

AIStudio 1.0

1.0

1. Introduction	2
1.1 System Requirements.....	2
1.2 Software Installation	2
1.3 Tutorial and Examples	2
1.4 Project and Files.....	2
2. Basics	4
2.1 Topography of User Interface	4
2.2 Navigating the Menus	4
3. Layout and Graphing	6
3.1 Setup Pages	6
3.2 Page Layout.....	6
3.3. Graph.....	6
3.3.1 Graph Grid, Cursor and Annotations	7
3.3.2 Graph Palette.....	7
4. Hardware (Video and Voice) Configuration	9
5. Python IDE.....	12
5.1 Creating Project.....	12
5.2 Writing Python Code.....	13
5.3 Compiling and Building Code	13
5.4 Debugging Code	13
5.4 Shell Windows	14
6. Interaction between Python and Data Views	15
7. Create and Run AI Model	16
7.1 Run AI Model.....	17
7.2 Display AI Model Data	17
8. Examples.....	18

1. Introduction

AIStudio provides an integrated development environment (IDE) for artificial intelligent (AI) application development.

Some of AIStudio capabilities include:

- Python IDE
- Data visualization
- AI model management
- Integrated with popular AI framework (tensorflow, pytorch, caffe)
- Integrated video and voice input/output

1.1 System Requirements

AIStudio runs under Windows 64bit operating systems. You also need to install Visual Studio 2015 redistribution, Python ≥ 3.6 and AI framework such as Tensorflow ≥ 1.4 and PyTorch.

1.2 Software Installation

Download the package from <https://github.com/ivapptech/AIStudio>. Run AIStudio.exe inside the Distribution\binary folder.

1.3 Tutorial and Examples

Inside AIStudio, there is an example folder. The graph examples show how to create python widget and interact with various view graphs. There are examples for common AI models such as inception, deep speech and mnist et al. These AI models are based on Tensorflow/Pytorch model zoo and revised using AIStudio library. They show how to use AIStudio to run and view AI models.

1.4 Project and Files

When you start AIStudio, it finds the previously opened project file and loads it. If the project cannot be

AIStudio 1.0

found, then it loads a default project. When you save a project, all your current work, which includes all user interface settings, is saved as an AIStudio project (*.aip). When you open the project, all your user interface settings are loaded. In addition to saving projects, you can also save each graph and layout page separately as an image file and export and import data from data sheet.

File types:

*.aip file: AIStudio project file

*.csv file: ASCII format data/text files.

*.bmp, *.jpg, *.tif files: different image format files.

2. Basics

The purpose of this section is to provide a quick software operational run-through, to provide a familiarization with operating and using AIStudio .

2.1 Topography of User Interface

Quick Access and Title Bar: The quick access toolbar displays shortcuts to commonly used command.

Main Menu Bar: The main menu bar provides a series of text commands running across the line, including: File, Edit, Model, Run, Graphing, Tools, and Help.

Solution Setup, Property and Tool Bars: Dockable bars to display model, manager script and data, show property and debug tools.

Graphing Layout: AIStudio can show many pages with flexible graph layout.

Status Bar: The status bar displays information about current run mode, run status, and other information.

2.2 Navigating the Menus

This section provides a detailed reference of AIStudio major menu items.

File Menu includes submenus to manager project files, save/load data files and images, and print images.

Edit Menu includes submenus for script editing.

Model Menu includes submenus to manage AI model.

Run Menu includes submenus to run and debug scripts.

Graphing Menu includes submenus to setup view layout and graphs.

Tools Menu includes submenus to activate different tools.

Help Menu includes submenus to get some on-line help to the operation of certain features of the software. You need to install adobe reader (PDF file) to open the help file. In addition, The 'About'

AIStudio 1.0

informs the user to the release version and date of AIStudio .

Besides the main menu bar, there are various popup menus for different view object. These popup menus can be activated by right clicking the object. The tool bar also provides quick access for some command menu functions.

3. Layout and Graphing

The graphing window includes a page tab and page panel. The page tab provides tools to add and delete pages. Each page panel has its layout with any number of graphs.

3.1 Setup Pages

AIStudio can contain many graph pages. To add, delete or rename a page, right click the page tab to bring up the page popup menu and select the corresponding menu command. The page tab also contains page adding and deleting buttons at the right side.

3.2 Page Layout

Page layout panel provides very flexible grid-style layout capabilities.

- a) There are 16 standard layouts, which provide 1 to 4 column and row selections.
- b) Menu commands to add additional column and row.
- c) Menu commands to merge grid cells.
- d) By double clicking the title bar, you can switch between a freely movable cell and a grid cell.
- e) Each cell title bar has two buttons to maximize and restore the cell size.
- f) Debug tools, model / data management are presented as dockable bars.
- g) Menu command to show pages in full screen mode.
- h) Right click the page bar to select menu commands to add, delete, rename and split pages.

3.3. Graph

A graph consists of one or many plots of data, one or many sets of axes, grids, ticks, legend, annotations and other associated graphic elements. Plots are graphic representation of data.

After the page layout has been setup, you can specify its graph type for each grid cell. A graph includes a title bar and graph image. For the title bar, besides displaying the graph image caption, it has following functions

- a) Click the title bar icon, a quick access popup dialog will be bring up to access common graph properties.
- b) Double click the title bar to switch between grid layout cell and freely-movable cell styles.
- c) It has window maximum/restore/close buttons.
- d) Right click to show context popup menu.

For every graph, you can specify its appearance which includes grid style, frame and plot area color and more. The entire area of a graph is called frame area and the area within the axes is called plot area.

3.3.1 Graph Grid, Cursor and Annotations

In a graph, grid lines can be displayed at each of the major and minor tick marks. You can specify the grid properties such as line style, color and width. AIStudio provides tools to annotate your graphs with text, lines, arrows, and other shapes. To add text label, select the text icon from the drawing toolbar and click on the desired location in the page, then edit the text in the rich text edit dialog. To draw shapes, select the respective icon on the drawing toolbar, then use mouse to draw the shape in the page window. To edit text or shape, select the edit icon in the drawing toolbar, then click the shape you want to edit or double click the text to open the text edit dialog for text editing.

