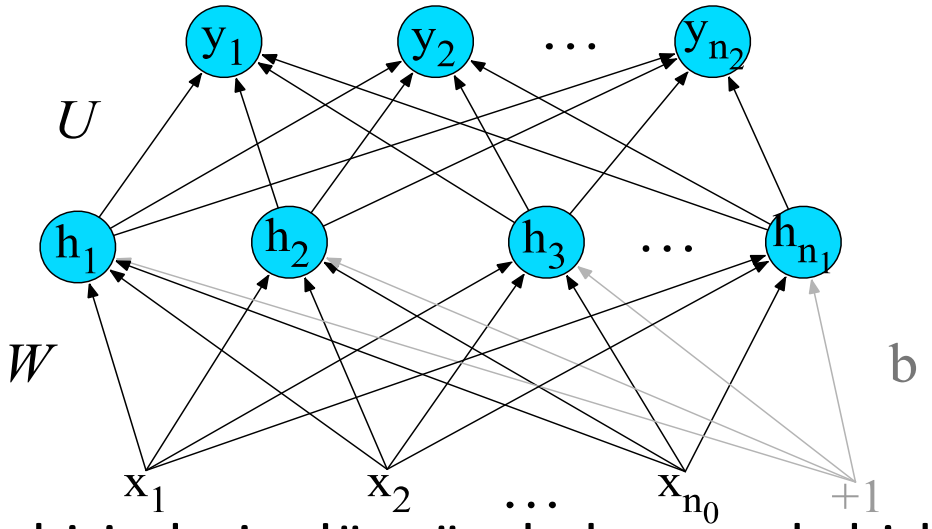
İleri beslemeli Sinir Ağları/Feedforward Neural Networks



İleri beslemeli ağ, birimlerin döngüsel olmayarak birbirlerine bağlı olduğu çok katmanlı bir ağdır; her katmandaki birimlerden gelen çıktılar bir üst katmandaki birimlere iletilir ve **alt katmanlara hiçbir çıktı geri iletilmez.**

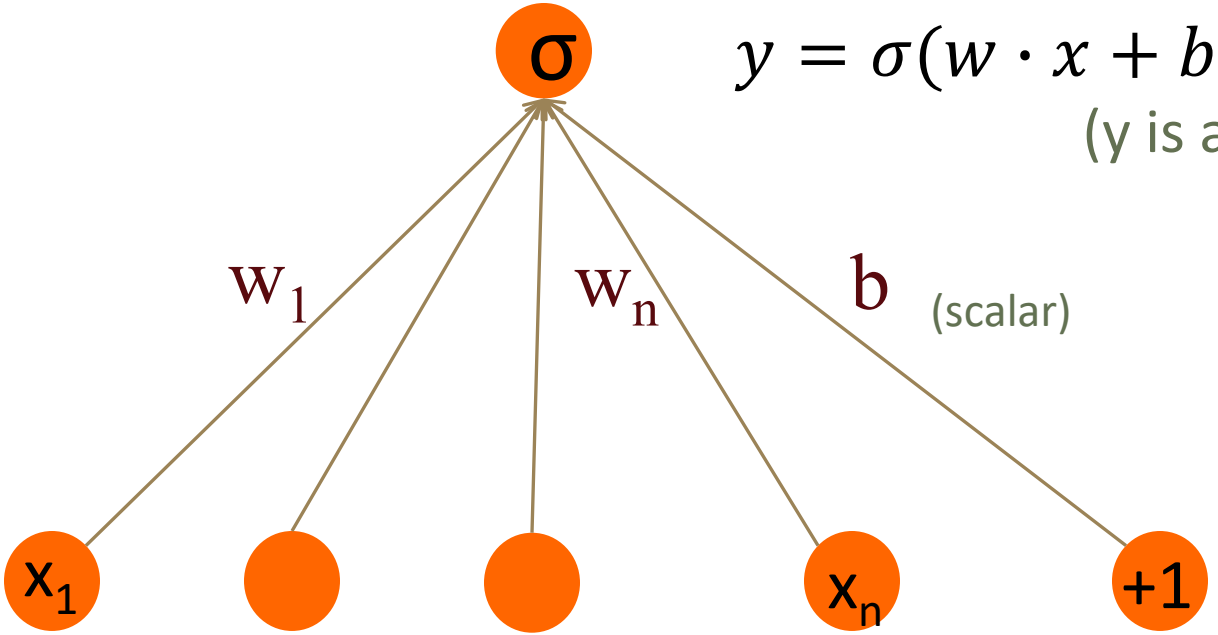
İleri beslemeli ağlara çok katmanlı algılayıcılar /multi-layer perceptrons (MLP) adı verilir.

