```cpp
#pragma once

#include <stdint.h>
#include <vector>
#include <string>
#include <fstream>

#include <boost/archive/xml_iarchive.hpp>
#include <boost/archive/xml_oarchive.hpp>
#include <boost/serialization/vector.hpp>

#include "GA.h"
#include "MathException.h"
#include "SoilMathTypes.h"
#include "FFT.h"

namespace SoilMath
{
    class NN
    {
    public:
        NN(uint32_t inputneurons, uint32_t hiddenneurons, uint32_t outputneurons);
        NN();
        virtual ~NN();

        Predict_t Predict(ComplexVect_t input);
        static Predict_t PredictLearn(ComplexVect_t input, Weight_t inputweights, Weight_t hiddenweights, uint32_t inputneurons,
          uint32_t hiddenneurons, uint32_t outputneurons);
        void SetInputWeights(Weight_t value) { iWeights = value; };
        void SetHiddenWeights(Weight_t value) { hWeights = value; };
        void SetBeta(float value) { beta = value; };

        void Learn(InputLearnVector_t input, OutputLearnVector_t cat, uint32_t noOfDescriptorsUsed );

        void SaveState(string filename);
        void LoadState(string filename);

        Weight_t iWeights;
        Weight_t hWeights;

    private:
```

```
41          std::vector<float> iNeurons;
42          std::vector<float> hNeurons;
43          std::vector<float> oNeurons;
44
45          uint32_t hiddenNeurons;
46          uint32_t inputNeurons;
47          uint32_t outputNeurons;
48          float beta;
49
50          bool studied = false;
51          friend class boost::serialization::access;
52          template <class Archive>
53          void serialize(Archive & ar, const unsigned int version)
54          {
55              ar & BOOST_SERIALIZATION_NVP(inputNeurons);
56              ar & BOOST_SERIALIZATION_NVP(hiddenNeurons);
57              ar & BOOST_SERIALIZATION_NVP(outputNeurons);
58              ar & BOOST_SERIALIZATION_NVP(iWeights);
59              ar & BOOST_SERIALIZATION_NVP(hWeights);
60              ar & BOOST_SERIALIZATION_NVP(beta);
61              ar & BOOST_SERIALIZATION_NVP(studied);
62          }
63
64      };
65  }
```