Uncertainty in Deep Learning

Yarin Gal

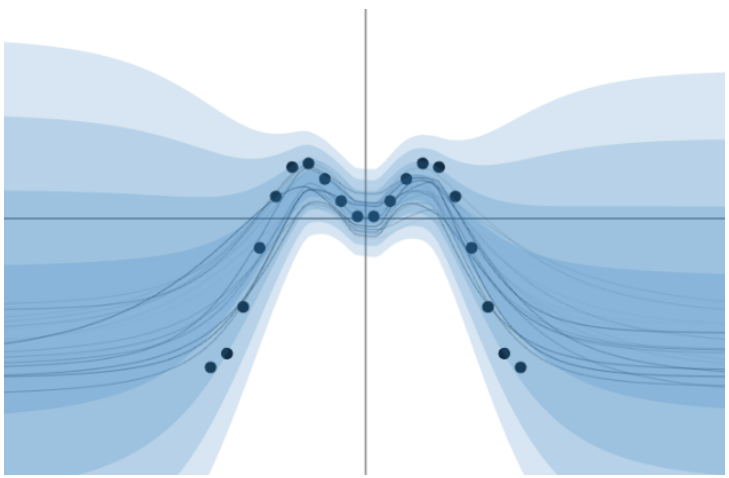
University of Oxford
yarin@cs.ox.ac.uk

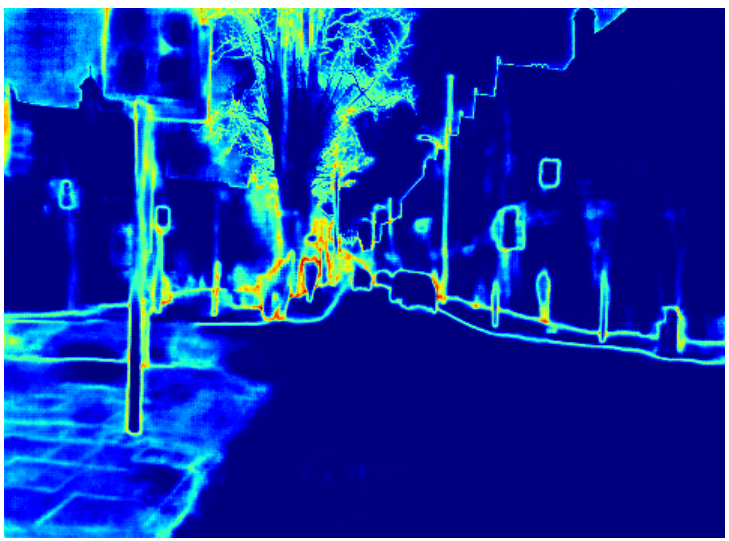
UNIVERSITY OF OXFORD

Introduction

Lecture Structure

- ML applied in the real world, and when we cannot trust it
- Sources of error and uncertainty
- What is uncertainty in deep learning





With great power...

- Many engineering advances in ML
- Systems applied to toy data
  $\rightarrow$ deployed in real-life settings
- Control slowly **handed-over** to automated systems; w many scenarios which can affect our daily lives
  - Medical: automated decision making or recommendation systems
  - Automotive: autonomous control of drones and self driving cars
  - High frequency trading: ability to affect economic markets on global scale
  - But all of these can be quite dangerous...

With great power...

- Many engineering advances in ML
- Systems applied to toy data
  $\rightarrow$ deployed in real-life settings
- Control slowly **handed-over** to automated systems; w many scenarios which can affect our daily lives
  - Medical: automated decision making or recommendation systems
  - Automotive: autonomous control of drones and self driving cars
  - High frequency trading: ability to affect economic markets on global scale
  - But all of these can be quite dangerous...

With great power...

- Many engineering advances in ML
- Systems applied to toy data
  $\rightarrow$ deployed in real-life settings
- Control slowly **handed-over** to automated systems; w many scenarios which can affect our daily lives
  - Medical: automated decision making or recommendation systems
  - Automotive: autonomous control of drones and self driving cars
  - High frequency trading: ability to affect economic markets on global scale
  - But all of these can be quite dangerous...

With great power...

- Many engineering advances in ML
- Systems applied to toy data
  $\rightarrow$ deployed in real-life settings
- Control slowly **handed-over** to automated systems; w many scenarios which can affect our daily lives
  - Medical: automated decision making or recommendation systems
  - Automotive: autonomous control of drones and self driving cars
  - High frequency trading: ability to affect economic markets on global scale
  - But all of these can be quite dangerous...

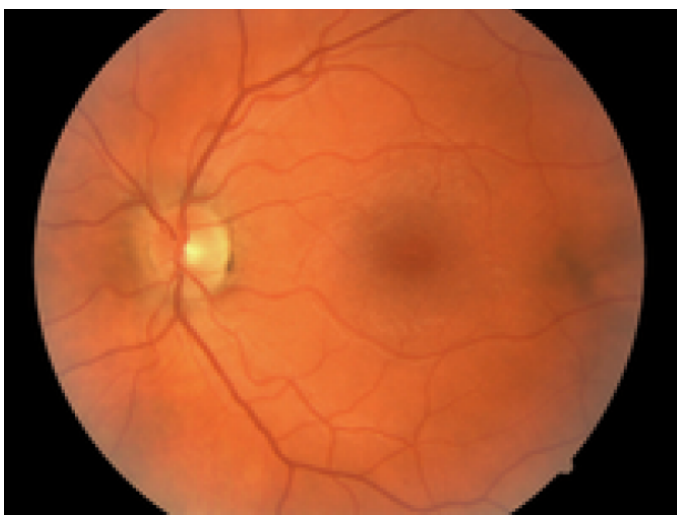
With great power...

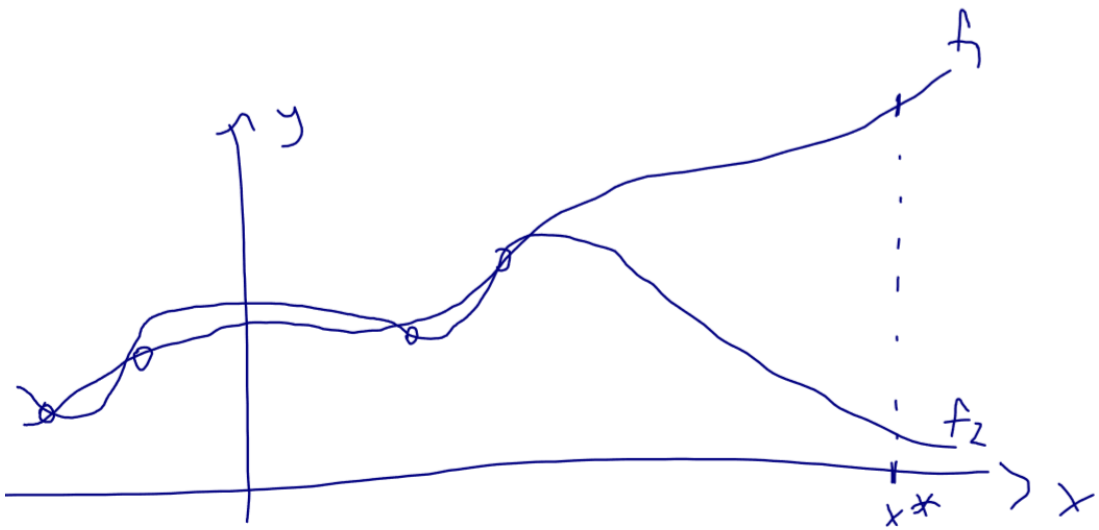
- Many engineering advances in ML
- Systems applied to toy data
  $\rightarrow$ deployed in real-life settings
- Control slowly **handed-over** to automated systems; w many scenarios which can affect our daily lives
  - Medical: automated decision making or recommendation systems
  - Automotive: autonomous control of drones and self driving cars
  - High frequency trading: ability to affect economic markets on global scale
  - But all of these can be quite dangerous...

Example: Medical Diagnostics

- We're given *dAIbetes*: an exciting new startup (not really)
  - claims to automatically diagnose diabetic retinopathy
  - accuracy 99% on their 400 train/test patients
  - engineer trained **two** deep learning systems to predict probability y given input fondus image x.

The engineer runs their systems on your fondus image x^* (RHS):



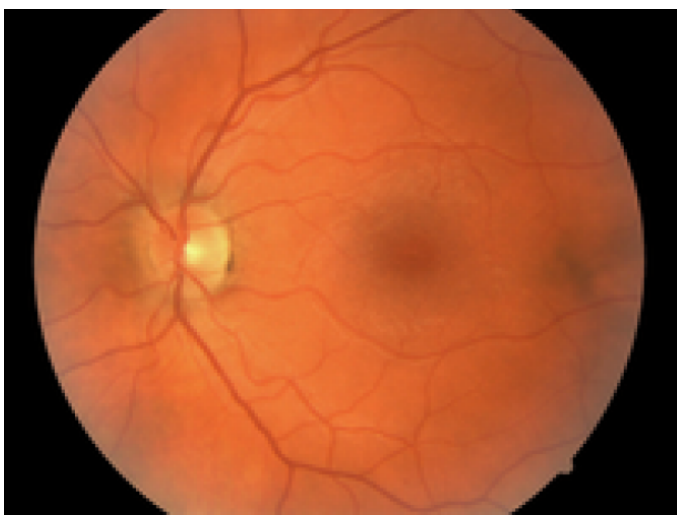


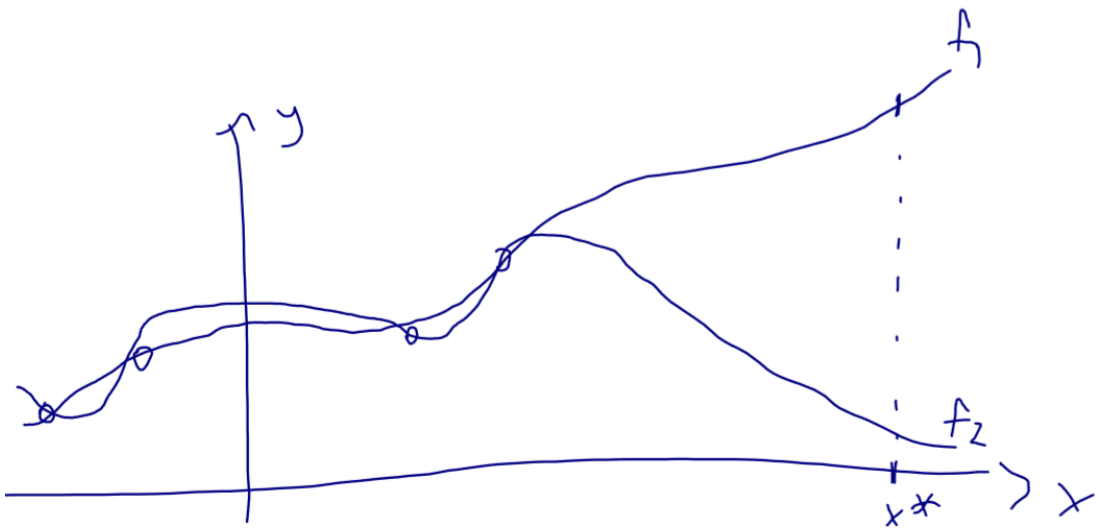
- Which model f_1, f_2 would you want the engineer to use for your diagnosis?

Example: Medical Diagnostics

- We're given *dAIbetes*: an exciting new startup (not really)
  - claims to automatically diagnose diabetic retinopathy
  - accuracy 99% on their 400 train/test patients
  - engineer trained **two** deep learning systems to predict probability y given input fondus image x.

