

PIETRO TERNA

目錄 (Table of Contents)

SWARM 中的经济仿真:

基于智能体建模与面向对象设计

Economic Simulations in Swarm:

Agent-Based Modelling and Object
Oriented Programming

[意] 弗兰西斯·路纳

(Francesco Luna)

[意] 本尼迪克特·史蒂芬森

(Benedikt Stefansson)

景体华 景旭 凌宁 等译

社会科学文献出版社

Social Sciences Documentation Publishing House

第三章 SWARM 在经济实验中的应用：关于智能体行为一致性自适应的神经网络方法

Pietro Terna

University of Turin

*Department of Economics and Finance G. Prato,
corso Unione Sovietica 218bis, 10134 Torino, Italy*

terna@econ.unito.it

摘要：我们的重点是基于智能体的模型在社会科学领域中的应用，并且关注由于模型结构所带来的主计算问题。为了简化这一任务，我们引入了一套规范化的环境——规则——智能体方案。最后，为了观察在没有预先规定行为规则情况下复杂性的涌现，在 SWARM 中我们引入了一种神经网络工具（交叉目标方法）。该方法对于建立人工实验室、并针对人工世界中的智能体的学习，自适应的一致性及其相互作用进行实验是有用的。

1. 引言

这一工作的目的是提供一种工具。在 SWARM 中所写的 BP-CT 软件包，为社会科学家进行仿真实验提供了一条捷径。

第 2 节：“社会科学领域中的智能体模型”，介绍了有关智能体模型的一些设想，强调它们在社会科学领域中的有用性，同时也考虑了它们在计算经济学广阔领域中的应用。

第 3 节：“ERA 方案”，处理由于模型结构所带来的主计算问题，引入了规范化的环境 - 规则 - 智能体方案 (ERA)，并初步用此方案创建了 BP-CT 软件包。

第 4 节：“神经网络”，简要解释了由处理单元构成的网络作为学习计算工具的思想。

第 5 节：“交叉目标方法 (CT)”。“我们展示了在神经网络算法基础上的人工智能体的结构，这些算法可以在没有先验行为规则的情况下，通过一个个反复实验过程来修正。

第 6 节：“SWARM 的 BP-CT 软件包”，解释了这一新的软件包。

第 7 节：“一个基于 CT 方案的行为与效果的实验，用以得到自发的 hayekian 市场”。我们建立并运行了一个实例，得到了一个无中介的 hayekian 市场。

第 8 节：关于“未来改进”的计划。

2. 社会科学领域中的智能体模型

选择基于智能体的模型范例，使我们进入了一个广阔的、未经开发的领域。这里的方法和技术大部分都是有待完善的。在这里我们应用自下而上的方法，把能够对环境或其他智能体的刺激

做出反应的软件段（我们的智能体）放在一起。那么，为什么要这么做呢？

从下面的网站上可以得到一个有趣的概览：

- (1) 人工生命与基于智能体的经济学读物摘要^①；
- (2) 基于智能体的计算经济学网站^②；
- (3) 基于智能体的经济学与人工生命简介^③；
- (4) Buenos Aries 大学的复杂系统^④；
- (5) 建立计算经济的模型^⑤；
- (6) 基于个体的模型^⑥；

我们还可以查阅的著作有：Conte (1997) 等人的社会仿真，Russel 和 Norvig (1995) 的人工智能文献，Beltratti (1996) 等人的经济学文献；还有伦敦经济与政治科学院的复杂性研究项目^⑦。

目前，基于智能体建模技术的最清晰的介绍来自于 1996 年 Epstein 和 Axtell 对 Sugarscape 模型作的介绍。Epstein 和 Axtell 这样写到：

Herbert Simon 热衷论证社会科学是一门事实上的硬科学。比如，许多具有决定性意义的社会过程是十分复杂的，它们不能简单的被分解为单独的子过程——经济学的、人口统计学的、文化的、空间的，分别加以分析，然后综合给出社会过程整体的恰当分析。然而，社会科学正是以这种方式

① <http://www.econ.iastate.edu/tesfatsi/sylalife.htm>.

② <http://www.econ.iastate.edu/tesfatsi/ace.htm>.

③ <http://www.econ.iastate.edu/tesfatsi/getalife.htm>.

④ <http://www.cea.uba.ar/aschu/complex.html>.

⑤ <http://zia.hss.cmu.edu/econ/>.

⑥ <http://hmt.com/cwr/ibm.html>.

⑦ <http://www.lse.ac.uk/lse/complex/>.

组成的，我们可以或多或少的将它分类细化为经济学，人口统计学，政治学等领域。

社会科学所以是硬科学，还因为某些控制实验方法是硬的。特别要指出的是，验证关于个人行为与宏观规律相互关系的假设是非常困难的。这个假设是：如果个人行为依照某种方式遵循特定规则，那么，整个社会将显示出一些特有的性质。可是，众多不同的微观个人行为又是怎样形成社会整体的宏观规律的呢？

另一个受到众多社会学家关注的基本问题是，理性的行为——一个完美的、掌握完整信息，有着超常计算能力的个体，往往将固定的（未展开的）外生效益函数极大化，而忽略了人的关系。然而，自然方法论并没有放宽关于个体的那些假设。

可以说，在模型构建中，抑制现实世界智能体的异质性是社会科学的通常做法。这方面，要么是明显的，如 Kirman 宏观经济学在 1992 年提出的具有代表性的代理人模型；要么是隐含的，当我们用高度集成的模型表示社会过程时，就是这样。这种模型在提供强大洞察力的同时，也滤掉了异质性的所有逻辑结果。几乎没有社会科学家能够否认这些逻辑结果的极端重要性，然而自然方法论迄今为止还没有系统地研究高度异质性的群体。

最后，可以明确地说，多数社会科学，特别是博弈论和一般均衡理论，一直专注于静态平衡，这自然忽略了时间的动态特性。而且，在强调这一点时，许多社会科学家都声称在社会系统中没有能够研究这种非均衡动态的自然方法论。

回答这一长段典型引语的，恰恰就是社会仿真和基于智能体的人工实验。这意味着我们有能力，以谨慎的态度从最基本的开

始，去构造智能体的互相作用，去观察他们行为中所出现的情况（如果真有情况出现的话）。从某种意义上说，这与我们在计算经济学这一广泛领域中探索时所完成的任务相同。计算经济学是对 Judd (1996) 的理论的补充，因为计算经济学验证和解释了一些理论问题，而这些问题涉及用实际的或假设的参数对方程进行数值解和迭代。但是，正如 Judd 指出的，我们在这里也有可能使用计算（或基于智能体的，我附加的）结果作为总的结论。原理如下：

计算机性能的提高为现今经济科学的开发提供了机遇与开发潜力。然而，像任何新工具一样，这里也存在许多关于经济学中计算的恰当作用的争论。有些作用是明显的，没什么争议的。大多数人都认为计算在经济分析当中是非常必要的，它还是一些政策的讨论提供了必要的指导。理论家们承认实例在说明一般结论时是有用的。然而，在研究一个经济理论时，当有人提出使用计算机和计算机生成的“实例”来替代传统的假设—理论—证明的分析模式的可能性时，争论就变得异常激烈了。在经济学中，计算的首要作用是标准研究方法的补充和有用的组成部分。而且，计算理论的活跃逐渐呈现出对传统理论分析的替代作用。在本文中，我们在提出“它们是补充还是替代”的同时，将着眼于研究计算方法在经济理论中的潜在作用和它们与标准理论分析的关系。

但是，怎样构建基于智能体的模型呢？最基本的是，我们必须构造 Gilbert 和 Terna, 1999a 那样的计算机模型，智能体（软件段）在这一模型中运行着（行为）。

首先，我们必须将智能体看做对象：这是一些包含数据和基

于这些数据进行运作的规则的程序段。这些规则提供了对来自对象外部的消息做出反应所必要的机制。

其次，我们必须通过每个智能体对象内部变量的状态以及同时产生于智能体整体行为的结果，来观察单个智能体的行为。因此，具有同步实验时钟就是软件的一个重要特征。我们必须确认：观测是建立在整体水平上的，而且所采集的关于智能体内部状态的信息，必须在实验的调度表上保持一致。