İkili Lojistik Regresyon (1- düzeyli ağ betimlemesi)

Output layer
(σ node)

$$y = \sigma(w \cdot x + b)$$

(y is a scalar)



Input layer
vector x

Giriş tabakası (inpur layer) düzey olarak sayılmaz.

İkili Logistik Regresyon

- Lojistik regresyon hesaplamalarında ikili lojistik regresyon tek katmanlı bir ağ olarak düşünülebilir.
- Giriş değerlerinin bulunduğu düzey, katman olarak alınmaz.
- x vektörü olarak adlandırılan ve katman 0 giriş katmanıdır. adlandırılır.
- x vektörü ağırlık matrisi olan w ile çarpılır, doğrusal sapma/önyargı b değeri eklenir.
- 1. katman, çıkış katmanı, wx + b değerinin sigmoid fonksiyonu olarak bir skaler bir y değeri verecektir.

Lojistik Regresyon

Doğal dil işlemede, lojistik regresyon sınıflandırma için temel denetimli (supervised) makine öğrenme algoritmasıdır.

Logistic regresyon sinir ağlarıyla da çok yakın bir ilişkiye sahiptir.

Lojistik regresyon, bir gözlemi iki sınıftan birine sınıflandırmak için kullanılabilir.

Bazen ikiden fazla sınıfa ihtiyaç olur.

3 yönlü duygu sınıflandırması (olumlu, olumsuz veya nötr) istenebilir.

Denetimli (Supervised) Öğrenme Nedir?

- Makine öğrenmesinde modelin etiketlenmiş verilerle eğitilmesidir.
  - Hem girdiler (örneğin sorular) hem de çıktılar (örneğin soruların doğru cevapları) mevcuttur.
- Lojistik regresyon, denetimli öğrenmede sınıflandırma problemlerini çözen algoritmalardan biridir.
  - Bir verinin belirli bir sınıfa ait olma olasılığını hesaplar.
- Diğer denetimli öğrenme algoritması regresyondur.
  - Süreklilik gösteren sayısal değerler tahmin edilir.

Logistik Regresyon (devam)

- Herhangi bir kelimenin sözcük tipi 10, 30 ya da 50 farklı sözcük türü arasından seçilebilir
- Diğer bir ifade ile bir ifadenin adlandırılmış varlık (named entity) türünün, kişi, yer, kuruluş gibi etiketler arasından seçim yaparak belirlenmesidir.
- Bu tip problemlerde, softmax regresyonu olarak da adlandırılan çok terimli lojistik regresyon kullanılır.

Çok Terimli Lojistik Sınıflandırıcı

- $p(y_k = 1/x)$ değerini hesaplamak için sigmoid fonksiyonunun genelleltirilmiş biçimi olan softmax fonksiyonu kullanılır.

- Softmax fonksiyonu, k adet rastgele değerden oluşan bir vektör olan $z = [z_1\ z_2 \ldots\ z_K]'$yı alır ve bunları her bir değerin [0,1] aralığında olduğu ve tüm değerlerin toplamının 1 olduğu bir olasılık dağılımına eşler.

Softmax Fonksiyonu: Genelleştirilmiş Sigmoid fonksiyonu

k boyutlu bir z vektörü için softmax aşağıdaki gibi hesaplanır:

$$\text{softmax}(z) = \left[\frac{\exp(z_1)}{\sum_{i=1}^{k}\exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^{k}\exp(z_i)}, \ldots, \frac{\exp(z_k)}{\sum_{i=1}^{k}\exp(z_i)}\right]$$

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{k}\exp(z_j)} \quad 1 \leq i \leq k$$

Örneğin $z = [0.6, 1.1, -1.5, 1.2, 3.2, -1.1]$ vektörü verildiğinde

$\text{softmax}(z) = [0.055, 0.090, 0.006, 0.099, 0.74, 0.010]$ olarak hesaplanır.

Nasıl hesaplanır?

