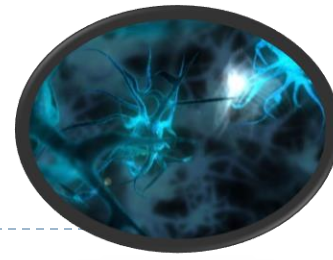


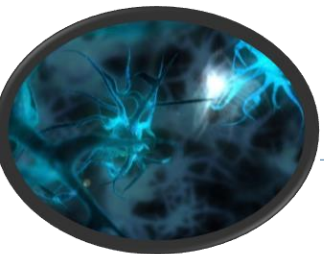
YAPAY SİNİR AĞLARINA GİRİŞ

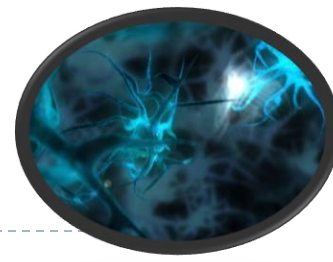
DR. ÖĞR. ÜYESİ BETÜL UZBAŞ

ART (Adaptif Rezonans Teori)



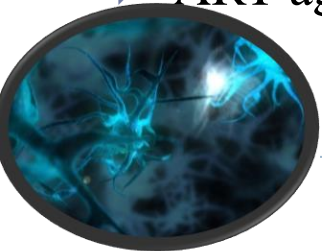
- ▶ ART ağıları danışmansız öğrenme yöntemlerindendir.
- ▶ Uyarlamalı Rezonans Teorisi (Adaptive Resonance Theory, ART) ağıları Grossberg'in 1976 yılında beynin fonksiyonlarını açıklamaya yönelik araştırmalarının sonucunda ortaya çıkmıştır.
- ▶ Bu ağların en önemli özellikleri gerçek zamanlı çalışabilmeleri ve çevrimiçi (*on-line*) öğrenebilmeleridir.
- ▶ Yeni durumlara adapte olabilmekte ART ağıları oldukça güçlüdürler. Bunu sağlayabilmek için bir taraftan öğrenirken bir taraftan da unutabilmektedirler.

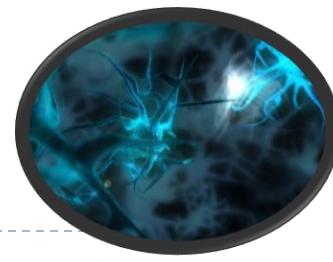




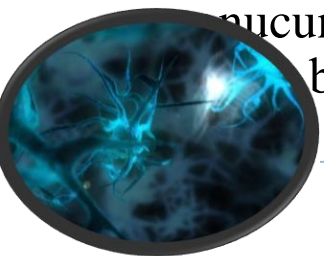
Hafıza (Bellek) Kavramı

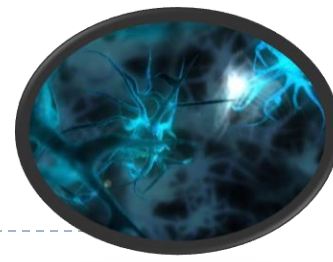
- ▶ ART ağlarında öğrenme doğru bilgilerin belirlenerek hafızaya alınması anlamına gelmektedir. Bilgilerin hafızaya alınması ve hafıza tutulması iki şekilde olmaktadır:
- ▶ Kısa Dönemli Hafıza (KDH): Bilgilerin geçici olarak tutulduğu ve zaman içerisinde de yok olduğu ve yerlerine başka bilgilerin saklandığı hafızalardır.
- ▶ Uzun Dönemli Hafıza (UDH): Bilgilerin sürekli tutulduğu ve kolay kolay unutulmadığı hafızadır. Bilginin silinmesi için çok uzun zamanın geçmesi gerekebilmektedir.
- ▶ ART ağlarında bilgiler hem KDH hem de UDH da tutulmaktadır.