现在让我们假设：一群智能体，具有公共（所有智能体共同的）和个体信息（每个智能体的特殊的），以及简单的内在局部规则。如果将其视为一个整体，他们就有可能表现出有趣的、复杂的或“涌现的”行为。这里有两种涌现性：未曾预知的和不可预知的。未曾预知的涌现性的一个事例是，当我们寻找一种均衡状态时，却出现了一些周期性的行为；这一周期行为的决定性因素隐藏在模型构造中。不可预知涌现性的一个事例，发生在由实验产生的数据出现混沌的情况下。尽管这种混沌在真实社会科学现象中是可以明显观察到的，但在智能体仿真的复原机制下却不容易观察到。

3. ERA 方案

Swarm 是这种结构的一个自然候选方案，但是我们需要某种程度的标准化，主要是在由简单模型导致复杂效果的时候。这里，基本模型的结构对实验的有效性和可接受性起着至关重要的作用。由于这一原因，我们在此引入了一个用于建立智能体仿真的一般性方案。

图 3.1 显示出的 Environment-Rules-Agents (ERA) 方案，其主要价值在于：它不仅保留了通过规则和一般数据仿照前后关系模型化的环境，而且还保留了在不同概念层次上具有个性数据的

智能体。为简化代码，我们建议智能体应避免直接交流，而是要通过环境互通信息。例如，环境允许每一个智能体得到周边成员列表。这一点并不是强制性的，但是，如果我们允许智能体之间进行直接通信的话，整个代码将变得更为复杂。

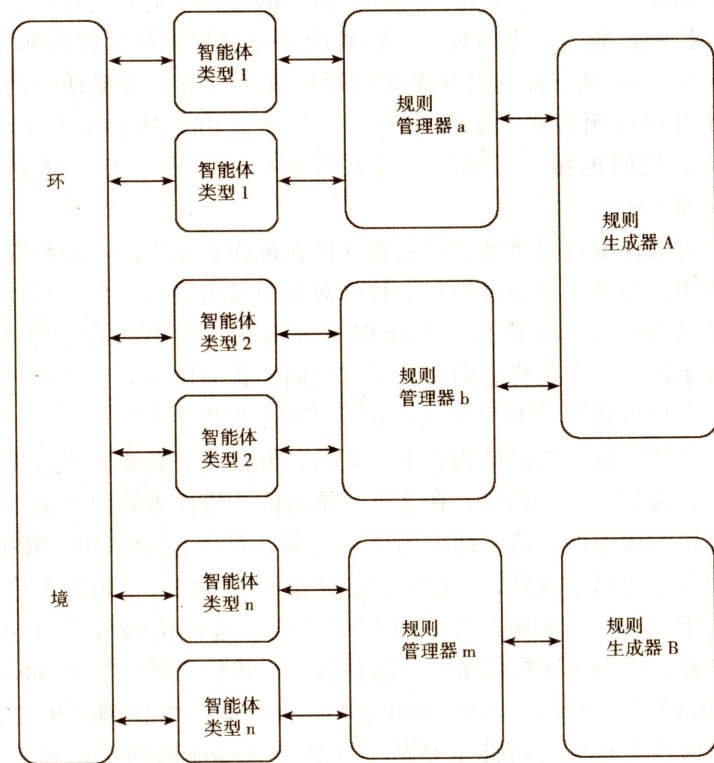


图 3.1 方案

为了简化代码设计，智能体的行为是由被称作 Rule-Masters（规则管理器）的外部对象所决定的，这可以被解释为是识别智

能体的一种抽象表示。无论是生产系统还是分类器系统，神经网络和遗传算法都是实现 Rule-Masters 的候选方案。

我们还会用到元规则 (meta-rules)，也就是用来修改规则的规则 (神经网络的训练就是一个例子)。Rule-Master 对象链接到 Rule-Maker (规则生成器) 对象，而 Rule-Maker 对象的作用是修改、控制智能体行为的规则。仿真的学习过程就是这样的例子。Rule-Masters 从智能体自身或专门承担收集和分发数据任务的智能体得到应用规则的必要信息。类似的，Rule-Makers 与 Rule-Masters 之间也相互影响，后者从智能体中获取数据以传送给 Rule-Makers。

智能体也可以将他们的数据存储在称作数据库的一种特殊对象当中，并且可以通过另一个特殊对象即通用接口对象，与环境以及其他智能体间产生交互作用 (数据库和接口并未在图 3.1 中表示出来，而是将它们简单地与它们的智能体一对一连接，数据库和接口被用在 BP-CT 软件包及其派生程序中)。

虽然，这一代码结构看上去复杂，但在我们需要修改仿真的时候，还是十分有益的。在这里，结构的严格性为我们带来了无可比拟的清晰性。这一结构的应用实例，可以在 SWARM BP-CT 应用程序代码中找到，其中的智能体是神经网络；也可以在第七节关于 hayekian 实验的专用代码中找到，这种代码可在 BP-CT 上创建，使用了同样的结构。这些软件包的代码可以在 Trento 的 CEEL 网站 (<http://www-ceel.gelso.unitn.it>) 中找到，也可以直接向作者索要 (可能也是从一些关于 Swarm 的网站得到的未加整理的片段)。

应用 ERA 结构的另一个优点，是它的模块性。这使得模型构建者在任何想要将基于智能体的神经网络模型转为其他诸如生产系统、分类器系统或遗传算法的时候，仅仅需要改变 Rule-Master 和 Rule-Maker 模块或对象就可以实现了。

按 Swarm 的术语，Environment 与 ModelSwarm 对象是相互协调的，而 ObserverSwarm 对象则在这一结构之外。

bp-ct 代码提供了一个基本软件包。利用它，我们可以通过以下几种方式获得任何应用程序：①如果我们保持神经网络的选择和 CT (见第五节) 框架不变，或者当我们用任何内部规则系统为神经网络提供输入和目标的时候，可以通过改变接口对象，得到需要的程序；②当我们用 BP-CT 软件包作为神经网络领域的后向传播工具的时候，它可以替换掉外部训练和校验集，得到需要的程序；③如果我们像以上所说的那样要转向别的实例的话，修改 Rule-Master 和 Rule-Maker 对象，也可以得到所需要的程序。

4. 神经网络

1986 年 Rumelhart 和 McClelland 选择神经网络作为执行 BP-CT 软件包的标准工具，利用的是矢量函数的功能，像在等式 3.1 中列出的那样，它在结构上与计量经济学中的非线性多元回归函数十分相似。

$$y = f(B f(Ax)) \quad (3.1)$$

我们可以想象一类简单的人工神经网络 (ANN)，多层感知器或前馈网络，用来连接如图 3.2 中的所示的几个处理单元。处理单元 (PE) 可以被看做极度简化的神经元，这里参数 w_{kj} 表示神经键的强度。 a_j 的值 (激活) 是 PE 或神经元的输入，用以计算它们的加权和 (这一操作表明：用这一传统名词“权重”来作为人工神经网络 (ANN) 中指定函数参数的专用术语)。和矩阵代数中的运算相同，它是由乘积 Ax 来完成的，这里 x 矢量包含激活值 a_j ，而 A 矩阵包含参数 w_{kj} 。PE 产生一个对加权和进行

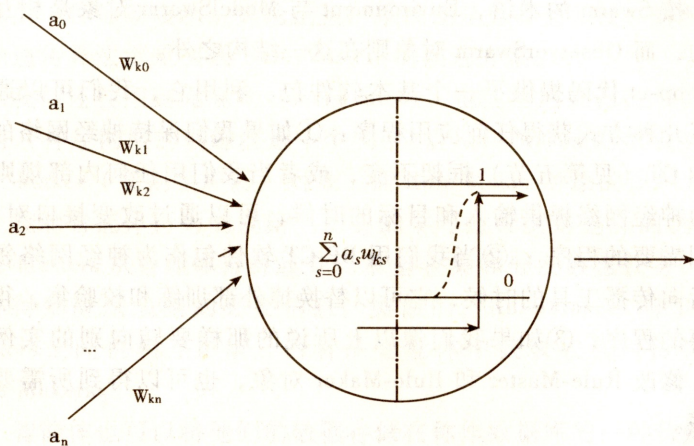


图 3.2 一个节点或人工神经元

非线性变换的输出，像方程 (3.2) 列出的那样，这里参数 k 决定函数与阶跃函数的近似程度*。

$$f(z) = \frac{1}{1 + e^{-kz}} \quad (3.2)$$

ANN 有一种简化的形式，我们在这里并没有用到，它忽略了外部的变换函数，从而简化成 $Bf(Ax)$ 。

A 和 B 的每一行都包含了一个处理单元 (PE) 的权重。在图 3.3 中我们可以看到一个神经网络的实例。它包含一个具有四个处理单元 PE (编号为 1 到 4) 的输入层，一个包含三个处理

