Statistical modeling: the two cultures

Leo Breiman

Statistical Science, 2001

There are two cultures in the use of statistical modeling to reach conclusions from data.

- One assumes that the data are generated by a given data model.
- The other uses algorithmic models and treats the data mechanism as unknown.

The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems.

In the mid-1980s two powerful new algorithms for fitting data became available: neural nets and decision trees. A new research community using these tools sprang up. Their goal was predictive accuracy. The community consisted of young computer scientists, physicists and engineers plus a few aging statisticians. They began using the new tools in working on complex prediction problems where it was obvious that data models were not applicable: speech recognition, image recognition, nonlinear time series prediction, handwriting recognition, prediction in financial markets.

TODAY

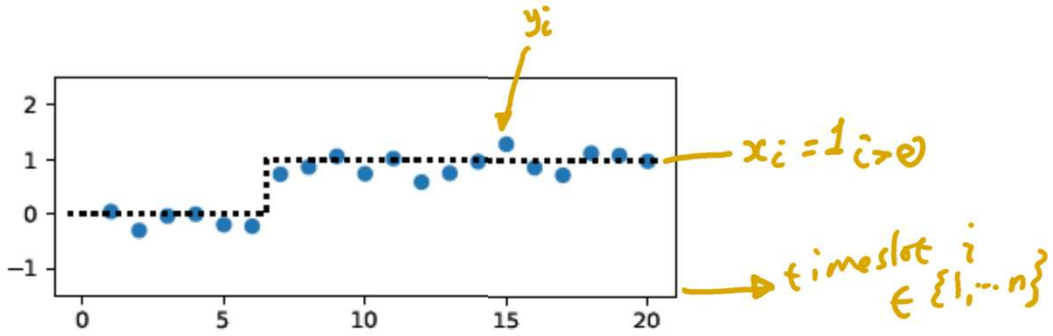
Mock exam question 1

Printouts are on the front bench, both sides.

(a) A 0/1 signal is being transmitted. The transmitted signal at timeslot $i \in \{1, \ldots, n\}$ 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 $\theta \in \{1, \ldots, n-1\}$ 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 $(y_1, \ldots, y_n)$, find an expression for the log likelihood, $\log \Pr(y_1, \ldots, y_n\,;\,\theta)$. 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
mle
pseudocode for mle

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

find expression for mle
give pseudocode for fit.

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

sec 1. specifying & fitting models

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

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

Observe that they turn *and* into *multiplication*:

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

In IA Discrete Maths [lecture 17] you used the indicator for a subset:

$$\chi_S(a) = 1_{a \in S}$$

(a) (i). "Find an expression for log likelihood"

$\log \Pr(\text{observed data}; \text{params})$ model: $Y_i \sim x_i + N(0, \varepsilon^2)$, $i \in \{1, \dots, n\}$

observed data: $y_1 \dots y_n$

params: θ unknown; ε known; $x_1 \dots x_n$ given; n given

LHS $\log \Pr(y_1 \dots y_n ; \theta)$

$= \sum_{i=1}^{n} \log \Pr(y_i ; \theta)$ modelling the Y_i as independent.

$\Pr_{Y_i}(y_i ; \theta)$

$\Pr_Y(y_i ; x_i)$

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

since $Y_i \sim x_i + N(0, \varepsilon^2) \sim N(x_i, \varepsilon^2)$ by linearity of the Normal.

SANITY CHECK.

Every variable on RHS should either

- appear on LHS — y_i, θ
- be a constant — ε, n
- be a dummy variable — i

so I'd better replace x_i by $1_{i>\theta}$

$= \sum_{i=1}^{n} \log \left\{ \frac{1}{\sqrt{2\pi\varepsilon^2}} e^{-(y_i - 1_{i>\theta})^2 / 2\varepsilon^2} \right\}$

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

(ii) We need to find $\max_{\theta}$ of the log likelihood we just derived.

```
def loglik(θ): return ...
scipy.optimize.fmin( lambda θ: -loglik(θ), init_guess)
```

silly! we should opt. over $\theta \in \{1, \ldots, n-1\}$.

```
vals = [ (loglik(θ), θ)  for θ ∈ {1, ..., n-1} ]
(_, θ̂) = max(vals)
```

