**Alultanulás és túltanulás jellemzői, felismerésük**

**Alultanulás**

**Jellemzői:**

A modell nem tanulja meg megfelelően az adatok mögötti mintákat, ezért alacsony teljesítményt nyújt mind a tanuló, mind a teszt adathalmazon. Egyszerű modellek, kevés paraméterrel (pl. lineáris regresszió egy bonyolult, nem lineáris adathalmazra). Magas bias (elfogultság): a modell leegyszerűsíti a problémát, nem képes komplexitást kezelni.

**Felismerése:**

A tanulási és tesztelési hiba magas. A hiba nem csökken a modell tanítása során.

**Túltanulás**

**Jellemzői:**

A modell túl jól megtanulja a tanuló adatok mintáit, beleértve a zajokat és a véletlen ingadozásokat is. Komplex modellek, sok paraméterrel (pl. mély neurális hálózat kevés adaton). Alacsony bias, magas variancia: a modell jól teljesít a tanuló adathalmazon, de gyengén a teszt adathalmazon.

**Felismerése:**

Alacsony hiba a tanuló adathalmazon, de magas hiba a teszt adathalmazon. A hiba gyorsan csökken a tanuló adatokon, de növekszik vagy stagnál a teszt adatokon.

**Hiperparaméterek**

* Ezek a paraméterek meghatározzák a tanulási folyamat körülményeit és a modell architektúráját, és közvetlenül befolyásolják a tanulási folyamat eredményességét és hatékonyságát.
* **Példák:** *tanulási ráta (α)*, regulárizációs paraméter (λ), rejtett rétegek száma és mérete egy neurális hálózatban, *epoch-szám*, *batch-méret*.
  + **Tanuális ráta (α):**  
    A tanulási ráta meghatározza, hogy milyen mértékben változtatjuk meg a modell paramétereit a gradiens módszer során. Nagy tanulási ráta esetén a paraméterek nagyobb lépésekben frissülnek, míg kis tanulási ráta esetén kisebb lépésekben. A megfelelő tanulási ráta kiválasztása kritikus fontosságú a tanulási stabilitás és a konvergencia szempontjából.
  + **Epoch-száma:**  
    Az epoch-szám meghatározza, hogy a tanulási algoritmus hány teljes iterációt hajt végre a tanító adatokon. Egy epoch egy teljes tanító adat halmazt jelent. A megfelelő epoch-szám kiválasztása lehetővé teszi a modell számára, hogy tanuljon és illeszkedjen a tanító adatokhoz, anélkül hogy túl sokat tanulna.
  + **Batch-méret:**  
    A batch-méret meghatározza, hogy hány adatpontot használunk minden egyes frissítéshez a gradiens módszer során. A kisebb batch-méretek gyorsabb frissítéseket eredményeznek, míg a nagyobb batch-méretűek stabilabb frissítéseket biztosítanak. A megfelelő batch-méret kiválasztása lehetővé teszi a hatékony tanulást és a számítások optimalizálását.

**A modell betanításának és kiértékelésének lépései validációs halmazzal**

**1. Adatok felosztása:**

* **Tanuló halmaz (Training Set):** Ez az adathalmaz a modell tanításához használt adatokat tartalmazza. A modell ezen az adathalmazon tanulja meg a mintázatokat és a kapcsolatokat az input és output között.
* **Validációs halmaz (Validation Set):** A validációs halmazt arra használjuk, hogy értékeljük a modell teljesítményét és finomhangoljuk a hiperparamétereket. A modell tanulása során gyakran szükség van a modell paramétereinek beállítására (pl. tanulási ráta, regularizációs paraméterek stb.), és a validációs halmaz segítségével értékeljük ezeknek a beállításoknak a hatékonyságát.
* **Teszt halmaz (Test Set):** A teszt halmazt a modell végleges teljesítményének objektív értékelésére használjuk. Ezeket az adatokat a tanulási folyamat során nem használjuk fel semmilyen módon, így azok függetlenek a tanulástól és objektív képet adnak a modell teljesítményéről a valóságos környezetben.

**2. Tanítás (Training):**

A modellt tanítjuk a tanuló halmazon, azaz a bemeneti adatok és a hozzájuk tartozó címkék alapján. A tanulás során a modell próbálja megtanulni a bemenetek és a címkék közötti összefüggéseket.

**3. Validáció (Validation):**

A modell teljesítményét értékeljük a validációs halmazon, amelyet a tanulás során nem használtunk fel. A validációs halmazon történő kiértékelés segítségével finomhangoljuk a modell hiperparamétereit és beállításait annak érdekében, hogy optimalizáljuk a teljesítményt és elkerüljük a túltanulást.

**4. Tesztelés (Testing):**

A végleges modellt kiértékeljük a teszt halmazon, amelyet a tanulás során nem használtunk fel. A tesztelés célja, hogy objektív módon mérjük a modell teljesítményét a valóságos környezetben. A tesztelés eredménye alapján megállapíthatjuk, hogy mennyire jól teljesít a modell a valóságos adatokon.

