B657 Final Project – Andrew Corum

More Random Graphs for Neural Architecture Search

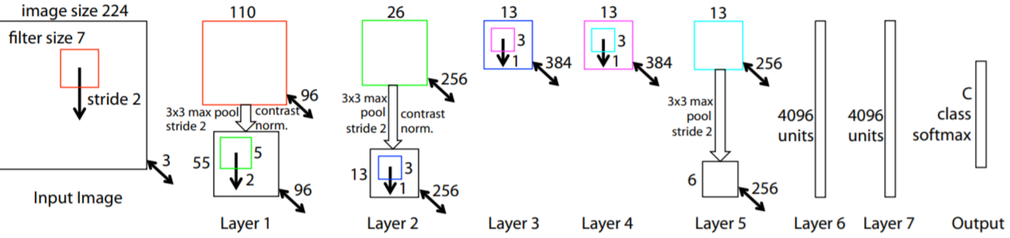
INDIANA UNIVERSITY

SECTION 1

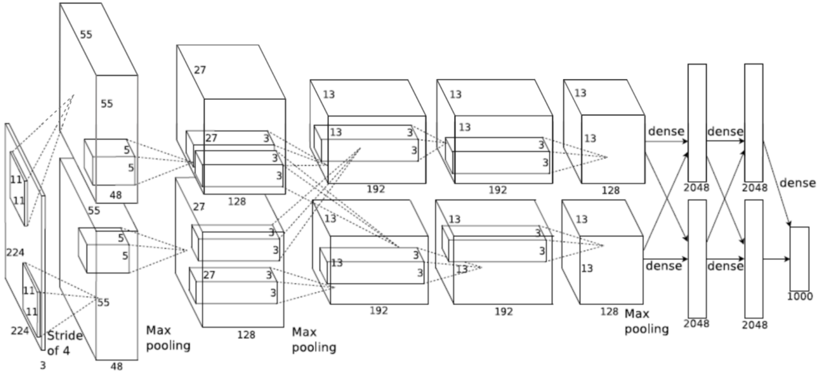
Background

CNNs for Image Classification

1. Images have too many inputs for full-connected NNs
   - Design of CNN architectures often done by hand (ZFNet, AlexNet, ResNet, DenseNet)



ZFNet



AlexNet

Neural Architecture Search

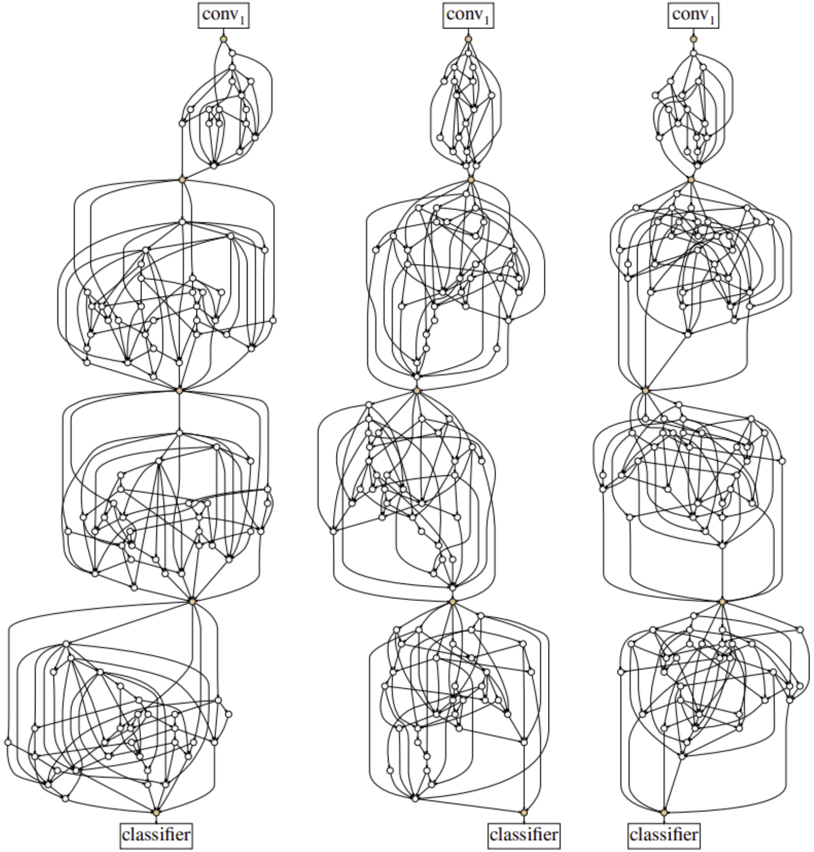
1. Neural architecture search (NAS) can be automated
   - NASNet
   - Create family of wiring patterns to sample from

