Unconventional Ways of Training Neural Networks

and What They Teach Us about Model Capacity

Deep Learning: Classics and Trends
5 Feb 2021

Rosanne Liu
http://rosanneliu.com

Conventionally, efforts in deep learning research wish to make neural networks train *easier*.

Conventionally, efforts in deep learning research wish to make neural networks train *easier*.

- Smart architectures
- Optimizers / regularizers
- Initialization schemes
- Normalization methods
- …

However a line of rather **unconventional** work focuses on training neural networks in rather *difficult* ways, most involving not optimizing in the original weight space.

I'd like to cover a few of those work that I was fortunate to have been closely admiring & directly involved in.

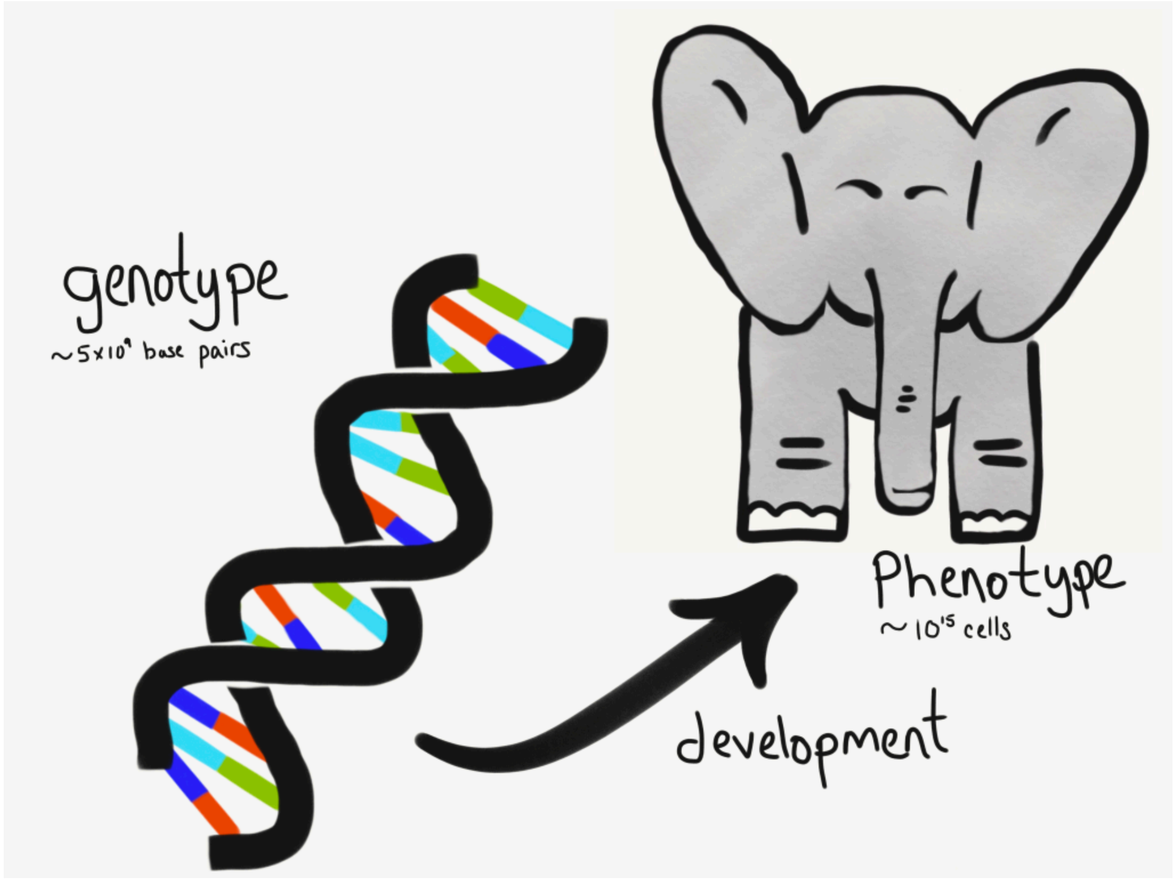
However a line of rather **unconventional** work focuses on training neural networks in rather *difficult* ways, most involving not optimizing in the original weight space.

I'd like to cover a few of those work that I was fortunate to have been closely admiring & directly involved in.

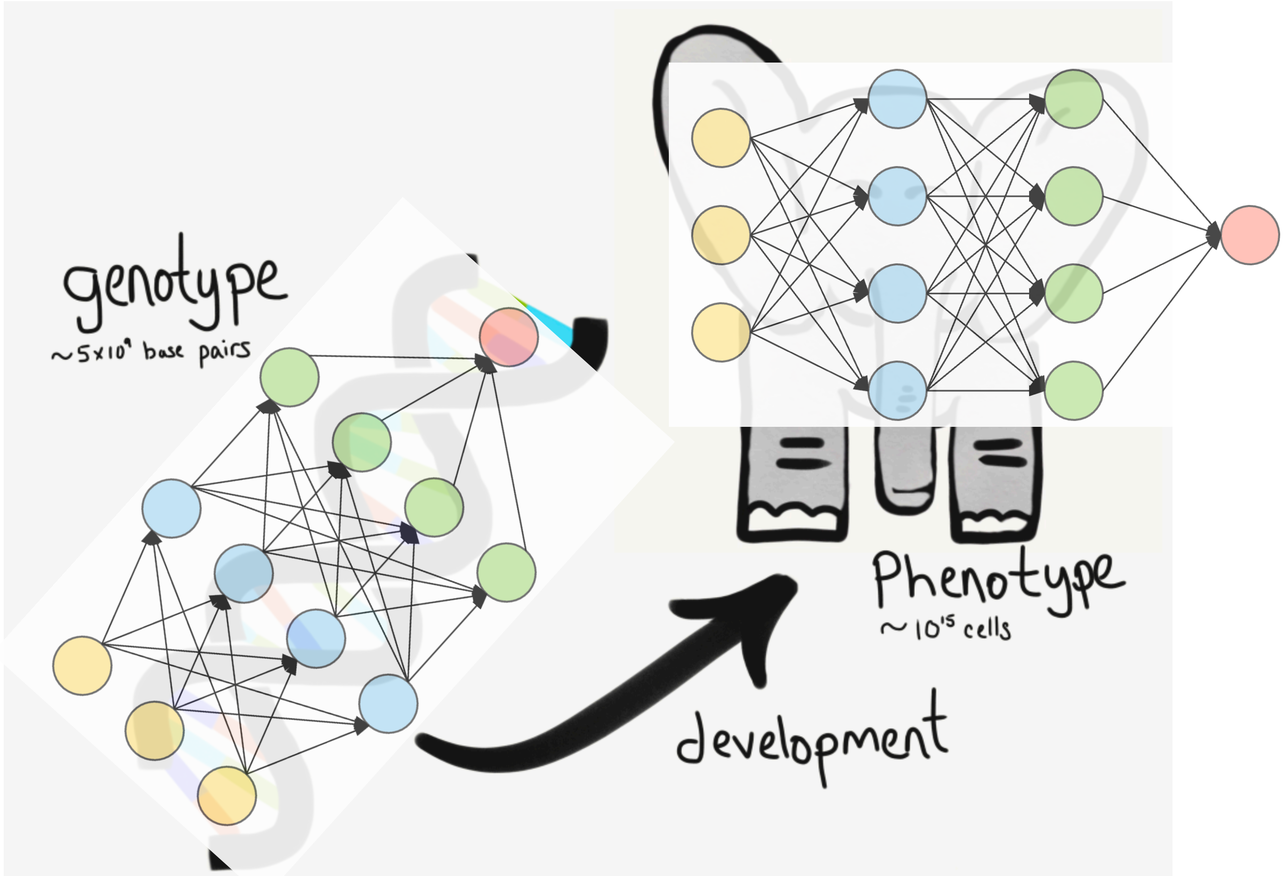
An arbitrary subset

1. Indirect encoding [Stanley et al. 2009; Ha et al. 2017]

1. Indirect encoding [Stanley et al. 2009; Ha et al. 2017]



1. Indirect encoding [Stanley et al. 2009; Ha et al. 2017]



1. Indirect encoding [Stanley et al. 2009; Ha et al. 2017]

A Hypercube-Based Indirect Encoding for Evolving Large-Scale Neural Networks

Accepted to appear in *Artificial Life journal* 15(2), Cambridge, MA: MIT Press, 2009

Kenneth O. Stanley (kstanley@cs.ucf.edu)
David D'Ambrosio (ddambro@cs.ucf.edu)
Jason Gauci (jgauci@cs.ucf.edu)
School of Electrical Engineering and Computer Science
University of Central Florida
4000 Central Florida Blvd.
Orlando, FL 32816-2362 USA

Keywords: Compositional Pattern Producing Networks, CPPNs, HyperNEAT, large-scale artificial neural networks, indirect encoding, generative and developmental systems

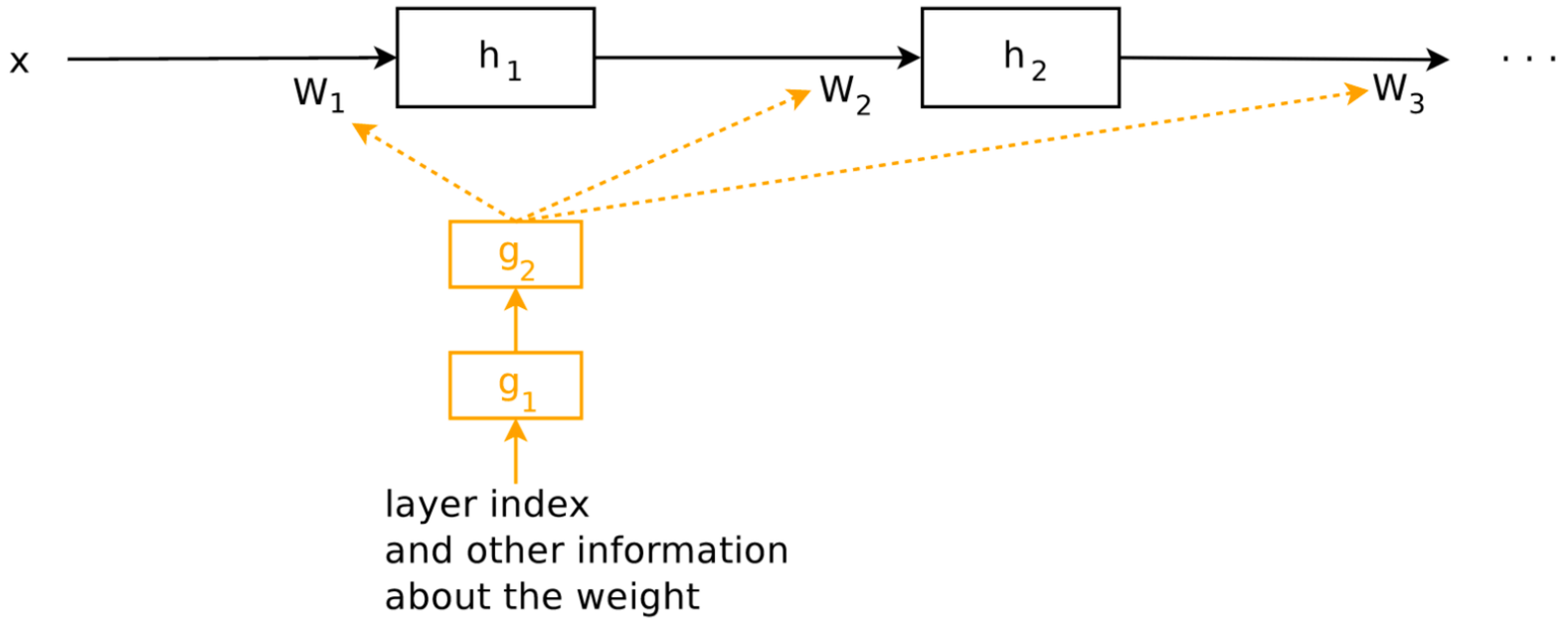
HYPERNETWORKS

David Ha*, Andrew M. Dai, Quoc V. Le
Google Brain
{hadavid,adai,qvl}@google.com

ABSTRACT

This work explores hypernetworks: an approach of using one network, also known as a hypernetwork, to generate the weights for another network. We apply hypernetworks to generate adaptive weights for recurrent networks. In this case, hypernetworks can be viewed as a relaxed form of weight-sharing across layers. In our implementation, hypernetworks are are trained jointly with the main network in an end-to-end fashion. Our main result is that hypernetworks can generate non-shared weights for LSTM and achieve state-of-the-art results on a variety of sequence modelling tasks including character-level language modelling, handwriting generation and neural machine translation, challenging the weight-sharing paradigm for recurrent networks.

A hypernetwork generates the weights for a feedforward network.



