

**CSE 541 Computer Vision**

**Project 3: Enhanced Deployment of YOLO V8 on Nvidia Jetson Nano through Model**

**Compression Techniques**

**Group: Visionary minds**

**Team Members**

|  |  |
| --- | --- |
| Enrollment Number | Name |
| AU2140154 | Harsh Loriya |
| AU2140179 | Urmil Jagad |
| AU2140205 | Rahul Patel |

**Week one:**

Team formation and project selection.

**Week two:**

Meeting/ discussions with Jay Sir in the research lab. Gave us an idea of what we have to do and all the things we need to understand for the project.

**YOLOv8:**

* "You Only Look Once version 8," or YOLOv8, is a state-of-the-art computer vision method. Consider that you are attempting to identify various objects, such as people, automobiles, or pets, in a picture. Computers can now accomplish it much more quickly and accurately because of YOLOv8.
* This is how it operates: YOLOv8 looks at the entire image once, instead of looking at it several times to locate things (thus the name "You Only Look Once"). Next, it creates what are known as "grids" out of the image's smaller pieces and makes predictions about what might be in each grid and if any items are present.
* The speed and precision of YOLOv8 are absolutely great features. It is incredibly useful for activities like surveillance, self-driving cars, and even merely tagging friends in pictures on social media because it can recognize objects in real-time.

**JetSon Nano:**

* The Jetson Nano is a small but powerful computer specifically designed to help with artificial intelligence tasks, like computer vision, robotics, and more. Think of it as a mini-brain that can process information and make decisions quickly.
* The Jetson Nano is unique due to its compact size and high efficiency. It's compact enough to fit in your palm, but it has the processing power to run sophisticated AI algorithms. Because of this, it's ideal for integrating AI into a wide range of gadgets, including smart cameras, drones, robotics, and even household appliances.
* The Jetson Nano is a fantastic device, and one of its best features is its real-time deep learning models (YOLOv8). This implies that it is capable of instantaneously analyzing picture or video streams and deciphering visual cues, including object recognition, motion tracking, and gesture interpretation.

**Project Presentation.**

**Week three:**

Mid Sem preparation and exams.

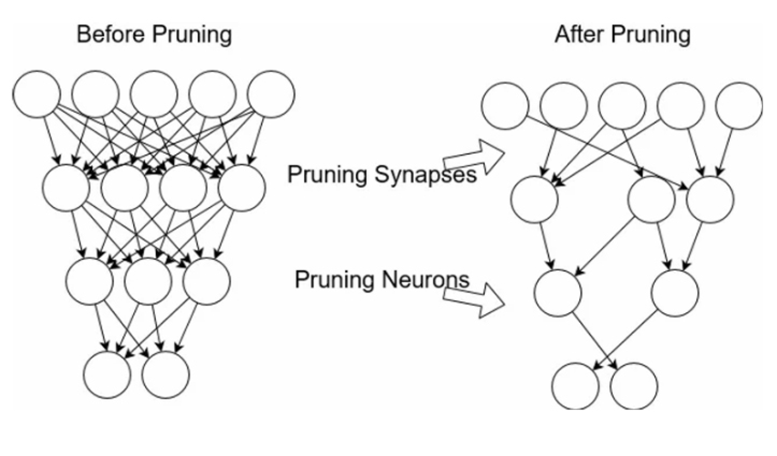
**Week four:**

More deep research on understanding of Model compression techniques:

Model Compression Techniques:

1. **Pruning:**

One effective method for lowering the amount of deep neural network parameters is pruning. Many parameters in DNNs are unnecessary as they don't really matter in the training process. Thus, these characteristics may be eliminated from the network after training with minimal impact on accuracy.

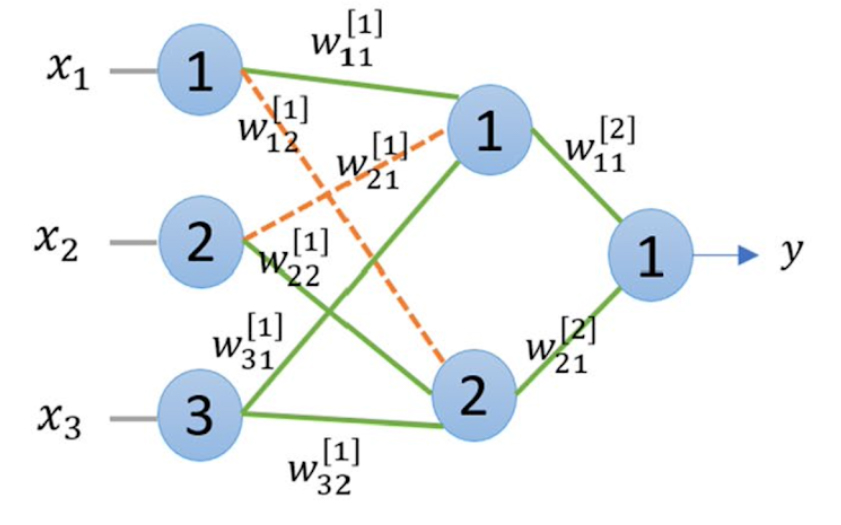


Models with pruning are compact and run faster. It lowers the total model size as well as the computational expense of network training. It is important that it also conserves energy and computation time.

