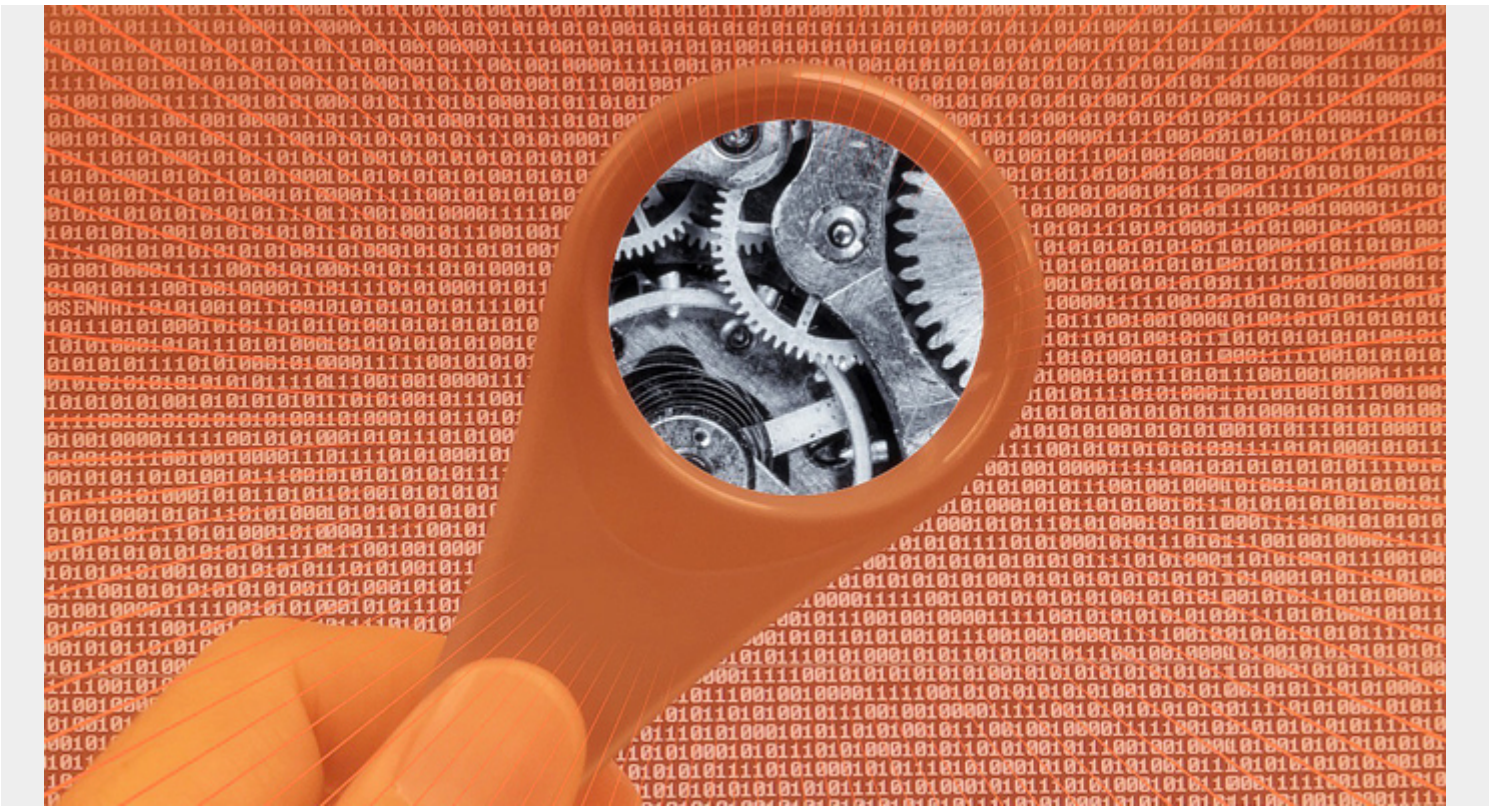
USING GPUS (GRAPHICAL PROCESSING UNITS) FOR MACHINE LEARNING



You are probably familiar with Nvidia as they have been developing graphics chips for laptops and desktops for many years now. But the company has found a new application for its graphic processing units (GPUs): machine learning. It is called **CUDA**.

Nvidia says:

> "CUDA® is a parallel computing platform and programming model invented by NVIDIA. It enables dramatic increases in computing performance by harnessing the power of the graphics processing unit (GPU)...CUDA-capable GPUs have hundreds of cores that can collectively run thousands of computing threads."

Contrast that number with the typical Intel or AMD chip, which has 4 or 8 cores.

This is important for machine learning because what this means is it can do matrix multiplication on a large scale and do it fast. That is important because, if you recall from our reading, to **train** a neural network means to find the weights and bias that yield the lowest cost using an **activation function**. The algorithm to do that is usually **gradient descent**. All of this requires multiplying very large matrices of numbers. This can be done in parallel since the order you do that does not matter.

To recall, finding the solution to a neural network means to apply different coefficient (weights) to each input variable, make a prediction, then see how accurate that prediction is. Then, using the gradient descent algorithm, we try different coefficients and repeat the process. You keep doing

that over and over until you reach the point where the neural network most accurately predicts whatever is is supposed to predict. With a large neural network of many thousands of **sigmoids** (nodes) that can take days.

To further make this simpler to understand, the neural network is represented as a series of **inputs X weights** W and a **bias** B, where X, W, and B are matrices. This yields an **output**, which is typically a small vector of says , as in the case of handwritten digit recognition. So we have:

X * W + B

X * W is the dot-product of 2 n-dimensional matrices. For example, for 2 dimensional matrices ,X and W, that is shown below where X is the first matrix, W is the 2nd, and X * W is the **dot product**. The dot product is the sum of multiplication of each corresponding row-column combination: