Multiplying Matrices Without Multiplying

Пантелеев Даниил, БПМИ 192

Постановка задачи

$A \in \mathbb{R}^{N \times D}, B \in \mathbb{R}^{D \times M}, N \gg D \geq M$

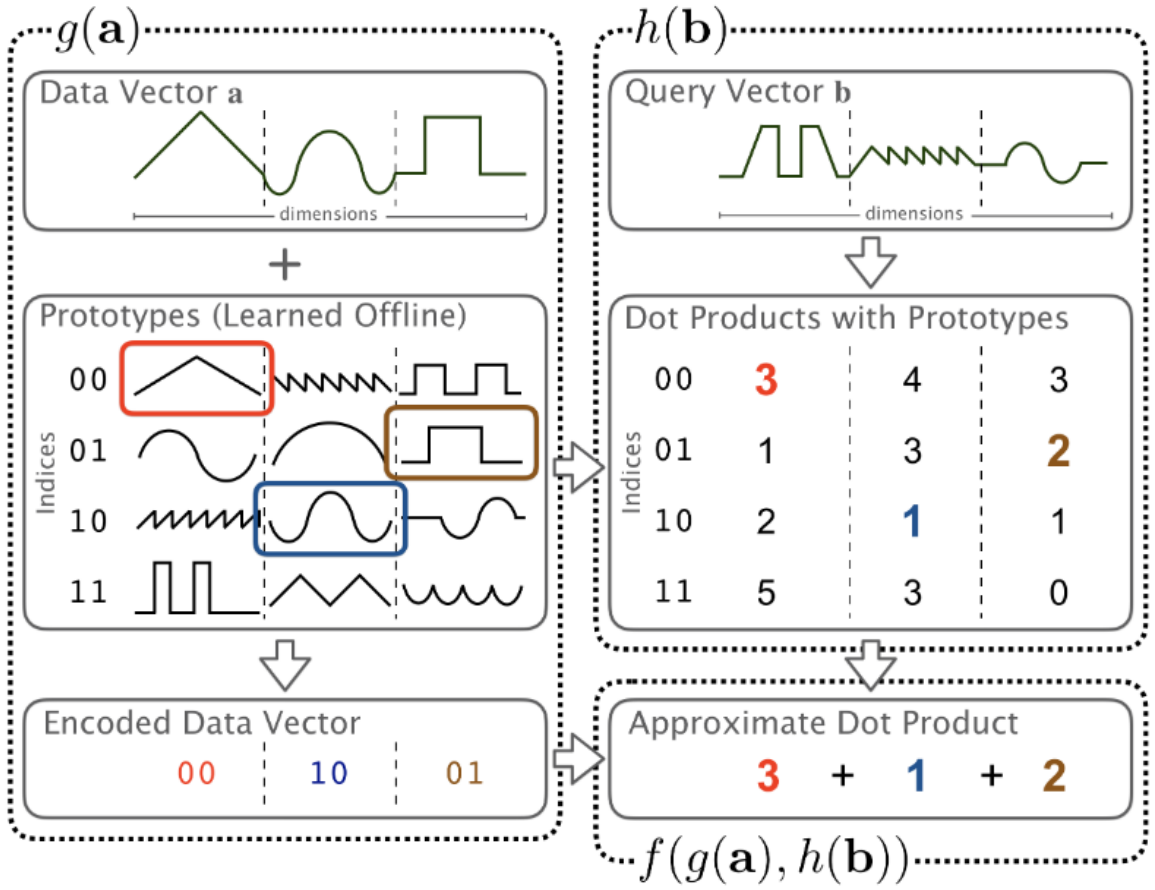
Найти функции $g(*), h(*), f(*)$ такие, что

$$\|\alpha f(g(A), h(B)) + \beta - AB\|_F < \varepsilon(\tau)\|AB\|_F$$

выполнено для как можно меньшего $\varepsilon(\tau)$

Product Quantization

1. Обучение прототипа
2. Хеш-функция, $g(a)$
3. Создание таблицы, $h(B)$
4. Аггрегация, $f(*,*)$



