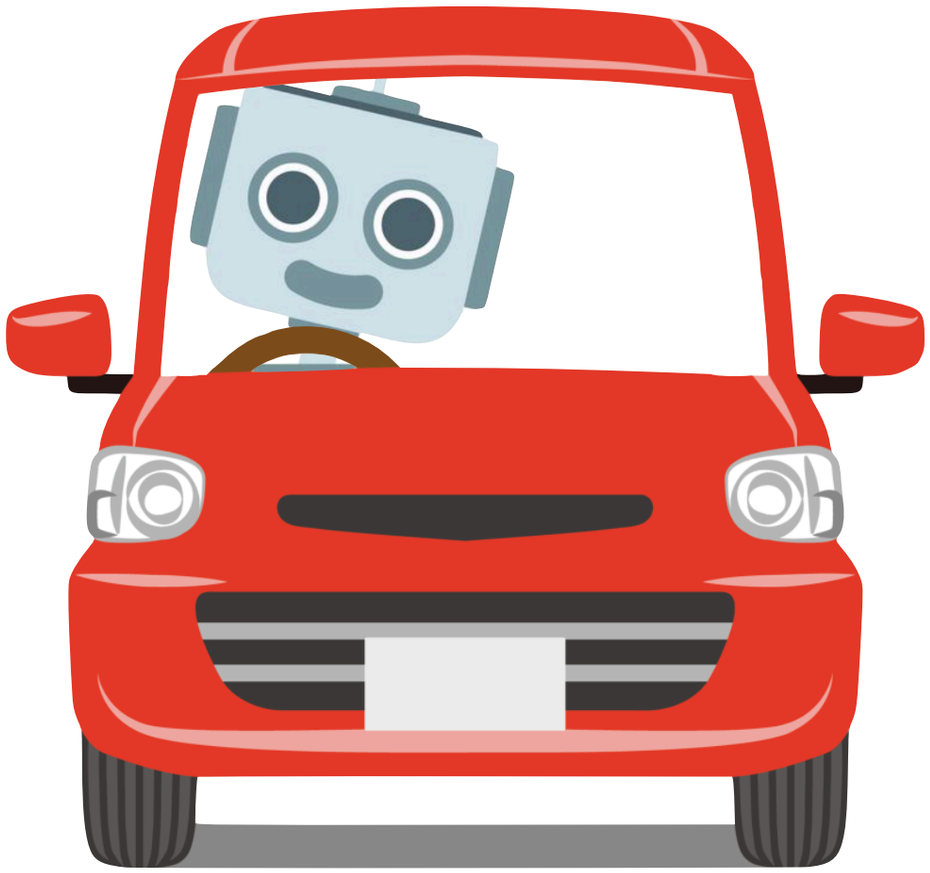
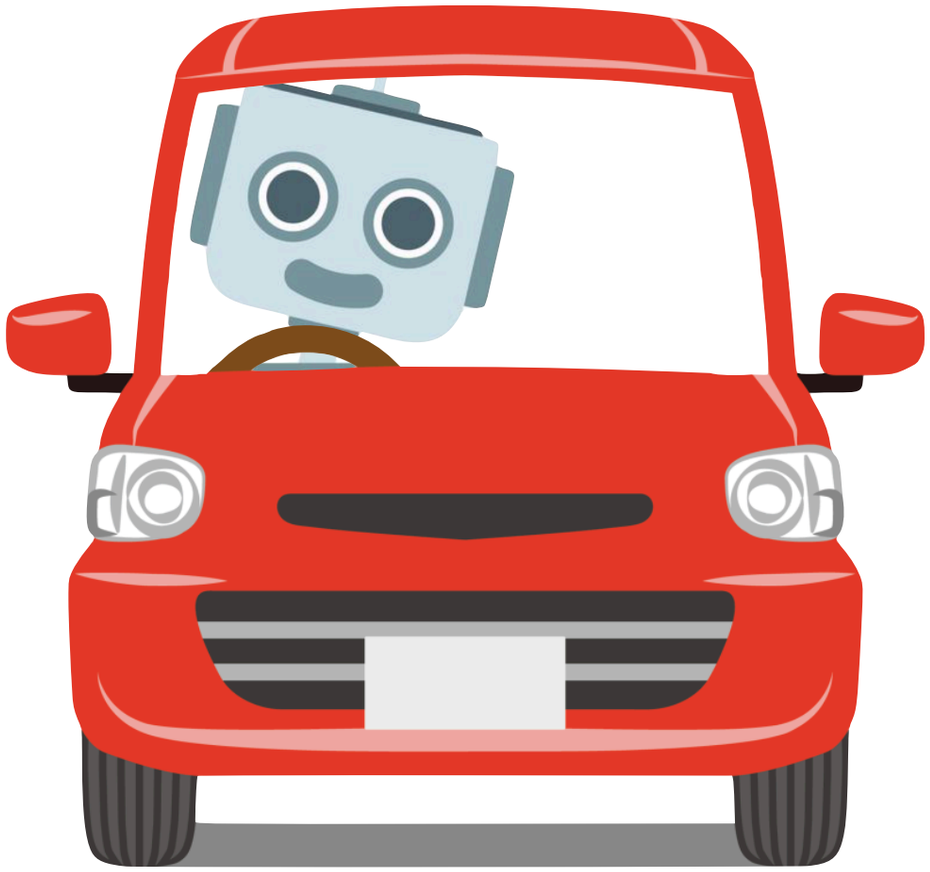
Learning to Reject and *Learning to Defer to an Expert*

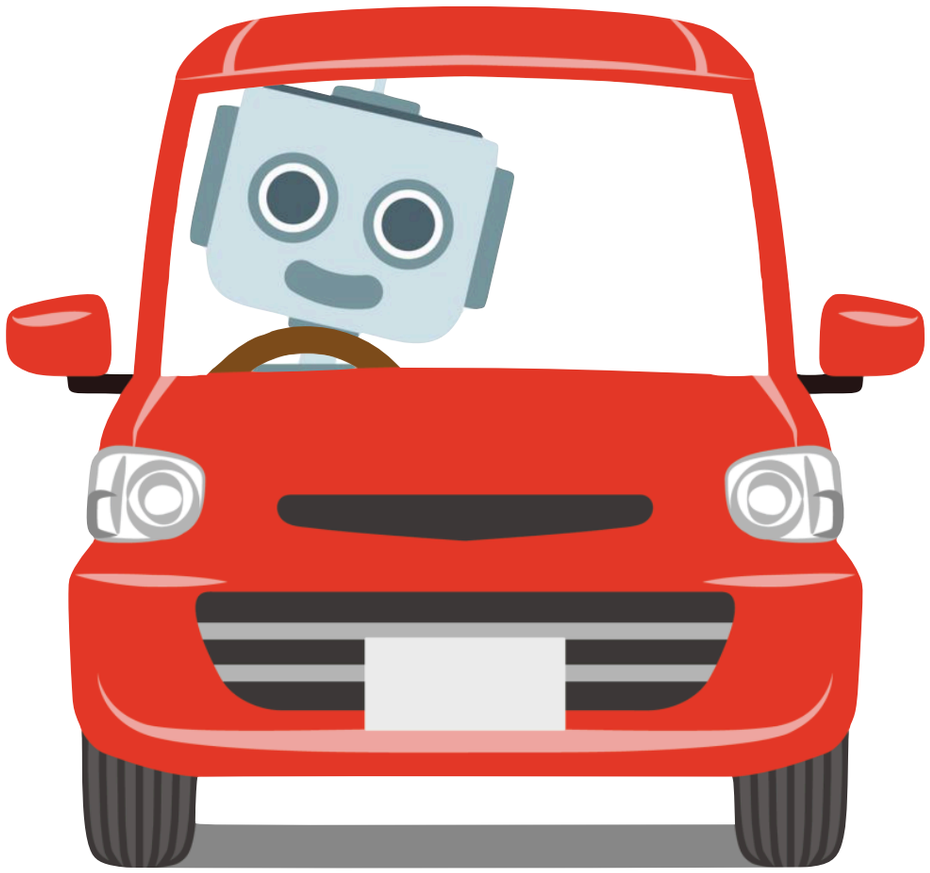
Eric Nalisnick

Johns Hopkins University

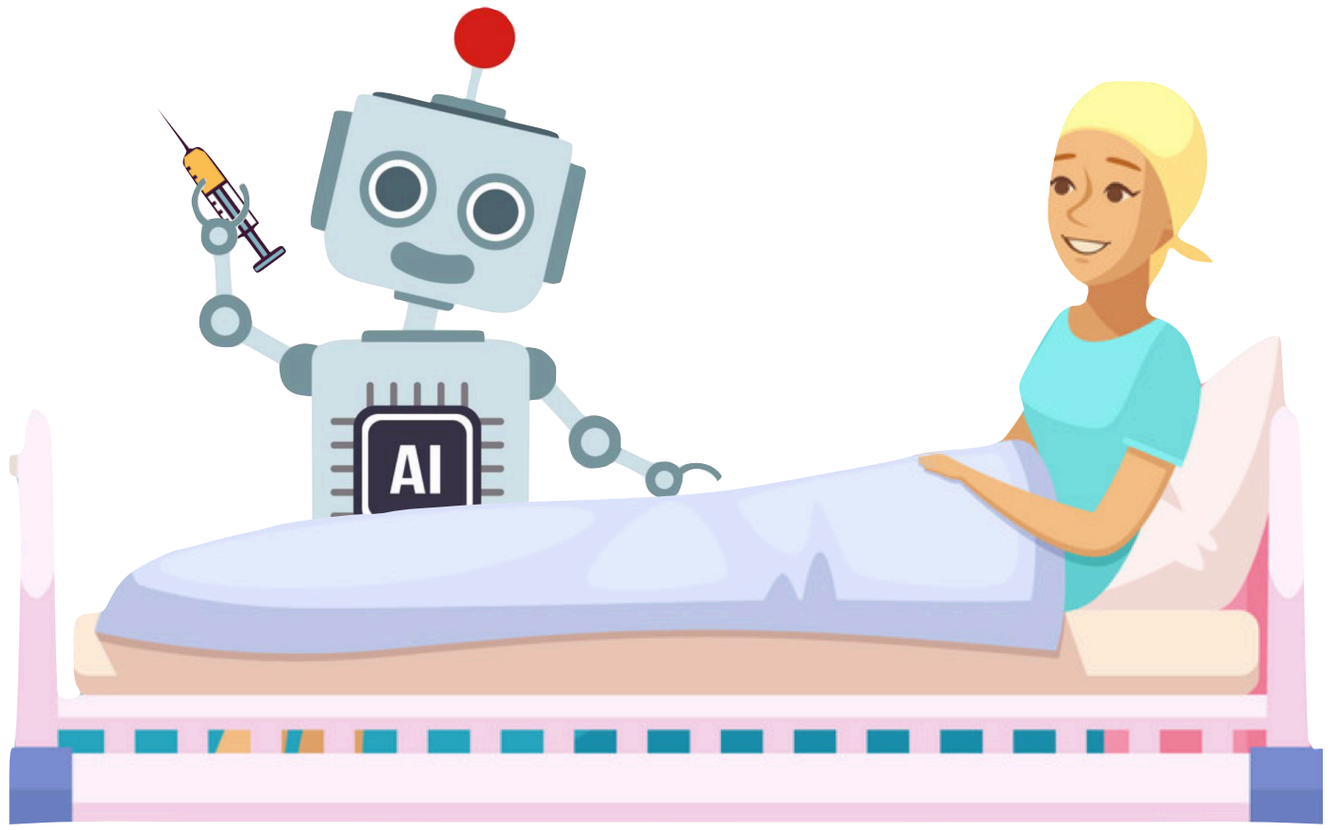


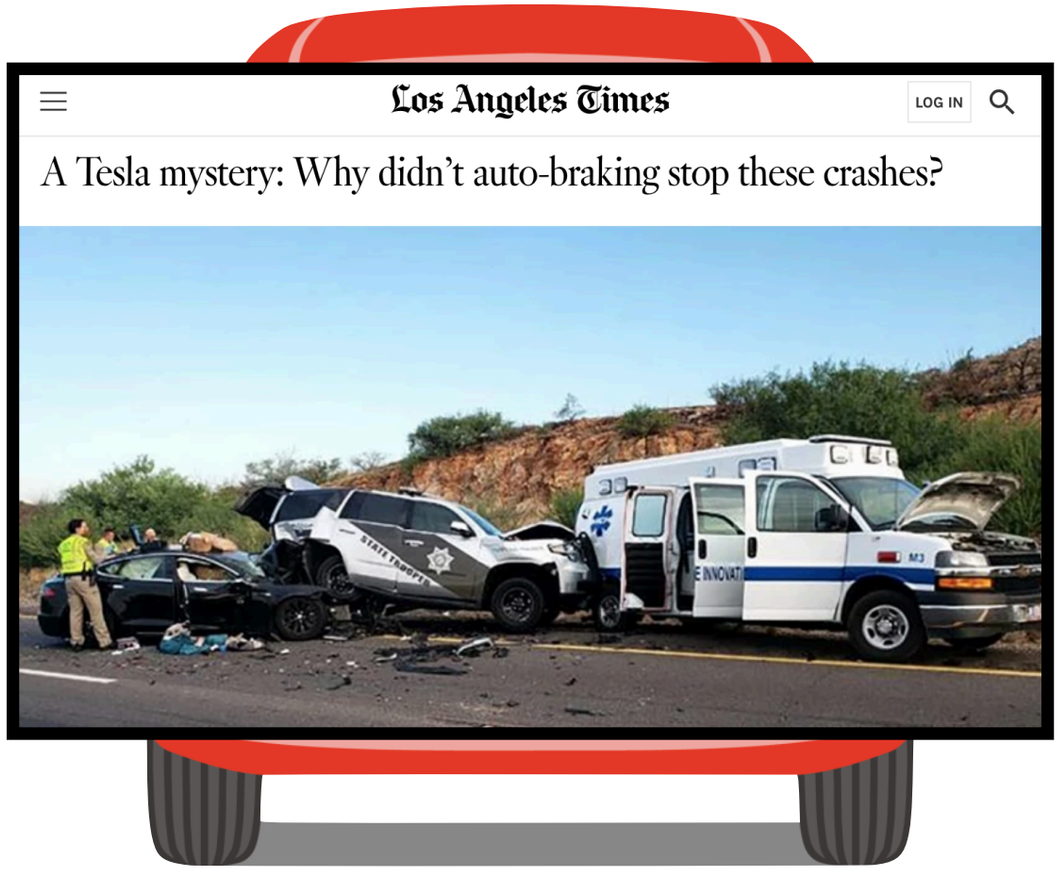


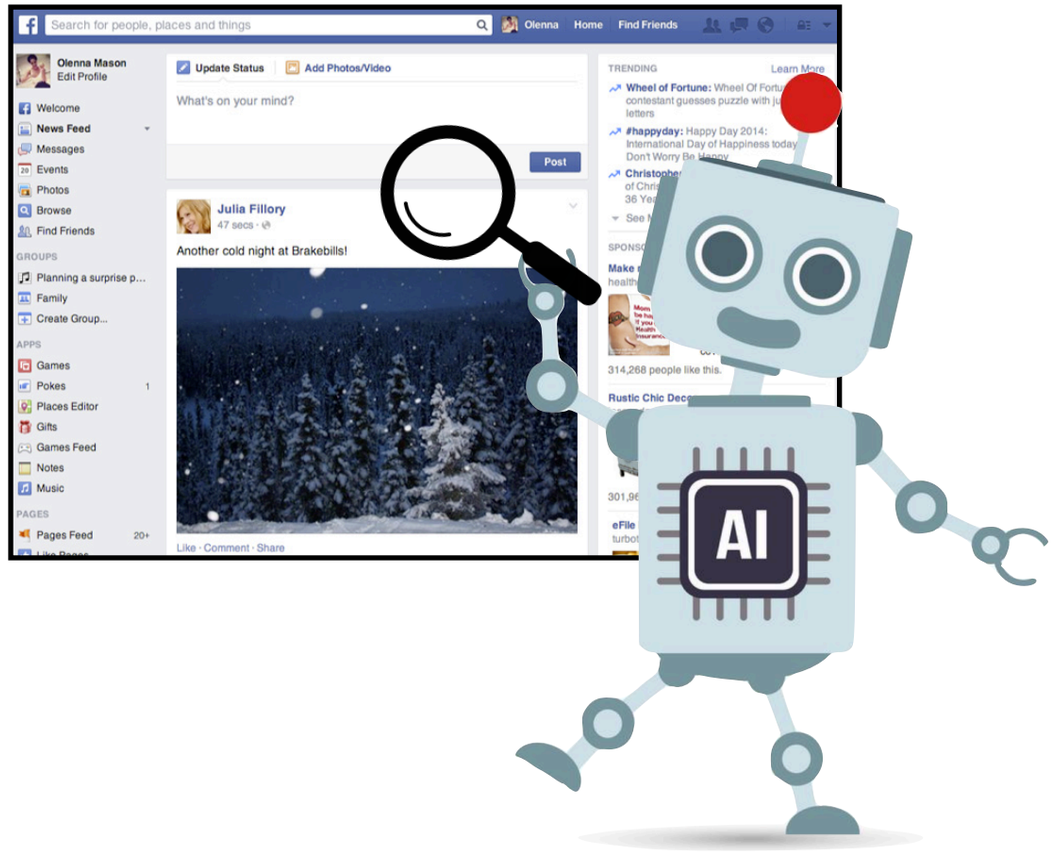

Search for people, places and things
Olenna
Home
Find Friends
Olenna Mason
Edit Profile
Welcome
News Feed
Messages
Events
Photos
Browse
Find Friends
GROUPS
Planning a surprise p...
Family
Create Group...
APPS
Games
Pokes
Places Editor
Gifts
Games Feed
Notes
Music
PAGES
Pages Feed
Update Status
Add Photos/Video
What's on your mind?
Post
Julia Fillory
47 secs
Another cold night at Brakebills!
Like · Comment · Share
TRENDING
Learn More
Wheel of Fortune:
#happyday:
Happy Day 2014: International Day of Happiness today. Don't Worry Be Happy
314,268 people like this.
AI

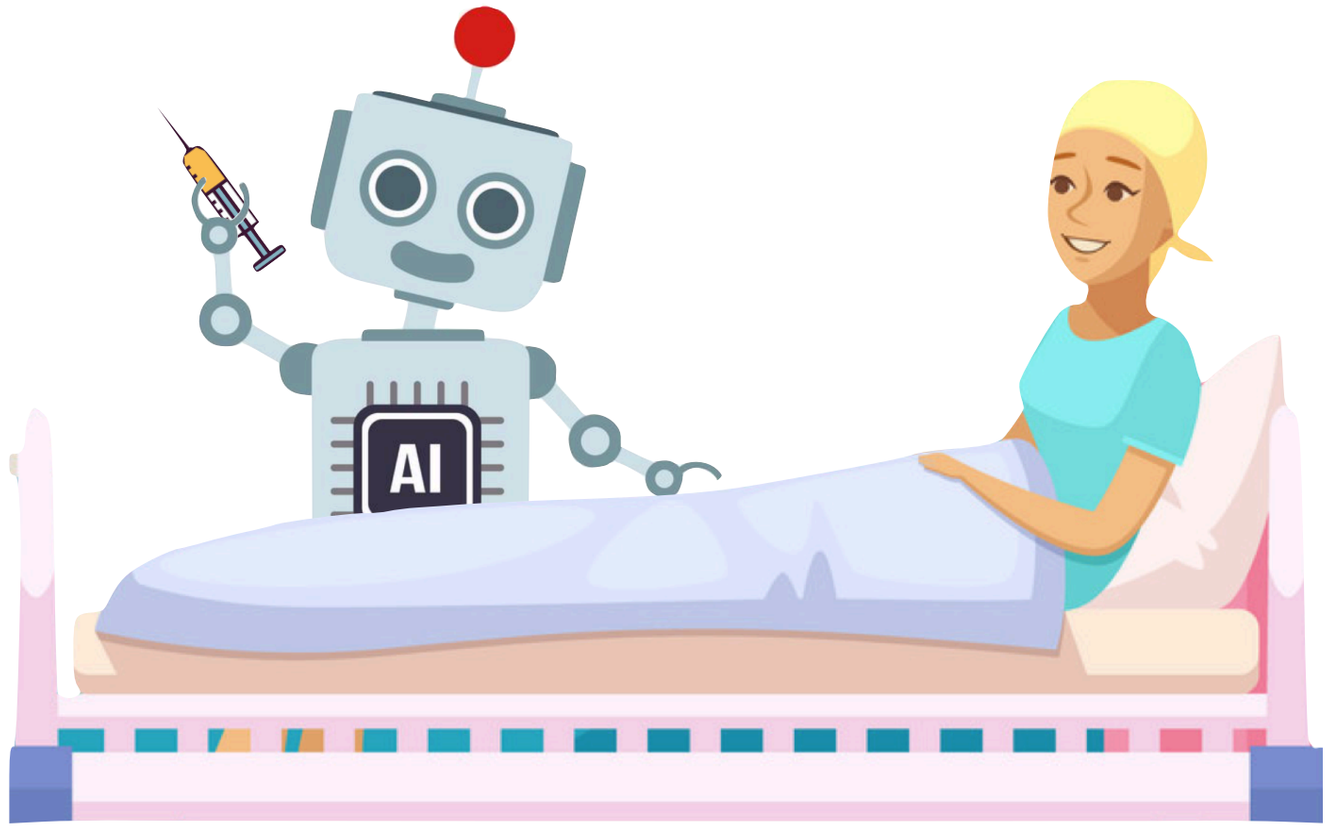


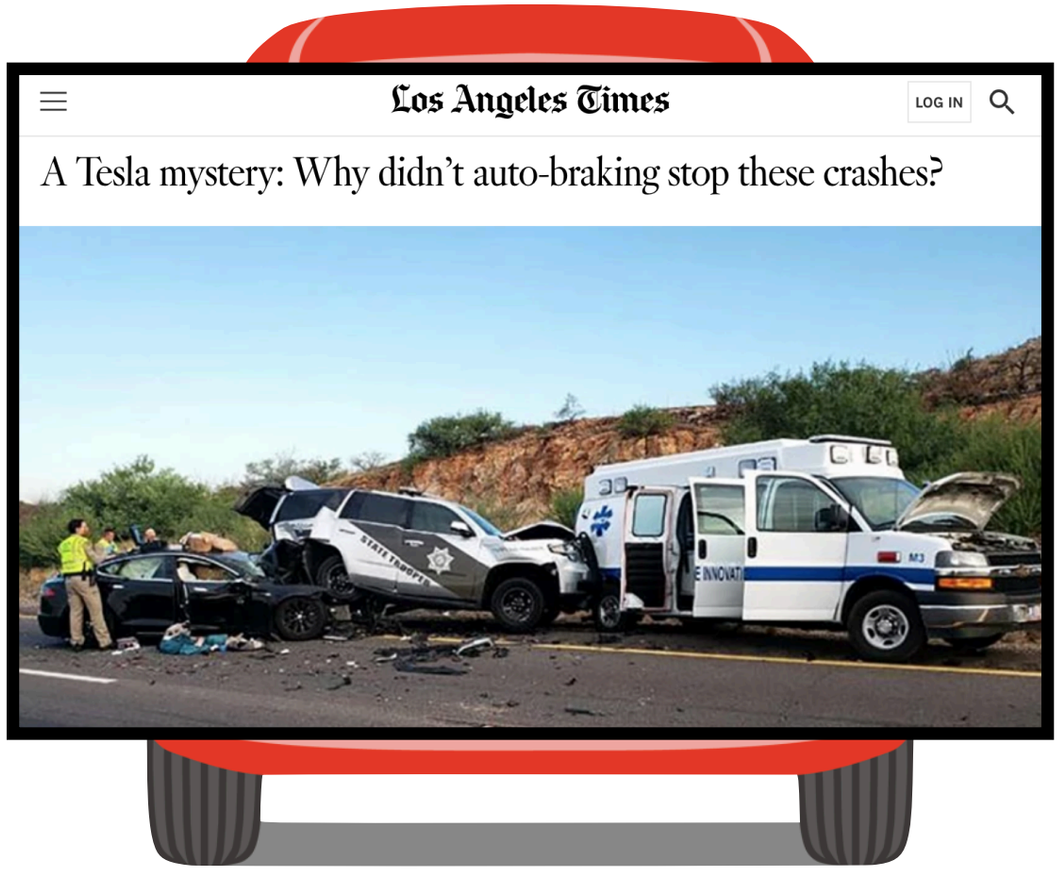

Search for people, places and things
Olenna
Home
Find Friends
Olenna Mason
Edit Profile
Welcome
News Feed
Messages
Events
Photos
Browse
Find Friends
GROUPS
Planning a surprise p...
Family
Create Group...
APPS
Games
Pokes
Places Editor
Gifts
Games Feed
Notes
Music
PAGES
Pages Feed
Update Status
Add Photos/Video
What's on your mind?
Post
Julia Fillory
47 secs
Another cold night at Brakebills!
Like · Comment · Share
TRENDING
Learn More
Wheel of Fortune:
#happyday:
Happy Day 2014: International Day of Happiness today. Don't Worry Be Happy
314,268 people like this.
AI

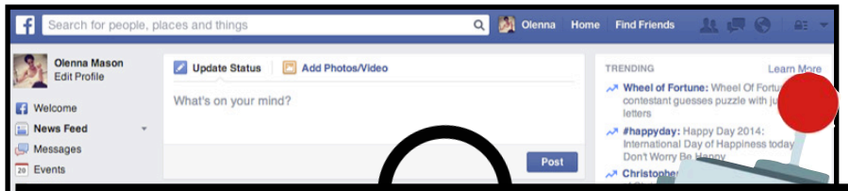

AI


Los Angeles Times
LOG IN
A Tesla mystery: Why didn't auto-braking stop these crashes?


Search for people, places and things
Olenna
Home
Find Friends
Olenna Mason
Edit Profile
Welcome
News Feed
Messages
Events
Photos
Browse
Find Friends
GROUPS
Planning a surprise p...
Family
Create Group...
APPS
Games
Pokes
Places Editor
Gifts
Games Feed
Notes
Music
PAGES
Pages Feed
Update Status
Add Photos/Video
What's on your mind?
Post
Julia Fillory
47 secs
Another cold night at Brakebills!
Like · Comment · Share
TRENDING
Learn More
Wheel of Fortune:
#happyday:
314,268 people like this.
AI


AI





from Net Politics *and* Digital and Cyberspace Policy Program

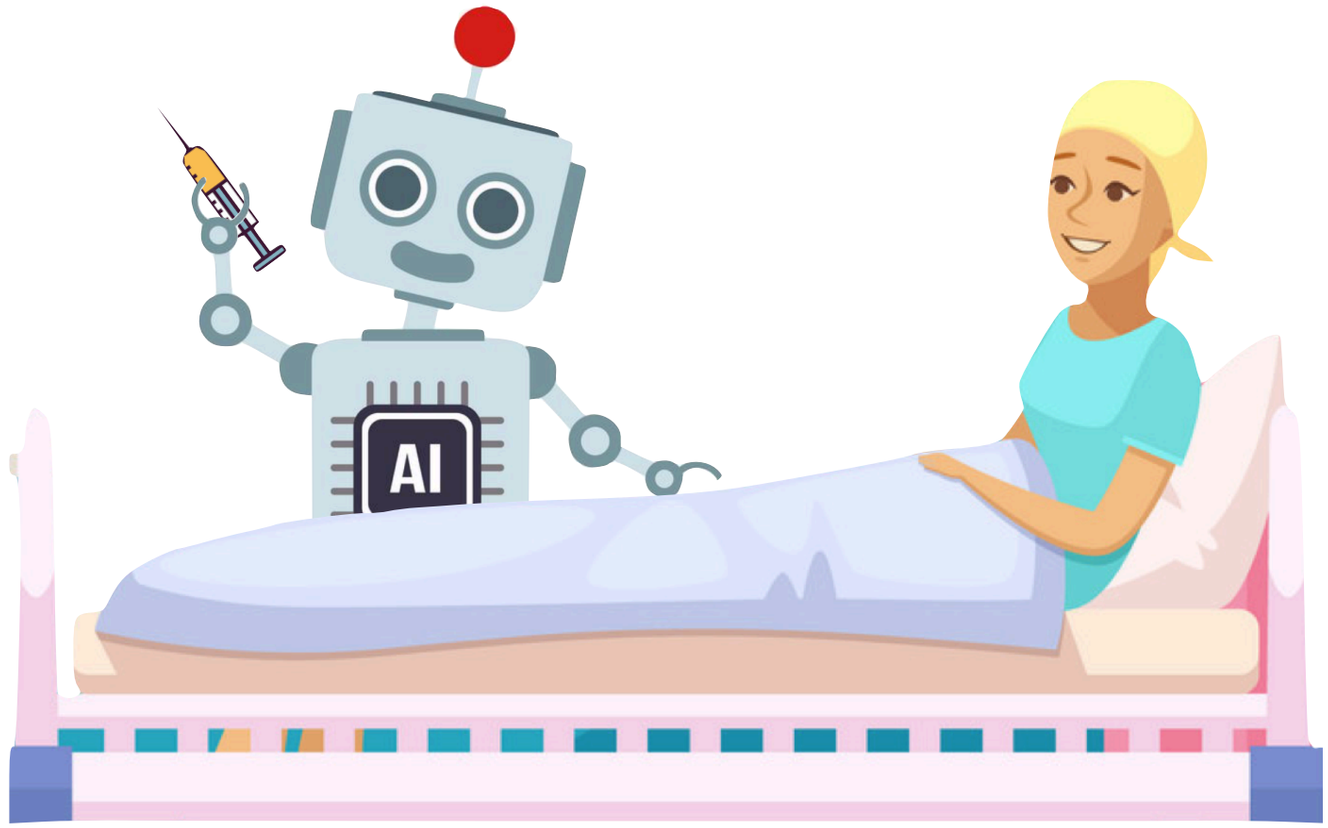
Facebook's Content Moderation Failures in Ethiopia

Facebook has failed to moderate content in underserved countries. Facebook and other social media companies must invest more in local content moderation, instead of relying on global AI systems.

Amhara militia members ride in the back of a truck towards a fight with the Tigray People's Liberation Front. Reuters/Tiksa Negeri

Blog Post by Caroline Allen, Guest Contributor

April 19, 2022 2:36 pm (EST)



Los Angeles Times

LOG IN

A Tesla mystery: Why didn't auto-braking stop these crashes?

Search for people, places and things

Olenna Home Find Friends

Olenna Mason
Edit Profile

Welcome
News Feed
Messages
Events

Update Status Add Photos/Video

What's on your mind?

Post

TRENDING Learn More

Wheel of Fortune: Wheel Of Fortu... contestant guesses puzzle with ... letters

#happyday: Happy Day 2014: International Day of Happiness today. Don't Worry Be Happy

Christophe...

from Net Politics *and* Digital and Cyberspace Policy Program

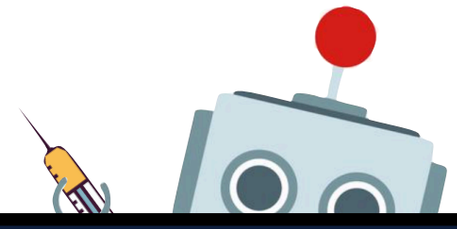
Facebook's Content Moderation Failures in Ethiopia

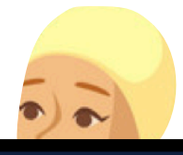
Facebook has failed to moderate content in underserved countries. Facebook and other social media companies must invest more in local content moderation, instead of relying on global AI systems.

Amhara militia members ride in the back of a truck towards a fight with the Tigray People's Liberation Front. Reuters/Tiksa Negeri

Blog Post by *Caroline Allen, Guest Contributor*

April 19, 2022 2:36 pm (EST)





ARTIFICIAL INTELLIGENCE

Google's medical AI was super accurate in a lab. Real life was a different story.

If AI is really going to ma...
works when real humar...

By Will Douglas Heaven

Medscape Tuesday, December 13, 2022

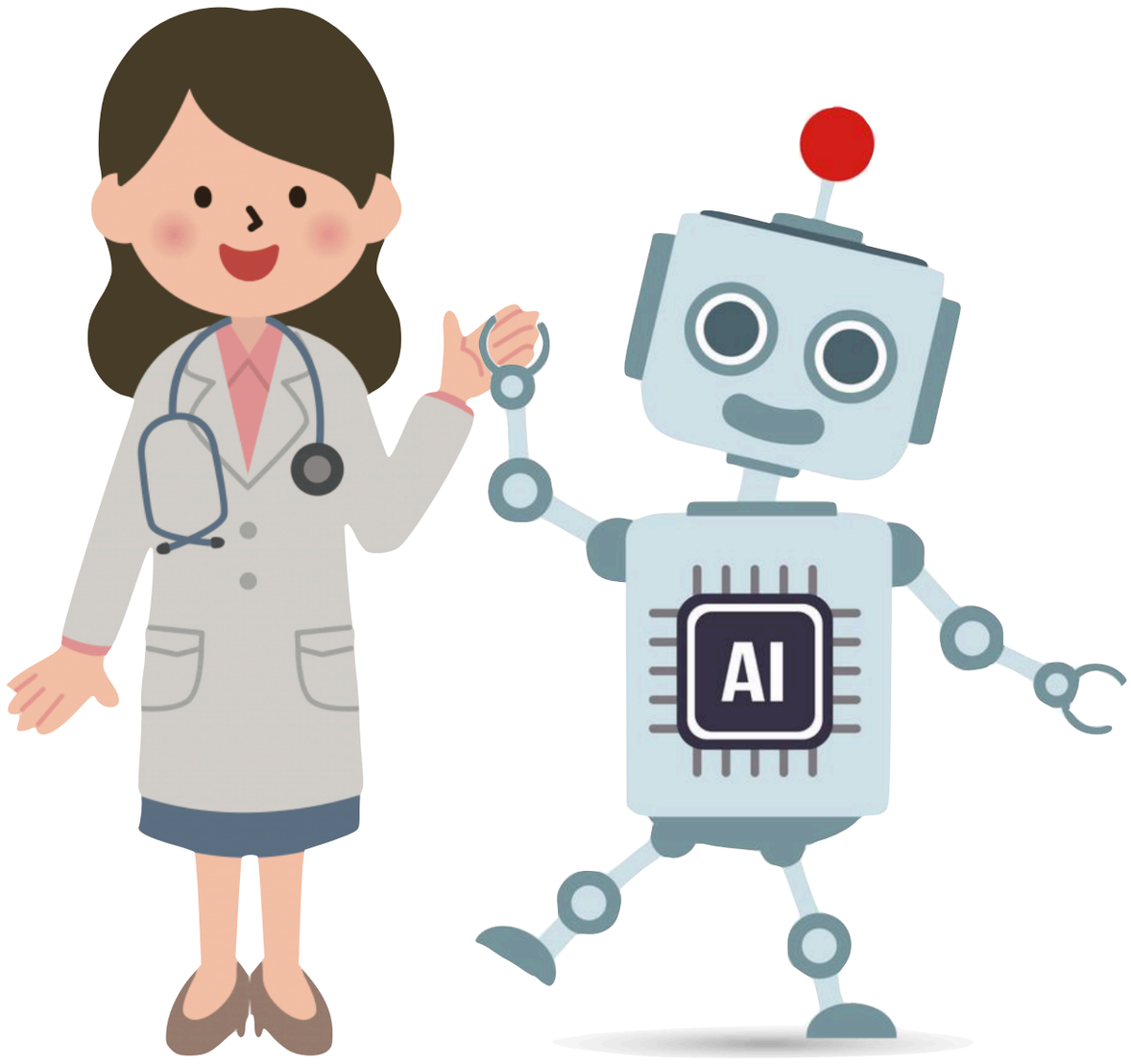
NEWS & PERSPECTIVE DRUGS & DISEASES CME & EDUCATION ACADEMY VIDEO DECISION POINT

News > Medscape Medical News > Conference News > CHEST 2022

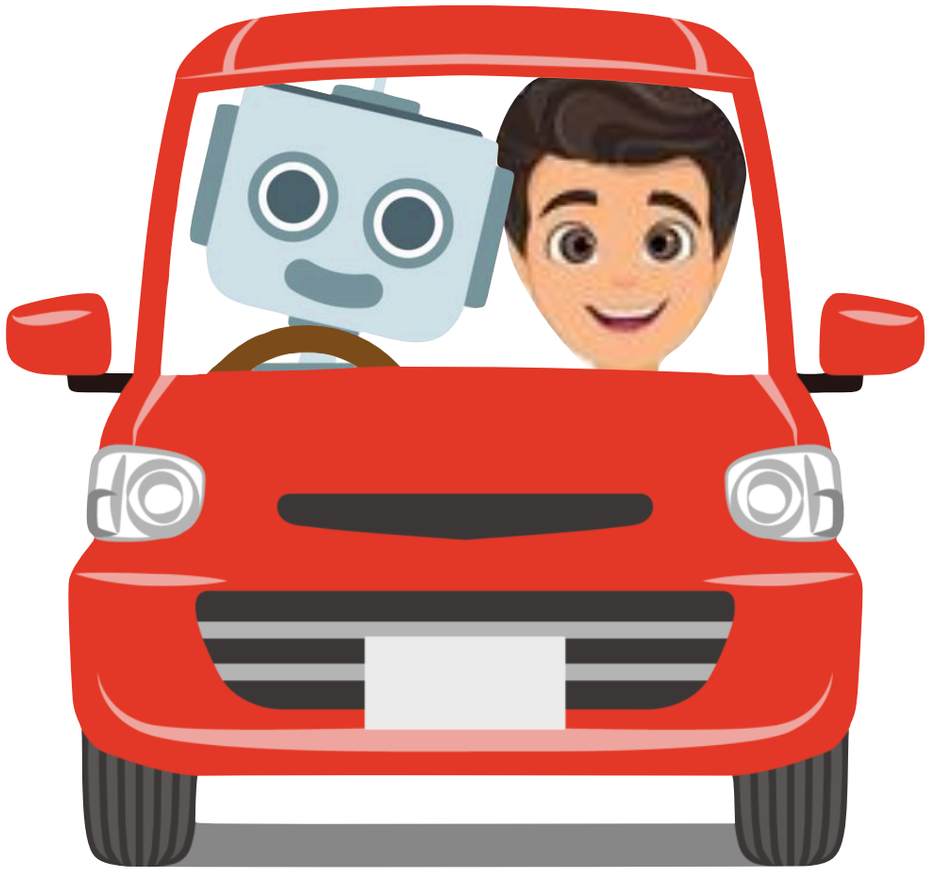
Sepsis Predictor Tool Falls Short in Emergency Setting

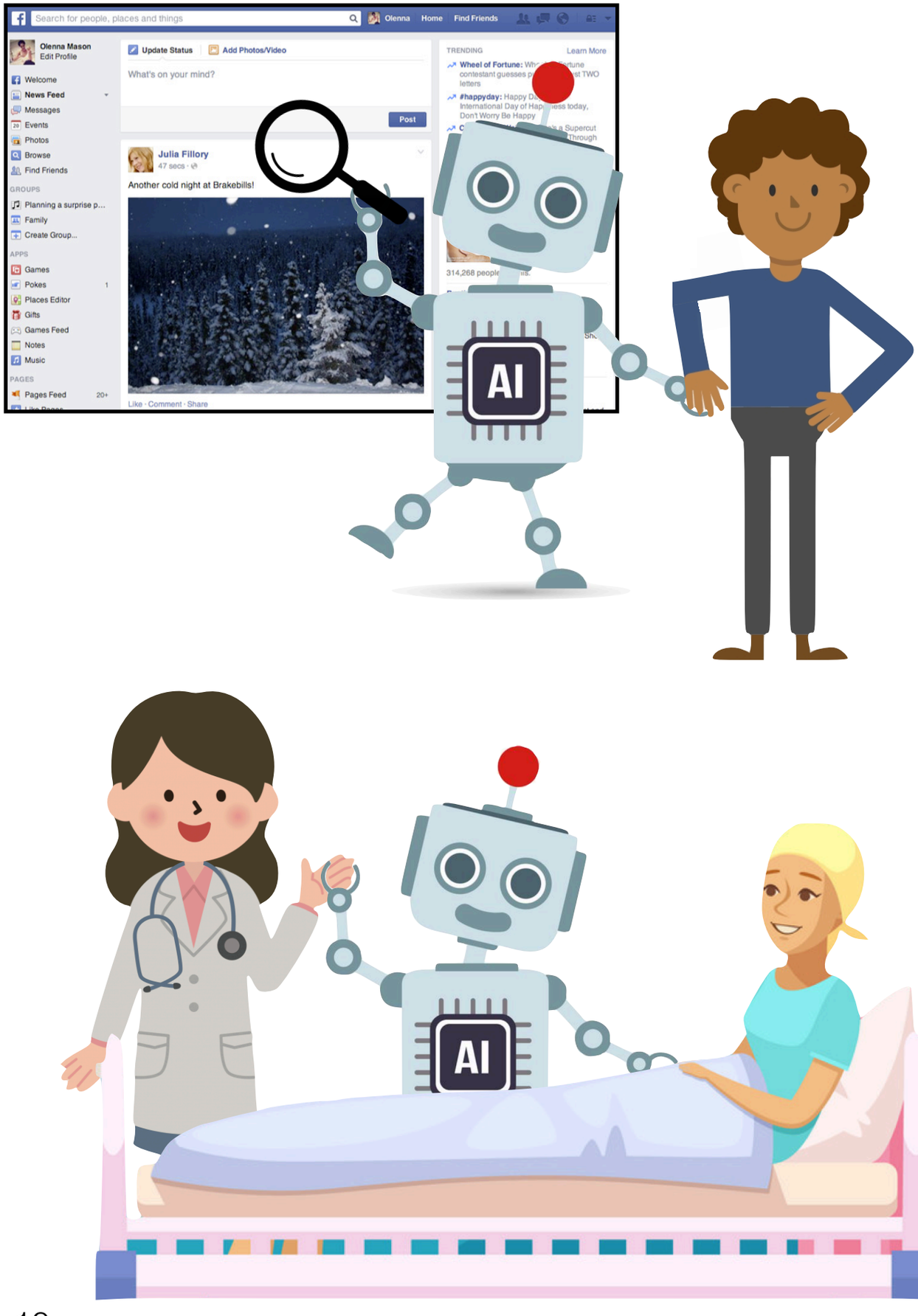
Heidi Splete

October 17, 2022



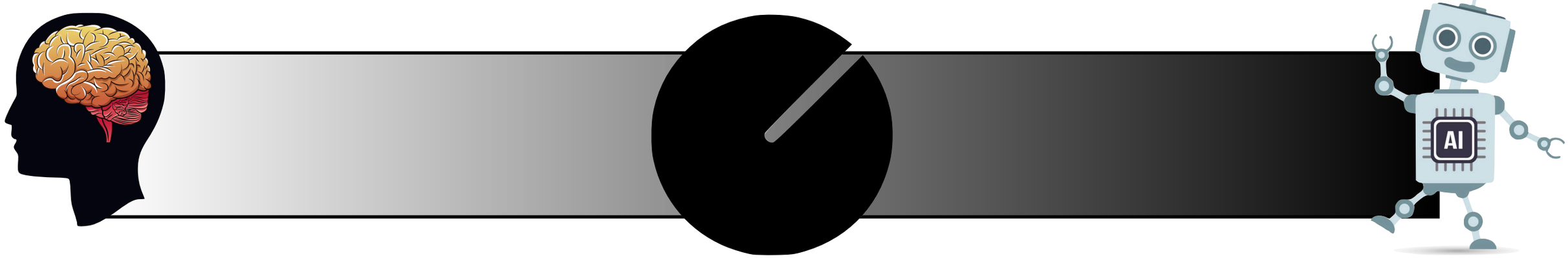
human-AI collaboration

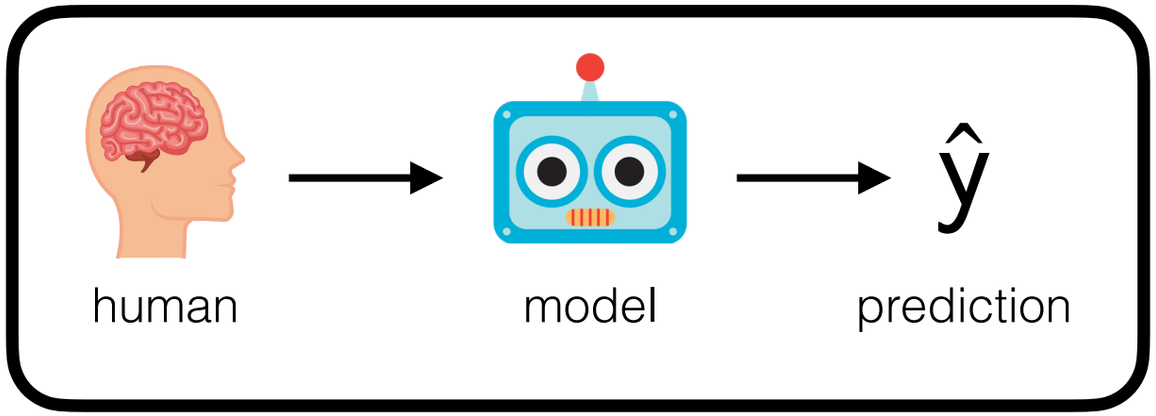


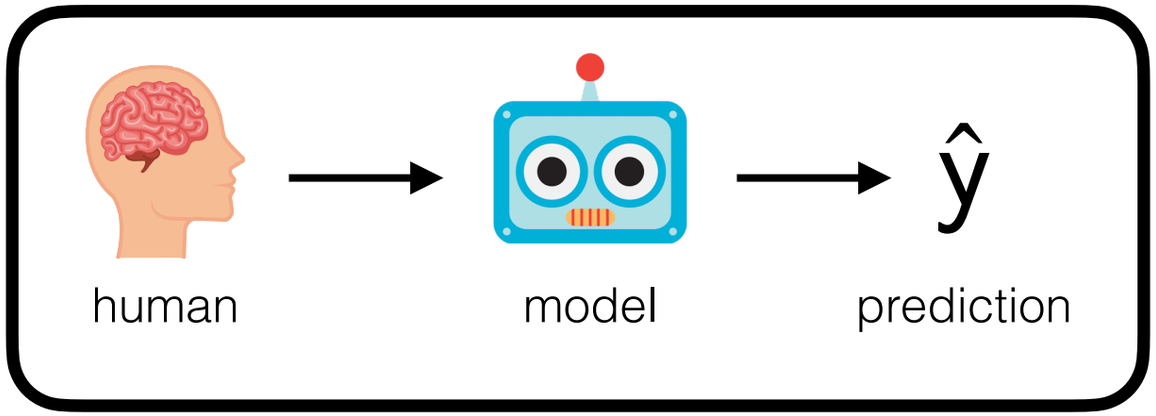

Search for people, places and things
Olenna
Home
Find Friends
Olenna Mason
Edit Profile
Welcome
News Feed
Messages
Events
Photos
Browse
Find Friends
GROUPS
Planning a surprise p...
Family
Create Group...
APPS
Games
Pokes
Places Editor
Gifts
Games Feed
Notes
Music
PAGES
Pages Feed
Update Status
Add Photos/Video
What's on your mind?
Post
Julia Fillory
47 secs
Another cold night at Brakebills!
Like · Comment · Share
TRENDING
Learn More
Wheel of Fortune:
#happyday:
AI

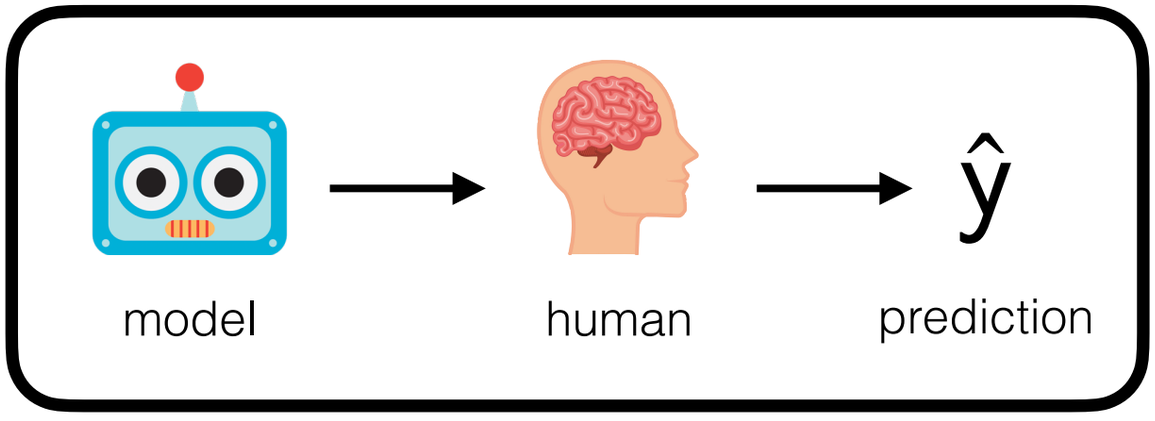
AI

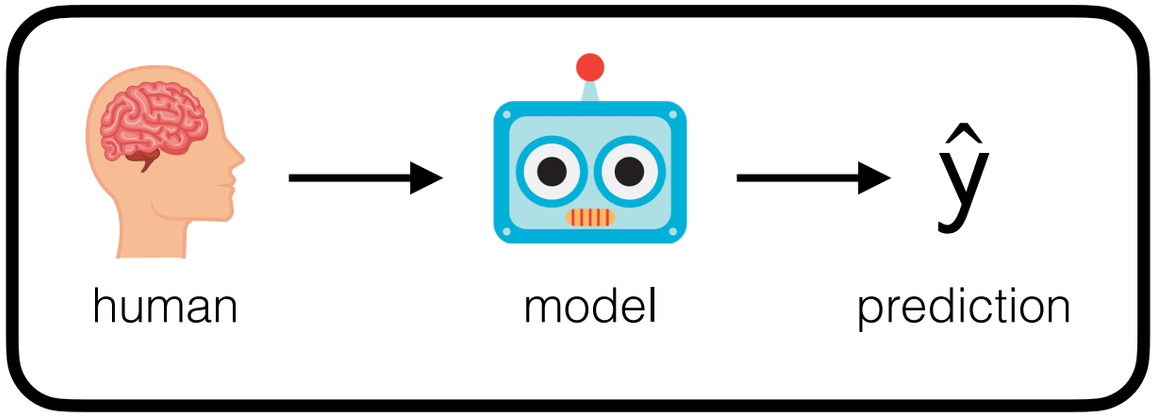
safe, gradual automation

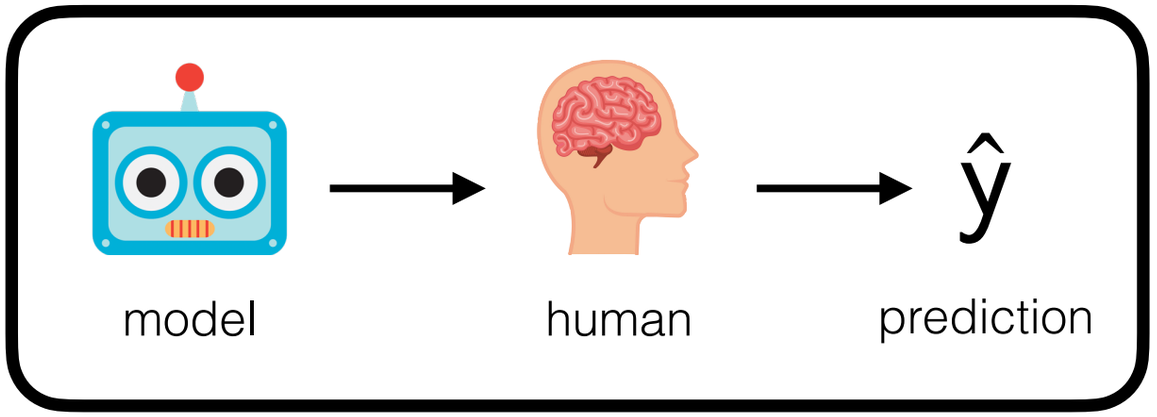


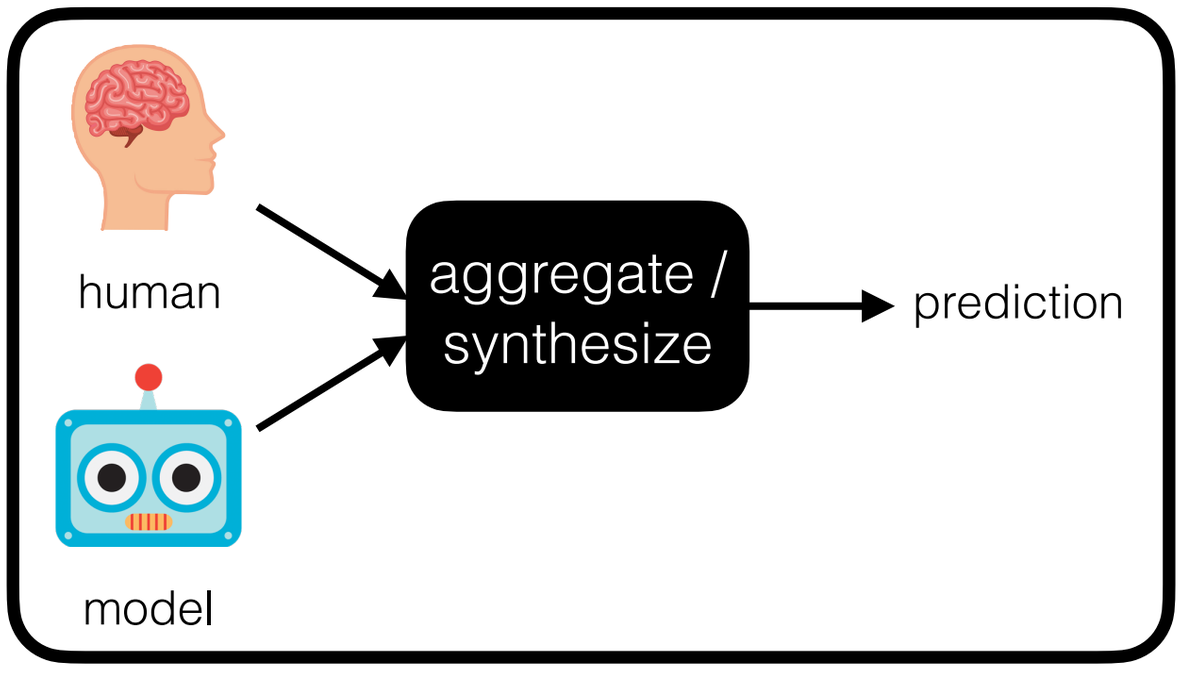

human
model
ŷ
prediction

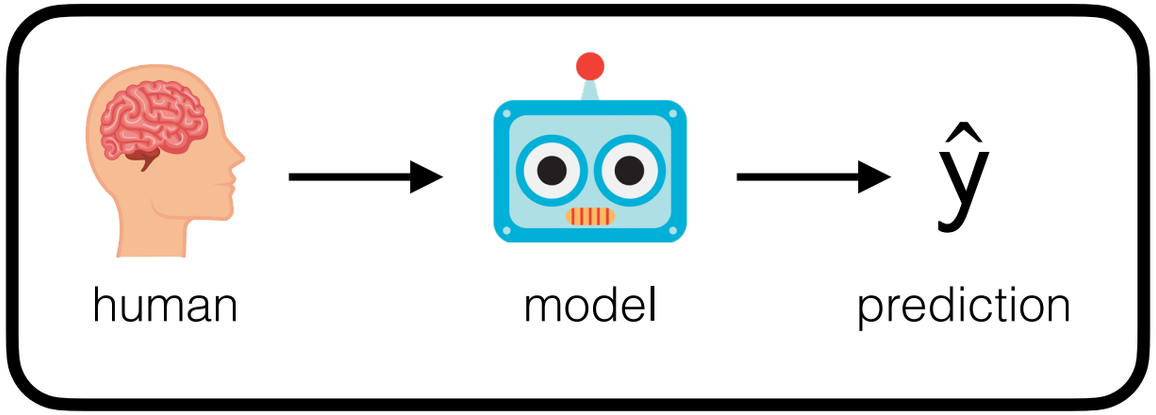

human
model
ŷ
prediction

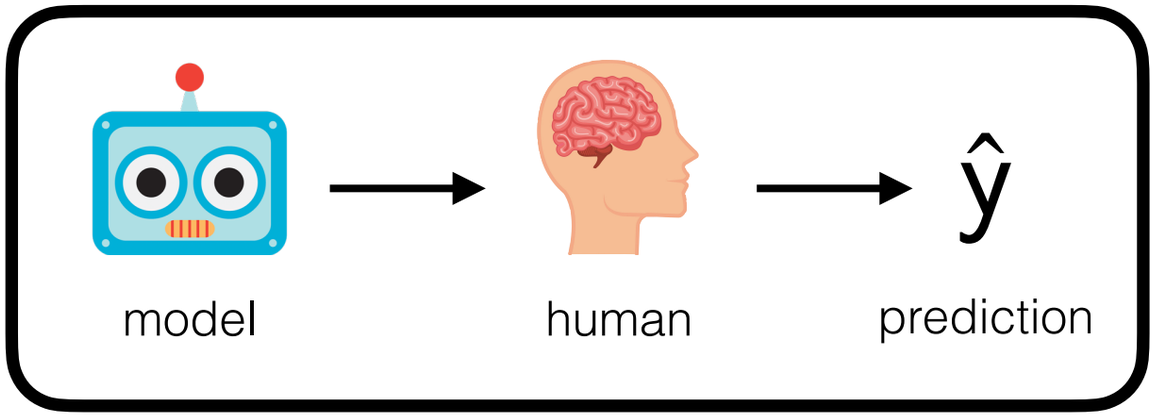

model
human
ŷ
prediction

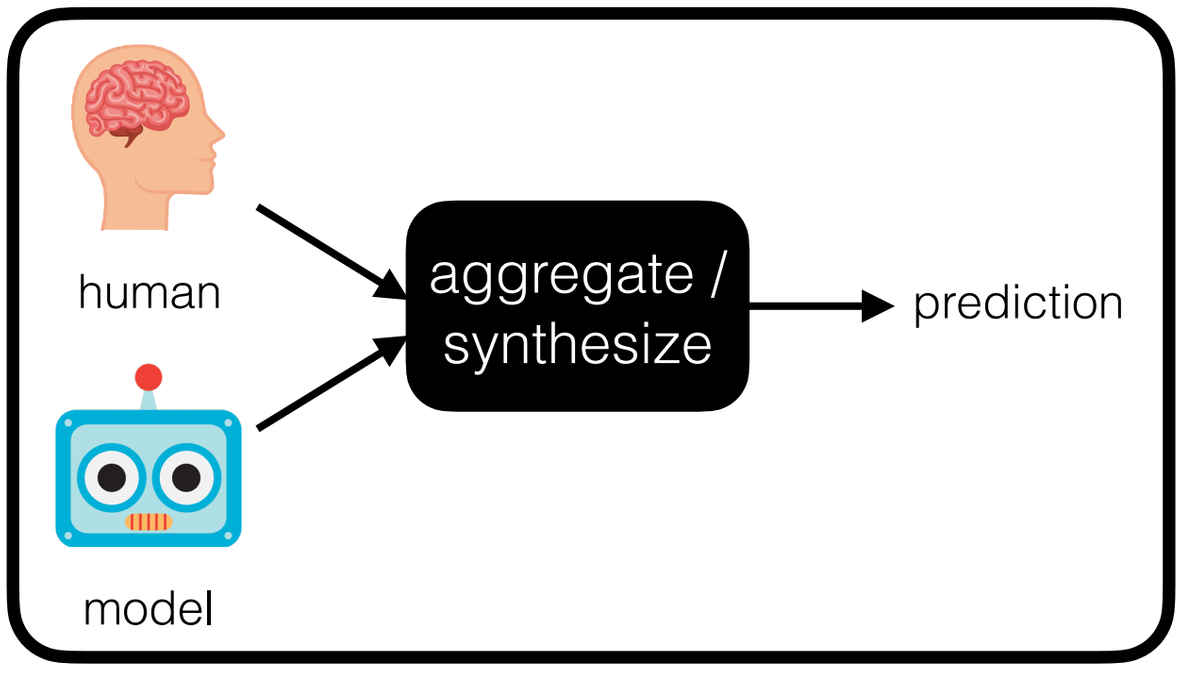

human
model
ŷ
prediction

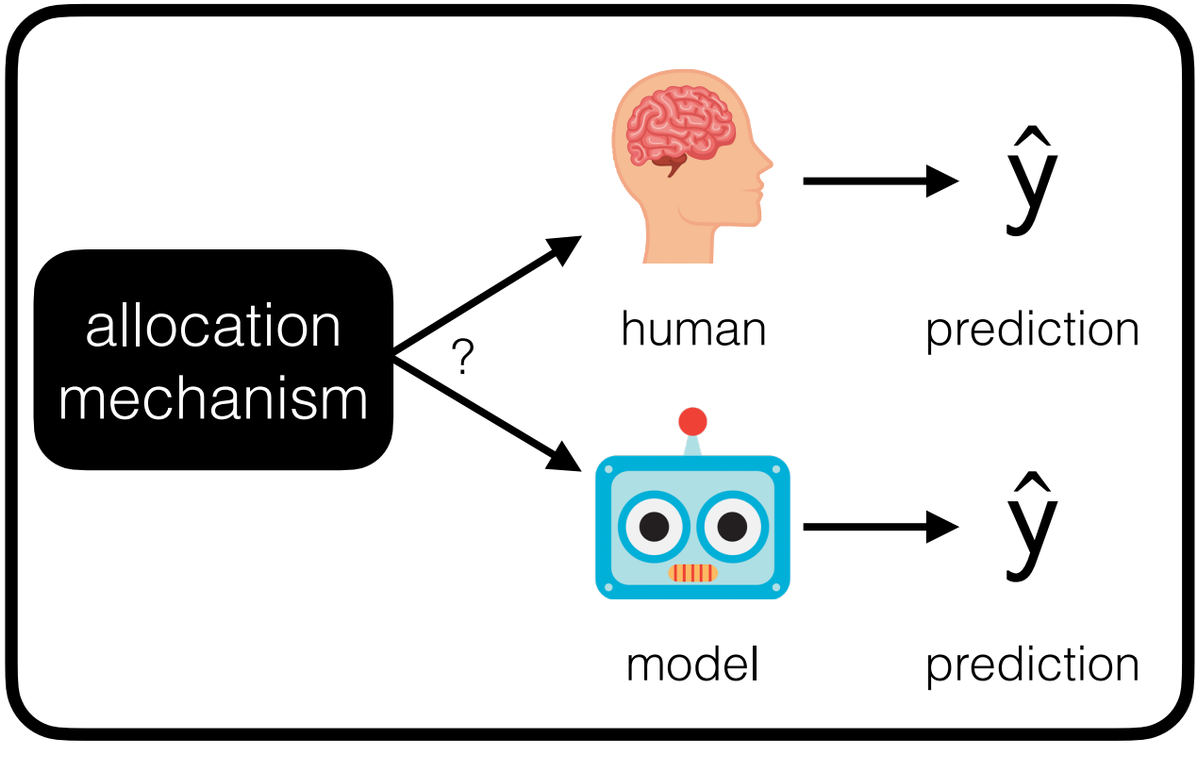

model
human
ŷ
prediction

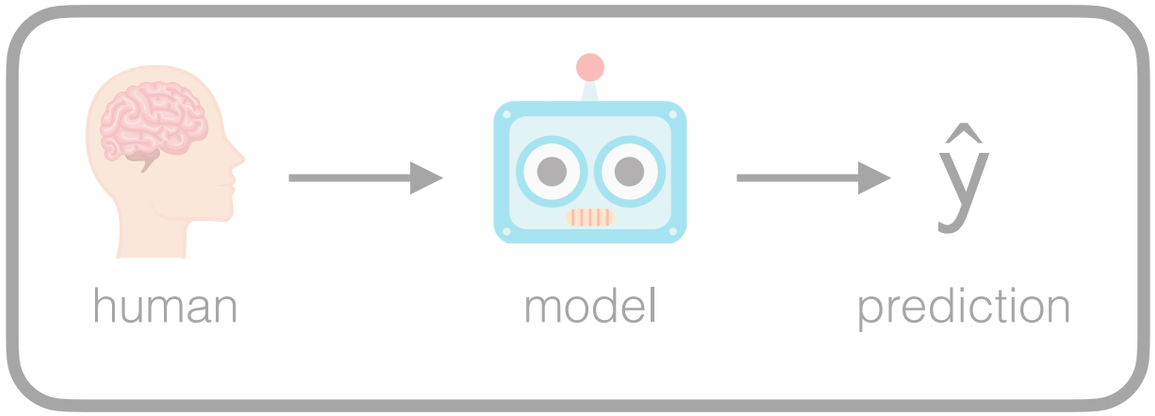

human
model
aggregate / synthesize
prediction

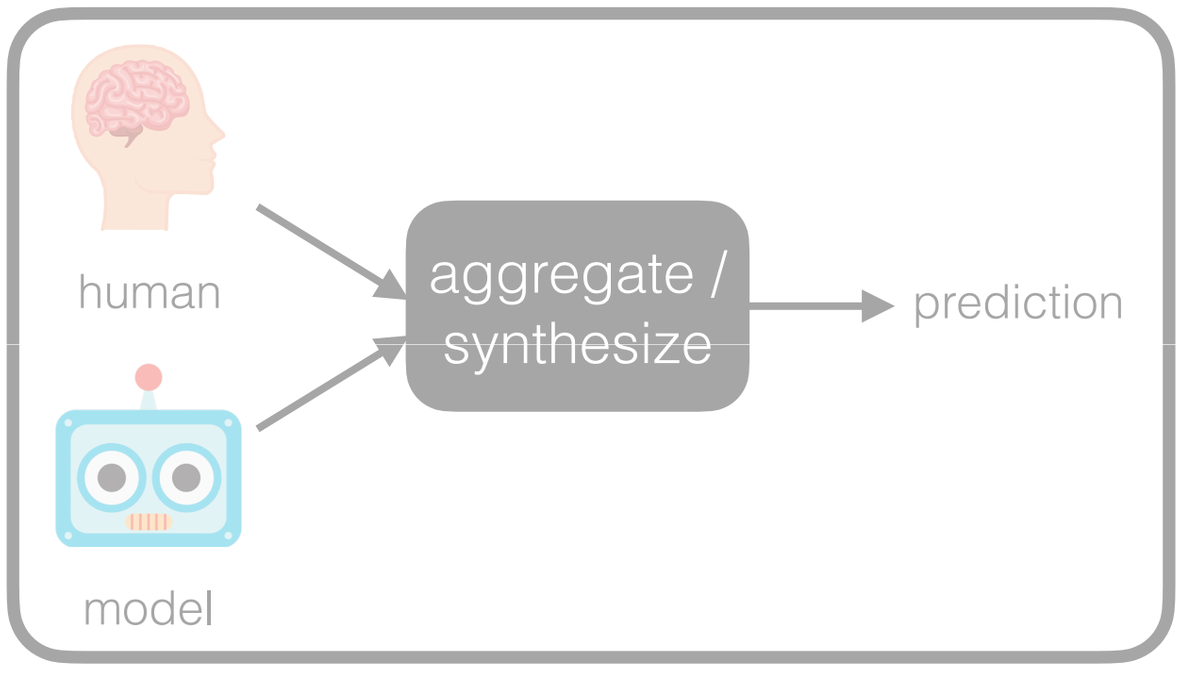

human
model
ŷ
prediction

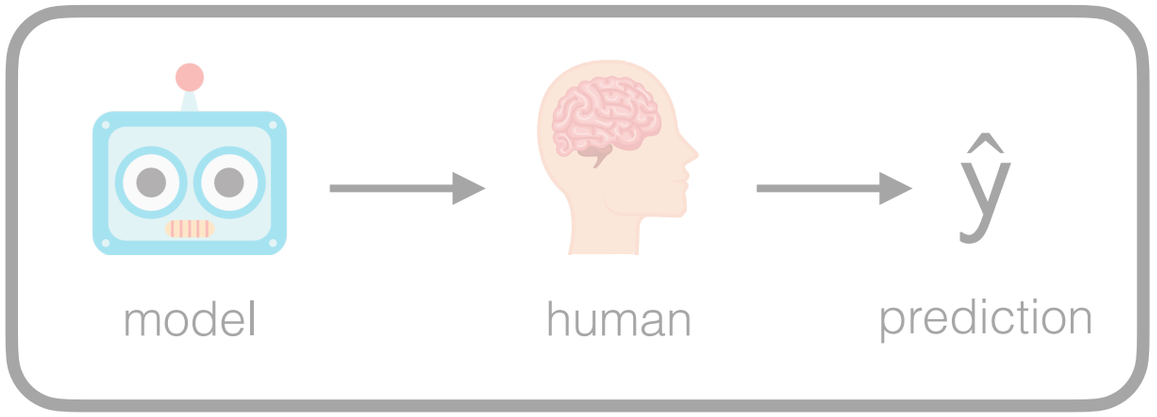

model
human
ŷ
prediction

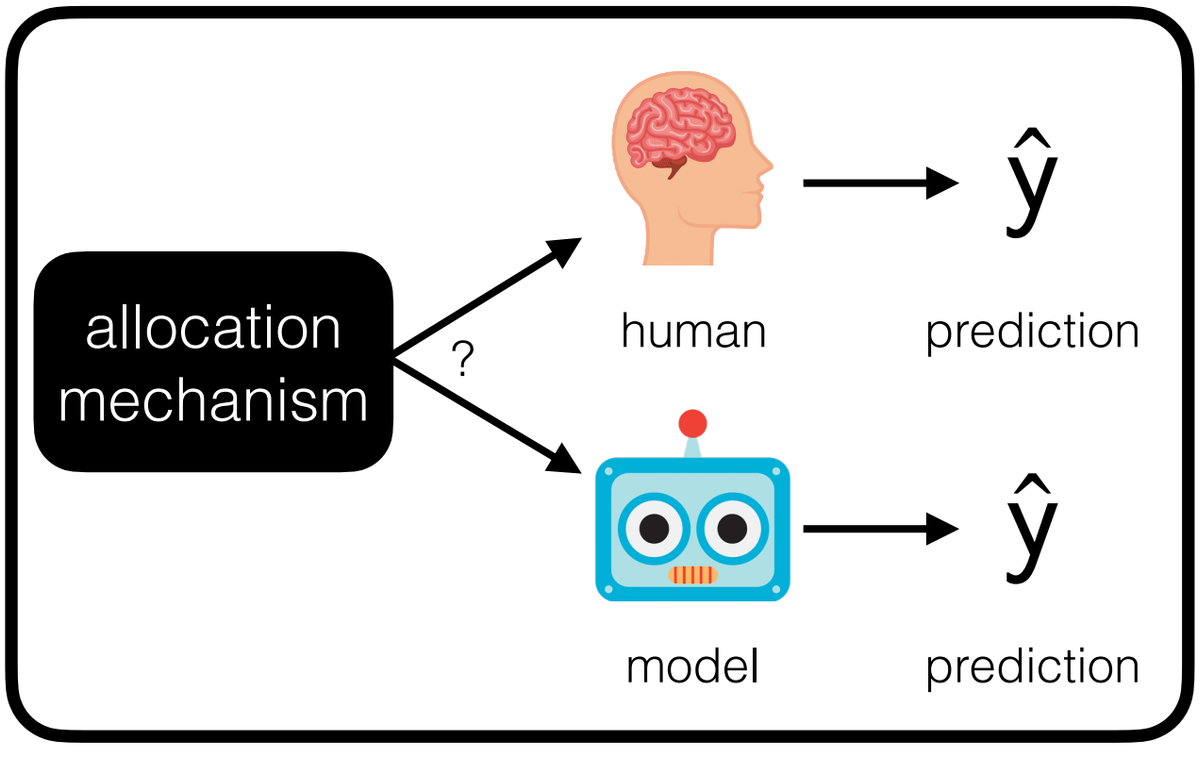

human
aggregate / synthesize
prediction
model


allocation mechanism
?
human
ŷ
prediction
model
ŷ
prediction


human
model
ŷ
prediction


human
aggregate / synthesize
prediction
model

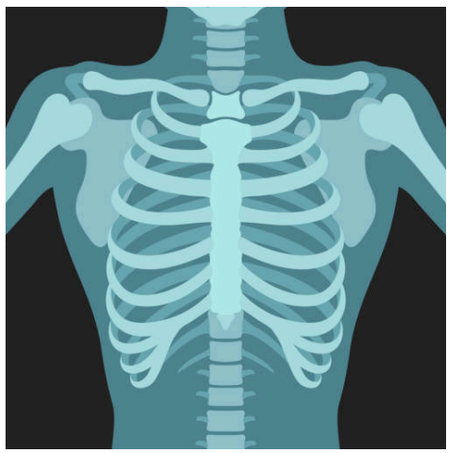

model
human
ŷ
prediction

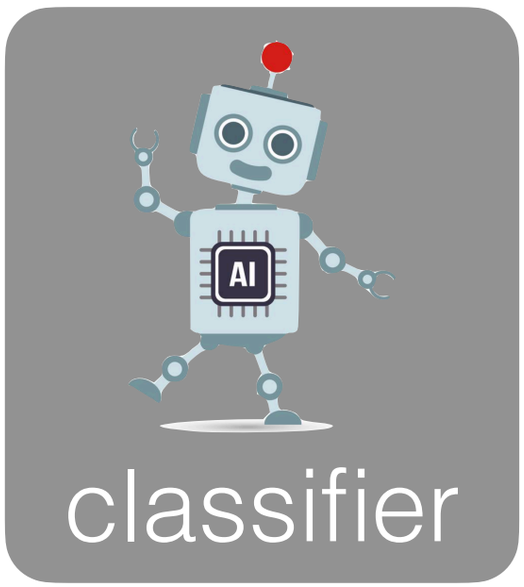

allocation mechanism
?
human
ŷ
prediction
model
ŷ
prediction

- Learning to reject
  - Bayes decision theory
  - consistent surrogate loss
- Learning to defer to an expert
  - Bayes decision theory
  - consistent surrogate loss
  - extension to multiple experts

- Learning to reject
  - Bayes decision theory
  - consistent surrogate loss
- Learning to defer to an expert
  - Bayes decision theory
  - consistent surrogate loss
  - extension to multiple experts

