Cheyenne Hathaway

ITAI 3377

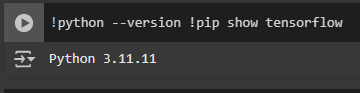
Lab 02

**Lab 02: Deploying an AI Model on a Simulated Edge Device**

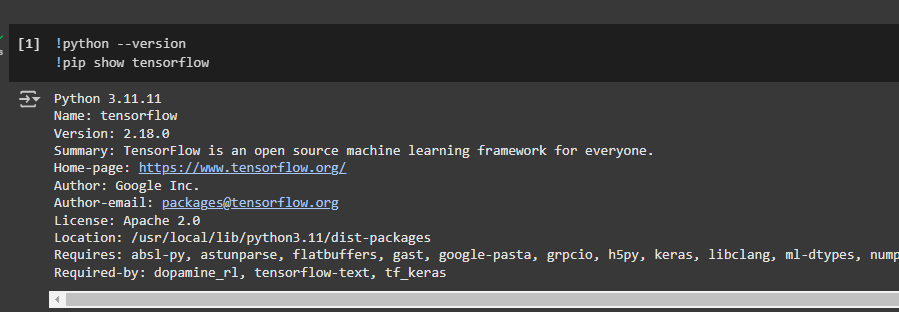
**Part 1: Setting Up the Development Environment**

**Step 1: Verify Python and TensorFlow Installation**

Step one was not exactly necessary for me as I regularly use Google Colab, which has Python readily available. I failed to move !pip show tensorflow to a new line, so it did not show me any version of TensorFlow. I took that to mean I did not have it, so I went ahead with the installation. However, I have in fact installed it before so it naturally showed the expected “Requirement already satisfied” message. When I reviewed it, I realized my mistake and fixed it, so it showed me the version I have.



Screenshot of what I initially got when I forgot to move the line for TensorFlow to a new line.



Screenshot after fixing the issue.

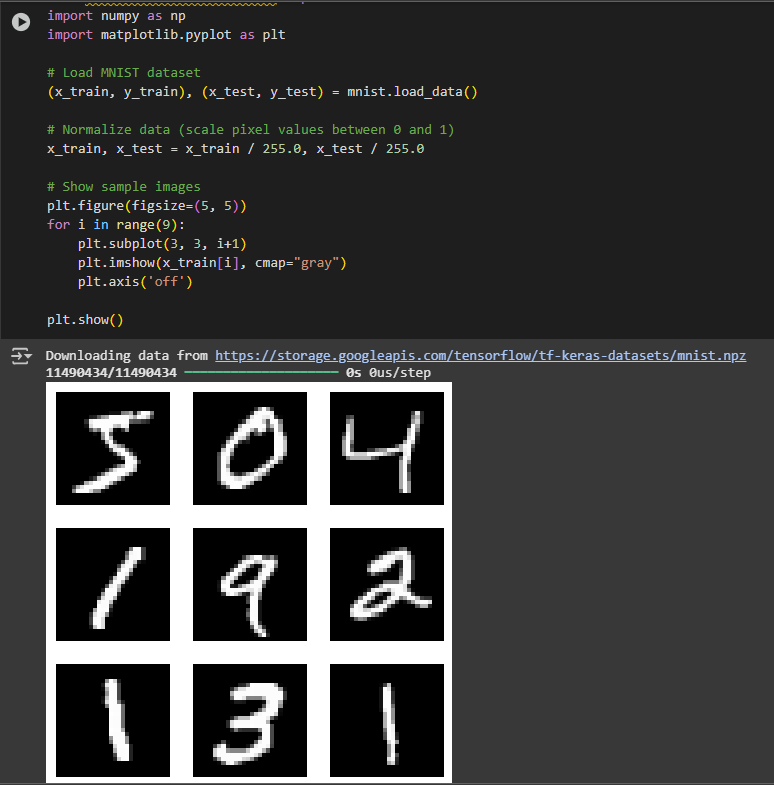
**Step 2: Install Jupyter Notebook (if working locally)**

I did not do this step as I prefer Google Colab over Jupyter Notebook.

**Part 2: Creating and Training an AI Model**

**Step 3: Load the MNIST Dataset**

After redoing the code with what I thought was proper structuring, I was getting a syntax error. Colab did not have a suggestion for the fix and I could not find where the problem was so I used ChatGPT. It turns out I had improperly separated lines cutting off part of the code, causing the syntax error. After applying the suggestion from ChatGPT, I ran this portion of the code again without an issue.