2. Current NAS search spaces are extremely narrow
   - Still constrain wiring patterns based on hand-made assumptions

RandWire (ICCV 2019)

1. Expand NAS search space by using randomly generated graphs

stage	output	*small regime*	*regular regime*
conv_1	112×112	3×3 conv, $C/2$	
conv_2	56×56	3×3 conv, C	**random wiring** $N/2, C$
conv_3	28×28	**random wiring** N, C	**random wiring** $N, 2C$
conv_4	14×14	**random wiring** $N, 2C$	**random wiring** $N, 4C$
conv_5	7×7	**random wiring** $N, 4C$	**random wiring** $N, 8C$
classifier	1×1	1×1 conv, 1280-d global average pool, 1000-d *fc*, softmax	

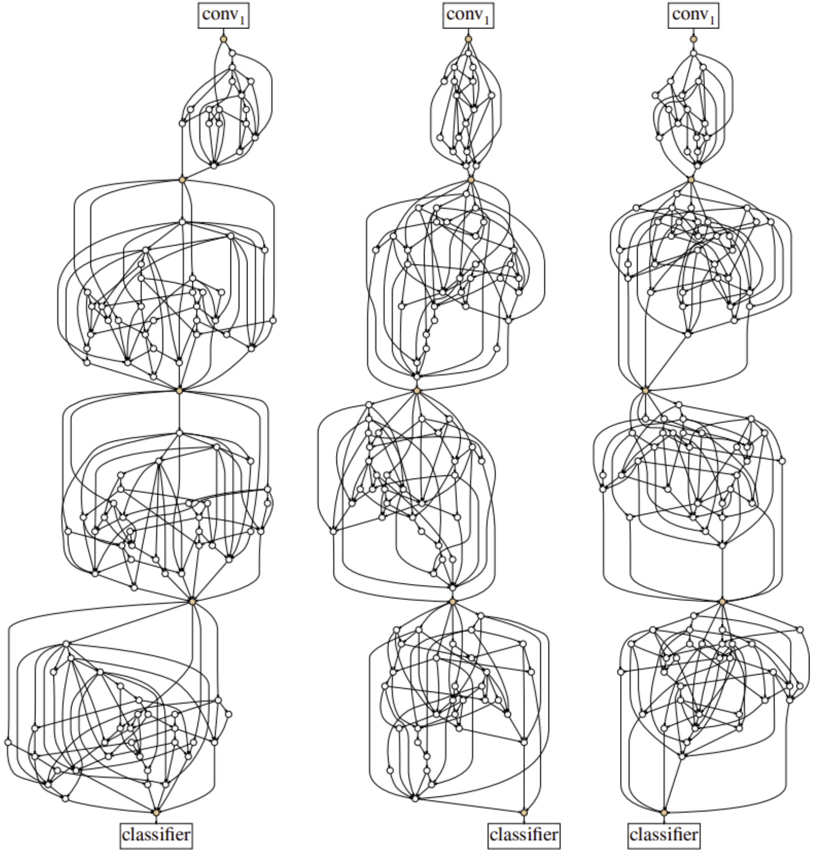


Ref: Xie, Kirillov, Gershick, and He (2019)

RandWire (ICCV 2019)

1. Competitive results, with even top designed CNNs

network	top-1 acc.	top-5 acc.	FLOPs (B)	params (M)
ResNet-50 [11]	77.1	93.5	4.1	25.6
ResNeXt-50 [52]	78.4	94.0	4.2	25.0
RandWire-WS, C=109	**79.0**$_{\pm 0.17}$	**94.4**$_{\pm 0.11}$	4.0$_{\pm 0.09}$	31.9$_{\pm 0.66}$
ResNet-101 [11]	78.8	94.4	7.8	44.6
ResNeXt-101 [52]	79.5	94.6	8.0	44.2
RandWire-WS, C=154	**80.1**$_{\pm 0.19}$	**94.8**$_{\pm 0.18}$	7.9$_{\pm 0.18}$	61.5$_{\pm 1.32}$



SECTION 2

Project Description

RandWire Reconstruction

Results of RandWire are surprising… are they reproduceable?