AIStudio provides single cursor which is useful for locating data and measurement cursor which is useful for sizing data.

3.3.2 Graph Palette

Each image graph has a color palette attached at one side. The color palette bar shows the relation between data scale and color.

- a) Right click the data scale to bring up the scale edit dialog, where you can modify maximum and minimum scale value.
- b) Right click the color palette bar to bring up the palette selection dialog, where you can select pre-saved palette or open the palette edit dialog.
- c) The color palette is constructed from colors at several thresholds. Both the threshold value and color can be varied and the colors between threshold can uniform or gradient.
- d) Double click the color bar to show/hide the threshold vale and color edit flags. So you can

AIStudio 1.0

revise the color palette without opening any editing dialogs.

AIStudio provides many graph types: Plot graph, Image graph, Bar graph, Pie graph, Audio waveform, Camera image, Data table and AI model diagram.

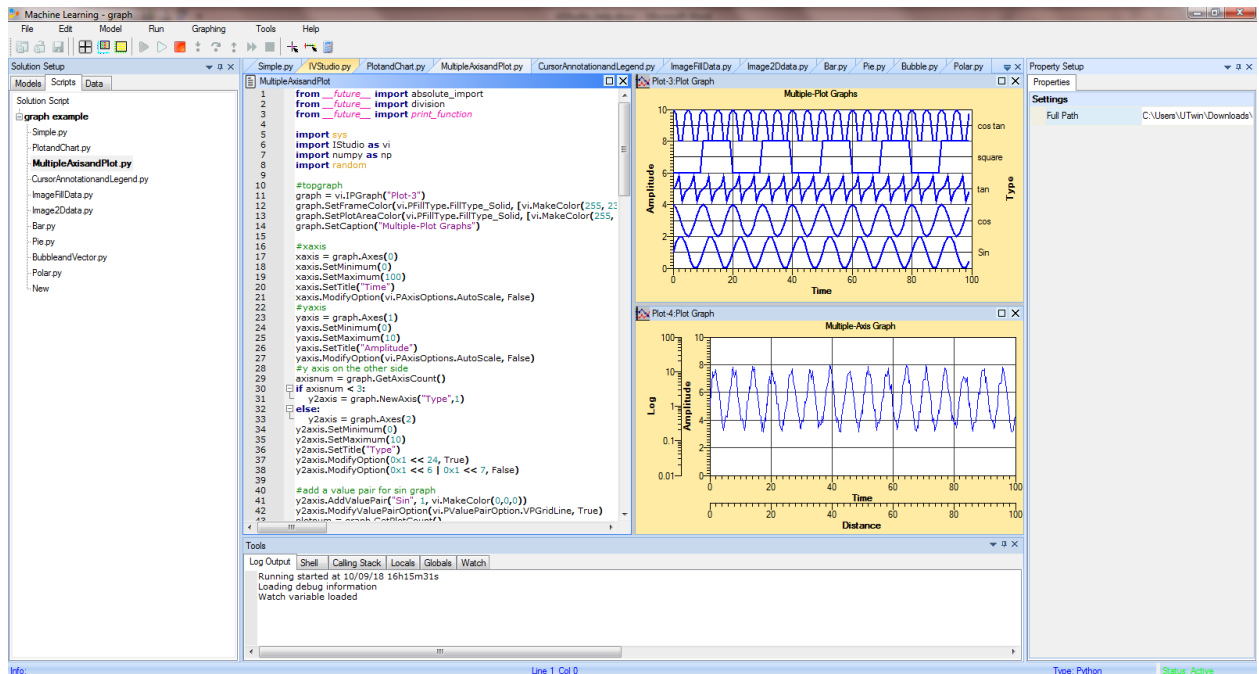


Figure 3.1: Python script with multiple plot graphs with multiple axes.

