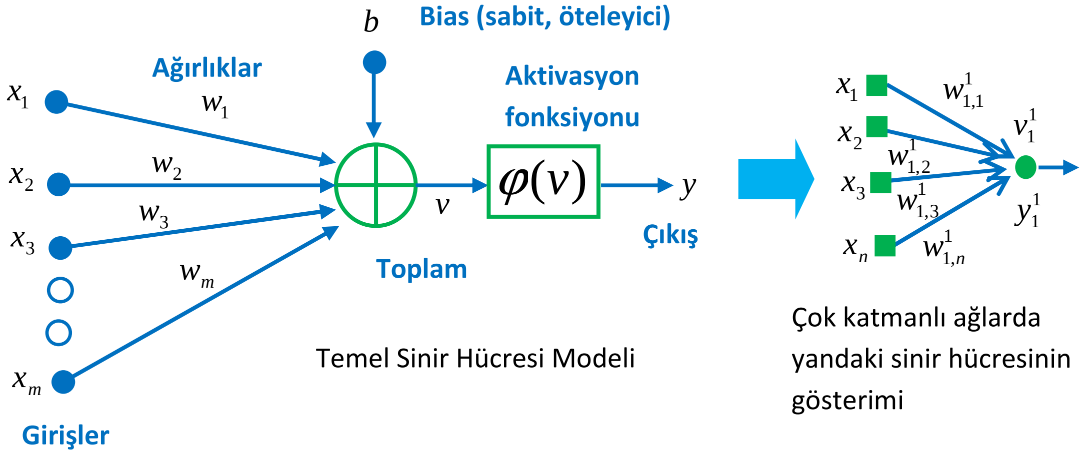
Yapay Sinir Ağları Temelleri II

Çok katmanlı Sinir Ağları:

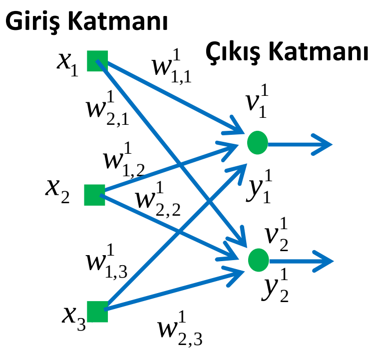
Tek bir sinir hücresi sınırlı sayıda bağlantı sağlar ve tek bir sinir hücresi ile oluşturulan öğrenme modelinin öğrenme kabiliyeti sınırlıdır. Yapay sinir ağı modelinin temsil edebileceği fonksiyon çeşitliliğini artırabilmesi (öğrenen model karmaşıklığını artırılması bu modelin elastisitesini artırır) çok sayıda sinir hücresinin birbiri ile belli bir düzende bağlantılandığı çok katmanlı sinir ağları ile sağlanır. Katman sayısı arttıkça daha karmaşık matematiksel ilişkilerin öğrenilebileceği görülmüştür. Ağın karmaşıklığının artışı hesaplama güçlüğü yanında ağın elle çiziminde sinir hücresinin bütün detayları ile gösterilmesini de zorlaştırır. Çok katmalı sinir ağlarını daha kolay çizebilmek için her bir sinir hücresi modeli (Ağırlıklı toplam ve aktivasyon fonksiyonu işlem grubu) bir düğüm noktası ile temsil edilir ve sinir hücreleri katmanlar halinde gruplanarak sinir ağları oluşturulur.



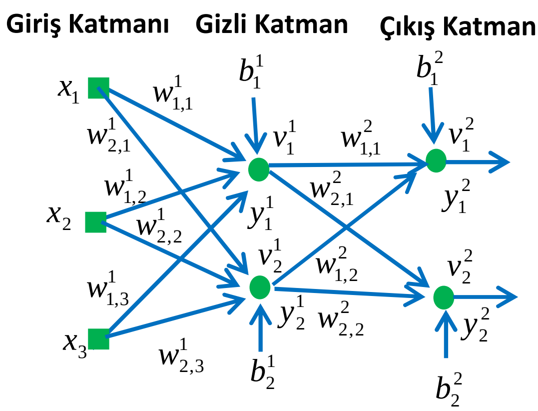
Temel Sinir Hücresi Modeli

Çok katmanlı ağlarda yandaki sinir hücresinin gösterimi

Aşağıda tek katmanlı ve çok katmanlı sinir ağlarına örnekler verilmiştir. Tek katmanlı sinir ağlarında bir giriş katmanı ve bir çıkış katmanı bulunur. Çok katmanlı sinir ağlarında ek olarak gizli katmanlar bulunur. Aşağıda çizilen tek katmanlı veya çok katmalı sinir ağ modelleri için bilgi(veri) ağın girişinden çıkışına doğru işlendiği (iletildiği) için bu tür ağlara ileri beslemeli (feedforward) ağ adı verilir. Aşağıdaki modellerde katmanlar arasında bütün sinir hücreleri birbiri ile bağlantılı olduğu için tam bağlı (fully connected) ileri beslemeli ağ olarak katagorize ediebilir.



Tek katmanlı yapay sinir ağı: 1 giriş katmanı + bir çıkış katmanı



1 **Çok katmanlı yapay sinir ağı**: 1 giriş katmanı + bir gizli katman+bir çıkış katmanı

Yukarıdaki temsilde,

x_j : j . girişi ifade eder. Örnek x_1 : birinci giriş, x_2 : ikinci giriş,

v_i^k : k . katman i . nörona (sinir hücresi) ait ağırlıklı toplam. Örnek v_2^2 : 2. katman ait 2. nöronun ağırlıklı toplamı

y_i^k : k . katman i . nörona ait çıkış. Örnek y_3^1 : 1. katman ait 3. neronun çıkışı

$w_{i,j}^k$: k . katman i . nöronuna j . girişinden gelen (yada gizli katmanda bir önceki katmanın j çıkışından) gelen ağırlık katsayısı

b_i^k : k . katman i . nörona bias katsayısı. Örnek b_1^2 : 2. katman ait 1. nöronun biası

Şimdi yukarıda verilen çok katmanlı ağ için matematiksel modeli (ağırlık toplam ve aktivasyon fonksiyonu) ifade edelim. Bu işlemi iki aşamalı olarak daha kolay ifade edebiliriz:

1) Birinci katman için (Girişlere bağlı katman)

1. katman i . nöron için ağılıklı toplam ağın girişlerine bağlı olarak yazılır.

$$\boxed{v_i^1 = \sum_{j=1}^{m} x_j w_{i,j}^1 + b = x_1 w_{i,1}^1 + x_2 w_{i,2}^1 + x_3 w_{i,3}^1 + .. + x_m w_{i,m}^1 + b_i^1}$$

Burada m giriş sayıdır. Bu ifade matris formunda da ifade edilebilir,

$$v_i^1 = x_1 w_{i,1}^1 + x_2 w_{i,2}^1 + x_3 w_{i,3}^1 + .. + x_m w_{i,m}^1 + b_i^1 = \begin{bmatrix} w_{i,1}^1 & w_{i,2}^1 & .. & w_{i,m}^1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_m \end{bmatrix} + b = W_i^1 X + b_i^1$$

$$\boxed{v_i^k = W_i^1 X + b_i^1}$$

elde edilir. Burada vektörler $W_i^1 = \begin{bmatrix} w_{i,1}^1 & w_{i,2}^1 & .. & w_{i,m}^1 \end{bmatrix}$ ve $X = \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_m \end{bmatrix}$ olarak ifade edilir.

Çıkışı yazılırsa,(Çıkış ağılıklı toplamın aktivasyon fonksiyonunda değerlendirilmesi ile elde edilir.)

$$\boxed{y_i^1 = \varphi(v_i^1)}$$

2) İkinci ve sonraki katmanlar için yazılırsa

2. katman ve sonraki katmanlarda i . nöron için ağırlıklı toplam bir önceki katman çıkışlarına bağlı olduğu için

$$\boxed{v_i^k = \sum_{j=1}^{p} y_j^{k-1} w_{i,j}^k + b_i^k = y_1^{k-1} w_{i,1}^k + y_2^{k-1} w_{i,2}^k + y_3^{k-1} w_{i,3}^k + .. + y_p^{k-1} w_{i,p}^k + b_i^k}$$

yazılır. Burada k katman indisi gizli ve ara katmalar için $k > 1$ değer alır. Ayrıca, y_j^{k-1} in bir önceki katmanın çıkışı olduğunu farkediniz. Ara katmanlarda bir önceki sinir hücresi çıkışı bir sonraki sinir hücresi için giriş olarak kabul edilir. Bu ifade matris formunda yazılırsa,

$$v_i^k = y_1^{k-1} w_{i,1}^k + y_2^{k-1} w_{i,2}^k + y_3^{k-1} w_{i,3}^k + .. + y_p^{k-1} w_{i,p}^k + b_i^k = \begin{bmatrix} w_{i,1}^k & w_{i,2}^k & .. & w_{i,p}^k \end{bmatrix} \begin{bmatrix} y_1^{k-1} \\ y_2^{k-1} \\ \vdots \\ y_p^{k-1} \end{bmatrix} + b_i^k = W_i^k Y^{k-1} + b_i^k$$

$$\boxed{v_i^k = W_i^k Y^{k-1} + b_i^k}$$

elde edilir. Burada vektörler $W_i^k = \begin{bmatrix} w_{i,1}^k & w_{i,2}^k & .. & w_{i,m}^k \end{bmatrix}$ ve $Y^{k-1} = \begin{bmatrix} y_1^{k-1} \\ y_2^{k-1} \\ \vdots \\ y_p^{k-1} \end{bmatrix}$

