Deep Neural Network (DNN) Surgery in Edge Computing

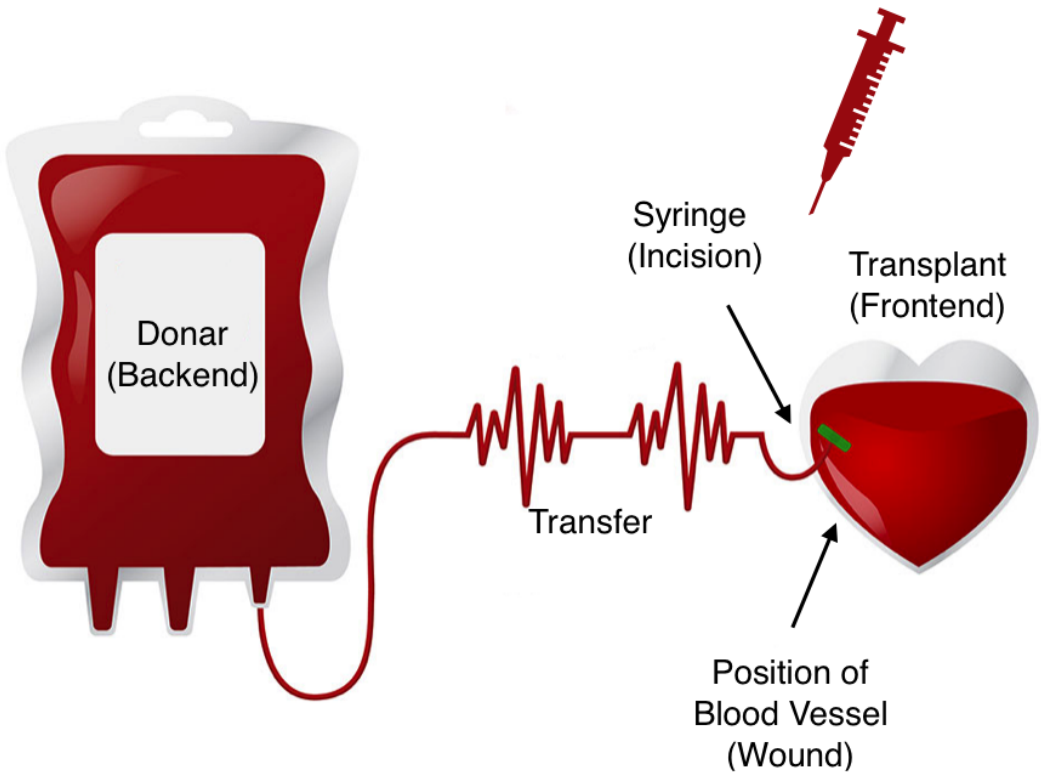
Yu Yang

University of Sydney

21 November 2018

Introduction

- Description: DNN Surgery executes part of inference at the edge and the rest is processed at the strong backend.
- Motivation: accelerate the inference speed with the device which has no strong computation power.
- Analogy with organ donation surgery or blood donation:



Various Architecture of DNN

- Sequential module

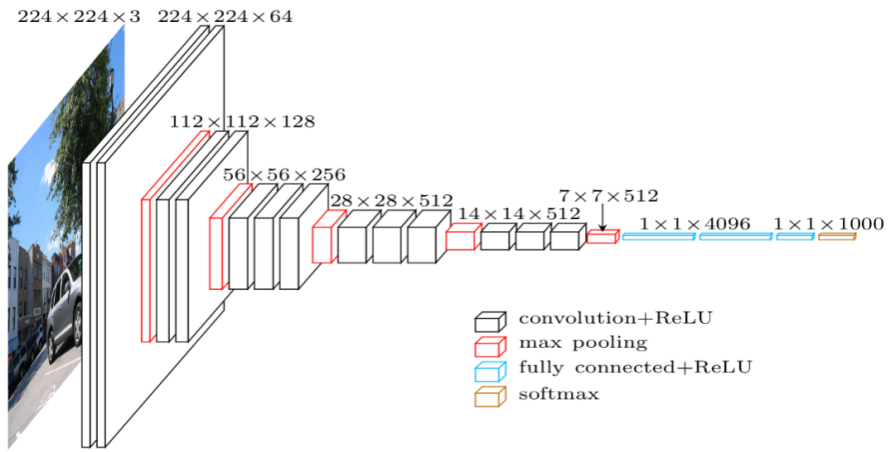


Figure 1: A visualization of the VGG architecture[1]

Various Architecture of DNN

- Residual module

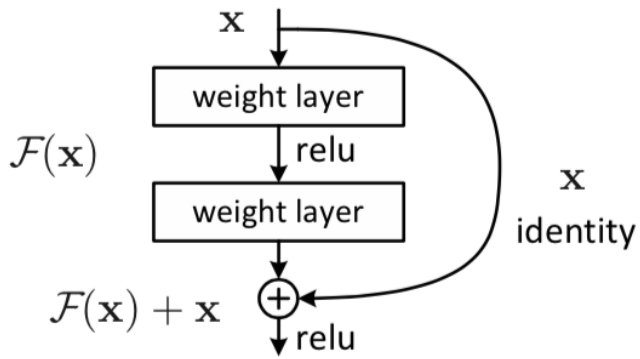


Figure 2: Residual learning: a building block[2]