(*b*) I have been monitoring average annual river levels for many years, and I have collected a dataset $(z_0, \ldots, z_n)$ 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]

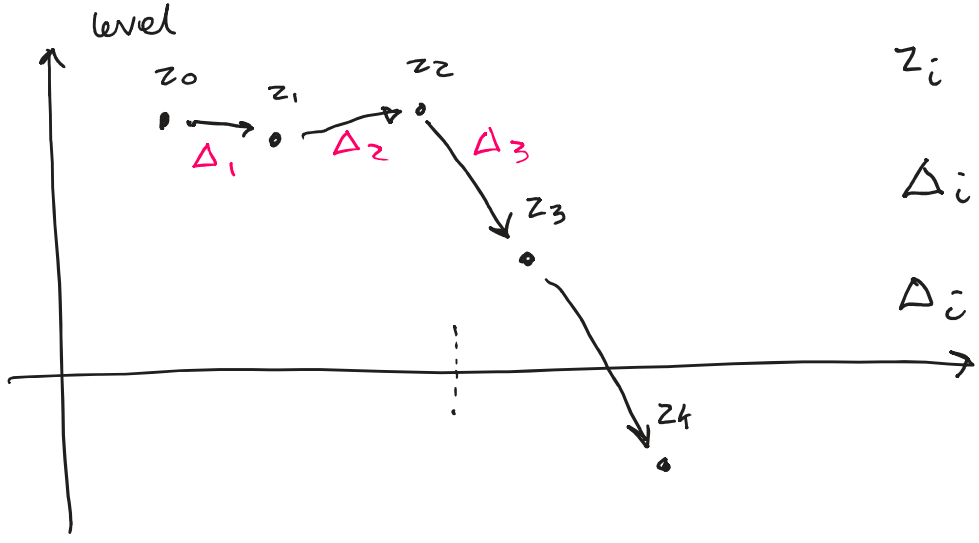
(*ii*) Explain how to fit your model. [5 marks]

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) (i)



$Z_i \approx Z_{i-1}$ + random variation.

$\Delta_i \sim N(0, \varepsilon^2)$ for $i \leq \theta$, before drought

$\Delta_i \sim N(-\mu, \varepsilon^2)$ for $i > \theta$, after drought

Model: $\Delta_i \sim N(-\mu 1_{i>\theta}, \varepsilon^2)$ where $\Delta_i = Z_i - Z_{i-1}$

μ, θ, ε unknown.

(ii) $\log \Pr(\delta_1, \dots \delta_n ; \mu, \theta, \varepsilon)$

$$= -\frac{n}{2} \log(2\pi\varepsilon^2) - \frac{1}{2\varepsilon^2} \sum_{i=1}^{n} (\delta_i + \mu \mathbb{1}_{i>\theta})^2$$

we want to estimate θ.

we want $\max_{\theta,\mu,\varepsilon} \log \Pr(\dots)$

$$= \max_{\theta} \left\{ \max_{\mu,\varepsilon} \log \Pr(\dots ; \theta, \mu, \varepsilon) \right\}.$$

do this bit by iteration

do this bit with scipy

THEOREM

$$\max_{x,y} f(x,y) = \max_{x}\left\{\max_{y} f(x,y)\right\}$$

Proof: By cartography.

$\max_{x,y} f(x,y)$

scans the entire area looking for the highest point

$\max_{y} f(x,y)$

scans a line of fixed x, looking for the highest point

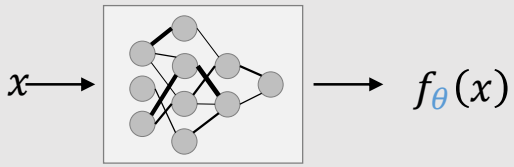
$\max_{x}\left\{\max_{y} f(x,y)\right\}$

look at all the vertical slices, and for each find the highest point
then pick the highest out of all these

