ECE763 Computer Vision: Models, Learning and Inference (including Deep Learning)

- HW1: out 01/17, due 01/31, [DONE]
- HW2: out 02/06, due 02/19 [DONE]
- HW3: out 02/27, due 03/16 [DONE]
- ~~HW4: out 03/17, due 03/31 [cancelled]~~
- HW5: out 04/01, due 04/15 [POSTED]
- Project1: out 02/04, due 02/27 [DONE]
- Project2: out 02/27, due 03/22 [DONE]
- Project3: out 04/01, due 04/30 (**no late days allowed**)

Instructor: Tianfu (Matt) Wu

NC STATE UNIVERSITY Electrical and Computer Engineering

Instructions and Notes

- *How to submit your solutions*: put your report (word or pdf or powerpoint) and results images (.png) if had in a folder named [your_unityid]_proj03 (e.g., twu19_proj03), and then compress it as a zip file (e.g., twu19_proj03.zip). Submit the zip file through **moodle**.
- *If you miss the deadline and still have unused late days, please send your zip file to TAs and me (0.5-day counted if later <= 6 hours based on email timestamp).*
- **Important Note**: We will **NOT** accept any replacement of submission after deadline ([+late days you use]), even if you can show the time stamp of the replacement is earlier than the deadline. So, **please double-check whether you submit correct files**.
- NOT LATE DAYS ALLOWED!

Instructions and Notes

- Project 3: 40%
  - Requirement:
    - final write-up (15%) and
    - self-contained reproducible code (25%): if your code is not reproducible, you will loose 20%; So, please make sure what you report in the write-up are consistent with the code.
  - Bonus points: 10
    - For novel ideas (including novel implementation of some existing techniques, e.g., in a significantly faster way under fair comparisons).
    - If you want to claim this, please add a **section 0** in your write-up to clearly present and justify **what's new. The novelty will be checked and judged by both TAs and the instructor.**

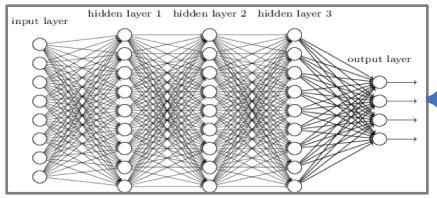
Babysitting the training of DNN

- **Problem**: Practice the babysitting method of training neural network as discussed in Lecture note 21-22-23 (more specifically, slide 76-slide 94).
  - **Step 0:** Data (you can start with existing data provided in different DL code platform, e.g., MNIST in PyTorch), and you must test your code using the face data you used in project 01.
  - **Step 1**: preprocess the data and select a data augment scheme. (Note: you can compare w/ and w/o this preprocessing step and the data-aug to see how they affect your training and testing performance)
  - **Step 2**: Choose the architecture. E.g. you can use something simple, a 3-layer FC network or the LeNet 5 (available in all deep learning platforms).
  - > Then please follow the slides to output detailed information of this babysitting procedure.

- **Hint**: You can reuse the tutorial code in different deep learning platform (e.g., the MNIST / CIFAR10 tutorial is available in almost all platforms).
  - E.g., If you use pytorch, you can find a lot of examples at https://pytorch.org/tutorials/
- **Requirement**: Although you can reuse the tutorial code, you need to modify the code to output different babysitting information (following slides 76-94 in lecture note 21-22-23). You need to snap your screenshots and paste those in your reports as shown in the slides. You also need to provide **self-contained reproducible code** .

Bonus Points for Creativity and Novelty

- Exploring new designs in, but not limited to, three aspects as follows.



"Representation": the space of neural architectures (*going deeper and/or wider*) → LeNet5, AlexNets, GoogLeNets, ResNets, ResNeXts, DenseNets, DualPathNets, and AOGNets, etc.

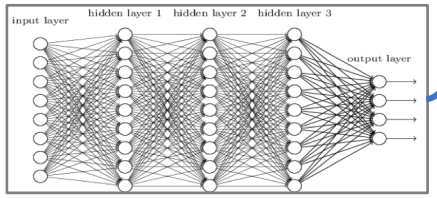
"Representation": the space of operations → vanilla conv., depthwise conv., separable conv., Atrous conv., deformable conv., different variants of ReLU, etc.

