

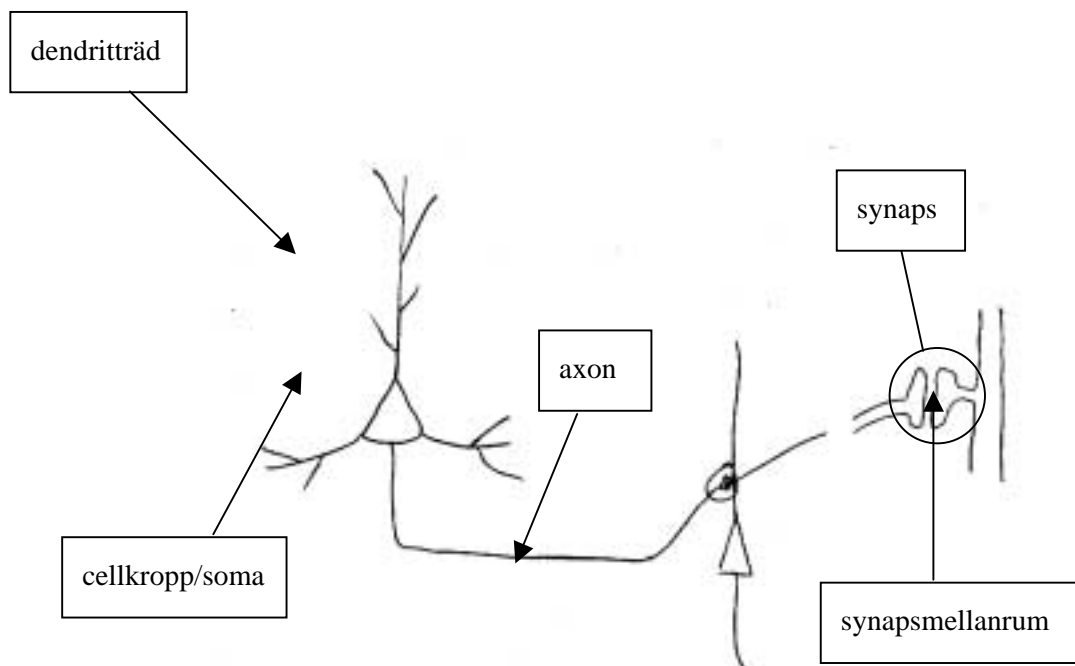
Tentamen i kursen
ARTIFICIELLA NEURONNÄT OCH ANDRA LÄRANDE SYSTEM (2D1432)

Lördagen den 9/3 2002, kl. 08⁰⁰ - 13⁰⁰

Inga hjälpmedel. Max poäng för varje uppgift inom parentes.

Uppgift 1

Rita en biologisk neuron och indikera cellkropp, dendritträd, axon, synaps och synapsmellanrum. Beskriv aktiviteten som äger rum i neuronet med fokus på de elektrokemiska signalernas väg från synaptiska insignaler till utsignal i form av en aktionspotential. (4)



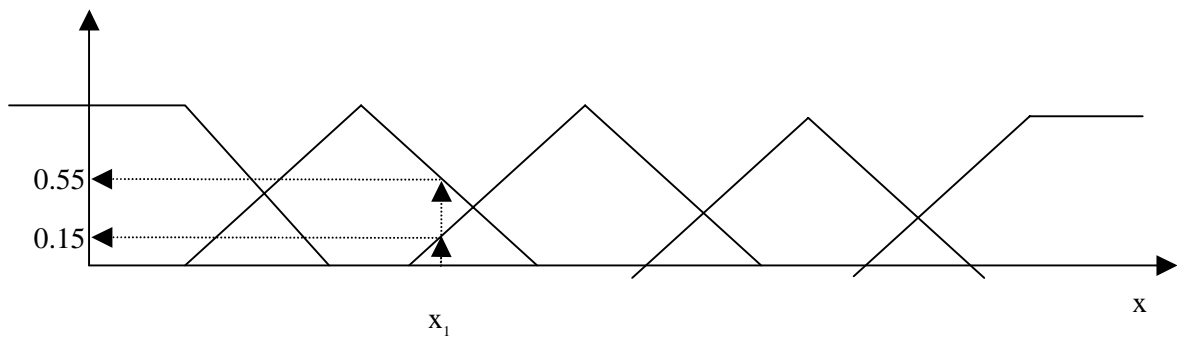
Från presynaptiska (sändande) nervceller kommer signaler som frisätter en kemisk transmittor i det synaptiska mellanrummet. Jonkanaler i den mottagande cellens membran öppnar och släpper in positiva joner (om excitatorisk synaps) vilket ger en EPSP. Dessa summeras från alla nära samtidigt aktiva synapser och om den summerade potentialen överstiger neuronets tröskel (vid "axon hillock") genereras en utsignal i form av en aktionspotential.

