19_to_22nd_April_Update

1. The inference runs for 5 images are completed for all the layers of Mobilenet-v2 encoder part and the results have been noted down in the shared Google sheet.

   Excel sheet:
   https://docs.google.com/spreadsheets/d/1tmXCuR8P1yGrYK8_bC07wBkrz_x6DhPl-a2OtFICZnw/edit?usp=sharing

   The screenshots of the inference are also added in the github repo for your reference: https://github.com/bALAJi-aDItHYa/MBM_implementation.git

2. **Observation:**
   The expected result values are provided in the “results-Encoder” tab in the spreadsheet.

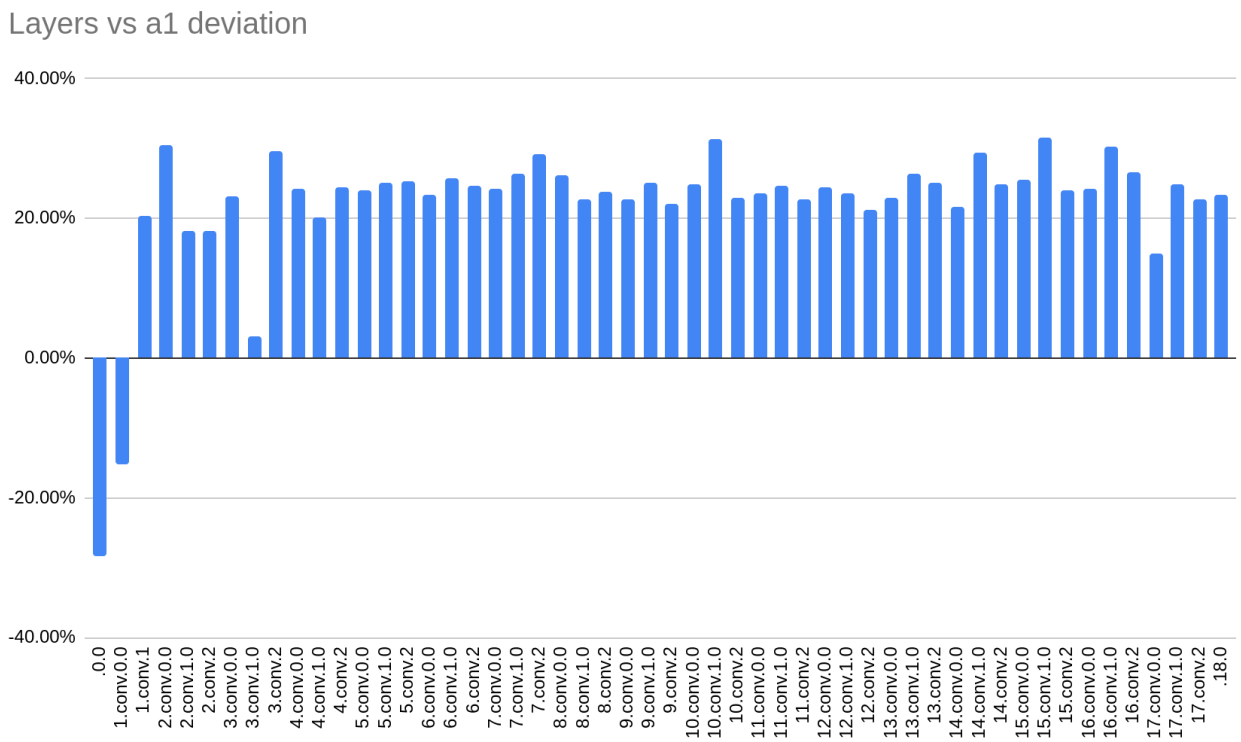
   Given below are the graphs representing the % deviation from the expected results for each layer after approximation is applied.

   * a1,a2,a3 - higher is better; rel, rms, log10 - lower is better

   A) For the error metrics a1, a2 and a3 -> the higher the value, the better it is. Hence if there is a higher positive deviation, it means the performance is better.

