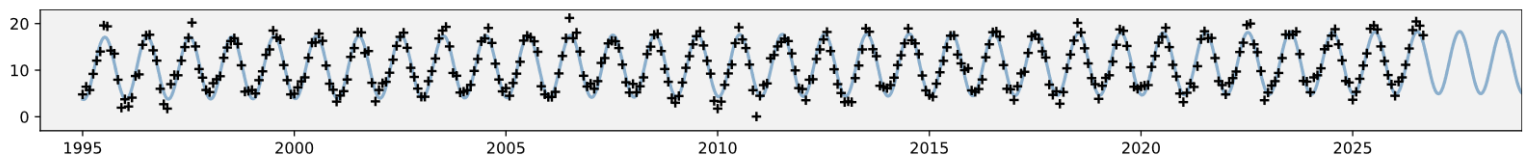


EXERCISE



Take the Cambridge monthly weather data from 1995 to the present.
[Code to load the data is in the slides for lecture 1.]

Estimate γ in the following model:

```
def rtemp(t,  $\alpha$ ,  $\phi$ , c,  $\gamma$ ,  $\sigma$ ):  
    pred = c +  $\alpha$  * sin(2* $\pi$ *(t+ $\phi$ )) +  $\gamma$ *t  
    return random.normal(loc=pred, scale= $\sigma$ )
```

0.003		6	
:			
0.023		7	
0.024			
0.025			
0.026			
0.027			
0.028			
0.029			
0.030		4	likely from way-off initial guesses
0.031			
0.032		0	
0.033			
0.034			
0.035			
0.036			
0.037			
0.038		0 3 3 3 3 3 3 3 3 4	
0.039		2 8	
0.040		8 8 8 8	fit the wrong data (2000 to present)
0.041		0 2	
0.042			
0.043			
0.044			
0.045			
0.046			
0.047		9	

TODAY

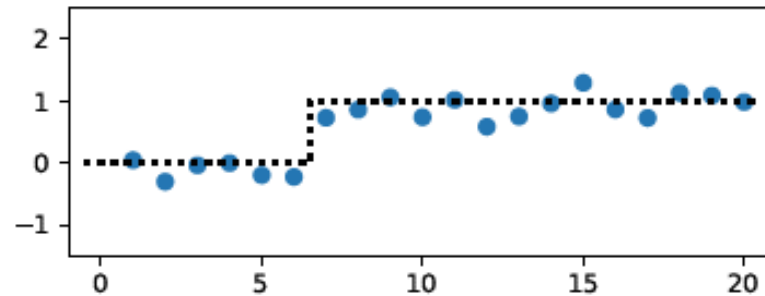
Mock exam question 1

Printouts are on the front bench, both sides.

- (a) A 0/1 signal is being transmitted. The transmitted signal at timeslot $i \in \{1, \dots, n\}$ is $x_i \in \{0, 1\}$, and we have been told that this signal starts at 0 and then flips to 1, i.e. there is a parameter $\theta \in \{1, \dots, n-1\}$ such that $x_i = 1_{i>\theta}$. The value of this parameter is unknown. The channel is noisy, and the received signal in timeslot i is

$$Y_i \sim x_i + \text{Normal}(0, \varepsilon^2)$$

where ε is known.



- (i) Given received signals $(y_1, \dots, y_n)$, find an expression for the log likelihood, $\log \Pr(y_1, \dots, y_n; \theta)$. Explain your working. [5 marks]
- (ii) Give pseudocode for finding the maximum likelihood estimator $\hat{\theta}$. [5 marks]

1. Skim read for keywords. What's the topic?

log likelihood
pseudocode

2. Look for question verbs. What is it asking you to do?

find expr
give pseudocode

3. Think through the course. What sections are relevant?

\$1. Fitting & specifying models.

(a) (i) Find an expression for the log likelihood

// Model: $Y_i \sim x_i + \text{Normal}(0, \varepsilon^2)$

$$\text{LHS} = \log \Pr(y_1, \dots, y_n; \Theta)$$

Sanity check: what are the parameters in the model?

$$x_i = 1_{i>0} \text{ - formula is given}$$

Θ - unknown

ε - known

$$= \sum_{i=1}^n \log \Pr_{Y_i}(y_i; \Theta)$$

$$= \sum_i \log \left\{ \frac{1}{\sqrt{2\pi\varepsilon^2}} e^{-(y_i - x_i)^2 / 2\varepsilon^2} \right\}$$

since $Y_i \sim N(x_i, \varepsilon^2)$, by linearity of the normal

$$= -\frac{n}{2} \log(2\pi\varepsilon^2) - \frac{1}{2\varepsilon^2} \sum_i (y_i - 1_{i>0})^2$$

— it's good manners to write our answer in terms of the unknown parameter Θ , as the question asks.

$$= \text{RHS.}$$

SANITY CHECK. Do the variables on LHS match those on the right?

