CS 453X: Class 3

Jacob Whitehill

Linear algebra

- A **column vector** is a (n x 1) matrix.
- A **row vector** is a (1 x n) matrix.
- The **transpose** of (n x k) matrix **A**, denoted $\mathbf{A}^T$, is (k x n).
- Multiplication of matrices **A** and **B**:
  - Only possible when: **A** is (n x k) and **B** is (k x m)
  - Result: (n x m)
- The **inner product** between two column vectors (same length) **x**, **y** can be written as: $\mathbf{x}^T\mathbf{y}$
- The **Hadamard** (element-wise) **product** between two matrices **A** and **B** is written as $\mathbf{A} \odot \mathbf{B}$.

Weakness of our feature set

- So far, the feature we have considered are very weak:
    - Is pixel (r_1,c_1) brighter than pixel (r_2,c_2)?
- We can't even express simple relationships such as:
    - "(r_1,c_1) is at least 5 bigger than (r_2,c_2)"
    - "2 times (r_1,c_1) is bigger than (r_2,c_2)"
    - "2 times (r_1,c_1) plus 4 times (r_2,c_2) is larger than (r_3,c_3)".

Linear regression

- We can harness these more complex relationships using **linear regression**.
- Let’s switch back to the age estimation problem…

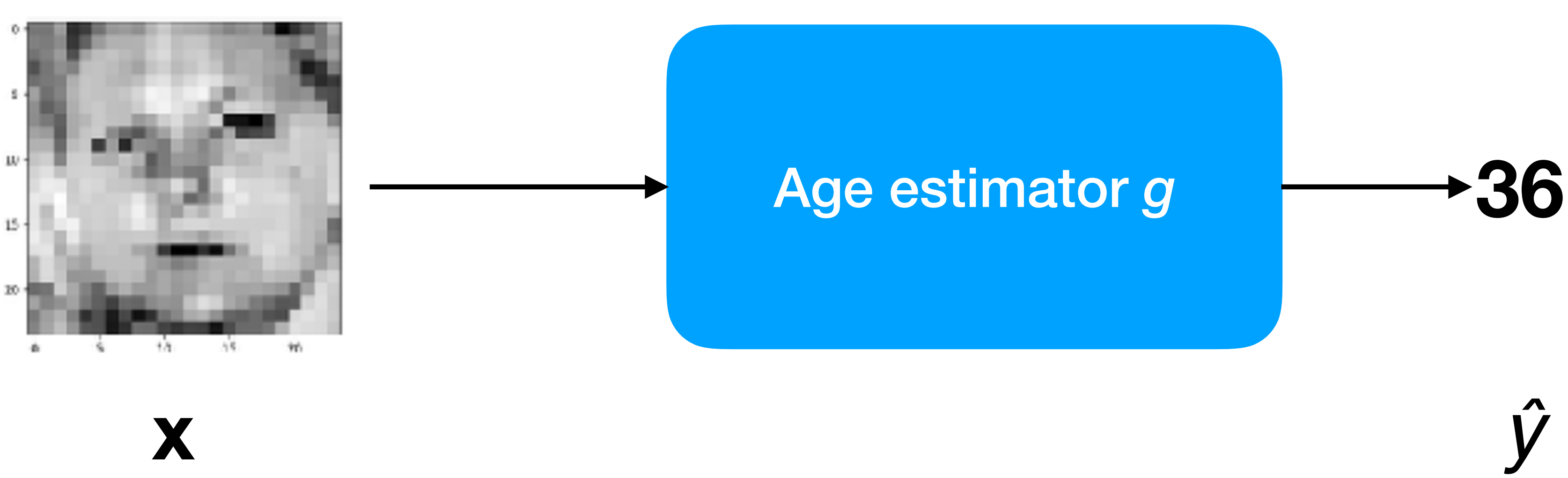
Linear regression

- Linear regression is built as a linear combination of all the inputs **x**:

image pixels

$$\hat{y} = g(\mathbf{x}; \mathbf{w}) = \sum_{j=1}^{m} \mathbf{x}_j \mathbf{w}_j = \mathbf{x}^\top \mathbf{w}$$

- Here, we treat the image **x** as a *vector* (even though it represents a 2-d image).



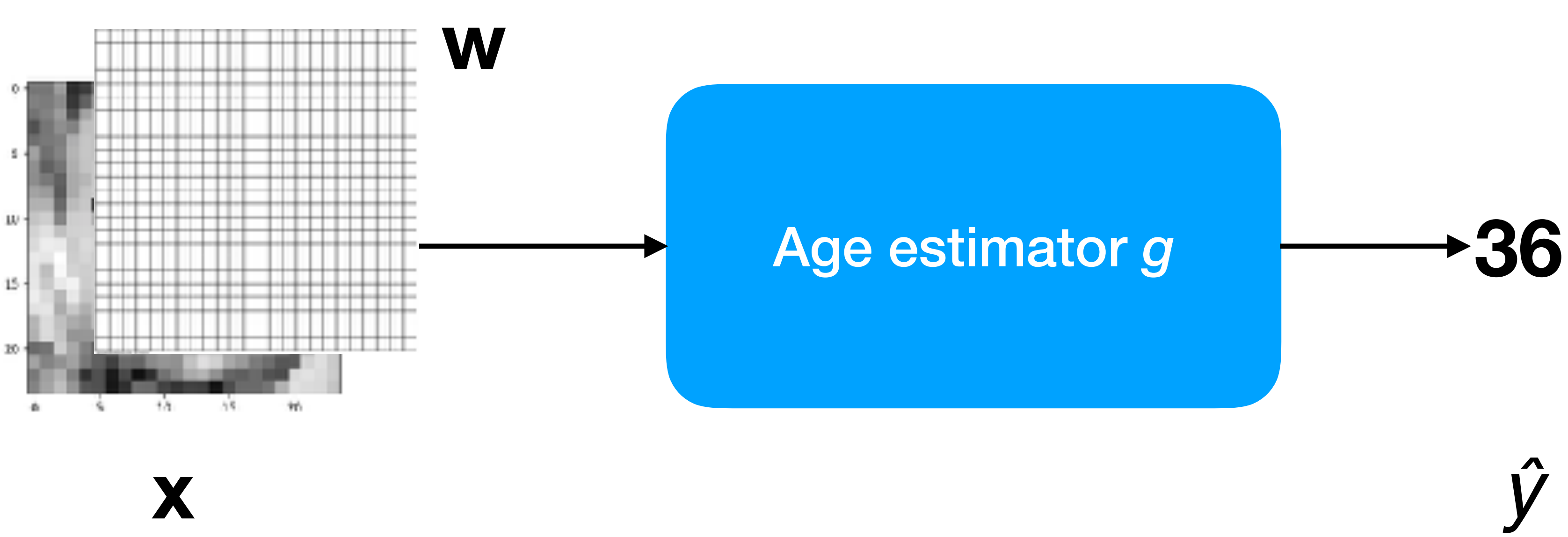
Linear regression

- Linear regression is built as a linear combination of all the inputs **x**:

image pixels

$$\hat{y} = g(\mathbf{x}; \mathbf{w}) = \sum_{j=1}^{m} \mathbf{x}_j \mathbf{w}_j = \mathbf{x}^\top \mathbf{w}$$

- Vector **w** represent an "overlay image" that weights the different pixel intensities of **x**.



Linear regression

- Image a 2x2 pixel “image” **x** and a weight matrix **w**:

2	5
0	3

x

1	3
2	4

w

- Then $\hat{y}$ = 2*1 + 5*3 + 0*2 + 3*4 = 22

Linear regression

- How should we choose each "weight" $\mathbf{w}_j$?
- Let's define the **loss** function that we seek to minimize:

$$
\begin{aligned}
f_{\mathrm{MSE}}(\mathbf{y}, \hat{\mathbf{y}}; \mathbf{w}) &= \frac{1}{2n} \sum_{i=1}^{n} \left(g(\mathbf{x}^{(i)}; \mathbf{w}) - y^{(i)} \right)^2 \\
&= \frac{1}{2n} \sum_{i=1}^{n} \left(\mathbf{x}^{(i)\top} \mathbf{w} - y^{(i)} \right)^2
\end{aligned}
$$

The 2 in the denominator will
slightly simplify the algebra later…

Linear regression

- **w** is an unconstrained real-valued vector; hence, we can use differential calculus to find the minimum of f_{MSE}.
- Just derive the gradient of f_{MSE} w.r.t. **w**, set to 0, and solve.
- Since f_{MSE} is a convex function, we are guaranteed that this critical point is a global minimum.

Matrix/vector calculus

- For a real-valued function $f : \mathbb{R}^m \rightarrow \mathbb{R}$, we define the gradient w.r.t. **w** as:

$$\nabla_{\mathbf{w}} f = \begin{bmatrix} \frac{\partial f}{\partial \mathbf{w}_1} \\ \vdots \\ \frac{\partial f}{\partial \mathbf{w}_m} \end{bmatrix}$$

- In other words, the gradient is a column vector containing all first partial derivatives w.r.t. **w**.

Solving for w

- The gradient of f_{MSE} is thus:

$$\begin{aligned}
\nabla_{\mathbf{w}} f_{\text{MSE}}(\mathbf{y}, \hat{\mathbf{y}}; \mathbf{w}) &= \nabla_{\mathbf{w}} \left[\frac{1}{2n} \sum_{i=1}^{n} \left({\mathbf{x}^{(i)}}^\top \mathbf{w} - y^{(i)} \right)^2 \right] \\
&= \frac{1}{2n} \sum_{i=1}^{n} \nabla_{\mathbf{w}} \left[\left({\mathbf{x}^{(i)}}^\top \mathbf{w} - y^{(i)} \right)^2 \right] \\
&= \quad ?
\end{aligned}$$

1. $$\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}^{(i)} \left({\mathbf{x}^{(i)}}^\top \mathbf{w} - y^{(i)} \right)$$

2. $$\frac{1}{n} \sum_{i=1}^{n} \left({\mathbf{x}^{(i)}}^\top \mathbf{w} - y^{(i)} \right) \mathbf{x}^{(i)}$$

3. $$\frac{1}{n} \sum_{i=1}^{n} \left({\mathbf{x}^{(i)}}^\top \mathbf{w} \mathbf{x}^{(i)} - \mathbf{x}^{(i)} y^{(i)} \right)$$

4. $$\frac{1}{n} \sum_{i=1}^{n} \left({\mathbf{x}^{(i)}}^\top \mathbf{w} - y^{(i)} \right) {\mathbf{x}^{(i)}}^\top$$

Solving for w

- The gradient of f_{MSE} is thus:

$$
\begin{aligned}
\nabla_{\mathbf{w}} f_{\mathrm{MSE}}(\mathbf{y}, \hat{\mathbf{y}}; \mathbf{w}) &= \nabla_{\mathbf{w}} \left[\frac{1}{2n} \sum_{i=1}^{n} \left({\mathbf{x}^{(i)}}^\top \mathbf{w} - y^{(i)} \right)^2 \right] \\
&= \frac{1}{2n} \sum_{i=1}^{n} \nabla_{\mathbf{w}} \left[\left({\mathbf{x}^{(i)}}^\top \mathbf{w} - y^{(i)} \right)^2 \right] \\
&= \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}^{(i)} \left({\mathbf{x}^{(i)}}^\top \mathbf{w} - y^{(i)} \right)
\end{aligned}
$$

1. $\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}^{(i)} \left({\mathbf{x}^{(i)}}^\top \mathbf{w} - y^{(i)} \right)$ **Correct**

Solving for **w**

- By setting to 0, splitting the sum apart, and solving, we reach the solution:

Solving for $\mathbf{w}$

- By setting to 0, splitting the sum apart, and solving, we reach the solution:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}\left(\mathbf{x}^{(i)\top}\mathbf{w}-y^{(i)}\right)$$

$$\begin{aligned}
0 &= \sum_i \mathbf{x}^{(i)}\mathbf{x}^{(i)\top}\mathbf{w} - \sum_i \mathbf{x}^{(i)}y^{(i)} \\
\sum_i \mathbf{x}^{(i)}\mathbf{x}^{(i)\top}\mathbf{w} &= \sum_i \mathbf{x}^{(i)}y^{(i)} \\
\mathbf{w} &= \left(\sum_i \mathbf{x}^{(i)}\mathbf{x}^{(i)\top}\right)^{-1}\sum_i \mathbf{x}^{(i)}y^{(i)}
\end{aligned}$$

Matrix notation

- Let's define a matrix **X** to contain all the training images:

$$\mathbf{X} = \begin{bmatrix} | & & | \\ \mathbf{x}^{(1)} & \dots & \mathbf{x}^{(n)} \\ | & & | \end{bmatrix}$$

  - In statistics, **X** is called the **design matrix.**

- Let's define vector **y** to contain all the training labels:

$$\mathbf{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

Matrix notation

- Using summation notation, we derived:

$$\mathbf{w} = \left(\sum_{i=1}^{n} \mathbf{x}^{(i)} {\mathbf{x}^{(i)}}^\top\right)^{-1} \left(\sum_{i=1}^{n} \mathbf{x}^{(i)} y^{(i)}\right)$$

- Using matrix notation, we can write the solution as:

where $\mathbf{X} = \begin{bmatrix} | & & | \\ \mathbf{x}^{(1)} & \dots & \mathbf{x}^{(n)} \\ | & & | \end{bmatrix}$

1. $\mathbf{w} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$
2. $\mathbf{w} = (\mathbf{X} \odot \mathbf{X})^{-1} \odot \mathbf{X}^\top \mathbf{y}$
3. $\mathbf{w} = \left(\mathbf{X}\mathbf{X}^\top\right)^{-1} \mathbf{X}\mathbf{y}$

Matrix notation

- The solution for the optimal **w** can be re-written as:

$$\mathbf{w} = \left(\mathbf{X}\mathbf{X}^\top\right)^{-1}\mathbf{X}\mathbf{y}$$

- To compute this, do **not** use np.linalg.inv.
- Instead, use np.linalg.solve, which avoids explicitly computing the matrix inverse.
- Show demo.