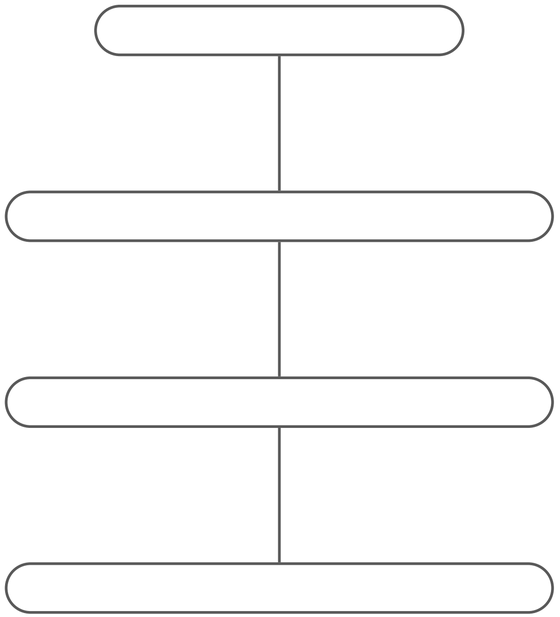
2. Intrinsic dimension [Li et al. 2018]

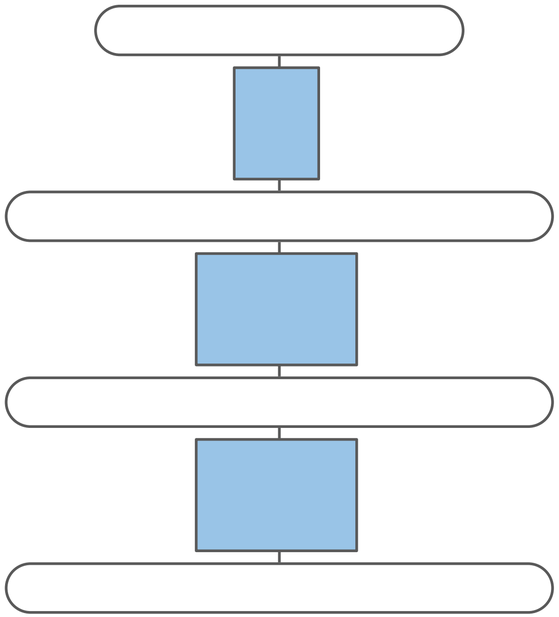
Published as a conference paper at ICLR 2018

MEASURING THE INTRINSIC DIMENSION OF OBJECTIVE LANDSCAPES

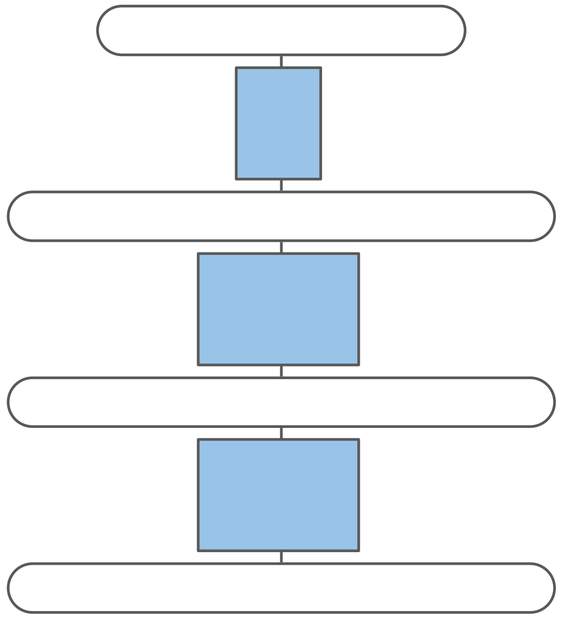
Chunyuan Li *
Duke University
cl319@duke.edu

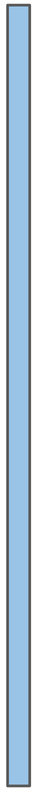
Heerad Farkhoor, Rosanne Liu, and Jason Yosinski
Uber AI Labs
{heerad,rosanne,yosinski}@uber.com



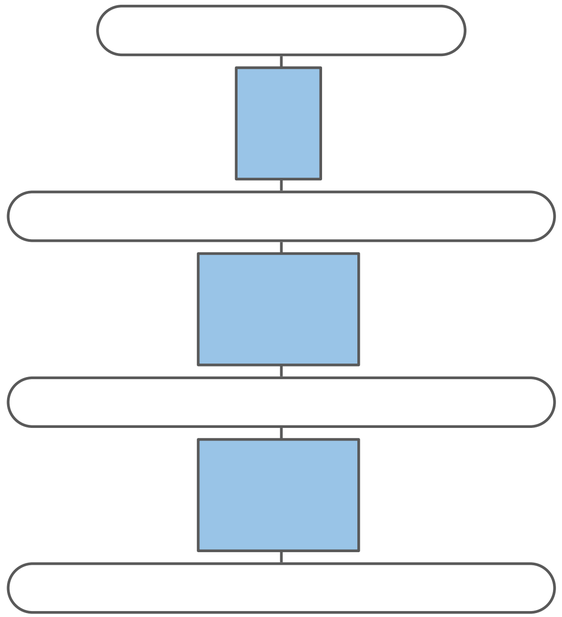


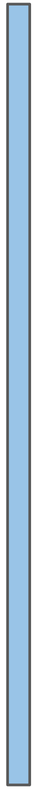
$$\theta \in R^D$$





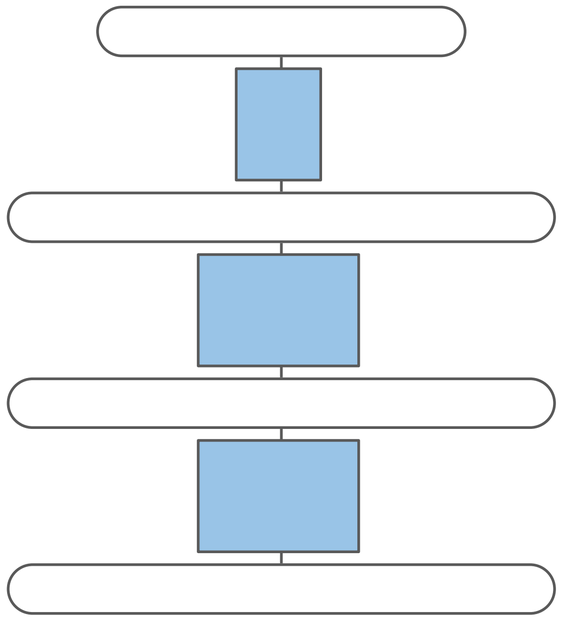
$\theta \in R^D$

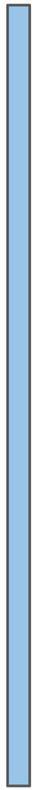




$\theta' \in R^d$

$\theta \in R^D$

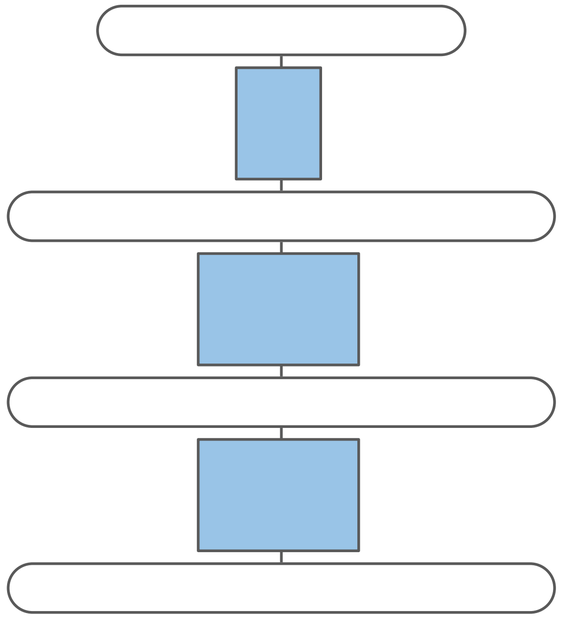


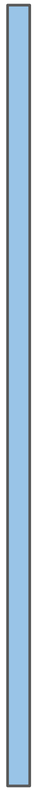


$\theta' \in R^d$

Only train this!

$\theta \in R^D$

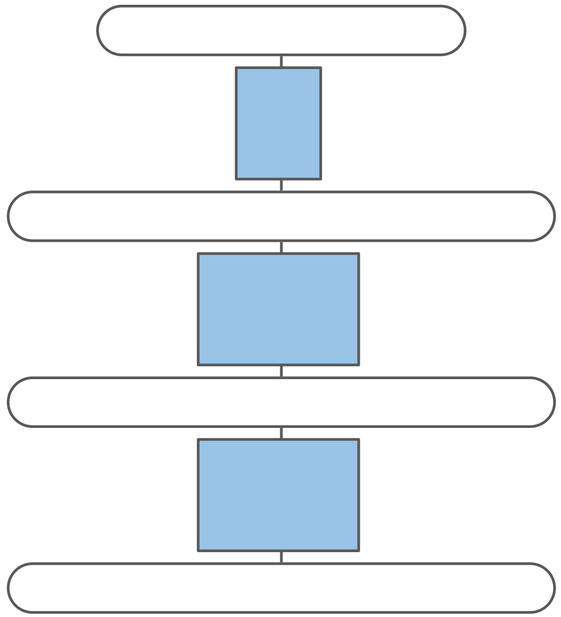


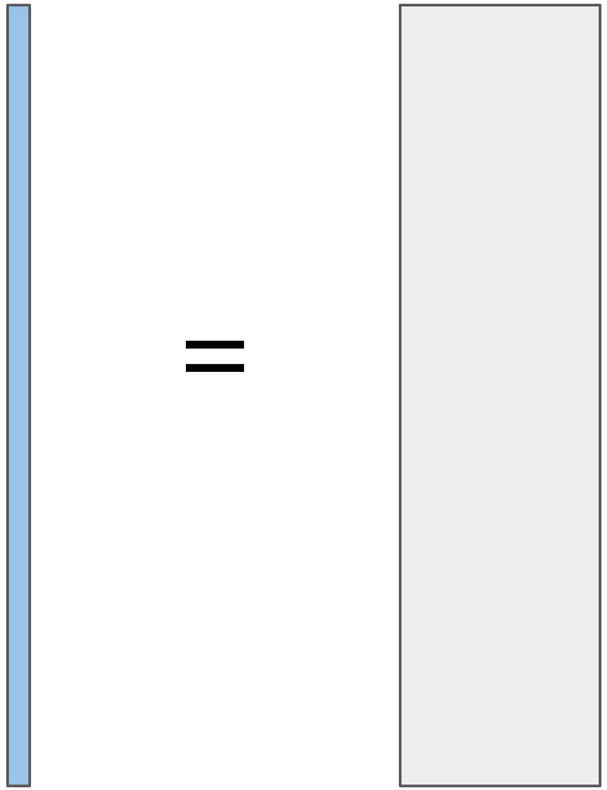


$\Delta\theta' \in R^d$

Only train this!

$$\Delta\theta \in R^D$$



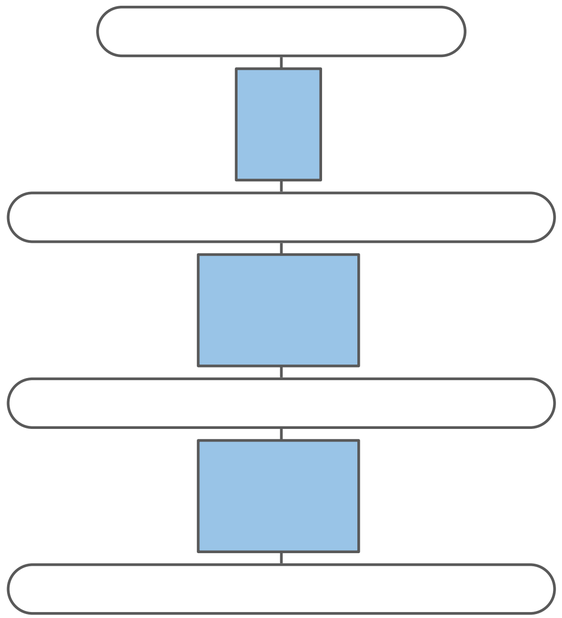


$=$

$*$

$$\Delta\theta' \in R^d$$

Only train this!



$\Delta\theta \in R^D$

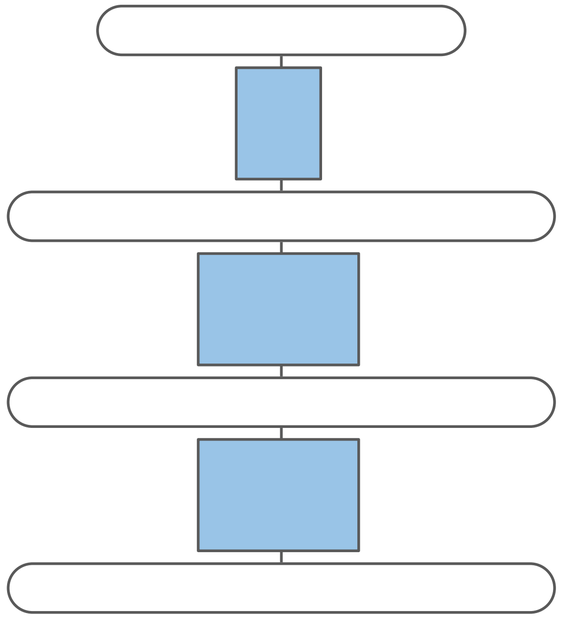
Randomized and frozen

$=$

$*$

$\Delta\theta' \in R^d$

Only train this!



$\Delta\theta \in R^D$

Randomized and frozen

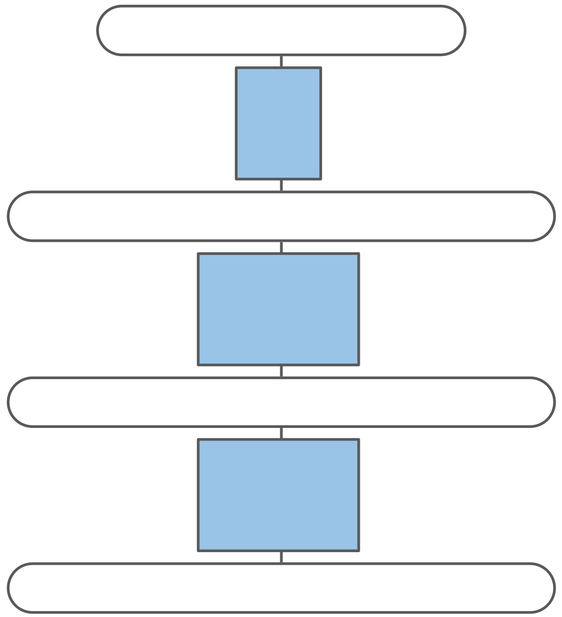
$=$

$*$

$\Delta\theta' \in R^d$

Only train this!