- k boyutlu bir z vektörü için softmax şu şekilde tanımlansın.
- Herhangi bir model için bir resmin Kedi, Köpek, Karga, Keçi, Kuzu, Kuzgun olduğuna dair şu ham skorlar, logitler aşağıdaki şekilde üretilsin.

$$z = [0.6, 1.1, -1.5, 1.2, 3.2, -1.1]$$

$$\text{softmax}(z) = [0.05, 0.09, 0.01, 0.1, 0.74, 0.01]$$

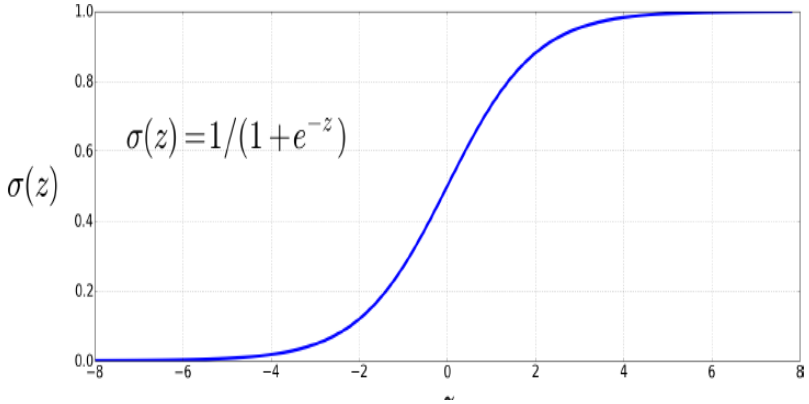
Logit nedir?

- Bir yapay sinir ağının veya regresyon modelinin "ham değerleridir.
- Veriyi herhangi bir model ile işleyip 0 ile 1 arasında bir olasılığa dönüştürmeden hemen önceki aşamada ürettiği işlenmemiş, sınırsız sayısal değerlerdir.
- $z = w \cdot x + b$, sigmoid fonksiyonunun girdisidir ve logit denir.
- logit fonksiyonunun sigmoid fonksiyonunun tersidir.
- Logit fonksiyonu, p/1-p olasılık oranının logaritmasıdır

$$\operatorname{logit}(p) = \sigma^{-1}(p) = \ln \frac{p}{1-p}$$

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+\exp(-z)}$$

Sigmoid fonksiyonu



$$\begin{aligned} P(y=1) &= \sigma(\mathbf{w}\cdot\mathbf{x}+b) \\ &= \frac{1}{1+\exp(-(\mathbf{w}\cdot\mathbf{x}+b))} \\ P(y=0) &= 1-\sigma(\mathbf{w}\cdot\mathbf{x}+b) \\ &= 1-\frac{1}{1+\exp(-(\mathbf{w}\cdot\mathbf{x}+b))} \\ &= \frac{\exp(-(\mathbf{w}\cdot\mathbf{x}+b))}{1+\exp(-(\mathbf{w}\cdot\mathbf{x}+b))} \end{aligned}$$

Ağırlıklı özelliklerin toplamı sigmoid fonksiyonuna uygulanarak , 0 ile 1 arasında bir sayı elde edilir. Bunun bir olasılığa dönüştürülmesi, p(y = 1) ve p(y = 0) durumlarının toplamının 1'e eşit olarak hesaplanmasıdır.

Multinomial Logistic Regression as a 1-layer Network

Fully connected single layer network

Output layer
(softmax nodes)