   B) For the error metrics rel, rms and log10 -> the lower the value, the better it is. Hence if there is lesser positive deviation, it means the impact on the performance is lesser due to approximation.

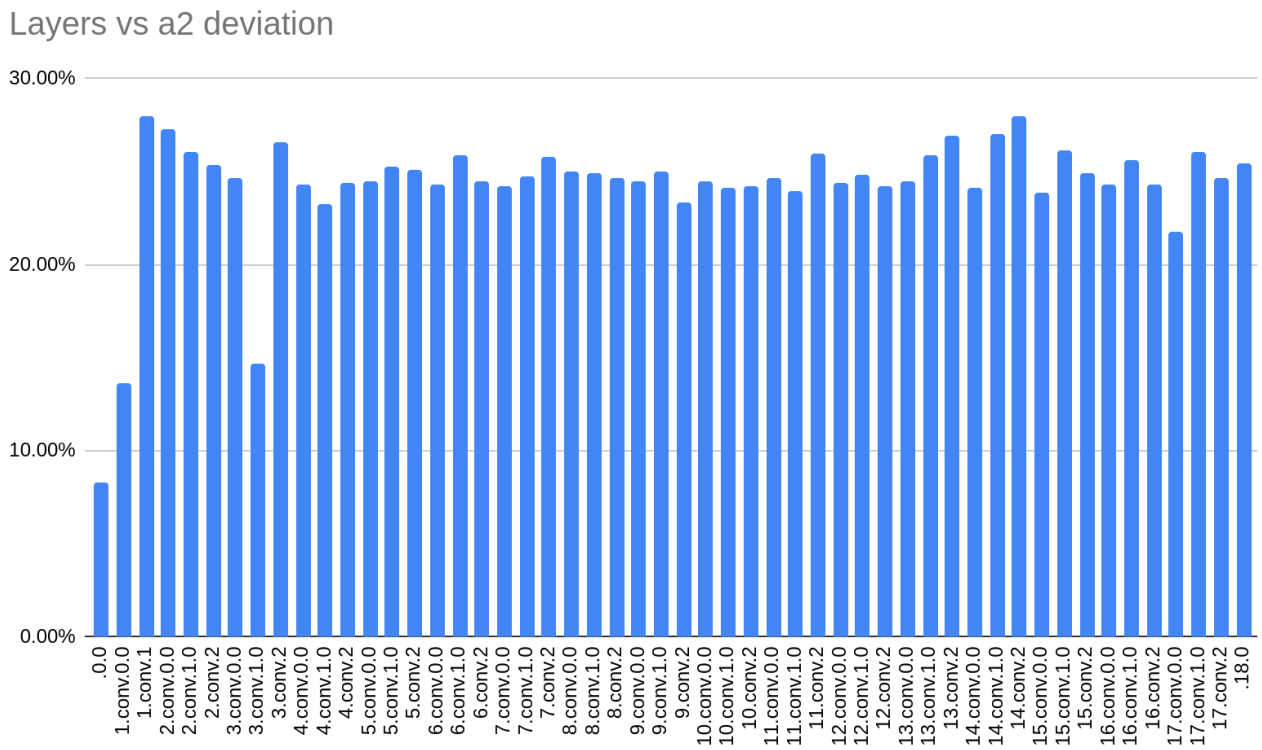
Layers vs a1 deviation



This graph represents the encoder layers vs a1 deviation from expected result. It is observed that when approximation is applied on the 0.0 and 1.conv.0.0 layers, the a1 result is lesser than the expected value by 28.37% and 15.16% respectively. In the rest of the layers, the a1 result is higher than expected value.

On an average there is an improvement of 22.22% in the a1 error metric when the approximation is applied.

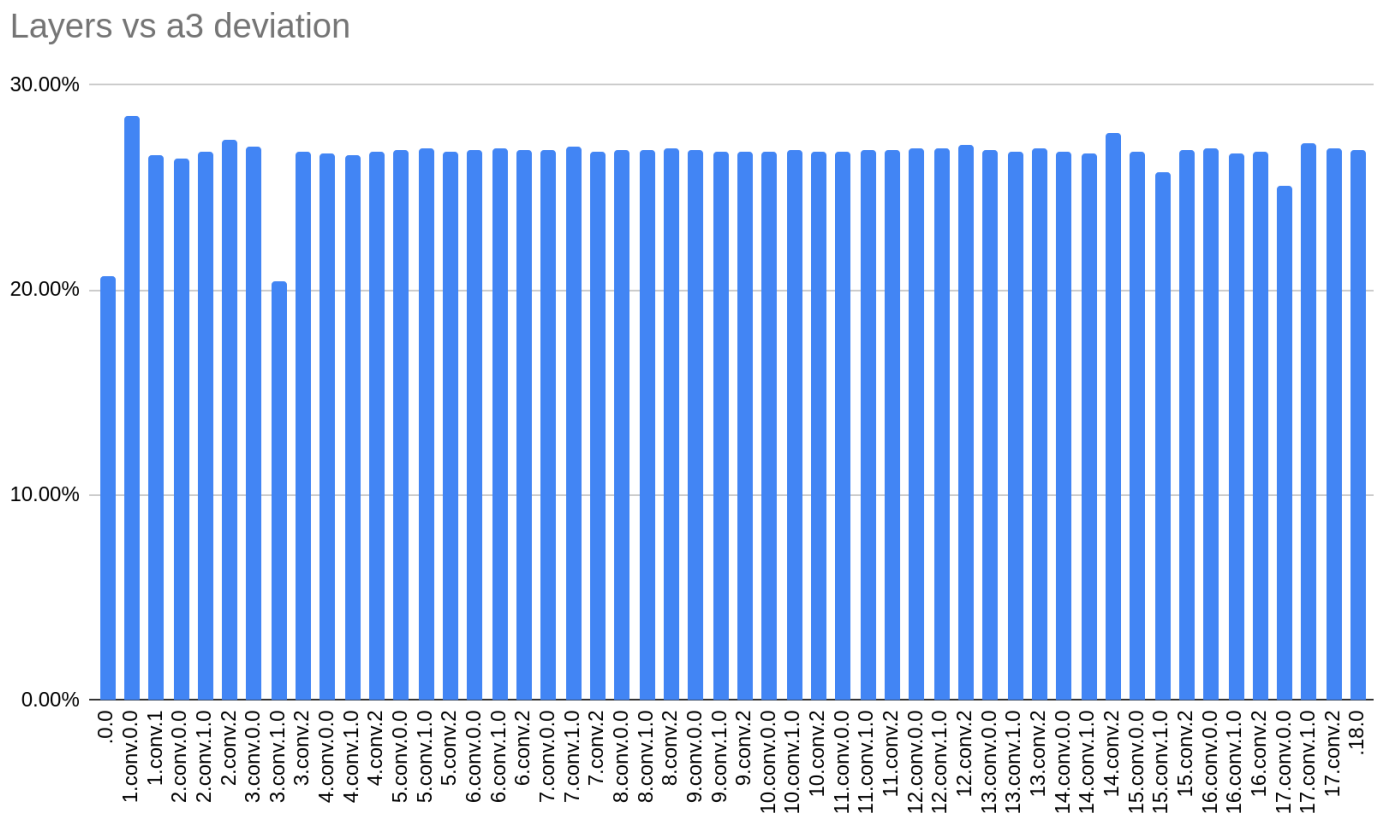
Layers vs a2 deviation



It is observed that, in all the layers when the approximation is applied there is improvement in the a2 error metric, as all of them have positive deviation.

On an average there is an improvement of 24.27% in the a2 error metric when the approximation is applied.

