Matrix-vector multiplication in terms of dot-products

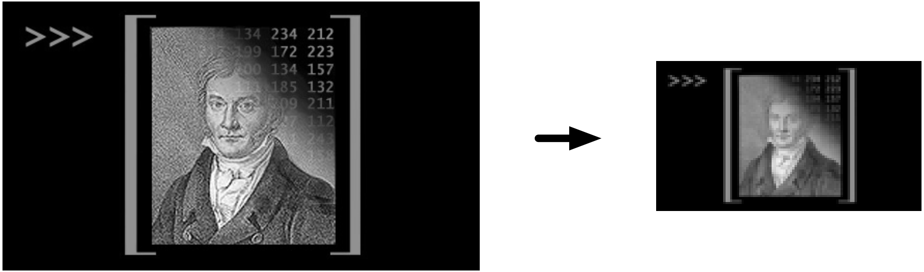
Let M be an $R \times C$ matrix.

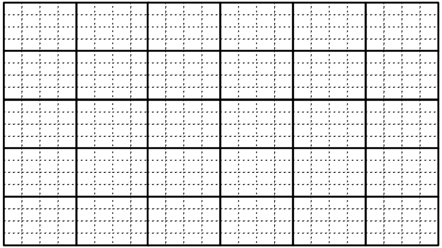
Dot-Product Definition of matrix-vector multiplication: $M * \mathbf{u}$ is the R-vector $\mathbf{v}$ such that $\mathbf{v}[r]$ is the dot-product of row r of M with $\mathbf{u}$.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 10 & 0 \end{bmatrix} * [3, -1] = [\ [1, 2] \cdot [3, -1],\ [3, 4] \cdot [3, -1],\ [10, 0] \cdot [3, -1]\ ]$$

$$= [1, 5, 30]$$

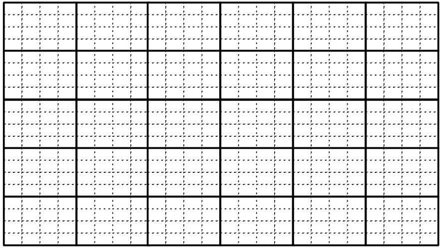
Applications of dot-product definition of matrix-vector multiplication: Downsampling





- Each pixel of the low-res image corresponds to a little grid of pixels of the high-res image.
- The intensity value of a low-res pixel is the *average* of the intensity values of the corresponding high-res pixels.

Applications of dot-product definition of matrix-vector multiplication: Downsampling



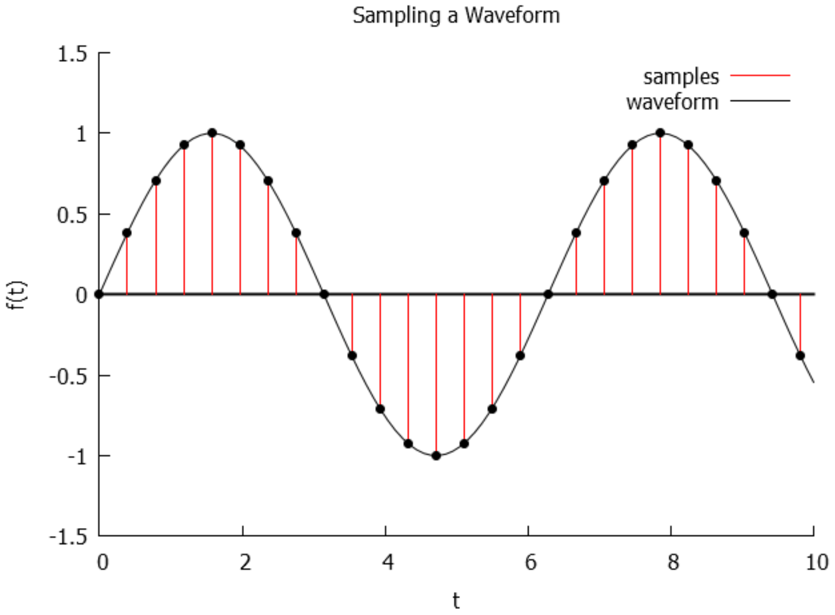
- Each pixel of the low-res image corresponds to a little grid of pixels of the high-res image.
- The intensity value of a low-res pixel is the *average* of the intensity values of the corresponding high-res pixels.

- Averaging can be expressed as dot-product.
- We want to compute a dot-product for each low-res pixel.
- Can be expressed as matrix-vector multiplication.

Applications of dot-product definition of matrix-vector multiplication: blurring

- To blur a face, replace each pixel in face with average of pixel intensities in its neighborhood.
- Average can be expressed as dot-product.
- By dot-product definition of matrix-vector multiplication, can express this image transformation as a matrix-vector product.
- Gaussian blur: a kind of weighted average

Applications of dot-product definition of matrix-vector multiplication: Audio search



Applications of dot-product definition of matrix-vector multiplication: Audio search

Lots of dot-products!

5	-6	9	-9	-5	-9	-5	5	-8	-5	-9	9	8	-5	-9	6	-2	-4	-9	-1	-1	-9	-3
2	7	4	-3	0	-1	-6	4	5	-8	-9												

5	-6	9	-9	-5	-9	-5	5	-8	-5	-9	9	8	-5	-9	6	-2	-4	-9	-1	-1	-9	-3
	2	7	4	-3	0	-1	-6	4	5	-8	-9											

5	-6	9	-9	-5	-9	-5	5	-8	-5	-9	9	8	-5	-9	6	-2	-4	-9	-1	-1	-9	-3
		2	7	4	-3	0	-1	-6	4	5	-8	-9										

5	-6	9	-9	-5	-9	-5	5	-8	-5	-9	9	8	-5	-9	6	-2	-4	-9	-1	-1	-9	-3
			2	7	4	-3	0	-1	-6	4	5	-8	-9									

5	-6	9	-9	-5	-9	-5	5	-8	-5	-9	9	8	-5	-9	6	-2	-4	-9	-1	-1	-9	-3
				2	7	4	-3	0	-1	-6	4	5	-8	-9								

5	-6	9	-9	-5	-9	-5	5	-8	-5	-9	9	8	-5	-9	6	-2	-4	-9	-1	-1	-9	-3
					2	7	4	-3	0	-1	-6	4	5	-8	-9							

5	-6	9	-9	-5	-9	-5	5	-8	-5	-9	9	8	-5	-9	6	-2	-4	-9	-1	-1	-9	-3
						2	7	4	-3	0	-1	-6	4	5	-8	-9						

Applications of dot-product definition of matrix-vector multiplication: Audio search

Lots of dot-products!

- Represent as a matrix-vector product.
- One row per dot-product.

To search for $[0, 1, -1]$ in $[0, 0, -1, 2, 3, -1, 0, 1, -1, -1]$:

$$\begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 2 \\ -1 & 2 & 3 \\ 2 & 3 & -1 \\ 3 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix} \quad * \quad [0, 1, -1]$$

Formulating a system of linear equations as a matrix-vector equation

Recall the *sensor node* problem:

- In each of several test periods, measure total power consumed:

$$\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$$

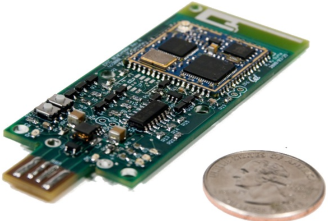
- For each test period, have a vector specifying how long each hardware component was operating during that period:

$$\mathbf{duration}_1, \mathbf{duration}_2, \mathbf{duration}_3, \mathbf{duration}_4, \mathbf{duration}_5$$

- Use measurements to calculate energy consumed per second by each hardware component.

  Formulate as system of linear equations

$$\begin{aligned}
\mathbf{duration}_1 \cdot \mathbf{x} &= \beta_1 \\
\mathbf{duration}_2 \cdot \mathbf{x} &= \beta_2 \\
\mathbf{duration}_3 \cdot \mathbf{x} &= \beta_3 \\
\mathbf{duration}_4 \cdot \mathbf{x} &= \beta_4 \\
\mathbf{duration}_5 \cdot \mathbf{x} &= \beta_5
\end{aligned}$$



Formulating a system of linear equations as a matrix-vector equation

Linear equations

$$\begin{aligned} \mathbf{a}_1 \cdot \mathbf{x} &= \beta_1 \\ \mathbf{a}_2 \cdot \mathbf{x} &= \beta_2 \\ &\vdots \\ \mathbf{a}_m \cdot \mathbf{x} &= \beta_m \end{aligned}$$

Each equation specifies the value of a dot-product.

Rewrite as

$$\begin{bmatrix} \mathbf{a}_1 \\ \hline \mathbf{a}_2 \\ \hline \vdots \\ \hline \mathbf{a}_m \end{bmatrix} * \mathbf{x} = [\beta_1, \beta_2, \ldots, \beta_m]$$

Matrix-vector equation for sensor node

Define `D = {'radio', 'sensor', 'memory', 'CPU'}`.

Goal: Compute a `D`-vector $\mathbf{u}$ that, for each hardware component, gives the current drawn by that component.

Four test periods:

- total milliampere-seconds in these test periods $\mathbf{b} = [140, 170, 60, 170]$
- for each test period, vector specifying how long each hardware device was operating:
  - $\mathbf{duration}_1$ = `Vec(D, 'radio':.1, 'CPU':.3)`
  - $\mathbf{duration}_2$ = `Vec(D, 'sensor':.2, 'CPU':.4)`
  - $\mathbf{duration}_3$ = `Vec(D, 'memory':.3, 'CPU':.1)`
  - $\mathbf{duration}_4$ = `Vec(D, 'memory':.5, 'CPU':.4)`

To get $\mathbf{u}$, solve $A * \mathbf{x} = \mathbf{b}$ where $A = \left[\begin{array}{c} \mathbf{duration}_1 \\ \hline \mathbf{duration}_2 \\ \hline \mathbf{duration}_3 \\ \hline \mathbf{duration}_4 \end{array}\right]$

Triangular matrix

Recall: We considered *triangular* linear systems, e.g.

$$\begin{array}{rcl}
[\ 1,\ 0.5,\ -2,\ 4\ ] \cdot \mathbf{x} & = & -8 \\
[\ 0,\ 3,\ 3,\ 2\ ] \cdot \mathbf{x} & = & 3 \\
[\ 0,\ 0,\ 1,\ 5\ ] \cdot \mathbf{x} & = & -4 \\
[\ 0,\ 0,\ 0,\ 2\ ] \cdot \mathbf{x} & = & 6 \\
[\ 0,\ 0,\ 0,\ 2\ ] \cdot \mathbf{x} & = & 6
\end{array}$$

We can rewrite this linear system as a matrix-vector equation:

$$\begin{bmatrix} 1 & 0.5 & -2 & 4 \\ 0 & 3 & 3 & 2 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 2 \end{bmatrix} * \mathbf{x} = [-8, 3, -4, 6]$$

The matrix is a *triangular* matrix.

Definition: An $n \times n$ *upper triangular* matrix A is a matrix with the property that $A_{ij} = 0$ for $i > j$. Note that the entries forming the upper triangle can be be zero or nonzero.

We can use backward substitution to solve such a matrix-vector equation.

Triangular matrices will play an important role later.

Computing sparse matrix-vector product

To compute matrix-vector or vector-matrix product,

- could use dot-product or linear-combinations definition.
  (You'll do that in homework.)

- However, using those definitions, it's not easy to exploit sparsity in the matrix.

"Ordinary" Definition of Matrix-Vector Multiplication: If M is an $R \times C$ matrix and $\mathbf{u}$ is a C-vector then $M * \mathbf{u}$ is the R-vector $\mathbf{v}$ such that, for each $r \in R$,

$$v[r] = \sum_{c \in C} M[r, c]u[c]$$

Computing sparse matrix-vector product

"Ordinary" Definition of Matrix-Vector Multiplication: If M is an $R \times C$ matrix and $\mathbf{u}$ is a C-vector then $M * \mathbf{u}$ is the R-vector $\mathbf{v}$ such that, for each $r \in R$,

$$v[r] = \sum_{c \in C} M[r, c]u[c]$$

Obvious method:

for i in R:
    $v[i] := \sum_{j \in C} M[i, j]u[j]$

But this doesn't exploit sparsity!

Idea:

- Initialize output vector $\mathbf{v}$ to zero vector.
- Iterate over nonzero entries of M, adding terms according to ordinary definition.

initialize $\mathbf{v}$ to zero vector
for each pair (i, j) in sparse representation,
    $v[i] = v[i] + M[i, j]u[j]$

Algebraic properties of matrix-vector multiplication

Proposition: Let A be an $R \times C$ matrix.

- For any C-vector $\mathbf{v}$ and any scalar α,

$$A * (\alpha\, \mathbf{v}) = \alpha\, (A * \mathbf{v})$$

- For any C-vectors $\mathbf{u}$ and $\mathbf{v}$,

$$A * (\mathbf{u} + \mathbf{v}) = A * \mathbf{u} + A * \mathbf{v}$$

Algebraic properties of matrix-vector multiplication

To prove

$$A * (\alpha\, \mathbf{v}) = \alpha\, (A * \mathbf{v})$$

we need to show corresponding entries are equal:

Need to show

$$\text{entry } i \text{ of} \quad A * (\alpha\, \mathbf{v}) = \text{entry } i \text{ of} \quad \alpha\, (A * \mathbf{v})$$

Proof: Write $A = \begin{bmatrix} \mathbf{a}_1 \\ \hline \vdots \\ \hline \mathbf{a}_m \end{bmatrix}$.

By dot-product def. of matrix-vector mult,

$$\begin{aligned} \text{entry } i \text{ of} \quad A * (\alpha\, \mathbf{v}) &= \mathbf{a}_i \cdot \alpha\, \mathbf{v} \\ &= \alpha\, (\mathbf{a}_i \cdot \mathbf{v}) \end{aligned}$$

by homogeneity of dot-product

By definition of scalar-vector multiply,

$$\begin{aligned} \text{entry } i \text{ of } \alpha\, (A * \mathbf{v}) &= \alpha\, (\text{entry } i \text{ of } A * \mathbf{v}) \\ &= \alpha\, (\mathbf{a}_i \cdot \mathbf{v}) \end{aligned}$$

by dot-product definition of matrix-vector multiply

QED

Algebraic properties of matrix-vector multiplication

To prove

$$A * (\mathbf{u} + \mathbf{v}) = A * \mathbf{u} + A * \mathbf{v}$$

we need to show corresponding entries are equal:

Need to show

$$\text{entry } i \text{ of} \quad A * (\mathbf{u} + \mathbf{v}) = \text{entry } i \text{ of} \quad A * \mathbf{u} + A * \mathbf{v}$$

Proof: Write $A = \begin{bmatrix} \mathbf{a}_1 \\ \hline \vdots \\ \hline \mathbf{a}_m \end{bmatrix}$.

By dot-product def. of matrix-vector mult,

$$\begin{aligned} \text{entry } i \text{ of} \quad A * (\mathbf{u} + \mathbf{v}) &= \mathbf{a}_i \cdot (\mathbf{u} + \mathbf{v}) \\ &= \mathbf{a}_i \cdot \mathbf{u} + \mathbf{a}_i \cdot \mathbf{v} \end{aligned}$$

by distributive property of dot-product

By dot-product def. of matrix-vector mult,

$$\begin{aligned} \text{entry } i \text{ of} \quad A * \mathbf{u} &= \mathbf{a}_i \cdot \mathbf{u} \\ \text{entry } i \text{ of} \quad A * \mathbf{v} &= \mathbf{a}_i \cdot \mathbf{v} \end{aligned}$$

so

$$\text{entry } i \text{ of} \quad A * \mathbf{u} + A * \mathbf{v} = \mathbf{a}_i \cdot \mathbf{u} + \mathbf{a}_i \cdot \mathbf{v}$$

QED