



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Szélessávú Hírközlés és Villamosságtan Tanszék



Adaptív moduláció támogatása gépi tanulási módszerekkel műholdas rádiócsatornán

Neurális hálók paraméterei és eredményei

Deli Tamás

2021

Tartalomjegyzék

1. Bevezetés	5
1.1. Problémakör	5
1.2. Alphasat műhold	5
1.2.1. Aldo Paraboni kísérlet	6
1.2.2. Műhold pálya	8
1.3. Diverziti kísérlet vevőállomásai	9
1.4. Gépi tanulás neurális hálóval	11
1.5. Nyers adatok	12
1.6. Mesterséges intelligencia	14
1.7. A neurális hálóról általánosan	15
1.7.1. Visszacsatolt neurális hálók	17
1.8. Miért LSTM ?	17
1.9. Az LSTM bemutatása	18
2. Jelenleg elérhető mesterséges intelligencia vezérelt metódusok	20
2.1. Műholdas rádiócsatornán	20
2.1.1. Regressziós SNR predikció Online Random Regression Forest (ORRF) alkalmazásával	20
2.1.2. Artificial Neural Network (ANN) alapú csillapítás predikció Q-sávban	24
2.1.3. ACM Reinforcement Learning Neural Network (RLNN) segítségével Average Exploration (AE) stratégiával	25
2.2. Földi összeköttetésnél	25
2.2.1. ACM OFDM csatornán Radial Basis Function Neural Network (RBFNN) segítségével	26
2.2.2. ACM 5G csatornán Reinforcement Learning (RL) Q-learning segítségével	27
2.2.3. ACM 5G csatornán Back Propagation Artificial Neural Network (BP-ANN)	27
2.2.4. Online ACM Support Vector Machine (SVM) gépi tanulással	29
2.2.5. Precíziós SNR becslés konvolúciós neurális háló (CNN) segítségével spektrális teljesítménysűrűség (PSD) képek alapján	33
3. Predikció, bináris klasszifikációra, LSTM visszacsatolt neurális hálóval	35
3.1. Adatok előkészítése	35
3.1.1. Adatok struktúrája	35
3.1.2. Adatok szűrése	36
3.1.3. Mérési hibák interpolálása	37
3.1.4. Nem kívánt intervallumok törlése a jelből	38
3.1.5. Feldolgozott hónapok összeillesztése	39

3.2.	Adatkezelés	39
3.3.	Legjobb eredményt elérő neurális háló	40
3.3.1.	Inicializálás	43
3.3.2.	Adatok jellemvonásai (Feature-ök)	44
3.3.3.	Adathalmaz felosztása	46
3.3.4.	Sztenderdizálás	46
3.3.5.	Hiperparaméterek	47
3.3.6.	Betanítás folyamata	47
3.4.	Eredmények értékelése	48
3.4.1.	Predikciós képességek vizsgálata	48
3.5.	Fading jelölés mesterséges intelligenciával	52
3.6.	LSTM rejtett réteg szélesség optimalizálása	53
3.6.1.	Szekvenciális adatok autokorrelációja	54
3.6.2.	75 széles LSTM rétegek	54
4.	Konklúzió	57
5.	Függelék	58

Kivonat

A milliméteres hullámhosszúságú elektromágneses hullámok kommunikációs célokra történő alkalmazása napjaink egyik nagy kihívása. Az 5. generációs mobilhálózat (5G) egyik nagy újítása a 10 GHz feletti sávok bevezetése. Korábban főként műhold-Föld és földfelszíni pont-pont összeköttetések létesítésére alkalmazták és alkalmazzák mind a mai napig ezen magasabb frekvenciákat.

Vezetéknélküli átvitelek esetében a fizikai közeg viselkedése szignifikáns eltéréseket mutat a magasabb frekvenciák esetén az alacsonyabbakhoz képest. Általánosságban elmondható, hogy szabadtéri alkalmazás esetén az időjárás nagy mértékben képes befolyásolni (különösképpen az eső) a csillapítást, melynek oka hogy az alkalmazott elektromágneses hullám hullámhossza összemérhetővé válik a közegben található légköri alkotóelemekkel. Ez a hatás különösképpen igaz műholdas összeköttetések esetén. A meteorológiai jelenségek időben nem állandóak, ugyanakkor jelenlétük szignifikáns jelszint csökkenést tud okozni, sokszor nagy gyorsasággal. A nagymértékű jelszintváltozást fading-nek nevezzük, mely problémát a kommunikációs rendszereknek kezelniük kell.

Az információ átvitel során mindig az a cél, hogy a lehető leggyorsabban történjen meg, egy elvárt hibahatáron belül. Egy változékony közegben ezt adaptívan változtatott modulációval és kódolással (ACM) lehet megvalósítani, melynek célja, hogy a kommunikációs csatornában minél több információt továbbítsunk az elvárt paramétereknek megfelelően.

Dolgozatom fő témája műholdas összeköttetések esetén megvalósított mesterséges intelligencia alapú fading predikció, mely a jövőbeni tervek szerint egy ACM algoritmus alapját képezheti. Az eljárás egy mélytanuló neurális hálón alapul, mely a vett jelszint felhasználva egy bináris klasszifikációt valósít meg, amelynek eredménye a fading esemény előrejelzése a csatornán.

Abstract

The use of millimeter-wavelength electromagnetic waves for communication purposes is one of the great challenges of our time. One of the major innovations of the 5th generation mobile network (5G) is the introduction of bands above 10 GHz. In the past, these higher frequencies have been used and are still used to establish satellite-to-Earth and terrestrial point-to-point connections.

In the case of wireless transmissions, the behavior of the physical medium shows significant differences at higher frequencies compared to lower ones. In general, in outdoor applications, the weather can greatly affect (especially rain) the attenuation, due to the fact that the wavelength of the applied electromagnetic wave becomes comparable to the atmospheric components in the medium. This effect is especially true for satellite connections. Meteorological phenomena are not constant over time, however, their presence can cause a significant decrease in signal levels, often at high speeds. Large signal level changes are called fading, a problem that communication systems need to address.

When transmitting information, the goal is always to happen as quickly as possible, within an expected margin of error. In a variable medium, this can be accomplished by adaptive variable modulation and coding (ACM), which aims to transmit as much information as possible in the communication channel according to the expected parameters.

The main topic of my dissertation is artificial intelligence-based fading prediction implemented in the case of satellite connections, which according to future plans can form the basis of an ACM algorithm. The method is based on a deep-learning neural network that implements a binary classification using the received signal level, resulting in the prediction of a fading event on the channel.

1. fejezet

Bevezetés

1.1. Problémakör

A modern világban nagy szükség van a rádiós kommunikációra. A jövőben arra számíthatunk, hogy az IoT elterjedésével és az eszközök növekvő adatsebességi igénye miatt, egyre zsúfoltabbak lesznek a ma használt frekvencia sávok. Az 5G-ben megfigyelhetjük, hogy az adatsebesség szignifikáns növekedéséhez az architektúrális változások és a technológia fejlődése mellett meghatározó az új frekvenciasávok használata [1]. A nagyobb, még nem használt frekvenciasávokon nagyobb sáv szélességeket használhatunk fel, így nagyobb adatsebességeket érhetünk el. Egy AWGN csatornán a Shannon-Hartley formula szerint (1.1) kapható meg a csatorna kapacitása:

$$C [bps] = B [Hz] \cdot \log_2 \left(1 + \frac{S}{N} \right). \quad (1.1)$$

Ahol a C a csatorna kapacitása, B a csatorna sáv szélessége, S a jel és N a zaj teljesítménye. A nagyobb frekvenciákon előjönnek olyan problémák, amelyeket MHz-es frekvenciatartományban el lehetett hanyagolni. A legnagyobb problémát a troposzférában lévő meteorológiai jelenségek jelentik, ezek közül is kiemelendő az eső csillapítása. Megfigyelhető, hogy a mobilszolgáltatók mikrohullámú összeköttetései nagy esőzések esetén kimerítik a fading tartalékot és kiesnek a szolgálatból. A műhold-Föld közötti kommunikációnál is ugyanúgy problémát okoz az eső. Ennek fizikai oka az, hogy a frekvencia növekedtével az esőcsepp mérete ($1 - 5\text{ mm}$) egyre jobban összemérhetővé válik az elektromágneses hullám hullámhosszával.

Mivel a 40 GHz körüli műholdas kommunikáció még gyerekcipőben jár, ezért vizsgálatra szorul. Ezeket a tudományos kísérleteket hivatott elvégezni az Alphasat műholdon elhelyezett hullámterjedési és kommunikációs kísérlet [2].

1.2. Alphasat műhold

Az Alphasat, más néven Inmarsat-4A F4 az ESA (Európai Űrügynökség) és Inmarsat (műholdakkal foglalkozó telekommunikációs cég) együttműködésével készült geoszinkron pályán lévő műhold. Ezt a nagy, körülbelül 6,5 tonnás űreszközt 2013-ban bocsájtották föl az Ariane 5ECA rakétával. Több tudományos kísérlet kapott helyet a holdon. Ilyen kísérletek a GEO-LEO optikai kapcsolatot vizsgáló TDP (Technology Demonstration Payload), egy Star Tracker aktív pixel detekcióval, űrkörnyezetet vizsgáló TDP és a dolgozat témáját képező Aldo Paraboni hullámterjedési vizsgálatot és kommunikációs

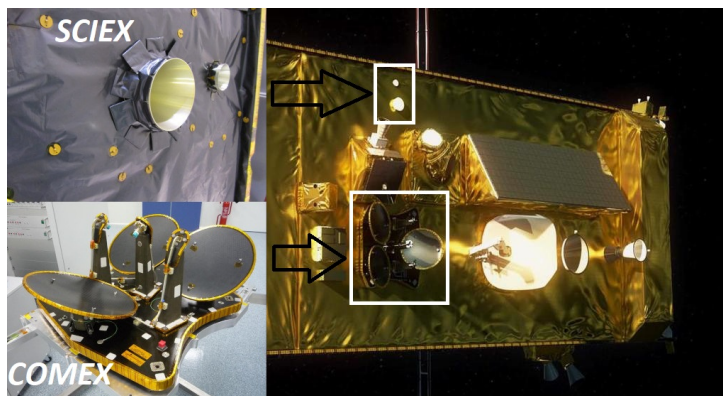
kísérletet végző TDP [2], [3].

1.2.1. Aldo Paraboni kísérlet

Ez a műhold fedélzetén a TDP#5-ös kísérlet, mely két elemből áll.

A COMEX kísérletből (Communication experiment), amely egy transzponder. Az egyik parabola antennán vett jelet relézi egy másik parabola antennára.

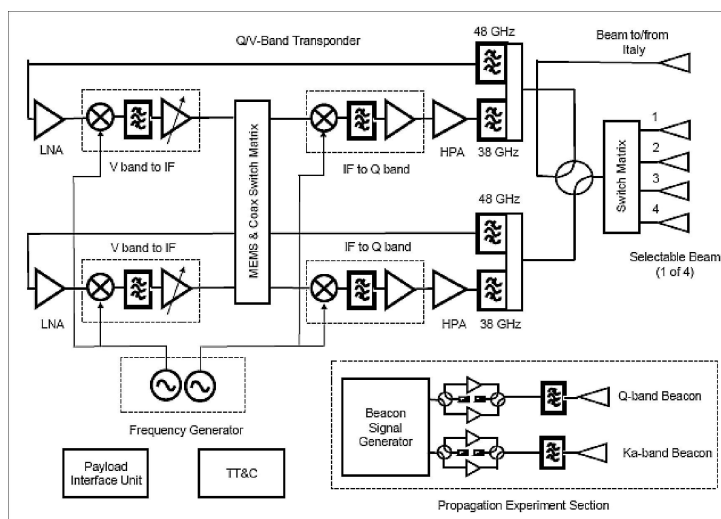
A SCIEX kísérletből (Scientific experiment), amely két különböző frekvencián, két különböző polárizációjú modulálatlan vivőt bocsájt ki két tölcserantennából. A Q sávú beacon jel (39,402 GHz) frekvenciában közel áll a COMEX Q sávú frekvenciáihoz (37,9 GHz és 38,1 GHz), ezért meg lehet vele ítélni a Q sávú DVB-S2 csatorna minőségét.



1.1. ábra. Az Alphasat műhold Föld felé néző oldala két kiemelt komponenssel [4] [5].

1.1. táblázat. SCIEX modulálatlan vivők tulajdonságai és ekvivalens izotropikus teljesítményük [4].

	Ka-sávú beacon	Q-sávú beacon
Frekvencia	19,701 GHz	39,402 GHz
Hullámhossz	15,22 mm	7,613 mm
Polarizáció	Lineáris Vertikális	Lineáris 45°
EIRP	21,5 dBW	29,3 dBW



1.2. ábra. Az Aldo Paraboni kísérlet sematikus rajza [2].

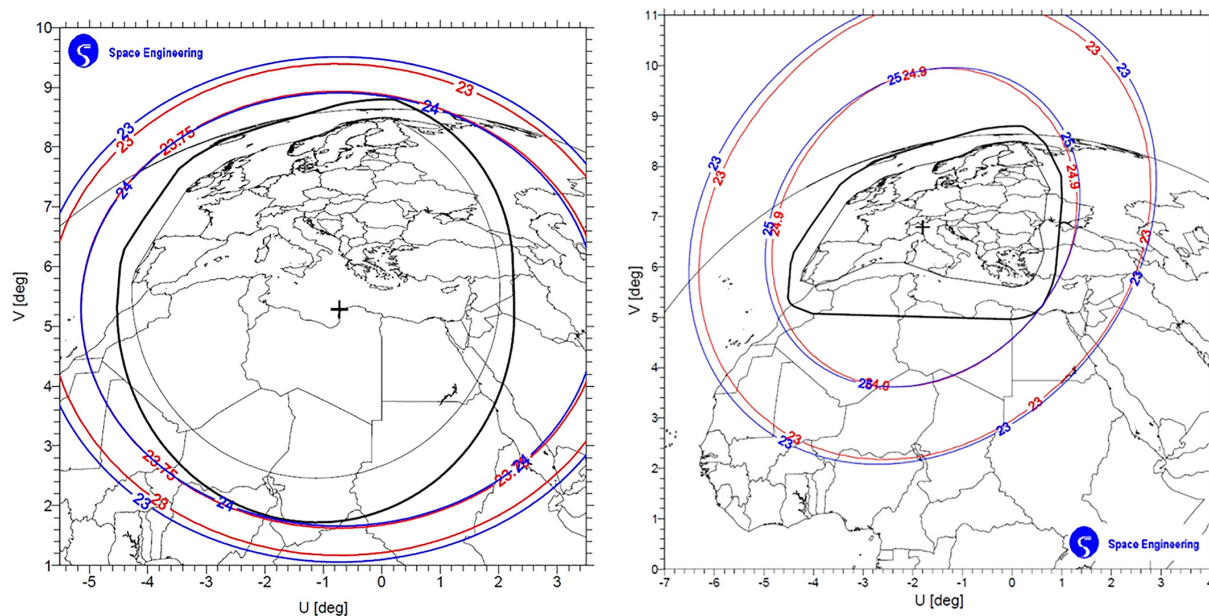
Az 1.2-ábra mutatja a SCIEEX és a COMEX kísérlet sematikus rajzát. Megfigyelhetőek rajta, hogy 48 GHz-en veszi az adást a rendszer, majd ezt IF-re (közép frekvenciára) keverve egy kapcsoló mátrixon keresztül a megfelelő irányba indul el, hogy Q sávú frekvenciára (38 GHz) felkeverjük és erősítsük. Ez az ábra felső része, mely a COMEX kommunikációs kísérletet képviseli. A műholdon három antennát helyeztek el, melyekből kettő Olaszországra irányul, a harmadik pedig az ausztriai Graz felé. Az utóbbinak a jele Budapesten is vehető, így tudott a BME is bekapcsolódni a kísérletbe.

Az 1.2-ábra jobb alsó sarkában a SCIEEX tudományos kísérlet sematikus rajza szerepel. A modulálatlan vevők különböző frekvencián és különböző polarizációval kerülnek ki a tölcserantennából. Tudjuk, hogy azok a jelek, amelyek sokat haladnak az atmoszférában depolarizálódnak, valamint az eső is depolarizálja a jelet a változó közeg miatt, melyet ki is használnak az újabb típusú meteorológiai radarokban. Ezért fontos, hogy a magas frekvencián a polarizációt is vizsgáljuk. Jelen esetben több információt szolgáltat két beacon jel 45°-os polarizációs különbséggel, mint ha egyszerűen vertikális és horizontális polarizációval sugároznánk ki.

Mivel magasabb frekvencián az antenna hatásos felülete kisebb,

$$A_{eff} = G_r \cdot \frac{\lambda^2}{4\pi}, \quad (1.2)$$

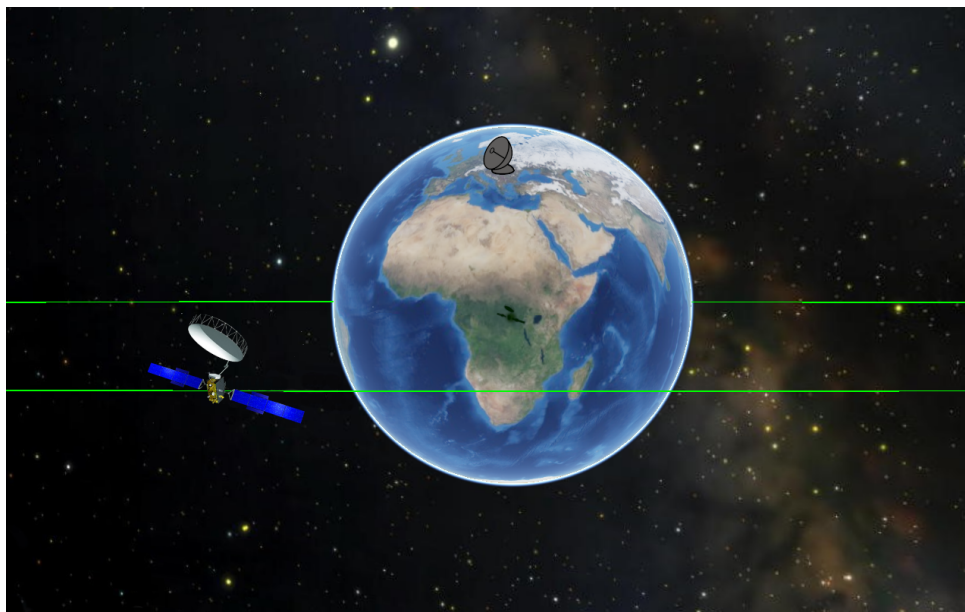
ezért azonos teljesítménnyel kisebb jelet tudnánk venni azonos távolságon. A 1.2- képletben a G_r a vevőantenna nyeresége, a λ a hullámhossz és A_{eff} az antenna hatásos felülete. A Q sávban megfigyelhető nagyobb szakaszcsillapítást a teljesítménnyel vagy antennával kompenzálva körülbelül azonos jelerősséghez juthatunk Európában. Ez a különbség megfigyelhető a 1.1-es táblázatban. Q sávban nagyobb az ekvivalens izotropikus teljesítmény. A 1.3-ábra szemlélteti a Ka és Q sávban generált beacon jelek lefedettségét Európában és környezetében. A kék koncentrikus vonalak a számított, a piros körök a mért antenna nyereséget jelölik.



1.3. ábra. A Ka (bal oldalon) és Q (jobb oldalon) sávú modulálatlan vevők lefedettsége [6]

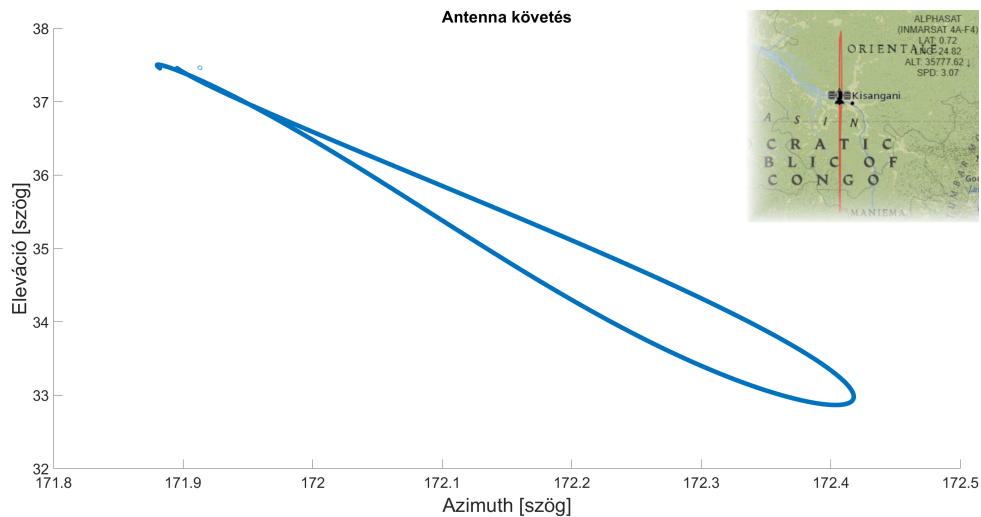
1.2.2. Műhold pálya

Az Alphasat geoszinkron pályán kering (a Föld forgásával megegyező irányban haladó, kis inklinációjú pálya, melynek keringési periódusa megegyezik a Föld forgási idejével, azaz 23 óra, 56 perc, 4 másodperccel), aminek az a következménye, hogy nem egy pontra kell nézni az égen, hanem követni kell. A műhold egy elnyújtott nyolcast ír le, melyet analemmának nevezünk és űrdinamikai okai a nem zérus excentricitásban (elliptikus pálya körtől való eltérését mérő paraméter) és az enyhe inklinációban (egyenlítői síkkal bezárt szög) keresendők. Ha nem követnénk a műholdat, akkor kikerülne az antenna kis irányélességi szögéből ($1,7^\circ$).



1.4. ábra. Műhold, földi állomás és vetület elhelyezkedése [7].

Az 1.4-es ábra mutatja, hogy a hold vetülete nagyjából Afrika közepére esik és itt jelenik meg az analemma ground track. Mivel a vevőállomás nem az egyenlítőn van, hanem 47° -os szélességi körön, nem 90° -os elevációval látszik az égen, hanem körülbelül 35° -os elevációval. Greenwich-től jobbra vagyunk, keleti hosszúságban 19° -ra, az antennát nem teljesen dél felé kell forgatnunk, hanem északról az óramutató járásával megegyező irányba, körülbelül 172° -ra. Ilyen elevációs és azimuth értékek mellett oszcillál a műhold, elevációban fokokat, azimuth-ban pedig tizedfokokat.



1.5. ábra. Analemma görbéből adódó oszcilláló azimuth és elevációs szög, [8]

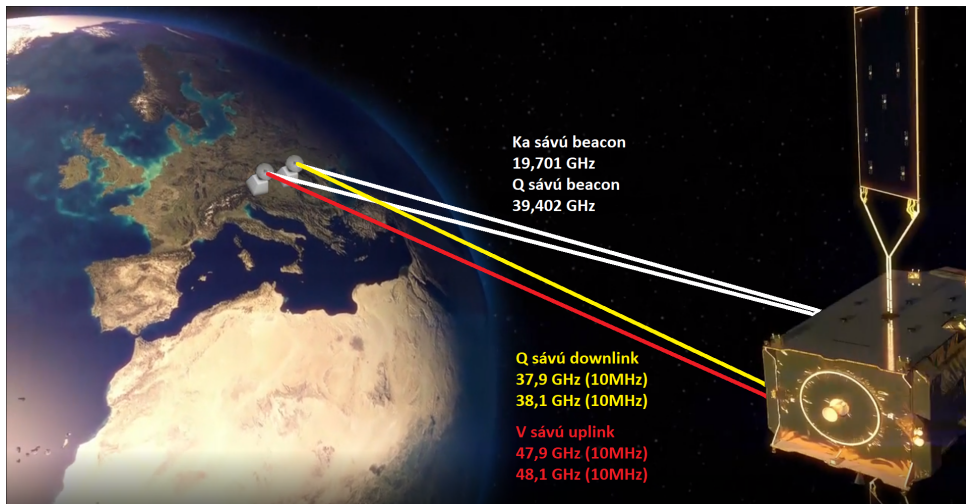
1.3. Diverziti kísérlet vevőállomásai

Két vevőállomás van, mely a COMEX és a SCIEX kísérletekben szerepet játszik a dolgozat keretein belül, ezek a Budapesten található vevőállomás és a Graz-ban található adó/vevőállomás. A dolgozatban említettekén kívül más vevőállomások is vizsgálják a beacon jeleket és használják a COMEX kísérlet transzponder módon működő parabola antennáit. A diverziti kísérlet felállása a 1.6 és a 1.7 -ábrán látszik. A két vevő körülbelül 275 km-re helyezkedik el egymástól, ezért előfordulhat, hogy csak az egyik állomás felett esik az eső.



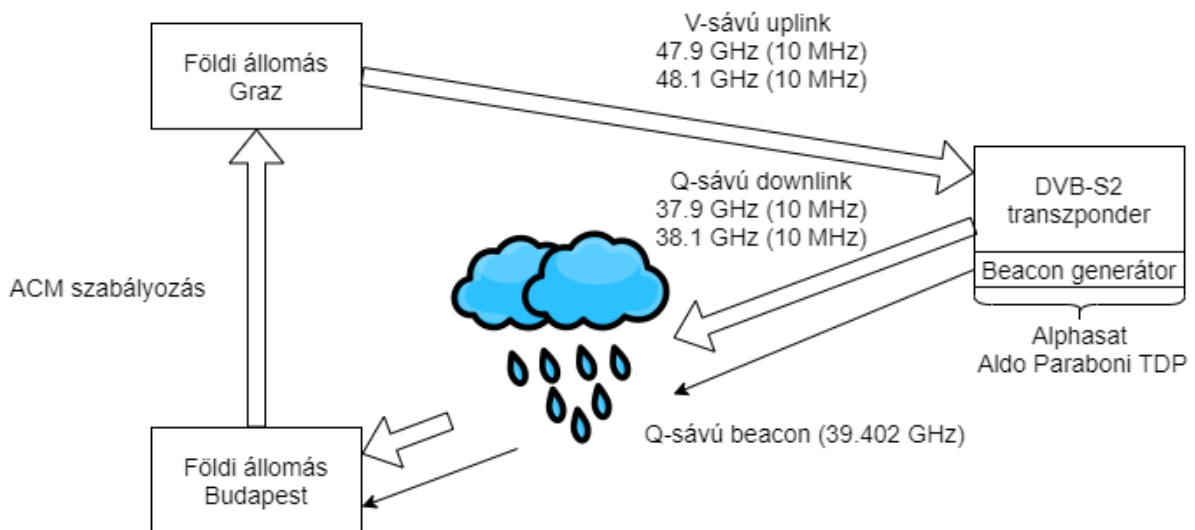
1.6. ábra. Diversity kísérlet földi állomásai [9] [6].