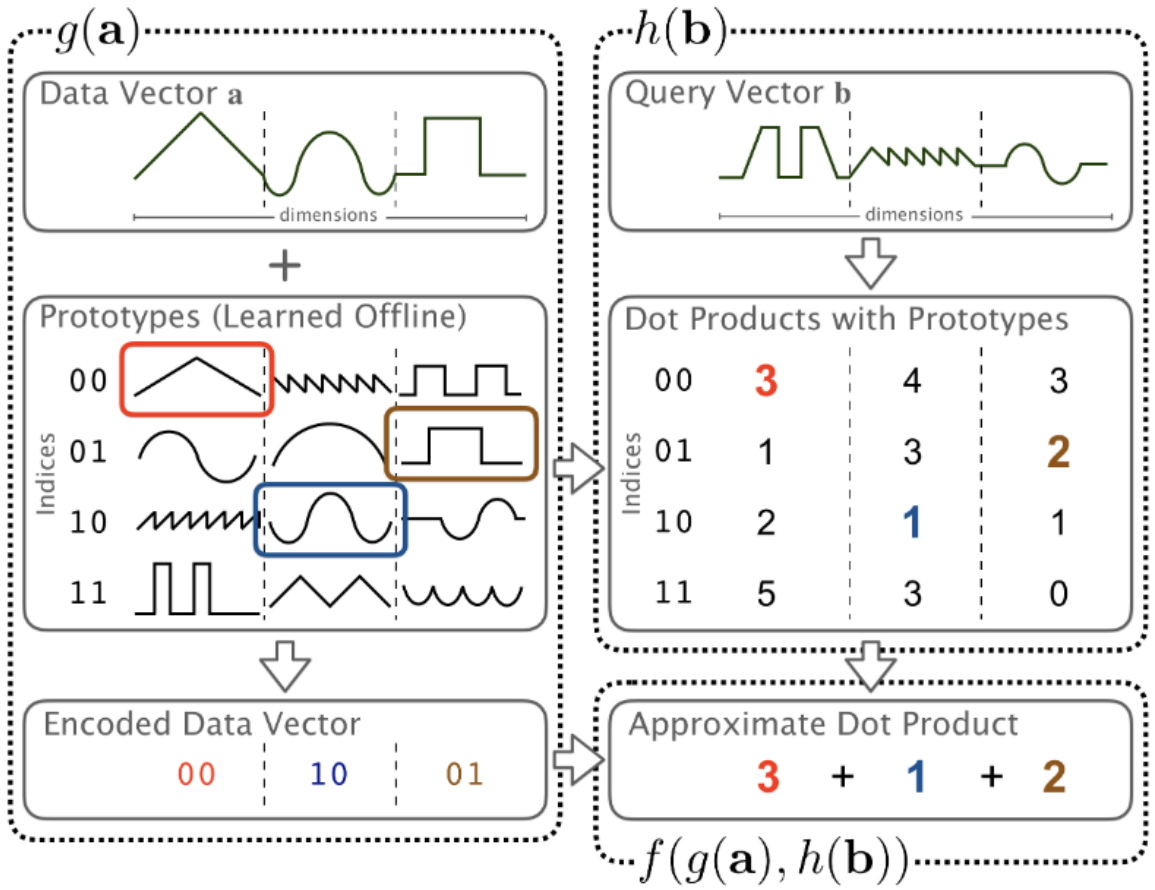
Product Quantization

Важно:

PQ дает хороший прирост производительности только при

$N, M \gg D$, а у нас

$N \gg D \geq M$



Multiplying Matrices Without Multiplying

Матрица $\tilde{A}$

Строки $\tilde{A}$ и A из одного распределения

Хотим:

придумать такую функцию $g(a)$, чтобы максимально оптимизировать PQ,
при этом используя тренировочную матрицу $\tilde{A}$

Multiplying Matrices Without Multiplying

Идея:

Обучим двоичное дерево на $\widetilde{A}$, чтобы оно хешировало векторы и распределяло их по корзинам $\mathcal{B}_j^i$

Здесь i - номер уровня вершины, а j - номер вершины

Multiplying Matrices Without Multiplying

Алгоритм хэширования: (MADDNESSHASH)

На вход получаем вектор x, индексы $j^1, \ldots, j^4$, пороги $v^1, \ldots, v^4$

Для каждого t от 1 до 4:

1. Сравниваем j^t-ый элемент вектора x с порогом v^t
2. В зависимости от результата определяем вектор в правого или левого сына вершины

Multiplying Matrices Without Multiplying

Добавление нового уровня t в дерево:

Для каждой из корзин $\mathcal{B}_1^{t-1} \dots \mathcal{B}_{2^{t-1}}^{t-1}$:

1. Выбираем индексы разбиений с помощью эвристики
2. Выбираем оптимальный (с точки зрения суммы по всем корзинам) порог
3. Получаем новые корзины с помощью новых порогов

Multiplying Matrices Without Multiplying

Бонус №1: оптимизация g(b)

$P \in \mathbb{R}^{KC \times D}$ - матрица с диагональными блоками, состоящими из K предобученных образцов

$\tilde{A} \approx GP$, где G - матрица, помогающая выбрать нужный образец

Тогда P можно найти, решив задачу наименьших квадратов:

$$P \approx (G^T + \lambda I)^{-1} G^T \tilde{A}$$

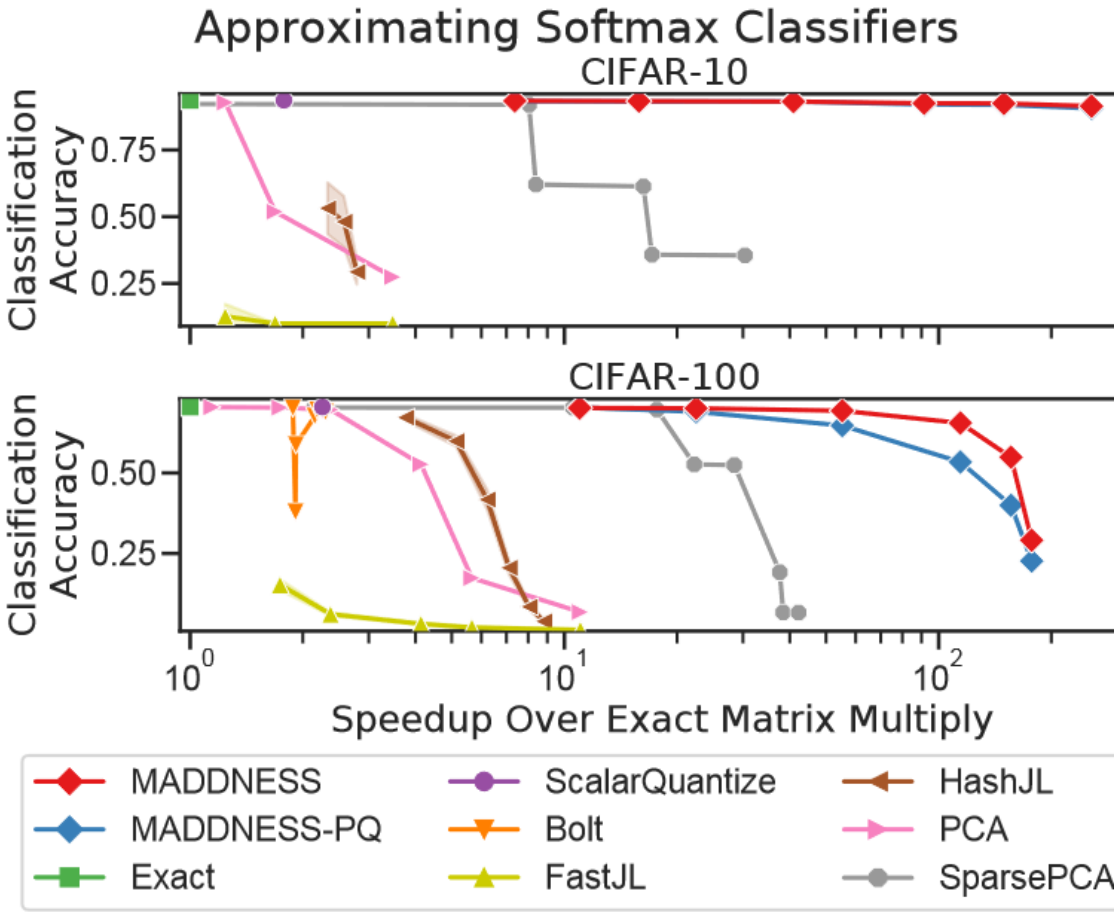
Multiplying Matrices Without Multiplying

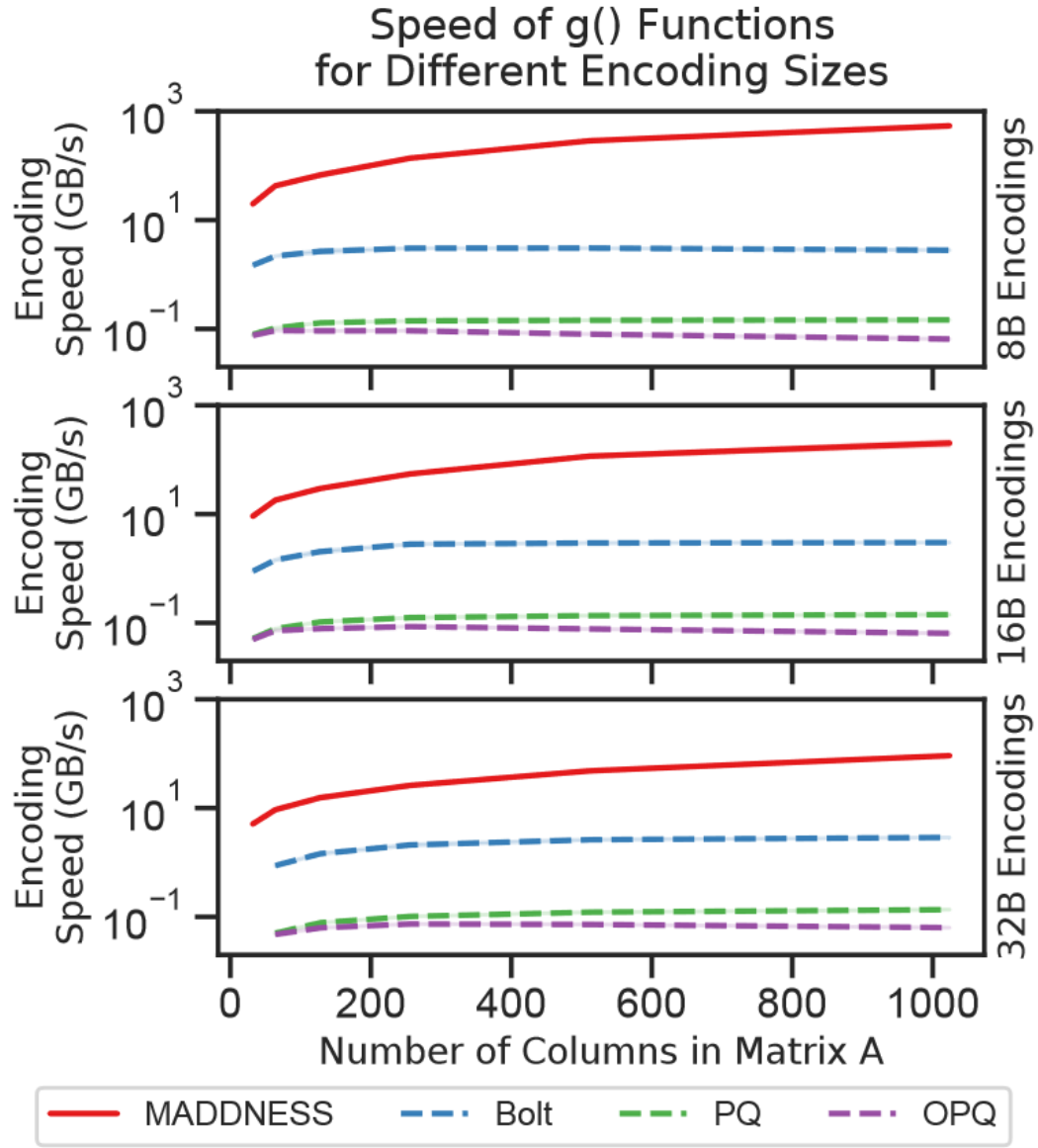
Бонус №2: оптимизация f(∗, ∗)

$T \in \mathbb{R}^{M \times C \times K}$ - тензор таблиц с образцами

Тогда $f(g, h) := \sum_{c=1}^{C} T_{m,c,k}, k = g^{(c)}(a_n)$

В оригинальной реализации для вычисления суммы используется инструкция суммы, а здесь мы используем функцию усреднения `vpavgb`, считая $\frac{(a+b+1)}{2}$ для каждой пары, потом пары пар, и т.д.


Approximating Softmax Classifiers
CIFAR-10
CIFAR-100
Classification Accuracy
0.75
0.50
0.25
10^0
10^1
10^2
Speedup Over Exact Matrix Multiply
MADDNESS
MADDNESS-PQ
Exact
ScalarQuantize
Bolt
FastJL
HashJL
PCA
SparsePCA


Speed of g() Functions
for Different Encoding Sizes
Encoding Speed (GB/s)
10^3
10^1
10^{-1}
8B Encodings
16B Encodings
32B Encodings
0
200
400
600
800
1000
Number of Columns in Matrix A
MADDNESS
Bolt
PQ
OPQ