Auto-Compressing Networks

Vaggelis Dorovatas, Georgios Paraskevopoulos, Alexandros Potamianos

Introduction

In this work, we investigate **the effect of inter-layer connectivity** and **propose a residual variant**, coined as ***Auto-Compressing Networks.***

Importance of Inter-Layer Connectivity

Artificial Neural Networks

FFNs → ResNets

- Multi-path architectures; **short & long connections**
- **Altered information flow & gradient dynamics**
- **Solved vanishing gradients** of FFNs;

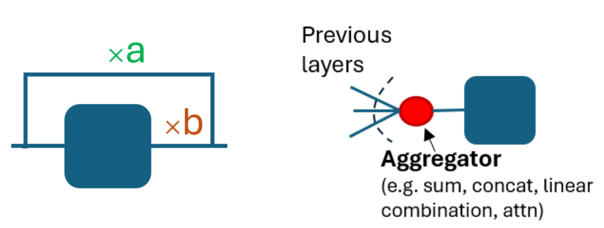
Biological Neural Networks

- **Short & Long connections (Small-world)**
- **Altered BNN connectivity** leads to **distinct cognitive profiles:** *Dyslexia vs Autism*

Residual Architectures

- Highway Networks: $h_i = (1 - C) \cdot x + C \cdot f(h_{i-1})$
- Residual Networks: $h_i = I \cdot x + I \cdot f(h_{i-1})$
- **Residual Variants:**

$h_i = H \cdot x + T \cdot f(h_{i-1})$



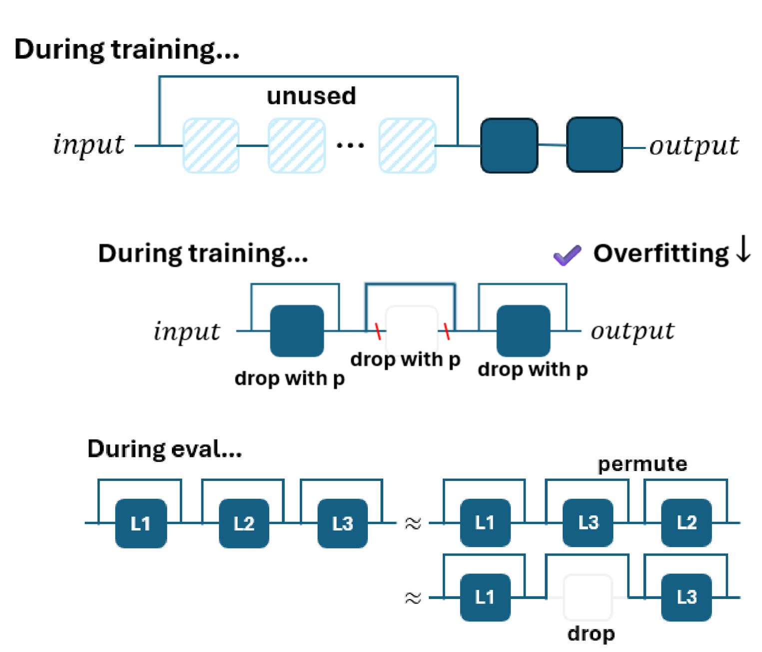
- **Most variants** explore **different aggregation mechanisms** for improving performance, convergence, ...