"Learning": the space of parameters → Initialization, Normalization, Pre-training,etc.

"Learning": the space of regularizer → Data augmentation (appearance and geometry transformation), DropOut, Weight Decay, etc

Bonus Points for Creativity and Novelty

- Potential Topic 1: New design of network architectures



"Representation": the space of neural architectures (*going deeper and/or wider*) → LeNet5, AlexNets, GoogLeNets, ResNets, ResNeXts, DenseNets, DualPathNets, and AOGNets, etc.

- Understand existing neural architectures
- Be critical and creative to think about different designs
  - Exploring the space of DAGs for network topologies.
  - E.g., exploiting your background domain knowledge, if you work in the circuit design field, or if you are familiar with molecular structures in chemistry or fluid dynamics in physics or theoretical studies of DAGs in mathematics, you can leverage different principles of designing efficient and effective "information flows" in those fields. **[Be creative!]**
  - Some baselines: you can think about how to combine different existing designs together.
- Validate your ideas on CIFAR10 (and/or CIFAR100)
- If your ideas are very promising, I will try to help you test on ImageNet using a large GPU server.
  - If the results are good, we can submit to computer vision/AI conferences.

Bonus Points for Creativity and Novelty

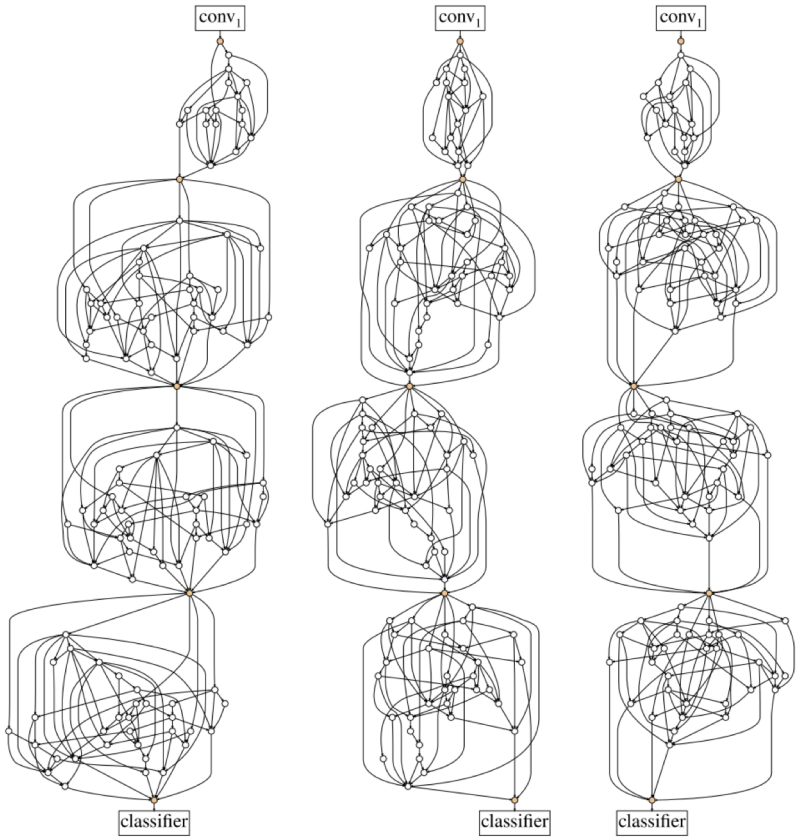
- Potential Topic 1: New design of network architectures
- Examples to inspire you

Exploring Randomly Wired Neural Networks for Image Recognition

Saining Xie Alexander Kirillov Ross Girshick Kaiming He

Facebook AI Research (FAIR)