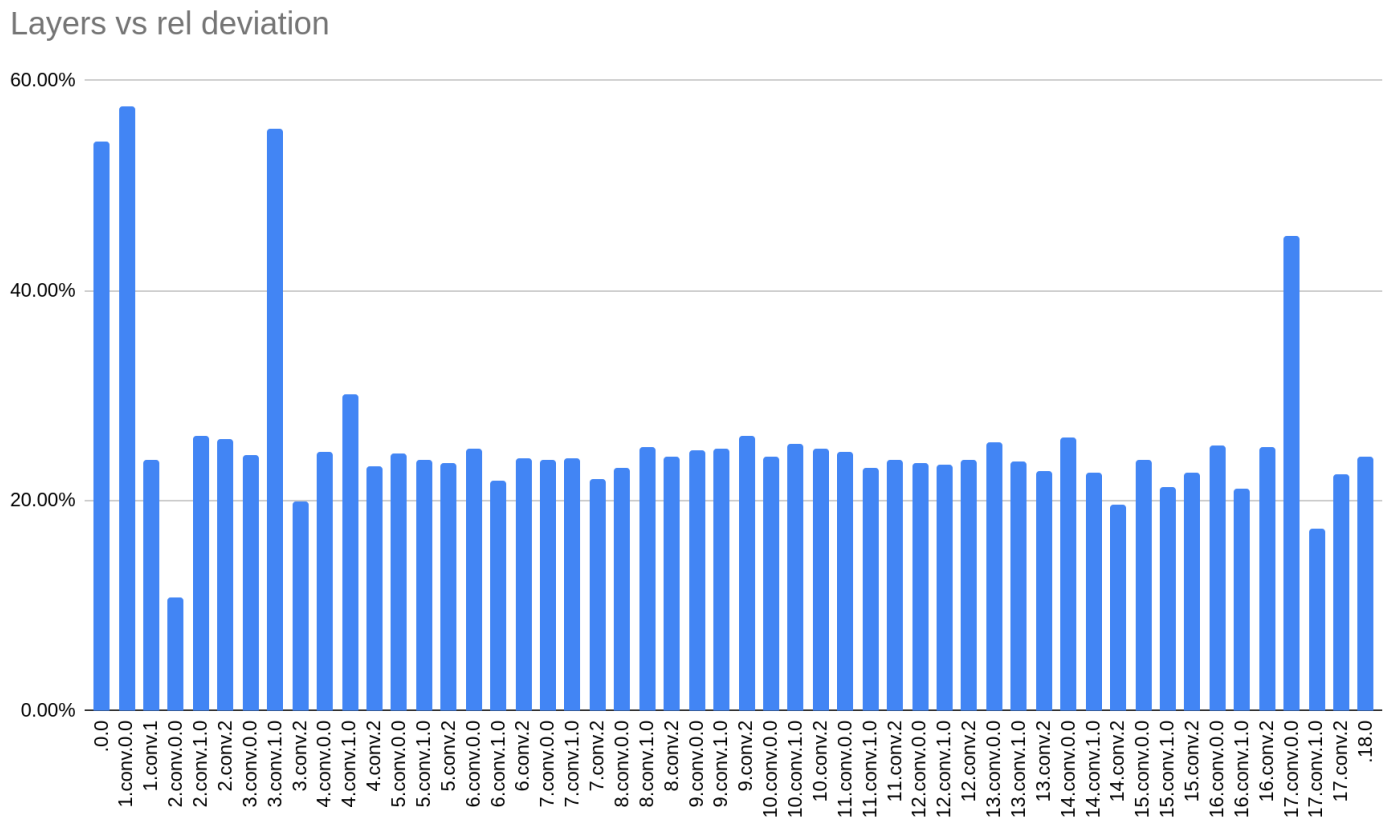
Layers vs a3 deviation



There is a positive deviation in the a3 error metric with respect to all the layers. Hence, there is an improvement in performance after approximation is applied in all the layers.

On an average there is an improvement of 26.55% in the a3 error metric when the approximation is applied.