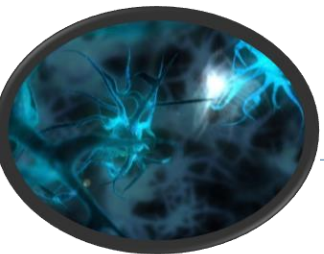


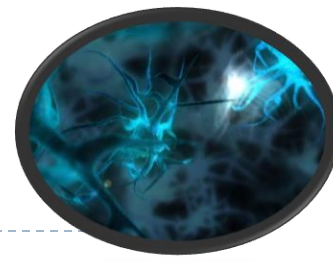
- ▶ ART ağlarının 3 temel özelliği vardır.
- ▶ **1. Normalizasyon:** Biyolojik sistemler bulundukları çevredeki değişimlere duyarlıdır ve bu değişikliklere uyum sağlayabilme özelliklerine sahiptir. Örnek olarak, insanların fazla gürültülü bir ortama ya da karanlık bir ortama uyum sağlaması, çevresindeki olayları normalize ettiğini göstermektedir.
- ▶ **2. Ayırıştırabilme:** İnsanların fark edilmesi zor olan ayrıntıları ayırıştırabilme yeteneği bazen hayat kurtaran bir özelliktir. Örneğin insan, dinlenen bir kaplanla saldırmak üzere olan bir kaplan arasındaki farkı kolayca anlayabilecek bir sisteme sahiptir.
- ▶ **3. Ayrıntıların Saklandığı Kısa Dönemli Hafıza:** Belirlenen farklılıklar ve çevresel değişimler davranışlara dönüşmeden önce hafızada saklanmaktadır. Bilgilerin geçici olarak tutulduğu ve zaman içinde yok olarak yerine yeni bilgilerin saklandığı yer KDH dir. Ancak sürekli aynı şeylerin tekrar edilmesi durumunda bilgiler KDH yerine UDH de saklanabilmektedir. Uzun dönemli bilgilerin sürekli tutulduğu ve kolay kolay unutulmadığı hafıza türüdür.



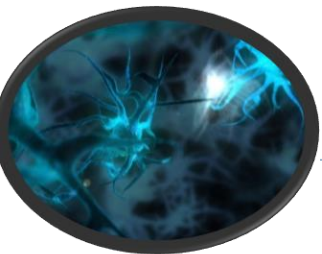
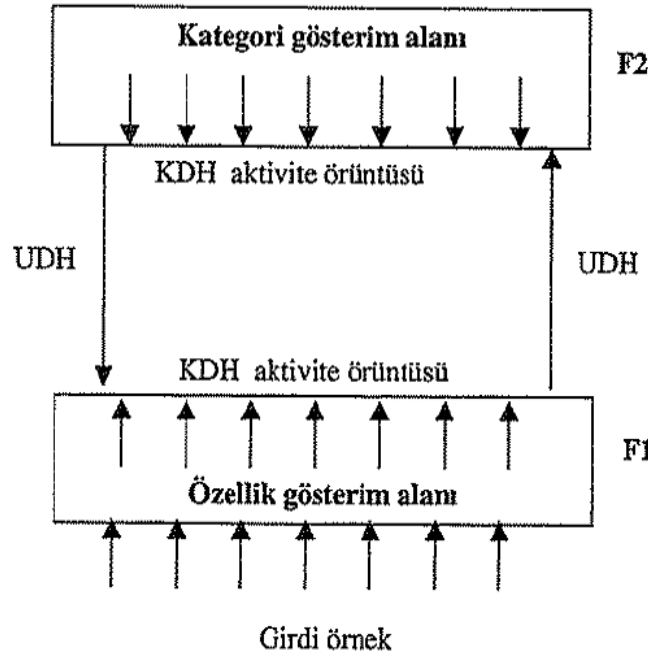


- ▶ ART ağıları gerçek zamanlı olarak oldukça hızlı ve kararlı bir öğrenme yeteneğine sahiptir.
- ▶ ART çevrimiçi öğrenip çalışabilir, esnektir.
- ▶ ART danışmansız öğrenme yöntemine sahiptir.





- ART ağıları, genel olarak iki katmandan oluşur. Bu katmanlar F1 ve F2 olarak isimlendirilmiştir. F1 katmanı girdinin özelliklerini, F2 katmanı kategorileri (ayrıştırılmış sınıfları) gösterir. Bu iki katman birbirlerine UDH ile bağlanırlar. Girdi bilgileri F1 katmanından alınarak, F2 katmanında sınıflandırma yapılmaktadır