Google and Read the paper



Bonus Points for Creativity and Novelty

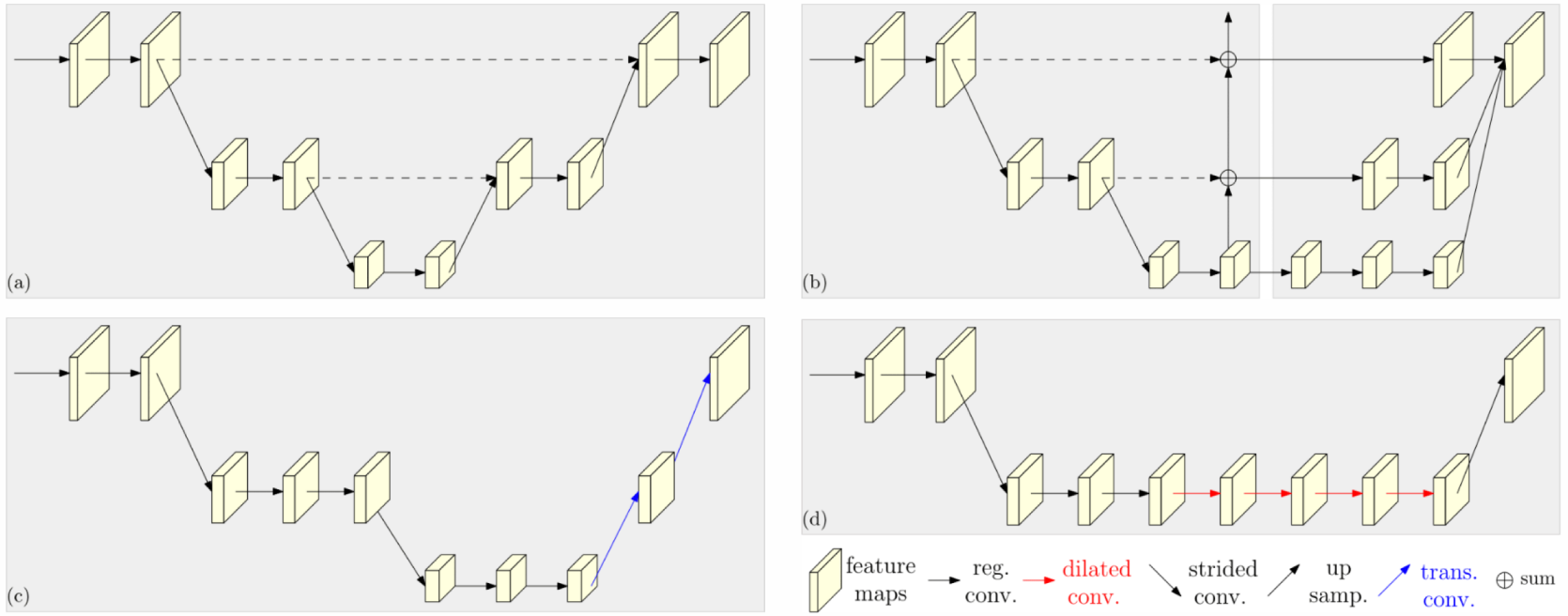
- Potential Topic 1: New design of network architectures
- Examples to inspire you

Deep High-Resolution Representation Learning for Human Pose Estimation

Ke Sun Bin Xiao Dong Liu Jingdong Wang

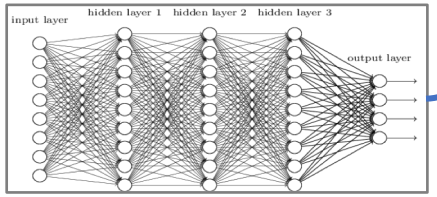
Google and Read the paper

(a) Hourglass [1]; (b) Cascaded pyramid networks [2]; (c) Simple baseline [3]; (d) Combined with dilated convolutions [4].



Bonus Points for Creativity and Novelty

- Potential Topic 2: New design of node operations



"Representation": the space of operations → vanilla conv., depthwise conv., separable conv., Atrous conv., deformable conv., different variants of ReLU, etc.