Pruning a model can happen both during and after training. There are several methods for doing the pruning process.

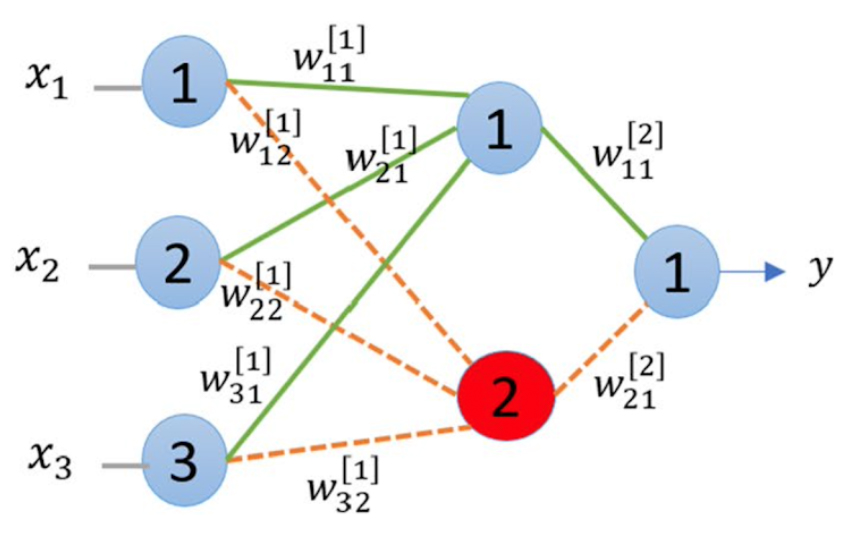
* **Weight Pruning:**

Some predefined thresholds are introduced, and weights below those thresholds are pruned.



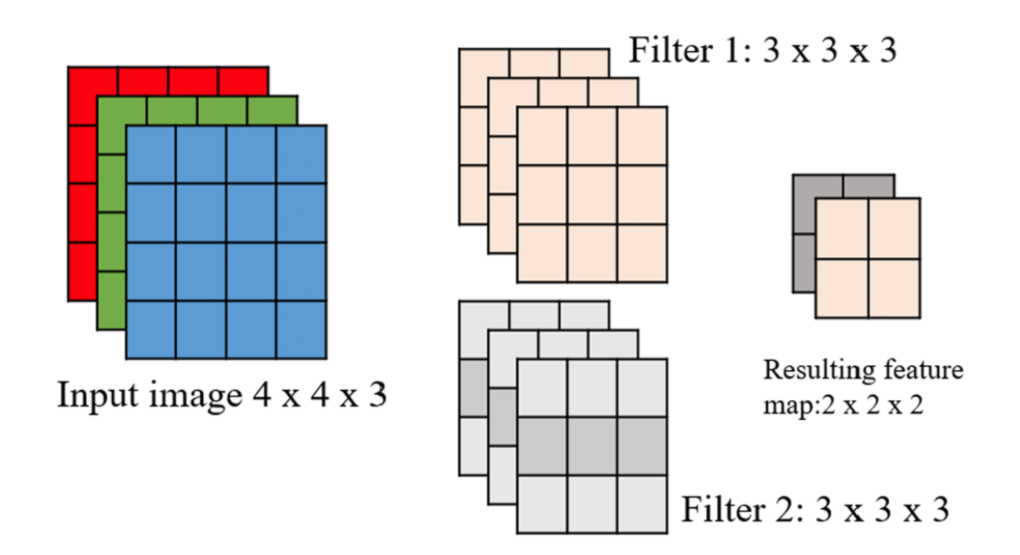
* **Neuron Pruning:**

Pruning the neurons rather than eliminating the weights.