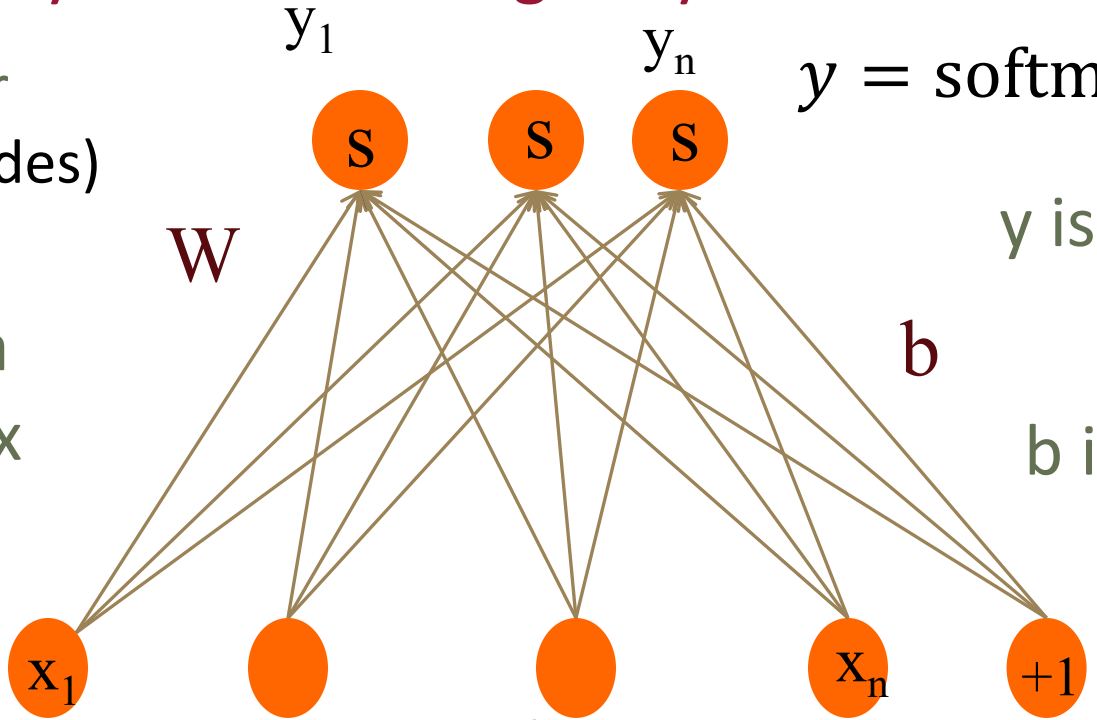
$$y = \text{softmax}(Wx + b)$$

y is a vector

W is a matrix

b is a vector

Input layer
scalars



Multinomial Logistic Regression have a softmax to turn the output values into probabilities instead of a single sigmoid at the output

Softmax Fonksiyonu: Genelleştirilmiş Sigmoid

k boyutlu bir z vektörü için softmax şu şekilde tanımlanır:

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)} \quad 1 \leq i \leq k$$

Sigmoid fonksiyonu gibi, bu da üstel bir fonksiyondur. k boyutlu bir z vektörü için softmax şu şekilde tanımlanır:

:

$$\text{softmax}(z) = \left[\frac{\exp(z_1)}{\sum_{i=1}^{k} \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^{k} \exp(z_i)}, \ldots, \frac{\exp(z_k)}{\sum_{i=1}^{k} \exp(z_i)} \right]$$

$z = [0.6, 1.1, -1.5, 1.2, 3.2, -1.1]$

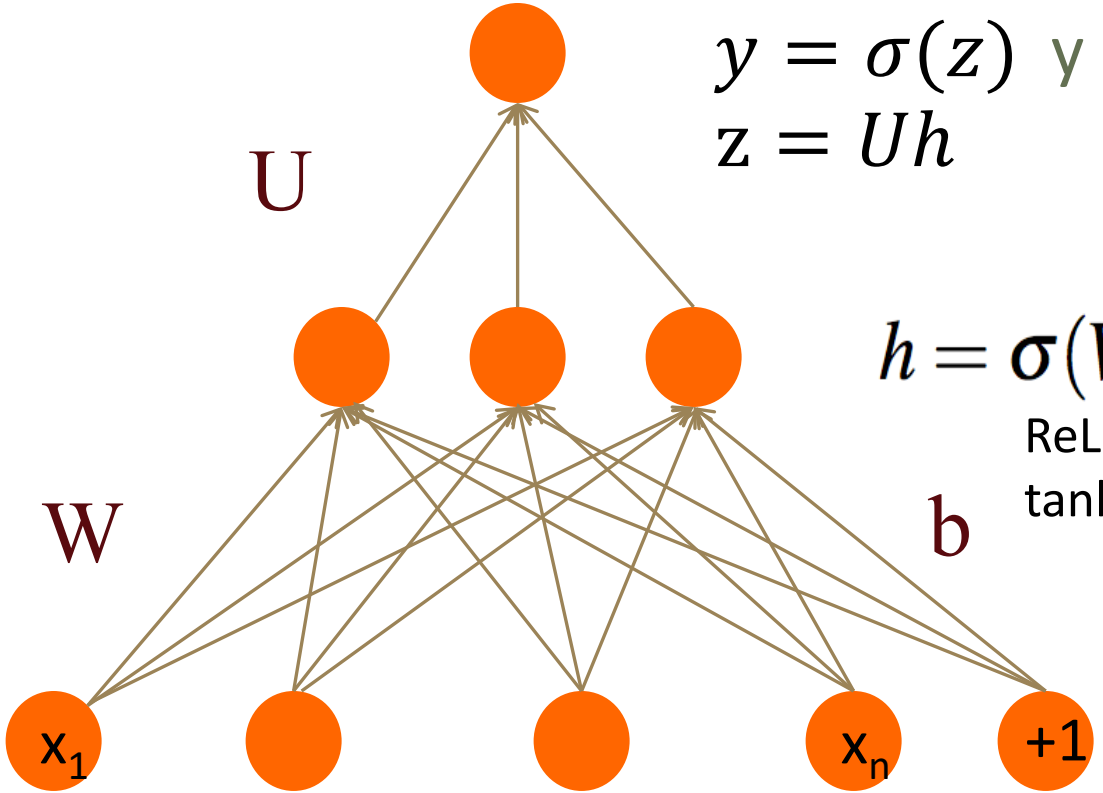
$$\text{softmax}(z) = [0.055, 0.090, 0.006, 0.099, 0.74, 0.010]$$

