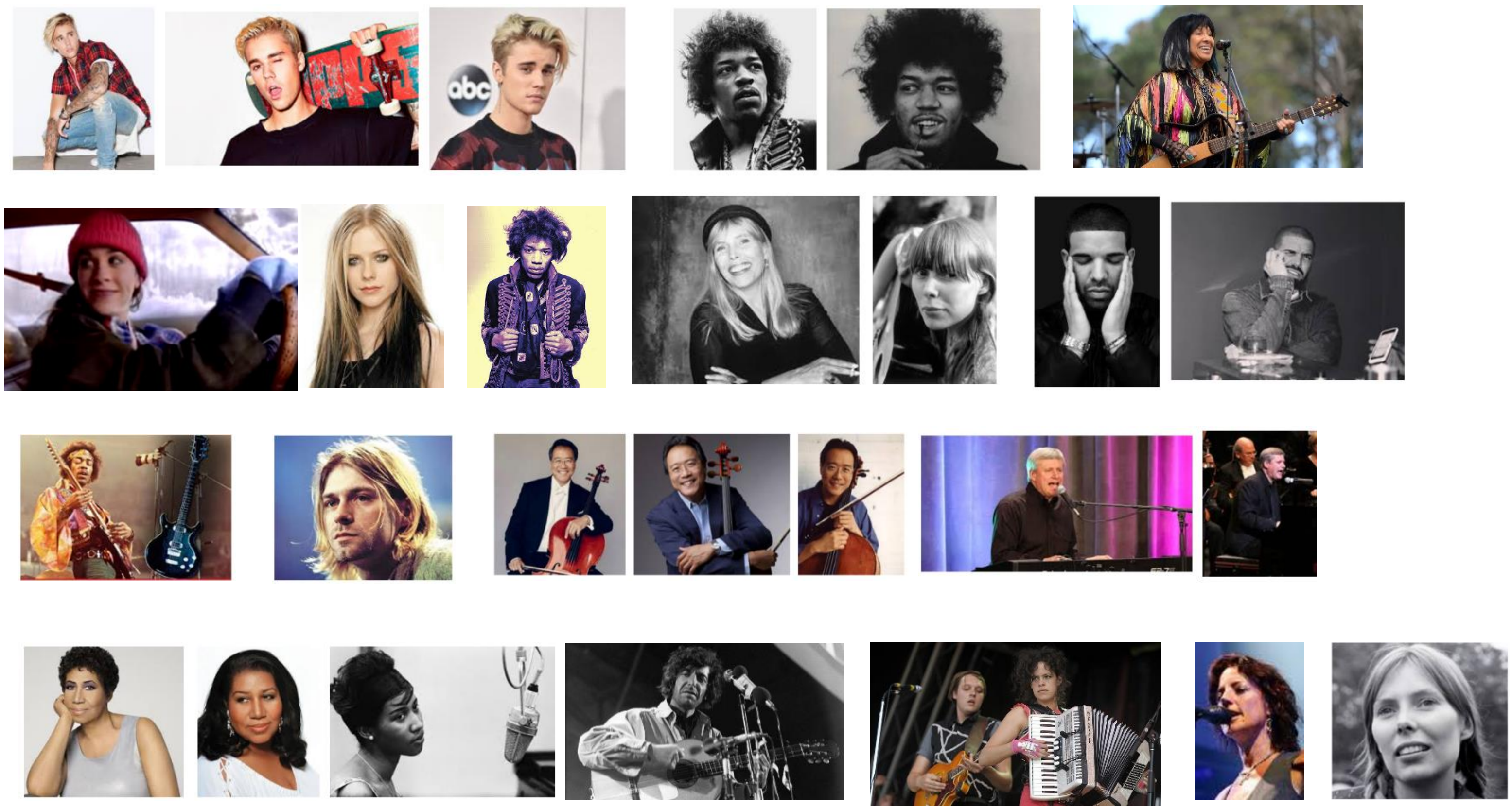
Face Classification with Logistic Regression



Slides from:
Derek Hoiem
ESLII, Friedman, Hastie and Tibshirani

SML310: Projects in Data Science, Fall 2019

Michael Guerzhoy

The Task: Supervised Learning

- Given a set of labelled examples (the *training set*), determine/predict the labels of a set of unlabelled examples (the *test set*)
  - Training set:

    Train Example 1: $(x_1^{(1)}, x_2^{(1)}, \dots., x_m^{(1)})$ Label: $y^{(1)}$

    Train Example 2: $(x_1^{(2)}, x_2^{(2)}, \dots., x_m^{(2)})$ Label: $y^{(2)}$

    …

    Train Example N: $(x_1^{(N)}, x_2^{(N)}, \dots., x_m^{(N)})$ Label: $y^{(N)}$
  - Test set:

    Test Example 1: $(x_1^{(N+1)}, x_2^{(N+1)}, \dots., x_m^{(N+1)})$ Label: $y^{(N+1)}$

    Test Example 2: $(x_1^{(N+2)}, x_2^{(N+2)}, \dots., x_m^{(N+2)})$ Label: $y^{(N+2)}$

    …

    Test Example K: $(x_1^{(N+K)}, x_2^{(N+K)}, \dots., x_m^{(N+K)})$ Label: $y^{(N+K)}$

Machine Learning vs. Intro to Programming

- Intro to Programming ***done badly***

```
def double_list(L):
  for e in L:
    e *= 2
  return L

>>> double_list([0, 0])
[0, 0]
>>> double_list([1, 2])
[1, 2]
```

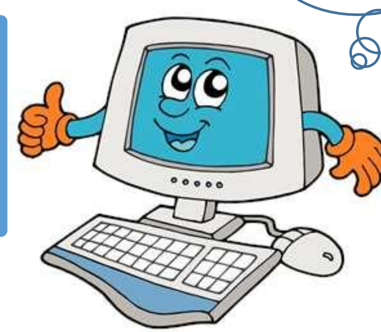
Change the for to while?

Shotgun debugging

- Machine Learning ***done right***

>>> $h_{(0,1.2,0.1)}$([0, 0])
[0, 0]
>>> $h_{(0,1.2,0.1)}$([1, 2])
[1.3, 2.8]

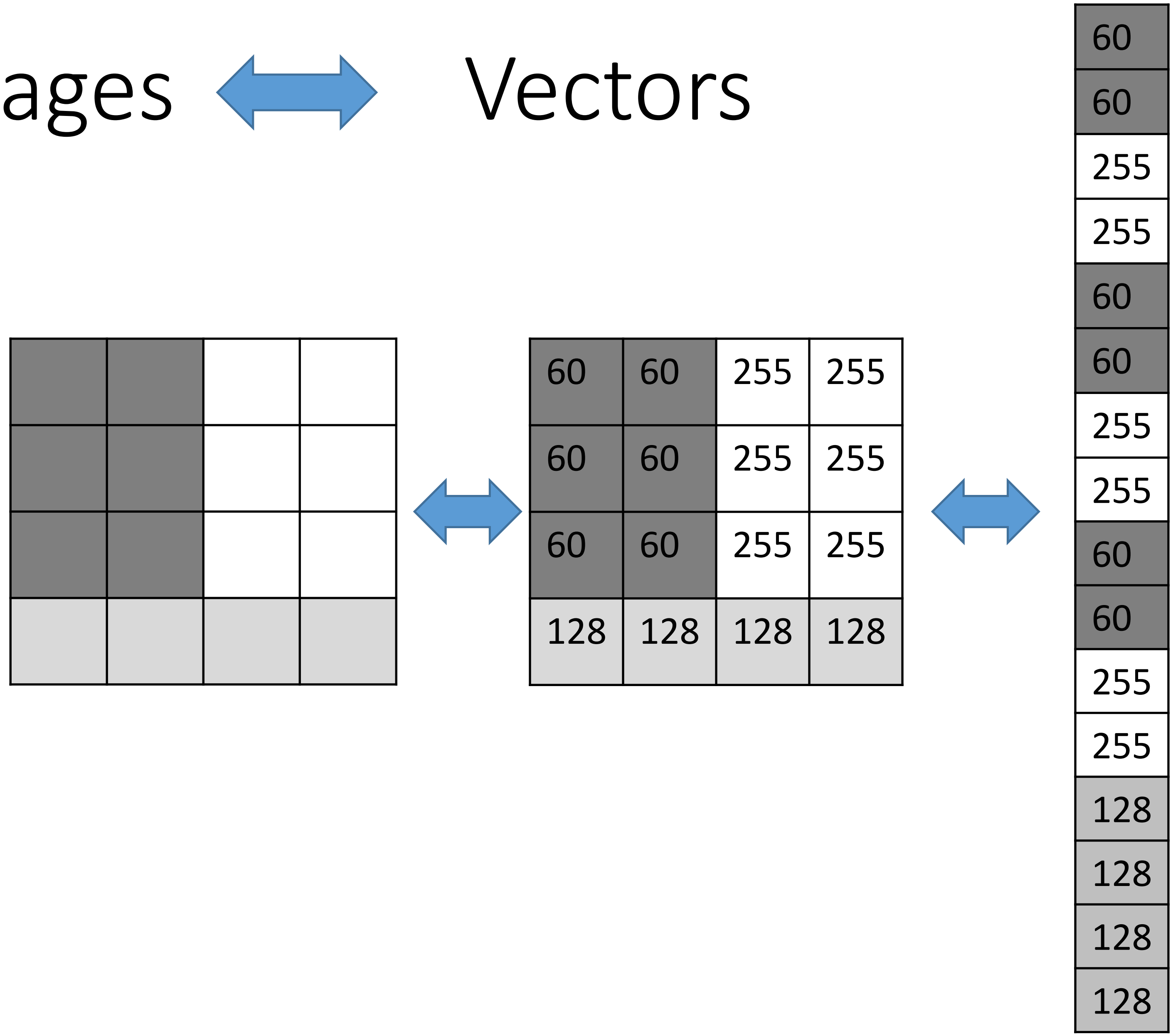
$$h_{(\theta_1,\theta_2,\theta_3)}(x) = \theta_1 + \theta_2 x + \theta_3 x^2$$



Change θ_2 to 1.3?

Machine learning (kind of)

Images ⟷ Vectors



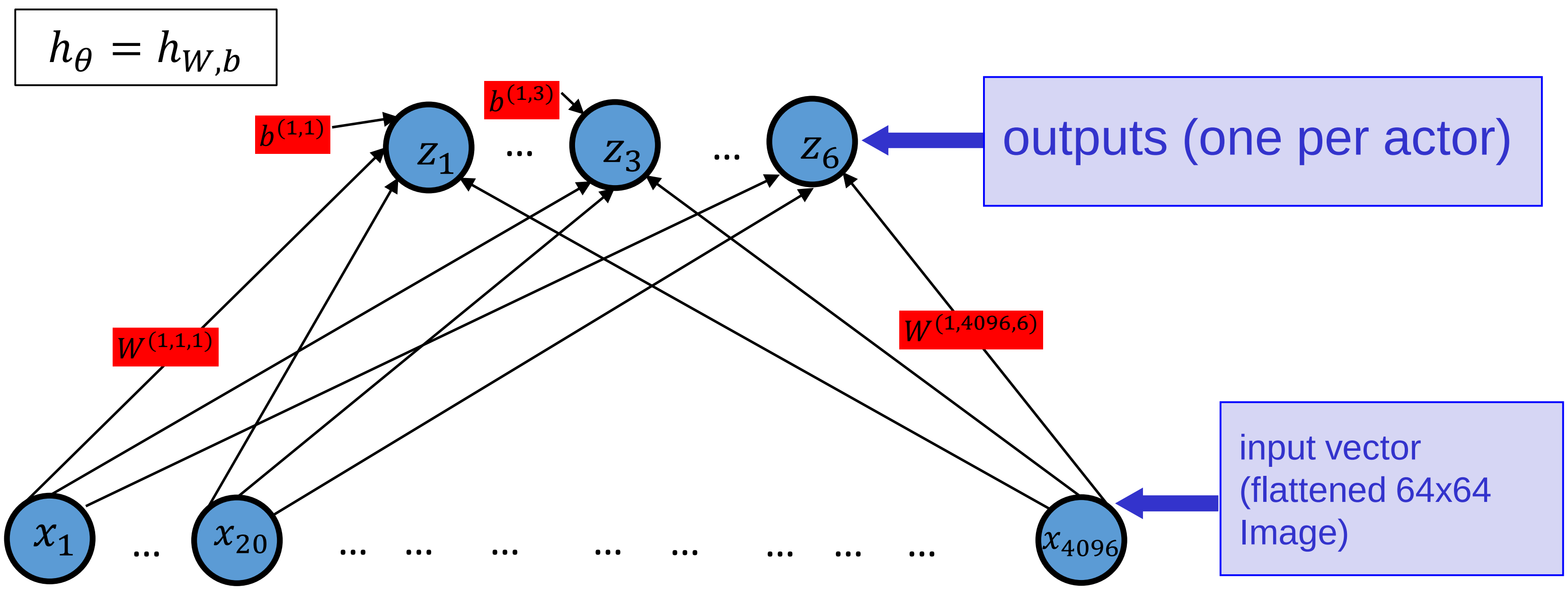
Project 4 task

- Training set: 6 actors, with 100 64×64 photos of faces for each
- Test set: photos of faces of the same 6 actors
- Want to classify each face as one of ['Fran Drescher', 'America Ferrera', 'Kristin Chenoweth', 'Alec Baldwin', 'Bill Hader', 'Steve Carell']

Multiclass Classification

$$z_k = \sigma\left(\sum_{j=1}^{4096} W^{(1,j,k)} x_j + b^{(1,k)}\right)$$

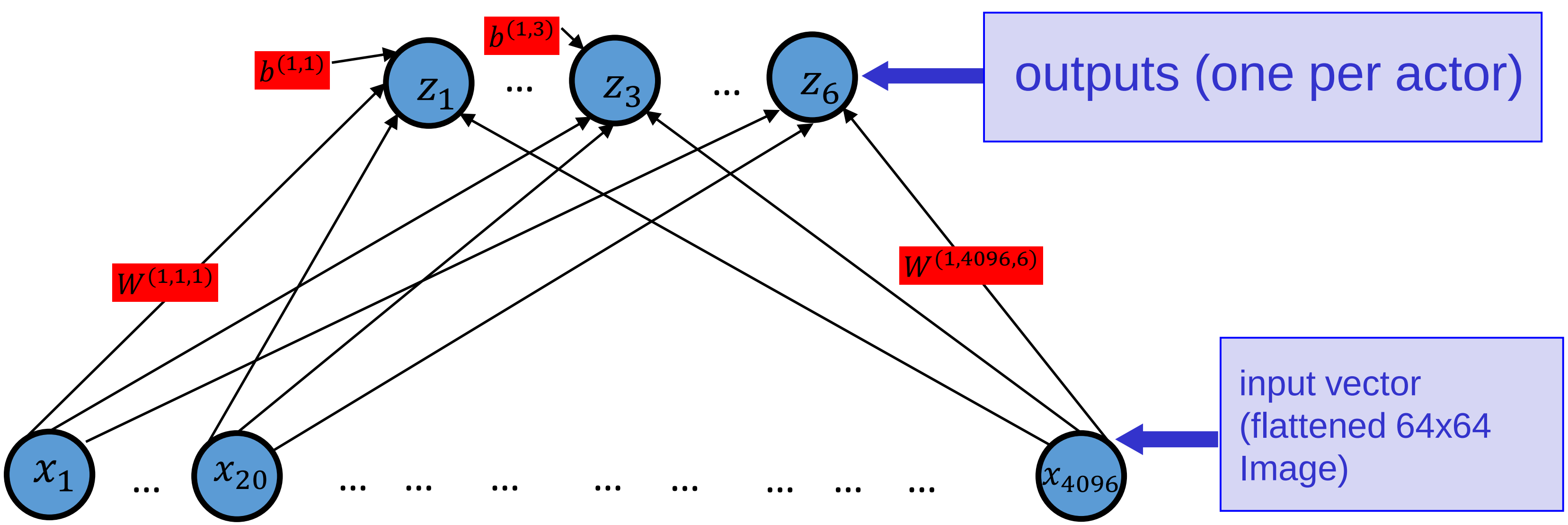
x



The transformation with σ is not necessary here, but will be useful later

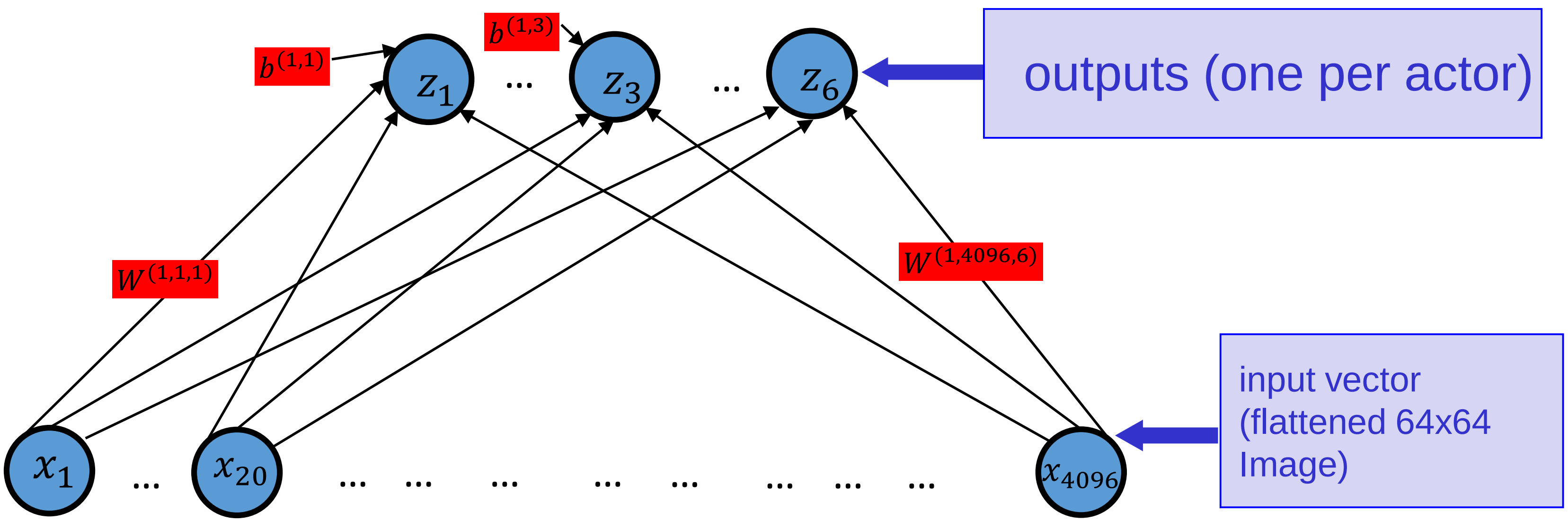
Fitting the model

- Adjust the W's (4096 × 6 coefs) and b's (6 coefs)
  - Try to make it so that if
    x is an image of actor 1, z is as close as possible to (1, 0, 0, 0, 0, 0)
    x is an image of actor 2, z is as close as possible to (0, 1, 0, 0, 0, 0)
    ……



Face recognition

- Compute the z for a new image x
- If z_k is the largest output, output name k



An interpretation

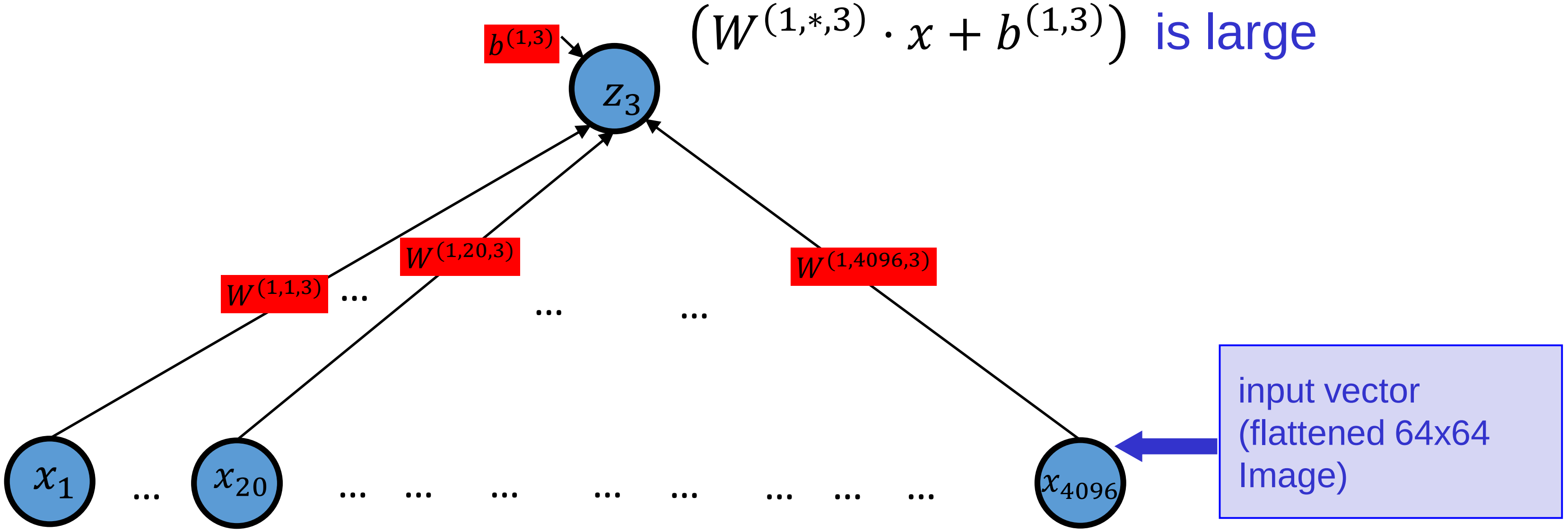
z_1 is large if $W^{(1,*,1)} \cdot x$ is large
z_2 is large if $W^{(1,*,2)} \cdot x$ is large

z_3 is large if $W^{(1,*,3)} \cdot x$ is large

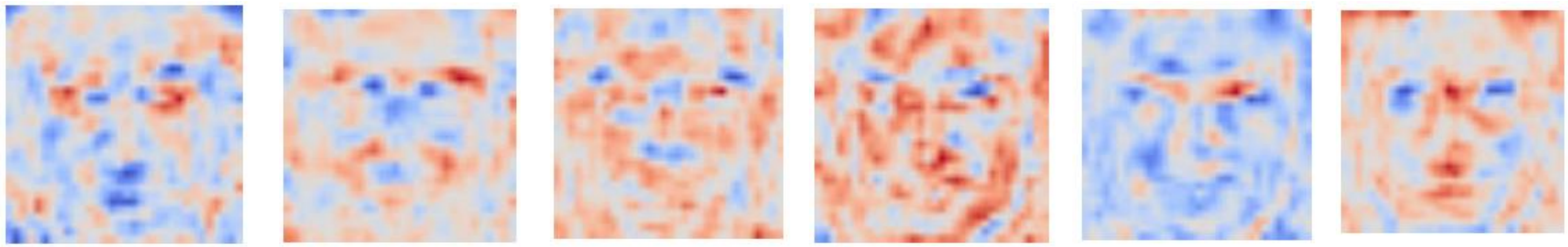
….

$W^{(1,*,1)}$, $W^{(1,*,2)}$, …, $W^{(1,*,6)}$ are *templates* for the faces of actor 1, actor 2, …, actor 6

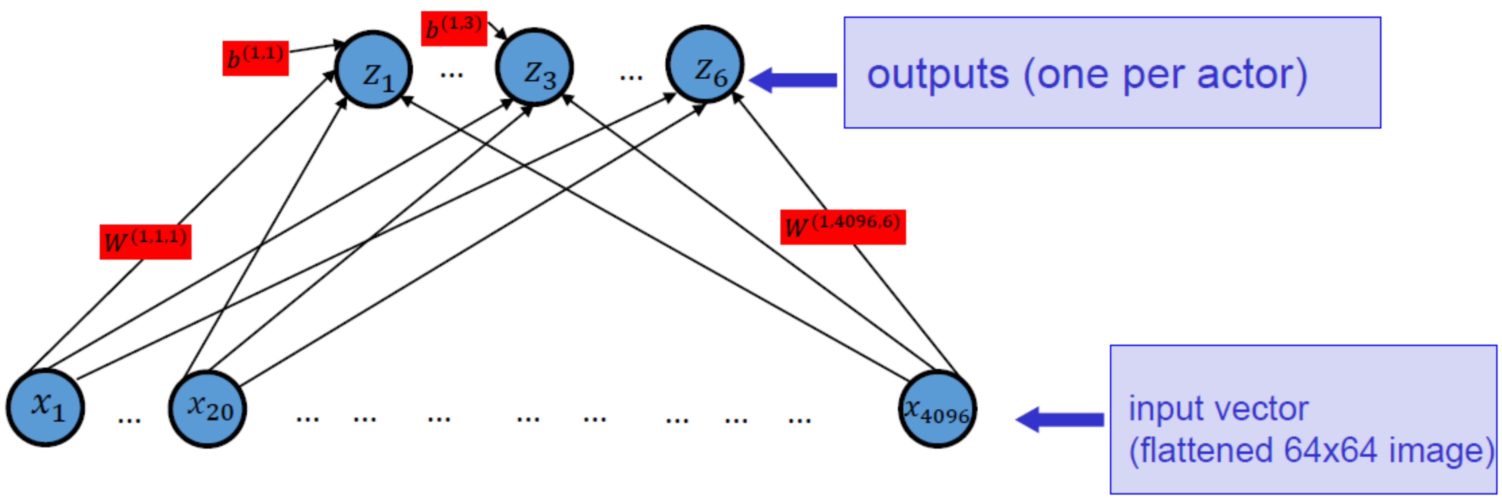
Actor 3 output activated:
$\left(W^{(1,*,3)} \cdot x + b^{(1,3)}\right)$ is large



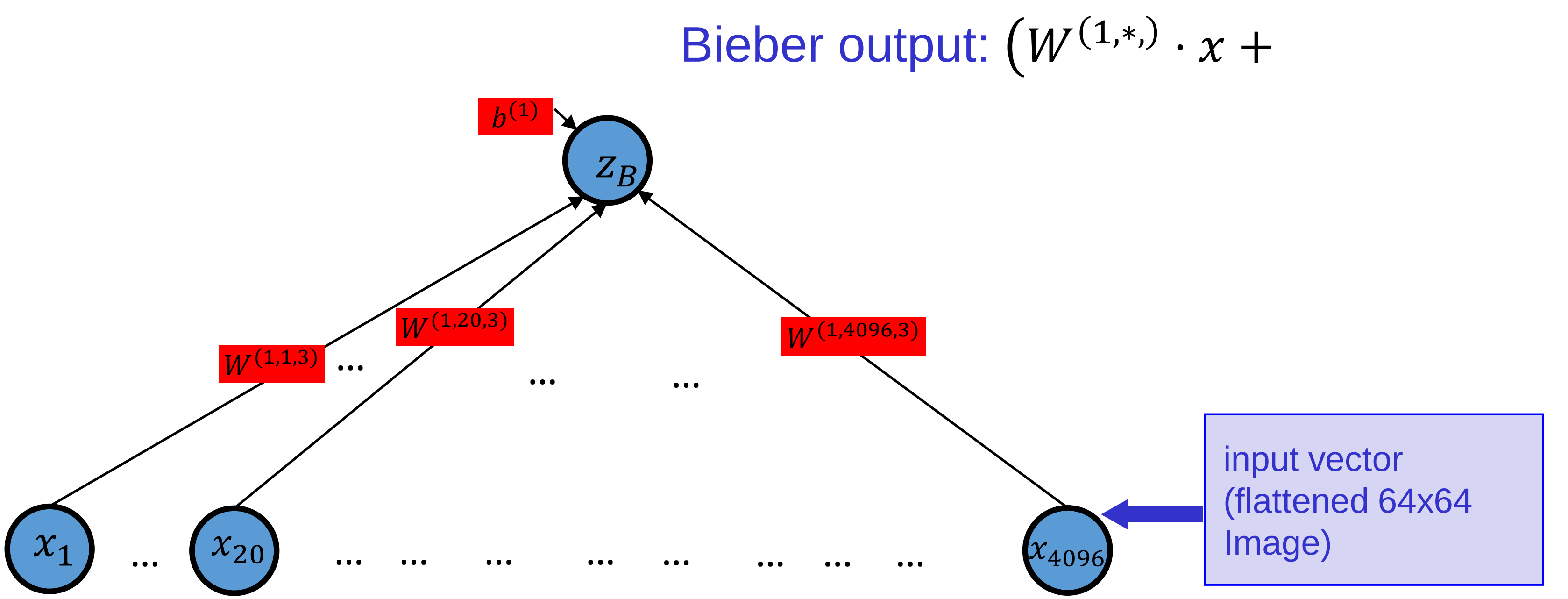
Visualizing the parameters W



Baldwin $W^{(1,*,1)}$ | Carrel $W^{(1,*,2)}$ | Hader $W^{(1,*,3)}$ | Ferrera $W^{(1,*,4)}$ | Drescher $W^{(1,*,5)}$ | Chenoweth $W^{(1,*,6)}$



Connection to Logistic Regression



The structure we saw before is (for the appropriate cost function) Multinomial Logistic Regression. (If we have just one output, the procedure is Binomial Logistic regression)