Uppgift 2

En kontinuerlig insignal kan representeras på åtminstone två tämligen olika sätt i ett neuronnet, med signalstyrka eller någon form av fuzzy/intervallkodning. Förklara i detta sammanhang dessa två metoder och illustrera hur en insignal "fuzzifieras", dvs hur transformering från invärde till fuzzyrepresentation går till. (2)

Att enhetens signalstyrka representerar en viss insignal är precis som det låter en enkel avbildning, t ex linjär, mellan signal i yttvärlden och enhetens signalstyrka.

Ett alternativ är att insignalen kvantiseras i ett antal nivåer och att en enhet aktiveras för en viss nivå medan övriga är tysta. Detta kan också göras på ett mjukare, graderat sätt t ex som i fuzzy representation med styckvis linjära, triangulära funktioner eller med gaussklockor. Figuren visar i principen hur en insignal x_1 avkodas vid fuzzy representation:



Koden har fem komponenter (en för varje mjukt intervall i figuren). x_1 ger koden $[0 \ 0.55 \ 0.15 \ 0 \ 0]$.

Uppgift 3

Förklara skillnaden mellan perceptroninlärningsregeln och deltaregeln för ett enlagers ANN. Ange särskilt från vilka principer reglerna har härletts och under vilka förutsättningar algoritmerna konvergerar mot en lösning som klassificerar alla tränings exempel korrekt. (4)

Perceptron rule is derived from the manipulation of a hyperplane separating the two classes, if an example is wrongly classified ($t \neq y$) the weight vector w is either moved towards or away from the input x learning rule : $\Delta\omega = \alpha (t-y) x$

Delta rule is derived from gradient descent (with respect to the weights) of the quadratic error between target and network output. $\Delta\omega = -\alpha \partial E / \partial \omega$

Perceptron converges if and only if training patterns are linearly separable and learning rate is sufficiently small, in that case it finds a solution that classifies all examples correctly

Delta rule always converges to the solution with minimum quadratic error if learning rate is sufficiently small (decrease over time) even if the examples are not linearly separable, in case of a one layer ANN delta rule converges to the unique global minimum (no local minima)

the network at minimal error might still classify examples incorrectly (see lab1) even if the training data is linearly separable, it will never classify all examples correctly if the data is linearly non-separable.

Uppgift 4

Bevisa formellt (ej grafiskt) att nedanstående problem är olösbart för en enlagers perceptron med två insignaler.

$x_1 = [-1, 1]$	$t_1 = 1$
$x_2 = [-1, -1]$	$t_2 = 0$
$x_3 = [1, -1]$	$t_3 = 1$
$x_4 = [1, 1]$	$t_4 = 0$

Ledning: utgå från insignal-utsignalrelationen beskrivna som olikheter vilka begränsar vikts- och biastermer. (4)

Perceptron

If $w \bullet x \geq \theta$ then $y = 1$

If $w \bullet x < \theta$ then $y = 0$

Constraints for four input output pairs (x_i, t_i)

$$-w_1 x_1 + w_2 x_2 \geq \theta \quad (1)$$

$$-w_1 x_1 - w_2 x_2 < \theta \quad (2)$$

$$w_1 x_1 - w_2 x_2 \geq \theta \quad (3)$$

$$w_1 x_1 + w_2 x_2 < \theta \quad (4)$$

$$\text{add (1) + (3)}$$

$$0 \geq \theta \quad (5)$$

add (2) + (4)

$$0 < \theta \quad (6)$$

(5) and (6) contradiction, θ must be either positive or strictly negative but can't be both.

