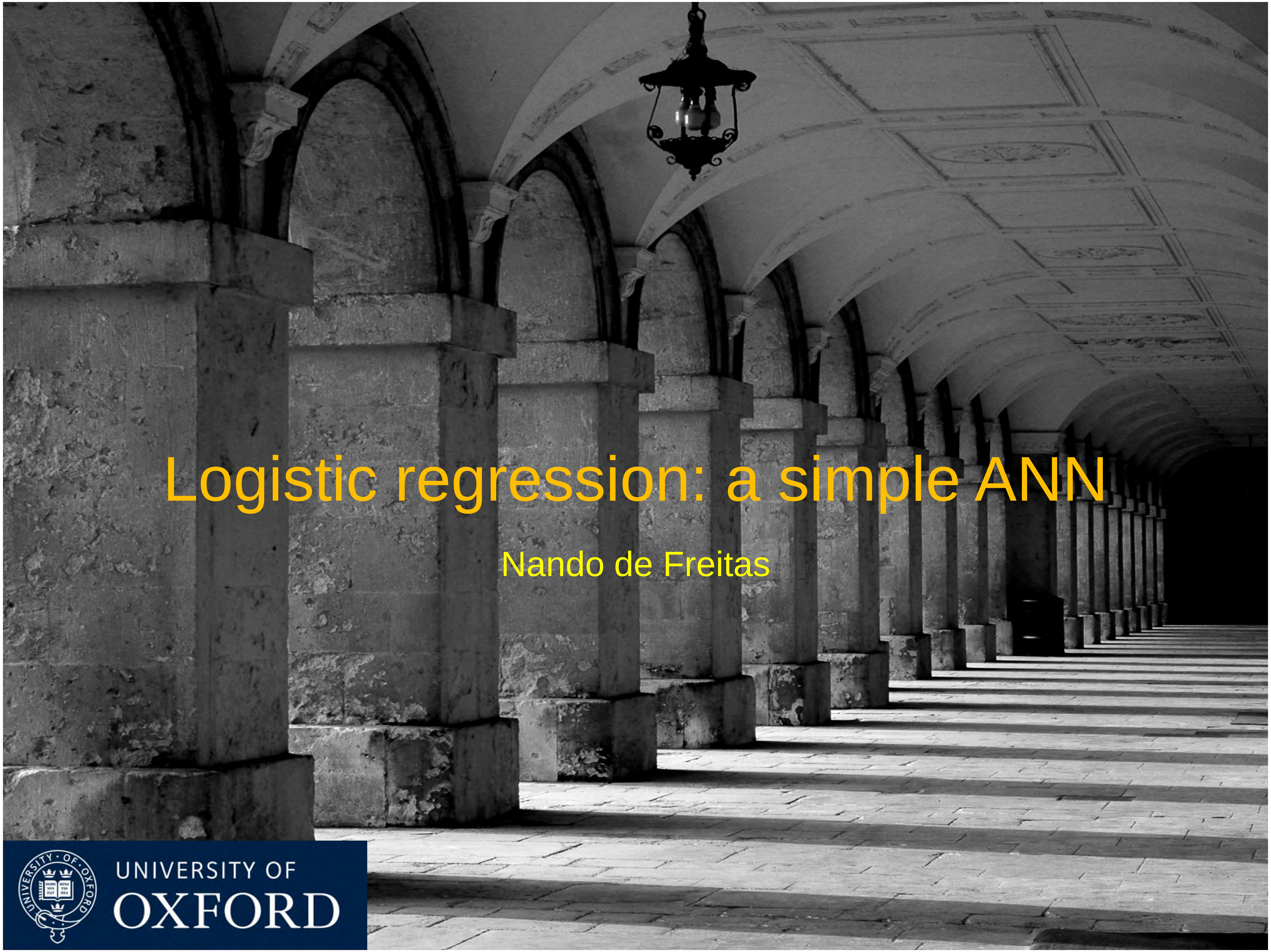


Logistic regression: a simple ANN

Nando de Freitas

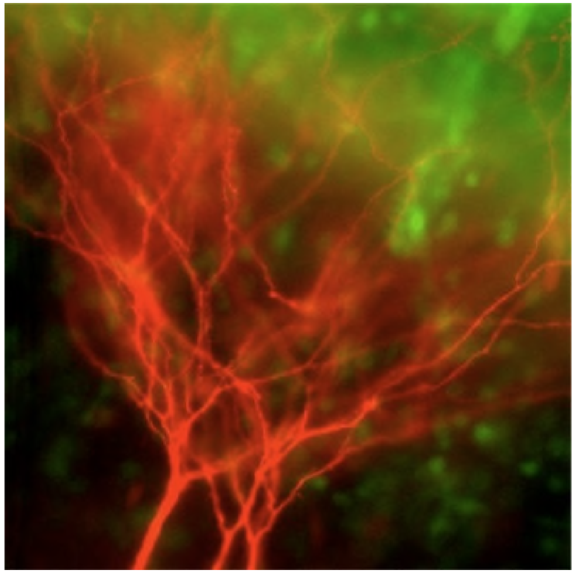
UNIVERSITY OF OXFORD

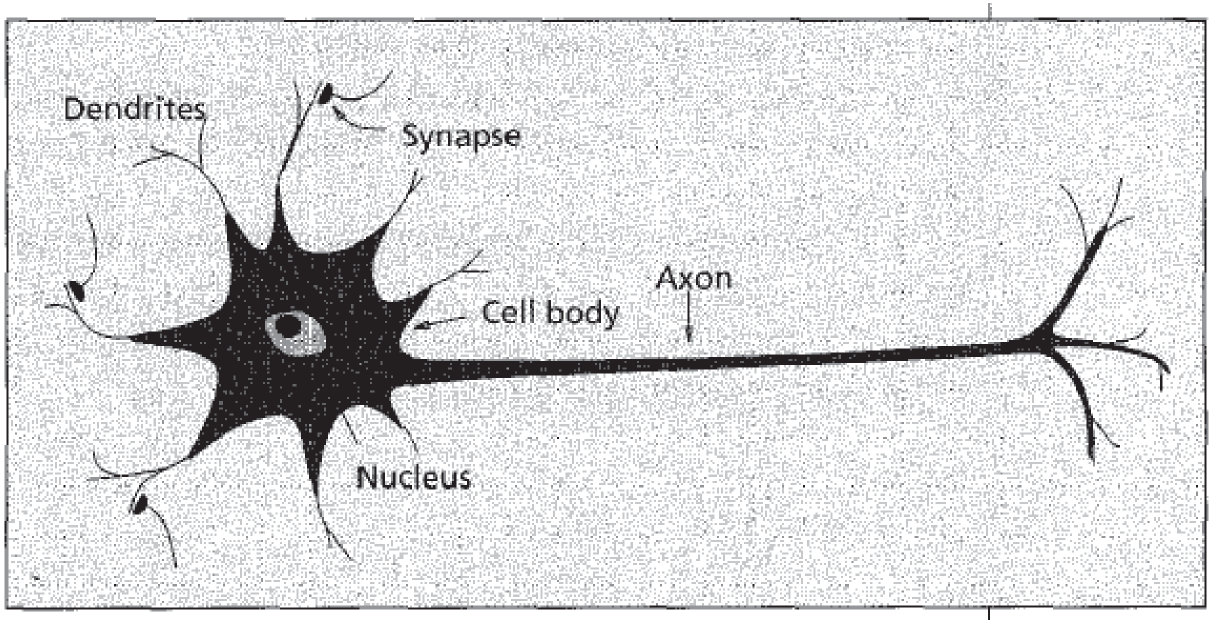
Outline of the lecture

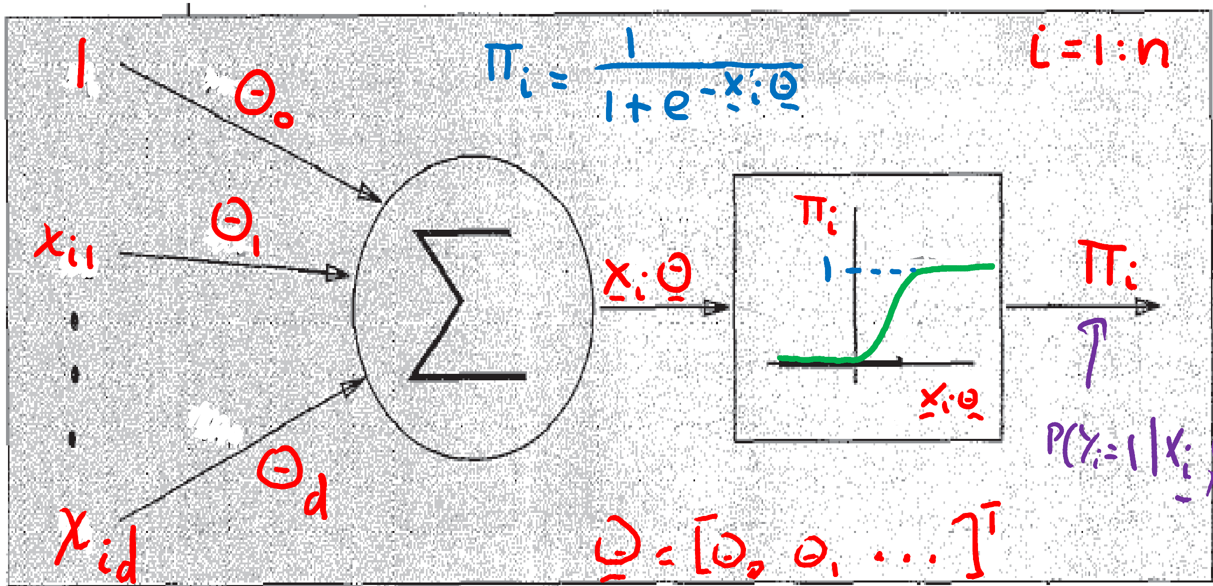
This lecture describes the construction of binary classifiers using a technique called Logistic Regression. The objective is for you to learn:

- ❑ How to apply logistic regression to discriminate between two classes.
- ❑ How to formulate the logistic regression likelihood.
- ❑ How to derive the gradient and Hessian of logistic regression.
- ❑ How to incorporate the gradient vector and Hessian matrix into Newton's optimization algorithm so as to come up with an algorithm for logistic regression, which we call IRLS.
- ❑ How to do logistic regression with the softmax link.

McCulloch-Pitts model of a neuron



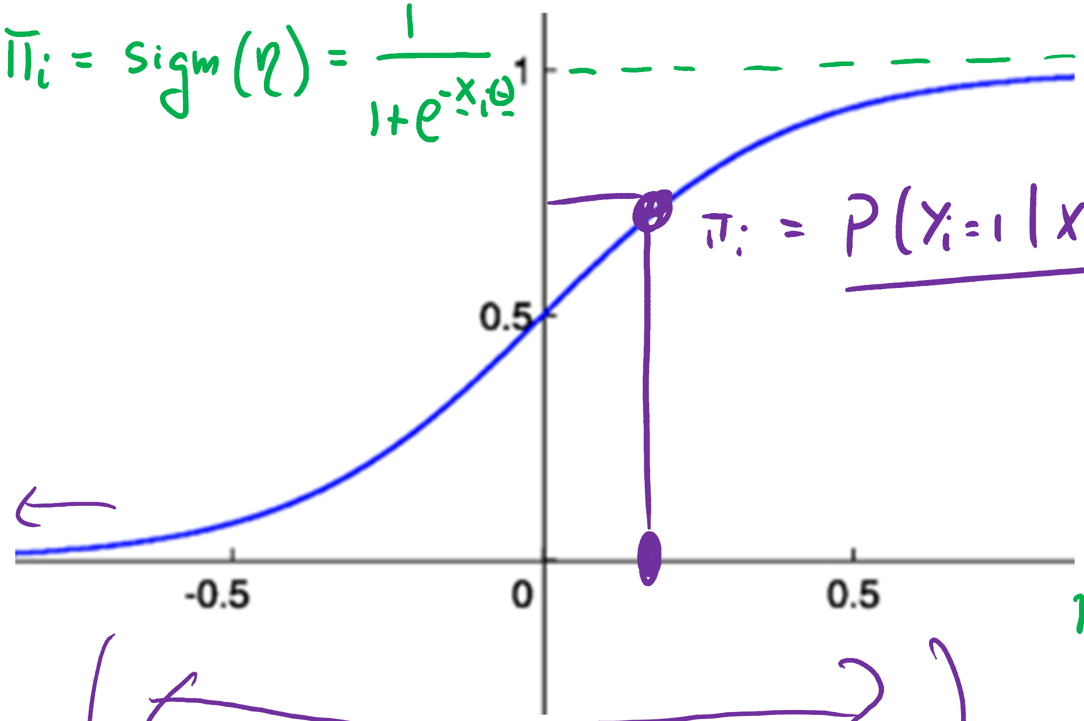




Sigmoid function

sigm(η) refers to the **sigmoid** function, also known as the **logistic** or **logit** function:

$$\text{sigm}(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^{\eta}}{e^{\eta}+1}$$



Linear separating hyper-plane

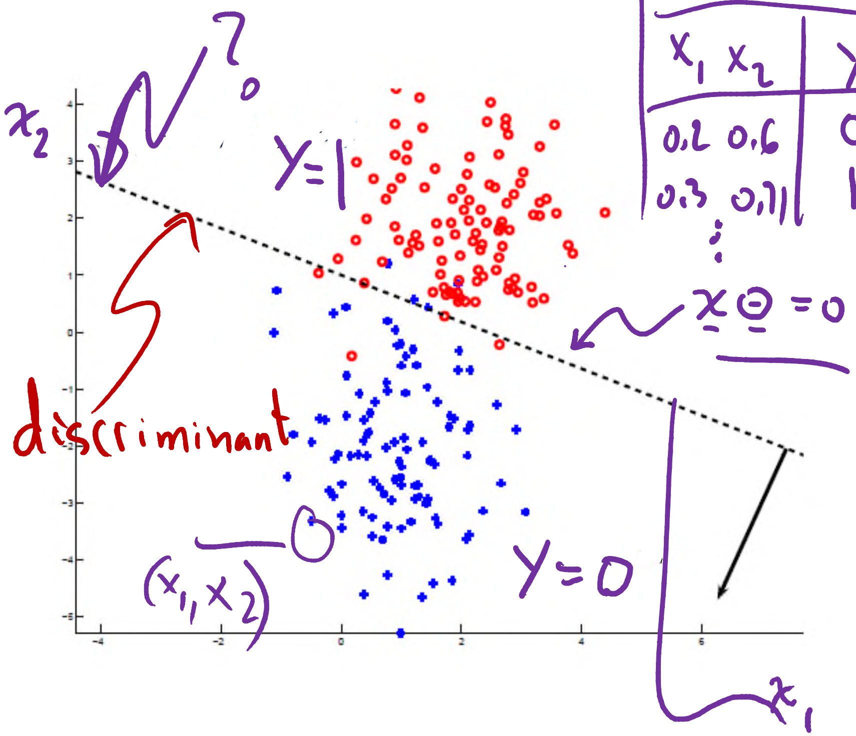
$$P(y_i=1|\underline{x}_i,\underline{\theta}) = \text{sigm}(\underline{x}_i\underline{\theta}) \overset{\text{When}}{=} \frac{1}{2} \quad \overset{\longrightarrow}{\underset{\longleftarrow}{\text{when}}} \quad \underline{x}_i\underline{\theta} \overset{\text{when}}{=} 0$$

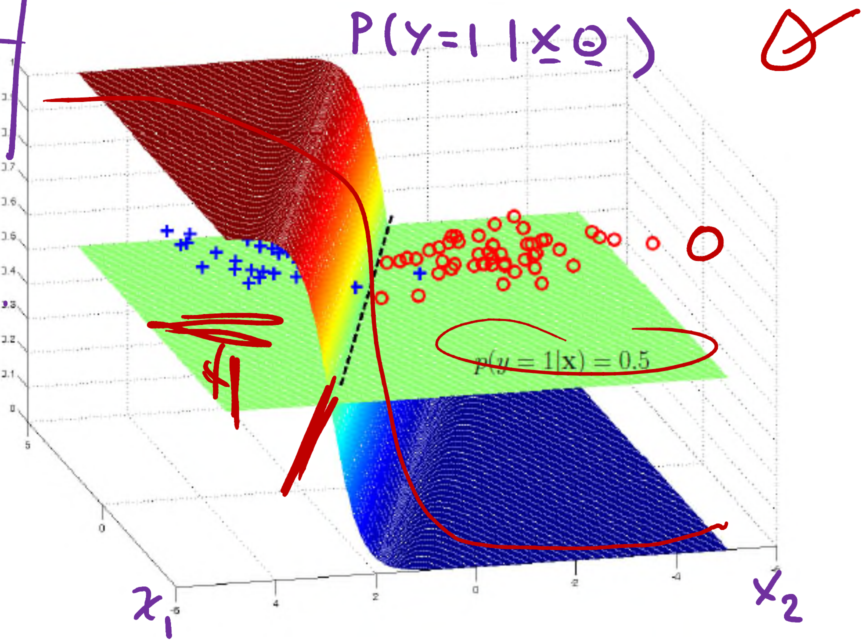
EQUATION OF A PLANE.

i.e. $\frac{1}{1+e^{-0}} = \frac{1}{2}$

DATA

x_1	x_2	y
0.2	0.6	0
0.3	0.11	1
⋮		



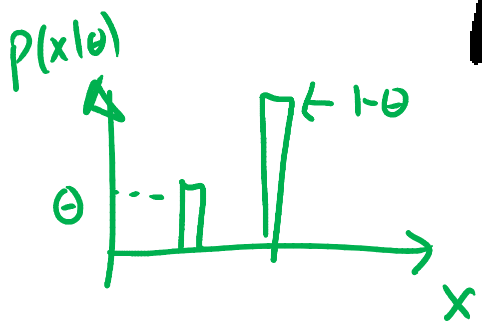


[Greg Shakhnarovich]

Bernoulli: a model for coins

A ***Bernoulli random variable r.v.*** ***X*** *takes values in {0,1}*

$$p(x|\theta) = \begin{cases} \theta & \text{if } x=1 \\ 1-\theta & \text{if } x=0 \end{cases}$$



Where θ **2** *(0,1). We can write this probability more succinctly as follows:*

$$P(x|\theta) = \theta^{x}(1-\theta)^{1-x} = \begin{cases} \theta & x=1 \\ 1-\theta & x=0 \end{cases}$$

Ber(x|θ)

Entropy

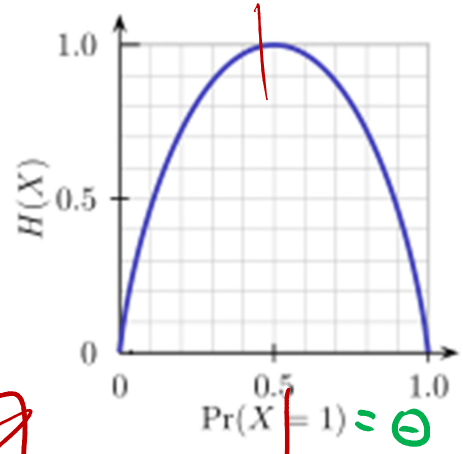
In information theory, entropy ***H*** *is a measure of the uncertainty associated with a random variable. It is defined as:*

$$H(X) = -\sum_{x} p(x|\theta) \log p(x|\theta)$$

<u>*Example:*</u> *For a Bernoulli variable* ***X****, the entropy is:*

$$H(x) = -\sum_{x=0}^{1} \theta^{x}(1-\theta)^{1-x} \log\left[\theta^{x}(1-\theta)^{1-x}\right]$$

$$= -\left[(1-\theta)\log(1-\theta) + \theta\log\theta\right]$$



Logistic regression

The logistic regression model specifies the probability of a binary output $y_i \in \{0, 1\}$ given the input $\mathbf{x}_i$ as follows:

$p(y_i | \mathbf{x}_i, \boldsymbol{\theta})$

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) &= \prod_{i=1}^{n} \text{Ber}(y_i|\text{sigm}(\mathbf{x}_i\boldsymbol{\theta})) \\
&= \prod_{i=1}^{n} \left[\frac{1}{1+e^{-\mathbf{x}_i\boldsymbol{\theta}}}\right]^{y_i} \left[1 - \frac{1}{1+e^{-\mathbf{x}_i\boldsymbol{\theta}}}\right]^{1-y_i}
\end{aligned}
$$

π_i π_i

where $\mathbf{x}_i\boldsymbol{\theta} = \theta_0 + \sum_{j=1}^{d} \theta_j x_{ij}$

$\pi_i = P(y_i = 1 | x_i \, \theta)$

$1 - \pi_i = P(y_i = 0 | x_i \, \theta)$

$$C(\theta) = -\log P(y | x, \theta)$$

$$= -\sum_{i=1}^{n} y_i \log \pi_i + (1-y_i) \log (1-\pi_i)$$

cross-entropy

Gradient and Hessian of binary logistic regression

The gradient and Hessian of the negative loglikelihood, $J(\boldsymbol{\theta}) = -\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$, are given by:

$$
\begin{aligned}
\mathbf{g}(\mathbf{w}) &= \frac{d}{d\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \mathbf{x}_i^T (\pi_i - y_i) = \mathbf{X}^T (\boldsymbol{\pi} - \mathbf{y}) \\
\mathbf{H} &= \frac{d}{d\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\theta})^T = \sum_i \pi_i (1 - \pi_i) \mathbf{x}_i \mathbf{x}_i^T = \mathbf{X}^T \mathrm{diag}(\pi_i (1 - \pi_i)) \mathbf{X}
\end{aligned}
$$

where $\pi_i = \mathrm{sigm}(\mathbf{x}_i \boldsymbol{\theta})$

One can show that $\mathbf{H}$ is positive definite; hence the NLL is **convex** and has a unique global minimum.

To find this minimum, we turn to batch optimization.

Iteratively reweighted least squares (IRLS)

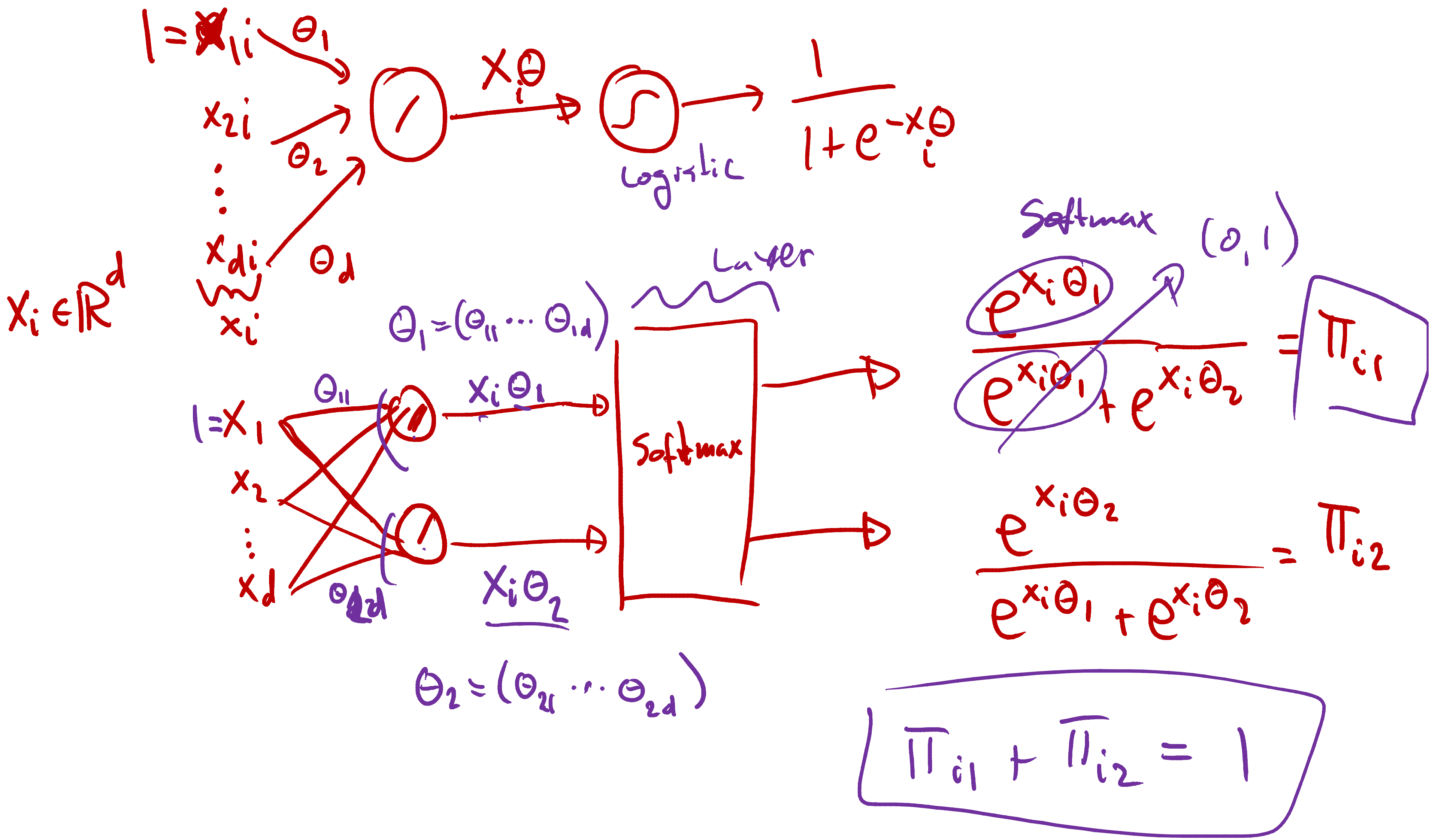
For binary logistic regression, recall that the gradient and Hessian of the negative log-likelihood are given by

