Referência em neural networks: Yan LeCun, from NYU

Multiclass classification with neuron networks:

$$y^{(i)} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \text{ or } \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \text{ or } \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \text{ or } \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

etc.

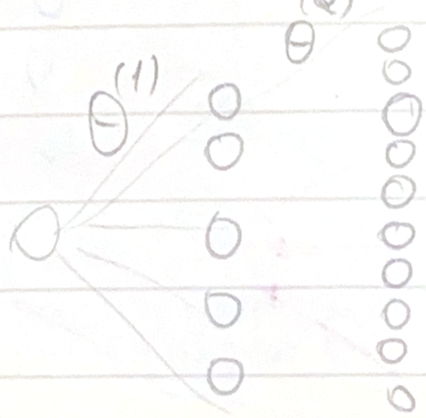
instead of $y = 1, 2, 3$ or 4

So, the training set looks like:

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots \quad (x^{(m)}, y^{(m)})$$

So $h_\theta(x^{(i)}) \cong y^{(i)}$

$h_\theta(x^{(i)}) \in \mathbb{R}^4$



$[\theta^{(2)}]_{10\times 6}$

Coqueiros, 04 de novembro de 2018

Aplicação / Implementação em OCTAVE

Função Custo p/ Reg. Log. c/ Reg.:

```
J = -1/m * sum( y' * log(sigmoid (X*theta))...
    +(1-y') * log(1-sigmoid(X*theta)))...
    + lambda/(2*m) * sum(theta(2:end).^2);
```

Gradientes:

```
temp = theta;
temp(1) = 0;
grad = 1/m *(X'*(sigmoid(X*theta)-y)...
    + lambda * temp);
```

(n+1)x1 ; mx1 ; (n+1)xm ; [mx(n+1)]x[(n+1)x1] - mx1