

Wei Gao

Curriculum Vitae

Beihang University

Beijing, China

☎ (+86) 178 01004314

✉ weigao1996@outlook.com



Education Experience

- 2014–2015 **School of Aerospace, Beihang University**, Beijing.
- 2015–2019 **Computer Science and Technology, Beihang University**, Beijing, GPA:3.78/4.0.
Rank:8/230

Awards

- 2015–2016 **BUAA Fengru Cup Innovation Contest**:First Prize
- 2016–2017 **BUAA Fengru Cup Science and Technology Works Competition**: Third Prize
- 2016–2017 **ACM-ICPC Regional Programming Contest(Shenyang)**:1st in Silver Metal
- 2017–2018 **Sensetime Outstanding Internship**

Internship Experience

- Oct 2018 – **Research Intern**, PEKING UNIVERSITY, PROF. ZHANXIN ZHU, Beijing, China.
Now Bachelor Thesis: The fast training optimization algorithm research of student neural network.
Research Area: Theoretical Research of Shallow Layer Neural Network Representation.
- Aug 2018 – **Visiting Scholar**, UC MERCED, PROF. MING-HSUAN YANG, Merced, US.
Oct 2018 Fast Compacted Multi Video Style Transfer (will Submit to IJCAI 2019)
- Sept 2017 – **Research Intern**, SENSETIME, JUNJIE YAN, LI QUANQUAN.
June 2018 Research in compressing neural networks and designing efficient neural networks
- June 2017 – **Research Intern**, BEIHANG UNIVERSITY, PROF. JIANLEI YANG.
Sep 2018 Research in compressing neural networks and running compressed model on Nvidia tx2 and FPGA

Paper

- Technical **Pruning with Hints: A framework for model acceleration**, Beijing, China.
Report **Author:** Wei Gao, Yi Wei, Quanquan Li, Hongwei Qin, Wanli Ouyang, Junjie Yan

Projects

Runing LeNet on FPGA, Verilog, C.

Implement Convolution layer, Pooling layer, InnerProduct layer by Verilog HDL and Write C code control Zynq runing LeNet. It gives me a new view to think about the way of acceleration of model.

NetAdapter-implementation, PyTorch.

Pruning model with a greedy search strategy. Implement for Paper **NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications(ECCV2018)**. Utilize Multi-process distribute training to finetune generated child network. I achieve a comparable performance with the result of paper.

3D-Autoencoder, *PyTorch*.

To my best of knowledge, I firstly design a 3D autoencoder framework. I planned to apply 3D-autoencoder in video style transfer task but failed. I utilize C3D(From Paper C3D: Generic Features for Video Analysis) as encoder. For decoder design and training, I learn a lot of tricks about 3D-Decoder.

ResNet50 Channel-Pruning Model, *Matlab, Caffe*.

Pruning model with lasso and least-square method.Implement for Paper **Channel Pruning for Accelerating Very Deep Neural Networks**(ICCV2017) on Resnet50(ImageNet).I achieve 2x acceleration and Top-5 Accuracy reduces 3.11(from 92.98 to 89.87). From this project, I know better about the advantage of Channel-Level Pruning strategy and the importance of reconstructing weights.

Taylor-Pruning Model, *PyTorch*.

Pruning model with minimum loss.Implement for Paper **Pruning Convolutional Neural Networks for Resource Efficient Inference**(ICLR2017) on VGG, ResNet, Inception Struture and Classification, Key Point, Detection Tasks. I use the flops constrain and weight reconstruction to enhance the result from 83 to 85.304(Top5@Accuracy) as well. Now, the algorithm has been used in products developed by SenseTime.

Adaptive Pruning Framework, *PyTorch*.

Designing a framework which can be easily used to accelerating most existing neural network model. For users without any accelerating context, they can acquire an accelerated model by adding no more than 5 line code.