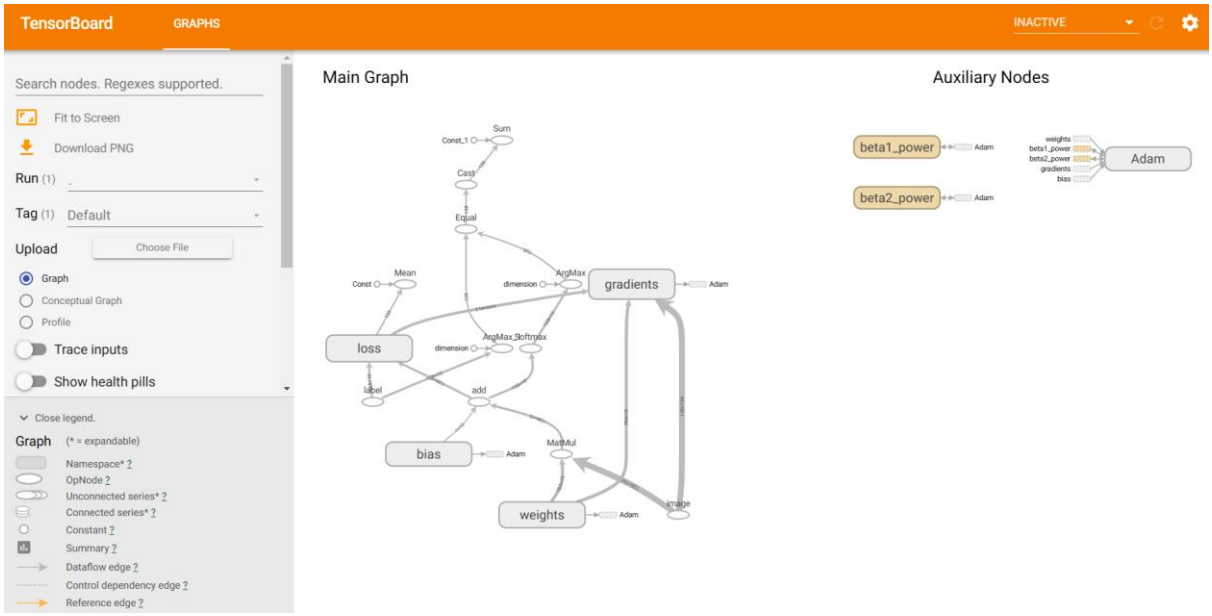
ASSIGNMENT 1 – REPORT

For a1a.py (TASK 1)

Description –

In this task, I used the Logistic Regression model. It is one of the simplest and most common techniques for supervised learning. It is a statistical method used to predict a binary outcome given a set of independent variables. It uses a logistic function to model a probability of a certain class or event existing such as the class of the given handwritten digit in our case.

TensorBoard graphs –



Accuracy –

```
Epoch 23, Train Loss: 0.2336287945508957, Train Accuracy: 93.6690902709961, Test Accuracy: 92.77999877929688
Epoch 24, Train Loss: 0.2315855771303177, Train Accuracy: 93.70545959472656, Test Accuracy: 92.72999572753906
Epoch 25, Train Loss: 0.22962485253810883, Train Accuracy: 93.68727111816406, Test Accuracy: 92.68000030517578
Epoch 26, Train Loss: 0.22973507642745972, Train Accuracy: 93.74545288085938, Test Accuracy: 92.63999938964844
Epoch 27, Train Loss: 0.23076367378234863, Train Accuracy: 93.81272888183594, Test Accuracy: 92.75
Epoch 28, Train Loss: 0.22718219459056854, Train Accuracy: 93.75091552734375, Test Accuracy: 92.65999603271484
Epoch 29, Train Loss: 0.22834576666355133, Train Accuracy: 93.77818298339844, Test Accuracy: 92.76000213623047
Epoch 30, Train Loss: 0.22685138881206512, Train Accuracy: 93.82181549072266, Test Accuracy: 92.68000030517578
```

Time spent –

I spent about 20 minutes on this task. I went through the template code, read a bit about logistic regression, and understood the utils methods in detail as well. It was a relatively straightforward task.

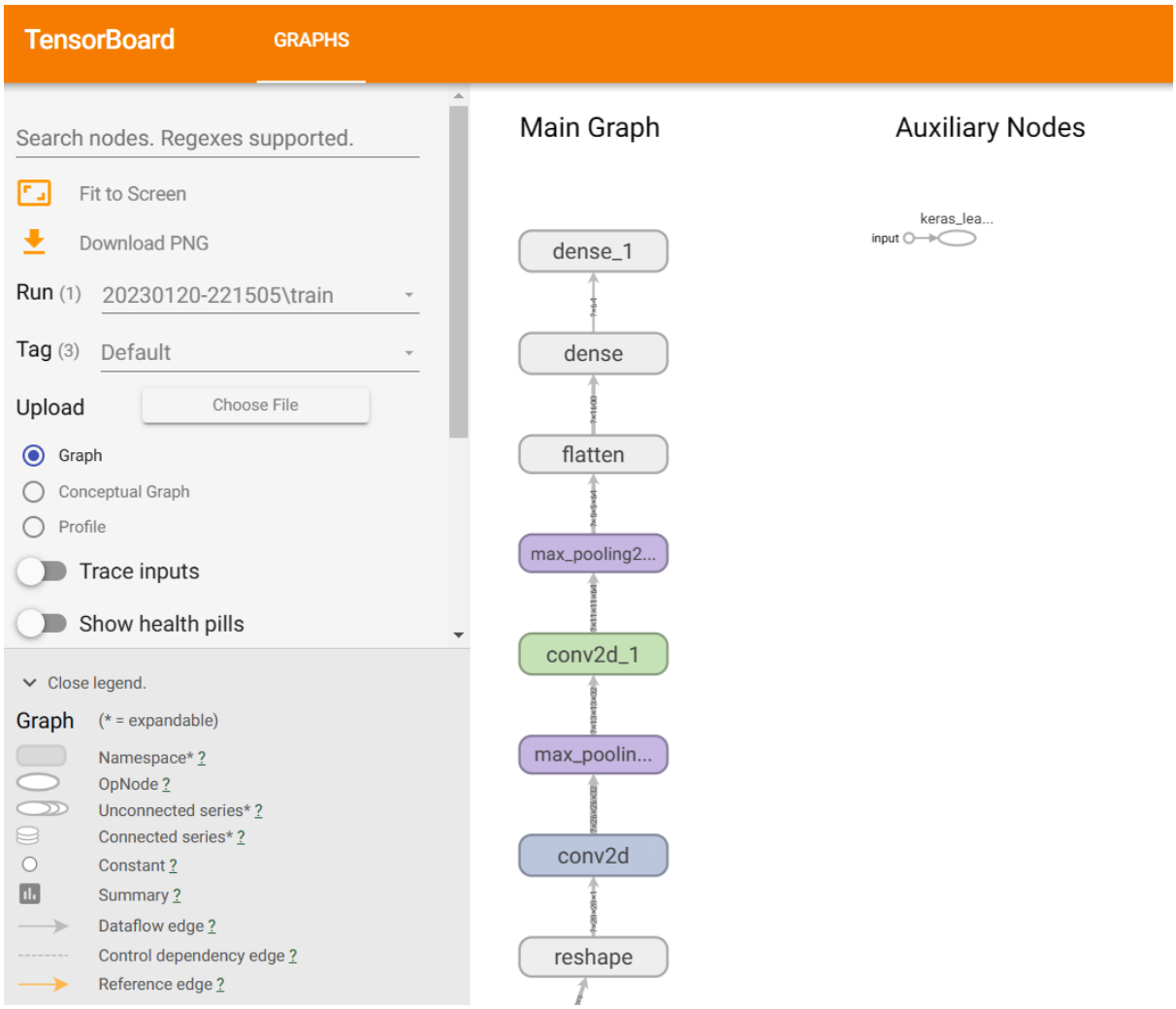
For a1b.py (TASK 2)

Description –

In this task, I have used a Convolutional Neural Network (CNN) model. CNN works well to learn spatial hierarchies of features, which means it is good to classify images-based data. The model is composed of several layers –

- The first layer is a reshape layer, which reshapes the input data to a specific shape (28, 28, 1) that is suitable for the convolutional layers.
- The next two layers are convolutional layers with 32 and 64 filters respectively. These layers extract features such as edges, corners, etc. from the input images.
- After each convolutional layer, there is a max pooling layer which reduces the spatial dimensions and the number of parameters in the model and thus helps reduce overfitting.
- Then there is a flatten layer which flattens the output to a 1D vector.
- The next layer is a dense layer with 64 units and relu activation function, which is a fully connected layer that helps to learn more complex features.
- The final layer is an output layer with 10 units and a softmax activation function. The softmax function is used to output the probability of each of the 10 classes and the class with the highest probability is selected as the output class.

TensorBoard graphs –



Accuracy –

```
Epoch 2/5
1875/1875 [==============================] - 25s 13ms/step - loss: 0.0463 - accuracy: 0.9857
Epoch 3/5
1875/1875 [==============================] - 24s 13ms/step - loss: 0.0317 - accuracy: 0.9906
Epoch 4/5
1875/1875 [==============================] - 23s 12ms/step - loss: 0.0237 - accuracy: 0.9923
Epoch 5/5
1875/1875 [==============================] - 23s 12ms/step - loss: 0.0169 - accuracy: 0.9945
313/313 [==============================] - 1s 3ms/step - loss: 0.0269 - accuracy: 0.9920
Test accuracy:  0.9919999837875366
```

Time spent –

I spent almost 60 minutes on this task. First, I had to research what kind of model I should use for the given dataset. Then I had to go through some TensorFlow and Keras documentation for CNN models. Then I had to play around with the model layers so that the training in relatively fast and gives good accuracy. Also, I had to understand callbacks for TensorBoard as well.

Interesting Problems

- Machine Learning models would not give such a high (>99%) accuracy, but Deep Learning models could easily reach higher accuracies.
- Had to learn Keras alongside TensorFlow to implement CNN in a better manner.
- The decision to use which model depends a lot on the kind of dataset.
- The graphs for both models were very different and the one of Logistic Regression was complex to understand.
- The TensorBoard had a lot of functionalities that I did not understand in the first glance.