Project Goals:

- Reconstruct RandWire
  - Found *PyTorch* implementation [ref], with a few mistakes
  - Wanted to fix and re-create in *Tensorflow*
- Evaluate RandWire against other architectures:
  - AlexNet
  - TinyCNN
  - Hand-tuned CNN

SECTION 3

Implementation

Random DAGs

- 3 different DAG generators
  1. Erdos-Renyi (ER)
  2. Barabasi-Albert (BA)
  3. Watts-Strogatz (WS)

- Still not purely random DAGs (inherent structure due to features of each DAG generator)

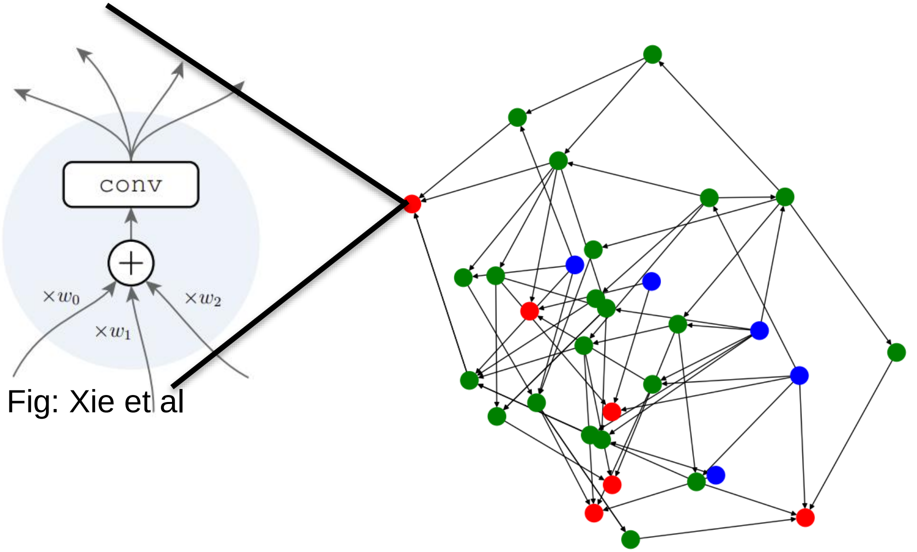
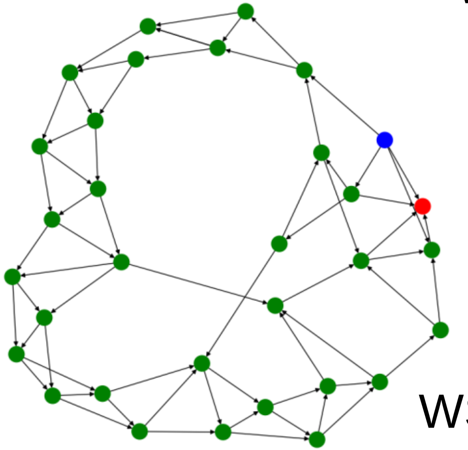


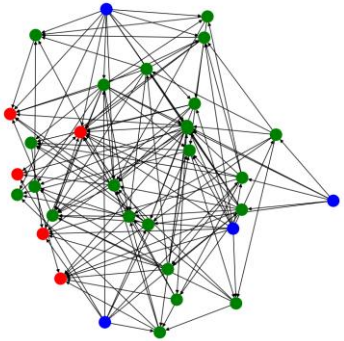
Fig: Xie et al

WS(4, .75)

