**Mi a gépi tanulás?**

A gépi tanulás egy olyan számítástechnikai tudományterület, amely algoritmusokat és statisztikai modelleket alkalmaz arra, hogy a számítógépek automatikusan tanuljanak és javuljanak tapasztalatokból anélkül, hogy kifejezetten beprogramoznák őket. A cél az, hogy a gép felismerje a mintákat az adatokban, és ezen minták alapján előrejelzéseket vagy döntéseket hozzon.

**Milyen feladatok megoldására ideális?**

A gépi tanulás számos különböző feladat megoldására alkalmas, például:

* **Kép- és hangfelismerés:** Az arcok, tárgyak, beszéd azonosítása.
* **Természetes nyelv feldolgozás (NLP):** Szövegértés, fordítás, érzelemfelismerés.
* **Ajánlórendszerek:** Filmek, termékek ajánlása a felhasználók korábbi viselkedése alapján.
* **Orvosi diagnosztika:** Betegségek felismerése és diagnosztizálása orvosi adatok alapján.
* **Pénzügyi előrejelzések:** Részvényárfolyamok, piaci trendek előrejelzése.
* **Játékok és szimulációk:** Stratégiák kidolgozása és optimalizálása.

**Felügyelt és felügyeletlen tanulás**

**Felügyelt tanulás (Supervised Learning)**

A felügyelt tanulás során az algoritmus egy előre definiált, címkézett adathalmaz alapján tanul. A bemeneti adatokhoz tartozik egy célérték (label), amit az algoritmusnak meg kell tanulnia előrejelezni.

**Feladatai:**

* **Klasszifikáció (Classification):** Az adatok kategóriákba sorolása (pl. email spam vagy nem spam).
* **Regresszió (Regression):** Folytonos célértékek előrejelzése (pl. házárak, hőmérséklet).

**Felügyeletlen tanulás (Unsupervised Learning)**

A felügyeletlen tanulás során az algoritmusnak nincs címkézett adathalmaz, hanem az adatokban rejlő struktúrákat és mintákat kell felfedeznie.

**Feladatai:**

* **Klaszterezés (Clustering):** Az adatok csoportosítása (pl. ügyfelek szegmensekbe sorolása).
* **Asszociációs szabályok (Association Rules):** Az adatok közötti kapcsolatok felfedezése (pl. vásárlási szokások elemzése).

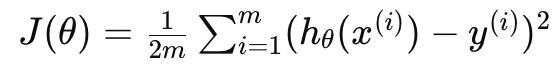
**Formális definíciója és folyamatának lépései**

A gépi tanulási folyamat általában a következő lépésekből áll:

* **Adatgyűjtés:** Az adatok összegyűjtése a problématerületről.
* **Adat-előkészítés:** Az adatok tisztítása, átalakítása és előkészítése.
* **Adatfelosztás:** Az adathalmaz felosztása tanító (training) és teszt (test) adatokra.
* **Modellek kiválasztása:** Megfelelő gépi tanulási algoritmusok kiválasztása.
* **Tanítás (Training):** Az algoritmus tanítása a tanító adatokkal.
* **Értékelés (Evaluation):** A modell teljesítményének értékelése a teszt adatok alapján.
* **Optimalizálás (Optimization):** A modell finomhangolása és optimalizálása.
* **Telepítés (Deployment):** A végleges modell alkalmazása valós adatokon.

**Hipotézisfüggvény és költségfüggvény fogalma, példák**

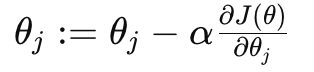
**Hipotézisfüggvény (Hypothesis Function)**  
A hipotézisfüggvény egy matematikai függvény, amely a bemeneti adatok alapján előrejelzéseket tesz. Például egy lineáris regresszió esetében a hipotézisfüggvény a következőképpen nézhet ki: **hθ(x)=θ0+θ1x**, ahol **θ0** és **θ1** a modell paraméterei, **x** pedig a bemeneti változó.

**Költségfüggvény (Cost Function)**  
A költségfüggvény egy olyan függvény, amely a modell predikcióinak és a valóságos értékeknek a különbségét méri. Ezáltal segít meghatározni, hogy a modell mennyire pontosan prediktálja a kimeneti változókat az adott bemeneti adatok alapján. Célunk a költség minimalizálása a modell paramétereinek optimalizálásával. Egy gyakori költségfüggvény a négyzetes hibaösszeg (mean squared error, MSE): [](https://github.com/Miki0195/4.Felev/blob/main/Neuronh%C3%A1l%C3%B3k/vizsga/Negyzeteshibaosszeg.png), ahol **m** a minták száma, **hθ(xi)** az **i**-edik minta előrejelzése, **yi** az **i**-edik minta valós értéke.

**Optimalizáció és tárgya**

Az optimalizáció célja a költségfüggvény minimalizálása, azaz olyan paraméterek (θ) megtalálása, amelyek minimalizálják a modell hibáját.

**Gradiens módszer (Gradient Descent)**  
A gradiens módszer egy iteratív optimalizációs algoritmus, amely a költségfüggvény gradiensét használja a paraméterek frissítésére és mindig arra lépekedünk amerre a legnagyobb a lejtése (gradiens felé) a költségfüggvénynek az aktuális helyen. Gradiens használata különösen fontos, mivel segítségével tudjuk meghatározni a függvény minimumát vagy maximumát.  
*Másképp: A gradiens módszer egy iteratív optimalizációs algoritmus, amely a gradiens irányában történő lépések sorozatával közelíti meg a függvény minimumát.*  
A frissítés szabálya:

[](https://github.com/Miki0195/4.Felev/blob/main/Neuronh%C3%A1l%C3%B3k/vizsga/gradiensmodszer.png)  
ahol α a tanulási ráta, amely meghatározza a lépés nagyságát.

**Update-módszer**  
Az update-módszer (vagy optimalizálási algoritmus) egy olyan módszer vagy algoritmus, amely a neurális hálózat súlyait és paramétereit frissíti és optimalizálja a tanulási folyamat során. Az update-módszer lényegében meghatározza, hogy hogyan módosítsuk a hálózat paramétereit annak érdekében, hogy a költségfüggvény értéke csökkenjen, és a hálózat a tanuló adatokra jobban illeszkedjen.

**Tanulási ráta (Learning Rate)**  
A tanulási ráta egy hiperparaméter, amely befolyásolja, hogy a gradiens módszer milyen nagy lépéseket tesz az optimalizálás során. Túl nagy tanulási ráta esetén az algoritmus nem konvergálhat, míg túl kicsi tanulási ráta esetén az algoritmus lassan konvergál.

**Garanciák a megtalált megoldásra**  
A gradiens módszer nem garantálja, hogy a globális minimumot találja meg, különösen nem konvex költségfüggvények esetén, de gyakran jó közelítést ad a gyakorlatban.

**Feature Scaling (Jellemzők skálázása)**

A feature scaling egy előfeldolgozási technika, amely a bemeneti változók (jellemzők) értékeit egy közös skálára hozza. Gyakori módszerek:

* **Normálás (Normalization):** Az értékeket 0 és 1 közé skálázza.
* **Standardizálás (Standardization):** Az értékeket 0 átlag és 1 szórás köré skálázza.

Feature scaling fontos, mert a különböző skálán lévő jellemzők zavarhatják az optimalizációs algoritmusokat, mint például a gradiens módszert.