

Title: Soft Computing Author: Eva Volná, University of Ostrava, Czech Republic	Lágy számítási modell Szerző: Eva Volná, University of Ostrava, Cseh Köztársaság
Soft computing became a formal area of study in computer science in the early 1990s. Prof. Lotfi Aliasker Zadeh has defined <i>soft computing</i> as a number of techniques and methods that deal with real-world situations in the same way that people deal with them. In computer science, soft computing (sometimes referred to as Computational Intelligence - CI) is the use of inexact solutions to computationally hard tasks such as the solution of NP-complete problems, for which there is no known algorithm that can compute an exact solution in polynomial time. Soft computing differs from conventional computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. In effect the role model for soft computing is the human mind. Soft computing forms the basis of a considerable amount of machine learning techniques. The principal constituents of soft computing are <i>fuzzy logic</i>, <i>evolutionary computation</i>, and <i>neural networks</i>. Generally speaking, soft computing techniques resemble biological processes more closely than traditional techniques, which are largely based on formal logical systems. Soft computing techniques are intended to complement each other.	A lágy számítási modell (soft computing) az 1990-es évek elején vált az informatika formális területévé. Lotfi Aliasker Zadeh professzor a lágy számítási modellt olyan technikák és módszerek együtteseként definiálta, amelyek a valós világ különböző szituációival olyan módon foglalkoznak, mint ahogyan azt az ember teszi. Az informatikában, a lágy számítási modelleket (amelyekre időnként számítástechnikai intelligenciaként - Computational Intelligence - CI hivatkoznak) használják az olyan, számítástechnikai szempontból nehéz feladatok nem-egzakt megoldására, mint az NP-teljes problémák, ahol is nincsenek olyan ismert algoritmusok, amelyek polinomiális időn belül ki tudnák számítani az egzakt megoldásokat. A lágy számítástechnikai modellek abban különböznek a hagyományos számítástechnikától, hogy itt elfogadható a precizitás hiánya, a bizonytalanság, a részigazság és a becslés. Hatásában a lágy számítási modell az emberi agy szerepét tölti be. A lágy számítási modell képezi az alapját a gépi tanulási technikák tekintélyes mennyiségének. A lágy számítási modellek alapvetően fuzzy logikából, evolúciós informatikából és neurális hálózatokból állnak. Általánosan, a lágy számítási technikák sokkal inkább a biológiai folyamatokra hasonlítanak, mint a nagy mértékben a formális logikai rendszereken alapuló hagyományos technikákra. A lágy számítási technikák célja egymás kiegészítése.
Fuzzy logic. Fuzzy logic is based on the theory of fuzzy sets, which is a generalization of the classical set theory. Classical logic only permits conclusions which are either true or false. In fuzzy logic the truth values of variables may be any real number between 0 and 1 inclusive.	Fuzzy logika. A fuzzy logika a fuzzy halmazok elméletén alapul, ami a klasszikus halmaz elméletek egy általánosítása. A klasszikus logika csak olyan következtetéseket engedélyez, amelyek vagy igazak, vagy hamisak. A fuzzy logikában egy változó igaz értéke bármely 0 és 1

Therefore, the membership function is a curve that defines how each point in the input space is mapped to a membership value between 0 and 1. The input space is referred to as the universe of discourse. The universe of discourse X is a continuous space, we usually partition X into several fuzzy sets. These fuzzy sets, which usually carry names that conform to adjectives appearing in our daily linguistic usage, such as "large", "medium" or "small" are called linguistic variables.	közötti valós szám lehet, beleértve magát a 0-t és az 1-t is. Ezért a tagsági függvény egy görbe, amely definiálja, hogy az input terület egyes pontjai hogyan rendelhetők hozzá egy 0 és 1 közötti tagsági értékhez. Az input területre úgy hivatkozunk, mint a társalgás univerzuma. Az X társalgási univerzum egy folytonos terület, mi ezt az X-et rendszerint számos fuzzy halmazra osztjuk. Ezeket a fuzzy halmazokat, amelyek általában olyan neveket hordoznak, amelyek a napi használatunkban előforduló mellékneveknek felelnek meg, mint amilyenek a "large", "medium" vagy "small", nyelvi változóknak nevezzük.
Creating a system with fuzzy logic contains four steps: 1. <i>Rule base</i>: It contains the set of IF-THEN rules provided by the experts to govern the decision making system, on the basis of linguistic information. 2. <i>Fuzzification</i>: It is used to convert inputs i.e. crisp numbers into fuzzy sets. 3. <i>Inference engine</i>: It determines the matching degree of the current fuzzy input with respect to each rule and decides which rules from the rule base are to be fired according to the input field. 4. <i>Defuzzification</i>: It is used to convert the fuzzy sets obtained by inference engine into a crisp value.	Egy rendszer fuzzy logikával történő létrehozása négy lépésből áll: 1. Szabály alapú: Tartalmazza a HA-AKKOR szabályok azon halmazát, amiket a szakértők a nyelvi információk alapján a döntéshozó rendszerek irányítása érdekében hoznak létre. 2. Fuzzifikálás: Ezt a bemeneteknek, vagyis a hagyományos (crisp) számokat fuzzy halmazokká történő alakítására használják. 3. Interferencia motor: Meghatározza, hogy az aktuális fuzzy bemenet milyen mértékben felel meg az egyes szabályoknak, és meghatározza, hogy a bemeneti mezőnek megfelelően az alapszabályokból mely szabályokat kell élesíteni. 4. Defuzzifikálás: Az interferencia motor által kapott fuzzy halmazok crisp értékekké történő konvertálására használatos.
Evolutionary computation. In computer science, evolutionary computation is a family of algorithms for global optimization inspired by biological evolution. In technical terms, they are a family of population-based trial and error problem solvers with a metaheuristic or stochastic optimization	Evolúciós számítástechnika. Az informatikában az evolúciós számítástechnika a biológiai evolúció által inspirált globális optimalizációra szolgáló algoritmusok családja. Technikai kifejezéssel, ezek egy populáció-alapú kísérlet és hiba probléma megoldók családját képezik,