Az 1.6-ábra bal oldalán a Graz-i vevőállomás látszik egy torony tetején elhelyezve. A Graz-i vevőállomás a műholdra V-sávban DVB-S2 formátumú adást továbbít a COMEX kísérletre. A műhold ezt 10 GHz-el kisebb frekvencián transzpondálja, melyet a jobb oldalon lévő Budapesti állomás nagyobb, jobb oldali antennájával veszünk. Mindkét vevőállomáson vizsgáljuk a Ka és a Q sávú beacon jelek állapotát. A budapesti állomás két kisebb antennája felelős ezeknek a beacon jeleknek a vételére. A nagyobb antennával a DVB-S2 adatfolyamot és a Q-sávú beacon jelet is vesszük szintén. A Graz-i állomás a beacon jelek vételét végrehajtja, ezen kívül a DVB-S2 adás spektruma mellé helyez egy beacon jelet.



1.7. ábra. Rádiós kapcsolatok a földi állomások és a műhold között,[1]

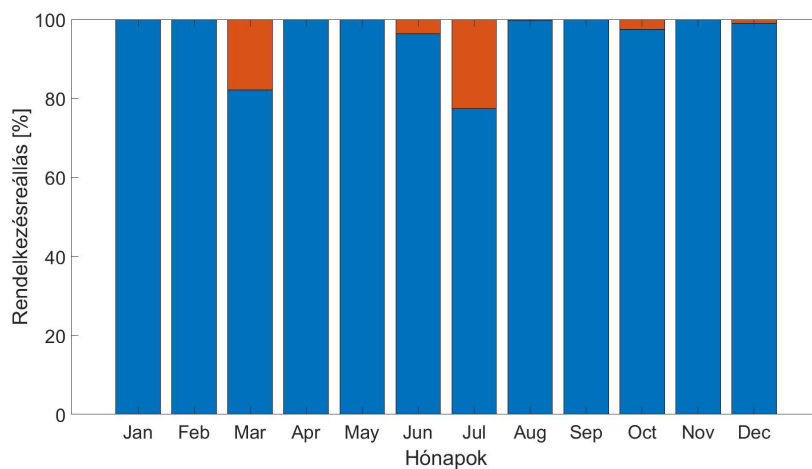
A 1.7-ábra mutatja, hogy két 10 MHz-es sávban történik az adás és a vétel is, valamint a vételi és az adott frekvenciák között 10 GHz különbség van. A DVB-S2 szabvány, a jobb spektrális hatékonyság mellett lehetővé teszi az adaptív modulációt és kódolást (ACM), ezáltal a csatorna zavartűrőképességét lehet állítani, így ha esik az eső és ez fading jelenséget okoz a csatornán, akkor visszaválthatunk egy kisebb állapotszámú modulációra, növelve azt a jel-zaj viszonyt, amelyet a csatorna még elvisel. Például egy jó minőségű rádiós csatornán használhatunk 16 APSK-modulációt, és a jel-zaj viszony romlásával, visszaválthatunk QPSK-ra. A kódolást is a csatorna minőségéhez igazíthatjuk, így szükség esetén egy 8/9-es kódarányú QPSK-ról visszaválthatunk egy 5/6-os kódarányú QPSK-ra.



1.8. ábra. Mérési elrendezés sematikus ábrája

A kísérlet lényege az, hogy a BME V1 épület tetején lévő vevő megállapítja a műholdas rádiócsatorna minőségét (jelenleg a DVB-S2 csatorna jel/zaj viszonyának (SNR) mérésével) és ezt interneten keresztül eljuttatja a Graz-i állomásra. A Graz-i állomás ennek megfelelően állítja a modulációt és a kódolást. Így szeretnénk minél jobb rendelkezésreálláshoz jutni. A Graz-i állomás képes adaptívan adóteljesítményt is állítani, de a műhold már erre nem képes, ezért a műhold és Budapest közötti csatornát csak a modu-

lációval és a kódolással tudjuk a csatornára szabni. Jelen pillanatban a rendelkezésreállítás az áramszünetek és egyéb hibák miatt a 1.9-es ábra szerint alakult 2018-ban.



1.9. ábra. Rendelkezésreállítás 2018-ban.

1.4. Gépi tanulás neurális hálóval

A rendszer jelenlegi felépítésében a Budpesten mérhető DVB-S2 jel/zaj viszony becslésével szabályozza a grazi feladóállomáson alkalmazott kódolási és modulációs módot. Ez alapján a rendszer egy diverziti ACM működést valósít meg.

Az a feltételezésünk, hogy egy neurális hálót betanítva a Q sávú becon jelerősséggel, képes lenne a hálózat prediktálni, hogy a jelszint jelenlegi állapotához tartozó effektív moduláció és kódolásról a következő pillanatban kell-e váltanunk, valamint milyen állapotú és kódarányú modulációt kéne alkalmaznunk. Amennyiben a neurális háló jobban tudja vezérelni a modulációt és a kódarányt, ezzel kihasználva a DVB-S2 által nyújtott lehetőségeket, akkor egy jobb minőségű rendelkezésreálláshoz jutnánk.

A cél első közelítésben egy bináris klasszifikáció, ahol a kész rendszer nagy valószínűséggel meg tudja mondani, hogy a következő pillanatban fading esemény lesz, vagy sem. Egy tanszéki szakdolgozat már foglalkozott a jelen probléma megoldásával szintén neurális hálót alkalmazva [10]. A dolgozat kétféle megoldási módszert javasol.

Az egyik megoldásban fading eseményeket veszünk ki a jelszintből és ezeket felcímkézzük fading illetve nem fading eseményként. Ezekkel betanítva a neurális hálót kapunk egy olyan rendszert, amely egy adatsorra meg tudja mondani, hogy fading vagy nem fading eseményt lát. Tehát ennek a hálónak a bemenetét "végighúzva" egy hosszabb adatsoron képesek leszünk megmondani, hogy hol van a hosszabb adatsorban fading. Ennek a módszernek a hátránya az, hogy a fadingek nem azonos hosszúak, ezért függ a háló bemenetére adott adatsor hosszától, hogy milyen eseményeket tudunk klasszifikálni. Ennek egy megoldása az, hogy a hosszú adatsoron többször futtatjuk végig a háló bemenetét különböző ablakmérettel. Ezekre a különböző ablakméretekre adott válaszokat valahogy kombinálva, jobb minőségben lennénk képesek a felismerésre. Erre egy egész jó megoldás született, azonban betanításbeli problémák álltak fenn vele.

A második megoldásban egy hosszú adatsort veszünk és nem bontjuk fel részekre felcímkézve. A hosszú adatsor minden egyes értékéhez hozzárendelünk címkéket (osztályokat). Tehát minden egyes pontban tudjuk hogy fading esemény van vagy sem.

$$\bar{x} = [x_1 \quad x_2 \quad \dots \quad x_n] \quad (1.3)$$

$$\bar{l} = [l_1 \quad l_2 \quad \dots \quad l_n] \quad (1.4)$$

\bar{x} -vektorhoz hozzárendelünk egy \bar{l} -vektort, ahol az \bar{x} tartalmazza a vett jelszint értékeit másodpercenként és \bar{l} tartalmaz egy bináris osztályozást. A fading események és a nem fading események intervallumokat szelnek ki a hosszú adatsorból, ahol két fading közötti intervallum vagy inter-fading intervallum vagy tiszta ég melletti zavartalan időszak. Ezzel a felcímkézett adatsorral tanítjuk be a neurális háló. A neurális háló a több millió hosszú sorvektoron elkezdi végighaladni és minden egyes lépésnél hoz egy predikciót hogy milyen osztályba tartozhat a következő érték. Tehát ha x_{1000} a bemenete a neurális hálónak akkor próbáljuk megmondani, hogy l_{1001} a fading vagy a nem fading osztályba tartozik. Majd beadjuk az x_{1001} -et és próbáljuk újra meghatározni, hogy mi lehet l_{1002} .

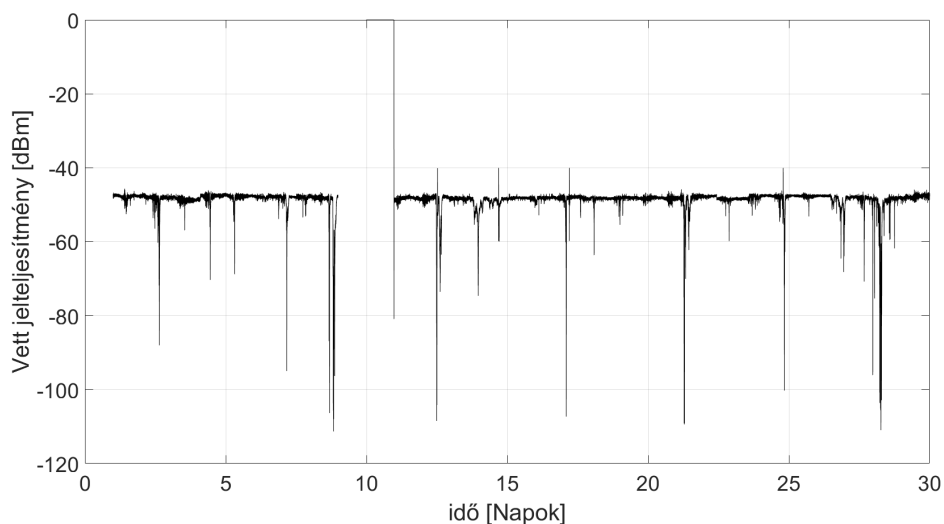
$$\hat{l} = [\hat{l}_1 \quad \hat{l}_2 \quad \dots \quad \hat{l}_n] \quad (1.5)$$

Így végül egy \hat{l} -vektort hozunk létre. Az \hat{l} és az \bar{l} különbsége határozza meg a háló jószágát, ahol az \bar{l} -a ground truth. A betanító próbálja úgy változtatni a hálót, hogy a \hat{l} minél közelebb legyen \bar{l} -hoz. Az $\bar{x} = [x_1 \quad x_2 \quad \dots \quad x_n]$ sorvektorban az \mathbf{n} körülbelül 60 millió, mivel a vevőállomás a jelszintet másodpercenként menti el egy txt fájlba. A dolgozat kereteiben 2 évre visszamenő adattal dolgozom, névszerint a 2018 és 2019 években mért adatokkal.

Én a dolgozatomban a második módszer használom, melyre az előző dolgozatban kevés szó esett, valamint nem tűnt kidolgozottnak a megoldás. Véleményem szerint ez a módszer egy egyszerűbb és jobb minőségű megvalósításhoz vezet, valamint elegánsabb.

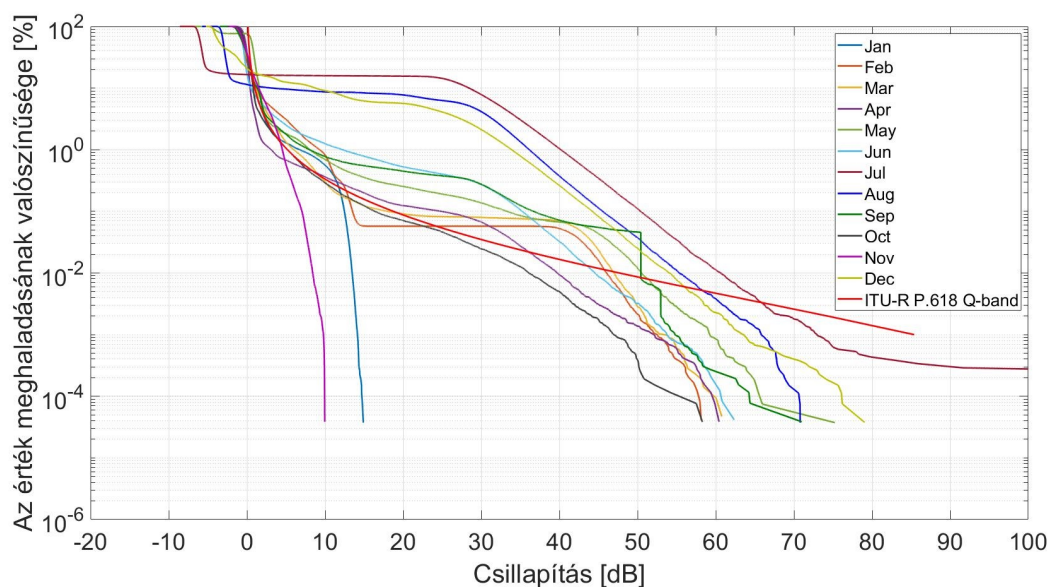
1.5. Nyers adatok

A 1.10 ábrán látszik, a SCIEEX kísérletből származó Q sávú modulálatlan beacon jel vett teljesítménye. Az ábrán megfigyelhetőek troposzférikus jelenségek, mint az eső és szcintilláció. A szcintilláció a légkör refrakciós indexének a dinamikus változása. Ezt a jelenséget figyelhetjük meg, ha télen átnézünk egy kémény felett, ahonnan meleg levegő száll fel. A törésmutató gyors változása „elmosza” a mögötte lévő dolgokat. Ez csillagászati szempontól is kritikus. Minél többet halad a jel a troposzférában, annál jobban ki lesz téve a szcintillációnak, ezért a skandináv országokban, melyek rövidebb szélességi körön fekszenek, a geostacionárius műholdak adásánál már meghatározó zavaró tényező. A szcintilláció felismerhető a jel alapszintjéhez képesti pozitív és negatív irányú gyors oszcillációról, mely nem haladja meg a 2-3 dB-t. Egy előbbi dolgozattól, amely a szcintillációval foglalkozik kiderült, hogy hajnali 3 kor, 10 és 14 óra között, valamint 18 órakor jelenik meg szignifikánsan ez a jelenség [11].



1.10. ábra. Q-sávú beacon jel vett teljesítménye a BME V1 tetjén lévő nagy parabola antennával.

A 1.10 ábrán megfigyelhetőek még mérési hibák is. A hosszabb kiesések áramszünetre vezethetők vissza. Hibák a jelben az alapszinthez képesti (-50 dBm) nagyobb hirtelen ugrások, melyek egyik másodpercről a másikra 5 dB-nél nagyobb jelerősségváltozást mutatnak. A jelet vételi jelszintjét troposferikus zavarok és hidrometeorok nélkül is perturbálja a vevő zajának hőmérsékletfüggése és az, hogy a műholdon lévő antennák nem forognak a BME vevőállomása után, ezért a jelszint az iránykarakterisztika változása szerint csökkenhet. Egyéb perturbáló hatás a pályaadatok (TLE, Two-line element set) elévülése. A kétsoros adatok egy ideális képet nyújtanak melyet a NORAD (North American Aerospace Defense Command) radarokkal derít fel. Ezt az ideális képet idővel (~5 nap) eltorzítja a műholdpálya instabilitása, pályakorrekciók, és az egyéb nem ideális hatások amelyek a műholdat érik. Ezek a hatások szükségszerűvé teszik a pályaadatok napi frissítését, mivel csak rövidtávon minősül megfelelőnek az ideális pályaadatokból való extrapolálás.

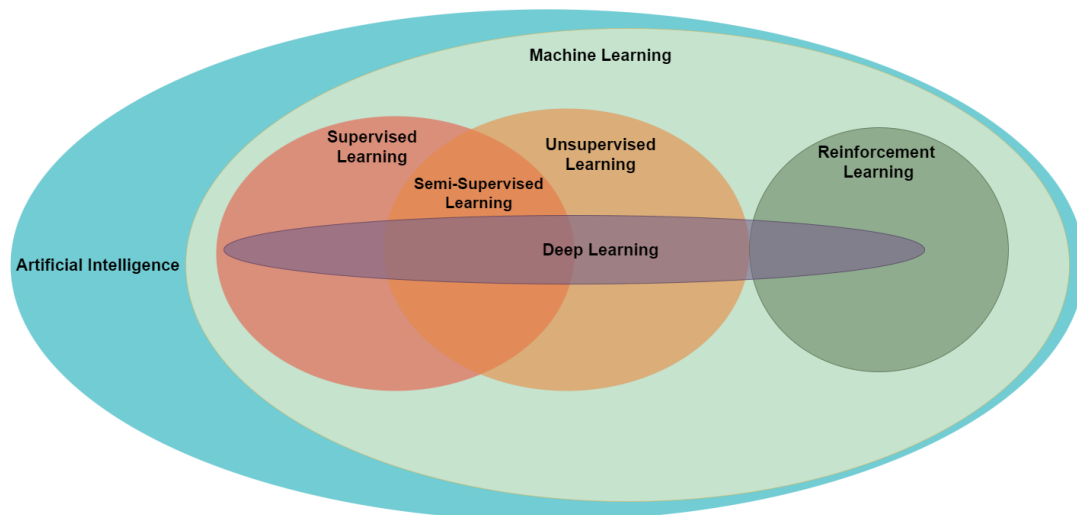


1.11. ábra. Komplementis eloszlás 2018 minden hónapjára.

Az adatoknak ismert a csillapítás komplementis eloszlása is hónapokra bontva (1.11-ábra), tehát tudjuk, hogy milyen hónapokban, milyen tulajdonságai vannak a csillapításnak. Az ITU-R P.618 [12] alapján számolt Q sávú eloszlásfüggvény több adatból empirikusan meghatározott görbe, melyre némelyik hónapban jól, némelyikben kevésbé illeszkedik a rádiós csatorna eloszlásfüggvénye. Ennek az oka egyrészt az, hogy kevesebb adatból becsülünk eloszlásfüggvény és a hónapok más tulajdonságai vannak esőintenzitás szempontjából. Másik oka a jelben lévő hibák. Az görbéken megfigyelhető, hogy egy bizonyos csillapításon túl nem mehetünk, mert ha elhasználtuk a fading tartalékunkat, akkor nem tudunk értelmes jelszintet előállítani.

1.6. Mesterséges intelligencia

Ebben a fejezetben elsőként áttekintést fogok nyújtani a mesterséges intelligencia csoportosításáról, valamint ezeknek a csoportoknak az alaptulajdonságairól. Ezek után a neurális hálóról fogok bővebben írni, mely elvezet a visszacsatolt neurális hálókhoz. A visszacsatolt neurális hálókban belül is az LSTM jelent egy fontosabb témakört, mivel a dolgozatomban ezt a típusú neurális hálót használom.



1.12. ábra. Mesterséges intelligencia osztályozása

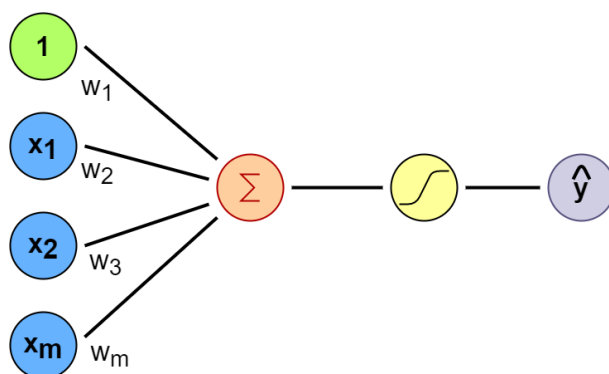
Maga a mesterséges intelligencia vagy angolul Artificial Intelligence (AI) a számítástechnika egyik területe. A Mesterséges Intelligencia elég tág fogalom, a 1.12-ábrán az egyszerűség kedvéért az adattudományt nem ábrázoltam, de érdemes tudni, hogy a mesterséges intelligenciának nem egyetlen részhalmaza a Machine Learning (ML). Megfigyelhető az MI három logikailag különvált ám néhol összefonódó csoportja az 1.12-ábrán. A Supervised Learning jelentése, hogy felügyeljük a tanulást, vagyis mi mondjuk meg explicit módon melyik bemeneti adatra, milyen kimenetet várunk. Ettől gyökeresen eltér az Unsupervised Learning, ahol is nem mondjuk meg mit várunk el, hanem azt várjuk, hogy az algoritmus maga találja meg a logikai csoportokat az adatok között. Amennyiben túl sok az adat, vagy a az adatok címkézése túl sok időt venne igénybe, ekkor használhatjuk a Semi-supervised learning-et. A Semi-supervised learning logikailag az Unsupervised és a Supervised learning között helyezkedik el. A mesterséges intelligencia ezen területe, megoldást nyújt a fel nem címkézett adatok hasznosítására bizonyos feltételek mellett. Mivel általában az adatgyűjtés kevesebb energiabefektetéssel jár, mint az adatok felcímkézése, ahol is egy hozzáértő embernek kell elvégeznie a címkézést, ezért előfordulhat, hogy nagy

az adathalmaz, viszont kevés a betanító adat a Supervised learning algoritmusához. Ilyen probléma lehet például a fehérje struktúrák csoportosítása illetve az olaj jelenlétének a megállapítása. Ekkor alkalmazhatunk Semi-supervised learning metódusokat (Generative modellek, Laplace-i regularizáció), melyek bizonyos törvényszerűségek mentén fel tudják címkézni a címkézetlen adatokat, betanítani a modellt és esetleg iteratív módon konvergálni egy kézzel felcímkézett hasonló méretű adathalmaz mellett vett pontosságához is. A Reinforcement learning egy ismét eltérő módszer. Itt egy Agent operál a környezetben (Environment) A_0 cselekvéssel, mely a környezetet S_0 -állapotba viszi. Ez az S_0 -állapot megfigyelésre kerül az Agent által, melyre egy újabb A_1 cselekvéssel válaszol. Eközben az Agent R_0 jutalmat kap (reward) az A_0 cselekedetért. Megfigyelhetjük még, hogy a Deep Learning, mint a neurális hálók több rejtett réteggel rendelkező csoportja, mind a három alapelvű csoportban megtalálható. A Supervised learningben egyértelmű a jelenléte, hisz a képfelismerés, mint konvolúciós neurális hálós megoldás, gyakori téma, ha machine learning-ról van szó, mivel a képeket kézzel kell címkézni. A Reinforcement learning-ben is könnyen értelmezhető a jelenléte, mivel egy neurális háló lehet az a modell, amely esetleg klasszifikációval a környezet megfigyelése alapján egy cselekedetet választ. Az Unsupervised irányban talán nem egyértelmű, de léteznek olyan neurális hálók amelyek specifikusan klaszterezésre vannak felkészítve, így nincs szükségük címkékre.

A supervised learning-en belül beszélhetünk Support Vector Machine-ről (SVM), lineáris regresszióról, logisztikus regresszióról, Naive Bayes módszerről, döntési fákról (decision tree), K-nearest neighbor (kNN) metódusról, neurális hálókról. Unsupervised tanulásnál beszélhetünk klaszterezésről, k-means algoritmusról, anomália detekcióról, neurális hálókról. Reinforcement tanulás esetén algoritmusok szintjén beszélhetünk diszkrét vagy folytonos állapot és cselekvés térről. Teljes diszkrét esetben például Monte Carlo, Q-learning, SARSA algoritmusokat használhatunk. Diszkrét cselekvés térben és folytonos állapot térben Deep Q Network (DQN) -et használhatunk például. Folytonos cselekvés és állapot térben például Deep Deterministic Policy Gradient (DDPG) algoritmussal oldhatjuk meg a feladatot [13].

1.7. A neurális hálóról általánosan

A neurális hálók a gépi tanulás (machine learning) alá tartoznak [14], [15]. A neurális háló egy olyan struktúra amely képes egy tetszőleges nemlineáris függvény közelítésére.



1.13. ábra. Egyszerű perceptron struktúrája.

A 1.13-ábra egy egyszerű három bemenetű perceptront (mesterséges neuront) ábrázol. A bemeneteket súlyozzuk majd összeadjuk. Ezt az összeget egy aktivációs függvénybe

illesztjük (mely lehet lineáris és nemlineáris), így jutunk az egyszerű perceptron kimenetéhez:

$$\hat{y} = f \left(w_0 + \sum_{i=1}^m x_i \cdot w_i \right). \quad (1.6)$$

A 1.6-képletben az \hat{y} a perceptron kimenete, w_0 az előfeszítés, x_i a bemeneteket jelöli és ezeknek a súlyozása a w_i , valamint az f függvény az aktivációs függvény. A bemenetek alapján a feladatnak megfelelő függvénykapcsolatot szeretnénk elérni. Supervised learning esetén adott y , amely az elvárt kimenet. Valahogy jellemeznünk kell, hogy milyen messze járunk az elvárt kimenettől az \hat{y} -al. Ezt a jellemzést a Loss function testesíti meg.

$$L(f(x^{(i)}; W), y^{(i)}) \quad (1.7)$$

A kimenet az $x^{(i)}$ bemeneti változóktól és a W súlyoktól függ, tehát $f(x^{(i)}; W) = \hat{y}^{(i)}$. A Loss function a kimenet és az elvárt kimenet függvénye. A betanítás során a több ezer dimenziós Loss function globális minimumát keressük. Próbáljuk úgy beállítani a W vektort, hogy a Loss function minimumában legyünk. Bináris klasszifikációnál használhatjuk például az alábbi Loss function-t, melyet Cross-entropy, tehát kereszt entrópia néven illetünk.

$$J(W) = -\frac{1}{n} \sum_{i=1}^n y^{(i)} \log(f(x^{(i)}; W)) + (1 - y^{(i)}) \log(1 - f(x^{(i)}; W)) \quad (1.8)$$

Ahol a 1.8-egyenletben a $J(W)$ a hibafüggvény, mely a súlyoktól függ, $y^{(i)}$ az elvárt kimenet az i -edik mintánál és $x^{(i)}$ az i -edik bemenete a hálózatnak. Szeretnénk megtalálni azt a W vektort, amellyel minimalizáljuk a Loss függvényt:

$$W = \arg \min_W J(W). \quad (1.9)$$

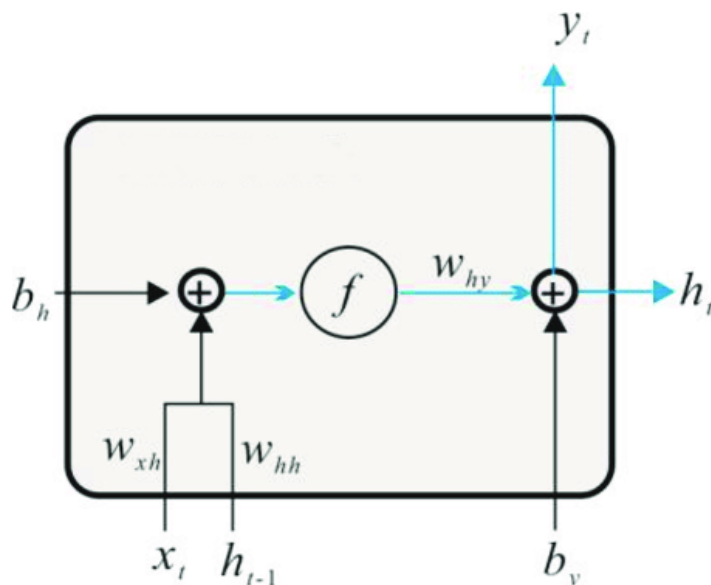
Tehát az elvárt kimenet és az előállított kimenet közötti hibát szeretnénk minimalizálni a súlyok és előfeszítések függvényében. A legegyszerűbb felállásban deriváljuk a Loss function-t a súlyok szerint és a deriváltat egyrészt szorozzuk -1 -el mivel nem a minimum felé mutat, valamint szorozzuk a tanulási rátával (η).

$$W \leftarrow W - \eta \frac{\partial J(W)}{\partial W} \quad (1.10)$$

A 1.10-képletben a $J(W)$ a Loss-function, az η a tanulási ráta és a W a súlyokat tartalmazó vektor. Itt arról van szó, hogy a legegyszerűbb esetben Gradient Descent metódust használva a mostani súlyokat felváltjuk a Loss-function deriváltjával korrigált súlyokkal, így ha például a W_{11} súly hibafüggvény szerinti deriváltja kis pozitív szám, akkor a W_{11} -es súlyt csökkentenünk kell egy kis számmal. Ennek az iteratív módszernek a sebességét továbbá megszabja az η tanulási ráta. A fenti példa egy statikus Feed Forward Network-re igaz [15]. hasonló alapokra építenek a bonyolultabb optimalizáló eljárások is. A hibafüggvény súlyok szerinti deriváltjával való módszert Back Propagation- nek hívjuk és matematikailag a deriváltak lánc szabályával lehet feloldani, hogy egy a hálózat elején lévő súlyt milyen irányba változtassunk meg kicsit, hogy csökkenjen a hibafüggvény.

