Regression

Half sum of square errors (to avoid values in derivative)

Where:

number of output features

number of training examples

target of the output corresponding to the example

prediction of the output corresponding to the example

, is the cost of observation

Where:

, is the total loss

Where:

, is the target output matrix

Where:

index of a layer (excluding input and including output)

: activation of neuron in layer

activation function

Where:

, weighted sum of neuron in layer

weight from neuron in layer to neuron in layer (Input is considered   
), note that is the bias and . (In code biases are included at the end, instead).

, is the number of hidden units in the layer

Limit conditions:

Where:

, is the number of layer

# Derivatives

Where:

, is the vector of weights formed by all .

Where:

, is the index of the neuron

Last layer weights:

Using chain rule.

For does not depend on , when , then:

In general:

Hence:

For the future:

# Second to last layer

Summation is kept for all outputs are influenced by inner weight.

, to exclude the bias.

Every term:

Only the weights that connect the corresponding neurons.

Finally:

# Finally

# Classification

TO DO

# Example 1

1 input, 3 hiddens and 1 output

A diagram of a network

Description automatically generated