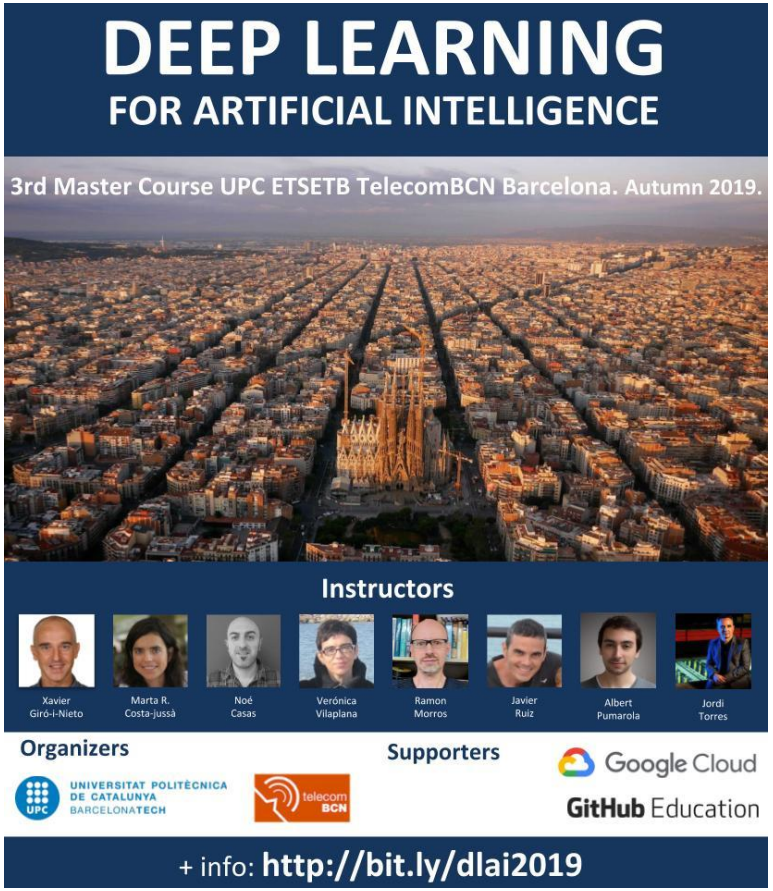


#DLUPC

Day 2 Lecture 3

Softmax Regression

[course site]

Xavier Giro-i-Nieto
xavier.giro@upc.edu

Associate Professor
Universitat Politècnica de Catalunya
Technical University of Catalonia

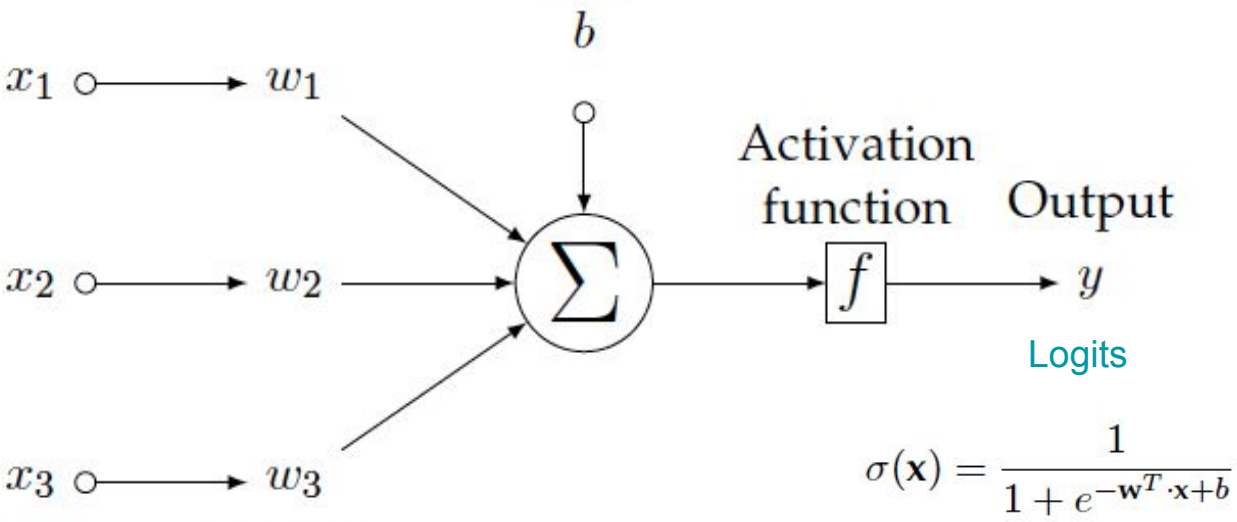
Acknowledgements

Santiago Pascual

Kevin McGuinness
kevin.mcguinness@dcu.ie

Research Fellow
Insight Centre for Data Analytics
Dublin City University

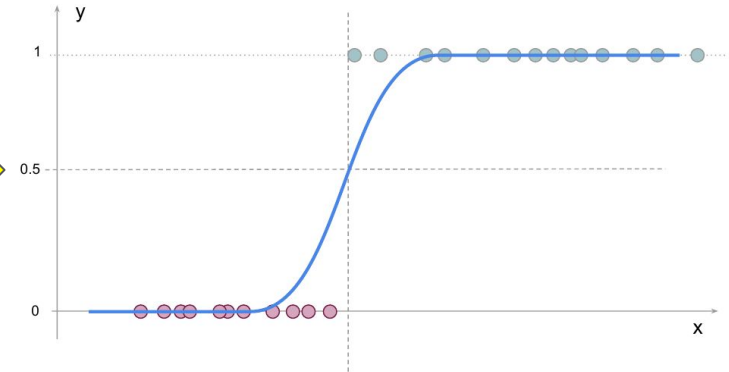
Previously… Logistic Regression



y > ϒ → class 1
(eg. green)

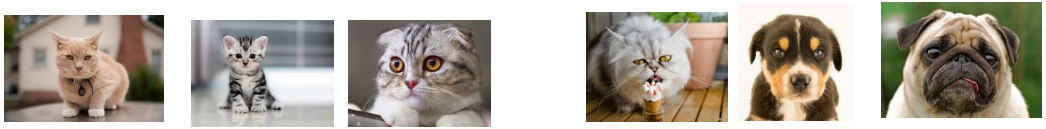
y < ϒ → class 2
(eg. red)

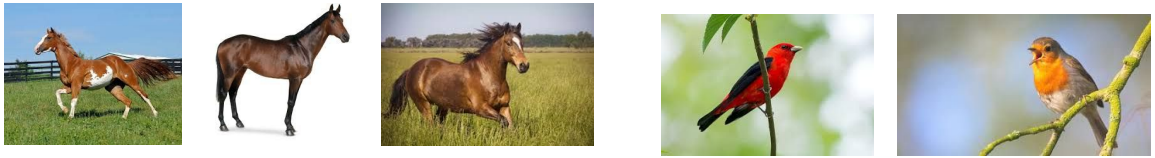
Threshold (ϒ)



Softmax regression: Multiclass (N classes)

Question: How could the logistic regression be adapted to a problem with more than 2 classes (N) ?

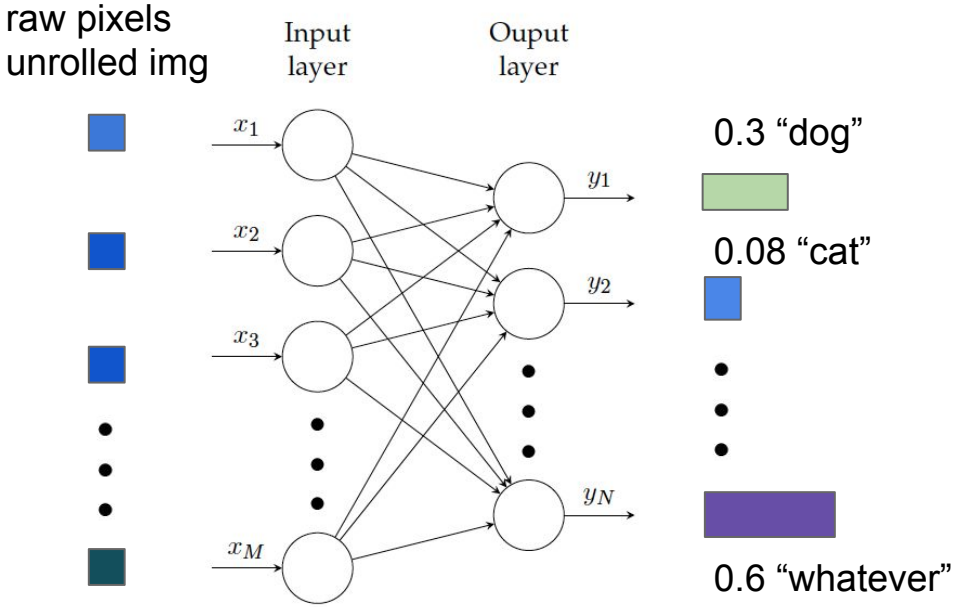




Softmax regression: Multiclass (N classes)

Question: How could the binary classifier with logistic regression to a problem with more than 2 classes (N) ?

A multiclass classification problem can be solved by assigning a perceptron for each class and choosing the **maximum** logit…

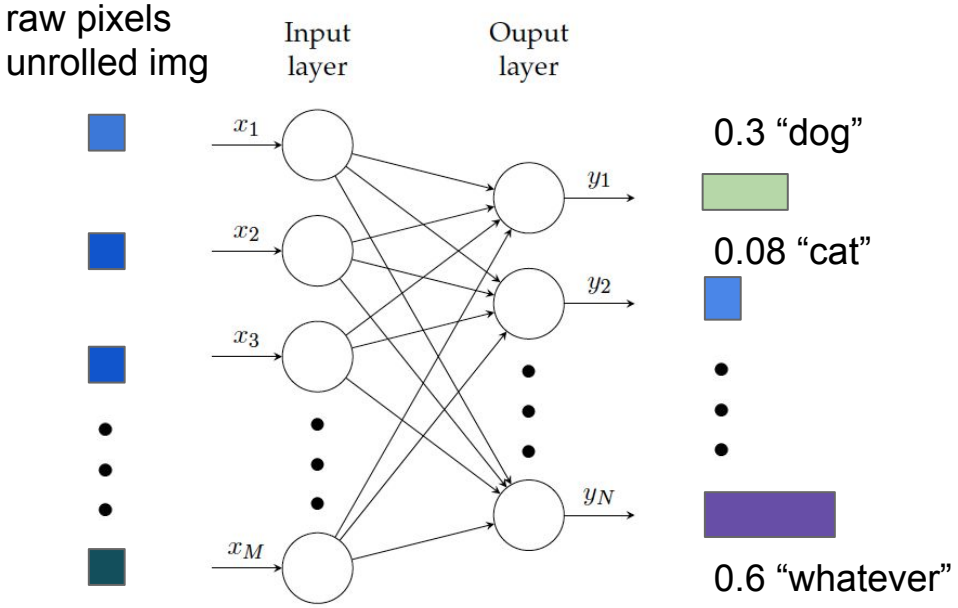


Softmax regression: Multiclass (N classes)

Question: How could the binary classifier with logistic regression to a problem with more than 2 classes (N) ?

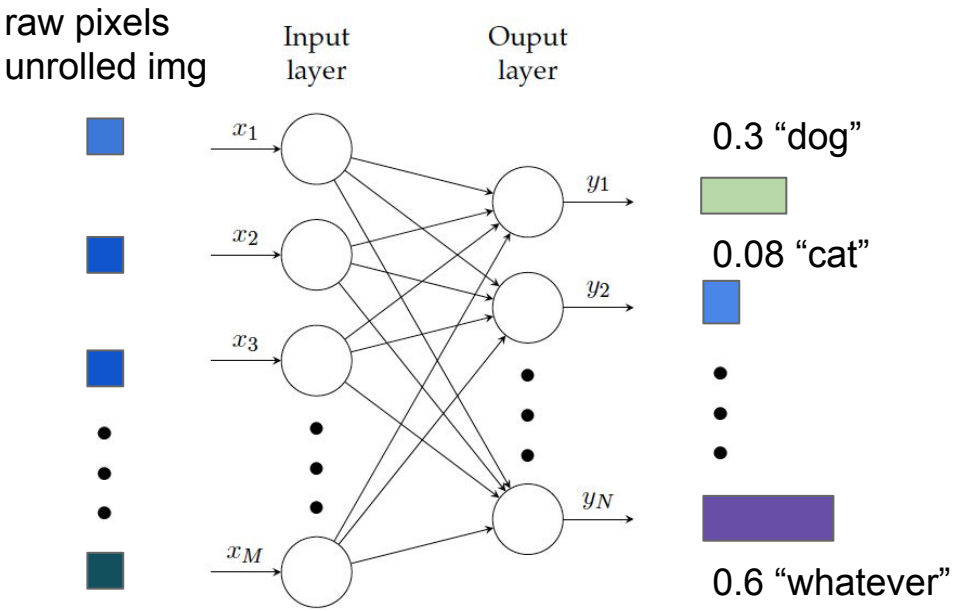
A multiclass classification problem can be solved by assigning a perceptron for each class and choosing the **maximum** logit…

but the max function is **non-differentiable**.



Softmax regression: Multiclass (N classes)

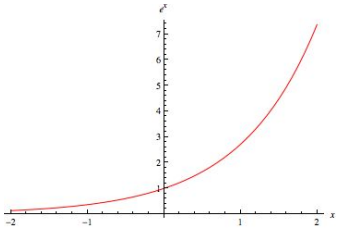
The output logits are normalized with the softmax function, which is **differentiable**:



Softmax regression

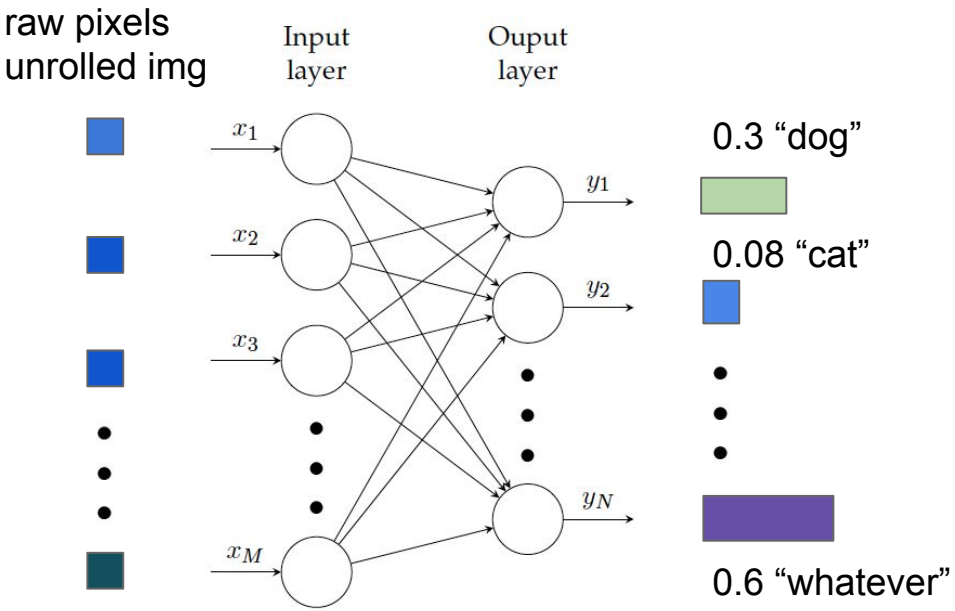
$$P(y = k|\mathbf{x}) = \frac{\exp \mathbf{x}^T \mathbf{w}_k}{\sum_{n=1}^{N} \exp \mathbf{x}^T \mathbf{w}_n}$$

Exponential *exp(·)* boosts higher logits (max effect).



Softmax regression: Multiclass (N classes)

The output logits are normalized with the softmax function, which is **differentiable**:



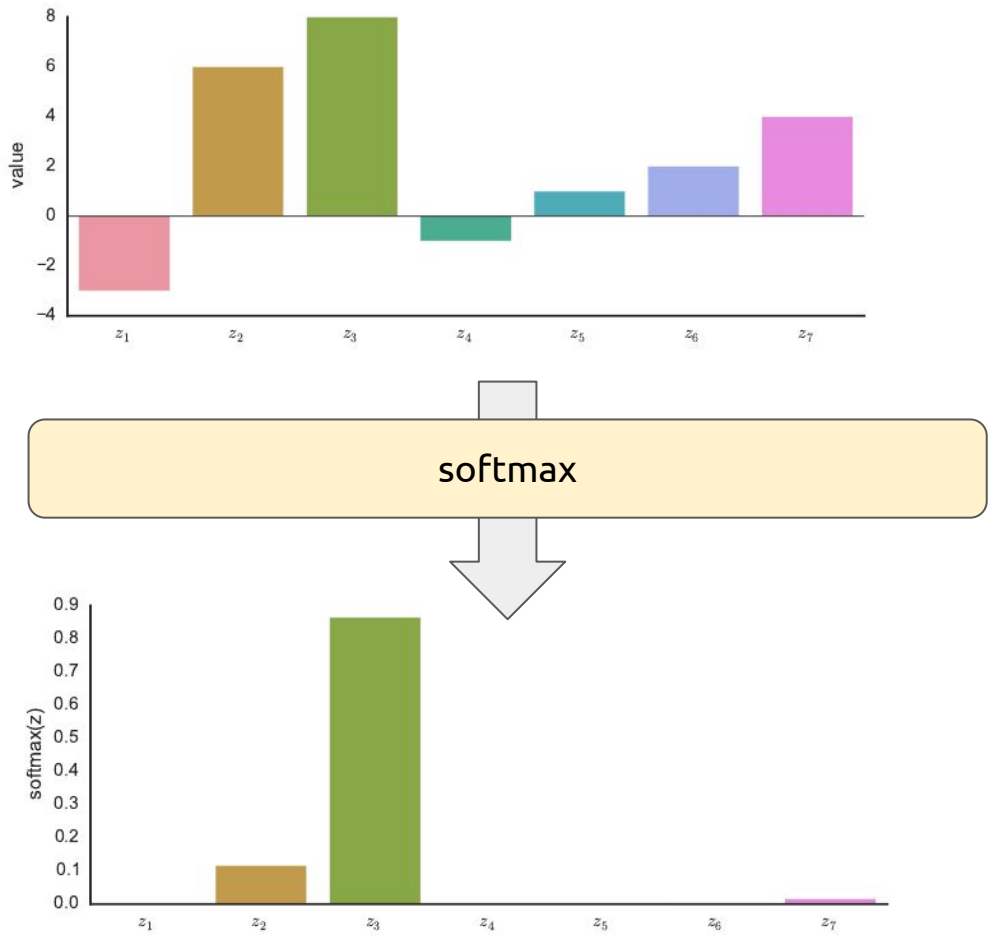
Softmax regression

$$P(y = k|\mathbf{x}) = \frac{\exp \mathbf{x}^T \mathbf{w}_k}{\sum_{n=1}^{N} \exp \mathbf{x}^T \mathbf{w}_n}$$

Normalization factor so that the sum of probabilities sum up to 1.

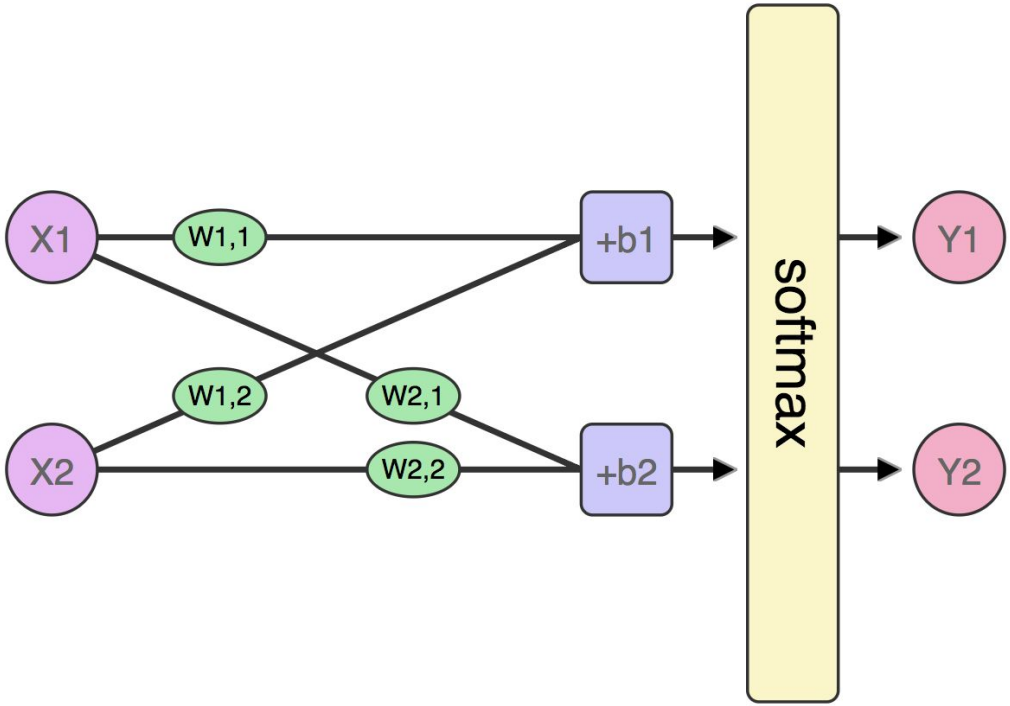
Softmax regression

$$\text{softmax}(\mathbf{x}) = \frac{1}{\sum_{j=1}^{K} \exp(x_j)} \begin{bmatrix} \exp(x_1) \\ \exp(x_2) \\ \vdots \\ \exp(x_K) \end{bmatrix}$$



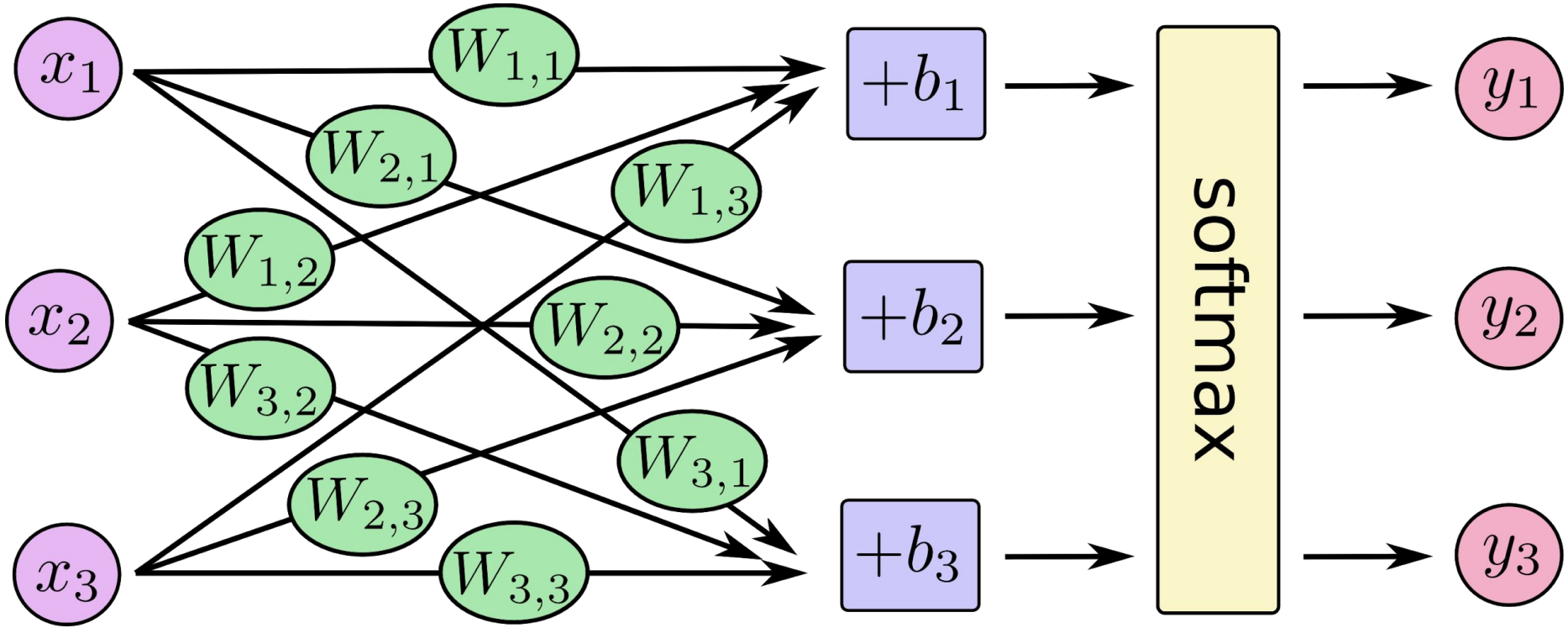
Softmax regression: Binary case

Example: Binary classification can also be solved with two perceptrons + softmax.



J. Alammar, "A visual and interactive guide to the Basics of Neural Networks" (2016)

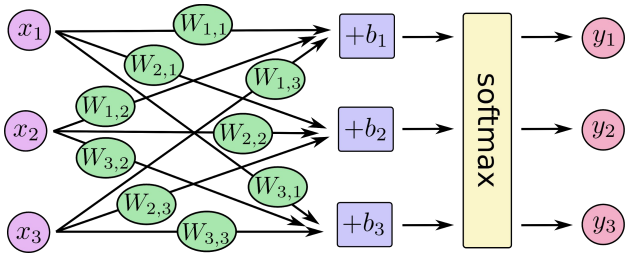
Softmax regression: Multiclass (3 classes)



TensorFlow, "MNIST for ML beginners"

Softmax regression: Multiclass (3 classes)

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax} \left(\begin{matrix} W_{1,1} x_1 + W_{1,2} x_2 + W_{1,3} x_3 + b_1 \\ W_{2,1} x_1 + W_{2,2} x_2 + W_{2,3} x_3 + b_2 \\ W_{3,1} x_1 + W_{3,2} x_2 + W_{3,3} x_3 + b_3 \end{matrix} \right)$$

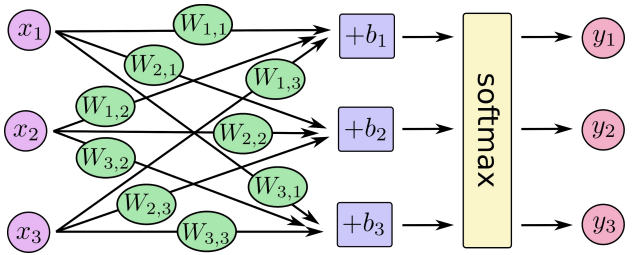


TensorFlow, "MNIST for ML beginners"

Softmax regression: Multiclass (3 classes)

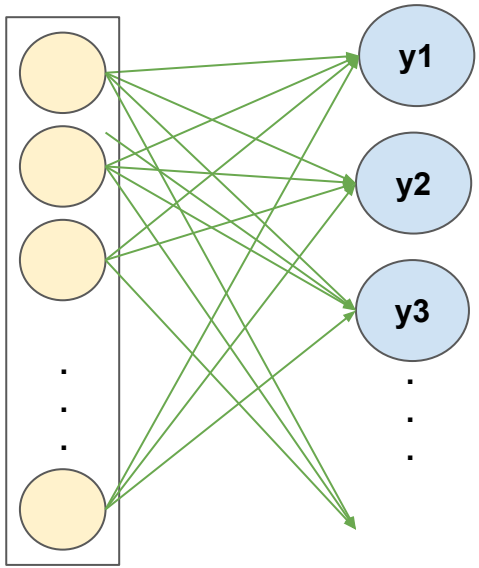
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax}\left(\begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} \\ W_{2,1} & W_{2,2} & W_{2,3} \\ W_{3,1} & W_{3,2} & W_{3,3} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

$$y = \text{softmax}(Wx + b)$$



TensorFlow, "MNIST for ML beginners"