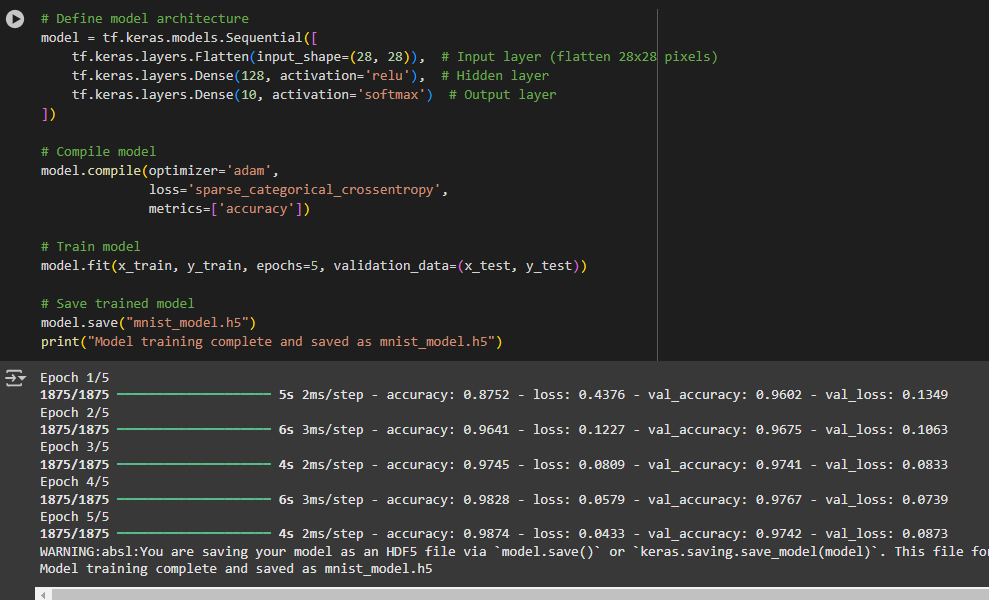


Screenshot of the output.

**Step 4: Define and Train a Neural Network**

For this step, again I just restructured the given code. I did not have as hard a time with this one, though it did suggest saving in a different way. I ignored that for the sake of this project, but I also used ChatGPT for a quick search on other save options. The suggestion was simply to put model.save(“mnist\_model.keras”)instead of model.save(“mnist\_model.h5”. I did run this just to double check before changing it back, and there was at the very least not an issue with saving the file this way versus using h5. However, I did not run the rest of the code using the update, so I am unsure if this would have made a difference to the overall project or not.

For further explanation of what this step means, training the network essentially consists of giving it labeled data so it can learn and minimize discrepancies between the predicted answer and the actual answer.

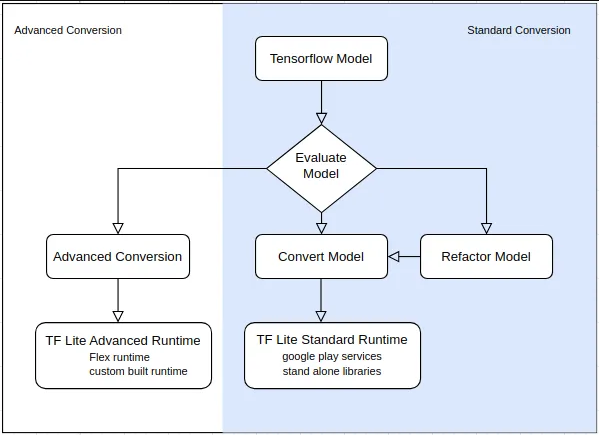
Screenshot showing the code and output of the training logs and accuracy values. It also shows the mentioned warning, though I had to cut it off for readability. 

**Part 3: Converting and Saving the Model**

**Step 5: Convert the Model to TensorFlow Lite Format**

Here, I used the given code (again, after editing for structure so it could run), to convert the model to TFLite. This will allow it to run on smaller devices as it will now require less resources. I was surprised at how simple this was and how short the code for this portion of the project is. It is just a matter of loading the model, using two lines of code to convert it, and then saving the newly converted model.

