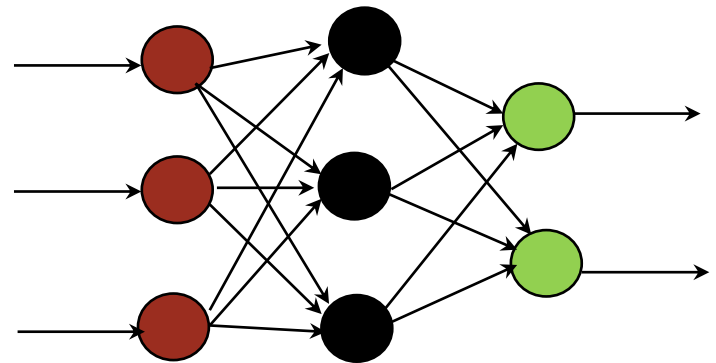


Класификация 3

Невронни мрежи

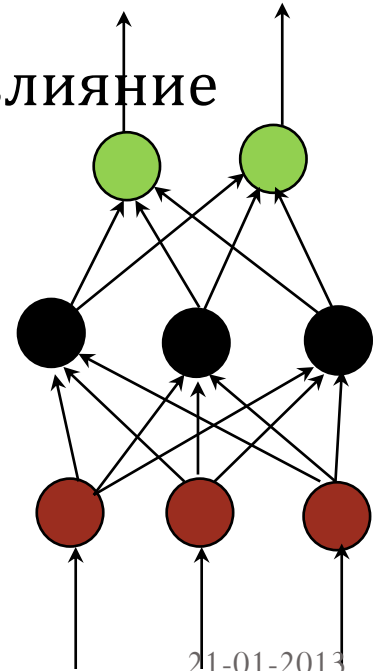
Невронни мрежи

- Модел на мрежа от елементи, наподобяващи неврони
- Елементите са свързани помежду си
- Три вида елементи:
 - **ВХОДНИ**
 - **ИЗХОДНИ**
 - **междинни (скрити)**

