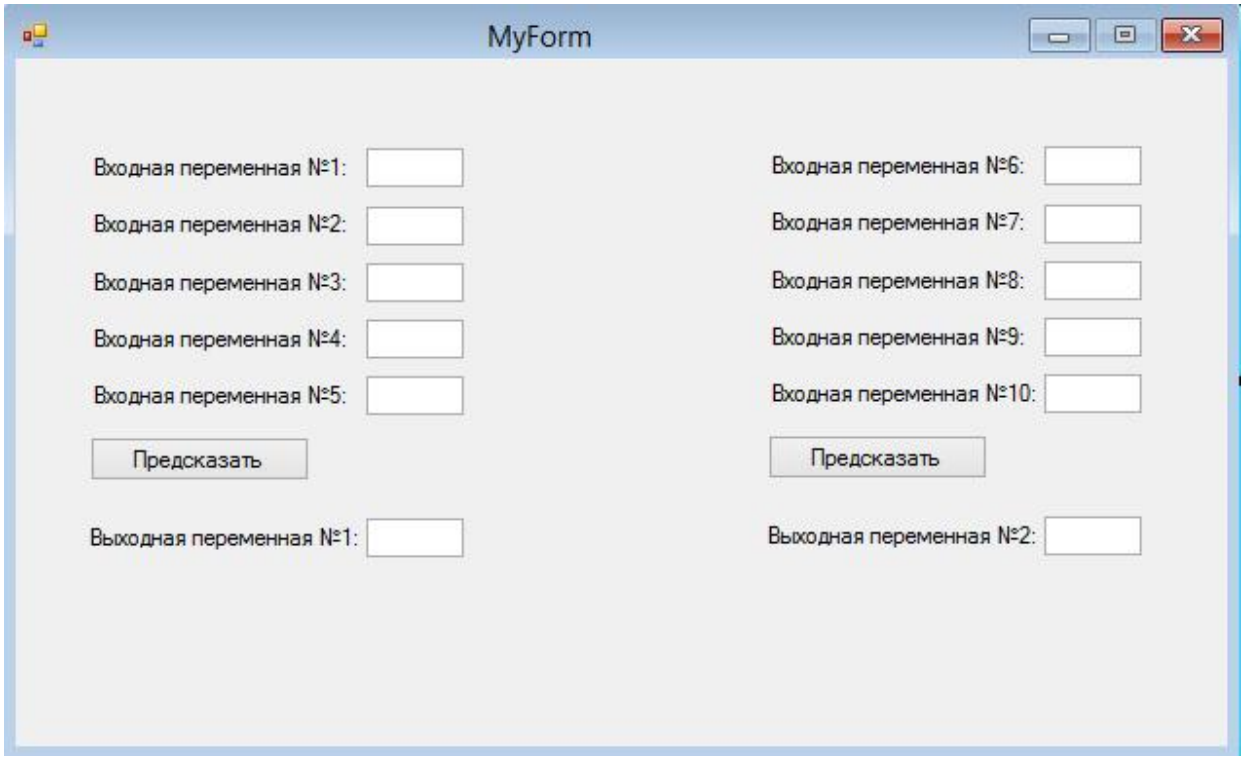

Примерный вид оконного приложения

Код первого консольного приложения:

```
//Analysis Type - Regression
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
#include <iostream>
#include <clocale>


double MLP5_4_1_input_hidden_weights[4][5] =
{
 {-3.13607837778505e+001, -3.01096042608715e+000, 1.03242212902347e+001, -1.86472411662761e+000, 6.96748325395164e+000 },
 {-6.06224264734459e-001, 4.11774854241150e+000, -1.29471976285977e+001, 6.34883696259913e+000, -6.84831424018010e+000 },
 {7.56390673202040e+000, 8.72850256071793e+000, 5.99740351097183e+000, -6.21364190444941e+000, -2.32371646178518e+000 },
 {8.78318955077932e+000, 2.05466204301580e+000, 8.97213948503104e-001, -4.12798464682834e+000, -3.92178399361126e-001 }
};

double MLP5_4_1_hidden_bias[4] = { 6.45465131904432e+000, 1.24980072908703e+001, -8.66765672968151e+000, 1.03425361273145e+000 };

double MLP5_4_1_hidden_output_wts[1][4] =
{
 {1.13577855675452e+000, 2.37836057439123e+000, 2.95426898724852e+000, 4.58618674308364e+000 }
};

double MLP5_4_1_output_bias[1] = { -4.64584893638802e+000 };

double MLP5_4_1_max_input[5] = { 2.70000000000000e+001, 5.78000000000000e-001, 2.53000000000000e+001, 1.47000000000000e+004,
7.00000000000000e+000 };

double MLP5_4_1_min_input[5] = { 3.00000000000000e+000, 3.00000000000000e-002, 1.22000000000000e+001, 3.60000000000000e+003,
2.00000000000000e+000 };

double MLP5_4_1_max_target[1] = { -1.43000000000000e+000 };

double MLP5_4_1_min_target[1] = { -4.30000000000000e+000 };

double MLP5_4_1_input[5];
double MLP5_4_1_hidden[4];
double MLP5_4_1_output[1];

double MLP5_4_1_MeanInputs[5] = { 0.00000000000000e+000, 0.00000000000000e+000, 0.00000000000000e+000, 0.00000000000000e+000,
0.00000000000000e+000 };

void MLP5_4_1_ScaleInputs(double* input, double minimum, double maximum, int size)
{
	double delta;
	long i;
```

```
	for (i = 0; i < size; i++)
	{
		delta = (maximum - minimum) / (MLP5_4_1_max_input[i] - MLP5_4_1_min_input[i]);
		input[i] = minimum - delta * MLP5_4_1_min_input[i] + delta * input[i];
	}
}

void MLP5_4_1_UnscaleTargets(double* output, double minimum, double maximum, int size)
{
	double delta;
	long i;
	for (i = 0; i < size; i++)
	{
		delta = (maximum - minimum) / (MLP5_4_1_max_target[i] - MLP5_4_1_min_target[i]);
		output[i] = (output[i] - minimum + delta * MLP5_4_1_min_target[i]) / delta;