input
features

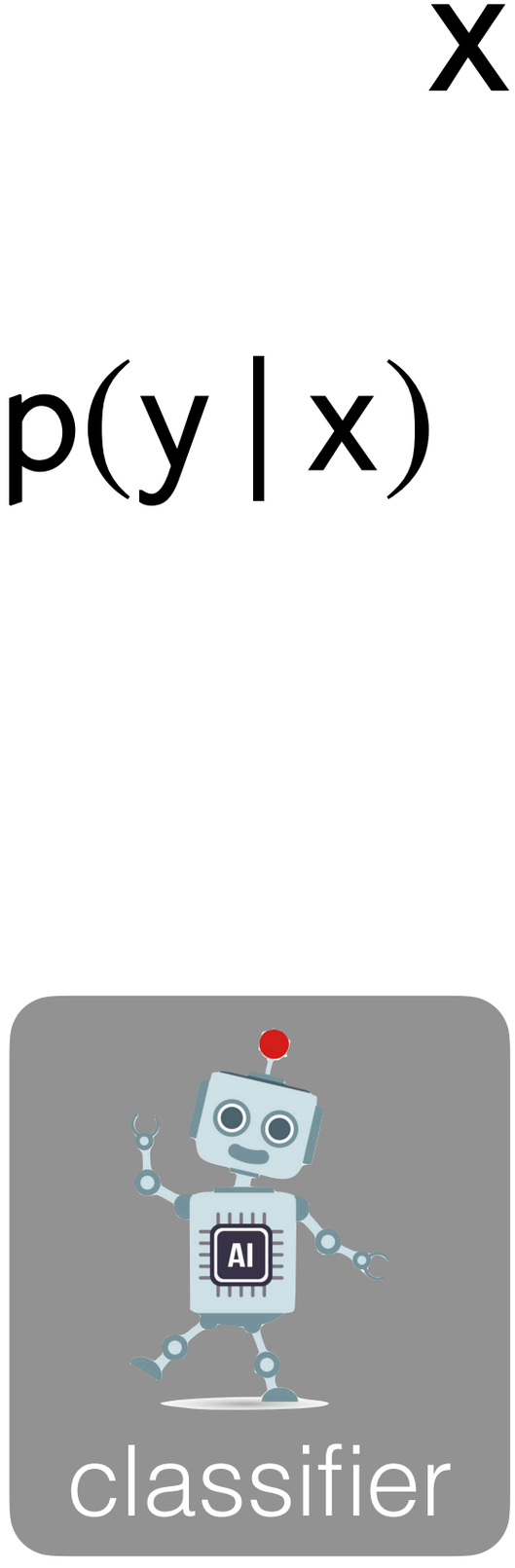


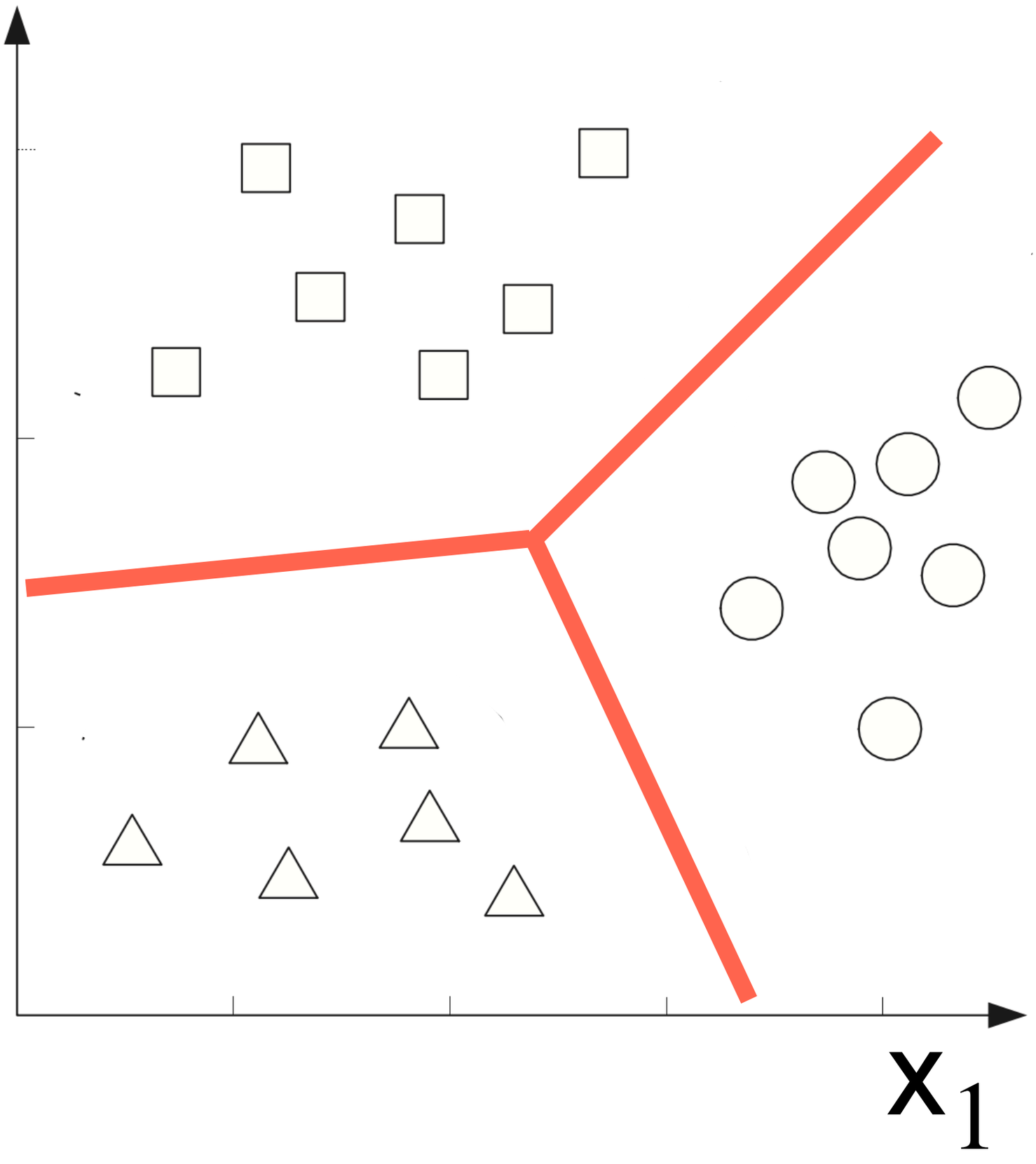


$\hat{y}$

traditional classifier

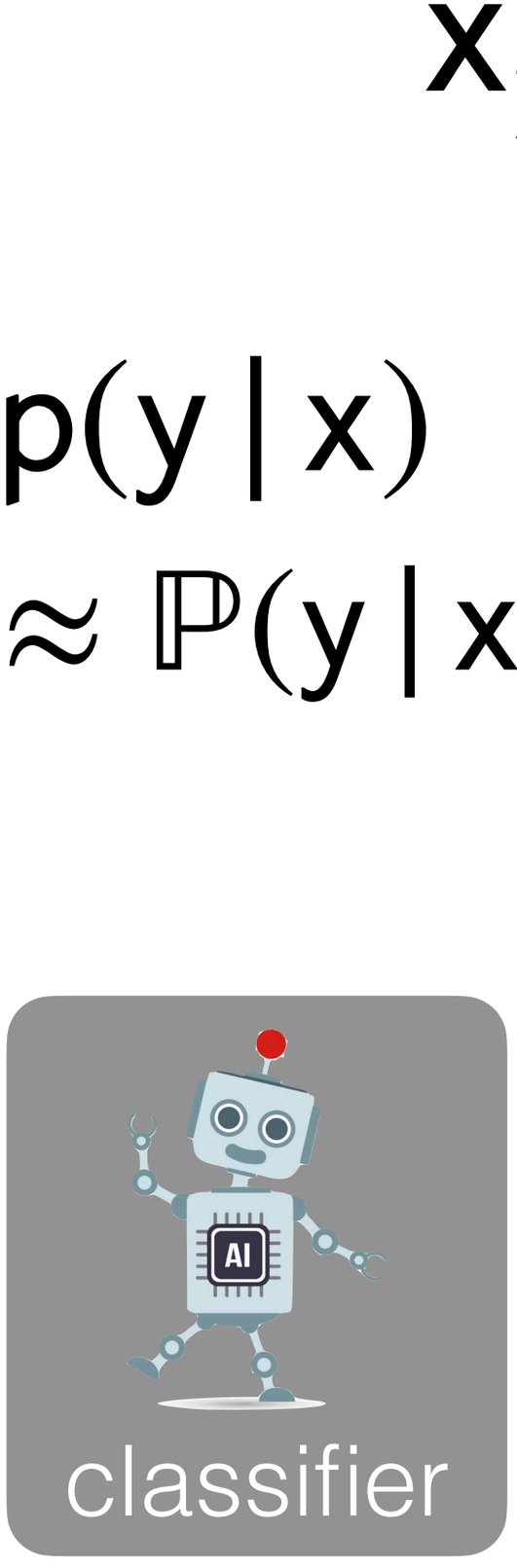
$p(y \mid x)$

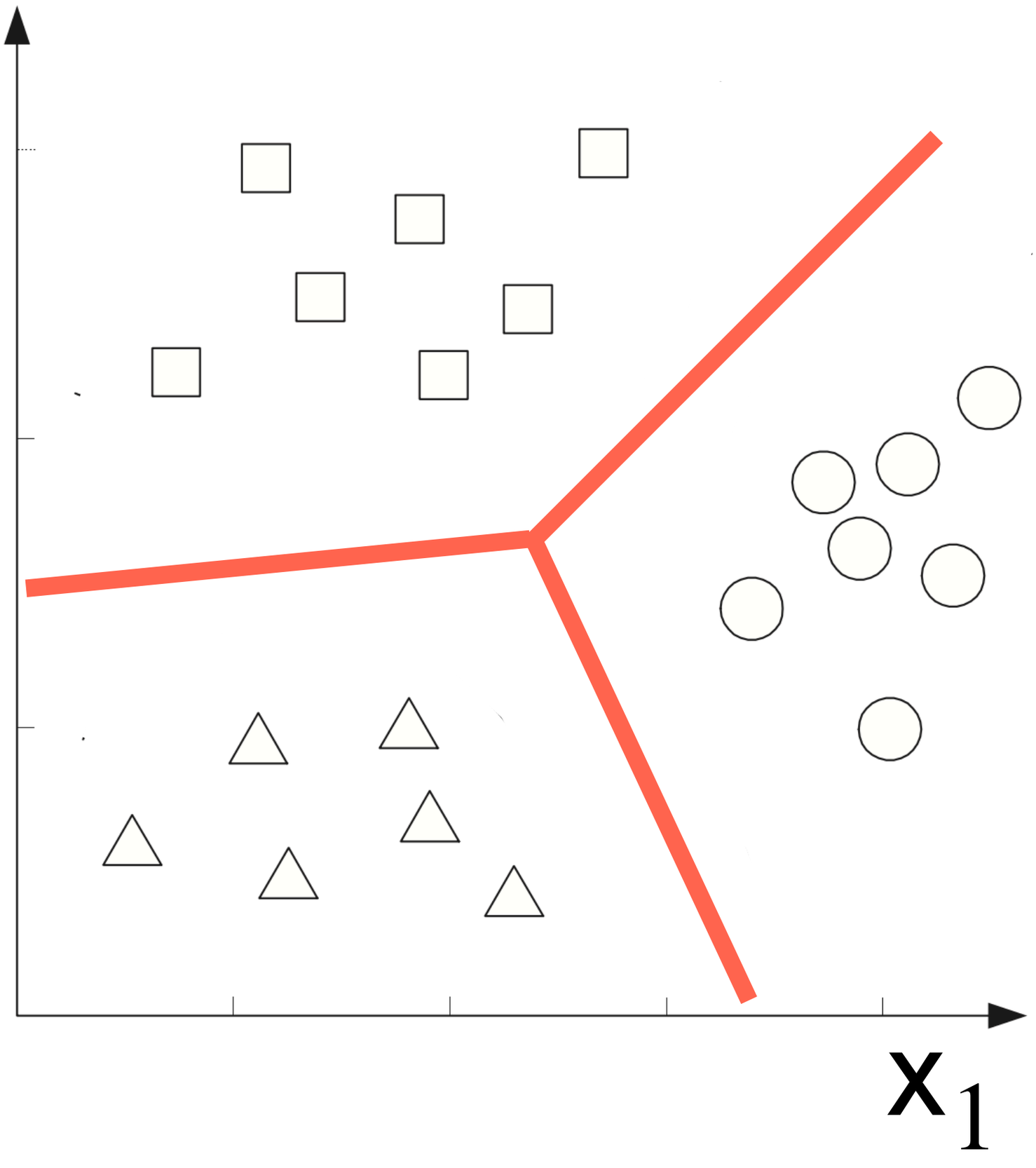




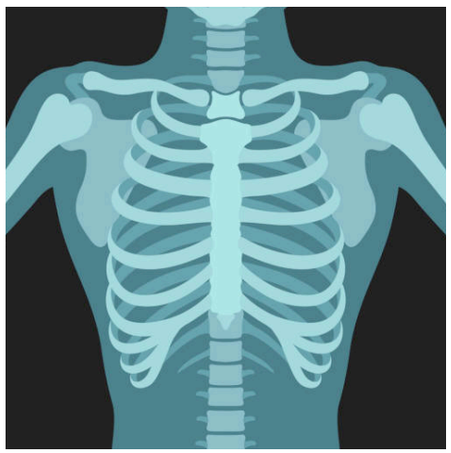
$p(y \mid x)$

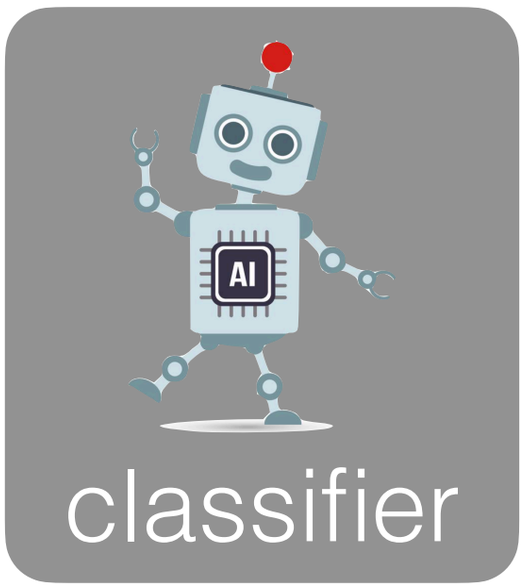
$\approx \mathbb{P}(y \mid x)$





input
features

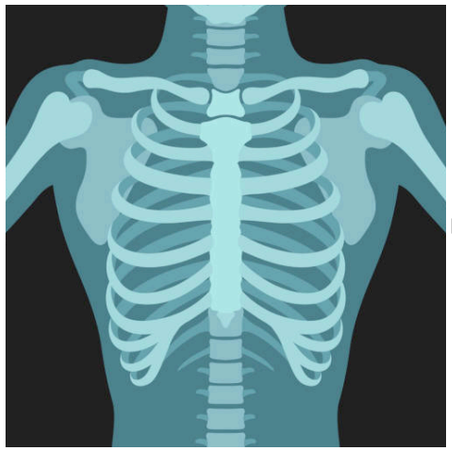


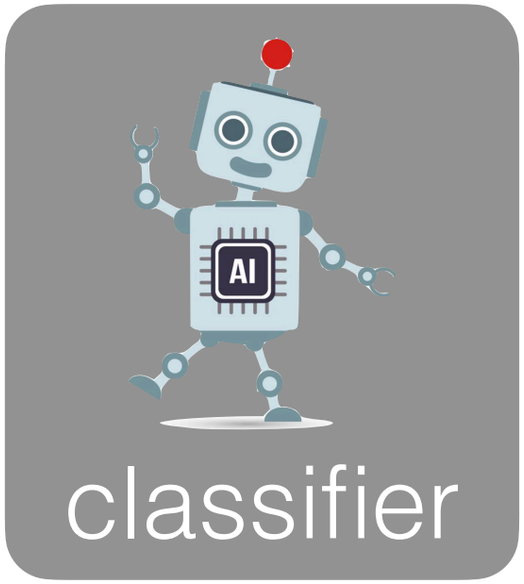


$\hat{y}$

traditional classifier

input features



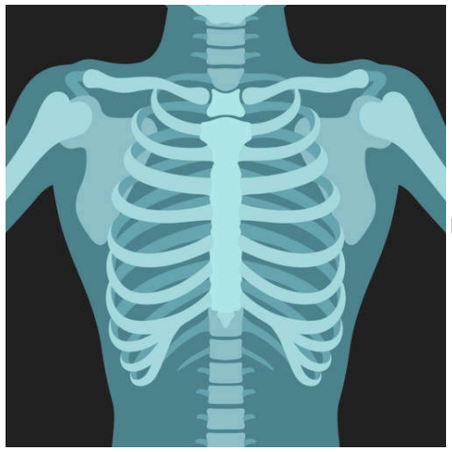


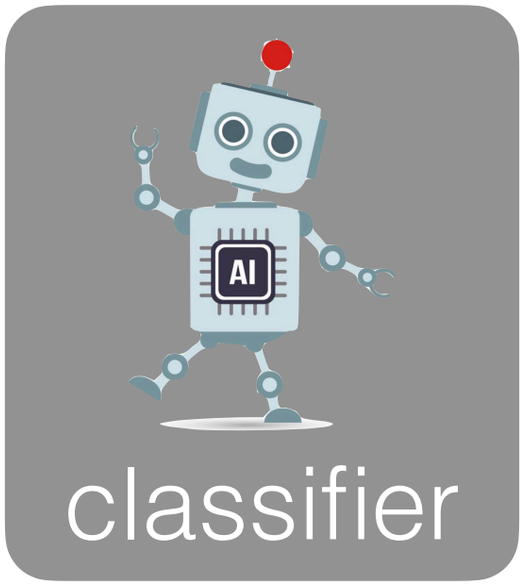
$\hat{y}$

abstain

learning to reject

input features





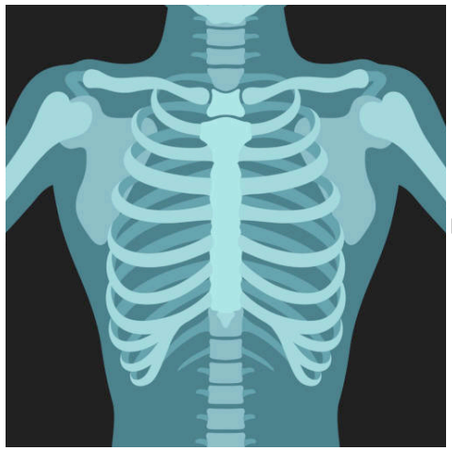
$\hat{y}$

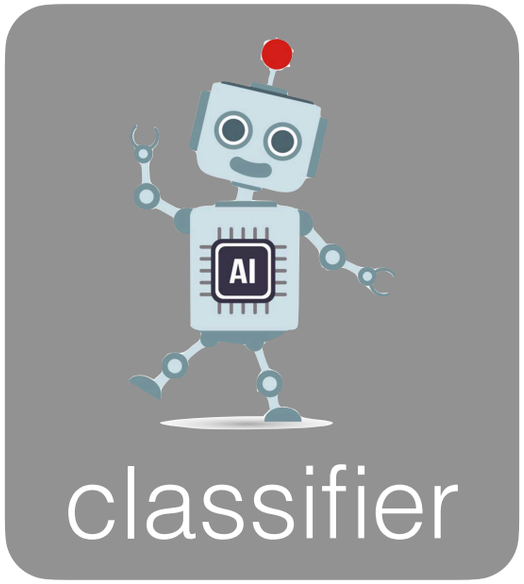
abstain

“I don’t know”

learning to reject

input features



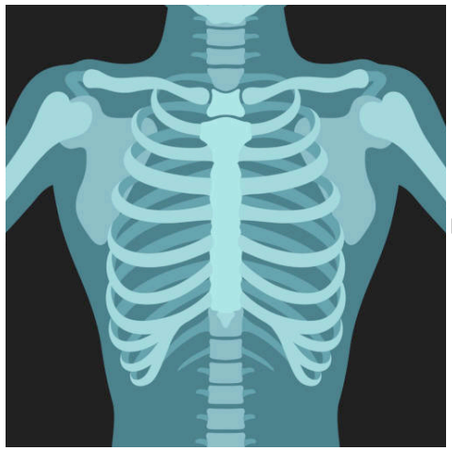


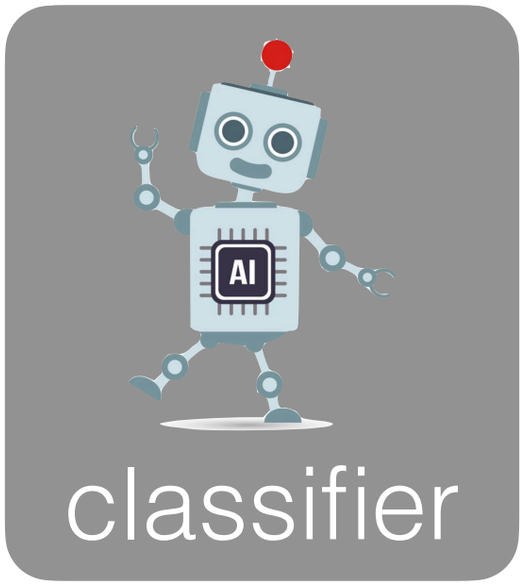
$\hat{y}$

abstain

learning to reject

input features





?

$\hat{y}$

abstain

learning to reject

- Learning to reject
  - **Bayes decision theory**
  - consistent surrogate loss

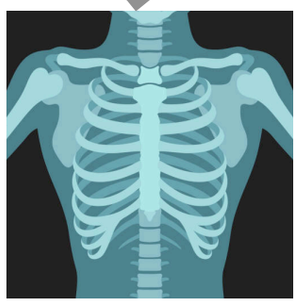
- Learning to defer to an expert
  - Bayes decision theory
  - consistent surrogate loss
  - extension to multiple experts

Derivation from Bayes decision theory

$$\delta : \mathcal{X} \rightarrow \{ \mathcal{Y} \cup \perp \}$$

Derivation from Bayes decision theory

$$\delta : \mathcal{X} \rightarrow \{ \mathcal{Y} \cup \perp \}$$



$\hat{\mathbf{y}}$

abstain

Derivation from Bayes decision theory

$$\delta : \mathcal{X} \rightarrow \{\mathcal{Y} \cup \perp\}$$

$$\mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}\right) = \begin{cases} 0 & \textbf{if} \quad \delta(x) = y \\ 1 & \textbf{if} \quad \delta(x) \neq y \\ \lambda & \textbf{if} \quad \delta(x) = \perp \end{cases}$$

Derivation from Bayes decision theory

$$\delta : \mathcal{X} \rightarrow \{\mathcal{Y} \cup \perp\}$$

$$\mathrm{c}\left(\delta(\mathrm{x}), \mathrm{y}\right) = \begin{cases} 0 & \textbf{if} \quad \delta(x) = y \\ 1 & \textbf{if} \quad \delta(x) \neq y \\ \lambda & \textbf{if} \quad \delta(x) = \perp \end{cases}$$

cost of rejecting

$\lambda \in (0, 1)$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}\right)$$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}\right)$$

when we choose to make a prediction…

$$\delta(\mathsf{x}) = \hat{\mathsf{y}} = \underset{\mathsf{y}}{\arg\max}\, \mathbb{P}(\mathsf{y} \,|\, \mathsf{x})$$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}\right)$$

when we choose to make a prediction…

$$\delta(\mathsf{x}) = \hat{\mathsf{y}} = \arg\max_{\mathsf{y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x})$$

$$\mathfrak{R}(\mathsf{x}, \hat{\mathsf{y}}) \;=\; \sum_{\mathsf{y} \neq \hat{\mathsf{y}}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \;=\; 1 - \mathbb{P}(\hat{\mathsf{y}} \,|\, \mathsf{x})$$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}\right)$$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}\right)$$

when we choose to reject…

$$\delta(\mathsf{x}) = \bot$$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}\right)$$

when we choose to reject…

$$\delta(\mathsf{x}) = \bot$$

$$\mathfrak{R}(\mathsf{x}, \bot\,) = \sum_{\mathsf{y} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \lambda = \lambda$$

Derivation from Bayes decision theory

Thus we need to choose the action with the minimum risk:

$$\mathfrak{R}(\mathrm{x}, \hat{\mathrm{y}}) \;=\; 1 - \mathbb{P}(\hat{\mathrm{y}} \,|\, \mathrm{x})$$

VS

$$\mathfrak{R}(\mathrm{x}, \perp) = \lambda$$

Derivation from Bayes decision theory

$$\delta^*(x) = \begin{cases} \arg\max_y \mathbb{P}(y \mid x) & \text{if } 1 - \max_y \mathbb{P}(y \mid x) \leq \lambda \\ \perp & \text{otherwise} \end{cases}$$

Derivation from Bayes decision theory

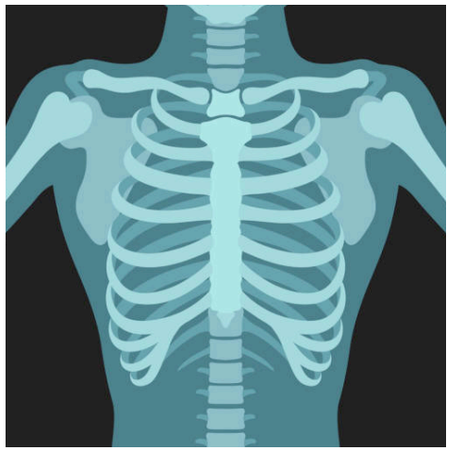
$$\delta^*(\mathsf{x}) = \begin{cases} \arg\max_{\mathsf{y}} \mathbb{P}(\mathsf{y} \mid \mathsf{x}) & \text{if } 1 - \max_{\mathsf{y}} \mathbb{P}(\mathsf{y} \mid \mathsf{x}) \leq \lambda \\ \perp & \text{otherwise} \end{cases}$$

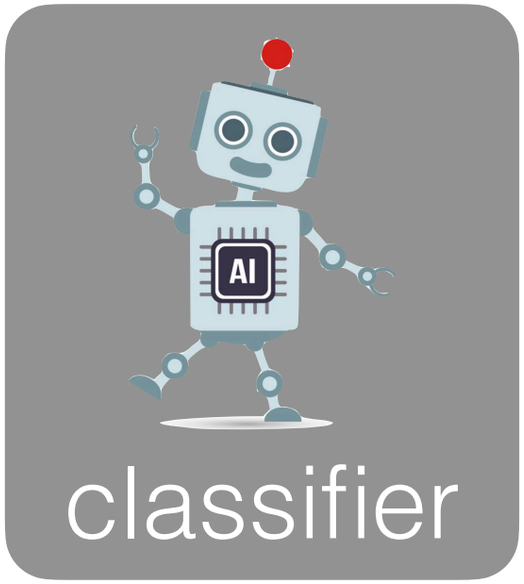
rearranging a bit…

$$\delta^*(\mathsf{x}) = \begin{cases} \arg\max_{\mathsf{y}} \mathbb{P}(\mathsf{y} \mid \mathsf{x}) & \text{if } \max_{\mathsf{y}} \mathbb{P}(\mathsf{y} \mid \mathsf{x}) \geq 1 - \lambda \\ \perp & \text{otherwise} \end{cases}$$

learning to reject

input features



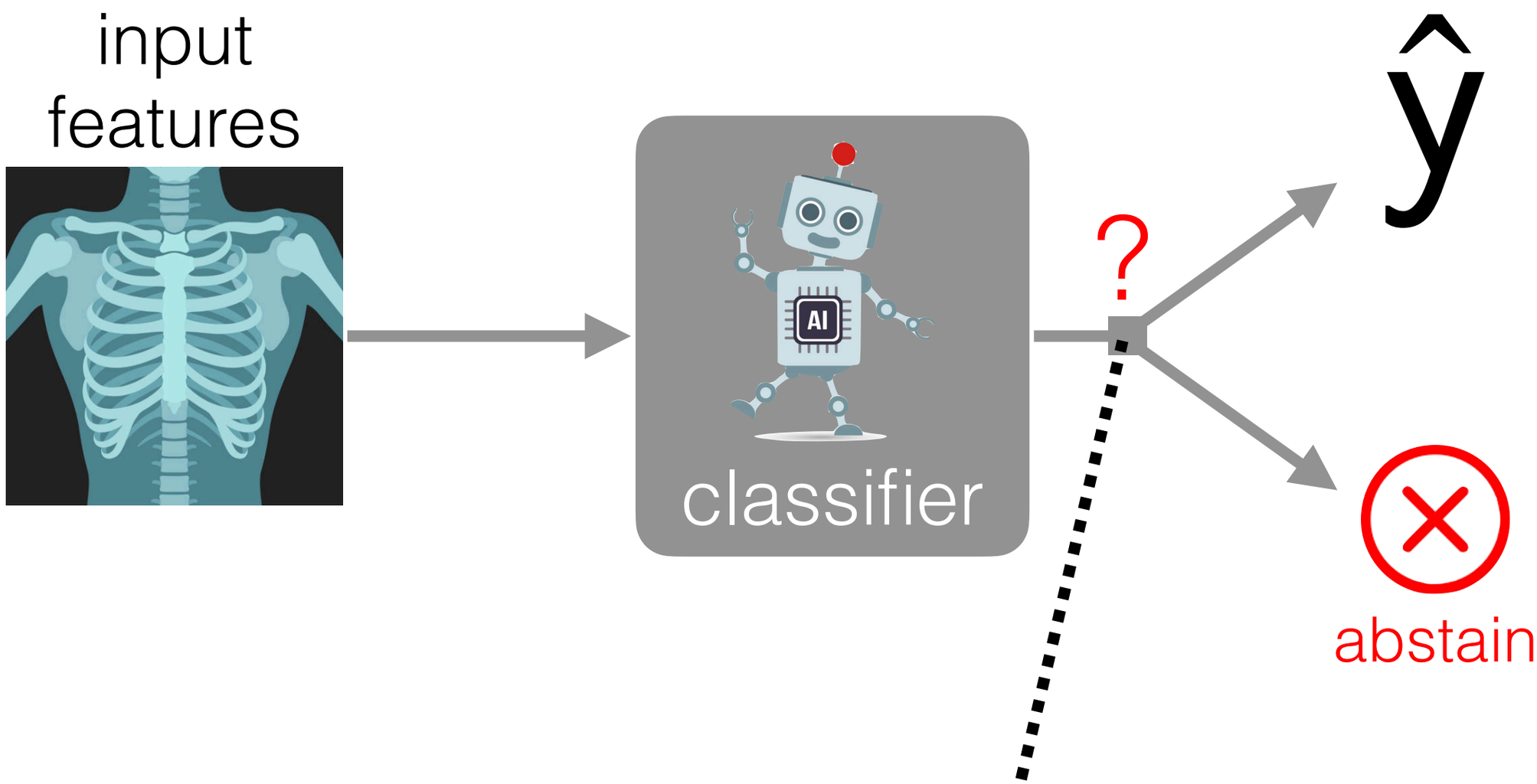


?

$\hat{y}$

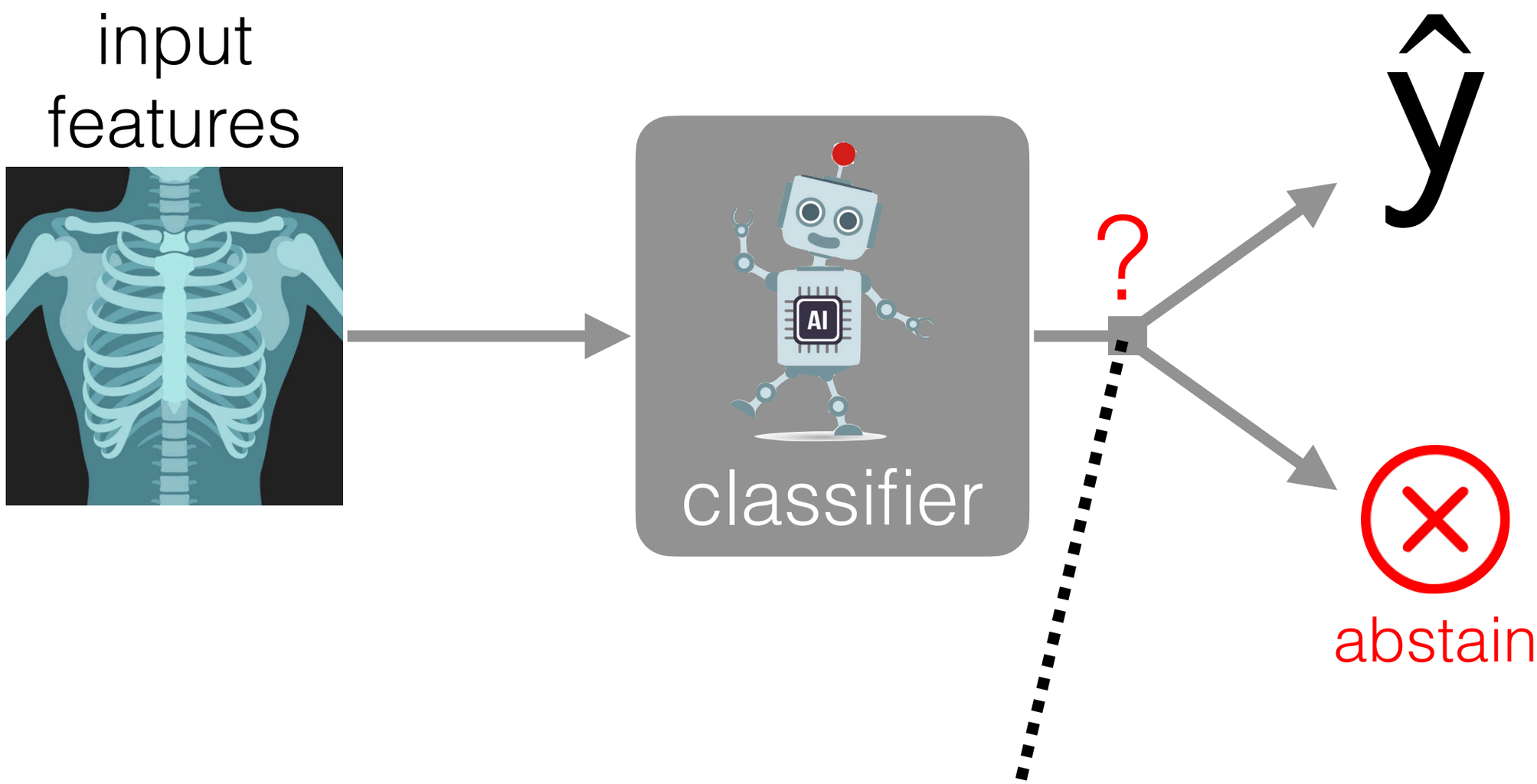
abstain

learning to reject



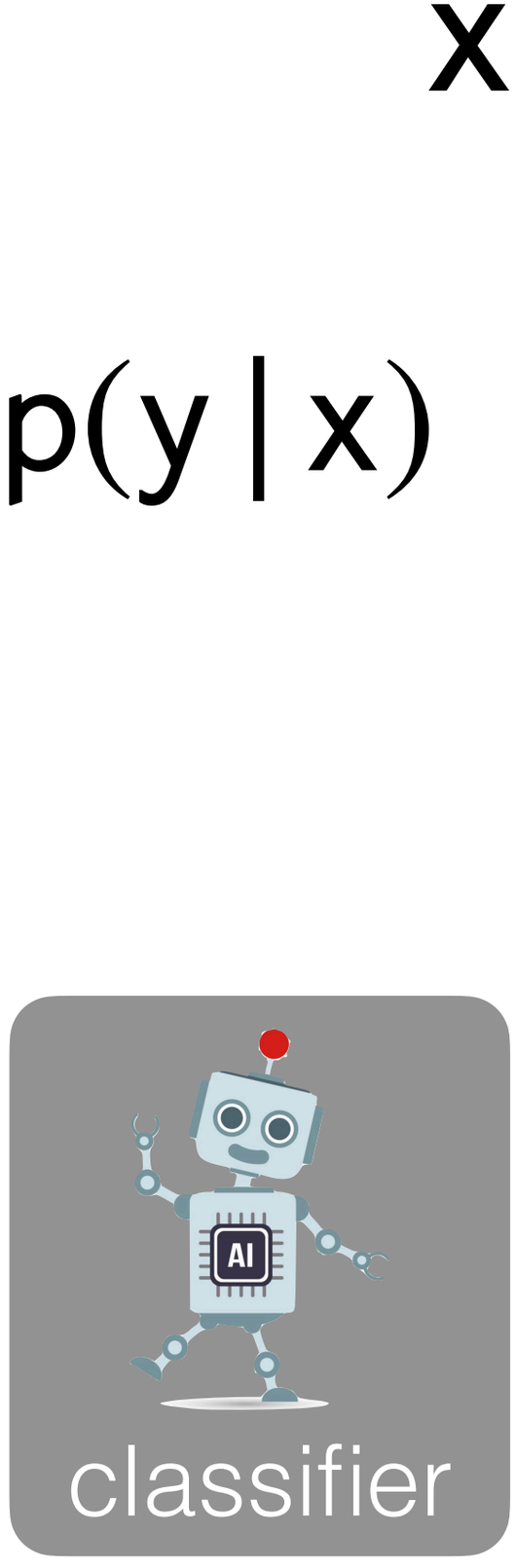
$$\delta^*(x) = \begin{cases} \hat{y} & \text{if } \max_y \mathbb{P}(y \mid x) \geq 1 - \lambda \\ \perp & \text{otherwise} \end{cases}$$