Neural networks are probability models*

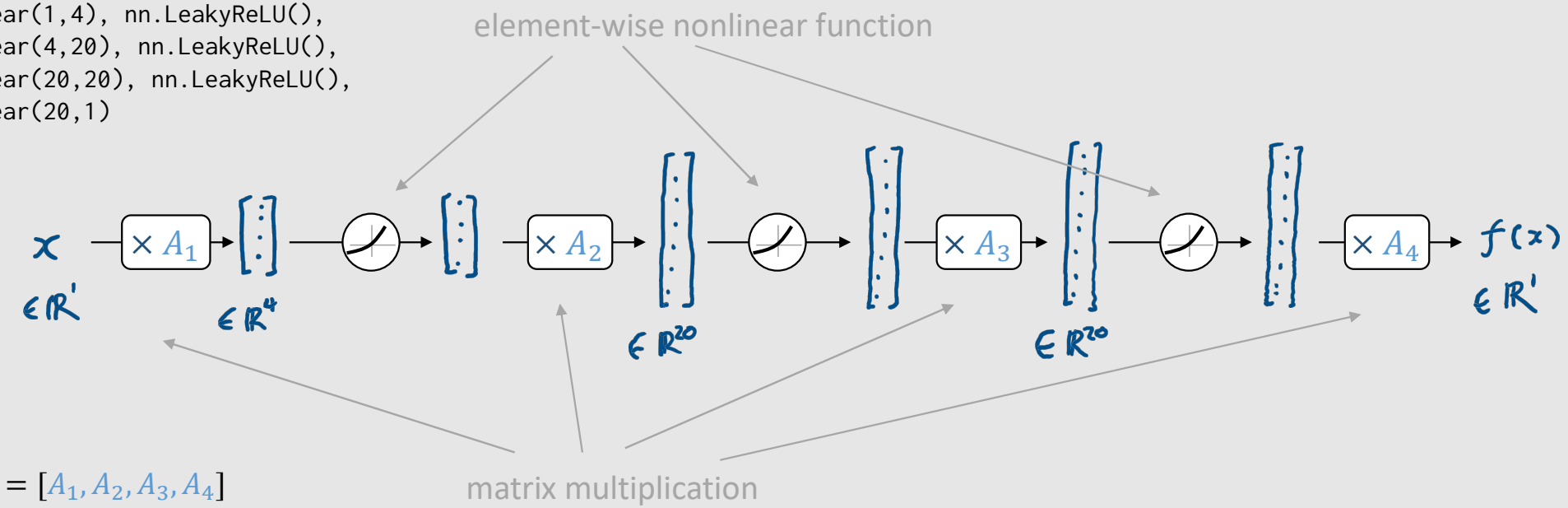
* non-examinable

A neural network is a deterministic parametric function, typically consisting of matrix multiplication and nonlinear functions.



edge weights θ

```
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)
      )
```

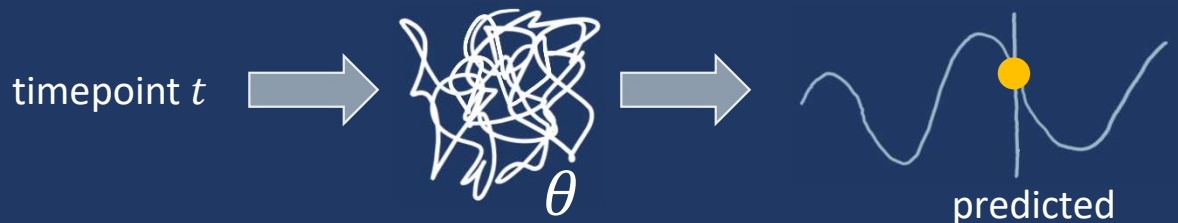


parameters $\theta = [A_1, A_2, A_3, A_4]$

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.



ground-truth labels:
Let y_i be the actual observed temperature at time t_i

it's our job as modellers to find θ so as to minimize prediction error, e.g.

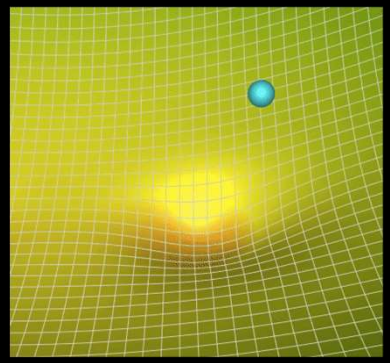
pick θ to minimize

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

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 is

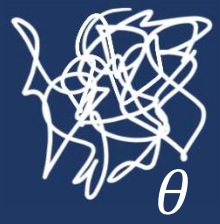
- it's a parametric function
  $$t \mapsto \hat{y}_\theta(t)$$
- it's differentiable with respect to its parameters θ

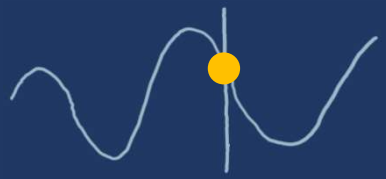
The loss function L is arbitrary, but it had better be differentiable with respect to *pred*.

This course teaches a different way to think of modelling …

PREDICTION MINDSET

timepoint t



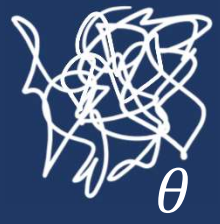


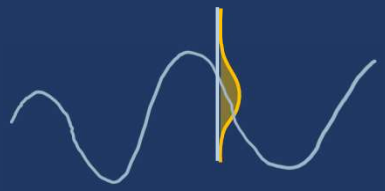
predicted temperature $\hat{y}$ at timepoint t

Goal:
minimize prediction error

PROBABILITY MODELLER'S VIEW

timepoint t



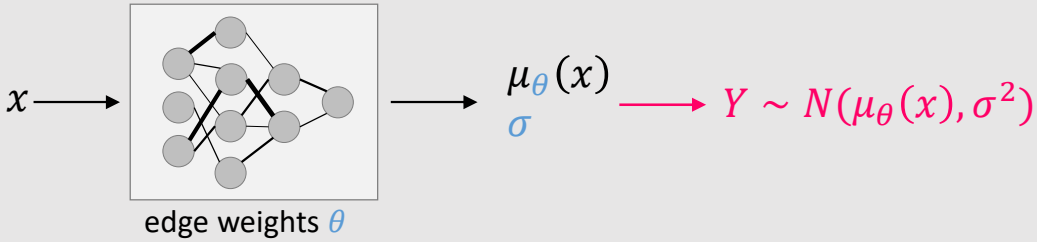


random variable for temperature Y at timepoint t

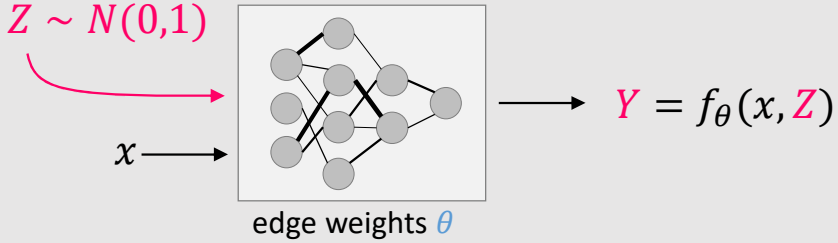
Goal:
maximize the log likelihood of the dataset