"Affected" but not directly trained



$\Delta\theta \in R^D$

Randomized and frozen

$\Delta\theta' \in R^d$

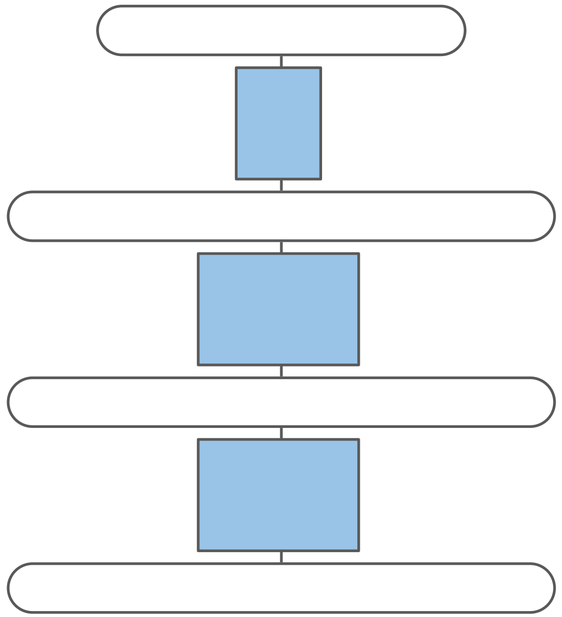
$=$

$*$

Only train this!

"Affected" but not directly trained

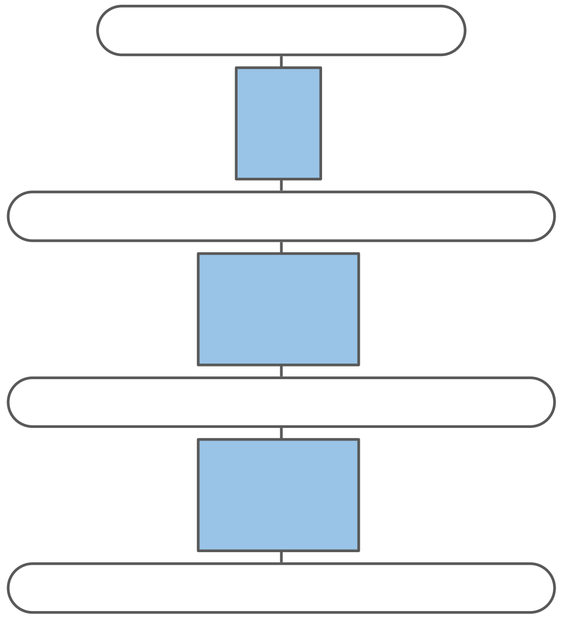
$$\Delta\theta \in R^D$$



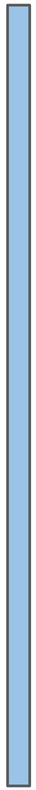
$$=$$

$$\Delta\theta' \in R^d$$

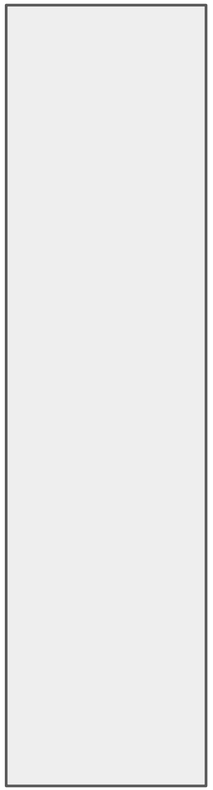
$$*$$



$$\Delta\theta \in R^D$$



=



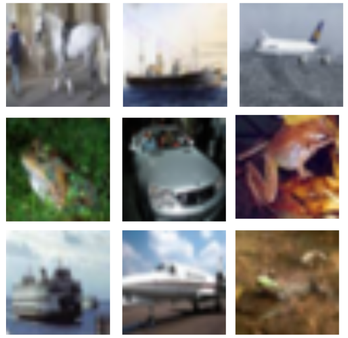
*

How small can this d be?

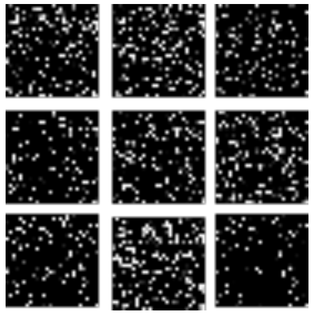
$$\Delta\theta' \in R^d$$

MNIST

CIFAR 10



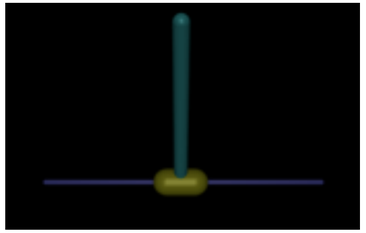
MNIST Shuffled-pixels



Humanoid

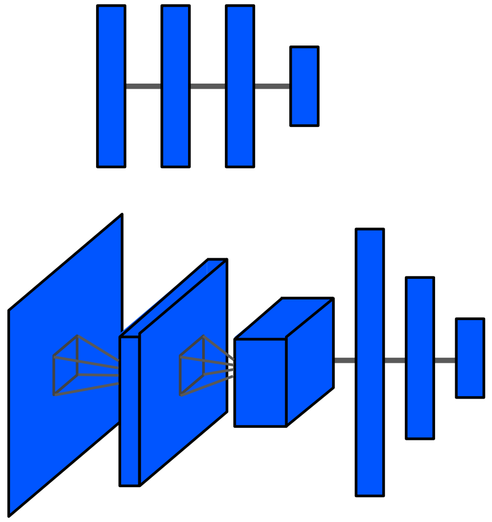
Pong

Inverted Pendulum



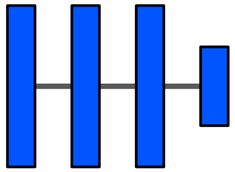
MNIST

MNIST

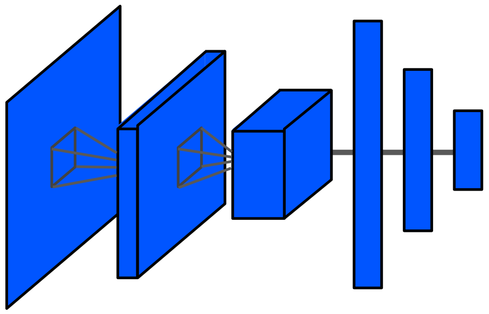


MNIST

Int. Dim.



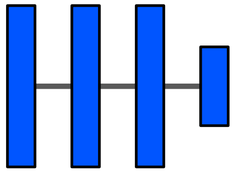
750



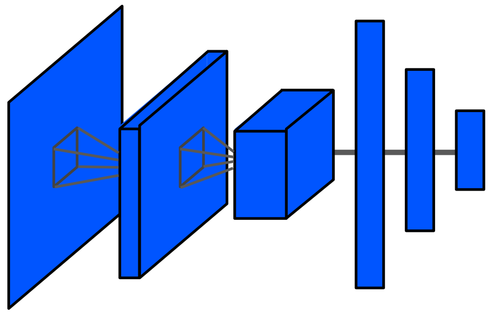
290

MNIST

Int. Dim.

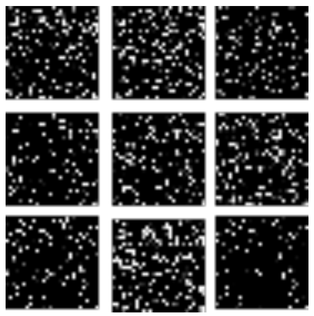


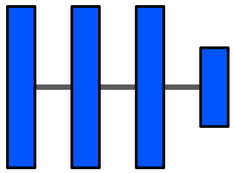
750



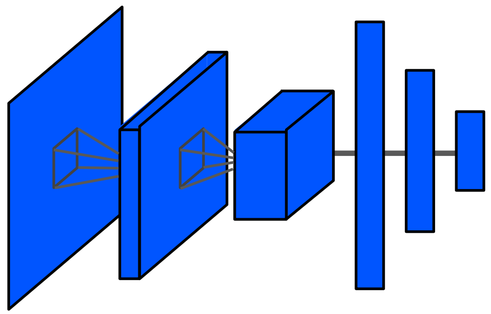
290

MNIST Shuffled-pixels





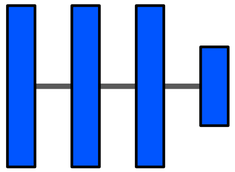
750



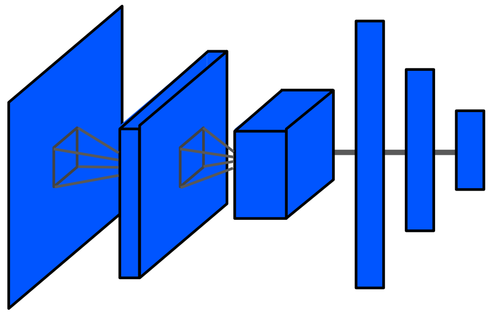
1400

MNIST

Int. Dim.

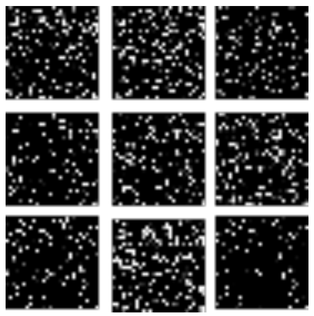


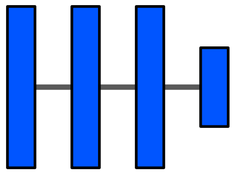
750



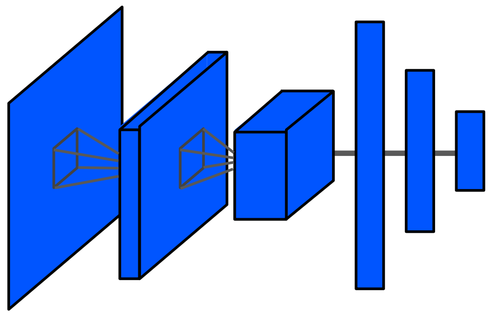
290

MNIST Shuffled-pixels





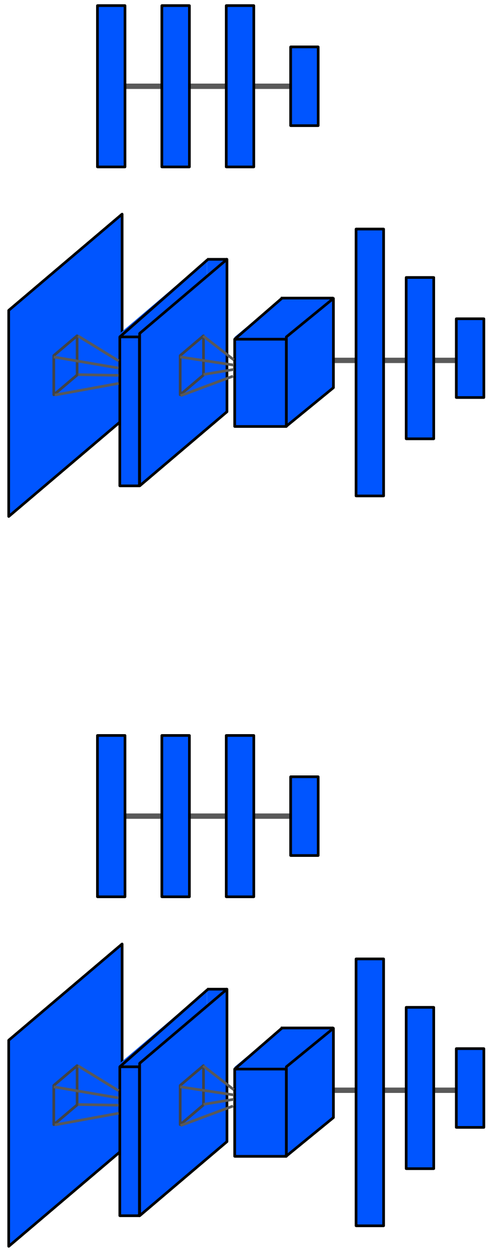
750



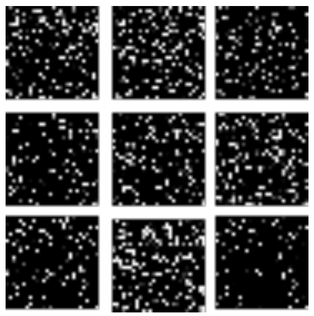
1400

MNIST

Int. Dim.

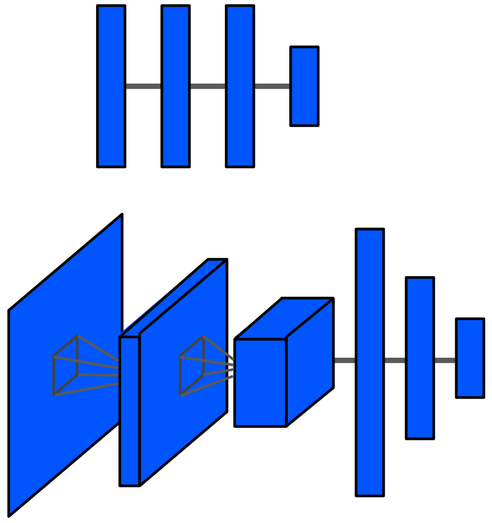


MNIST Shuffled-pixels



MNIST

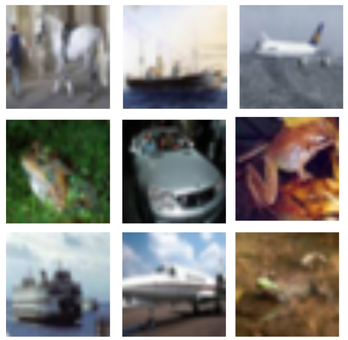
Int. Dim.