Uppgift 5

Förklara termen "generalisering"! Hur kan man uppskatta prestanda (residualfel) hos ett neuronät på framtida ännu oseedda exempel? Hur avgör man när man ska avbryta träningen när man använder "early stopping" för att undvika överträning? (4)

Med "generalisering" avses prestanda (hur litet fel) på en representativ testmängd som inte på något sätt använts vid träningen av en lärlärande algoritm, t ex ANN. Litet fel på oseedda indata betyder god generalisering.

För att uppskatta det kvarvarande felet på oseedda exempel kan man dela upp sina inmönster i en tränings- och en valideringsmängd. Man tränar bara på träningsmängden och efter träningen testas felet på valideringsmängden. Det värde man då får ger då en uppskattning av felet på oseedda exempel.

Med "early stopping" avses att man tränar på detta sätt och följer hur felet på valideringsmängden utvecklas med antal epoker i träningen. Man avbryter träningen när detta fel nått sitt minimum. Då börjar nämligen nätet lära sig egenskaper hos träningsmängden som inte generaliserar till oseedda mönster.

Uppgift 6

Antag att vi har ett diskret Hopfieldnät med 4 enheter och att vi önskar lagra de två vektorerna $[-1, 1, -1, -1]$ och $[1, -1, 1, -1]$. Beräkna nätets viktsmatris och förklara hur du gjort det. Vad är energin hos de två lagrade mönstren? Till vilket tillstånd konvergerar inmönstret $[1, 1, 1, 1]$ om man använder synkron uppdatering? (4)

Vi har två mönster

$$x_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \quad x_2 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

Inläring sker med "Hebbsk regel", summa av ytterprodukter

$$\begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} -1 & 1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & -2 & 2 & 0 \\ -2 & 2 & -2 & 0 \\ 2 & -2 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} = W$$

Energien hos x_1 resp x_2 är

$$E_1 = -\frac{1}{2} \left\{ \begin{bmatrix} 2 & -2 & 2 & 0 \\ -2 & 2 & -2 & 0 \\ 2 & -2 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \right\}^T \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = -10$$

$$E_2 = -\frac{1}{2} \left\{ \begin{bmatrix} 2 & -2 & 2 & 0 \\ -2 & 2 & -2 & 0 \\ 2 & -2 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \right\}^T \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = -10$$

Vad ger mönster [1 1 1 1] för attraktor (synkron uppdatering):

$$\begin{bmatrix} 2 & -2 & 2 & 0 \\ -2 & 2 & -2 & 0 \\ 2 & -2 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ 2 \\ 2 \end{bmatrix} \quad \text{sign} \left\{ \begin{bmatrix} 2 \\ -2 \\ 2 \\ 2 \end{bmatrix} \right\} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

Iterera en gång till för test av stabilitet:

$$\begin{bmatrix} 2 & -2 & 2 & 0 \\ -2 & 2 & -2 & 0 \\ 2 & -2 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ -6 \\ 6 \\ 2 \end{bmatrix} \quad \text{sign} \left\{ \begin{bmatrix} 6 \\ -6 \\ 6 \\ 2 \end{bmatrix} \right\} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \quad \therefore \text{stabil!}$$

Inmönstret [1, 1, 1, 1] konvergerar alltså mot [1, -1, 1, 1] (som ju är inversen av x_1).

