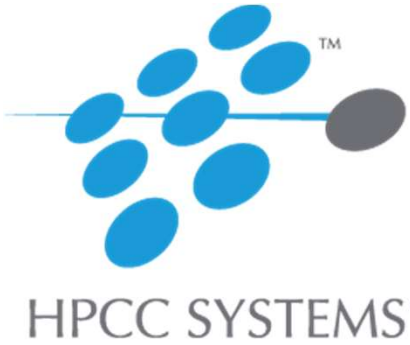
1. Create the data use to populate the TensData RECORD.
2. Create a Tensor using that data and any provided meta-data and pack it all into an efficient block oriented form that we call **slices**.

Lesson Completed!

Please continue directly to the next lesson:

GNN Tutorial (Part 2 of 2)

HPCC SYSTEMS®



Machine Learning with HPCC

Lesson 6

Machine Learning – Generalized Neural Networks Tutorial

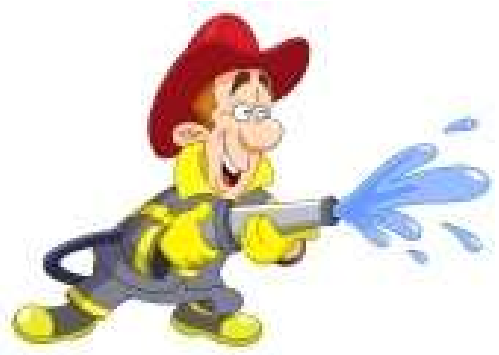
Part 2 of 2 – Examining the GNN Tutorial ECL

Tutorial

GNN Tutorial

Lesson Completed!

More lessons coming soon!



HPCC SYSTEMS®