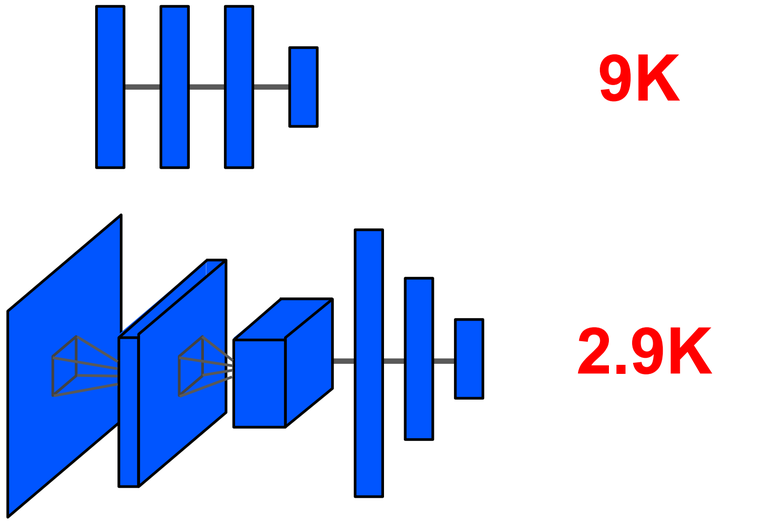
750

290

CIFAR 10



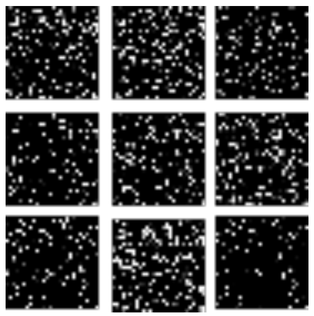
Int. Dim.

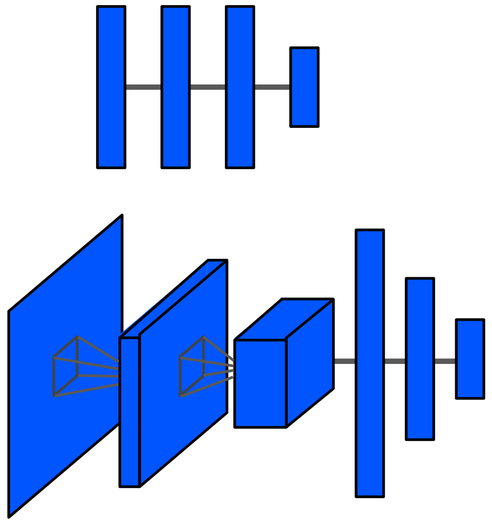


9K

2.9K

MNIST Shuffled-pixels



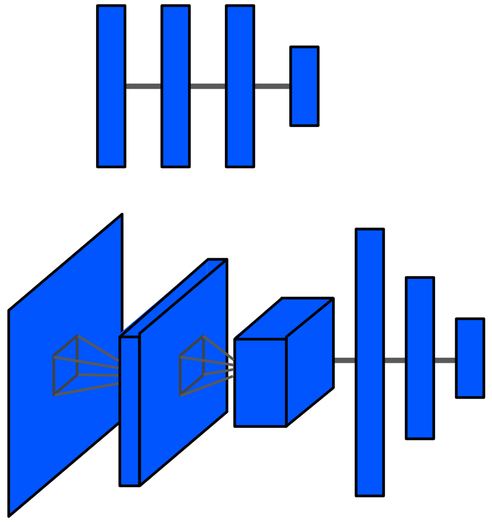


750

1400

MNIST

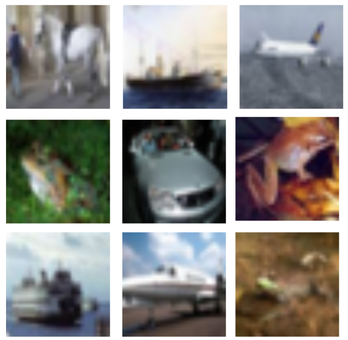
Int. Dim.



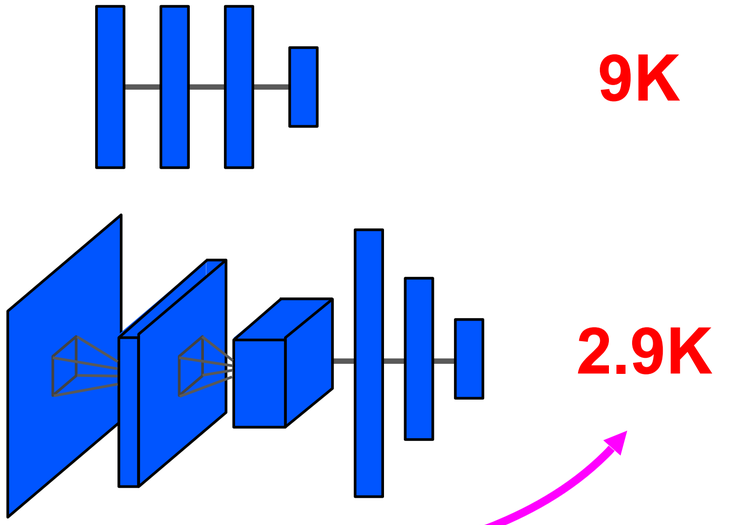
750

290

CIFAR 10



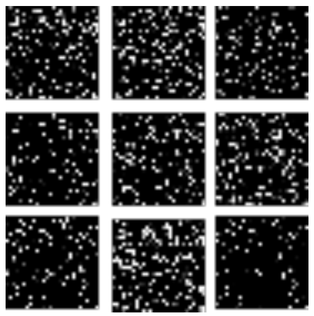
Int. Dim.

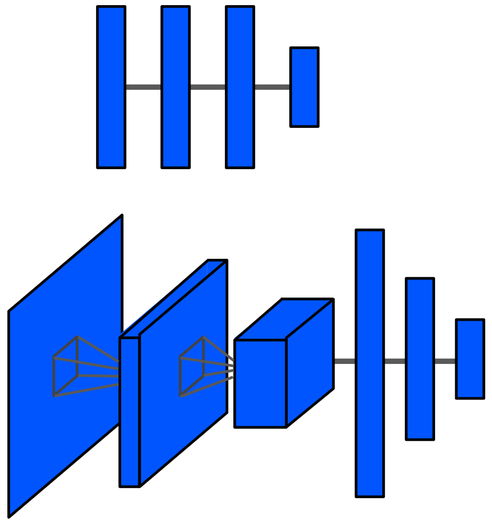


9K

2.9K

MNIST Shuffled-pixels



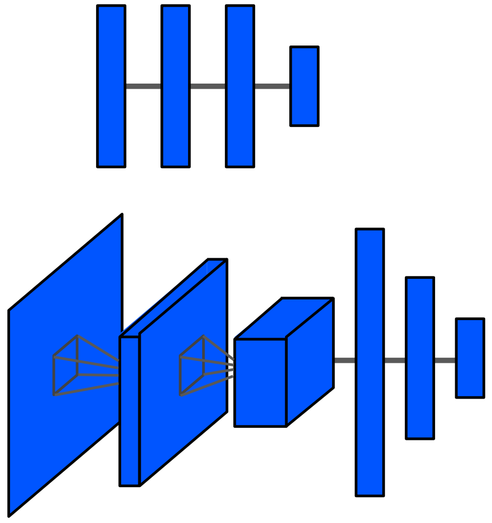


750

1400

MNIST

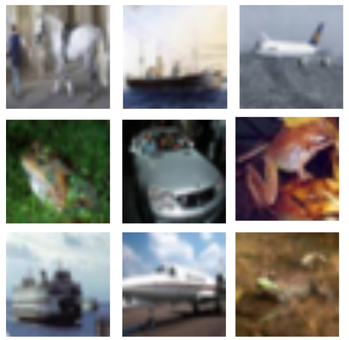
Int. Dim.



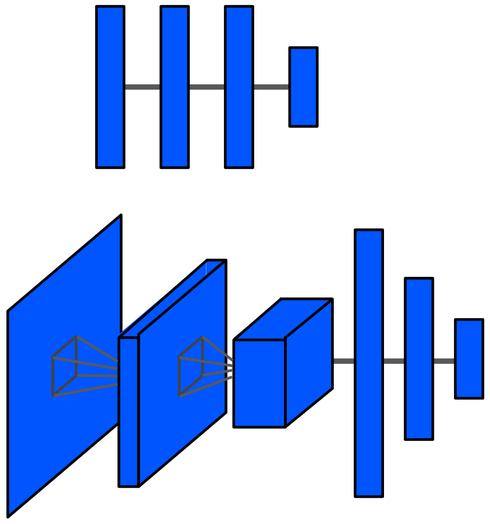
750

290

CIFAR 10



Int. Dim.



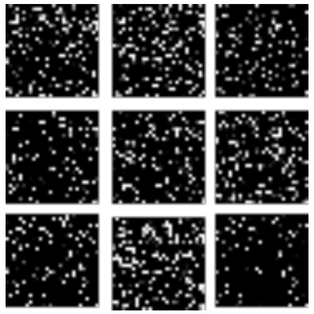
9K

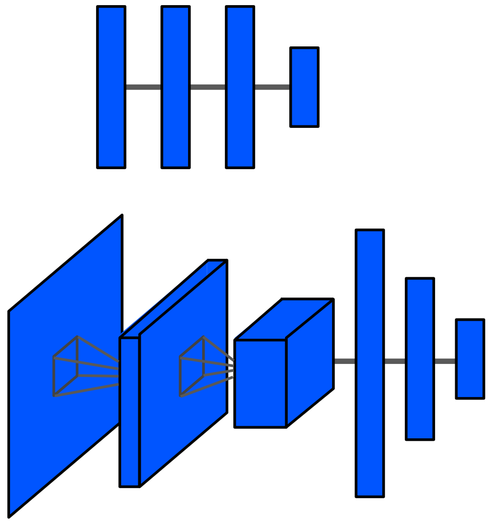
2.9K

SqueezeNet

>500K

MNIST Shuffled-pixels



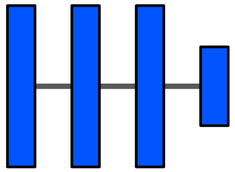


750

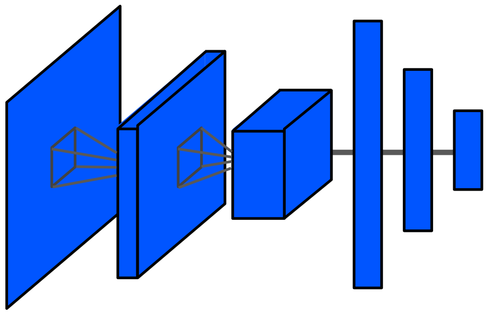
1400

MNIST

Int. Dim.

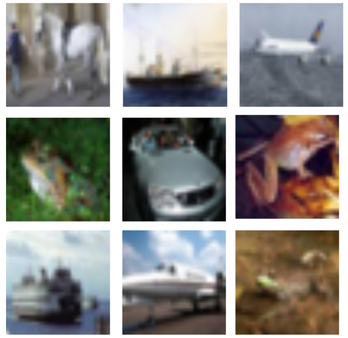


750

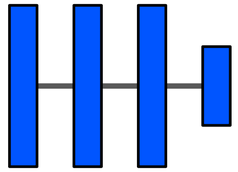


290

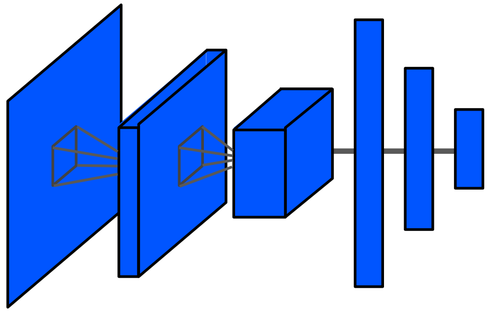
CIFAR 10



Int. Dim.

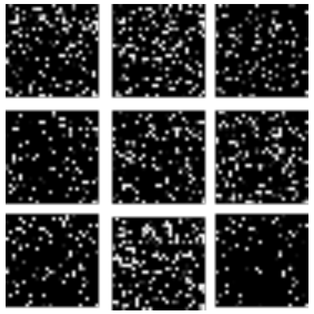


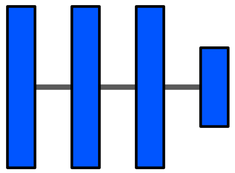
9K



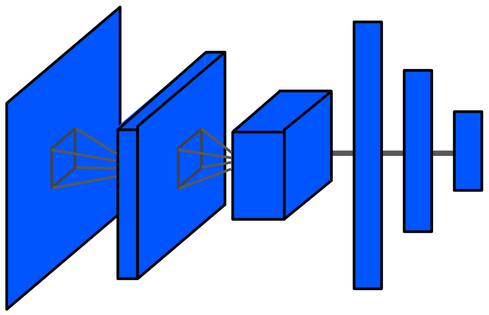
2.9K

MNIST Shuffled-pixels





750



1400

SqueezeNet

>500K

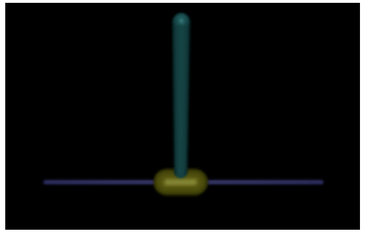
Humanoid

Int. Dim. = 700

Pong

6000

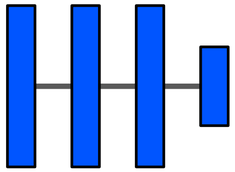
Inverted Pendulum



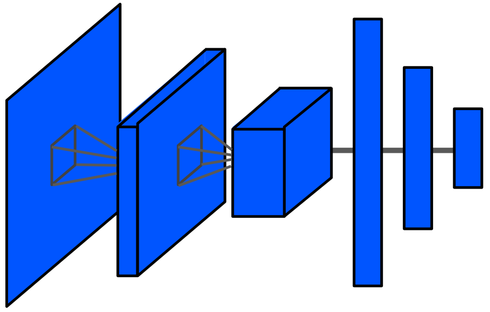
4

MNIST

Int. Dim.