The engineer runs their systems on your fondus image x^* (RHS):



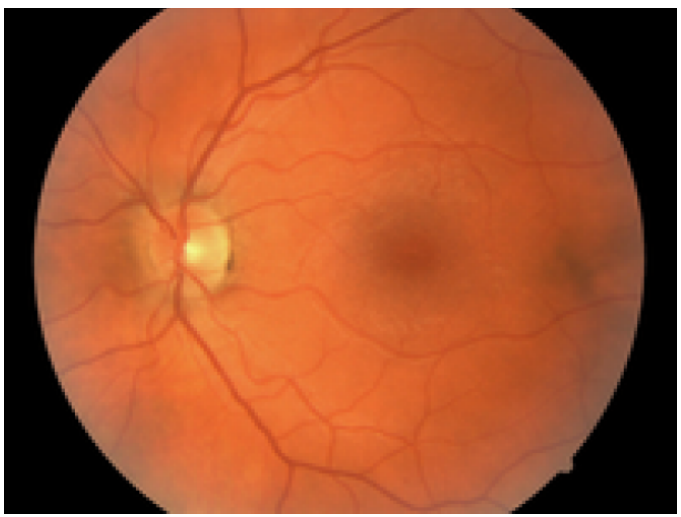


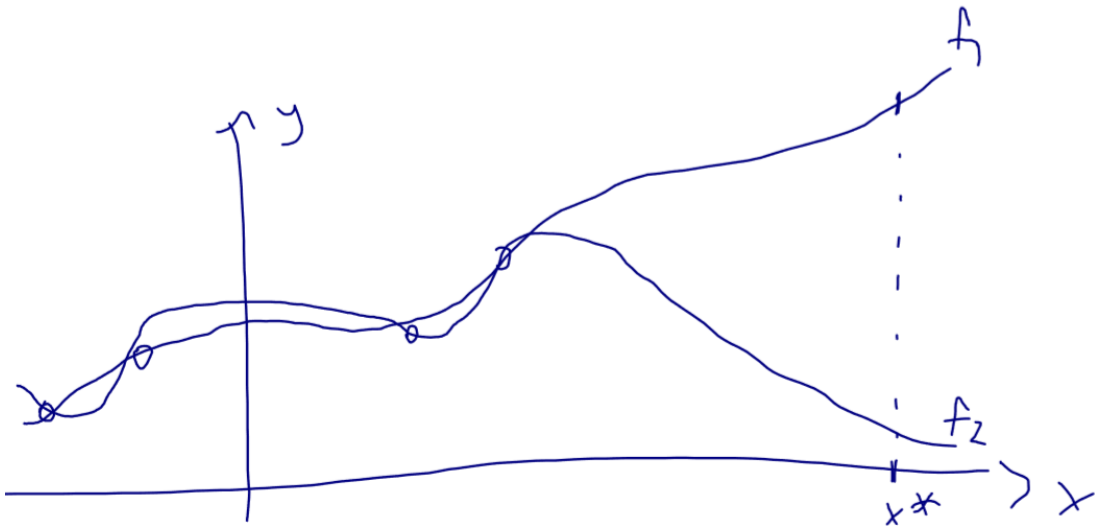
- Which model f_1, f_2 would you want the engineer to use for your diagnosis?

Example: Medical Diagnostics

- We're given *dAIbetes*: an exciting new startup (not really)
  - claims to automatically diagnose diabetic retinopathy
  - accuracy 99% on their 400 train/test patients
  - engineer trained **two** deep learning systems to predict probability y given input fondus image x.

The engineer runs their systems on your fondus image x^* (RHS):

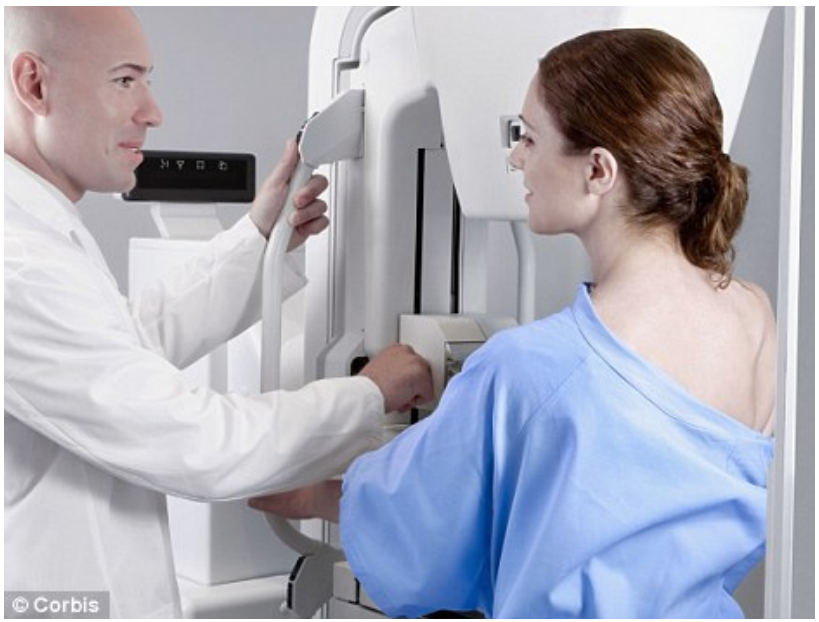




- Which model f_1, f_2 would you want the engineer to use for your diagnosis? **None of these!** ('I don't know')

Example: Medical Diagnostics (cnt)

In medical / robotics / science...



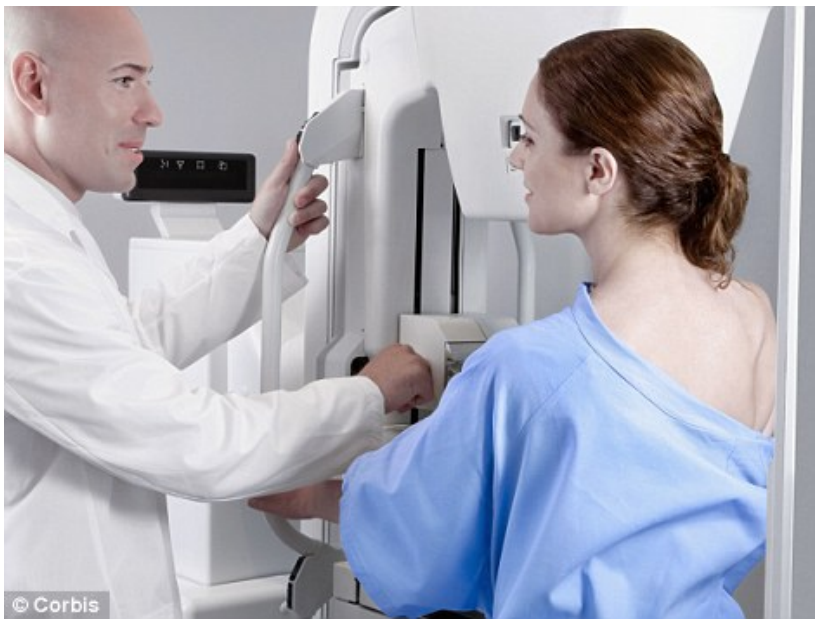
X **can't** use ML models giving a single point estimate (single value) in prediction

V **must** use ML models giving an answer that says '10 but I'm uncertain'; or '10 ± 5'

- Give me a **distribution** over possible outcomes!

Example: Medical Diagnostics (cnt)

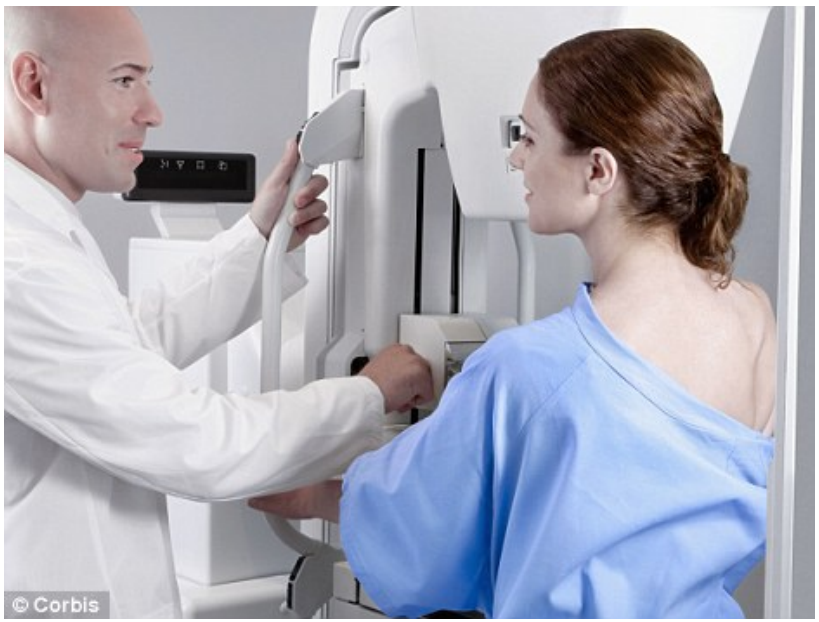
In medical / robotics / science...



- X **can't** use ML models giving a single point estimate (single value) in prediction
- V **must** use ML models giving an answer that says '10 but I'm uncertain'; or '10 ± 5'
- Give me a **distribution** over possible outcomes!

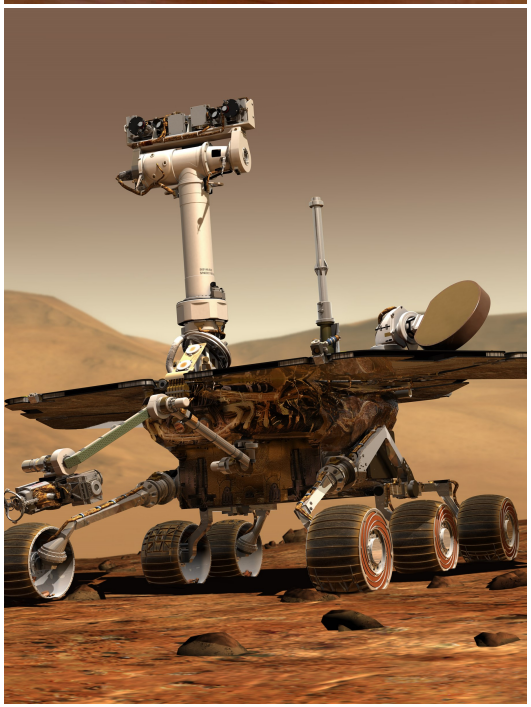
Example: Medical Diagnostics (cnt)

In medical / robotics / science...



- X **can't** use ML models giving a single point estimate (single value) in prediction
- V **must** use ML models giving an answer that says '10 but I'm uncertain'; or '10 ± 5'
- ▶ Give me a **distribution** over possible outcomes!

Example: Autonomous Driving



Autonomous systems

- Range from simple robotic **vacuums** to **self-driving cars**
- Largely divided into systems which
  - control behaviour w **rule-based** systems
  - learn and **adapt to environment**

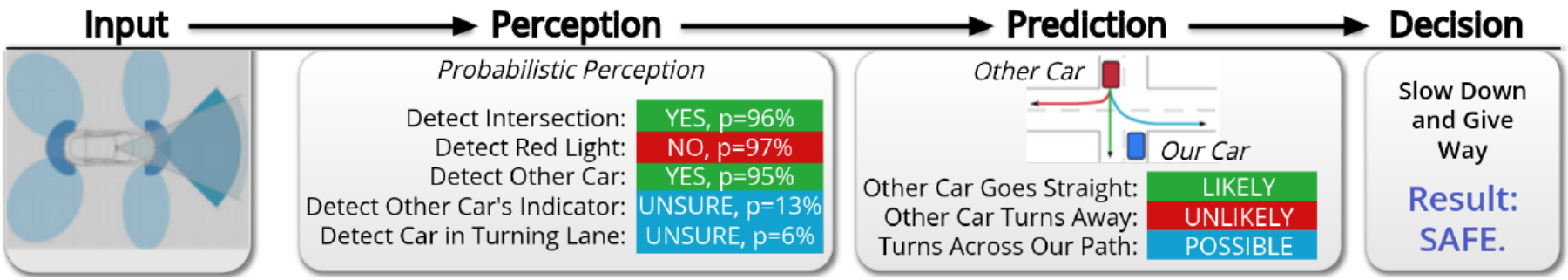
Both can use of ML tools in their pipelines

- ML for low-level feature extraction (**perception**)
- reinforcement learning (control)

Example: Autonomous Driving (cnt)

Example ML pipeline in **self-driving cars**

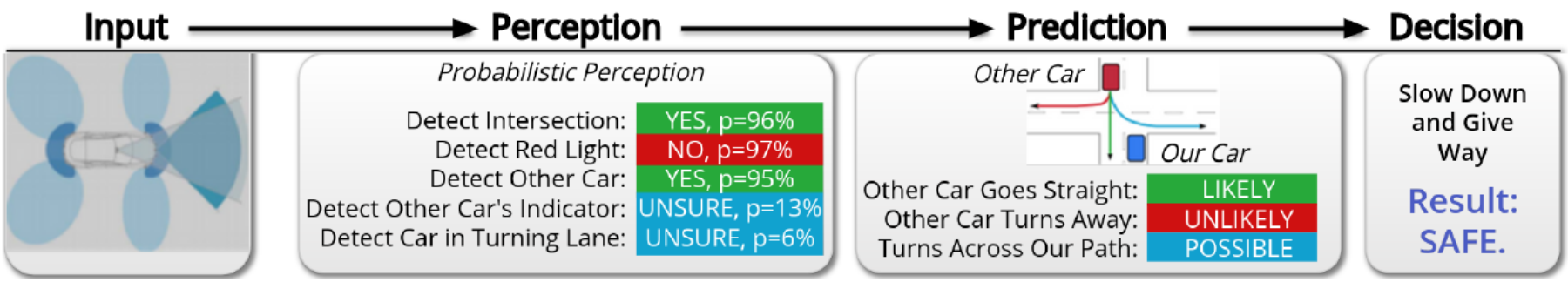
- **process raw sensory input** w perception models
  - eg image segmentation to build the car's *internal repr of the world*: find where other cars and pedestrians **are** (examples below; we'll come back to this at end of course)
- output fed into prediction model
  - eg where other car will **go**
- output fed into 'higher-level' decision making procedures
  - eg **rule based system** ("cyclist to your left → do not steer left")