Various Architecture of DNN

- Inception module

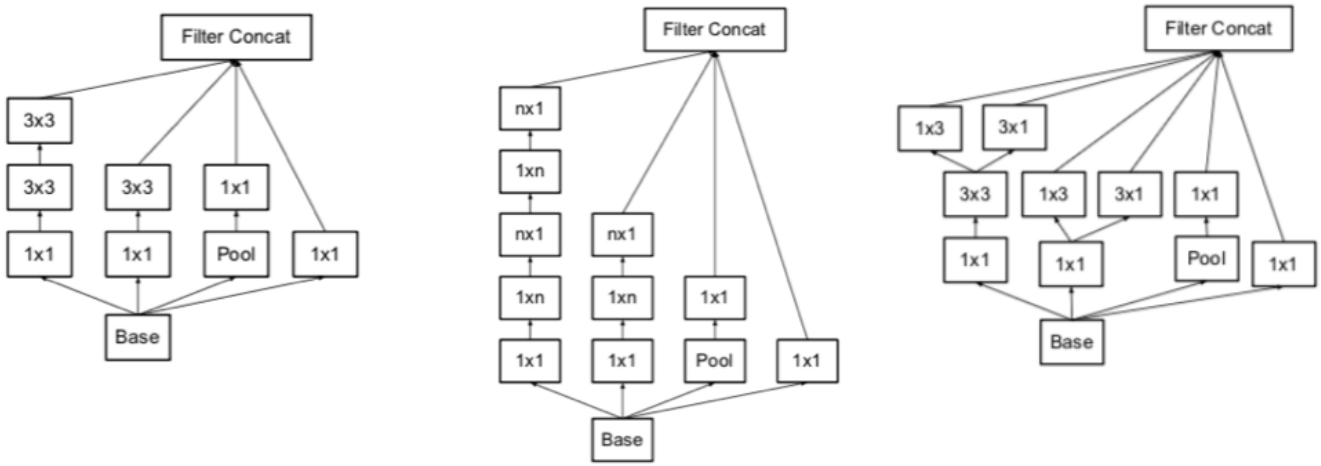


Figure 3: Inception variants in Inception-v2:Left:Inception modules where each 5x5 convolution is re-placed by two 3x3 convolution; Middle: Inception modules after the factorization of the nxn convolutions; Right:Inception modules with expanded the filter bank outputs[3]

Various Architecture of DNN

- Inception-ResNet Module

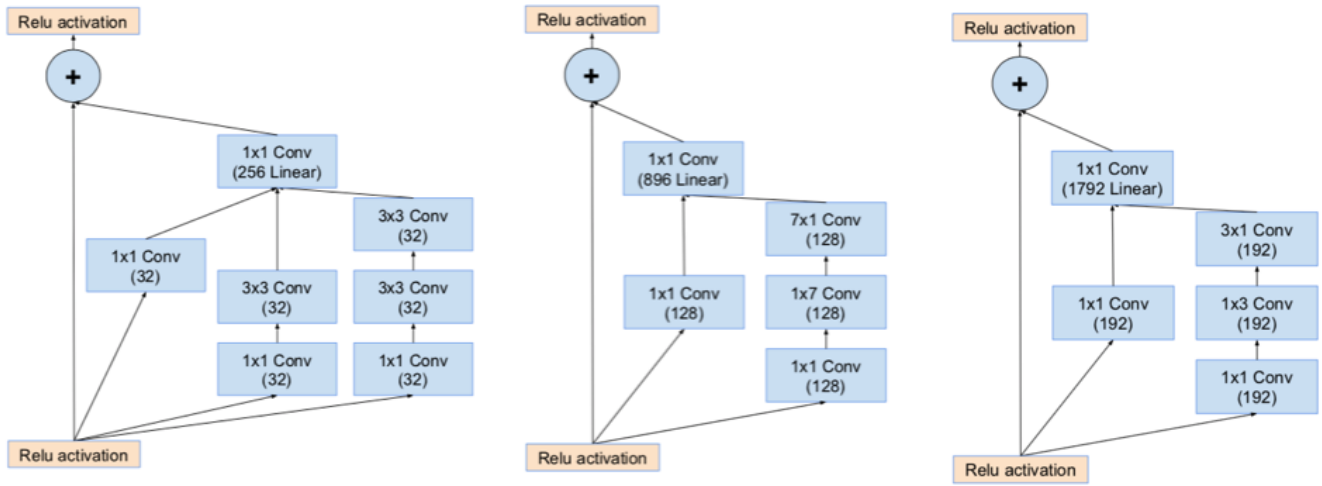


Figure 4: Inception variants in Inception-ResNet-v1 network: Left:The schema for 35 $\times$ 35 grid (Inception-ResNet-A) module; Middle: The schema for 17 $\times$ 17 grid (Inception-ResNet-B) module; Right:The schema for 8$\times$8 grid (Inception-ResNet-C) module[4]

