Difference between CPU and GPU operations

Let's first see difference b/w architectures Of GPU and CPU to understand their respective operations .
CPU is composed of very few cores, but those cores are individually very powerful and smart whereas **GPU** is composed of a large number of weaker cores.

Therefore CPUs can handle very few computations(may be complex or simple) at a time whereas GPUs can handle a large number of simple computations at a time.
CPUs perform all the computations **sequentially** whereas GPUs divide the task into smaller computations and perform them **parallely** with the help of large no. of cores**.**

But Sometimes carrying out computations in GPU is not so efficient . As sometimes we need to transfer the data from CPU to GPU to carry out parallel computations in GPU. If the data is large, GPU can have a **Bandwidth bottleneck issue** i.e. transferring a large amount of data to GPU might be slow.

But if our computation is very minute, say we have to **add two 4*4 matrices.** Even if this task can be done parallely with 16 parallel (addition) operations. But a very few number of cores are required for the same and as we know a single CPU core is more powerful than a single GPU core, CPU will outperform GPU in such tasks.

Also, if some computation/task involves recurrence i.e current output depends upon its previous value, then such tasks cannot be performed parallely. They must be done sequentially. So CPUs are best to do such tasks.

Now let's talk particularly for Deep Learning
In typical neural networks, there are a million parameters which define the model and require large amounts of data to learn these parameters. This is a computationally intensive process which takes a lot of time.In typical neural networks, there are a million parameters which define the model and requires large amounts of data to learn these parameters. Therefore it is important to

come up with parallel and distributed algorithms which can run much faster and which can drastically reduce training times.
These are the ways to use multiple cores to speed up the training process using GPUs:

- Use the cores to process multiple images at once, in each layer.
- Use multiple cores to perform Gradient descent of multiple mini-batches in parallel.
- Use GPU for computationally intensive calculations like matrix multiplication.

Also, if we talk about the task of **convolution** (in CNNs) in which a sliding kernel (of a fixed size, and is usually square in shape) which does an element-wise multiplication on each of the pixels of image it has an overlap with, and sums all the elements together. Here also, for each overlapping, calculation can be done parallely using GPU.