Example: Autonomous Driving (cnt)

Example ML pipeline in **self-driving cars**

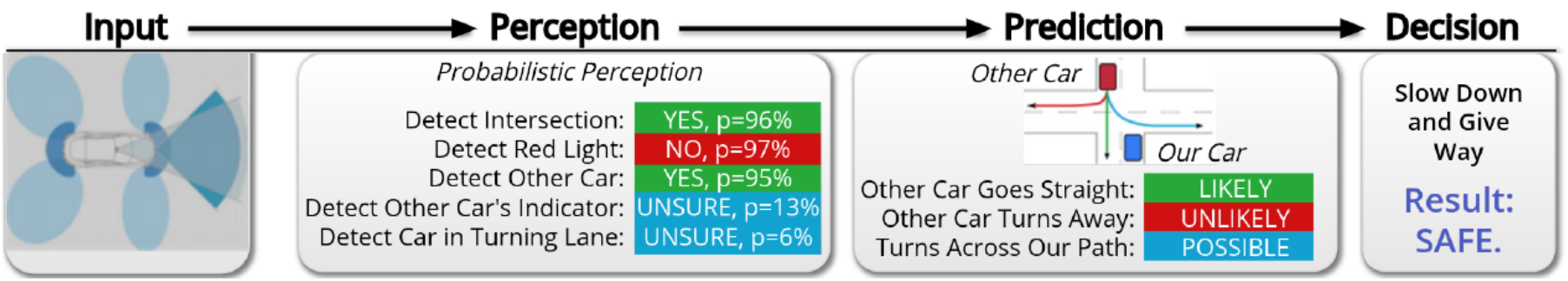
- **process raw sensory input** w perception models
  - eg image segmentation to build the car's *internal repr of the world*: find where other cars and pedestrians **are** (examples below; we'll come back to this at end of course)
- output fed into prediction model
  - eg where other car will **go**
- output fed into 'higher-level' decision making procedures
  - eg **rule based system** ("cyclist to your left → do not steer left")



Example: Autonomous Driving (cnt)

Example ML pipeline in **self-driving cars**

- **process raw sensory input** w perception models
  - eg image segmentation to build the car's *internal repr of the world*: find where other cars and pedestrians **are** (examples below; we'll come back to this at end of course)
- output fed into prediction model
  - eg where other car will **go**
- output fed into 'higher-level' decision making procedures
  - eg **rule based system** ("cyclist to your left $\rightarrow$ do not steer left")



Example: Autonomous Driving (cnt)

UNIVERSITY OF OXFORD

Real-world failure: assisted driving

- first fatality of assisted driving (June 2016)
- low-level system failed to distinguish white side of trailer from bright sky

TESLA MODEL S MODEL 3 MODEL X MODEL Y ROADSTER

Blog Videos

A Tragic Loss

The Tesla Team • 30 June 2016

We learned yesterday evening that NHTSA is opening a preliminary evaluation into the performance of Autopilot during a recent fatal crash that occurred in a Model S. This is the first known fatality in just over 130 million miles where Autopilot was activated. Among all vehicles in the US, there is a fatality every 94 million miles. Worldwide, there is a fatality approximately every 60 million miles. It is important to emphasize that the NHTSA action is simply a preliminary evaluation to determine whether the system worked according to expectations.

Following our standard practice, Tesla informed NHTSA about the incident immediately after it occurred. What we know is that the vehicle was on a divided highway with Autopilot engaged when a tractor trailer drove across the highway perpendicular to the Model S. Neither Autopilot nor the driver noticed the white side of the tractor trailer against a brightly lit sky, so the brake was not applied. The high ride height of the trailer combined with its positioning across the road and the extremely rare circumstances of the impact caused the Model S to pass under the trailer, with the bottom of the trailer

WIRED

TESLA'S AUTOPILOT WAS INVOLVED IN ANOTHER DEADLY CAR CRASH

TESLA

WIRED SUBSCRIBE

TESLA'S LATEST AUTOPILOT DEATH LOOKS JUST LIKE A PRIOR CRASH

Example: Autonomous Driving (cnt)

UNIVERSITY OF OXFORD

Real-world failure: assisted driving

- first fatality of assisted driving (June 2016)
- low-level system failed to distinguish white side of trailer from bright sky

TESLA MODEL S MODEL 3 MODEL X MODEL Y ROADSTER

Blog Videos

A Tragic Loss

The Tesla Team • 30 June 2016

We learned yesterday evening that NHTSA is opening a preliminary evaluation into the performance of Autopilot during a recent fatal crash that occurred in a Model S. This is the first known fatality in just over 130 million miles where Autopilot was activated. Among all vehicles in the US, there is a fatality every 94 million miles. Worldwide, there is a fatality approximately every 60 million miles. It is important to emphasize that the NHTSA action is simply a preliminary evaluation to determine whether the system worked according to expectations.

Following our standard practice, Tesla informed NHTSA about the incident immediately after it occurred. What we know is that the vehicle was on a divided highway with Autopilot engaged when a tractor trailer drove across the highway perpendicular to the Model S. Neither Autopilot nor the driver noticed the white side of the tractor trailer against a brightly lit sky, so the brake was not applied. The high ride height of the trailer combined with its positioning across the road and the extremely rare circumstances of the impact caused the Model S to pass under the trailer, with the bottom of the trailer

WIRED

TESLA'S AUTOPILOT WAS INVOLVED IN ANOTHER DEADLY CAR CRASH

TESLA

WIRED SUBSCRIBE

TESLA'S LATEST AUTOPILOT DEATH LOOKS JUST LIKE A PRIOR CRASH

If system had identified its own uncertainty:

- alert user to take control over steering
- propagate uncertainty to decision making

Example: Autonomous Driving (cnt)

UNIVERSITY OF OXFORD

Real-world failure: assisted driving

- first fatality of assisted driving (June 2016)
- low-level system failed to distinguish white side of trailer from bright sky

TESLA MODEL S MODEL 3 MODEL X MODEL Y ROADSTER

Blog Videos

A Tragic Loss

The Tesla Team • 30 June 2016

We learned yesterday evening that NHTSA is opening a preliminary evaluation into the performance of Autopilot during a recent fatal crash that occurred in a Model S. This is the first known fatality in just over 130 million miles where Autopilot was activated. Among all vehicles in the US, there is a fatality every 94 million miles. Worldwide, there is a fatality approximately every 60 million miles. It is important to emphasize that the NHTSA action is simply a preliminary evaluation to determine whether the system worked according to expectations.

Following our standard practice, Tesla informed NHTSA about the incident immediately after it occurred. What we know is that the vehicle was on a divided highway with Autopilot engaged when a tractor trailer drove across the highway perpendicular to the Model S. Neither Autopilot nor the driver noticed the white side of the tractor trailer against a brightly lit sky, so the brake was not applied. The high ride height of the trailer combined with its positioning across the road and the extremely rare circumstances of the impact caused the Model S to pass under the trailer, with the bottom of the trailer

WIRED

TESLA'S AUTOPILOT WAS INVOLVED IN ANOTHER DEADLY CAR CRASH

TESLA

WIRED SUBSCRIBE

TESLA'S LATEST AUTOPILOT DEATH LOOKS JUST LIKE A PRIOR CRASH

But industry is starting to use uncertainty in the pipeline

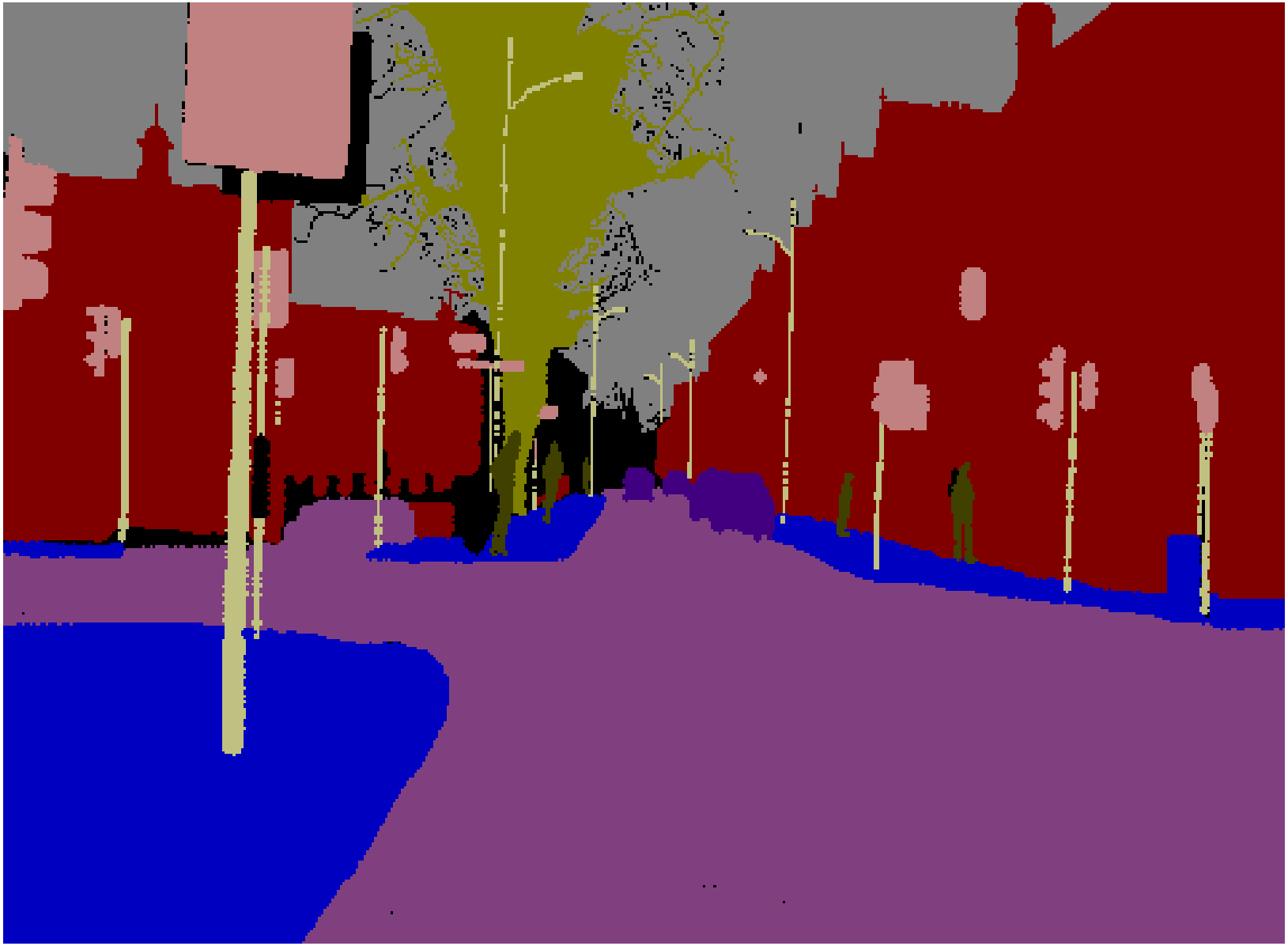
- eg predict distribution of pedestrian locations in X timesteps
- or uncertainty in perception components