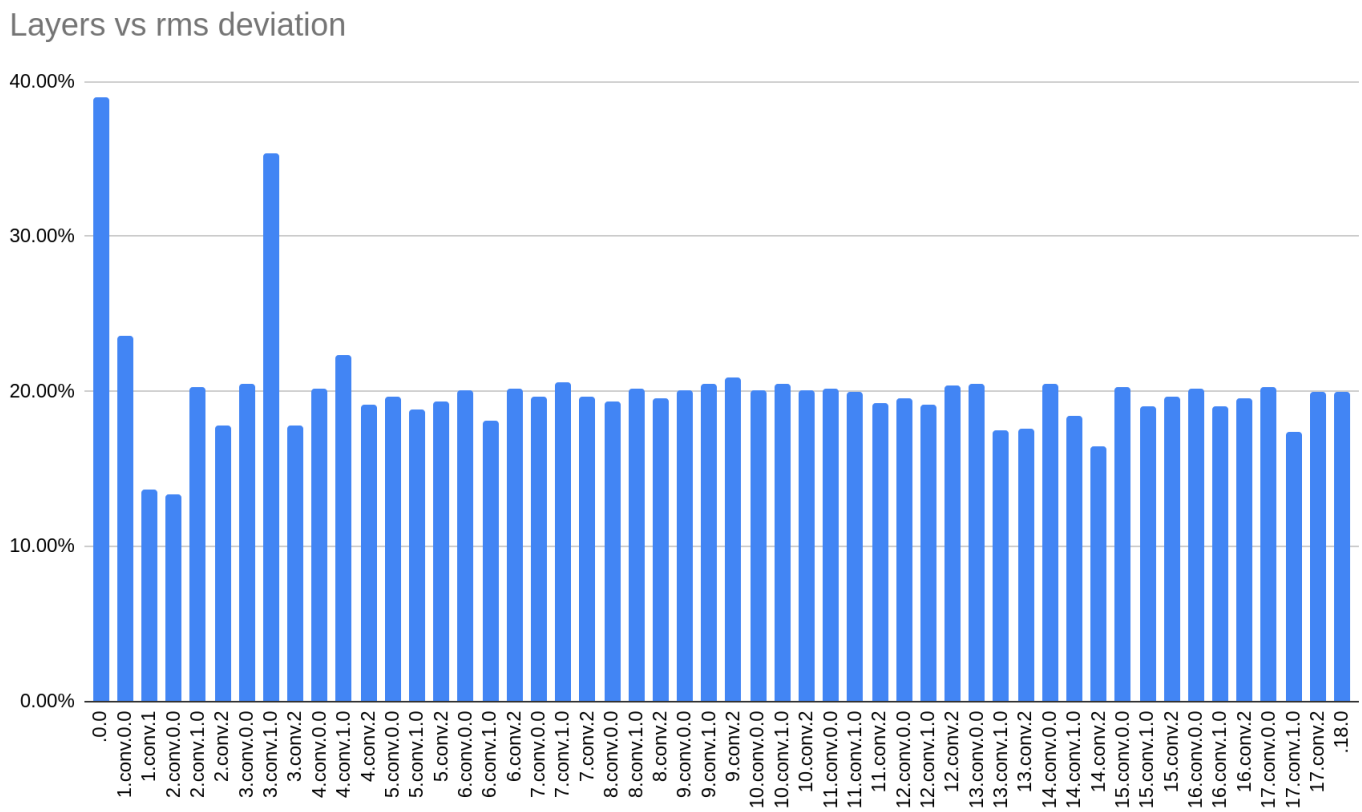
Layers vs rel deviation



On an average, there is a deviation of 25.83% from the expected value of the rel error metric.

The layers 0.0, 1.conv.0.0, 3.conv.1.0 and 17.conv.0.0 perform poorly with high deviations of 54.21%, 57.50%, 55.34% and 45.17% respectively.