	}
}

double MLP5_4_1_logistic(double x)
{
	if (x > 100.0) x = 1.0;
	else if (x < -100.0) x = 0.0;
	else x = 1.0 / (1.0 + exp(-x));
	return x;
}

void MLP5_4_1_ComputeFeedForwardSignals(double* MAT_INOUT, double* V_IN, double* V_OUT, double* V_BIAS, int size1, int size2, int layer)
{
	int row, col;
	for (row = 0;row < size2; row++)
	{
		V_OUT[row] = 0.0;
		for (col = 0;col < size1;col++)V_OUT[row] += (*(MAT_INOUT + (row * size1) + col) * V_IN[col]);
		V_OUT[row] += V_BIAS[row];
		if (layer == 0) V_OUT[row] = tanh(V_OUT[row]);
		if (layer == 1) V_OUT[row] = MLP5_4_1_logistic(V_OUT[row]);
	}
}

void MLP5_4_1_RunNeuralNet_Regression()
{
	MLP5_4_1_ComputeFeedForwardSignals((double*)MLP5_4_1_input_hidden_weights, MLP5_4_1_input, MLP5_4_1_hidden, MLP5_4_1_hidden_bias, 5,
4, 0);
	MLP5_4_1_ComputeFeedForwardSignals((double*)MLP5_4_1_hidden_output_wts, MLP5_4_1_hidden, MLP5_4_1_output, MLP5_4_1_output_bias, 4, 1,
1);
}

int main()
{
```

```
	setlocale(LC_ALL, "Rus");

	int cont_inps;
	int i = 0;
	int keyin = 1;
	while (1)
	{
		printf("\n%s\n", "Enter values for Continuous inputs (To skip a continuous input please enter -9999)");
		printf("%s", "Cont. Input-0(Входная переменная №1): ");// Если значение переменной 8
				scanf_s("%lg", &MLP5_4_1_input[0]);
		printf("%s", "Cont. Input-1(Входная переменная №2): ");// Если значение переменной 0,073
		scanf_s("%lg", &MLP5_4_1_input[1]);
		printf("%s", "Cont. Input-2(Входная переменная №3): ");// Если значение переменной 14,8
		scanf_s("%lg", &MLP5_4_1_input[2]);
		printf("%s", "Cont. Input-3(Входная переменная №4): ");// Если значение переменной 10200
		scanf_s("%lg", &MLP5_4_1_input[3]);
		printf("%s", "Cont. Input-4(Входная переменная №5): ");// Если значение переменной 4,5
		scanf_s("%lg", &MLP5_4_1_input[4]);
		for (cont_inps = 0;cont_inps < 5;cont_inps++)
		{
			//Substitution of missing continuous variables
			if (MLP5_4_1_input[cont_inps] == -9999)
				MLP5_4_1_input[cont_inps] = MLP5_4_1_MeanInputs[cont_inps];
		}
		MLP5_4_1_ScaleInputs(MLP5_4_1_input, 0, 1, 5);
		MLP5_4_1_RunNeuralNet_Regression();
		MLP5_4_1_UnscaleTargets(MLP5_4_1_output, 0, 1, 1);
		printf("\n%s%.14e", " Predicted Output of Выходная переменная = ", MLP5_4_1_output[0]);// То значение переменной -3,89
		printf("\n\n%s\n", "Press any key to make another prediction or enter 0 to quit the program.");
		keyin = _getch();
