



Live Agile Artificial Neural Network Programming Sessions

Live Agile Artificial Neural Network Programming Sessions

Fun note:

1000+ lines of basic neural network code was written in Java from memory/scratch by God Bennett for the session below. God can also write the same 1000+ lines of neural network code at any given time from memory/scratch, [after years of bi-annual practice](#). Note that memorizing this neural network code as God does, is reasonably not required, although writing one out step by step is beneficial, as discussed below.

In an agile setting where months may not be available for data science courses, this is an optimal way of quickly getting a developer of starting or intermediate coding skills to understand overall layout of machine learning.

Since artificial neural networks can represent a wide variety of utilities/functions as seen in [Universal Approximation Theorem](#), and since neural networks power most cognitive/smart apps today, the agile artificial intelligence/machine learning process concerns 2 topics:

1. Most basic Neural network programming in detail (in BlueJ or any IDE of choice. BlueJ is good for quickly visualizing how classes connect, as coding takes place.)
2. Data preprocessing.

December, Friday 18, 2020, a live Neural Network coding Session done by God Bennett at NCB with small QA team: <https://github.com/JordanMicahBennett/Live-Agile-Artificial-Neural-Network-Programming-Sessions>

Code/video for a live recorded session of a basic neural network done by God Bennett (Legally changed name from: Jordan Bennett):

<https://www.youtube.com/watch?v=OhRkCb8XGj0>

Brain inspired computer code or smart apps, called **AGI** or **Artificial General Intelligence** (predicted to happen by [as soon as 2029 or sooner](#)), will perhaps one day be [mankind's last invention](#)! (Interested parties can see [MIT's AGI course here](#).)

For now though, AGI's predecessor, called **Artificial Narrow Intelligence**, also called **Artificial Intelligence**, can do amazing stuff like [diagnose diseases better than human doctors](#), enable [self driving cars](#), or [give game characters the ability to learn to navigate game environments without human aid](#)!

Crucially, [where Ai is already enhancing banking](#), [fortunately the Jamaican government](#) has recognized the impact that artificial intelligence already brings, and what shall likely happen futureward. I speak more about this in the cleaner articles found on the newspaper tab [on this experimental platform of mine](#).

Quick Overview of Seminar:

- 1.) All of successful artificial intelligence algorithms today perform something called [error minimization](#).



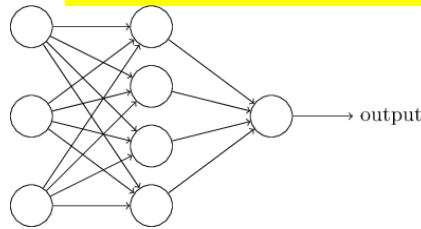
- 2.) They work **similar** to how biological brains work.



- 3.) For eg, with a **high error rate**, a child will first **wrongly** identify objects in the world in his/her earlier years of life.

3.b.) That error rate gets **smaller** or is minimized, as the child gets better at identifying objects in the world; in the early years, a parent can guide the child by saying this is a cat or this is a dog etc, i.e. the parent helps the child to **correctly label** objects in world.

3.c.) After a while, even without parental guidance, the child will be quite good at identifying objects, and his/her error rate at object identification would have been **minimized substantially**.



4.) Artificial Intelligence works in a similar way; for a particular task, they start out **terrible** with high error rates, then they get far better after being exposed to many instances of correctly labelled things, until they get to a point of doing the task well, even without being exposed to correctly labelled data.

5.) Artificial neural networks power most smart apps today.

On December 18 2020, I spent about 1 hour to write a basic artificial neural network in java from scratch/memory without using libraries or the internet. (The aim was to guide NCB QA staff along in writing a basic artificial neural network, as I wrote the neural network code)

This exercise was really geared towards preparing more NCB programmers to better apply machine learning libraries.

6.) Though optional, understanding basic neural nets (even a non-math heavy, but programmatic understanding instead) can afford the programmer better grasp of applying machine learning libraries such as **tensorflow** built by Google or **azure ml** by Microsoft or other ones by other parties like **the one I used to help code the artificial neural network based credit card fraud detection system demo at National Commercial Bank**. (We **didn't** go through [the math behind basic neural nets](#), but we went through the relatively simple live aforementioned programming example instead, re xor neural network.)

Regards,