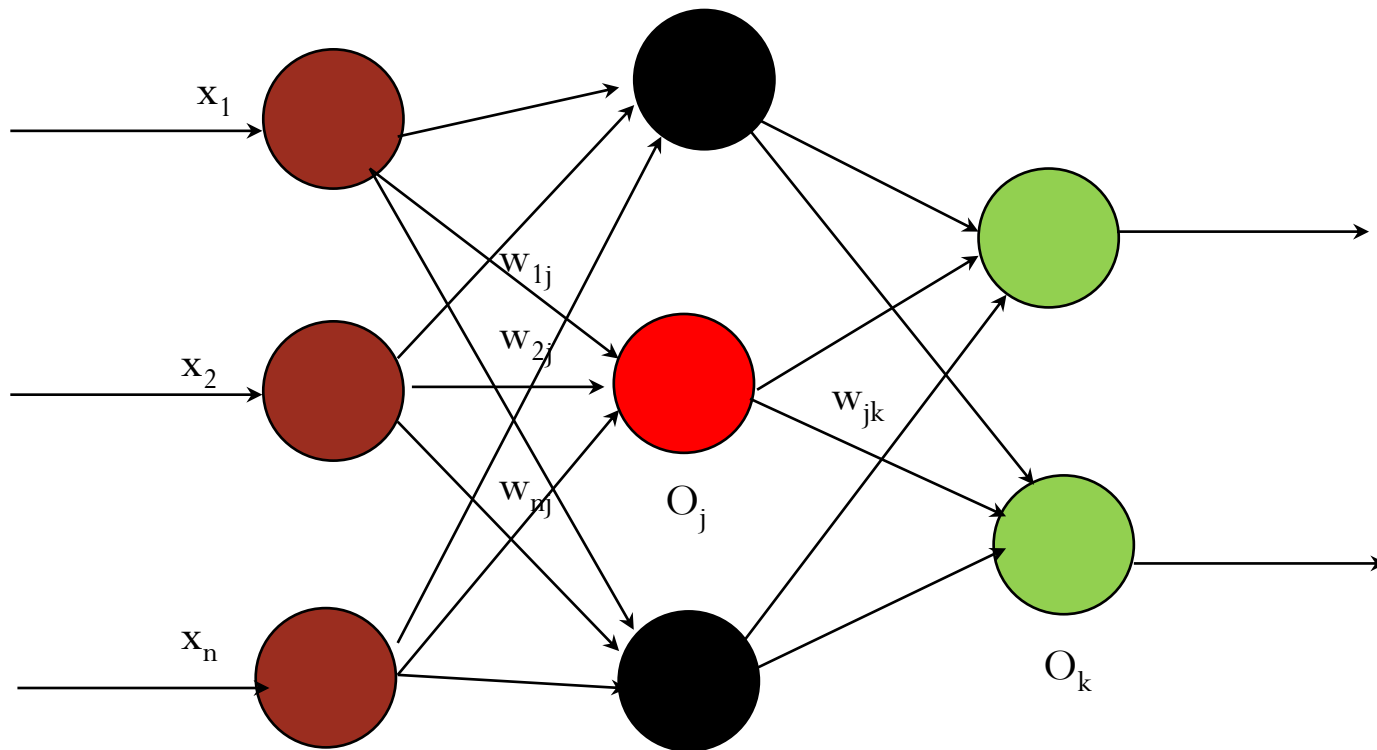


Невронни мрежи

- Влияния
 - ребра на граф
 - на елементите от първото ниво влияят входните параметри
 - всяко следващо ниво е подложено на влияние от предходните нива
- Тегла на влиянията
 - степени на влияние
 - асоциирани към ребрата на графа



Multilayer Feed-Forward Neuron Network



Multilayer Feed-Forward Neuron Network

- Множество слоеве
- Всеки слой се състои от елементи
- С елементите от входния слой се асоциират стойностите на атрибутите в записите от обучаващата извадка $x_1, x_2, x_3, \dots, x_n$
- След преминаване през входния слой, входните стойности се претеглят и така преизчислени се предават на следващия, скрит слой
- Всеки елемент от скрития слой се изчислява като претеглена сума на всички изходи от предходния слой
- Изходите от скрития слой могат да бъдат входове за друг скрит слой, или за изходния слой

Невронни мрежи като класификатори

- Целта е да се прогнозира принадлежността на обект към клас, на базата на оценка на входните параметри
- На етапа на обучение на модела се настройват теглата на влиянията
- На етапа на приложение на модела
 - се търсят нелинейни функционални зависимости
 - изчисляват се в зависимост от топологията на мрежата и стойностите на теглата
 - стойностите на теглата се настройват, за дадат възможно най-точни прогнози за принадлежността на входните атрибути към определен от изходните атрибути клас

Невронни мрежи като класификатори

- За входни данни се използват измерените стойности от обучаващата извадка – **входен слой**
- Входните стойности със своите тегла определят **скрит слой**
 - може да има повече от един скрит слой
- Изходите от един слой са входове на следващия
- Претеглените изходи от последния скрит слой определят **изходния слой**
- Изходите на изходния слой се прилагат за прогнозиране
- Достатъчно голям брой скрити слоеве и обучения могат да възпроизведат приблизително всяка функция – **нелинейна регресия**

Приложение на метода

- Избират се определени стойности на определени входни атрибути и се изследва как други входни атрибути влияят на изходния атрибут, за който се прави прогноза
- Например
 - потенциален клиент на средна възраст (40-50 годишен), с две деца и собствено жилище
 - ще си купи ли велосипед?
 - ако има високи доходи, вероятно ще си купи, но ако живее на повече от 25 км от местоработата си – няма да си купи

Проектиране на модела

- Определяне на топологията
- Определяне на входните стойности
- Определяне на изходите
 - по един изход за всеки клас
- Обучаване на мрежата по избран алгоритъм
 - определяне на тегла
- Топологията и теглата могат да се променят по време на обучението

Определяне на топологията

- Избор на брой на невроните във входния слой
- Избор на брой на скритите слоеве
- Избор на брой на невроните във всеки слой
- Избор на брой на невроните в изходния слой

- Няма еднозначни правила за най-добър брой скрити слоеве
- Прилага се емпирични методи на проба и грешка