* 译者注：如果 $f(z)$ 中的自变量 z 是一个 n 维向量 $[z_i]_n$ ， $f(z)$ 是一个 n 维

向量函数即： $f(z) = \left[\frac{1}{1 + e^{-kz_i}} \right]_n$ 。

单元 PE (编号 5 到 7) 的内部或称隐含层，以及一个包含两个处理单元 PE (编号 8 和 9) 的输出层。请注意，如果去掉隐含层的话，ANN 就退化成为简单的非线性结构 $f(Ax)$ 。

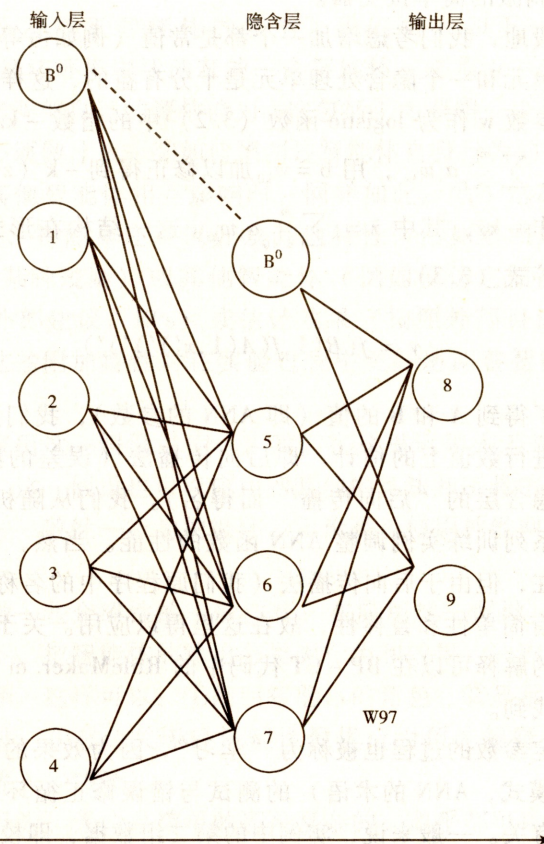


图 3.3 一个简单的人工神经网络

举个例子，图 3.3 中的矩阵 A (两行三列) 包含了处理单元

(PE) 8 和 9 的 w_{kj} 值, 就像图中标出的 w_{97} 一样; 矩阵 B (三行四列) 包含了处理单元 (PE) 5 到 7 的 w_{kj} 值; 向量 x 包含了四个输入处理单元 (PE) 的值, 在这里其仅仅是外界环境到达 ANN 的刺激的简单接受器。

一般地, 我们考虑增加一个都是常值 (例如恒等于 1) 的输入处理单元和一个隐含处理单元是十分有益的, 这样我们可以定义一个参数 w 作为 logistic 函数 (3.2) 中的指数 $-kz$ 的偏移值, 这里 $z = \sum_{s=1}^m a_s w_{ks}$, 用 $b = w_{k0}$ 加以修正得到 $-k(z+b)$, 可以直接写作 $-k\bar{z}$, 其中 $\bar{z} = \sum_{s=0}^m a_s w_{ks}$ 。这一结构在形式上仍可以表示为等式 (3.3)。

$$y = f(B(1, f(A(1, x'))')) \quad (3.3)$$

为了得到 A 和 B 的值 (即 ANN 的参数), 我们必须用简单梯度法进行数值上的估计, 即后向传播法 (误差的测量由输出层传向隐含层的“后向传播”而得名)。我们从随机参数开始, 依据一系列训练实例调整 ANN 函数的性能。当然, 也有其他的方法存在, 但由于后向传播法 (我们原程序中的名称为 BP 的部分) 具有简单性和鲁棒性, 故在这里得以应用。关于 BP 方法的更详细的解释可以在 BP-CT 代码中的 RuleMaker.m 文件的解释部分中找到。

确定参数的过程也被称为“学习”, 因为效果的质量显然与实例 (模式, ANN 的术语) 的测试与错误修正循环中数值迭代的次数有关。一般来说, 实例中的第二组数据, 即校验组, 在学习阶段是不被应用的, 它们被用来检验 ANN 函数的特性。

神经网络工作和训练的特殊方式, 使得它们能够重建任何函数性能, 就像“万能逼近器”一样, 这是 White 在 1990 年提出的。

5. 交叉目标方法 (CT)

5.1 概念简述

为了开发基于智能体的实验, 我们引入如下一般假设 (GH): 在经济环境中活动的一个智能体, 必须以一种连贯的方式开发和调整自己的评估能力, 这包括①为获得一个明确效果该智能体必须做什么; ②如何预测该智能体自身行为的效果。当智能体与其他智能体相互影响时, 同样如此。除了这种内在一致性 (IC), 智能体还可以开发其他特性, 例如参照外部环境 (例如按照某种规则) 或其他智能体 (例如模仿它们) 采取行动 (按照外部建议, EPs) 或估计效果 (按照外部目标, EOs) 的能力。这些附加特性对在实验过程中更好地调整智能体是十分有用的。

为了应用这一假设 (GH), 我们在这里运用人工神经网络。我们发现, 这一假设也可以应用到其他的算法和工具中, 重现经验—学习—一致性—行为的循环, 无论是否使用神经网络都是如此。

这里做一些介绍性的一般注释: 在应用这一假设 (GH) 的所有情况下, 把智能体的输出分类为行为和效果, 一直被用来作为初始选择。这样可以: ①阐明智能体的角色; ②显示模型的合理性和效果; ③避免关于经济理性最优行为预先解释的必要性 (Beltratti 等人, 1996)。

经济行为, 无论是简单的还是复杂的, 似乎都可以直接作为 IC, EPs 和 EOs 的副产品。对于外部观察者来说, 我们的人工适应性主体 (AAAs) 是按一定目的和计划操作的。很清楚, 经济行为不具有这样的象征性实体, 它是观察者抽象的。在这里, 我们想到的相似之处是, 对现实世界智能体行为的观察与分析也会

遇到相同的问题。而且，对于一个外部观察者来说，AAAs 似乎总是一个具有最大化行为的合理性范例。

复杂性在智能体外部可以更容易看到，从它们的交互作用、适应和学习的框架中涌现出的复杂性，通常比智能体内部要多。同样地，理性（以及奥林匹克的理性）都可以作为环境约束和智能体有限能力的副产品，在智能体外部找到。就像最优化也可以作为交互和约束的副产品，作为智能体意向的外部涌现一样。

主要的问题是：智能体的行为，显然是有目标的，即增加或减少某种东西，但却不能由此认为，所有的注意力都要关注从智能体内部搜寻复杂性，而且不能犹豫。根据我们的假设，以及由此产生的交叉目标方法（CT），可以说，我们的工作，是从生命技术边缘向有限理性下的理性 AAAs 的人工世界开拓：从他们的交互作用中，复杂性、优化行为，以及奥林匹克的理性都可以涌现出来，但要在智能体外部进行。

最后，我们要考虑借鉴 Arthur 在 1990 年提出的有限理性研究项目的观点：

为了设计一个再现特殊环境中人类行为的学习系统，我们不仅要关注重现人类学习进程，而且要重现人类学习的模式，可能的话，甚至要包括可能偏离完美理性的方式。理想地说，我们不仅要关注高度符合人类学习曲线的那些曲线，而且，更要致力于研究那些能够通过图灵测试，但难以从人类行为中区分出来的弱点、偏差和错误的学习行为。对不知道该行为究竟是算法产生还是人为产生的观察者而言，这种区分是不可能的。

为了实现这一理想目标，又避免掉入所建模型过于复杂而无法管理的陷阱，我们设想将人工智能体，建立在一个能够通过反

复实验得到校正的算法上。一方面，我们的智能体甚至比新古典主义模型中考虑的还要简单，因为他们的目标和工具并没有那些新古典主义模型中假设的那样强有力。但从另一方面讲，我们的智能体有时又要复杂得多，因为，他们具有持续不断地运用已有的工具学习周围环境的特性。

交叉目标（CTs）的名称，来源于这样的技术，它用来计算出所需训练 ANNs 的目标。ANNs 代表构成我们实验的人工自适应智能体（AAAs）。

沿着这一假设（GH），这些 AAAs 的主要特性，是取得行为与相应结果之间的内部一致性。通常按照假设（GH），在很多（经济上的）情况下，对智能体的行为的评估可被分为两部分：行为量化的数据（要做什么）和行为效果的预测。于是，我们要确定 ANN 的两类输出，同样地，对于 AAA 也要包括：①要实施的行为；②对这些行为效果的推断。

