Classification

CISC 7026: Introduction to Deep Learning

University of Macau

ML

Many problems in ML can be reduced to **regression** or **classification**

ML

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

ML

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

Classification asks which one

ML

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

Classification asks which one

- Is this a dog or muffin?

ML

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

Classification asks which one

- Is this a dog or muffin?
- Will it rain tomorrow? Yes or no?

ML

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

Classification asks which one

- Is this a dog or muffin?
- Will it rain tomorrow? Yes or no?
- What color is this object?

ML

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many
- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

Classification asks which one
- Is this a dog or muffin?
- Will it rain tomorrow? Yes or no?
- What color is this object?

Now let us look at classification

Classification

1. Define an example problem

Classification

1. Define an example problem
2. Primer on probability

Classification

1. Define an example problem
2. Primer on probability
3. Define our machine learning model f

Classification

1. Define an example problem
2. Primer on probability
3. Define our machine learning model f
4. Define a loss function $\mathcal{L}$

Classification

1. Define an example problem
2. Primer on probability
3. Define our machine learning model f
4. Define a loss function $\mathcal{L}$
5. Use $\mathcal{L}$ to learn the parameters θ of f

Does this look familiar?

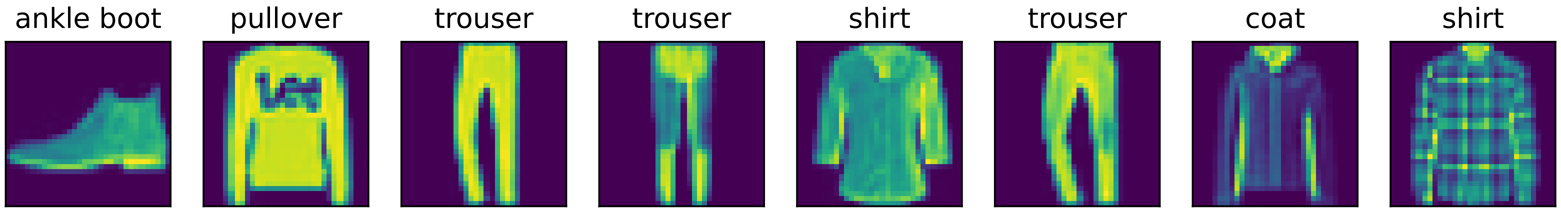
Classification

1. **Define an example problem**
2. Primer on probability
3. Define our machine learning model f
4. Define a loss function $\mathcal{L}$
5. Use $\mathcal{L}$ to learn the parameters θ of f

Task: Given an pictures of clothes, predict their text descriptions

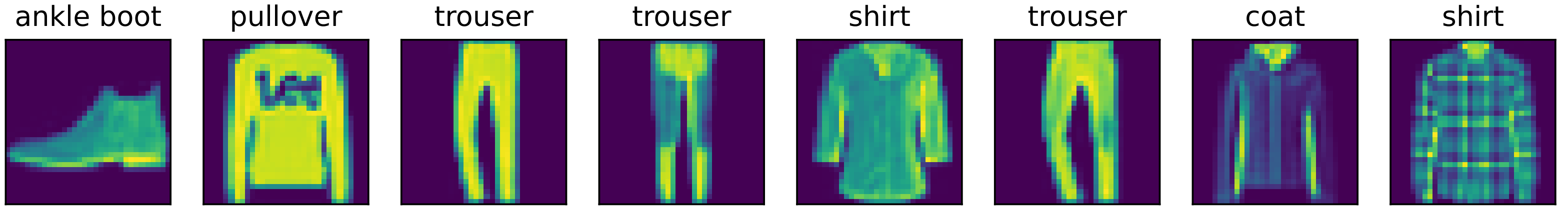
Task: Given an pictures of clothes, predict their text descriptions

X :



Task: Given an pictures of clothes, predict their text descriptions

X :

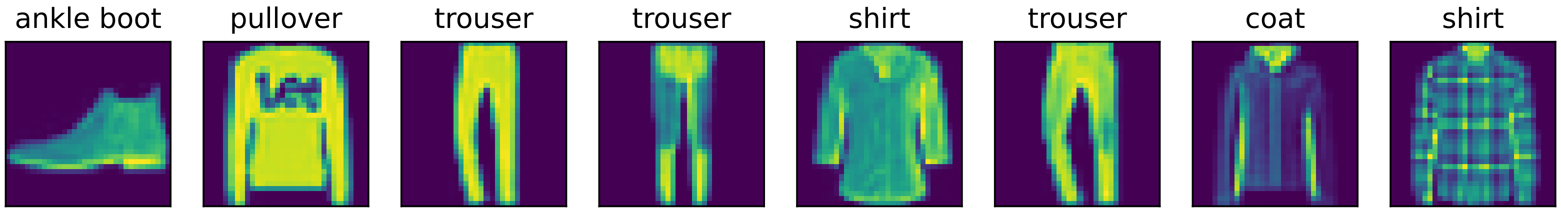


$$Y : \{\text{T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot}\}$$

Approach: Learn the parameters θ that produce **class probabilities**

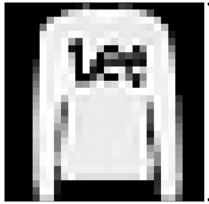
Task: Given an pictures of clothes, predict their text descriptions