Определяне на входните стойности

- Броят на входните елементи се избира така, че всяка стойност от домейните с възможни стойности на всички входни атрибути, да бъде представена
 - Например, ако атрибутът a има три възможни стойности $\{a_1, a_2, a_3\}$, за него са необходими три елемента на входния слой
 - Всички елементи се инициализират с 0 и само част от тях се активират при различните обучаващи тестове
- Всички стойности се представят в интервала $[0.0—1.0]$ - **нормализация**

Алгоритъм за класификация Backpropagation

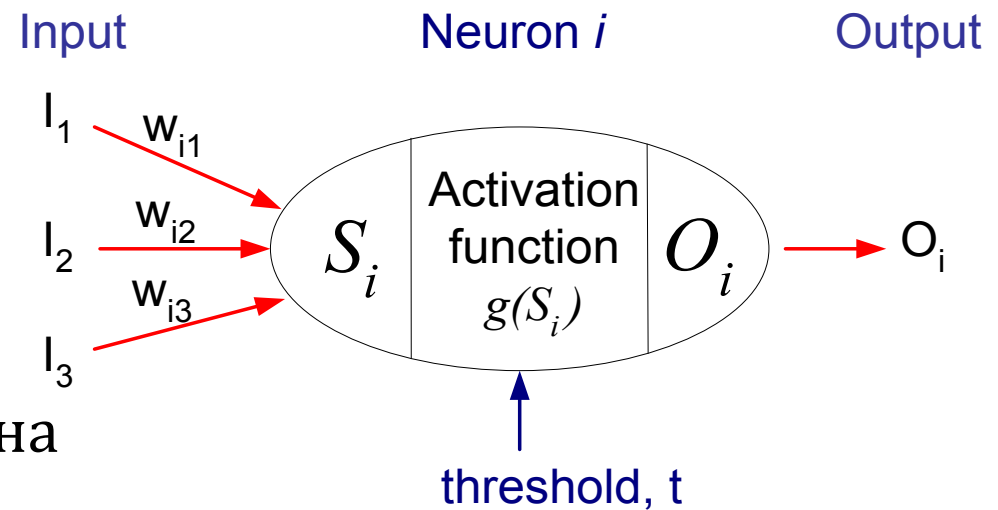
- Интерактивен процес
- Тестване на невронната мрежа за установяване на точността на тестовите прогнози чрез сравняване с известни
- Настройване на теглата на данните така, че да се минимизира средно-квадратичната грешка между прогнозната и действителната стойност
- Изчисляват се входовете и изходите на всеки неврон отпред-назад **feed-forward**
- Проверява се грешката
- Промяната на теглата се извършва отзад-напред, **backwards**, от изхода към скритите слоеве

Стъпки на алгоритъма

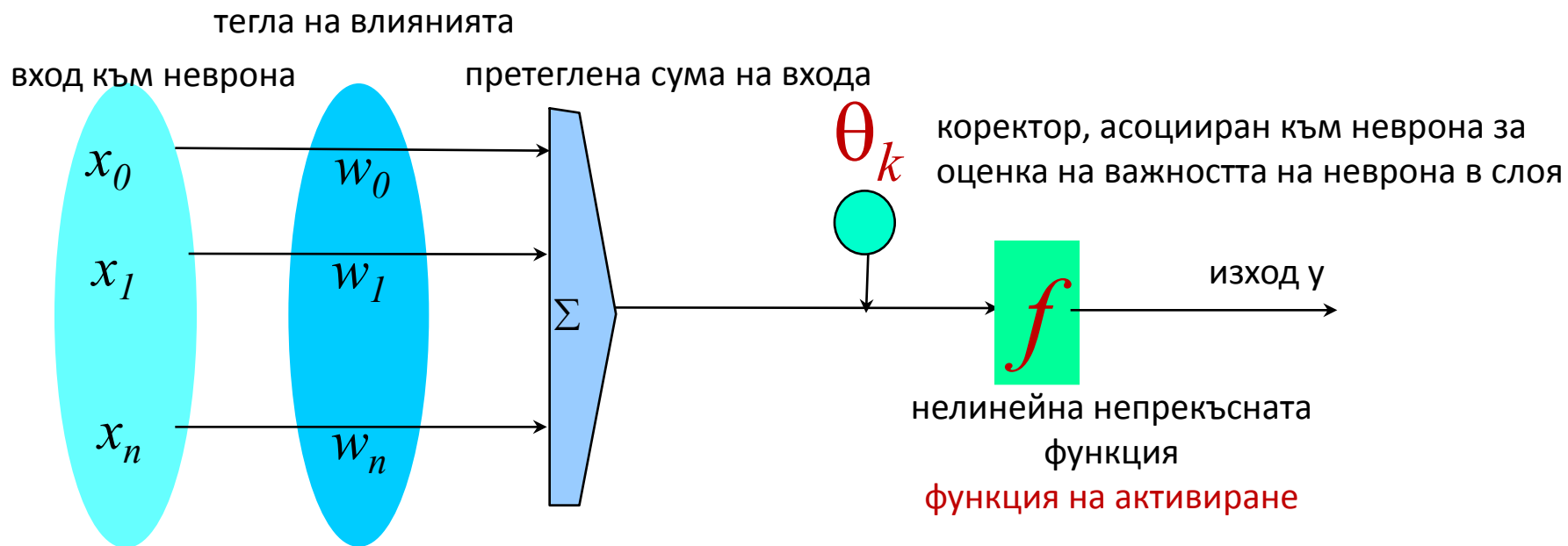
- Инициализиране на теглата и корекционните коефициенти
- Интеративно:
 - Предаване на входните стойности към следващото ниво
 - Връщане и коригиране на грешката, чрез обновяване на теглата и корекционните коефициенти
- Край: при изпълнение на определено условие, напр.
 - за минимален размер на грешката
 - моделът не се подобрява
 - т.н.

Модел на неврон

- ВХОД
 - стойности на входни параметри I
 - тегла на влиянията w
 - начална стойност на неврона, праг θ
- ИЗХОД
 - вход за следващо ниво
- функция на активиране на неврона
 - нелинейна функция, която се прилага върху изчислената стойност на комбинацията от входове $g(S)$



Модел на неврон

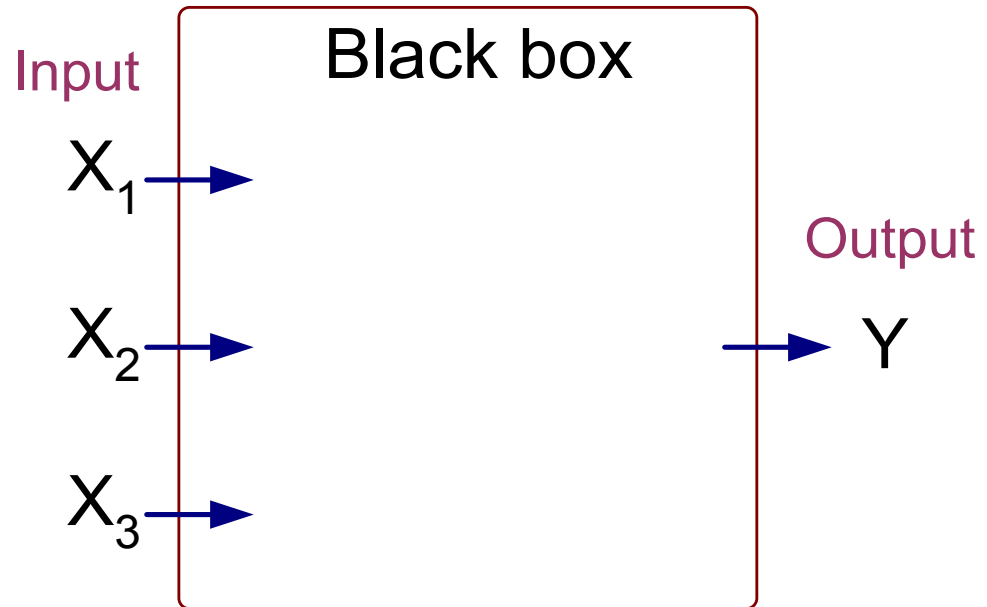


Скритите неврони използват *hyperbolic tangent function* (*tanh*) като функция на активиране, а изходните неврони – *sigmoid*.

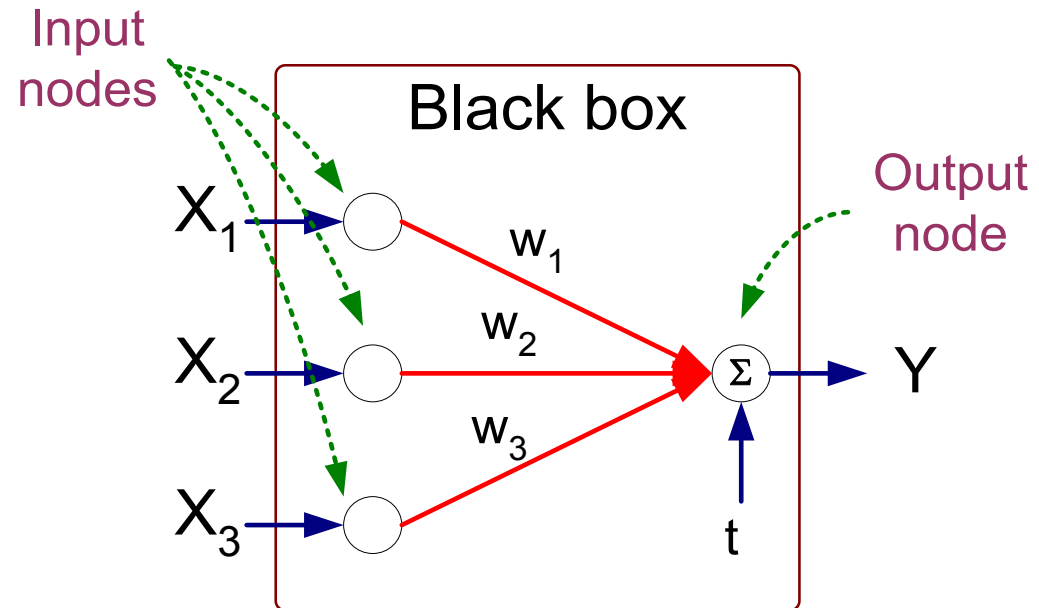
$$y = \text{sign}\left(\sum_{i=0}^n w_i x_i - \theta_k\right)$$

Пример

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



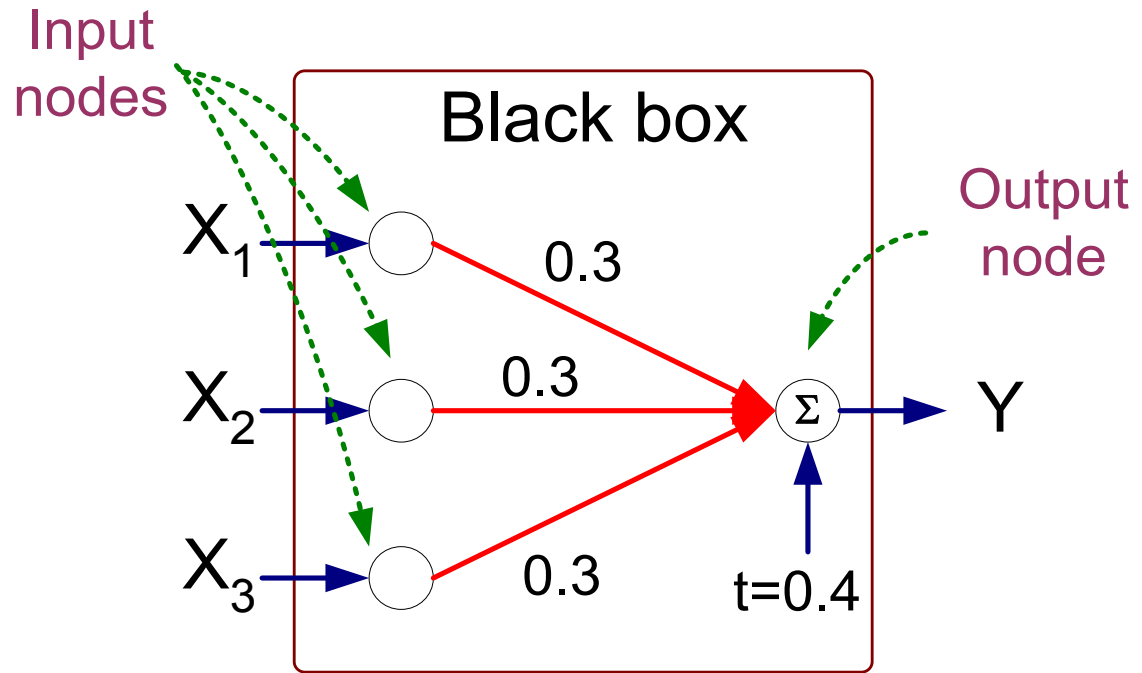
Пример



$$Y = I\left(\sum_i w_i X_i - t\right)$$

Пример

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

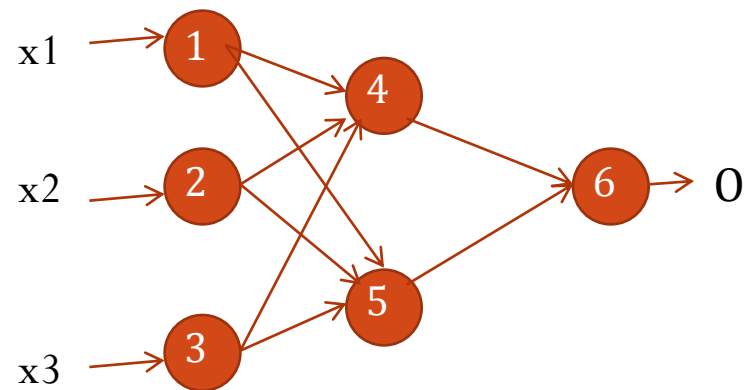


$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

$$\text{където } I(z) = \begin{cases} 1 & \text{ако } z \text{ e true} \\ 0 & \text{ако } z \text{ e false} \end{cases}$$

Пример

пар ам	стойн ост
x1	1
x2	0
x3	1
w14	0.2
w15	-0.3
w24	0.4
w25	0.1
w34	-0.5
w35	0.2
w46	-0.3
w56	-0.2
θ_4	-0.4
θ_5	0.2
θ_6	0.1



$$O_i = \frac{1}{1 + e^{-I_j}}$$

нев рон	вход към неврона	изход от неврона	грешка
4	$(1*0.2+0*0.4 + 1*-0.5) - 0.4 = -0.7$	$1/(1+e^{0.7}) = 0.332$	
5	$(1*-0.3+0*0.1+1*0.2)+0.2 = 0.1$	$1/(1+e^{-0.1}) = 0.525$	
6	$(0.332*-0.3 + 0.525 * -0.2)+0.1 = - 0.105$	$1/(1+e^{0.105}) = 0.474$	

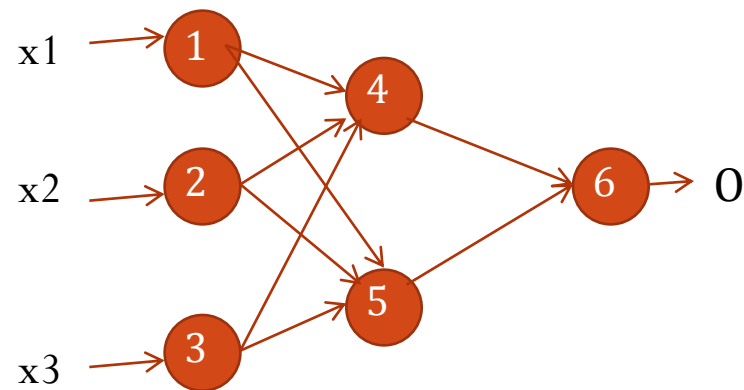
Оценка на грешката

- За изходния слой: разлика с очакваната стойност $Err_j = O_j * (1 - O_j) * (T_j - O_j)$
- За междинните слоеве: натрупана грешка от предходните слоеве $Err_j = O_j * (1 - O_j) * \sum_k Err_k w_{jk}$
- T – цел, очаквана изходна стойност за тези входни стойности