Two-Layer Network with scalar output
İki Katmanlı Ağlar

Output layer
(σ node)

hidden units
(σ node)

Input layer
(vector)



$y = \sigma(z)$ y is a scalar

$z = Uh$

$h = \sigma(Wx + b)$

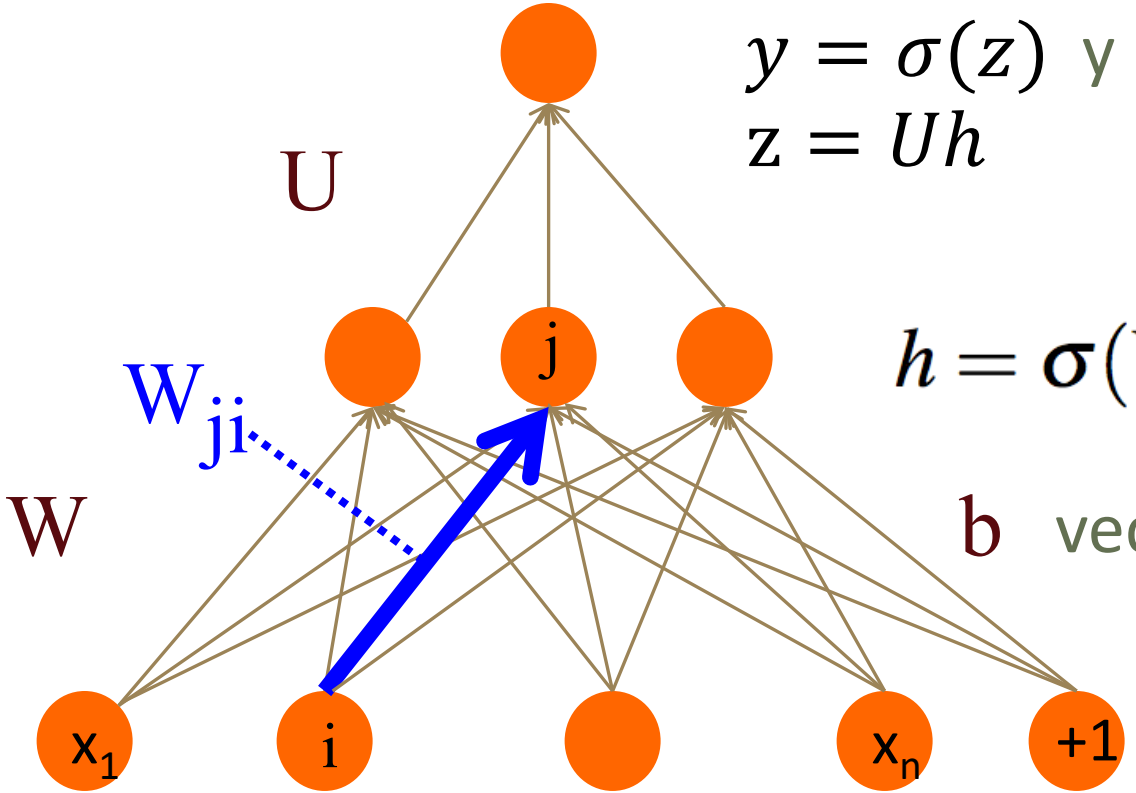
ReLU ya da
tanh olabilirdi

Two-Layer Network with scalar output

Output layer
(σ node)

hidden units
(σ node)

Input layer
(vector)



$y = \sigma(z)$ y is a scalar

$z = Uh$

$h = \sigma(Wx + b)$

b vector

Each element *Wji* of the weight matrix *W* represents the weight of the connection from the *i*th input unit *xi* to the *j*th hidden unit *hj*.

Two-Layer Network with scalar output

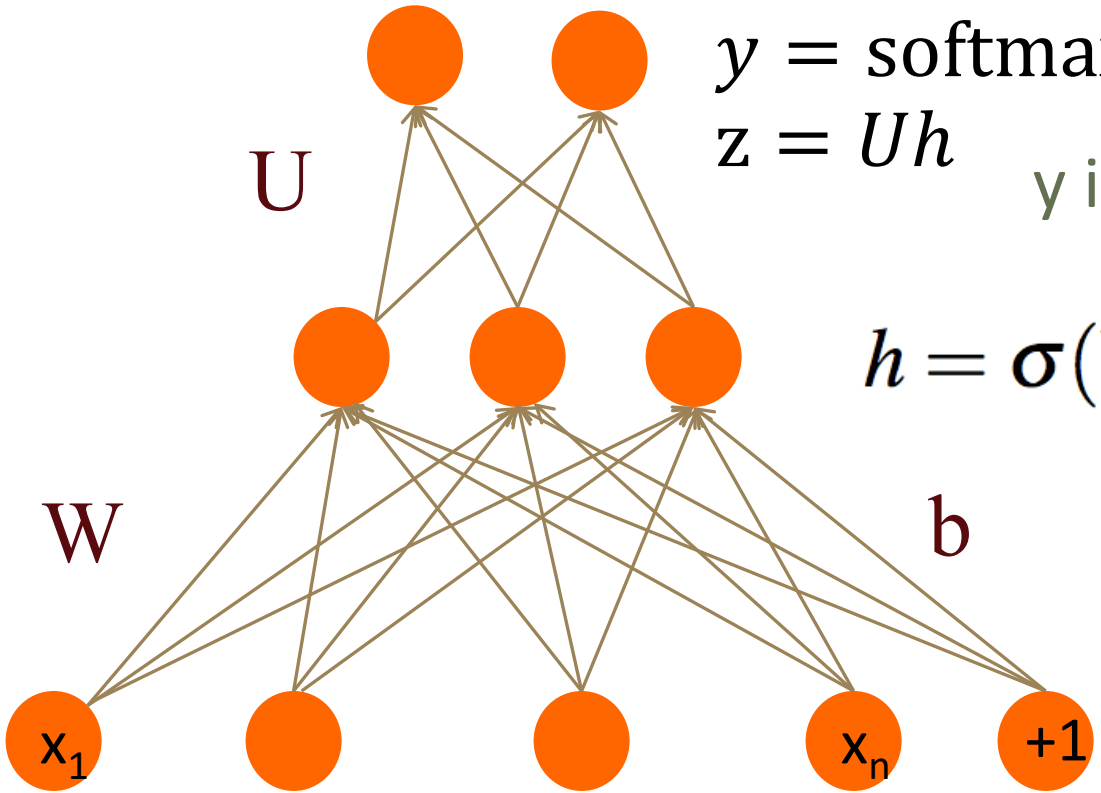
- σ fonksiyonu bir vektöre uygulanır. Bu, σ(·)'nin vektöre eleman eleman uygulanmasıdır. g[z1,z2,z3] = [g(z1),g(z2),g(z3)]
- XOR fonk. olması durumunda elde edilen h değeri (gizli katman) girişin bir değerini (betimlenişini) oluşturur.
- Çıkış katmanının rolü, bu yeni değer h'yi alıp bir çıktı hesaplamaktır.
- Çıktı gerçek değerli bir sayı olacaktır.
- Ağın amacı bir sınıflandırmaya karar vermektir.
- Duygu sınıflandırması gibi ikili bir görev yapılıyorsa, tek bir çıkış düğümü olabilir ve değeri y, olumlu ve olumsuz duygu olasılığıdır.
- Çıkış katmanı bir ağırlık matrisi, yani U matrisini içerir.
- Modellerin bir kısmı çıkış katmanında önyargı vektörü b içermez.
- Ağırlık matrisi, ara çıktı z'yi üretmek için giriş vektörü (h) ile çarpılır (z=Uh)
- Elde edilen değeri olasılığa dönüştürmek için sigmoid fonksiyonunu kullanılır.

Two-Layer Network with softmax output

Output layer
(σ node)

hidden units
(σ node)

Input layer
(vector)



$$y = \text{softmax}(z)$$
$$z = Uh$$

y is a vector

$$h = \sigma(Wx + b)$$

ReLU
ya da tanh
olabilir.

Çok terimli Sınıflandırma

- İkiden fazla sınıflandırma yapmaktır.