-
- ▶ ART ağlarının çalışması, aşağıdan yukarı (F1'den F2'ye) bilgi işleme ve yukarıdan aşağı (F2'den F1'e) bilgi işleme olmak üzere iki şekilde olmaktadır.
 - ▶ Ağa sunuluna girdilerin sınıflandırılmaması durumunda yeni bir kategori oluşturmaya yardım eden oryantasyon modülü vardır.
 - ▶ F1 ve F2 katmanları arasında hem aşağıdan yukarı , hem de yukardan aşağı ağırlık vektörleri vardır.
 - ▶ Girdiler F1 katmanından ağa sunulur ve yukarı doğru ağırlıklar ile F2 katmanından çıktı değerleri hesaplanır. F2 katmanında en yüksek çıktı değerini oluşturan proses elemanı yarışmayı kazanan proses elemanı belirlenir.
 - ▶ Kazanan elemanın girdi vektörünü göstermesi için ona bağlı hafızaki vektörünün birbirine benzemesi gerekmektedir. Bu benzerliği benzerlik katsayısı ile karar verilir.
 - ▶ Eğer iki vektör benzerse girdi vektörü o sınıfın elemanı sayılır ve ağırlıklar girdi vektörüne göre güncellenir. Arada benzerlik yoksa oryantasyon sistemi o girdi vektörü için yeni bir sınıf oluşturur. Bu nedenle bir ART ağında örnek sayısı kadar sınıf sayısı oluşturabilir.
-



-
- ▶ Değişik ART ağları geliştirilmiştir.
 - ▶ ART1 ağı geliştirilmiş ilk ART ağıdır. Bu ağ sadece ikili (binary) değerlerden oluşan girdi vektörünü kabul eder.
 - ▶ ART2 ağları ise sürekli değerleri de kabul etmektedir.
 - ▶ Benzer şekilde ART3, Fuzzy ART, ARTMAP gibi başka modellerde geliştirilmiştir.



ART ağlarının diğer yapay sinir ağlarından farkları:

- ▶ ART ağları gerçek zamanlı olarak oldukça hızlı ve kararlı bir şekilde öğrenme yeteneklerine sahiptirler. Diğer bir çok ağda olmayan bu yetenek sayesinde ART ağları, donanımla desteklenerek gerçek zamanlı öğrenen bilgisayarlar kullanılmaktadırlar.
- ▶ Diğer ağların çoğu çevrimdışı öğrenme yeteneğine sahiptir. Ama gerçek hayatta genellikle durağanlık olmadığından, ağların eğitimlerinin sürekli yenilenmesi gerekmektedir. ART ağları sürekli öğrenme ve beklenmedik durumların üstesinden gelme yetenekleri ile sınırsız karmaşıklık altında çalışabilmektedirler.
- ▶ ART ağları kendi kendine öğrenen yapıya sahiptirler. Hiçbir danışmanın yardımına ihtiyaç duymadıkları gibi, yeni karşılaştıkları bir durumu sınıflandıramazlarsa bu yeni durum için yeni bir sınıf oluşturarak ağın büyümesini sağlarlar. Bu sayede sınırlı bir yapıdan kurtulmuş, ağın kapasitesi dahilinde bir sınırsızlık elde edilmiş olur.
- ▶ ART ağlarında girdi otomatik olarak normalize edilir ve çok düşük veya çok fazla değerdeki girdilerin etkisi azaltılmış olur.



-
- A. Initialize each top-down weight $t_{ij}(0) = 1$;
- B. Initialize bottom-up weight $b_{ji}(0) = \frac{1}{n+1}$;
- C. **While** the network has not stabilized, **do** $n+1$
1. Present a randomly chosen pattern $x = (x_1, \dots, x_n)$ for learning
 2. Let the active set A contain all nodes; calculate

$$y_j = b_{j,1} x_1 + \dots + b_{j,n} x_n \quad \text{for each node } j \in A;$$
 3. **Repeat**
 - a) Let j^* be a node in A with largest y_j , with ties being broken arbitrarily;
 - b) Compute $s^* = (s_1^*, \dots, s_n^*)$ where $s_i^* = t_{ij^*} x_i$;
 - c) Compare similarity between s^* and x with the given vigilance parameter ρ :

$$\text{if } \frac{\sum_{i=1}^n s_i^*}{\sum_{i=1}^n x_i} \leq \rho \text{ then remove } j^* \text{ from set } A$$

else associate x with node j^* and update weights:

$$b_{ji^*}(\text{new}) = \frac{t_{ij^*}(\text{old})x_i}{0.5 + \sum_{i=1}^n t_{ij^*}(\text{old})x_i} \quad t_{ij^*}(\text{new}) = t_{ij^*}(\text{old})x_i$$
- Until** A is empty or x has been associated with some node j
4. If A is empty, then create new node whose weight vector coincides with current input pattern x ;
- end-while**
-

ART Example Computation

For this example, let us assume that we have an ART-1 network with 7 input neurons ($n = 7$) and initially one output neuron ($n = 1$).

Our input vectors are

$\{(1, 1, 0, 0, 0, 0, 1),$
 $(0, 0, 1, 1, 1, 1, 0),$
 $(1, 0, 1, 1, 1, 1, 0),$
 $(0, 0, 0, 1, 1, 1, 0),$
 $(1, 1, 0, 1, 1, 1, 0)\}$

and the vigilance parameter $\rho = 0.7$.

