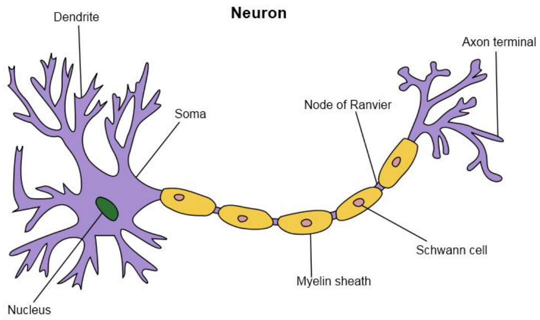
Evolúciós algoritmusok 6. előadás

Neurális hálók

Az agy felépítéséről

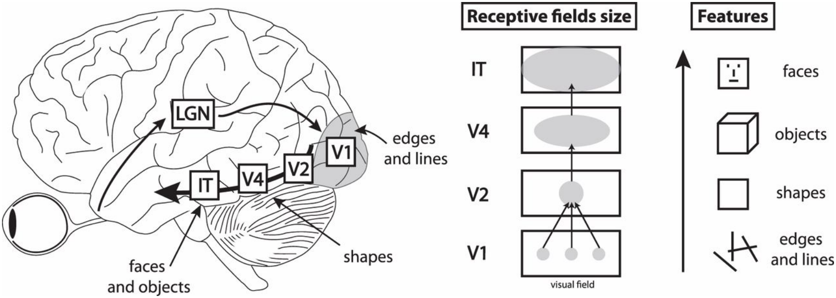
Az idegrendszer legkisebb egysége a neuron. Neuronnak nevezzük az idegsejt és összes nyúlványainak együttesét. A neuronok ingerlékeny sejtek, amelyek ingerfelvételre és idegi ingerületek vezetésére specializálódtak.



Az agy felépítéséről

Az információ dendriteken keresztül érkezik. A sejt egyetlen kimenete az axon. Az agy ilyen idegsejtek hálózatából áll, részben ismert topológiával (adott funkciót ellátó területekre, és ezen belül szintekre osztható).

Például az agy felszínének nagyjából egyharmada a képfeldolgozásra specializálódott neuronokból áll, amelyek különböző szintekbe szerveződnek. Az alacsonyabb szinteken többek között vonalak felismerése és irányának meghatározását végzi az agy, a magasabb szinteken ismeri fel a formákat és tárgyakat.



A kép forrása: Mauro Manassi; Bilge Sayim; Michael H. Herzog (Journal of Vision)

Neurális hálózatok

Az agy nagyon leegyszerűsített mesterséges utánzata a neurális háló(zat), a ma használt legmodernebb kép- és szövegfeldolgozó eszköz. Matematikai szempontból egy nemlineáris függvényt approximál egyszerű függvények kompozíciójával.

A neurális hálózat lokális feldolgozást végző műveleti elemek (neuronok) rendezett topológiájú, kapcsolt rendszeréből áll.

Egy neurális háló felépítésekor meg kell terveznünk az elemi neuronokat és a köztük lévő kapcsolatot (topológiát). Ez a hálózat ekkor még sok szabad paraméterrel rendelkezik. Az optimális paraméterek meghatározását nevezzük a neurális háló tanításának. Adott paraméterekkel a bemenetből a háló kimenetének elkészítése az előhívás.

