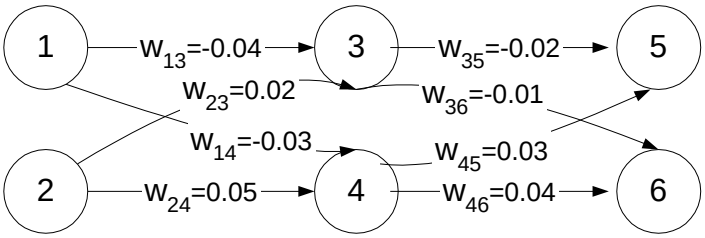
Given the network:



In C++, internal representation of this simple network is a 6 x 6 matrix for weights and two two-dimensional vectors vector< vector< double > > *output* and *error* for keeping the outputs and errors of the nodes. A node in *output*[i][j] corresponds to i-th layer and j-th node in that layer. To calculate the index of that node in the matrix *weights,* we can use a simple hash function *h(i,j)=2*i+j.* A dummy weight w_{ok} for each node *k* is stored at *weights[k][k].* With this representation, implementation of the back-propagation algorithm is straightforward.

For example, the output for node 1 on Figure above will correspond to *output*[*0*][*0*], for node 2 to *output*[*0*][*1*], for node 3 to *output*[*1*][*0*], for node 4 to *output*[*1*][*1*], for node 5 to *output*[*2*][*0*] and for node 6 to *output*[*2*][*1*]. For example, weight w_{13} will be stored at *weights*[*hash*(*0, 0*)][*hash*(*1, 0*)] = *weights*[2*0+0][2*1 + 0] = *weights*[0][2] (because node 1 has indices (0,0) and node 3 has indices (1, 0) in two-dimensional vectors); and weight w_{46} will be stored in *weights*[*hash*(*1, 1*)][*hash*(*2, 1*)] = *weights*[2*1+1][2*2 + 1] = *weights*[3][5].