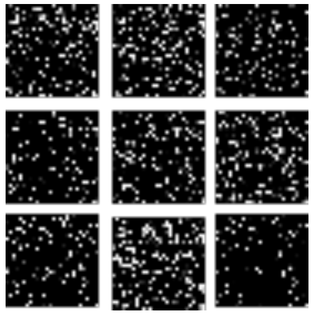


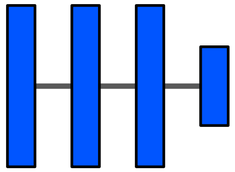
750



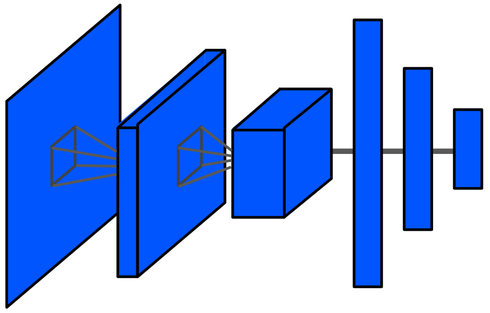
290

MNIST Shuffled-pixels



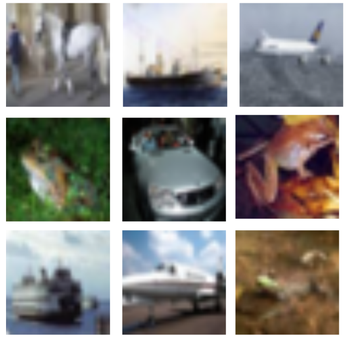


750

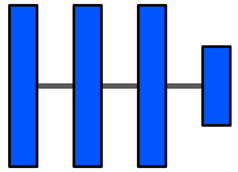


1400

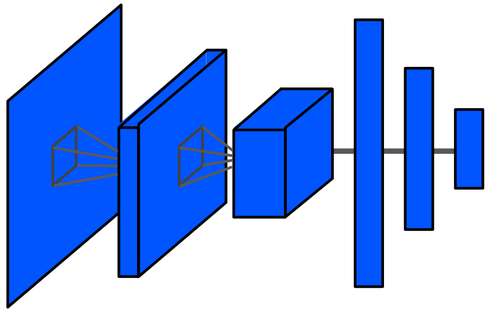
CIFAR 10



Int. Dim.



9K



2.9K

SqueezeNet

>500K

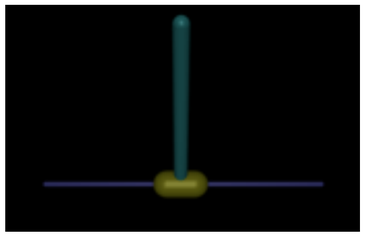
Humanoid

Int. Dim. = 700

Pong

6000

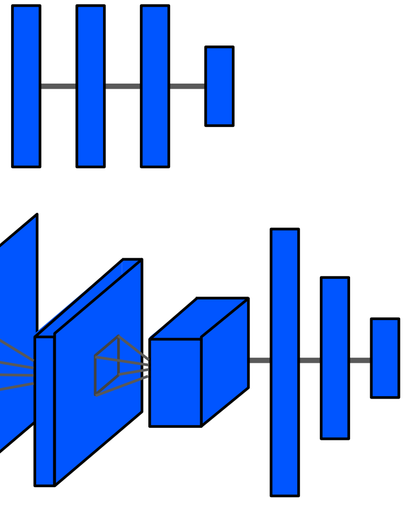
Inverted Pendulum



4

MNIST

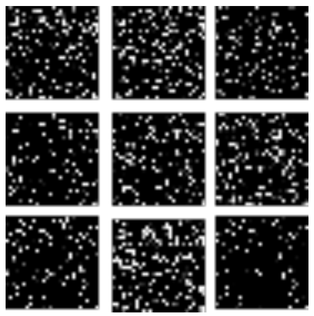
Int. Dim.

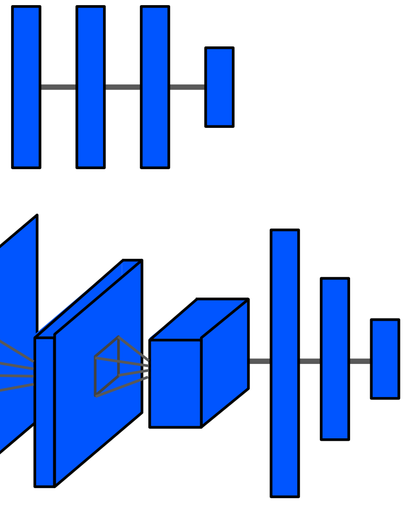


750

290

MNIST Shuffled-pixels

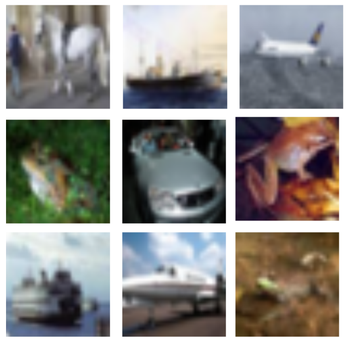




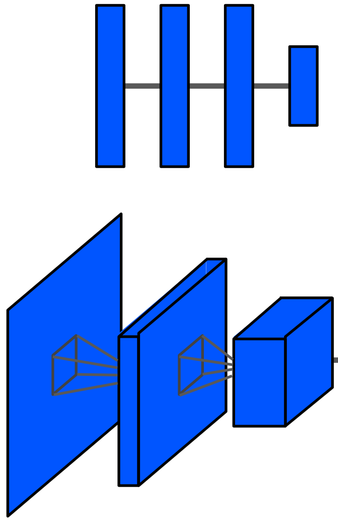
750

1400

CIFAR 10



Int. Dim.



9K

2.9K

SqueezeNet

>500K

Humanoid

Int. Dim. = 700

Pong

6000

Inverted Pendulum

4

Summary of “Intrinsic Dimension”

- We use this weird way of training networks in a random subspace, with a dimension much smaller than that of the original space
- We found the minimum dimension trainable is a fairly stable metric across a family of models for a given dataset
- It says something about model capacity: as we add more parameters, we increase how much the solution set covers the space
- Model compressibility & Minimum Description Length

3. Supermasks [Zhou et al. 2019; Ramanujan et al. 2019; Wortsman et al. 2020]

3. Supermasks [Zhou et al. 2019; Ramanujan et al. 2019; Wortsman et al. 2020]

Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask

Hattie Zhou Uber hattie@uber.com
Janice Lan Uber AI janlan@uber.com
Rosanne Liu Uber AI rosanne@uber.com
Jason Yosinski Uber AI yosinski@uber.com

Abstract

The recent "Lottery Ticket Hypothesis" paper
simple approach to creating sparse networks
in models that are trainable from scratch, but
initial weights. The performance of these netw
of the non-sparse base model, but for reasons th
paper we study the three critical components o
showing that each may be varied significantly w
Ablating these factors leads to new insights fo
as they do. We show why setting weights to z
you need to make the reinitialized network train, and why masking behaves like
training. Finally, we discover the existence of Supermasks, masks that can be
applied to an untrained, randomly initialized network to produce a model with
performance far better than chance (86% on MNIST, 41% on CIFAR-10).

What's Hidden in a Randomly Weighted Neural Network?

Vivek Ramanujan *† Mitchell Wortsman *‡ Aniruddha Kembhavi †‡
Ali Farhadi ‡ Mohammad Rastegari ‡

Supermasks in Superposition

Mitchell Wortsman* University of Washington
Vivek Ramanujan* Allen Institute for AI
Rosanne Liu ML Collective
Aniruddha Kembhavi† Allen Institute for AI
Mohammad Rastegari University of Washington
Jason Yosinski ML Collective
Ali Farhadi University of Washington

Abstract

We present the Supermasks in Superposition (SupSup) model, capable of sequen-
catastrophic forgetting. Our approach
twork and for each task finds a subnet-
ormance. If task identity is given at test
ved with minimal memory usage. If not
g gradient-based optimization to find a
ts which minimizes the output entropy.
step is often sufficient to identify the
also showcase two promising extensions.
ly without task identity information, as
they may detect when they are uncertain about new data and allocate an additional
supermask for the new training distribution. Finally the entire, growing set of
supermasks can be stored in a constant-sized reservoir by implicitly storing them
as attractors in a fixed-sized Hopfield network.

Network Pruning

Table 1: Network pruning can save 9× to 13× parameters with no drop in predictive performance.

Network	Top-1 Error	Top-5 Error	Parameters	Compression Rate
LeNet-300-100 Ref	1.64%	-	267K	
LeNet-300-100 Pruned	1.59%	-	**22K**	**12**×
LeNet-5 Ref	0.80%	-	431K	
LeNet-5 Pruned	0.77%	-	**36K**	**12**×
AlexNet Ref	42.78%	19.73%	61M	
AlexNet Pruned	42.77%	19.67%	**6.7M**	**9**×
VGG-16 Ref	31.50%	11.32%	138M	
VGG-16 Pruned	31.34%	10.88%	**10.3M**	**13**×

[Han et al. 2015]

Lottery Ticket Hypothesis [Frankle & Carbin, 2019]

THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

ABSTRACT

Neural network pruning techniques can reduce the parameter counts of trained networks by over 90%, decreasing storage requirements and improving computational performance of inference without compromising accuracy. However, contemporary experience is that the sparse architectures produced by pruning are difficult to train from the start, which would similarly improve training performance.

We find that a standard pruning technique naturally uncovers subnetworks whose initializations made them capable of training effectively. Based on these results, we articulate the *lottery ticket hypothesis*: dense, randomly-initialized, feed-forward networks contain subnetworks (*winning tickets*) that—when trained in isolation—reach test accuracy comparable to the original network in a similar number of iterations. The winning tickets we find have won the initialization lottery: their connections have initial weights that make training particularly effective.

We present an algorithm to identify winning tickets and a series of experiments that support the lottery ticket hypothesis and the importance of these fortuitous initializations. We consistently find winning tickets that are less than 10-20% of the size of several fully-connected and convolutional feed-forward architectures for MNIST and CIFAR10. Above this size, the winning tickets that we find learn faster than the original network and reach higher test accuracy.

Lottery Ticket Algorithm [Frankle & Carbin, 2019]

1. Initialize a mask m of value ones. Randomly initialize the parameters w of a network $f(x; w \odot m)$.

2. Train the parameters w of the network $f(x; w \odot m)$ to completion. Denote initial weight values as w_i and final weight values as w_f.

3. Mask criterion: Use the mask criterion $M(w_i, w_f) = |w_f|$ to produce a score for each unmasked weight. Set the mask values of weights with the bottom p% of score to 0, and keep the remaining mask values at 1.

4. Mask-1 Action: Rewind weights with mask value 1 back to their initial values and allow them to train in subsequent rounds.

5. Mask-0 Action: Set weights with mask value 0 to 0 and freeze during subsequent rounds.

Deconstructing Lottery Tickets [Zhou et al. 2019]

Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask

Hattie Zhou — Uber — hattie@uber.com

Janice Lan — Uber AI — janlan@uber.com

Rosanne Liu — Uber AI — rosanne@uber.com

Jason Yosinski — Uber AI — yosinski@uber.com

Abstract

The recent "Lottery Ticket Hypothesis" paper by Frankle & Carbin showed that a simple approach to creating sparse networks (keeping the large weights) results in models that are trainable from scratch, but only when starting from the same initial weights. The performance of these networks often exceeds the performance of the non-sparse base model, but for reasons that were not well understood. In this paper we study the three critical components of the Lottery Ticket (LT) algorithm, showing that each may be varied significantly without impacting the overall results. Ablating these factors leads to new insights for why LT networks perform as well as they do. We show why setting weights to zero is important, how signs are all you need to make the reinitialized network train, and why masking behaves like training. Finally, we discover the existence of Supermasks, masks that can be applied to an untrained, randomly initialized network to produce a model with performance far better than chance (86% on MNIST, 41% on CIFAR-10).

Deconstructing Lottery Tickets [Zhou et al. 2019]

1. Initialize a mask m of value ones. Randomly initialize the parameters w of a network $f(x; w \odot m)$.

2. Train the parameters w of the network $f(x; w \odot m)$ to completion. Denote initial weight values as w_i and final weight values as w_f.

3. Mask criterion: Use the mask criterion $M(w_i, w_f) = |w_f|$ to produce a score for each unmasked weight. Set the mask values of weights with the bottom p% of score to 0, and keep the remaining mask values at 1.

4. Mask-1 Action: Rewind weights with mask value 1 back to their initial values and allow them to train in subsequent rounds.

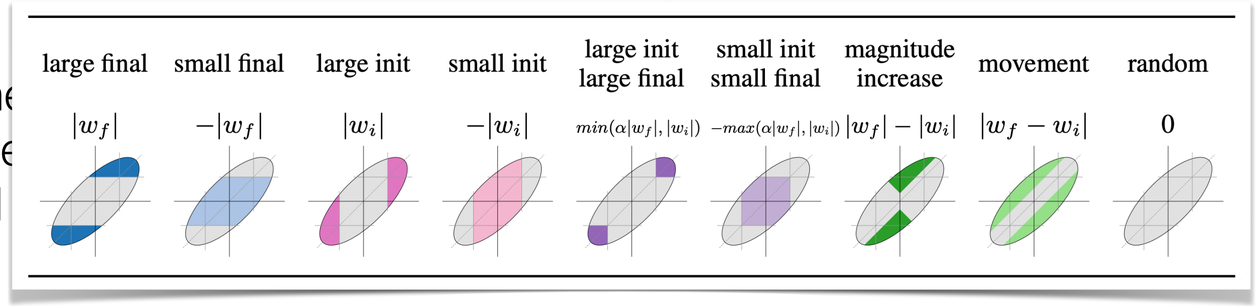
5. Mask-0 Action: Set weights with mask value 0 to 0 and freeze during subsequent rounds.

Deconstructing Lottery Tickets [Zhou et al. 2019]

1. Initialize a mask m of value ones. Randomly initialize the parameters w of a network $f(x; w \odot m)$.

2. Train the parameters w of the network $f(x; w \odot m)$ to completion. Denote initial weight values as w_i and final weight values as w_f.

3. Mask criterion: Use the
for each unmasked we
p% of score to 0, and



4. Mask-1 Action: Rewind weights with mask value 1 back to their initial values and allow them to train in subsequent rounds.

5. Mask-0 Action: Set weights with mask value 0 to 0 and freeze during subsequent rounds.

Deconstructing Lottery Tickets [Zhou et al. 2019]

5 Supermasks

The hypothesis above suggests that for certain mask criteria, like large_final, that masking is training: *the masking operation tends to move weights in the direction they would have moved during training*. If so, just how powerful is this training operation? To answer this, we can start from the beginning—not training the network at all, but simply applying a mask to the randomly initialized network.

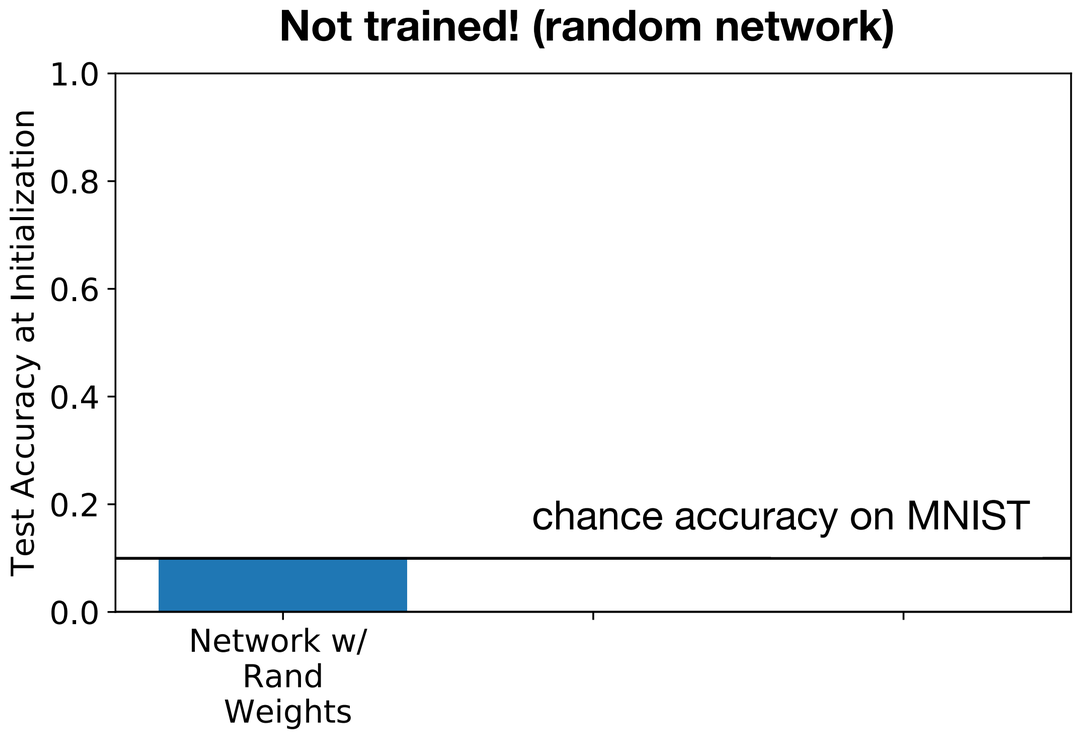
It turns out that with a well-chosen mask, an untrained network can already attain a test accuracy far better than chance. This might come as a surprise, because if you use a randomly initialized and untrained network to, say, classify images of handwritten digits from the MNIST dataset, you would expect accuracy to be no better than chance (about 10%). But now imagine you multiply the network weights by a mask containing only zeros and ones. In this instance, weights are either unchanged or deleted entirely, but the resulting network now achieves nearly 40 percent accuracy at the task! This is strange, but it is exactly what we observe with masks created using the large_final criterion.

In randomly-initialized networks with large_final masks, it is not implausible to have better-than-chance performance since the masks are derived from the training process. The large improvement in performance is still surprising, however, since the only transmission of information from the training back to the initial network is via a zero-one mask based on a simple criterion. We call masks that can produce better-than-chance accuracy without training of the underlying weights "Supermasks".

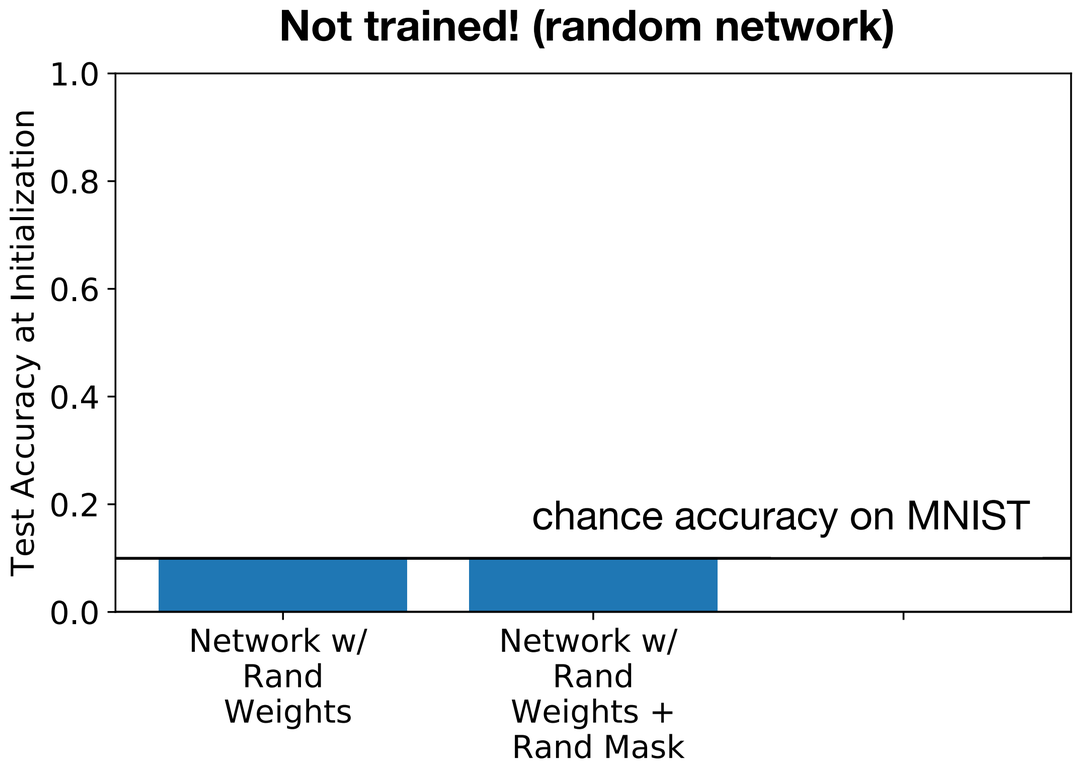
We now turn our attention to finding better Supermasks. First, we simply gather all masks instantiated in the process of creating the networks shown in Figure 2, apply them to the original, randomly initialized networks, and evaluate the accuracy without training the network. Next, compelled by the demonstration in Section 3 of the importance of signs and in Section 4 of keeping large

[4] Additional control variants of this experiment can be seen in Supplementary Information Section S3.

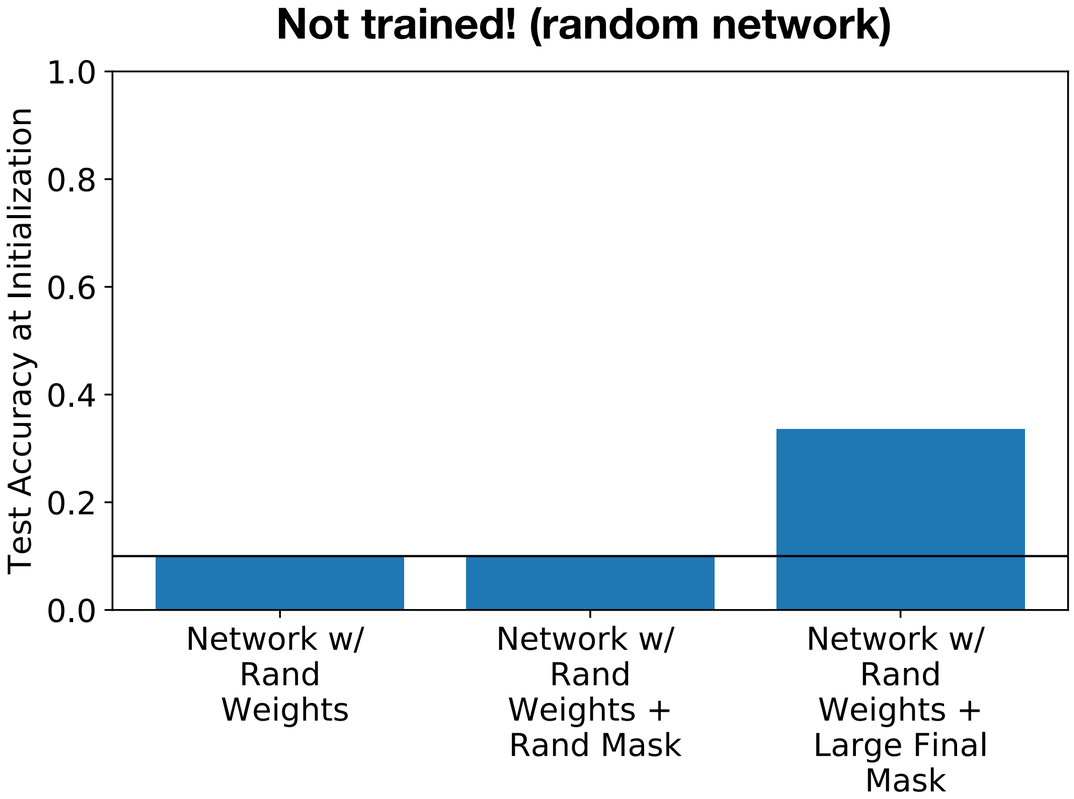
Curiously Effective Masks at Initialization



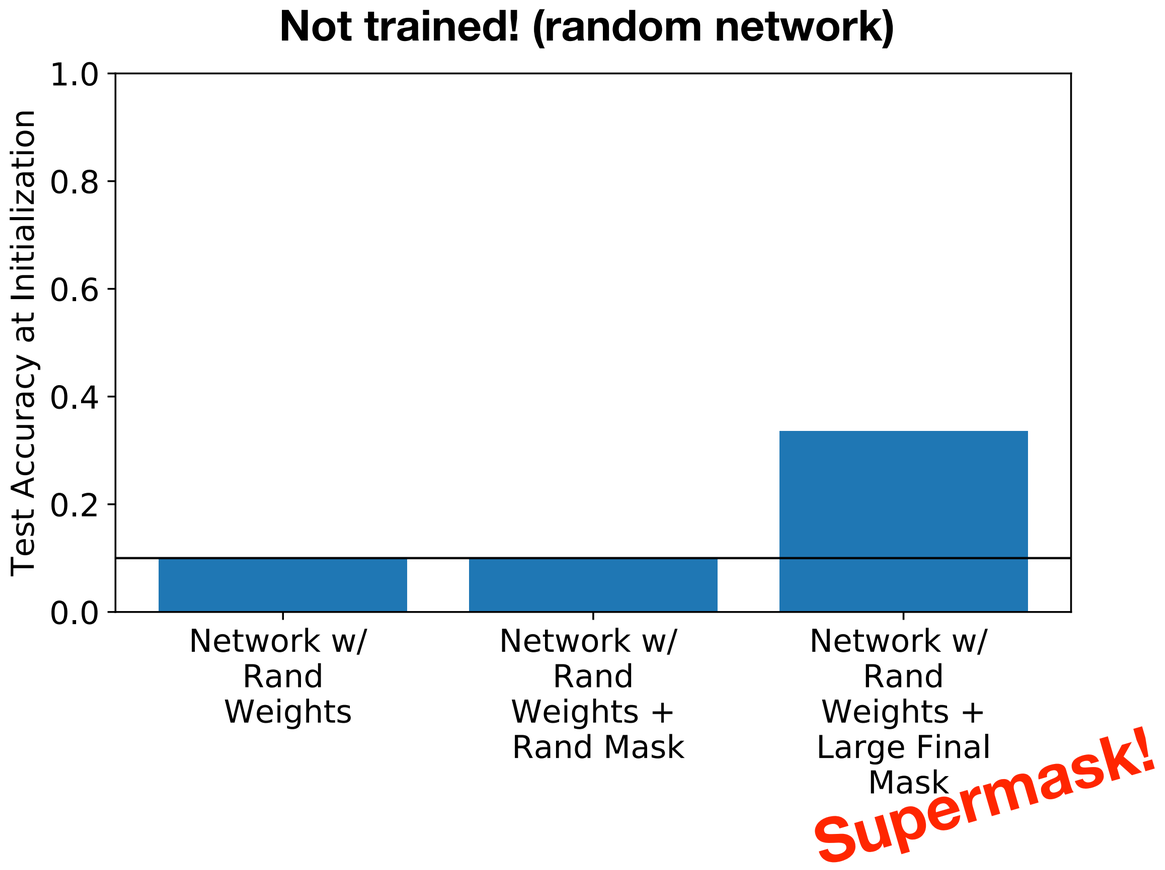
Curiously Effective Masks at Initialization



Curiously Effective Masks at Initialization



Curiously Effective Masks at Initialization



Directly training Supermasks

- $w' = w_i \odot g(m)$
- w' is the effective weight of the network
- g is a point-wise function that transform a matrix of continuous values into binary values
- $g(m) = \mathrm{Bern}(S(m))$

Sigmoid

bernoulli sampler

Directly training Supermasks: Works!