1.7.1. Viisszacsatolt neurális hálók

A szekvenciális adatok feldolgozására RNN-re (Recurrent Neural Network) van szükség. Az RNN egy visszacsatolt hálózat, így az előző pillanatokból származó információ is szerepet tud játszani a jelen pillanatbeli kimenet képzésében.



1.14. ábra. RNN cella [16].

A 1.14-ábra egy általános visszacsatolt neurális hálózati elemet szemléltet, ahol az y_t a t időpontbeli kimenet, x_t a t időpontbeli bemenet, h_t a t időpontbeli belső állapot, h_{t-1} az előző időpontbeli belső állapot, b_h és b_y az előfeszítések, W_{hy} , W_{xh} és W_{hh} a súlyozások, valamint f az aktivációs függvény, mely lehet például tangens hiperbolikus.

Létezik még egy GRU típusú visszacsatolt neurális hálózati struktúra, melyet kipróbáltam a betanítás során, ám nem járt megfelelően pontos eredménnyel, ezért a következőkben LSTM rétegekkel foglalkozok.

1.8. Miért LSTM ?

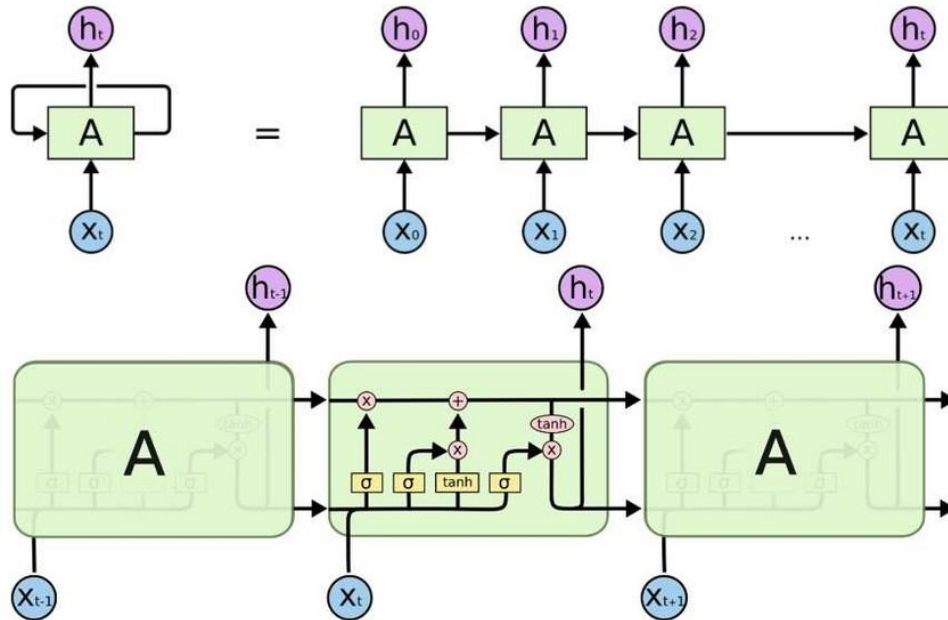
A dolgozatban megoldani kívánt probléma egyértelműen visszacsatolt neurális hálót követel meg. Egy szimpla RNN is képes visszamenőleg információkat tárolni, azonban két probléma van vele. Egyrészt nem képes hosszútávú függőségeket tárolni, csak rövidebb ideig képes fenntartani az előzményeket. Másrészt az elszálló és a 0-hoz tartó gradiens (exploding gradient, vanishing gradient) is nehezíti a betanítás sikerességét. Ezekre a problémákra nyújt megoldást az LSTM (Long Short-Term Memory). Tehát képes tárolni régebbi információkat és a gradiensek a belső felépítésből adódóan nem szállnak el vagy tartanak a 0-hoz. Ezen kívül megemlítendő, hogy a BPTT (Back Propagation Throug Time) során a Loss function szerinti deriváltakat egyszerűbb előállítani, mivel egy csatornán halad keresztül a derivált az időn keresztül, ez is segíti a betanítás lépéseit.

Indok még az LSTM használatára, hogy a jelenlegi problémára megoldást kereső dolgozatban is LSTM hálózatot használtak és használható eredményekhez jutott a dolgozat [10]. Valamint a MATLAB példaprogramok, név szerint a Japán magánhangzók felismerése és az EKG jelek klasszifikációja is sikeresen alkalmazza ezt a típusú visszacsatolt

hálózatot, ezért bizalmam volt abban, hogy képes lesz felismerni fading jelalakokat az adatsorban.

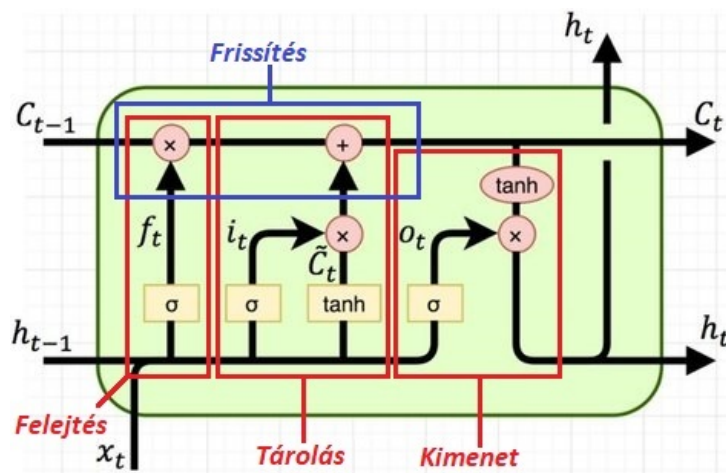
1.9. Az LSTM bemutatása

Az LSTM a Long Short-Term Memory-t jelenti. Ez is jelzi, hogy képes hosszabb időre tárolni az előző bemeneteknek megfelelő információt.



1.15. ábra. LSTM réteg kiterítve időtartományban [17].

A fenti 1.15 -ábrán a bal felső sarokban egy visszacsatolt cellát láthatunk. Ez jelen esetben egy LSTM réteget jelöl. Megfigyelhetjük, hogy a réteg által keltett értékek, mint a kimenet és a cella állapot a következő időpillanatban a réteg bemenetét fogja képezni az általános x_t bemenetével együtt. Ezt a modellt, ha időtartományban kibontjuk, a jobb felső ábrához jutunk. Az alsó szegmens szemlélteti, hogy itt egy LSTM celláról van szó, melynek egy sajátos belső struktúrája van.



1.16. ábra. LSTM cella struktúrája [17].

A 1.16-es ábra egy LSTM cellát mutat. Az LSTM cella az előző időpontbeli értékekből az előző kimenetet és az előző belső állapotot viszi tovább. A kimenet átadása képviseli a rövid időtartamú memóriát és a belső állapot képviseli a hosszú távú memóriát. A felső csatornán szabad utat kap a derivált az időn keresztüli áthaladáshoz, így effektíven ki lehet számolni, hogy hogyan állítsuk a háló paramétereit a jobb megoldás érdekében [18].

$$\begin{aligned}
i_t &= \sigma(x_t U^i + h_{t-1} W^i) \\
f_t &= \sigma(x_t U^f + h_{t-1} W^f) \\
o_t &= \sigma(x_t U^o + h_{t-1} W^o) \\
\tilde{C}_t &= \tanh(x_t U^g + h_{t-1} W^g) \\
C_t &= \sigma(f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t) \\
h_t &= \tanh(C_t) \cdot o_t
\end{aligned} \tag{1.11}$$

Ahol az x_t a bemeneti vektor, h_t a jelen pillanatbeli kimenet, h_{t-1} az előző kimenet, C_t a jelen pillanatbeli belső állapot, C_{t-1} az előző belső állapot, W^i, W^o, W^g és W^f a kimeneti függvény súlyozásai, U^i, U^f, U^o és U^g a bemeneti vektor súlyozásai. Megfigyelhetők a belső struktúrán különböző részegységek, melyek működését az 1.11-es egyenletek írják le. A felejtés kapu egy szigmoid logisztikus görbe szerinti aktivációs függvényt használ, mely bármely bemenetét 0 és 1 közé helyezi. Mivel a nemlineáris aktivációs függvény az előző időpontbeli belső állapottal (C_{t-1}) szorozva van, ezért a bemeneti értékével ($x_t U^f + h_{t-1} W^f$) modulálni lehet, hogy mennyi információ jusson be a jelenlegi belső állapotba. A tárolás kapu szabályozza, hogy a milyen információk jussanak be a belső állapotba. A kékkel kijelölt frissítésnek az a jelentése, hogy f_t -vel megmondjuk mennyit felejtünk és i_t -vel megmondjuk mennyit adjunk hozzá a belső állapothoz. A kijelölt kimenet részen egyszerűen előállítjuk a kimenetet (h_t) a jelenlegi bemenet, az előző kimenet (h_{t-1}) és a jelenlegi belső állapot (C_t) alapján.

2. fejezet

Jelenleg elérhető mesterséges intelligencia vezérelt metódusok

A következőkben be fogok mutatni pár különböző típusú mesterséges intelligencián alapuló ACM módszert. Két csoportba fogom ezeket sorolni, műhold-Föld közötti rádiócsatornára értelmezett megoldásokra és földi összeköttetések közötti rádiócsatornára értelmezett megoldásokra.

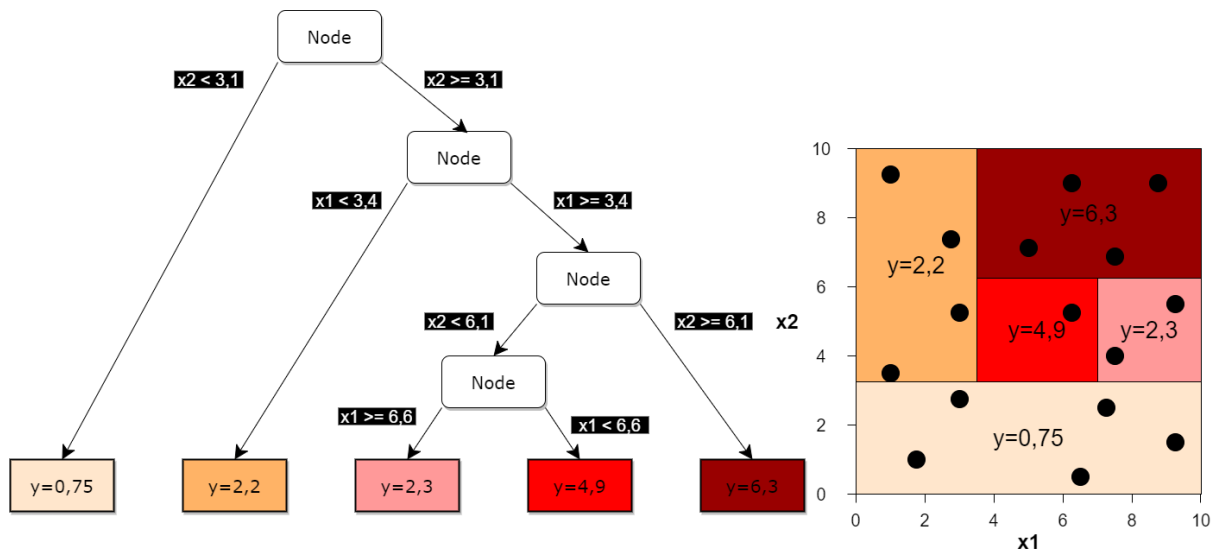
2.1. Műholdas rádiócsatornán

Ebben az alfejezetben a Föld-műhold közötti rádiócsatornán alkalmazott létező ACM metódusokat fogom bemutatni, melyeket mesterséges intelligenciával oldottak meg.

2.1.1. Regressziós SNR predikció Online Random Regression Forest (ORRF) alkalmazásával

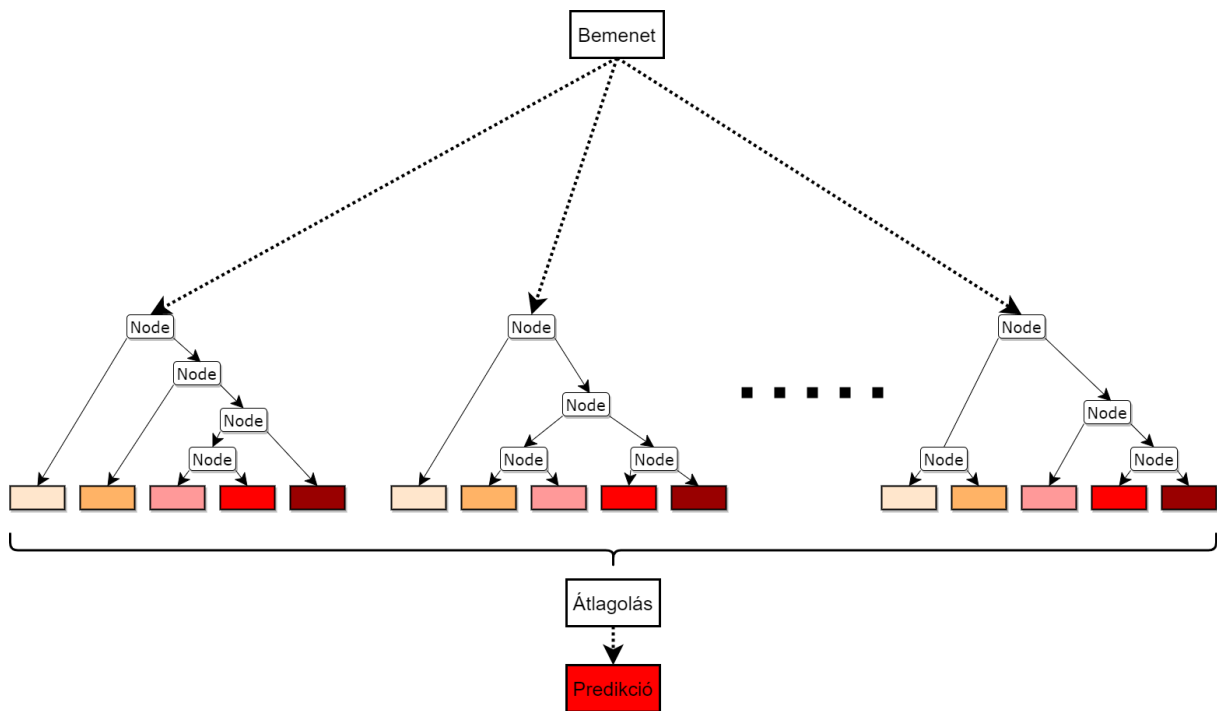
A kísérlet egyik földi állomásán zajló tudományos kísérlet célja megegyezik a dolgozat tematikájával. Azonos műholdon és rádiós csatornán dolgoznak, az a különbség, hogy SNR adatokkal dolgoznak és ezekből állítanak elő egy skalár SNR predikciót. Ezt a dolgozatot fogom bemutatni részletesebben [19].

Az Online Random Regression Forest (ORRF) tehát döntési fákkal dolgozik melyeket végigjárva egy értékhez jutunk. Az elmúlt években a hagyományos véletlenszerű erdők (RF) online sémája nagy figyelmet keltett, mivel olyan szekvenciális adatokat tud kezelni, amelyek eloszlása megváltozik az előrejelzési folyamat során. Ahelyett, hogy módosítanánk az off-line tanított RF-ek szerkezetét, a regressziós előrejelzés során memóriával ruházzuk fel az RF-eket egy online súlytanulási módszer segítségével, amelyet Online Weight Learning Random Regression Forest-nek (OWL-RFR) nevezünk. Pontosabban, a levelek súlyát egy új, adaptív sztochasztikus gradiens módszer alapján frissítik. Így a levélszintű súly tárolja a múltbeli adatpontokból származó információkat a jövőbeli korrelált előrejelzéshez. A faszintű súlyhoz képest, amely csak azonnali memóriával rendelkezik az aktuális előrejelzéshez, a levélszintű súly hosszú távú memóriát nyújthat. Az OWL-RFR-rel végzett numerikus kísérletek figyelemre méltó javulást mutatnak a prediktív pontosságban számos gépi tanulási adatkészletben, összehasonlítva a hagyományos RF-kkel és más online módszerekkel. A levélszintű súly hosszú távú memóriáját használó súlymeghatározás hatékonyabb, mint a faszintű súlytól való közvetlen függőség.



2.1. ábra. Regressziós fa működési elve [20].

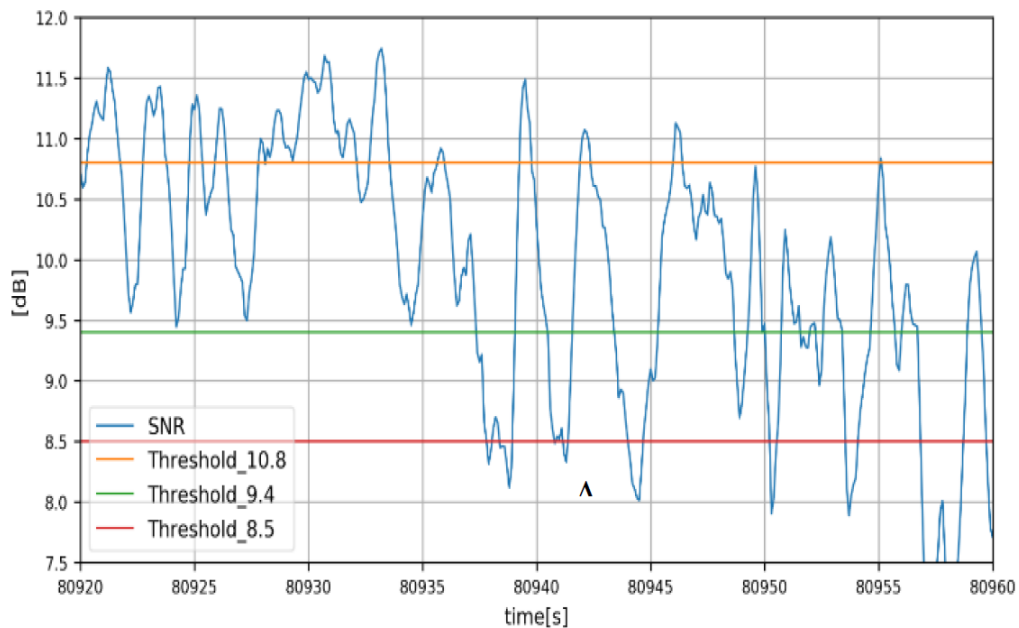
A 2.1 bemutatja a regressziós fa működését két prediktor esetén. A fán végighaladva egy skalár kimeneti értékhez érünk. Megfigyelhető hogy a kimeneti értékek merőleges szeleteket vágnak ki a létező bemeneti kombinációkat ábrázoló síkból. A regressziós fa nemlineáris függőségeket is képes kezelni ezen tulajdonsága miatt. Itt nem egyenes illesztésről van szó, mint a lineáris regressziónál, hanem inkább a cél függvény szakaszaival való közelítésről.



2.2. ábra. Regressziós fák aggregálása és eredményeiknek átlagolása.

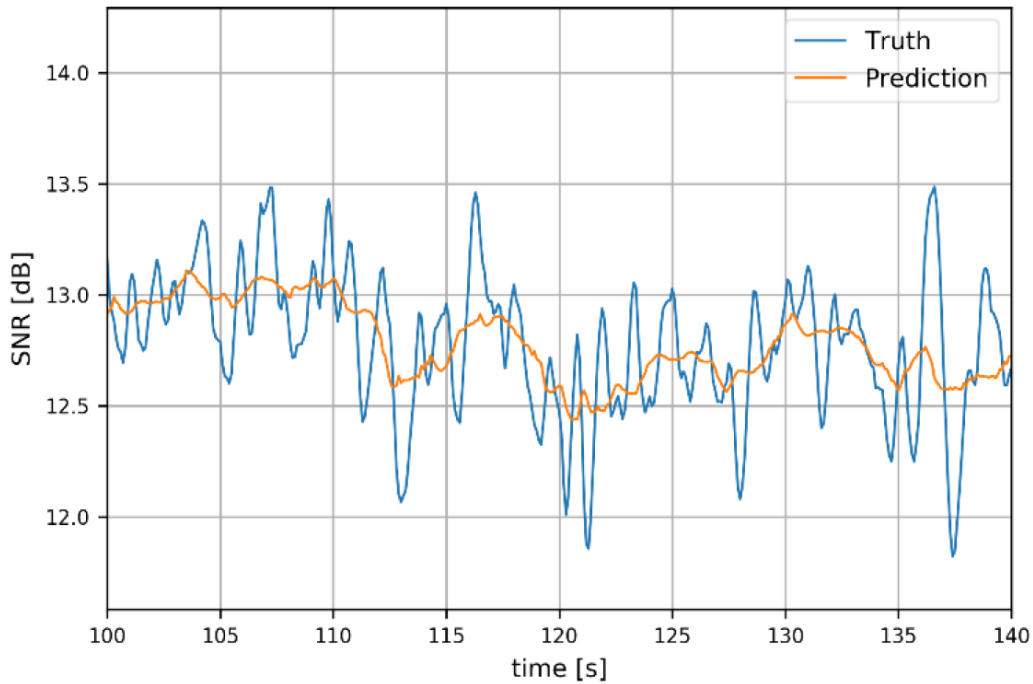
A döntések pontosítása végett több párhuzamosan létező fa eredményét átlagolják (ensemble, aggregation). Több dimenzió esetén, több fának különböző bemeneti variációkat táplálnak be (bootstrap). Ezeknél a módszereknél fontos pontosan specifikálni a fák számát a megfelelő pontossághoz. Ebben az esetben a fák száma az erdőben 20, me-

lyek mélysége 10 volt. Az online tanulás miatt az algoritmus képes percek alatt igazodni bizonyos környezetekhez.



2.3. ábra. ACM döntési határai jel/zaj viszony alapján [19].

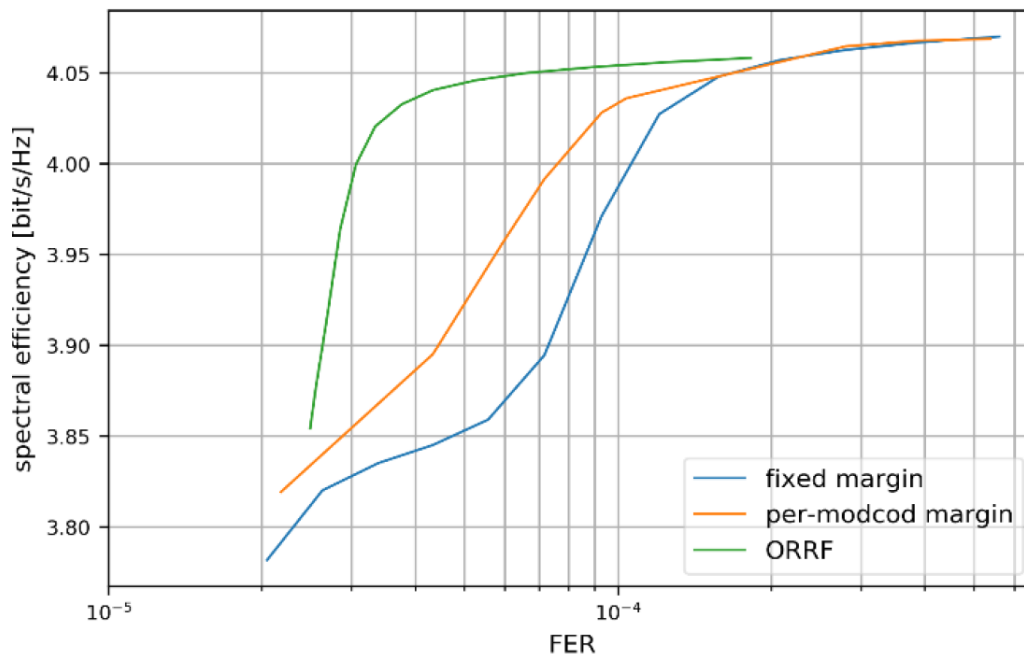
A 2.3-ábra az SNR mért értékét jelöli kézzel. Más színekkel konstans vonalak is láthatóak az ábrán, ezek azok a határok, amelyek felett vagy alatt más MCS-t használunk. A cél az, hogy úgy válasszuk meg a MCS-t, hogy ne lőjjünk túl az elvárt robosztusságon. Itt ebben az esetben egy fix határok alapján működő rendszer van ábrázolva. Ennek a módszernek a hátránya, hogy olyan időszakokban, amikor a vételi jelszint nagyon fluktuál, nagyobb távolságra van szükség az SNR határok között, mint amikor egy tiszta időben lévő ("clear sky") jelszintet nézünk. Ezt az ismeretet fel tudjuk használni úgy, hogy amikor nagy az SNR fluktuációja (pl. esik az eső) egy nagyobb határtávolságokkal rendelkező sémát alkalmazunk, valamint amikor a rádiós csatornán az SNR viszonylag stabil értéket mutat (tiszta az ég), akkor egy kisebb SNR határtávolságú sémát alkalmazunk. Ezt a módszert hívjuk per-modcod határoknak.