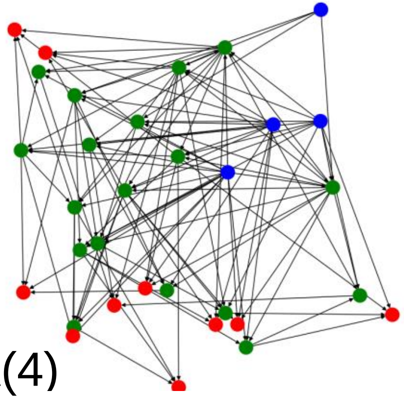


WS(4, .1)

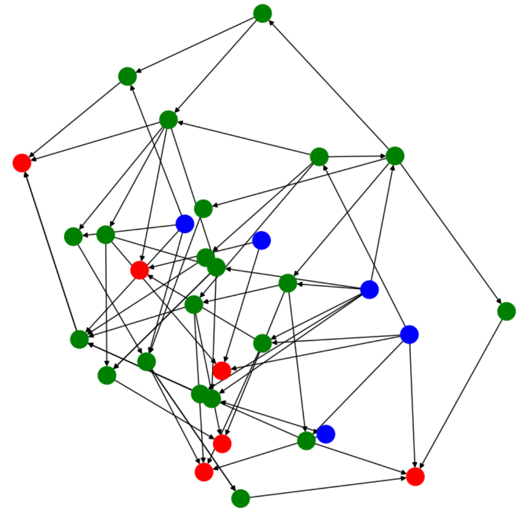
Random DAGs (N=32)



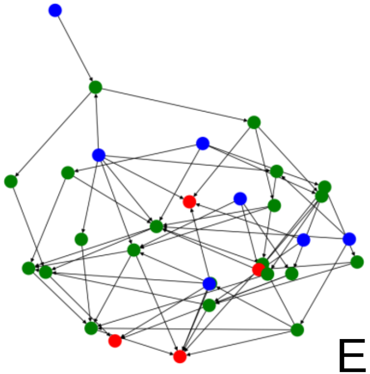
ER(.3)



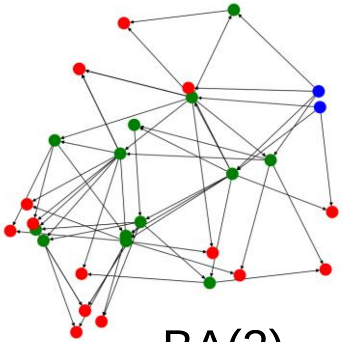
BA(4)



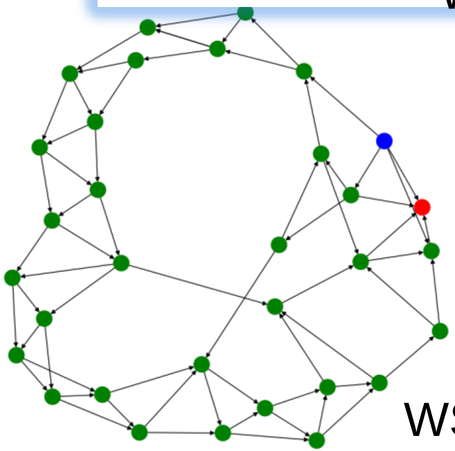
WS(4, .75)



ER(.15)



BA(2)



WS(4, .1)