无论是从行为的，还是从相应效果的角度来训练网络，所需的目标，都是用交叉方式建立起来的，由此引出了交叉目标的名字。前者建立在与效果推断有关的网络输出相容的方式上，为的是开发使决策活动接近预期结果的能力，使行为尽量接近期望的效果。与前者相类似，后者是持续地建立在与行为推断相关的网络输出上，以改善智能体估计自身行为决策产生效果的能力。

交叉目标，作为 GH 假设的一个实现，可以重现经济主体的行为，通常在内部采用简单的方式，而外部产生的是复杂的效果。

为了开发经济主体的自主行为而引入的交叉目标方法，可以说是建立行为模型非常有效的算法，且不受约束性的或非约束性的优化技术的影响。这一方法的核心是，通过推测和行动进行学习：主体控制能力可以在既不定义目标，也不定义最大化效果的

情况下得到开发。该方法便于建立在 ANN 的基础之上（但也可以用同样用其他数学工具作为基础）。

5.2 技术细节

我们选择神经网络作为开发交叉目标方法的途径，很大程度上是因为神经元函数具有内在适应能力。这里我们将应用第四节介绍的前馈多层网络。

图 3.4 描述了一个在交叉目标规划下 AAA 的学习和行为。这个 AAA 必须在一个信息集合（输入元素是 $I_1, \dots, I_k$ ）的基础上，做出对自身行为以及相关效果的推断。考虑到 IC 的要求，学习过程的目标如下：①一方面，被仿真的主体发生行为的实际效果，通过可计算的规则进行量化；②另一方面，这些行为需要与推断的效果相匹配。后一种情况我们必须运用后向规则，即使这种后向不确定时就会出现一些问题。Beltratti 等 1996 年对于交叉目标方法的技术解释，我们在下面给予简单介绍：

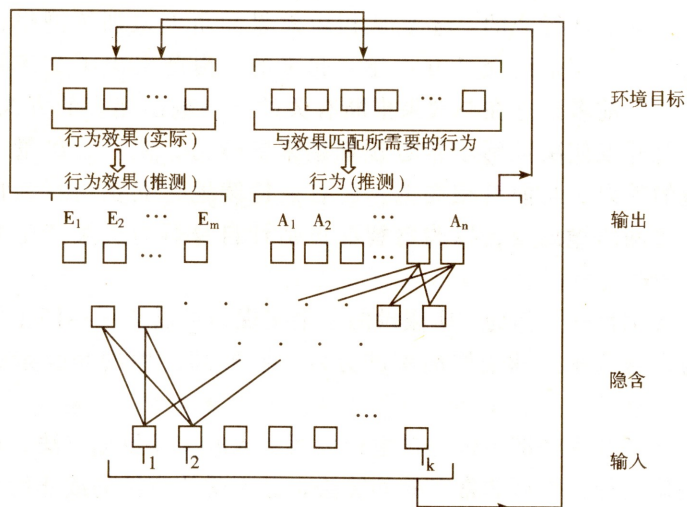


图 3.4 交叉目标方案

首先要说明的是关于学习和交叉目标（CT）：分析在这一过程中权重的变化，我们可以发现，连接输入元素和隐含元素间的权重矩阵很少变化或根本不变，而从隐含到输出层的权重矩阵则是按相应规律变化的。只有隐含层到输出权重的变化决定 ANN 对于外界环境变化的连续适应能力，这时隐含层元素的输出值基本维持为一个常量。这一状况是由目标（由交叉目标方法产生）非常微小的变化和学习循环下降的次数所共同决定的。

由此形成的网络效果显然要经过训练。这里仿真的经济智能体要具有一种在局部做出决策的能力，即：使得输出逐渐适应最终目标，而不考虑输入值。这是短期学习和长期学习所不同的地方。

下面给出一些定义：①短期学习：在执行阶段，智能体不断地修改他们的权重（主要是从隐含层到输出层），以适应交叉目标方法自己产生的目标。②长期学习，根据过去的情况，当我们有效地通过大量的学习循环，将输入映像成目标（与执行阶段产生方式相同），使 ANN 能够明确应用那些在执行和学习阶段中开发出的隐含规则。

第二点说明，无论是外部目标（EOs）还是外部策略（EPs）：如果应用，这些值就在执行和适应阶段替代交叉目标，并被一致地包容到由过去的学习所建立的数据集中。除了由行为产生的目标外，对效果的推断也可以经过训练而趋近于一个值，该值由一个简单规则给出，财富的增加就是一个例子。这就是交叉目标方法术语中的 EO，它的间接效果是通过交叉目标修正行为，使之与（修正后的）效果推断更加相一致。反之，对于完成行为的推断也可以由 EP 进行修正，同时也间接的影响到相应的效果的推断。如果 EO，EP 和 IC 在决定行为时发生冲突，智能体内部也会涌现复杂性，但从有限理性下的理性观点来看，一般是不需要优化和完全理性设置的。

现在我们可以介绍关于交叉目标的一些技术解释了。在图 3.4 的通用方案的帮助下,我们发现:①模型输入的数据主要来自外部环境和其他智能体的行为。②他们可以依赖仿真人工主体的以前行为,也可以不依赖它们。③目标只有在行为发生时才能知道。

交叉目标算法是一个学习和执行的算法:智能体行为是我们生成相关信息所必需的。由此信息我们可以构成目标,通过这些目标又可以训练 ANN 仿真主体。训练集在这里不能由通常的方法创建,因为与 ANN 的输入和输出相关联的规则必须能够在引入 AAAs 的实验里“探测到”。

学习与执行在“一天”的四个阶段发生。这里所谓的“一天”指的是,对输出进行估计,将误差后向传播,并更改神经网络权重的完整过程所需要的四个阶段的总和。初始权重是在给定范围内随机确定的。

观察图 3.4,我们可以按以下顺序介绍这四个阶段。

(1) ANN 的输出:要完成的行为,在图 3.4 的右侧给出。这些行为产生的结果也在该图的左侧列出来。行为和效果是根据输入和网络权重推断出来的。

(2) 网络左侧的目标:由行为产生的对应效果的目标,如图 3.4 输出层左侧所推断的,是由独立推断的行为来描述的。用这种方法,使得对结果的推断能够更接近实际行为产生的真实效果。

(3) 网络右侧的目标:第二阶段在效果一侧测量到的目标和 ANN 的输出的差,可以反过来解释为行为校正的起点,用以逼近推断的效果。所以它们被用于建立推断行为机制的目标。由于上述反推公式往往还没有定义,所以在所有建立的目标中校正是随机的;同时,当同一目标涉及几个校正的时候,只有模值最大的那个才能够被采用。用这种方式,我们可以模拟一个主体需

要遵守一些独立的不一致的命令时的实际行为:可能在所有命令中,值最大的那个在这里才会被执行。

(4) 后向传播:在学习的时候,权重持续修正,以使所获得的推断更接近于推断行为产生的效果,同时推断的行为也要与推断的效果更为一致。这样,我们有两个学习过程,都是以对网络中相对一侧元素的推断为基础。

从由智能体的结果产生的值所形成的时间通路的角度看,这个双向适应过程,通过智能体之间相互作用和上面提到的长期学习,确保了重要的自开发行为的涌现。

我们现在可以正式说明执行与学习交叉目标算法了:首先引入一类效果 E_1 ,他们是由两个行为 A_1 和 A_2 产生的。这一效果的目标是:

$$\hat{E}_1 = f(A_1, A_2) \quad (3.4)$$

这里 $f(*)$ 定义为具有一定计算关系的行为与效果之间的关系。

这里我们的目的是要使得输出 E_1 (由网络进行推断) 与行为 A_1 和 A_2 产生的效果的正确测量值尽量接近, $\hat{E}_1$ 的误差为:

$$e = \hat{E}_1 - E_1$$

或者,按照 ANN 更新的惯例,取 $\hat{E}_1 - E_1$ 平方的一半。为尽量减小这一误差,我们通过网络权重将它后向传播。

我们现在的目的,是找到与效果端输出尽量一致的行为,作为我们的网络输出。因此我们必须校正 A_1 和 A_2 ,使他们尽量接近 $\hat{A}_1$ 和 $\hat{A}_2$,它们是和输出 E_1 相一致的行为。我们无法分开描述 A_1 和 A_2 的目标。从 (3.4) 我们得到:

程序从两个文件中（文件的名字在代码中确定，并可以修改）读取模式的数据。大约 30 个成批训练学习后，我们的 ANN 的 XOR 输出，就像图 3.6 表示的那样，完全符合图 3.7 中所期望的输出（目标）。