Network	mask ⊙ init	mask ⊙ S.C.	learned mask ⊙ init	learned mask ⊙ S.C.	DWR learned mask ⊙ init	DWR learned mask ⊙ S.C.	trained weights
MNIST FC	79.3	86.3	95.3	96.4	97.8	**98.0**	97.7
CIFAR Conv2	22.3	37.4	64.4	**66.3**	65.0	66.0	69.2
CIFAR Conv4	23.7	39.7	65.4	66.2	71.7	**72.5**	75.4
CIFAR Conv6	24.0	41.0	65.3	65.4	76.3	**76.5**	78.3

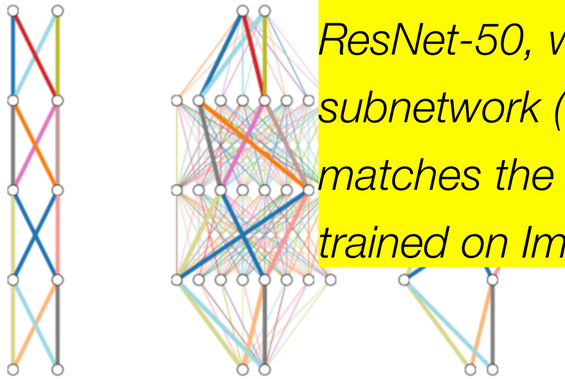
Magic: someone made it work for ImageNet 🥲

What's Hidden in a Randomly Weighted Neural Network?

Vivek Ramanujan *† Mitchell Wortsman *‡ Aniruddha Kembhavi †‡
Ali Farhadi †‡ Mohammad Rastegari †‡

Abstract

Training a neural network is synonymous with learning the values of the weights. In contrast, we demonstrate that randomly weighted neural networks contain subnetworks which achieve impressive performance without ever training the weight values. Hidden in a randomly weighted Wide ResNet-50 [28] we show that there is a subnetwork (with random weights) that is smaller than, but matches the performance of a ResNet-34 [8] trained on ImageNet [3]. Not only do these "untrained subnetworks" exist, but we provide an algorithm to effectively find them. We empirically show that as randomly weighted neural networks with fixed weights grow wider and deeper, an "untrained subnetwork" approaches a network with learned weights in accuracy.



A neural network τ which achieves good performance | Randomly initialized neural network N | A subnetwork τ' of N

Hidden in a randomly weighted Wide ResNet-50, we show that there is a subnetwork (with random weights) that matches the performance of a ResNet-34 trained on ImageNet.

More Magic: someone theoretically proved it 🥲

Proving the Lottery Ticket Hypothesis: Pruning is All You Need

Eran Malach [*1] **Gilad Yehudai** [*2] **Shai Shalev-shwartz** [1] **Ohad Shamir** [2]

Abstract

The lottery ticket hypothesis (Frankle and Carbin, 2018), states that a randomly-initialized network contains a small subnetwork such that, when trained in isolation, can compete with the performance of the original network. We prove an even stronger hypothesis (as was also conjectured in Ramanujan et al., 2019), showing that for every bounded distribution and every target network with bounded weights, a sufficiently over-parameterized neural network with random weights contains a subnetwork with roughly the same accuracy as the target network, without any further training.

without any training. (Ram
lowing conjecture: a suffic
network with random initi
that achieves competitive a
trained network), without a
be viewed as a stronger ve
esis.

In this work, we prove this
case of over-parameterized neural networks. Moreover, we differentiate between two types of subnetworks: subnetworks where specific weights are removed (*weight-subnetworks*) and subnetworks where entire neurons are removed (*neuron-subnetworks*). First, we show that a ReLU network of arbitrary depth l can be approximated by finding a *weight-*

Our work aims to give theoretical evidence to these empirical results. We prove the latter conjecture, stated in (Ramanujan et al., 2019), in the case of deep and shallow neural networks.

Leverage Supermasks for Continual Learning [Wortsman et al. 2020]

Supermasks in Superposition

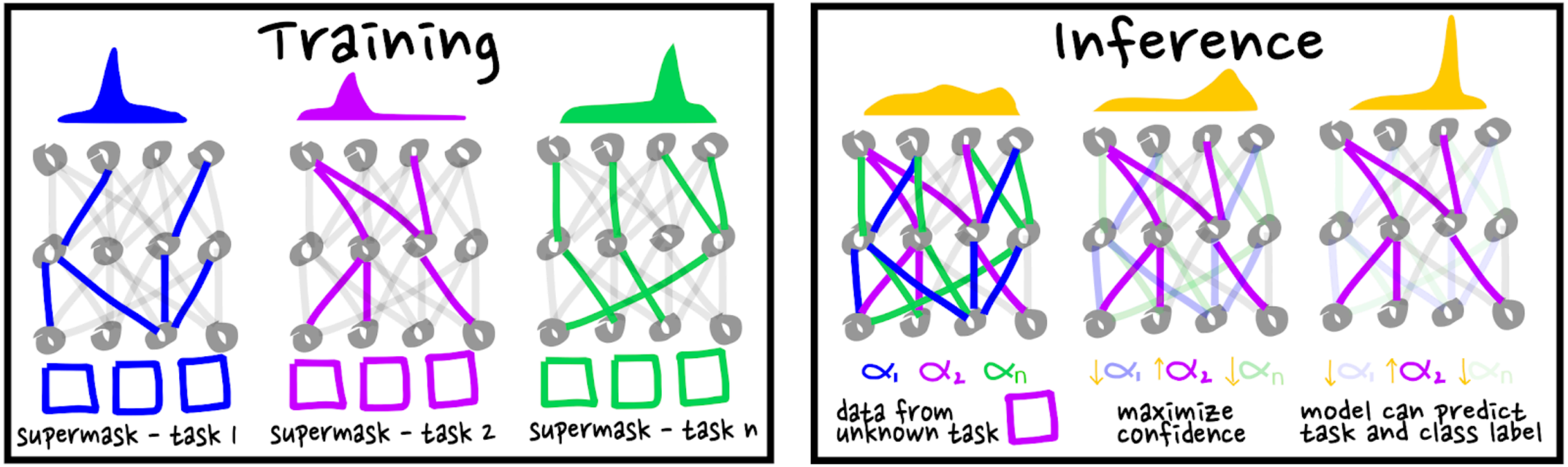
Mitchell Wortsman* University of Washington
Vivek Ramanujan* Allen Institute for AI
Rosanne Liu ML Collective
Aniruddha Kembhavi† Allen Institute for AI

Mohammad Rastegari University of Washington
Jason Yosinski ML Collective
Ali Farhadi University of Washington

Abstract

We present the Supermasks in Superposition (SupSup) model, capable of sequentially learning thousands of tasks without catastrophic forgetting. Our approach uses a randomly initialized, fixed base network and for each task finds a subnetwork (supermask) that achieves good performance. If task identity is given at test time, the correct subnetwork can be retrieved with minimal memory usage. If not provided, SupSup can infer the task using gradient-based optimization to find a linear superposition of learned supermasks which minimizes the output entropy. In practice we find that a single gradient step is often sufficient to identify the correct mask, even among 2500 tasks. We also showcase two promising extensions. First, SupSup models can be trained entirely without task identity information, as they may detect when they are uncertain about new data and allocate an additional supermask for the new training distribution. Finally the entire, growing set of supermasks can be stored in a constant-sized reservoir by implicitly storing them as attractors in a fixed-sized Hopfield network.

Leverage Supermasks for Continual Learning [Wortsman et al. 2020]



What it means for model capacity

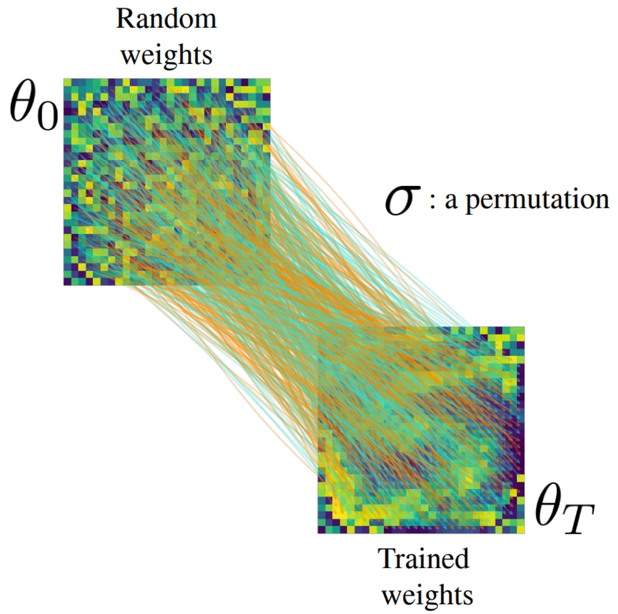
4. Train-by-Reconnect [Qiu et al. 2020]

Train-by-Reconnect: Decoupling Locations of Weights from Their Values

Yushi Qiu **Reiji Suda**
Graduate School of Information Science and Technology, The University of Tokyo
{yushi621, reiji}@is.s.u-tokyo.ac.jp

Abstract

What makes untrained deep neural networks (DNNs) different from the trained performant ones? By zooming into the weights in well-trained DNNs, we found that it is the *location* of weights that holds most of the information encoded by the training. Motivated by this observation, we hypothesized that weights in DNNs trained using stochastic gradient-based methods can be separated into two dimensions: the location of weights, and their exact values. To assess our hypothesis, we propose a novel method called *lookahead permutation* (LaPerm) to train DNNs by reconnecting the weights. We empirically demonstrate LaPerm's versatility while producing extensive evidence to support our hypothesis: when the initial weights are random and dense, our method demonstrates speed and performance similar to or better than that of regular optimizers, e.g., *Adam*. When the initial weights are random and sparse (many zeros), our method changes the way neurons connect, achieving accuracy comparable to that of a well-trained dense network. When the initial weights share a single value, our method finds a weight agnostic neural network with far-better-than-chance accuracy.



5. Train BN and only BN [Frankle et al. 2020]

Training BatchNorm and Only BatchNorm: On the Expressivity of Random Features in CNNs

Jonathan Frankle*
MIT CSAIL
jfrankle@mit.edu

David J. Schwab
CUNY Graduate Center, ITS
Facebook AI Research
dschwab@fb.com

Ari S. Morcos
Facebook AI Research
arimorcos@fb.com

Abstract

Batch normalization (BatchNorm) has become an indispensable tool for training deep neural networks, yet it is still poorly understood. Although previous work has typically focused on studying its normalization component, BatchNorm also adds two per-feature trainable parameters—a coefficient and a bias—whose role and expressive power remain unclear. To study this question, we investigate the performance achieved when training *only* these parameters and freezing all others at their random initializations. We find that doing so leads to surprisingly high performance. For example, sufficiently deep ResNets reach 82% (CIFAR-10) and 32% (ImageNet, top-5) accuracy in this configuration, far higher than when training an equivalent number of randomly chosen parameters elsewhere in the network. BatchNorm achieves this performance in part by naturally learning to disable around a third of the random features. Not only do these results highlight the under-appreciated role of the affine parameters in BatchNorm, but—in a broader sense—they characterize the expressive power of neural networks constructed simply by shifting and rescaling random features.

6. Train Perturbations [Dathathri et al. 2019]

PLUG AND PLAY LANGUAGE MODELS: A SIMPLE APPROACH TO CONTROLLED TEXT GENERATION

Sumanth Dathathri *
CMS, Caltech

Andrea Madotto *
HKUST

Janice Lan
Uber AI

Jane Hung
Uber AI

Eric Frank
Uber AI

Piero Molino
Uber AI

Jason Yosinski †
Uber AI

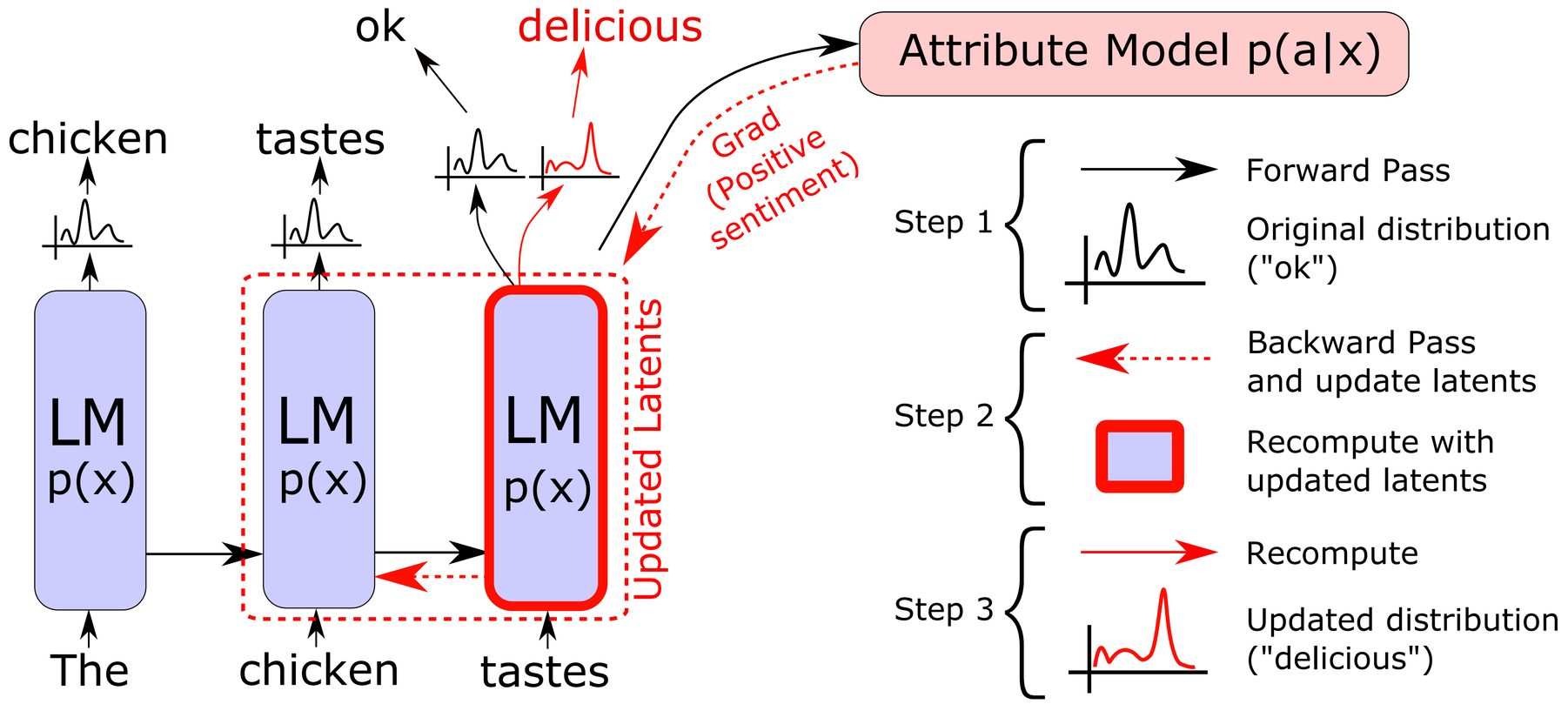
Rosanne Liu †
Uber AI

dathathris@gmail.com, amadotto@connect.ust.hk
{janlan, jane.hung, mysterefrank, piero, yosinski, rosanne}@uber.com

