4.1 Tensors

A tensor is a multi-dimensional array, which are used in a variety of applications, such as weights and activations in deep neural networks. The order of a tensor (also known as the modes of a tensor) is the number of dimensions $N$ of that tensor. An element $(i, j, k)$ of a third-order tensor $X$ is denoted by $X_{i,j,k}$. Fibers are defined by fixing every index but one; they are a higher-dimensional analogue of matrix rows and columns. Slices are defined by fixing all but two indices, i.e. two-dimensional sections of a tensor. Examples of fibers and slices are seen in figure 4.1. The (Frobenius) norm of a tensor is defined as

$$||X||_F = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} |X_{i_1i_2 \ldots i_N}|^2},$$

where the $j$-th dimension fiber is in $\mathbb{R}^{I_j}$.

![Sample fibers and slices of an order 3 tensor](image)

Figure 4.1: Sample fibers and slices of an order 3 tensor
4.2 Tensor Multiplication

4.2.1 Definition

The $n$-mode (matrix) product of a tensor $A \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ with a matrix $B \in \mathbb{R}^{p \times d_n}$ is done element-wise as below.

$$(A \times_n B)_{i_1, \ldots, i_{n-1}i_n+1 \ldots, i_N} = \sum_{i_n=1}^{d_n} A_{i_1i_2 \ldots i_n \ldots i_N} B_{ji_n}$$

In other words, each mode-$n$ fiber of $A$ is multiplied by the matrix $B$.

4.2.2 Approximate Tensor Multiplication

The algorithm for approximate tensor multiplication is shown in Figure 4.2. The central idea is to reduce the dimensions of the tensor $A$ and matrix $B$ with sampling to get $C$ and $R$, and perform an $n$-mode matrix product with $C$ and $R$ using the classical algorithm. The complexity of this algorithm is $O(d_1 \ldots d_{n-1} md_n \ldots d_N p)$.

![Algorithm 1 Approximate Tensor n-Mode Product via Sampling](image)

Figure 4.2: Algorithm for approximate tensor multiplication

We now look at the mean and variance of the multiplication estimator. Define

$$M_{i,j} \triangleq (A \times_n B)_{i_1, \ldots, i_{n-1}j_{i_{n+1}} \ldots, i_N} = \sum_{i_n=1}^{d_n} A_{i_1i_2 \ldots i_n \ldots i_N} B_{ji_n}$$

and

$$\hat{M}_{i,j} \triangleq \sum_{i_n=1}^{m} \frac{1}{p_{i_n}} A_{i_1i_2 \ldots i_n \ldots i_N} B_{ji_n}.$$ 

This estimator is unbiased, i.e. $\mathbb{E}[\hat{M}_{i,j}] = M_{i,j}$. The variance is

$$\text{Var}[\hat{M}_{i,j}] = \frac{1}{m} \sum_{i_n=1}^{d_n} \frac{1}{p_{i_n}} A_{i_1i_2 \ldots i_n \ldots i_N}^2 B_{ji_n}^2 - \frac{1}{m} (M_{i,j})^2.$$
To achieve the optimal multiplication estimator, we want to solve the following minimization problem.

$$\minimize_p \mathbb{E} ||\hat{M} - M||^2_F = \minimize_p \sum \var[\hat{M}_{ij}].$$

After some math, we find that the optimal $p$ is defined by

$$p_k = \frac{||A...k...||_F||B_k||_F}{\sum_k ||A...k...||_F||B_k||_F}.$$

### 4.3 Verifying Matrix Multiplication

We now consider a different problem. Suppose we are given three $n \times n$ matrices $A, B, M$. We want to verify whether $AB = M$. The naive method is to multiply $A$ and $B$ with the classical method and compare each point in the product and $M$ individually, which is $O(n^3)$. It turns out that a randomized algorithm can do this in $O(n^2)$ and no faster.

