**Az MLP Modell Gyengeségei Képfeldolgozásban**

A Multilayer Perceptron (MLP) modell képes összetett mintázatok felismerésére, de képfeldolgozásban számos gyengesége van.

**Nagy Bemeneti Méret:**

* A képek általában nagy méretűek (például egy 28x28 pixeles kép 784 bemeneti neuront igényel).
* A nagy bemeneti méret miatt az MLP-k rengeteg paraméterrel rendelkeznek, ami nagy számítási és memóriaigényt eredményez.

**Lokális Mintázatok Figyelmen Kívül Hagyása:**

* Az MLP minden bemeneti pixelt egyformán kezel, figyelmen kívül hagyva a lokális mintázatokat.
* A képekben a szomszédos pixelek közötti korrelációk fontosak, de az MLP-k nem képesek ezeket hatékonyan kihasználni.

**Transzlációs Invariancia Hiánya:**

* Az MLP nem képes felismerni, ha egy objektum különböző helyeken jelenik meg a képen.
* Minden pozícióra különböző súlyokat kell tanulnia, ami növeli a modell komplexitását.

**Diszkrét Konvolúció és Konvolúciós Réteg**

**Diszkrét Konvolúció**

A diszkrét konvolúció egy matematikai művelet, amely két diszkrét függvény (például egy kép és egy szűrő) kombinációját eredményezi. A konvolúciós művelet a bemeneti adatok és a szűrő közötti korrelációkat számítja ki.

Matematikailag egy képre és egy szűrőre alkalmazott diszkrét konvolúció így néz ki: ...

**Konvolúciós Réteg**

A konvolúciós réteg egy neuronális hálózatban olyan réteg, amely konvolúciós műveleteket végez. Minden szűrő (kernel) végigcsúszik a bemeneti képen, és egy kimeneti térképet (feature map) hoz létre.

**Szűrők (Kernelek):**

* A szűrők kis méretűek (például 3x3, 5x5), és a bemeneti kép különböző régióira alkalmazva új feature map-eket hoznak létre.
* A szűrők paramétereit a hálózat tanulja meg.

**Aktivációs Függvények:**

* A konvolúciós műveletek után az aktivációs függvény (például ReLU) alkalmazása segít a nemlinearitás bevezetésében.

**Pooling Réteg**

A pooling réteg csökkenti a térbeli dimenziókat, miközben megőrzi a fontos információkat. Ez segít a modellnek a transzlációs invariancia elérésében.

**Max Pooling:**

* A leggyakrabban használt pooling módszer, amely a pooling ablakban található legnagyobb értéket választja ki.
* **Példa:** Egy 2x2-es pooling ablak esetén a négy érték közül a legnagyobbat választjuk ki.

**Average Pooling:**

* Az pooling ablakban található értékek átlagát számítja ki.

**Konvolúciós Háló Működése**

Olyan neurális hálózat, amely különösen hatékony a képfeldolgozási feladatok megoldására. A lényege abban rejlik, hogy a hálózat tanulja meg az adott képekben található fontos mintázatokat és jellemzőket.

A konvolúciós hálózatok kiválóak a képfeldolgozási feladatokban, mert képesek lokalizálni és kinyerni a fontos jellemzőket a képek különböző részeiből, és a tanulás során kevésbé függenek a képek általános pozíciójától vagy kisebb transzlációs változásaitól. A konvolúciós rétegek paramétereinek (pl. szűrő mérete, szám, stride, padding stb.) megfelelő beállítása kulcsfontosságú a hálózat hatékonysága és teljesítménye szempontjából.

**Konvolúciós Háló Felépítése (LeNet-5 Architektúra)**

A LeNet-5 az egyik első sikeres konvolúciós neurális háló (CNN) architektúra, amelyet kifejezetten a képfeldolgozásra terveztek.

**Bemeneti Réteg:**

* A bemeneti kép mérete 32x32 pixeles.

**C1 Konvolúciós Réteg:**

* Hat 5x5 méretű szűrőt használ, ami 6 különböző 28x28 méretű feature map-et eredményez.
* Az aktivációs függvény a sigmoid.

**S2 Pooling Réteg:**

* A C1 réteg 2x2-es ablakokat használó average pooling rétege.
* Ez 6 különböző 14x14 méretű feature map-et eredményez.