2.4. ábra. SNR mért értékek és predikciós értékek [19].

A 2.4-ábrán a dolgozat eredménye látható időtartományban. Itt a kézzel jelölt jel a mért SNR és a narancssárgával jelölt az SNR predikció, melyet a regressziós fák átlagolt eredményéből kapunk. Megfigyelhető egyfajta periodicitás az SNR jelalakban, ez nem csak a fehér zajnak tudható be, hanem kvázi-periodicitásnak is mondható. A dolgozati korrelációs analízise a szekvenciális adatoknak azt mutatta, hogy a korreláció ± 10 mintán belül gyorsan esik, ez azt jelenti, hogy a mesterséges intelligenciás megoldás csak rövid távú mintákat tudna követni a jelalakban.

A dolgozatban prezentált minőségbeli összehasonlítási eredményeket az Alphasat műholdról érkező Q sávú modulálatlan vivő segítségével alkották meg, melyet Graz-ban egy 3 méteres antennával vesznek. A teljesítmény értékeket 10 minta/s mintavételezési sebességgel vették figyelembe. Annak érdekében, hogy modellezzék az SNR variációkat, V sávba kellett átskálázni az adatokat. Ezt az ITU-R P.618-11 es ajánlásának segítségével tették meg. A lenti 2.5-ábra a különböző metodikájú ACM megoldások spektrális hatékonyságát vizsgálja a FER függvényében.



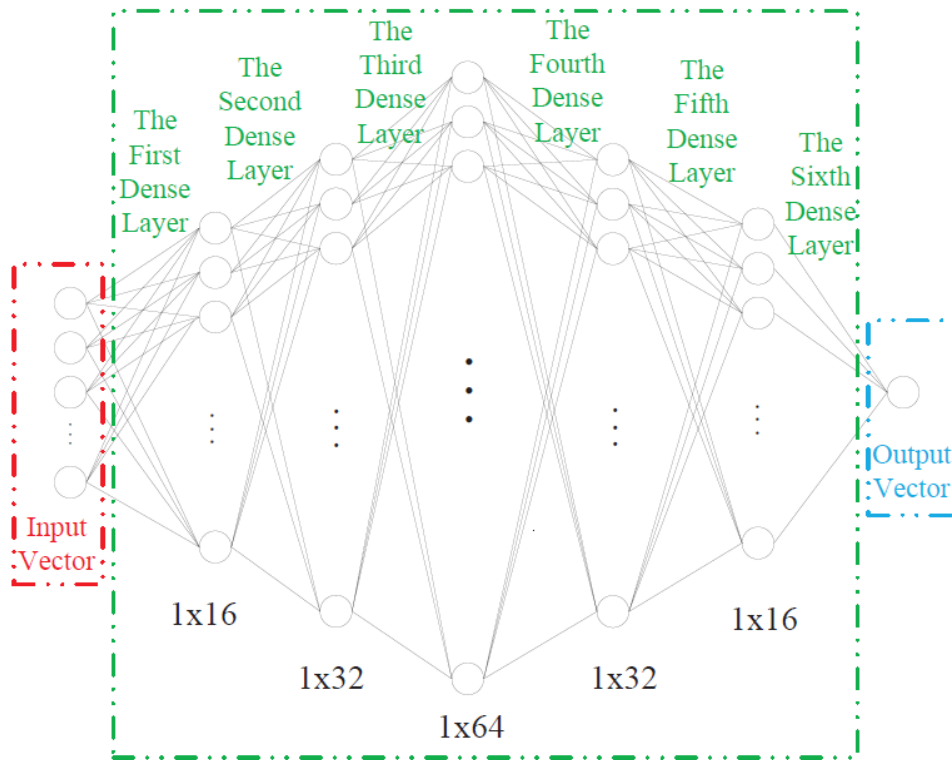
2.5. ábra. Spektrális hatékonyság Frame Error Rate (FER) függvényében [19]

A görbék minnél feljebb és minnél jobban balra futnak, annál jobb a rendszer. Itt a legrosszabb metódus a fix határokkal rendelkező rendszer, mely kisebb spektrális hatékonyságot mutat, több kerethibával. A per-modcod növeli a rendszer minőségét a dinamikája miatt. A legjobb eredményt az ORRF típusú rendszer éri ebben az esetben.

2.1.2. Artificial Neural Network (ANN) alapú csillapítás predikció Q-sávban

Ez a dolgozat [21] az Aldo Paraboni Q-sávú beacon jelét használja, melynek a csillapítása az egyik bemeneti változó a betanítás során. A megoldás egy FFN-ön alapul. Egy 6 dense rétegű fully connected rétegek összekapcsolásából álló neurális háló szolgáltat predikciókat. A Multi Layer Perceptron (MLP) bemenetei különböző típusú meteorológiai eszközök és az előző percbeli csillapítási érték. A dolgozat próbálkozik az előző csillapítás és a disdrometer nélküli predikcióval, ennek az eredménye az, hogy a disdrométer nem szignifikáns a predikció pontosságában, viszont az előző csillapítási érték szerepet játszik a megfelelő regressziós pontosságban. A dolgozatban a komplexitást csökkentése érdekében $1/60Hz$ es mintavételezéssel dolgoznak.

A 2.6-ábra demonstrálja az ANN architektúrát, mely regressziós feladatot lát el.



2.6. ábra. ANN architektúra túlzott csillapítás predikcióra

2.1.3. ACM Reinforcement Learning Neural Network (RLNN) segítségével Average Exploration (AE) stratégiával

Ebben a dolgozatban [22] műholdas kommunikációra optimalizáltak mesterséges intelligenciát. Az RLNN feloldása a Reinforcement Learning Neural Network. Ez azt jelenti, hogy Reinforcement learning típusú mesterséges intelligenciát alkalmaznak, amely gyökeresen eltér a supervised és az unsupervised learning témakörétől, azonban a környezetben a megfigyeléseket figyelembe véve egy neurális háló kimenete dönti el a környezetben végrehajtandó cselekedetet, tehát a reinforcement learning struktúrában az agent a supervised learning-hez közel álló neurális háló. Az RLNN alkalmazhat deep neural network (DNN)-öt is, amely lehetővé teszi az igen komplex feladatok megoldására is, melyben akár szerepelhet képfelismerés is.

Average Exploration stratégiával stabilizálni és általánosítani lehet a betanítást, így több alkalommal fog általános és megbízhatóan konvergens betanításokat produkálni. Amikor a környezet felfedezésére van szükség, lehet alkalmazni Q-learning vagy SARSA algoritmust. Ezen eljárás szerint szeparálva van a fő hálózat és a cél hálózat. A neurális háló egy 4 rétegű fully connected struktúra.

A modulációs sémák a következők, BPSK, QPSK, 8PSK, 16PSK.

2.2. Földi összeköttetésnél

Ebben az alfejezetben a földi összeköttetéseken alkalmazott létező ACM metódusokat fogom bemutatni, melyeket mesterséges intelligenciával oldottak meg.

2.2.1. ACM OFDM csatornán Radial Basis Function Neural Network (RBFNN) segítségével

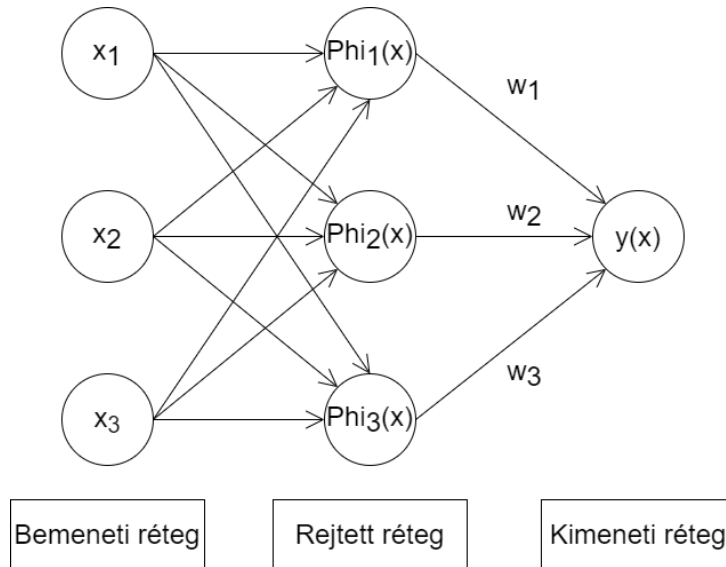
Ismét egy más típusú mesterséges intelligenciával való ACM megoldást fogok prezentálni [23]. A radiális bázisfüggvény egy olyan függvény, melynek értéke az origó vagy valamilyen center távolságától függ, ahol a távolság általában Euklideszi távolság, tehát L_2 norma, melyet $\| \cdot \|$ -el jelölünk.

$$\begin{aligned}\Phi(x) &= \hat{\Phi}(\|x\|) \\ \Phi(x) &= \hat{\Phi}(\|x - c\|)\end{aligned}\quad (2.1)$$

A 2.1-egyenletek a radiális bázisfüggvény definíció, mivel az a függvény, amely ezt teljesíti, nevezhető radiális bázisfüggvénynek. Gyakran kollekciónban használják őket, $\{\Phi_k\}_k$, melyek bázisként szolgálnak és kifizítenek egy teret valamilyen függvény térben.

$$y(x) = \sum_{i=0}^N w_i \Phi(\|x - x_i\|) \quad (2.2)$$

Radiális bázisú függvények összegével lehet folytonos függvényeket $y(x)$ közelíteni. Mivel a mesterséges intelligencia tulajdonképpen egy függvényközelítés, ahol a bemenetektől az elvárt kimeneteket képző függvényt keressük, ezért van relevanciája ennek a metodikának. Ez esetben N radiális bázisfüggvény x_i középpel és súlyozva állítja össze a célfüggvényt. Ez a technika tekinthető Feed Forward Network-nek (FFN). Ez a neurális háló általában 3 rétegből áll, a bemeneti, a rejtett és a kimeneti rétegekből.



2.7. ábra. Neurális háló struktúra radiális bázisfüggvények alkalmazásához

A hálózat kimenete a lineáris kombinációja a bemenet radiális bázisfüggvényének. Más szóval a rejtett réteg radiális bázis függvényeket használ aktivációs függvényként, mint például a Gauss függvény.

$$\Phi_i(x) = e^{-\frac{(x-x_i)^2}{r^2}} \quad (2.3)$$

A bemenetek ez esetben SNR, BER és Present Modulation (PM) és Code Rate (CR)

voltak. Az FNN a legjobb pontosságot 500 neuronnal érte el, mely 89 % volt. Az RBFNN-t használják időbeli adatok predikciójára és klasszifikációra. Itt egy skalár kimenetről van szó amely vezérli a modulációt és a kódolást.

2.2.2. ACM 5G csatornán Reinforcement Learning (RL) Q-learning segítségével

Egy 5G rádiócsatornán való adaptív modulációt és kódolást vezélő mesterséges intelligenciát fogok összefoglalni pár mondatban [24]. Az adaptív modulációt Reinforcement Learning-el tanított modell vezérli. A betanításnál a bemeneti értékei az agent-nek a Channel Quality Indicator (CQI) és a reward. Ezek függvényében kell megtalálni a megfelelő Modulation and Codig Scheme-et (MCS). A környezet a bázis állomásnak a megfelelő MCS kiválasztása esetén a Block Error Rate (BLER) és a spektrális hatékonyság függvényében ad jutalmat.

2.2.3. ACM 5G csatornán Back Propagation Artificial Neural Network (BP-ANN)

Ebben a dolgozatban [25], egy perceptronokból felépülő mesterséges intelligenciát fogok összefoglalni, mely vezérli az adaptív modulációt és kódolást. Az ACM technológiák csatorna állapot információ alapján választják ki a megfelelő MCS-et. A vevő által visszajuttatott információk, már nem feltétlen lesznek megfelelőek, mivel időben gyorsan változhat a csatorna. Ez a dolgozat Back Propagation Artificial Neural Network-el (BP-ANN) kívánja elérni a legnagyobb elérhető adatsebességet és spektrális hatékonyságot. A neurális háló a csatornán fellépő interferenciát, zaj becslést és csatorna becslést használta bemenetként. Eddig az MCS szelekció kognitív heterogén hálózatok által történt deep reinforcement learning segítségével. Az Enhanced mobile broadband (eMBB) nagy adatsebességet nyújt az 5G rendszerben. A sebesség csúcserő értéke elérheti a 20 Gbps értéket is, ahol a várható értéke az adatsebességnek egy felhasználóra meghaladhatja a 100 Mbps-ot. A rádiós csatornán OFDM-et használnak, melyeket Physical Resource Block-okra (PRB) osztanak, mely 12 alvivőt aggregál.

A BP-ANN tulajdonképpen egy egyszerű fully connected multilayered feedforward hálózatot jelent. A BP-ANN használható, hogy multidimenzionális adatokra állítsunk fel egy függvény közelítést, amely egy klasszifikációként szolgálhat. A rejtett réteg a nemlineáris számítások eszközölésére hivatott, mely csökkenti a hibát. A neuronok minden rétegben össze vannak kötve a szomszédjaikkal. A kapcsolatokhoz társuló súly a backpropagation eljárással határozandó meg, innen a neve a BP-ANN-nek.

Amennyiben a bemenetek

$$x = x_1, x_2, \dots, x_n, \quad (2.4)$$

a súlyok a bemenettől a rejtett rétegegig $\phi_{i,j}$, a rejtett rétegtől a kimeneti rétegegig $\phi_{j,k}$, a kimenet pedig

$$y = y_1, y_2, \dots, y_n. \quad (2.5)$$

A neurális háló használata során (forward propagation) a bemeneti réteg és a rejtett réteg közötti kapcsolatot az alábbi formula adja meg:

$$q_j = \delta_1 \left(\sum_{i=1}^m \phi_{ji}^{(1)} + \phi_{j0}^{(1)} \right), \quad (2.6)$$

ahol a $q = q_1, q_2, \dots, q_n$ a rejtett rétegek értékei és a δ az aktivációs függvény. Általában a négyzetes hibafüggvényt szokták használni ilyen esetekben, mely az alábbi módon néz ki:

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2. \quad (2.7)$$

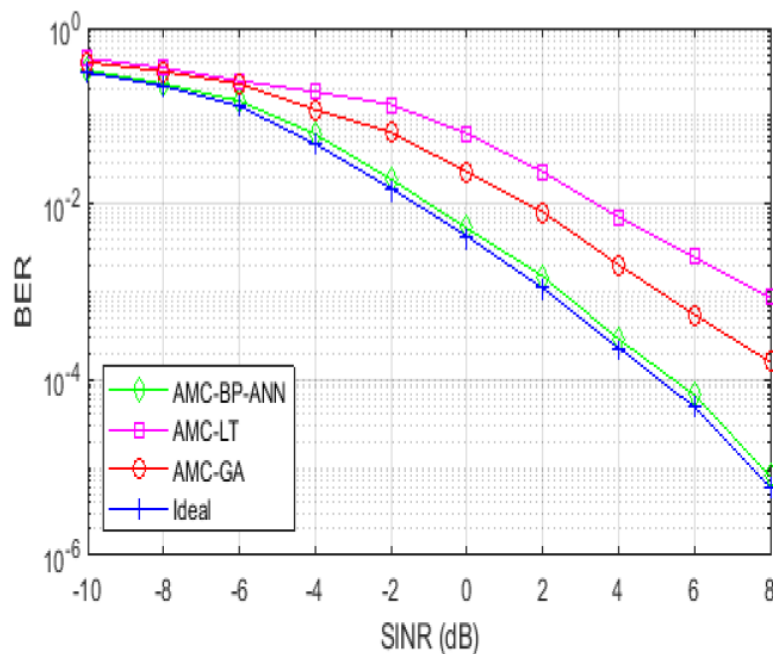
Sokszor aktivációs függvényként a szigmoid logisztikus görbét szokták alkalmazni, azonban van két hátránya ennek. Elsősorban ez a függvény könnyen szaturál, így a gradiens eltűnéséhez vezet. Másodsorban az asszimmetrikus intervalluma miatt a kimenete nem 0-közepű. Ezek nagyban nyomot hagynak a leszálló gradiens módszer (gradient descent) eredményében. A tangens hiperbolikus függvény is hasonló, mint a szigmoid, ám annak 0 közepű a kimenete. ReLU (Rectified Linear Unit) függvény egy lineáris aktivációs függvény, amelyet lineáris transzformációknál használhatunk. A gradient descent metódus során gyorsabban konvergálhatunk egy ilyen függvénnyel, mint a tangens hiperbolikusszal, vagy a szigmoiddal, ráadásul nem okoz vanishing gradient problémát. A fully connected feed forward hálózat ebben a dolgozatban ReLU függvényt használ aktivációs függvényként a rejtett rétegekben.

A hálózatba betáplált feature-ök a $H, \hat{H}, \Delta H, SINR, \omega_n^{typeI}, \omega_n^{typeII}$

$I_{total}, I_{mod3}, I_{mod4}, I_{mod30}, N^2, \hat{N}^2, \Delta N^2, BER$ voltak. A kimenet pedig egy SINR közelítés. Az eltemetett rétegek számától nagyban függ a megoldás. Nem létezik jelenleg sztenderd metódus. A dolgozat a Kolmogorov axiómát használja a rejtett rétegek számának meghatározására, ahol a node-ok száma v és a rétegek száma n .

$$v = 2n + 1 \quad (2.8)$$

A neurális háló a rejtett rétegeinek növelésével egyre kisebb hibát mutat. Ez a tendencia körülbelül 18 rejtett rétegnél megáll, így ezt a hálózatot teszteli a dolgozat.



2.8. ábra. Különböző ACM megoldások távolsága az ideális görbétől, BER függése az SINR től [25].

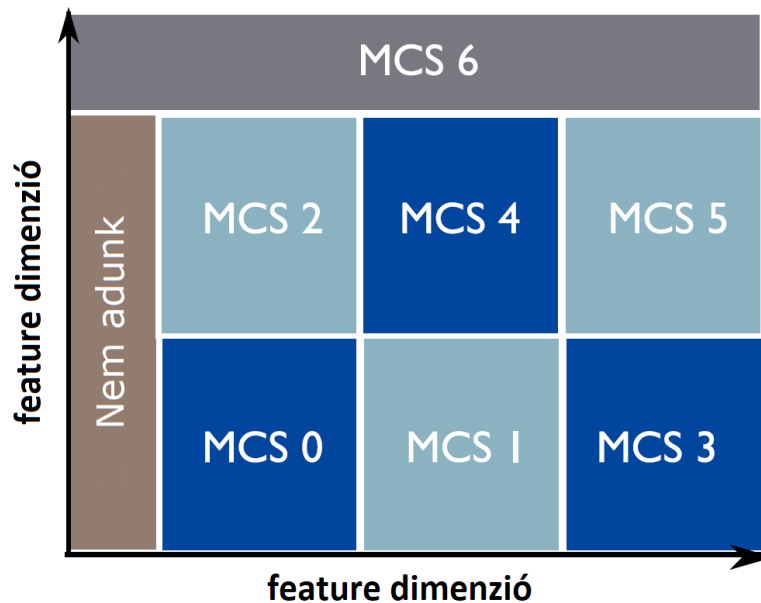
A BP-ANN rendszert egyéb megoldásokkal hasonlítja össze a 2.8-ábra. Itt megjelenik

a genetikus algoritmus, a keresési táblás megoldás is. A jel/zaj viszony függvényében az ideálisan legkevesebb bithiba arányt a BP-ANN metódus éri el. Ezt követi minőségben a genetikus algoritmus, majd a keresési táblás megoldás.

2.2.4. Online ACM Support Vector Machine (SVM) gépi tanulás-sal

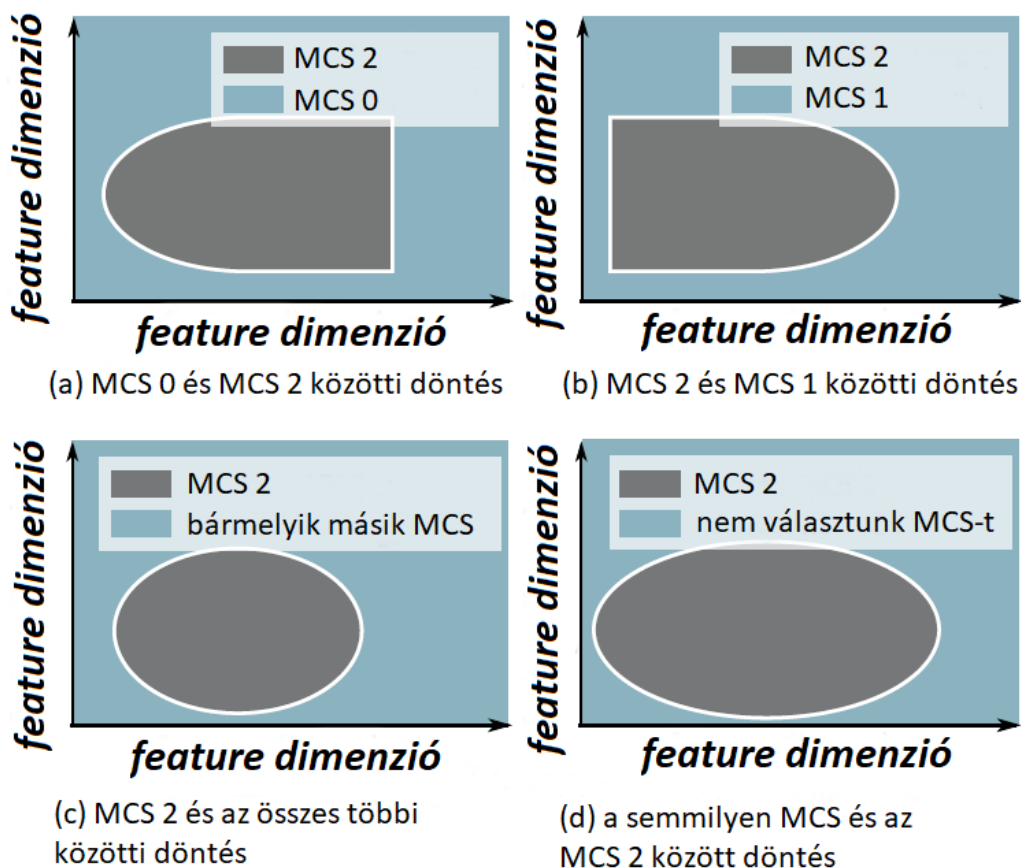
Ez a dolgozat [26] egy újszerű online SVM módszert használ, amely képes alkalmazkodni és optimalizálni az ACM-t az egyedi, potenciálisan dinamikus csatorna karakterisztikákhoz. Az online kifejezés jelentése ebben az esetben az, hogy a betanító adatok folyamatosan frissülnek valós időben, így a rendszer modell evolúciója nyomon követhető. A cél nem csak a megoldás, hanem a megoldás komplexitásának a mértéke is mérhető. A gyakorlatban az ACM egy look-up-table -ből találja meg az optimális kommunikációs paramétereket. Ennek a megoldásnak több hátulütője is van, amely miatt MIMO-OFDM környezetben nem megfelelő metodika. Az online algoritmus a gyors ACM-hez a keretek sikeres átvitelének mért indikációját használja fel (sikeres/hibás), valamint a rádiós csatorna minőségét megfogalmazó adatokat. A dolgozat említ egy már megvalósított gyors ACM klasszifikációs eljárást ugyanerre a problémára, mely Nearest Neighbor metodikával készült el. Ezt a megoldási módot gyakorlatilag limitálta a túlzott memóriaigénye. A dolgozat bemutatja, hogy SVM-el drasztikusan csökken a komplexitása a mesterséges intelligenciának a Nearest Neighbour módszerhez képest.

A vezeték nélküli hálózati modellben kereten alapuló átvitel történik az RX és a TX között. Minden egyes kerethez a TX kiválasztja a megfelelő paramétereket a fizikai jelalak generálásához. Ezeket a paramétereket jelen esetben egy változóval, a MCS-el le tudjuk írni.



2.9. ábra. Kétdimenziós feature sík, egymást kizáró osztályokkal [26]

A kétdimenziós feature térben, ahol a jelalak elállításához szükséges paraméterek egymást kizáró területeket alkotnak, klasszifikáció kell eszközölnünk. Tehát egy megfelelő MCS-t kell választanunk, hogy a keretünk a rádiócsatornán kívánt valószínűséggel áthaladjon. SVM technikával nehéz jó minőségű többdimenziós klasszifikációt eszközölni, ezért bináris klasszifikációra vezetjük vissza az MCS választást a megelőző adatok alapján.



2.10. ábra. (a) és (b) one-versus-one klasszifikációk a feature síkban, (c) one-versus-all és (d) one-versus-none klasszifikációkat ábrázolnak [26].

Többféle bináris klasszifikációra vezethető vissza a multidimenziós klasszifikációs döntés. Egyrészt visszavezethetjük például a 2.10-ábra a, szegmensének megfelelően, különböző MCS-ek közötti bináris klasszifikációra, ám ekkor sok bináris klasszifikációt kell végrehajtanunk a teljes feature tér lefedéséhez. Választhatjuk azt a megoldást is, hogy csak egy bizonyos MCS és bármelyik MCS között próbálunk választani. Megtehetjük, hogy egy olyan kategóriát hozunk létre, amely diszjunkt minden MCS-el (nem adunk), így választhatjuk azt a bináris klasszifikációt is. Tehát a semelyik MCS sem és a valamelyik MCS között döntünk. Ezt a 2.10-ábra d, szegmense szemlélteti. A dolgozat a semmilyen MCS és valamelyik MCS között bináris klasszifikációt használja. A többdimenziós klasszifikációt felbontja bináris klasszifikációkra, így az SVM algoritmus jobb minőségben képes elhatárolni két kategóriát egymástól, mint sok kategóriát egyszerre. Amennyiben \mathcal{M} a modulációs sémák száma, akkor

$$\mathcal{M} \frac{(\mathcal{M} - 1)}{2}, \quad (2.9)$$

bináris klasszifikációra bontható fel a multidimenziós klaszifikáció.

Ilyen típusú alkalmazásnál két függvény egymásba ágyazásával jutunk eredményhez a legmegbízhatóbban.

$$f(x) = g(h(x)); f : \mathbb{R}^p \rightarrow \mathbb{R} \rightarrow \{0, 1, \dots, \mathcal{M} - 1\} \quad (2.10)$$

Ahol $h(x)$ megállapítja a struktúra határait, hogy valós számra vetítse a régiókat a feature

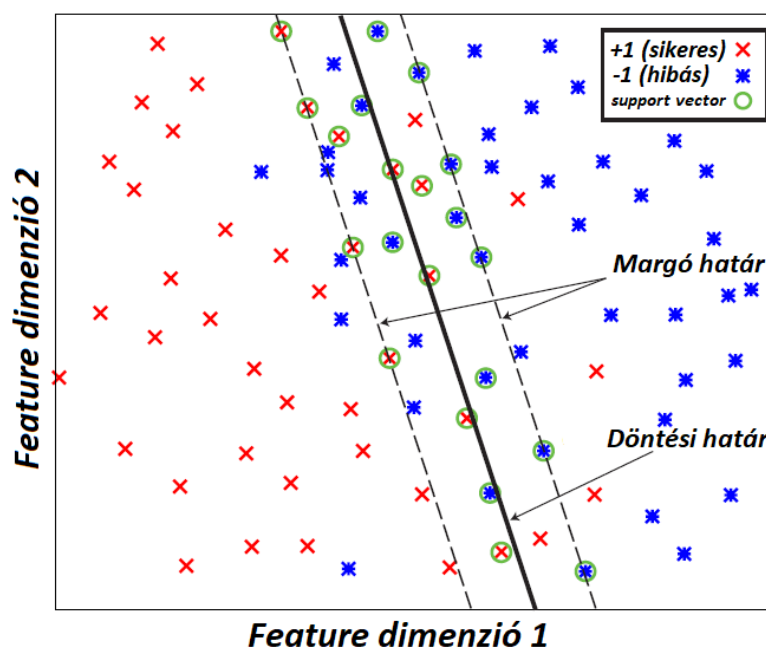
térben. Ez után $g(x)$ a valós számegegyenesbe való vetítést pozitív egész számokbeli vetítéssé dolgozza át.

Az ACM-t tehát SVM bináris klasszifikációval oldja meg a dolgot. Ez a módszer a mostani időkben azért kapott nagyobb figyelmet, mivel bonyolult nemlineáris csoport határok mentén is képes megfelelő eredményt hozni, hasonló módon mint a neurális hálók, azonban kisebb komplexitású modell szükséges a megoldáshoz. Ezen kívül az optimalizáció is könnyebb a neurális hálókhoz képest, ahol is túltanulás és interpolációs problémák léphetnek fel.

Az SVM képes megtalálni azt a vetítést, amely egy határvonal létrehozásával jól szeparálja térben a két csoportot. Ebben az esetben két csoport van, ahol is ezeket

$$y \in \{+1, -1\}, \quad (2.11)$$

címkekkel jelöljük.



2.11. ábra. Feature sík döntési határokkal [26].

Léteznek lineáris magfüggvények, amelyek az alábbiak szerint vannak definiálva:

$$K_{lin}(x_n, x_{n'}) := x_n^T x_{n'}. \quad (2.12)$$

Eszerint hipersík határokat tudunk létrehozni osztályok között. Polinomiális magfüggvényél:

$$K_{lin}(x_n, x_{n'}) := (x_n^T x_{n'})^d. \quad (2.13)$$

Végül radiális bázisú magfüggvénnyel

$$K_{rbf} := \exp(-\gamma \|x_n - x_{n'}\|^2), \quad \gamma > 0, \quad (2.14)$$

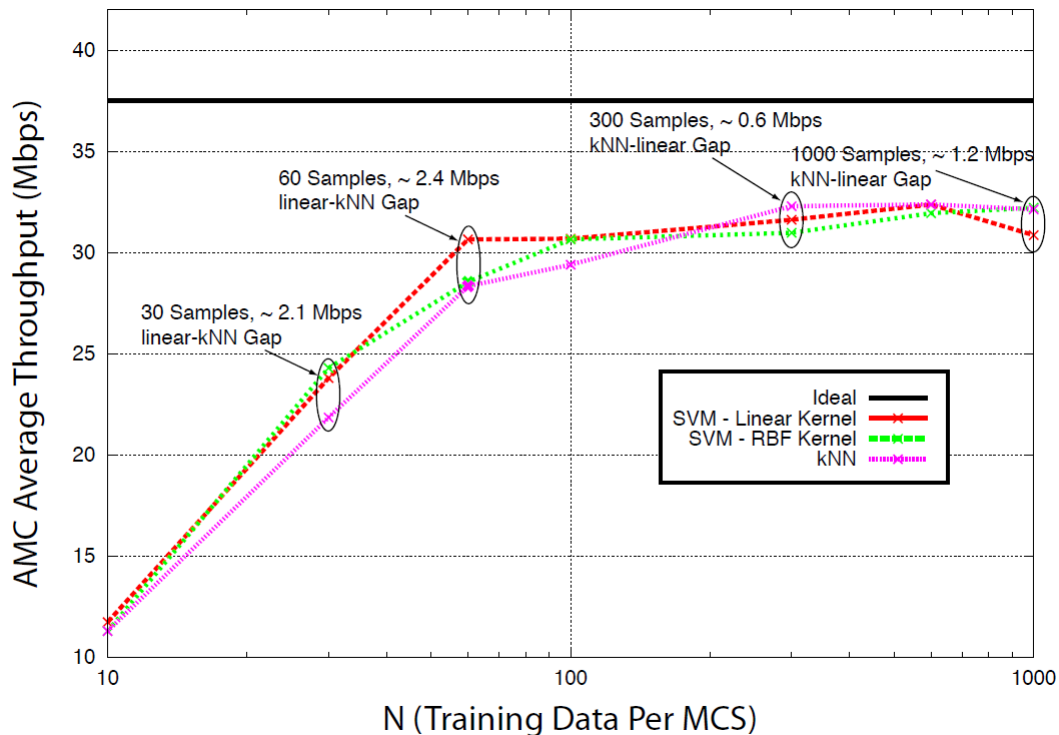
a fenti képlet definiálja a határokat exponenciálisan skálázott Euklideszi távolsággal.

A dolgot IEEE 802.11n szabványt használva, numerikus számításokat mutat be az alábbi paraméterű rádiós csatornán:

- 20 MHz es csatorna sávszélességek

- 2 vevő és 2 adó antenna
- MCS 0 -tól MCS 15 -ig lévő sémákkal (MIMO és SISO)
- 10 %-os FER megkötéssel ($F=0,1$)
- tökéletes RX szinkronizációt és csatorna becslést feltételezve

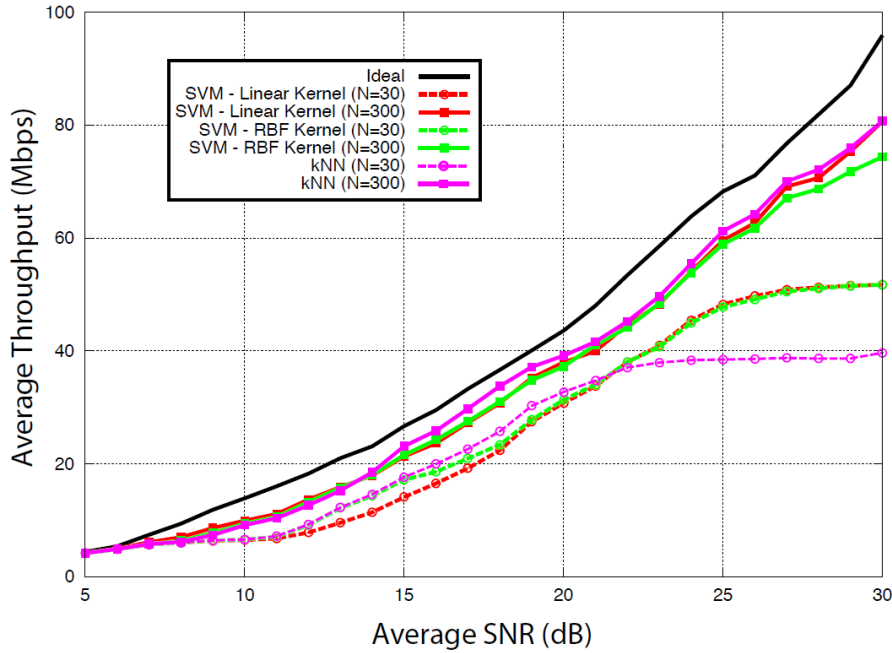
A csatorna szimulációt elvégezték SVM lineáris illetve radiális bázisú magfüggvénnyel is, valamint Nearest Neighbor klasszifikációval is.



2.12. ábra. ACM átlagos átviteli sebesség a betanítási adatok számossága függvényében [26].

Megfigyelhető a 2.12-ábrán, hogy a kNN algoritmus lassabban tanul a leglassabban. Mind a három megoldási módszer azonos közelségbe kerül az ideális átviteli sebességhez, azonban nem mindegy milyen komplexitású modellel vagyunk képesek ezt elérni. Az előzőekben említett kNN-el memória problémák vannak. A radiális bázisú magfüggvény szintén jó megoldás és azonos karakterisztikát mutat, azonban több betanító adatot igényel. Úgy tűnik, hogy nem éri meg feltétlenül a radiális bázisú függvénnyel foglalkozni, mert ez csak felesleges komplexitást visz a modellbe a lineáris magfüggvényhez képest.

Az 2.13-ábrán a növekvő jel-zaj viszony és az adatátviteli sebesség kapcsolata látható. Megfigyelhető, hogy $N = 30$ -nál egyik megoldási módszernél sem megfelelő a spektrális hatékonyság az SNR növekedtével. Ebben a helyzetben a kNN az SVM hez képest is alul teljesít. Újra megfigyelhető, hogy az SVM radiális és lineáris bázisú megoldások azonos karakterisztikával rendelkeznek és megfelelő spektrális hatékonyságot érnek el $N = 300$ -nál.

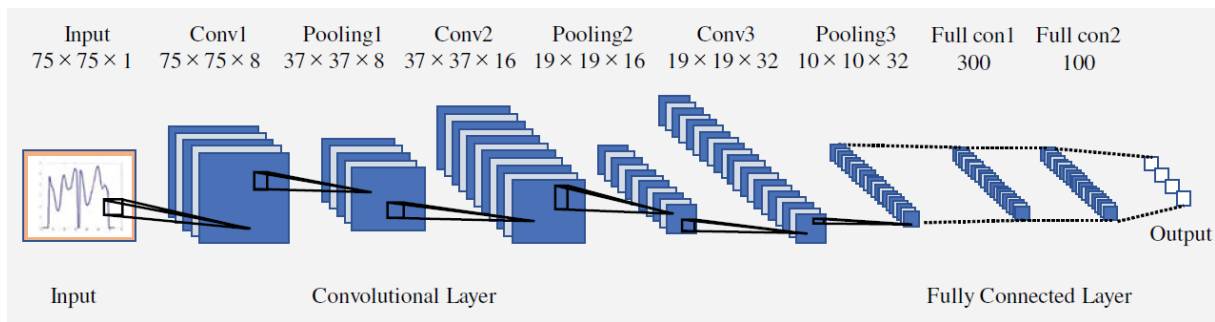


2.13. ábra. Átlagos átviteli sebesség az SNR függvényében, különböző paraméterű megoldásokra [26]

Összefoglalva a dolgozatban tárgyalt online ACM valós időben frissíti a betanító adatokat, megfigyelve, hogy milyen MCS, adott helyzetben sikeres keret átvitelhez vezetett-e vagy nem. A lineáris és a radiális bázisfüggvényű metódus illetve a kNN is megfelelő MCS választásokat eszközöl, azonban komplexitásban, betíntási időben, spektrális hatékonyságban a lineáris bázisú SVM a jó választás.

2.2.5. Precíziós SNR becslés konvolúciós neurális háló (CNN) segítségével spektrális teljesítménysűrűség (PSD) képek alapján

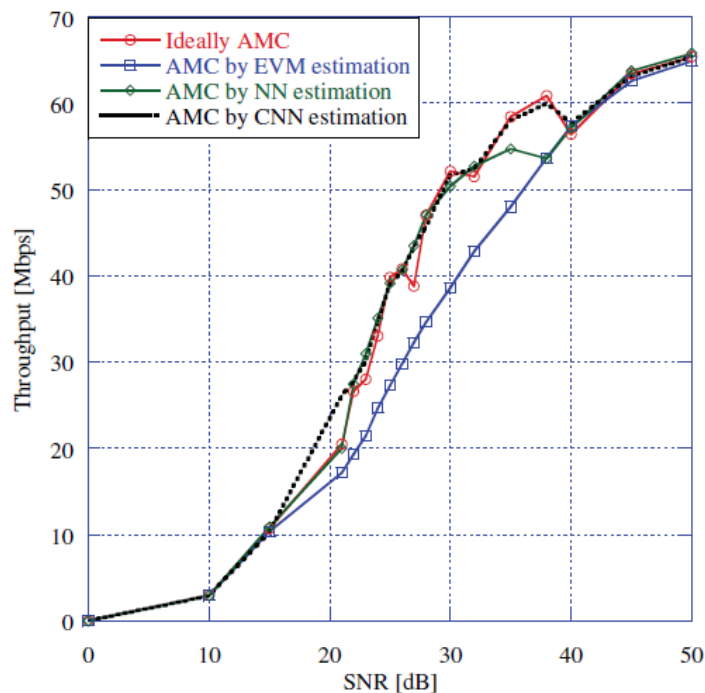
Ebben a dolgozatban [27] egy pontos SNR becslést végrehajtó rendszert mutattak be. A konvolúciós neurális hálót az időtarományból transzformált frekvencitarománybeli teljesítménysűrűség görbék képeivel tanították be. Egy egy képhez egy SNR tartomány tartozott, így a rendszer egy képen látható jelforma alapján egy klasszifikációt végez. Ez a megoldás előnyös magas mobilitást igénylő rendszerekben, mivel a PSD frekvenciarománybeli struktúrájára kevésbé hat a Doppler eltolódás, mint egy időtarománybeli megoldásban. A CNN sokkal részletesebb feature-öket megtalálni az adatokban mint a neurális háló, a bemeneti kép feature-jeinek a konvolválásával.



2.14. ábra. Konvolúciós neurális háló rétegei [27]

A fenti ábrán (2.14) egy általános konvolúciós háló megoldás struktúrája látható. A bemeneti képet konvolúciós és max-Pooling réteg párosokon keresztül vezetve, melyek között ReLU aktivációs függvények vannak, fel tudjuk fedezni az adat magasabb szintű tulajdonságait. Ezek után teljesen összekötött perceptronok vezetnek a kimeneti rétegre.

OFDM csatornára alkalmazza a dolgozat az ACM metódust. A szimulációt 20 MHz es sávszélességű csatornán végzi, ahol 15 jelutat feltételezünk Rayleigh fading szerint, valamint 3 modulációs sémát (QPSK, 16QAM, 64QAM) és 4 konvolúciós kódarányt (1/2, 2/3, 3/4, 5/6) különböztetünk meg. A betanítás során sztohasztikus gradiens leereszkedést (SGD) használtak. Minden SNR tartományhoz 5000 spektrális kép tartozott. 900000 spektrális kép generálódott a szimulációk során, ennek a 80 %-át használták betanításra, a többit pedig tesztelésre.



2.15. ábra. Különböző ACM metódusok összehasonlítása [27]

A 2.15-ábra az SNR függvényében ábrázolja az átviteli sebességet. Az EVM ezen az ábrán Error Vector Amplitude közelítést ábrázol. Az ideális görbét a CNN és az NN is megközelíti, azonban a CNN a 30-40 dB-es jel/zaj viszony környékén való kódarány váltásoknál jobban szerepel.

3. fejezet

Predikció, bináris klasszifikációra, LSTM visszacsatolt neurális hálóval

Ebben a fejezetben be fogom mutatni a munkámat az adatok feldolgozásától a jelölésen keresztül a betanításig és a tesztelésig. A fejezet végén szó fog esni a neurális háló optimalizációjáról LSTM rejtett réteg szélesség és mozgóablak szélesség tekintetében.

3.1. Adatok előkészítése

A mesterséges intelligencia alkalmazásánál általában a legtöbb időt az adatfeldolgozás szokta elvinni. Fontos olyan tiszta környezetbe helyezni a betanítási folyamatot, ahol a hibák vagy pontatlanságok nem perturbálják az adatokban rejlő struktúra felismerhetőségét. A nyers adatokból ki kell venni az inkonklúzív jelenségeket és hibákat, majd jelen esetben össze kell illeszteni az egyes hónapok adatait, hogy egy teljes összefüggő adathalmazt kapjunk időtartományban.

3.1.1. Adatok struktúrája

Az Alphasat földi vevőállomását vezérlő és az adatokat rögzítő számítógép gyűjti és tárolja a mérési adatokat egy szövegfileban. Minden naphoz külön file generálódik, ahol minden mérés időbélyeggel kerül rögzítésre. Az egyes napok adatai txt fájlokban adóttak, melyeket össze kell fűzni. A txt fájlokban az alábbi sorok szerepelnek:

048.70; 171.9129; 37.4625; 1

048.74; 171.9129; 37.4625; 2

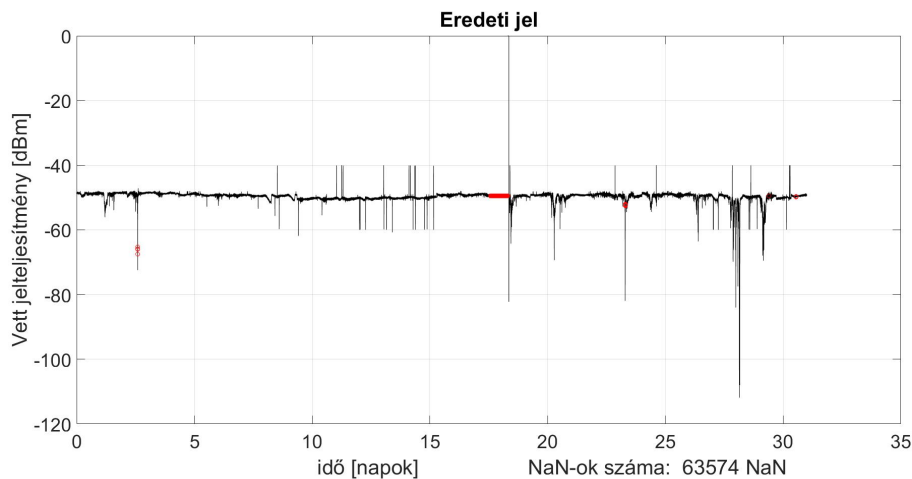
...

052.53; 171.8821; 37.4540; 86400

A betanítás szempontjából az első oszlop érdekes, mely a vett jelerősség szintet tartalmazza két tizedesjegy pontossággal dBm-ben. A második és a harmadik oszlopból kiolvashatóak a műhold aktuális azimuth és az eleváció szögei. A negyedik oszlopban pedig egy jelölés jelenik meg, mely a nap eltelt másodperceinek felel meg.

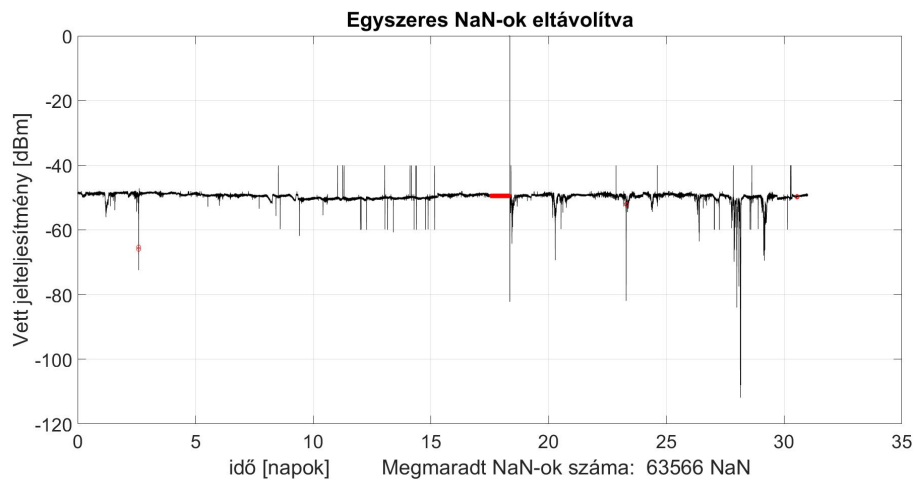
3.1.2. Adatok szűrése

Az adatokban előfordulnak több tízezer hosszúságú nem szám értékű intervallumok, ezek a mérés kiesése miatt vannak benne a jelszint adatokban. Az előfeldolgozás során, amely MATLAB-al történik, a kieső adatokat NaN-al jelöljük. Vannak csupán egy másodperces NaN intervallumok is. Minden nem szám értéket el kell távolítani a jeltől. Ez történhet interpolálással vagy törléssel, vagy egy konstans érték beállításával. Én egyszeres NaN-okra interpolációt használtam, mely a szomszédos két értékből meghatározza, hogy milyennek kéne lennie a középső értéknek. A többszörös NaN-okat egy konstans érték beszúrásával elimináltam temporálisan. A konstans értéket a jel NaN intervallumának végéről vettem. Ez nem a jel helyettesítését szolgálja, mivel a neurális háló lehet, hogy egy konstans szinttel nem tudna boldogulni, ha előtte tiszta ég mellett is csak hullámozó értékeket látott volna. Ezért ezek a többszörös NaN-ok helyetti konstans értékek jelölésül szolgálnak, hogy az adatok előkészítésének a törlés fázisában eltávolítsam őket, mert egy konstans érték jól észrevehető a zajjal terhelt jelben. A következőekben végig fogok menni egy hónapon, melyet egy MATLAB szkripttel feldolgoztam a NaN-ok szempontjából.



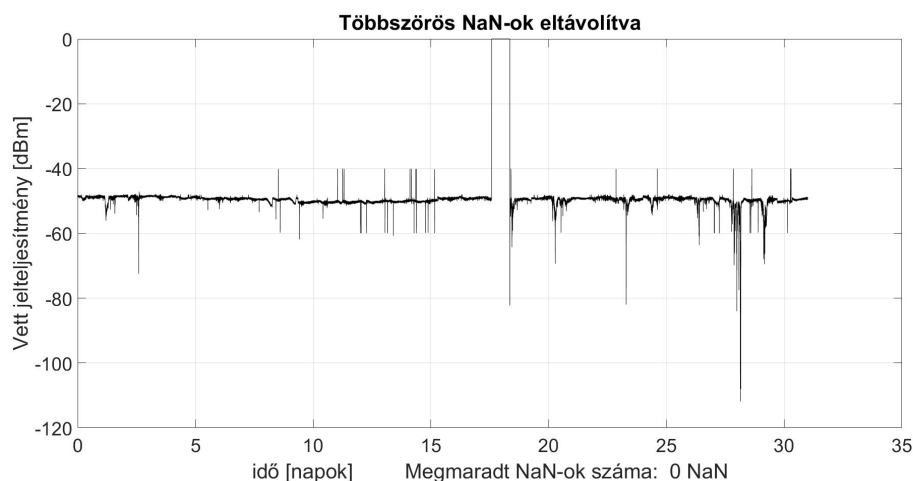
3.1. ábra. 2018 októbere, a piros jelölés a NaN-ok helye

Első lépésben a 3.1-ábrát rajzoltam ki, ahol megjelöltem az időtengelyen a NaN értékeket és összeszámoltam őket. Jelen esetben egy hosszabb intervallum miatt van ennyi mérési eredmények nélküli hely. Ezeket a helyeket egy piros körrel jelöltem és a NaN-ok környezetétől függően próbáltam kitalálni, hogy milyen teljesítményt vettünk volna ebben a pillanatban. Ezért rendelkeznek a NaN helyek vett teljesítménybeli értékekkel.



3.2. ábra. Egyszeres NaN-ok eltávolítva, a többszörös NaN-ok megjelölve

A 3.2-ábra jobb és bal oldalán látszik, hogy néhány egyszeres hiba kiinterpolálódott. Ezek után már csak azok az intervallumok maradtak, amelyek egynél több összefüggő NaN-okból álló intervallumot képeznek.

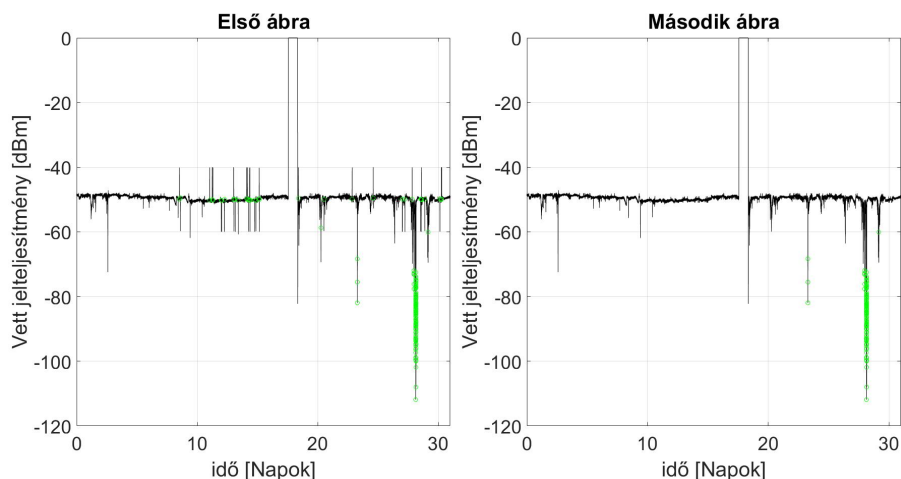


3.3. ábra

A többszörös NaN intervallumokat jobb oldalról vett konstans értékkel helyettesítve meg számoljuk a NaN-ok számát az adatsorban. Látszik, hogy a két szkript eliminálta a hiányos helyeket a jelben. A kisebb összefüggő hiányos intervallumok is jól látszanak közelről nézve a törlés fázisban, ezért csak nagyon rövid szegmensek maradtak valóban konstans értékkel helyettesítve.