Comparison of DNN Framework

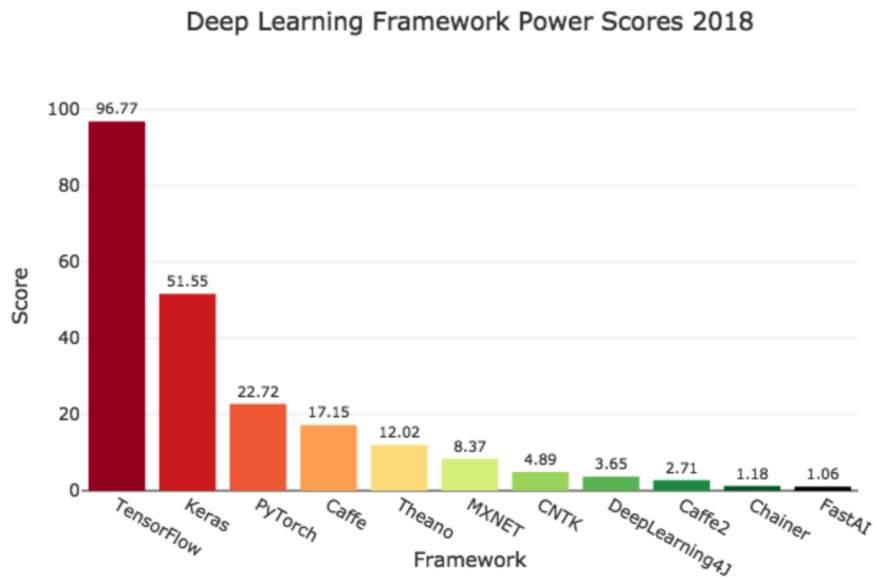
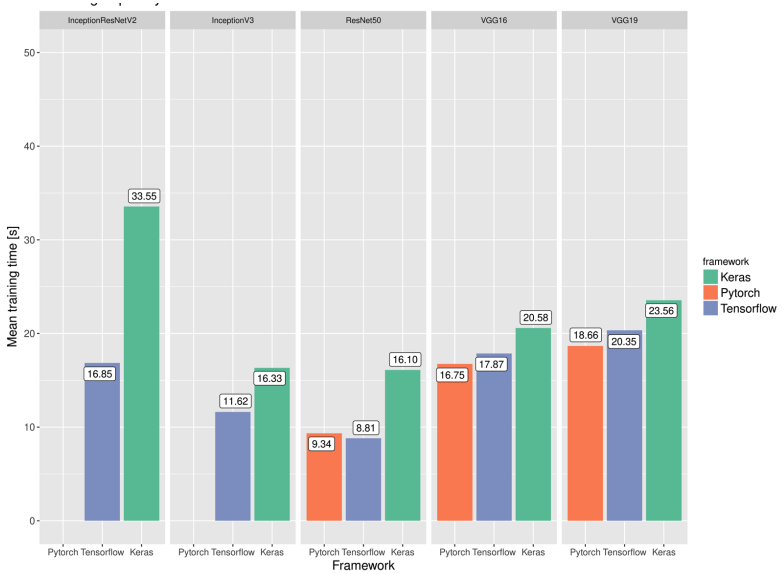
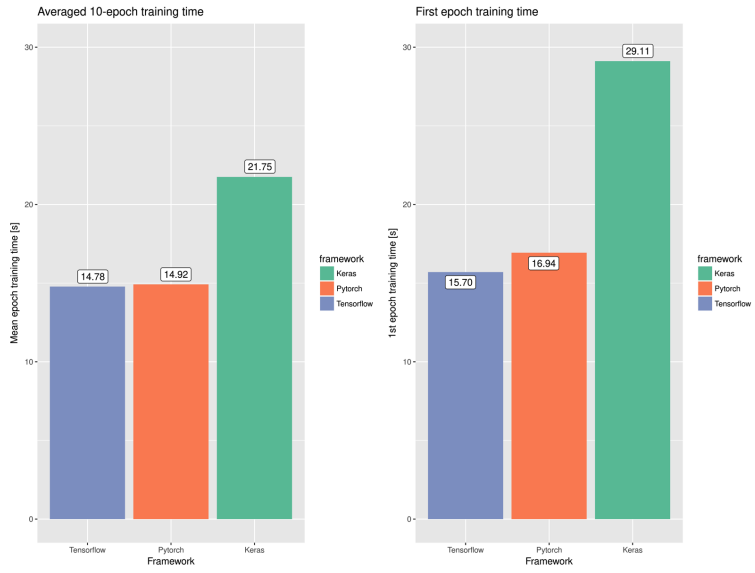


Figure 5: Deep Learning Framework Power Scores in 2018[5]

Comparison of DNN Framework



(a) Framework compared by models[6]



(b) Compare the first epoch and mean training time[6]

Figure 6: Comparison of DNN Framework

Inference Experiments of Model

- Comparison of Load Time

Table 1: Comparison Load Time between Keras and Tensorlow

model	load time in keras(s)	load time in tensorflow(s)
VGG19	3.7	3.3
Resnet50	8.3	6.6
Inception-v2	36.2	19.6
NASNetLarge	54.4	35.9
NASNetMobile	41.1	23.6
MobileNet-v2	3.6	3.0