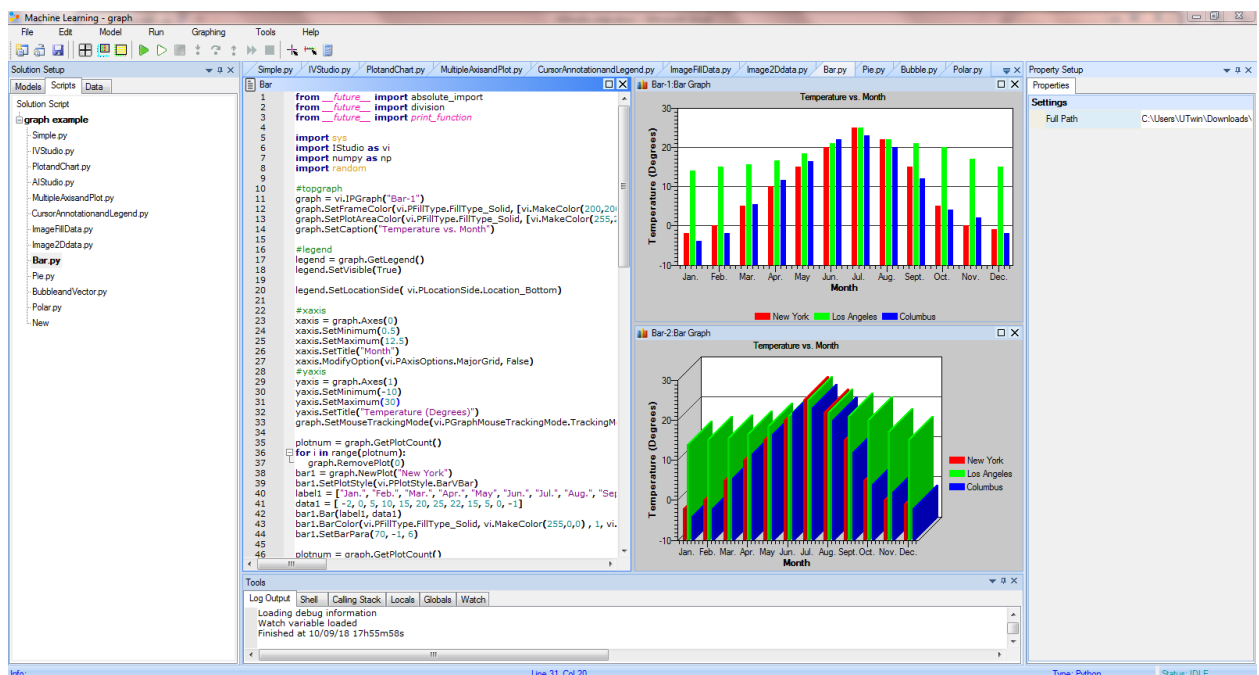


Figure 3.2: Python script with multiple bar graphs

4. Hardware (Video and Voice) Configuration

An AI system usually includes video and voice data. AIStudio supports video and voice input/output. All devices setup are configured through “Device Configuration” dialog. To add support for a new type of device, device manufacturers should provide an AIStudio-compatible library for their devices. Users can add a new type of devices by registering their device library in the “Device Configuration” dialog.

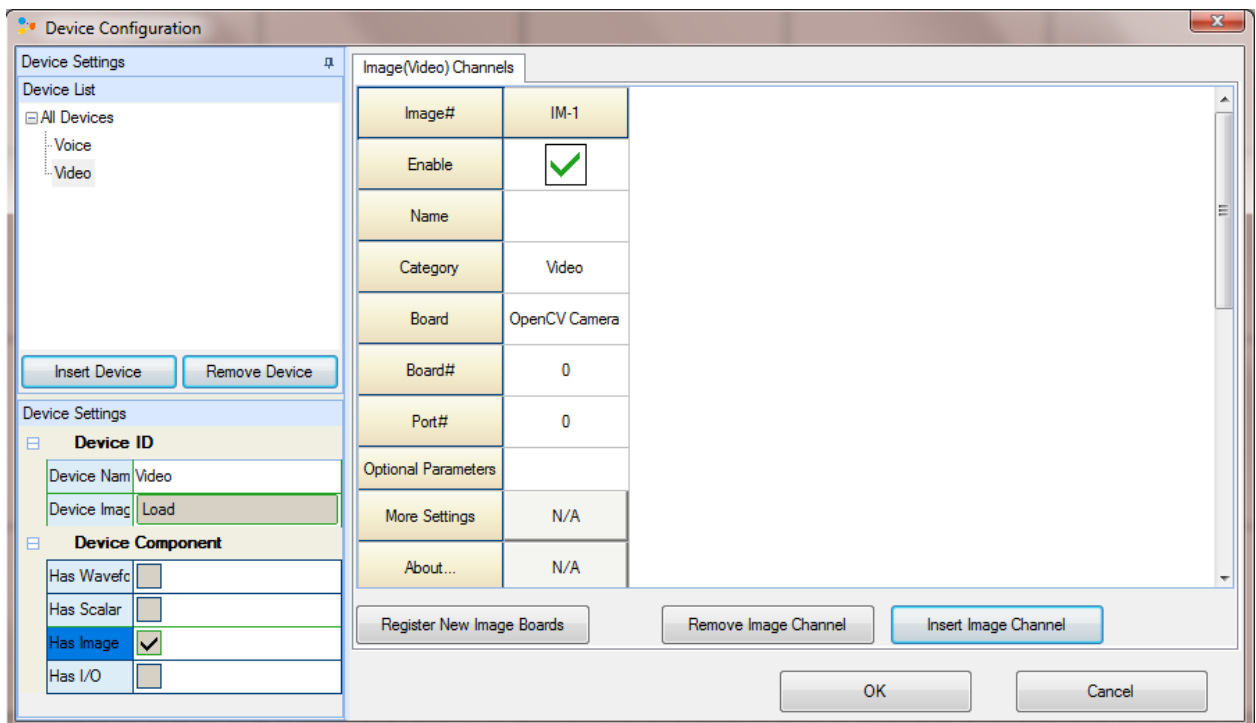


Figure 4.1: Imaging and waveform devices management

Video and Audio Devices

Each video and voice devices are specified by its name, board and port number. A camera graph (by specify image channel using device name) can be used to display alive video or replay video files (Figure 4.2). An audio waveform graph can be used to display waveform or replay wave data(Figure 4.3). Python script can acquire and play video or audio data by specifying the device name.