3.1.3. Mérési hibák interpolálása

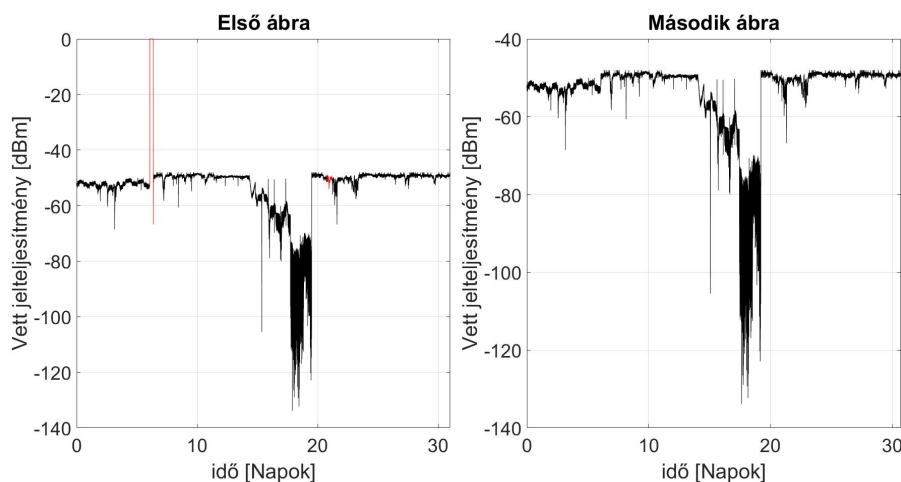
Úgy találtam, hogy a hirtelen ugrásokat egy pontról pontra való haladás során való különbség számításával ki lehet mutatni. Ahol az előző ponthoz képest a következő pontban 5 dBm-el nagyobb vagy kisebb érték volt, megjelöltem. Mivel a fading események is létrehozhatják ezeket a gyors változásokat, amit a 3.4-es ábrán is láthatunk, ezért nem lehet minden ilyen ugrást kiinterpolálni.



3.4. ábra. 2018 októbere

Az volt a tapasztalatom, hogy az 5 dBm-es beállítással a jel alapszintjéhez közel ki lehet interpolálni az ugrásokat, mert ezek a hibáknak feleltek meg. Ahol a fading eseményben szerepet játszó ugrásokat távolítottam volna el a jelből ott ezt a módszert helyettesítettem egy felülről szűréssel, ahol a jelben csak a pozitív ugrásokat interpoláltam ki. 2018 októberében úgy véltem, hogy nem okoz adatvesztést a pozitív és a negatív irányban való szűréssel való interpolálás, ezért ezt használtam. Az első ábrán látszik, hogy minden hirtelen ugrás megjelöltem, majd a második ábrán ezeket az ugrásokat interpoláltam, de csak azokat, amelyek a jel alapszintjéhez közel állnak.

3.1.4. Nem kívánt intervallumok törlése a jelből

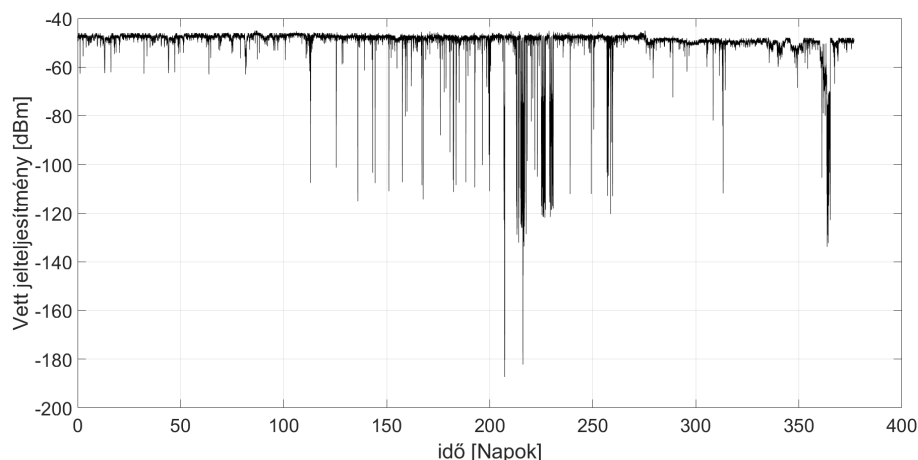


3.5. ábra. 2018 decembere

A 3.5-ös első ábráján látszik a már mérési hibáktól megszűrt jel. Ezen a jelen pirossal jelöltem meg azokat a részeket amelyeket nem kívánnék a betanító adatok közé berakni. A nem egyértelmű kisebb jelölések közelről nézve a többszörös hiba intervallumokból adódó konstans értékek miatt voltak megjelölve. A második ábrán ezek az értékek törölve lettek a hónapból.

3.1.5. Feldolgozott hónapok összeillesztése

A fenti metodikát követve szűrtem meg a 2018 és 2019 hónapjait. A 2018-as évet összeillesztve az alábbi adatsort kaptam.

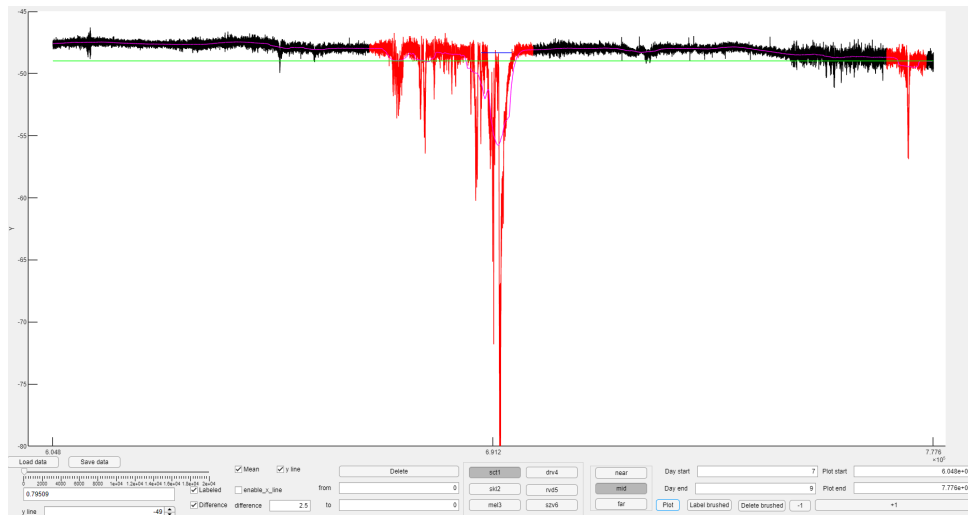


3.6. ábra. Szűrt adatsor a 2018-as évre

2019 éve kevesebb mérési hibát tartalmaz, viszont 2018-már több napos intervallumokat tartalmaz, amelyeket ki kellett vennem, hogy a hálózat tiszta körülmények között tanulhasson. A legtisztább körülmények teremtése végett két évből csupán 468 nap maradt meg.

3.2. Adatkezelés

Az adatok címkézésére egy MATLAB app-ot fejlesztettem, mellyel könnyedén megjelölhetem a fading eseményeket. Az események megjelölése során az a tapasztalatom, hogy a jelölés egyértelműsége végett szabályokat kell beiktatni. A jelölés során azonos nézetet kell használni, hogy az emberi hiba elkerülhető legyen, mivel határesetben a nézet is módosíthatja azt hogy az ember fading eseménynek bélyegzi az intervallumot vagy még nem. Szabályokat kell szabni jeljeljesítményben is, hogy mikortól számít fadingnek az intervallum és ezt érdemes megjelölni a címkéző számára a jelenlegi jelszinthez képest.



3.7. ábra. MATLAB applikáció fading jelölésre.

A 3.7-ábrán látható app a jelölést hivatott segíteni, segédvonalakat nyújtva a jelölő számára, hogy a jelölés egyértelmű és konzekvens legyen. Az app-ban magentával van jelölve a jel mozgóátlaga, zölddel van jelölve a mozgóátlaghoz képest 2 dBm-el kisebb konstans érték és kékkel van jelölve a mozgóátlag eltéréseiből számított jelcsökkenési intervallum. Ezek mind segítséget hivatottak nyújtani a jelöléshez.

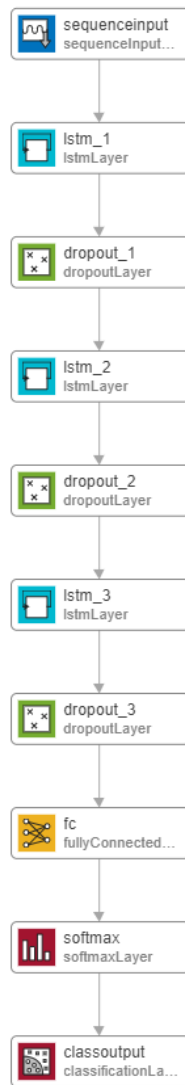
Bár a jelölés egyszerűbb ebben a programba, így is megfigyelhetőek a kész, címkézett adatsorban hibák, melyeken még javítani kell.

A program fel van készítve a több osztályú jelölésre is, nem csak bináris klasszifikációra. Végigjelöltem a 2 éves adatsor úgy, hogy 6 osztályba soroltam jelalakokat és úgy is hogy csak bináris osztályokba soroltam a jelenségeket. Az osztályok a következők voltak többosztályú jelölésnél: szcintilláció, sekély fading, mély fading, nagyon hirtelen és rövid fading, nagy jelváltozással járó hosszú zavar (hiba) és szintugrás. A nem bináris klasszifikációs feladatra a jelölés inkonzekvenciája miatt nem tanult be jól a neurális háló. Valószínű, hogy regressziós problémaként jobb konzekvenciával jobb eredményt kaptam volna, de ezután a próbálkozás után maradtam a bináris klasszifikáció mellett, ahol csupán fading és nem fading osztályok léteznek.

3.3. Legjobb eredményt elérő neurális háló

Mind az adatok feldolgozására, mind a betanításra MATLAB programot használtam. A neurális háló szintéziséhez és betanító algorimus meghívásához a Deep Learning Toolbox 14.1 -es verzióját használtam.

Az adatok másodperces mintavételezésben adottak, viszont az ilyen sűrűségű mintavételezésből származó adathalmaz meghaladná az alkalmazott számítógép memóriakapacitását, ezért ezt perces mintavételezésre állítottam. Ezzel a módosítással is képes a hálózat jó pontosságot elérni, ezért ezzel dolgoztam. Az alábbi ábrán látható az általam használt neurális háló felépítése.



3.8. ábra. Legjobb pontosságot elérő neurális háló struktúrája.

Sok kísérletezéssel arra jutottam, hogy több egyirányú LSTM réteget használjak, ne csak egyet. Bár jobb megoldást hoz a több réteg, ám a hálózat komplexitásának növelése nem feltétlenül éri meg az általa nyert pontosságot, ettől függetlenül, jelen esetben a legjobb pontosságot nyújtó neurális hálót prezentálok. Ezek után az LSTM rétegek után Dropout rétegeket tettem, mivel ez is javított a klasszifikációs pontosságon. A Dropout réteg egy meghatározott valószínűséggel eldob kapcsolatokat a neurális hálóban. Azok között a kapcsolatok között sorsol eldobandó vonalakat, amelyek között a Dropout réteg tartózkodik. Ezzel a módszerrel elkerülhetjük, hogy bizonyos kapcsolatok túl dominánssá váljanak és elnyomják a többi kapcsolatot, így általánosabb megoldás felé visz minket a betanítás folyamata. Általánosan 30-50%-os valószínűséget szoktak paraméternek állítani a Dropout rétegnél. Ennél a problémánál a 60%-os kapcsolat eldobási valószínűség vált be. Az egymást követő LSTM és Dropout párok után egy Fully Connected réteg kapott helyet. Ez a réteg összefogja az LSTM rétegektől érkező információt, úgy hogy már a bináris klasszifikációnak megfelelő vonalakat kapjuk. Ezért is 2 a Fully Connected rétegnek a bemenő paramétere, ami azt jelenti, hogy az utolsó LSTM réteg a Dropout-on keresztül két, a 1.7-ben bemutatott perceptron-hoz kapcsolódik. Ez a két perceptron egy Softmax réteggel áll kapcsolatban, amely az osztályok számának megfelelő számú kimenettel rendelkezik. Bináris klasszifikáció esetén 2-vel. Ezen a két kimenetén a különböző

osztályokhoz tartozó valószínűséget állít elő, tehát a két bemenet értékének az összege mindig 1 lesz. A hálózat Softmax rétege a választ valószínűségben fogalmazza meg. Ezt a valószínűséget kell egy olyan rétegen átengedni amely egy határozott választ ad. Ez a réteg a Classification Output réteg. Ez lenne hivatott a valószínűségből eldönteni, hogy fading vagy nem fading esemény fog a következő percben történni. Mivel a predictAndUpdateState függvényt használtam, ezért csak a különböző csoportok valószínűségét kaptam meg predikcióként, tehát a Softmax réteg eredményeit a Classification Output réteg nem alakította egy meghatározott szint szerint fading vagy nem fading osztállyá. A Classification Output rétegen megjelenik a fading és a nem fading következő percre vetített valószínűsége, ezt a MATLAB classify parancsával lehetett volna osztály predikcióvá tenni. Mivel a predictAndUpdateState függvényt használtam, ezért magam vetítettem le az osztály valószínűségeket csoportosításra egy, határ szabásával. Ennek az a hátránya, hogy nem fogom tudni úgy beállítani a kimenet osztályozásának a tulajdonságait ahogy azt a sok adat alapján az optimalizáló beállítaná. Az előnye az, hogy mivel a valószínűség van a kezemben, mint kimenet, ezért állítani tudom a határ által, hogy már kisebb valószínűségnél is jelezzen fading-et. Ezzel a hálózat által betanult megoldást ki tudom élezni a fading jelenség minél hamarabbi felismerésére. Amennyiben kisebb fading valószínűségeknél is úgy állítom be a határt, hogy az már fading-nek minősüljön, akkor viszont több nem fading jelenséget fog a hibásan prediktálni a hálózat. Tehát a predictAndUpdateState függvény használata egy plusz szabadsági fokot ad a kezembe, azonban a predikció mihamarabbi felismerése rosszab fading predikciókat hoz átlagban, így fordítva is igaz, hogy jobb átlagos fading predikciónál később tudjuk felismerni a fading jelenséget.

Name	Type	Activations	Learnables
sequenceinput Sequence input with 4 dimensions	Sequence Input	4	-
lstm_1 LSTM with 100 hidden units	LSTM	100	InputWeights 400×4 RecurrentWe... 400×... Bias 400×1
dropout_1 60% dropout	Dropout	100	-
lstm_2 LSTM with 100 hidden units	LSTM	100	InputWeights 400×... RecurrentWe... 400×... Bias 400×1
dropout_2 60% dropout	Dropout	100	-
lstm_3 LSTM with 100 hidden units	LSTM	100	InputWeights 400×... RecurrentWe... 400×... Bias 400×1
dropout_3 60% dropout	Dropout	100	-
fc 2 fully connected layer	Fully Connected	2	Weights 2×100 Bias 2×1
softmax softmax	Softmax	2	-
classoutput crossentropyex with classes 'nem fading' and 'fading'	Classification Output	-	-

3.9. ábra. Neurális hálózat struktúrája, aktivációs függvényeinek és betanulandó értékeinek a száma.

A 3.9-ábrán szintén látszik a neurális hálózat felépítése, valamint megfigyelhető a Learnables oszlopban, hogy milyen súlyokat és előfeszítés értékeket kell beállítani, hogy a globális minimumhoz elérkezzünk, amennyiben ezt a hálózat felépítése lehetővé teszi. A

hálózat felépítése megfelelő a nemlineáris függvény közelítésére, mivel a pontossága elfogadható.

A függelékben megfigyelhető 5.1-es kódblokk felelős a neurális háló rétegeit összeilleszteni egymás után, valamint ezeket a rétegeket felparaméterezni. Megfigyelhető, hogy 4 bemenete van a hálózatnak. A bemenete a sztenderdizált eredeti jel és a három feature. Az LSTM rétegek argumentuma, hogy hány Hidden Unit-tal realizáljuk őket. A Hidden Unit-ok száma megfeleltethető az LSTM cella emlékező képességének. 100 Hidden Unit-tal a cella, az előző 100 bemenetéhez tartozó információkat képes tárolni. A perces mintavételezés miatt ez elegendő lehet.

Az alábbi felépítésű neurális hálókat próbáltam ki, melyek után a bemutatott eredményre jutottam, ahol is a klasszifikációs pontosság mentén döntöttem el mi számít jó vagy jóvv eredménynek.

- 1-2-3-4-5 db LSTM- Dropout pár egymás után. A 4 vagy 5 db nem hozott jobb eredményt, viszont a 3 db jobb volt, mint az 1 vagy a 2.
- A Dropout réteg beállítása 30-40-50 és 60% között. A 60 % os Dropout-ok hozták a legjobb pontosságot.
- A 3 db LSTM- Dropout pár mellett a Hidden Unitok számát is kipróbáltam különböző állásokban. Kipróbáltam úgy, hogy az összes Hidden Unit szám 25, 50, 75 100, 150, 300, 500. A 25 és az 50 nem volt megfelelő. A 300, 500 túl sokáig futott és nem is kecsegtetett túl jó eredménnyel. Maradt a 75 és 150 közötti szám. Innen is a 100-as volt a legjobb. A 300 és az 500 vagy akár 1000 egy gyorsabb géppel akár hozhatott volna jó eredményt, de én csak maximum 30 epoch-ig vizsgáltam. Próbáltam még a Hidden Unit-ok számának a változtatását úgy is, hogy a bemenetközeli LSTM például 300 és sorban 150 és 75 értéket kaptak a többi cellák, valamint fordítva is, hogy 75 el kezdtem. Ezek sem vezettek jobb klasszifikációs pontossághoz. Maradt a homogén 100 Hidden Unit szám.

3.3.1. Inicializálás

Elsőnek alaphelyzetbe állítjuk a MATLAB véletlenszám generátorát.

```
1 rng default;
```

A véletlenszám generátor egy pseudo-random generátor, mely az idő előrehaladtával változik. Állítsuk be a kezdeti értékre. Ennek a generátornak a segítségével állítjuk elő a belépő neurális háló belső értékeit. Mivel a neurális háló betanítás során változtatható értékei véletlenszerűen állnak a kezdeti pillanatban, ezért enyhén változhat a betanítás eredménye. Például a súlyok bizonyos állása mellett csak egy lokális minimumot tudhat elérni a betanító algoritmus.

A másodperces mintavételezést az adatok betöltése után átváltom perces mintavételezéssé az alábbi kóddal.

```
1 x_m=x(1:60:end);  
2 y_m=y(1:60:end);
```

A továbbiakban az `_m` jelölés a perces mintavételezésű adatokra utal.

3.3.2. Adatok jellemvonásai (Feature-ök)

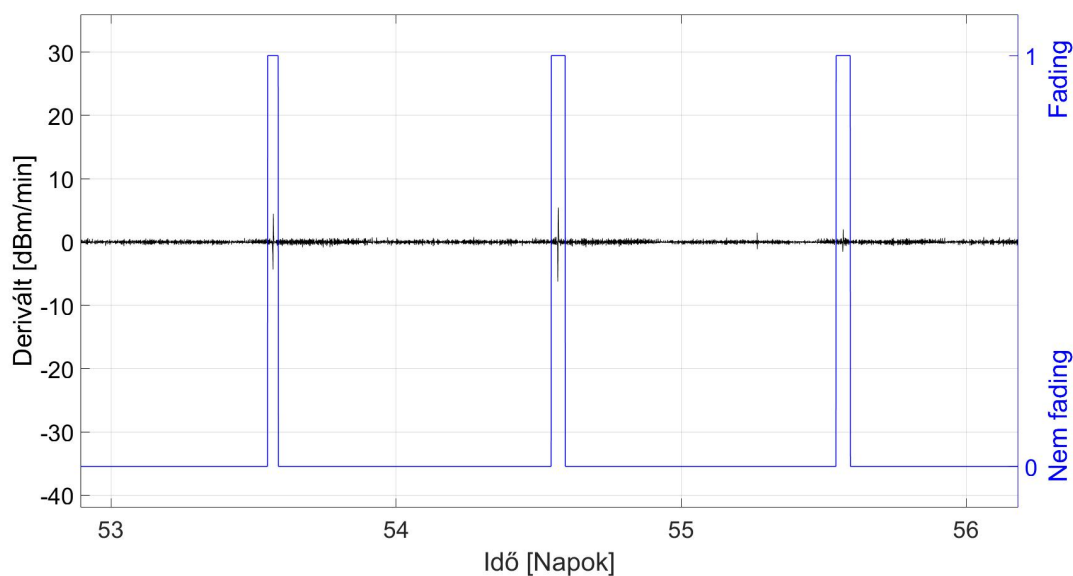
A nyers adatok betáplálása a neurális hálóba általában nem vezet kellően jó megoldáshoz, vagy csak nagyon sok adat alapján vezethet, ezért a jel különböző tulajdonságait adjuk a hálózat bemenetére. Olyan adatokat kell generálni amelyek tükrözik a jel egy fontos tulajdonságát. Ez az adat legyen kellően korrelált a meghatározandó osztályokkal és legyen azonos hosszúságú vektor, mint a nyers adatok. A neurális hálózat bemenetére 4 különböző adat kerül. Ezek közül az új feature-ök az utolsó három.

- Az eredeti jelteljesítmény
- Derivált
- Mozgóátlag
- Mozgóablakos tapasztalati szórás

A függelékben látható 5.2-es MATLAB kódblokk felelős a feature-ök generálásáért. A feature-öket kipróbáltam külön-külön is beadni a neurális hálózat bemenetére. Külön körülbelül 60%-os pontossághoz vezettek, együtt pedig 90%-os pontosság volt a jellemző a betanító adatokra.

Derivált

A derivált, mint meredekség adja magát, hogy ha a jelteljesítmény függvényében a negatív irányú változásokat szeretnénk kimutatni, jó tulajdonsága lehet a nyers adatnak. Maga a derivált az alábbi módon néz ki, perc alapú, idő szerint deriválva a jelteljesítményt (3.10-ábra).



3.10. ábra. Jelteljesítmény idő szerinti deriváltja.

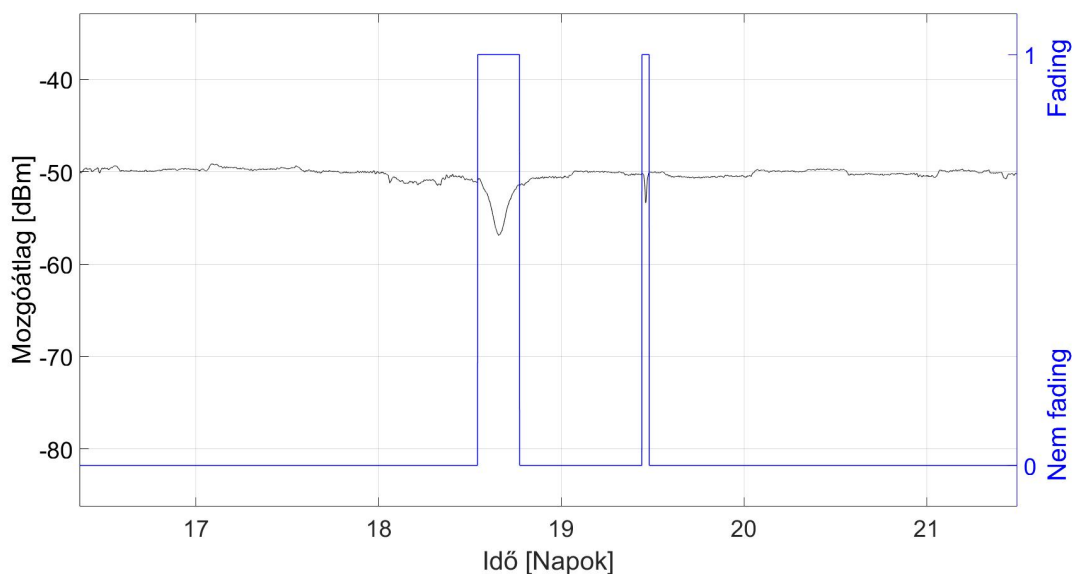
Mozgóátlag

Mozgóátlagot movmean paranccsal tudunk létrehozni, melynek egyik argumentuma, hogy minek készítsük el a mozgóátlagát és a másik argumentuma, hogy milyen ablakot vegyünk.

Általában mozgóátlag a kiszámolandó pontra szimmetrikus ablakkal megy végig a jelen. Mivel mi jelen esetben az architektúra predikciós képességét szeretnénk felhasználni, ezért ezt nem tehetjük meg. Ennek az az oka, hogy ha olyan feature-t készítünk, amely jelen értéke összefüggésben van a jövőbeli időpillanattal, akkor a feature jósol, amely nagy mértékben növeli a hálózat predikciós képességeit, úgy hogy ez tulajdonképpen hamis predikciós képesség. Felhasználva a hálózatot egy olyan rendszert kapnánk, ahol előre be kell táplálni a jövőbeli értékeket, hogy megjósolja a jövőbeli értékeket ami rossz. Bár a szimmetrikus mozgóablak hamis predikciós képességeket ad a hálózatnak, de ezt fel lehet használni egy másik alkalmazásra. Mivel a mai napig kézzel végzik a fading események jelölését, ezért segítséget nyújthat egy hálózat, amely képes felismerni és megjelölni ezeket. Ha valós alkalmazásban az adatok nem időről időre jönnek sorban, hanem adott a teljes adathalmaz, akkor egy ilyen szimmetrikus mozgóablakkal felruházott bementettel a neurális hálót be lehet tanítani, a meghatározandó jelalakok felismerésére. A dolgozat a predikciós képességeket vizsgálja, ezért ennek megfelelően asszimmetrikus mozgóablakkal végezzük az átlagolást.

```
1 movmean(x_m', [10 0]);
```

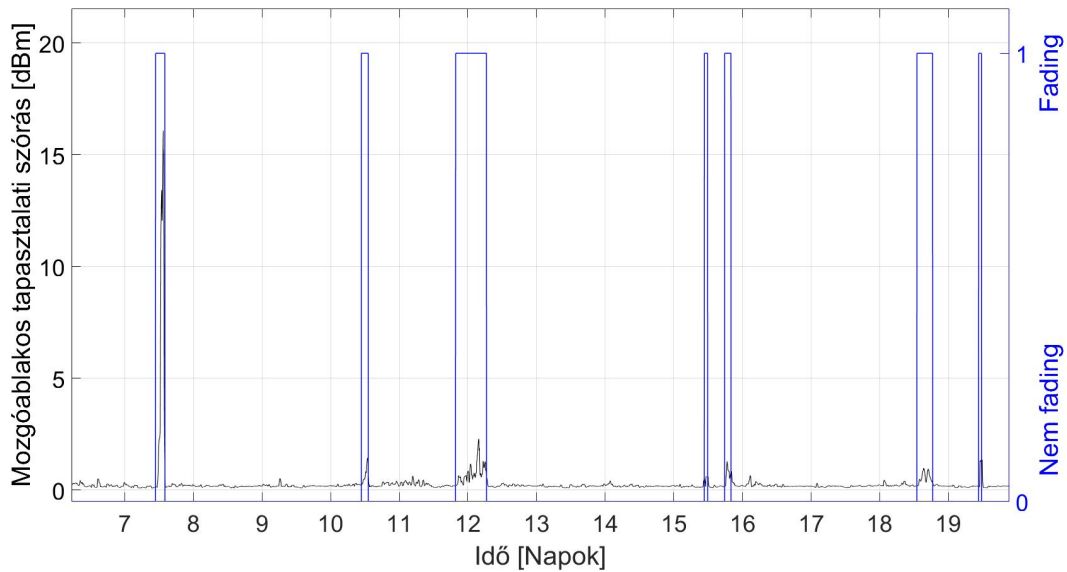
A [10 0] beállítás azt jelenti, hogy a jelen pillanatban vizsgált ponthoz képest 10-el korábbi és 0-val későbbi értéket veszünk az adathalmazból és ezeket átlagoljuk. Ehhez a konkrét beállításhoz úgy jutottam, hogy kirajzoltam a generált feature-t mellette az elvárt osztállyal. Ez látszik a 3.11-ábrán. A beállítások mellett betanítottam a hálózatot, majd finomítottam a mozgóátlagképzés beállításán.