**Túltanulás kezelése**

**1. Mintaelemek számának növelése**

**Cél:** Az adatok mennyiségének növelése, hogy a modell jobban általánosíthatóvá váljon.

**Működés:**

* **Több adatgyűjtés:** Minél több adat áll rendelkezésre a modell tanításához, annál jobban képes a modell felismerni az általános mintákat, és kevésbé fogja megtanulni a zajokat és az adatok véletlenszerű változását.
* **Adatforrások bővítése:** Új adatokat gyűjthetünk különböző forrásokból, például további mérések, kísérletek, vagy külső adatbázisok felhasználásával.

**Előnyök:**

* Csökkenti a modell varianciáját, ezáltal jobban általánosíthat.
* Nagyobb adathalmazoknál a modell robusztusabbá válik.

**Hátrányok:**

* Adatgyűjtés idő- és erőforrás-igényes lehet.
* Az adatok minősége is fontos, nem csak a mennyiségük.

**2. Paraméterek számának csökkentése**

**Cél:** Egyszerűbb modellek használata kevesebb paraméterrel, hogy elkerüljük a túlkomplexitást és a túltanulást.

**Működés:**

* **Modellek egyszerűsítése:** Például kisebb mélységű döntési fák, kevesebb rejtett réteg vagy neuron egy neurális hálózatban.
* **Felesleges paraméterek eltávolítása:** A nem informatív vagy redundáns változók eltávolítása az adathalmazból.

**Előnyök:**

* Csökkenti a modell varianciáját és egyszerűbbé teszi a modellt.
* Javítja a modell érthetőségét.

**Hátrányok:**

* Ha túlzottan leegyszerűsítjük a modellt, alultanuláshoz vezethet.

**3. Early Stopping**

**Cél:** A modell tanításának leállítása, amikor a validációs hiba elkezd növekedni, hogy elkerüljük a túltanulást.

**Működés:**

* **Tanulási folyamat monitorozása:** A tanítás során folyamatosan figyeljük a validációs halmazon mért hibát.
* **Stop feltétel:** A modell tanítását leállítjuk, amikor a validációs hiba növekedni kezd, még ha a tanulási hiba tovább csökken is.

**Előnyök:**

* Megakadályozza a modell túlillesztését a tanuló adatokra.
* Hatékonyan javítja a modell általánosítási képességét.

**Hátrányok:**

* Megfelelő stop kritérium kiválasztása néha nehéz lehet.
* Előfordulhat, hogy a modell nem éri el a potenciálisan legjobb teljesítményt.

**4. Adatok augmentációja**

**Cél:** Az adatok mesterséges növelése különböző transzformációkkal, hogy a modell ne tanulja meg a bemeneti adatok véletlenszerű zajait.

**Működés:**

* **Képek esetén:** Forgatás, eltolás, méretezés, tükrözés, színek módosítása, zaj hozzáadása stb.
* **Szöveges adatok esetén:** Szinonimák cseréje, mondatok átrendezése stb.

**Előnyök:**

* Növeli az adathalmaz változatosságát anélkül, hogy új adatokat kellene gyűjteni.
* Segíti a modell robusztusabbá válását a különböző adatvariációkkal szemben.

**Hátrányok:**

* Az augmentációs technikák nem minden adatfajtára alkalmazhatóak egyformán jól.
* A rosszul megválasztott augmentációk ronthatják a modell teljesítményét.

**5. Zaj hozzáadása**

**Cél:** Kis mértékű zaj hozzáadása a bemeneti adatokhoz, hogy a modell ne tanulja meg a zajokat és ne váljon túlérzékennyé az adatok véletlenszerű változására.

**Működés:**

* **Zaj típusok:** Gaussi zaj, sós és bors zaj, stb.
* **Zaj mértéke:** A zaj mértéke szabályozható, hogy ne torzítsa túl az eredeti adatokat.

**Előnyök:**

* Segíti a modell robusztusabbá válását és javítja az általánosítási képességet.
* Növeli a modell toleranciáját a valós világban előforduló adatvariációkkal szemben.

**Hátrányok:**

* Túl sok zaj hozzáadása ronthatja a modell teljesítményét.
* Nehéz megtalálni az optimális zajszintet.

**6. Dropout**

**Cél:** Neurális hálózatok esetén bizonyos neurális egységek véletlenszerűen kikapcsolása a tanítás során, hogy a hálózat ne váljon túlzottan függővé bizonyos neuronoktól.

**Működés:**

* **Kikapcsolási arány:** Általában a rétegek neuronjainak 20-50%-át véletlenszerűen kikapcsolják a tanítás minden iterációjában.
* **Kikapcsolási mechanizmus:** A kikapcsolt neuronok nem vesznek részt a számításokban az adott iteráció során, de a következő iterációban újra bekapcsolódhatnak.

**Előnyök:**

* Csökkenti a neurális hálózat túlillesztését és növeli a robusztusságát.
* Javítja a modell általánosítási képességét, mivel a hálózat nem válik túlzottan függővé egy-egy részterülettől.

**Hátrányok:**

* Lassíthatja a tanulási folyamatot.
* Néha bonyolult lehet a megfelelő dropout arány kiválasztása.