Inference Experiments of Model

- Comparison Size between Models

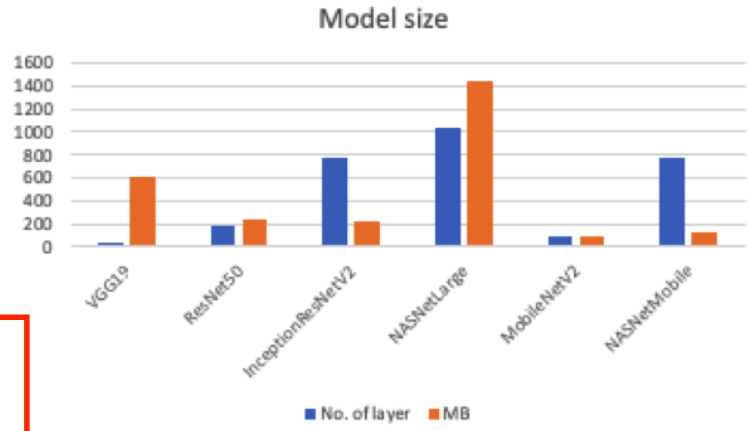
Table 2: Comparison Size between Models

Model	parameter	Trainable parameter	Nontrainable parameter	No. of layer	Total memory	MB
VGG19	143667240	143667240	0	26	640938048	611.246
ResNet50	25636712	25583592	53120	177	251027696	239.399
InceptionResNetV2	55873736	55813192	60544	782	224484300	214.085
NASNetLarge	88949818	88753150	196668	1041	1518039956.0	1447.716
NASNetMobile	5326716	5289978	36738	771	132069684	125.951
MobileNetV2	425386	4231976	21888	93	85015948	81.078

Inference Experiments of Model

- Comparison of Model According to Accuracy and Size

Model	No. of layer	MB	ImageNet validation		Accuracy ranking	Sort by size
			Top-1 Accuracy	Top-5 Accuracy		
VGG19	26	611.246	0.713	0.900	6	1
ResNet50	177	239.399	0.749	0.921	4	3
InceptionResNetV2	782	214.085	0.803	0.953	2	5
NASNetLarge	1041	1447.716	0.825	0.960	1	6
MobileNetV2	93	81.078	0.744	0.919	3	2
NASNetMobile	771	125.951	0.713	0.901	5	4



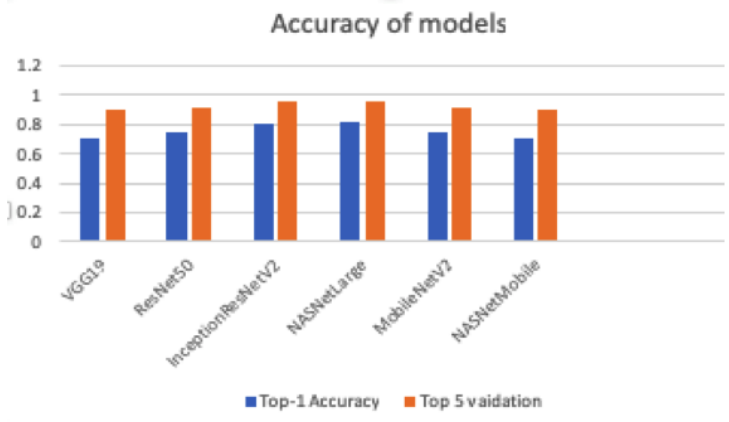
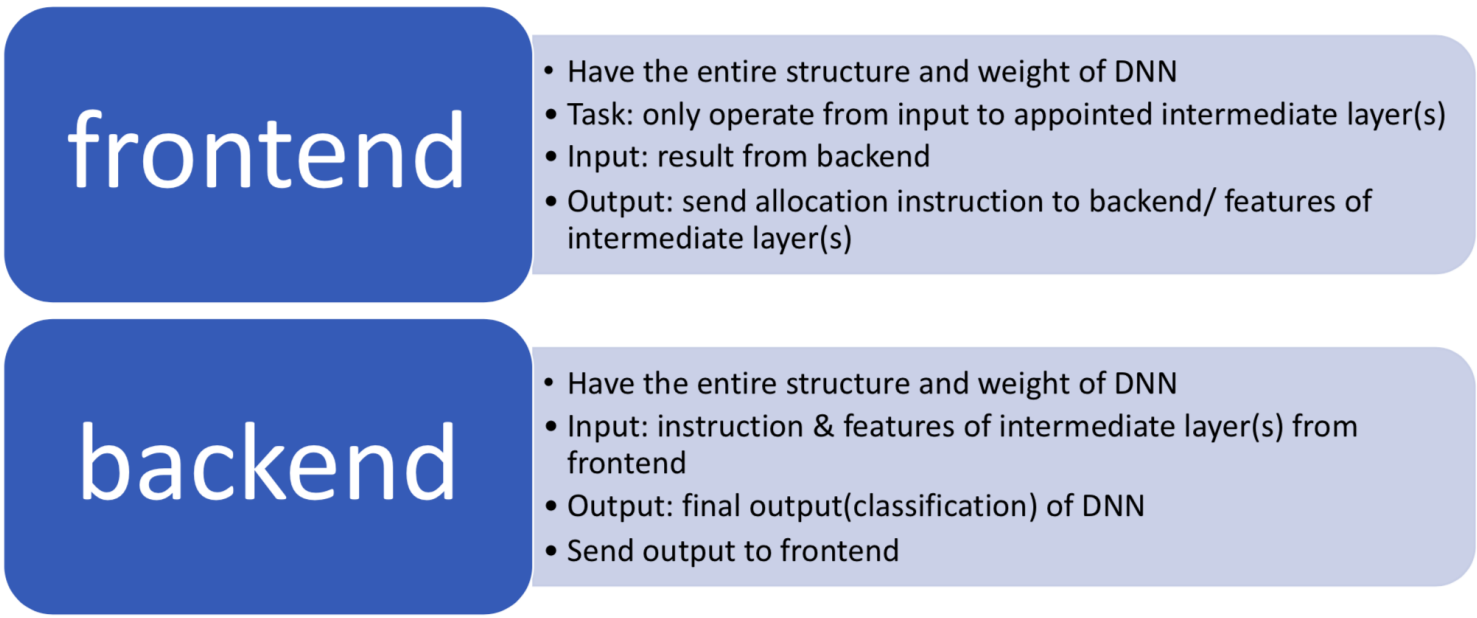


