Kambiz Majidzadeh
Institute of Applied Mathematics, Baku State University, Baku, Azerbaijan
kambiz823@yahoo.com

APPLICATION OF ARTIFICIAL NEURAL NETWORK IN OPTIMAL PURSUIT PROBLEM WITH RESPECT TO DOMAIN

In the present paper, the convex domain's space is constructed and a scalar product is introduced. The derivative of a domain function in this space is defined. Using this approach, a method is proposed to investigate an optimal control pursuit problem with respect to domain. Unlike the traditional problem, here the controller and trajectory are a domain at each moment of time. In other words, controller and trajectory are a domain function. At first we show the existence of the solution of Cauchy problem where the process is described, and then prove the maximum principle for the considered optimal control problem. Using the obtained results, we offer an algorithm for its numerical solution and train it to neural network.

Keywords: *support function, optimal pursuit, artificial neural network, network training.*

1. Introduction

Wide classes of practical problems are reduced to studying the change of the form of the considered object or body in time. Diffusion processes, the processes of ecology and biology, the problems on change of the form of a body subject to temperature, different problems of elasticity theory and etc. are among these problems [1, 2]. As a rule, while investigating these problems, the change of separate points of a body in time is studied. But, sometimes, a great importance is directly change of the form of a body instead of the change of separate points. Besides, while studying the change of separate points of a body, there arises necessity to accept some additional assumptions. This reduces to inadequacy of the constructed mathematical model of the process and real description of the object under investigation.

Here, we use a new variation notion of domains given in the paper [3] and define the change of the form of a body in time, i.e. define the change velocity of domain. Therefore, this enables to write equations characterizing the motion of domain. Thus, if in traditional mathematical models the process was characterized by the motion equations of a point, in the second approach the process is characterized by the motion equation of the domain. Our aim is to develop the traditional pursuit problem to the considered case and using the results obtained for this problem to create neural networks enabling to study such problems.

2. Space of convex domains

We define a space of domains at first. Denote by M the set of all convex bounded domains $D \subset R^n$.

Consider the addition and multiplication operations by non-negative number in M for any $A, B \in M$ as follows:

$$A + B = \{c = a + b;\ a \in A,\ b \in B\},$$
$$\lambda A = \{\lambda a:\ a \in A\},\ \lambda \geq 0.$$

Note that M is not a linear space (the subtraction operation is not defined in M). Let's take the pairs $(A, B) \in M \times M$ and define the operations:

$$(A_1, B_1) + (A_2, B_2) = (A_1 + A_2,\ B_1 + B_2),$$
$$\lambda(A, B) = (\lambda A,\ \lambda B),\ \ if\ \ \lambda \geq 0,$$
$$(-1)\cdot(A, B) = (B, A),$$
$$(A, B) \approx (C, D)\quad if\quad A + D = B + C$$

As a zero element of this space we take the pair $(0,0)$, i.e. the set of elements $(A,\ A)$, $A\in M$. The set of all pairs $(A,B)\in M\times M$ forms a structure of a linear space. Below we will introduce a scalar product on this space. For any $A\in M$ the function

$$P_A(x)=\sup_{l\in A}(l,x),\ x\in R^m \tag{1}$$

is called a support function of the set A. This function is continuously convex and positive-homogeneous, i.e. $P_A(\lambda x)=\lambda P_A(x),\ \lambda\geq 0$. Moreover, it is known [4] that for each continuously convex positive-homogeneous function $P(x)$ there exists a convex bounded set $A\in M$ such that

$$P\ (x)=P_A(x).$$

The set $A\in M$ is reconstructed as a subdifferential of the function $P_A(x)$ at the origin [4], i.e.

$$A=\partial P_A(0)=\{l\in R^m : P_A(x)\geq (l,x), x\in R^m\}. \tag{2}$$

Let's take any $a=(A_1,A_2),\ \ b=(B_1,B_2),\ \ A_i,B_i\in M,\ i=1,2$, and define the scalar product as

$$a\bullet b=\int_{S_B} p(x)q(x)ds. \tag{3}$$

Here, $p(x)=p_{A_1}(x)-p_{A_2}(x)$, $g(x)=p_{B_1}(x)-p_{B_2}(x)$, where $p_{A_i}(x), p_{B_i}(x)$ are support functions of the sets A_i, B_i, respectively; S_B is a surface of a unit ball B. It is natural to define a norm in this space as

$$\| a\|_{ML_2}=\sqrt{a.a}=\left(\int_{S_B}\left(p_{A_1}(x)-p_{A_2}(x)\right)^2 ds\right)^{1/2}. \tag{4}$$

Note that for the one-dimensional case $n=1$, the formula (3) turns to

$$a\bullet b=\ p(-1)g(-1)+\ p(1)g(1).$$

It can be shown that this definition satisfies all requirements of a scalar product. We denote this space by ML_2. The distance between the sets $A\in M$ and $B\in M$ is defined as the norm of

$$a=(A,0)-(B,0)=(A,B).$$

Let at the time $t\in[0,T]$ the area under investigation be in the form $D(t)$. We define the speed of the evolution of the domain $D(t)$ as:

$$\frac{\partial P_{D(t)}(x)}{\partial t}=\lim_{\Delta t\to 0}\frac{P_{D(t+\Delta t)}(x)-P_{D(t)}(x)}{\Delta t},\quad x\in S_B.$$

If there exist the domains $V_1(t),\ V_2(t)\in M,\ \ t\in[0,T]$ such that

$$\frac{\partial P_{D(t)}(x)}{\partial t}=P_{V_1(t)}(x)-P_{V_2(t)}(x), \tag{5}$$

we write $\dot d(t)=(V_1(t),V_2(t))\in M\times M$. For instance, if $D(t)=B_t$ is a ball of radius t, with the center at the origin, then $P_{D(t)}=t\cdot\|x\|$. In this case, $\dot D(t)=(B_1,0)$. If $D(t)$ is defined as follows

$$D(t)=\{(x_1,x_2):0\leq x_1\leq at, 0\leq x_2\leq bt\},$$

then $\dot d(t)=(D(1),0)$.

For any t let's consider the pair $d(t)=(D_1(t),\ D_2(t))\in M\times M$. Writing $d(t)=(D_1(t),0)-(D_2(t),0)$, provided $\dot d_1(t),\dot d_2(t)\in M\times M$ we can define

$$\dot d(t)=\dot d_1(t)-\dot d_2(t)\in M\times M.$$

3. Problem statement

Now let's assume that the motion of the object under investigation is described by the following system of equations:

$$\dot{Y}(t) = a_1(t)Y(t) + b_1(t)V(t), \quad Y(0) = Y_0, \tag{6}$$

$$\dot{Z}(t) = a_2(t)Z(t) + b_2(t)V(t), \quad Z(0) = Z_0. \tag{7}$$

Here, $a_i(t), b_i(t), i = 1,2$, are the given functions, $Y_0, Z_0 \in M_0$ are the given closed convex bounded sets, $V(t)$ is a controller and a closed convex bounded set for each $t \in [0,T]$. This function is said to be a domain function. We'll show that under certain conditions the solution of problem (6)–(7) is also a closed convex bounded set. By U denote a set of controls $V(t) \in V_0$, $t \in [0,T]$ such that $\|V(t)\| \in L_2(0,T)$. In other words,

$$U = \{V = V(t) \in U_0, \ \overset{\circ}{\forall} t \in [0,T], \ \|V(t)\| \in L_2(0,T)\}. \tag{8}$$

Here, U_0 is a convex subset of the class of a closed convex bounded sets M.

Our aim is to find a domain function $V(t) \in U$ giving minimum to the following functional

$$J(v) = \|Y(T) - Z(T)\|^2 + \mu \int_0^T \|V(t)\|^2 dt \to \min, \tag{9}$$

where, $\mu \geq 0$ is a given number.

We consider the existence of a solution from the class of closed bounded domain functions of problem (6), (7) for arbitrary $V(t) \in U$ at first.

Theorem 1. Let $b_i(t) \geq 0, \ i = 1, 2, \ t \in [0,T]$. Then, for arbitrary $V = V(t) \in U$ problem (6), (7) has a unique solution in the class of closed convex bounded domain functions.

Proof. Take an arbitrary $V = V(t) \in U$ and consider the following function

$$P(x;t) = \exp\left(\int_0^t a_1(\tau)d\tau\right)\left[P_{Y_0}(x) + \int_0^t b_1(\tau)\exp\left(-\int_0^\tau a_1(s)ds\right)P_{V(t)}(x)d\tau\right]. \tag{10}$$

By verifying we can see that the function $P(x;t)$ is a solution of the problem

$$\frac{\partial P(x;t)}{\partial t} = a_1(t)P(x;t) + b_1(t)P_{V(t)}(x)), \quad P(x;0)) = P_{Y_0}(x). \tag{11}$$

As $V(t)$ is a convex set for Y_0 and each $t \in [0,T]$, their support functions $P_{Y_0}(x), \ P_{V(t)}(x)$ are convex and homogeneous functions.

Show that if $a(t) \geq 0$, then $\int_0^t a(s)P_{V(s)}(x)ds$ is convex with respect to $x \in R^m$, Indeed, for an arbitrary $x_1, x_2 \in R^m$,

$$\int_0^t a(s)P_{V(s)}\left(\frac{x_1 + x_2}{2}\right)ds \leq \int_0^t a(s)\left[\frac{1}{2}P_{V(s)}(x_1) + \frac{1}{2}P_{V(s)}(x_2)\right]ds =$$

$$= \frac{1}{2}\int_0^t a(s)P_{V(s)}(x_1)ds + \frac{1}{2}\int_0^t a(s)P_{V(s)}(x_2)ds.$$

Taking this into account, it is clear that the function $P(x;t)$ determined as (10) is a convex homogeneous function. Then, there exists a convex, bounded domain function $Y(t)$ such that

$P_{Y(t)}(x) = P(x;t)$. Taking into account relation (11), it is clear that $Y(t)$ is a solution of problem (6). In the same way, we can show that problem (7) has a unique solution. The theorem is proved.

4. Maximum principle

Assume $\Psi = (\Psi_1(t), \Psi_2(t))$ is a solution of the following problem

$$\dot{\Psi}_1(t) = -a_1(t)\Psi_1(t), \quad \Psi_1(T) = -2[Y(T) - Z(T)], \tag{12}$$

$$\Psi_2(t) = -a_2(t)\Psi_2(t), \quad \Psi_2(T) = -2[Y(T) - Z(T)]. \tag{13}$$

Problem (12), (13) is a problem conjugated to problem (6)–(9). Since $Y(T) \in M$, $Z(T) \in M$, in the same way we can show that $\Psi_1(t) \in M \times M$, $\Psi_2(t) \in M \times M$, $t \in [0,T]$. Now, for the given problem prove the following theorem that is the analogy of the maximum principle.

Theorem 2. Let $V^*(t) \in M$ be an optimal solution of problem (6)–(9). Then, the following relations are satisfied for almost all $t \in (0,T)$,

$$\begin{aligned} & b_1(t)v^*(t) \bullet \Psi_1^*(t) + b_2(t)V^*(t) \bullet \Psi_2^*(t) - \mu\|V^*(t)\|^2 = \\ & = \max_{V \in U_0}[b_1(t)V \bullet \Psi_1^*(t) + b_2(t)V \bullet \Psi_2^*(t) - \mu\|V\|^2], \ \overset{\circ}{\forall} t \in (0,T), V \in U_0. \end{aligned} \tag{14}$$

Here, $\Psi_i^* = \Psi_i^*(t), i = 1,2$ is a solution of conjugated problem (12)–(13) corresponding to $V^* = V^*(t)$.

Proof. Take the two possible variants as $\{V(t), Y(t), Z(t)\}, \{\overline{V}(t), \overline{Y}(t), \overline{Z}(t)\}$ and make the following denotation

$$\Delta Z(t) = \overline{Z}(t) - Z(t), \ \Delta Y(t) = \overline{Y}(t) - Y(t), \ \Delta V(t) = \overline{V}(t) - V(t). \tag{15}$$

It is clear that the following relations will be satisfied:

$$\Delta\dot{Y}(t) = a_1(t)\Delta Y(t) + b_1(t)\Delta V(t), \quad \Delta Y(0) = 0, \tag{16}$$

$$\Delta\dot{Z}(t) = a_2(t)\Delta Z(t) + b_2(t)\Delta V(t), \quad \Delta Z(0) = 0, \tag{17}$$

Multiply equation (16) by the functions $\Psi_1(t) \in M \times M$, and equation (17) by the functions $\Psi_2(t) \in M \times M$, $\forall t \in [0,T]$, integrate and sum. Then

$$\begin{aligned} & \int_0^T [\Delta\dot{Y}(t) \bullet \Psi_1(t) - a_1(t)\Delta Y(t) \bullet \Psi_1(t) - b_1(t)\Delta V(t) \bullet \Psi_1(t)]dt + \\ & + \int_0^T [\Delta\dot{Z}(t) \bullet \Psi_2(t) - a_2(t)\Delta Z(t) \bullet \Psi_2(t) - b_2(t)\Delta V(t) \bullet \Psi_2(t)]dt = 0. \end{aligned}$$

Taking into account initial conditions we get:

$$\begin{aligned} & \Delta Y(T) \bullet \Psi_1(T) - \int_0^T [\Delta Y(t) \bullet \dot{\Psi}_1(t) + a_1(t)\Delta Y(t) \bullet \Psi_1(t))]dt - \\ & - \int_0^T b_1(t)\Delta V(t) \bullet \Psi_1(t)dt + \Delta Z(T) \bullet \Psi_2(T) - \\ & - \int_0^T [\Delta Z(t) \bullet \Psi_2(t) + a_2(t)\Delta Z(t) \bullet \Psi_2(t))]dt - \int_0^T b_2(t)\Delta V(t) \bullet \Psi_2(t)dt = 0. \end{aligned} \tag{18}$$

Now, calculate the increment of functional (9)

$$\Delta J \equiv J(\overline{V}) - J(V) = 2(Y(T) - Z(T)).[\Delta Y(T) - \Delta Z(T)] + \\ + \mu\int_0^T \left\|\overline{V}(t)\right\|^2 dt - \mu\int_0^T \left\|V(t)\right\|^2 dt + \max_{t\in[0,T]}\left\|\Delta Y(t)\right\| \quad . \tag{19}$$

Here, we make denotation $\left\|\Delta Y(t)\right\| = \left\|(\Delta Y(t),0)\right\|$. Add relation (18) to (19), (Take into account that right hand side is equal zero). Then

$$\Delta J = 2(Y(T) - Z(T)) \bullet [\Delta Y(T) - \Delta Z(T)] + \Delta Y(T) \bullet \Psi_1(T) - \\ - \int_0^T \left[\Delta Y(t) \bullet \dot{\Psi}_1(t) + a_1(t)\Delta Y(t) \bullet \Psi_1(t))\right]dt - \int_0^T b_1(t)\Delta V(t) \bullet \Psi_1(t)dt + \\ + \Delta Z(T) \bullet \Psi_2(T) - \int_0^T \left[\Delta Z(t) \bullet \dot{\Psi}_2(t) + a_2(t)\Delta Z(t) \bullet \Psi_2(t))\right]dt - \\ - \int_0^T b_2(t)\Delta V(t) \bullet \Psi_2(t)dt + \mu\int_0^T \left\|\overline{V}(t)\right\|^2 dt - \mu\int_0^T \left\|V(t)\right\|^2 dt + o\left(\max_{t\in[0,T]}\left\|\Delta Y(t)\right\|\right) = 0.$$

Take into account that the functions $\Psi_1 = \Psi_1(t),\ \Psi_2 = \Psi_2(t)$ are the solutions of problem (12)–(13):

$$\Delta J = \mu\int_0^T \left\|\overline{V}(t)\right\|^2 dt - \mu\int_0^T \left\|V(t)\right\|^2 dt - \int_0^T b_1(t)\overline{V}(t) \bullet \Psi_1(t)dt + \int_0^T b_1(t)V(t) \bullet \Psi_1(t)dt - \\ - \int_0^T b_2(t)\overline{V}(t) \bullet \Psi_2(t)dt + \int_0^T b_2(t)V(t) \bullet \Psi_2(t)dt + \int_0^T o\left(\max_{t\in[0,T]}\left\|\Delta Y(t)\right\|\right)dt\ . \tag{20}$$

From the form (10) of problem (6)–(7) it is clear that:

$$\left|P_{Y(t)}(x;\overline{V}) - P_{Y(t)}(x;V)\right| \le C_1\int_0^t \left|P_{\overline{V}(t)}(x) - P_{V(t)}(x)\right|d\tau\ ,$$

$$\left|P_{Z(t)}(x;\overline{V}) - P_{Z(t)}(x;V)\right| \le C_2\int_0^t \left|P_{\overline{V}(t)}(x) - P_{V(t)}(x)\right|d\tau\ ,$$

Here and further, by $C_i = const,\ i = 1,2,...$ we'll denote positive constants.

Take this into account in (20) we get:

$$\max_{t\in[0,T]}\left\|\Delta Y(t)\right\|^2 \le C_3\left(\int_0^T \left|\Delta V(t)\right|dt\right)^2,\ \ \max_{t\in[0,T]}\left\|\Delta Z(t)\right\|^2 \le C_3\left(\int_0^T \left|\Delta V(t)\right|dt\right)^2. \tag{21}$$

Now, let $\{V^*(t), Y^*(t), Z^*(t)\}$ be a solution of problems (6)–(9). Then

$$\Delta J = \mu\int_0^T \left\|V(t)\right\|^2 dt - \mu\int_0^T \left\|V^*(t)\right\|^2 dt - \int_0^T b_1(t)V(t) \bullet \Psi_1(t)dt + \\ - \int_0^T b_2(t)V(t) \bullet \Psi_2(t)dt + \int_0^T b_1(t)V(t) \bullet \Psi_1(t)dt + \\ + \int_0^T b_2(t)V^*(t) \bullet \Psi_2(t)dt + \int_0^T o\left(\max_{t\in[0,T]}\left\|\Delta Y(t)\right\|\right)dt \ge 0\ ,\quad \forall V \in U. \tag{22}$$

Let $t \in [0,T]$ be the Lebesgue points set [9] of the function

$$B(t)V^*(t) \bullet \psi^*(t) - \beta_1\left\|V^*(t)\right\|^2 - B(t)V(t) \bullet \psi^*(t) + \beta_1\left\|V(t)\right\|^2.$$

Take any number $\varepsilon > 0$ and $V \in U_0$. Select the following domain control:

$$V(\tau) = \begin{cases} V, & \tau \in [t, t+\varepsilon] \\ V^*(\tau), & \tau \in [0,T] \setminus [t, t+\varepsilon] \end{cases} .$$

It is clear from (8) that $V = V(t) \in U$. Then, from (21) we get:

$$\left\| Y(t) - Y^*(t) \right\|_{CF}^2 \le \varepsilon C_3 \int_t^{t+\varepsilon} \left\| V(t) - V^*(t) \right\|^2 dt ,$$

$$\left\| Z(t) - Z^*(t) \right\|_{CF}^2 \le \varepsilon C_3 \int_t^{t+\varepsilon} \left\| V(t) - V^*(t) \right\|^2 dt .$$

Here,

$$\|Y\|_{CF} = \max_{x \in S_B} |P_Y(x)|$$

Take this into account in (22). Then, divide

$$\Delta J = \mu \int_t^{t+\varepsilon} \|V\|^2 dt - \mu \int_t^{t+\varepsilon} \left\|V^*(t)\right\|^2 dt - \int_t^{t+\varepsilon} b_1(t) V.\Psi_1(t) dt + \int_t^{t+\varepsilon} b_1(t) V^*(t).\Psi_1(t) dt - \int_t^{t+\varepsilon} b_2(t) V.\Psi_2(t) dt +$$

$$+ \int_t^{t+\varepsilon} b_2(t) V^*(t).\Psi_2(t) dt + o(\varepsilon) \ge 0 , \quad \forall V \in U_0$$

into $\varepsilon \to 0$ and get relation (14). The theorem is proved.

5. Algorithm for numerical solution

For pursuit problems (6)–(9) we obtained an optimality condition in the form (14). This condition is an analogy of the maximum principle for the problem under investigation. Using the obtained optimality condition, give an algorithm for numerical solution of problems (6)–(9).

Step 1. Take the initial domain $V^{(0)} = V^{(0)}(t) \in U$, solve problems (6)–(7) and find the domain functions $Y^{(0)} = Y^{(0)}(t), Z^{(0)} = Z^{(0)}(t)$.

Step 2. Taking these solutions into account in problems (12), (13), we find the solutions $\Psi_1^{(0)} = \Psi_1^{(0)}(t),\ \Psi_2^{(0)} = \Psi_2^{(0)}(t)$ of the conjugated problem.

Step 3. Having solved the following problem

$$\int_0^T \left[b_1(t)\Psi_1^{(0)}(t) + b_2(t)\Psi_2^{(0)}(t) - 2\mu V^{(0)}(t) \right] V(t) \to \max,\ V = V(t) \in U , \tag{23}$$

we find the domain function $\overline{V}^{(0)}(t)$.

Step 4. The next domain function $V^{(1)} = V^{(1)}(t)$ is found from the following relation:

$$V^{(1)} = (1-\alpha)\overline{V}^{(0)} + \alpha V^{(0)}, \quad 0 < \alpha < 1.$$

Here, α may be selected by different ways. So, our main aim is to provide completion of the following monotonicity condition

$$J(V^{(1)}) \le J(V^{(0)}) .$$

The process is continued until some accuracy condition is satisfied. The accuracy condition may be as

$$\left| J(V^{(1)}) - J(V^{(0)}) \right| < \varepsilon ,$$

$$\left\| V^{(1)} - V^{(0)} \right\| < \varepsilon$$

and etc. As the suggested method is similar to the conditional gradient method, under certain conditions its construction is shown in [10].

As it is seen, at each iteration, problems (6), (7) are solved in the second step, problems (12), (13) in the third step. To solve these problems discretize them. For that partition the segment $[0,T]$ into n parts by the step $h=\frac{T}{n}$. In this case, we can write (6), (7) in the following way:

$$Y(i+1)=(1+ha_1(i))Y(i)+hb_1(i)V(i), \quad Y(0)=Y_0, \tag{24}$$

$$Z(i+1)=(1+ha_2(i))Z(i)+hb_2(i)V(i), \quad Z(0)=Z_0,\ i=\overline{1,n}. \tag{25}$$

Notice that unlike the traditional problems, here $V(i), Y(0), Z(0)$ are convex sets for each $i=\overline{1,n}$. As it is seen, as the quantity $h=\frac{T}{n}$ is small, and $b_1(t)\geq 0,\ b_2(t)\geq 0$ and each new domain $Y(i), Z(i)$ found by the new method stated above is the sum of convex domains, they also will be convex domains. Thus, discrete problems (24), (25) may be solved for each control $V(t)$ selected in the first step. Unlike problems (6), (7), the domain functions $\Psi_1(t),\ \Psi_2(t)$ being a solution of problems (12), (13) will not be convex for each $t\in[0,T]$. For that, equation (12) may be written in the form:

$$\begin{gathered}\dot{\Psi}_1^{(1)}(t)=-a_1(t)\Psi_1^{(1)}(t), \quad \Psi_1^{(1)}(T)=-2Y(T),\\ \dot{\Psi}_1^{(2)}(t)=-a_1(t)\Psi_1^{(2)}(t), \quad \Psi_1^{(2)}(T)=-2Z(T).\end{gathered} \tag{26}$$

Then, the solution of problem (12) may be shown in the form

$$\Psi_1(t)=\Psi_1^{(1)}(t)-\Psi_1^{(2)}(t).$$

Since $Y(T)$ and $Z(T)$ are convex domains, the domain functions $\Psi_1^{(1)}(t),\ \Psi_1^{(2)}(t)$ will be convex sets for each $t\in[0,T]$. In the same way, having written equation (13) in the form (26), we can show it as

$$\Psi_2(t)=\Psi_2^{(1)}(t)-\Psi_2^{(2)}(t).$$

Then, as it is seen, the functional (23) considered in the third step of iteration by the formula of scalar product (1) may be written in the form:

$$\int_{S_B}\int_0^T\left[b_1(t)P_{\Psi_1^{(0)}(t)}(x)+b_2(t)P_{\Psi_2^{(0)}}(x)-2\mu P_{V^{(0)}(t)}(x)\right] \quad P_{V(t)}(x)ds\rightarrow\max,\ V=V(t)\in U. \tag{27}$$

Here,

$$P_{\Psi_1\ (t)}(x)=P_{\Psi_1^{(1)}(t)}(x)-P_{\Psi_1^{(2)}(t)}(x)$$

$$P_{\Psi_2(t)}(x)=P_{\Psi_2^{(1)}(t)}(x)-P_{\Psi_2^{(2)}(t)}(x).$$

Denote

$$A_0(t,x)=b_1(t)P_{\Psi_1^{(0)}(t)}(x)+b_2(t)P_{\Psi_2^{(0)}}(x)-2\mu P_{V^{(0)}(t)}(x).$$

Then we can write problem (27) briefly in the form

$$\int_{S_B}\int_0^T A^{(0)}(t,x)P_{V(t)}(x)ds\rightarrow\max,\ V=V(t)\in U. \tag{28}$$

Thus, having solved discrete domain equation (24), (25) at each iteration we must find the maximum of the functional of the form (28). As equations (24), (25) are discrete, to minimize functional (28) we must discretize it. In this case, functional (28) may be written as follows:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}A_{ij}^{(0)}P_{ij}\rightarrow\max. \tag{29}$$

The condition in the set (8) may be written in different ways. For example, if $V = V(t) \in U_0$, this condition may be written as [5,6]

$$U_0 = \{V \subset R^2 : V_0 \subset V \subset V_1\}$$
$$P_0(x) \le P_V(x) \le P_1(x)$$

Here, V_0, V_1 are the given sets, $P_0(x), P_1(x)$ are their support functions, respectively. Discrete analogy of the considered condition may be written as

$$P_{ij}^{(0)} \le P_{ij} \le P_{ij}^{(1)},\ i = \overline{1,n},\ j = \overline{1,m} \tag{30}$$

6. Application of a neural network to numerical solution of the problem

Thus, in the second step of each iteration we must solve discrete domain equations (24),(25) and in the third step the linear programming problems in the form (29), (30). To solve these problems, many complexes of programs have been elaborated [9, 11]. Using these programs, the indicated problem may be solved. But, since discretization is carried out both with respect to t and x, the solution of problems requires a long iteration process. Therefore, in some cases, it is expedient to solve such problems by a neural network. In spite of the fact that in this case, the artificial neural network training process is too long, a neural network is created once and it finds the solution of the considered problem very quickly [10, 11]. A neural network may be applied to each of the mentioned problems or both of them simultaneously. In order to apply a neural network to the solution of such type problems we give some remarks concerning the training process of a neural network. For that, consider the linear programming problems (29), (30). For training the multi-layer neural network, we need input and output data. Therefore, changing the coefficients both in the functional and restrictions, we carry out training process in a neural network. Here we notice that it is not necessary to carry out the training process by changing all the coefficients, i.e. the data $A_{ij}^{(0)},\ P_{ij}^{(0)},\ P_{ij}^{(1)},\ i = \overline{1,n},\ j = \overline{1,m}$. It is possible to retain a part of coefficients as data of our problem and change the other coefficients. Though for obtaining these data we spend much time for iteration, we can use the programs concerning linear programming problems. But a great number of variables causes to some rough errors in the obtained answers. This may reduce to some errors in the solution given by the created neural network as well [7, 8].