		if (keyin == 48)break;
	}
	return 0;
}
```

Код второго консольного приложения:

```
//Analysis Type - Regression
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
#include <iostream>
#include <clocale>



double MLP_5_4_1_input_hidden_weights[4][5] =
{
 {-8.33317387264603e-001, -6.70554624534129e-001, 3.09761676361377e+000, -2.48958302493199e-001, -3.54657848453916e+000 },
 {-1.13157858502933e+001, 7.71842587001804e+000, -1.92003714594431e+000, 5.96311620330257e+000, -4.25633637934041e+000 },
 {-1.41734981196917e+001, 7.17454194026011e+000, -2.62225184841414e+000, 7.14401610724097e+000, -4.23121514526772e+000 },
 {5.19963799659024e+000, -5.63189634756797e-001, 3.64528529788290e+000, 2.34290568799957e+000, -1.24549567578635e+000 }
};

double MLP_5_4_1_hidden_bias[4] = { -2.58835460229572e+000, 3.69479027878497e-002, 3.46651677539524e-001, -5.98309734016395e+000 };

double MLP_5_4_1_hidden_output_wts[1][4] =
{
 {-7.36338615041376e-001, 1.28400064302483e+000, -1.19720218099090e+000, 9.88608658594499e-001 }
};

double MLP_5_4_1_output_bias[1] = { 1.54099576879180e-001 };

double MLP_5_4_1_max_input[5] = { 1.38000000000000e+001, 1.47000000000000e+004, 5.15000000000000e-001, 3.50000000000000e+000,
3.00000000000000e+001 };

double MLP_5_4_1_min_input[5] = { 6.40000000000000e+000, 3.60000000000000e+003, 3.00000000000000e-002, 2.00000000000000e+000,
3.00000000000000e+000 };

double MLP_5_4_1_max_target[1] = { 2.83000000000000e+001 };

double MLP_5_4_1_min_target[1] = { 9.00000000000000e-001 };

double MLP_5_4_1_input[5];
double MLP_5_4_1_hidden[4];
double MLP_5_4_1_output[1];

double MLP_5_4_1_MeanInputs[5] = { 9.64313725490196e+000, 8.18235294117647e+003, 2.17666666666667e-001, 2.65843137254902e+000,
1.23980392156863e+001 };

void MLP_5_4_1_ScaleInputs(double* input, double minimum, double maximum, int size)
{
	double delta;
```

```
	long i;
	for (i = 0; i < size; i++)
	{
		delta = (maximum - minimum) / (MLP_5_4_1_max_input[i] - MLP_5_4_1_min_input[i]);
		input[i] = minimum - delta * MLP_5_4_1_min_input[i] + delta * input[i];
	}
}

void MLP_5_4_1_UnscaleTargets(double* output, double minimum, double maximum, int size)
{
	double delta;
	long i;
	for (i = 0; i < size; i++)
	{
		delta = (maximum - minimum) / (MLP_5_4_1_max_target[i] - MLP_5_4_1_min_target[i]);
		output[i] = (output[i] - minimum + delta * MLP_5_4_1_min_target[i]) / delta;
	}
}

