

Multiclass classification with neural networks

Softmax regression

Filippo Biscarini
Senior Scientist
CNR, Milan (Italy)

Nelson Nazzicari
Research fellow
CREA, Lodi (Italy)



Multiclass classification



(a)



(b)



(c)



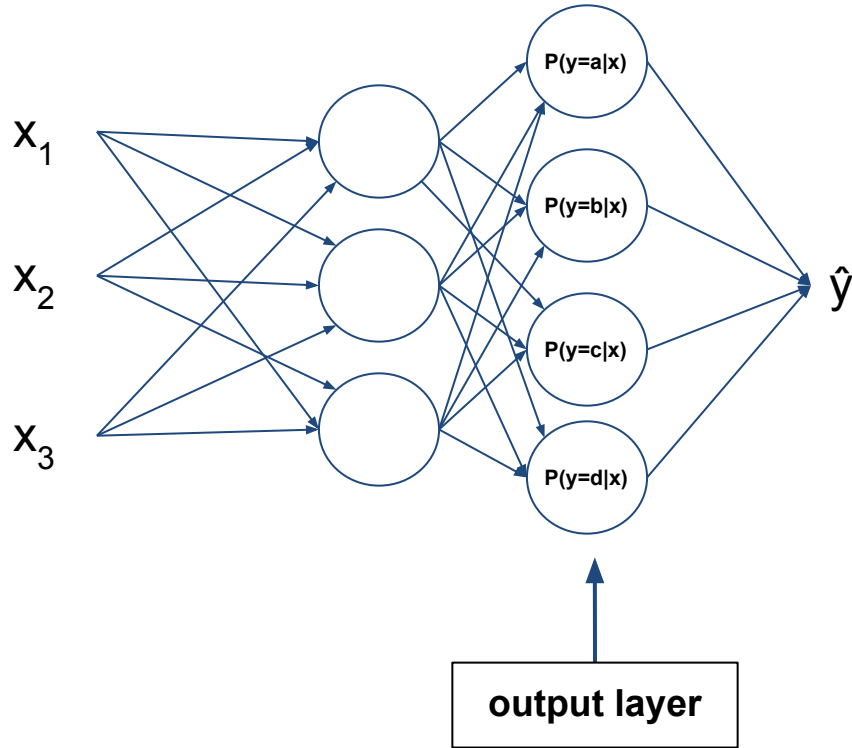
(d)

From: Wang et al., 2015 (Entropy)

- **more than two classes** to recognize
- in this case, 4 classes: bananas, green apples, red apples, oranges
- need to extend logistic regression to estimating the **probabilities** of samples **to belong** to each of the **four classes**



