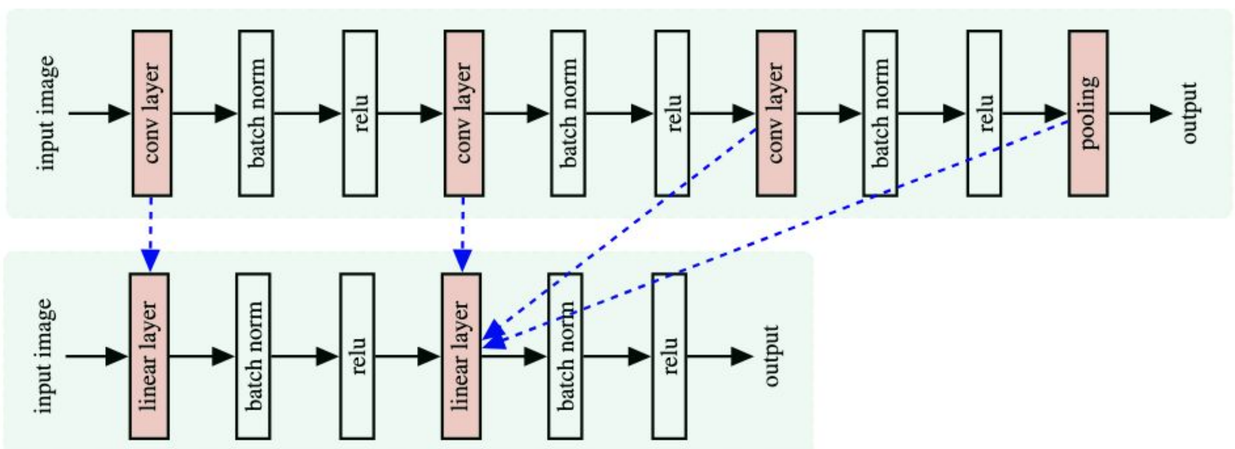
Training the Untrainable: Introducing Inductive Bias via Representational Alignment

Vighnesh Subramaniam, David Mayo, Colin Conwell, Tomaso Poggio, Boris Katz, Brian Cheung, Andrei Barbu
MIT CSAIL; CBMM. https://untrainable-networks.github.io/

Abstract

- What makes networks like ConvNets trainable but networks like fully-connected networks difficult to train for a task like image classification? **Can we make FCNs trainable?**
- Our method does so by transferring the **inductive bias** from one network to another via representational alignment.



Guidance

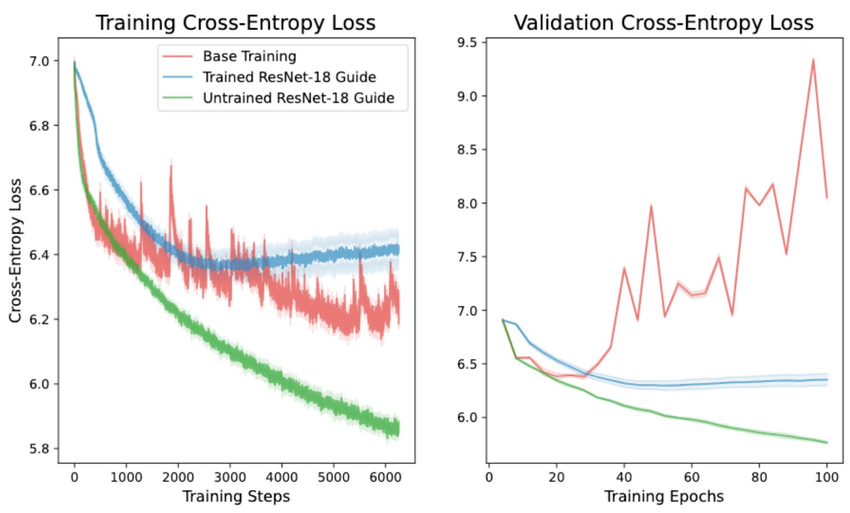
- **Guidance:** increase representational alignment between activations of an untrainable *target* network and activations of a trainable *guide* network during training.
  - Guidance transfers the **inductive bias** from one network to another.
- **Representation Alignment:** Similarity via Centered Kernel Alignment (CKA). Increase CKA at each training step.
- **Architectural vs Training Inductive Biases:** Guide network can be trained, transferring knowledge or randomly initialized, transferring architectural properties.
  - This distinguishes guidance from *distillation!*