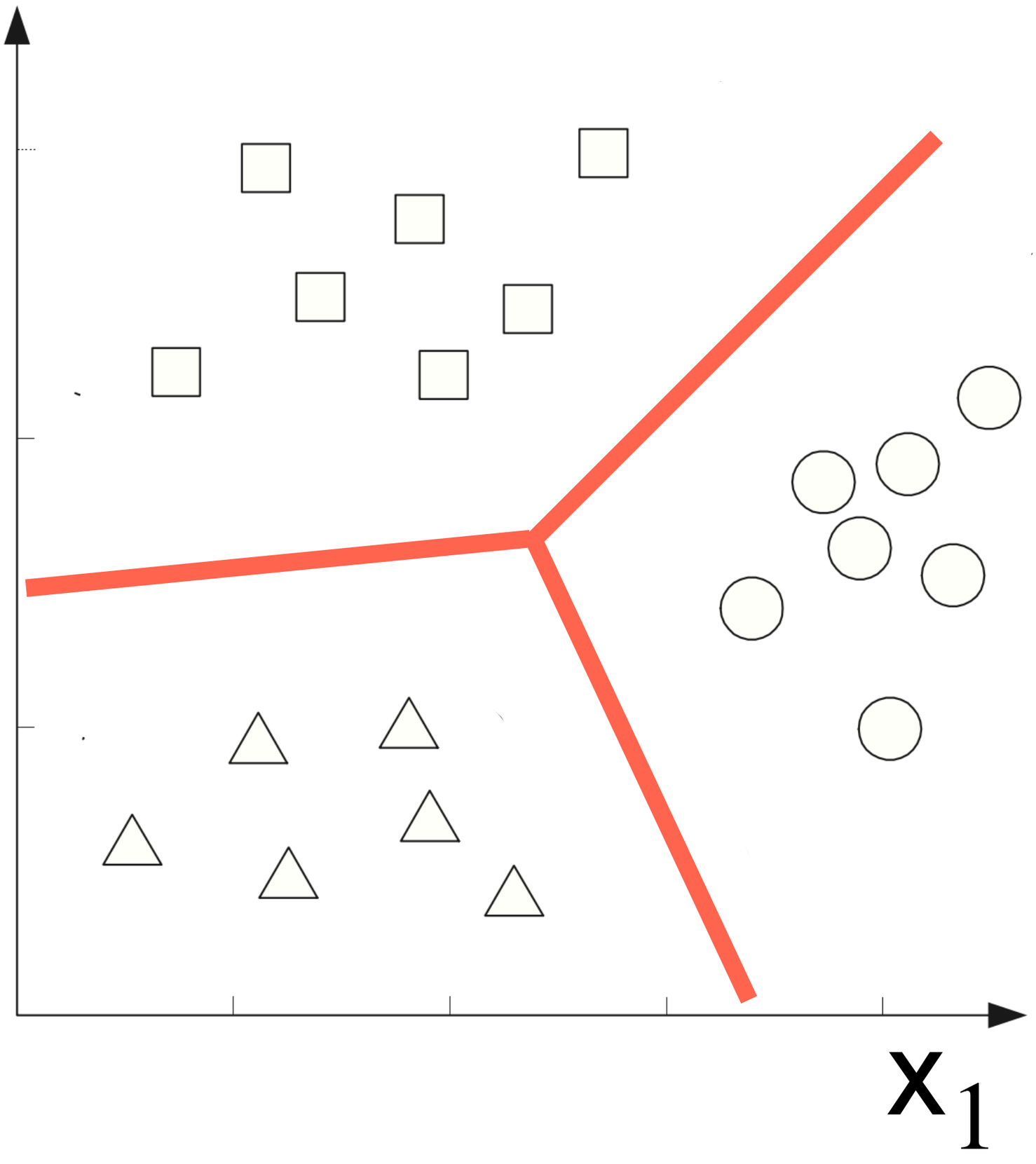
learning to reject

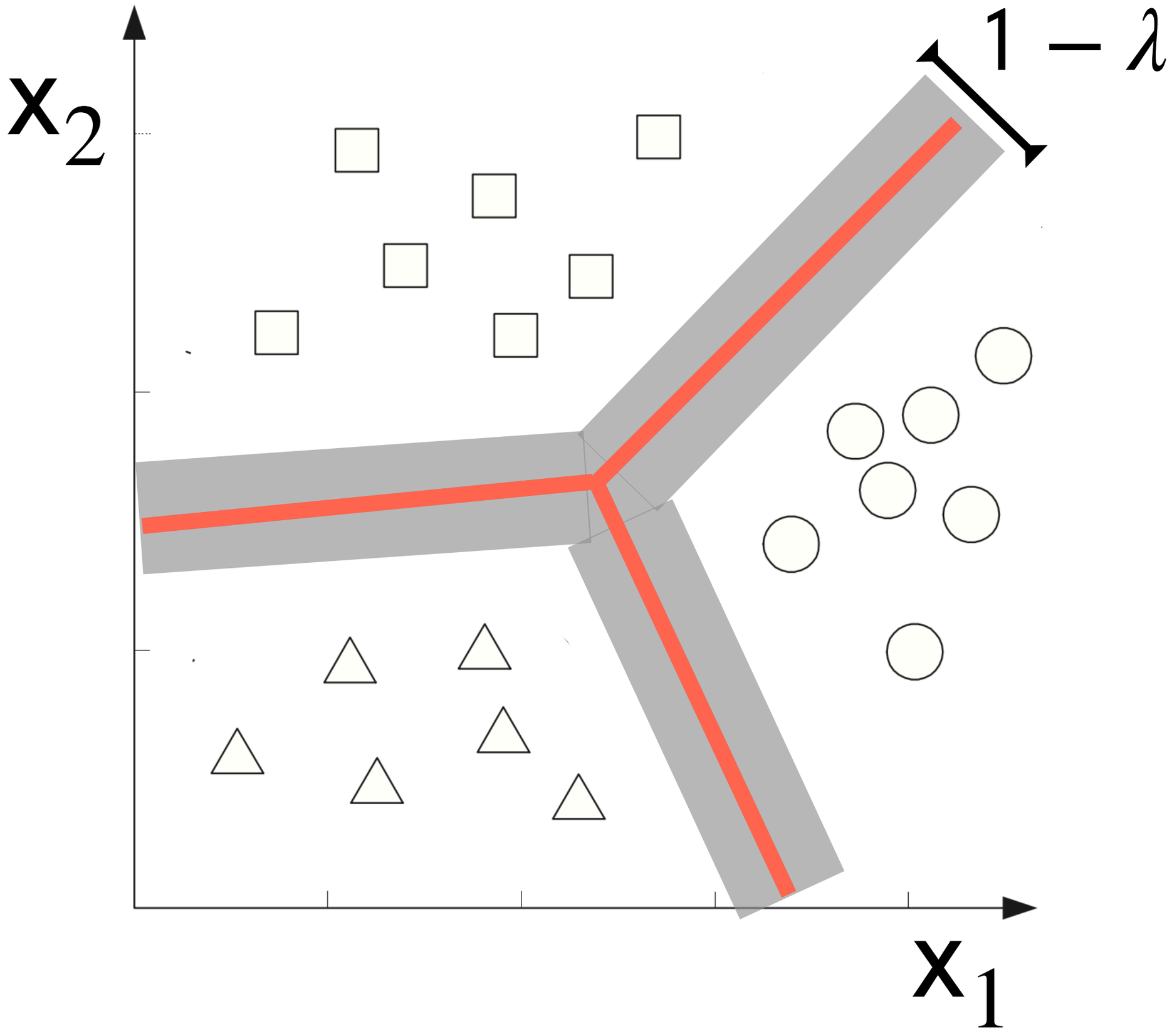


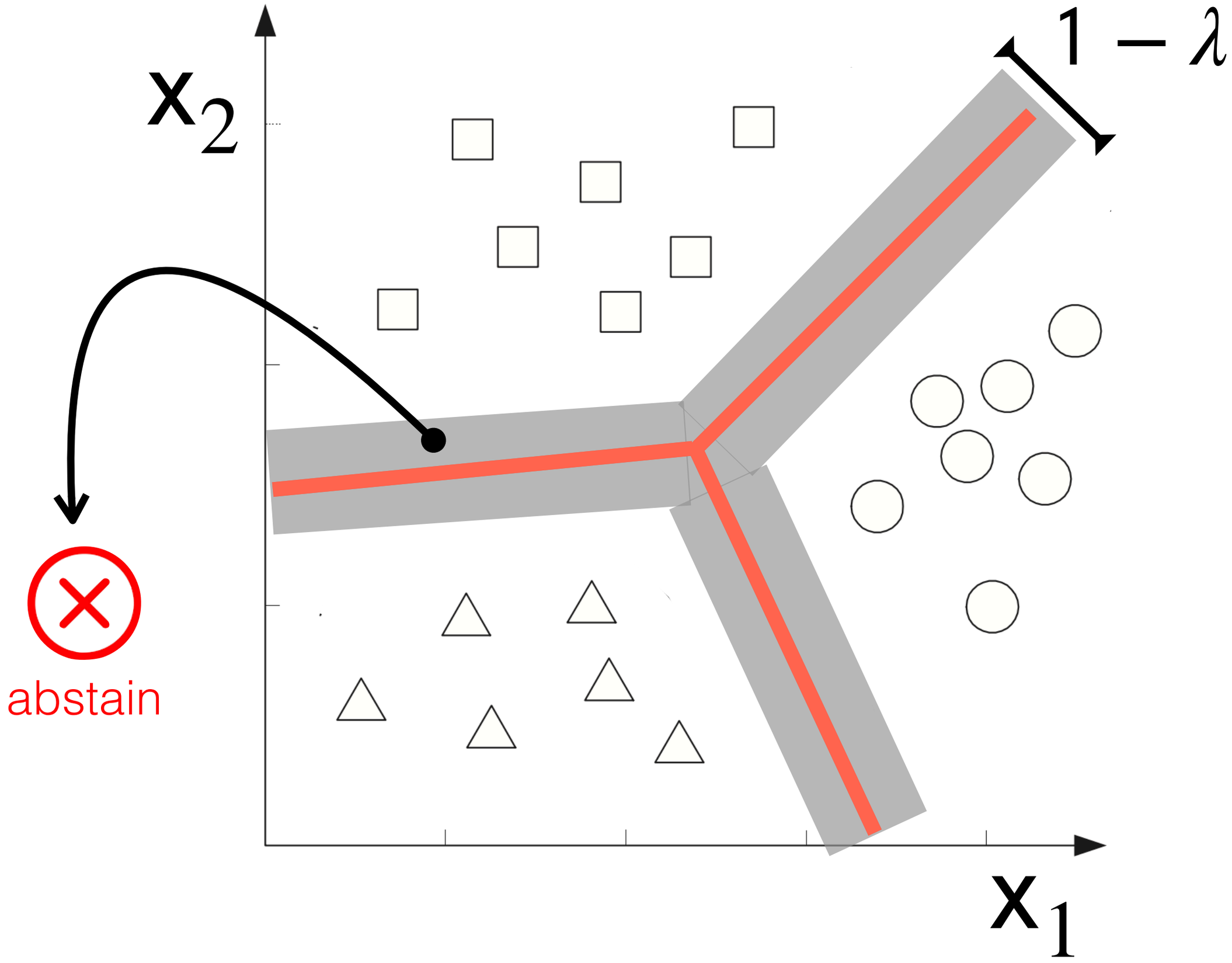
$$\hat{\delta}(x) = \begin{cases} \hat{y} & \text{if } \max_y p(y \mid x) \geq 1 - \lambda \\ \perp & \text{otherwise} \end{cases}$$

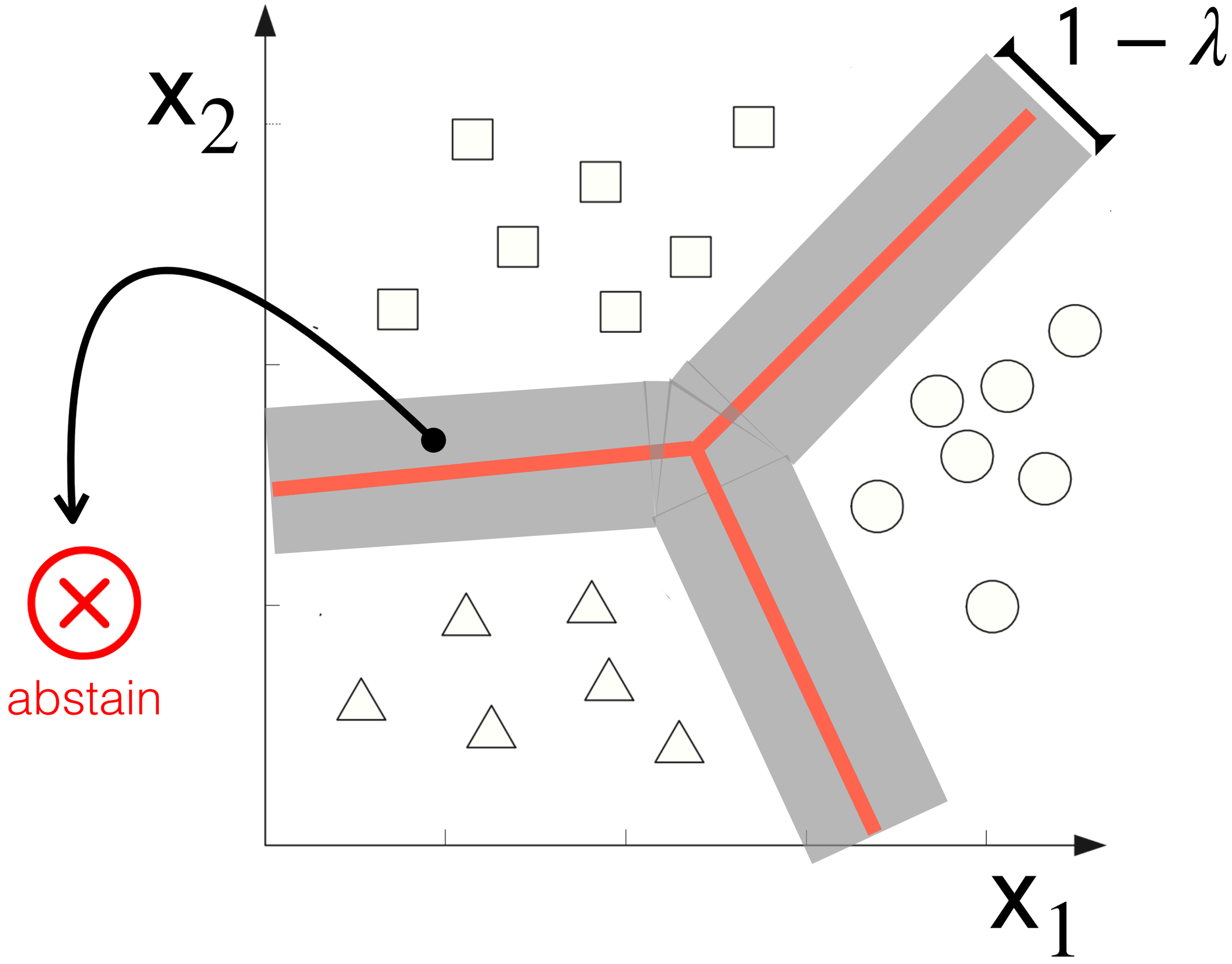
$p(y \mid x)$





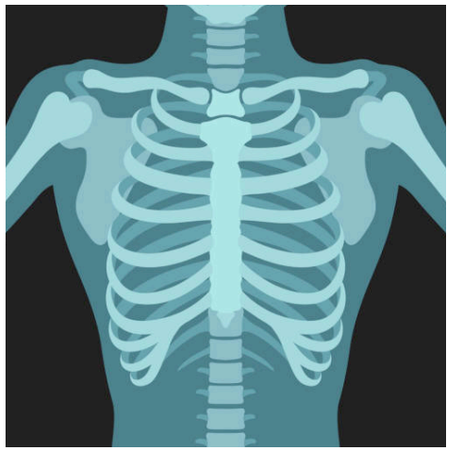

x_2
$1 - \lambda$
x_1

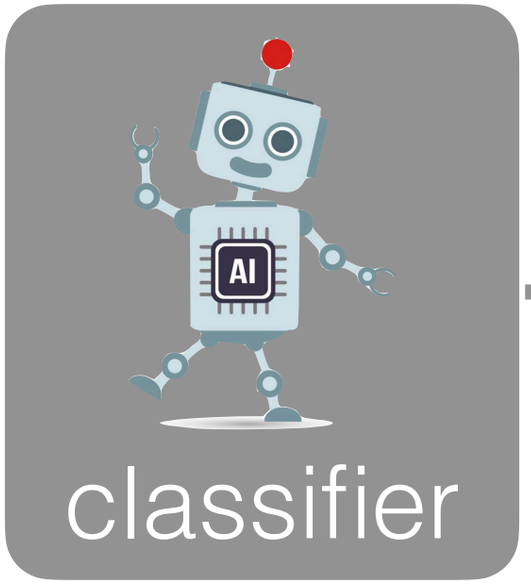

x_2
$1-\lambda$
abstain
x_1


x_2
$1-\lambda$
abstain
x_1

learning to reject

input features





$\hat{y}$

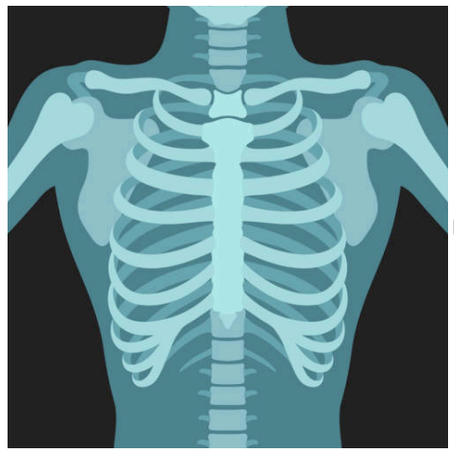
abstain

Any Problems?

$$\hat{\delta}(x) = \begin{cases} \hat{y} & \text{if } \max_y p(y \mid x) \geq 1 - \lambda \\ \perp & \text{otherwise} \end{cases}$$

learning to reject

input features



$\hat{y}$

abstain

Need p(y|x) to be a really good approximation of the true class conditional probabilities

$$\hat{\delta}(x) = \begin{cases} \hat{y} & \text{if } \max_y p(y \mid x) \geq 1 - \lambda \\ \perp & \text{otherwise} \end{cases}$$

- Learning to reject
  - Bayes decision theory
  - consistent surrogate loss
- Learning to defer to an expert
  - Bayes decision theory
  - consistent surrogate loss
  - extension to multiple experts

developing a surrogate loss

$$\hat{\delta}(x) = \begin{cases} \hat{y} & \text{if } \max_y p(y \mid x) \geq 1 - \lambda \\ \perp & \text{otherwise} \end{cases}$$

key idea: the decision rule above is valid for *any* choice of λ. If we can choose a specific λ ahead of time, we only need to calibrate the model around that λ.

developing a surrogate loss

$$\hat{\delta}(x) = \begin{cases} \hat{y} & \text{if } \max_y p(y \mid x) \geq 1 - \lambda \\ \perp & \text{otherwise} \end{cases}$$

key idea: the decision rule above is valid for *any* choice of λ. If we can choose a specific λ ahead of time, we only need to calibrate the model around that λ.

$$\text{if } \mathbb{P}(y \mid x) \geq 1 - \lambda, \text{ then } p(y \mid x) \geq 1 - \lambda$$

$$\text{if } \mathbb{P}(y \mid x) < 1 - \lambda, \text{ then } p(y \mid x) < 1 - \lambda$$

softmax implementation

softmax implementation

training data

$$\mathfrak{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n \right\}_{n=1}^{N}$$

softmax implementation

training data

$$\mathfrak{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n \right\}_{n=1}^{N}$$

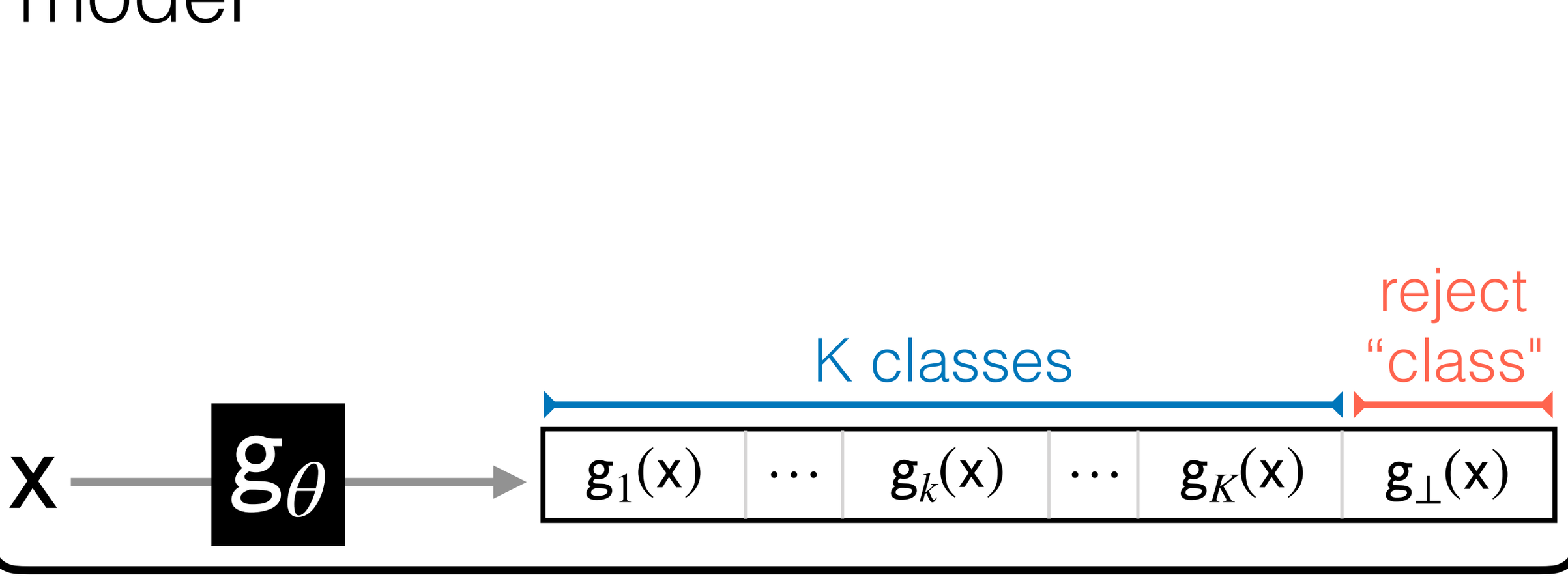
model

softmax implementation

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n \right\}_{n=1}^{N}$$

model



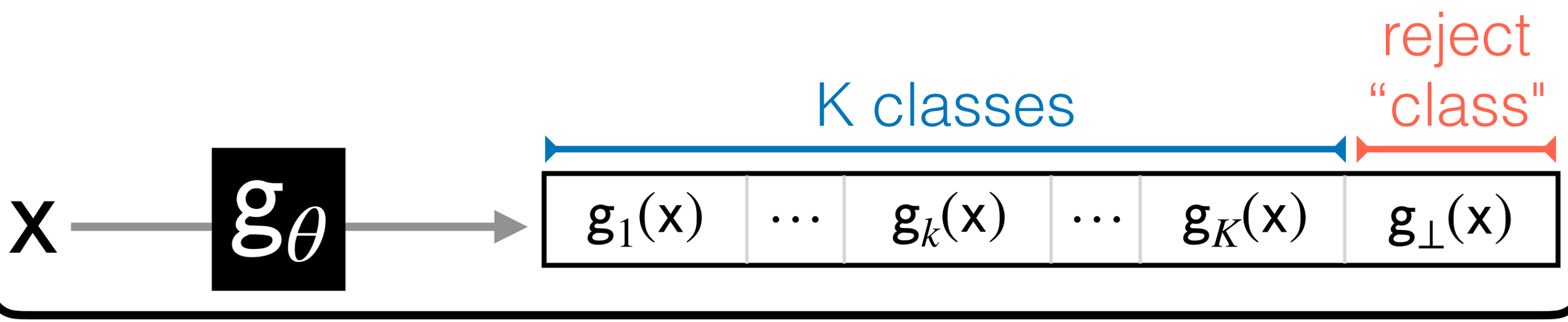
softmax implementation

training data

$$\mathfrak{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n \right\}_{n=1}^{N}$$

model

$$\mathbf{g}_k(\mathbf{x}) \in \mathbb{R}$$

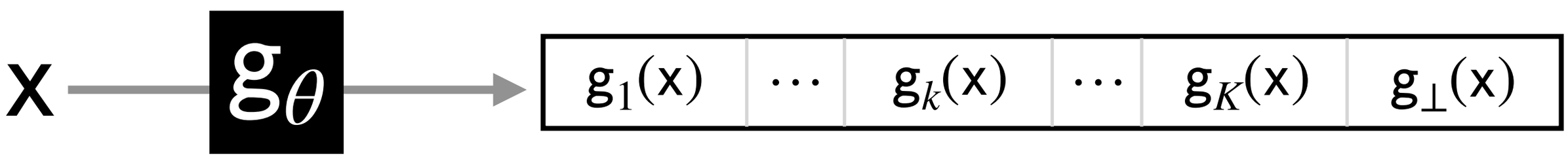


softmax implementation

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n \right\}_{n=1}^{N}$$

model

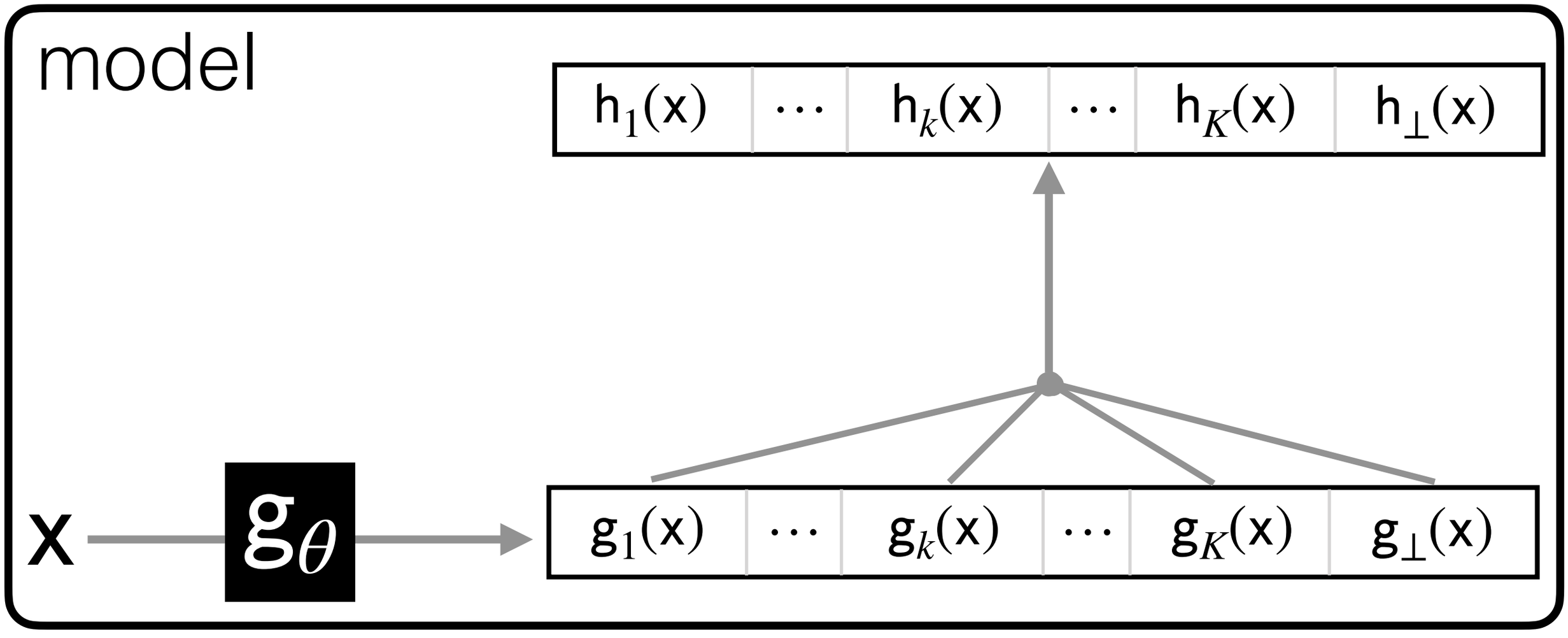


softmax implementation

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n \right\}_{n=1}^{N}$$

model



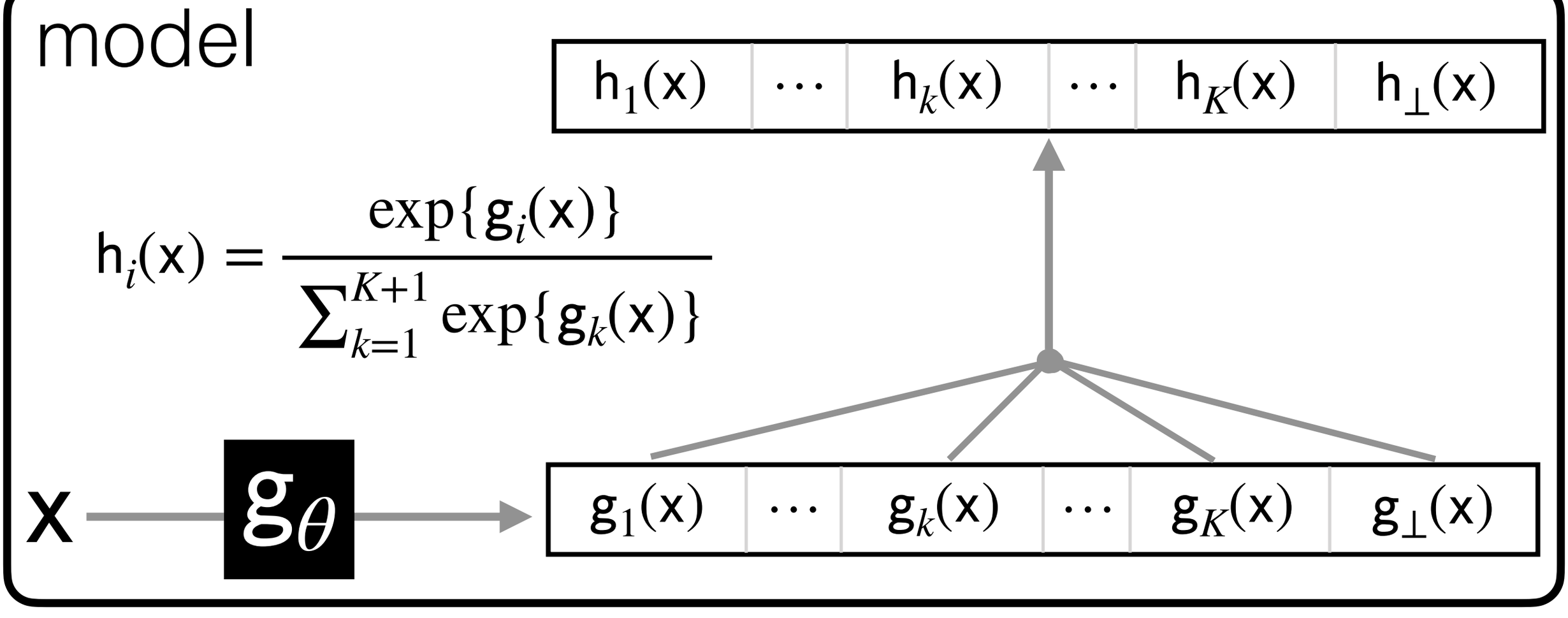
softmax implementation

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathbf{x}) = \frac{\exp\{\mathsf{g}_i(\mathbf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathbf{x})\}}$$



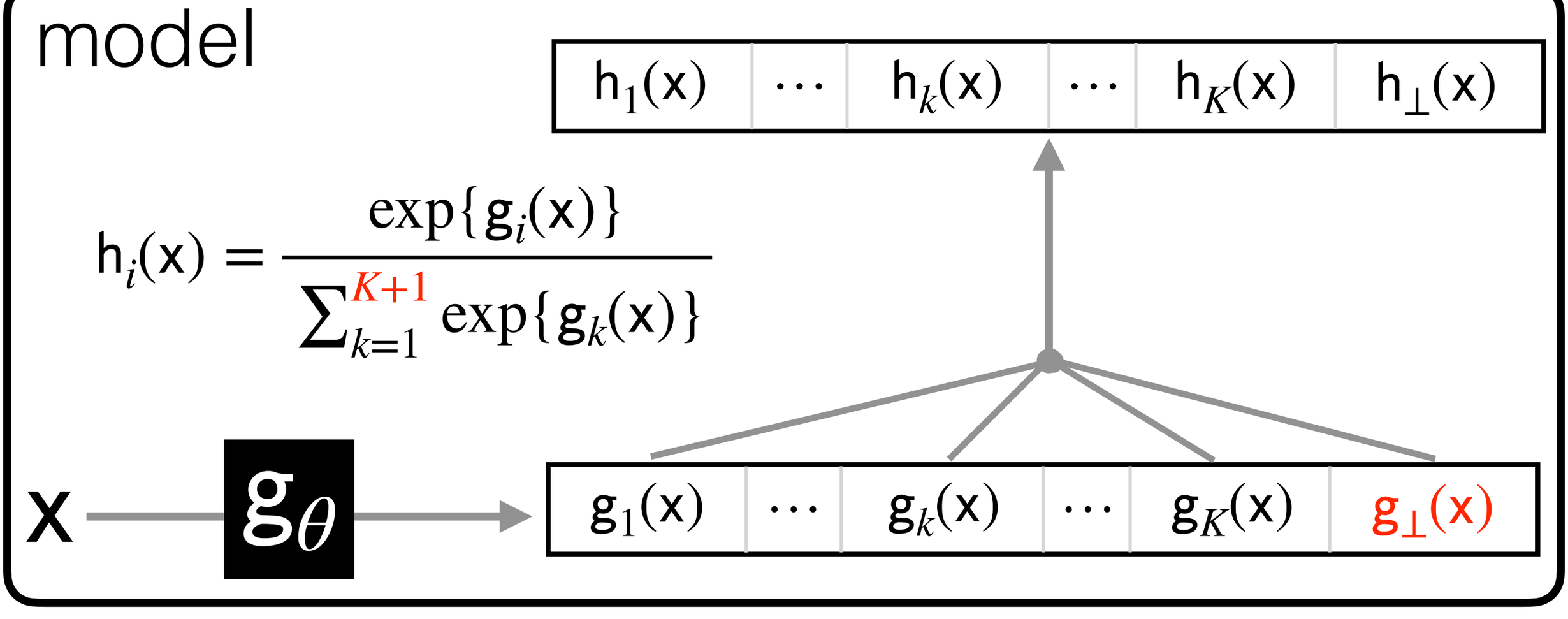
softmax implementation

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathbf{x}) = \frac{\exp\{\mathsf{g}_i(\mathbf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathbf{x})\}}$$



