Task 6 - Model pruning

Problem Statement:
You may have trained different models ranging from MobileNets to GPT-3 (just kidding). One of the challenges we face when we deploy a model in outside world is that it's too bulky and slow and one of the ways to solve that is to use model pruning. Show us the way how it's done with explanation (impact on size, performance and accuracy)

Explanation of the code:
Though I am aware of the concept of Model Pruning, I hadn't employed it in any of my projects before. So, Thankyou team for the task.

In my search, I came across numerous articles on Model pruning. Tensorflow documentation proved the most useful. Most of the articles performed Model pruning on Sequential Models for the MNIST dataset.

For the Task, I have demonstrated Model pruning on a Face Mask Detector I recently implemented. The detector architecture is as folows:

- **Base Model:**
  - **MobileNet** pre-trained model , freezing its layers during training.
- **Fully connected layers:**
  - Dense Layers with 128 Neurons
  - Dropout Layer with 20% Dropout
  - Dense Layer with 2 output neurons

1. I have performed **Magnitude based weight pruning** using **tensorflow_model_optimization** package.
2. **tfmot.sparsity.keras.prune_low_magnitude( ) –** This function was used to wrap my model with the pruning function so that the weight matrix gets sparse during training. The idea is using this function with K% sparsity ensures K% of the layer weights are zero.
3. I configured the **Pruning Schedule** with a **Polynomial Decay function**, so that the sparsity of the model increases with the number of epochs from **0.0 to 0.40** (ie. **0 to 40% sparsity**)
4. **Update Pruning Step** call-back was used to update the pruning wrappers with the optimizer.

The impact of Pruning on my model size, performance and accuracy is as shown:

```
Pruned CNN - Test loss: 0.11053787916898727 / Test accuracy: 0.9528985619544983
Regular CNN - Test loss: 0.016787197440862656 / Test accuracy: 0.9963768124580383
Size of gzipped original model: 9047471.00 bytes
Size of gzipped pruned model: 6492795.00 bytes
```

The size of my model reduced 9 MB to 6.4 MB after pruning combined with a standard compression algorithm. Though accuracy has dropped slightly from 99.6% to 95%, the model still performs considerable well.

Reference:

1. https://www.tensorflow.org/model_optimization/guide/pruning/pruning_with_keras
2. https://www.tensorflow.org/model_optimization/guide/pruning
3. https://medium.com/@souvik.paul01/pruning-in-deep-learning-models-1067a19acd89
4. https://www.machinecurve.com/index.php/2020/09/23/tensorflow-model-optimization-an-introduction-to-pruning/