Networks and Tasks

- Image Classification: ImageNet
  - Target: Deep FCN, Wide FCN, Deep ConvNet
    Guide: ResNet-18, ResNet-50
- Sequence Modeling: Copy-Paste, Parity, Language Modeling
  - Target: Vanilla RNN; Guide: Transformer

Image Classification

Preventing overfitting in fully-connected networks



- Guidance will prevent overfitting in FCNs and do this with a **randomly-initialized guide network (ResNet-18 in this case)!**

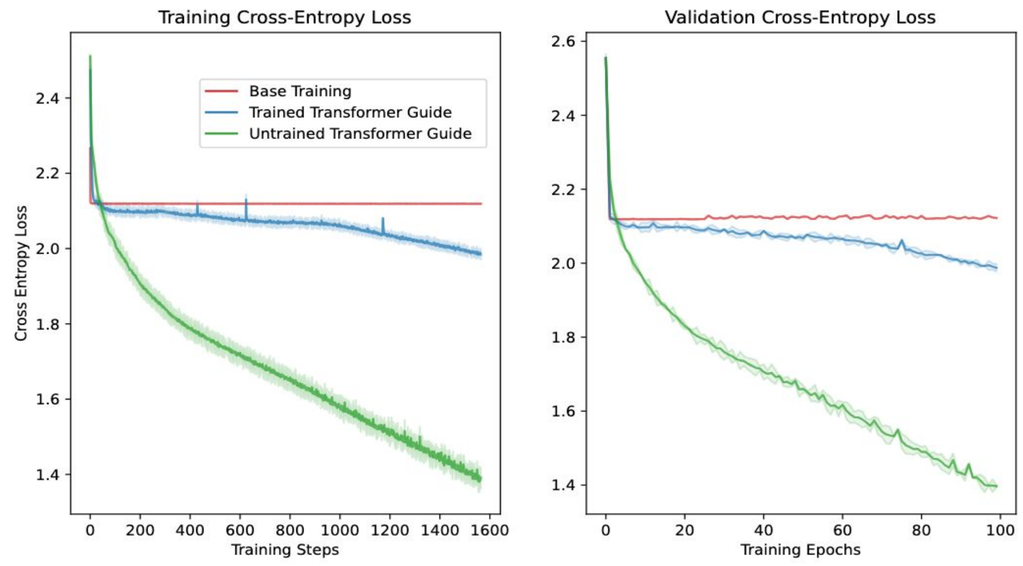
ImageNet Performance Gains

Experiment	ImageNet Top-5 Validation Accuracy (↑)
ResNet-18	89.24
Untrained ResNet-18	0.24 ± 0.043
ResNet-50	92.99
Untrained ResNet-50	0.54 ± 0.029
Deep FCN	1.65 ± 0.51
ResNet-18 → Deep FCN	7.50 ± 1.51
Untrained ResNet-18 → Deep FCN	**13.10** ± 0.72
Wide FCN	34.09 ± 1.21
ResNet-18 → Wide FCN	**43.01** ± 0.92
Untrained ResNet-18 → Wide FCN	39.47 ± 0.31
Deep ConvNet	70.02 ± 1.52
ResNet-50 → Deep ConvNet	**78.91** ± 2.16
Untrained ResNet-50 → Deep ConvNet	68.17 ± 2.54

- Guidance improves image classification performance in traditionally difficult to train networks.
- Underfitting in Deep ConvNets and Wide FCNs is less of a concern!

Sequence Modeling

RNNs incorporate memory!



- When **guided by an untrained transformer, vanilla RNNs do better at copying**, showing stronger incorporation of memory.
- Vanilla RNNs have been abandoned due to memory limitations but we show this may not be necessary!