Пример

пар ам	стойн ост
x1	1
x2	0
x3	1
w14	0.2
w15	-0.3
w24	0.4
w25	0.1
w34	-0.5
w35	0.2
w46	-0.3
w56	-0.2
04	-0.4
05	0.2
06	0.1

$$O_i = \frac{1}{1 + e^{-I_j}}$$



$$Err_j = O_j * (1 - O_j) * (T_j - O_j)$$

$$Err_j = O_j * (1 - O_j) * \sum_k Err_k w_{jk}$$

нев рон	вход	изход	грешка
4		$1/(1+e^{0.7}) = 0.332$	$0.332*(1-0.332)*0.1311*-0.3 = -0.0087$
5		$1/(1+e^{-0.1}) = 0.525$	$0.525*(1-0.525)*0.1311*-0.2 = -0.0065$
6		$1/(1+e^{0.105}) = 0.474$	$0.474*(1-0.474)*(1-0.474) = 0.1311$

Корекции

- Корижиране на теглата

$$w_{ij} += (l) \text{Err}_j O_i$$

l – learning rate, константа между 0.0 и 1.0

$$l = 1/t$$

t – пореден номер на итерацията от обучаващата извадка

- Корижиране на праговете

$$\theta_j += (l) \text{Err}_j$$

- След корекциите, обучението продължава

Коригиране Backpropagation

тегло/коэффициент		нова стойност
w46	-0.3	$-0.3 + 0.9 * 0.1311 * 0.332 = -0.261$
w56	-0.2	$-0.2 + 0.9 * 0.1311 * 0.525 = -0.138$
w14	0.2	$0.2 + 0.9 * -0.0087 * 1 = 0.192$
w15	-0.3	$-0.3 + 0.9 * 0.0065 * 1 = -0.306$
w24	0.4	$0.4 + 0.9 * -0.0087 * 1 = 0.4$
w25	0.1	$0.1 + 0.9 * -0.0065 * 0 = 0.1$
w34	-0.5	$-0.5 + 0.9 * -0.0087 * 1 = -0.508$
w35	0.2	$0.2 + 0.9 * -0.0065 * 1 = 0.194$
θ_6	0.1	$0.1 + 0.9 * 0.1311 = 0.218$
θ_5	0.2	$0.2 + 0.9 * 0.0065 = 0.194$
θ_4	-0.4	$-0.4 + 0.9 * 0.0087 = 0.408$

Условия за край на обучението

- всички тегловни коефициенти са по-малки от праговете
- количеството на неправилно класифицирани записи е по-малко от предварително зададен праг
- вече са приложени предварително определен брой обучения

Предимства и недостатъци на метода

- Предимства

- приложим за класификация и прогнозиране
- висока точност на прогнозиране
- устойчивост на шум и грешки
- подходящ за анализ на непрекъснати атрибути
- подходящ за разпознаване на нетренирани извадки
- приложим в много области
 - медицината
 - разпознаване на ръкописен текст
 - разпознаване на говор
 - и др.

- Недостатъци

- продължителен етап на обучение
- труден за разбиране и интерпретиране
- съдържа параметри, определяни емпирично

Критерии за избор на модел

- **Точност**
 - на определяне на класа на изхода
- **Скорост**
 - време за построяване на модела (training time)
 - време за прилагане (classification/prediction time)
- **Устойчивост**: справяне с шума и липсващите стойности
- **Мащабируемост**: приложение в големи бази данни
- **Интерпретиране**
 - разбиране на модела и резултатите
- и др.

Заключение

- **Класификация** - форма на анализ на данни, при която се генерират модели, които описват значими класове.
- Методи за класификация - **дърво на решенията, Бейс, класификация, основана на правила, най-близки съседни, невронни мрежи** и др.
- **Оценяване на методите** – точност, чувствителност, време за обучение, приложимост върху различни категории и количества данни и др.
- Нито един метод не се определя като най-добър, всеки може да бъде най-подходящ в различни условия