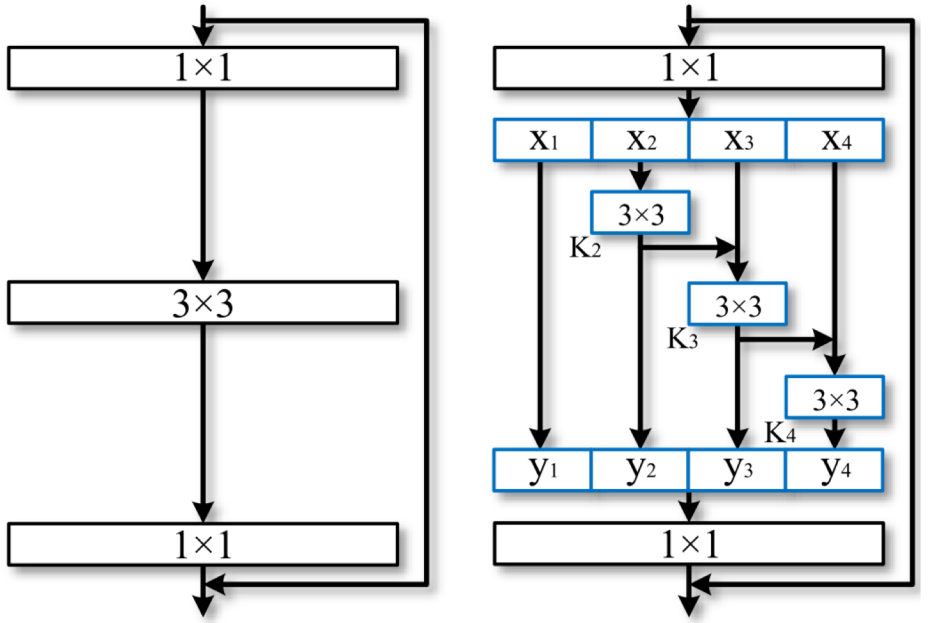
- Understand existing operations
- Be critical and creative to think about different designs
  - Consider existing network architectures such as ResNets or DenseNets (PyTorch has default implementation), and how to extend their operations (e.g., they use the Bottleneck operation)
  - Exploring the combinations of different primitive operations using some micro-topologies similar in the spirit to network-in-network (https://arxiv.org/abs/1312.4400).
  - Examples (understand how they improved the existing networks):
    - SENet: https://arxiv.org/abs/1709.01507
    - Non-local networks: https://arxiv.org/abs/1711.07971 https://arxiv.org/abs/1811.11721
    - Image Transformer: https://arxiv.org/abs/1802.05751
    - HetConv https://arxiv.org/pdf/1903.04120.pdf
  - Exploiting your background domain knowledge on methods of how to explore spatial and channel contextual information, or attention mechanism for operation design. **[Be creative!]**
- Validate your ideas on CIFAR10 (and/or CIFAR100)
- If your ideas are very promising, I will try to help you test on ImageNet or other big datasets using a large GPU server.
  - If the results are good, we can submit to computer vision/AI conferences.

Bonus Points for Creativity and Novelty

- Potential Topic 2: New design of node operations
- Examples to inspire you

Res2Net: A New Multi-scale Backbone Architecture

Shang-Hua Gao*, Ming-Ming Cheng*, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr



(a) Bottleneck block (b) Res2Net module

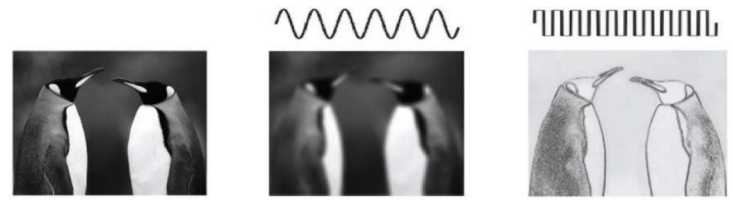
Fig. 2: Comparison between the bottleneck block and the proposed Res2Net module (the scale dimension $s = 4$).

Bonus Points for Creativity and Novelty

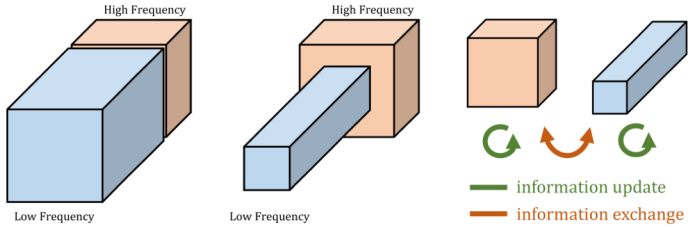
- Potential Topic 2: New design of node operations
- Examples to inspire you

Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks with Octave Convolution

Yunpeng Chen[†‡], Haoqi Fang[†], Bing Xu[†], Zhicheng Yan[†], Yannis Kalantidis[†],
Marcus Rohrbach[†], Shuicheng Yan[‡♭], Jiashi Feng[‡]
[†]Facebook AI, [‡]National University of Singapore, [♭]Qihoo 360 AI Institute



(a) Separating the low and high spatial frequency signal [1, 12].

