This work introduces the JIT-compiled cublas胎・kernel for elementwise operations. It evaluates the efficiency of c～c × c scaling vs quadratic elementwise scaling. The basic building block is a batch-reduce GEMM kernel that computes an elementwise operation (3) by implementing and optimizing this single kernel for a specific feature map size. It is used in practice by implementing and optimizing this single kernel for all DL primitives.

The introduction of the JIT-compiled cublas胎・kernel for elementwise operations enables efficient usage of tensor compiler frameworks such as PlaidML. The following figures highlight the micro-kernel targeting an NVIDIA 2080 Ti platform. The kernel takes the following arguments: the input/output tensors and the number of threads. The output tensor is computed by applying element-wise operations on all input features. The following figures illustrate the performance of the batch-reduce GEMM kernel on various GPUs.

The kernel is optimized for high efficiency, allowing it to scale well on modern GPUs. The performance of the batch-reduce GEMM kernel is compared with other state-of-the-art methods.

The following tables summarize the performance of the batch-reduce GEMM kernel on various GPUs. The kernel shows significant speedup compared to other methods. The performance is measured in terms of GFLOPS and images per second. The kernel is optimized for high efficiency, allowing it to scale well on modern GPUs. The performance of the batch-reduce GEMM kernel is compared with other state-of-the-art methods.

The following figures illustrate the performance of the batch-reduce GEMM kernel on various GPUs. The kernel shows significant speedup compared to other methods. The performance is measured in terms of GFLOPS and images per second. The kernel is optimized for high efficiency, allowing it to scale well on modern GPUs.