character. In evolutionary computation, an initial set of candidate solutions is generated and iteratively updated. Each new generation is produced by stochastically removing less desired solutions, and introducing small random changes. In biological terminology, a population of solutions is subjected to natural selection (or artificial selection) and mutation. As a result, the population will gradually evolve to increase in fitness, in this case the chosen fitness function of the algorithm. Evolutionary computation techniques can produce highly optimized solutions in a wide range of problem settings, making them popular in computer science.	metaheurisztikus vagy sztochasztikus optimalizációs jellemzővel. Az evolúciós számítástechnikában a lehetséges megoldások egy kiindulási halmazát generálják, és azt iterációs módon aktualizálják. Minden új generáció úgy jön létre, hogy sztochasztikusan eltávolítják a kevésbé kívánatos megoldásokat, és kicsi, random változtatásokat vezetnek be. A biológiai terminológiában a megoldások populációján megtörténik a természetes (vagy mesterséges) kiválasztódás és a mutációk. Ennek eredményeként a populáció fokozatosan egyre nagyobb állóképességre tesz szert, ebben az esetben az algoritmus kiválasztott állóképesség funkciójára. Az evolúciós informatikai technikák nagy mértékben optimalizált megoldásokat tudnak nyújtani a probléma halmazok széles körénél, és népszerűvé tudják tenni azokat a számítástechnikai tudományban.
Evolutionary computation proceeds in the following steps: 1. Test each chromosome to see how good it is at solving the problem at hand and assign a fitness score accordingly. The fitness score is a measure of how good that chromosome is at solving the problem to hand. 2. Select two members from the current population. The chance of being selected is proportional to the chromosomes fitness. Roulette wheel selection is a commonly used method. 3. Dependent on the crossover rate crossover the bits from each chosen chromosome at a randomly chosen point. 4. Step through the chosen chromosomes bits and flip dependent on the mutation rate. Repeat step 2, 3, 4 until a new population has been created.	Az evolúciós számítástechnikai eljárások a következő lépésekből állnak: 1. Minden egyes kromoszóma tesztelése annak megállapítása céljából, hogy mennyire felelnek meg az adott probléma megoldásához, és ennek megfelelően egy megfeleléségi pontszám hozzárendelése. A megfeleléségi pontszám annak a mértéke, hogy az adott kromoszóma mennyire jó a fennálló probléma megoldásához. 2. Két egyed kiválasztása az aktuális populációból. Annak az esélye, hogy valamely egyed kiválasztásra kerül, arányos a kromoszómák megfeleléségi pontszámával. Egy általánosan használt módszer a rulett kerek kiválasztás. 3. A kereszteződési aránytól függően minden egyes kiválasztott kromoszóma bitjeinek keresztezése egy véletlenszerűen kiválasztásra kerülő pontban. 4. Valamennyi kiválasztott kromoszóma biten végig kell menni, és a mutációs aránytól függően meg kell fordítani azokat. A 2, 3, 4 lépések ismétlése az új populáció létrehozásáig.