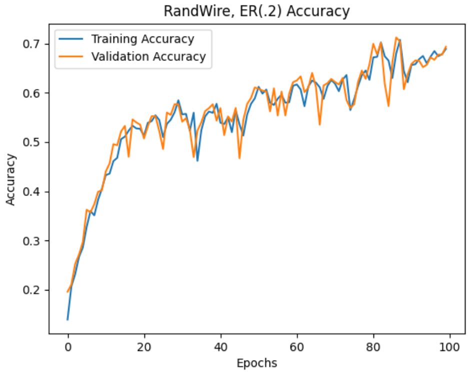
SECTION 4

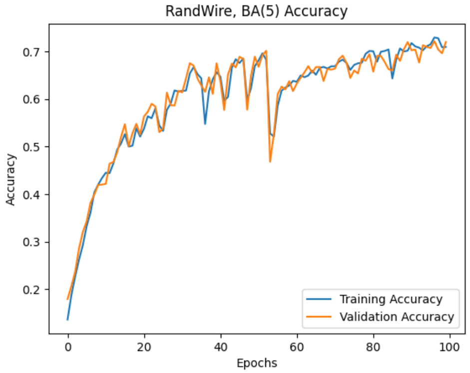
Results

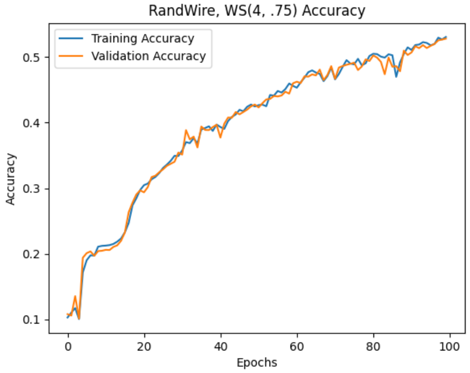
Data

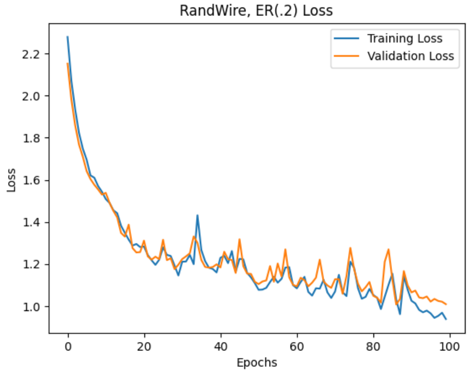
- Used MNIST - contains 70,000 handwritten digits (28x28)
  - Note: RandWire paper used ImageNet (~1.4 million images, 1000 classes, >150GB)
- MNIST is more manageable
- However, then my RandWire construction must differ from the paper’s