AIStudio 1.0

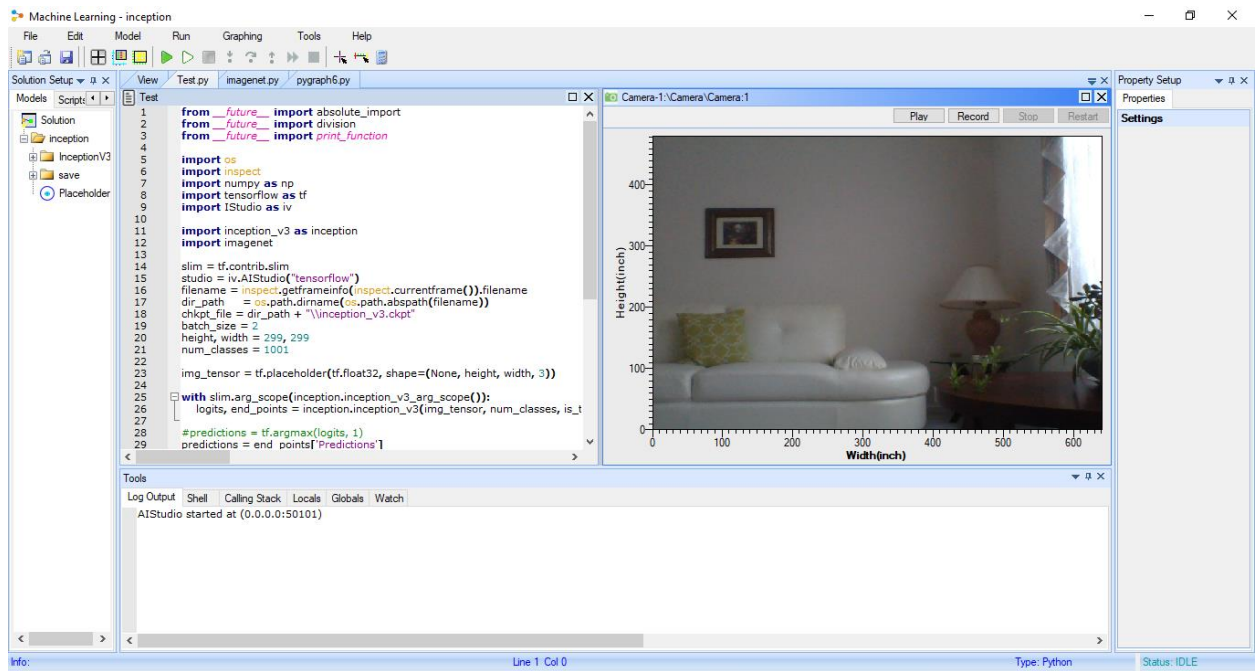


Figure 4.2: Camera graph for video devices.

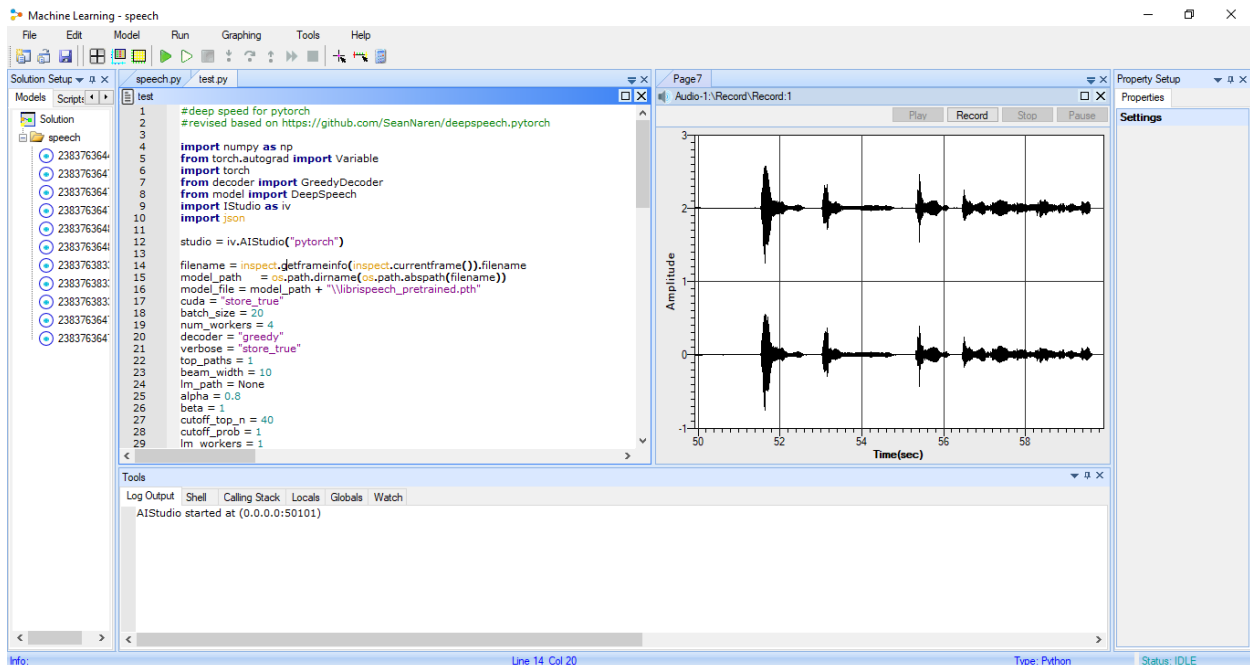


Figure 4.3: Audio graph for audio devices.

I/O Control Board

Digital I/O boards are used for toggling control line or inputting control status. You can add/remove I/O line. Each I/O is specified by its board type, board and port number boards and state.

Scalar Board

Scalar boards are used for acquiring non-waveform data such as temperature and pressure. For scalar channels, you can specify the data name, unit name, board type, board and port numbers.

5. Python IDE

AIStudio provides an IDE tools for run and debug Python script. The following images shows the AIStudio IDE with a list of script files, script editor and debug windows.

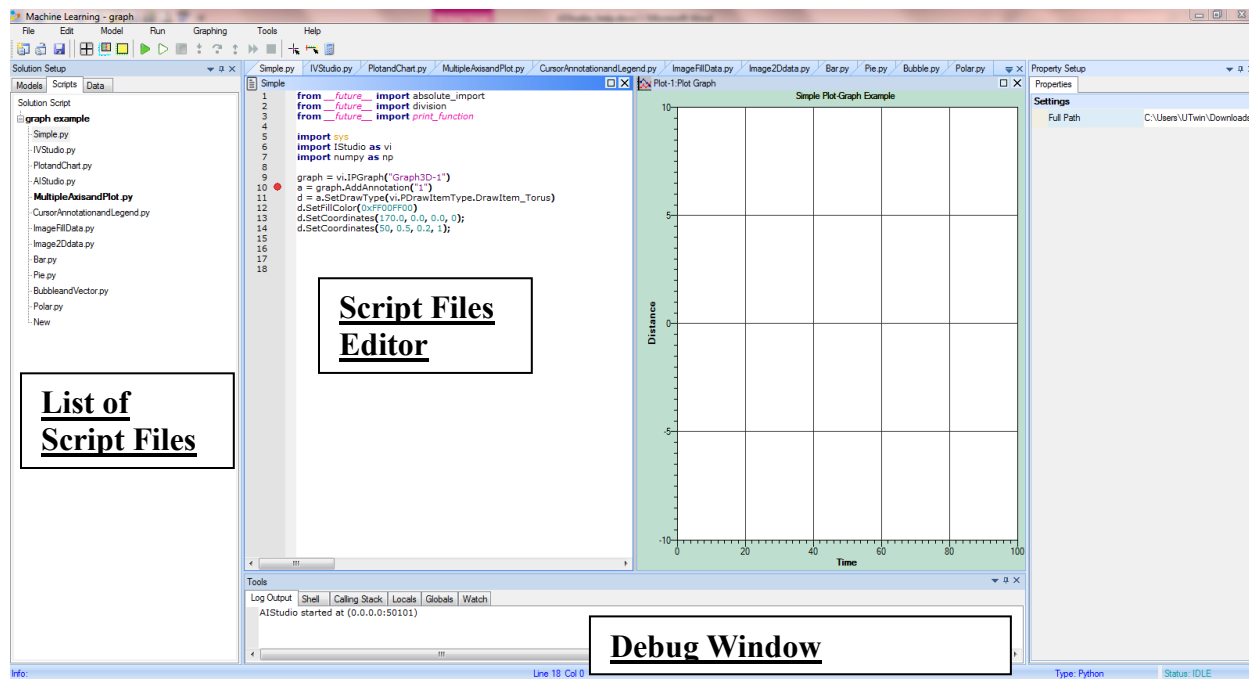


Figure 5.1: Python IDE

5.1 Creating Project

You can use AIStudio to view individual script files, more commonly you will be working in a project. An AIStudio project is a collection of script files, AI models, data and view graphs. Each AIStudio project can contain multiple AI models.

Each model contains a list of script. The scripts are organized as a tree and you add new folder, delete and add items into the script tree.

5.2 Writing Python Code

If you are a developer, the editor window is where you will probably spend most of your time. AIStudio includes editors for Python script.

You can edit individual files in the text editor by clicking **File | Open File**. To edit files in an open project, click on the file name in script list. The code is colorized. You can have lots of text editor tabbed windows open at once. You can split each window independently. AIStudio provides many productivity features such as syntax highlighting, code indentation and find/replace.

5.3 Compiling and Building Code

The Run menu contains all commands to run and debug your scripts. You can select **Start Debugging (F5)** to run your code in debug mode. Using **Start Without Debugging (Ctrl + F5)** to run script without debugging. Using **Syntax Check (Ctrl + F7)** to check syntax.

The default start script (main script) for a model is highlighted in the script list. If you select command from the Run menu, then the default start script will be run. If you want to run a specific script file, then right click the file and in the popup menu, select **Run** or **Syntax Check**.

5.4 Debugging Code

AIStudio's debugger enables you to debug code. You can step through code one statement at a time and inspect variables as you go, and you can set breakpoints.

Debugger Windows

To see a list of debugger windows, select **Run** menu and **Debug Windows** submenu.

The output from the build, including an error or success messages, appear in the **Output** window. You can use a **Watch** window to specify a variable (or an expression) that you want to keep an eye on. Right click the **Watch** Window to bring up a popup menu, where you can select to insert, remove and view variables. The **Locals** window shows you the variables that are currently in scope. The **Globals** window shows you the global variables. The **Call Stack** window shows the order in

which methods and functions are getting called. You can double-click a line to go look at that source code and that also changes the current scope being inspected by the debugger.

Using Breakpoints

You can set breakpoints when you want to stop debugger execution, perhaps to see the state of code variables or to look at the call stack. A breakpoint is indicated by a red circle in the front of code line. Using the Run menu, script editor context menu or shortcut, you can toggle breakpoint for the current line, delete/disable /enable all breakpoints.

5.4 Shell Windows

The shell window provides a Python console. The console window (opened with the **Run > Debug Windows > Shell Console** menu commands) lets you enter arbitrary Python code and see immediate results. Once the console window is open, you can start entering code line-by-line at the `>>>` prompt. The console window executes each line as you enter it, which includes importing modules, defining variables, and so on

6. Interaction between Python and Data Views

AIStudio provides a rich set of widgets (2-D and 3-D graphs, data grid, camera view, and voice waveform view) to view data. Each widget has its unique ID. Python script creates a python widget based on the widget's ID. All available python widgets are defined in script\IVStudio.py file. Python widget classes include:

- **IPGraph:** 2-D and 3-D graphs. A graph includes IPPlot, IPAxis, IPCursor, IPDrawItem, IPDrawGroup, IPAnnotation, IPLight3D, IPPalette, and IPLegend python classes.
- **IPDraw:** 2-D and 3-D drawing.
- **IPDataGrid:** Excel style data grid.
- **IPCamera:** a widget to show video.
- **IPAudio:** a widget to display waveform.

Using python widget class, you can change widget appearance, plot data and get data.

7. Create and Run AI Model

AIStudio provides many tools to make it easier for creating, debugging and deploying AI models. An AI model includes model definition (graphdef), related python script and data files(checkpoints, summary data). These three AI model properties are presented in the solution setup bar in Model, Script and Data tabs.

The model tab includes a tree list of operators in model's graphdef. You can import a model by loading its graphdef file. The operator list is organized based on the operator's name scope. You can view the model graphically using the Model Diagram view. Click any operator in the list will highlight its diagram node in the model diagram and its properties will be displayed in the property bar.

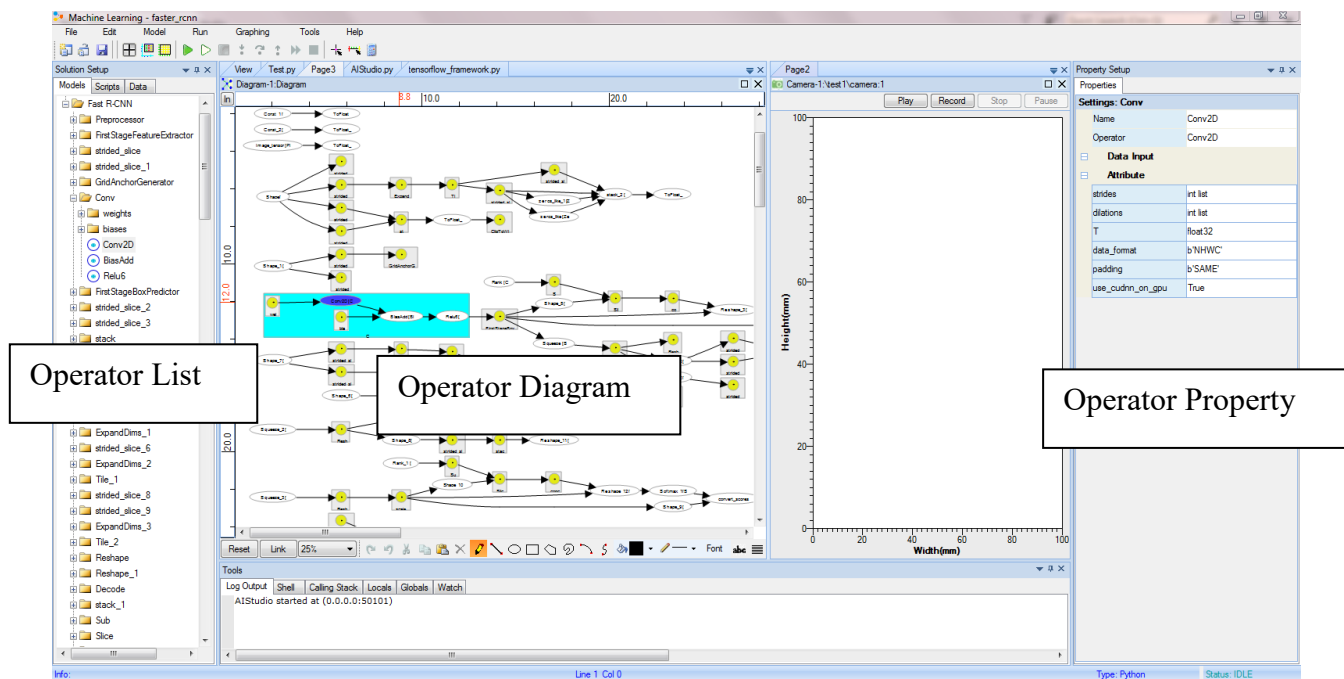


Figure 7.1: AI development IDE

For each model, there is a corresponding folder in the script list. You can organize the model-related python script inside this folder.

When running an AI model, you can create checkpoint and save summary data. All these data are organized in the Data tab.

7.1 Run AI Model

AIStudio provides an abstract interface to integrate AI framework (tensorflow, Pytorch, ...). This abstract interface provides framework-independent methods to save graphdef, save and restore checkpoint. AIStudio also provides methods to save summary data (tensor, image, waveform, scalar value) during running. These saved summary data can be viewed during running or replay. Thus you can observe “real-time” loss value, weight distribution and any model data while you are training your model. You can use the live camera/audio graph to feed real-time data into your model. You can also use the drawing widget to feed handwriting image into your model.

7.2 Display AI Model Data

AIStudio has powerful data view capability. You can use these graph widgets to view data during training or replay. To view replay data, in the data list, select these files you want to view and add them into the display file list. Then when you open graph’s format dialog (right click the graph and select Format Graph), you can select the data saved in these files to view. If the data is a tensor, you can slice the data.

8. Examples

Below is a list of example projects inside the example folder

\example\graph\graph.aip: this project shows Python scripts with data visualization.

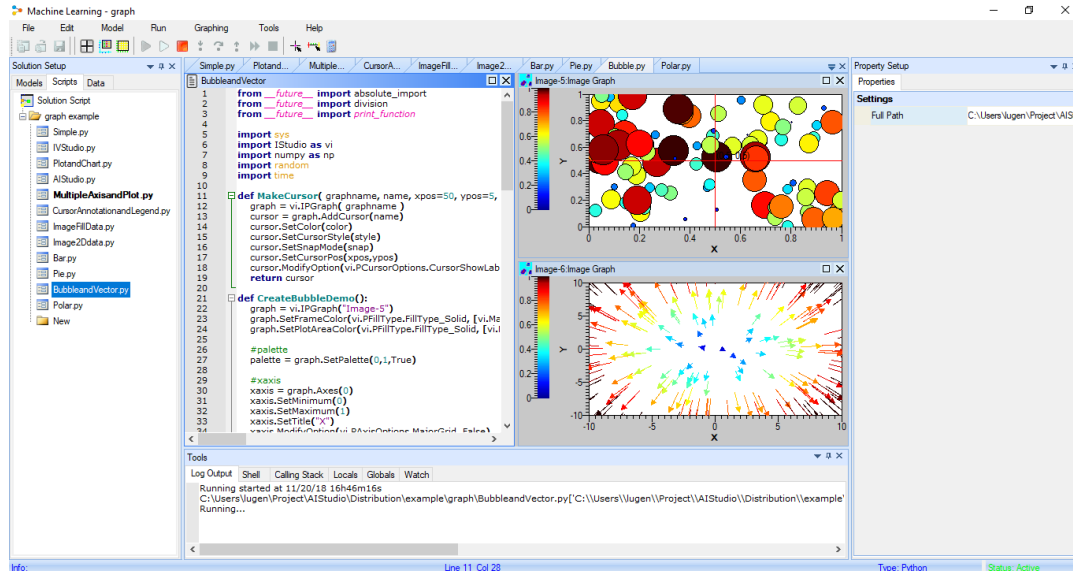


Figure 8.1: data visualization

\example\mnist\mnist.aip: this project shows the tensorflow MNIST example with data preview, displaying neural network weight, loss, accuracy, and handwriting input.

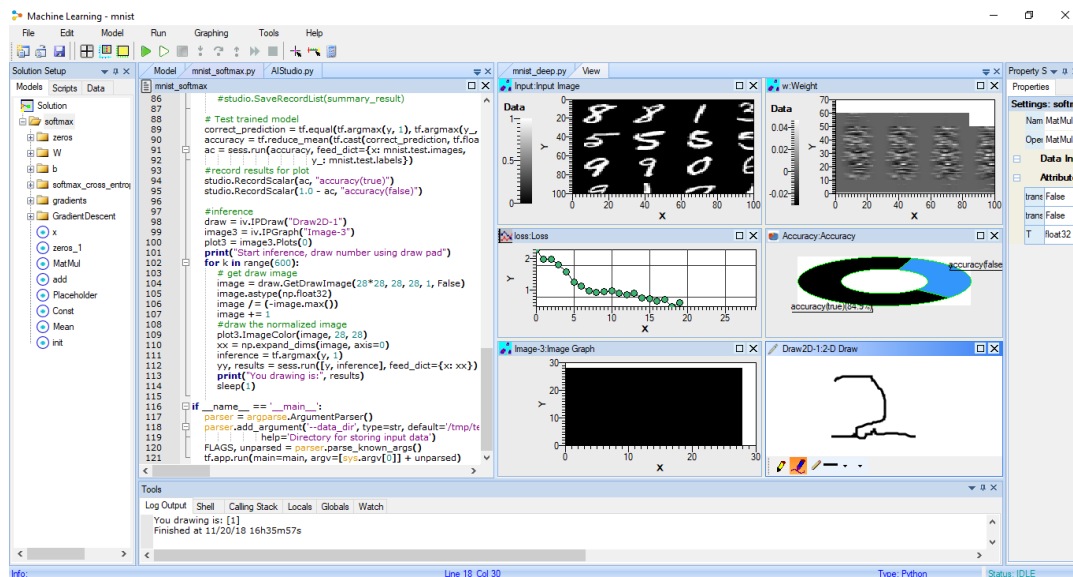


Figure 8.2: MNIST example

AIStudio 1.0

\example\inception\inception.aip: this project shows Google's inception 3 model. It displays model graph diagram, real-time image recognition using camera graph.

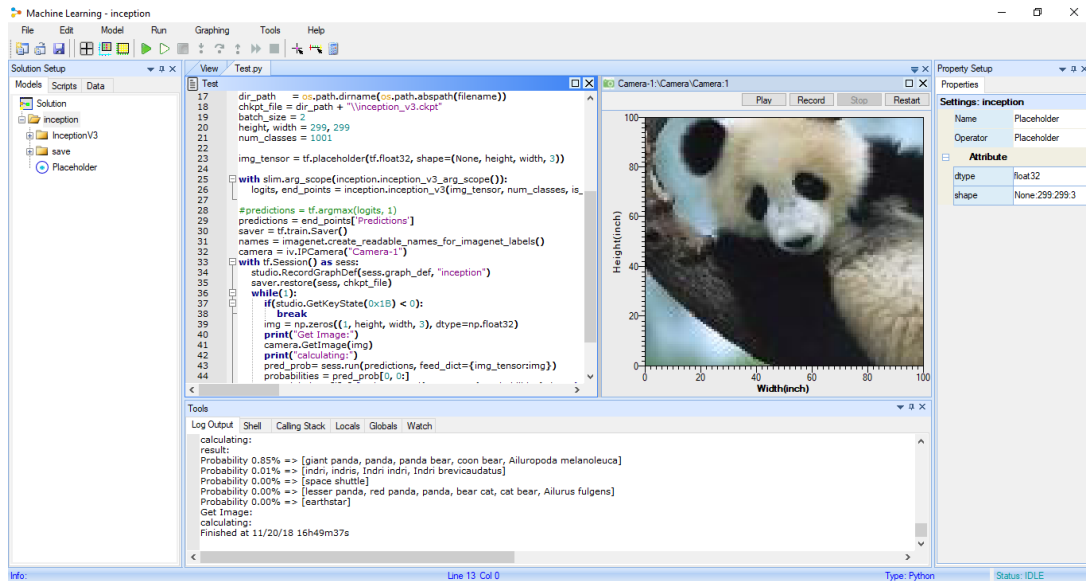


Figure 8.3: Google's inception 3 model

\example\r-cnn\faster-rcnn.aip: this project shows the fast R-CNN model. It displays model graph diagram, real-time image recognition using camera graph.

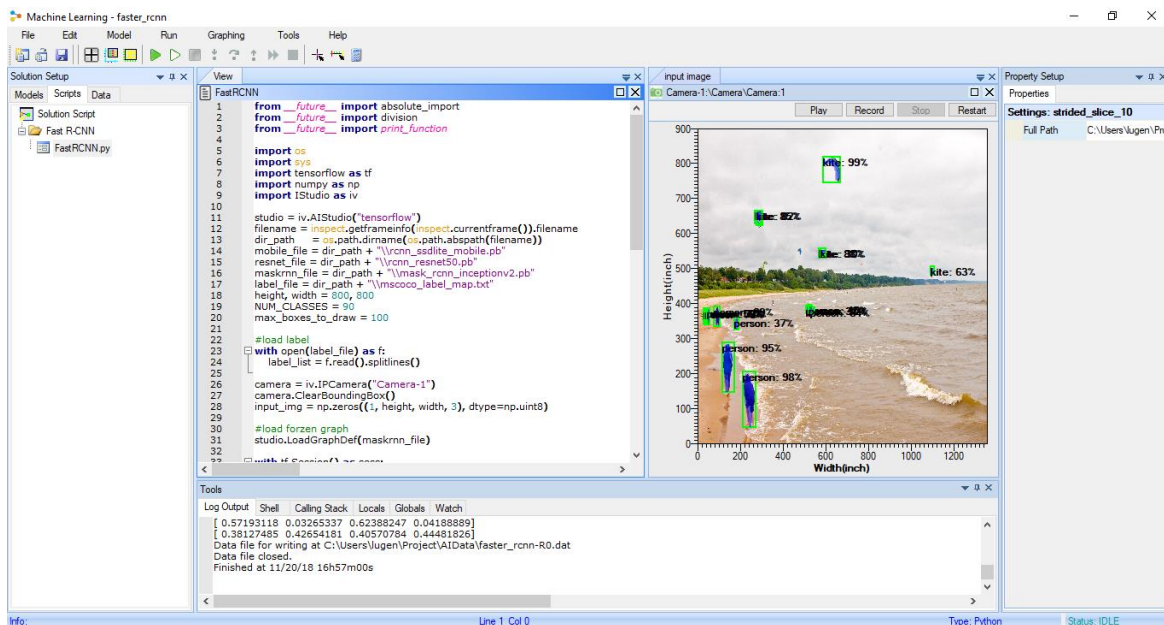


Figure 8.4: Faster R-CNN model

AIStudio 1.0

\example\rnn\rnn.aip: this project shows tensorflow RNN model. It displays a popup dialog to let you input sampling prefix, then it will train a language mode and generate sampling text.

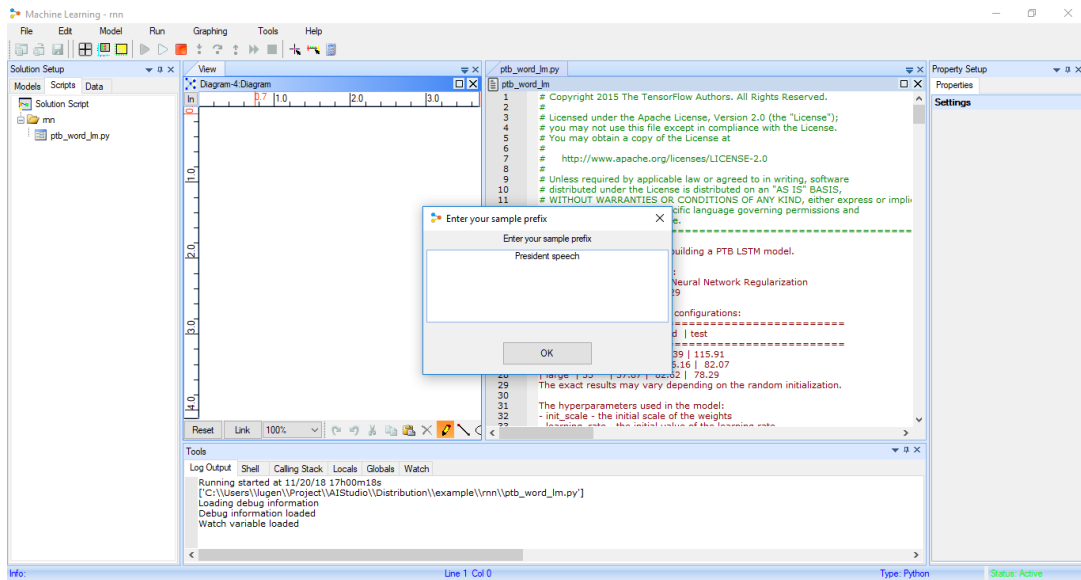


Figure 8.5: RNN model

\example\speech\speech.aip: this project shows Baidu deep speech model based on PyTorch. It acquires audio data from audio devices and converts it to text.

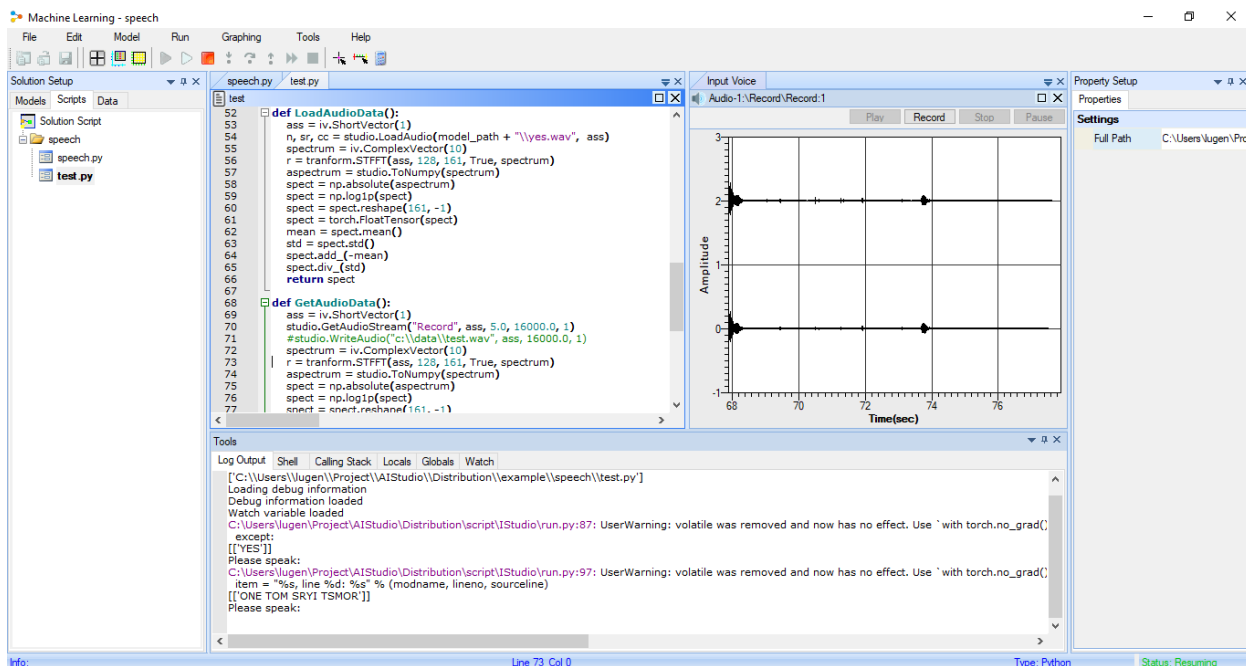


Figure 8.6: PyTorch deep speech model