

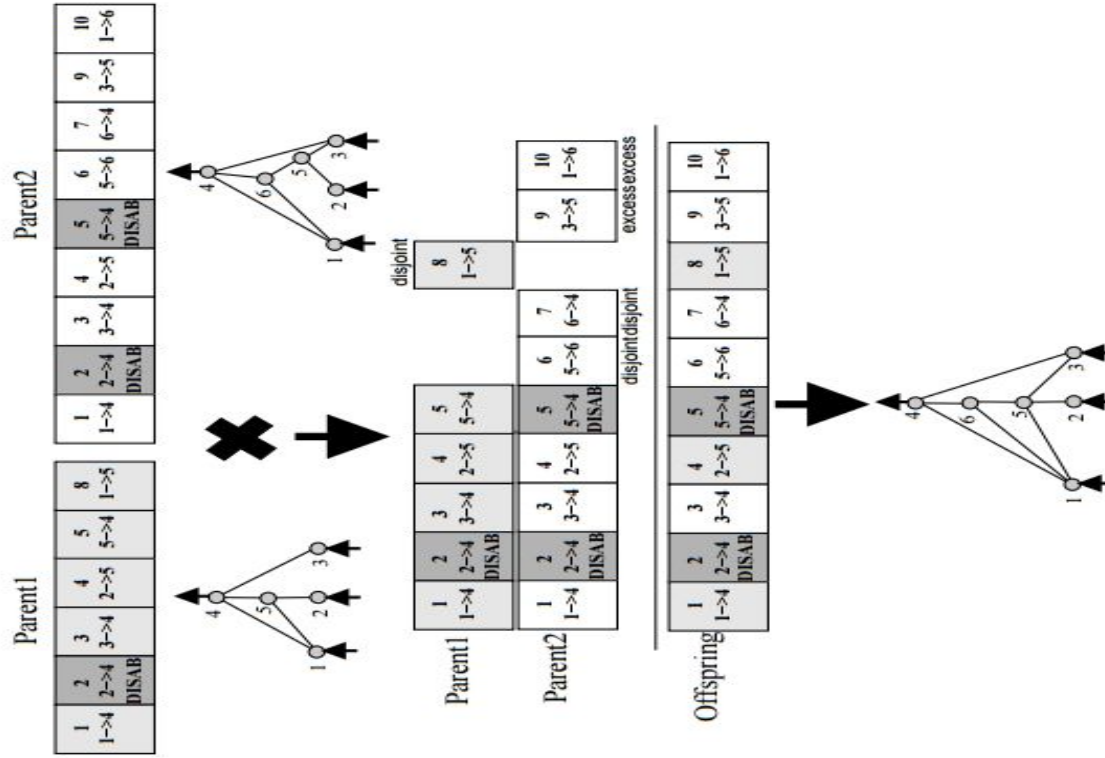
# NeuroEvolution of Augmenting Topologies (NEAT) in TensorFlow Eager

Open Source Project

Cristian ([cb2015@cam.ac.uk](mailto:cb2015@cam.ac.uk))