Figure 7: Comparison between Models

Introduction of Surgery

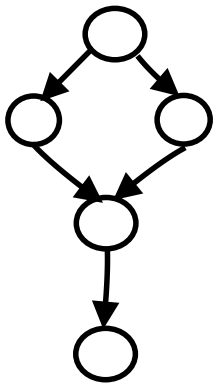
- Main Architecture of Surgery



Introduction of Surgery

- Software and Hardware Specification of the Experiment Used
  1. Frontend: raspberry pi 3()
     - Model B 16GB NOOBS
     - SAMSUNG microSD 16GB
  2. Backend: Mac Pro
     - Processor 2.9 GHz Intel Core i5
     - Memory 8 GB
  3. Python version: python 3.6.6
  4. IDE: pycharm
  5. Tensorflow: 1.11.0
  6. Keras: 2.2.2

Introduction of Surgery

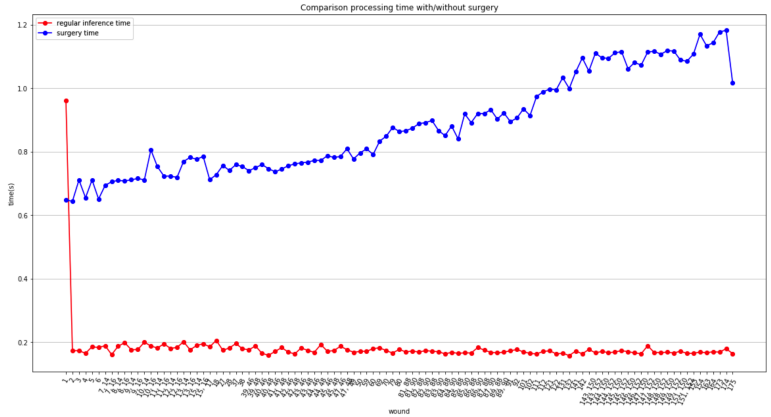


- Support Module
  - Graph model: plot model with index(figure human organs)
  - Incision support module:
    1. wound generation (any organ that might cause disease): generate candidate wound based on the architecture of model(include ring detector)
    2. wound to incision (from discussion to practice)
    3. cut generate (ready to do surgery)
    4. feasible wound generation (candidate operation plan)

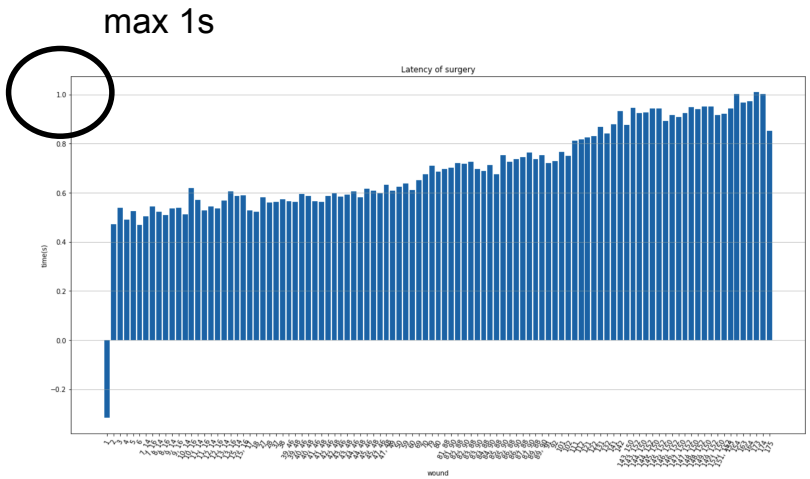
Figure 8: Analog of Medical Surgery[7]