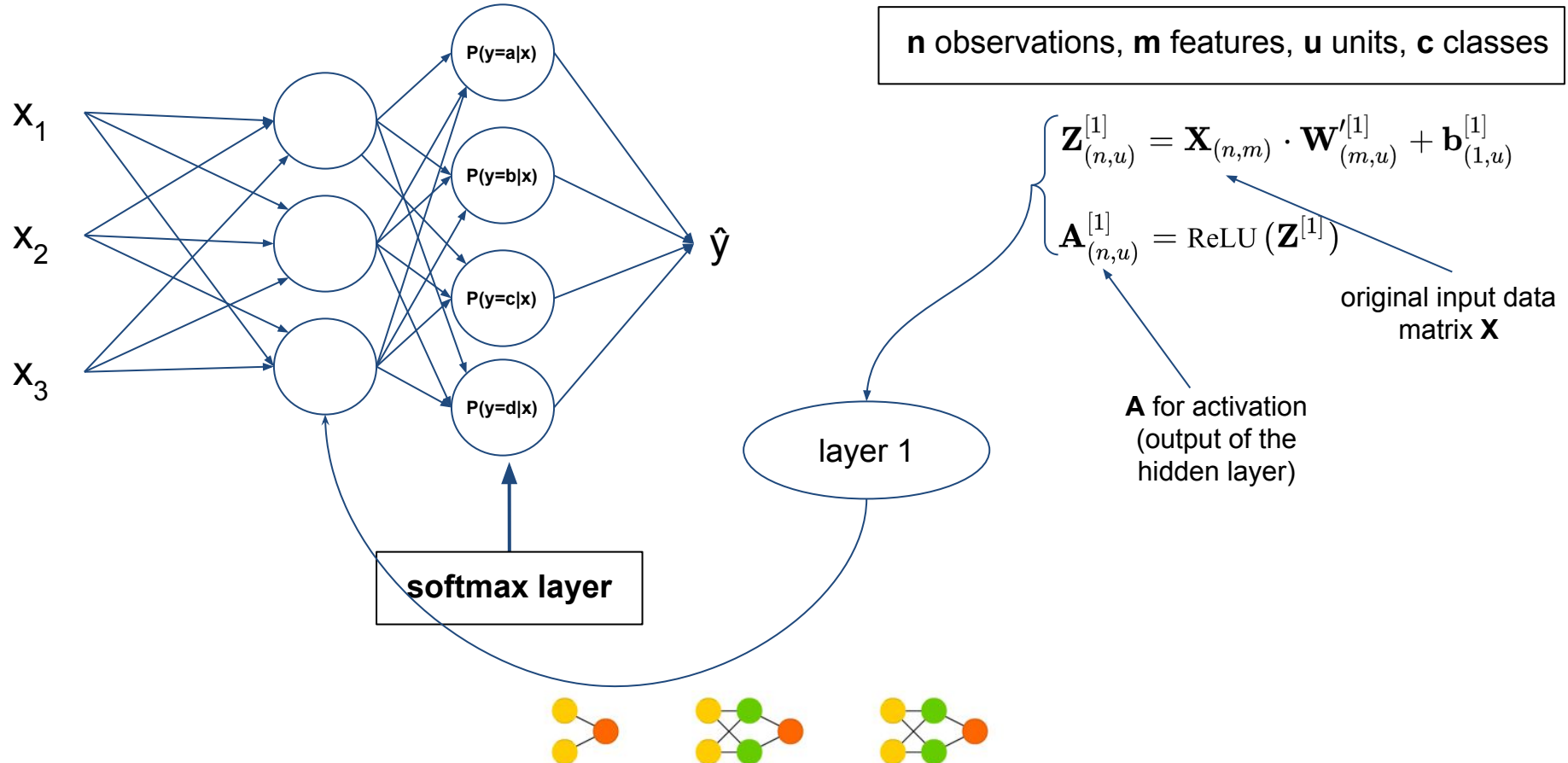
Multiclass classification



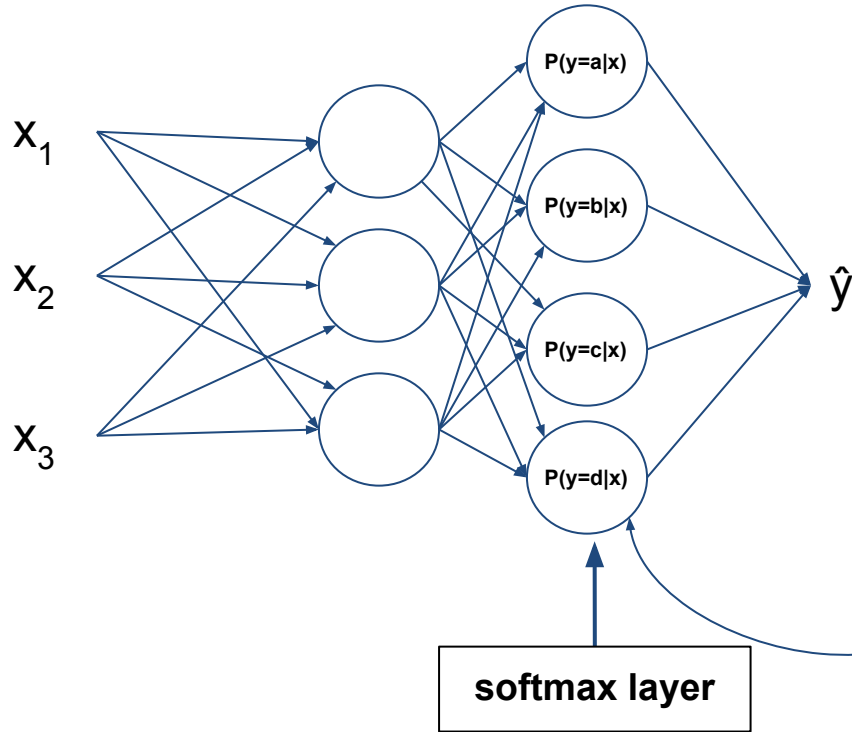
- the **output layer** has now **4 nodes** (instead of just one as in binary classification)
- the **activation** function is now the **softmax function**
- this is why the output layer is also called the **softmax layer**, and multiclass classification is called **softmax regression**