LHS: $y_1, \dots, y_n, \Theta$

RHS: n ε i y_i Θ
| | |
given known dummy



(ii) Give pseudocode for finding MLE $\hat{\theta}$

We need to find $\max_{\theta} \log \Pr(y_1, \dots, y_n; \theta)$

naïve answer:

~~def loglik(θ): return ---- (formula we just derived)~~
~~scipy.optimize.fmin(lambda θ : -loglik(θ), initguess)~~ $\leftarrow$ b, by eye?

scipy.optimize.fmin is only for optimizing over continuous parameters, we want $\theta \in \{1, \dots, n-1\}$.

vals = [(loglik(θ), θ) for $\theta \in \{1, \dots, n-1\}$]

(-, $\hat{\theta}$) = max(vals)

Indicator functions

ex 1.3.6, §2.2.1

Indicator functions are a notational trick that lets us turn if/then conditions into algebra. The indicator function is defined to be

$$1_A = \begin{cases} 1 & \text{if statement } A \text{ is true} \\ 0 & \text{if statement } A \text{ is false} \end{cases}$$

$$1_{A \text{ and } B} = 1_A \cdot 1_B$$

It turns AND into MULTIPLICATION

In IA Discrete Maths [lecture 17] you used the indicator for a subset: $\chi_S(a) = 1_{a \in S}$

There are standard numerical random variables that you should know:

DISCRETE RANDOM VARIABLES

Binomial $X \sim \text{Bin}(n, p)$	$\mathbb{P}(X = x) = \binom{n}{x} p^x (1 - p)^{n-x}$ $x \in \{0, 1, \dots, n\}$	For count data, e.g. number of heads in n coin tosses
Poisson $X \sim \text{Pois}(\lambda)$	$\mathbb{P}(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$ $x \in \{0, 1, \dots\}$	For count data, e.g. number of buses passing a spot
Categorical $X \sim \text{Cat}([p_1, \dots, p_k])$	$\mathbb{P}(X = x) = p_x$ $x \in \{1, \dots, k\}$	For picking one of a fixed number of choices

CONTINUOUS RANDOM VARIABLES

Uniform $X \sim U[a, b]$	$\text{pdf}(x) = \frac{1}{b - a}$ $x \in [a, b]$	A uniformly-distributed floating point value
Normal / Gaussian $X \sim N(\mu, \sigma^2)$	$\text{pdf}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$ $x \in \mathbb{R}$	For data about magnitudes, e.g. temperature or height
Pareto $X \sim \text{Pareto}(\alpha)$	$\text{pdf}(x) = \alpha x^{-(\alpha+1)}$ $x \geq 1$	For data about “cascade” magnitudes, e.g. forest fires
Exponential $X \sim \text{Exp}(\lambda)$	$\text{pdf}(x) = \lambda e^{-\lambda x}$ $x > 0$	For waiting times, e.g. time until next bus
Beta $X \sim \text{Beta}(a, b)$	$\text{pdf}(x) \propto x^{a-1} (1 - x)^{b-1}$ $x \in (0, 1)$	Arises in Bayesian inference

There are standard numerical random variables that you should know:

Useful properties of the Normal distribution:

- If we rescale a Normal, we get a Normal
- If we add independent Normals, we get a Normal

$a + b N(0,1) \sim a + N(0, b^2) \sim N(a, b^2)$
for constants a, b
like Uniform, Exponential

$N(\mu, \sigma^2) + N(\nu, \rho^2) \sim N(\mu + \nu, \sigma^2 + \rho^2)$
like Poisson, (Binomial)

These two properties are known as “linearity of the Normal distribution”.

Normal / Gaussian
 $X \sim N(\mu, \sigma^2)$

$$\text{pdf}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$

$x \in \mathbb{R}$

For data about magnitudes, e.g. temperature or height

(b) I have been monitoring average annual river levels for many years, and I have collected a dataset $(z_0, \dots, z_n)$ where z_i is the level in year i since I started monitoring. I believe that for the first few years the level each year was roughly what it was the previous year, plus or minus some random variation; but that some year a drought started, and since then the level has decreased on average each year. I would like to estimate when the drought started. I do not know the other parameters.

(i) Propose a probability model for my dataset. [5 marks]

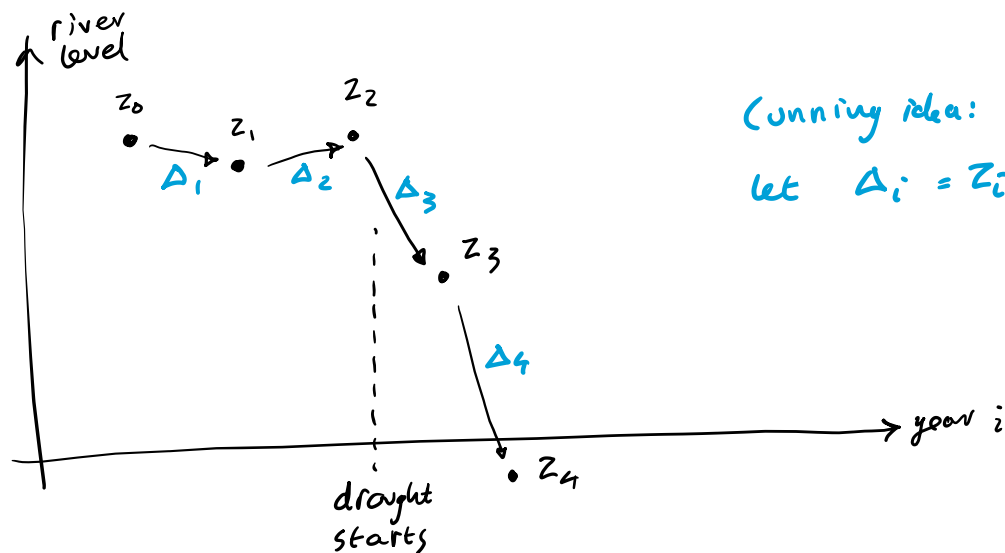
(ii) Explain how to fit your model. [5 marks]

This "propose a probability model" is open-ended and scary. How should we even begin to think about it?

1. Skim read for keywords. What's the topic?
2. Look for question words. What is it asking you to do?
3. Think through the course. What sections are relevant?
4. Read the whole question. What's the link?

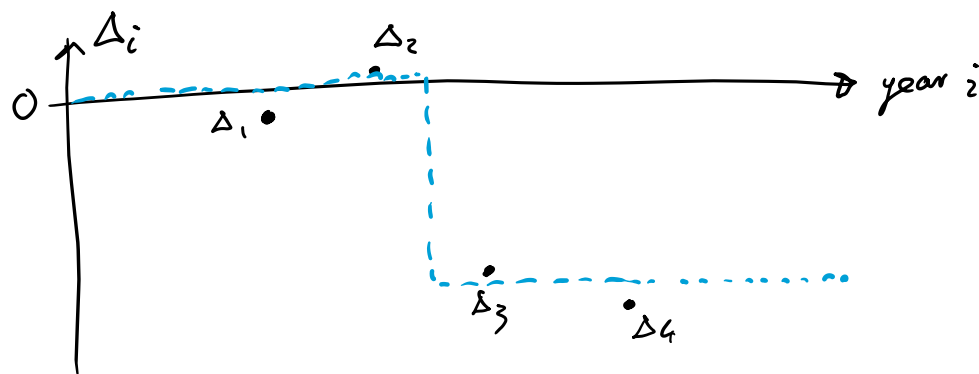
Part (a) gave us a hammer. Can we see part (b) as a nail?

(b) For this sort of question, I find a sketch helps.



(Cunning idea: look at changes in Z .

let $\Delta_i = Z_i - Z_{i-1}$.



Proposed model:

$\Delta_i \sim N(0, \epsilon^2)$ for $i \leq \Theta$

$\Delta_i \sim N(-\mu, \epsilon^2)$ for $i > \Theta$

Equivalently,

$\Delta_i \sim N(-\mu 1_{i>\Theta}, \epsilon^2)$.

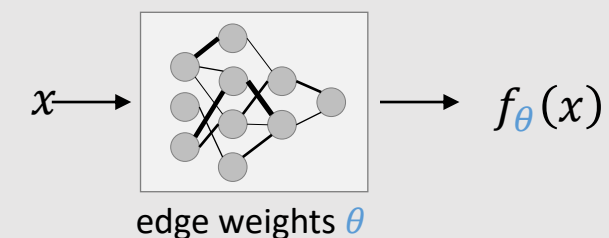
TASK: write out $\log \Pr(\delta_1, \dots, \delta_n; \mu, \epsilon, \Theta)$ — just the same as before

then find MLE for μ, ϵ, Θ . — watch out! $\mu, \epsilon \in \mathbb{R}$ so you can use `scipy.optimize.fmin`
 Θ is discrete.

