Forward Pass and Backpropagation Summary in matrix

One hidden layer

Activation function : sigmoid f()

Input: $X_{1 x N}$ N inputs

Weights_input_to_hidden: W_i_h$_{N x M}$

Weights_hidden_to_output: W_h_o$_{M x K}$

$Y_{1 x K}$

Forward pass:

Hidden_layer_input$_{1 x M}$ = $X_{1 x N}$ × W_i_h$_{N x M}$

Hidden_layer_output$_{1 x M}$ = f(Hidden_layer_input) |$_{1 x M}$

Output_layer_input$_{1 x K}$= Hidden_layer_output$_{1 x M}$ × W_h_o$_{M x K}$

Output_y$_{1 x K}$= f(output_layer_in)|$_{1 x K}$

Backpropagation

Error = (Y - $\hat{Y}$)$_{1 x K}$

output_error_term:

$\delta_{1 x K}$ = Error $*$ f'(output_layer_input$_{1 x K}$) (* element wise product)

Hidden_error_term :

$\delta_{1 x M}$ = $\delta_{1 x K}$ × W_h_o$_{K x M}$ $*$ f'(hidden_layer_input$_{1 x M}$)

Delta_w_h_o$_{M x K}$ = learnrate * Hidden_layer_output$_{Mx 1}$ × $\delta_{1 x K}$

Delta_w_i_h$_{N x M}$ = learnrate * $X_{N x 1}$ × $\delta_{1 x M}$

Be very careful about how you calculate the error usually it is

Error = ($\hat{Y}$ -Y)$_{1 x K}$

When you update the weight you need to add "-" but Udacity uses

Error = (Y - $\hat{Y}$)$_{1 x K}$

In its homework. Then gradient descend sign "-" is already done Delta_w_i_h$_{N x M}$