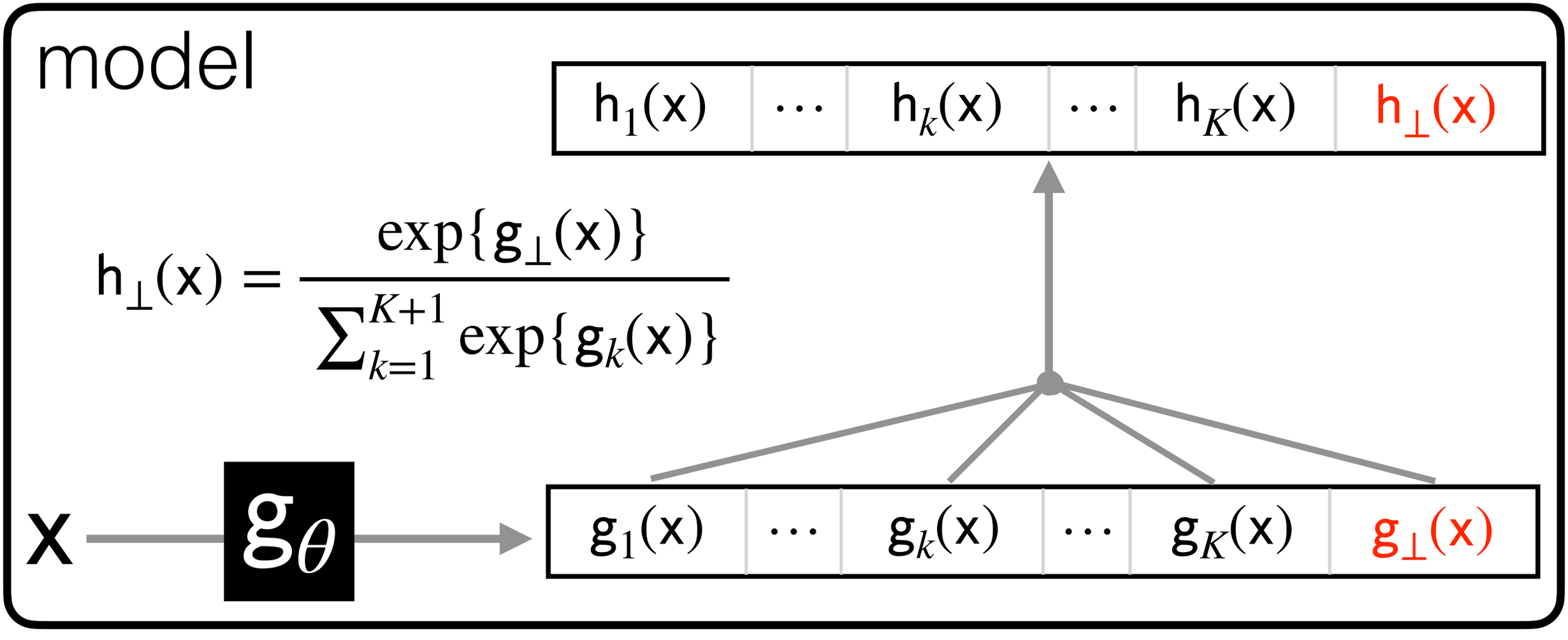
softmax implementation

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_\perp(\mathbf{x}) = \frac{\exp\{\mathsf{g}_\perp(\mathbf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathbf{x})\}}$$



softmax implementation

training data

$$\mathcal{D} = \left\{ \mathbf{x}_n, \mathbf{y}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathbf{x}) = \frac{\exp\{\mathsf{g}_i(\mathbf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathbf{x})\}}$$

softmax implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \lambda) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; (1-\lambda) \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

softmax implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \lambda) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; (1-\lambda) \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

softmax implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \lambda) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; (1-\lambda) \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

softmax implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n \right\}_{n=1}^{N}$$

model

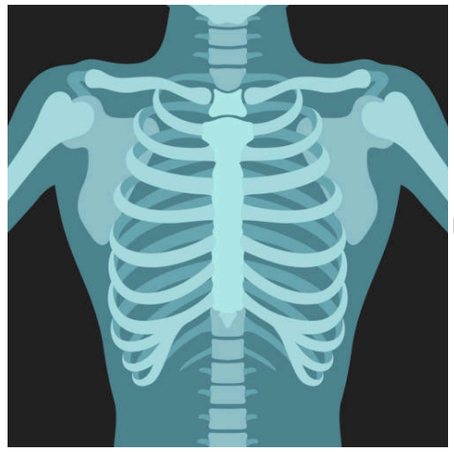
$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \lambda) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; (1-\lambda) \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

learning to reject

input features



$\hat{y}$

abstain

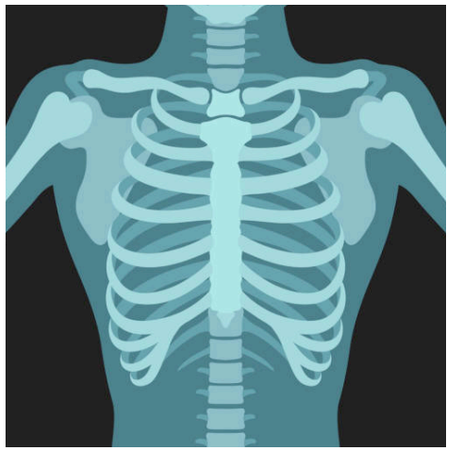
reject if…

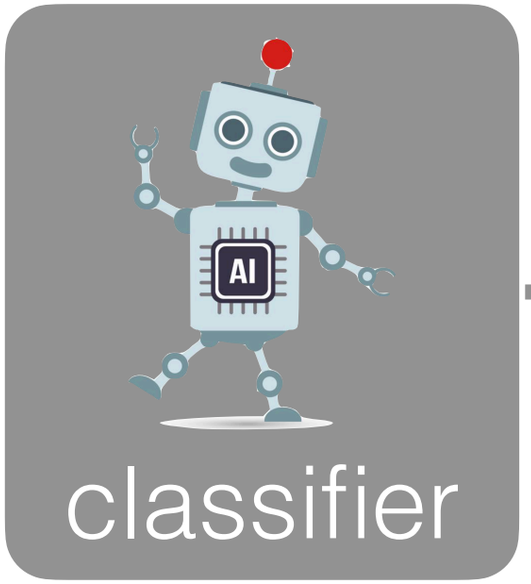
$$\max_{y \in [1,K]} h_y(x) < h_\perp(x)$$

- Learning to reject
  - Bayes decision theory
  - consistent surrogate loss
- Learning to defer to an expert
  - Bayes decision theory
  - consistent surrogate loss
  - extension to multiple experts

learning to reject

input features





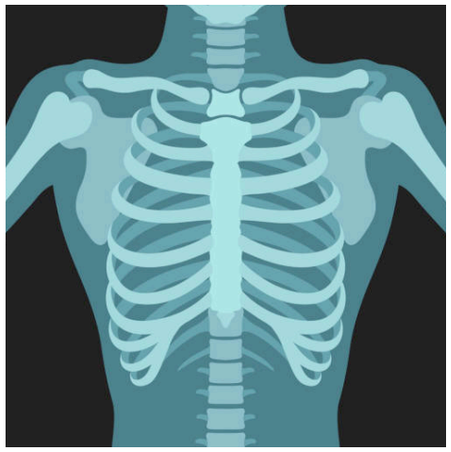
$\hat{y}$

abstain

What are the limitations of learning to reject?

learning to reject

input features



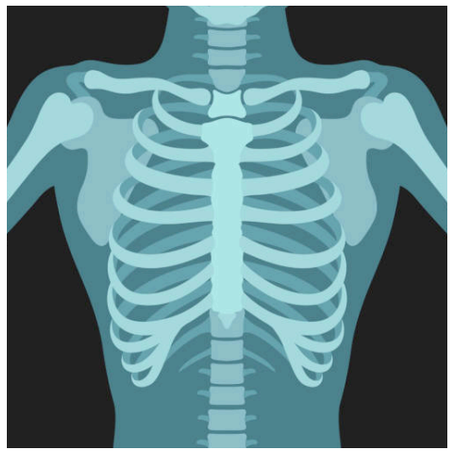
$\hat{y}$

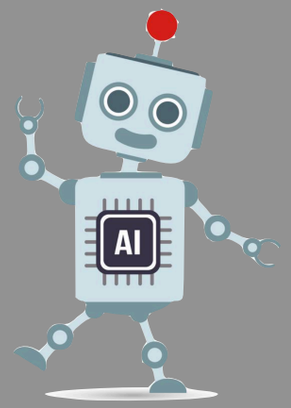
abstain

What happens after the model rejects?

What are the limitations of learning to reject?

input features



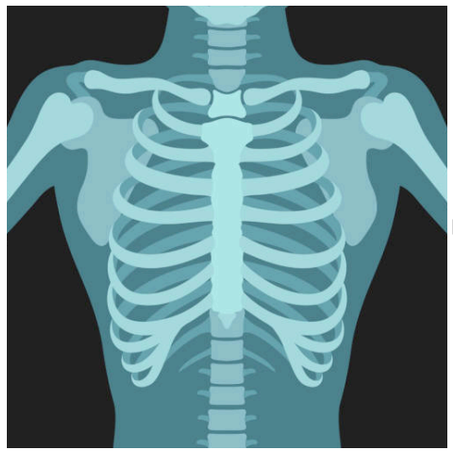


classifier

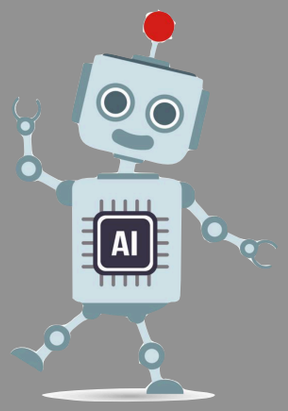
expert

learning to defer (to an expert)

input features



allocation mechanism

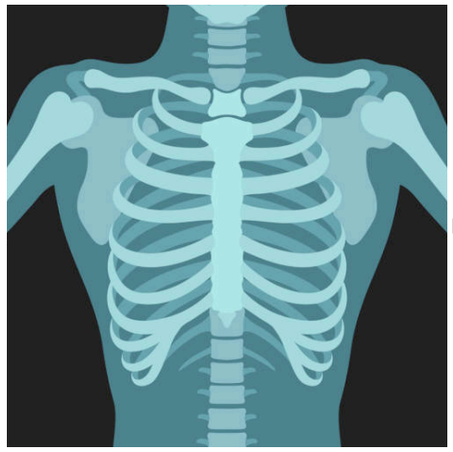


classifier

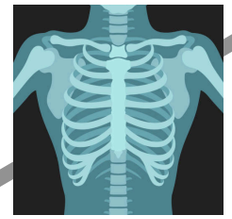
expert

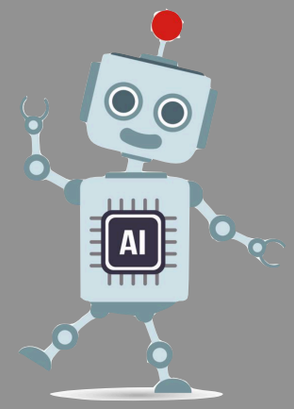
learning to defer (to an expert)

input features



allocation mechanism



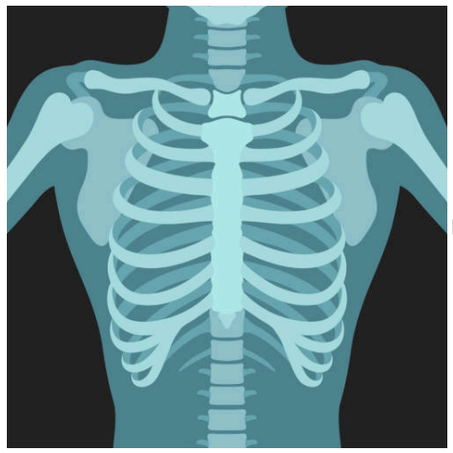


classifier

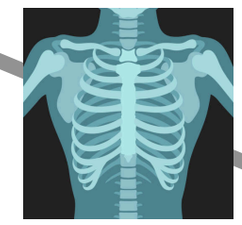
expert

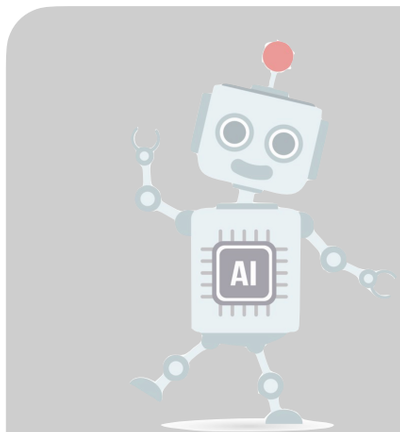
learning to defer (to an expert)

input features



allocation mechanism



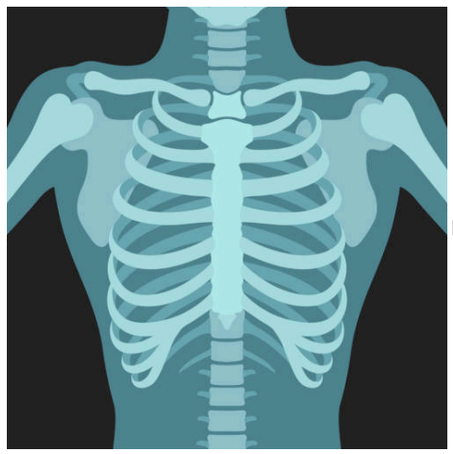


classifier

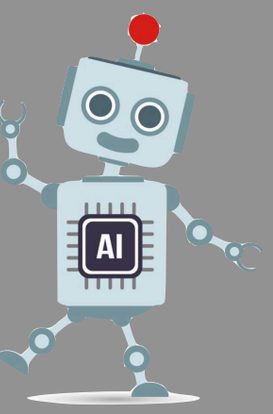
expert

learning to defer (to an expert)

input features



allocation mechanism



classifier

expert

input features

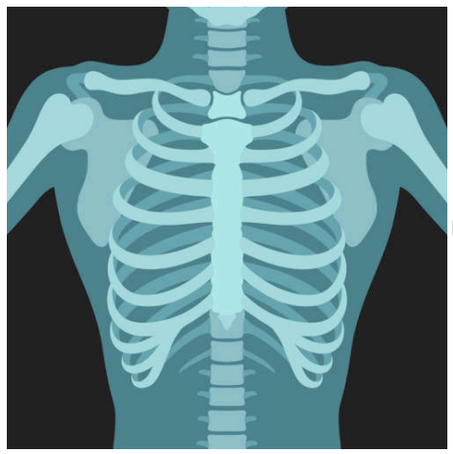
x

(black box)

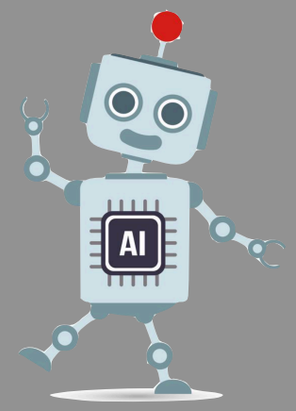
m

prediction

input features



allocation mechanism



classifier

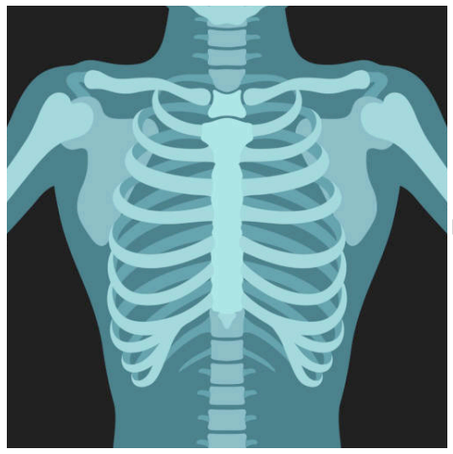
expert

now we will incorporate information about what will happen if the model doesn't predict and the human does instead.

- Learning to reject
  - Bayes decision theory
  - consistent surrogate loss

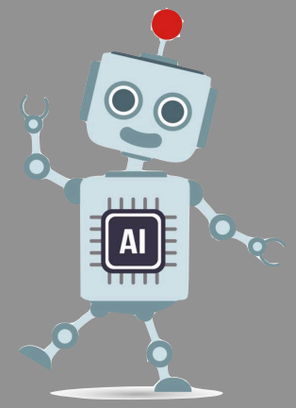
- Learning to defer to an expert
  - **Bayes decision theory**
  - consistent surrogate loss
  - extension to multiple experts

input features



allocation mechanism

???



classifier

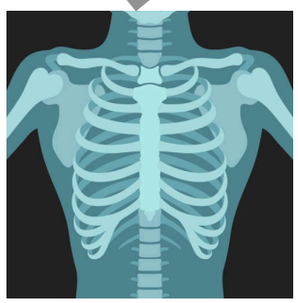
expert

Derivation from Bayes decision theory

$$\delta : \mathcal{X} \rightarrow \{ \mathcal{Y} \cup \perp \}$$

Derivation from Bayes decision theory

$$\delta : \mathcal{X} \rightarrow \{ \mathcal{Y} \cup \perp \}$$



$\hat{\mathbf{y}}$

expert

Derivation from Bayes decision theory

$$\delta : \mathscr{X} \to \{\mathscr{Y} \cup \perp\}$$

$$\mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}, \mathsf{m}\right) = \begin{cases} 0 & \text{if } \delta(\mathsf{x}) = \mathsf{y} \\ 1 & \text{if } \delta(\mathsf{x}) \neq \mathsf{y} \\ \mathbb{I}[\mathsf{m} \neq \mathsf{y}] & \text{if } \delta(\mathsf{x}) = \perp \end{cases}$$

Derivation from Bayes decision theory

$$\delta : \mathcal{X} \to \{\mathcal{Y} \cup \perp\}$$

$$c\left(\delta(x), y, m\right) = \begin{cases} 0 & \text{if } \delta(x) = y \\ 1 & \text{if } \delta(x) \neq y \\ \mathbb{I}[m \neq y] & \text{if } \delta(x) = \perp \end{cases}$$

0-1 loss applied to expert's prediction

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \sum_{\mathsf{m} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathbb{P}(\mathsf{m} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}, \mathsf{m}\right)$$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \sum_{\mathsf{m} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathbb{P}(\mathsf{m} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}, \mathsf{m}\right)$$

when we choose to make a prediction…

$$\delta(\mathsf{x}) = \hat{\mathsf{y}} = \arg\max_{\mathsf{y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x})$$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \sum_{\mathsf{m} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathbb{P}(\mathsf{m} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}, \mathsf{m}\right)$$

when we choose to make a prediction…

$$\delta(\mathsf{x}) = \hat{\mathsf{y}} = \arg\max_{\mathsf{y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x})$$

$$\mathfrak{R}(\mathsf{x}, \hat{\mathsf{y}}) \;=\; \sum_{\mathsf{y} \neq \hat{\mathsf{y}}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \;=\; 1 - \mathbb{P}(\hat{\mathsf{y}} \,|\, \mathsf{x})$$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \sum_{\mathsf{m} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathbb{P}(\mathsf{m} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}, \mathsf{m}\right)$$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \sum_{\mathsf{m} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathbb{P}(\mathsf{m} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}, \mathsf{m}\right)$$

when we choose to defer…

$$\delta(\mathsf{x}) = \bot$$

Derivation from Bayes decision theory

$$\mathfrak{R}(\mathsf{x}, \delta) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \sum_{\mathsf{m} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathbb{P}(\mathsf{m} \,|\, \mathsf{x}) \cdot \mathsf{c}\left(\delta(\mathsf{x}), \mathsf{y}, \mathsf{m}\right)$$

when we choose to defer…

$$\delta(\mathsf{x}) = \bot$$

$$\mathfrak{R}(\mathsf{x}, \bot) \;=\; \sum_{\mathsf{y} \in \mathcal{Y}} \sum_{\mathsf{m} \in \mathcal{Y}} \mathbb{P}(\mathsf{y} \,|\, \mathsf{x}) \cdot \mathbb{P}(\mathsf{m} \,|\, \mathsf{x}) \cdot \mathbb{I}[\mathsf{m} \neq \mathsf{y}]$$

$$= \;\; \mathbb{P}_{\mathsf{m},\mathsf{y}}(\mathsf{m} \neq \mathsf{y} \,|\, \mathsf{x})$$

Derivation from Bayes decision theory

Thus we need to choose the action with the minimum risk:

$$\mathfrak{R}(x, \hat{y}) = 1 - \mathbb{P}(\hat{y} \mid x)$$

VS

$$\mathfrak{R}(x, \perp) = \mathbb{P}_{m,y}(m \neq y \mid x)$$

Derivation from Bayes decision theory

$$\delta^*(x) = \begin{cases} \arg\max_y \mathbb{P}(y \mid x) & \text{if } 1 - \max_y \mathbb{P}(y \mid x) \leq \mathbb{P}_{m,y}(m \neq y \mid x) \\ \perp & \text{otherwise} \end{cases}$$

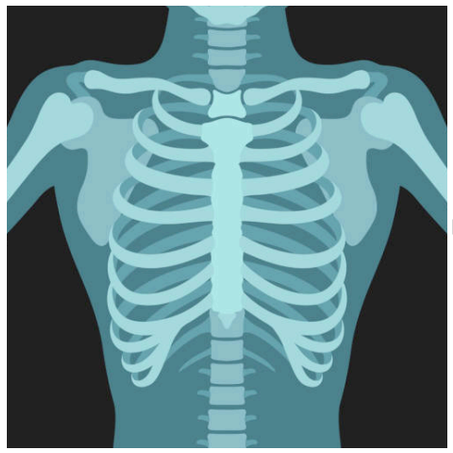
Derivation from Bayes decision theory

$$\delta^*(x) = \begin{cases} \arg\max_y \mathbb{P}(y \mid x) & \text{if } 1 - \max_y \mathbb{P}(y \mid x) \leq \mathbb{P}_{m,y}(m \neq y \mid x) \\ \perp & \text{otherwise} \end{cases}$$

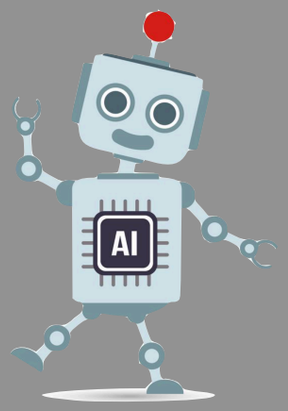
rearranging a bit…

$$\delta^*(x) = \begin{cases} \arg\max_y \mathbb{P}(y \mid x) & \text{if } \max_y \mathbb{P}(y \mid x) \geq \mathbb{P}_{m,y}(m = y \mid x) \\ \perp & \text{otherwise} \end{cases}$$

input features



allocation mechanism

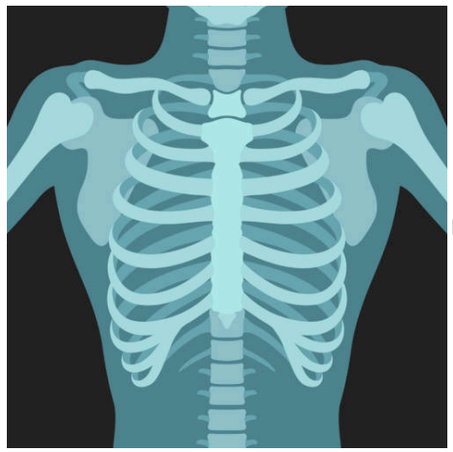


classifier

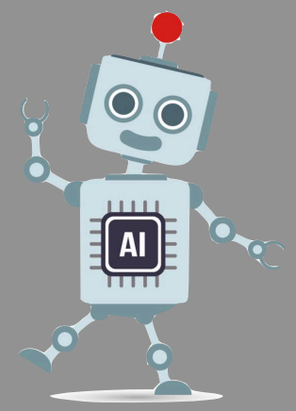
expert

$$\delta^*(x) = \begin{cases} \arg\max_y \mathbb{P}(y \mid x) & \text{if } \max_y \mathbb{P}(y \mid x) \geq \mathbb{P}_{m,y}(m = y \mid x) \\ \perp & \text{otherwise} \end{cases}$$

input features



allocation mechanism

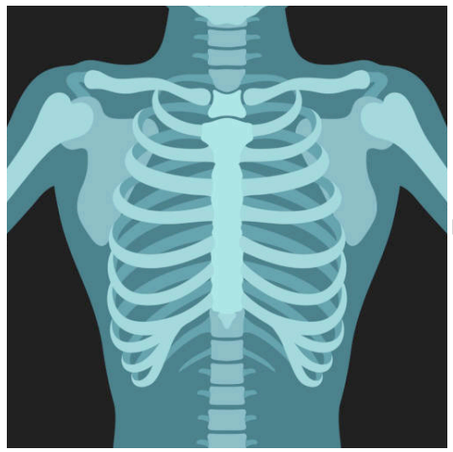


classifier

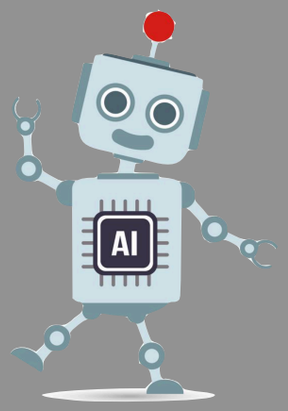
expert

$$\hat{\delta}(x) = \begin{cases} \arg\max_y p(y \mid x) & \text{if } \max_y p(y \mid x) \geq p(m = y \mid x) \\ \perp & \text{otherwise} \end{cases}$$

input features



allocation mechanism



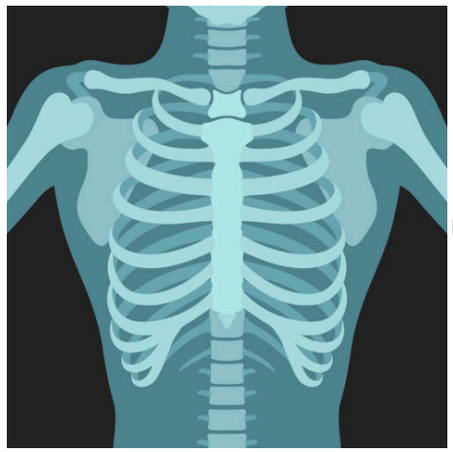
classifier

expert

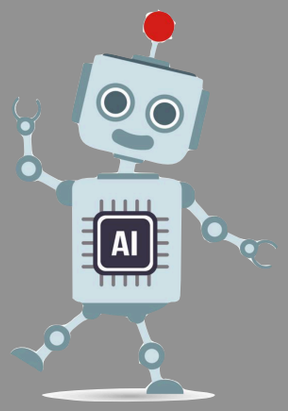
Any Problems?

$$\hat{\delta}(x) = \begin{cases} \arg\max_y p(y \mid x) & \text{if } \max_y p(y \mid x) \geq p(m = y \mid x) \\ \perp & \text{otherwise} \end{cases}$$

input features



allocation mechanism



classifier

expert

Any Problems?

$$\hat{\delta}(x) = \begin{cases} \arg\max_y p(y \mid x) & \text{if } \max_y p(y \mid x) \geq p(m = y \mid x) \\ \perp & \text{otherwise} \end{cases}$$

- Learning to reject
  - Bayes decision theory
  - consistent surrogate loss
- Learning to defer to an expert
  - Bayes decision theory
  - **consistent surrogate loss**
  - extension to multiple experts

softmax implementation [Mozannar & Sontag, 2020]

softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

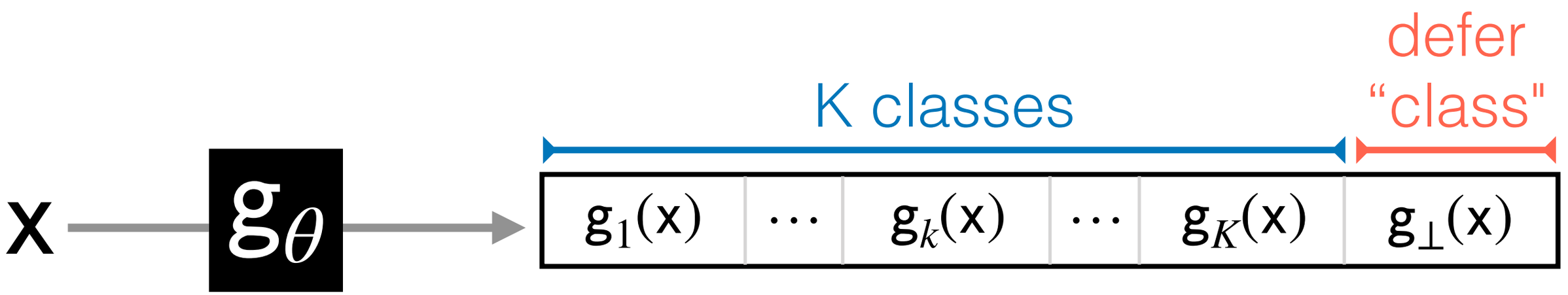
model

softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model



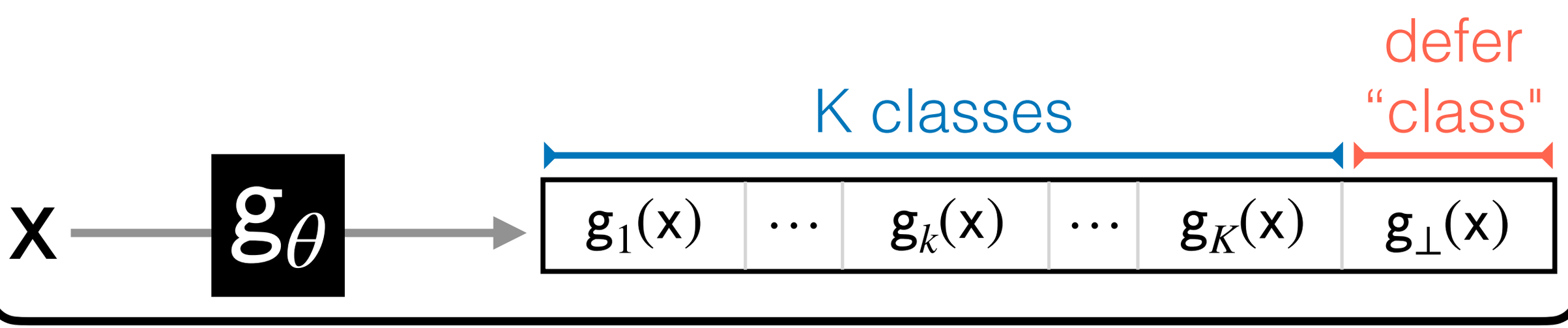
softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{g}_k(\mathsf{x}) \in \mathbb{R}$$