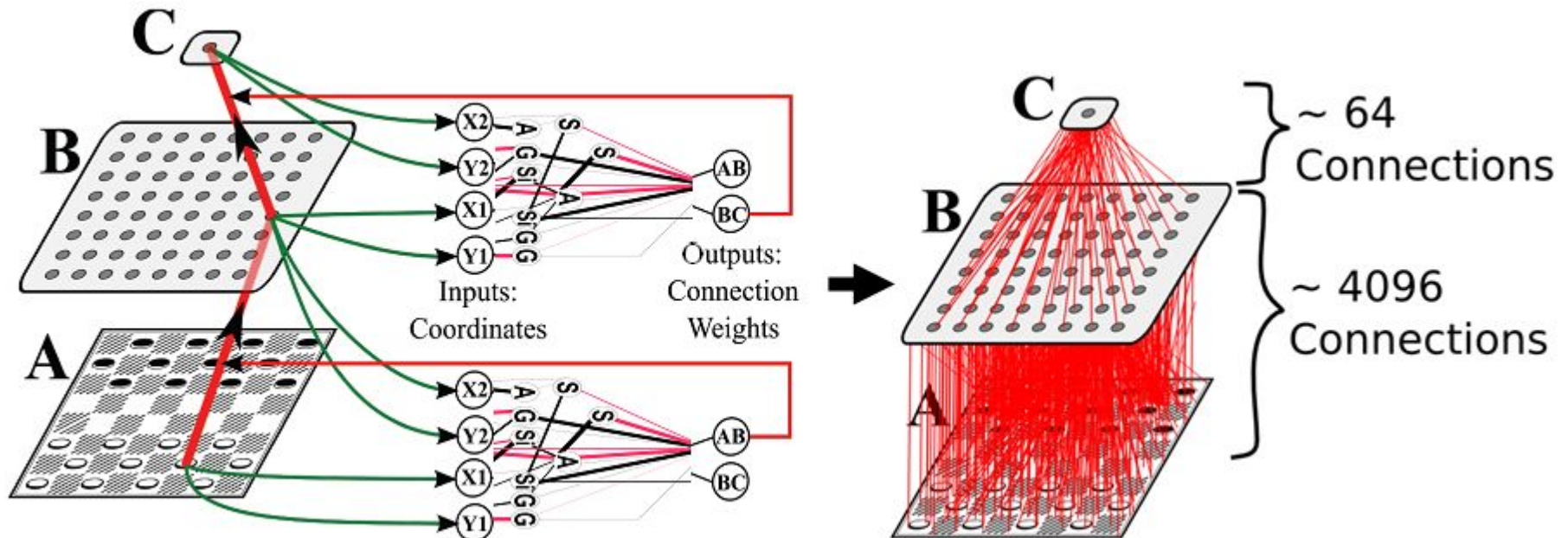
# NEAT

NEAT is a popular genetic algorithm that learns both the topology of a neural network and the weights.



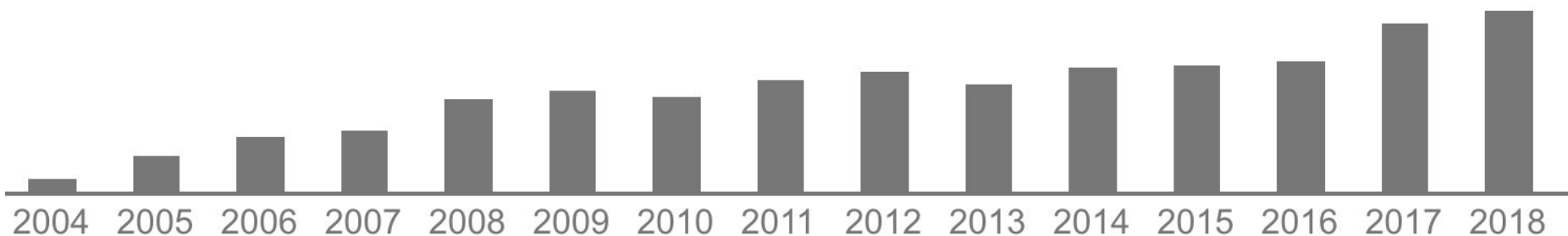
# HyperNEAT

Its variant, HyperNEAT is able to evolve Deep(er) Neural Networks with complex structures similar to those in the brain.



# NEAT

NEAT is still a popular algorithm, widely used in the research community. It has 254 citations in 2018 and over 2000 since it was proposed.



However, there is no mature TensorFlow implementation because until recently. TensorFlow did not support dynamic computation graphs.

# NEAT: Not in TensorFlow Fold



crisbodnar commented on 29 Oct



Would it be possible to write an algorithm like NEAT which evolves the topology of a neural network in Fold? From my understanding, Fold does not support dynamic computation graphs in this sense, but these dynamic graphs are rather input dependent.



crisbodnar changed the title from **NEAT implementation** to **NEAT Algorithm implementation** on 29 Oct



delesley commented on 29 Oct

Collaborator



The original NEAT algorithm evolved small neural networks at the level of individual neurons. That obviously won't work with tensorflow; calling a python function to schedule a tensorflow operation to invoke a cuda kernel to multiply two scalars would have ridiculously high overhead.

However, if you want to implement NEAT at the level of NN layers, rather than individual neurons, then Tensorflow Fold should work quite well. You'll want to use the loom library, not the blocks library.

First, create a separate LoomOp class for every NN operation you want to support. Second, implement NEAT on a population of programs. Each program in the population is a DAG of NN operations. Write a recursive python function which traverses the DAG for every program in the population, and invokes the appropriate LoomOp for each node. There is a calculator example in loom that shows how to do this for arithmetic expressions. Loom will handle the dynamic batching for you, and evaluate the LoomOps for all programs using TensorFlow.

It might be easier to build a prototype using TensorFlow Eager, and then switch to loom. Loom should give you a nice ~30x speedup over eager due to dynamic batching.

-DeLesley



# Project Goal

The goal of the project is to build an open source NEAT library in TensorFlow Eager and compare its performance against the Uber Research PyTorch library.



# Plan

1. Analyse the Uber Research PyTorch NEAT library and NEAT-Python.
2. Build vanilla NEAT in TensorFlow Eager.
3. Compare it against PyTorch on a simple RL task such as Cartpole.
4. If time allows it, other versions of NEAT such as HyperNEAT can be implemented.

# The End

Thank you!  
Questions?