Prerequisites of surgery

- Comparison of Surgery Times and Inference Time



(a) Surgery Times and Inference Time

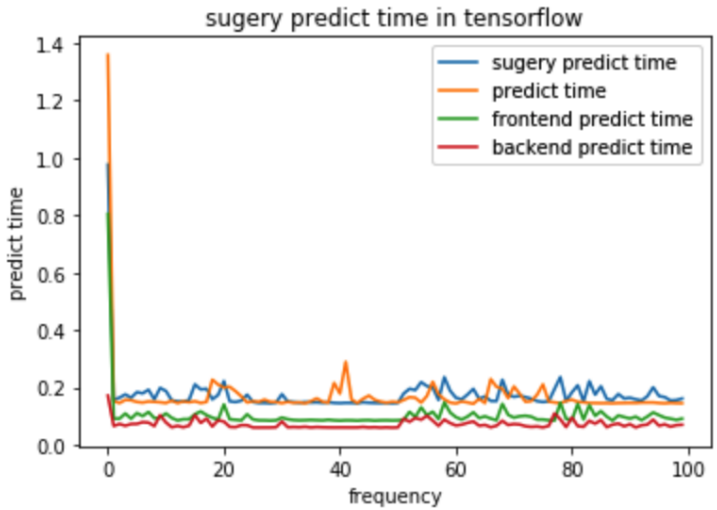


(b) Surgery Latency

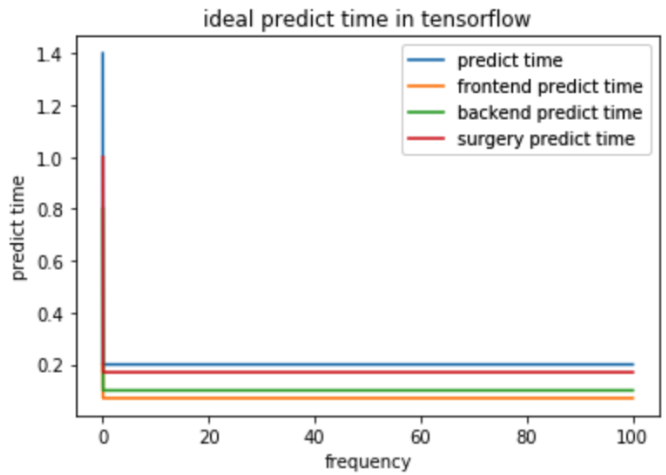
Figure 9: Comparison of Surgery Times and Inference Time

Prerequisites of Surgery

- Uncertainty of Program Running Time



(a) Surgery Times



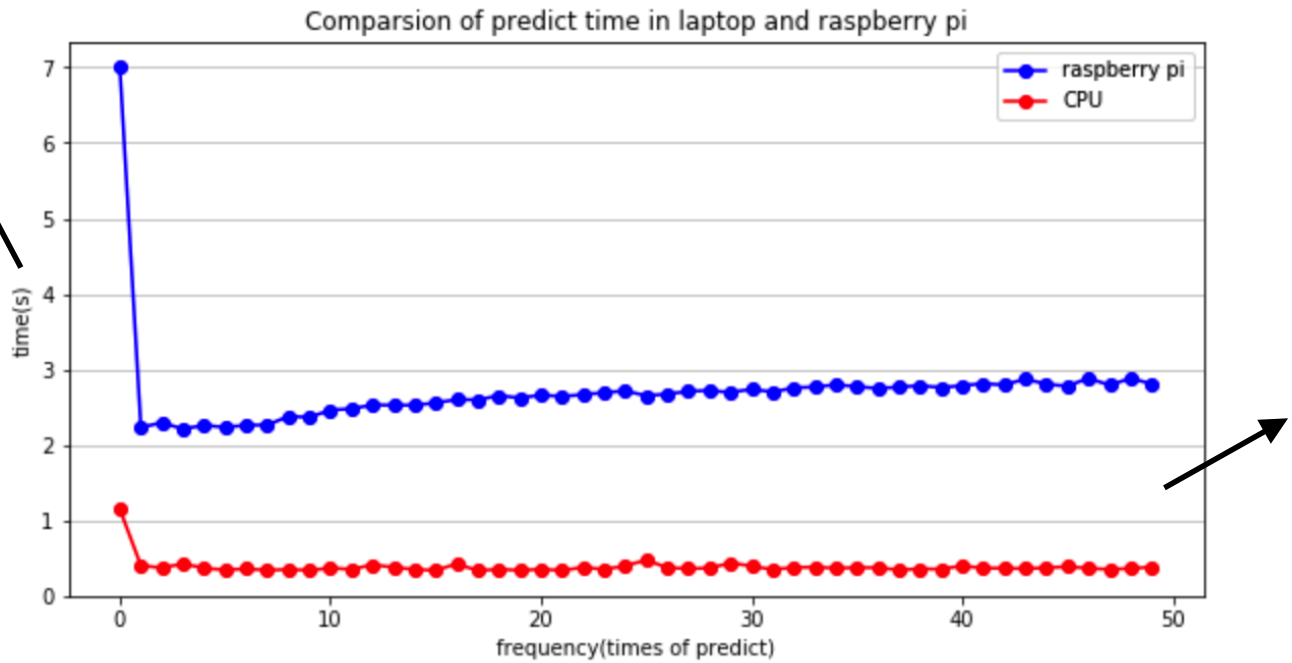
(b) Ideal Surgery Time

Figure 10: Comparison between Surgery Times and ideal surgery time

Prerequisites of Surgery

- Different Surgery Time in Laptop and Raspberry pi

It takes more time during the first time



expected acceleration: 1-2s

Figure 11: Different surgery time in laptop and Raspberry pi

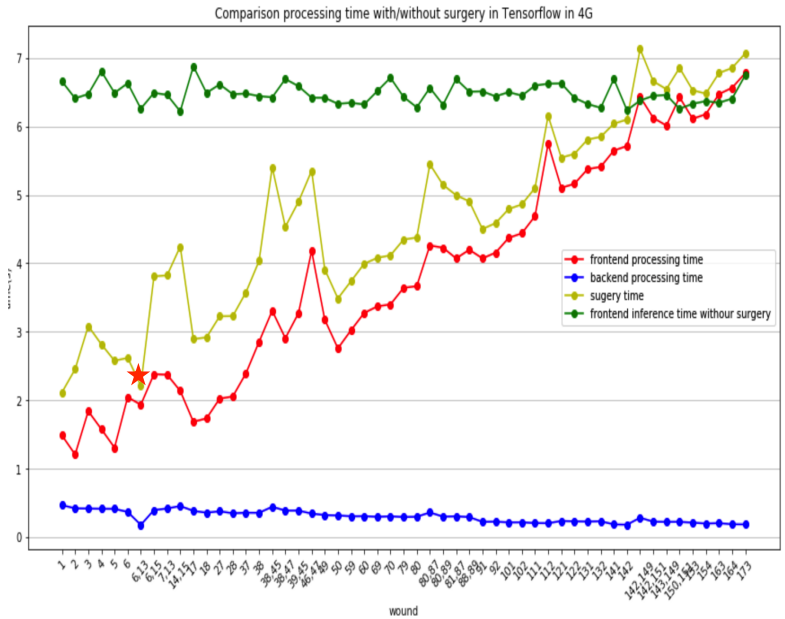
Prerequisites of Surgery

- Ideal transfer environment:
  - Downlink Speed(α):100 Mbps
  - Uplink Speed(β): 45 Mbps
  - intermediate feature size(byte)(γ)
  - result size(δ): 25 bytes in Resnet 50
  - Transfer time(T):