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

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

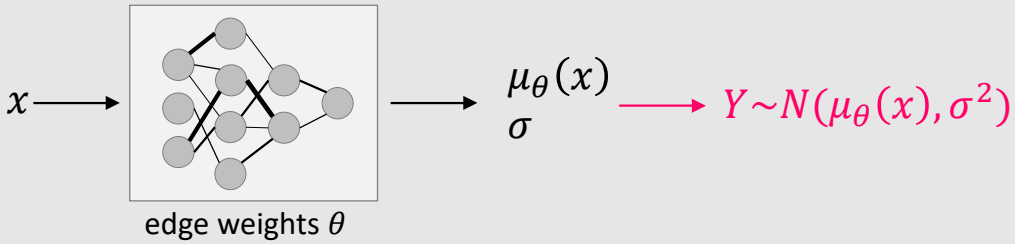


- Hard mode: give it some noise as input

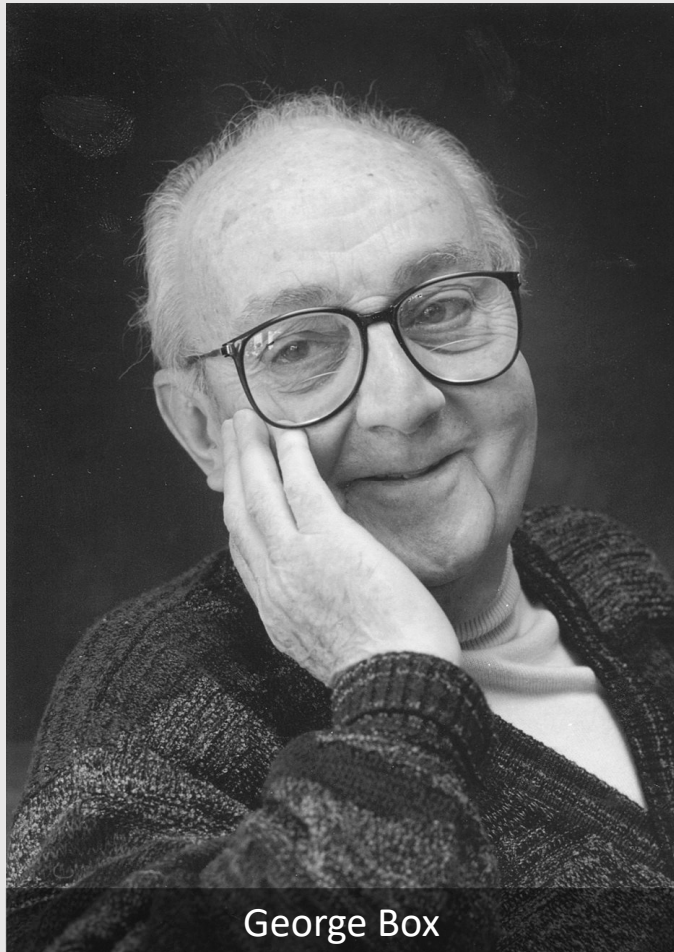


(We then have to compute the likelihood $\Pr_Y(y)$. This is what makes these easy or hard.)

Q. Why does the response have to be Gaussian?



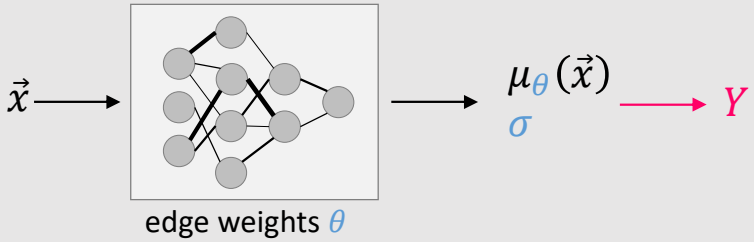
- It doesn't!
- All models are wrong
- … but some are useful


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 tuple of predictor variables, $\vec{x} = (x_1, \ldots, x_d)$,
and $\mu_\theta : \mathbb{R}^d \to \mathbb{R}$ is a differentiable function



- 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 $x_i \in \mathbb{R}^{28\times28}$ 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/

image x

ground truth:
Let y_i be the actual observed label in the dataset

it's our job as modellers to find θ so as to maximize prediction accuracy, i.e.