Taking into account what has been said, to get the input and output data denote arbitrarily a part of problem data satisfying the restrictions. It is easy to do this under restrictions (30). For example, a part of coefficients is taken as a zero. Assume that in this case the obtained restrictions are not empty. Notice that if the sets V_0, V mentioned above are domains containing a zero, we can take the coefficients as a zero. Let $p_1, p_2, ..., p_s$ be new variables of the obtained small-dimensional problem. Then, we get the following small – dimensional problem:

$$\sum_{k=1}^{s} a_k^{(0)} p_k \to \max. \tag{31}$$

$$p_k^{(0)} \le p_k \le p_k^{(1)},\ k = \overline{1,s},. \tag{32}$$

Here, the numbers $a_k^{(0)},\ p_k^{(0)},\ p_k^{(1)}$ are the values with respect to the appropriate indices $A_{ij}^{(0)},\ P_{ij}^{(0)},\ P_{ij}^{(1)},\ i = \overline{1,n},\ j = \overline{1,m}$. Notice that taking an aggregate of different variables as a zero, we can get a great number of such small – dimensional problems. Here, having chosen arbitrarily the data $a_k^{(0)},\ p_k^{(0)},\ p_k^{(1)}$, we solve problems (31), (32) by the known methods and get a solution corresponding to these data. So, we get any number of input data consisting of the aggregate of

$a_k^{(0)}$, $p_k^{(0)}$, $p_k^{(1)}$, $k=\overline{1,s}$ and the same number of output data $p_1, p_2, \dots, p_s$ being a solution of appropriate problems (31), (32).

The used neural network is non-liner "*feed forward- distributed time delay-backpropagation*" with levenbery-marqwardt algorithm (train *LM*) contain of three layers:

1st layer is *Input layer* that contain *6 neuron* and *tansig* transfer function.
2nd layer is *Hidden layer* that contain *12 neuron* and *tansig* transfer function.
3rd layer is *Output layer* that contain *3 neuron* and *purelin* transfer function.

Now we can start to creating a neural network through command structure:

```
NET=newff (minmax(P),[6,12,3],{'tansig','tansig','purelin'},'trainlm');
NET=init (NET);
NET.trainparam.mc=0.8;
NET.trainparam.show=70;
NET.trainparam.lr_inc=1.03;
NET.trainparam.epochs=500;
NET.trainparam.lr=0.02;
NET.trainparam.goal=1e-5;
[NET,tr]=train (NET,P,T);
```

```
Network  =
   Neural Network object:
     architecture:
      numInputs: 1
      numLayers: 3
      biasConnect: [1; 1; 1]
      inputConnect: [1; 1; 1]
      layerConnect: [0 0 0; 1 0 0; 1 1 0]
      outputConnect: [0 0 1]
      numOutputs: 1  (read-only)
      numInputDelays: 0  (read-only)
      numLayerDelays: 0  (read-only)
     subobject structures:
      inputs: {1x1 cell} of inputs
      layers: {3x1 cell} of layers
      outputs: {1x3 cell} containing 1 output
      biases: {3x1 cell} containing 3 biases
      inputWeights: {3x1 cell} containing 3 input weights
      layerWeights: {3x3 cell} containing 3 layer weights
     functions:
      adaptFcn: 'trains'
      divideFcn: 'dividerand'
      gradientFcn: 'gdefaults'
      initFcn: 'initlay'
      performFcn: 'mse'
      plotFcns: {'plotperform','plottrainstate','plotregression'}
      trainFcn: 'trainlm'
     parameters:
      adaptParam: .passes
      divideParam: .trainRatio, .valRatio, .testRatio
```

gradientParam: (none)
initParam: (none)
performParam: (none)
trainParam: .show, .showWindow, .showCommandLine, .epochs,
.time, .goal, .max_fail, .mem_reduc,
.min_grad, .mu, .mu_dec, .mu_inc,

weight and bias values:

IW: {3x1 cell} containing 3 input weight matrices
LW: {3x3 cell} containing 3 layer weight matrices
b: {3x1 cell} containing 3 bias vectors

After initialize and adjust the train and weight parameter, the final created form of the artificial neural network is shown as (fig.1).

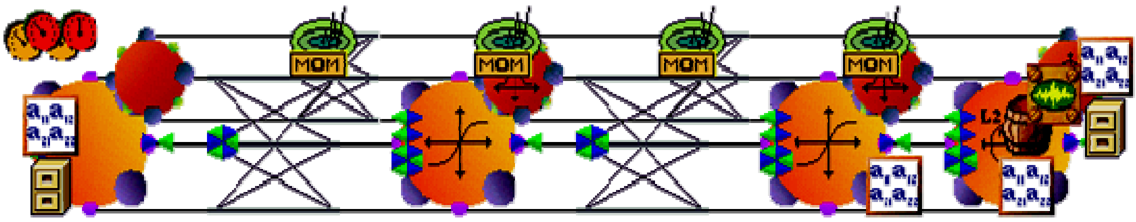

Fig.1. Network architecture

Here, we introduce the remaining part of output data as a zero. Re-initializing the training process corresponding to these data, we construct a neural network. To minimize the error, the number of data should be increased. Then we introduce the aggregate $A_{ij}^{(0)}$, $P_{ij}^{(0)}$, $P_{ij}^{(1)}$, $i=\overline{1,n}$, $j=\overline{1,m}$, to the constructed network as input variables.