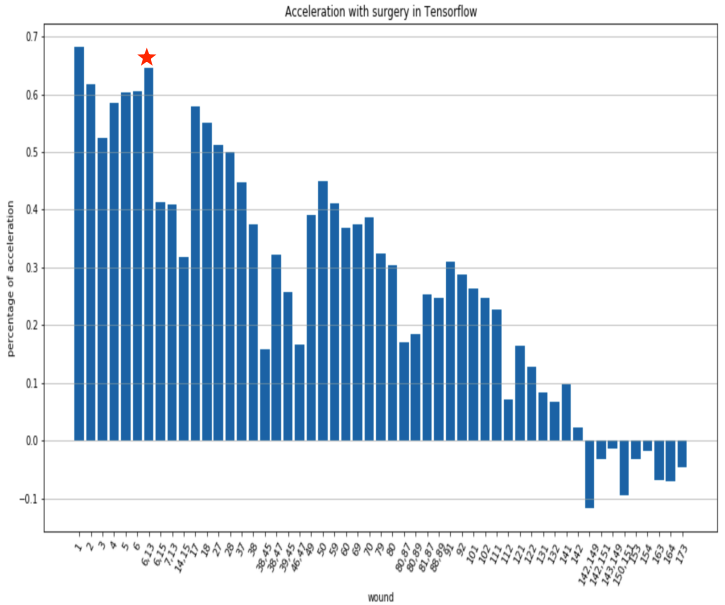
$$T = (\delta + \gamma)/1024/(\alpha * 125) + (\delta + \gamma)/1024/(\beta * 125)$$

Find the Desirable Wound in Resnet50

- Execute Each Surgery Once:



(a) Comparison processing time with/without surgery



(b) Accelerate time with surgery

Figure 12: Execute each surgery once to find out the desirable Incision

Find the Desirable Wound in Resnet50

- Minimum Speed(Mbps)

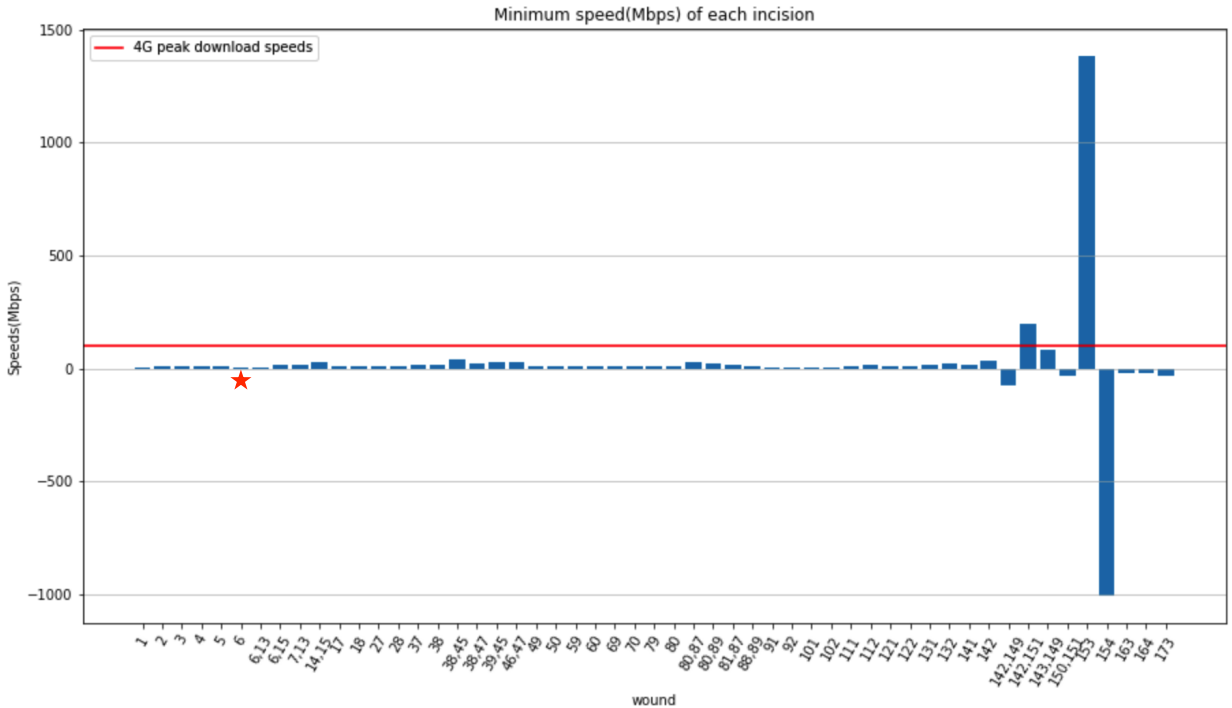
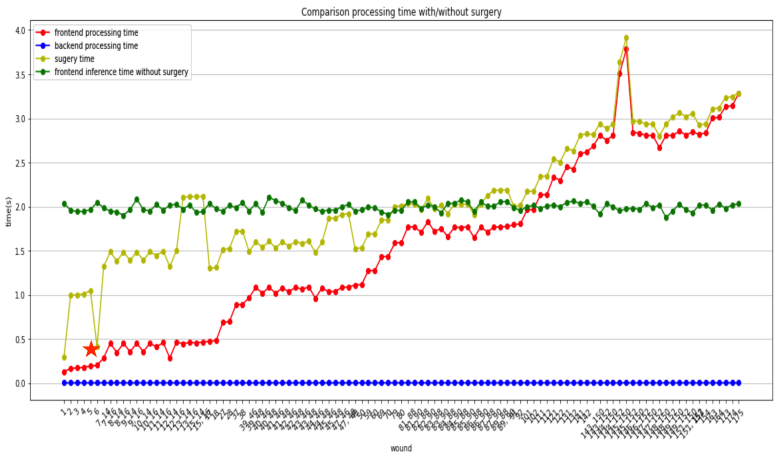