Teaser: Uncertainty in Autonomous Driving

UNIVERSITY OF OXFORD

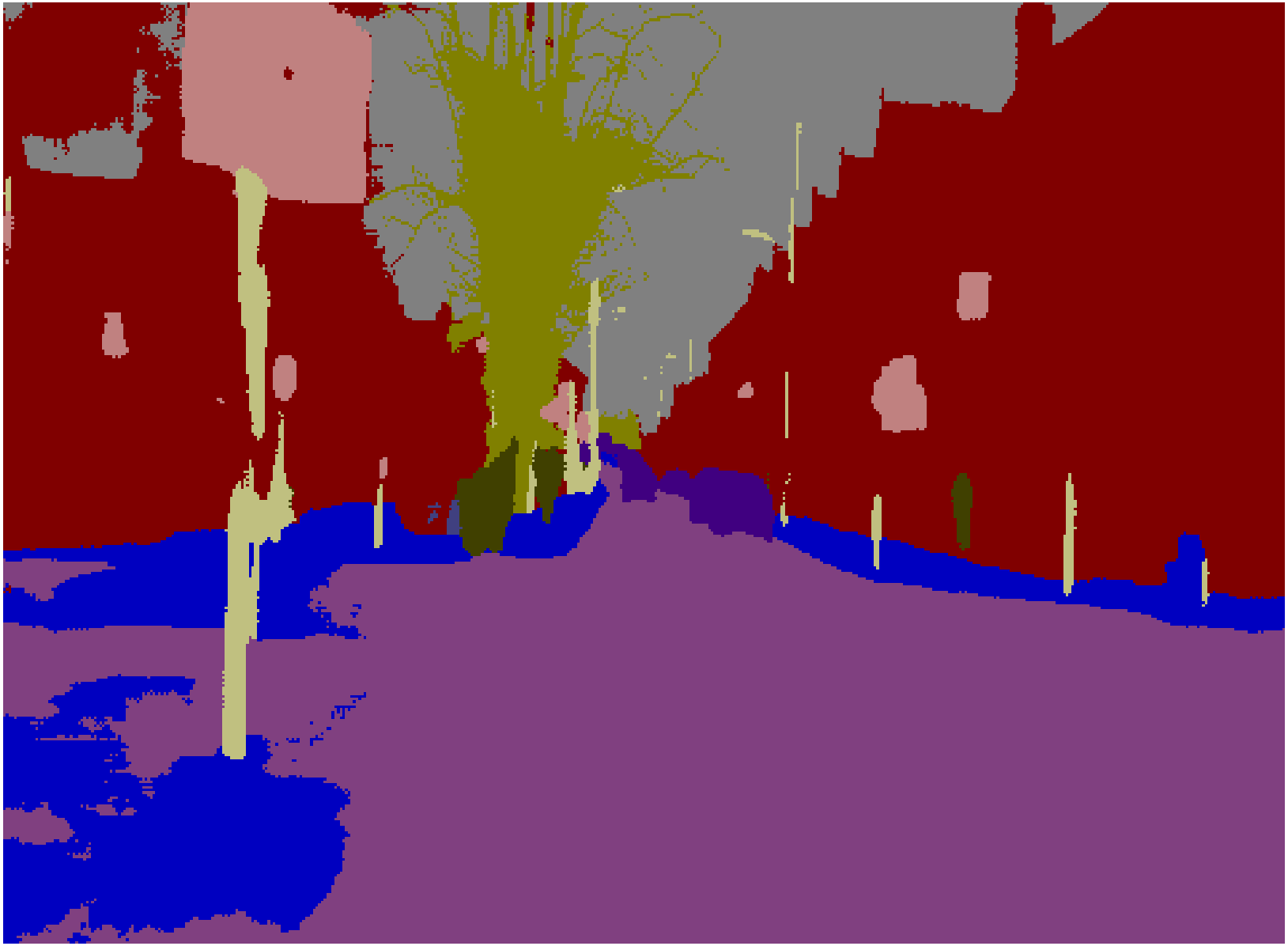
Teaser: Uncertainty in Autonomous Driving

UNIVERSITY OF OXFORD



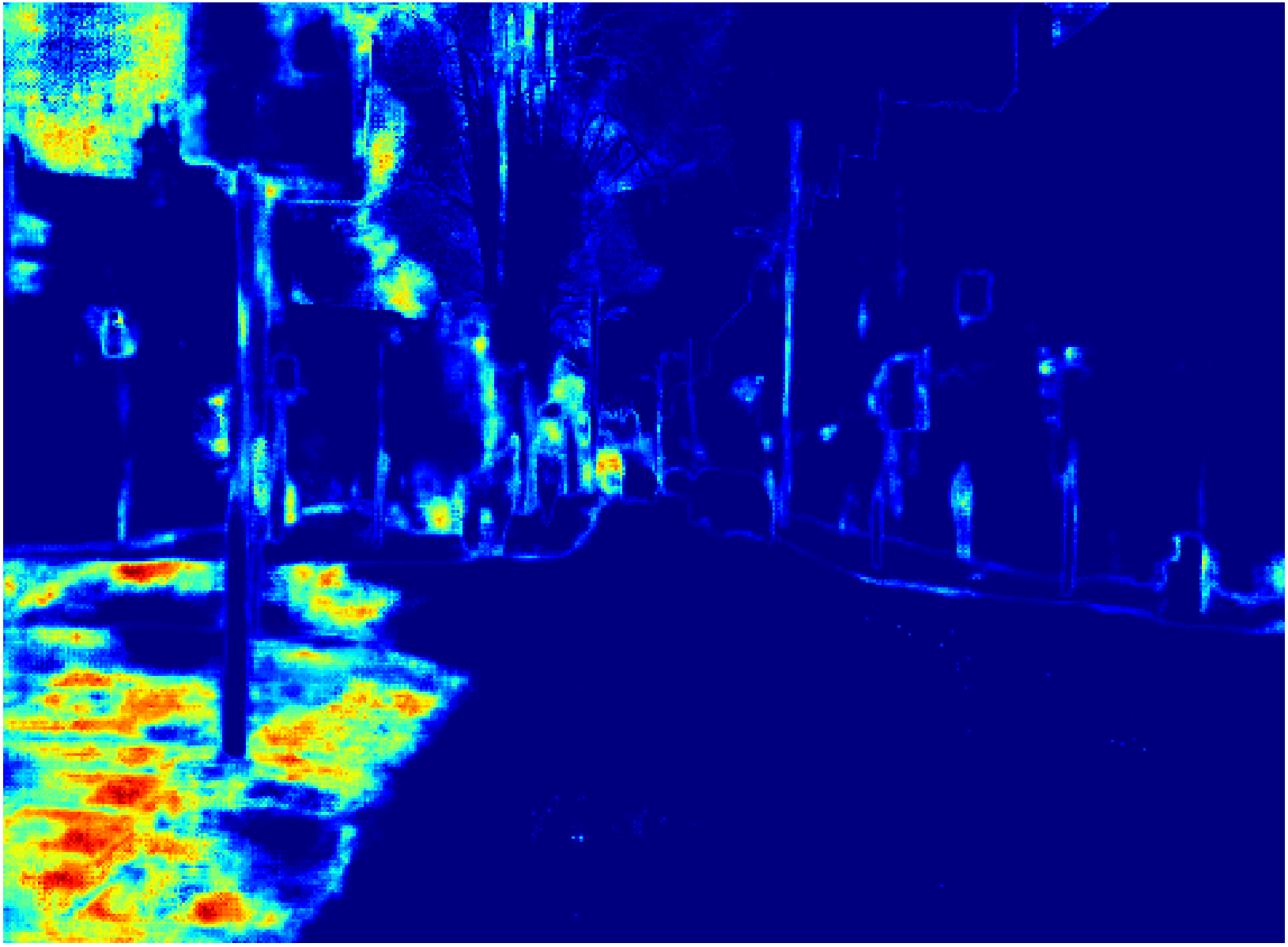
Teaser: Uncertainty in Autonomous Driving

UNIVERSITY OF OXFORD



Teaser: Uncertainty in Autonomous Driving

UNIVERSITY OF OXFORD



Example: Autonomous Driving (in industry)

UNIVERSITY OF OXFORD

Efficient Computation of Collision Probabilities for Safe Motion Planning*

Andrew Blake, Alejandro Bordallo, Majd Hawasly,
Svetlin Penkov, Subramanian Ramamoorthy†, Alexandre Silva

***Abstract*— We address the problem of safe motion planning. As mobile robots and autonomous vehicles become increasingly more prevalent in human-centered environments, the need to ensure safety in the sense of guaranteed collision free behaviour has taken renewed urgency. Achieving this when perceptual modules provide only noisy estimates of objects in the environment requires new approaches. Working within a probabilistic framework for describing the environment, we present methods for efficiently calculating a probabilistic risk of collision for a candidate path. This may be used to stratify a set of candidate trajectories by levels of a safety threshold. Given such a stratification, based on user-defined thresholds, motion synthesis techniques could optimise for secondary criteria with the assurance that a primary safety criterion is already being satisfied. A key contribution of this paper is the use of a 'convolution trick' to factor the calculation of integrals providing bounds on collision risk, enabling an $O(1)$ computation even in cluttered and complex environments.**

I. INTRODUCTION

[3]. Therefore, we can only treat such perception modules as being able to provide us with a probability distribution [4] over poses of the various objects in the scene. Achieving safe motion planning in such a setting will require the motion planning methods to turn these into probabilities of unsafe events (such as a collision of the robot with another agent), providing at least approximate assurance regarding the (non-)occurrence of these events.

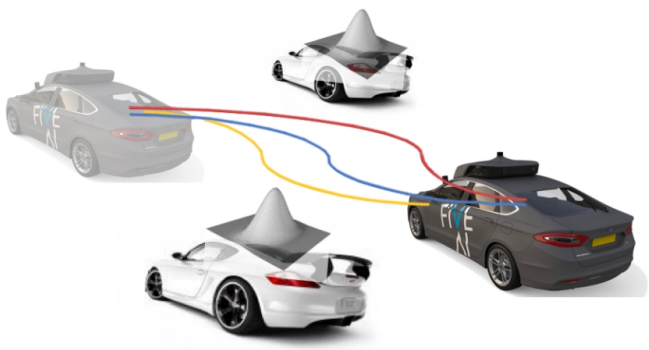


Fig. 1: A visualisation of the motion planning problem

methods. For instance, the Rapidly-exploring Random Tree (RRT) algorithm can utilise this probability within the search process. Likewise, a variational formulation of optimal control [5] could include this within the cost terms.

B. Related Work

The issue of safety in control and motion planning has been investigated from a number of different methodological

All authors are affiliated with FiveAI Ltd., Edinburgh, UK; Authors are listed in alphabetical order.
†Corresponding author: s.ramamoorthy@five.ai

Example: Autonomous Driving (in industry)

UNIVERSITY OF OXFORD

[cs.RO] 6 Mar 2019

Uncertainty-Aware Driver Trajectory Prediction at Urban Intersections

Xin Huang[1,2], Stephen G. McGill[1], Brian C. Williams[2], Luke Fletcher[1], Guy Rosman[1]

***Abstract*— Predicting the motion of a driver's vehicle is crucial for advanced driving systems, enabling detection of potential risks towards shared control between the driver and automation systems. In this paper, we propose a variational neural network approach that predicts future driver trajectory distributions for the vehicle based on multiple sensors.**

Our predictor generates both a conditional variational distribution of future trajectories, as well as a confidence estimate for different time horizons. Our approach allows us to handle inherently uncertain situations, and reason about information gain from each input, as well as combine our model with additional predictors, creating a mixture of experts.

We show how to augment the variational predictor with a physics-based predictor, and based on their confidence estimations, improve overall system performance. The resulting combined model is aware of the uncertainty associated with its predictions, which can help the vehicle autonomy to make decisions with more confidence. The model is validated on real-world urban driving data collected in multiple locations. This validation demonstrates that our approach improves the prediction error of a physics-based model by 25% while successfully identifying the uncertain cases with 82% accuracy.

I. Introduction

put deterministic results efficiently. However, these methods fail to capture the uncertain nature of human actions. Probabilistic predictions are very useful in many safety-critical tasks such as collision checking and risk-aware motion planning. They can express both the intrinsically uncertain

This work was part of X. Huang's internship at Toyota Research Institute (TRI). However, this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

[1]Toyota Research Institute, Cambridge, MA 02139, USA {stephen.mcgill,luke.fletcher,guy.rosman}@tri.global

[2]Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA 01239, USA {xhuang,williams}@csail.mit.edu

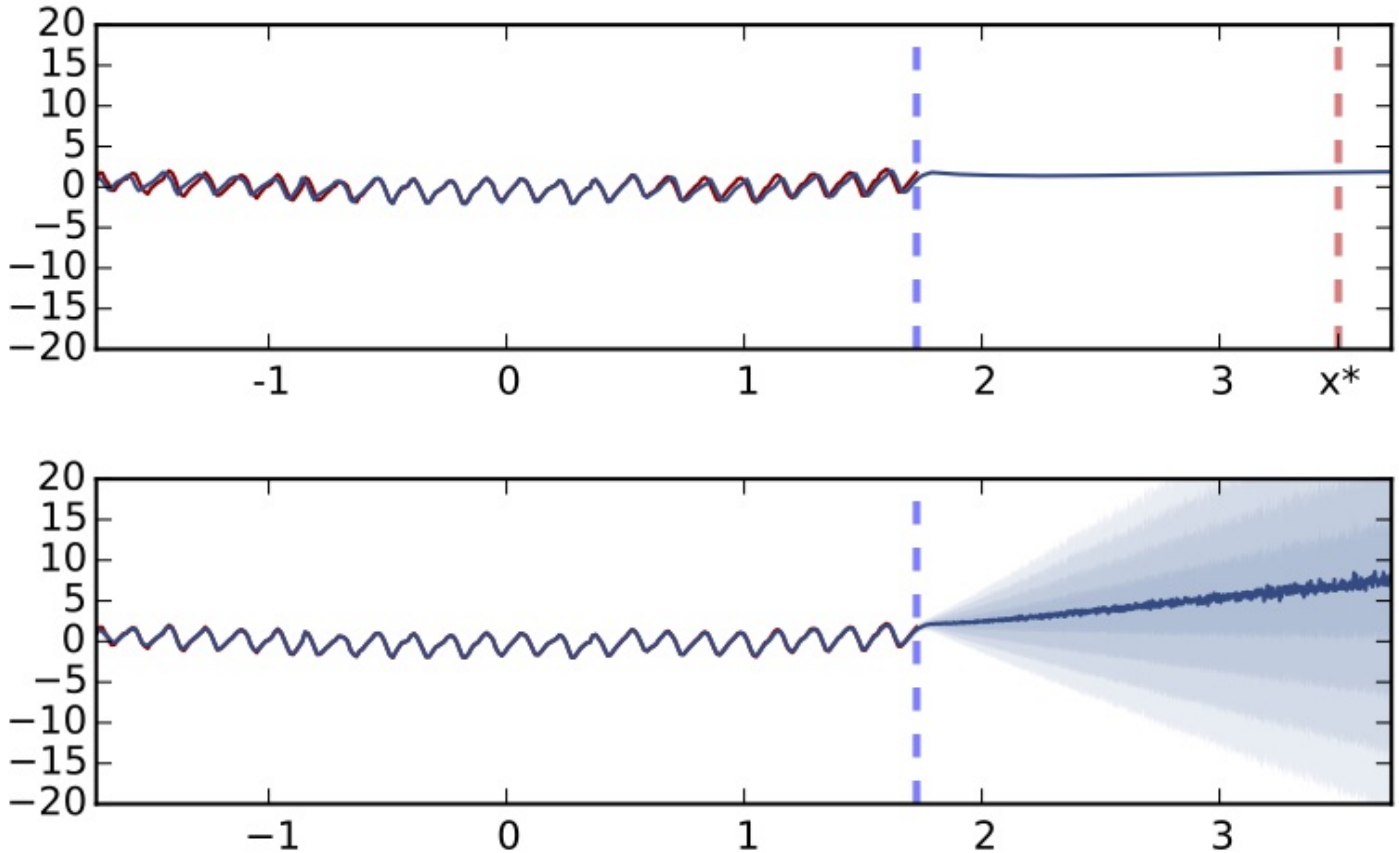
Video demonstration is available at https://youtu.be/clR08hRdtlM

Fig. 1: Illustration of a motivating example where a vehicle is in front of an intersection. The sampled predicted trajectories using our approach are plotted in blue, where the groundtruth future trajectory is plotted in red. In parallel autonomy, the autonomous system can leverage the predicted driver trajectory to avert risk and improve the driving. This requires the system to be confident of its predicted trajectories.

urban driving prediction. Intersections, for instance, were responsible for 40% of crashes happened in the United States in 2008 [9]. We therefore focus on predicting trajectories for vehicles driving in urban environments. This is more challenging than highway trajectory prediction due to more complicated environments with different road shapes and dynamic objects, as well as the variety of available driving actions for the driver. Additionally, in many cases it is crucial to be aware of the confidence of the prediction. In cases where those predictions cannot be accurately made, a later planning or parallel autonomy layer can take this into account, avoiding catastrophic outcomes due to mispredictions.

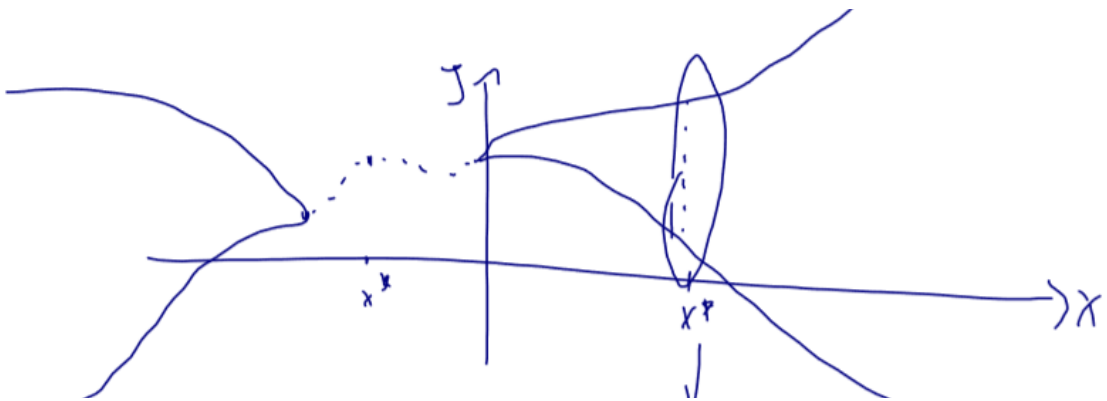
Sources of uncertainty

- Above are some *examples* of sources of uncertainty
- Many different sources of uncertainty



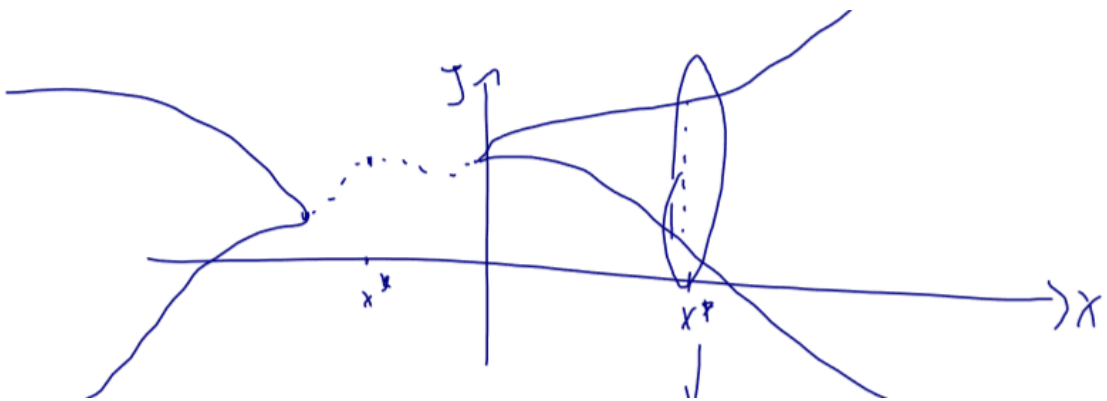
Sources of uncertainty (cnt)

- Uncertainty in model parameters that best explain data
  - *model* is defined by its parameters (parametrising a function)
  - large number of possible models can explain a dataset
  - uncertain which model parameters to choose to predict with
  - affects how we predict with new test points



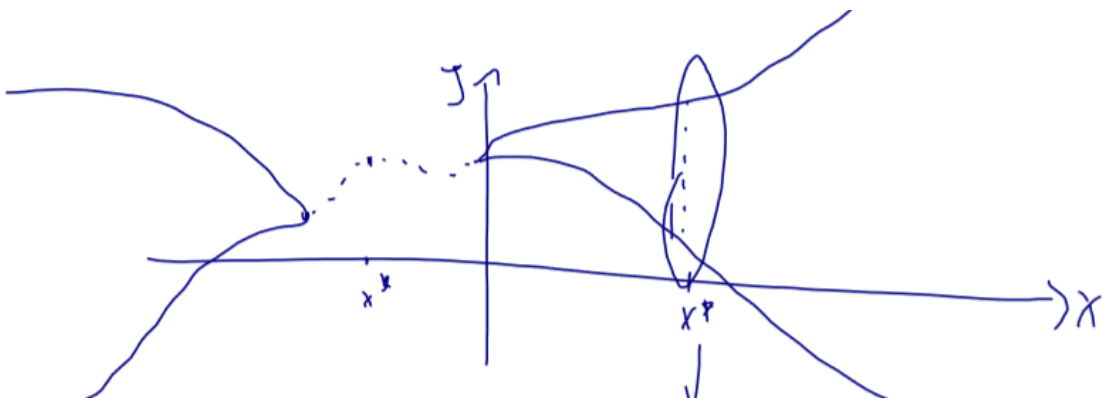
Sources of uncertainty (cnt)

- Eg when test data is very dissimilar to training data
  - model trained on diabetes fondus photos of subpopulation A
  - never saw subpopulation B
    - "images are outside data distribution model was trained on"
  - desired behaviour
    - return a prediction (attempting to extrapolate)
    - +information that image lies outside data distribution
  - Retraining model w subpop. B labels → low uncertainty on these



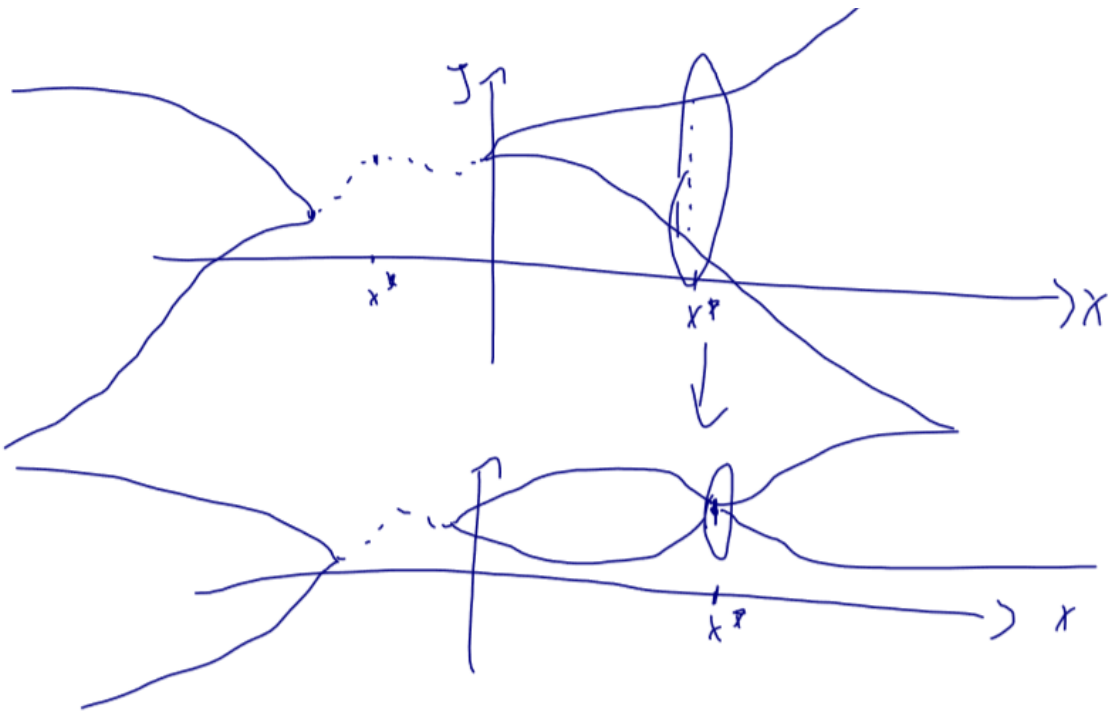
Sources of uncertainty (cnt)

- Eg when test data is very dissimilar to training data
  - model trained on diabetes fondus photos of subpopulation A
  - never saw subpopulation B
    - "images are outside data distribution model was trained on"
  - desired behaviour
    - return a prediction (attempting to extrapolate)
    - +information that image lies outside data distribution
  - Retraining model w subpop. B labels → low uncertainty on these



Sources of uncertainty (cnt)

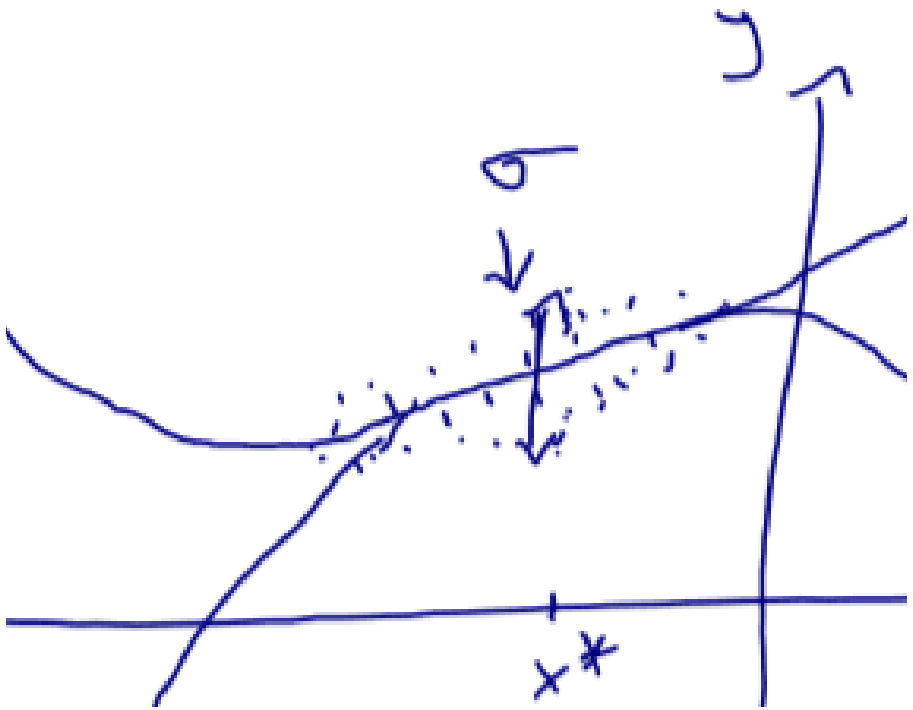
- Eg when test data is very dissimilar to training data
  - model trained on diabetes fondus photos of subpopulation A
  - never saw subpopulation B
    - "images are outside data distribution model was trained on"
  - desired behaviour
    - return a prediction (attempting to extrapolate)
    - +information that image lies outside data distribution
  - Retraining model w subpop. B labels $\rightarrow$ low uncertainty on these