Multiclass classification



Multiclass classification



n observations, **m** features, **u** units, **c** classes

$$\begin{cases} \mathbf{Z}_{(n,u)}^{[1]} = \mathbf{X}_{(n,m)} \cdot \mathbf{W}_{(m,u)}'^{[1]} + \mathbf{b}_{(1,u)}^{[1]} \\ \mathbf{A}_{(n,u)}^{[1]} = \text{ReLU}(\mathbf{Z}^{[1]}) \end{cases}$$

$$\mathbf{Z}_{(n,c)}^{[2]} = \mathbf{A}_{(n,u)}^{[1]} \cdot \mathbf{W}_{(u,c)}'^{[2]} + \mathbf{b}_{(1,c)}^{[2]}$$

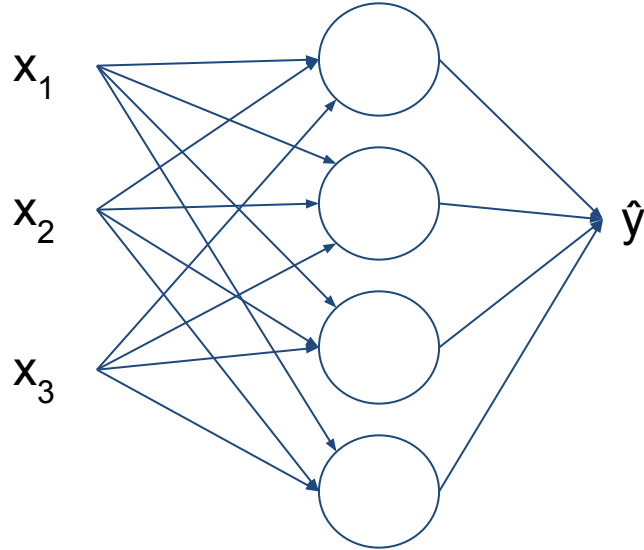
$$\hat{\mathbf{y}}_{(n,c)} = \text{softmax}(\mathbf{Z}^{[2]})$$

softmax activation

c, classes: $\hat{\mathbf{y}}$ is now a matrix (4 probabilities per sample)



Multiclass classification



- as we did for binary classification, we can also represent softmax regression with no hidden layers
- this would essentially reduce to multinomial logistic regression, with no neural networks involved
- we'll see this implementation in the ipynb file 4c, part 1



Loss function for softmax regression

$$L(\hat{y}, y) = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y}))$$

loss function for logistic
regression

$$L(\hat{y}, y) = - \sum_{j=1}^c y_j \cdot \log(\hat{y}_j)$$

loss function for softmax
regression

- generalization of the loss function for logistic regression over c classes
- you then sum up the loss for each sample and divide by the number of samples \rightarrow cost function for the entire dataset



Loss function for softmax regression

$$L(\hat{y}, y) = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y}))$$

loss function for logistic
regression

$$L(\hat{\mathbf{y}}, \mathbf{y}) = - \sum_{j=1}^c y_j \cdot \log(\hat{y}_j)$$

loss function for softmax
regression

these are actually vectors of length c (number of classes)!



Multiclass classification

- demonstration 04c
- exercise 04c.1 (multiclass logistic regression)

→ `code_04c_keras_multiclass_classification.ipynb`



Neural networks models: recap

- exercise 04d.1 (write your own code)

→ `code_04d_neural_networks_exercise.ipynb`

