

# From “datascience” in RocketChat: Search for “facebookresearch”

arbitrage

[GitHub](#)

[facebookresearch/open\\_lth](#)

A repository in preparation for open-sourcing lottery ticket hypothesis code. - facebookresearch/open\_lth

jrb

Not to downplay the importance of the lottery ticket hypothesis paper, but the results from “Training BatchNorm and Only BatchNorm” paper (linked in that repo) are some of the most mind blowing results I’ve seen in DL literature, lately.

arbitrage

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 its normalization component, BatchNorm also adds two per-feature trainable parameters: a coefficient and a bias. However, the role and expressive power of these parameters remains 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, a sufficiently deep ResNet reaches 83% accuracy on CIFAR-10 in this configuration. Interestingly, BatchNorm achieves this performance in part by naturally learning to disable around a third of the random features without any changes to the training objective. 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.”

jrb

In other words, a randomly initialized network with frozen weights, can be trained to 83% accuracy by just learning the batch mean and variance at every layer. That’s bizarre and a completely unexpected result!

on CIFAR-10

with a CNN.

bor

isn’t that pretty close to what that paper found on evolving really weird neural network architectures. Didn’t really need to train the NN’s.

high on my todo list

not sure if this was the paper, but something along these lines

[Google AI Blog](#)

## [Using Evolutionary AutoML to Discover Neural Network Architectures](#)

Posted by Esteban Real, Senior Software Engineer, Google Brain Team The brain has evolved over a long time, from very simple worm brains 500...

[Google AI Blog](#)

## [Google AI Blog](#)

The latest news from Google AI

Using Evolutionary AutoML to Discover Neural Network Architectures

Posted by Esteban Real, Senior Software Engineer, Google Brain Team The brain has evolved over a long time, from very simple worm brains 500...

evolution baby

I see that way back wsouza is doing something like that already - evolving neural networks

vantratone

Extreme Learning Machines just make a single-layer NN with random weights (sort of) and they can work amazingly well

(on some things)

bor

feed the mmc

jrb

vantratrone Rocket is similar to ELMs but with 1d convolutions for time series data.

Evolutionary computing is great for finding novel architectures. But it's nowhere near competitive to gradient descent in terms of training efficiency.

Why not do both?