
Whitepaper on “Pruning Neural Network for Inferencing on Vitis-AI/DNNDK & FPGA” LogicTronix-WPL-053

As the computation power evolves more and more every year, state of art machine learning models are also getting larger, complex, and computation heavy. In the Cloud infrastructure and research environment computation requirement of machine learning models is not a priority over the accuracy of the models.

In recent years newer and better deep learning models are being developed which require more amount of computing, memory, and power. This can become a bottleneck in situations where real-time inferencing is required. Such cases arise in Edge and Embedded Applications. Edge Based Deployments don't get to enjoy the same level of computation and memory flexibility as seen in the Cloud Infrastructures. A lot of thought and compromises goes into finalizing the deployment in the edge cases to balance between model performance, available computing, and memory resources, and power efficiency. Such limitations could limit the neural network from achieving maximum performance.

LogicTronix specializes in the deployments of machine learning models at the edge. To achieve the right balance between the model performance, model accuracy, and power efficiency of the model a lot of different methods are utilized. Today we are going to show how pruning helps us in our process of deploying machine learning models that are smaller in size, memory and power efficient, and fast at inference with minimal loss in accuracy.

In a neural network, there are a large number of components and connections. Some of these connections, after a few iterations, become redundant and do not contribute much to the output of the network. These connections can be removed without impacting the accuracy of the model. Removing these connections is referred to as pruning.

Pruning is a way to reduce the size of the neural network through compression. The basic principles of pruning include removing unimportant weighted information using second derivative data. This results in better generalization results, improved speed of processing the results, and reduced size as well.

Among various pruning methods, we are currently using channel pruning to show the results. Channel Pruning is a kind of structural model compression approach which can not only compress the model size, but it can also accelerate the inference speed directly. Each layer can be pruned at a different ratio depending on the task but for the sake of simplicity Uniform Layer Pruning is used. Uniform Layer Pruning means each convolution layer is pruned with the same pruning ratio.

Top level illustration of how pruning works is shown in Figure 1.

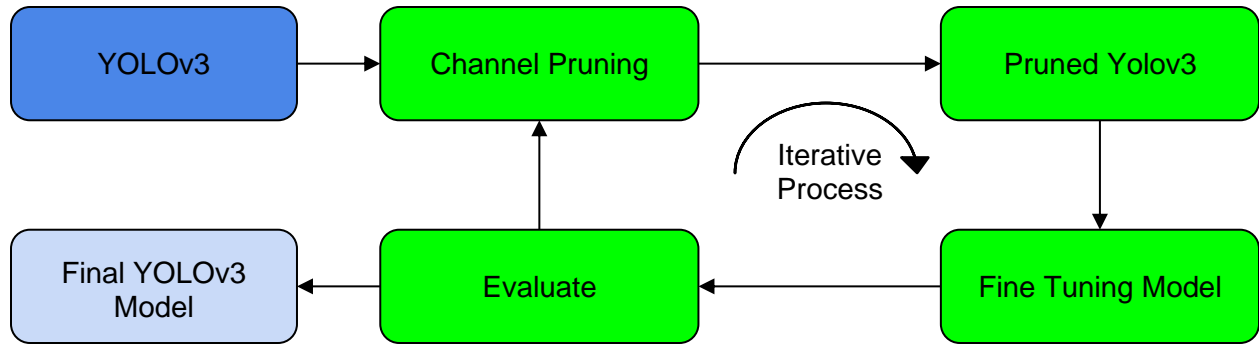


Figure 1: Pruning Process

Pruning Results

In this White Paper, we mainly focus on Object Detection and Recognition tasks using the Yolov3 model.

The results of channel pruning of three different models are shown in Table 1.

Table 1: Channel Pruning result of License Plate, BDD and Face+ Person Model

Model	Global threshold (Pruning ratio)	Input Size	GFLOPS	mAP	Size
License Plate Original Yolov3	-	512*512	98.901	86.09	246.3 MB
		640*640	154.532	88.83	
		960*960	347.698	84.16	
License Plate Pruned Model 1st Iteration	0.5	512*512	25.204	86	62.7 MB
		640*640	39.981	86.87	
		960*960	88.607	82.20	
License Plate Pruned Model 2nd Iteration	0.5	512*512	11.231	83.60	16.6 MB
		640*640	17.549	86.09	
		960*960	39.485	78.52	

BDD Model Original Yolov3	-	512*512	99	42.96	246.5 MB
		640*640	154.687	45.97	
		960*960	348.046	42.43	
BDD Pruned Model 1st Iteration	0.5	512*512	39.464	42.83	112.5 MB
		640*640	61.662	46.61	
		960*960	138.739	44.93	
Face+Person Original Yolov3	-	512*512	98.912	77.61	246.3
		640*640	154.549	76.95	
		960*960	347.736	62.26	
Face+Person Pruned Model 1st Iteration	0.5	512*512	40.001	70.12	113.8
		640*640	62.502	74.93	
		960*960	140.629	63.59	
Face+Person Pruned Model 2nd Iteration	0.5	512*512	18.254	76.25	50.1
		640*640	28.522	73.11	
		960*960	64.174	50.16	

The above table shows us the different models precision at different input sizes during pruning. All the models are trained on 512x512 image size and the input image sizes are changed only during inferencing. The computation requirement also increases as the input size is increased which is shown on the GFLOPS column. GFLOPS shows the computation requirement of a particular model.

We can see that as we pruned the models, the GFLOPS also decreases which results in improved inferencing speed.

Global threshold is a parameter that ranks each layer of the neural network and removes those channels in that layer which are below the specified threshold.

mAP determines the precision of the model and as we can see that the pruning did not affect the model's accuracy by a lot of margins. Even at the second iteration, the mAP loss is only up to 5 to 10% but the return in performance is up to 9 times more on some models.

Conclusion

In this white paper, we showed the results of Channel Pruning and how it affected the performance of the models. This work shows that the pruned model requires less GFLOPS which finally allows to implement the model on lower series of FPGA devices than without pruning.

In the next revision of this white paper, we will show some different methods as discrimination-aware channel pruning, weight sparsification and model quantization and see how these method affect our model performance. Availability of such techniques has allowed us to deploy even the compute heavy deep learning models on real time embedded systems.

Revision History

The following table shows the revision history of this white paper.

Date	Version	Details
May 1, 2020	v1.0	Initial Release

About LogicTronix

LogicTronix provide turnkey solutions to customers on accelerating Machine Learning algorithms for various applications including ADAS, Surveillance, Computer Vision, etc.

If you are interested in pruning the machine learning models to meet the performance requirements or to deploy a custom machine learning model using Xilinx DPU contact us at our email.