X :

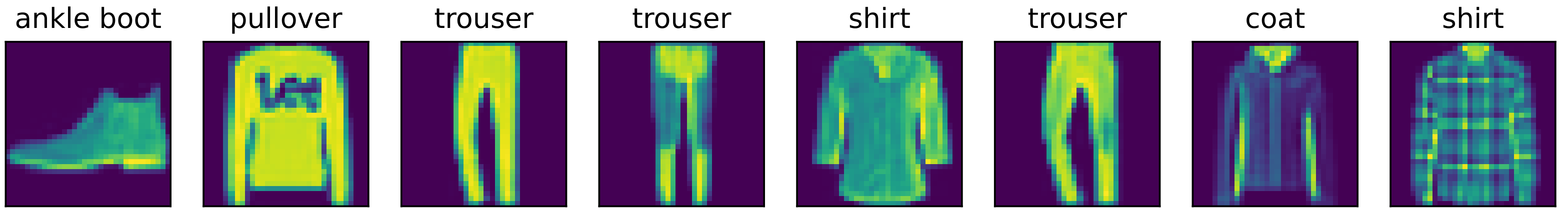


Y : {T-shirt, Trouser, Pullover, Dress, Coat,
Sandal, Shirt, Sneaker, Bag, Ankle boot}

Approach: Learn the parameters θ that produce **class probabilities**

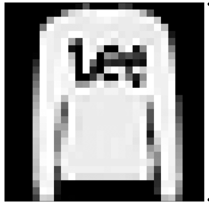
$f(x, \theta) = P(y \mid x) = P\Big(\text{boot} \mid$  $\Big)$

Task: Given an pictures of clothes, predict their text descriptions

X : 

Y : {T-shirt, Trouser, Pullover, Dress, Coat,
Sandal, Shirt, Sneaker, Bag, Ankle boot}

Approach: Learn the parameters θ that produce **class probabilities**

$$f(x, \theta) = P(y \mid x) = P\left(\text{boot} \mid \right.$$  $$\left.\right)$$

Classification

1. **Define an example problem**
2. Primer on probability
3. Define our machine learning model f
4. Define a loss function $\mathcal{L}$
5. Use $\mathcal{L}$ to learn the parameters θ of f

Classification

1. Define an example problem
2. **Primer on probability**
3. Define our machine learning model f
4. Define a loss function $\mathcal{L}$
5. Use $\mathcal{L}$ to learn the parameters θ of f

In probability, we have **experiments** and **outcomes**

In probability, we have **experiments** and **outcomes**

An experiment yields one of many possible outcomes

In probability, we have **experiments** and **outcomes**

An experiment yields one of many possible outcomes

Flip a coin Heads

In probability, we have **experiments** and **outcomes**

An experiment yields one of many possible outcomes

Flip a coin	Heads
Walk outside	Rain

In probability, we have **experiments** and **outcomes**

An experiment yields one of many possible outcomes

Flip a coin	Heads
Walk outside	Rain
Grab clothing from closest	Coat

In probability, we have **experiments** and **outcomes**

An experiment yields one of many possible outcomes

Flip a coin	Heads
Walk outside	Rain
Grab clothing from closest	Coat

The **sample space** S defines all possible outcomes for an experiment

The **sample space** S defines all possible outcomes for an experiment

Flip a coin $\quad\quad$ $S = \{\text{heads}, \text{tails}\}$

The **sample space** S defines all possible outcomes for an experiment

Flip a coin	$S = \{\text{heads}, \text{tails}\}$
Walk outside	$S = \{\text{rain}, \text{sun}, \text{wind}, \text{cloud}\}$

The **sample space** S defines all possible outcomes for an experiment

Flip a coin	$S = \{\text{heads}, \text{tails}\}$
Walk outside	$S = \{\text{rain}, \text{sun}, \text{wind}, \text{cloud}\}$
Grab clothing from closet	$S = \{\text{T-shirt}, \text{Trouser}, \text{Pullover}, \text{Dress}, \text{Coat}, \text{Sandal}, \text{Shirt}, \text{Sneaker}, \text{Bag}, \text{Ankle boot}\}$

An **event** is a subset of the sample space

An **event** is a subset of the sample space

Flip a coin $\{\text{heads}\}$

An **event** is a subset of the sample space

Flip a coin	$\{\text{heads}\}$
Walk outside	$\{\text{rain}, \text{cloud}, \text{wind}\}$

An **event** is a subset of the sample space

Flip a coin	$\{\text{heads}\}$
Walk outside	$\{\text{rain}, \text{cloud}, \text{wind}\}$
Grab clothing from closet	$\{\text{Sneaker}\}$

The **probability** measures how likely an event is to occur

The **probability** measures how likely an event is to occur

The probability must be between 0 (never occurs) and 1 (always occurs)

The **probability** measures how likely an event is to occur

The probability must be between 0 (never occurs) and 1 (always occurs)

$$0 \leq P(A) \leq 1; \quad \forall A \in S$$

The **probability** measures how likely an event is to occur

The probability must be between 0 (never occurs) and 1 (always occurs)

$$0 \leq P(A) \leq 1; \quad \forall A \in S$$

The probabilities must sum to one

The **probability** measures how likely an event is to occur

The probability must be between 0 (never occurs) and 1 (always occurs)

$$0 \leq P(A) \leq 1; \quad \forall A \in S$$

The probabilities must sum to one

$$\sum_{A \in S} P(A) = 1$$

Flip a coin

$$P(\text{Heads}) = \frac{1}{2}$$

Flip a coin $\quad P(\text{Heads}) = \frac{1}{2}$

Walk outside $\quad P(\text{Rain}) = 0.05$

Flip a coin	$P(\text{Heads}) = \frac{1}{2}$
Walk outside	$P(\text{Rain}) = 0.05$
Grab clothing from closet	$P(\text{Dress}) = 0$

Flip a coin $P(\text{Heads}) = \frac{1}{2}$

Walk outside $P(\text{Rain}) = 0.05$

Grab clothing from closet $P(\text{Dress}) = 0$

For **mutually exclusive** events, we can sum together probabilities

For **mutually exclusive** events, we can sum together probabilities

$$P(A \cup B) = P(A) + P(B)$$

For **mutually exclusive** events, we can sum together probabilities

$$P(A \cup B) = P(A) + P(B)$$

Grab clothing from closet

$$P(\text{Shirt}) = 0.1, P(\text{Bag}) = 0.05$$

$$P(\text{Shirt} \cup \text{Bag}) = 0.15$$

Be careful!

For **mutually exclusive** events, we can sum together probabilities

$$P(A \cup B) = P(A) + P(B)$$

Grab clothing from closet

$$P(\text{Shirt}) = 0.1, P(\text{Bag}) = 0.05$$

$$P(\text{Shirt} \cup \text{Bag}) = 0.15$$

Be careful!

Walk outside

$$P(\text{Rain}) = 0.05, P(\text{Sun}) = 0.4$$
$$P(\text{Rain} \cup \text{Sun}) \neq 0.45$$

If events are **independent**, we can multiply their probabilities

If events are **independent**, we can multiply their probabilities

$$P(A \cap B) = P(A) \cdot P(B)$$

If events are **independent**, we can multiply their probabilities

$$P(A \cap B) = P(A) \cdot P(B)$$

Be careful!

If events are **independent**, we can multiply their probabilities

$$P(A \cap B) = P(A) \cdot P(B)$$

Be careful!

Flip a coin

$$P(\text{Heads}) = 0.5, P(\text{Tails}) = 0.5$$
$$P(\text{Heads} \cap \text{Tails}) \neq 0.25$$

If events are **independent**, we can multiply their probabilities

$$P(A \cap B) = P(A) \cdot P(B)$$

Be careful!

Flip a coin

$$P(\text{Heads}) = 0.5, P(\text{Tails}) = 0.5$$
$$P(\text{Heads} \cap \text{Tails}) \neq 0.25$$

Events can be **conditionally dependent**

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Events can be **conditionally dependent**

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Flip a coin

$$P(\text{Heads} \cap \text{Tails}) = 0$$

$$P(\text{Tails}) = 0.5$$

$$P(\text{Heads} \mid \text{Tails}) = \frac{0}{0.5} = 0$$

Events can be **conditionally dependent**

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Flip a coin

$$P(\text{Heads} \cap \text{Tails}) = 0$$

$$P(\text{Tails}) = 0.5$$

$$P(\text{Heads} \mid \text{Tails}) = \frac{0}{0.5} = 0$$

Walk outside

$$P(\text{Rain} \cap \text{Cloud}) = 0.2$$

$$P(\text{Cloud}) = 0.4$$

$$P(\text{Rain} \mid \text{Cloud}) = \frac{0.2}{0.4} = 0.5$$

TODO: Random variable, distribution

Classification

1. Define an example problem
2. **Primer on probability**
3. Define our machine learning model f
4. Define a loss function $\mathcal{L}$
5. Use $\mathcal{L}$ to learn the parameters θ of f

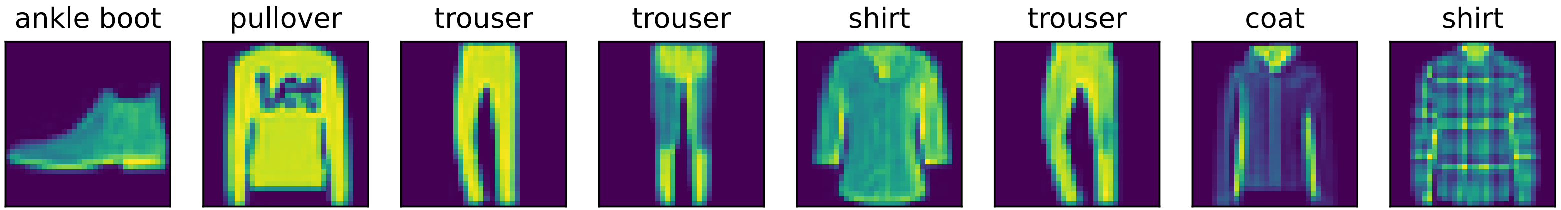
Relax

Back to the problem...

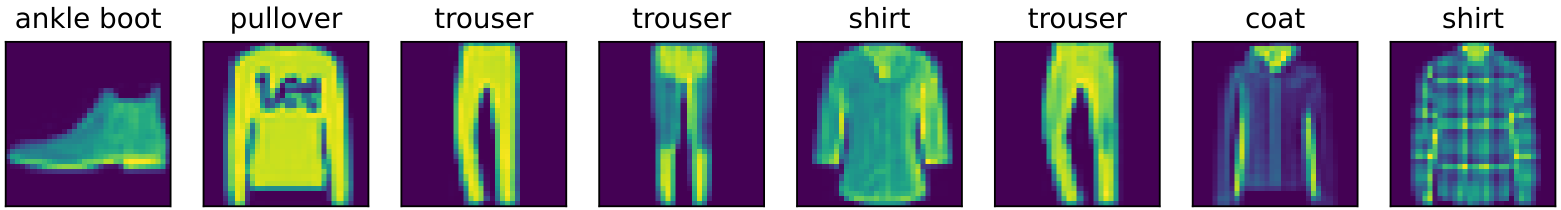
Task: Given a picture of clothes, predict the text description

Task: Given a picture of clothes, predict the text description

X :



Task: Given a picture of clothes, predict the text description

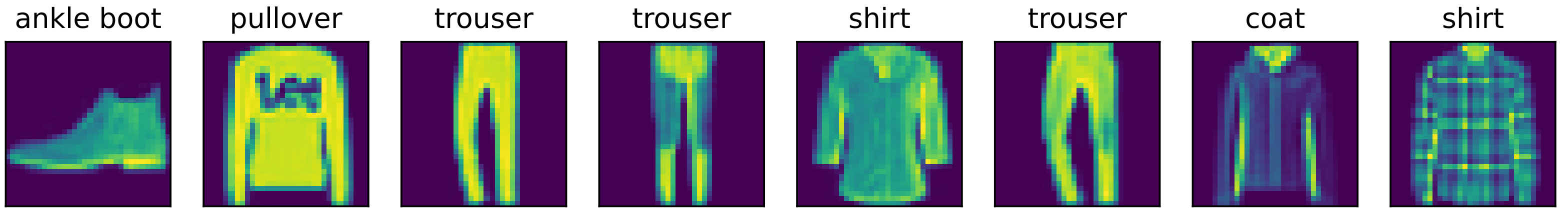
X : 

Y : {T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot}

Approach: Learn the parameters θ that produce **event probabilities**

Task: Given a picture of clothes, predict the text description

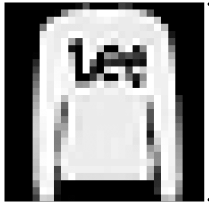
X :



Y : {T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot}

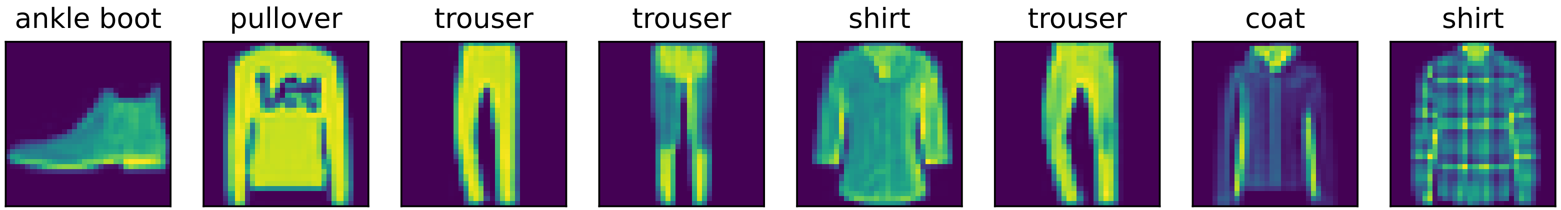
Approach: Learn the parameters θ that produce **event probabilities**

$$f(x, \theta) = P(y \mid x) = P\left(\text{boot} \mid \right)$$



Task: Given a picture of clothes, predict the text description

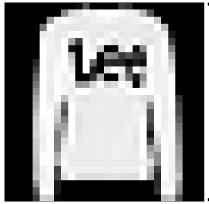
X :



Y : {T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot}

Approach: Learn the parameters θ that produce **event probabilities**

$$f(x, \theta) = P(y \mid x) = P\left(\text{boot} \mid \right.$$



$$\left. \right)$$

Classification

1. Define an example problem
2. Primer on probability
3. **Define our machine learning model f**
4. Define a loss function $\mathcal{L}$
5. Use $\mathcal{L}$ to learn the parameters θ of f

We will again start with a linear model

We will again start with a linear model

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} W \\ b \end{bmatrix}\right) = Wx + b$$

We will again start with a linear model

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} W \\ b \end{bmatrix}\right) = Wx + b$$

However, the probabilities must sum to one which is nonlinear...

We will again start with a linear model

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} W \\ b \end{bmatrix}\right) = Wx + b$$

However, the probabilities must sum to one which is nonlinear...

We introduce the **softmax** operator to ensure all probabilities sum to 1

We will again start with a linear model

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} W \\ b \end{bmatrix}\right) = W x + b$$

However, the probabilities must sum to one which is nonlinear...

We introduce the **softmax** operator to ensure all probabilities sum to 1

The softmax operator is heavily used in machine learning, especially where probabilities pop up

The softmax operator is heavily used in machine learning, especially where probabilities pop up

It maps a vector of real numbers to a vector of probabilities

The softmax operator is heavily used in machine learning, especially where probabilities pop up

It maps a vector of real numbers to a vector of probabilities

$$\text{softmax} : \mathbb{R}^n \mapsto \Delta^{n-1}$$

The softmax operator is heavily used in machine learning, especially where probabilities pop up

It maps a vector of real numbers to a vector of probabilities

$$\text{softmax} : \mathbb{R}^n \mapsto \Delta^{n-1}$$

$$\Delta^{n-1} = \left\{ \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} \middle| \sum_{i=1}^{n} p_i = 1 \right\}$$

The simplex operator Δ just means that the outputs of softmax sum to 1

The softmax operator is heavily used in machine learning, especially where probabilities pop up

It maps a vector of real numbers to a vector of probabilities

$$\mathrm{softmax} : \mathbb{R}^n \mapsto \Delta^{n-1}$$

$$\Delta^{n-1} = \left\{ \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} \middle| \sum_{i=1}^{n} p_i = 1 \right\}$$

The simplex operator Δ just means that the outputs of softmax sum to 1

$$\text{softmax}\left(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} = \begin{bmatrix} \frac{e^{x_1}}{e^{x_1}+e^{x_2}+\ldots e^{x_n}} \\ \frac{e^{x_2}}{e^{x_1}+e^{x_2}+\ldots e^{x_n}} \\ \vdots \\ \frac{e^{x_n}}{e^{x_1}+e^{x_2}+\ldots e^{x_n}} \end{bmatrix}$$

Using the softmax function, we learn the probability for each class/event

$$f(\boldsymbol{x}, \boldsymbol{\theta}) : \mathbb{Z}^n \mapsto \Delta^{|Y| - 1}$$

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} W \\ b \end{bmatrix}\right) = \text{softmax}(Wx + b)$$

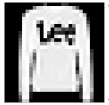
Using the softmax function, we learn the probability for each class/event

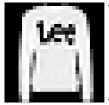
$$f(\boldsymbol{x}, \boldsymbol{\theta}) : \mathbb{Z}^n \mapsto \Delta^{|Y| - 1}$$

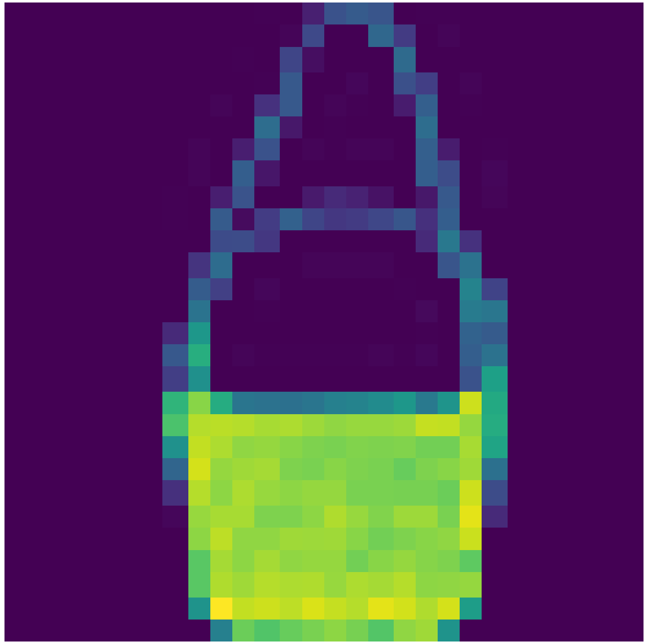
$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} W \\ b \end{bmatrix}\right) = \text{softmax}(Wx + b)$$

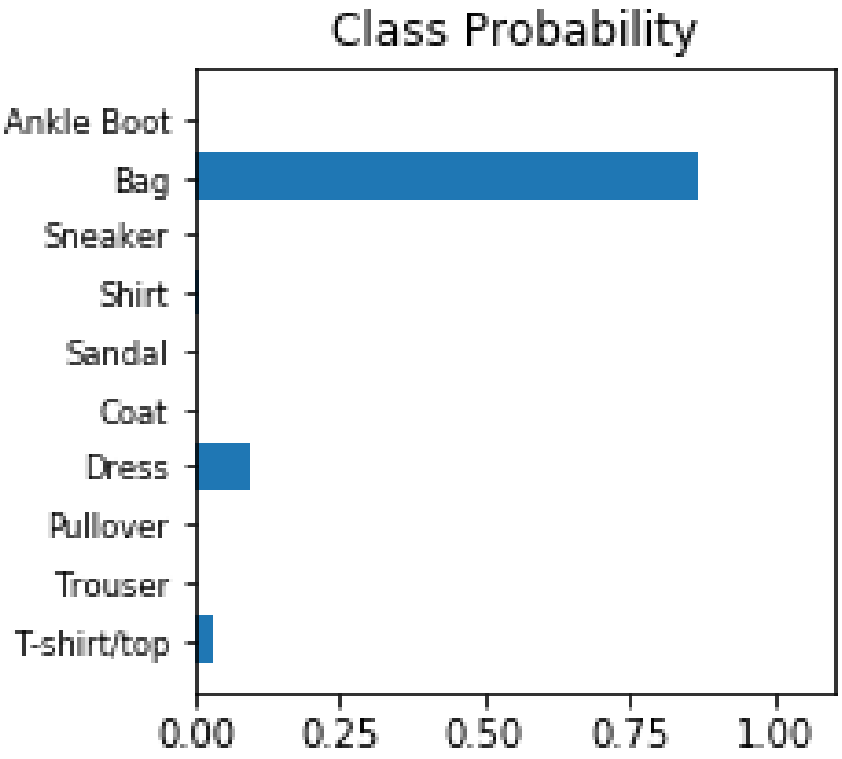
Each output dimension determines a specific class/event probability

$$f(x, \boldsymbol{\theta}) = \begin{bmatrix} P\left(\text{Ankle boot} \mid \quad \right) \\ P\left(\text{Bag} \mid \quad \right) \\ \vdots \end{bmatrix}$$



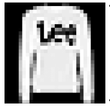


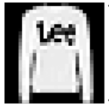



Class Probability
Ankle Boot
Bag
Sneaker
Shirt
Sandal
Coat
Dress
Pullover
Trouser
T-shirt/top
0.00
0.25
0.50
0.75
1.00

Question: Why do we output probabilities instead of just a one-hot vector

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \begin{bmatrix} P(\text{Shirt} \mid \quad) \\ P(\text{Bag} \mid \quad) \end{bmatrix}$$





$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Answer: We do not always know the correct answer. There is always uncertainty.

Relax

Classification

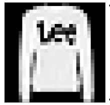
1. Define an example problem
2. Primer on probability
3. **Define our machine learning model** f
4. Define a loss function $\mathcal{L}$
5. Use $\mathcal{L}$ to learn the parameters θ of f

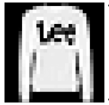
Classification

1. Define an example problem
2. Primer on probability
3. Define our machine learning model f
4. **Define a loss function $\mathcal{L}$**
5. Use $\mathcal{L}$ to learn the parameters θ of f

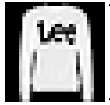
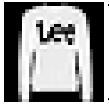
We use squared error for regression, what about classification?

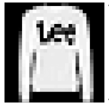
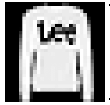
We use squared error for regression, what about classification?

$f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \Bigg[\; P\big(\text{Shirt} \mid$  $\big)$

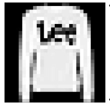
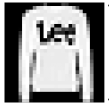
$P\big(\text{Bag} \mid$  $\big) \;\Bigg] = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}$

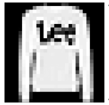
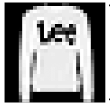
We use squared error for regression, what about classification?

$f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \begin{bmatrix} P(\text{Shirt} \mid$  $) \\ P(\text{Bag} \mid$  $) \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}$

$\boldsymbol{y}_i = \begin{bmatrix} P(\text{Shirt} \mid$  $) \\ P(\text{Bag} \mid$  $) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

We use squared error for regression, what about classification?

$f(\boldsymbol{x}_i, \boldsymbol{\theta}) =$ [$P\big(\text{Shirt} \mid$  $\big)$; $P\big(\text{Bag} \mid$  $\big)$] $= \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}$

$\boldsymbol{y}_i =$ [$P\big(\text{Shirt} \mid$  $\big)$; $P\big(\text{Bag} \mid$  $\big)$] $= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$$f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}, \boldsymbol{y}_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}, \boldsymbol{y}_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

We could compute the sum of square errors

$$f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}, \boldsymbol{y}_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

We could compute the sum of square errors

$$(0.6 - 1)^2 + (0.4 - 0)^2$$

$$f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}, \boldsymbol{y}_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

We could compute the sum of square errors

$$(0.6 - 1)^2 + (0.4 - 0)^2$$

In practice, this does not work very well

$$f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}, \boldsymbol{y}_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

We could compute the sum of square errors

$$(0.6 - 1)^2 + (0.4 - 0)^2$$

In practice, this does not work very well

Instead, we use the **cross-entropy loss**

$$f(\boldsymbol{x}_i, \boldsymbol{\theta}) = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}, \boldsymbol{y}_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

We could compute the sum of square errors

$$(0.6 - 1)^2 + (0.4 - 0)^2$$

In practice, this does not work very well

Instead, we use the **cross-entropy loss**

Let us derive it

We can model $f(\boldsymbol{x}, \boldsymbol{\theta})$ and $\boldsymbol{y}$ as probability distributions

We can model $f(\boldsymbol{x}, \boldsymbol{\theta})$ and $\boldsymbol{y}$ as probability distributions

How do we measure the difference between probability distributions?

We can model $f(\boldsymbol{x}, \boldsymbol{\theta})$ and $\boldsymbol{y}$ as probability distributions

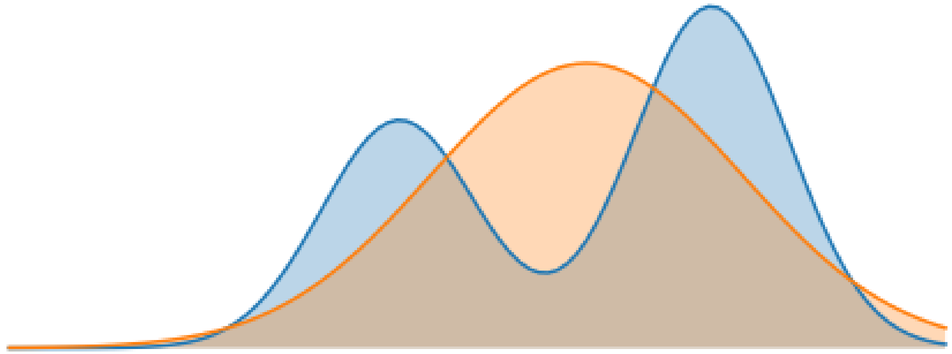
How do we measure the difference between probability distributions?

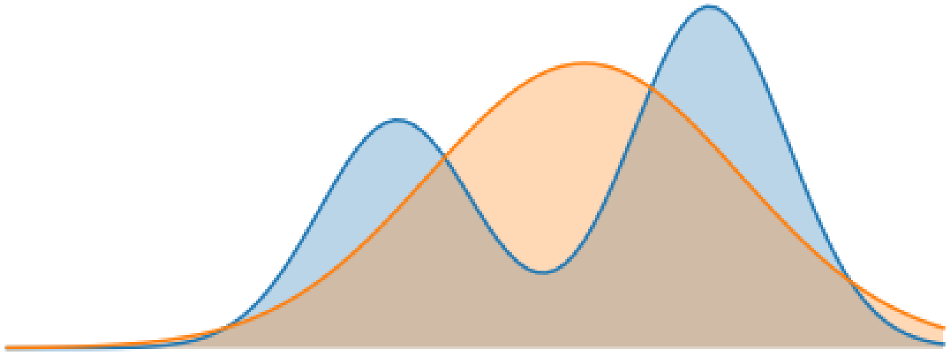
We use the **Kullback-Leibler Divergence (KL)**

We can model $f(\boldsymbol{x}, \boldsymbol{\theta})$ and $\boldsymbol{y}$ as probability distributions

How do we measure the difference between probability distributions?

We use the **Kullback-Leibler Divergence (KL)**





$$\mathrm{KL}(P, Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

TODO: Should be $f(y_i \mid x, \theta)$

$\mathrm{KL}(P, Q) =$	$\sum_i P(i) \log \frac{P(i)}{Q(i)}$	KL divergence

TODO: Should be $f(y_i \mid x, \theta)$

$$\mathrm{KL}(P, Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$ KL divergence

$$\mathrm{KL}(P(\boldsymbol{y} \mid \boldsymbol{x}), f(\boldsymbol{x}, \boldsymbol{\theta})) = \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log \frac{P(y \mid \boldsymbol{x})}{f(\boldsymbol{x}, \boldsymbol{\theta})}$$ Plug in f, y

TODO: Should be $f(y_i \mid x, \theta)$

$$\mathrm{KL}(P, Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$ KL divergence

$$\mathrm{KL}(P(\boldsymbol{y} \mid \boldsymbol{x}), f(\boldsymbol{x}, \boldsymbol{\theta})) = \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log \frac{P(y \mid \boldsymbol{x})}{f(\boldsymbol{x}, \boldsymbol{\theta})}$$ Plug in f, y

$$\mathrm{KL}(P(\boldsymbol{y} \mid \boldsymbol{x}), f(\boldsymbol{x}, \boldsymbol{\theta})) = \sum_{y \in Y} P(y \mid \boldsymbol{x})[\log P(y \mid \boldsymbol{x}) - \log f(\boldsymbol{x}, \boldsymbol{\theta})]$$ Log rule

TODO: Should be $f(y_i \mid x, \theta)$

$$\mathrm{KL}(P, Q) = \quad \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad \text{KL divergence}$$

$$\mathrm{KL}(P(\boldsymbol{y} \mid \boldsymbol{x}), f(\boldsymbol{x}, \boldsymbol{\theta})) = \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log \frac{P(y \mid \boldsymbol{x})}{f(\boldsymbol{x}, \boldsymbol{\theta})} \quad \text{Plug in } f, y$$

$$\mathrm{KL}(P(\boldsymbol{y} \mid \boldsymbol{x}), f(\boldsymbol{x}, \boldsymbol{\theta})) = \sum_{y \in Y} P(y \mid \boldsymbol{x})[\log P(y \mid \boldsymbol{x}) - \log f(\boldsymbol{x}, \boldsymbol{\theta})] \quad \text{Log rule}$$

$$= \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log P(y \mid \boldsymbol{x}) - \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \quad \text{Split sum}$$

TODO: Should be $f(y_i \mid x, \theta)$

$$\mathrm{KL}(P, Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \qquad \text{KL divergence}$$

$$\mathrm{KL}(P(\boldsymbol{y} \mid \boldsymbol{x}), f(\boldsymbol{x}, \boldsymbol{\theta})) = \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log \frac{P(y \mid \boldsymbol{x})}{f(\boldsymbol{x}, \boldsymbol{\theta})} \qquad \text{Plug in } f, y$$

$$\mathrm{KL}(P(\boldsymbol{y} \mid \boldsymbol{x}), f(\boldsymbol{x}, \boldsymbol{\theta})) = \sum_{y \in Y} P(y \mid \boldsymbol{x})[\log P(y \mid \boldsymbol{x}) - \log f(\boldsymbol{x}, \boldsymbol{\theta})] \qquad \text{Log rule}$$

$$= \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log P(y \mid \boldsymbol{x}) - \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \qquad \text{Split sum}$$

$$= -\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \qquad \text{First term constant}$$

TODO: Should be $f(y_i \mid x, \theta)$

$$\mathrm{KL}(P, Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad \text{KL divergence}$$

$$\mathrm{KL}(P(\boldsymbol{y} \mid \boldsymbol{x}), f(\boldsymbol{x}, \boldsymbol{\theta})) = \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log \frac{P(y \mid \boldsymbol{x})}{f(\boldsymbol{x}, \boldsymbol{\theta})} \quad \text{Plug in } f, y$$

$$\mathrm{KL}(P(\boldsymbol{y} \mid \boldsymbol{x}), f(\boldsymbol{x}, \boldsymbol{\theta})) = \sum_{y \in Y} P(y \mid \boldsymbol{x})[\log P(y \mid \boldsymbol{x}) - \log f(\boldsymbol{x}, \boldsymbol{\theta})] \quad \text{Log rule}$$

$$= \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log P(y \mid \boldsymbol{x}) - \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \quad \text{Split sum}$$

$$= -\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \quad \text{First term constant}$$

This is the cross-entropy loss!

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = -\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta})$$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = -\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta})$$

By minimizing the loss, we make $f(\boldsymbol{x}, \boldsymbol{\theta})$ output the same probability distribution as $\boldsymbol{y}$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = -\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta})$$

By minimizing the loss, we make $f(\boldsymbol{x}, \boldsymbol{\theta})$ output the same probability distribution as $\boldsymbol{y}$

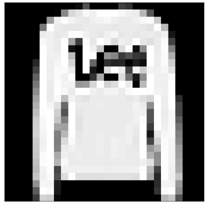
$$\min_{\theta} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \min_{\theta} \left[-\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \right]$$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = -\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta})$$

By minimizing the loss, we make $f(\boldsymbol{x}, \boldsymbol{\theta})$ output the same probability distribution as $\boldsymbol{y}$

$$\min_{\theta} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \min_{\theta} \left[-\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \right]$$

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = P(\boldsymbol{y} \mid \boldsymbol{x}) = P\left(\begin{bmatrix} \text{boot} \\ \text{dress} \\ \vdots \end{bmatrix} \middle| \right.$$



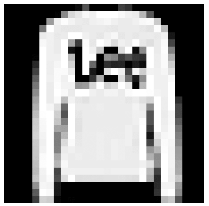
$$\left. \right)$$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = -\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta})$$

By minimizing the loss, we make $f(\boldsymbol{x}, \boldsymbol{\theta})$ output the same probability distribution as $\boldsymbol{y}$

$$\min_{\theta} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \min_{\theta} \left[-\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \right]$$

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = P(\boldsymbol{y} \mid \boldsymbol{x}) = P\left(\begin{bmatrix} \text{boot} \\ \text{dress} \\ \vdots \end{bmatrix} \middle| \right.$$



$$\left. \right)$$

Our loss was just for a single image

Our loss was just for a single image

$$\min_{\theta} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \min_{\theta} \left[-\sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \right]$$

Our loss was just for a single image

$$\min_{\theta} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \min_{\theta} \left[- \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \right]$$

Find the parameters over all images

Our loss was just for a single image

$$\min_{\theta} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \min_{\theta} \left[- \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \right]$$

Find the parameters over all images

$$\min_{\theta} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \min_{\theta} \left[- \sum_{i=1}^{n} \sum_{y_i \in Y} P(y_i \mid \boldsymbol{x}_i) \log f(\boldsymbol{x}_i, \boldsymbol{\theta}) \right]$$

Our loss was just for a single image

$$\min_\theta \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \min_\theta \left[- \sum_{y \in Y} P(y \mid \boldsymbol{x}) \log f(\boldsymbol{x}, \boldsymbol{\theta}) \right]$$

Find the parameters over all images

$$\min_\theta \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \min_\theta \left[- \sum_{i=1}^{n} \sum_{y_i \in Y} P(y_i \mid \boldsymbol{x}_i) \log f(\boldsymbol{x}_i, \boldsymbol{\theta}) \right]$$

Classification

1. Define an example problem
2. Primer on probability
3. Define our machine learning model f
4. **Define a loss function $\mathcal{L}$**
5. Use $\mathcal{L}$ to learn the parameters θ of f

Relax

Classification

1. Define an example problem
2. Primer on probability
3. Define our machine learning model f
4. Define a loss function $\mathcal{L}$
5. **Use $\mathcal{L}$ to learn the parameters θ of f**

Unlike linear regression, we use a softmax to model probabilities

Unlike linear regression, we use a softmax to model probabilities

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} W \\ b \end{bmatrix}\right) = \operatorname{softmax}(Wx + b)$$

Unlike linear regression, we use a softmax to model probabilities

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} W \\ b \end{bmatrix}\right) = \operatorname{softmax}(W x + b)$$

There is no closed-form solution!

Unlike linear regression, we use a softmax to model probabilities

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} W \\ b \end{bmatrix}\right) = \mathrm{softmax}(W x + b)$$

There is no closed-form solution!

We need to use iterative solvers to find theta

Unlike linear regression, we use a softmax to model probabilities

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} W \\ b \end{bmatrix}\right) = \text{softmax}(W x + b)$$

There is no closed-form solution!

We need to use iterative solvers to find theta

We will come back to this when discussing neural networks