Initially, all top-down weights are set to $t_{1,j}(0) = 1$, and all bottom-up weights are set to $b_{1,j}(0) = 1/8$.

ART Example Computation

For the first input vector, $(1, 1, 0, 0, 0, 0, 1)$, we get:

$$y_1 = \frac{1}{8} \cdot 1 + \frac{1}{8} \cdot 1 + \frac{1}{8} \cdot 0 + \frac{1}{8} \cdot 0 + \frac{1}{8} \cdot 0 + \frac{1}{8} \cdot 0 + \frac{1}{8} \cdot 1 = \frac{3}{8}$$

Clearly, y_1 is the winner (there are no competitors).

Since we have:

$$\frac{\sum_{l=1}^7 t_{l,1} x_l}{\sum_{l=1}^7 x_l} = \frac{3}{3} = 1 > 0.7,$$

the vigilance condition is satisfied and we get the following new weights:

$$b_{1,1}(1) = b_{1,2}(1) = b_{1,7}(1) = \frac{1}{0.5 + 3} = \frac{1}{3.5}$$

$$b_{1,3}(1) = b_{1,4}(1) = b_{1,5}(1) = b_{1,6}(1) = 0$$



ART Example Computation

Also, we have:

$$t_{l,1}(1) = t_{l,0}(0)x_l$$

We can express the updated weights as matrices:

$$B(1) = \begin{bmatrix} \frac{1}{3.5} & \frac{1}{3.5} & 0 & 0 & 0 & 0 & \frac{1}{3.5} \end{bmatrix}^T$$

$$T(1) = [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1]^T$$

Now we have finished the first learning step and proceed by presenting the next input vector.



ART Example Computation

For the second input vector, (0, 0, 1, 1, 1, 1, 0), we get:

$$y_1 = \frac{1}{3.5} \cdot 0 + \frac{1}{3.5} \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + \frac{1}{3.5} \cdot 0 = 0$$

Of course, y_1 is still the winner.

However, this time we do not reach the vigilance threshold:

$$\frac{\sum_{l=1}^7 t_{l,1} x_l}{\sum_{l=1}^7 x_l} = \frac{0}{4} = 0 < 0.7.$$

This means that we have to generate a second node in the output layer that represents the current input.

Therefore, the top-down weights of the new node will be identical to the current input vector.

ART Example Computation

The new unit's bottom-up weights are set to zero in the positions where the input has zeroes as well.

The remaining weights are set to:

$$1/(0.5 + 0 + 0 + 1 + 1 + 1 + 1 + 0)$$

This gives us the following updated weight matrices:

$$B(2) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 1/4.5 & 1/4.5 & 1/4.5 & 1/4.5 & 0 \end{bmatrix}^T$$

$$T(2) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}^T$$



ART Example Computation

For the third input vector, (1, 0, 1, 1, 1, 1, 0), we have:

$$y_1 = \frac{1}{3.5}; \quad y_2 = \frac{4}{4.5}$$

Here, y_2 is the clear winner.

This time we exceed the vigilance threshold again:

$$\frac{\sum_{l=1}^7 t_{l,2} x_l}{\sum_{l=1}^7 x_l} = \frac{4}{5} = 0.8 > 0.7.$$

Therefore, we adapt the second node's weights.

Each top-down weight is multiplied by the corresponding element of the current input.



ART Example Computation

The new unit's bottom-up weights are set to the top-down weights divided by $(0.5 + 0 + 0 + 1 + 1 + 1 + 1 + 0)$.

It turns out that, in the current case, these updates do not result in any weight changes at all:

$$B(3) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 1/4.5 & 1/4.5 & 1/4.5 & 1/4.5 & 0 \end{bmatrix}^T$$

$$T(3) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}^T$$



ART Example Computation

For the fourth input vector, (0, 0, 0, 1, 1, 1, 0), it is:

$$y_1 = 0; \quad y_2 = \frac{3}{4.5}$$

Again, y_2 is the winner.

The vigilance test succeeds once again:

$$\frac{\sum_{l=1}^7 t_{l,2} x_l}{\sum_{l=1}^7 x_l} = \frac{3}{3} = 1 > 0.7.$$

Therefore, we adapt the second node's weights.

As usual, each top-down weight is multiplied by the corresponding element of the current input.



ART Example Computation

The new unit's bottom-up weights are set to the top-down weights divided by

$(0.5 + 0 + 0 + 0 + 1 + 1 + 1 + 0)$.

