

# Logistic regression: a simple ANN

Nando de Freitas



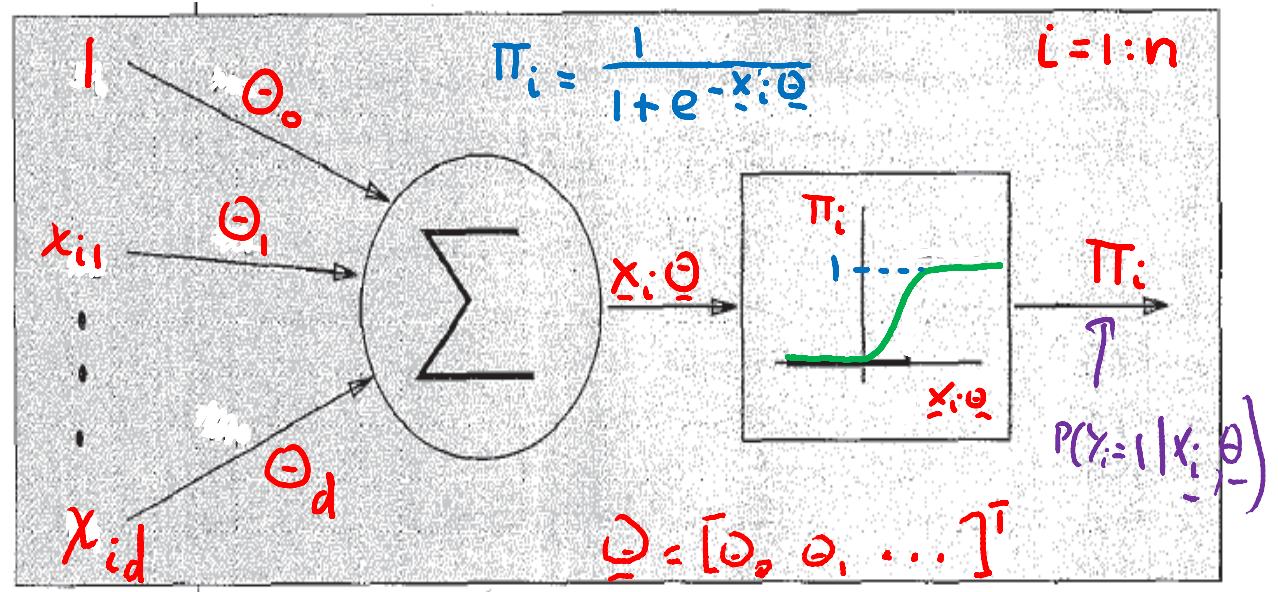
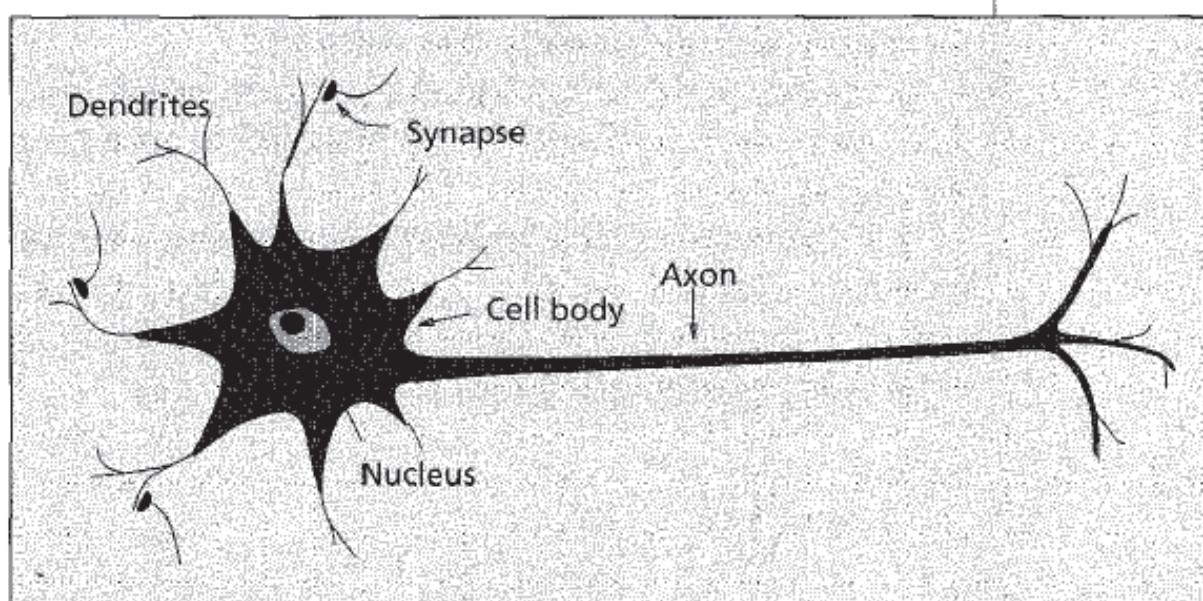
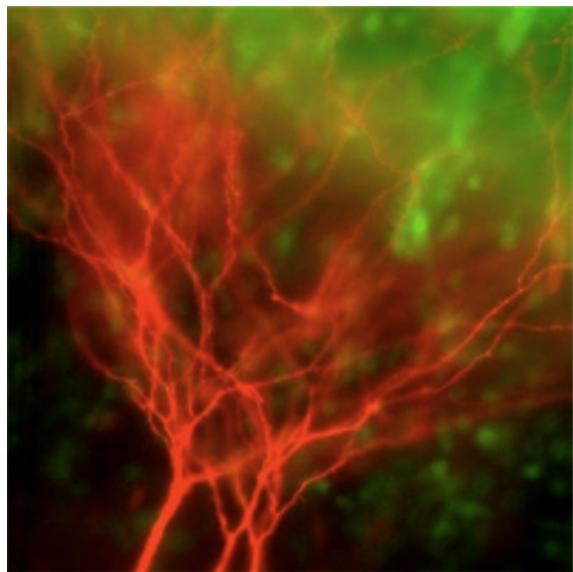
UNIVERSITY OF  
**OXFORD**

# 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.

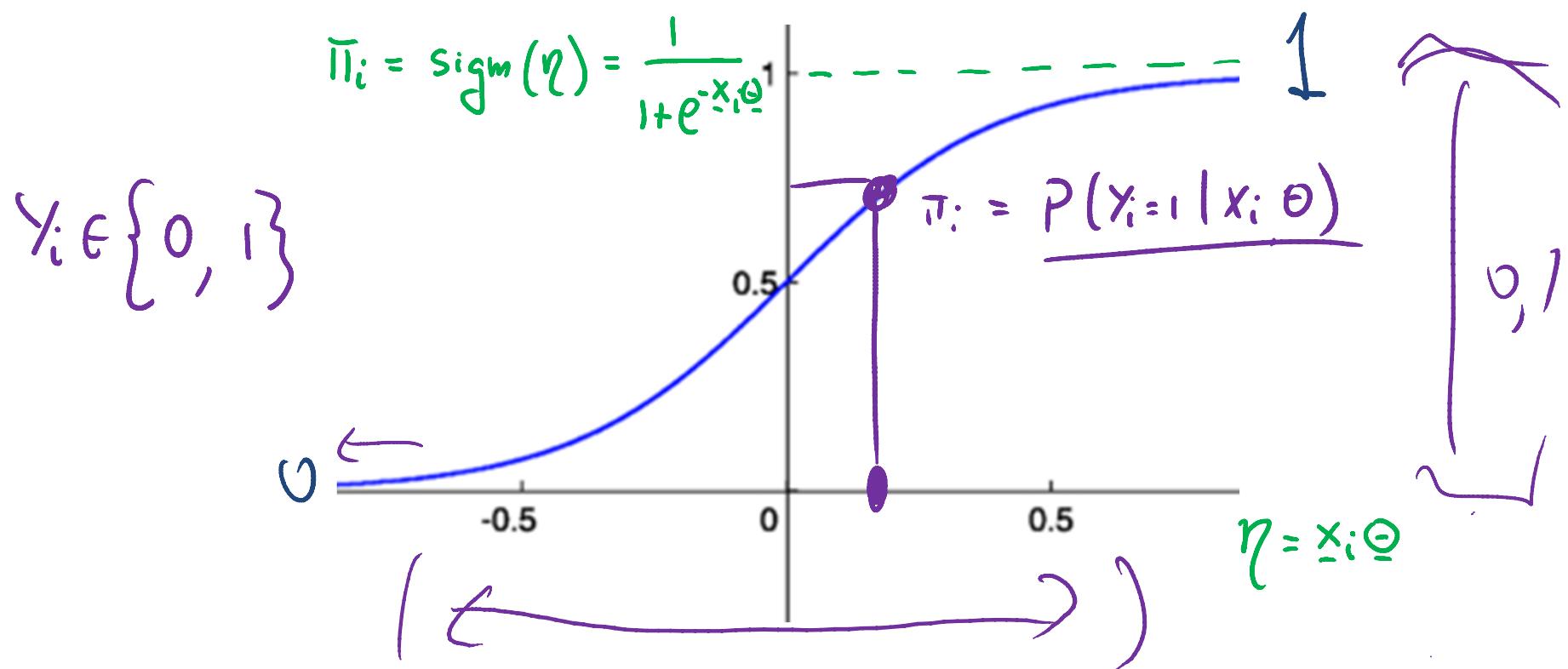
# McCulloch-Pitts model of a neuron



# Sigmoid function

$\text{sigm}(\eta)$  refers to the **sigmoid** function, also known as the **logistic** or **logit** function:

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



# Linear separating hyper-plane

$$P(y_i=1 | \underline{x}_i, \underline{\theta}) = \text{Sigm}(\underline{x}_i \cdot \underline{\theta}) = \frac{1}{1+e^{-\underline{x}_i \cdot \underline{\theta}}}$$

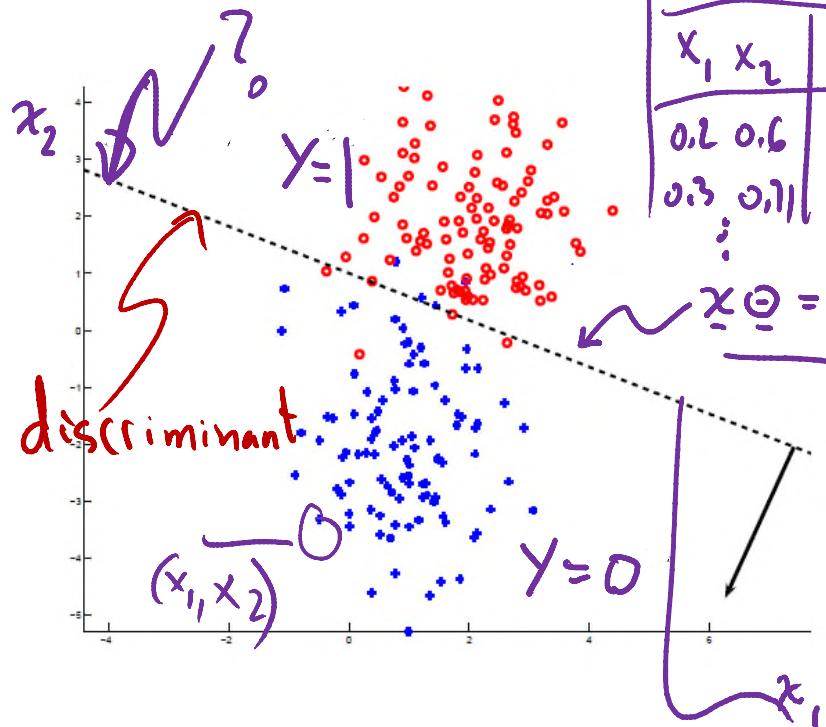
When

i.e.

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

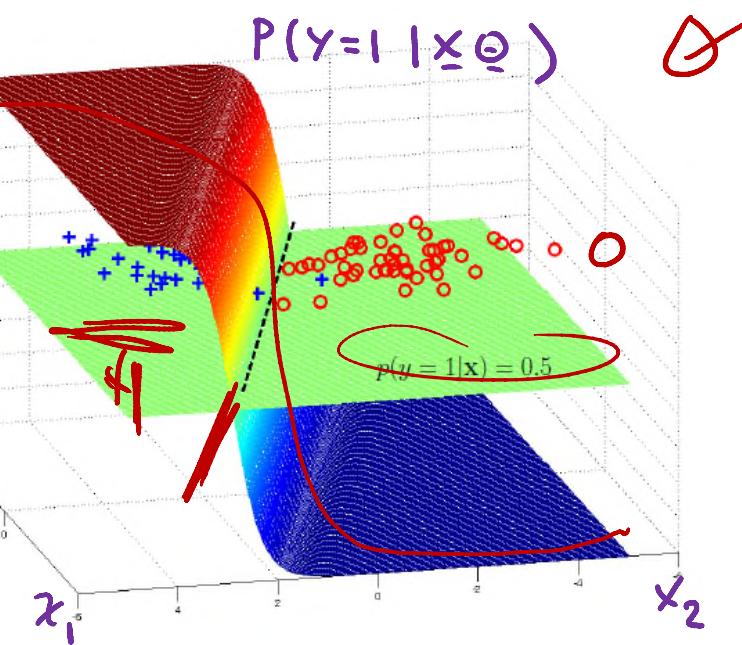
$$\underline{x}_i \cdot \underline{\theta} = 0$$

EQUATION OF A  
PLANE.



DATA		
$x_1$	$x_2$	$y$
0.2	0.6	0
0.3	0.7	1
⋮	⋮	⋮

$$\underline{x} \cdot \underline{\theta} = 0$$



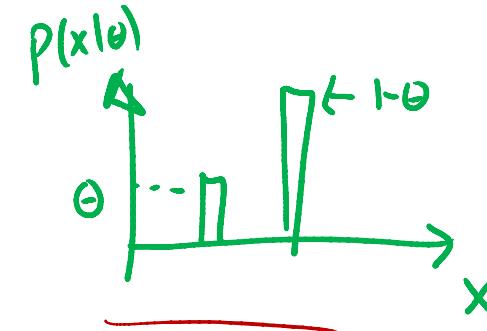
[Greg Shakhnarovich]

# 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  $\theta \in (0, 1)$ . We can write this probability more succinctly as follows:

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

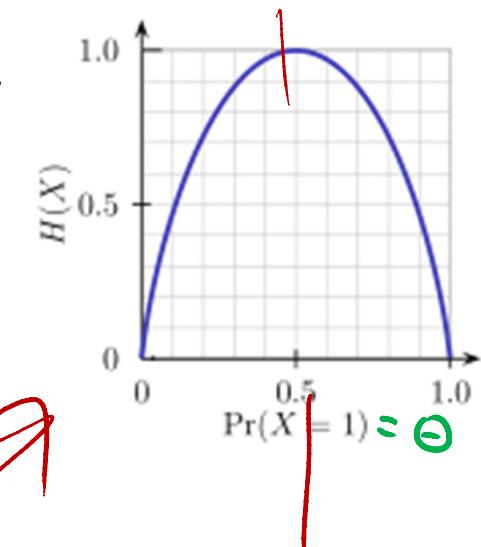
# Entropy

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

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

Example: For a Bernoulli variable  $\mathbf{X}$ , the entropy is:

$$\begin{aligned} H(x) &= - \sum_{x=0}^1 \Theta^x (1-\Theta)^{1-x} \log [\Theta^x (1-\Theta)^{1-x}] \\ &= -[(1-\Theta) \log (1-\Theta) + \Theta \log \Theta] \end{aligned}$$



# 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:

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

$p(y_i | x_i, \theta)$

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

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

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

$$C(\boldsymbol{\theta}) = -\log P(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta})$$

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

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:

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

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

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)

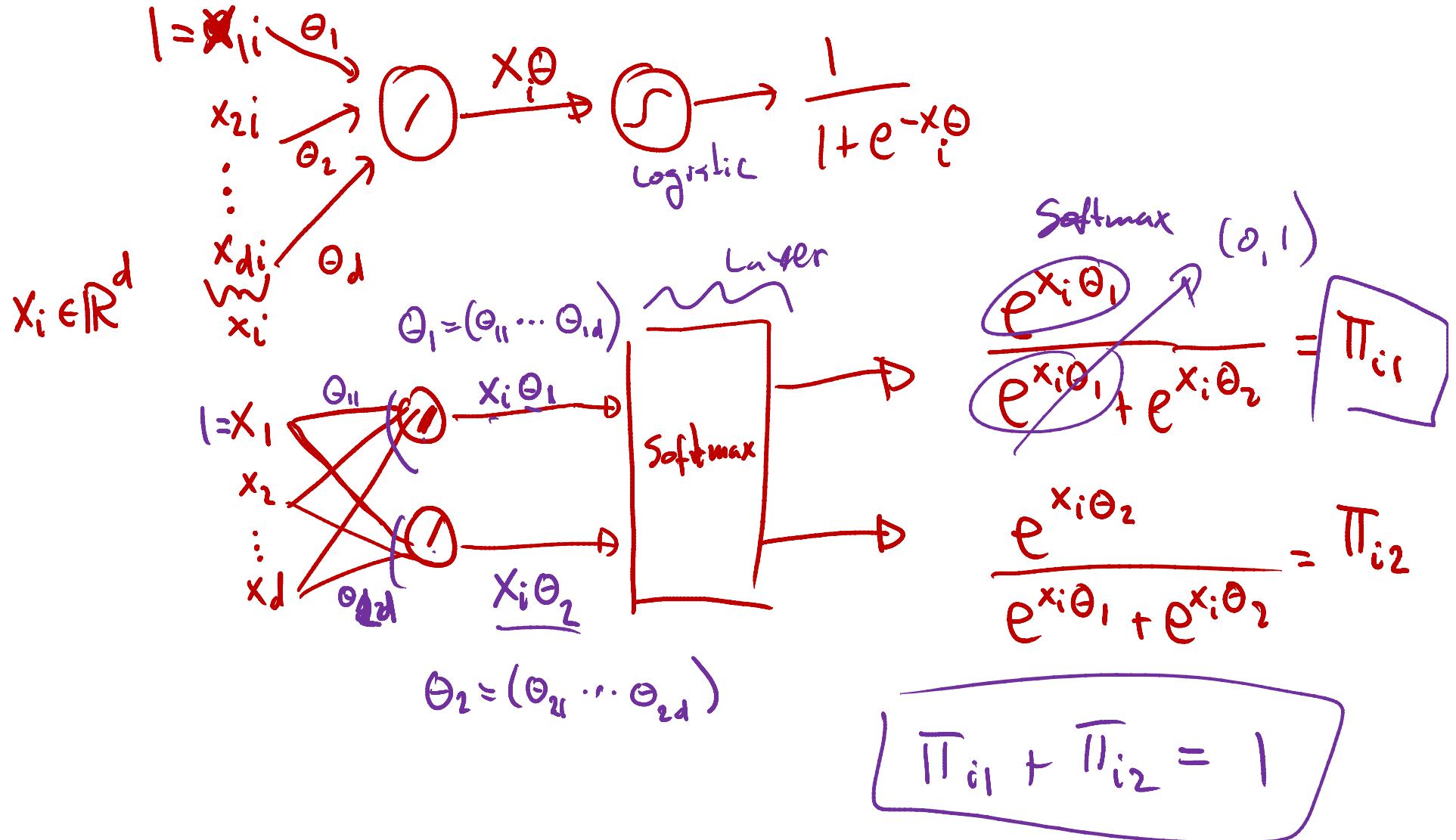
For binary logistic regression, recall that the gradient and Hessian of the negative log-likelihood are given by

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

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

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

# Softmax formulation



# Likelihood function

INDICATOR:

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

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

$$P(y_i|x_i, \Theta) = \begin{cases} \pi_{i1} = \frac{e^{x_i \Theta_1}}{e^{x_i \Theta_1} + e^{x_i \Theta_2}} & \text{if } y_i = 0 \\ \pi_{i2} = \frac{e^{x_i \Theta_2}}{e^{x_i \Theta_1} + e^{x_i \Theta_2}} & \text{if } y_i = 1 \end{cases}$$

# Negative log-likelihood criterion

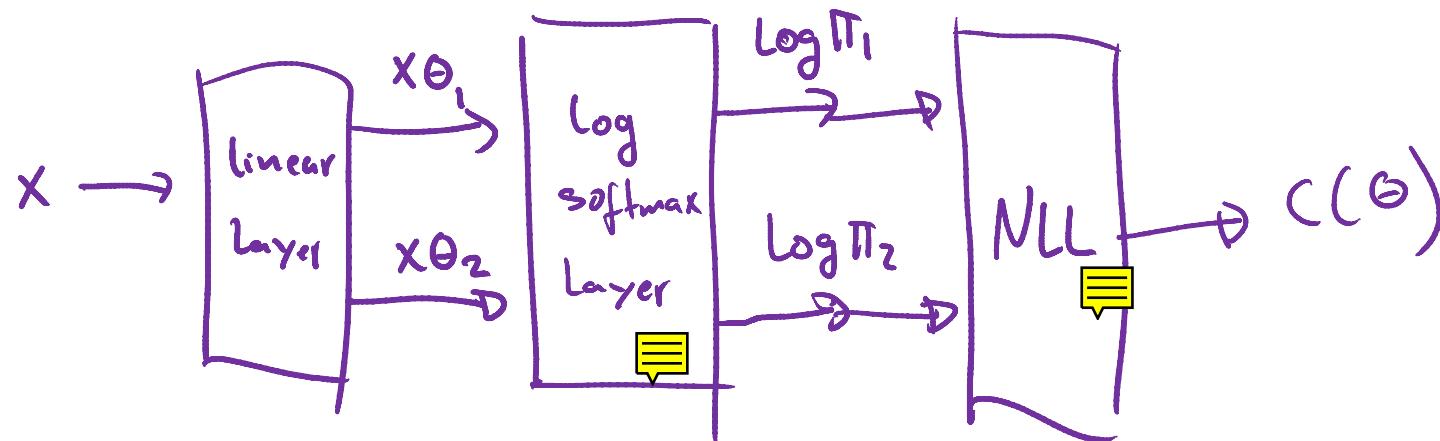
$$P(y|x,\theta) = \prod_{i=1}^n \pi_{i1}^{y_i} \pi_{i2}^{1-y_i}$$

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

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

Logsoftmax  
Lsf

# Neural network representation of loss



# Manual gradient computation

$$\frac{\partial C(\theta)}{\partial \theta_2} = \frac{\partial}{\partial \theta_2} \left( - \sum_{i=1}^n \Pi_0(y_i) \log \Pi_{i1} + \Pi_1(y_i) \log \Pi_{i2} \right)$$

$$= - \sum_{i=1}^n \left( \Pi_0(y_i) \frac{\partial \log \Pi_{i1}}{\partial \theta_2} + \Pi_1(y_i) \frac{\partial \log \Pi_{i2}}{\partial \theta_2} \right)$$

$$\begin{aligned} \frac{\partial}{\partial \theta_2} (\log \Pi_{i1}) &= \frac{\partial}{\partial \theta_2} \log \left( \frac{e^{x_i \theta_1}}{e^{x_i \theta_1} + e^{x_i \theta_2}} \right) = \frac{\partial}{\partial \theta_2} \left( x_i \overset{\theta_1}{\cancel{\theta_1}} - \log (e^{x_i \theta_1} + e^{x_i \theta_2}) \right) \\ &= 0 - \frac{x_i e^{x_i \theta_2}}{e^{x_i \theta_1} + e^{x_i \theta_2}} = -x_i \Pi_{i2} \end{aligned}$$

$$\frac{\partial}{\partial \theta_2} \log \Pi_{i2} = x_i (1 - \Pi_{i2})$$

# Manual gradient computation

$$\frac{\partial C(\theta)}{\partial \theta_2} = - \sum_{i=1}^n I_o(y_i) x_i \pi_{i2} + I_1(y_i) x_i (1 - \pi_{i2})$$

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

Check

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

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

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

# 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.