double MLP_5_4_1_logistic(double x)
{
	if (x > 100.0) x = 1.0;
	else if (x < -100.0) x = 0.0;
	else x = 1.0 / (1.0 + exp(-x));
	return x;
}

void MLP_5_4_1_ComputeFeedForwardSignals(double* MAT_INOUT, double* V_IN, double* V_OUT, double* V_BIAS, int size1, int size2, int layer)
{
	int row, col;
	for (row = 0;row < size2; row++)
	{
		V_OUT[row] = 0.0;
		for (col = 0;col < size1;col++)V_OUT[row] += (*(MAT_INOUT + (row * size1) + col) * V_IN[col]);
		V_OUT[row] += V_BIAS[row];
		if (layer == 0) V_OUT[row] = MLP_5_4_1_logistic(V_OUT[row]);
	}
}

void MLP_5_4_1_RunNeuralNet_Regression()
{
	MLP_5_4_1_ComputeFeedForwardSignals((double*)MLP_5_4_1_input_hidden_weights, MLP_5_4_1_input, MLP_5_4_1_hidden,
MLP_5_4_1_hidden_bias, 5, 4, 0);
	MLP_5_4_1_ComputeFeedForwardSignals((double*)MLP_5_4_1_hidden_output_wts, MLP_5_4_1_hidden, MLP_5_4_1_output, MLP_5_4_1_output_bias,
4, 1, 1);
}

int main()
{
```

```
	setlocale(LC_ALL, "Rus");

	int cont_inps;
	int i = 0;
	int keyin = 1;
	while (1)
	{
		printf("\n%s\n", "Enter values for Continuous inputs (To skip a continuous input please enter -9999)");
		printf("%s", "Cont. Input-0(Входная переменная №6): ");// Если значение переменной 10
		scanf_s("%lg", &MLP_5_4_1_input[0]);
		printf("%s", "Cont. Input-1(Входная переменная №7): ");// Если значение переменной 4200
		scanf_s("%lg", &MLP_5_4_1_input[1]);
		printf("%s", "Cont. Input-2(Входная переменная №8): ");// Если значение переменной 0,148
		scanf_s("%lg", &MLP_5_4_1_input[2]);
		printf("%s", "Cont. Input-3(Входная переменная №9): ");// Если значение переменной 2
		scanf_s("%lg", &MLP_5_4_1_input[3]);
		printf("%s", "Cont. Input-4(Входная переменная №10): ");// Если значение переменной 4
		scanf_s("%lg", &MLP_5_4_1_input[4]);
		for (cont_inps = 0;cont_inps < 5;cont_inps++)
		{
			//Substitution of missing continuous variables
			if (MLP_5_4_1_input[cont_inps] == -9999)
				MLP_5_4_1_input[cont_inps] = MLP_5_4_1_MeanInputs[cont_inps];
		}
		MLP_5_4_1_ScaleInputs(MLP_5_4_1_input, 0, 1, 5);
		MLP_5_4_1_RunNeuralNet_Regression();
		MLP_5_4_1_UnscaleTargets(MLP_5_4_1_output, 0, 1, 1);
		printf("\n%s%.14e", "Predicted Output of Выходная переменная №2 = ", MLP_5_4_1_output[0]);// То значение переменной 5,34
		printf("\n\n%s\n", "Press any key to make another prediction or enter 0 to quit the program.");
		keyin = _getch();
		if (keyin == 48)break;
	}
	return 0;
}
```