As the result of training the neural network, it will give us the output P_{ij}, $i=\overline{1,n}$, $j=\overline{1,m}$. This output will be an approximate solution of problems (29), (30).

Notice that if we take into account that the class of vectors obtained as the result of taking separate groups of variables as a zero may play as a basis, we can say that the training process is perfect.

7. Conclusion

Using new approach, a method is proposed to investigate an optimal control pursuit problem with respect to domain. At first we considered the existence of the solution of Cauchy problem where the process is described, and then prove the maximum principle for the considered optimal control problem. Using the obtained results, we offer an algorithm for its numerical solution. After that, we trained such problems to create the neural networks. This allowed applying neural networks to solve the problem.

References

1. J.Sokolowski, J.-P.Zolesio, Introduction to shape optimization // Shape Sensitivity Analysis, Springer, Heidelberg, 1992.
2. J.Haslinger, R.A.E.Makinen, Introduction to shape optimization // theory, approximation and computation, SIAM, Philadelphia, 2003.
3. A.A.Niftiyev, C.I.Zeynalov, H.C.Efendiyeva, Optimal control problem relatively to domain evolution // International Journal of Applied Mathematics (IJAM). 2010, v.23, no.3, pp. 527-538.

4. V.F.Demyanov, A.M.Rubinov, Bases of non-smooth analysis and quasidifferential calculus, M.: “Nauka”, 1990.
5. Y.S.Gasimov, A.Nashaoui, A.A.Niftiyev, Nonlinear eigenvalue problems for p-laplacian // Optimization Letters, 2010, no.4, pp.67-84.
6. A.A.Niftiyev, E.R.Akhmadov, Variational statement of an inverse problem for a domain // Journal Differential equation, 2007, v.43, no.10, pp.1410-1416.
7. K.Majidzadeh, Application of neural networks to the problems on finding the form in dynamic processes // Transaction of Azerbaijan Nat. Acad. of Sciences, Information Technology, Baku, 2011, no.3, pp. 141-147.
8. R.M.Aliguliyev, K.Majidzadeh , Y.Ghasemi, Application of neural networks to finding optimal form // Problems of Information Technology, Baku, 2010, no.2, pp. 47-51.
9. F.P.Vasilyev, Optimization methods**.** Moscow, **“**Factorial Press”, 2002, 824 p.
10. Ward Cheney, Will Light A., Course in Approximation Theory. Brooks/Cole, 2000.
11. R.M.Alguliev, R.M.Aliguliyev, R.K.Alekperov, An approach to optimal task assignment in a distributed system // Journal Automation and Information Sciences, 2004, v.36, no.10, pp.51-55.

UOT 517.97; 519.68
Kambiz Məcidzadə
Bakı Dövlət Universitetinin Tətbiqi Riyaziyyat Elmi-Tədqiqat İnstitutu, Bakı, Azərbaycan
kambiz823@yahoo.com

Oblasta nəzərən optimal izləmə məsələsinə neyron şəbəkənin tətbiqi

Məqalədə əvvəlcə qabarıq çoxluqlar cütünün fəzası qurulur və bu fəzada skalyar hasil verilir. Bu metrikaya görə oblast funksiyanın törəməsi təyin olunur. Bundan istifadə edərək, oblasta nəzərən optimal izləmə məsələsi tədqiq olunur. Oblastların hərəkəti diferensial tənliklər sistemi ilə təsvir olunur. Sistemə təsir edən xarici qüvvəni elə seçmək tələb olunur ki, hər iki oblast zamanın verilmiş son anında bir-birinə yaxın olmuş olsunlar. Bu məsələ üçün maksimum prinsipinə analoji nəticə isbat olunur və bunun əsasında ədədi həll alqoritmi təklif olunur. Daha sonra, alınmış nəticələrdən istifadə edərək, belə məsələlərin neyron şəbəkələrin köməyi ilə həlli sxemi verilir.
Açar sözlər: *dayaq funksiyası, optimal izləmə, süni neyron şəbəkələr, şəbəkənin öyrədilməsi.*

УДК 517.97; 519.68
Маджид-заде Камбидж
kambiz823@yahoo.com
Научно-Исследовательский Институт Прикладной Математики при БГУ, Баку, Азербайджан

Применение нейронной сети к оптимальной задаче преследования по отношению области

В работе сначала строится пространство пары выпуклых областей и вводится скалярное произведение. По этой метрике определяется производная область функции. Используя это, рассматривается одна задача оптимального преследования относительно области. Движение области описывается системой дифференциальных уравнений. Требуется определить внешние силы так, чтобы их формы в конечном моменте времени были бы ближе друг-другу. Для этой задачи получен аналог принципа максимума и предложен алгоритм для ее решения. Далее, используя полученные результаты, задается схема применения нейронных сетей для решения рассматриваемой задачи.
Ключевые слова: *функция поддержки, оптимальное преследование, искусственная нейронная сеть, сетевое обучение.*