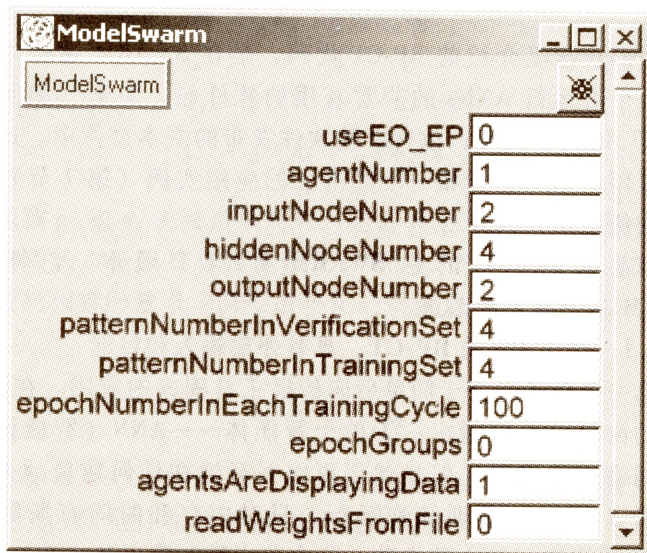


图 3.5 BP-CT 在具有外部输入和目标情况下的控制窗口

6.2 作为 CT 的 BP-CT，输入和目标的内部生成

第二个例子，讲的是关于带有内部生成的输入和目标的程序的用法。这是走向 CT 应用的一个步骤。在图 3.8 的窗口中，我们指出与前一个例子中相同的选项，除了第三点，在那里训练集的模式个数我们取 -1，而校验集我们取 -10。负值在这里表示内部数据的生成：如果是负值，PatternNumberInTrainingSet 变量只能等于 -1；PatternInVerificationSet 变量的负值建立了自生模式的数目，以组成每一个成批训练集。

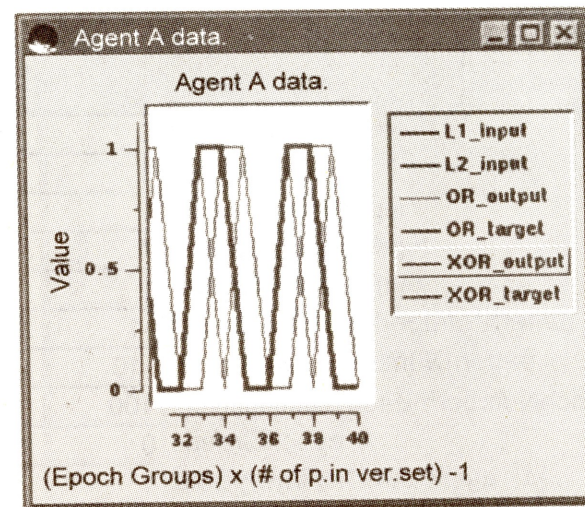


图 3.6 XOR 输出

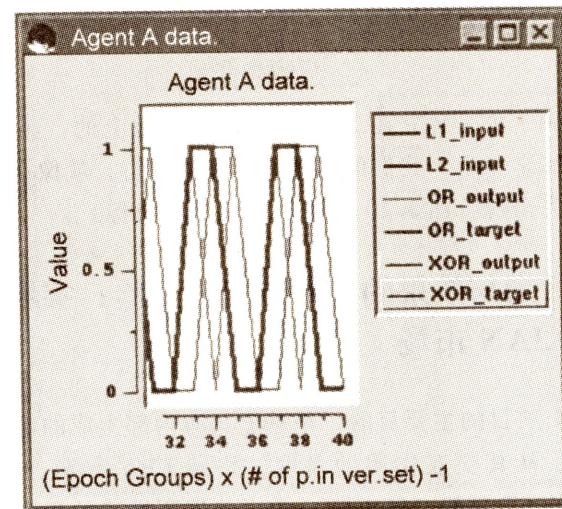


图 3.7 XOR 目标

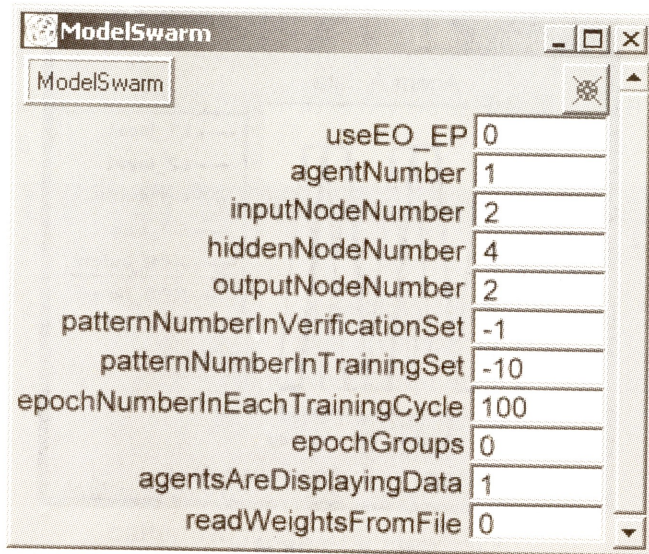


图 3.8 BP-CT 在具有输入和目标内部生成情况下的控制窗口

这个例子也是建立在 XOR 逻辑函数基础上的：经过大约 30 个成批训练学习后，我们的 ANN 的 XOR 输出，就像在图 3.9 中指出的，完全符合图 3.10 中期望的输出（目标）。

7. 行为与效果的 CT 方案的试验：实现自发的 HAYEKIAN 市场

BP-CT 项目的主要目的，是基于具有内部生成的输入和目标的智能体，从其相互作用的角度，按照 ERA 方案（第三节），并在上述的 CT 框架（第五节）中，依据所描述的规则结构来使用代码。

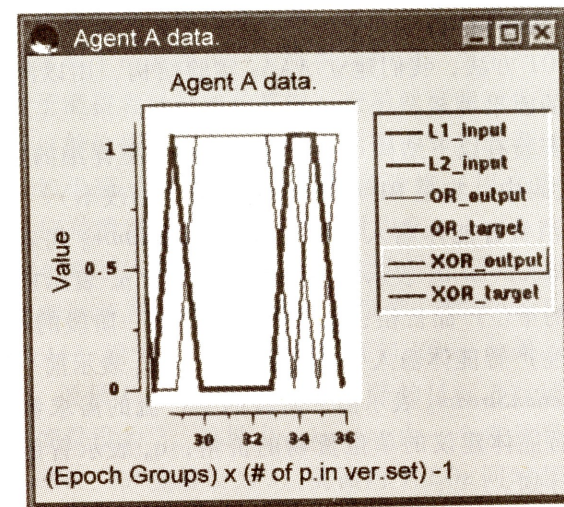


图 3.9 XOR 输出

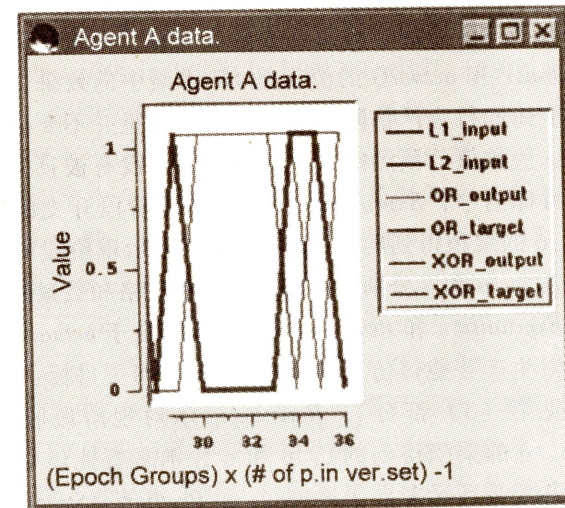


图 3.10 XOR 目标

7.1 实验结构

按照 CT 方法，我们建立 AAAs 实验结构，用以测试智能体学习能力的因果关系。

这一实验有以下结构（关于这一结构的更详细的定义在 CT-HAYEK 代码中，它是 BP-CT 对特定实验的修改本）。

(1) 消费智能体输入中的变量： $Expenditure_0$ 代表前一阶段的开支； $Requirement_0$ 表示前一阶段的需求； p_0 表示前一阶段智能体建议的单位产品价格； q_0 表示智能体前一阶段的购买打算。

(2) 生产智能体输入中的变量： $Revenue_0$ 表示前一阶段的收入； $ProductionStream_0$ 表示前一阶段的生产流的需求量； p_0 表示前一阶段智能体建议的单位货物的价格； q_0 表示智能体前一阶段的销售计划。

(3) 对消费智能体仿真的 ANN 输出及相关目标： $actualP$ ，为推断的交易价格（目标仅是交易价格，实际的或者是潜在的，见下文）； $actualQ$ 为推断的交易数量（目标仅是交易数量，可能为 0）； $actualP$ 和 $actualQ$ 的值就是 CT 术语中的效果，但它们是由其他智能体的行为严格限制的，由于在这里并不要战略上开发智能体对这些数值的预测能力，所以它们并没有被直接包括到行为的交叉目标当中； $Expenditure$ ，表示对现阶段开支的推断（目标是由 p 、 q 的乘积得到的）； $Requirement$ 表示现阶段的需求（目标为 $actualQ$ ）； p 和 q 分别是智能体对交易价格和数量的计划（目标来自于 $Expenditure$ 和 $Requirement$ ，间接来自于 $actualQ$ ）。

(4) 对生产智能体仿真的 ANN 输出及相关目标： $actualP$ 和 $actualQ$ [见第 (3) 部分]； $Revenue$ 表示对现阶段收入的推断（目标由 p 、 q 的乘积得到的）； p 和 q 分别是智能体对交易价格和数量的建议（目标来自于 $Revenue$ 和 $ProductionStream$ ，间接来自于 $actualQ$ ）。

(5) 我们有 10 个消费者和 10 个生产者。生产过程是未经

定义的。简单地假设：生产者在每一时间单元中提供生产的产品数量；消费者和生产者在每一天随机地相遇。当生产者提出的价格低于或等于消费者计划的价格时，交易就可能产生；交易价格是生产者提出的价格，交易的数量是消费者和生产者计划中最小的那一个；如果交易未能成功，则参考价格（用来决定当天的平均价格）应是消费者建议的，且数量为 0。

(6) 我们也引入外部目标 EOs，在发布的 CT-HAYEK 代码中可以找到详细地解释。在消费者的实例中：如果 $useEO_EP$ 变量（见图 3.11）设为 0，则我们没有任何 EOs；如果 $useEO_EP$ 变量设为 1，效果 $Expenditure$ 由一个下降的目标进行训练；如果 $useEO_EP$ 变量设为 2，效果 $Requirement$ 由一个常量目标进行训练；如果 $useEO_EP$ 变量设为 3，前面两个 EOs 同时起作用。CT 的效果是对 p 推断的修正，应记住 EOs 校正的是基本 ANN 效果端的输出，而且通过交叉目标 CT 也会影响到行为端；对 p 推断值的校正会促进或阻止由 $actualQ$ 测出的交换。在生产者一方的情况也大致类似，当 $useEO_EP$ 设为 1、2 或 3 的时候，分别对应不断增长的收入目标、维持常量的 $ProductionStream$ 目标。 $ProductionStream$ 可以解释为与现有的固定资本结构相关的适当产量。

根据第五节中介绍的短时间学习和长时间学习的差别，我们设计了以下实验：①短时间学习（图 3.12，3.15，3.18 和 3.21），智能体仅适应局部状况；②略长时间的学习（图 3.13，3.16，3.19，3.22）智能体开始开发全局适应能力；③最后是完全的长时间学习（图 3.14，3.17，3.20 和 3.23）。

在图 3.11 中，我们可以在没有外部目标的，略长时间学习的 CT-HAYEK 代码启动窗口看到：外部目标的选项；智能体数目的双重选择，和从两个模式号码箱中选择的负数所确定的交叉目标选项。

根据 Hayek 个人主义理念和 Hayek (1949) 经济秩序的理

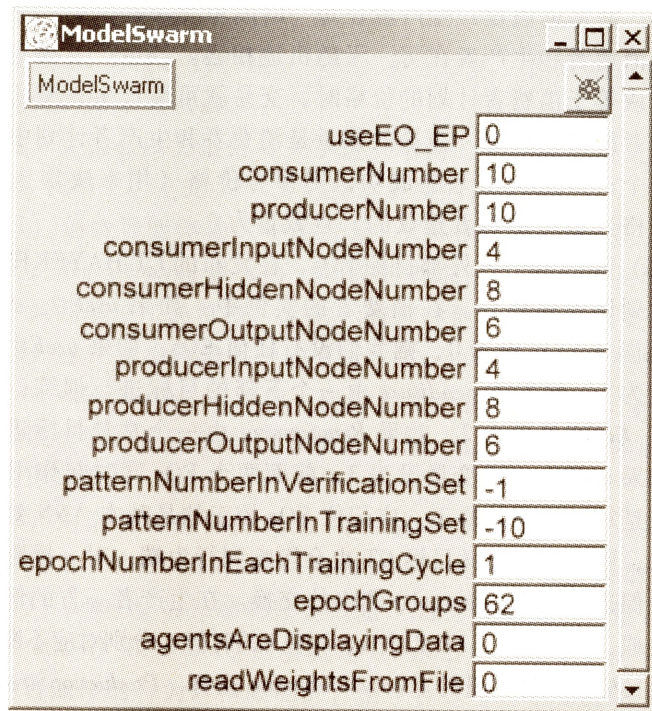


图 3.11 CT- Hayek 包中出现的 bp-ct 控制窗口

论，我们这里处理的是以完全个性化为基础的智能体行为，不是过于简化的人工拍卖市场。一个具有稳定价格的市场总是要出现的，它带有经验性的准均衡的特征。注意，我们在模型中完全排除了对于这一均衡的预先描述，而且智能体对平均价格并不知晓，以上这两项仅用于观察者在外部描述实验。

我们必须认真阅读图 3.12 ~ 3.23（从下向上）：①交易的平均数量（按比例除以10）；②最低价格；③平均价格；④最高价格。价格是由上述变量 p 所获得的。

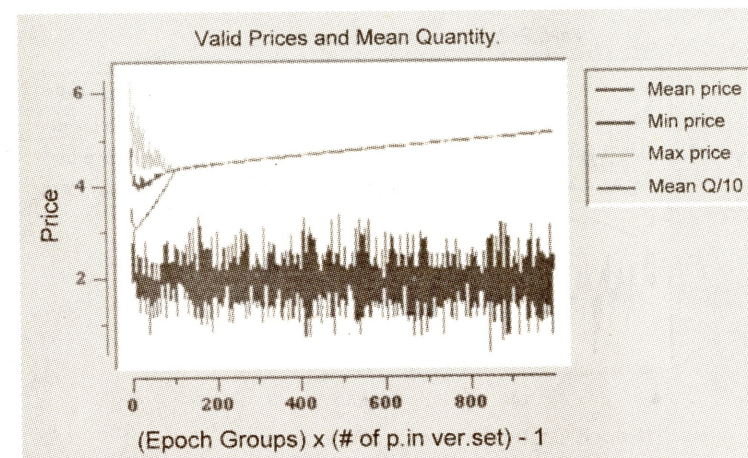


图 3.12 没有外部目标 EOs 时的短期学习

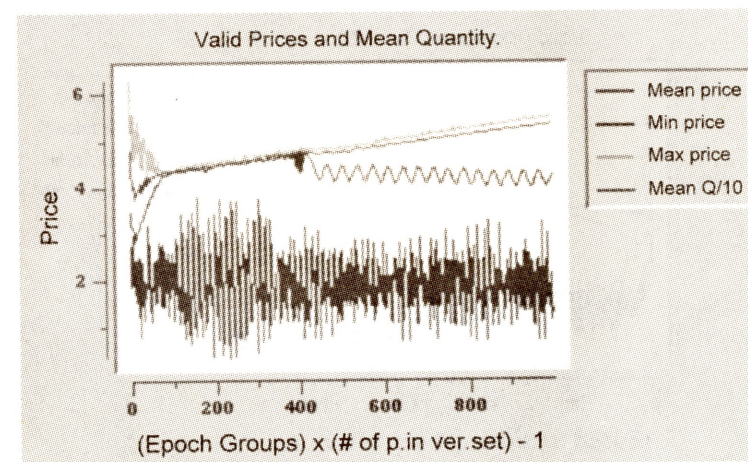


图 3.13 没有外部目标 EOs 时的略长时间的学习

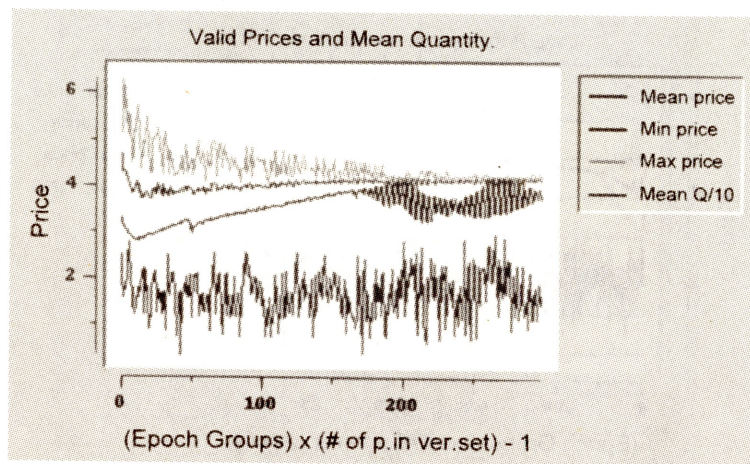


图 3.14 没有外部目标 EOs 时的长期学习

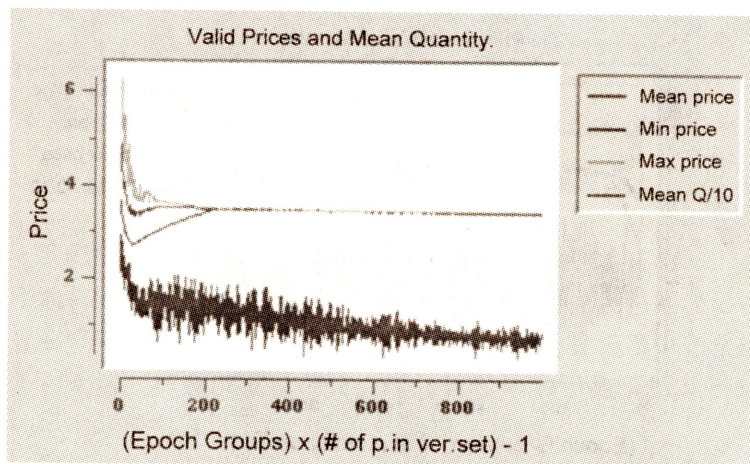


图 3.15 有外部目标 EOs 作用于消费和税收时的短期学习

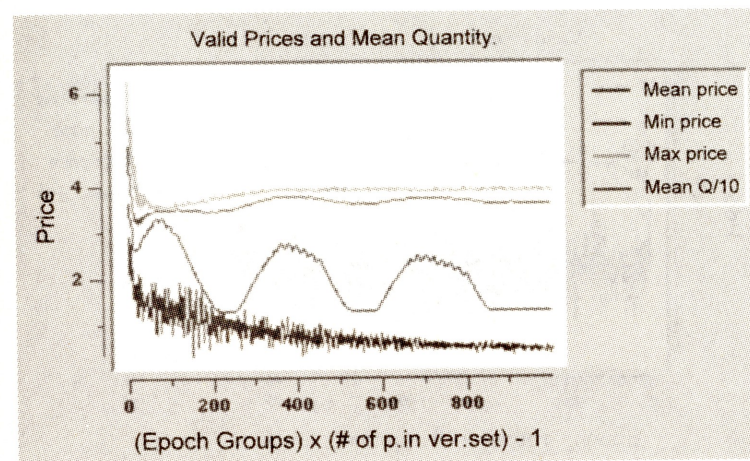


图 3.16 有外部目标 EOs 作用于消费和税收时的略长时间的学习

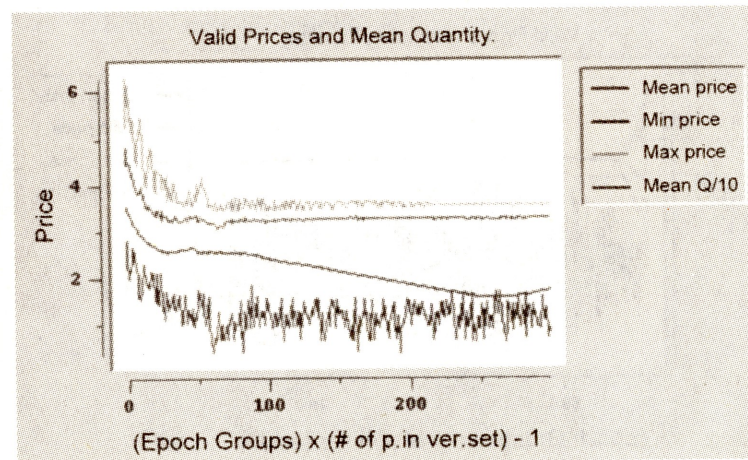


图 3.17 有外部目标 EOs 作用于消费和税收时的长期学习

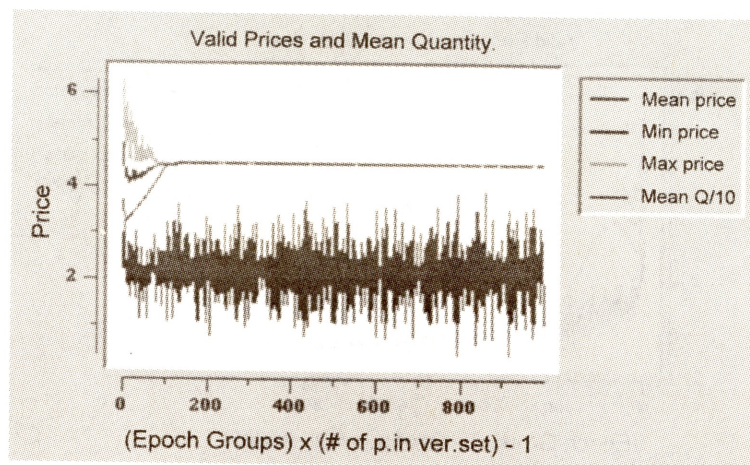


图 3.18 有外部目标 EOs 作用于需求和生产流时的短期学习

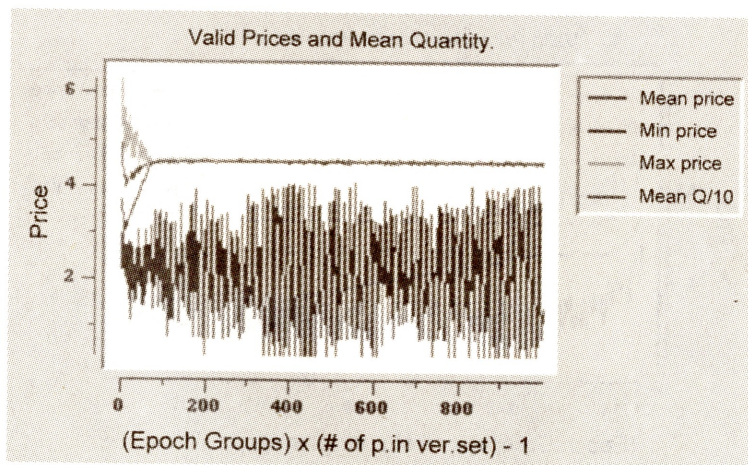


图 3.19 有外部目标 Eos 作用于需求与生产流的略长时间学习

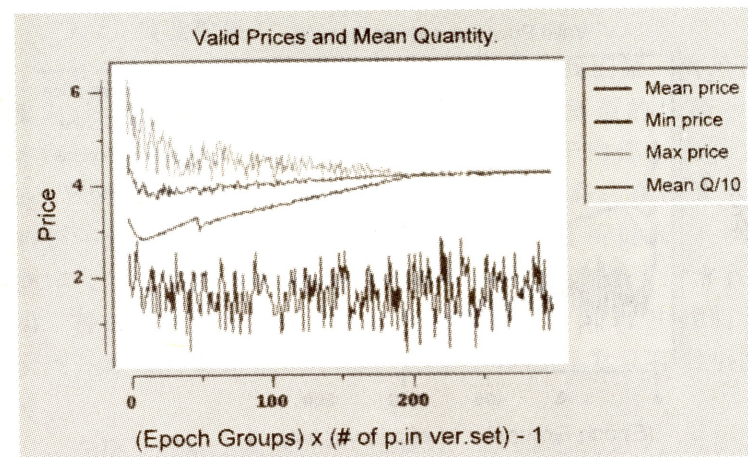


图 3.20 有外部目标 EOs 作用于支出和
生产流时的长期学习

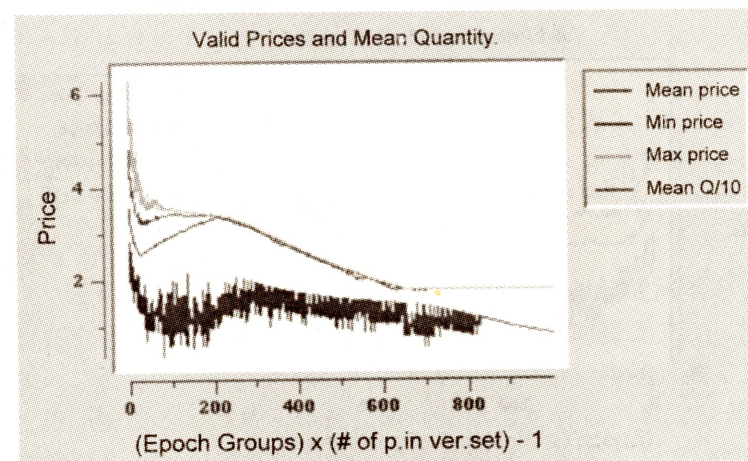


图 3.21 有外部目标 EOs 作用于消费、税收、
需求和生产流时的短期学习

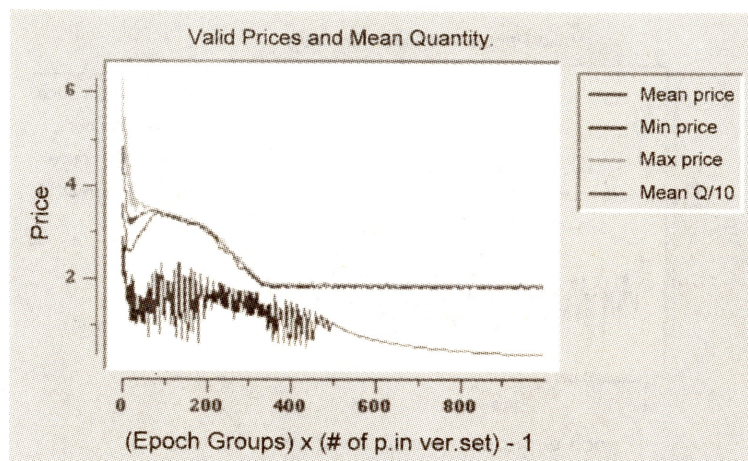


图 3.22 有外部目标 EOs 作用于消费、税收、需求和生产流时的略长时间的学习

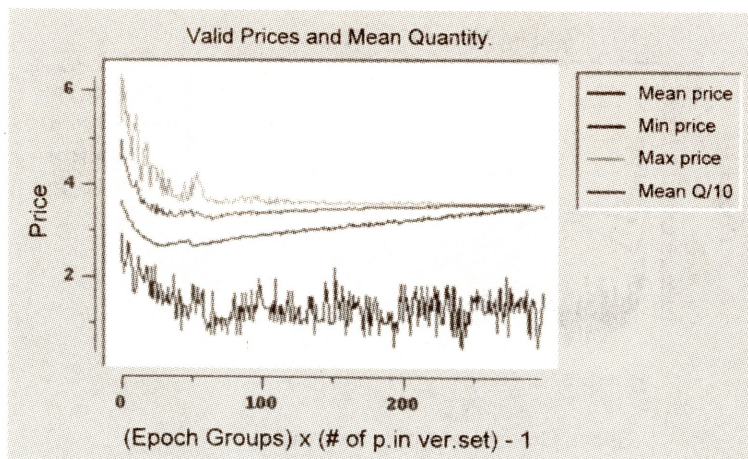


图 3.23 有外部目标 EOs 作用于消费、税收、需求和生产流时的长期学习

7.2 实验效果

我们将 *useEO_EP* 变量设为 0。市场在我们第一次运行模型时就自发地显示出均衡。我们得到一个复杂的均衡状态: 在图 3.12 的短时学习的情况下, 低价格由于内部生成的 (无 EOs) *Requirement* 和 *ProductionStream* 值而提高; 在图 3.13 的略长一点再学习情况下, 系统的复杂程度有所提高, 一些智能体 (均值通常非常接近最大值) 显示出脱离了正常状态; 第三种情况下, 系统要反复地再学习, 在图 3.14 中我们可以看到一种带有鲁棒性的平衡, 一些个体的行为只有细小的差异。

在模型的第二次运行中, *useEO_EP* 变量设为 1, 我们得到一个短时间学习的完全校准的模型, 见图 3.15。关于长时间学习, 在图 3.16 和 3.17 中我们可以看到一些行为分歧。如果单纯考虑由 EOs 确定最终目标, 智能体往往会缺乏经济理念, 去作一些无法完成的事情 (例如无限制地降低开支, 增加收入), 在再次学习过程中这一矛盾就会突现出来。

在模型的第三次运行中, *useEO-EP* 变量设为 2, 价格呈现出稳定的状态 (不是在数量上)。图 3.18、3.19 和 3.20 中的均衡状态, 表现出保持需求和生产流为常量的效果, 这里价格保持近似为常量, 但数量却大幅度震荡。

最后 *useEO_EP* 变量设为 3, 当智能体经过长时间学习, 可以在全局范围内使用时, 通过仿真我们可以同时降低支出, 增加收入, 稳定需求和生产流, 从而得到一个稳定的市场。数量仅发生微小的波动。当然不经过长时间地学习也可以达到这种情形, 就像图 3.21, 但通过学习会变得更加明显, 像图 3.22 和 3.23 中看到的那样。在图 3.22 中我们得到一个有趣的结果: 在价格经过最初阶段的降低后, 学习开始有效地起作用, 也能够得到稳定的市场。

8. 未来的发展

这里我们介绍一些将在今后版本的 BP-CT 中出现的改进。

我们将给每个智能体加上一个探测器,以便在模型运行时能够直接检测智能体的变量。一个更重要的进展,是今后的 CT 规则将由行为与效果的定义而自动产生(目前包含 CT 规则的公式是由手写的接口界面代码产生的)。