Effective Depth

Shortcut overuse during training → parts of networks unused

Dropping blocks during training reduces overfitting

Dropping or permuting blocks during eval results in similar performance

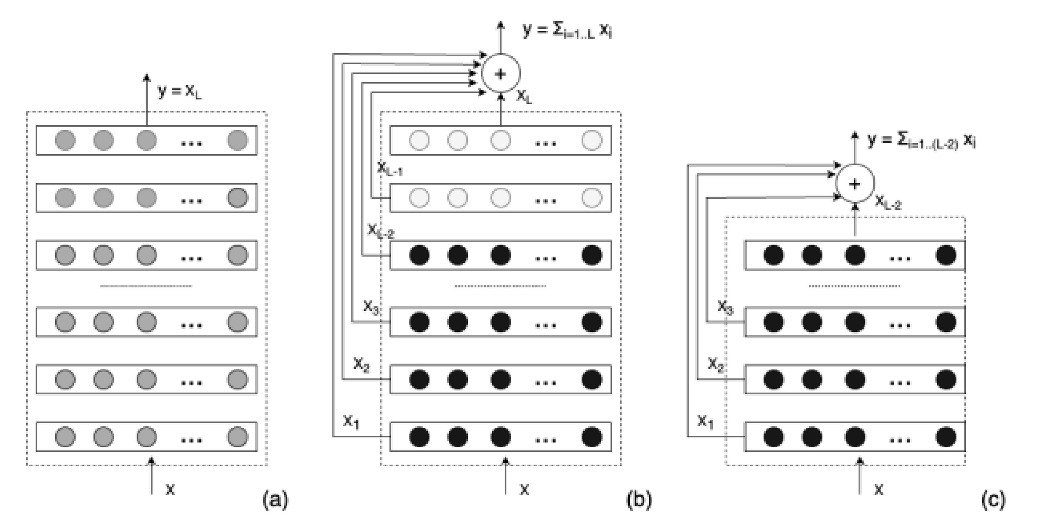


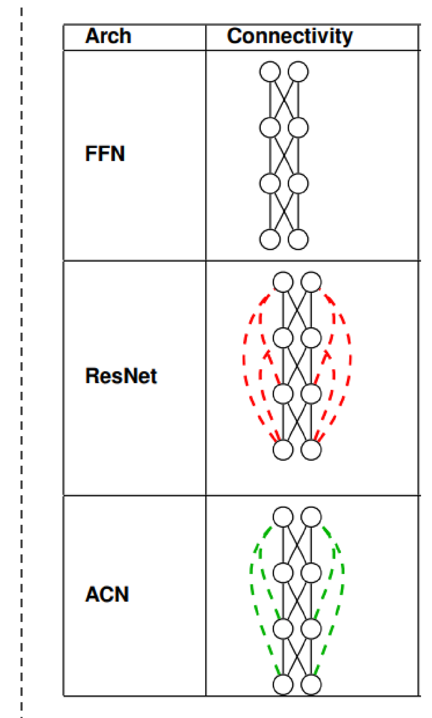
ResNets facilitate efficient training, but they may not use their resources (depth) efficiently.

Auto-Compressing Networks (ACNs)

An *ACN* of depth L:

$$x_i = f_i(x_{i-1}), \ for\ 0 < i < L, \quad x_L = y_A = \sum_{i=0}^{L-1} x_i$$





Decomposition of the Full Gradient

We analyse the gradient of **an intermediate layer i** for **1D linear case.**

- We can decompose it into:

FFN:

$$\frac{\partial y_F}{\partial w_i} = \underbrace{\left(\prod_{k=i+1}^{L} w_k\right)}_{\text{backward term}} \underbrace{\left(\prod_{m=1}^{i-1} w_m\right)}_{\text{forward term}} x_0$$

ResNet:

$$\frac{\partial y_R}{\partial w_i} = \underbrace{\left(\prod_{k=i+1}^{L} (1+w_k)\right)}_{\text{backward term}} \underbrace{\left(\prod_{m=1}^{i-1} (1+w_m)\right)}_{\text{forward term}} x_0$$

ACN:

$$\frac{\partial y_A}{\partial w_i} = \underbrace{\left(1+\sum_{j=i+1}^{L} \prod_{k=i+1}^{j} w_k\right)}_{\text{backward term}} \underbrace{\left(\prod_{m=1}^{i-1} w_m\right)}_{\text{forward term}} x_0$$

Forward Term
- Signal up to layer i
- ACNs equivalent to FFNs

Backward Term
- Signal from the loss
- Multiple paths in ACNs & ResNets
- ↪ The number decreases with depth

Implicit Layer-wise Dynamics of ACNs

ACNs feature an **asymmetric gradient structure**:

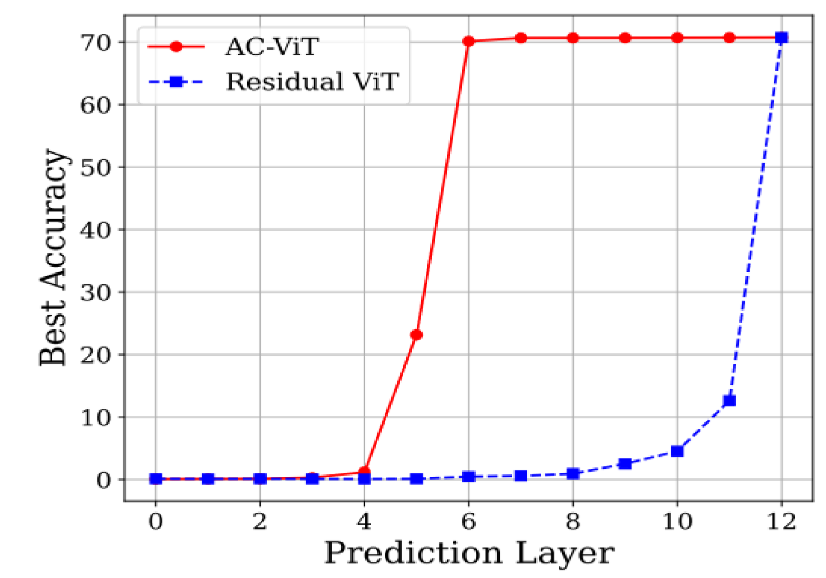
- ***Forward* term**: Identical to FFNs - **single path**.
- ***Backward* term**: Similar to ResNets - **multiple paths**, but linear in depth (vs exponential).
- **Layer-wise Training Dynamics**: Deeper layers receive weaker gradients because of:
  - ➢ A weaker forward component and
  - ➢ Fewer backward paths

Auto-Compression

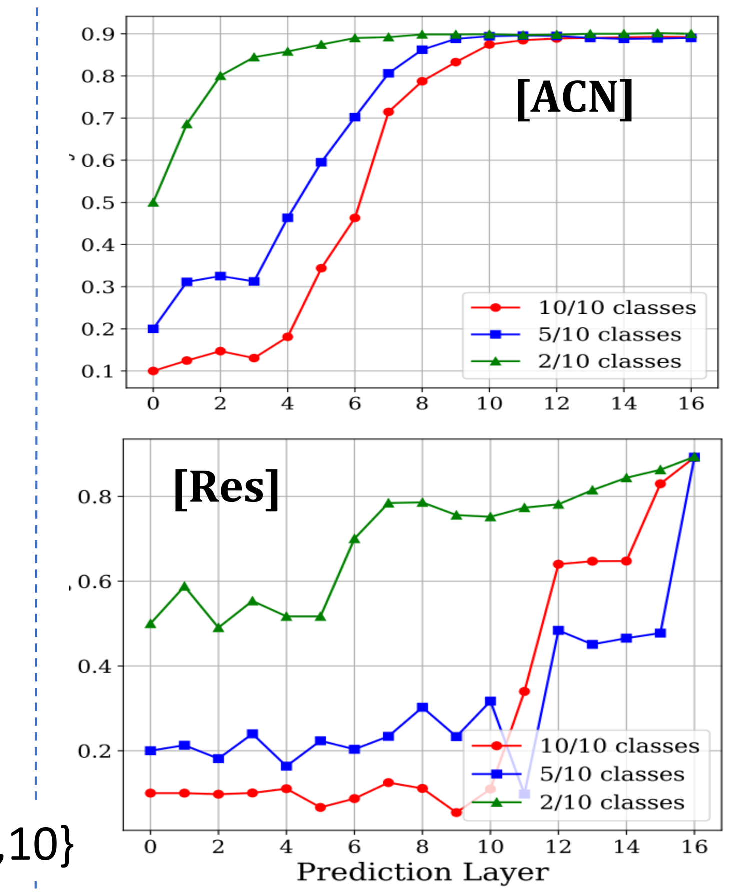
If the k bottom layers, that are trained at a faster rate, suffice to solve the task (minimize the loss), deeper layers remain unused:

⇒ **Implicit information compression**

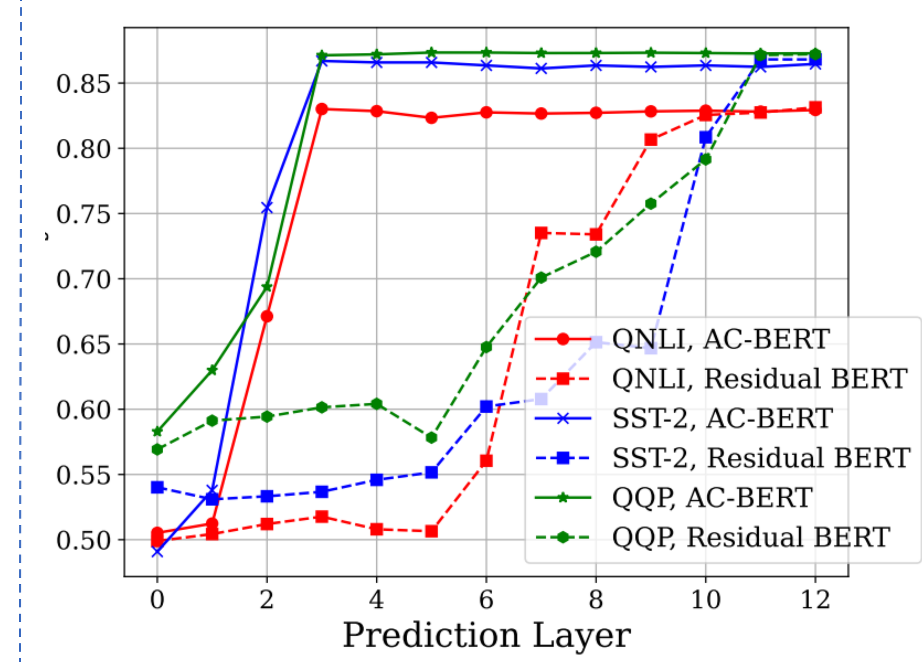
Do ACNs *compress* more?



(a) ViT-base/ImageNet-1K



(b) MLP-Mixer/CIFAR-{2,5,10}



(c) BERT/PT+GLUE

ACNs utilize their depth dynamically across experiments.

Do ACNs *generalize* better?

Robustness against Noise

- Setup
- ↪ ViT/ImageNet-1K

Model	Baseline	Gaussian Noise			Salt and Pepper Noise		
	w/o noise	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$	$p = 0.01$	$p = 0.05$	$p = 0.1$
Residual ViT	70.74	67.68	62.80	45.46	56.80	27.48	10.34
AC-ViT	70.76	69.50	64.54	51.89	59.80	36.35	19.98

Res architectures propagate noise through the residual connections.

Continual Learning

- Setup
- ↪ MLP-Mixer/**Split CIFAR-100**
- Algorithms:
- ↪ **naive fine-tuning[*] (*n*FT)**
- ↪ **Synaptic Intelligence[**] (SI)**

M.	Arch	Avg. Acc. (%) ↑			Avg. Forget. (%) ↓		
		L=5	L=10	L=15	L=5	L=10	L=15
*n*FT	AC-Mixer	32.97±2.4	32.94±5.3	31.61±2.2	46.55±2.2	45.46±5.8	46.91±2.4
	ResMixer	31.77±1.8	28.16±1	26.14±2.3	52.76±2.3	54.89±1.6	54.49±2.2
SI	AC-Mixer	44.5±2.2	46.1±1.3	**46.2±0.8**	35.7±2.1	33.8±0.4	**32±1.8**
	ResMixer	43.47±3.1	36.1±5	32.1±0.8	42.4±4.1	44.6±3.7	50±2.1

ACNs reduce forgetting by up to 18%.

***Deeper* ACNs forget less with SI.**

*directly train on each new task

**penalize changes to previous tasks' important params

Summary

- We proposed *ACNs*, that:
  - **perform on par with residual architectures** but **utilize the network depth dynamically.**
  - Through Auto-Compression, **they learn representations that generalize better.**

Limitations & Future Work

- **Resource Constraints** ↪ scale up experiments
- **Slower Training vs Faster Inference** ↪ research on ACN training optimization ↪ combine architectures