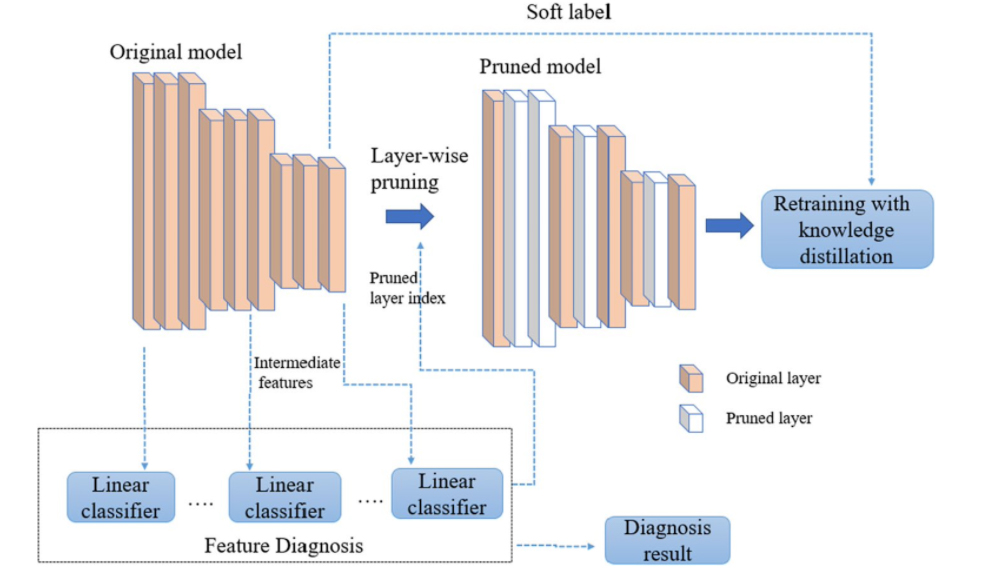
* **Filter Pruning:**

The least significant filters are eliminated from the network after being sorted in order of relevance. The L1/L2 norm can be used to determine the filters' significance.



* **Layer Pruning:**

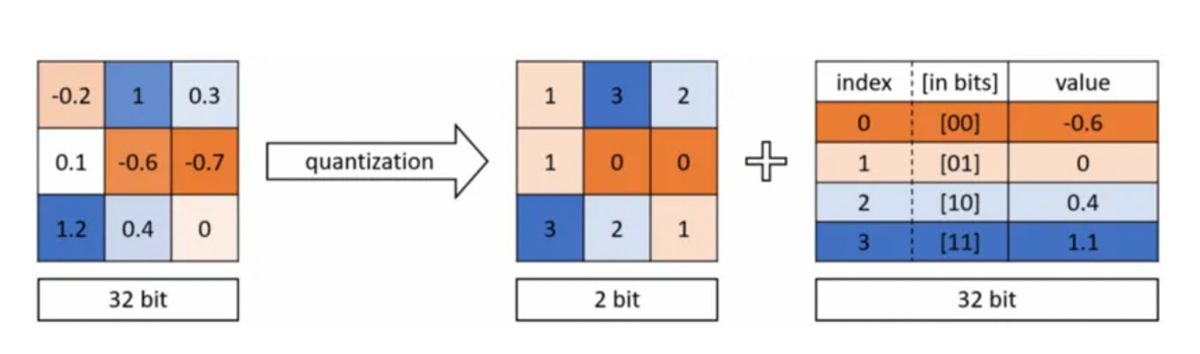
Layers can also be pruned



1. **Weight Quantization:**

In Deep Neural Network, weights are stored as 32-bit floating point numbers.

The original network is compressed by quantization, which lowers the number of bits needed to represent each weight. For example, the weights can be quantized to 16 bits, 8 bits, 4 bits, or even 1 bit. The size of the DNN can be significantly reduced by lowering the amount of bits used.

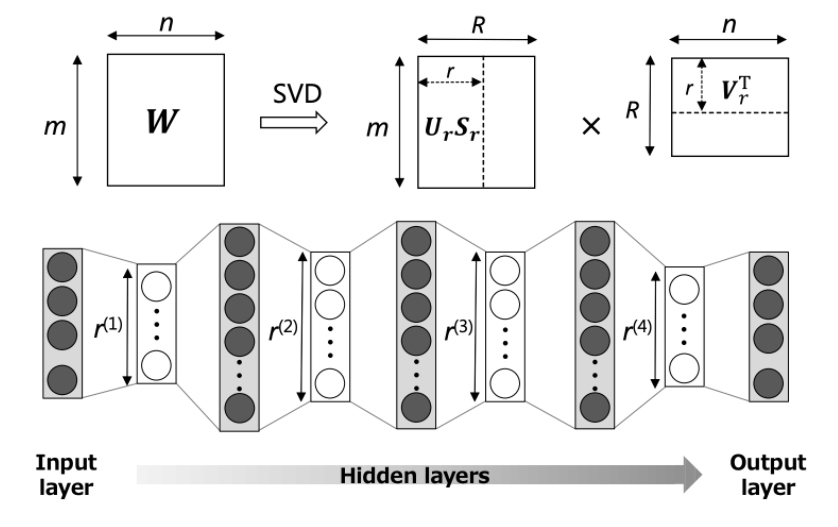


Quantization can be applied both during and after training. It can be applied to both convolutional and fully connected layers. However, quantized weights make neural networks harder to converge and make back-propagation infeasible.

1. **Low-Rank Factorization:**

Low-rank factorization identifies redundant parameters of deep neural networks by employing the matrix and tensor decomposition. A low-rank factorization technique helps by breaking down a large matrix into smaller matrices when it is necessary to reduce the size of the model.

A weight matrix A with m x n dimension and having a rank r can be decomposed into smaller matrices.



While the factorization of convolutional layers speeds up the inference process, the low-rank factorization of the dense layer matrices primarily reduces the amount of storage needed and makes the model storage-friendly.

Appropriate factorization and rank selection are essential for both model performance and accuracy. The primary difficulty in this case is that the decomposition process is highly computational and leads to more difficult implementations.

**Reference:**

Pokhrel, S. (n.d.). *4 popular model compression techniques explained*. https://xailient.com/blog/4-popular-model-compression-techniques-explained/#2\_The\_quantization\_technique