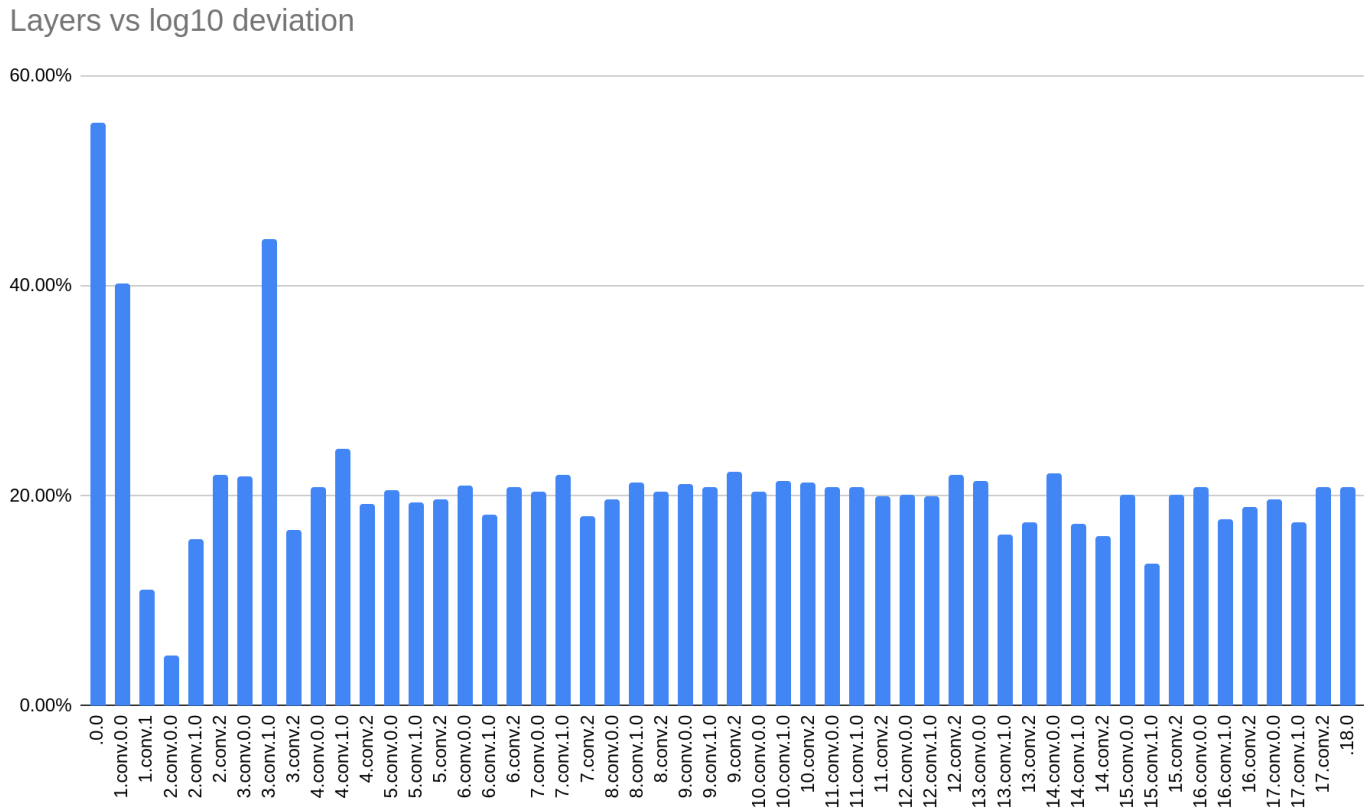
Layers vs rms deviation



On an average, there is a deviation of 20.09% from the expected rms error metric value.

The layers 0.0, 3.conv.2 perform poorly with higher deviations of 39.02% and 35.36% respectively.

Layers vs log10 deviation



On an average there is a deviation of 20.97% from the expected log10 error metric value.

The layers 0.0 and 3.conv.1.0 perform poorly with higher deviation of 55.51% and 44.49% respectively.

3. **Comparing Encoder and Decoder layers:**

   Average deviation values

	avg a1 deviation	**avg a2 deviation**	**avg a3 deviation**	**avg rel deviation**	**avg rms deviation**	**avg log10 deviation**
Encoder	22.22%	24.27%	26.55%	25.83%	20.09%	20.97%
Decoder	1.38%	13.38%	24.14%	63.78%	24.94%	31.89%

It is observed that, on an average when the approximation is applied on the encoder, the resultant metrics are better as compared to the decoder.

It can be noted that the average encoder **a1, a2, rel and log10** deviation values are much better when compared to the decoder deviation values of same error metrics.

The average deviation values of the error metrics **a3** and **rms** of encoder and decoder are comparable and are almost the same.