Introducing Trophic Incoherence to Traditional Neural Network Architectures

UNIVERSITY OF BIRMINGHAM

Asa Hopkins, Supervised by Dr. Samuel Johnson

Contact: asa.hopkins@strath.ac.uk
Github: https://github.com/Asa-Hopkins/Trophic-Networks

Artificial neural networks (NNs) at their core are an attempt to emulate the biological NNs found in the brains of animals, and can accomplish certain tasks more efficiently than more traditional computing methods. However, there are still ways that artifical NNs fall short of their biological counterparts. Most artificial NNs are made up of layers of nodes, with edges only being formed between adjacent layers. This kind of strict ordering is not seen in nature and the existence of edges connecting non-adjacent layers is important to the stability of larger natural systems, such as food chains and metabolic pathways. The extent to which this strict layering is broken is known as the trophic incoherence. This work investigates methods of adding trophic incoherence to artificial NNs, and the effects doing so has on the convergence speed during training (fast convergence is more energy efficient) and the accuracy after training is completed.

What is Trophic Incoherence?

Trophic incoherence is a way of measuring the amount of disorder in a directed graph, and it stems from the idea of a trophic level. We first define the trophic level of a node as the average of the trophic levels of the nodes that it takes inputs from, plus one. To completely define the trophic levels of the systems, nodes with no inputs are defined as having a trophic level of one.

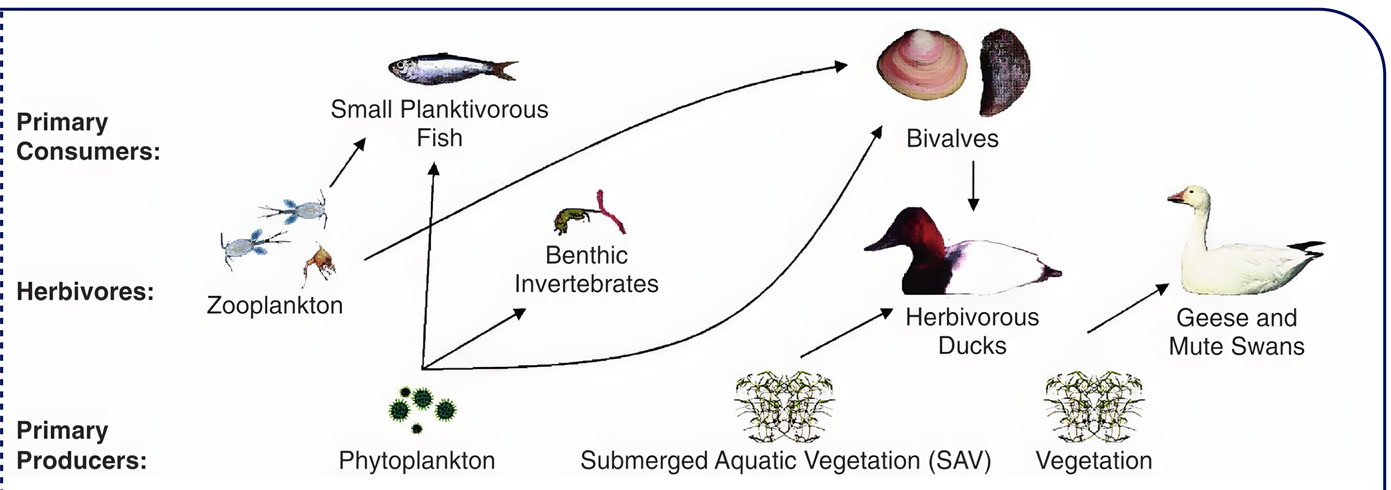
As an example, let's calculate the trophic level of the Herbivorous Ducks.

$$S_{Phyto}, S_{SAV} := 1, S_{Zoo} := 2$$

$$S_{Bivalve} = \frac{1}{2}\left(S_{phyto} + S_{Zoo}\right) + 1 = 2.5$$

$$S_{Ducks} = \frac{1}{2}\left(S_{SAV} + S_{Bivalve}\right) + 1 = 2.75$$

This places them somewhere between Herbivores and Primary Consumers

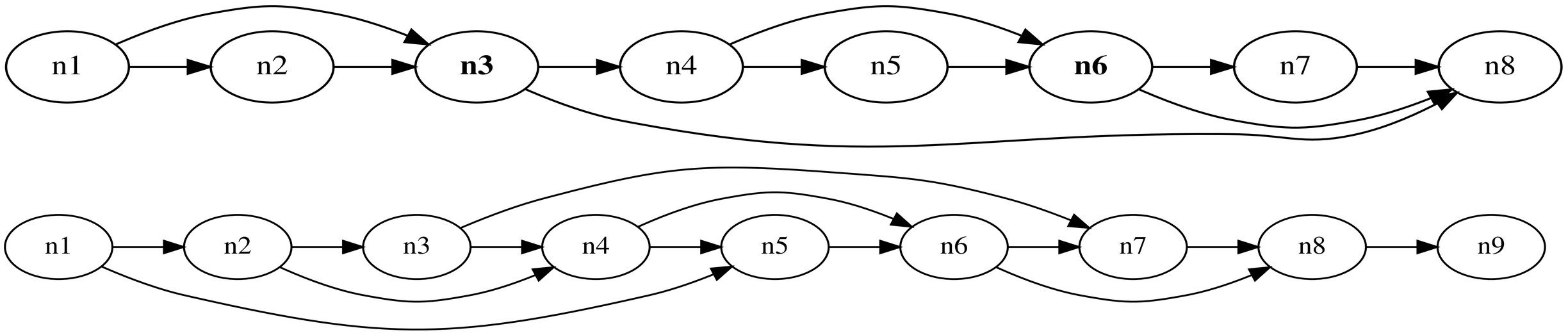


Consider the the "Herbivorous Ducks" in this image. They eat both primary producers and primary consumers, which makes it unclear whether they should be placed in herbivores or in secondary consumers. In this strictly layered diagram, there is no good option.

Connectivity

There are many ways in which incoherence can be introduced, and an early idea was to use randomly generated directed acyclic graphs, but this loses the advantage of efficient matrix multiplication for calculating node values. The approach that was eventually taken was to still have strict layers of nodes, but to allow connections to appear between non-adjacent layers. In the diagrams below, each circle represents an entire layer of nodes, and each connection represents a matrix of weights.

Even with this additional restriction, there are many possible choices of ways to add connections between layers. This is called the connectivity, and multiple connectivities have been tested.



Top: The MaxDist2 Model

Bottom: The ResNetX Model

Why Might Incoherence Help?

When calculating the derivative for gradient descent, each step in the connectivity graph represents one application of the chain rule in calculating the derivative with respect to that layer, and there is a tendency for this value to decrease as more applications of the chain rule are needed. The two connectivity models introduced, MaxDist1 and MaxDist2, ensure that a route to the output layer exists with only one or two steps respectively, meaning that this issue of vanishing gradients is mitigated.

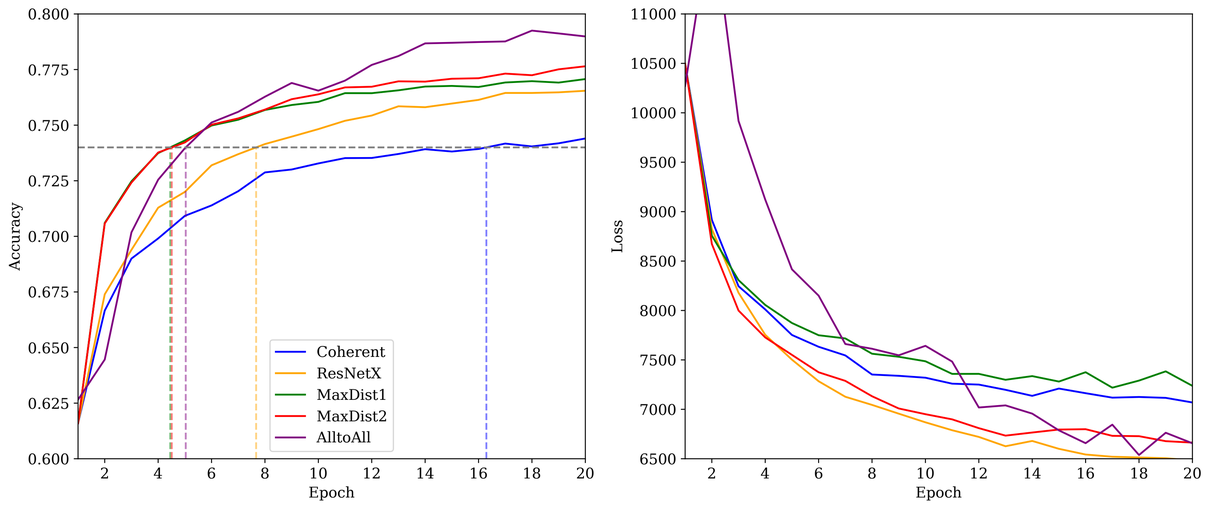
Neural Network Basics

Feed forward neural networks are a class of vector functions defined by the iteration