The algorithm for this method is known as Frievald’s Algorithm (1977). We first sample a random vector $r = [r_1, ..., r_n]^T$. We compute $Br$, then $A(Br)$. We compute $Mr$. Finally, we compare our two products. If $A(Br) \neq Mr$, then $AB \neq M$ with 100% probability. Otherwise, we return $AB = M$. Since there are three matrix-vector multiplications, we have a complexity of $O(n^2)$.

We would like to analyze the failure probability of this algorithm. Without knowing anything about the matrices $A, B, M$, we can’t guarantee a high or low probability for this algorithm. However, if we pick each $r_i$ in $r = [r_1, ..., r_n]^T$ in an i.i.d. fashion to be $+1$ or $−1$ with probability $\frac{1}{2}$, we can claim $\mathbb{P}[ABr = Mr] \leq \frac{1}{2}$. Note that we can also choose $r_i$ to be 0 or 1. To improve the error probability, we run the algorithm independently $k$ times. If we ever find an $r^k$ such that $ABr^k = Mr^k$, then the algorithm correctly returns $AB \neq M$. If we always find $ABr = Mr$, then the error probability is at most $\frac{1}{2^k}$. For $k = 25$, we have an error probability $\leq 10^{-9}$.

### 4.4 Concentration Bounds

In order to achieve tighter success probabilities, we look at concentration bounds. Specifically for approximate matrix multiplication (AMM), the size of the sample is $m = \frac{1}{\delta \epsilon^2}$. We would like to have $m$ not depend on the failure probability $\delta$.

#### 4.4.1 Specific Bounds

We provide a quick refresher on common bounds. Markov’s Inequality states that for $Z > 0$ and $t > 0$,

$$\mathbb{P}[Z > a] \leq \frac{\mathbb{E}Z}{a}.$$
Chebyshev’s Inequality is as follows. Let \( X \) be a random variable with expectation \( \mathbb{E}[X] \) and variance \( \text{Var}[X] \). Then,
\[
\mathbb{P}[|X - \mathbb{E}[X]| \geq t] \leq \frac{\text{Var}[X]}{t^2}.
\]

Lastly, Chernoff’s Bound has several versions with better constants, but we present this one. Let \( X_1, \ldots, X_m \) be independent random variables \( \in [0, 1] \) and let \( \mu = \mathbb{E}X_1 \). Then
\[
\mathbb{P}[\left| \frac{1}{m} \sum_{i=1}^{m} X_i - \mu \right| > t\mu] \leq 2e^{-m\frac{t^2}{3\mu}}.
\]

We will use this result in the following discussions.

### 4.4.2 Application 1: Monte Carlo Approximations

We look at applications in Monte Carlo Approximations. Suppose we want to estimate \( \pi \). We uniformly sample \( z_1, \ldots, z_m \) i.i.d. from \( [0, 1]^2 \). We define the random variable \( Z_i \) below.
\[
Z_i = \begin{cases} 
1 & \|z_i\|_2 \leq 1 \\
0 & \text{otherwise}
\end{cases}
\]

Thus, \( \mathbb{P}[Z_i = 1] = \frac{\pi}{4} \). Applying the Chernoff Bound, we get
\[
\left| \frac{1}{m} \sum_{i=1}^{m} Z_i - \frac{\pi}{4} \right| \leq \epsilon \frac{\pi}{4}
\]

with probability at least \( 1 - 2e^{-m\epsilon^2/\pi} \). We can pick \( m \geq \frac{12}{\pi^2} \log \frac{2}{\delta} \) and obtain an estimate of \( \hat{\pi} \) such that \( (1 - \epsilon)\pi \leq \hat{\pi} \leq (1 + \epsilon)\pi \) with probability at least \( 1 - \delta \). The range \( [(1 - \epsilon)\pi, (1 + \epsilon)\pi] \) is a confidence interval.