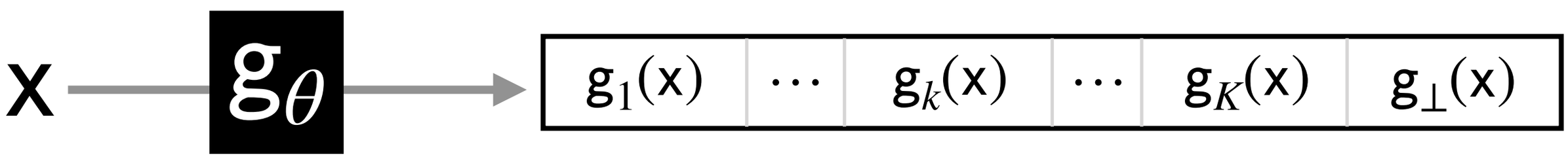


softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model

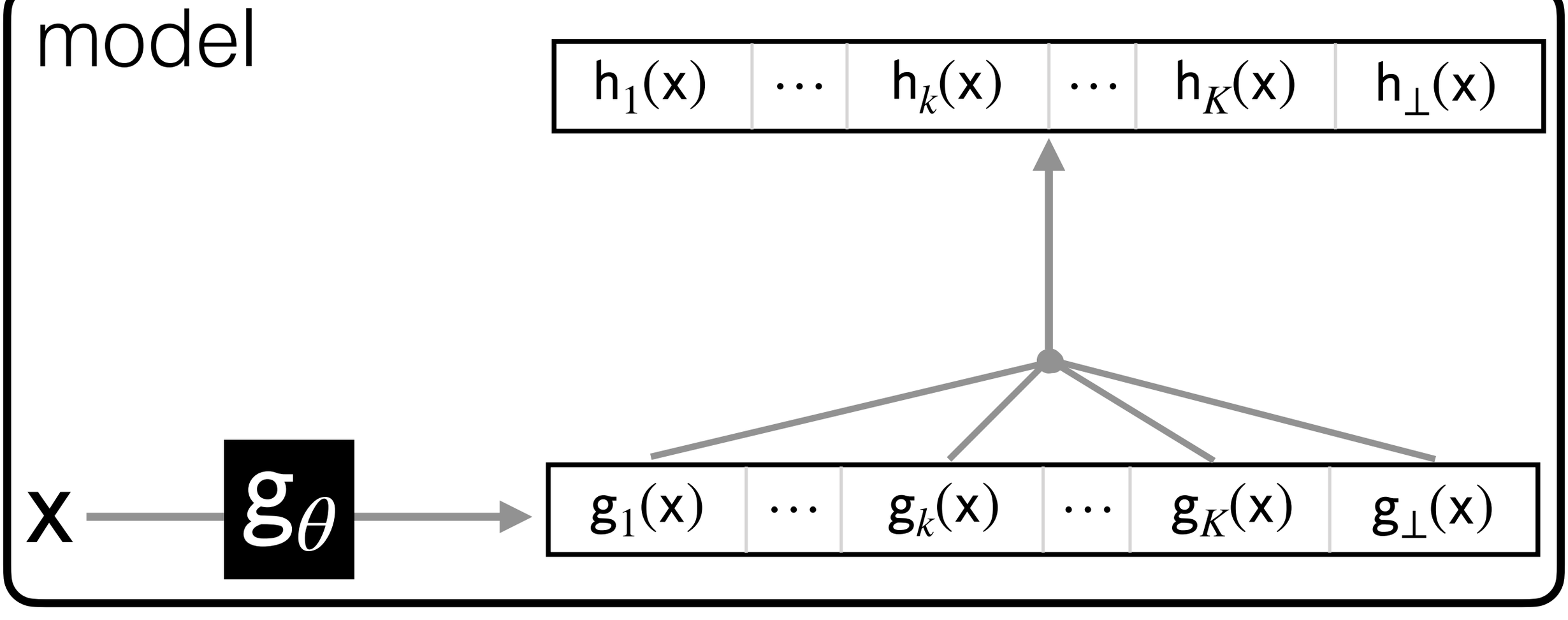


softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model



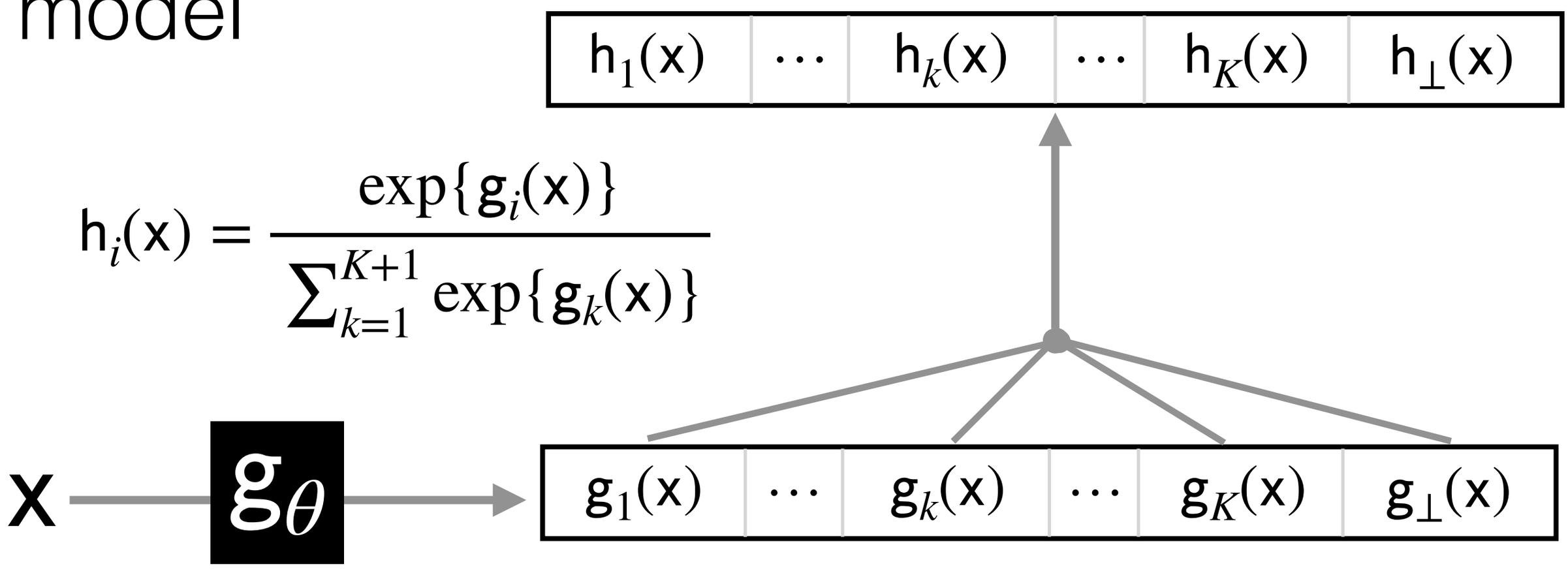
softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$



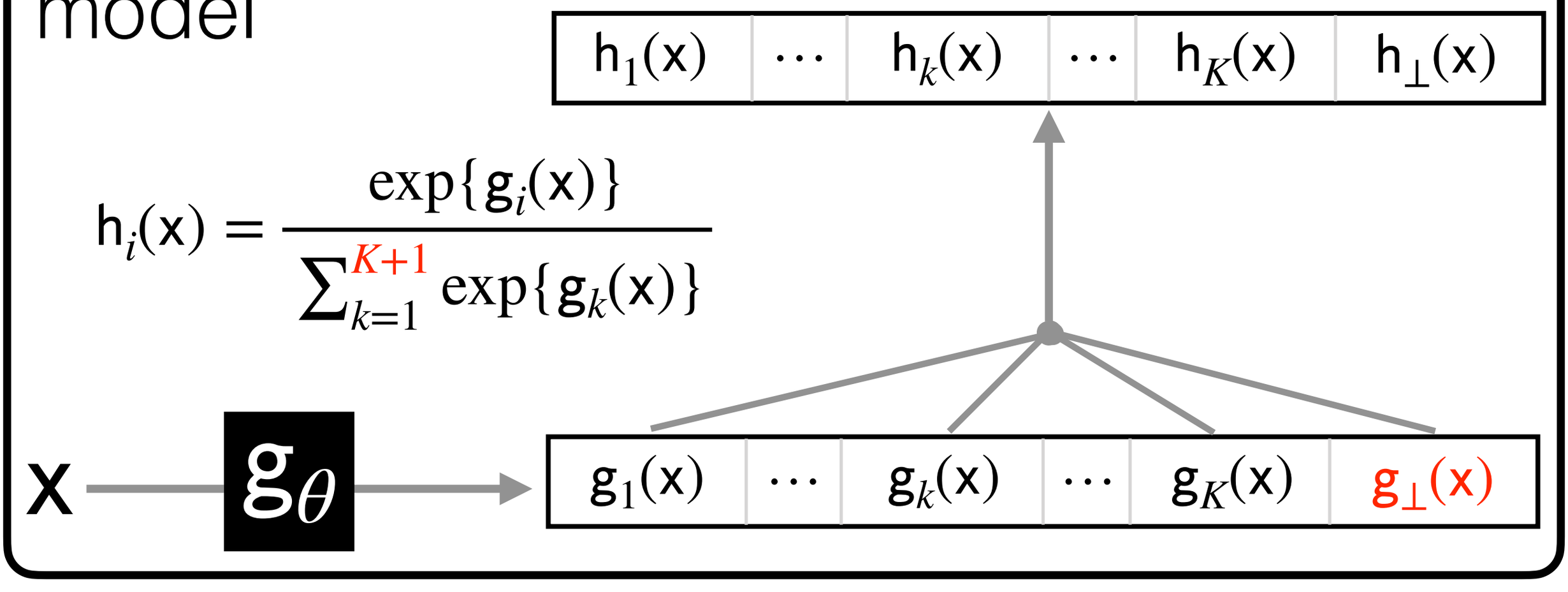
softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$



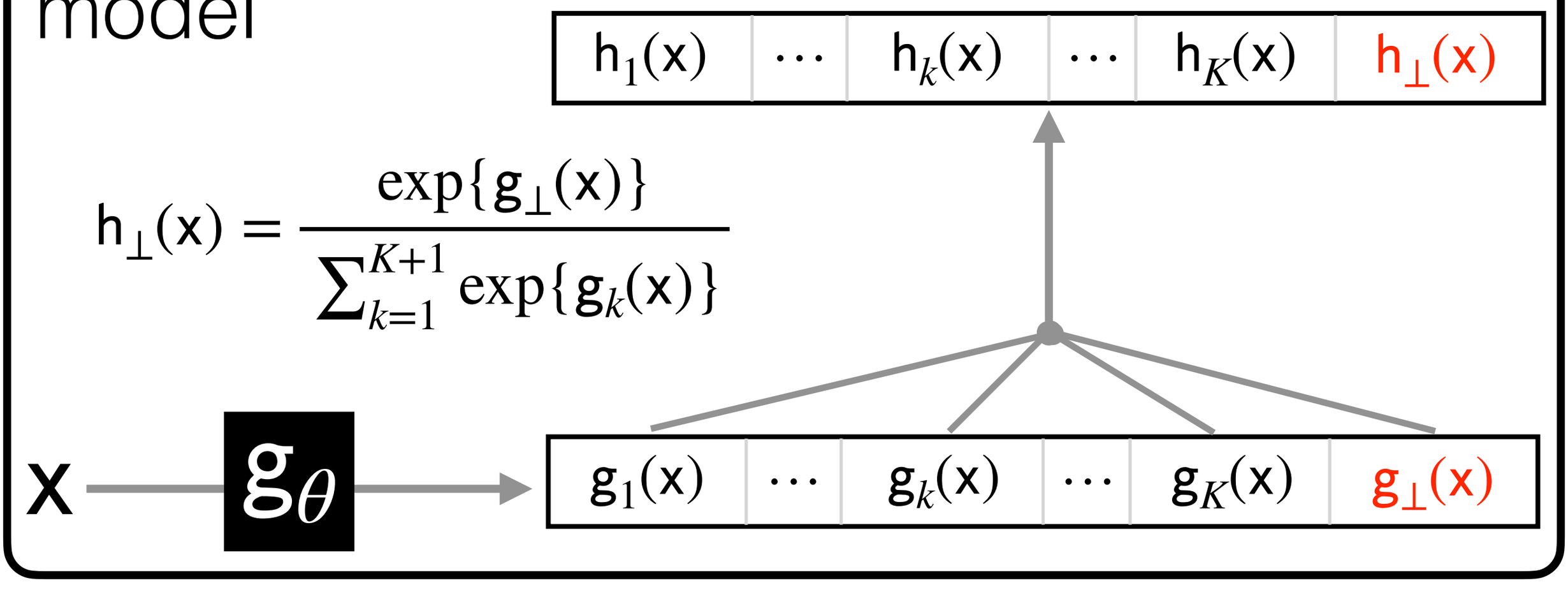
softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_\perp(\mathsf{x}) = \frac{\exp\{\mathsf{g}_\perp(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$



softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \mathsf{m}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; \mathbb{I}[\mathsf{y} = \mathsf{m}] \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \mathsf{m}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; \mathbb{I}[\mathsf{y} = \mathsf{m}] \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \mathsf{m}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; \mathbb{I}[\mathsf{y} = \mathsf{m}] \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

softmax implementation [Mozannar & Sontag, 2020]

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_n \right\}_{n=1}^{N}$$

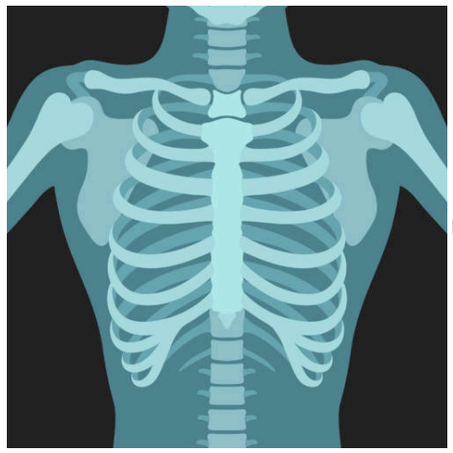
model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+1} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

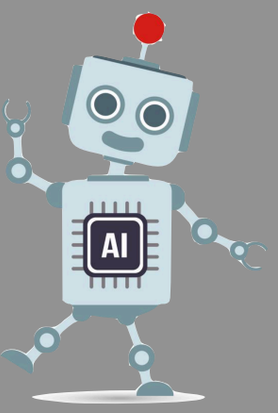
loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \mathsf{m}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) \; - \; \mathbb{I}[\mathsf{y} = \mathsf{m}] \cdot \log \mathsf{h}_{\perp}(\mathsf{x})$$

input features



allocation mechanism



classifier

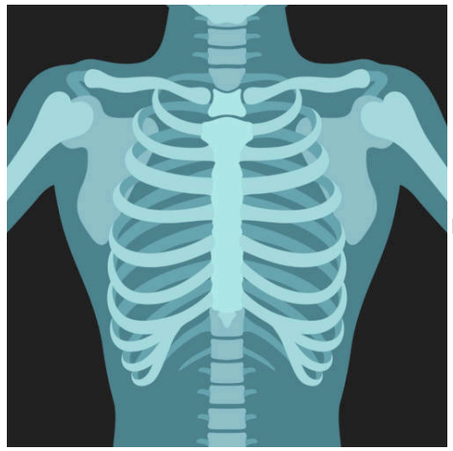
expert

defer to expert if…

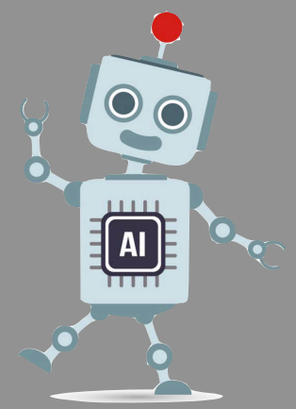
$$\max_{y \in [1,K]} \; \mathsf{h}_y(\mathsf{x}) \; < \; \mathsf{h}_\perp(\mathsf{x})$$

- Learning to reject
  - Bayes decision theory
  - consistent surrogate loss
- Learning to defer to an expert
  - Bayes decision theory
  - consistent surrogate loss
  - **extension to multiple experts**

input features



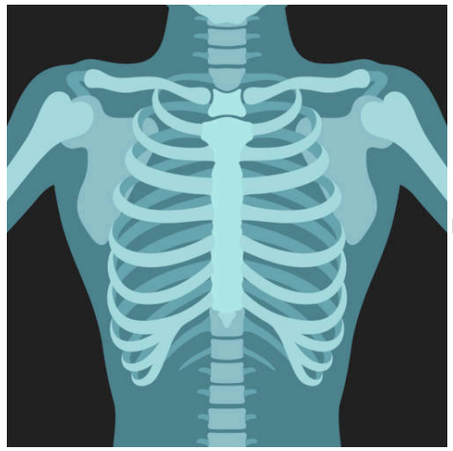
allocation mechanism



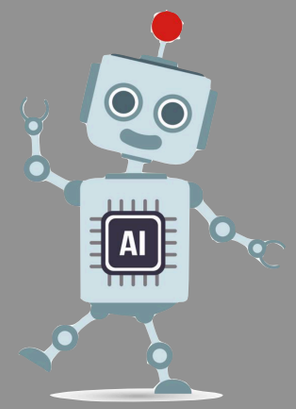
classifier

expert

input features



allocation mechanism



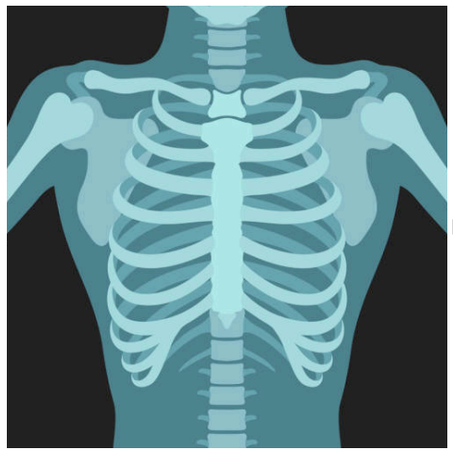
classifier

expert #1

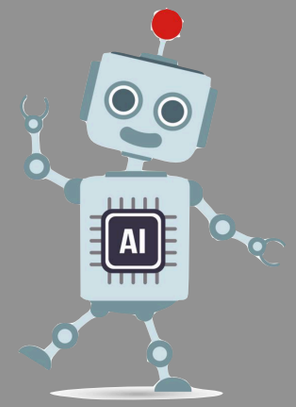
expert #3

expert #2

input features



allocation mechanism



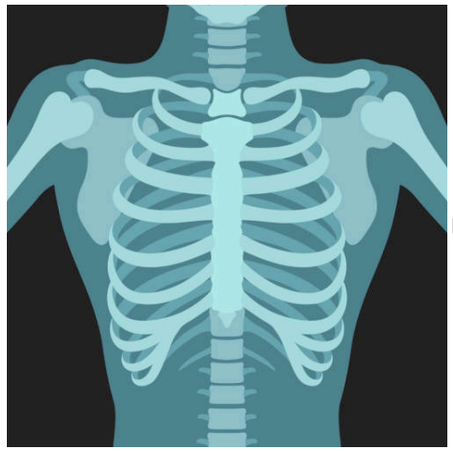
classifier

expert #1

expert #3

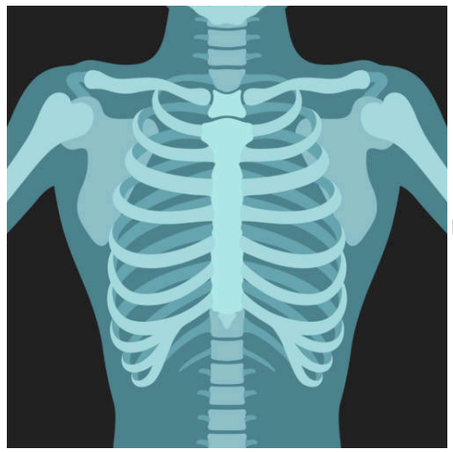
expert #2

input features



allocation mechanism

input features



allocation mechanism

?

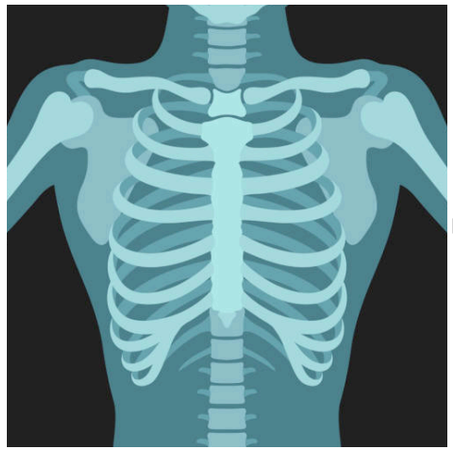
classifier

expert #3

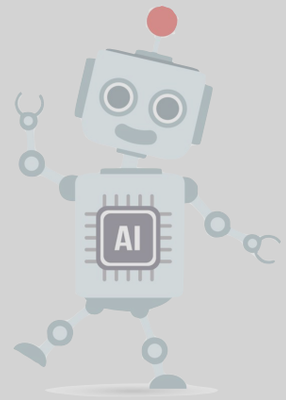
expert #1

expert #2

input features



allocation mechanism



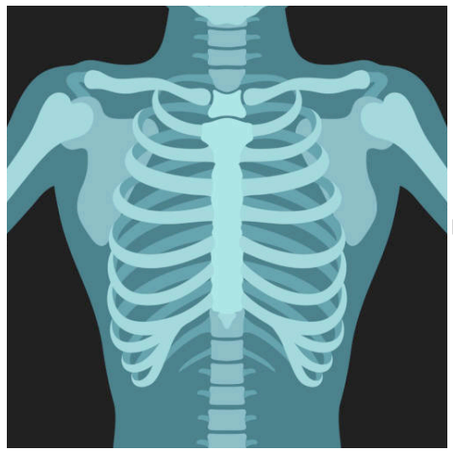
classifier

expert #1

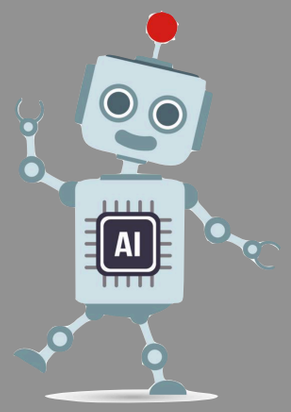
expert #3

expert #2

input features



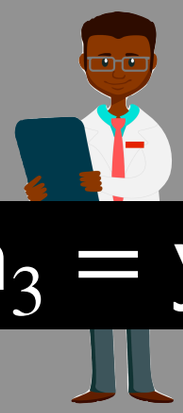
allocation mechanism



classifier

$\mathbb{P}(m_1 = y \mid x)$

expert #1



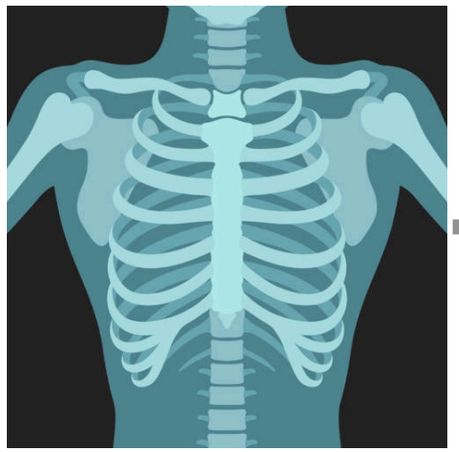
$\mathbb{P}(m_3 = y \mid x)$

expert #3

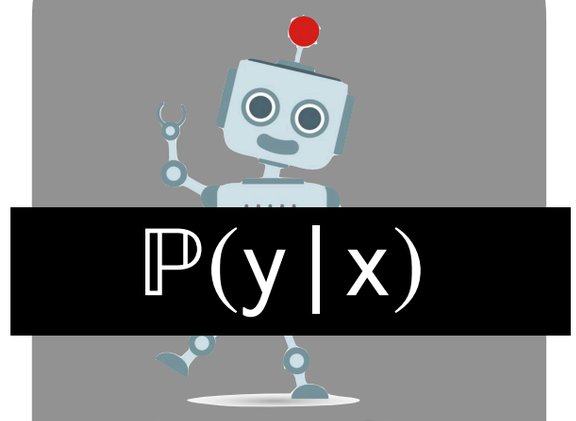
$\mathbb{P}(m_2 = y \mid x)$

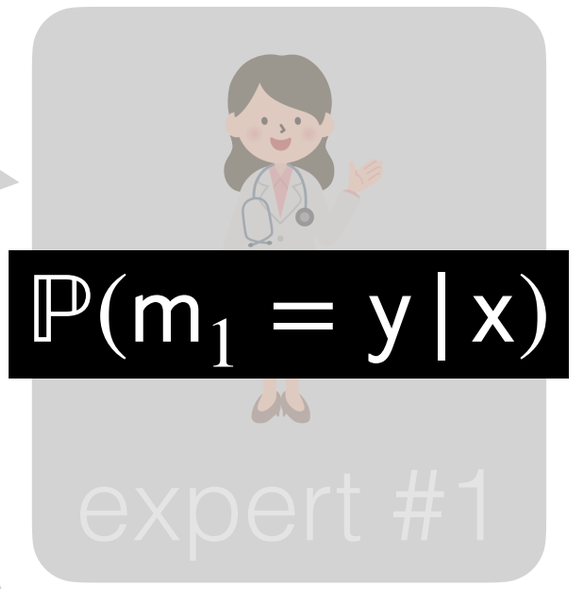
expert #2

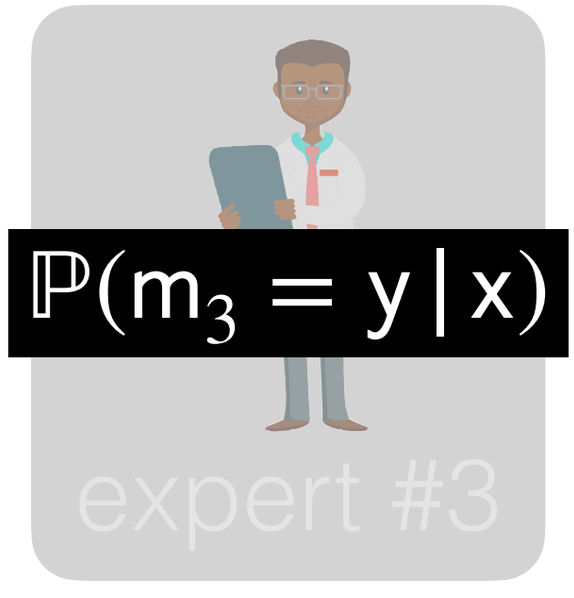
input features

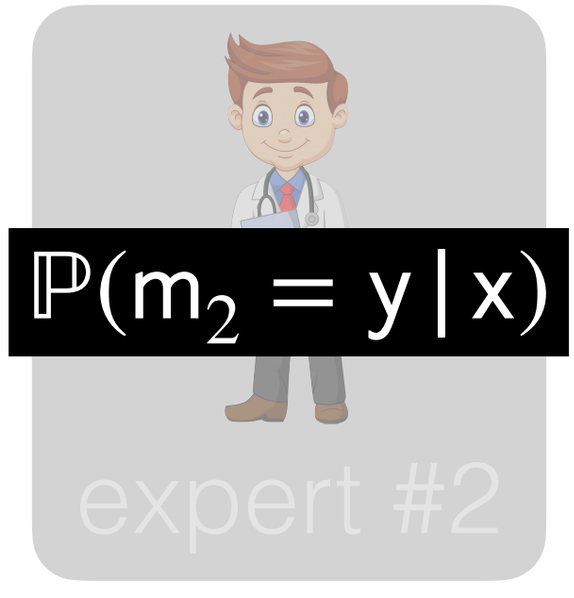


allocation mechanism





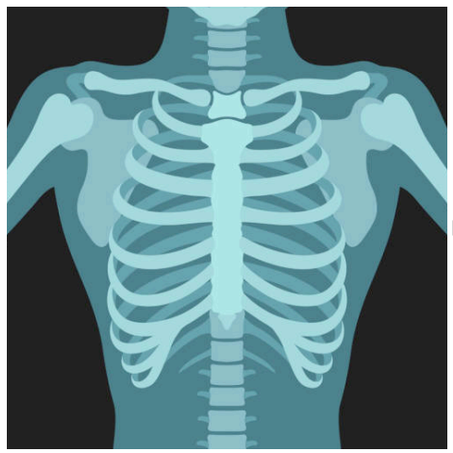




use classifier if…

$$\max_y \ \mathbb{P}(y \mid x) > \mathbb{P}(m_j = y \mid x), \ \forall j$$

input features



allocation mechanism

classifier

$\mathbb{P}(y \mid x)$

expert #1

$\mathbb{P}(m_1 = y \mid x)$

expert #3

$\mathbb{P}(m_3 = y \mid x)$

expert #2

$\mathbb{P}(m_2 = y \mid x)$

else, pick best expert:

$$\arg\max_j \ \mathbb{P}\left(m_j = y \mid x\right)$$

multi-expert implementation

training data

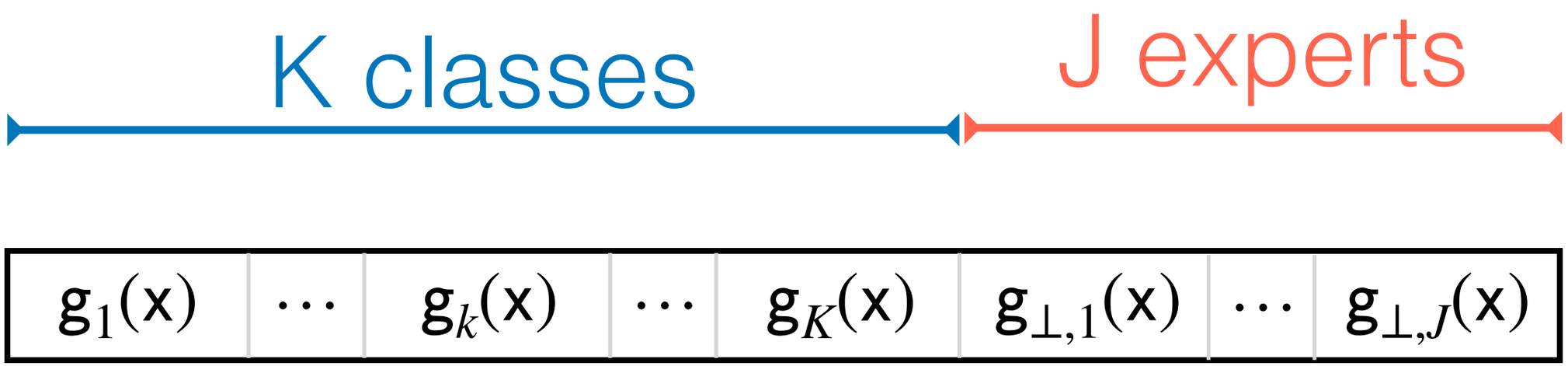
$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_{n,1}, \ldots, \mathsf{m}_{n,J} \right\}_{n=1}^{N}$$

multi-expert implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_{n,1}, \ldots, \mathsf{m}_{n,J} \right\}_{n=1}^{N}$$

model

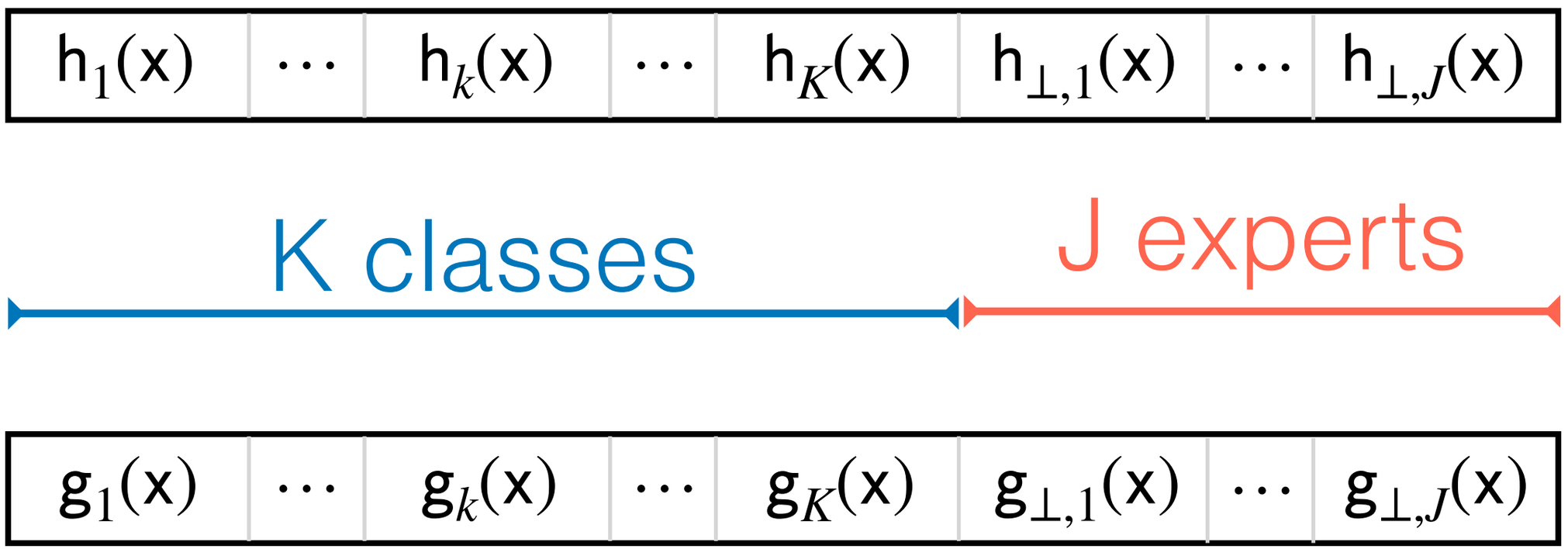


multi-expert implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_{n,1}, \ldots, \mathsf{m}_{n,J} \right\}_{n=1}^{N}$$

model

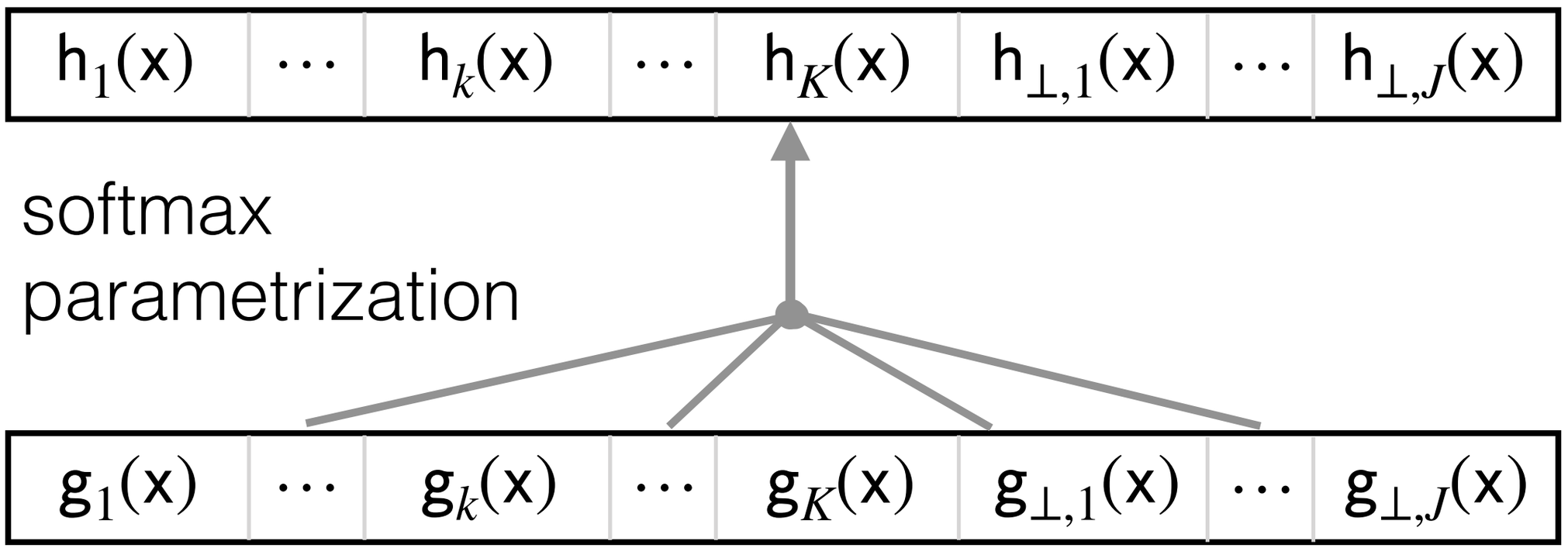


multi-expert implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_{n,1}, \ldots, \mathsf{m}_{n,J} \right\}_{n=1}^{N}$$

model



multi-expert implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_{n,1}, \ldots, \mathsf{m}_{n,J} \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+J} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

multi-expert implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_{n,1}, \ldots, \mathsf{m}_{n,J} \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+J} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \{\mathsf{m}_j\}_{j=1}^{J}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) - \sum_j \mathbb{I}\left[\mathsf{y} = \mathsf{m}_j\right] \cdot \log \mathsf{h}_{\perp,j}(\mathsf{x})$$

multi-expert implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_{n,1}, \ldots, \mathsf{m}_{n,J} \right\}_{n=1}^{N}$$

model

$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+J} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

loss function

$$\ell(\theta; \mathsf{x}, \mathsf{y}, \{\mathsf{m}_j\}_{j=1}^{J}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) - \sum_j \mathbb{I}\left[\mathsf{y} = \mathsf{m}_j\right] \cdot \log \mathsf{h}_{\perp,j}(\mathsf{x})$$

multi-expert implementation

training data

$$\mathfrak{D} = \left\{ \mathsf{x}_n, \mathsf{y}_n, \mathsf{m}_{n,1}, \ldots, \mathsf{m}_{n,J} \right\}_{n=1}^{N}$$

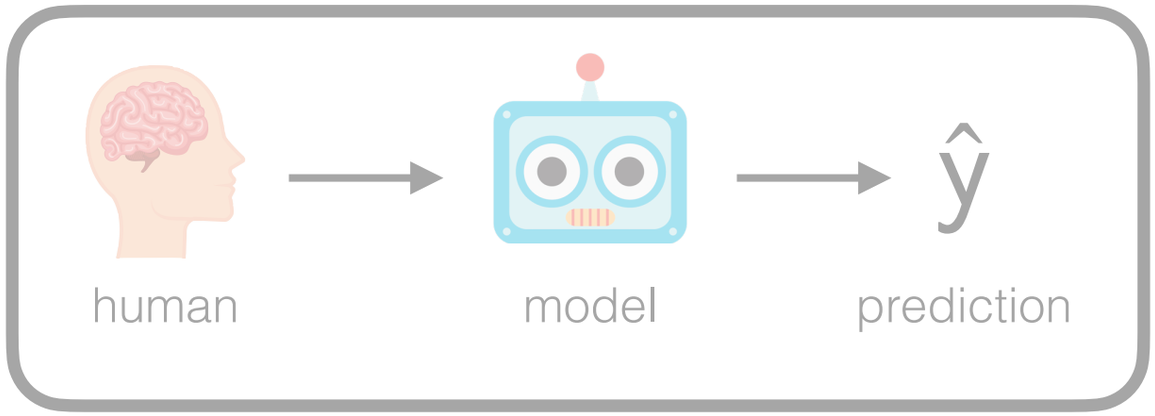
model

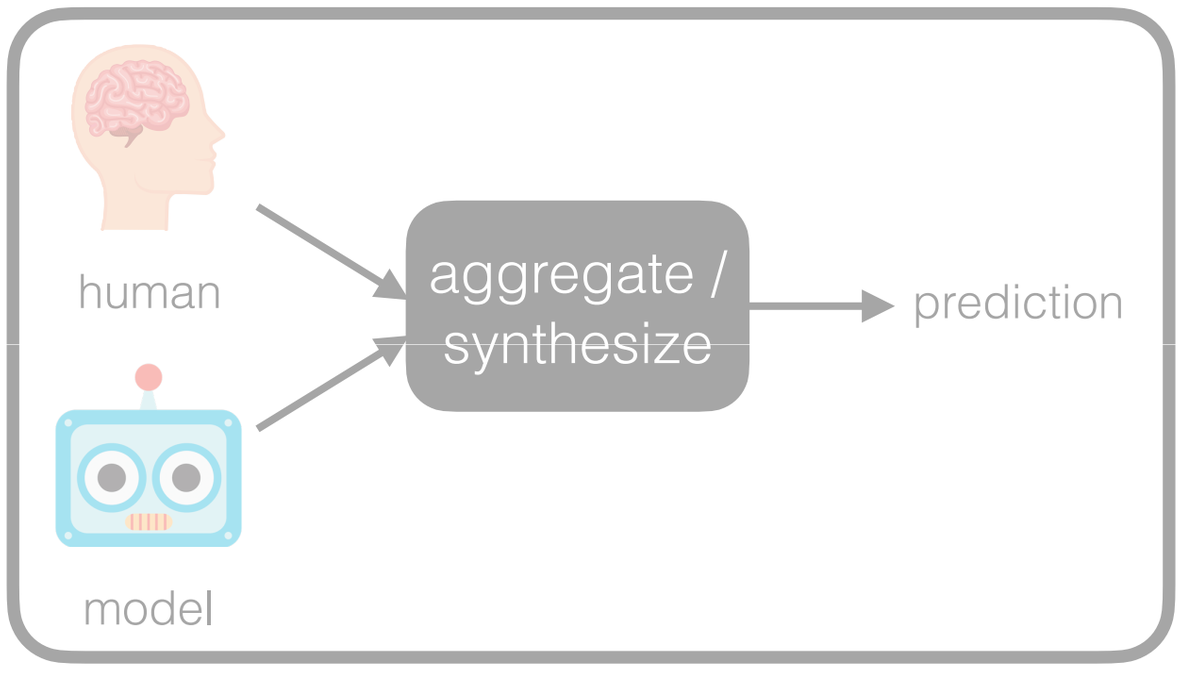
$$\mathsf{h}_i(\mathsf{x}) = \frac{\exp\{\mathsf{g}_i(\mathsf{x})\}}{\sum_{k=1}^{K+J} \exp\{\mathsf{g}_k(\mathsf{x})\}}$$

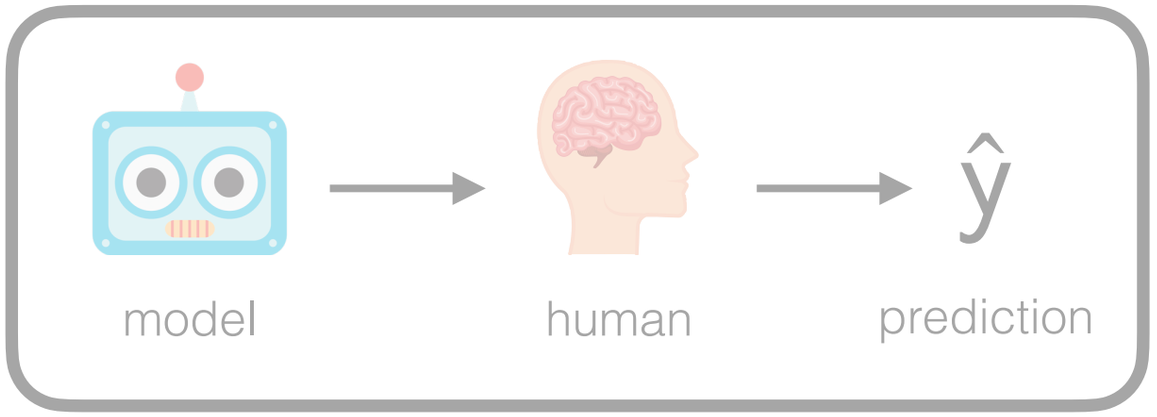
loss function

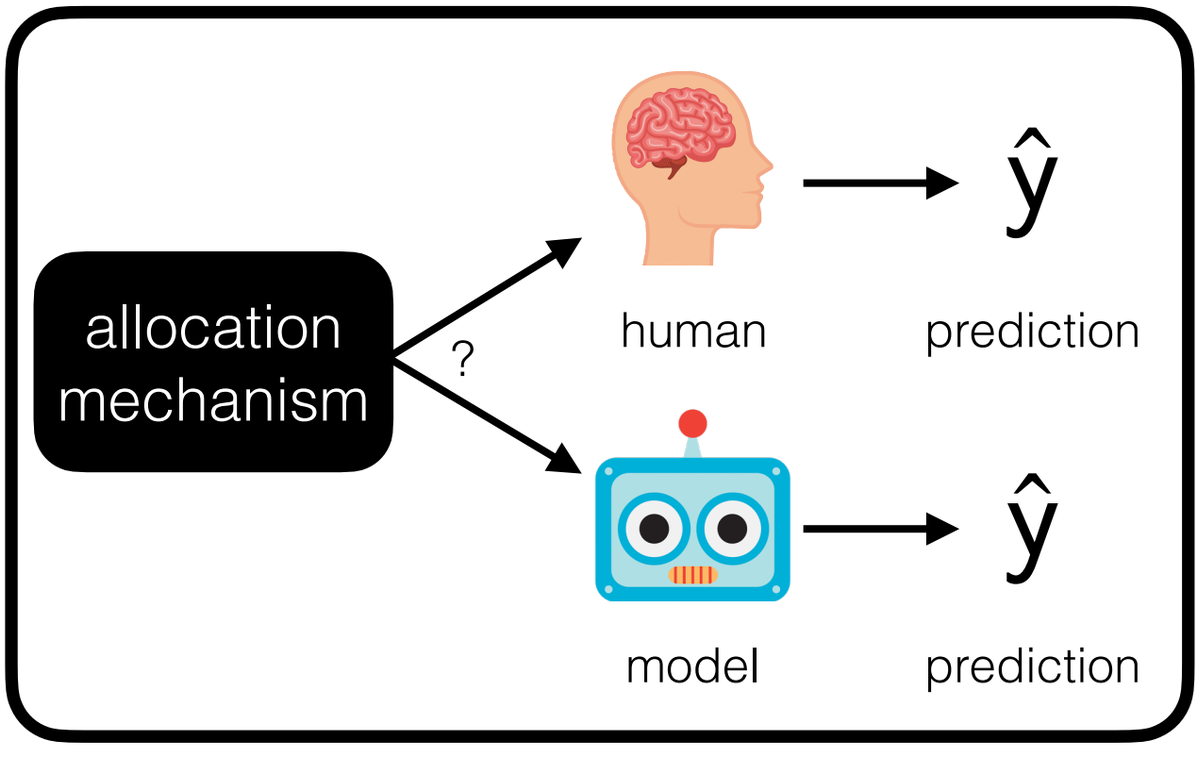
$$\ell(\theta; \mathsf{x}, \mathsf{y}, \{\mathsf{m}_j\}_{j=1}^{J}) = -\log \mathsf{h}_{\mathsf{y}}(\mathsf{x}) - \sum_j \mathbb{I}\left[\mathsf{y} = \mathsf{m}_j\right] \cdot \log \mathsf{h}_{\perp,j}(\mathsf{x})$$

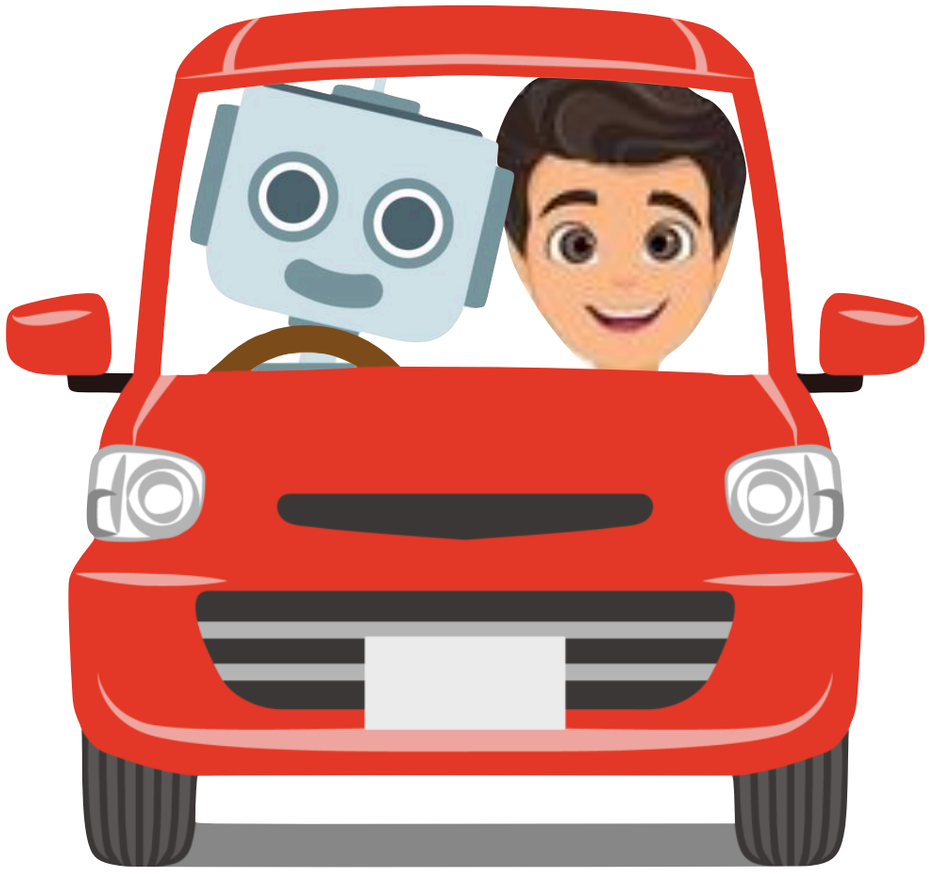
- Learning to reject
  - Bayes decision theory
  - consistent surrogate loss
- Learning to defer to an expert
  - Bayes decision theory
  - consistent surrogate loss
  - extension to multiple experts

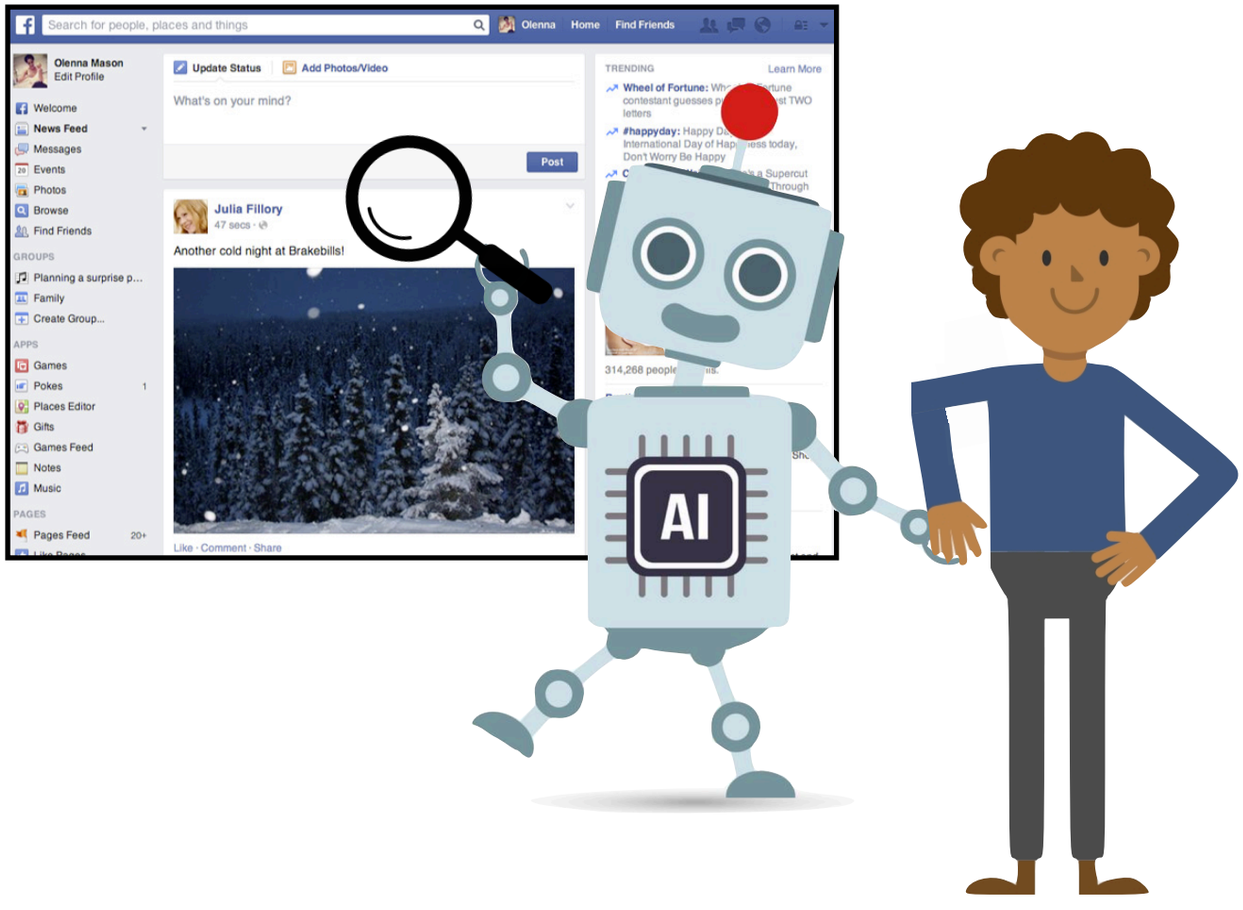

human
model
ŷ
prediction

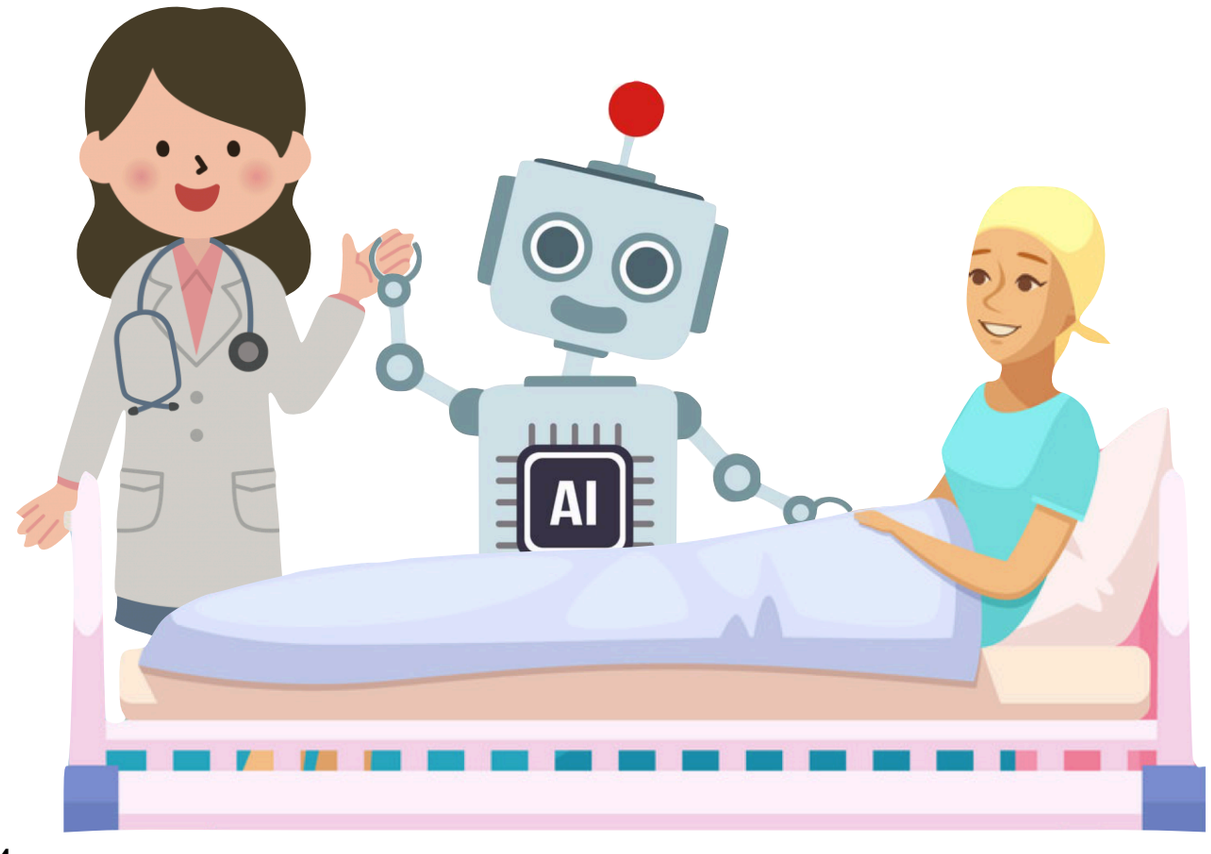

human
aggregate / synthesize
prediction
model


model
human
ŷ
prediction


allocation mechanism
?
human
ŷ
prediction
model
ŷ
prediction




Search for people, places and things
Olenna
Home
Find Friends
Olenna Mason
Edit Profile
Welcome
News Feed
Messages
Events
Photos
Browse
Find Friends
GROUPS
Planning a surprise p...
Family
Create Group...
APPS
Games
Pokes
Places Editor
Gifts
Games Feed
Notes
Music
PAGES
Pages Feed
Update Status
Add Photos/Video
What's on your mind?
Post
Julia Fillory
47 secs
Another cold night at Brakebills!
Like · Comment · Share
TRENDING
Learn More
Wheel of Fortune:
#happyday:
AI


AI