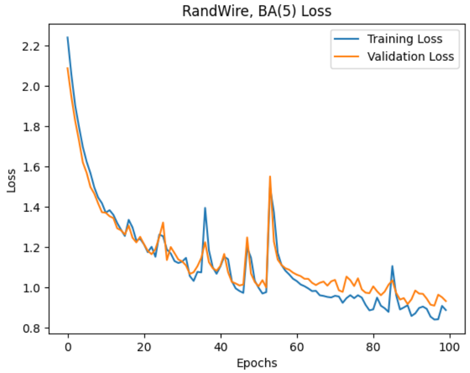
RandWire Learning curves

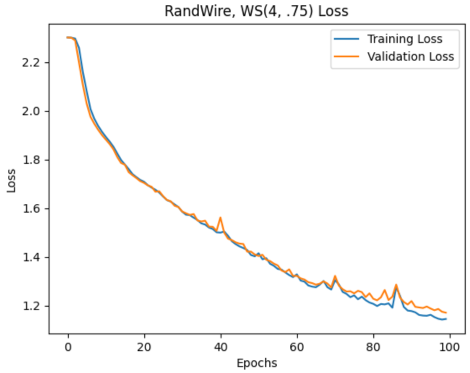




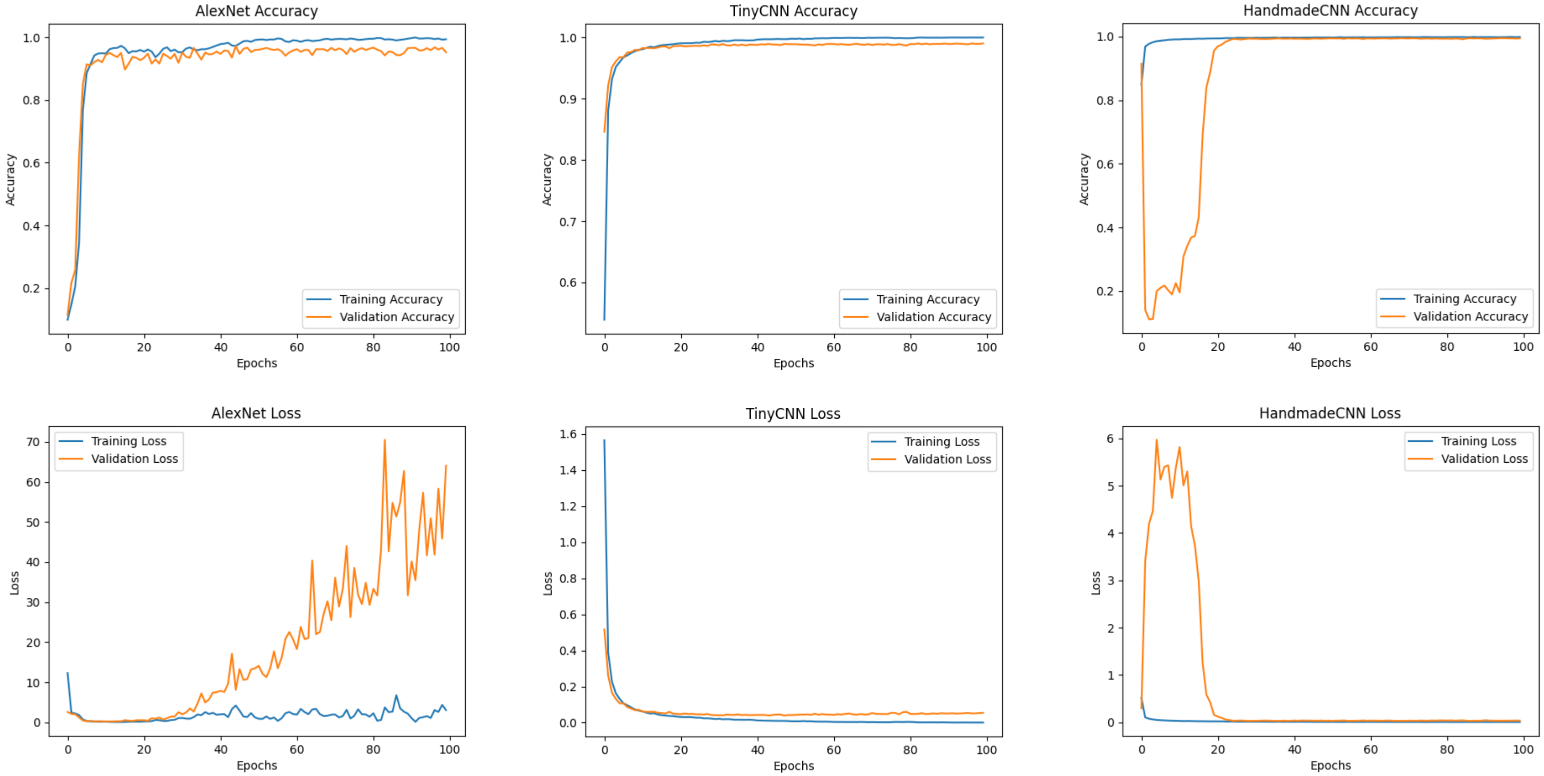








Other CNN Learning Curves



Results on test data

Model	Test Accuracy	Test Loss
RandWire, ER(.2)	69.49%	1.017
RandWire, BA(5)	71.67%	0.9562
RandWire, WS(4, .75)	54.31%	1.146
AlexNet*	96.40%	57.07
TinyCNN	99.17%	0.04160
HandmadeCNN	99.46%	0.02386

SECTION 5

Discussion

Discussion

1. My RandWire did not perform as well as shown in Xie et al. Why?
   - Different dataset (RandWire built to perform on larger images)
     - This issue also seemed to plague implementation of AlexNet
   - Different optimizer for learning
   - Fewer resources. Not able to perform large NAS over many different random DAGs.

2. My RandWire did fit the dataset with much better than random performance

Questions?

Feel free to ask questions in the zoom chat.

Thanks!

Key References: (see final report for remaining references)

Papers:

- S. Xie, A. Kirillov, R. Girshick, and K. He. Exploring randomly wired neural networks forimage recognition. In2019 IEEE/CVF International Conference on Computer Vision (ICCV)
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deepconvolutional neural networks.Commun. ACM
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale imagerecognition. InICLR, 2015
- isualizing and understanding convolutional networks. InEuropean conference on computer vision

Code:

- SeungWonPark. Randwirenn, 2019.https://github.com/seungwonpark/RandWireNN/tree/0850008e9204cef5fcb1fe508d4c99576b37f995

Images:

- ZFNet: https://arxiv.org/pdf/1311.2901.pdf