pick θ to minimize

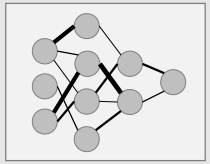
$$\sum_i L(y_i, \hat{y}_\theta(x_i))$$

where

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

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

input x



some arbitrary function, parameters θ

3

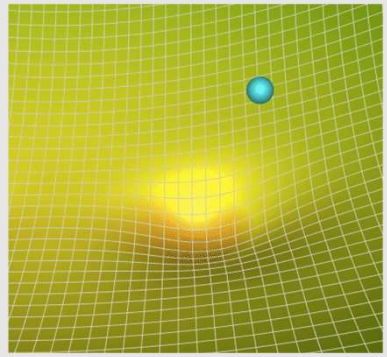
predicted digit $\hat{y}$

ground truth y

Loss function

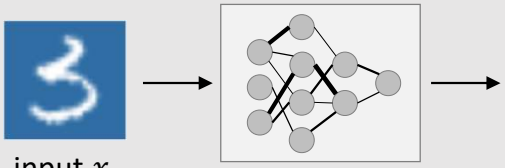
$L(y, \hat{y}) = 1_{y \neq \hat{y}}$

this loss function is non-differentiable



but gradient descent requires the landscape to be differentiable

PROBABILITY MODELLER'S VIEW



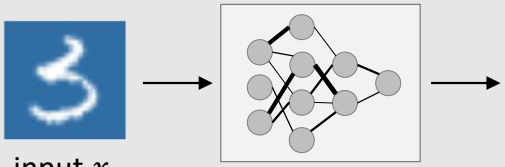
input x

some arbitrary function, parameters θ

random output Y

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

PROBABILITY MODELLER'S VIEW



input x

some arbitrary function, parameters θ

$\vec{p}(x) = [p_0, \ldots, p_9]$ $\longrightarrow$ $Y \sim \mathrm{Cat}(\vec{p}(x))$

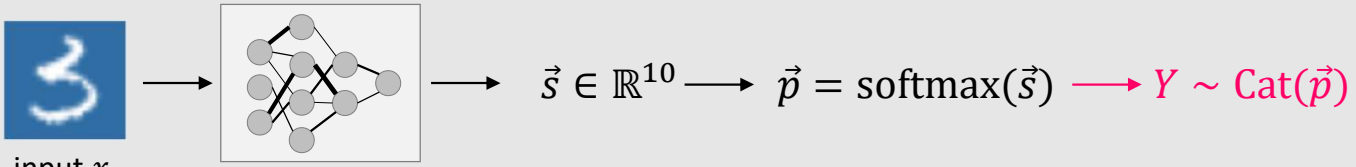
How can we make sure that $\vec{p}$ is a valid probability vector?

(We need $p_i \in [0,1]$ for each i, and $\Sigma_i p_i = 1$.)

PROBABILITY MODELLER'S VIEW

Softmax function:

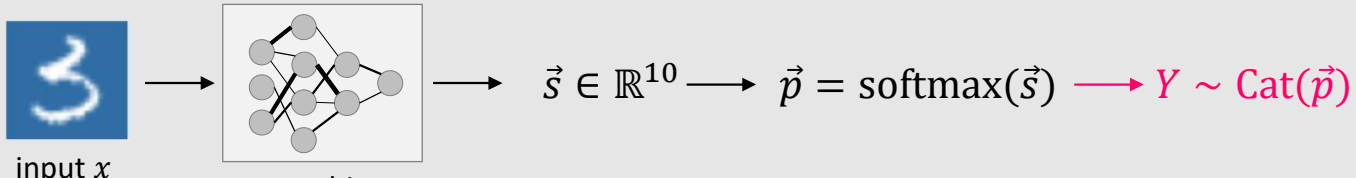
$$p_k = \frac{e^{s_k}}{\Sigma_{\ell=0}^{9} e^{s_\ell}}$$



How do we fit this model to the data?