3.11. ábra. Jelteljesítmény mozgóablakos átlaga.

Mozgóablakos tapasztalati szórás

A mozgóátlaghoz hasonló a mozgóablakos tapasztalati szórás. Itt a kirajzolással állapítottam meg, hogy milyen relációban van az elvárt kimenet és a generált feature.



3.12. ábra. Jelteljesítmény mozgóablakos tapasztalati szórása.

```
1 movstd(x_m', [50 0]);
```

A fenti kódban az $[50\ 0]$ jelentése, hogy a jelen pillanathoz képest 50 előző és 0 későbbi mérési eredményből számítunk tapasztalati szórást. A 3.12-ábrán megfigyelhető, hogy elég tisztán együtt jár a feature és az elvárt kimenet.

3.3.3. Adathalmaz felosztása

Szükségünk van három elkülönített adatsorra. Ezek a Train, Validation és a Test adatok. Az eredeti adathalmazt háromfelé osztva az alábbi kóddal juthatunk ezekhez hozzá.

```
1 [train , test , validation] = divideblock(data , .70 , .15 , .15);
```

A kód jelentése az, hogy a hosszú, már feature-öket is tartalmazó vektorunkat 70%, 15% és szintén 15%-os arányban sorban felszeleteljük és elkülönítjük. Így a vektor elejének nagy része a betanító adat lesz, az utána következő 15% teszt adat és a maradék pedig validációs adat lesz. Azért jön előbb a teszt adat, mert az adathalmaz utolsó szelete nem olyan szép. Validációs adatnak megteszi, hisz ugyanúgy hasonló jelleget öltenek benne a jelenségek, viszont kevésbé lehet megítélni a hálózat viselkedését vele.

3.3.4. Sztenderdizálás

A neurális hálóban nagy számban találhatóak aktivációs függvények. Ezek jelen esetben többnyire Sigmoid és Tangens-hiperbolikus függvények. Közös tulajdonságuk ezeknek, hogy a kimenetük egy elég szűk tartományon mozog. 0 és 1 vagy -1 és 1 között. Érdeemes a hálózat bemenetére olyan adatokat adni, amelyek ugyanilyen tartományban mozognak, így nem vesszük telítésbe az aktivációs függvényeket, valamint nem is tesszük zérussá. Mindkét jelenség csak hátráltatná a betanítás folyamatát, a Loss function, paramétere szerinti deriválása során. A függelékben szereplő 5.3-as kódszegmens testesíti meg a sztenderdizálás folyamatát. Ennél a kódszegmensnél már felosztottuk a feature-öket és elvárt bemenetet párhuzamosan tartalmazó vektort három részre. Vegyük a betanító adathalmaznak a soronkénti várható értékét és a tapasztalati szórását. Ez azt jelenti,

hogy minden egyes feature-höz rendelünk egy várható értéket és egy szórást. Ezekkel az értékekkel sztenderdizáljuk magát a betanító adatot és ugyanezekkel a konstansokkal sztenderdizáljuk a validációs adatot és a teszt adatot, sőt a hálózat felhasználásakor is ezekkel tesszük 0 várható értékűvé és egységnyi szórásúvá a bemeneti adatokat. Mivel a kész hálózat ezekkel a paraméterekkel lett betanítva, ezért ezt el kell menteni és minden későbbi felhasználásnál ezt kell használni.

3.3.5. Hiperparaméterek

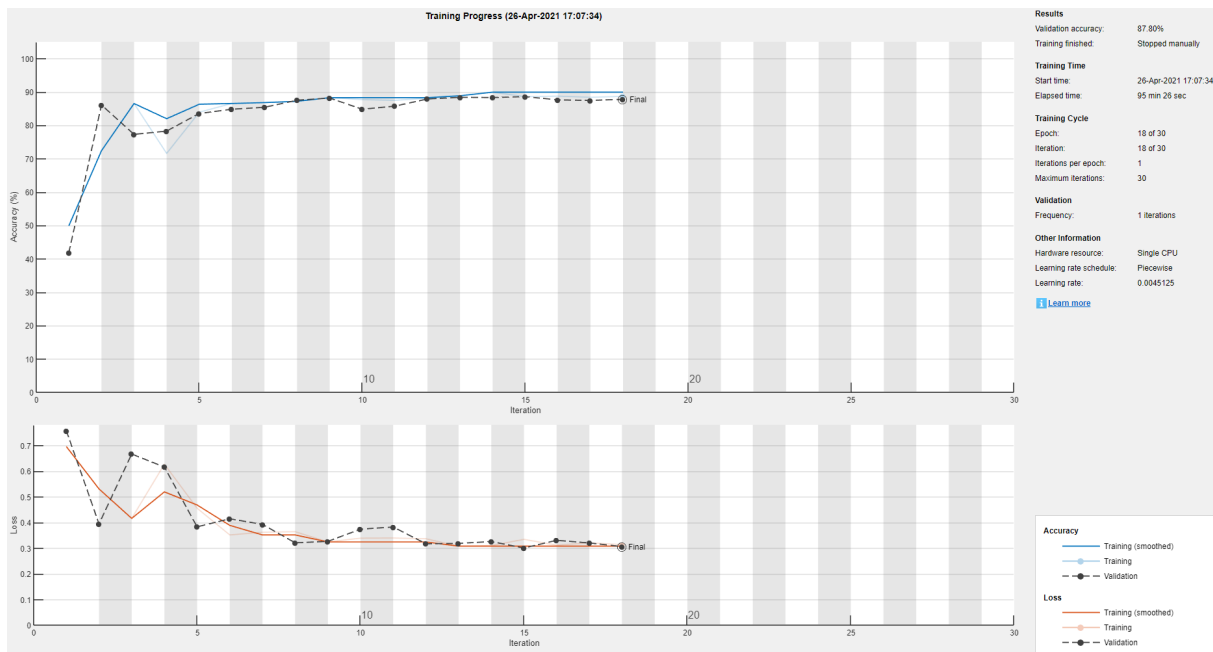
A hiperparaméterek azok a paraméterek amelyek a hálózat betanításának körülményeit szabják meg. Ilyen például az, hogy milyen legyen a tanulási ráta. A hiperparamétereket beállító kódrészlet a függelékben olvasható (5.4). A betanítás során, ha 30 epoch-nál tartunk az azt jelenti, hogy 30-szor ment végig a teljes adatsor a hálózaton. Be lehet állítani egy maximális számot, amennyinél többször ne fusson keresztül a teljes adatsor a hálózaton. Ez jelen esetben 30. Látható, hogy használok validációs adatokat. A validációs frekvencia megmondja, hogy a validációs adatokat hány epoch-onként futtassuk le a betanítás közben. A validációs adat egy a betanító adatoktól független adatsor. A függetlensége miatt jól mutatja, hogy mikor haladunk túl az általános megoldáson és tanulunk rá túlságosan a betanító adatokra. Az a cél, hogy a bármely bemenetre a hálózat ugyanolyan jól reagáljon. Amennyiben a betanító adatokon kezd szignifikánsan jobb lenni a klasszifikációs pontosság, mint a validációs adatokon, akkor érdemes leállítani a betanítást. Be lehet állítani még a tanulási ráta szorzófaktorát is, valamint azt, hogy hány epoch-onként szorozzuk meg a tanulási rátát. Jelen esetben 8 epoch-onként 0,95-el meg lesz szorozva a tanulási ráta. Fontos tulajdonsága a betanításnak az optimalizáló, amellyel keressük a Loss function globális minimumát a hálózat paramétereinek függvényében. Az Adam optimalizálót használom, mely az AdaGrad és az RMSProp előnyeit ötvözi. Más optimalizálókat is kipróbáltam, mint az RMSProp és az SGDM, ám várhatóan az Adam volt a legjobb.

3.3.6. Betanítás folyamata

A betanítást egyszerűen a lenti kódrészlettel lehet elindítani, a szükséges paramétereket megadva, mint a betanító adatok és címkék, valamint a hálózat rétegei és betanítási paraméterek.

```
1 net = trainNetwork(XTrain, YTrain, layers, options);
```

A fent látható függvény betanítási görbéket generál, ez látható a 3.13-ábrán.

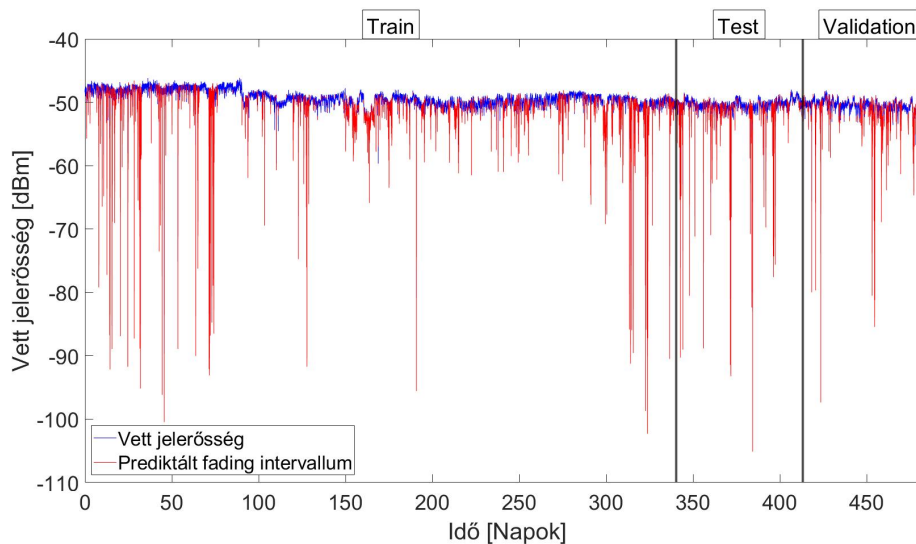


3.13. ábra. Neurális háló sikeres betanítási folyamata MATLAB környezetben.

3.4. Eredmények értékelése

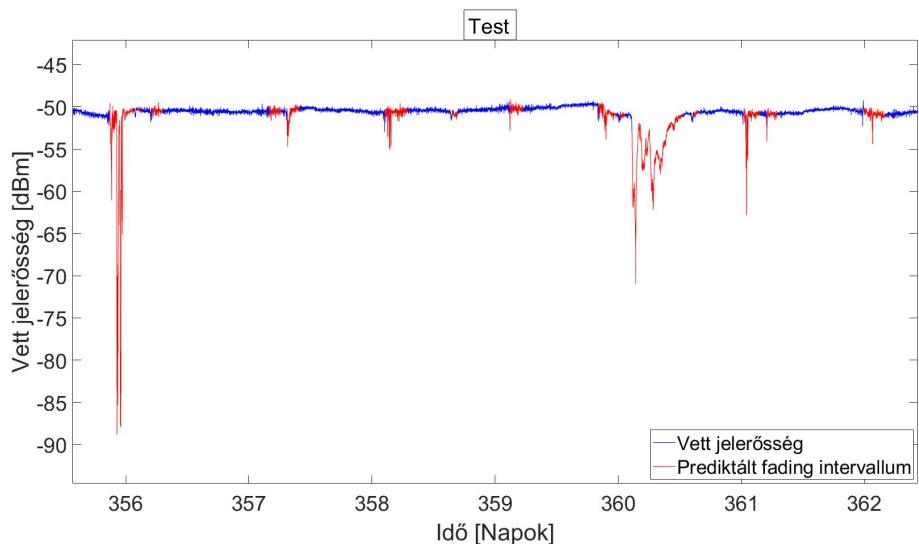
3.4.1. Predikciós képességek vizsgálata

A 3.14-ábrán az egész adathalmazon végigfuttatott hálózat eredménye látható. Ki vannak jelölve azok az időpontok, ahol szétválasztottam az adathalmazt. A következőkben a Test adatokon való viselkedést fogom bemutatni.



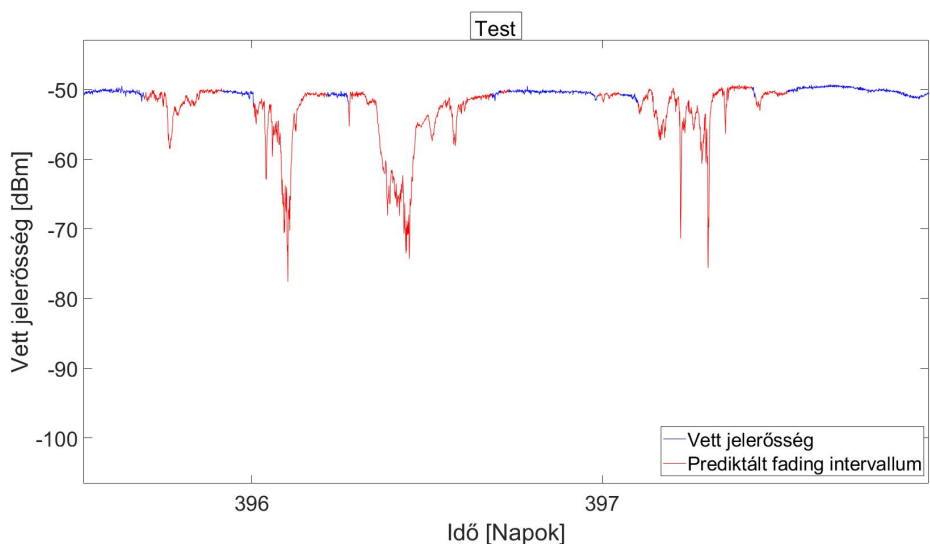
3.14. ábra. Szűrt, kétéves adatsoron vett fading predíciója a vizsgált neurális hálónak.

Belenagyítva a Test intervallumba a 3.15 ábrát kapjuk. Megfigyelhető, hogy hol hibázik a hálózat és, hogy hol szolgál jó predikciókkal.



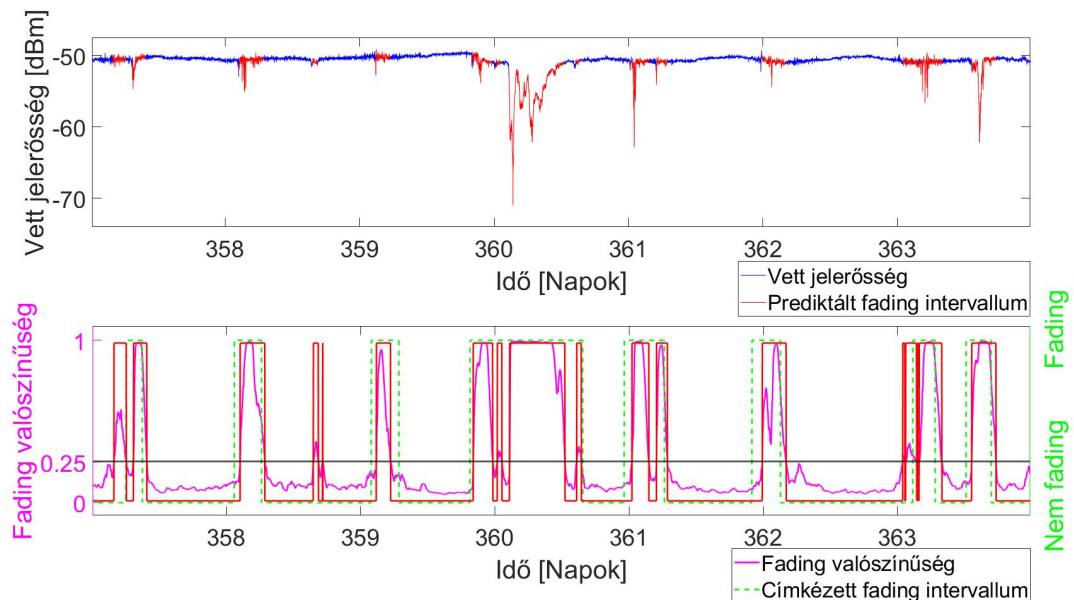
3.15. ábra. Test adathalmazból kiemelt fading predikció.

A nagyobb szcintilláció szerű jelenségekre még mindig érzékeny és a jel gyors negatív változására is könnyen reagál. Ezt jobb feature-ökkel és több adattal úgy gondolom hogy lehet orvosolni. Mivel úgy lettek megjelölve a fading események, hogy még a kritikus jelváltozás előtt fadingnek címkézzük a jelenséget, ezért ez lehet az oka az erős szcintillációra való érzékenységnek. Ennek a címkézésnek pont az volt a célja, hogy ha van valamilyen folyamat amely megelőzi a fading jelenséget és ezt esetleg a hálózat fel tudja ismerni, akkor tanítsuk be erre. Nekem úgy tűnik, hogy a mozgóablakos tapasztalati szóráson keresztül látunk valamit, ami a fading jelenséget megelőzi, vagy legalábbis elég korán tudunk reagálni a jelszint változásra, ahhoz, hogy hasznos legyen a hálózat predikciókra. A 3.17-ábrán ismét a Test adatokban lévő predikciók figyelhetőek meg, egy más időintervallumban.



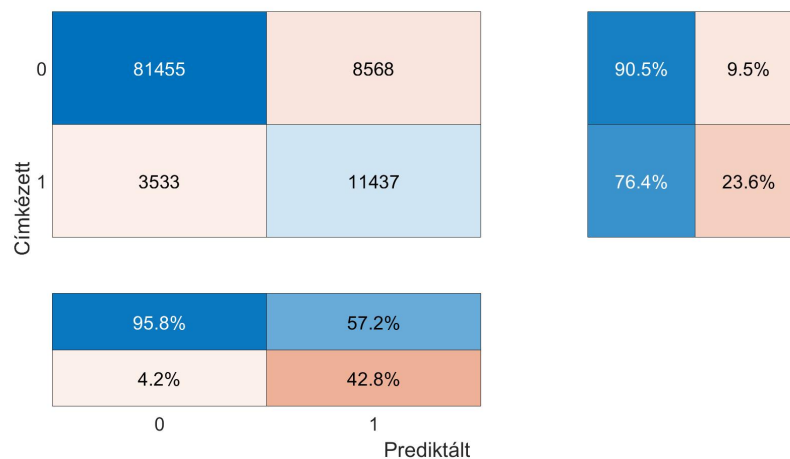
3.16. ábra. Test adathalmazból kiemelt fading predikció.

A 3.17-ábra egy már bemutatott szegmensével foglalkozik a Test-adatoknak. Mivel belátunk a rendszer valószínűségi értékeibe, ezért ábrázolni tudom a fading valószínűséget, amellyel jobban meg tudjuk figyelni, hogy mire hogyan reagál a hálózat.



3.17. ábra. Predikciós eredmények 25%-os döntéshatárral.

Ebben az esetben a fading valószínűség 25%-os értékénél prediktáljuk a következő percbeli osztályt fadingnek. Ez a vonal fel is van tüntetve az alsó szegmensen. Mivel még a jelváltozás előtt egy kicsivel jelöltem fading-et, ezért a piros és a zöld szaggatott vonalak közelsége mutatja a predikciós képesség minőségét. Néhol későn, néhol egész korán rájön a rendszer, hogy fading következik. A 3.18-ábra egy konfúziós mátrixot ábrázol. A konfúziós mátrix középső részeiben az összes minta különböző halmazokra osztva jelenik meg. A legsötétebb kék például azoknak a mintáknak a csoportja, amelyeket én nem fadingnek címkéztem és a neurális háló is nem fadingnek prediktálta.



3.18. ábra. Konfúziós mátrix 25%-os döntéshatárral

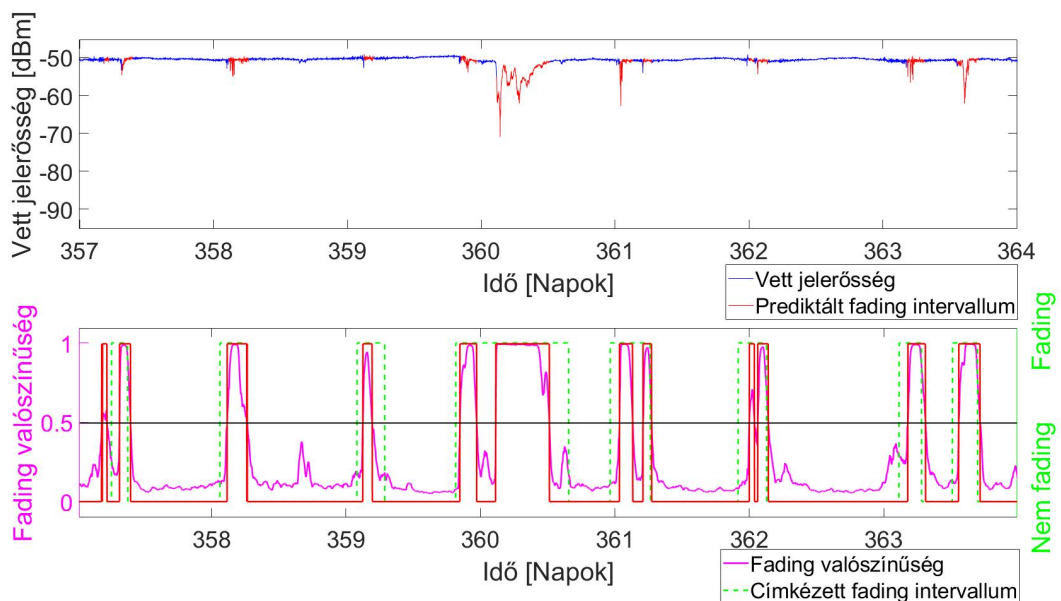
A konfúziós mátrix arról is ad információt, hogy körülbelül mekkora a nem fading és a fading adatminták számossága. Ez a mátrix egy hasznos eszköz, mivel kiszámolható belőle az előrejelzés általános és specifikus helyessége.

$$\frac{\text{helyes predikcio}}{\text{osszes predikcio}} \cdot 100$$

A fenti képlet az előrejelzés általános pontosságát mutatja meg. Ezt a következő módon lehet megállapítani.

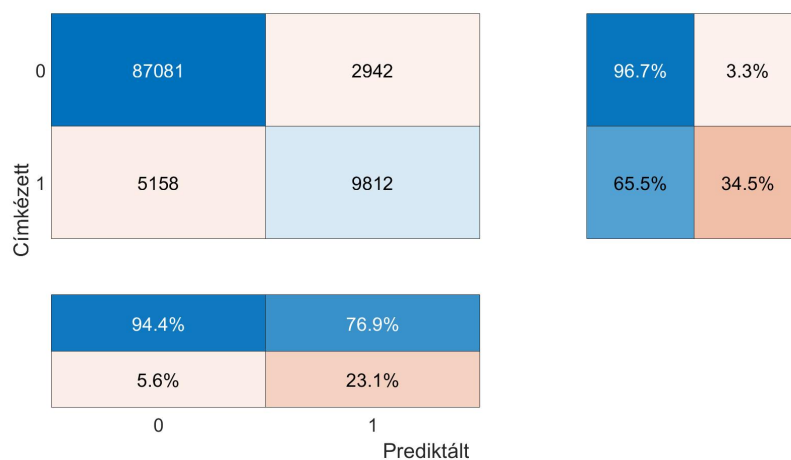
$$\frac{81455 + 11437}{81455 + 8568 + 3533 + 11437} \cdot 100 = 88,47\%|_{0.25 \text{ döntéshatar}}$$

A következőekben megvizsgálom a rendszer választ az 50%-os döntéshatárra (3.19-ábra). Hangsúlyoznom kell, hogy a Softmax réteg kijutása a rendszer kimenetére egy hiba, ám érdekes megvizsgálni a rendszer által látott valószínűségeket. A valószínűségi határ állításával tulajdonképpen szabályozni lehet, hogy a predikciós képessége legyen jobb a hálózatnak, mivel kisebb fading valószínűségekre is fadinget prediktálhatunk, vagy a címkézés és a prediktált értékek legyenek közelebb egymáshoz. Amennyiben elveszünk a rendszer predikciós képességéből, növekedni fog az általános pontosság.



3.19. ábra. Predikciós eredmények 50%-os döntéshatárral.

A 3.20-ábra ismét egy konfúziós mátrixot szemléltet.



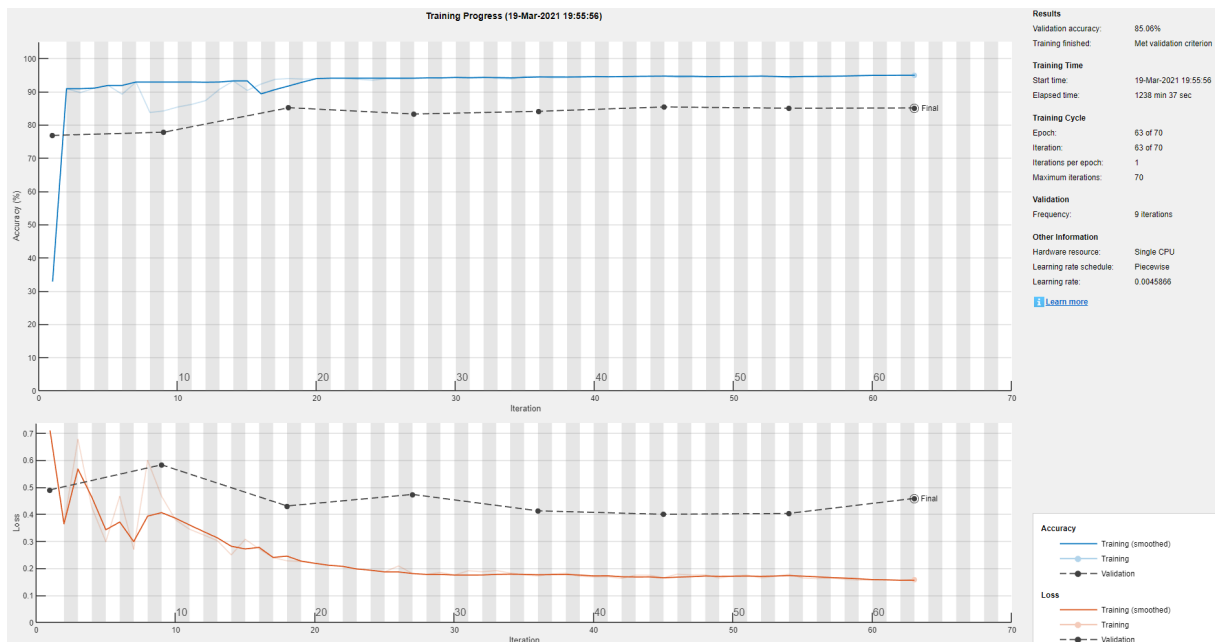
3.20. ábra. Konfúziós mátrix a Test adatokon 50 %-os döntéshatárral.

