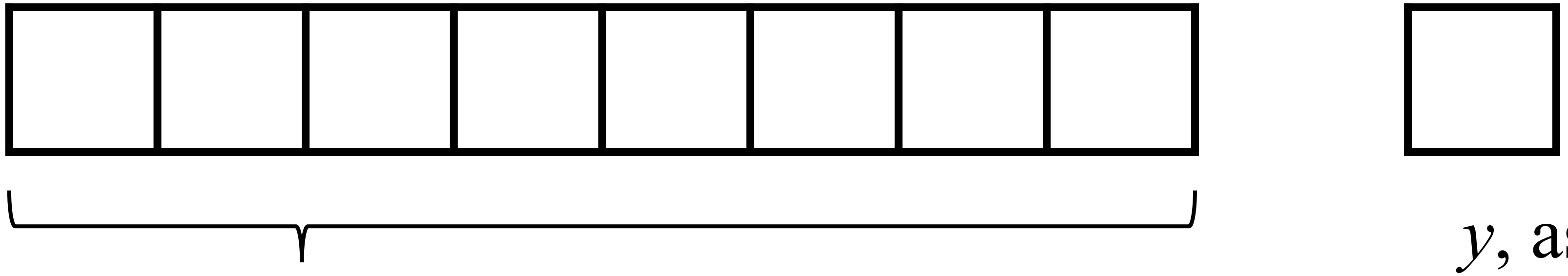
Logistic Regression

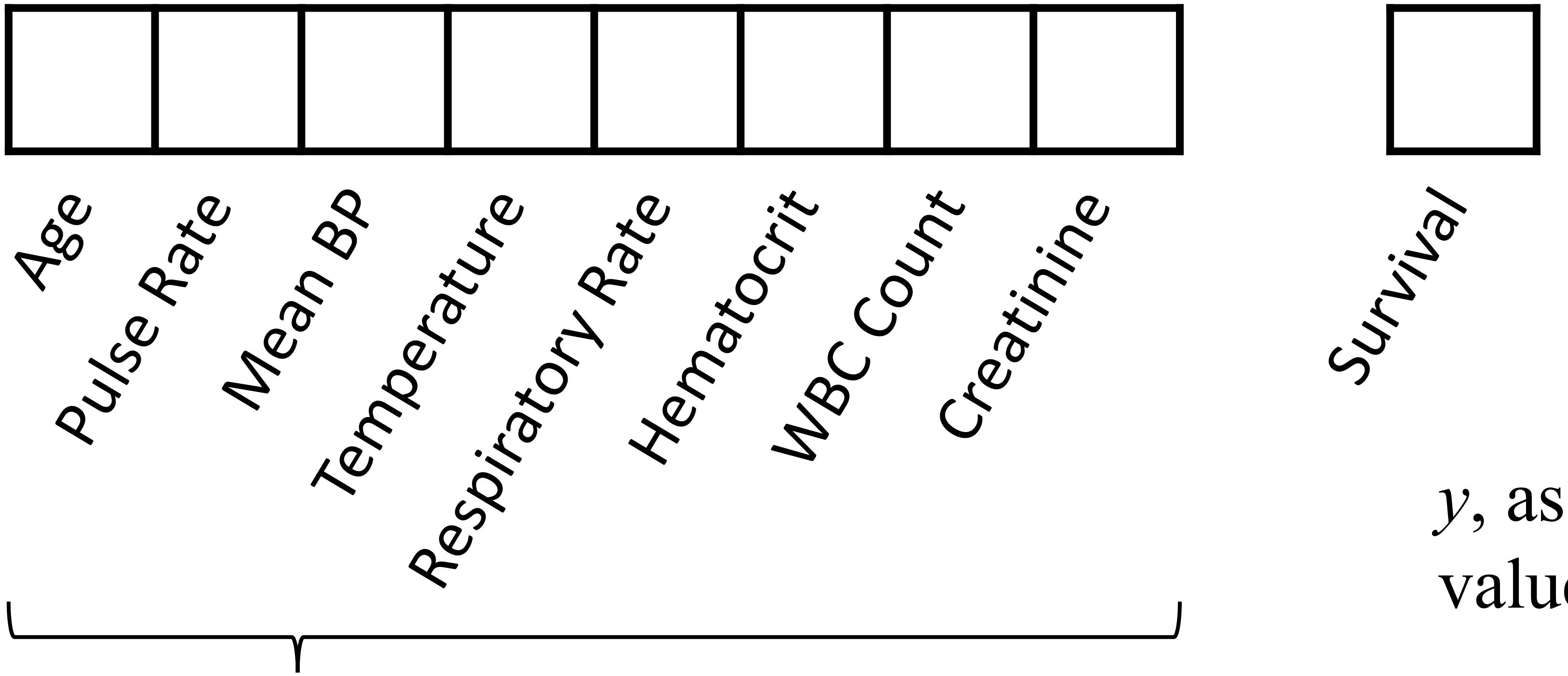
Matthew Engelhard

features x -> prediction y: a predictive model



End goal: predict y from x

Simple models often work well for clinical data!

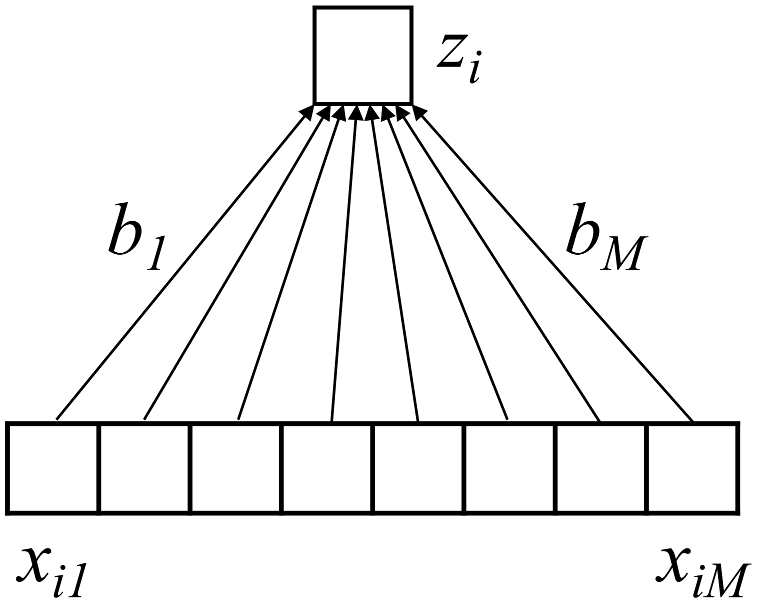


y, associated value or label

x, data/features for a subject or patient

End goal: predict odds of hospital mortality (APACHE III)

Can we use a linear model?



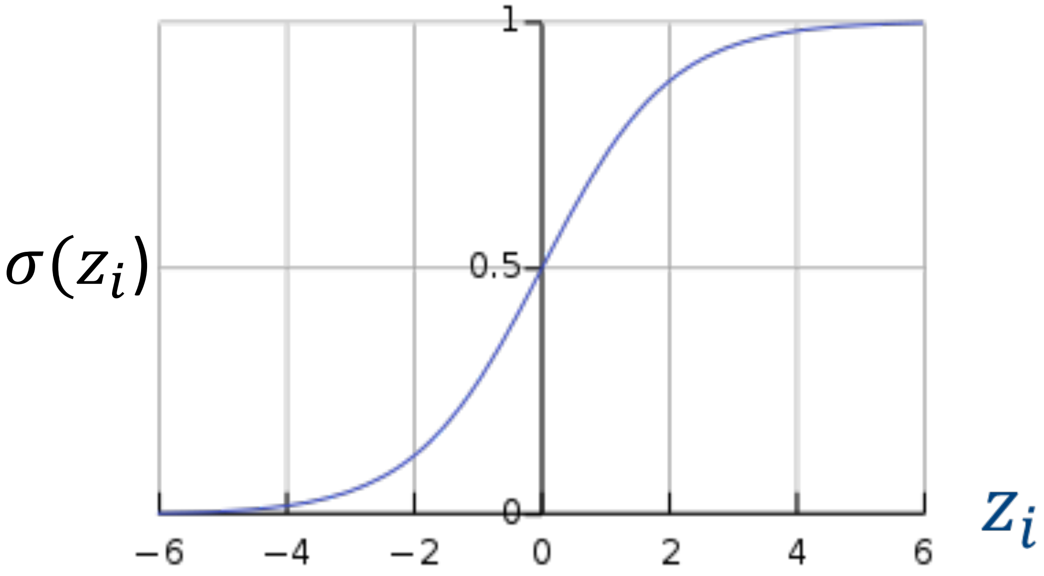
$$z_i = b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM}$$