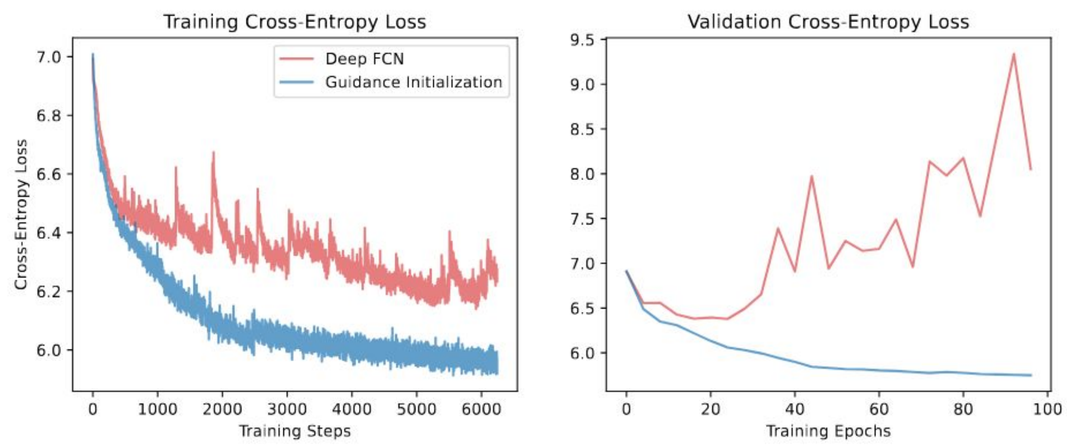
Improving Sequence Modeling

Experiment	Copy-Paste Accuracy (↑)	Parity Accuracy (↑)	Language Modeling Perplexity (↓)
RNN	14.35 ± 0.01	100	69.19 ± 1.89
Untrained RNN	—	2.32 ± 0.41	—
Transformer	96.98	71.98 ± 3.16	34.15
Untrained Transformer	1.04 ± 0.81	—	51948.8 ± 90.44
RNN → Transformer	—	**78.49** ± 2.16	—
Untrained RNN → Transformer	—	70.38 ± 4.17	—
Transformer → RNN	23.27 ± 1.02	—	**40.01** ± 1.54
Untrained Transformer → RNN	**42.56** ± 1.51	—	59.61 ± 2.33

- We can improve both RNNs and Transformers at incorporating memory and sequential state. RNNs teach Transformers and Transformers teach RNNs!
- We make **RNNs competitive with Transformers on language modeling.**
- Transformers struggle with certain sequence tasks like parity and we show that these can picked up by aligning with an RNN!

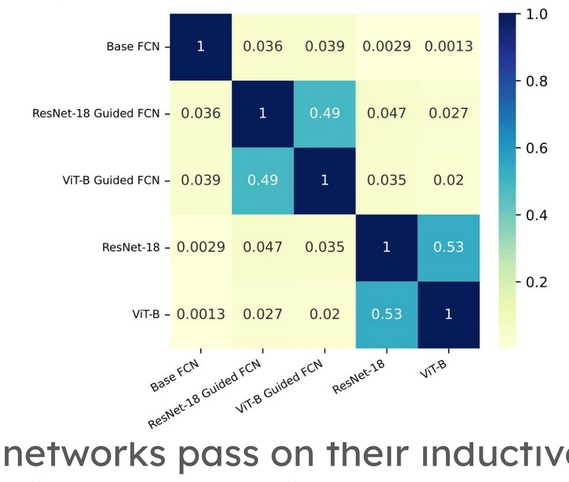
Analyses with Guidance

FCN Initialization



- **Guidance can find new initialization strategies.** We first optimize representational alignment between the FCN and an untrained ResNet on noise for 150 steps. Then optimize on the task. This has no overfitting!

Error Consistency



- Do guide networks pass on their inductive bias to the targets? Using error consistency, we see that they do!

Conclusion

- Guidance provides a method to transfer inductive biases between networks.
- Can we get RNN language models? FCN image classifiers?
- Can we learn what makes a network prevent overfitting? Underfitting? Mathematical properties of neural networks?
- Can we find better ways to compare neural networks?