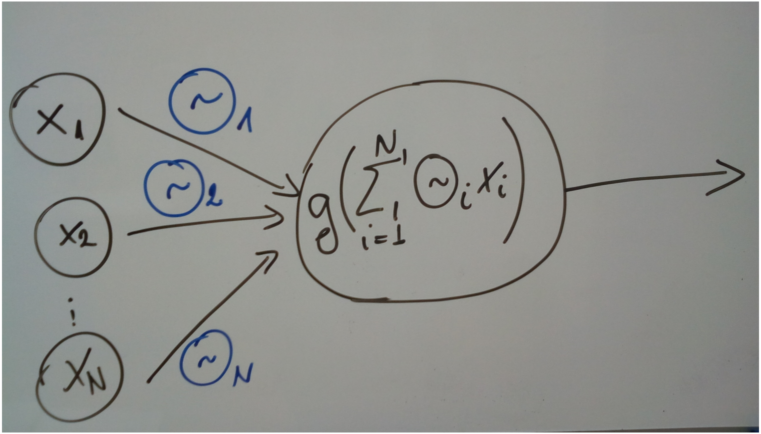
Egy neuron felépítése MLP hálózatoknál

Egy elemi neuronnak több bemenete van és egy kimenete. A kimenet a legegyszerűbb esetben a bemenetek lineáris kombinációján keresztül függ a bemenettől, azaz

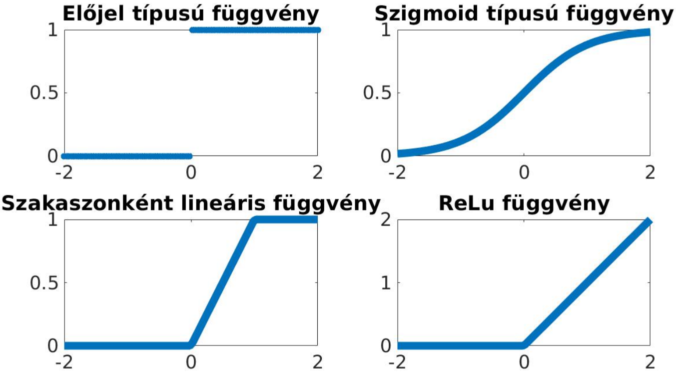
$$g\left(\sum_{i=1}^{N}\theta_i x_i\right)$$

alakban írható. A θ_i mennyiségeket nevezzük súlyoknak, g-et aktivációs függvénynek. Ez utóbbi lehet például egy szakadással rendelkező előjel függvény, vagy egy simább, szigmoid típusú. Létezik még a kettő közötti átmenetnek tekinthető szakaszonként lineáris (de folytonos) függvény.

Egy neuron



Aktivációs függvény típusai



Pár szó a topológiáról

A hálózat topológiája egy irányított gráffal írható le, amely azt mondja meg, hogy mi az egyes neuronok bemenete és kimenete. Az élek típusa alapján egy neuron lehet

- bemeneti (ezek kapják a hálózat bemenetét, és továbbítják a többi neuron felé),
- kimeneti (továbbítják az információt a környezet felé),
- rejtett (a bemenetük és kimenetük is csak másik neuron lehet).

Pár szó a topológiáról

A legegyszerűbb esetben (előrecsatolt többrétegű hálózat) az egyforma típusú neuronok rétegeket alkotnak: van egy bementi, egy kimeneti, illetve tetszőleges számú rejtett réteg, amelyekből minden él előrefelé (a bemeneti rétegtől a kimeneti réteg felé) mutat.

Ez a felépítés rengeteg egymástól független súly meghatározását jelenteni. Bizonyos feladatokban azonban lehet egyszerűsíteni: például képfelismerésnél ha bizonyos csoportok ugyanazt a munkát végzik, csak a kép más-más részeire, akkor vehetjük azonosnak a súlyokat minden ilyen csoportban.

Elméleti eredmények

Tétel (Cybenko, 1989)

Legyen $f(\underline{x})$ egy $K \subset \mathbb{R}^n$ kompakt halmazon értelmezett folytonos függvény, $\varepsilon > 0$ tetszőleges. Ekkor létezik olyan egy rejtett réteggel rendelkező neurális háló, szigmoid aktivációs függvénnyel, melynek $h_\theta(\underline{x})$ kimenetére

$$\max_{x \in K} |f(\underline{x}) - h_\theta(\underline{x})| < \varepsilon$$

A bizonyítás alapötlete az $f(\underline{x})$ függvény egyenletes folytonossága, ami miatt lépcsős függvénnyel jól közelíthető. A lépcsős függvény pedig neurális hálóból jól közelíthető, de a szükséges neuronok száma nyitott kérdés. Később a tételt mások jelentősen általánosították. Bár a bizonyítás konstruktív, ennek ellenére háló építésére a gyakorlatban nem használható.