  Sözcük türü etiketi atamak gibi, daha fazla çıktı birimi eklenir.

- Ekleme her sınıf için bir tanedir. Yani, her potansiyel sözcük türü için bir çıktı düğümü olabilir ve bu düğümün çıktı değeri, o sözcük türünün olasılığıdır.

Çıktı katmanı, çıktı düğümleri arasında bir olasılık dağılımı verir.

- Bu dağılımı hesaplamak için, çok terimli lojistik regresyonda olduğu gibi softmax kullanılır.

  Tek gizli katmanlı bir sinir ağı sınıflandırıcısını, girdinin gizli katman temsili olan bir h vektörünü oluşturmak ve ağın h'deki özelliklerini standart lojistik regresyon olarak çalıştırmaktır.

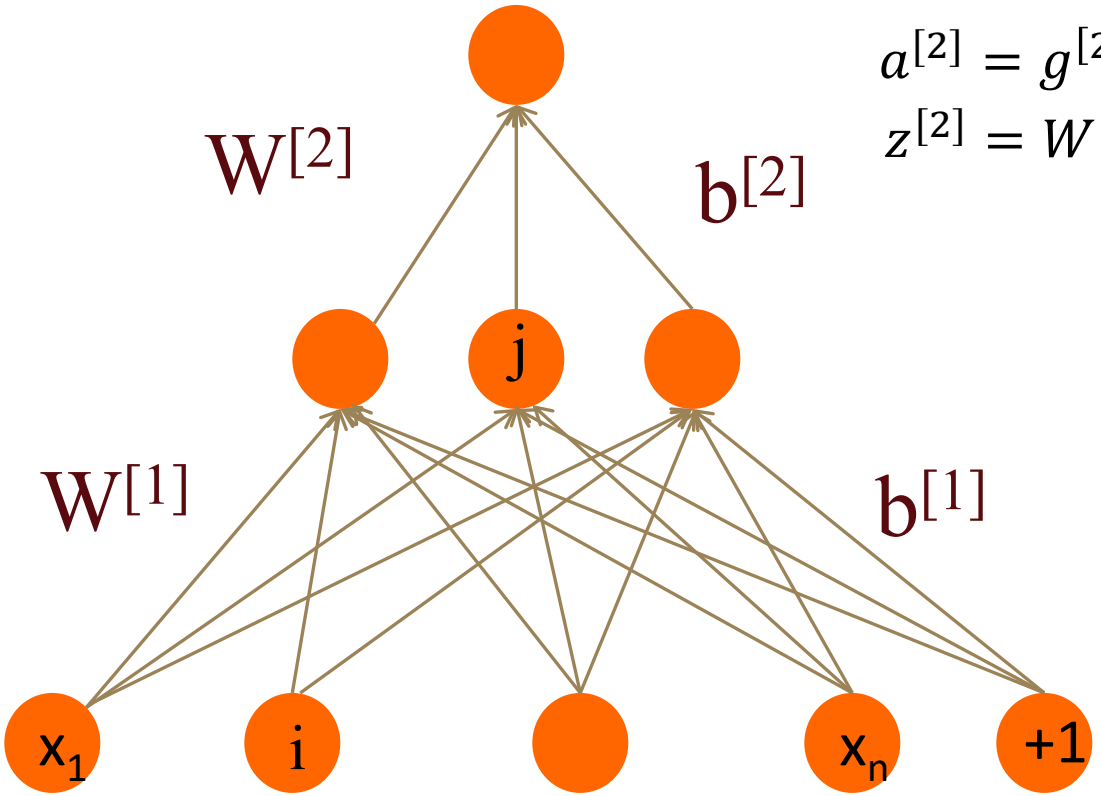
Sinir Ağları ve Lojistik Regresyon İlişkisi

- Bir sinir ağı, lojistik regresyona benzer. Ama bazı farklılıkları vardır.
  - Sinir ağları birçok katmana sahip olduğu bilinmektedir. Derin bir sinir ağlarının katman katman lojistik regresyon sınıflandırıcılarına benzemeleri ortak özelliklerinden biridir.
  - Özellik şablonlarıyla (feature templates) özellikleri oluşturmak yerine, ağın önceki katmanları özellik (feature) betimlemelerini (değerlerini) kendileri oluşturur.