The logistic function converts z_i to a probability

$$z_i = b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM} + \cdots + b_0$$

Extra Constant
(i.e. intercept)
(i.e. bias)

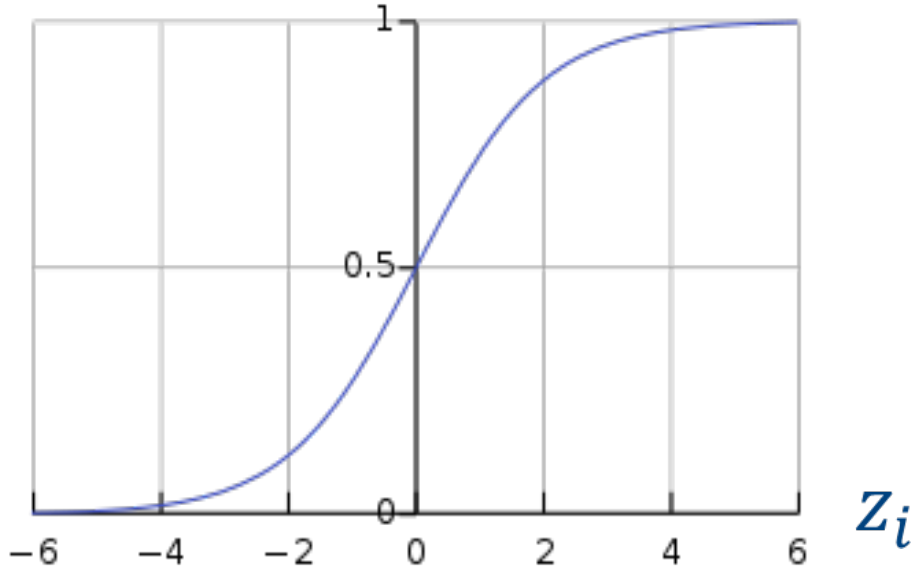
$$p(y_i = 1 | x_i) = \sigma(z_i)$$



The logistic function converts z_i to a probability

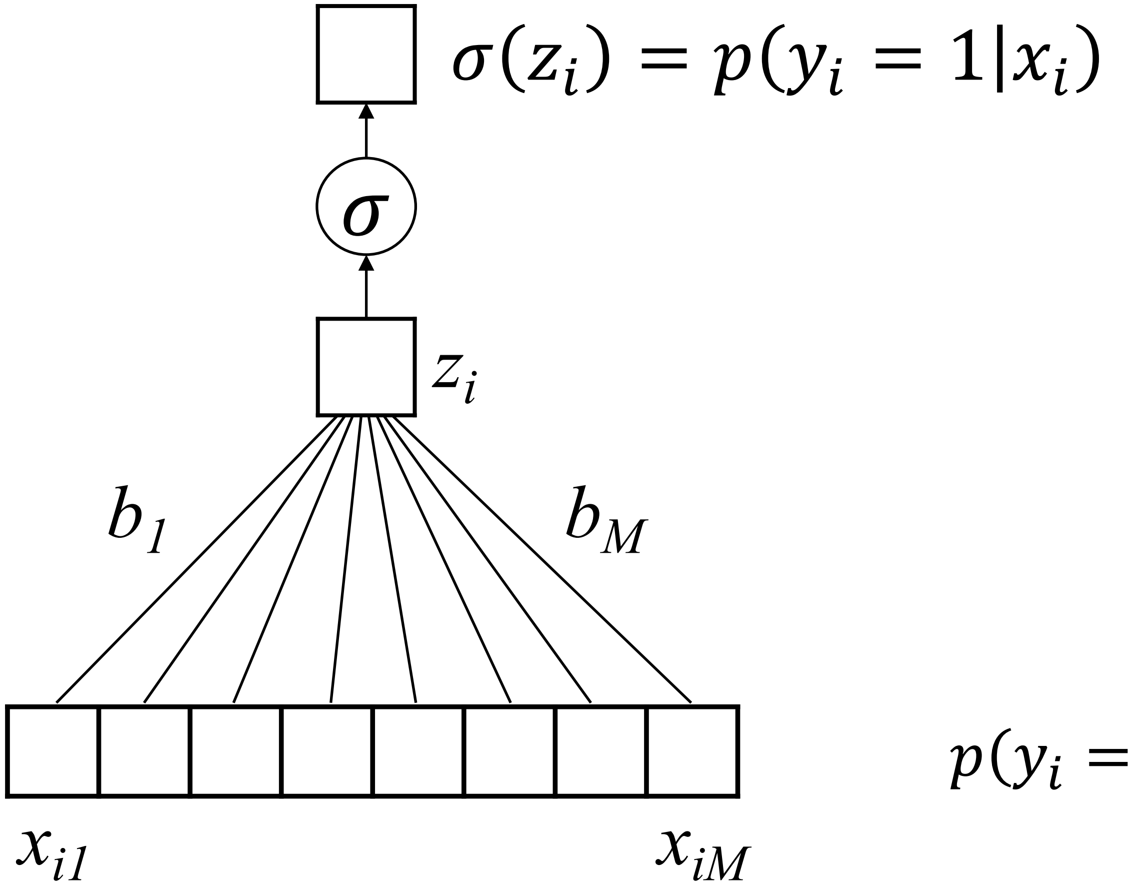
$$z_i = b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM} + \cdots + b_0$$

$$p(y_i = 1 | x_i) = \sigma(z_i) = \frac{\exp(z_i)}{1+\exp(z_i)}$$



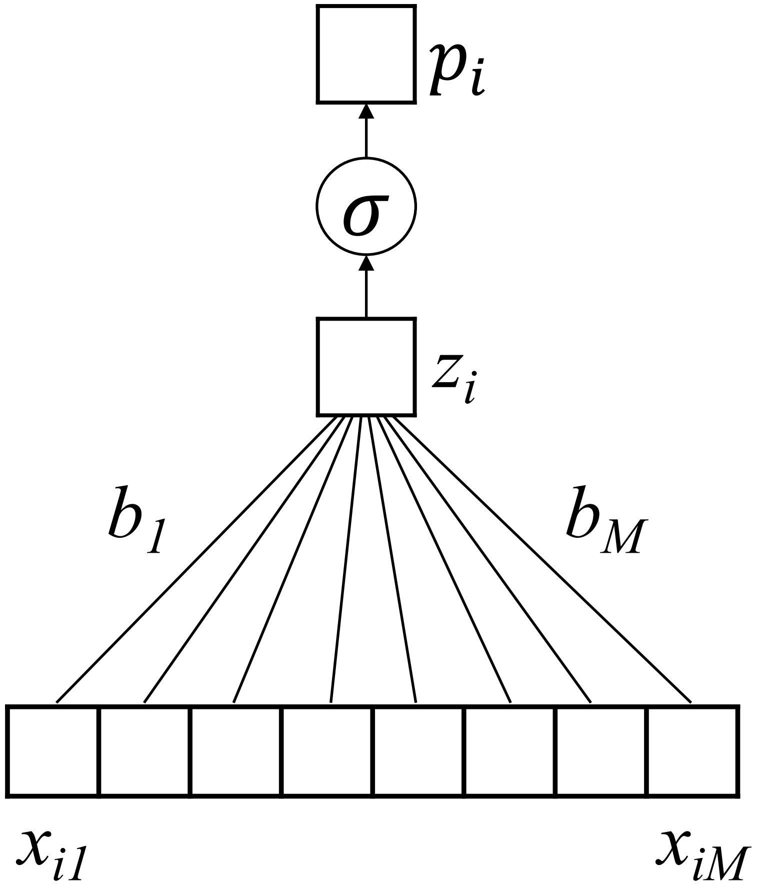
- Large and positive z_i indicates that event $y_i = 1$ is likely
- Large and negative z_i indicates that event $y_i = 0$ is likely

Logistic Regression: a linear model with a logistic "link" function that converts the prediction to a probability



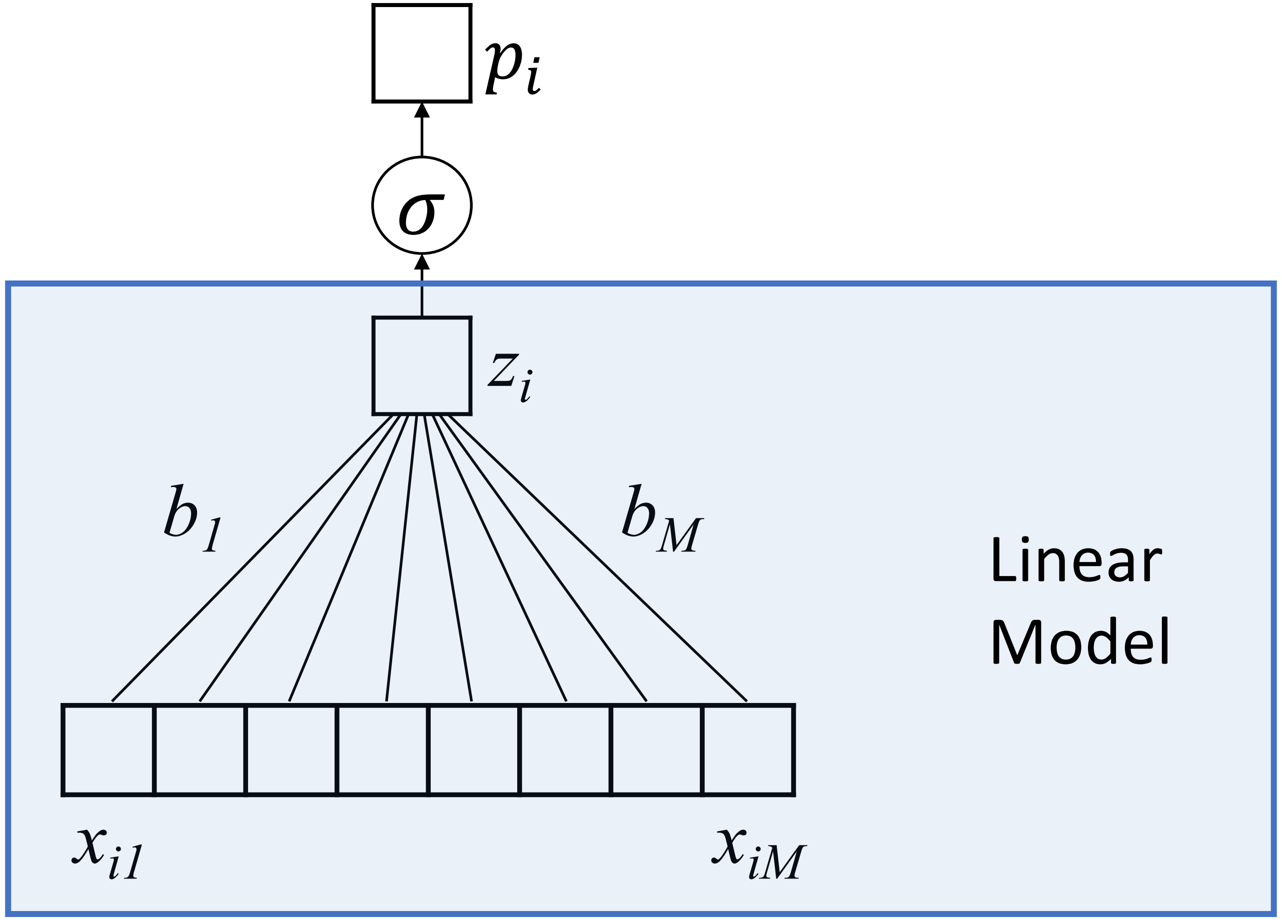
$$p(y_i = 1|x_i) = \sigma(b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM})$$

Logistic Regression: a linear model with a logistic "link" function that converts the prediction to a probability



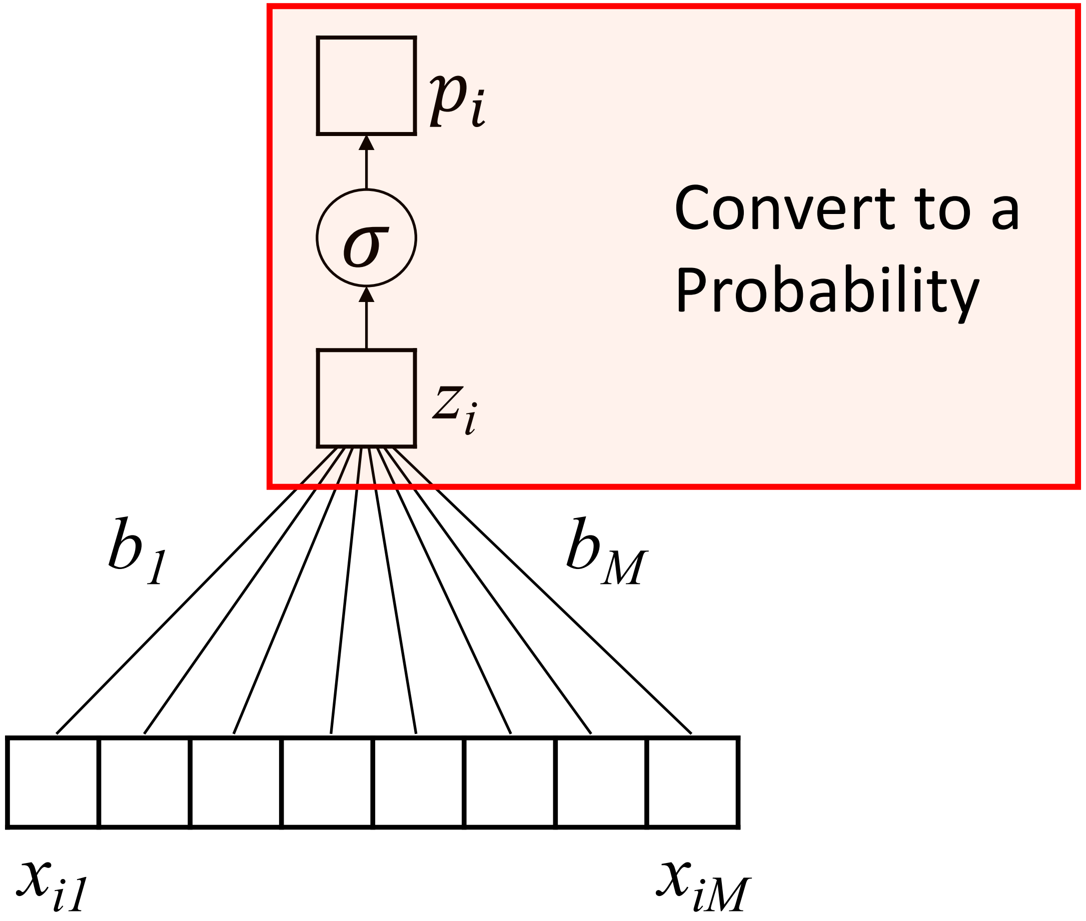
$$p_i = \sigma(b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM})$$

Logistic Regression



$$p_i = \sigma(b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM})$$

Logistic Regression



$$p_i = \sigma(b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM})$$

ICU Mortality Prediction

A clinical example

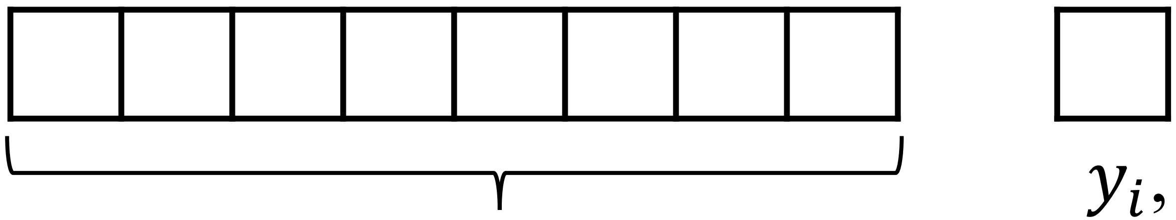
Example: ICU Mortality Prediction

- Outcome:

$$y_i = \begin{cases} 1, \text{patient } i \text{ dies} \\ 0, \text{patient } i \text{ lives} \end{cases}$$

- Features: On admission, what is patient i's

$$\{\text{age}, \text{sex}, \text{temperature}, \text{blood pressure}, \dots\}$$



x_i, features for patient i

y_i, did patient i die

Example: ICU Mortality Prediction

- Outcome:

$$y_i = \begin{cases} 1, \text{patient } i \text{ dies} \\ 0, \text{patient } i \text{ lives} \end{cases}$$

- Features: On admission, what is patient i's:

$$\{1\text{: age}, 2\text{: sex}, 3\text{: temperature}, 4\text{: blood pressure} \ldots\}$$

$$z_i = b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM} + b_0$$

Age

Blood Pressure

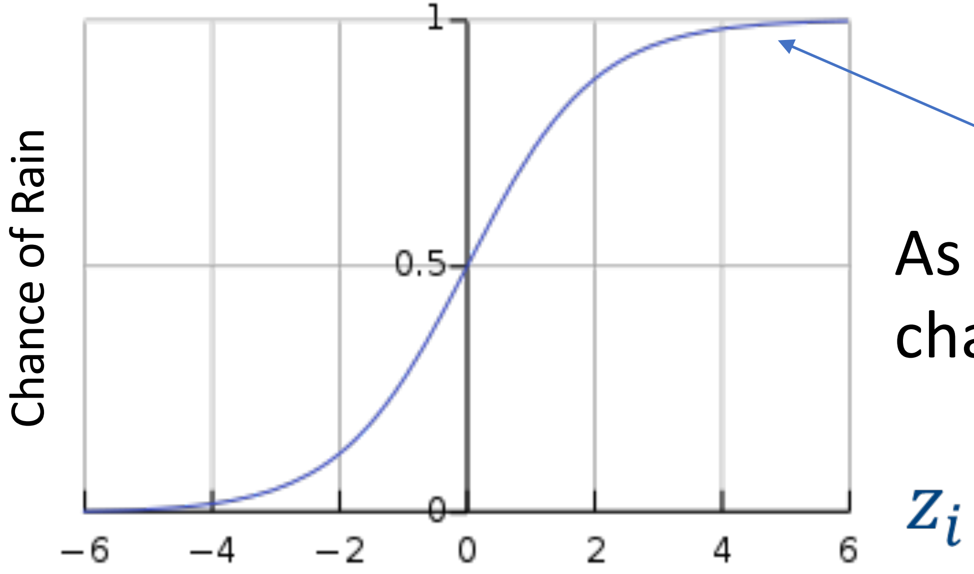
- If increased age increases odds of mortality, b_1 should be positive

Impact on the Sigmoid Function

$$z_i = b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM} + b_0$$

Age

$$p(y_i = 1|x_i) = \sigma(z_i)$$



As the value z_i increases, the chance of mortality increases

The logistic function just converts the patient's log-odds (of mortality) to the corresponding probability.

An example:

- Suppose the patient's predicted log odds = 2
- Convert log odds to odds by exponentiating: e^2 = 7.4
- The odds are always relative to 1; in other words, they are 7.4x more likely to die than not
- Convert odds to probability = 7.4 / (1 + 7.4) = 0.88

Building the Training Set

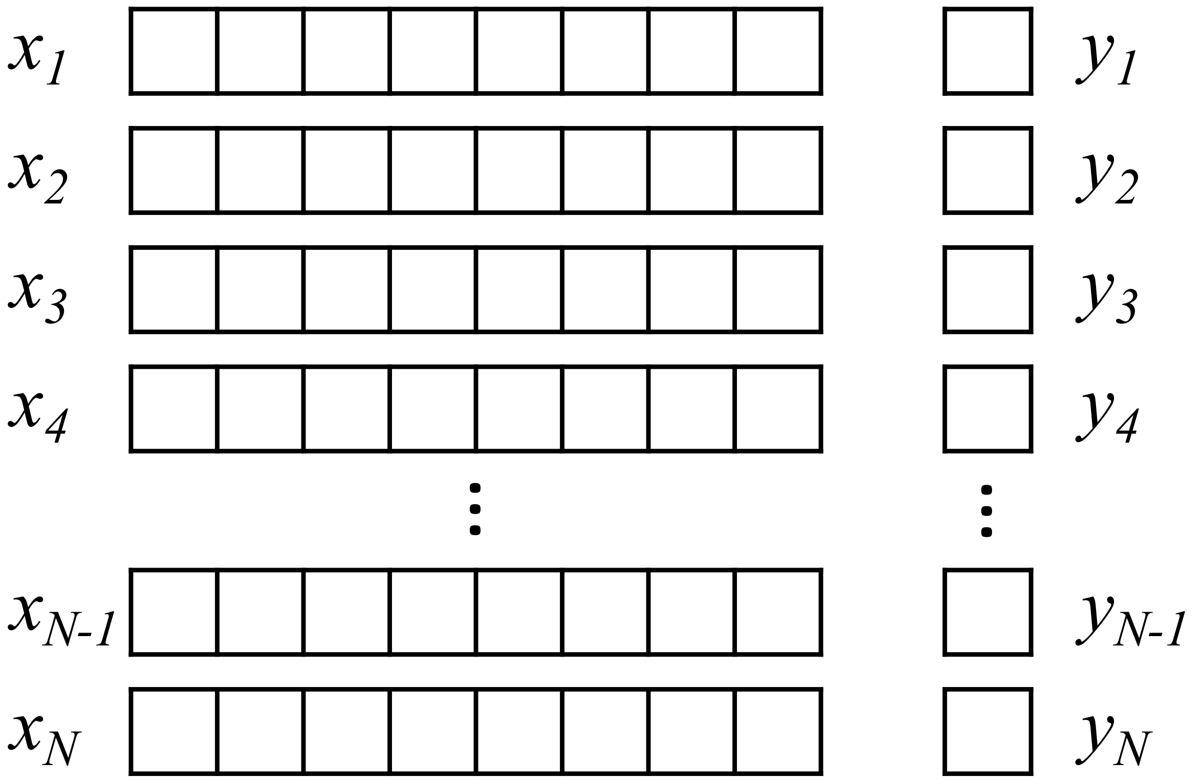
- We want to learn the model parameters

$$b = (b_0, \ldots, b_M)$$

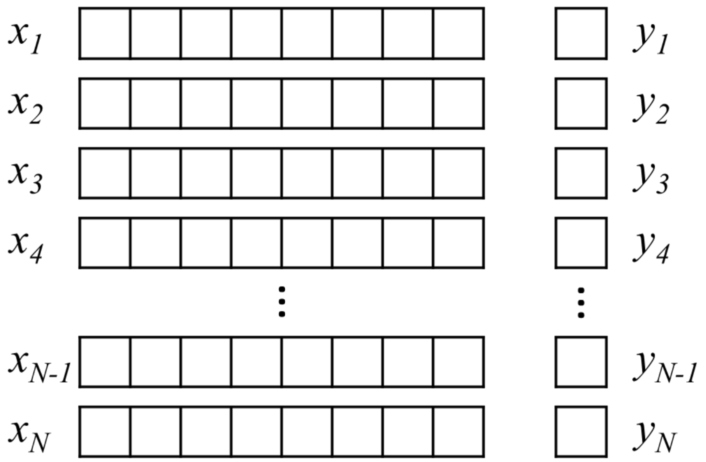
- This requires *training data*; we will find the *b* that match it best

- Record data from N patients
  - Capture features:

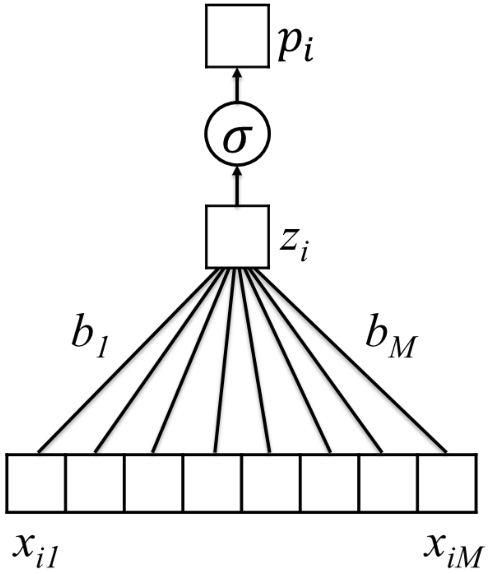
    {age, sex, temp, BP, …}
  - Did they survive?



Learning Model Parameters

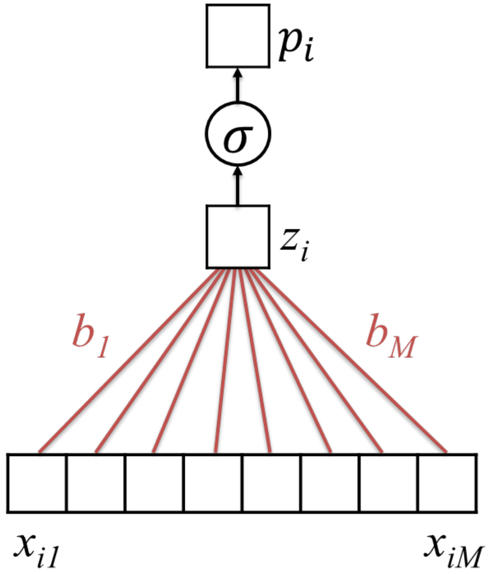


Training Set



$$p_i = \sigma(b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM})$$

Untrained Logistic Regression Model (or “Network”)

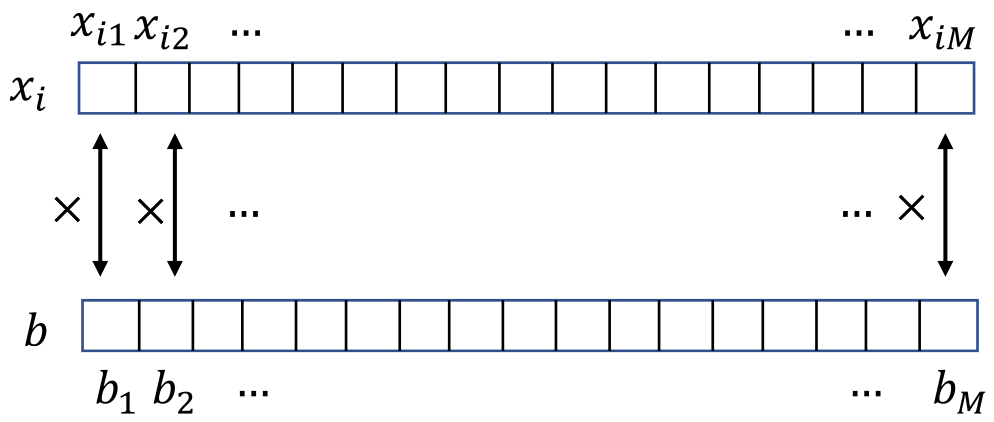


$$b = (b_0, \ldots b_M)$$

Trained Model (with learned parameters)

Simplifying our Notation…

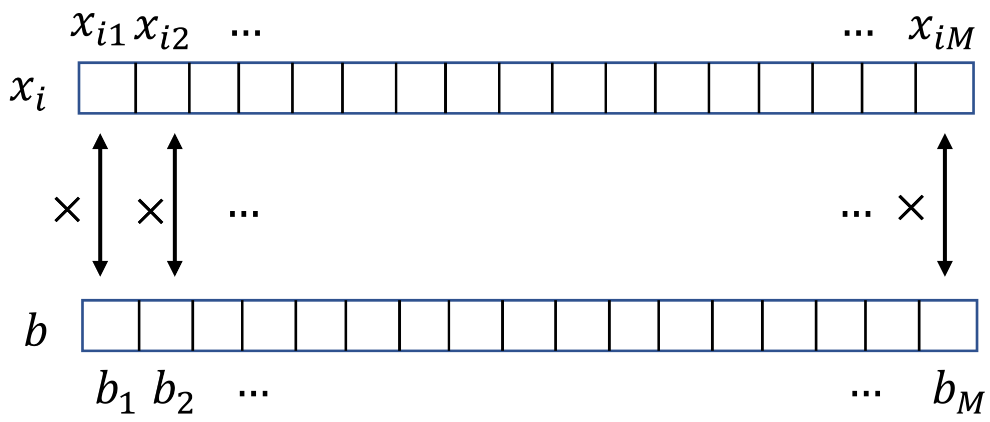
$$z_i = b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM}$$



$$\Rightarrow \sum_{m=1}^{M} x_{im} b_m$$

Simplifying our Notation…

$$z_i = b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM}$$



$$\sum_{m=1}^{M} x_{im} b_m$$

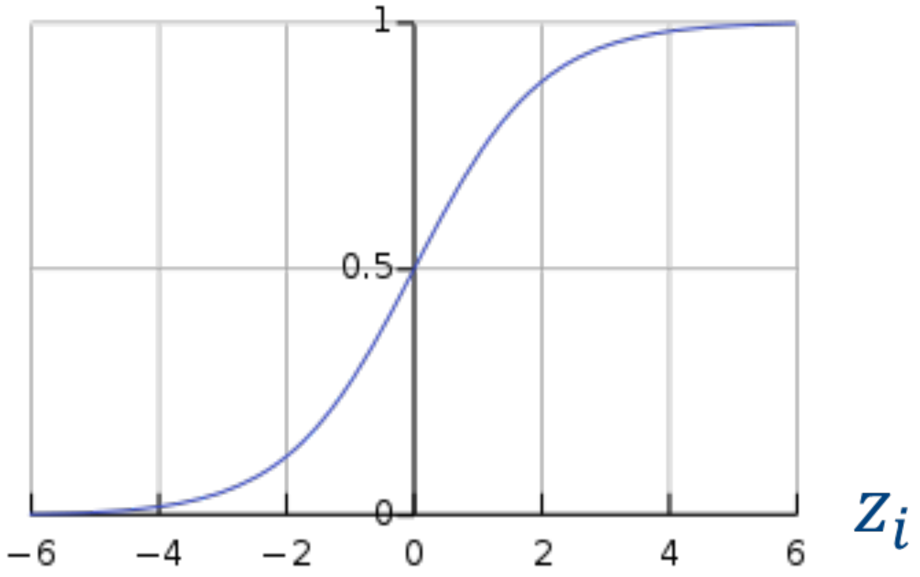
Compact Notation: $x_i \odot b$ (*inner* or *dot* product)

Interpretation of Logistic Regression

$$z_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + b_M x_{iM}$$

$$= b_0 + x_i \odot b$$

$$p(y_i = 1 | x_i) = \sigma(z_i)$$



- May think of vector b as a template or filter (will visualize to make clear)
- If x_i is aligned/matched with b, then $x_i \odot b$ will be large
- The parameter b_0 is a bias to correct for class prevalence

Recognizing Handwritten Digits

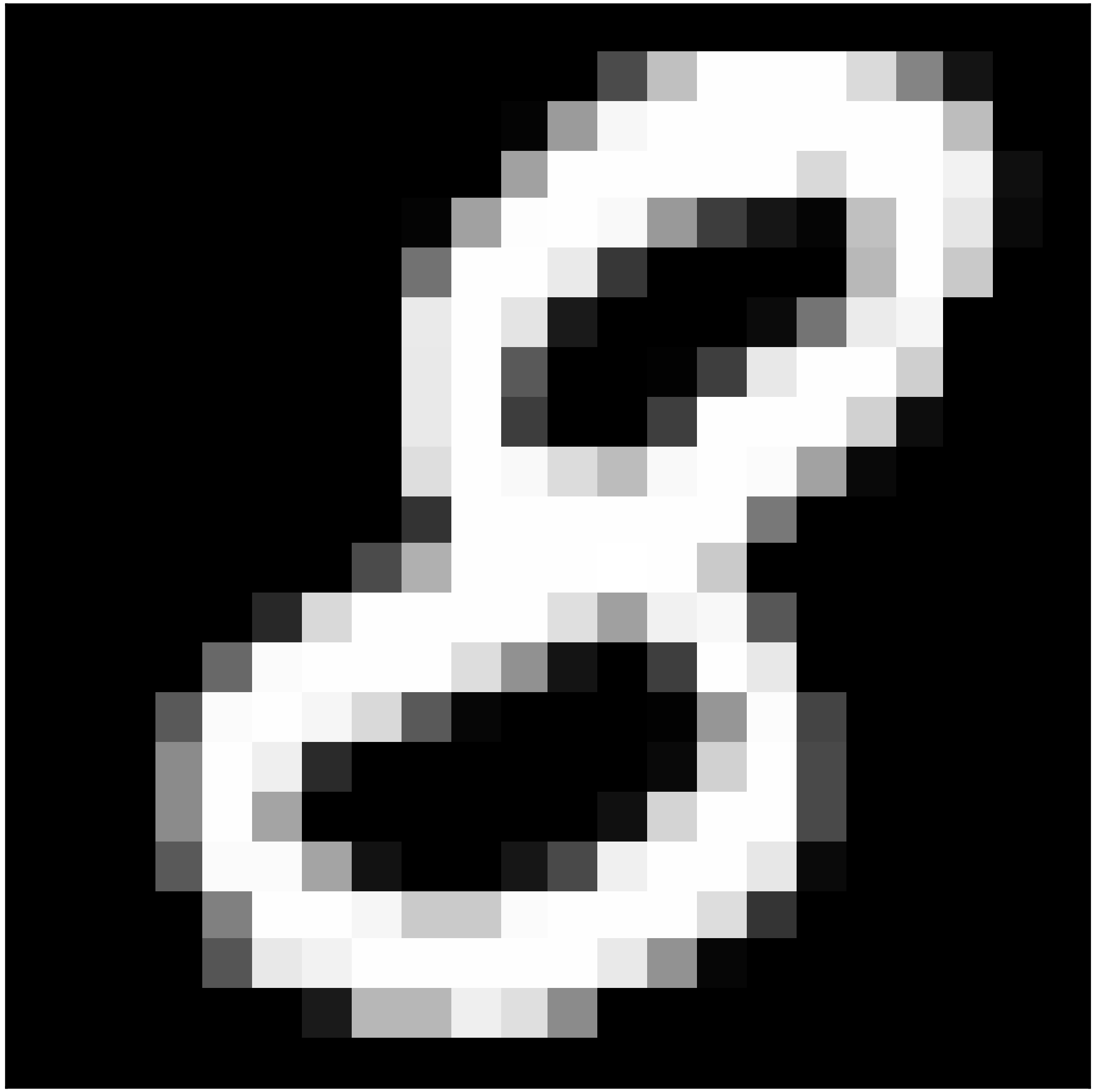
A visual example

The MNIST Dataset

- The Modified National Institute of Standards and Technology (MNIST) contains pictures of handwritten digits (0,1,2,…)

- Want to be able to tell what digit each image is (*e.g.,* optical character recognition)

Images are Encoded as Numbers



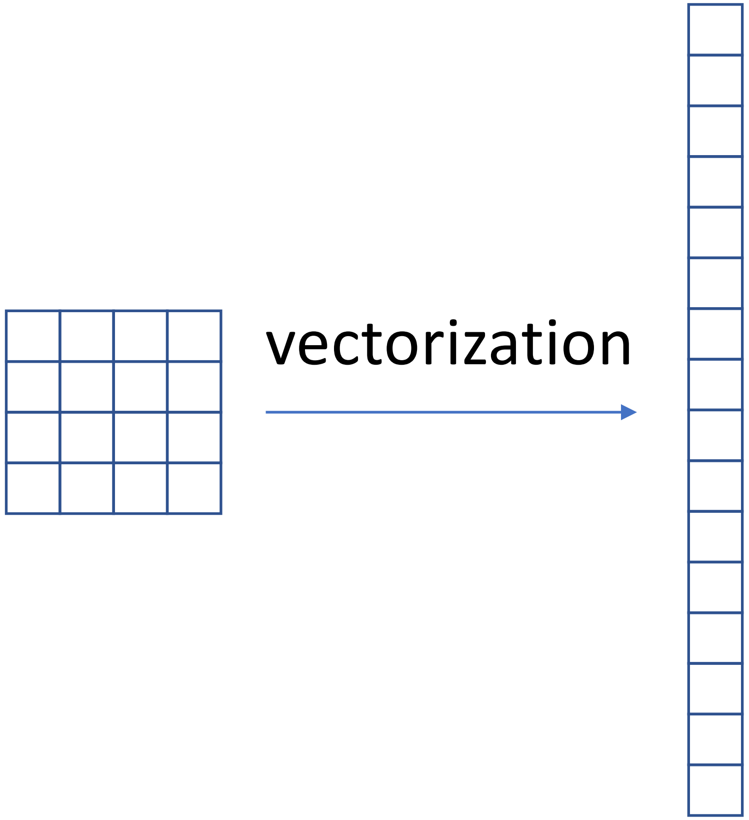
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	118	200	223	155	155	23	0	0	0	0	0
0	0	0	0	0	0	0	0	6	124	214	253	253	253	253	254	229	213	67	0	0	0
0	0	0	0	0	0	0	43	198	253	254	253	247	175	175	236	253	253	108	0	0	0
0	0	0	0	0	0	9	212	253	253	224	58	19	0	0	16	139	247	247	71	0	0
0	0	0	0	0	0	58	254	254	186	14	0	0	0	0	0	0	139	254	118	0	0
0	0	0	0	0	0	58	253	183	8	0	0	0	0	0	0	36	222	253	118	0	0
0	0	0	0	0	0	54	250	128	2	0	0	0	6	71	192	237	253	247	71	0	0
0	0	0	0	0	0	0	213	253	50	0	18	123	198	253	254	253	247	85	0	0	0
0	0	0	0	0	0	0	69	241	227	136	200	253	253	253	254	192	34	0	0	0	0
0	0	0	0	0	0	0	0	175	254	256	254	254	254	149	59	0	0	0	0	0	0
0	0	0	0	0	0	0	101	253	253	254	253	253	253	42	0	0	0	0	0	0	0
0	0	0	0	0	8	138	247	253	243	159	196	243	253	199	0	0	0	0	0	0	0
0	0	0	0	12	183	253	253	253	50	0	0	49	253	214	0	0	0	0	0	0	0
0	0	0	0	180	254	253	213	50	2	0	0	71	253	214	0	0	0	0	0	0	0
0	0	0	0	234	217	97	10	0	0	0	23	207	254	215	0	0	0	0	0	0	0
0	0	0	174	253	156	0	0	0	0	45	215	253	253	95	0	0	0	0	0	0	0
0	0	9	210	253	163	5	19	49	130	244	253	251	137	4	0	0	0	0	0	0	0
0	0	13	229	253	254	192	253	253	253	254	253	137	0	0	0	0	0	0	0	0	0
0	0	0	160	253	254	253	253	253	253	239	132	4	0	0	0	0	0	0	0	0	0
0	0	0	18	112	194	254	254	254	163	59	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Vectorization

- We will start talking about deep learning *without* using the structure of the image

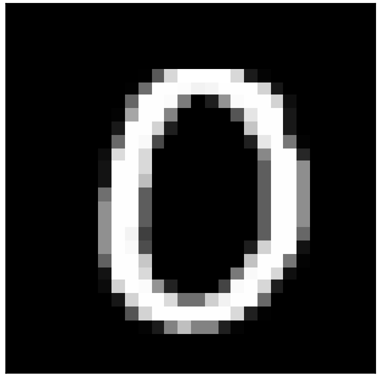
- Later, in block 2, we will consider how to take advantage of this structure

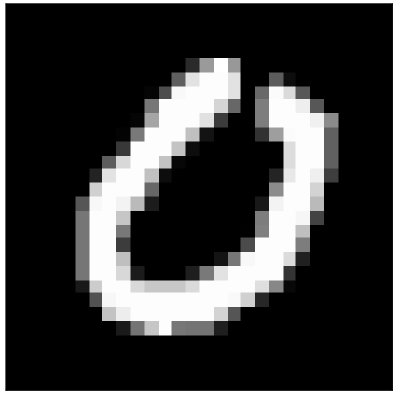
- To convert an image into an unstructured set of numbers, we *vectorize* (or *flatten*) it

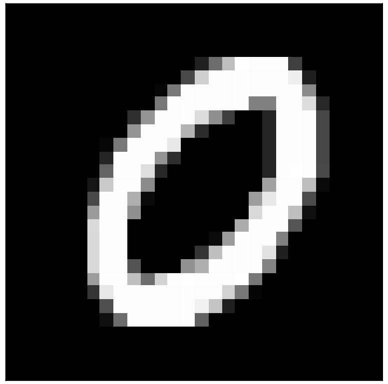


Start With The Binary Case

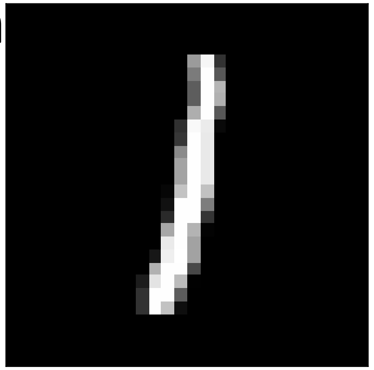
Zeros

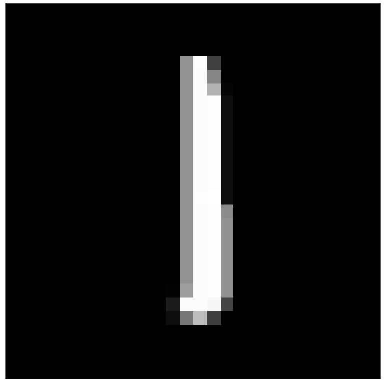


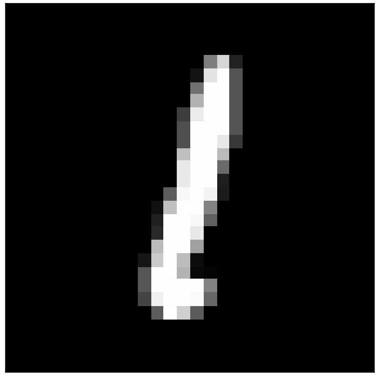




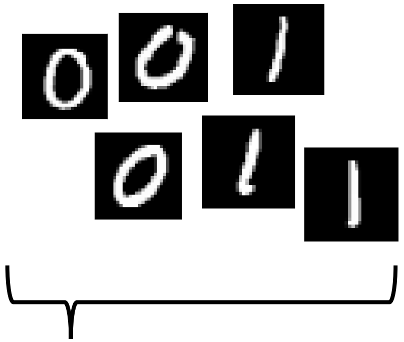
On





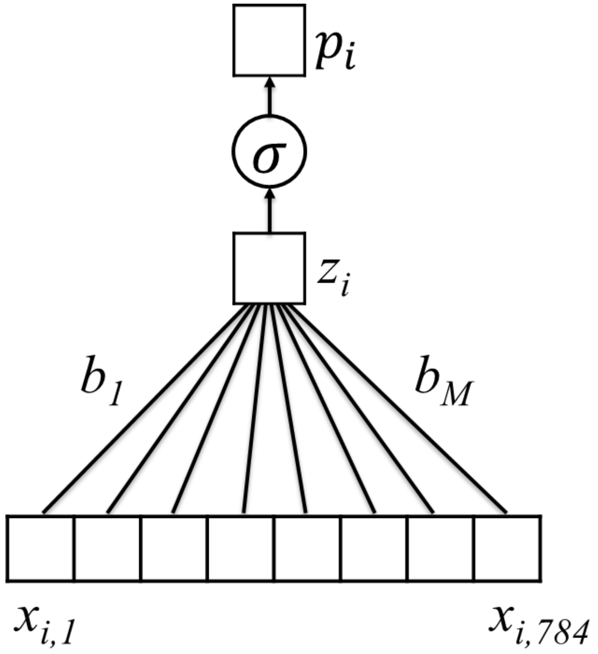


Learning on MNIST



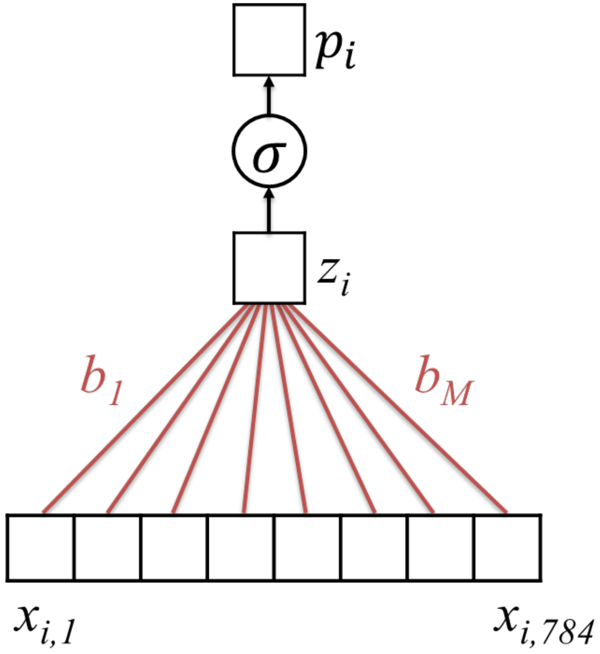
Training set:
28 x 28 images

Vectorize



$p_i = \sigma(b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_M x_{iM})$

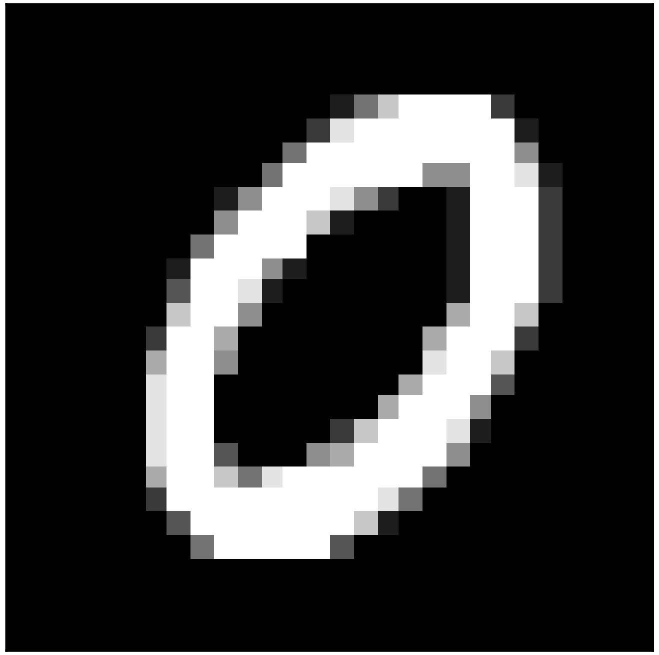
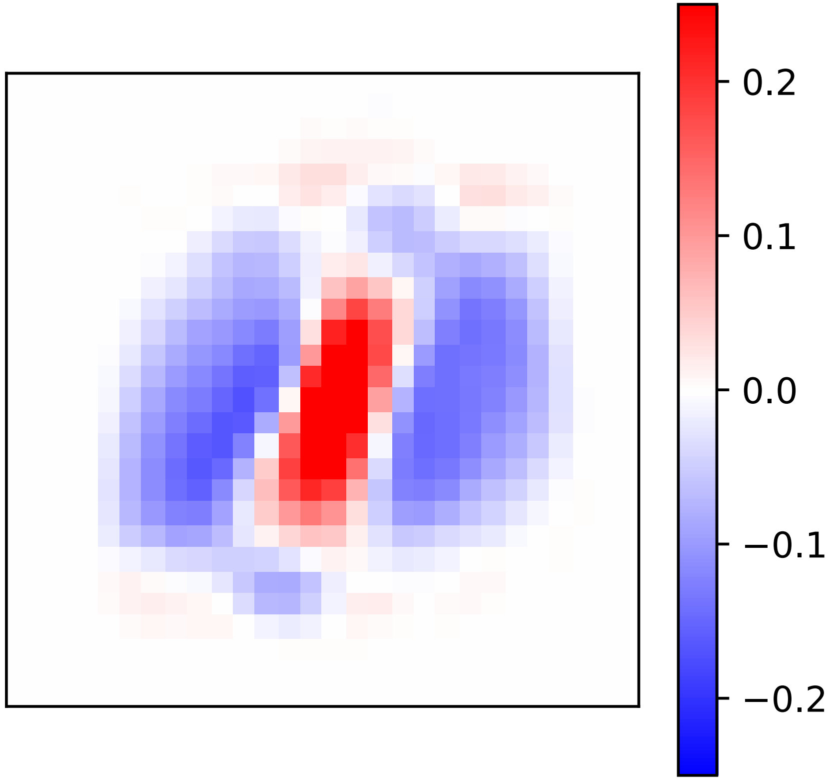
Untrained Logistic Regression Model (or "Network")



$b = (b_0, \dots b_M)$

Trained Model (with learned parameters)

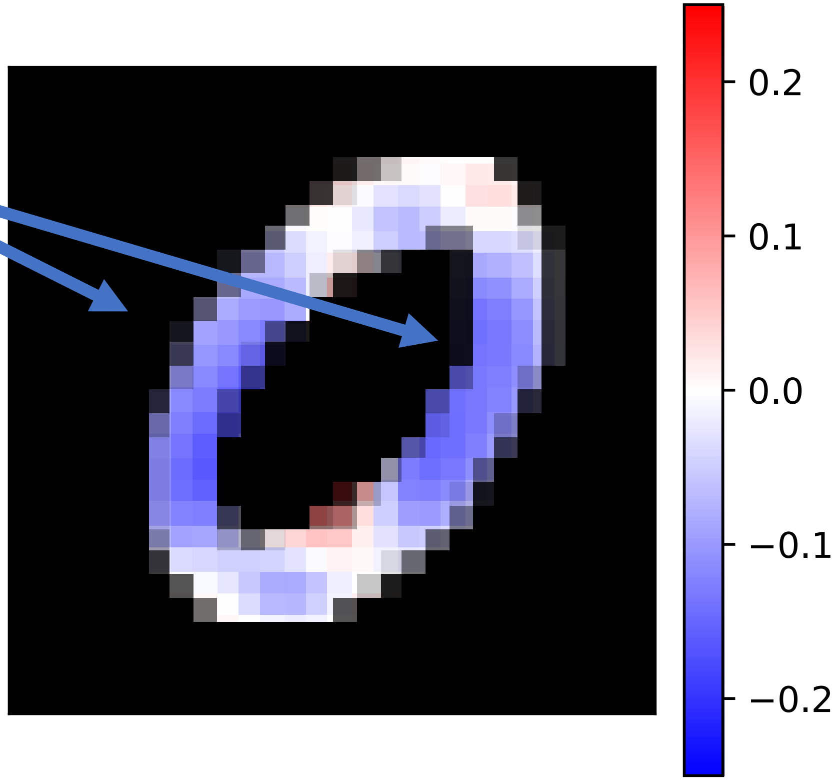
Zooming in on 0/1

$\sigma($  $\odot$ 

$)$

Zooming in on 0/1

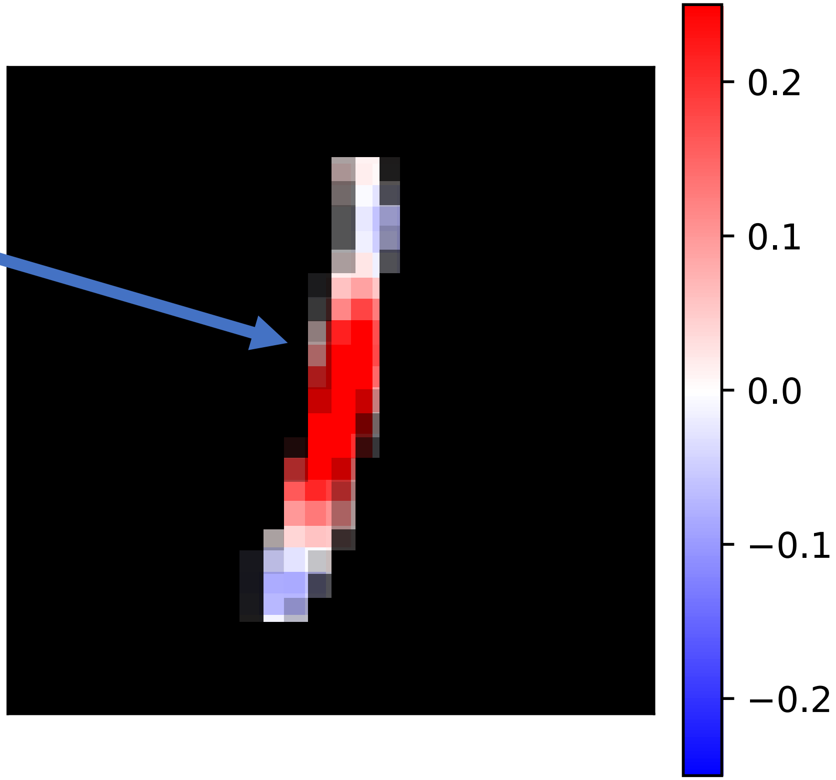
Negative Sections

$\sigma($  $) = 0.006$

Zooming in on 0/1

Positive Section

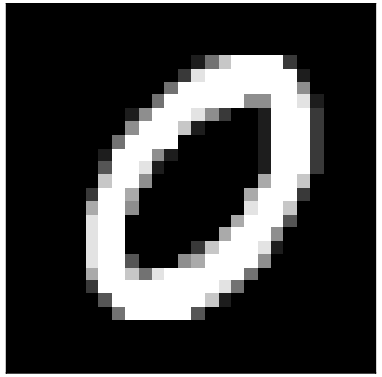
$\sigma($



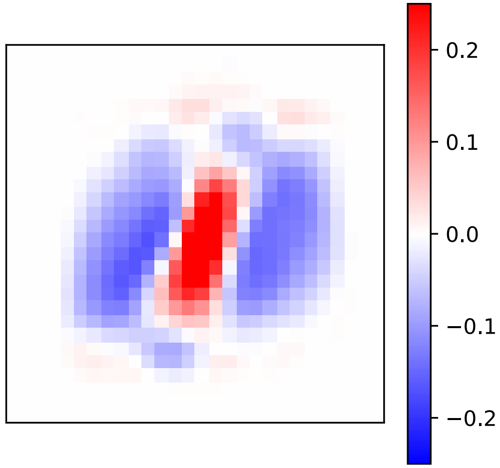
$) = .991$

Learned Weights for 0/1

$\sigma($



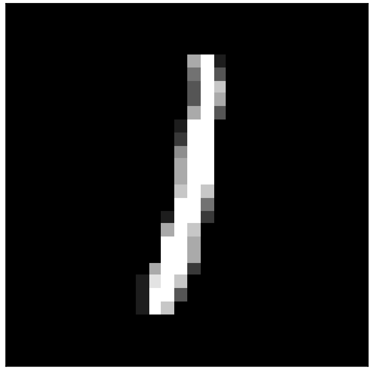
$\odot$



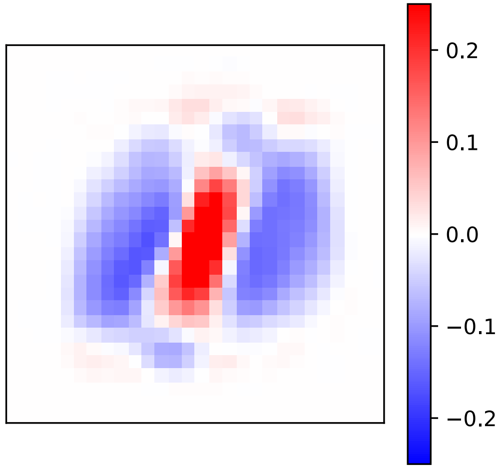
$) = .006$

We think that this is a "zero" (.6% chance it is a "one")

$\sigma($



$\odot$

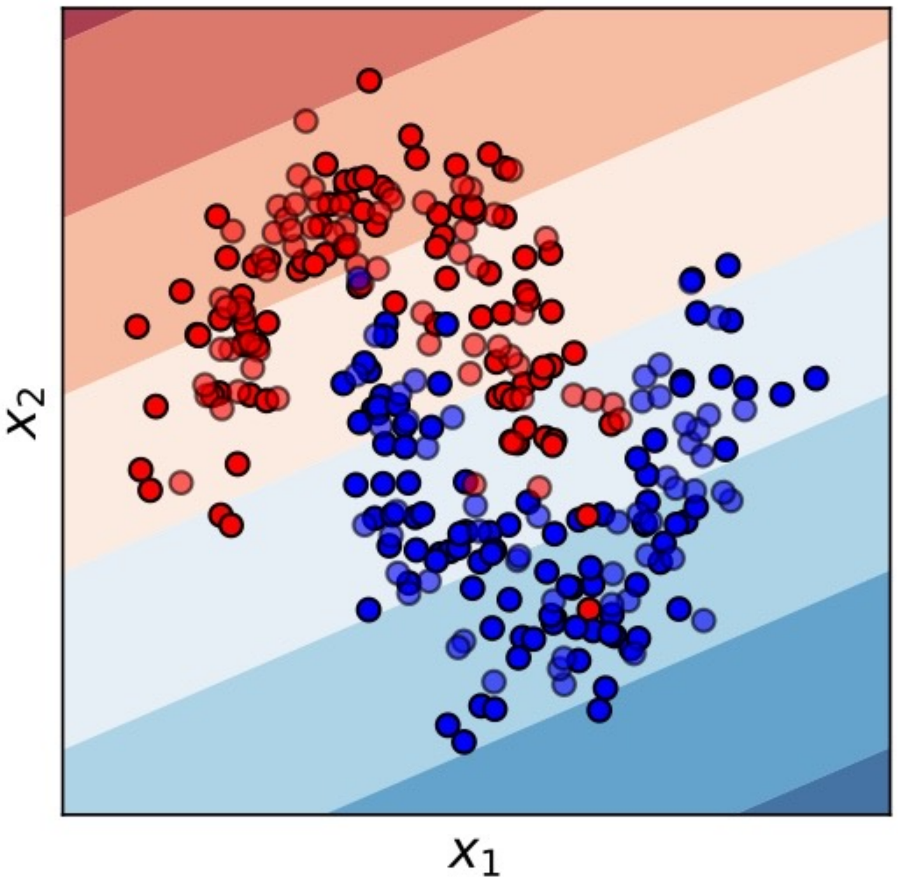


$) = .991$

We think that this is a "one" (99.1% chance it is a "one")

Logistic Regression is a “Linear” Classifier

- A “generalized linear model”
- Can only split data by linear trends



Summary

- Logistic regression is commonly used in machine learning to predict events and/or binary labels. It is simple but often quite effective.
- Logistic regression consists of a linear model coupled with a logistic “link” function that converts predictions to valid probabilities.
- We may view its parameters as a *filter*; when the features and filter are similar (dissimilar), the predicted probability is high (low).
- For many problems, however, we will not want to limit ourselves to a linear decision surface. In the next lecture, we will extend logistic regression to address this limitation.