Bonus Lecture: The Swiss Army Knife of Tensor Networks: Singular Value Decomposition (SVD)

Hsin-Yuan Huang (Robert)

Caltech

1 Introduction

In the study of many-body quantum systems and other complex structures, we often encounter large tensors. A tensor network is a powerful language for representing and manipulating these tensors. One of the most fundamental operations in any tensor network algorithm is the *Singular Value Decomposition (SVD).* It allows us to decompose a single, large tensor into a network of smaller, more manageable tensors. This is the key to compressing information, finding canonical representations (like in Matrix Product States), and simplifying calculations. This note will walk you through the simple but profound idea of applying SVD to a generic tensor.

2 From Tensor to Matrix: The Reshaping Step

A tensor is a multi-dimensional array of numbers. Let's consider a generic tensor T of rank-k, which means it has k indices (or "legs" in the tensor network language). We write its components as $T_{i_1,i_2,\ldots,i_k}$. This tensor is not necessarily a matrix (since a matrix has $k = 2$).

To use SVD, which is a tool for matrices, we must first *reshape* our tensor into a matrix. We do this by partitioning its indices into two groups.

1. **Choose a partition.** Let's pick a subset of the indices, $S \subset \{1, 2, \ldots, k\}$. The remaining indices form the complement set, S^c.

2. **Combine indices.** We group all the indices corresponding to S into a single "super-index," let's call it a. Similarly, we group the indices in S^c into another super-index, b.

For example, if we have a rank-4 tensor T_{i_1,i_2,i_3,i_4}, we could choose the partition $S = \{1, 3\}$ and $S^c = \{2, 4\}$. Then we define:

$$a = (i_1, i_3) \quad \text{and} \quad b = (i_2, i_4)$$

The tensor T_{i_1,i_2,i_3,i_4} can now be viewed as a matrix A with row index a and column index b:

$$A_{a,b} \equiv T_{i_1,i_3,i_2,i_4}$$

The dimension of the row space is the product of the dimensions of indices i_1 and i_3, and similarly for the column space.

3 Performing the SVD

Now that we have a matrix A, we can perform a standard Singular Value Decomposition. The SVD of A is given by:

$$A = U\Sigma V^\dagger$$

where:

- U is a unitary matrix ($U^\dagger U = I$) whose columns are the left singular vectors.
- Σ is a rectangular diagonal matrix containing the non-negative singular values, σ_λ.
- $V^\dagger$ is a unitary matrix ($V^\dagger V = I$) whose rows are the right singular vectors.

In index notation, the decomposition is written as:

$$A_{a,b} = \sum_\lambda U_{a,\lambda}\Sigma_{\lambda,\lambda}V^\dagger_{\lambda,b}$$

The index λ runs up to the rank of the matrix, which is the number of non-zero singular values.

4 Back to Tensor Networks

The final step is to interpret this matrix decomposition back into the language of tensors. We have expressed our original tensor T as a sum over a new index λ. This is precisely a tensor contraction! We can split the SVD expression into two new tensors. A common choice is to absorb the singular values into the U matrix:

$$A_{a,b} = \sum_\lambda (U_{a,\lambda}\Sigma_\lambda)(V^\dagger_{\lambda,b})$$

We can now define two new tensors, let's call them T_A and T_B:

- $(T_A)_{a,\lambda} = U_{a,\lambda}\Sigma_\lambda$
- $(T_B)_{\lambda,b} = V^\dagger_{\lambda,b}$

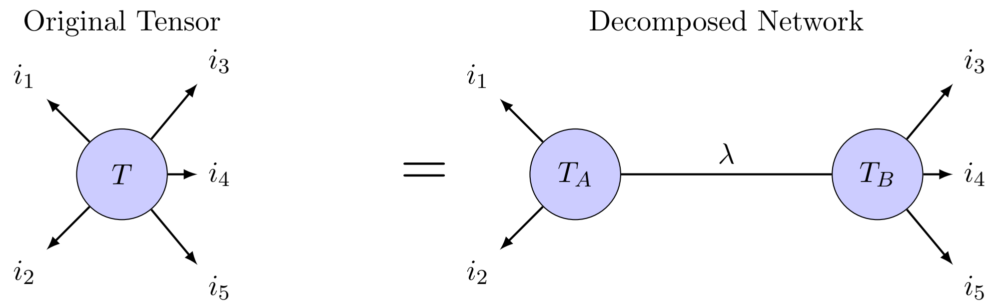
Finally, we "un-group" the super-indices a and b to recover the original legs of our tensor:

$$T_{i_x\in S,\ i_x\notin S} = \sum_\lambda (T_A)_{i_x\in S,\lambda}(T_B)_{\lambda,i_x\notin S}$$

We have successfully decomposed our original rank-k tensor T into two smaller tensors, T_A and T_B, connected by a new internal index (or "bond") λ. The size of this bond, known as the *bond dimension*, is determined by the number of singular values we keep. If we truncate zero (or small and negligible) singular values, this process compresses the tensor.

5 The Diagrammatic Picture

The entire process is beautifully and intuitively captured by a tensor network diagram. Let's visualize the SVD of a rank-5 tensor T_{i_1,i_2,i_3,i_4,i_5} where we partition the indices into $\{i_1, i_2\}$ and $\{i_3, i_4, i_5\}$.



In the diagram:

- The original tensor T is a single node with five legs.
- After SVD, it becomes two nodes, T_A and T_B.
- The external legs are partitioned between the two new tensors according to our choice of S.
- The new internal line connecting them represents the contracted index λ. Its dimension is the bond dimension.

This simple decomposition is the building block for nearly all advanced tensor network algorithms.