Fast Matrix Exponentiation

With Applications

Nurseiit Abdimomyn

UNIST

October 25, 2018

Outline

Matrix Multiplication

Consider two matrices:

Matrix Multiplication

Consider two matrices:

- Matrix A is $n * k$ dimensional.

Matrix Multiplication

Consider two matrices:

- Matrix A is $n * k$ dimensional.
- Matrix B is $k * m$ dimensional.

Matrix Multiplication

Consider two matrices:

- Matrix A is $n * k$ dimensional.
- Matrix B is $k * m$ dimensional.

Notice that A's columns and B's rows number are identical!

Matrix Multiplication

Consider two matrices:

- Matrix A is $n * k$ dimensional.
- Matrix B is $k * m$ dimensional.

Notice that A's columns and B's rows number are identical!

Then we define matrix $C = A * B$ as:

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nm} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nk} \end{bmatrix} * \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \dots & \dots & \dots & \dots \\ b_{k1} & b_{k2} & \dots & b_{km} \end{bmatrix}$$

Matrix Multiplication

Consider two matrices:

- Matrix A is $n * k$ dimensional.
- Matrix B is $k * m$ dimensional.

Notice that A's columns and B's rows number are identical!

Then we define matrix $C = A * B$ as:

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nm} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nk} \end{bmatrix} * \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \dots & \dots & \dots & \dots \\ b_{k1} & b_{k2} & \dots & b_{km} \end{bmatrix}$$

Thus, C is an $n * m$ dimensional matrix.

Which is calculated as: $c_{ij} = \sum_{r=1}^{k} a_{ir} * b_{rj}$

Couple of things to notice

- Matrix C has $n * m$ elements and each element is computed in k steps with given formula.

Couple of things to notice

- Matrix C has $n * m$ elements and each element is computed in k steps with given formula.
  Thus, we can obtain C in $O(n * m * k)$, given A and B.

Couple of things to notice

- Matrix C has $n * m$ elements and each element is computed in k steps with given formula.
  Thus, we can obtain C in $O(n * m * k)$, given A and B.
- If $n = m = k$ (i.e. both A and B have n rows and n columns), then C has n rows and n columns, and can be computed in $O(n^3)$.

Some useful properties of matrix multiplication

- It is *not commutative*: $A * B \neq B * A$ in general case.

Some useful properties of matrix multiplication

- It is *not commutative*: $A * B \neq B * A$ in general case.
- It is *associative*:
  $A * B * C = (A * B) * C = A * (B * C)$ in case $A * B * C$ exists;

Some useful properties of matrix multiplication

- It is *not commutative*: $A * B \neq B * A$ in general case.
- It is *associative*:
  $A * B * C = (A * B) * C = A * (B * C)$ in case $A * B * C$ exists;
- If you have a matrix with n rows and n columns,
  then multiplying it by I_n gives the same matrix.
  i. e. $I_n * A = A * I_n = A$.

Some useful properties of matrix multiplication

- It is *not commutative*: $A * B \neq B * A$ in general case.
- It is *associative*:
  $A * B * C = (A * B) * C = A * (B * C)$ in case $A * B * C$ exists;
- If you have a matrix with n rows and n columns,
  then multiplying it by I_n gives the same matrix.
  i. e. $I_n * A = A * I_n = A$.
  Where I_n is a matrix with n rows and n columns of such form:

$$I_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Outline

Matrix exponentiation

Suppose we need to find some $n * n$ dimensional square matrix A to the power p or A^p

Matrix exponentiation

Suppose we need to find some $n * n$ dimensional square matrix A to the power p or A^p

We can do so via:

```
function matpow_naive(A, p):
  result = I_n
  for i = 1..p:
    result = result * A
  return result
```

Which will run in $O(n^3 * p)$

Fast Matrix exponentiation

Can we do it any faster?

Fast Matrix exponentiation

Can we do it any faster?
Yes, we can apply the *BinPower* algorithm here:

```
function matBinPow(A, p):
  result = I_n
  while p > 0:
    if p % 2 == 1:
      result = result * A
    A = A * A
    p = p / 2
  return result
```

Which will run in $O(n^3 * \log p)$

Outline

Finding Nth Fibonacci number

Fibonacci numbers, F_n are defined as:

- $F_0 = F_1 = 1$
- $F_i = F_{i-1} + F_{i-2}$ for $i > 1$.

Finding Nth Fibonacci number

Fibonacci numbers, F_n are defined as:

- $F_0 = F_1 = 1$
- $F_i = F_{i-1} + F_{i-2}$ for $i > 1$.

We want to calculate $F_n \ mod \ M$, where $n < 10^{18}$ and $M = 10^9 + 7$.

Finding Nth Fibonacci number

Suppose we have a vector (matrix with one row and several columns) of (F_{i-2}, F_{i-1}) and we want to multiply it by some matrix M, so that we get (F_{i-1}, F_i) as a result.

Thank you!