$$\begin{aligned}
\mathbf{g}_k &= \mathbf{X}^T(\boldsymbol{\pi}_k - \mathbf{y}) \\
\mathbf{H}_k &= \mathbf{X}^T\mathbf{S}_k\mathbf{X} \\
\mathbf{S}_k &:= \mathrm{diag}(\pi_{1k}(1-\pi_{1k}), \ldots, \pi_{nk}(1-\pi_{nk})) \\
\pi_{ik} &= \mathrm{sigm}(\mathbf{x}_i\boldsymbol{\theta}_k)
\end{aligned}$$

The Newton update at iteration $k+1$ for this model is as follows (using $\eta_k = 1$, since the Hessian is exact):

$$\begin{aligned}
\boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k - \mathbf{H}^{-1}\mathbf{g}_k \\
&= \boldsymbol{\theta}_k + (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y} - \boldsymbol{\pi}_k) \\
&= (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}\left[(\mathbf{X}^T\mathbf{S}_k\mathbf{X})\boldsymbol{\theta}_k + \mathbf{X}^T(\mathbf{y} - \boldsymbol{\pi}_k)\right] \\
&= (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}\mathbf{X}^T\left[\mathbf{S}_k\mathbf{X}\boldsymbol{\theta}_k + \mathbf{y} - \boldsymbol{\pi}_k\right]
\end{aligned}$$

Softmax formulation



Likelihood function

INDICATOR:

$$\mathbb{I}_c(y_i) = \begin{cases} 1 & \text{if } y_i = c \\ 0 & \text{otherwise} \end{cases}$$

$$P(y \mid x, \theta) = \prod_{i=1}^{n} \pi_{i1}^{\mathbb{I}_0(y_i)} \, \pi_{i2}^{\mathbb{I}_1(y_i)}$$

$$P(y_i \mid x_i, \theta) = \begin{cases} \pi_{i1} = \dfrac{e^{x_i \theta_1}}{e^{x_i \theta_1} + e^{x_i \theta_2}} & \text{if } y_i = 0 \\[2ex] \pi_{i2} = \dfrac{e^{x_i \theta_2}}{e^{x_i \theta_1} + e^{x_i \theta_2}} & \text{if } y_i = 1 \end{cases}$$

Negative log-likelihood criterion

$$P(y \mid x, \theta) = \prod_{i=1}^{n} \pi_{i1}^{\mathbb{I}_0(y_i)} \, \pi_{i2}^{\mathbb{I}_1(y_i)}$$

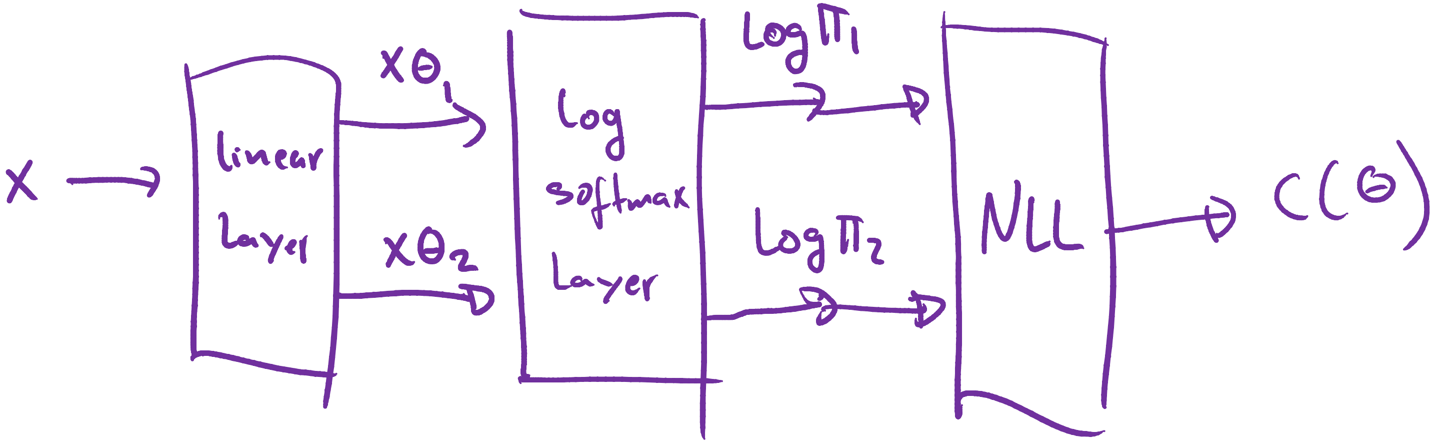
$$P(y_i \mid x_i, \theta) = \begin{cases} \pi_{i1} = \dfrac{e^{x_i\theta_1}}{e^{x_i\theta_1} + e^{x_i\theta_2}} & \text{if } y_i = 0 \\ \pi_{i2} = \dfrac{e^{x_i\theta_2}}{e^{x_i\theta_1} + e^{x_i\theta_2}} & \text{if } y_i = 1 \end{cases}$$

$$C(\theta) = -\log P(y \mid x, \theta) = -\sum_{i=1}^{n} \mathbb{I}_0(y_i) \underbrace{\log \pi_{i1}}_{\text{Logsoftmax}} + \mathbb{I}_1(y_i) \log \pi_{i2}$$

Logsoftmax

LSf

Neural network representation of loss



Manual gradient computation

$$\frac{\partial C(\theta)}{\partial \theta_2} = \frac{\partial}{\partial \theta_2}\left(- \sum_{i=1}^{n} I_0(y_i) \log \pi_{i1} + I_1(y_i) \log \pi_{i2} \right)$$

$$= - \sum_{i=1}^{n} \left(I_0(y_i) \frac{\partial}{\partial \theta_2} \log \pi_1 + I_1(y_i) \frac{\partial}{\partial \theta_2} \log \pi_{i2} \right)$$

$$\frac{\partial}{\partial \theta_2} \left(\log \pi_1 \right) = \frac{\partial}{\partial \theta_2} \log \left(\frac{e^{x_i \theta_1}}{e^{x_i \theta_1} + e^{x_i \theta_2}} \right) = \frac{\partial}{\partial \theta_2} \left(\cancelto{0}{x_i \theta_1} - \log \left(e^{x_i \theta_1} + e^{x_i \theta_2} \right) \right)$$

$$= 0 - \frac{x_i e^{x_i \theta_2}}{e^{x_i \theta_1} + e^{x_i \theta_2}} = - x_i \pi_{i2}$$

$$\partial / \partial \theta_2 \log \pi_2 = x_i \left(1 - \pi_{i2} \right)$$

Manual gradient computation

$y_i \in \{0,1\}$ $\pi_{i2} = P(y_i = 1 \mid x_i, \theta) = \pi_i$

$$\frac{\partial c(\theta)}{\partial \theta_2} = -\sum_{i=1}^{n} \mathbb{I}_0(y_i)(-x_i \pi_{i2}) + \mathbb{I}_1(y_i)\, x_i (1 - \pi_{i2})$$

Check

$$= -\sum_{i=1}^{n} (1-y_i)(-x_i)\pi_{i2} + y_i x_i (1 - \pi_{i2})$$

$$= -\sum_{i=1}^{n} -x_i \pi_{i2} + y_i x_i \pi_{i2} + y_i x_i - y_i x_i \pi_{i2}$$

$$= -\sum_{i=1}^{n} x_i (y_i - \pi_{i2})$$

Next lecture

In the next lecture, we develop an automatic layer-wise way of computing all the necessary derivatives known as back-propagation.

This is the approach used in Torch. We will review the torch nn class.