PROBABILITY MODELLER'S VIEW

0. Probability model for a single datapoint:



1. Likelihood of a ground-truth datapoint y:

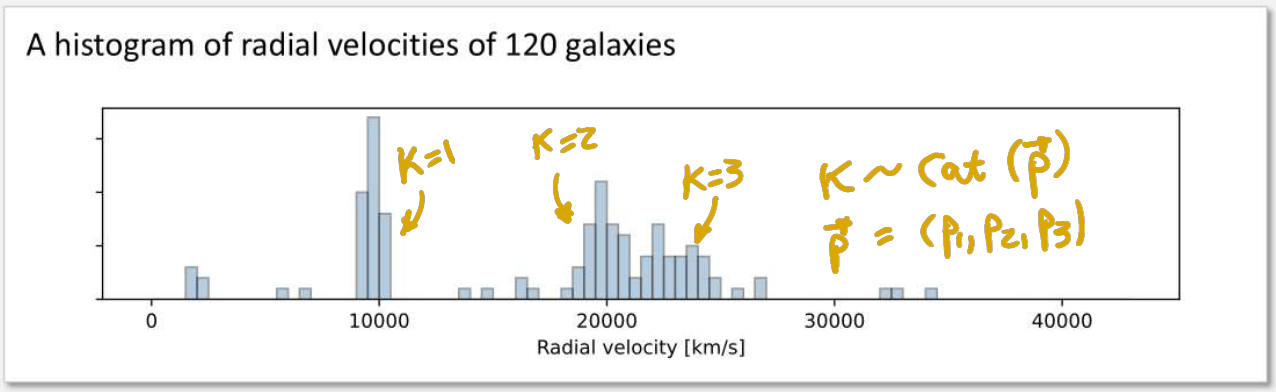
$$\Pr_Y(y; x, \theta) =$$

2. Log likelihood of the whole dataset:

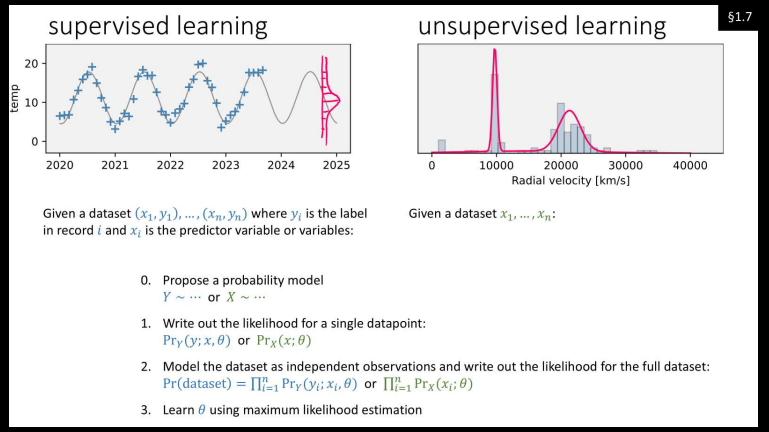
$$\log \Pr(y_1, \ldots, y_n) =$$ … we end up with the famous "softmax cross-entropy loss function"

3. Find the maximum likelihood estimator $\hat{\theta}$ using gradient descent

Categorical random variable (lec3)



Supervised learning (lec3)



Softmax reparameterization trick (lec2)

Exercise 1.4.2 (Constraints / softmax transformation)
Find the maximum of

$$f(p_1, p_2, p_3) = 0.2 \log p_1 + 0.5 \log p_2 + 0.3 \log p_3$$

over $p_1, p_2, p_3 \in (0,1)$ such that $p_1 + p_2 + p_3 = 1$.

fmin is not able to handle constraints.
Coming trick: instead of maximizing over $p_1, p_2, p_3 \in (0,1)$ s.t. $p_1 + p_2 + p_3 = 1$
let's maximize over $(s_1, s_2, s_3) \in \mathbb{R}^3$, and set $p_i = \frac{e^{s_i}}{e^{s_1} + e^{s_2} + e^{s_3}}$.

PREDICTION MINDSET

features

5

predicted label $\hat{y}$

Goal:

pick θ to maximize prediction accuracy

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

PROBABILITY MODELLER'S VIEW

$$\begin{cases} 5 \text{ with prob } 45\% \\ 3 \text{ with prob } 41\% \\ 6 \text{ with prob } \ 5\% \\ \dots \end{cases}$$

random predicted label Y

Goal:

fit my Y_θ model to the dataset by maximizing log likelihood

Statistical modeling: the two cultures

Leo Breiman

Statistical Science, 2001

There are two cultures in the use of statistical modeling to reach conclusions from data.

- One assumes that the data are generated by a given data model.
- The other uses algorithmic models and treats the data mechanism as unknown.

The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems.

In the mid-1980s two powerful new algorithms for fitting data became available: neural nets and decision trees. A new research community using these tools sprang up. Their goal was predictive accuracy. The community consisted of young computer scientists, physicists and engineers plus a few aging statisticians. They began using the new tools in working on complex prediction problems where it was obvious that data models were not applicable: speech recognition, image recognition, nonlinear time series prediction, handwriting recognition, prediction in financial markets.

Neural networks can be viewed as probability models for data.

- ❖ This view solves problems with the “prediction mindset”
- ❖ It unifies prediction, description e.g. clustering, and generative AI
- ❖ It opens the door to deeper thinking about generalizability

- Office hours in the cafe area
  1–1.30pm today