This gives us the following new weight matrices:

$$B(4) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 0 & 1/3.5 & 1/3.5 & 1/3.5 & 0 \end{bmatrix}^T$$

$$T(4) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}^T$$



ART Example Computation

Finally, the fifth input vector, (1, 1, 0, 1, 1, 1, 0), gives us:

$$y_1 = \frac{2}{3.5}; \quad y_2 = \frac{3}{3.5}$$

Once again, y_2 is the winner.

The vigilance test fails this time:

$$\frac{\sum_{l=1}^7 t_{l,2} x_l}{\sum_{l=1}^7 x_l} = \frac{3}{5} = 0.6 < 0.7.$$

This means that the active set A is reduced to contain only the first node, which becomes the uncontested winner.



ART Example Computation

The vigilance test fails for the first unit as well:

$$\frac{\sum_{l=1}^7 t_{l,1} x_l}{\sum_{l=1}^7 x_l} = \frac{2}{5} = 0.4 < 0.7.$$

We thus have to create a third output neuron, which gives us the following new weight matrices:

$$B(5) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 0 & 1/3.5 & 1/3.5 & 1/3.5 & 0 \\ 1/5.5 & 1/5.5 & 0 & 1/5.5 & 1/5.5 & 1/5.5 & 0 \end{bmatrix}^T$$

$$T(5) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}^T$$



ART Example Computation

In the second epoch, the first input vector, (1, 1, 0, 0, 0, 0, 1), gives us:

$$y_1 = \frac{3}{3.5}; \quad y_2 = 0; \quad y_3 = \frac{2}{5.5}$$

Here, y_1 is the winner, and the vigilance test succeeds:

$$\frac{\sum_{l=1}^7 t_{l,1} x_l}{\sum_{l=1}^7 x_l} = \frac{3}{3} = 1 > 0.7.$$

Since the current input is identical to the winner's top-down weights, no weight update happens.



ART Example Computation

The second input vector, (0, 0, 1, 1, 1, 1, 0), results in:

$$y_1 = 0; \quad y_2 = \frac{3}{3.5}; \quad y_3 = \frac{3}{5.5}$$

Now y_2 is the winner, and the vigilance test succeeds:

$$\frac{\sum_{l=1}^7 t_{l,2} x_l}{\sum_{l=1}^7 x_l} = \frac{3}{3} = 1 > 0.7.$$

Again, because the current input is identical to the winner's top-down weights, no weight update occurs.



ART Example Computation

The third input vector, (1, 0, 1, 1, 1, 1, 0), give us:

$$y_1 = \frac{1}{3.5}; \quad y_2 = \frac{3}{3.5}; \quad y_3 = \frac{4}{5.5}$$

Once again, y_2 is the winner, but this time the vigilance test fails:

$$\frac{\sum_{l=1}^7 t_{l,2} x_l}{\sum_{l=1}^7 x_l} = \frac{3}{5} = 0.6 < 0.7.$$

This means that the active set is reduced to $A = \{1, 3\}$.

Since $y_3 > y_1$, the third node is the new winner.



ART Example Computation

The third node does satisfy the vigilance threshold:

$$\frac{\sum_{l=1}^7 t_{l,3} x_l}{\sum_{l=1}^7 x_l} = \frac{4}{5} = 0.8 > 0.7.$$

This gives us the following updated weight matrices:

$$B(8) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 0 & 1/3.5 & 1/3.5 & 1/3.5 & 0 \\ 1/4.5 & 0 & 0 & 1/4.5 & 1/4.5 & 1/4.5 & 0 \end{bmatrix}^T$$

$$T(8) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}^T$$



ART Example Computation

For the fourth vector, $(0, 0, 0, 1, 1, 1, 0)$, the second node wins, passes the vigilance test, but no weight changes occur.

The fifth vector, $(1, 1, 0, 1, 1, 1, 0)$, makes the second unit win, which fails the vigilance test.

The new winner is the third output neuron, which passes the vigilance test but does not lead to any weight modifications.

Further presentation of the five sample vectors do not lead to any weight changes; the network has thus stabilized.



-
- ▶ **Öztemel, E.**, 2003. *Yapay Sinir Ağları*, Papatya Yayıncılık, İstanbul.
 - ▶ Yapay Sinir Ağları ve Tahmin Modellemesi Üzerine Bir Uygulama, 2006 , İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü.
 - ▶ <https://www.cs.umb.edu/~marc/cs672/net11-23.pdf>
[Erişim Tarihi: 02.01.2021]