Uppgift 7

Belöningsinlärning ("reinforcement learning", RL) behandlar en klass av problem känd som "Markov decision processes". Hur skiljer sig RL från övervakad inlärning i termer av information som är tillgänglig, besluten som tas, den fundamentala strukturen hos problemet, och målet med inlärningsprocessen? Diskutera dessa skillnader mot bakgrund av ett system som ska lära sig att spela "checkers"! Vilka tränings exempel ges till ett program som använder RL respektive övervakad inlärning. (4)

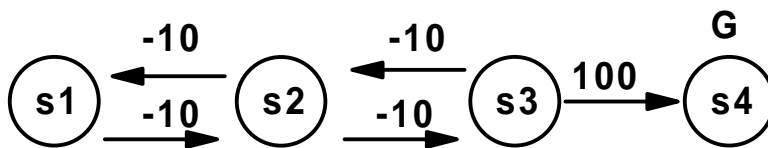
RL learns by trial error through exploration of actions and states that result in high rewards, it is a dynamic process of active exploration. The only information available to or required by the RL system is the current state s , the set of possible actions a , the next state s' that results from applying action a in state s and the reward r associated to that transition. The objective of the RL learning process is to maximize the expected future discounted rewards, in other words to find an optimal policy that choose action that maximizes the expected reward. RL employs a state or state-action value function that describes the estimated future rewards attainable in a particular state or for a particular state action pair. By exploring actions and observing states and rewards the RL system updates its knowledge/belief about this value function. It also updates its policy based on the value function. Rewards can be delayed and feedback about good or bad actions is not immediate, therefore the system faces the temporal credit assignment problem. RL is learning with a critic, namely the system only indirectly obtains information about good or bad actions instead of right and wrong actions. The system is only told what to learn but not how to learn it. In the game checkers the state corresponds to the current board state or features of this board states, actions are the possible moves for the agent in the current state and rewards are given for winning (positive reward) or losing (negative rewards) the game. Learning takes place online by letting the RL system either play against itself or an opponent.

In supervised learning there is no exploration but the objective is to minimize the error between the network output and the target value on future unseen instances (generalization). The goal is to mimic the game playing behavior of the trainer/expert who provided the examples. There is no explicit goal of winning the game, but only to generalize from a training set. The training data provided by a

checkers expert or a database of previous games consists of pairs of board states and the move to be played in that board state. Notice, that a supervised learning system for checkers will never play better than the expert who provided the examples and the system does not improve its performance over time after it has been trained. Therefore, the learning approach is static and learning takes place offline. The learning process consists of updating the weights of the network in such a way that the error between the network output and target values is minimized over the training data.

Uppgift 8

Betrakta följande problem med fyra tillstånd s_1, s_2, s_3, s_4 , där s_4 är ett absorberande måltillstånd som när det väl nåtts terminerar all belöning/bestrafning. I s_1 är den enda tillåtna handlingen att gå till höger till s_2 , medan i s_2 och s_3 agenten kan välja mellan höger eller vänster. Agenten får en belöning på +100 när den når måltillståndet och en bestraffning (negativ belöning) på -10 för alla andra handlingar. Vilken är den optimala policyn? Vilka är tillståndsvärdena $V(s_i)$ givet optimal policy med diskonteringsfaktor $\gamma=0.8$?



Beräkna tillståndsvärdena $V(s_i)$ med hjälp av "iterative policy evaluation" för en slumpmässig policy,

$$p(s_1, \rightarrow) = 1.0$$

$$p(s_2, \rightarrow) = p(s_2, \leftarrow) = 0.5$$

$$p(s_3, \rightarrow) = p(s_3, \leftarrow) = 0.5$$

Använd följande initialvärden: $V(s_1) = 10$, $V(s_2) = 30$, $V(s_3) = 70$, $V(s_4) = 0$ och iterera två gånger. Du behöver inte räkna med decimal noggrannhet (dvs avrunda 5.8 till 6). (6)

The optimal policy is to move to the right in each state, namely

$$\pi(s_1) = \rightarrow, \pi(s_2) = \rightarrow, \pi(s_3) = \rightarrow.$$

Or more formal in a probabilistic sense

$$\pi(s_1, \rightarrow) = 1.0, \pi(s_2, \rightarrow) = 1.0, \pi(s_3, \rightarrow) = 1.0, \pi(s_2, \leftarrow) = 0.0, \pi(s_3, \leftarrow) = 0.0$$