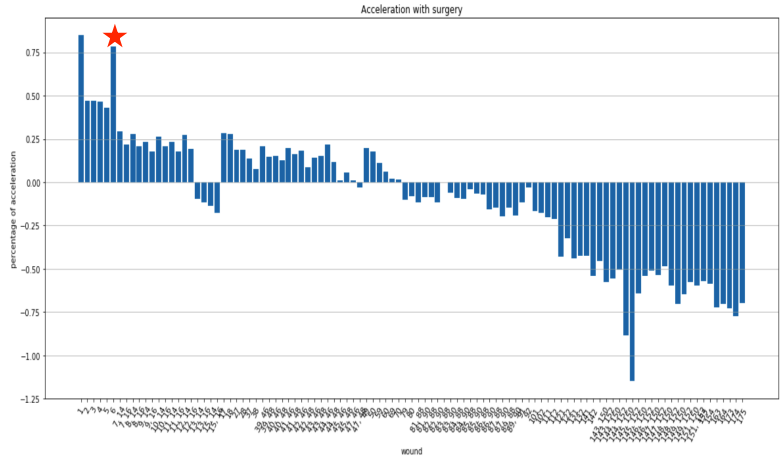
Figure 13: Minimum speed(Mbps) of each wound

Find the Desirable Wound in Resnet50

- Execute each surgery 100 time, and compute the average from second time to 100th time



(a) Comparison processing time with/without surgery



(b) Accelerate time with surgery

Figure 14: Execute each surgery 100 time to find out the desirable Incision

Find the Desirable Wound in Resnet50

- Minimum speed(Mbps)

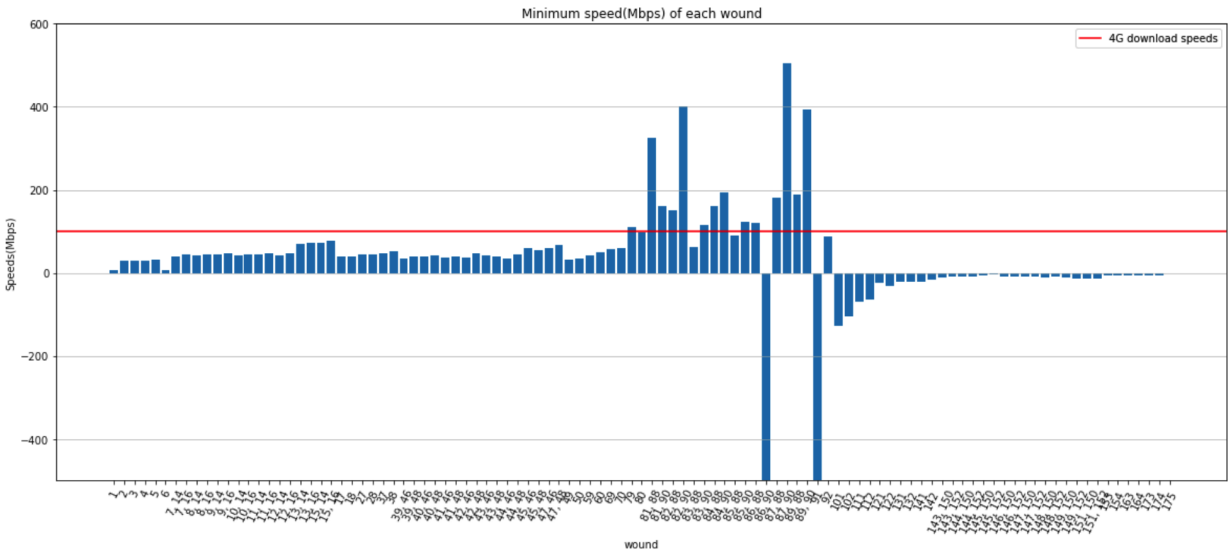


Figure 15: Minimum speed(Mbps) of each wound

Conclusion

- DNN Surery is practicable of model with residual module and inception module in 4G network
- In Resnet50, the optimal wound is after the 6th layer/before 7th layer
- There are 50 desirable wounds before the 70th layer, which means as long as executing an effective surgery before 70th layer, DNN Surery could accelerate the inference speed more or less.

References

- K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *ArXiv preprint arXiv:1409.1556*, 2014.
- K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.", in *AAAI*, vol. 4, 2017, p. 12.
- *Deep learning framework power scores 2018*, ://https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a.