1. 받은 값 N을 batch로 하고 기존의 weight들을 transpose해주고 Bias도 구현의 편리성을 위해 Broadcasting을 미리 해놓음. Input 값들에 Batch 차원을 추가해줌. N을 GPU개수로 나누어 각 GPU가 처리해야할 batch수를 계산해 놓음.
2. random_select을 제외한 모든 함수를 kernel 생성을 통해 병렬화 하였음. Softmax에서는 Sum 병렬화를 실패하였으며 matmul은 이전 과제와 같은 방식으로 하고 나머지 단순 연산들을 어렵지 않게 구현. 모든 커널 내에는 Tiling을 진행함.
3. Batch가 N이기 때문에 생성 과정에서 N번 돌던 for문 삭제.
4. 계속되는 Memory 주소 반복 사용을 막기 위해 미리 포인터를 생성해놓고 커널 함수 실행 전 cudaMalloc과 Host to Device cudaMemcpy를 해주고 종료 뒤에는 Device to Host cudaMemcpy. (사실 이 선언을 미리 다 해놓고 싶었는데 미리 해놓으면 자꾸 illegal access오류가 떠서 포기.)
5. 커널 실행을 횟수를 줄이기 위해서 n_gate, h_gate를 만들어서 가능한 연산을 한번에 묶어서 함. r_gate와 z_gate도 생성하였는데 이상하게 이 둘을 사용하면 round-off error가 심하게 나서 뺌.
6. MPI로 각 4개의 노드마다 배치를 나누어 실행하는 병렬화 구현
7. 최종 성능 N = 16382 기준

조교님 감사합니다. 한 학기 수고 많으셨습니다.