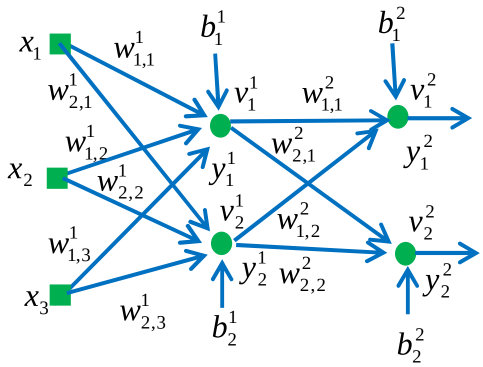
Çıkışı yazılırsa,

$$\boxed{y_i^k = \varphi(v_i^k)}$$

ÖNEMLİ NOT: İhtiyaç duyulduğunda, özellikle program yazarken indislerin yazılmasında bu formüller faydalı olacaktır ve gerektiğinde kaynaklardan bakılarak kullanılabilir. Bu nedenle formüllerin ezberlenmesi zorunlu değildir. Ağ üzerinde doğru indislere sahip değişkenler yazılırsa, girişten çıkışa <u>işaret(bilgi, veri) akış yönü ve ilgili matematiksel işlemleri takip ederek formül kolayca yazılabilir. Ağın şeması ve üzerinde değişkenlerin uygun şekilde belirtilmesi ağın matematiksel modelinin yazılması için yeterlidir.</u>

El ile çözümde ise kolay yol bağlantılar üzerinden <u>işaret akışını(değer akışı) ve ağın işlemlerini (ağırlıklı toplam ve aktivasyon fonksiyonu) takip ederek formül kullanmaya gerek kalmadan ağın çıkışına kadar hesaplamalar yapılabilir</u>. Aşağıdaki örneklerde sinir ağı üzerinde verilen değerler kullanılarak giriş değerinin bir işaret gibi çıkışa ulaşana kadar maruz kalacağı işlemlerin nümerik olarak hesaplanması gösterilmiştir. Bu noktada, sadece ağ üzerinde işaret(değer) akışlarının izlenmesi ve işlemlerin uygulanmasının yeterli olduğu görülür.

Örnek: Aşağıda verilen çok katmanlı ağ için ağırlık toplamları yazınız ve lineer aktivasyon fonksiyonu ($\varphi(v) = v$) kullanılması durumunda her bir sinir hücresi çıkışlarını ifade eden matematiksel modeli yazınız.



Burada yapılması gereken ağ üzerinde değişkenleri doğru şekilde yazdıktan sonra girişten-çıkışa işlem akışı takip edilerek ağırlıklı toplam ve nöron çıkışlarını yazmak.

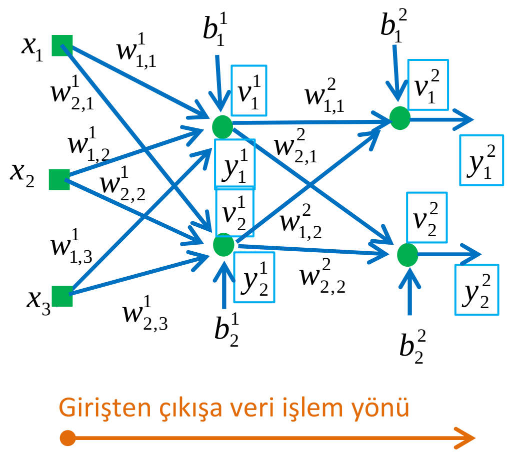
Birinci katman nöronları için yazalım:

$$v^1_1 = x_1 w^1_{1,1} + x_2 w^1_{1,2} + x_3 w^1_{1,3} + b^1_1$$

$$y^1_1 = \varphi(v^1_1) = v^1_1 \;\; => \boxed{y^1_1 = x_1 w^1_{1,1} + x_2 w^1_{1,2} + x_3 w^1_{1,3} + b^1_1}$$

$$v^1_2 = x_1 w^1_{2,1} + x_2 w^1_{2,2} + x_3 w^1_{2,3} + b^1_2$$

$$y^1_2 = \varphi(v^1_2) = v^1_2 \;\; => \boxed{y^1_2 = x_1 w^1_{2,1} + x_2 w^1_{2,2} + x_3 w^1_{2,3} + b^1_2}$$