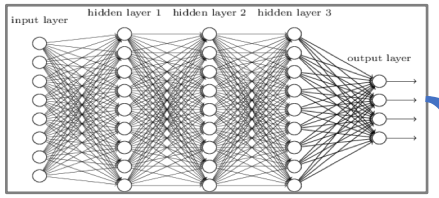


(b) (c) (d)

Figure 1: (a) Motivation. The spatial frequency model for vision [1, 12] shows that natural image can be decomposed into a low and a high spatial frequency part. (b) The output maps of a convolution layer can also be factorized and grouped by their spatial frequency. (c) The proposed multi-frequency feature representation stores the smoothly changing, low-frequency maps in a low-resolution tensor to reduce spatial redundancy. (d) The proposed Octave Convolution operates directly on this representation. It updates the information for each group and further enables information exchange between groups.

Bonus Points for Creativity and Novelty

- Potential Topic 3: New design of learning



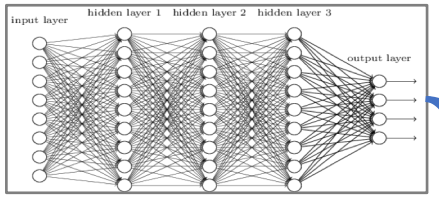
"Learning": the space of parameters → Initialization, Normalization, Pre-training,etc.

"Learning": the space of regularizer → Data augmentation (appearance and geometry transformation), DropOut, Weight Decay, etc

- Understand existing initialization and normalization methods
- Be critical and creative to think about different designs
  - Consider existing network architectures such as ResNets or DenseNets (PyTorch has default implementation), and how to train them with different initialization/normalization methods
  - Examples (understand how they improved the vanilla random initialization and the vanilla BatchNorm methods):
    - https://arxiv.org/abs/1502.01852
    - Fixup: https://arxiv.org/abs/1901.09321v1
    - BatchNorm, InstanceNorm, LayerNorm and GroupNorm: https://arxiv.org/abs/1803.08494 (and related references therein)
    - How does BatchNorm work: https://arxiv.org/abs/1805.11604
    - SwitchableNorm: https://github.com/switchablenorms/Switchable-Normalization
    - Attentive Norm: https://arxiv.org/abs/1908.01259
    - Feature response norm: https://arxiv.org/pdf/1911.09737.pdf
- Validate your ideas on CIFAR10 (and/or CIFAR100)
- If your ideas very promising, I will try to help you test on ImageNet or other big datasets using a large GPU server.
  - If the results are good, we can submit to computer vision/AI conferences.

Bonus Points for Creativity and Novelty

- Potential Topic 3: New design of learning



"Learning": the space of parameters → Initialization, Normalization, Pre-training,etc.

"Learning": the space of regularizer → Data augmentation (appearance and geometry transformation), DropOut, Weight Decay, etc

- Understand existing data augmentation "tricks"
- Be critical and creative to think about different designs
  - Consider existing network architectures such as ResNets or DenseNets (PyTorch has default implementation), and how to train them with different initialization/normalization methods
  - Examples (understand how they augment the training data):
    - Cutout: https://arxiv.org/abs/1708.04552
    - Mixup: https://arxiv.org/abs/1710.09412
    - Bag of tricks: https://arxiv.org/abs/1812.01187 (and references for the tricks therein)
    - Grid Mask: https://arxiv.org/abs/2001.04086
  - Any transformation functions that do not change the task semantics are applicable. **[Be creative!]**
- Validate your ideas on CIFAR10 (and/or CIFAR100)
- If your ideas very promising, I will try to help you test on ImageNet or other big datasets using a large GPU server.
  - If the results are good, we can submit to computer vision/AI conferences.