Artificial neural networks. Artificial neural networks are computing systems inspired by the biological neural networks. The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. An artificial neural network is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.	Mesterséges neurális hálózatok. A mesterséges neurális hálózatok olyan számítástechnikai rendszerek, amelyeket a biológiai neurális hálózatok inspiráltak. A neurális hálózat önmagában nem egy algoritmus, sokkal inkább egy keret ahhoz, hogy sok különböző gépi tanulási algoritmus együtt tudjon dolgozni, és fel tudja dolgozni a komplex beviteli adatokat. Az ilyen rendszerek „learn” példák alapján hajtják végre a feladatokat, általában anélkül, hogy bármilyen feladat-specifikus szabály beléjük lenne programozva. Egy mesterséges neurális hálózat mesterséges neuronoknak nevezett kapcsolódó egységek vagy csomópontok együttesén alapul, amelyek lazán modellezik a biológiai agy neuronjait. Minden egyes összeköttetés a biológiai agy szinapszisaihoz hasonlóan, egy jelet tud átvinni egy mesterséges neurontól egy másikhoz. Egy mesterséges neuron, amely kap egy jelet, fel tudja azt dolgozni, majd jelzést tud adni további, vele kapcsolatban levő mesterséges neuronoknak.
In common implementations, the signal at a connection between artificial neurons are a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. Typically, artificial neurons are aggregated into layers. Signals travel from the first layer (the input layer), to the last layer (the output layer). The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, noise filtering, playing board and video games and medical diagnosis etc.	Az általános megvalósítások során a mesterséges neuronok közötti összekötő jel egy valós szám, és az egyes mesterséges neuronok kimeneteit azok bemeneteihez kapcsolódó néhány, nem-lineáris függvény számolja ki. A mesterséges neuronoknak és éleknek tipikusan van egy súlyozásuk, amit tanulási folyamatként állítanak be. Tipikusan, a mesterséges neuronokat rétegekbe gyűjtik össze. A jelek az első rétegből (a bemeneti réteg) az utolsó réteg (a kimeneti réteg) felé haladnak. A mesterséges neurális hálózat (ANN) megközelítés eredeti célja az volt, hogy a problémát olyan módon oldják meg, ahogyan azt az emberi agy tenné. Idővel azonban a figyelem eltolódott a speciális feladatok megoldása felé, ami a biológiától való eltérést eredményezte. A mesterséges neurális hálózatokat a legkülönbözőbb feladatoknál használják, beleértve a számítógépes látást, a beszéd felismerést, a gépi fordítást, a zajszűrést, a játék fórumokat és video játékokat, valamint az orvosi diagnosztikát, stb.

--	--

Title: Soft Computing

Author: Eva Volná, University of Ostrava, Czech Republic

Soft computing became a formal area of study in computer science in the early 1990s. Prof. Lotfi Aliasker Zadeh has defined *soft computing* as a number of techniques and methods that deal with real-world situations in the same way that people deal with them. In computer science, soft computing (sometimes referred to as Computational Intelligence - CI) is the use of inexact solutions to computationally hard tasks such as the solution of NP-complete problems, for which there is no known algorithm that can compute an exact solution in polynomial time. Soft computing differs from conventional computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. In effect the role model for soft computing is the human mind.

Soft computing forms the basis of a considerable amount of machine learning techniques. The principal constituents of soft computing are *fuzzy logic*, *evolutionary computation*, and *neural networks*. Generally speaking, soft computing techniques resemble biological processes more closely than traditional techniques, which are largely based on formal logical systems. Soft computing techniques are intended to complement each other.

Fuzzy logic. Fuzzy logic is based on the theory of fuzzy sets, which is a generalization of the classical set theory. Classical logic only permits conclusions which are either true or false. In fuzzy logic the truth values of variables may be any real number between 0 and 1 inclusive. Therefore, the membership function is a curve that defines how each point in the input space is mapped to a membership value between 0 and 1. The input space is referred to as the universe of discourse. The universe of discourse X is a continuous space, we usually partition X into several fuzzy sets. These fuzzy sets, which usually carry names that conform to adjectives appearing in our daily linguistic usage, such as "large", "medium" or "small" are called linguistic variables.

Creating a system with fuzzy logic contains four steps:

1. Rule base: It contains the set of IF-THEN rules provided by the experts to govern the decision making system, on the basis of linguistic information.
2. Fuzzification: It is used to convert inputs i.e. crisp numbers into fuzzy sets.
3. Inference engine: It determines the matching degree of the current fuzzy input with respect to each rule and decides which rules from the rule base are to be fired according to the input field.
4. Defuzzification: It is used to convert the fuzzy sets obtained by inference engine into a crisp value.

Evolutionary computation. In computer science, evolutionary computation is a family of algorithms for global optimization inspired by biological evolution. In technical terms, they are a family of population-based trial and error problem solvers with a metaheuristic or stochastic optimization character. In evolutionary computation, an initial set of candidate solutions is generated and iteratively updated. Each new generation is produced by stochastically removing less desired solutions, and introducing small random changes. In biological terminology, a population of solutions is subjected to natural selection (or artificial selection) and mutation. As a result, the population will gradually evolve to increase in fitness, in this case the chosen fitness function of the algorithm.

Evolutionary computation techniques can produce highly optimized solutions in a wide range of problem settings, making them popular in computer science.

Evolutionary computation proceeds in the following steps:

1. Test each chromosome to see how good it is at solving the problem at hand and assign a fitness score accordingly. The fitness score is a measure of how good that chromosome is at solving the problem to hand.
2. Select two members from the current population. The chance of being selected is proportional to the chromosomes fitness. Roulette wheel selection is a commonly used method.
3. Dependent on the crossover rate crossover the bits from each chosen chromosome at a randomly chosen point.
4. Step through the chosen chromosomes bits and flip dependent on the mutation rate.

Repeat step 2, 3, 4 until a new population has been created.

Artificial neural networks. Artificial neural networks are computing systems inspired by the biological neural networks. The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules.

An artificial neural network is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

In common implementations, the signal at a connection between artificial neurons are a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. Typically, artificial neurons are aggregated into layers. Signals travel from the first layer (the input layer), to the last layer (the output layer).

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, noise filtering, playing board and video games and medical diagnosis etc.