$$L^{[m\times m]} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ Strips bias terms from Θ

Linear Regression:

$J(\theta) = \frac{1}{2m}(X\theta - y)^T(X\theta - y) + \frac{\lambda}{2m}[(L\theta)^T(L\theta)]$ Cost Function

$\nabla J = \frac{1}{m}(X^T(X\theta - y) + \lambda L\theta)$ Cost Gradient

$\theta := (1 - \frac{\alpha\lambda}{m}L\theta) - \frac{\alpha}{m}X^T(X\theta - y)$ Paramater Update Rule

Logistic Regression:

$\sigma(z) = \frac{1}{1+e^{-z}}$ Logistic/Sigmoid Function

$h = \sigma(X * \Theta')$ Hypothesis/Prediction Function

$J(\theta) = -\frac{1}{m}[y^T \log(h) + (1-y)^T \log(1-h)] + \frac{\lambda}{2m}[(L\theta)^T(L\theta)]$ Cost Function

$\nabla J = \frac{1}{m}(X^T(h - y) + \lambda L\theta)$ Cost Gradient

$\theta := (1 - \frac{\alpha\lambda}{m}L\theta) - \frac{\alpha}{m}X^T(h - y)$ Paramater Update Function

Neural Networks:

$\sigma'(z) = \sigma(z). * (1 - \sigma(z))$ Sigmoid Derivative

$a^l = \sigma(a^{l-1} * [\theta^{l-1}]^T)$ Activation/Output of layer l

$$Y^{(i)} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}, Y[k] = 1$$ One-Hot Form of Classification Result

$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{||L||}[Y. * log(a^{(L)}) + (1-Y). * log(1 - a^{(L)})] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\theta_{i,j}^{(l)})^2$ Cost function of NN used for Backpropogation, condensed down to a scalar

$\delta^{(L)} = a^{(L)} - Y$ Error of Neural Network Output

$\delta^{(l)} = \delta^{(l+1)}\widehat{\Theta}^{(l)} \odot \sigma'(a^{(l)}\Theta^{(l)})$ Error of layer l

$\widehat{\Theta}^{(l)} = \Theta^{(l)}[i,j]; i \in [1,2,3...m], j \in [2,3,4,...n]$ Reduced Paramater Matrix (No Bias Terms)

$\Delta^{(l)} = \delta^{(l+1)T} * a^{(l)}$ Cumulative Error of Layer l

$\nabla\frac{\partial J}{\partial\Theta^{(l)}} = \frac{1}{m}\Delta^{(l)} + \frac{\lambda}{m}\Theta^{(l)}L$ Gradient of the Cost Function for Layer l