In this step, the TF model is converted to a format compatible with TFLite. This can involve converting the dtypes or removing operations that will not be supported on the smaller device. It also may include removing or combining some operations. It is important to note that while the model will take up less memory, sometimes the precision of the model can be impacted. See the below image for a flowchart of how this process works:



Source: https://miro.medium.com/v2/resize:fit:640/format:webp/0\*dbx29n9t064ayexz

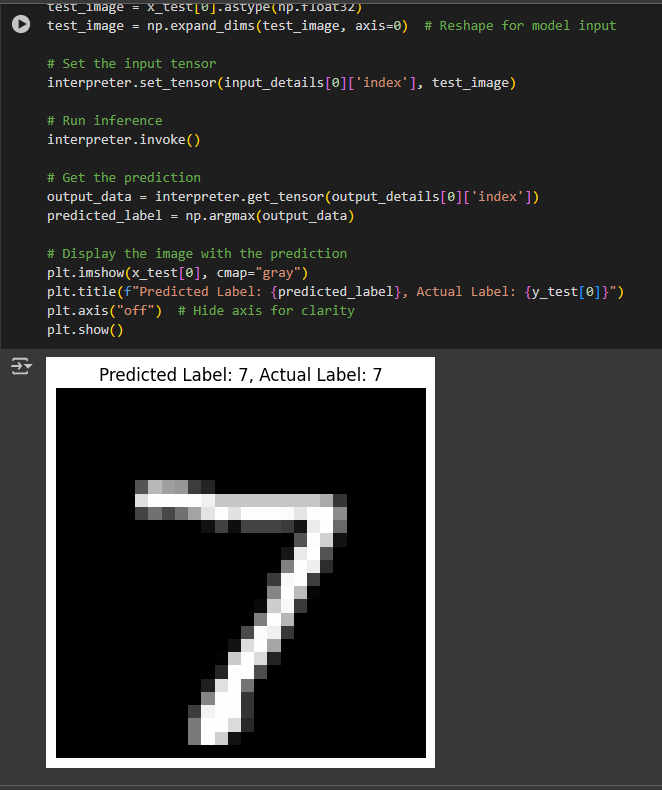
**Part 4: Loading and Running Inference with TensorFlow Lite**

**Step 6: Load the Converted Model Using TensorFlow Lite Interpreter**

This part of the assignment was again very simple and straightforward. The code required little editing as it was just loading the model and confirming that it was in fact loaded.

**Step 7: Perform Inference with TensorFlow Lite**

Here I was able to actually run the new TFLite model. This ensures that it was converted properly, loaded properly, and gave the expected predictions. While I did use Colab for this, which allows use of GPUs and TPUs, I still noticed this code ran faster than the previous code. While I cannot be sure of the accuracy of this as I tested it on a single image, the predicted label and actual label matched, which always allows for a sense of satisfaction in these projects.



Screenshot of the output of the TFLite model.

Overall, I have learned that I need to study the rules to formatting/structuring code, as I had to research far more than I expected to get the correct structure. I am used to platforms and projects that require less effort around structure, so it is something that I have yet to master. I believe having a better understanding of it will be what gives me both the knowledge base and confidence to stop classifying myself as a beginner when it comes to code.

As for real world application of TFLite, it allows models to be used on smaller devices, or devices that do not have the power of GPUs and TPUs behind it. This means these models can be used by a wider array of consumers, making this technology more accessible and more affordable. This could apply to countless situations, though the one that comes to mind first is for individuals in developing countries. It could aid in anything from farming to medicine, but it allows people from all over to use AI to better their lives and gives them access to technology previously reserved for people or companies with more resources, be it computational or financial.

**Resource used:**

*.*

* OpenAI, "ChatGPT (Version 4)," OpenAI, 2025. [Online]. Available: <https://openai.com>.
* [1] IBM, "Neural networks," *IBM Think*, [Online]. Available:<https://www.ibm.com/think/topics/neural-networks>. [Accessed: Feb. 4, 2025].
* Ishara Neranjana, "From TensorFlow to TFLite: How model conversion is done and how it affects neural network structure," *Medium*, March 21, 2023. [Online]. Available:<https://medium.com/@zone24x7_inc/from-tensorflow-to-tflite-how-model-conversion-is-done-and-how-it-affects-neural-network-structure-1d01086083e0>. [Accessed: Feb. 4, 2025].