最后,我们还要进一步调整,以便让分类器系统*或更一般的遗传算法自动或半自动地替代神经网络。

我们认为,用其他方式开发第五节中的 GH 是十分有用的,在其他背景下校验我们的效果重现性也是十分有用的。我们必须引入相同功能的算法,以便在没有 ANN 的帮助下重现短时间和长时间学习。我们的算法必须能够平滑修正它的输出。在作用和自适应阶段,为了做自适应行为的训练,可以遵循关于行为和行为效果推断的交叉目标方法的建议。这样的算法,虽然不一定完全相同,但必须具备对于输入和输出(目标)向量之间的较强的映射能力,以便能够在不用进一步学习的情况下产生同样的行为结果。

致 谢

这一研究是由 MURST 项目“Intermediazione finanziaria, funzionamento dei mercati ed economia reale.”资助的。

参考文献

1. Arthur, W. B. (1990). “A Learning Algorithm that Mimics Human

Learning.” Santa Fe Institute Working Paper 90 - 026.

2. Beltratti, A., Margarita, S., Terna, P. (1996). *Neural Networks for Economic and Financial Modelling*. London: International Thomson Computer Press.

3. Conte, R., Hegselmann, R., Terna, P. (eds.) (1997). *Simulating Social Phenomena*. Berlin: Springer.

4. Epstein, M. E. and Axtell, R. (1996). *Growing Artificial Societies Social Science from the Bottom Up*. Washington: Brookings Institution Press. Cambridge, MA: MIT Press.

5. Gilbert, N. and Terna, P. (1999). “How to build and use agent-based models in social science.” *Mind & Society*, 1, forthcoming.

6. Von Hayek, F. A. (1949). *Individualism and Economic Order*. London: Routledge.

7. Judd, K. L. (1996). “Computational Economics and Economic Theory: Substitutes or Complements?”! <http://bucky.Stanford.edu/?>.

8. Kirman, A. (1992). “Whom or What Does the Representative Agent Represent?” *Journal of Economic Perspectives*, 6, pp. 126 ~ 139.

9. Rumelhart, D. E. and McClelland, J. L. (1986). *Parallel Distributed Processing. Explorations in the Microstructure of Cognition*. Cambridge, MA: The MIT Press.

10. Russel, S. and Norvig, P. (1986). *Artificial Intelligence. A Modern Approach*. Upper Saddle River, NJ: Prentice - Hall.

11. White, H. (1990). “Connectionist Nonparametric Regression: Multilayer Feedforward Networks Can Learn Arbitrary Mappings.” *Neural Networks*, 5, pp. 535 ~ 550.

* 译者注:指学习分类器系统(Learning Classifier System)见《管理与社会经济系统仿真》宣惠玉等著 p. 133。