ABSTRACT

Large transformer-based language models (LMs) trained on huge text corpora have shown unparalleled generation capabilities. However, controlling attributes of the generated language (e.g. switching topic or sentiment) is difficult without modifying the model architecture or fine-tuning on attribute-specific data and entailing the significant cost of retraining. We propose a simple alternative: the Plug and Play Language Model (PPLM) for controllable language generation, which combines a pretrained LM with one or more simple attribute classifiers that guide text generation without any further training of the LM. In the canonical scenario we present, the attribute models are simple classifiers consisting of a user-specified bag of words or a single learned layer with 100,000 times fewer parameters than the LM. Sampling entails a forward and backward pass in which gradients from the attribute model push the LM's hidden activations and thus guide the generation. Model samples demonstrate control over a range of topics and sentiment styles, and extensive automated and human annotated evaluations show attribute alignment and fluency. PPLMs are flexible in that any combination of differentiable attribute models may be used to steer text generation, which will allow for diverse and creative applications beyond the examples given in this paper.

6. Train Perturbations [Dathathri et al. 2019]



I've talked about "weird ways of training NNs":

- Indirect encoding (e.g. HyperNetworks)
- Random subspace training (Intrinsic Dimension)
- SuperMasks
- Train by shuffling weight positions
- Train by renormalizing (e.g. BN parameters)
- Train activation perturbations (e.g. PPLM)

- [Dathathri et al. 2019] Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J. and Liu, R., 2019. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In International Conference on Learning Representations, 2019.
- [Frankle & Carbin, 2019] Frankle, J. and Carbin, M., 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In International Conference on Learning Representations, 2019.
- [Frankle et al. 2020] Frankle, J., Schwab, D.J. and Morcos, A.S., 2020. Training BatchNorm and Only BatchNorm: On the Expressive Power of Random Features in CNNs. arXiv preprint arXiv:2003.00152.
- [Ha et al. 2017] Ha, D., Dai, A. and Le, Q.V., 2016. Hypernetworks. arXiv preprint arXiv:1609.09106.
- [Han et al. 2015] Han, S., Pool, J., Tran, J. and Dally, W., 2015. Learning both weights and connections for efficient neural network. Advances in neural information processing systems, 28, pp.1135-1143.
- [Li et al. 2018] Li, C., Farkhoor, H., Liu, R. and Yosinski, J., 2018, February. Measuring the Intrinsic Dimension of Objective Landscapes. In International Conference on Learning Representations, 2018.
- [Malach et al. 2020] Malach, E., Yehudai, G., Shalev-Schwartz, S. and Shamir, O.. (2020). Proving the Lottery Ticket Hypothesis: Pruning is All You Need. Proceedings of the 37th International Conference on Machine Learning, 119:6682-6691
- [Qiu et al. 2020] Qiu, Y. and Suda, R., 2020. Train-by-Reconnect: Decoupling Locations of Weights from Their Values. Advances in Neural Information Processing Systems, 33.
- [Ramanujan et al. 2019] Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A. and Rastegari, M., 2020. What's Hidden in a Randomly Weighted Neural Network?. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11893-11902).
- [Stanley et al. 2009] Stanley, K.O., D'Ambrosio, D.B. and Gauci, J., 2009. A hypercube-based encoding for evolving large-scale neural networks. Artificial life, 15(2), pp.185-212.
- [Wortsman et al. 2020] Wortsman, M., Ramanujan, V., Liu, R., Kembhavi, A., Rastegari, M., Yosinski, J. and Farhadi, A., 2020. Supermasks in superposition. Advances in Neural Information Processing Systems, 33.
- [Zhou et al. 2019] Zhou, H., Lan, J., Liu, R. and Yosinski, J., 2019. Deconstructing lottery tickets: Zeros, signs, and the supermask. In Advances in Neural Information Processing Systems (pp. 3597-3607)

Unconventional Ways of Training Neural Networks

and What They Teach Us about Model Capacity

Unconventional Ways of ~~Training Neural Networks~~ Being a ML researcher

and What They Teach Us about ~~Model~~ Capacity our own

Unconventional Ways of ~~Training Neural Networks~~

Being a ML researcher

and What They Teach Us about ~~Model~~ Capacity

our own

A 12-month Field Study

Rosanne: 2016.4 to 2020.2

Conventional ML researcher

Rosanne: 2016.4 to 2020.2

Conventional ML researcher

Finished long academic training (Ph.D)

Affiliated

Being paid to do research

Environment: little diversity

Cares about: # papers/citations, h-index

Short-term goal: next paper

Long-term goal: title / pay / prestige

Ultimate goal: prove individual worth

Unfortunate byproduct: entitlement, negligence of others' pain, anxiety, feeling lost, feeling unable to change anything until I "get there"

Rosanne: 2016.4 to 2020.2

Rosanne: 2020.2 - 2021.2

Conventional ML researcher

Finished long academic training (Ph.D)

Affiliated

Being paid to do research

Environment: little diversity

Cares about: # papers/citations, h-index

Short-term goal: next paper

Long-term goal: title / pay / prestige

Ultimate goal: prove individual worth

Unfortunate byproduct: entitlement, negligence of others' pain, anxiety, feeling lost, feeling unable to change anything until I "get there"

Rosanne: 2016.4 to 2020.2	*Rosanne: 2020.2 - 2021.2*
Conventional ML researcher	**Unconventional ML researcher**
Finished long academic training (Ph.D)	
Affiliated	
Being paid to do research	
Environment: little diversity	
Cares about: # papers/citations, h-index	
Short-term goal: next paper	
Long-term goal: title / pay / prestige	
Ultimate goal: prove individual worth	
Unfortunate byproduct: entitlement, negligence of others' pain, anxiety, feeling lost, feeling unable to change anything until I "get there"	

Rosanne: 2016.4 to 2020.2	*Rosanne: 2020.2 - 2021.2*
Conventional ML researcher	**Unconventional ML researcher**
Finished long academic training (Ph.D)	
Affiliated	Unaffiliated
Being paid to do research	Not taking pay to do research
Environment: little diversity	
Cares about: # papers/citations, h-index	
Short-term goal: next paper	
Long-term goal: title / pay / prestige	
Ultimate goal: prove individual worth	
Unfortunate byproduct: entitlement, negligence of others' pain, anxiety, feeling lost, feeling unable to change anything until I "get there"	

Rosanne: 2016.4 to 2020.2	*Rosanne: 2020.2 - 2021.2*
Conventional ML researcher	**Unconventional ML researcher**
Finished long academic training (Ph.D)	
Affiliated	Unaffiliated
Being paid to do research	Not taking pay to do research
Environment: little diversity	Environment: more diversity
Cares about: # papers/citations, h-index	
Short-term goal: next paper	
Long-term goal: title / pay / prestige	
Ultimate goal: prove individual worth	
Unfortunate byproduct: entitlement, negligence of others' pain, anxiety, feeling lost, feeling unable to change anything until I "get there"	

Rosanne: 2016.4 to 2020.2	*Rosanne: 2020.2 - 2021.2*
Conventional ML researcher	**Unconventional ML researcher**
Finished long academic training (Ph.D)	
Affiliated	Unaffiliated
Being paid to do research	Not taking pay to do research
Environment: little diversity	Environment: more diversity
Cares about: # papers/citations, h-index	Cares about: # people I **help** get into research
Short-term goal: next paper	Short-term goal: next person to **help**
Long-term goal: title / pay / prestige	Long-term goal: a **help**ing community
Ultimate goal: prove individual worth	Ultimate goal: American dream for ML research
Unfortunate byproduct: entitlement, negligence of others' pain, anxiety, feeling lost, feeling unable to change anything until I "get there"	

ML Collective

http://mlcollective.org

Help people publish their 1st ML paper, help connect them to the right mentor, get into ML, know what to expect & what to do…

Help normalize the expectation of ML research

Help level the playing field by redistributing opportunity

Help build a community of collaborators, not competitors

Help widen the path into ML research, at least a little bit

…

ML Collective

http://mlcollective.org

+ publish great research!

Help people publish their 1st ML paper, help connect them to the right mentor, get into ML, know what to expect & what to do…

Help normalize the expectation of ML research

Help level the playing field by redistributing opportunity

Help build a community of collaborators, not competitors

Help widen the path into ML research, at least a little bit

…

Rosanne: 2016.4 to 2020.2	*Rosanne: 2020.2 - 2021.2*
Conventional ML researcher	**Unconventional ML researcher**
Finished long academic training (Ph.D)	
Affiliated	Unaffiliated
Being paid to do research	Not taking pay to do research
Environment: little diversity	Environment: more diversity
Cares about: # papers/citations, h-index	Cares about: # people I **help** get into research
Short-term goal: next paper	Short-term goal: next person to **help**
Long-term goal: title / pay / prestige	Long-term goal: a **help**ing community
Ultimate goal: prove individual worth	Ultimate goal: American dream for ML research
Unfortunate byproduct: entitlement, negligence of others' pain, anxiety, feeling lost, feeling unable to change anything until I "get there"	**Unfortunate byproduct: not taken as a serious researcher until they know you better**

Rosanne: 2016.4 to 2020.2		*Rosanne: 2020.2 - 2021.2*
Conventional ML researcher		**Unconventional ML researcher**
Finished long academic training (Ph.D)		
Affiliated		Unaffiliated
Being paid to do research		Not taking pay to do research
Environment: little diversity	←	Environment: more diversity
Cares about: # papers/citations, h-index	←	Cares about: # people I **help** get into research
Short-term goal: next paper	←	Short-term goal: next person to **help**
Long-term goal: title / pay / prestige	←	Long-term goal: a **help**ing community
Ultimate goal: prove individual worth	←	Ultimate goal: American dream for ML research
Unfortunate byproduct: entitlement, negligence of others' pain, anxiety, feeling lost, feeling unable to change anything until I "get there"		Unfortunate byproduct: not taken as a serious researcher until they know you better

Rosanne: 2016.4 to 2020.2

Rosanne: 2020.2 - 2020.12

Conventional ML researcher		Unconventional ML researcher
Finished long academic training (Ph.D)		
Affiliated		Unaffiliated
Being paid to do research		Not taking pay to do research
Environment: little diversity	←	Environment: more diversity
Cares about: # papers/citations, h-index	←	Cares about: # people I **help** get into research
Short-term goal: next paper	←	Short-term goal: next person to **help**
Long-term goal: title / pay / prestige	←	Long-term goal: a **help**ing community
Ultimate goal: prove individual worth	←	Ultimate goal: American dream for ML research
~~Unfortunate byproduct: entitlement, negligence of others' pain, anxiety, feeling lost, feeling unable to change anything until I "get there"~~		~~Unfortunate byproduct: not taken as a serious researcher until they know you better~~

We as ML researchers have more capacity than we think.

We as ML researchers have more capacity than we think.

Not only can we produce great research,

We as ML researchers have more capacity than we think.

Not only can we produce great research, but we are also **totally capable** of making society better and ourselves happier,

We as ML researchers have more capacity than we think.

Not only can we produce great research, but we are also **totally capable** of making society better and ourselves happier, by simply changing our objective function from “individual achievement” to “helping people, nurturing society.”

We as ML researchers have more capacity than we think.

Not only can we produce great research, but we are also **totally capable** of making society better and ourselves happier, by simply changing our objective function from “individual achievement” to “helping people, nurturing society.”

I’ve run such experiment and presented positive results so you don’t have to run it yourself.

We as ML researchers have more capacity than we think.

Not only can we produce great research, but we are also **totally capable** of making society better and ourselves happier, by simply changing our objective function from “individual achievement” to “helping people, nurturing society.”

I’ve run such experiment and presented positive results so you don’t have to run it yourself.

Cite me though! 😊

We as ML researchers have more capacity than we think.

Not only can we produce great research, but we are also **totally capable** of making society better and ourselves happier, by simply changing our objective function from “individual achievement” to “helping people, nurturing society.”

I’ve run such experiment and presented positive results so you don’t have to run it yourself.

Cite me though! 😊

Twitter: @savvyRL
Email: rosanne@mlcollective.org
Website: http://rosanneliu.com/