### 4.4.3 Application 2: Amplifying Probability of Success

Now we try to amplify the probability of success of a randomized algorithm. Suppose we have a randomized algorithm which produces an \( \epsilon \) approximation \( |\hat{x} - x^*| \leq \epsilon \) with probability at least 0.9. We repeat the algorithm \( m \) times independently, and take the median of the \( m \) outputs. Note that we take the median instead of the mean, because a failure case could result in very large/small values that shift the mean. Let the random variable \( X_i = 1 \) if the \( i \)-th trial is good, i.e. \( |\hat{x}_i - x^*| \leq \epsilon \). If at least half of the \( X_i \)'s are one, the median of the \( m \) outputs is also good, i.e. \( \left| \text{Median}(\hat{x}_i) - x^* \right| \leq \epsilon \). The Chernoff Bound implies that \( \left| \frac{1}{m} \sum_{i=1}^{m} X_i - 0.9 \right| \leq 0.9t \) with probability \( 1 - e^{-t^2/0.9m/3} \). Pick \( t = 0.4/0.9 \). Then, the median is an \( \epsilon \) approximation with probability at least \( 1 - e^{-0.059m} \), e.g., for \( m = 200 \), failure probability is \( \leq 7 \times 10^{-6} \).
4.4.4 Median for Approximate Matrix Multiplication

Since the Chernoff Bound implies that the majority of estimators are good, we would like to generalize the concept of a median to matrices. The median relies on the fact that $\mathbb{R}^1$ is ordered; however, matrices aren’t ordered. We could represent the median as the optimization problem, $\arg\min_y \sum |x_i - y|$, but solving this for matrices is computationally expensive. The central idea is to have some concept of “centrality”. We look at distances between estimates: the correct estimates will have many smaller distances, while the incorrect ones will have many larger distances.

We start with the AMM final probability bound. For any $\delta > 0$, set $m = \frac{1}{\delta \epsilon^2}$ to obtain

$$
P[\|AB - CR\|_F > \epsilon \|A\|_F \|B\|_F] \leq \delta.
$$

Suppose $\|A\|_F = \|B\|_F = 1$ and let $\epsilon = 0.1, \delta = 0.9$. Repeat the algorithm independently and obtain $C_1R_1, ..., C_tR_t$ in $t$ independent trials. Then, $\|AB - C_iR_i\|_F < 0.1$ with probability 0.9 for each $i$. However, we don’t know which ones are good, i.e. $\|AB - C_iR_i\|_F < 0.1$.

Let $X_i = 1$ if the $i$-th trial is good and $X_i = 0$ otherwise. The Chernoff Bound implies that $\frac{1}{m} \sum_{i=1}^m X_i \geq 0.5$ with probability $1 - e^{-0.059m}$, i.e. at least half of the matrices are good. Compute $\rho_i \triangleq \frac{1}{|\{j \neq i, \|C_iR_i - C_jR_j\|_F \leq 0.2\}|}$.

We now prove this lemma. We use the triangle inequality which states that

$$
\|X + Y\|_F \leq \|X\|_F + \|Y\|_F
$$

and the reverse triangle inequality which states that

$$
\|X + Y\|_F \geq \|X\|_F - \|Y\|_F.
$$

Letting $X = C_iR_i - AB, Y = AB - C_jR_j$, we get

$$
\|C_iR_i - C_jR_j\|_F \leq \|C_iR_i - AB\|_F + \|C_jR_j - AB\|_F
$$

and

$$
\|C_iR_i - C_jR_j\|_F \geq \|C_iR_i - AB\|_F - \|C_jR_j - AB\|_F.
$$

If $C_iR_i$ is good, i.e. $\|AB - C_iR_i\|_F < 0.1$, then it is close to at least half of the other $C_jR_j$’s. Thus, $\rho_i \geq \frac{1}{|\{j \neq i, \|C_iR_i - C_jR_j\|_F \leq 0.2\}|}$ by the triangle inequality. If $C_iR_i$ is bad, i.e. $\|AB - C_iR_i\|_F > 0.3$, then $\|C_iR_i - C_jR_j\|_F \geq 0.2$ by the triangle inequality and $\rho_i \leq \frac{t}{2}$. 

4-5