**C3 Konvolúciós Réteg:**

* 16 darab 5x5 méretű szűrőt használ, ami 16 különböző 10x10 méretű feature map-et eredményez.
* Az aktivációs függvény a sigmoid.

**S4 Pooling Réteg:**

* A C3 réteg 2x2-es ablakokat használó average pooling rétege.
* Ez 16 különböző 5x5 méretű feature map-et eredményez.

**C5 Konvolúciós Réteg:**

* 120 darab 5x5 méretű szűrőt használ, ami 120 különböző 1x1 méretű feature map-et eredményez (teljesen kapcsolódó réteg).

**F6 Teljesen Kapcsolódó Réteg:**

* 84 neuront tartalmaz, és a bemenete a C5 réteg kimenete.

**Output Réteg:**

* 10 neuront tartalmaz (egy-egy az osztályoknak), és a kimenet a softmax függvény alkalmazása után kerül ki.

**Hierarchikus Mintázatfelismerés**

A hierarchikus mintázatfelismerés egy olyan folyamat, amely során egy rendszer a bemeneti adatokat különböző absztrakt szinteken elemzi és dolgozza fel. Ebben az esetben a rendszer egy konvolúciós neurális hálózat (CNN) lehet, amely képes felismerni az alacsony szintű, egyszerűbb mintázatoktól kezdve a magasabb szintű, komplexebb mintázatokig terjedő jellemzőket.

Például, egy képfeldolgozási feladat során az alacsony szintű jellemzők lehetnek élek, sarkok vagy színek, míg a magasabb szintű jellemzők lehetnek objektumok, arcok vagy emberek.

A hierarchikus felismerés előnye, hogy a rendszer hatékonyan tudja reprezentálni és felismerni az egyre összetettebb jellemzőket, miközben azokat a bemeneti adatokat dolgozza fel, amelyeket egyre magasabb absztrakciós szinten kezel.

A konvolúciós hálózatok hierarchikus mintázatfelismerése több szinten történik:

**Alacsony Szintű Jellemzők:**

* Az alsóbb rétegek egyszerű mintázatokat tanulnak, mint például élek és sarkok.

**Középszintű Jellemzők:**

* A középső rétegek bonyolultabb mintázatokat tanulnak, mint például textúrák és formák.

**Magas Szintű Jellemzők:**

* A felsőbb rétegek komplex mintázatokat és objektumokat ismernek fel.

**Transzláció Invariancia**

A transzláció invariancia azt jelenti, hogy a rendszer kimenete (például osztályozás vagy regresszió) nem változik, ha a bemeneti adatok eltolódnak (transzláció). Más szavakkal, a rendszernek nem kell érzékelnie vagy figyelembe vennie az objektumok pontos pozícióját vagy elhelyezkedését a bemeneten ahhoz, hogy hatékonyan működjön.

A konvolúciós neurális hálózatok (CNN-ek) hajlamosak transzlációs invarianciát kialakítani, mivel a konvolúciós rétegek révén a hálózat képes az objektumok különböző pozícióiban található jellemzőket hatékonyan felismerni. Ezt a tulajdonságot pooling rétegek is erősítik, amelyek csökkentik a térbeli dimenziókat, miközben megőrzik a fontos információkat.

**Padding**

A padding azt jelenti, hogy nullákkal (vagy más értékekkel) kiegészítjük a bemeneti adatokat a konvolúciós művelet előtt. Ennek a célja lehet például a kimeneti feature map méretének megtartása vagy a szélek információinak megőrzése. Két fő célja van:

**Méret Megőrzése:**

* Biztosítja, hogy a kimeneti feature map mérete megegyezzen a bemeneti kép méretével.

**Szélek Információjának Megőrzése:**

* Megőrzi a bemeneti kép szélein található információkat, amelyeket a konvolúciós művelet egyébként elveszítene.

**Stride (Lépésköz)**

A stride az a lépésköz, amelyet a szűrő egy konvolúciós művelet során tesz. Két fő típus van:

**Stride = 1:**

* A szűrő egy pixellel lép tovább a bemeneti képen, ami nagyobb kimeneti feature map-et eredményez.

**Stride > 1:**

* A szűrő több pixellel lép tovább, ami kisebb kimeneti feature map-et eredményez.