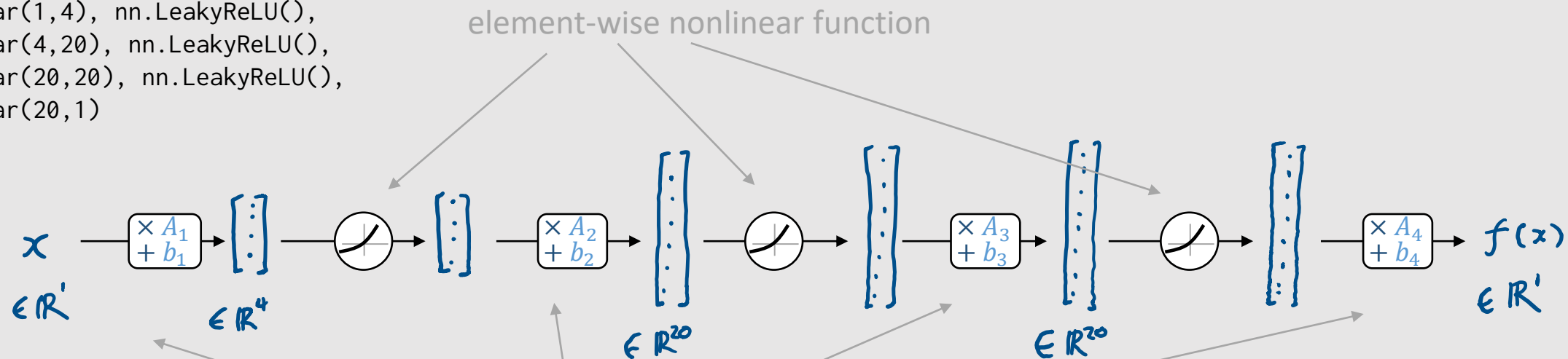
Neural networks
are probability
models*

* non-examinable

Think of a neural network as a black-box function with parameters.



```
f = nn.Sequential(
    nn.Linear(1,4), nn.LeakyReLU(),
    nn.Linear(4,20), nn.LeakyReLU(),
    nn.Linear(20,20), nn.LeakyReLU(),
    nn.Linear(20,1)
)
```



parameters $\theta = [A_1, A_2, A_3, A_4, b_1, b_2, b_3, b_4]$

matrix multiplication

THIS IS YOUR MACHINE LEARNING SYSTEM?

YUP! YOU POUR THE DATA INTO THIS BIG
PILE OF LINEAR ALGEBRA, THEN COLLECT
THE ANSWERS ON THE OTHER SIDE.

WHAT IF THE ANSWERS ARE WRONG?

JUST STIR THE PILE UNTIL
THEY START LOOKING RIGHT.



<https://xkcd.com/1838>

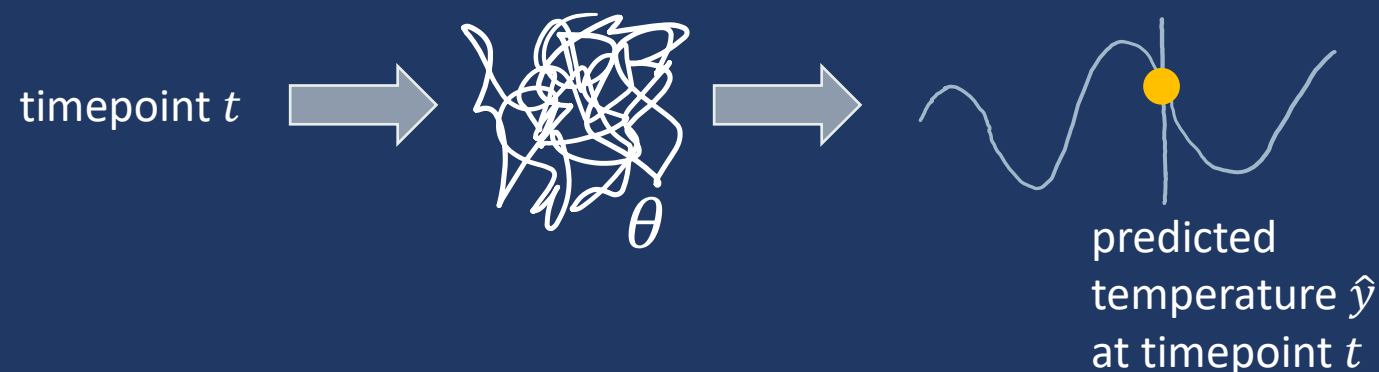
CC BY-NC 2.5

PREDICTION MINDSET IN MACHINE LEARNING

We're given a labelled dataset.

We want to learn to predict the label.

We do this by minimizing a loss function that measures prediction error.



ground-truth labels:

Let y_i be the actual observed temperature at time t_i

minimize prediction loss:

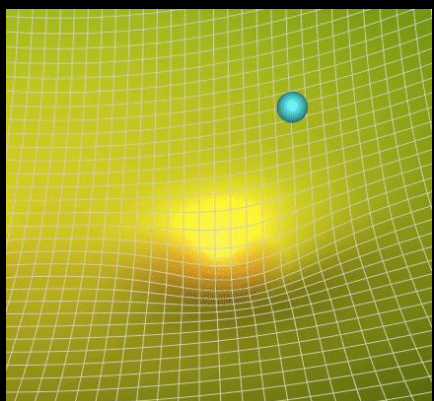
pick θ to minimize

$$\sum_i L(y_i, \hat{y}_\theta(t_i))$$

where

$$L(\text{obs}, \text{pred}) = (\text{obs} - \text{pred})^2$$

We can use gradient descent to fit θ



All that matters about the neural network function is

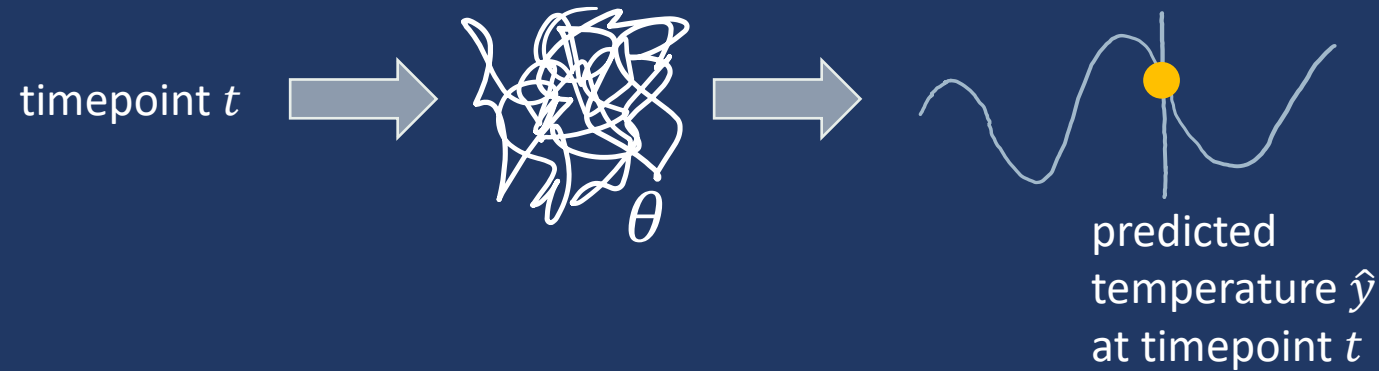
- it takes an input, the timepoint t
- it has parameters, θ
- it's differentiable with respect to θ

The loss function L is arbitrary, but it had better be differentiable with respect to pred .

This course teaches a different way to think of modelling ...

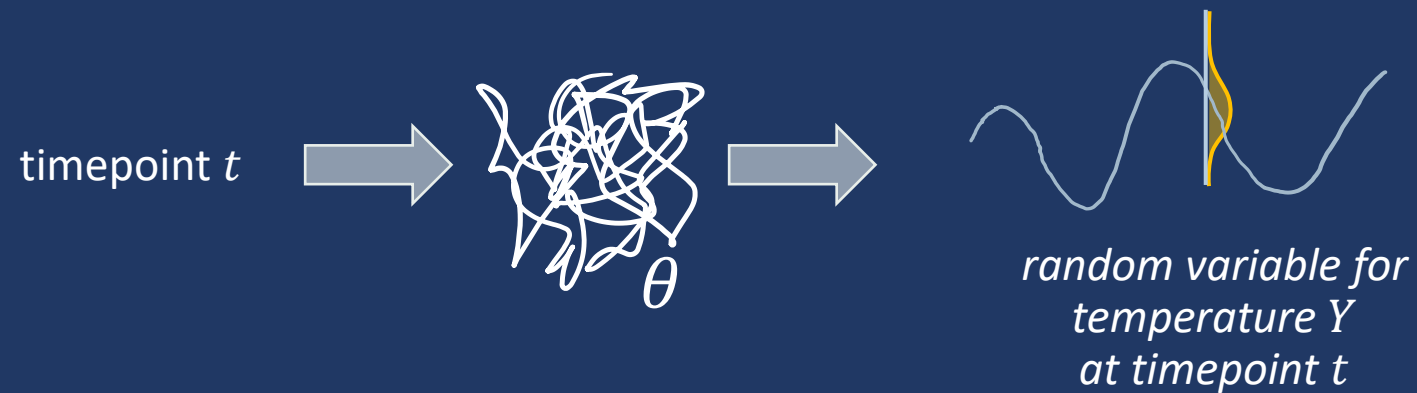
§3.2

PREDICTION MINDSET



Goal:
minimize prediction error

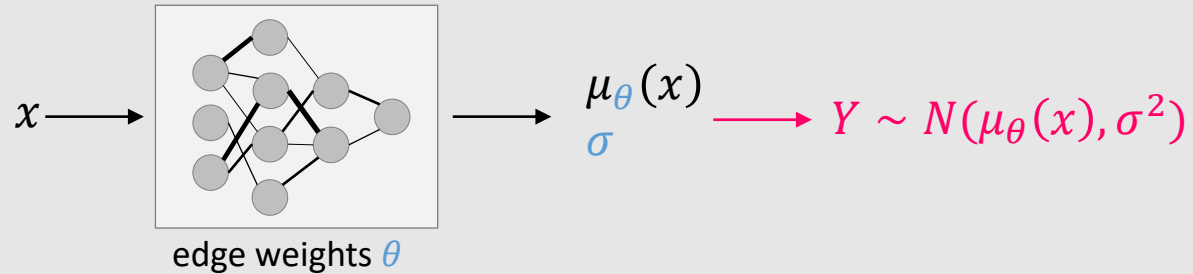
PROBABILITY MODELLER'S VIEW



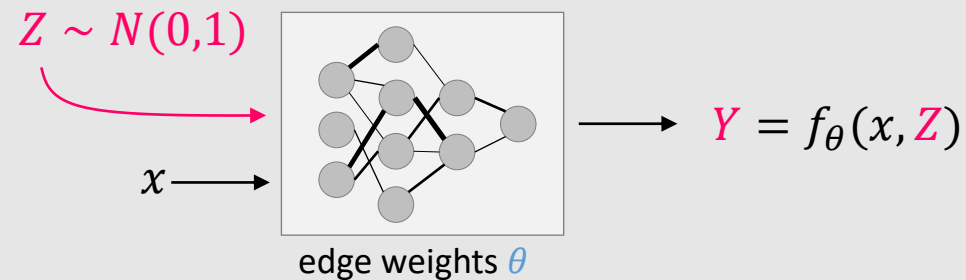
Goal:
fit the model by
maximum likelihood estimation

Q. How does a neural network output a random variable?

- ❖ Easy mode: have it output the parameters of a distribution

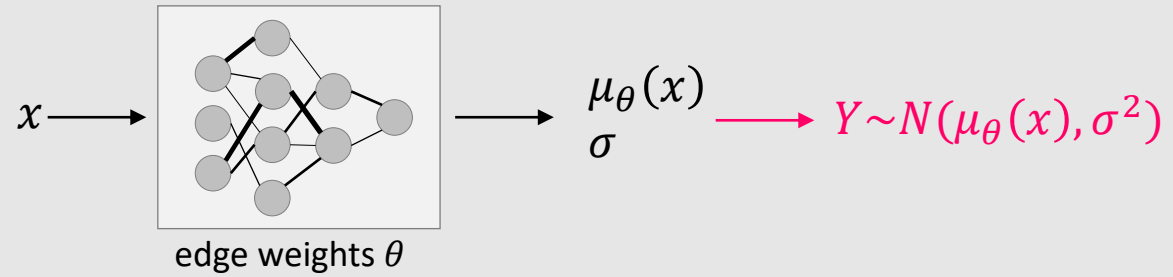


- ❖ Hard mode: give it some noise as input

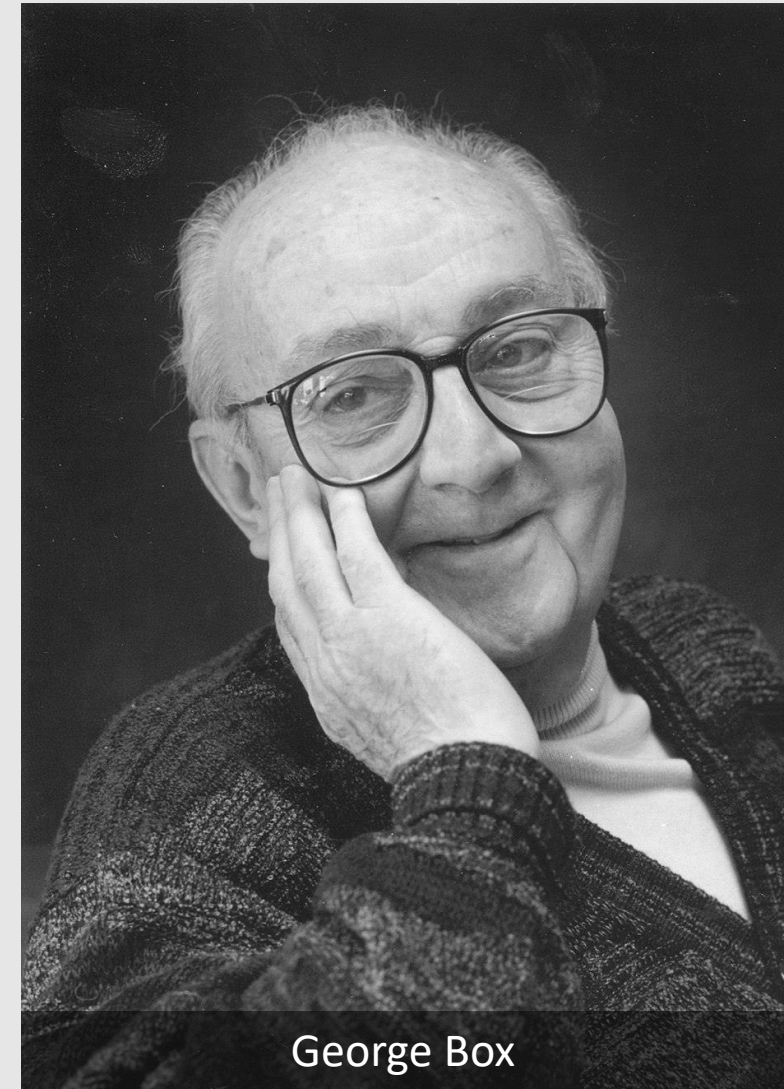


(We then have to compute the likelihood $\Pr_Y(y)$. This is what makes these easy or hard.)

Q. Why does the response have to be Gaussian?



- ❖ It doesn't!
- ❖ All models are wrong
- ❖ ... but some are useful

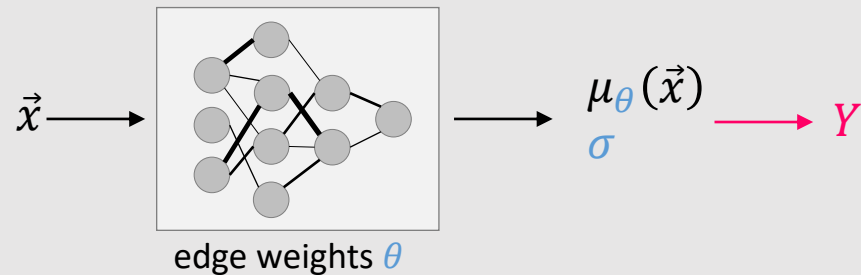


George Box

Q. How would we specify a probability model that uses several predictor variables?

$$Y \sim N(\mu_{\theta}(\vec{x}), \sigma^2)$$

where $\vec{x}$ is a vector of input variables



- ❖ Easy mode: when $\vec{x} \in \mathbb{R}^d$ for fixed dimension d
- ❖ Hard mode: when $\vec{x}$ is an arbitrary-length sequence, e.g. a text prompt

In this temperature-forecasting example,
there's not very much difference in practice
between the prediction mindset and the probability modelling view.

In other examples,
the prediction mindset runs into problems ...

Neural network classification

The MNIST database of handwritten images consists of records (x_i, y_i) where $x_i \in \mathbb{R}^{28 \times 28}$ is a greyscale image with 28×28 pixels, and $y_i \in \{0, \dots, 9\}$ is the digit.

We'd like to predict the digit, given an image. How might we learn to do this?

image	digit
	2
	1
	3
	1
	4

Data from <http://yann.lecun.com/exdb/mnist/>

ground truth:

Let y_i be the actual observed label in the dataset

minimize prediction loss:

pick θ to minimize

$$\sum_i L(y_i, \hat{y}_\theta(x_i))$$

where

$$L(\text{obs}, \text{pred}) = 1_{\text{obs} \neq \text{pred}}$$

PREDICTION MINDSET

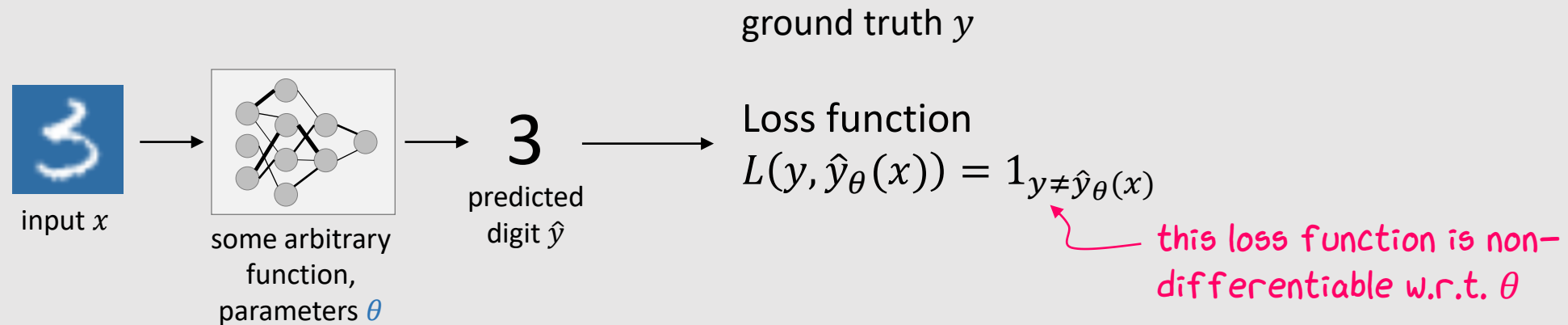
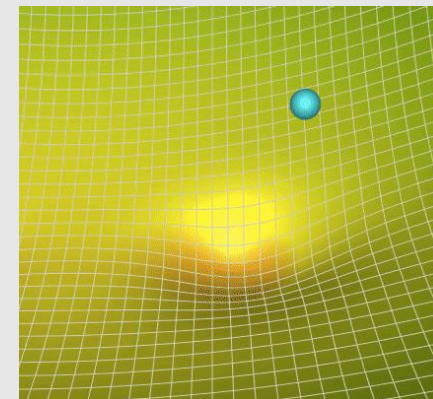


image x

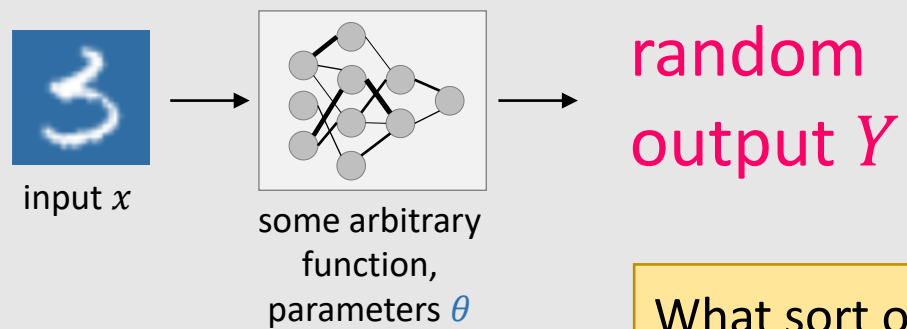


5
predicted
label
 $\hat{y}_\theta(x)$

Q. How would we train this neural network with gradient descent?



but gradient descent
requires the
landscape to be
differentiable



What sort of PROBABILITY MODEL might we use for the response Y ?

PREDICTION MINDSET



features



5
predicted
label $\hat{y}$

Goal:

maximize prediction accuracy

😞 but we have to fudge the loss
function to make it differentiable 🤪

PROBABILITY MODELLER'S VIEW



5 with prob 45%
3 with prob 41%
6 with prob 5%
...

random predicted label Y

Goal:

fit the model by
maximum likelihood estimation

Neural networks
are probability
models*
trained by
maximum likelihood estimation†

* Non-examinable

† Examinable. Very very examinable.