Derin Ağlar için Notasyon

- Katman numaraları köşeli parantez içindeki üst simgelerdir.
- Giriş katmanı 0'dan başlar.
- $W^{[1]}$, (birinci) gizli katman için ağırlık matrisi, $b^{[1]}$ ise (birinci) gizli katman için sapma vektörüdür.
- Ara katmanlar için ReLU veya tanh kullanılır.
- Çıkış katmanları için ise softmax özelliğinde bir aktivasyon fonksiyon $g(\cdot)$ şeklinde belirtilir.
- i. katmandan gelen çıktı $a^{[i]}$, ağırlıkların ve sapmaların kombinasyonu $z^{[i]}$ olarak betimlenir.
- Girişler x'i daha genel olarak $a^{[0]}$ olarak adlandırılır.

Multi-layer Notation



$$y = a^{[2]}$$

$$a^{[2]} = g^{[2]}(z^{[2]})$$ sigmoid ya da softmax

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$

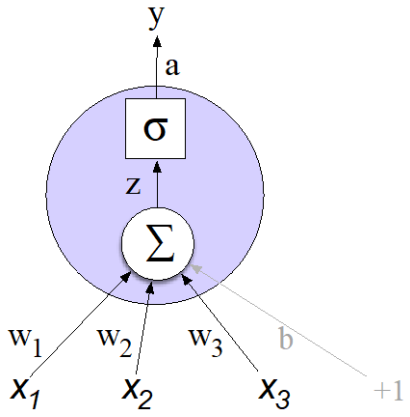
$$a^{[1]} = g^{[1]}(z^{[1]})$$ ReLU

$$z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$$

$a^{[0]}$

Multi Layer Notation

$$
\begin{aligned}
z^{[1]} &= W^{[1]}a^{[0]} + b^{[1]} \\
a^{[1]} &= g^{[1]}(z^{[1]}) \\
z^{[2]} &= W^{[2]}a^{[1]} + b^{[2]} \\
a^{[2]} &= g^{[2]}(z^{[2]}) \\
\hat{y} &= a^{[2]}
\end{aligned}
$$

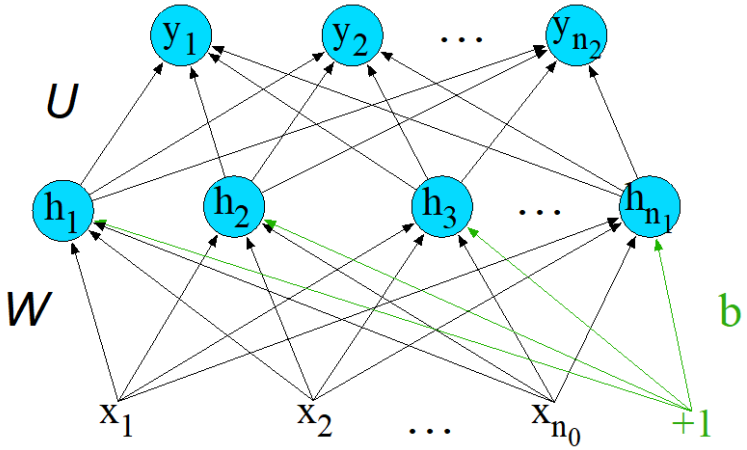
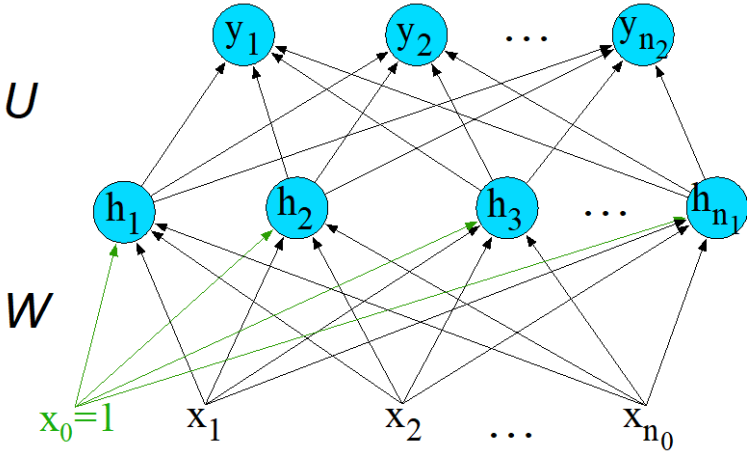


for i **in** 1..n

$$
\begin{aligned}
z^{[i]} &= W^{[i]}\, a^{[i-1]} + b^{[i]} \\
a^{[i]} &= g^{[i]}(z^{[i]}) \\
\hat{y} &= a^{[n]}
\end{aligned}
$$

Replacing the bias unit

Instead of:



do this: