

Gradient Memory

Theory: A neural network uses a one dimensional gradient array of three dimensional color vectors to store and fetch information. Having each information be assigned to a specific color vector.

Practical Example:

Let's assume a Neural Network named "X"

Input: 3D Color Vector [e.g: (1.0, 0.5, 0.62)]

Output: 3D Color Vector [e.g: (1.0, 0.65, 1.0)]

Access Memory (AM): 1D Gradient (Blank at start)

Comparable Memory (CM) : 1D Gradient (Color Spectrum)

Input Data: {

orange: (1.0, 1.0, 0.0),

rottenOrange: (0.0, 0.0, 1.0)

}

Input: orange **OR** rottenOrange

Output: accessMemory: [(1.0, 1.0, 0.0), (0.0, 0.0, 0.0), ...] **OR** [(0.0, 0.0, 1.0), (0.0, 0.0, 0.0), ...]

Logic: On first start it'll have it's Gradient be blank and have an extra comparable memory gradient which it'll compare it's input with.

For each input it'll compare it with **CM** and append the value in the index of the matched in the **CM** in **AM**.

This **AM** Gradient can now be used to execute specific tasks. For example the model could avoid any value near the colder region (like **rottenOrange**). It can learn and decide which object is more likely to be picked or what action is more logical to take based on the color vector assigned to the input variable.