The state values for the optimal policy are

$$V(s_4) = 0$$

$$V(s_3) = r(s_3, \rightarrow) + \gamma V(s_4) = 100 + 0.8 \cdot 0 = 100$$

$$V(s_2) = r(s_2, \rightarrow) + \gamma V(s_3) = -10 + 0.8 \cdot 100 = 70$$

$$V(s_1) = r(s_1, \rightarrow) + \gamma V(s_2) = -10 + 0.8 \cdot 70 = 46$$

Random policy: Bellmann equation:

$$V(s) = \sum_{a, s'} \pi(s, a) p(s'|s, a) (r(s, a) + \gamma V(s'))$$

Where $p(s'|s, a)$ is the probability of ending up in state s' when applying action a in state s , since the above environment is deterministic these $p(s'|s, a)$ are either 0 or 1 in our case. The Bellman equations for the concrete system become.

$$V(s_4) = 0$$

$$V(s_3) = 0.5 (r(s_3, \rightarrow) + \gamma V(s_4)) + 0.5 (r(s_3, \leftarrow) + \gamma V(s_2))$$

$$V(s_2) = 0.5 (r(s_2, \rightarrow) + \gamma V(s_3)) + 0.5 (r(s_2, \leftarrow) + \gamma V(s_1))$$

$$V(s_1) = r(s_1, \rightarrow) + \gamma V(s_2)$$

First iteration:

$$V(s_4) = 0,$$

$$V(s_3) = 0.5(100 + 0 - 10 + 0.8 \cdot 30) = 57$$

$$V(s_2) = 0.5(-10 + 0.8 \cdot 70 - 10 + 0.8 \cdot 10) = 22$$

$$V(s_1) = -10 + 0.8 \cdot 30 = 14$$

Second iteration

$$V(s_4) = 0$$

$$V(s_3) = 0.5(100 + 0 - 10 + 0.8 \cdot 22) = 54$$

$$V(s_2) = 0.5(-10 + 0.8 \cdot 57 - 10 + 0.8 \cdot 14) = 18$$

$$V(s_1) = -10 + 0.8 \cdot 22 = 8$$

The exact solution is

$$V(s_4) = 0, V(s_3) = 48.1, V(s_2) = 7.7, V(s_1) = -3.8$$

Uppgift 9

Beskriv och förklara kortfattat de två faserna i SOFM algoritmen!

(3)

(jfr Gurney s 128)

Kohonens "Self-Organizing Feature Map" algoritmen består av två huvudfaser:

- 1) Utseende av vinnare
- 2) Uppdatering av enheternas viktsvektorer.

Dessa går till som följer:

- 1) När ett nytt sampel/input presenteras matchas detta mot befintliga viktsvektorer. Exempelvis används euklidiskt avstånd i inrymden. Den viktsvektor som ligger närmast samplet utses till vinnare.
- 2) Viktsvektorerna hos vinnaren och dess grannoder flyttas mot samplet med någon viss inlärningshastighet α .

$$\Delta w_j = \alpha(x - w_j), \text{ nod } j \text{ tillhör grannskap.}$$

Både grannskapets storlek och inlärningshastigheten minskas med ökande antal iterationer.

Uppgift 10

Beskriv vad som menas med "resonans" i ett ART-nätverk, samt under vilka förutsättningar resonans uppkommer!