Sources of uncertainty (cnt)

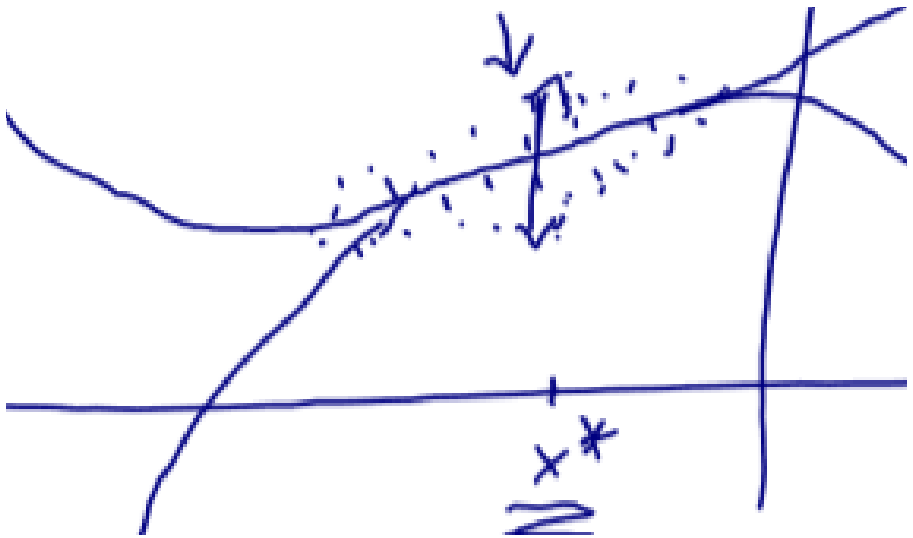
- Another source - training labels are noisy
  - measurement imprecision
  - expert mistakes
  - crowd sourced labels

even infinity data $\rightarrow$ ambiguity inherent in data itself



Uncertainty vs error

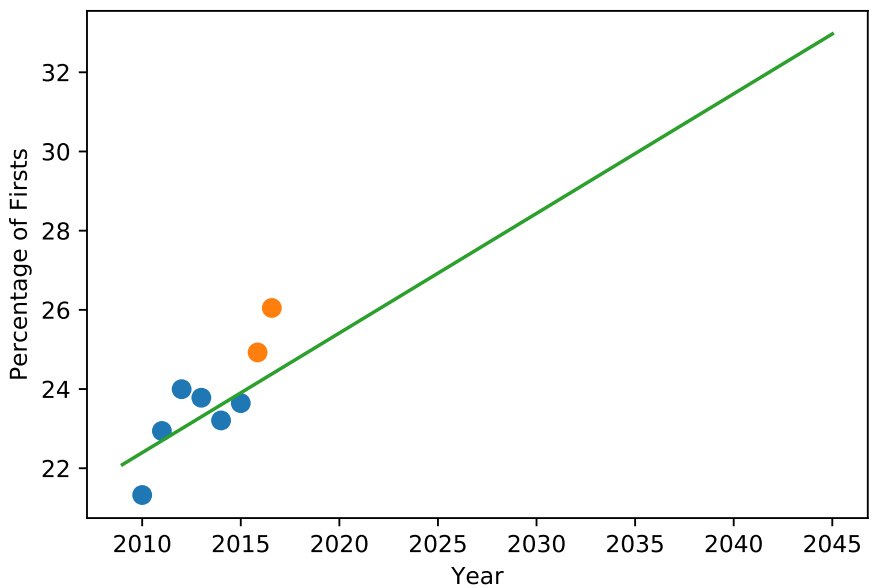
- Error is the discrepancy between the model's prediction and the observed outcome
- Uncertainty is a measure of reliability of the model's prediction
- We *would like* models that have high uncertainty when the model makes an error and low otherwise–but this is not guaranteed
  - This is a property of modelling assumptions, inference, optimisation, etc
  - Can you come up with an example of model error & low uncertainty?



Deep learning models are deterministic

Deep learning does not capture uncertainty:

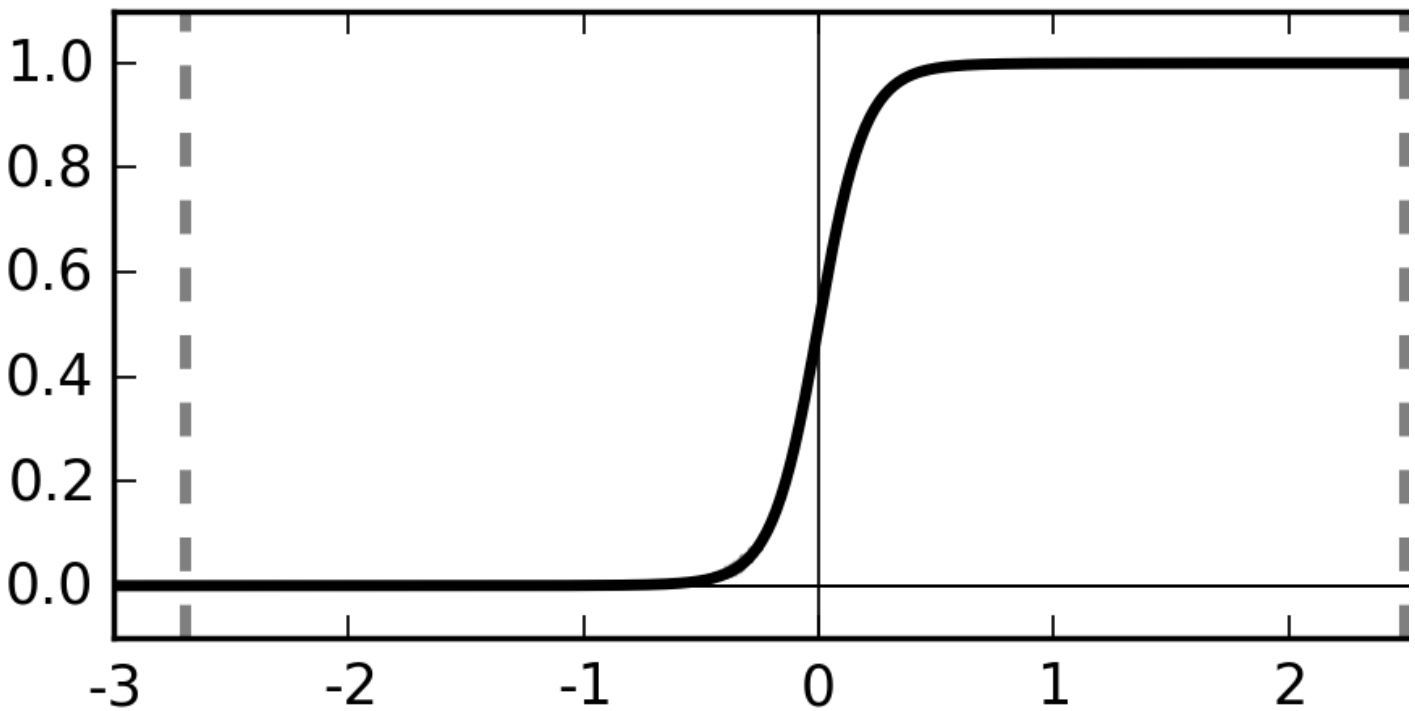
- regression models output a single scalar/vector
- classification models output a probability vector (only capturing one type of uncertainty from above!)



Deep learning models are deterministic

Deep learning does not capture uncertainty:

- regression models output a single scalar/vector
- classification models output a probability vector (only capturing one type of uncertainty from above!)



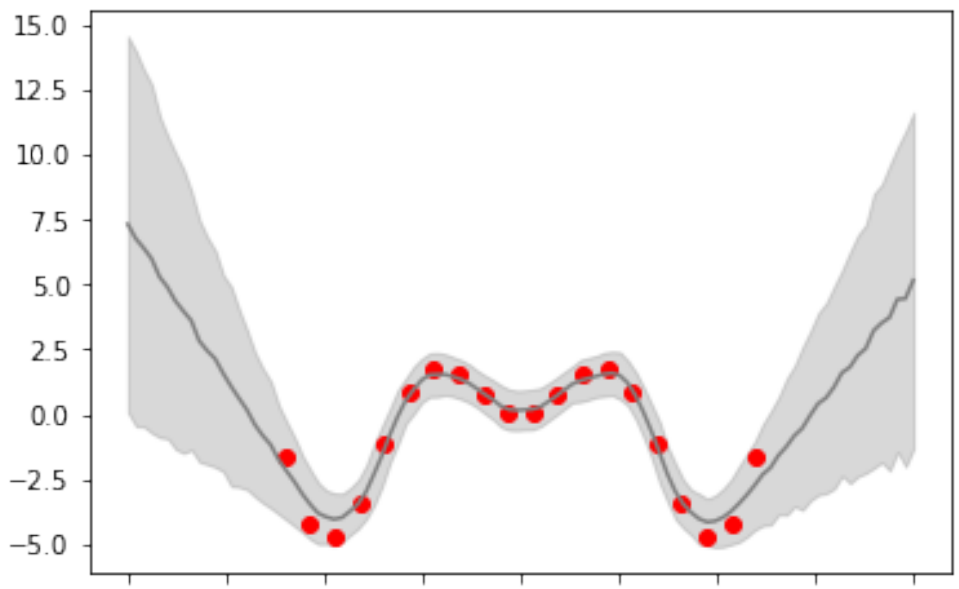
Deep learning models are deterministic

Deep learning does not capture uncertainty:

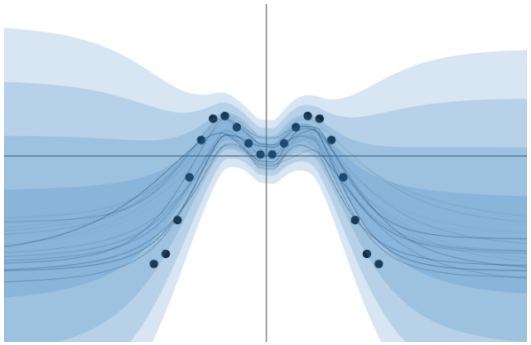
- regression models output a single scalar/vector
- classification models output a probability vector (only capturing one type of uncertainty from above!)

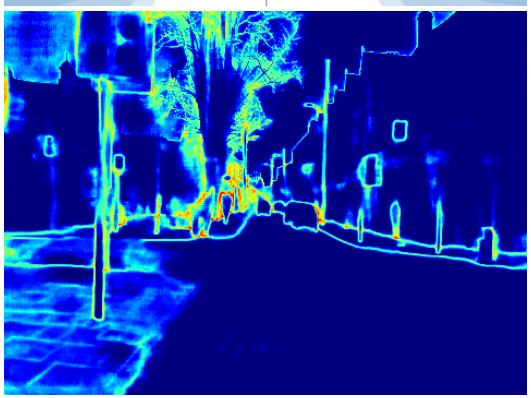
But when combined with *probability theory* **can** capture uncertainty in a principled way

$\rightarrow$ known as **Bayesian Deep Learning**



In the next couple of lectures...





We'll develop the tools to *understand*

- the formal language of **uncertainty** (Bayesian probability theory)
- tools to use this language in **ML** (Bayesian prob. modelling)
- and get a taste of defining **uncertainty** over functions.

Please bring a pen and paper! We'll do lots of fun derivations

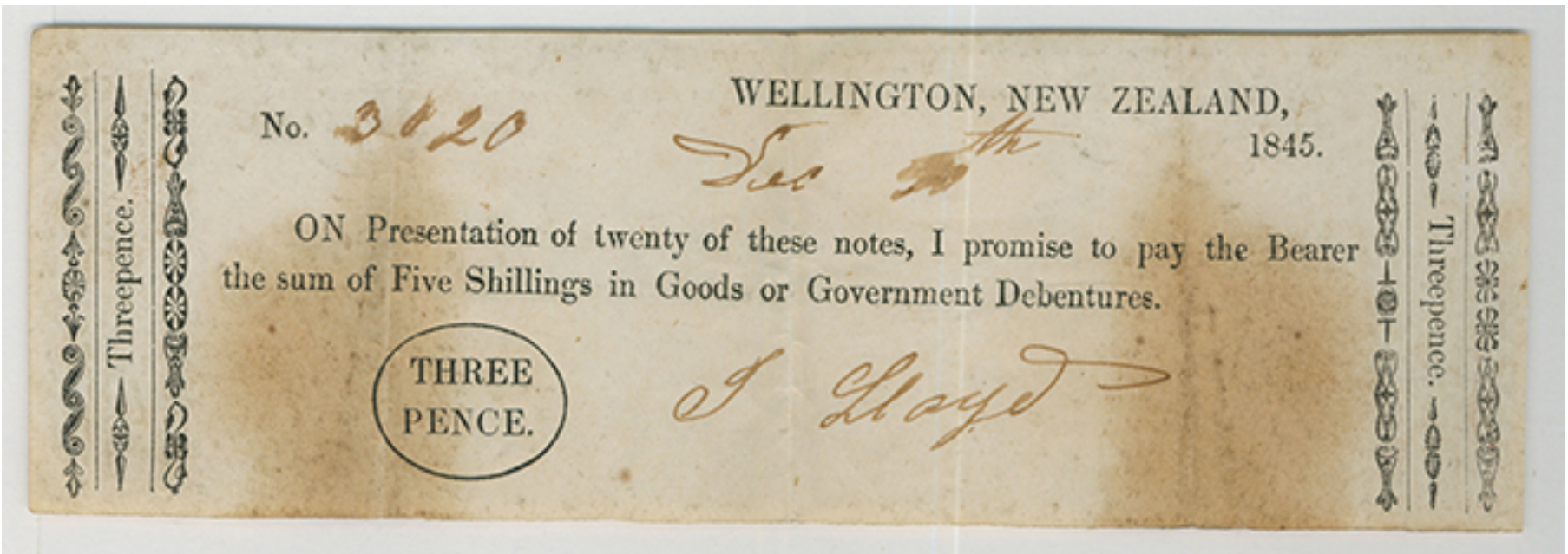
UNIVERSITY OF OXFORD

Bayesian Probability Theory: the Language of Uncertainty

Deriving the laws of *probability theory* from *rational beliefs*

Structure

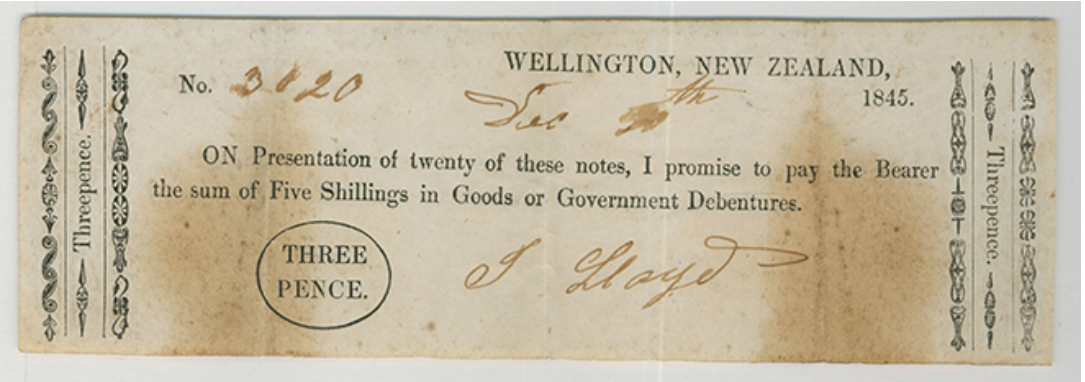
- Betting games
- ‘Belief’ as willingness to wager, and rational beliefs
- Deriving the laws of probability from rational beliefs, and interpretations of probability



Betting games (some philosophy for the soul)

UNIVERSITY OF OXFORD

Unit wager: a 'promise note' where seller commits to pay note owner £1 if a pre-specified event took place



- Eg "I promise to pay the bearer of this note £1 if Lucky Charm wins the horse race on 1/1/1845"
- Tradable – you can buy a wager from a bookie for 40p and sell to your friend for 50p
- If Lucky Charm won the race specified, you can cash the note to get £1
- If Lucky Charm lost the race, the note is worth nothing

Betting game 1a (some philosophy for the soul)

UNIVERSITY OF OXFORD

I toss a fair coin:

```
1 import numpy as np
2 def toss():
3   if np.random.rand() < 0.5:
4     print('Heads')
5   else:
6     print('Tails')
```

The pre-specified event is "toss outcome is heads", and bookie has a note to sell.

- would you pay p=£0.01 for a unit wager on 'heads'?
  - ie pay a penny to buy a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'
- p=£0.99?
  - ie pay 99 pence for a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'
- up to £0.05?, £0.95 or above?, up to £0.25?, ... [whiteboard]

Betting game 1a (some philosophy for the soul)

UNIVERSITY OF OXFORD

I toss a fair coin:

```
1 import numpy as np
2 def toss():
3   if np.random.rand() < 0.5:
4     print('Heads')
5   else:
6     print('Tails')
```

The pre-specified event is "toss outcome is heads", and bookie has a note to sell.

- would you pay p=£0.01 for a unit wager on 'heads'?
  - ie pay a penny to buy a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'
- p=£0.99?
  - ie pay 99 pence for a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'
- up to £0.05?, £0.95 or above?, up to £0.25?, ... [whiteboard]

Betting game 1a (some philosophy for the soul)

UNIVERSITY OF OXFORD

I toss a fair coin:

```
1 import numpy as np
2 def toss():
3   if np.random.rand() < 0.5:
4     print('Heads')
5   else:
6     print('Tails')
```

The pre-specified event is "toss outcome is heads", and bookie has a note to sell.

- would you pay p=£0.01 for a unit wager on 'heads'?
  - ie pay a penny to buy a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'
- p=£0.99?
  - ie pay 99 pence for a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'
- up to £0.05?, £0.95 or above?, up to £0.25?, ... [whiteboard]

Betting game 1a (some philosophy for the soul)

UNIVERSITY OF OXFORD

- p=£0.01 for a unit wager on 'heads'
  - ie pay a penny to buy a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'
- p=£0.99
  - ie pay 99 pence for a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'

- Let's collect some p's for more events (I need two volunteers from the audience)
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1a (some philosophy for the soul)

- p=£0.01 for a unit wager on 'heads'
  - ie pay a penny to buy a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'
- p=£0.99
  - ie pay 99 pence for a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'

- Let's collect some p's for more events (I need two volunteers from the audience)
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1a (some philosophy for the soul)

- p=£0.01 for a unit wager on 'heads'
  - ie pay a penny to buy a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'
- p=£0.99
  - ie pay 99 pence for a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'

- Let's collect some p's for more events (I need two volunteers from the audience)
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1a (some philosophy for the soul)

UNIVERSITY OF OXFORD

- p=£0.01 for a unit wager on 'heads'
  - ie pay a penny to buy a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'
- p=£0.99
  - ie pay 99 pence for a note where bookie commits to give you £1 if toss is 'heads', but they give you nothing if 'tails'

- Let's collect some p's for more events (I need two volunteers from the audience)
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1b (some philosophy for the soul)

UNIVERSITY OF OXFORD

```
1 import numpy as np
2 def toss():
3   if np.random.rand() < 0.5:
4     print('Heads')
5   else:
6     print('Tails')
```

New game:

- Now you're the bookie. You commits to paying £1 if outcome='heads' to the bearer of the note, but **you can choose whether to sell or not for p pence** (buyer always buys).
- a unit wager at £p for 'heads'
  - ie you get £p for the note, and have to pay £1 if heads
- would you **sell** up to £0.05?, £0.95 or above?, up to £0.25?, ... [whiteboard]

Betting game 1b (some philosophy for the soul)

- Now you're the bookie. You commits to paying £1 if outcome='heads' to the bearer of the note, but **you can choose whether to sell or not for p pence** (buyer always buys).
- a unit wager at £p for 'heads'
  - ie you get £p for the note, and have to pay £1 if heads

- Let's collect some p's for more events
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1b (some philosophy for the soul)

- Now you're the bookie. You commits to paying £1 if outcome='heads' to the bearer of the note, but **you can choose whether to sell or not for p pence** (buyer always buys).
- a unit wager at £p for 'heads'
  - ie you get £p for the note, and have to pay £1 if heads

- Let's collect some p's for more events
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1b (some philosophy for the soul)

UNIVERSITY OF OXFORD

- Now you're the bookie. You commits to paying £1 if outcome='heads' to the bearer of the note, but **you can choose whether to sell or not for p pence** (buyer always buys).
- a unit wager at £p for 'heads'
  - ie you get £p for the note, and have to pay £1 if heads

- Let's collect some p's for more events
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1b (some philosophy for the soul)

UNIVERSITY OF OXFORD

- Now you're the bookie. You commits to paying £1 if outcome='heads' to the bearer of the note, but **you can choose whether to sell or not for p pence** (buyer always buys).
- a unit wager at £p for 'heads'
  - ie you get £p for the note, and have to pay £1 if heads

- Let's collect some p's for more events
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1c (some philosophy for the soul)

UNIVERSITY OF OXFORD

```
1 import numpy as np
2 def toss():
3   if np.random.rand() < 0.5:
4     print('Heads')
5   else:
6     print('Tails')
```

- What if you're a trader, you set price £p; If customer is interested in buying the note you **must sell** it at £p, and if customer is interested in selling you a note you **must buy** it at £p.
  - **Customer decides** whether to sell to you, or buy from you
  - Customer sells: you pay £p to buy note where customer commits to pay £1 if heads
  - Customer buys: you get £p for note, and have to pay customer £1 if heads

- You **must** set a price: up to £0.05?, £0.95 or above?, ... [whiteboard]

Betting game 1c (some philosophy for the soul)

UNIVERSITY OF OXFORD

```
1 import numpy as np
2 def toss():
3   if np.random.rand() < 0.5:
4     print('Heads')
5   else:
6     print('Tails')
```

- What if you're a trader, you set price £p; If customer is interested in buying the note you **must sell** it at £p, and if customer is interested in selling you a note you **must buy** it at £p.
  - **Customer decides** whether to sell to you, or buy from you
  - Customer sells: you pay £p to buy note where customer commits to pay £1 if heads
  - Customer buys: you get £p for note, and have to pay customer £1 if heads
- You **must** set a price: up to £0.05?, £0.95 or above?, ... [whiteboard]

Betting game 1c (some philosophy for the soul)

UNIVERSITY OF OXFORD

- What if you're a trader, you set price £p; If customer is interested in buying the note you **must sell** it at £p, and if customer is interested in selling you a note you **must buy** it at £p.
  - **Customer decides** whether to sell to you, or buy from you
  - Customer sells: you pay £p to buy note where customer commits to pay £1 if heads
  - Customer buys: you get £p for note, and have to pay customer £1 if heads

---

- Let's collect some p's for more events
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1c (some philosophy for the soul)

UNIVERSITY OF OXFORD

- What if you're a trader, you set price £p; If customer is interested in buying the note you **must sell** it at £p, and if customer is interested in selling you a note you **must buy** it at £p.
  - **Customer decides** whether to sell to you, or buy from you
  - Customer sells: you pay £p to buy note where customer commits to pay £1 if heads
  - Customer buys: you get £p for note, and have to pay customer £1 if heads

---

- Let's collect some p's for more events
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1c (some philosophy for the soul)

UNIVERSITY OF OXFORD

- What if you're a trader, you set price £p; If customer is interested in buying the note you **must sell** it at £p, and if customer is interested in selling you a note you **must buy** it at £p.
  - **Customer decides** whether to sell to you, or buy from you
  - Customer sells: you pay £p to buy note where customer commits to pay £1 if heads
  - Customer buys: you get £p for note, and have to pay customer £1 if heads

---

- Let's collect some p's for more events
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Betting game 1c (some philosophy for the soul)

UNIVERSITY OF OXFORD

- What if you're a trader, you set price £p; If customer is interested in buying the note you **must sell** it at £p, and if customer is interested in selling you a note you **must buy** it at £p.
  - **Customer decides** whether to sell to you, or buy from you
  - Customer sells: you pay £p to buy note where customer commits to pay £1 if heads
  - Customer buys: you get £p for note, and have to pay customer £1 if heads

---

- Let's collect some p's for more events
  'I promise to pay the bearer of this note £1 if ...' [whiteboard]
  - 'code **if np.random.rand() < 0.3: print('Heads')** prints *Heads*'
  - 'It will rain tomorrow morning'
  - 'UK will win the next Eurovision'
  - 'DeepMind/OpenAI/etc will solve AGI within the next year'

Definition: Beliefs as willingness to wager

A person with degree of belief p in event A is assumed to be

- willing to pay anywhere up to £p for a unit wager on A
- and is willing to sell such a wager for any price higher equal to £p

This p captures the person's degree of **belief about the event A taking place** (aka **uncertainty**, confidence)

Some properties: This number p is

- subjective (might take different values for different people)
- but follows some regularities, as we'll see next

Definition: Rational beliefs

- Two notes (unit wagers):
  - Note #1: 'outcome=heads'
  - Note #2: 'outcome=tails'

  you decide p for note #1 and q for note #2; Customer decides whether to buy from you or sell to you each note at the price you determined.
- if $p + q < 1$ then customer will always buy from you both notes, #1 for £p and #2 for £q
- because whatever outcome happens, customer gets £1; but customer only paid $p + q < 1$, so they always make a profit of £$1 - (p + q) > 0$
- **I.e. you are guaranteed to lose £$1 - (p + q)$!**
- this is an example of a **Dutch book**

Definition: Rational beliefs

- Two notes (unit wagers):
  - Note #1: 'outcome=heads'
  - Note #2: 'outcome=tails'

  you decide p for note #1 and q for note #2; Customer decides whether to buy from you or sell to you each note at the price you determined.
- if $p + q < 1$ then customer will always buy from you both notes, #1 for £p and #2 for £q
- because whatever outcome happens, customer gets £1; but customer only paid $p + q < 1$, so they always make a profit of $£1 - (p + q) > 0$
- **I.e. you are guaranteed to lose $£1 - (p + q)$!**
- this is an example of a **Dutch book**

Definition: Rational beliefs

- Two notes (unit wagers):
  - Note #1: 'outcome=heads'
  - Note #2: 'outcome=tails'

  you decide p for note #1 and q for note #2; Customer decides whether to buy from you or sell to you each note at the price you determined.
- if $p + q < 1$ then customer will always buy from you both notes, #1 for £p and #2 for £q
- because whatever outcome happens, customer gets £1; but customer only paid $p + q < 1$, so they always make a profit of £$1 - (p + q) > 0$
- I.e. you are guaranteed to lose £$1 - (p + q)$!
- this is an example of a **Dutch book**

Definition: Rational beliefs

- Two notes (unit wagers):
  - Note #1: 'outcome=heads'
  - Note #2: 'outcome=tails'

  you decide p for note #1 and q for note #2; Customer decides whether to buy from you or sell to you each note at the price you determined.
- if $p + q < 1$ then customer will always buy from you both notes, #1 for £p and #2 for £q
- because whatever outcome happens, customer gets £1; but customer only paid $p + q < 1$, so they always make a profit of £$1 - (p + q) > 0$
- **I.e. you are guaranteed to lose £$1 - (p + q)$!**
- this is an example of a **Dutch book**

Definition: Rational beliefs

- Two notes (unit wagers):
  - Note #1: 'outcome=heads'
  - Note #2: 'outcome=tails'

  you decide p for note #1 and q for note #2; Customer decides whether to buy from you or sell to you each note at the price you determined.
- if $p + q < 1$ then customer will always buy from you both notes, #1 for £p and #2 for £q
- because whatever outcome happens, customer gets £1; but customer only paid $p + q < 1$, so they always make a profit of £$1 - (p + q) > 0$
- **I.e. you are guaranteed to lose £$1 - (p + q)$!**
- this is an example of a **Dutch book**

Definition: Rational beliefs

UNIVERSITY OF OXFORD

Dutch book:

- a set of unit wager notes where **you decide the odds** (wager price) and **customer decides whether to buy or sell** each note ... and you are guaranteed to always lose money! (regardless of the outcome of events)
- customer has to decide what to buy and what to sell based on wager odds only, without knowing in advance the outcome of events

Definition: Rational beliefs

UNIVERSITY OF OXFORD

Dutch book:

- a set of unit wager notes where **you decide the odds** (wager price) and **customer decides whether to buy or sell** each note ... and you are guaranteed to always lose money! (regardless of the outcome of events)
- customer has to decide what to buy and what to sell based on wager odds only, without knowing in advance the outcome of events

Set of beliefs is called **rational** if no Dutch book exists

ie there doesn't exist a set of wagers that will **always** lead to loss, regardless of event outcome

Formalism (rational beliefs = prob theory)

UNIVERSITY OF OXFORD

Setup

- Def sample space X of simple events (possible outcomes)
  - e.g. experiment flipping two coins X={HH, HT, TH, TT}
- Let A be an event (a subset of X). A holding true = at least one of the outcomes in A happened
  - e.g. "at least one heads" $\leftrightarrow$ A={HH, HT, TH}
- Write p_A for belief in event A (your wager on A happening, assuming all wagers are unit wagers)
- note: $\{p_A\}_{A \subseteq X}$ is a set of *numbers*, with a number assigned to each subset of X

Formalism (rational beliefs = prob theory)

UNIVERSITY OF OXFORD

Setup

- Def sample space X of simple events (possible outcomes)
  - e.g. experiment flipping two coins X={HH, HT, TH, TT}
- Let A be an event (a subset of X). A holding true = at least one of the outcomes in A happened
  - e.g. "at least one heads" $\leftrightarrow$ A={HH, HT, TH}
- Write p_A for belief in event A (your wager on A happening, assuming all wagers are unit wagers)
- note: $\{p_A\}_{A \subseteq X}$ is a set of *numbers*, with a number assigned to each subset of X

Claim: $\{p_A\}_{A \subseteq X}$ are rational beliefs iff $p(A) := p_A$ satisfies laws of probability theory, regardless of the individual to choose p_A

? what does it mean to have *different individuals* choosing different sets $\{p_A\}_{A \subseteq X}$?

- Already showed that $p_A + p_{A^c} < 1$ leads to a Dutch book
- Let's look at a few more...

Formalism (rational beliefs = prob theory)

UNIVERSITY OF OXFORD

Setup

- Def sample space X of simple events (possible outcomes)
  - e.g. experiment flipping two coins X={HH, HT, TH, TT}
- Let A be an event (a subset of X). A holding true = at least one of the outcomes in A happened
  - e.g. "at least one heads" $\leftrightarrow$ A={HH, HT, TH}
- Write p_A for belief in event A (your wager on A happening, assuming all wagers are unit wagers)
- note: $\{p_A\}_{A \subseteq X}$ is a set of *numbers*, with a number assigned to each subset of X

Claim: $\{p_A\}_{A \subseteq X}$ are rational beliefs iff $p(A) := p_A$ satisfies laws of probability theory, regardless of the individual to choose p_A

? what does it mean to have *different individuals* choosing different sets $\{p_A\}_{A \subseteq X}$?

- Already showed that $p_A + p_{A^c} < 1$ leads to a Dutch book
- Let's look at a few more...

Betting game 2 (rational beliefs = prob theory)

UNIVERSITY OF OXFORD

If $\{p_A\}_{A \subseteq X}$ are rational beliefs, then:

- $0 \leq p_A \leq 1 \qquad \forall A \subseteq X$
  - If $p > 1$: I sell you £p priced wager; You lose regardless of outcome
    - you give me £p > 1 for the wager, I need to give you at most £1
    - so $p > 1$ forms a Dutch book and is not rational
  - If $p < 0$: I buy from you for £p
    - you give me £p to sell me the wager, and then you give me £1 if heads

Betting game 2 (Exercise)

If $\{p_A\}_{A \subseteq X}$ are rational beliefs, then:

- $0 \leq p_A \leq 1 \qquad \forall A \subseteq X$
- Unit measure $p_X = 1$ (hint: what wager would you define?)

Exercise (discuss with your neighbour)

Betting game 2 (Exercise)

If $\{p_A\}_{A \subseteq X}$ are rational beliefs, then:

- $0 \leq p_A \leq 1 \qquad \forall A \subseteq X$
- Unit measure $p_X = 1$
- Two disjoint events satisfy $p_{A \cup B} = p_A + p_B$

Exercise (discuss with your neighbour)

Betting game 2 (rational beliefs = prob theory)

UNIVERSITY OF OXFORD

If $\{p_A\}_{A \subseteq X}$ are rational beliefs, then:

- $0 \leq p_A \leq 1 \qquad \forall A \subseteq X$
- Unit measure $p_X = 1$
- Two disjoint events satisfy $p_{A \cup B} = p_A + p_B$

- These are the axioms of probability theory: All laws of probability theory follow from these three axioms
- So we can call our set of numbers $\{p_A\}_{A \subseteq X}$ 'probabilities'
- $\rightarrow$ in ML, if you want your model to be rational (ie so no-one could take advantage of inconsistencies in your model's decisions), then **must** follow laws of probability theory

Betting game 2 (rational beliefs = prob theory)

UNIVERSITY OF OXFORD

If $\{p_A\}_{A \subseteq X}$ are rational beliefs, then:

- $0 \leq p_A \leq 1 \qquad \forall A \subseteq X$
- Unit measure $p_X = 1$
- Two disjoint events satisfy $p_{A \cup B} = p_A + p_B$

- These are the axioms of probability theory: All laws of probability theory follow from these three axioms
- So we can call our set of numbers $\{p_A\}_{A \subseteq X}$ 'probabilities'
- $\rightarrow$ in ML, if you want your model to be rational (ie so no-one could take advantage of inconsistencies in your model's decisions), then **must** follow laws of probability theory

Betting game 2 (rational beliefs = prob theory)

UNIVERSITY OF OXFORD

If $\{p_A\}_{A \subseteq X}$ are rational beliefs, then:

- $0 \leq p_A \leq 1 \qquad \forall A \subseteq X$
- Unit measure $p_X = 1$
- Two disjoint events satisfy $p_{A \cup B} = p_A + p_B$

- These are the axioms of probability theory: All laws of probability theory follow from these three axioms
- So we can call our set of numbers $\{p_A\}_{A \subseteq X}$ 'probabilities'
- $\rightarrow$ in ML, if you want your model to be rational (ie so no-one could take advantage of inconsistencies in your model's decisions), then **must** follow laws of probability theory

Betting game 2 (rational beliefs = prob theory)

UNIVERSITY OF OXFORD

If $\{p_A\}_{A \subseteq X}$ are rational beliefs, then:

- $0 \leq p_A \leq 1 \qquad \forall A \subseteq X$
- Unit measure $p_X = 1$
- Two disjoint events satisfy $p_{A \cup B} = p_A + p_B$

- These are the axioms of probability theory: All laws of probability theory follow from these three axioms
- So we can call our set of numbers $\{p_A\}_{A \subseteq X}$ 'probabilities'
- $\rightarrow$ in ML, if you want your model to be rational (ie so no-one could take advantage of inconsistencies in your model's decisions), then **must** follow laws of probability theory

Betting game 2 (rational beliefs = prob theory)

UNIVERSITY OF OXFORD

If $\{p_A\}_{A \subseteq X}$ are rational beliefs, then:

- $0 \leq p_A \leq 1 \qquad \forall A \subseteq X$
- Unit measure $p_X = 1$
- Two disjoint events satisfy $p_{A \cup B} = p_A + p_B$

- These are the axioms of probability theory: All laws of probability theory follow from these three axioms
- So we can call our set of numbers $\{p_A\}_{A \subseteq X}$ 'probabilities'
- $\rightarrow$ in ML, if you want your model to be rational (ie so no-one could take advantage of inconsistencies in your model's decisions), then **must** follow laws of probability theory

No arbitrary decisions!

Probability as belief vs frequency

Above known as **Bayesian probability theory**

- forms an **interpretation of the laws of probability** (a grounding of prob theory in reality), and formalises **our notion of uncertainty** in events
  - can be used eg for non-repeatable events ('who will win an election')
- contrast to 'Frequency as probability' interpretation of prob theory, which is
  - only applicable to repeatable events (eg, try to put prob on 'will Nixon win the election')
  - also other issues; eg p-hacking (leading to Psychology journal banning p values!)

(Note: there are problems w Bayesian arguments as well, out of scope for us)

Probability as belief vs frequency

Above known as **Bayesian probability theory**

- forms an **interpretation of the laws of probability** (a grounding of prob theory in reality), and formalises **our notion of uncertainty** in events
  - can be used eg for non-repeatable events ('who will win an election')
- contrast to 'Frequency as probability' interpretation of prob theory, which is
  - only applicable to repeatable events (eg, try to put prob on 'will Nixon win the election')
  - also other issues; eg p-hacking (leading to Psychology journal banning p values!)

(Note: there are problems w Bayesian arguments as well, out of scope for us)

Questions & discussion