



Machine Learning

Neural Networks: Learning

Implementation
note: Unrolling
parameters

Advanced optimization

```
function [jVal, gradient] = costFunction(theta)  
...  
optTheta = fminunc(@costFunction, initialTheta, options)
```

Handwritten annotations: \mathbb{R}^{n+1} (twice) and \mathbb{R}^{n+1} (vectors) with arrows pointing to gradient, theta, and initialTheta respectively.

Neural Network (L=4):

→ $\Theta^{(1)}, \Theta^{(2)}, \Theta^{(3)}$ - matrices (Theta1, Theta2, Theta3)

→ $D^{(1)}$, $D^{(2)}$, $D^{(3)}$ - matrices (D1, D2, D3)

“Unroll” into vectors

n unidades na
layer 1,2,3

Example

$$\underline{s_1} = 10, \underline{s_2} = 10, \underline{s_3} = 1$$

$$\rightarrow \Theta^{(1)} \in \mathbb{R}^{10 \times 11}, \Theta^{(2)} \in \mathbb{R}^{10 \times 11}, \Theta^{(3)} \in \mathbb{R}^{1 \times 11}$$

$$\rightarrow D^{(1)} \in \mathbb{R}^{10 \times 11}, D^{(2)} \in \mathbb{R}^{10 \times 11}, D^{(3)} \in \mathbb{R}^{1 \times 11}$$

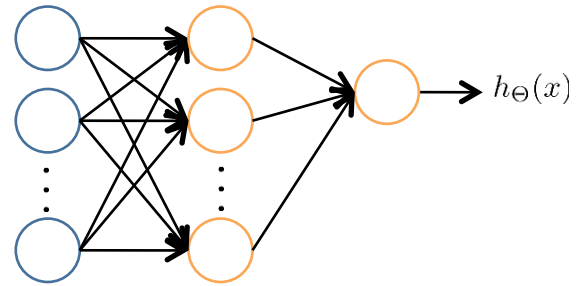
$$\rightarrow \text{thetaVec} = [\text{Theta1}(:); \text{Theta2}(:); \text{Theta3}(:)] ; \text{meter todos thetas num longo vector}$$

$$\rightarrow \text{DVec} = [\text{D1}(:); \text{D2}(:); \text{D3}(:)] ;$$

$$\text{Theta1} = \text{reshape}(\text{thetaVec}(1:110), 10, 11) ;$$

$$\rightarrow \text{Theta2} = \text{reshape}(\text{thetaVec}(111:220), 10, 11) ;$$

$$\rightarrow \text{Theta3} = \text{reshape}(\text{thetaVec}(221:231), 1, 11) ;$$



(10) (11)

(10) (11)

(10) (11)

Learning Algorithm

- Have initial parameters $\Theta^{(1)}, \Theta^{(2)}, \Theta^{(3)}$.
- Unroll to get `initialTheta` to pass to
- `fminunc(@costFunction, initialTheta, options)`

`function [jval, gradientVec] = costFunction(thetaVec)`

- From thetaVec, get $\Theta^{(1)}, \Theta^{(2)}, \Theta^{(3)}$. *reshape*
- Use forward prop/back prop to compute $D^{(1)}, D^{(2)}, D^{(3)}$ $J(\Theta)$
and $D^{(1)}, D^{(2)}, D^{(3)}$
Unroll _____ to get gradientVec.