(3)

Resonance occurs if there is a match between the template (represented by the currently winner-take all node in layer 2) and the input in an ART network. Matching takes place in layer 1, in which each node computes the bit-wise AND between the template and the input. Resonance occurs if the ratio between 1's in the bottom up input $|I|$ and the number of 1's in the intersection $|Z \cap I|$ between the input I and the template Z is larger than the vigilance parameter ρ

$$|Z \cap I| / |I| > \rho$$

If resonance occurs the short term memory STM in the ART network stabilizes and adaptation of the long term memory LTM (weights) occurs. Example:

$$I = 110111,$$

$$Z = 100011$$

$$Z \cap I = 100011,$$

$|Z \cap I| / |I| = 3/5$ which means if $\rho < 0.6$ resonance between I and Z occurs otherwise the next template is tried. The vigilance parameter determines the resolution of templates/prototypes, for large ρ fewer patterns belong to the same template but match the template better, for small ρ the templates represent more patterns. According to the 2/3 rule, two out of three inputs to nodes in layer 1 must be active

to activate the node. Since the gain to layer is turned off, these two inputs must come from input layer and layer 2.

If there is no resonance between an input and the currently active template (node in layer 2), the template in layer 2 is suppressed and the next template is tried either until resonance, or in case no template matches the input a new template is created.

Uppgift 11

När man använder en supportvektormaskin transformeras indata till ett "feature space". Förklara varför! Ge också exempel på och beskriv två tillämpningar av en SV-maskin. (4)

The idea with svm is to construct a hyperplane in order to maximize the the margin of separation between positive and negative examples. The transformation from input space to feature space permits patterns to be linearly separable provided (i) the transformation is non-linear and (ii) the dimensionality of the feature space is high enough. The construction of the hyperplane make use of vectors from the feature space rather than the input space. By computing the inner product between support vectors (subset discovered by the svm from the training data) and input vectors we can construct a hyperplane without consider the feature space in explicit form. aa

Possible example categories include: (a) the XOR problem as in the handouts, (b) noise removal (kernel based PCA in handouts), (c) pattern classification tasks (describe the above), (d) non-linear regression.

Uppgift 12

Genetiska algoritmer och ANN är båda adaptiva men på olika sätt. Beskriv kortfattat på vilket sätt de är inspirerade av biologin samt vilka viktiga begrepp och mekanismer respektive metod har ärvt från sin biologiska motsvarighet? (4)

Genetic algorithms are inspired by **natural evolution and genetics**.

GA key concepts and mechanisms: learning on the population level, focus is rather on adaptation than on learning, **population of competing individuals, concept of a fitness** that describes chance of reproducing offspring, **Darwinian selection survival of the fittest, genetic adaptation to the environment**, modification of the genotype – evaluation of the phenotype (no Lamarckian learning), **generation of new variants through mutation and recombination, concept of a genetic code (DNA)** that is a blueprint for a phenotype (animal, human)

Neural networks are inspired by the biological neural processes occurring in the **human and animal brains or nervous system**

Mechanism, key concepts: **learning** on an individual basis, distributed, parallel processing, **many computationally simple but highly connected processing elements (neurons) operate in parallel, learning takes place in the synaptic connections (weights), robustness towards noisy, incomplete patterns**, ability to generalize to previously unseen instances, learning from a set of examples, activity of a neuron depends on the summation of weighted inputs from other neurons, neuron fires when the input exceeds a certain threshold, Hebbian learning, the strength of a synaptic connection is increased if pre- and postsynaptic neuron are active (e.g. Hopfield network), inhibitory, excitatory connections, winner-takes all dynamics.

Uppgift 13

Antag att du önskar använda en genetisk algoritm för att "träna" ett framåtkopplat ANN på en träningsmängd av in/utmönster. Vad skulle GA-populationen bestå av? Vad skulle du använda som överlevnadsvärde ("fitness measure")? (3)

A chromosome genotype in GA population could encode

- a) the strength of synaptic weights in the ANN
- b) the architecture of the ANN, e.g. connectivity between neurons, number of layers, number of hidden neurons
- c) parameters of the learning algorithm, learning rate, momentum term

The fitness measure to be maximized would be

- a) for function approximation

the inverse of the sum of squared errors between the target values and the network output or if the GA is designed to minimize fitness the error itself.

- b) for classification tasks, binary outputs

number of correct classifications

Usually one would evaluate the fitness without any further supervised training, but in principle one could use the GA coded weights as initial values only and compute the fitness after supervised training of the weights.