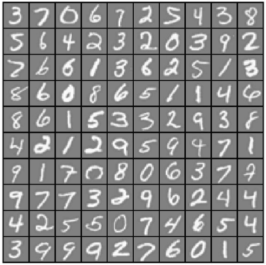
CNN

Specifikus problémára készített neurális hálók az adott problémát megoldó agyterület hierarchiáját utánozzák: képfeldolgozásnál például a konvolúciós hálókat alkalmazzák (sok egyforma kisebb egységet kapcsol össze). Egy alacsony szintű részegység a kép egy részletére (például a bal felső sarokra) old meg egy egyszerűbb problémát: például felismeri, hogy van-e az adott részleten függőleges vonal. Ezen alacsony szintű részegységek kimenetét használja a neurális háló következő szintje bemenetként.

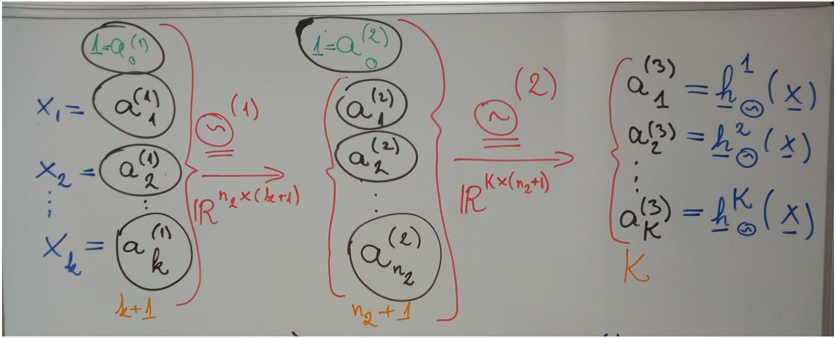
Egy kis érdekesség arról, hogy hány neuron és réteg van egy mai neurális hálóban: CNN

Mintapélda: osztályozás

Legyen a bemenetnek k dimenziója. Osztályozási feladatot oldunk meg, azaz szeretnénk meghatározni hogy a bemenet a K darab osztály melyikébe tartozik. Konkrét példa: 20×20 pixeles fekete-fehér képekről szeretnénk megmondani, hogy milyen számjegy van rajta. Itt összesen 100 darab példabemenet látható, mindegyikhez ismerjük az ábrázolt számjegyet is.

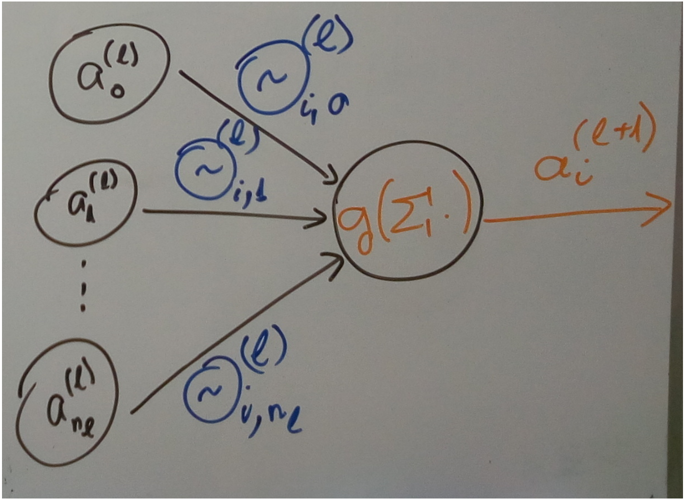


Példa: egy rejtett rétegű NN



A zölddel jelölt neuronok az úgynevezett 'bias' egységek. A mintapéldában $k = 400$ és $K = 10$. A kimenet mind a 10 neuronon egy 0 és 1 közötti szám, amelyik a legnagyobb, az lesz a tippünk az osztályra.

Példa: egy rejtett rétegű NN



Egy neuron közelről.

A súlyok kiszámítása

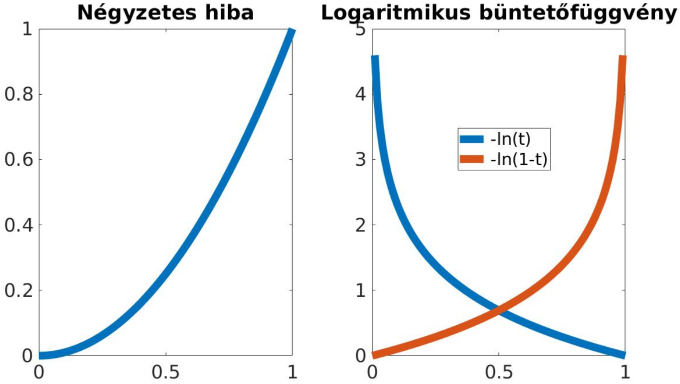
Ha már adott a háló topológiája, akkor az ismeretlen súlyok (az összes θ mátrix) meghatározása a következő feladat. Előre ismert valahány (tipikusan rengeteg) tanítópontban az approximálandó függvény értéke.

Így meghatározható, hogy az adott súlyokkal a háló milyen összhibát vét, célunk ennek minimalizálása. Ez egy (nagyméretű háló esetén nagyon sok dimenziós) függvény-optimalizálási feladat.

Jelen példánkban van összesen $401 \cdot 25 + 26 \cdot 10 = 10285$ θ paraméterünk, ez a feladat dimenziója.

Az NN által vétett összhiba mérése

Ahhoz, hogy meg tudjuk oldani a függvény optimalizálási feladatot egy mérőszámot kell rendelnünk a neurális háló kimenete és a kívánt kimenet eltéréséhez. Erre szigmoid aktivációs függvény esetén a logaritmikus büntetőfüggvény a célszerű. Más esetekben szóba jön a legkisebb négyzetes eltérés.



Az NN által vétett összhiba mérése

Tegyük fel, hogy m darab tanulópont áll rendelkezésre, ezekre ismert, hogy melyik osztályba tartoznak. Az y változó értéke 1 erre az osztályra, a többire 0. Ekkor a költségfüggvény:

$$J(\theta) = \frac{-1}{m} \sum_{\text{példák}} \sum_{k=1}^{\text{osztályok}} y \log(\text{k.kimenet}) + (1-y) \log(1-\text{k.kimenet})$$

Ha tökéletesen illeszkedne az adatainkhoz a háló (ami nem cél!), akkor 0 lenne az értéke, egyébként pozitív. Célunk azon θ súlyok megtalálása, amelyekre ezen költség minimális.

Az optimális súlyok meghatározása

Egy lehetőség (és ma ez a legmodernebbnek tekintett módszer), hogy az optimális súlyok kiszámítását gradiens-módszerrel végezzük. Ehhez szükséges a fenti költségfüggvény összes $\theta_{i,j}^{(l)}$ változó szerinti deriváltját meghatároznunk.

A deriváltakra a láncszabály felhasználásával egy rekurzív képletet tudunk felírni, a kimeneti rétegtől indulva és a rétegeken visszafelé haladva (ezért hívják 'backpropagation of error'-nak).

A súlyok kiszámítása - Backpropagation

Amennyiben a $g(z) = \frac{1}{1+\exp(-z)}$ szigmoid aktivációs függvényt használjuk a fenti költségfüggvénnyel, akkor a következő rekurziót írhatjuk fel a deriváltakra egyetlen tanulópont alapján:

$$\underline{\delta}^{(L)} = \underline{h}_\theta(\underline{x}) - \underline{y}$$

$$\underline{\delta}^{l} = \underline{\underline{\hat{\theta}}}^{(l)} \underline{\delta}^{(l+1)} .* g'(\underline{\underline{\theta}}^{(l-1)} \underline{a}^{(l-1)}),$$

ahol $\underline{\underline{\hat{\theta}}}^{(l)}$ mátrixot úgy kapjuk, hogy töröljük a 'bias' unit-hoz tartozó súlyokat. Majd ezen $\underline{\delta}$ mennyiségekből:

$$\frac{\partial}{\partial \theta_{i,j}^{(l)}} = \frac{1}{m} \sum_{példák} \delta_i^{(l+1)} a_j^{(l)}$$

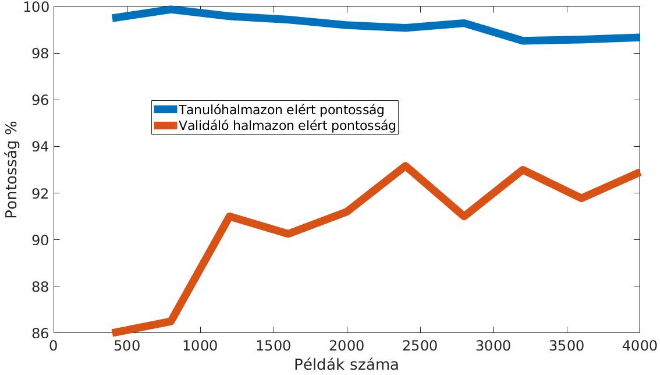
Validáció

A tanulási folyamat végén egy validációs halmazon teszteljük, hogy az 'optimális' súlyok mennyire jól oldják meg az adott feladatot. Ehhez még kezdetben két részre osztjuk a tanulópontjainkat. Az adatok egy részét (kb 75-80%-át) használjuk a súlyok beállítására. A maradékot az elkészült háló tesztelésére, az ezen a halmazon számított hiba a háló valódi hibája.

Egy nagyméretű háló fokozottan ki van téve az 'overfitting' veszélyének. Ekkor bár a költség 0 lesz, nem a feladatot tanultuk meg, hanem a teszthalmazt. Ennek elkerüléséhez még egy tagot hozzáveszünk a költségfüggvényhez, az összes súly négyzetösszegét valamilyen paraméterrel súlyozva. Ekkor persze a gradienst is újra kell számolnunk.

Learning curve

Érdemes megfigyelnünk a példák számának változtatásával hogyan változik a validációs hiba. Itt egy 25 rejtett neuronnal rendelkező háló hibájának alakulása a példák számának függvényében.



Learning curve

Ha a validációs hiba és a tanulóhalmazon elkövetett hiba még távol vannak egymástól, akkor valószínűleg sok a változónk a példák számához képest ('High variance'). Jelen példánkban van összesen 10285 θ paraméterünk, és 4000 tanulópontunk. Nagyobb adathalmaztól a paraméterek jobb illesztését várhatjuk, de ezt megszerezni költséges, a hálót tanítani pedig több idő.

Ha a validációs hiba és a tanulóhalmazon elkövetett hiba közel van egymáshoz, de egyformán magas, akkor a modell rossz, további tanuló adatoktól nem várhatunk javulást. ('High bias')

A háló teljesítménye

Tegyük fel, hogy csak annyi feladatunk, hogy eldöntsük a 9-es számjegy van-e a képen vagy sem. Ekkor elégedettek lennénk egy 90%-os teljesítménnyel? Pedig ha az adatok közül egyforma arányban fordulnak elő a számjegyek és a háló az adatoktól függetlenül nem-et válaszol, akkor eléri ezt a teljesítményt.

Ezért szokás bevezetni az F_1 score-t a teljesítménymérésére. Ehhez tekintsük egy 2 osztályba sorolási feladatot, ahol csak pozitív/negatív választ kell adni. A modell szenzitivitásának ('recall') nevezzük a valódi pozitív és összes pozitív eset arányát. A modell pontossága ('precision') a valódi pozitív és a pozitívnak jelzett esetek aránya. A két arány harmonikus közepe az F_1 score.

Az F_1 score

Teszt \ Valóság	Pozitív	Negatív
Pozitív	Valódi pozitív (VP)	Hamis pozitív (HP)
Negatív	Hamis negatív (HN)	Valódi negatív (VN)

Azaz a szenzitivitás:

$$R = \frac{VP}{VP + HN}$$

a pontosság:

$$P = \frac{VP}{VP + HP}$$

. Innen:

$$F_1 = \frac{2PR}{P + R}.$$