

Presentation Title

Subtitle

Author

Instituto Federal do Espírito Santo
Campus Vitória

August 17, 2018



Schedule

- 1 Overview
- 2 Blocks
- 3 Boxes
- 4 Lists
 - List items
 - Numbered list
 - Descriptive list
- 5 Tables
- 6 Figures
- 7 Equations and Codes
 - Equations
 - Programming



Overview



Overview

Normal text **Alert Text** Example Text **Emphasis Text**

Simple block

■ ...

Example block

■ ...

Alert block

■ ...

A purple box

An orange box

A gray box

My price table

Color	Price 1	Price 2	Price 3
Red	10.00	20.00	30.00
Green	20.00	30.00	40.00
Blue	30.00	40.00	50.00
Orange	60.00	90.00	120.00



Blocks



Blocks types

Simple block

- First point
- Second point
- Third point

Examples block

- First point
- Second point
- Third point

Alert block

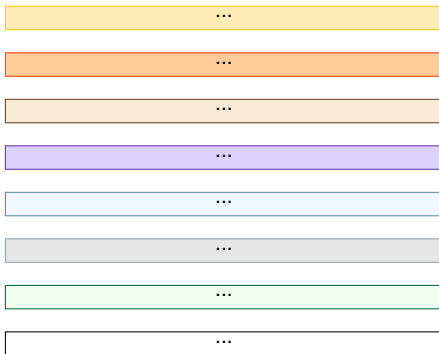
- First point
- Second point
- Third point



Boxes



Boxes



Lists



Items

- ...
- ...
- ...



Numbered

1 ...

2 ...

3 ...



Descriptive

Theme 1: ...

Theme 2: ...

Theme 3: ...



Tables



Tables 1

My price table		
Couleur	Prix 1	Prix 2
Rouge	10.00	20.00
Vert	20.00	30.00
Bleu	30.00	40.00
Orange	60.00	90.00

My price table		
Couleur	Prix 1	Prix 2
Rouge	10.00	20.00
Vert	20.00	30.00
Bleu	30.00	40.00
Orange	60.00	90.00

My price table		
Couleur	Prix 1	Prix 2
Rouge	10.00	20.00
Vert	20.00	30.00
Bleu	30.00	40.00
Orange	60.00	90.00

My price table		
Couleur	Prix 1	Prix 2
Rouge	10.00	20.00
Vert	20.00	30.00
Bleu	30.00	40.00
Orange	60.00	90.00



Tables 2

My price table		
Couleur	Prix 1	Prix 2
Rouge	10.00	20.00
Vert	20.00	30.00
Bleu	30.00	40.00
Orange	60.00	90.00

My price table		
Couleur	Prix 1	Prix 2
Rouge	10.00	20.00
Vert	20.00	30.00
Bleu	30.00	40.00
Orange	60.00	90.00

My price table		
Couleur	Prix 1	Prix 2
Rouge	10.00	20.00
Vert	20.00	30.00
Bleu	30.00	40.00
Orange	60.00	90.00

My price table		
Couleur	Prix 1	Prix 2
Rouge	10.00	20.00
Vert	20.00	30.00
Bleu	30.00	40.00
Orange	60.00	90.00



Figures



Figure Example

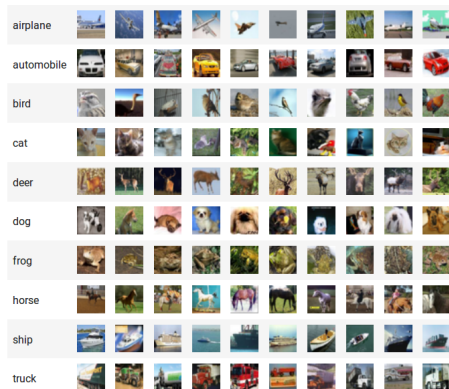


Figure: Example images from the CIFAR-10 dataset.



Equations and Codes



Equation Example

Some random equation:

$$\begin{aligned}\frac{\partial}{\partial \theta_k} J(\theta) &= \frac{\partial}{\partial \theta_k} \left[\frac{1}{m} \sum_{k=1}^m \log(1 + e^{-y^{(i)} \theta^T x^{(i)}}) \right] \\ &= \frac{1}{m} \sum_{k=1}^m \frac{1}{1 + e^{-y^{(i)} \theta^T x^{(i)}}} y^{(i)} x_k^{(i)} \\ &= -\frac{1}{m} \sum_{k=1}^m h_{\theta}(-y^{(i)} x^{(i)}) y^{(i)} x_k^{(i)}\end{aligned}$$



Code Example

```
def softmax_loss_naive(W, X, y, reg):  
    """  
    Softmax loss function, naive implementation (with loops)  
  
    Inputs have dimension D, there are C classes, and we operate on minibatches  
    of N examples.  
  
    Inputs:  
    - W: A numpy array of shape (D, C) containing weights.  
    - X: A numpy array of shape (N, D) containing a minibatch of data.  
    - y: A numpy array of shape (N,) containing training labels; y[i] = c means  
        that X[i] has label c, where 0 <= c < C.  
    - reg: (float) regularization strength  
  
    Returns a tuple of:  
    - loss as single float  
    - gradient with respect to weights W; an array of same shape as W  
    """
```

carbon
carbon.now.sh

1

¹Carbon