İkinci katman nöronları için yazalım:

$$v^2_1 = y^1_1 w^2_{1,1} + y^1_2 w^2_{1,2} + b^2_1$$

$$y^2_1 = \varphi(v^2_1) = v^2_1 \;\; => \boxed{y^2_1 = y^1_1 w^2_{1,1} + y^1_2 w^2_{1,2} + b^2_1}$$

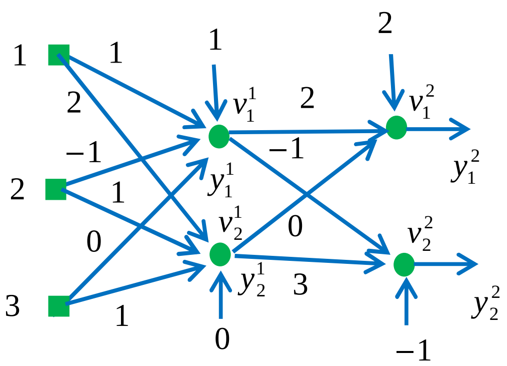
$$v^2_2 = y^1_1 w^2_{2,1} + y^1_2 w^2_{2,2} + b^2_2$$

$$y^2_2 = \varphi(v^2_2) = v^2_2 \;\; => \boxed{y^2_2 = y^1_1 w^2_{2,1} + y^1_2 w^2_{2,2} + b^2_2}$$

Not: Eğer son katmanda lineer aktivasyon fonksiyonu yerine sigmoid aktivasyon fonksiyonu $\varphi(v) = \dfrac{1}{1+e^{-v}}$ kullanılsaydı,

$$y_1^2 = \frac{1}{1+e^{-(y_1^1 w_{1,1}^2 + y_2^1 w_{1,2}^2 + b_1^2)}} \text{ ve } y_2^2 = \frac{1}{1+e^{-(y_1^1 w_{2,1}^2 + y_2^1 w_{2,2}^2 + b_2^2)}}$$ olacağını gösteriniz. Bu durumda modelin lineer olmayan bir çıkış karakteristiğine sahip olacağı görünüz.

Örnek: Aşağıdaki çok katmanlı ileri beslemeli yapay sinir ağında lineer aktivasyon fonksiyonu kullanılmıştır. Bu sinir ağının çıkışındaki değerlerini elde ediniz.



Burada yapılması gereken ağ üzerinde verilen değerleri kullanarak akışa göre ağırlıklı toplam ve nöron çıkışlarını yazmak.

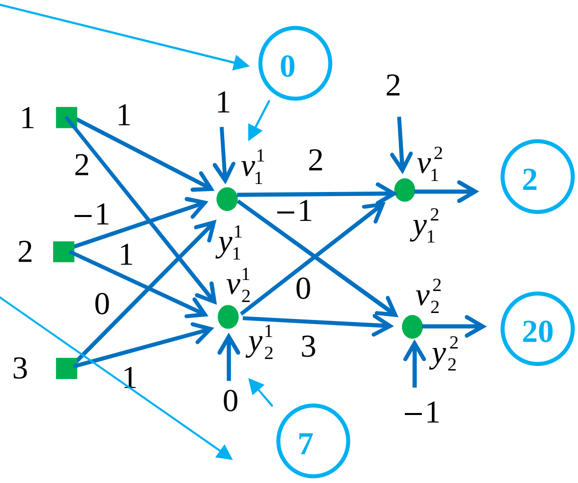
Birinci katman nöronları için yazalım:

$v_1^1 = 1*1 + 2*(-1) + 3*0 + 1 = 1 - 2 + 0 + 1 = 0$

Lineer aktivasyon fonksiyonu için $y_1^1 = v_1^1 \Rightarrow y_1^1 = 0$

$v_2^1 = 1*2 + 2*1 + 3*1 + 0 = 2 + 2 + 3 = 7$

Lineer aktivasyon fonksiyonu için $y_2^1 = v_2^1 \Rightarrow y_2^1 = 7$



İkinci katman nöronları için yazalım:

$v_1^2 = 0*2 + 7*0 + 2 = 0 + 0 + 2 = 2$

$y_1^2 = v_1^2 \Rightarrow y_1^2 = 2$

$v_2^2 = 0*(-1) + 7*3 + -1 = 1 + 21 - 1 = 20$

$y_2^2 = v_2^2 \Rightarrow y_2^2 = 20$