Software implementation

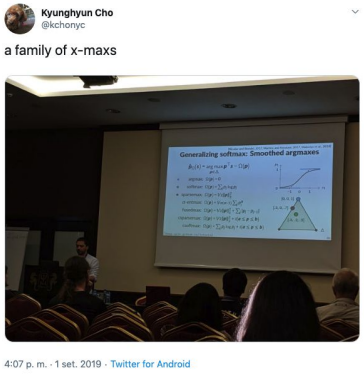


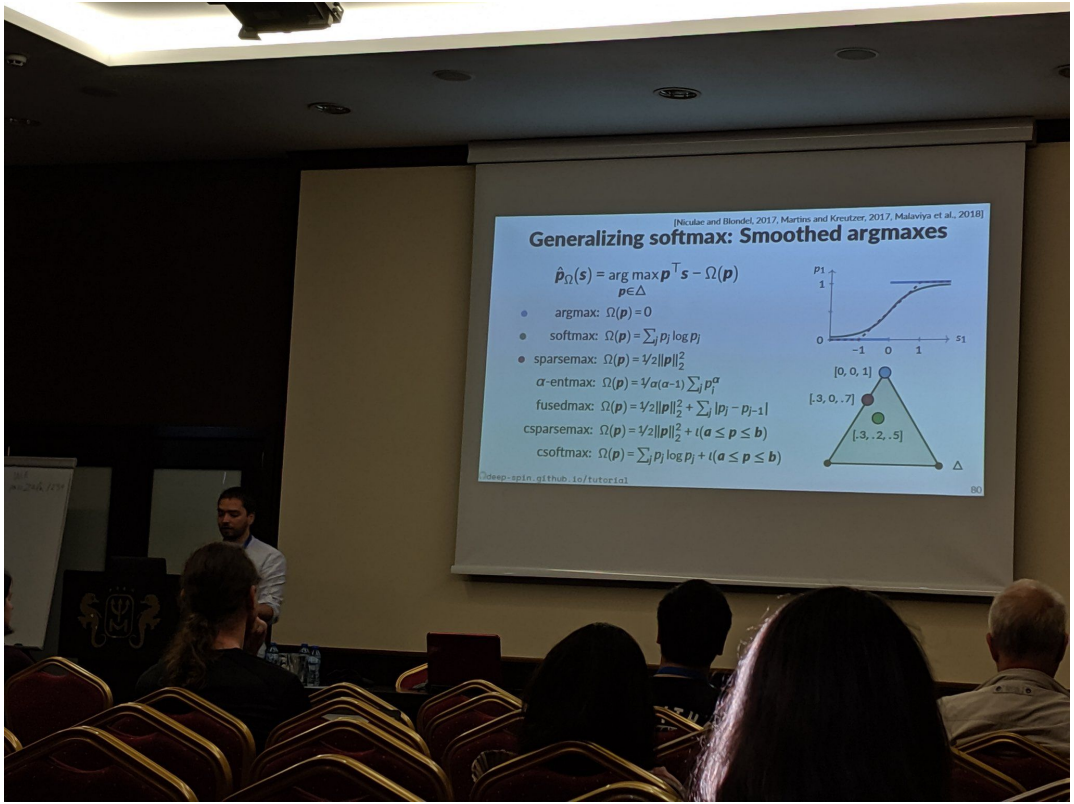
PYTORCH

```
smx = nn.Sequential(
    nn.Linear(NUM_INPUTS, NUM_OUTPUTS),
    nn.LogSoftmax(dim=1)
)
```

Learn more

Source:
Kyunghyun Cho (@kchonyc)





End-to-end differentiable relaxations:
https://deep-spin.github.io/tutorial/acl.pdf

Undergradese

What undergrads ask vs. what they're REALLY asking