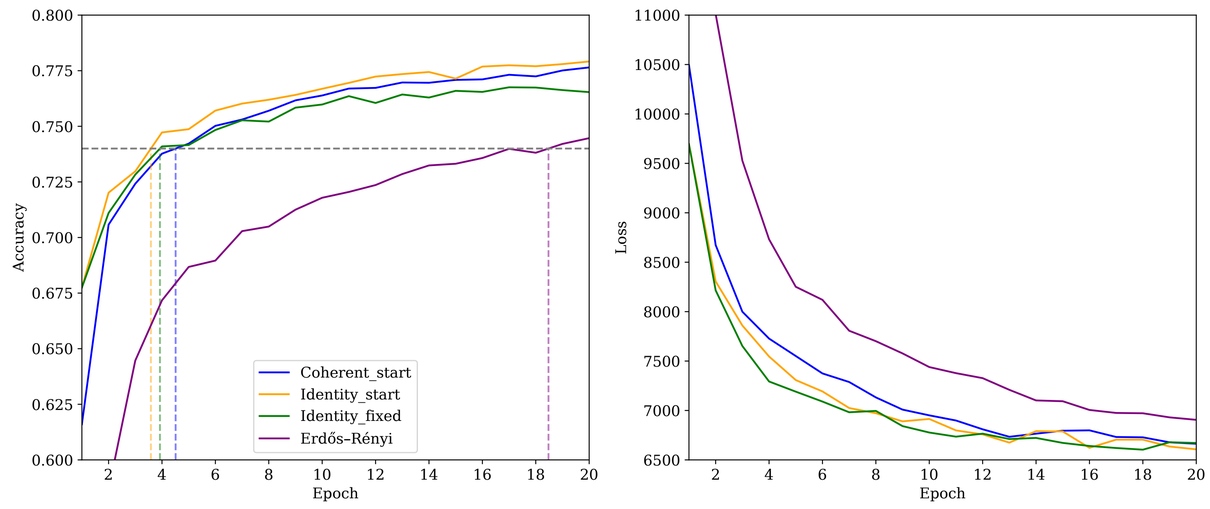
$$\mathbf{p}_{i+1} = \mathbf{W}_i \mathbf{n}_i, \qquad \mathbf{n}_{i+1} = f_i(\mathbf{p}_{i+1}).$$

Here $\mathbf{n}_0$ represents the input, $\mathbf{n}_l$ represents the output, each $\mathbf{W}$ is a matrix and each f is a (typically elementwise) nonlinear function. Initially, the elements of the matrices are random, but by inputting values which have a known output, an error can be calculated. The derivative of this error is then calculated, and a small step is taken in the direction of steepest descent to try and reduce it

Results for a 20 Layer Network on EMNIST

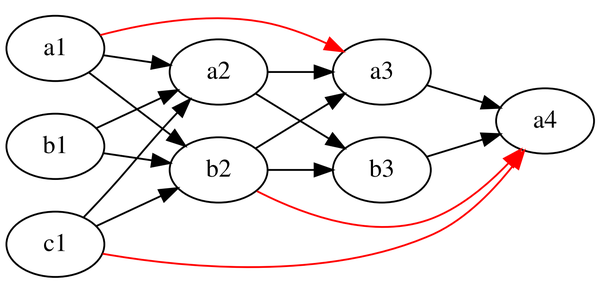


For this set of results, different connectivity models were tested. MaxDist1 and MaxDist2, show accelerated convergence and therefore require less training. In theory, an all-to-all connectivity should be ideal, as allowing more options for placing connections will never increase the global minimum, but in practise this makes the solution space harder to search.



For this set of results, MaxDist2 connectivity was used and different choices of inital connections were tried. It can be seen here that a good choice of initialisation can improve convergence by an epoch, although it is not obvious beforehand what choice is optimal.

University of Strathclyde Glasgow



UK Research and Innovation