A mátrixból az általános pontosságot a következők szerint számíthatjuk ki:

$$\frac{87081 + 9812}{87081 + 2942 + 5158 + 9812} \cdot 100 = 92,29\%|_{0.5 \text{ donteshatar}}$$

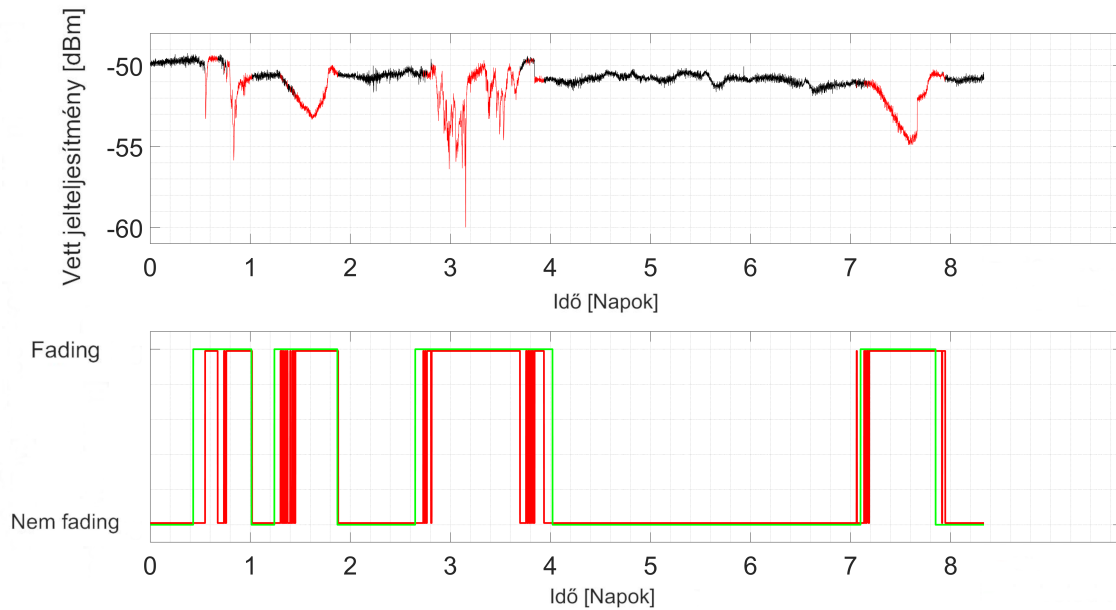
3.5. Fading jelölés mesterséges intelligenciával

Amennyiben szeretnénk olyan neurális hálót létrehozni, amely megjelöli a fading jelenségeket egy előre már adott szekvenciális adathalmazon, akkor megtehetjük, hogy jövőbeli értékeket is beadunk a rendszernek. Erre a feladatra az előzőekben bemutatott architektúra megfelel és effektív. Ebben a problémakörben is ezek a paraméterek hoztak jó eredményt. A feature-öket elállító kódrészletet a függelékben található 5.5-ábra szemlélteti. Mivel igazából ezzel a problémakörrel foglalkoztam először, ezért még nem szerepelt benne egy igen jó feature, a mozgóablakos tapasztalati szórás. Viszont szerepelt benne több különböző méretű mozgóátlaga a jelnek.

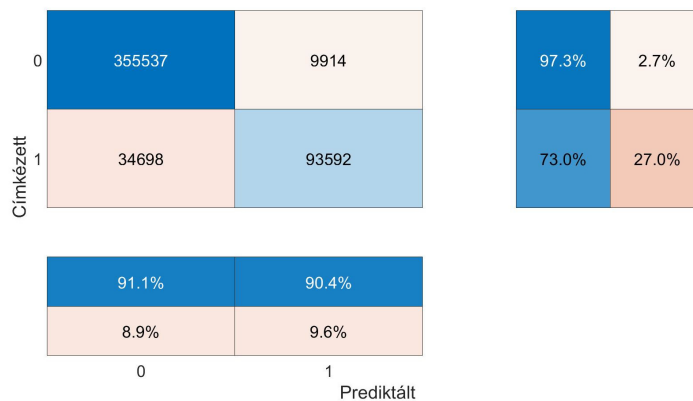


3.21. ábra. Fading jelölésre szánt neurális hálózat betanítási görbéje MATLAB környezetben.

A 3.21-ábrán a betanítás görbéi látszanak. A teszt adatokon 94%, a validációs adatokon 85%-os pontosságot mutatott a neurális háló. Ebben az esetben esetleg lehetett volna használni kétirányú LSTM réteget, mivel az egész adathalmaz adott. Az 3.22-ábrán ennek a betanított neurális hálónak a kimenete látható a Test-adatokon szemléltetve.



3.22. ábra. Test adatokon vett fading jelölés.



3.23. ábra. Konfúziós mátrix a Test adatokon fading jelölésre.

A 3.23-ábrán látható mátrix alapján számolható az általános pontosság a következők szerint:

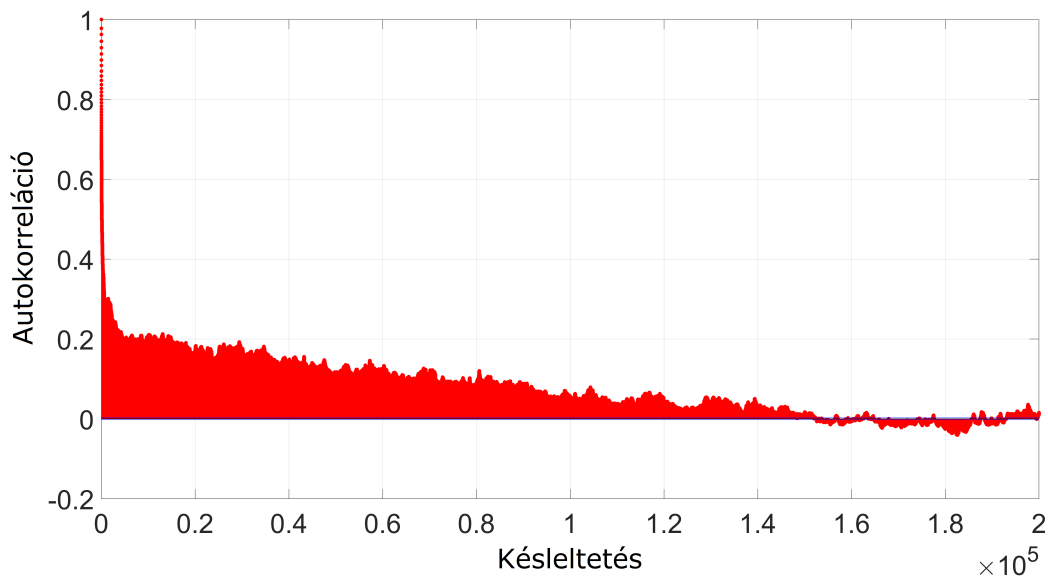
$$\frac{355537 + 93592}{355537 + 9914 + 34698 + 93592} \cdot 100 = 90,96\% .$$

Bár romlott az általános pontosság, ám a valóban fading és a fading-nek klasszifikált percek száma megnőtt. Tehát jövőbeli értékeket táplálva a hálózatba, jobban meg tudjuk állapítani, hogy fading jelenség zajlik -e.

3.6. LSTM rejtett réteg szélesség optimalizálása

A már létező, meghatározott struktúrájú hálózatot optimalizálni lehet a rétegek paramétereinek beállításával. Ebben az esetben kritikus és szignifikáns változások az LSTM szélesség megválasztásának függvényében történnek. A következőkben be fogom mutatni a különböző szélességű hálózatokat, melyeket azonos adatokon tanítottam be. Ezzel a módszerrel ki lehet választani a legmegfelelőbb réteg paramétert.

3.6.1. Szekvenciális adatok autokorrelációja



3.24. ábra. Teljes adatosoron vett autokorreláció.

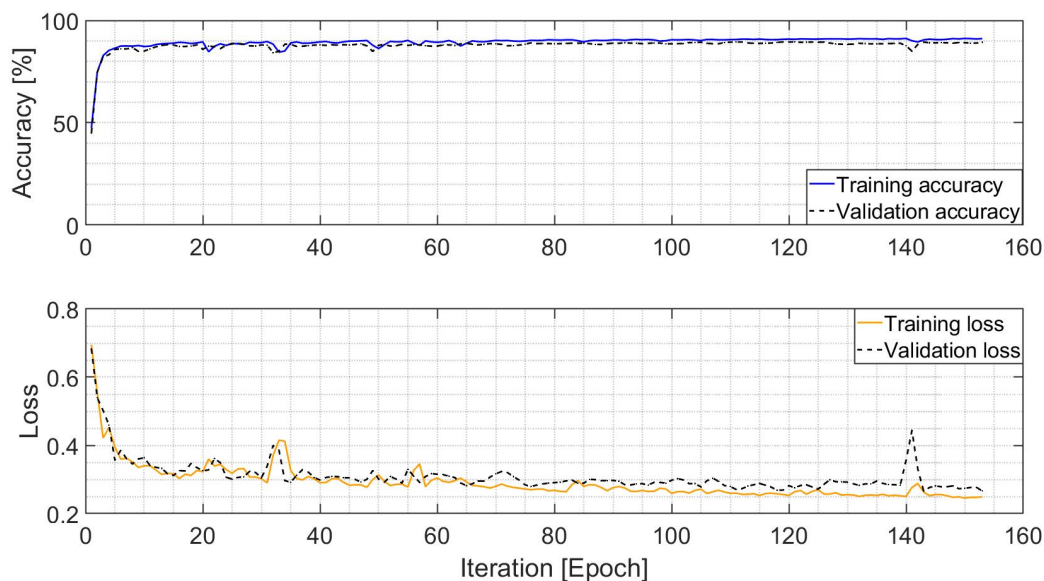
A 3.24-ábrán látható, hogy csak rövid távú függést mutatnak az egyes mérési pontok egymástól, tehát minél távolabbi mintára akarunk predikciót megfogalmazni, annál nehezebb lesz, valamint csak bizonyos mennyiségű időbeli emlékezőképességre van szüksége a hálózatnak, melynél több nem hozna jobb eredményt. Más dolgozatban is olvasható, hogy csak rövid távú időbeli korreláció van a jelteljesítménybeli értékek között [19]. Ennek a korrelációs függvénynek a meghatározása jelentette a kiindulási alapot az LSTM szélesség meghatározásához. A következő pontokban ábrázolni fogom a 10, 25, 50, 75, 100, 150, 200, 500 rejtett rétegű LSTM hálózatok betanítását. A betanításokat úgy végeztem, hogy minden egyes epoch (teljes adathalmaz átfutása a neurális hálón) után validációt végeztem és emellett elmentettem a neurális háló állapotát. Az elmentett több tíz neurális hálóból ezek után a betanító görbék alapján kiválasztottam a legáltalánosabb és legjobb pontosságú hálózatot és erre számoltam konfúziós mátrixot.

3.6.2. 75 széles LSTM rétegek

A 3.25-ábrán megfigyelhető felül a betanítás állapotát jellemző pontosság és alul a betanítás állapotát jellemző kereszt entrópia (1.8-egyenlet). Mindkét mérőszámnál külön van ábrázolva a betanító adatokon vett pontosság és kereszt entrópia, valamint a validációs adatokon vett pontosság és kereszt entrópia értéke. Megfigyelhető, hogy az iterációk száma Epoch mértékegységű. Ez azért van, mert a betanító adatokat nem osztottuk különböző részekre betanítás során, melyeket Bach-elésnek hívnak, hanem az egész adathalmaz egy egész maradt, így egy súly állítás a hálózatban az egész adathalmaz végigfutása után következik be. Mivel az egész adathalmaz egy időtartományban összefüggő entitás, ezért a dolgozatban leírt megoldás alapján nem is volna értelme ezt az adathalmazt több részre választani és ezekkel betanítani a hálózatot.

Bár nyolc féle szélességgel történt betanítás, ám itt csak az egyik hálózat betanítási görbéjét ábrázolom, mivel egészen hasonló karakterisztikát mutatnak, valamint a fontos

információ a nyolc hálózatnál az ezek közül kiválasztott legáltalánosabb megoldás Test adatokon vett konfúziós mátrixa.



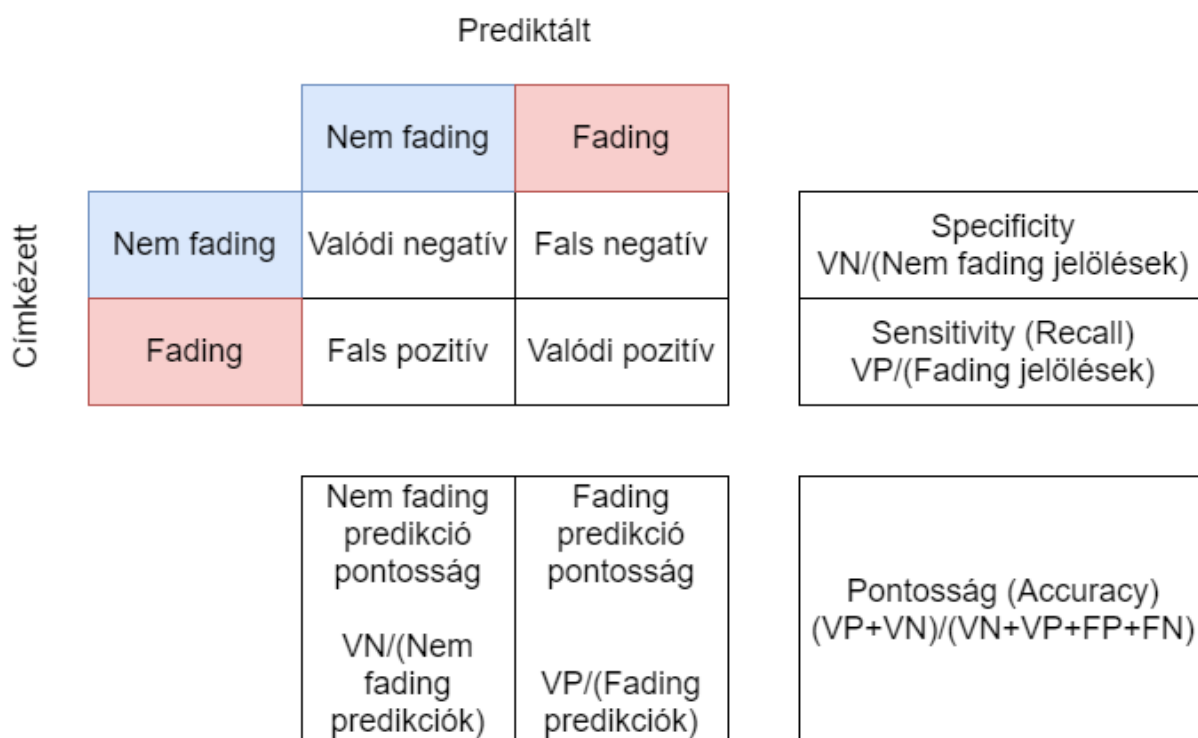
3.25. ábra. Betanítási folyamat 75 LSTM szélesség mellett.

A betanítást az előzőekben leírt okok miatt perces mintavételezéssel végeztem személyi számítógéppel (Intel® Core™ i5-4670 CPU with 32 GB memory). A rétegszám növekedésével megnő a neurális hálózat és a betanítási folyamat által igénybevett memória. Jelen esetben 32 GB memóriával 3 darab 500 szélességű LSTM réteggel tudtam elindítani a betanítást. Az alábbi táblázatban a teljes adathalmaz teljes átfutásához szükséges időt adom meg percben a különböző vizsgált hálózatokhoz.

3.1. táblázat. Vizsgált neurális hálók 1 Epoch-hoz tartozó betanítási ideje [28]

LSTM szélesség	10	25	50	75	100	150	200	500
1 Epoch, validációval [perc]	1,3	1,5	2,75	3,75	6,82	11,96	12,68	123,1

A 3.26-ábra egy konfúziós mátrixot ábrázol. Jelen esetben a pozitív esemény a fading jelenség, ezért a fading-nek címkézett adat, melyet fading-nek is prediktáltunk a valódi pozitív csoportba kerül. Amennyiben a fading-nek jelölt mintát nem fading-nek prediktáljuk, akkor a fals pozitív halmazba kerül. Az ábrán a VN jelölés a mátrixban lévő Valódi negatív csoport értékét jelenti, a többi jelölés hasonló logika alapján oldható fel.



3.26. ábra. Pontosság, specificity, sensitivity, fading predikciós pontosság és nem-fading predikciós pontosság kiszámolásának a módjai konfúziós mátrix esetén.

A 3.2-táblázat tartalmazza a konfúziós mátrix fontosabb elemeit, így jól leírja a neurális háló viselkedését. Ezeket a paramétereket a 3.26-ábra alapján lehet kiszámolni a Test adatokon vett konfúziós mátrixok alapján.

3.2. táblázat. Vizsgált neurális hálók különböző paraméterei [28]

LSTM szélesség	10	25	50	75	100	150	200	500
Pontosság	92,1%	92,6%	93,3%	92,5%	90,7%	92,1%	91,6%	88,3%
Nem fading predikció pontosság	95,0%	95,1%	94,9%	95,3%	96,0%	95,6%	95,4%	96,5%
Fading predikció pontosság	73,3%	76,2%	81,2%	74,6%	64,5%	71,8%	69,6%	58,4%
Specificity	95,8%	96,4%	97,4%	96,0%	92,9%	95,2%	94,7%	89,6%
Sensitivity	69,7%	69,9%	68,7%	71,4%	77,0%	73,5%	72,7%	80,6%

A 3.2-táblázatban megfigyelhető, hogy a memóriaméret növelésével a nem fading predikció pontossága elkezdi növekedni, azonban a fading predikció pontossága csökken. Jelen esetben a sensitivity paraméteren kéne javítani, mivel ez jellemzi a fadingnek jelölt és fading-nek prediktált események arányát. A jövőben a másodperces mintavételezéső adatok jobb sensitivity értékekehez vezethetnek. A különböző szélességű hálózatok közül ezek alapján a legérdemesebb a 75 hidden unit-ot tartalmazó LSTM réteget alkalmazni.

4. fejezet

Konklúzió

Dolgozatomban egy miliméteres hullámhosszú DVB-S2 adatfolyamú műholdas rádiócsatornán a mesterséges intelligencia által vezérelt adaptív moduláció és kódolás megvalósításának lehetőségét vizsgáltam. Ehhez szükség lenne egy N -számú csoportból álló klasszifikációt végrehajtó rendszerre, ahol a különböző osztályok az egyes ACM sémákat jelölnék. Mivel ez egy elég nehéz probléma, ezért az első megközelítésben elegendő egy bizonyos megoldási módszernél csak annyit megmondani, hogy képes-e felismerni a fading és nem fading csoportokat. A dolgozatomban egy ilyen kételemű csoportra végeztem klasszifikációt, ahol is megállapítottam, hogy az adott megoldási módszer (visszacsatolt neurális hálók) képes a probléma legegyszerűbb megfogalmazására jó választ nyújtani. Tehát érdemes ACM tekintetben LSTM rétegekkel foglalkozni, azonban szükségesek további jelteljesítménybeli jellemvonások és struktúrabeli optimalizálások, melyek segítségével egy többelemű klasszifikációt lehet majd végrehajtani.

5. fejezet

Függelék

```
1 numFeatures=4;
2 numResponses=2;
3 numHiddenUnits1=100;
4 numHiddenUnits2=100;
5 numHiddenUnits3=100;
6
7
8 layers = [ ...
9     sequenceInputLayer(numFeatures)
10    lstmLayer(numHiddenUnits1)
11    dropoutLayer(0.6)
12    lstmLayer(numHiddenUnits2)
13    dropoutLayer(0.6)
14    lstmLayer(numHiddenUnits3)
15    dropoutLayer(0.6)
16    fullyConnectedLayer(numResponses)
17    softmaxLayer
18    classificationLayer
19    ];
```

5.1. ábra. Rétegek összetűzését és paramétereit beállító kódszegmens.

```
1 x=linspace(1,1,1);
2 inst_slope = diff(x_m') ./ diff(x);
3 inst_slope(end+1)=inst_slope(end);
4
5 data(1,:)=x_m';
6 data(2,:)=inst_slope;
7 data(3,:)=movmean(x_m',[10 0]);
8 data(4,:)=movstd(x_m',[50 0]);
```

5.2. ábra. A jelteljesítmény jellemvonásait előállító kódrészlet fading predikcióra.

```

1 mu = mean(XTrain,2);
2 sig = std(XTrain,0,2);
3
4 dataTrainStandardized = (XTrain - mu) ./ sig;
5 XTrain = dataTrainStandardized;
6 dataTrainStandardized = (XValidation - mu) ./ sig;
7 XValidation=dataTrainStandardized;
8 dataTrainStandardized = (XTest - mu) ./ sig;
9 XTest=dataTrainStandardized;

```

5.3. ábra. Sztenderdizáció normális eloszlással.

```

1 options = trainingOptions('adam', ...
2     'MaxEpochs',30, ...
3     'ValidationData',{XValidation,YValidation}, ...
4     'ValidationFrequency',5, ...
5     'ValidationPatience',10, ...
6     'GradientThreshold',1, ...
7     'InitialLearnRate',0.005, ...
8     'LearnRateSchedule','piecewise', ...
9     'LearnRateDropPeriod',3, ...
10    'LearnRateDropFactor',0.96, ...
11    'Verbose',1, ...
12    'Plots','training-progress',...
13    'ExecutionEnvironment','auto');

```

5.4. ábra. Hiperparaméterek beállítása.

```

1 window1=10;
2 window2=20;
3 window3=100;
4
5 x=linspace(1,1,1);
6 inst_slope = diff(x_m') ./ diff(x);
7 inst_slope(end+1)=inst_slope(end);
8
9 data(1,:)=x_m';
10 data(2,:)=inst_slope;
11 data(3,:)=movmean(x_m',window1);
12 data(4,:)=movmean(x_m',window2);
13 data(5,:)=movmean(x_m',window3);

```

5.5. ábra. A jelteljesítmény jellemvonásait előállító kódrészlet fading felismerésre.

Irodalom

- [1] “Exploring new frontiers of frequency”, kut. jel., https://www.esa.int/Applications/Telecommunications_Integrated_Applications/Alphasat/Exploring_new_frontiers_of_frequency.
- [2] eoportal. (2021. máj.). “Alphasat I / Inmarsat-XL (Inmarsat-Extended L-band Payload) / Inmarsat-4A F4”. English. <https://eoportal.org/web/eoportal/satellite-missions/a/alphasat>, eoPortal, cím: <https://eoportal.org/web/eoportal/satellite-missions/a/alphasat>.
- [3] ESA, “Aldo Paraboni Q/V Band Payload”, techn. jel., http://www.esa.int/Applications/Telecommunications_Integrated_Applications/Alphasat/Aldo_Paraboni_Q_V_Band_Payload. cím: http://www.esa.int/Applications/Telecommunications_Integrated_Applications/Alphasat/Aldo_Paraboni_Q_V_Band_Payload.
- [4] D. C. H. László, “Műholdas kommunikáció”, http://eik.bme.hu/~csurgai/urtech/Sources/Csurgai_terjedes.pdf, 2021. máj., cím: http://eik.bme.hu/~csurgai/urtech/Sources/Csurgai_terjedes.pdf.
- [5] ESA, “Aldo Paraboni (TDP 5) Payload Q-band and Ka-band horns”, techn. jel., 2013. cím: https://www.esa.int/ESA_Multimedia/Images/2013/07/Aldo_Paraboni_TDP_5_Payload_Q-band_and_Ka-band_horns.
- [6] F. Cuervo, A. Martellucci, J. Rivera Castro, M. Schmidt és M. Schönhuber, “The Alphasat Aldo Paraboni scientific and communication experiments at Ka and Q/V bands in Austria”, *International Journal of Satellite Communications and Networking*, 37. évf., 5. sz., 437–448. old., 2019, <https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1308>. DOI: <https://doi.org/10.1002/sat.1308>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sat.1308>. cím: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1308>.
- [7] (2021. máj.). “Visualize orbit”. English. <https://celestrak.com/cesium/orbit-viz.php?tle=/satcat/tle.php?CATNR=39215&satcat=/pub/satcat.txt&orbits=1&pixelSize=3&samplesPerPeriod=90>, Celestrak, cím: <https://celestrak.com/cesium/orbit-viz.php?tle=/satcat/tle.php?CATNR=39215&satcat=/pub/satcat.txt&orbits=1&pixelSize=3&samplesPerPeriod=90>.
- [8] “Ground track”, 2021, <https://www.n2yo.com/satellite/?s=39215>.
- [9] L. Csurgai-Horváth, “Receiver station in Budapest for Q/V band satellite site diversity and adaptive coding and modulation experiments with Alphasat”, *International Journal of Satellite Communications and Networking*, 37. évf., 3. sz., 2019, <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sat.1270>. DOI: <https://doi.org/10.1002/sat.1270>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sat.1270>. cím: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1270>.

- [10] D. C.-H. L. Márton Boldizsár, “Neurális hálózatok alkalmazása fading jelenségek automatikus detekciójában”, <https://diplomaterv.vik.bme.hu/hu/Theses/Neuralis-halozatok-alkalmazasa-fading>, dipl., BME HVT. cím: <https://diplomaterv.vik.bme.hu/hu/Theses/Neuralis-halozatok-alkalmazasa-fading>.
- [11] D. C.-H. L. Szekeres Dorottya Viktória, “A szcintilláció mérése és modellezése műholdas rádiócsatornán”, <https://diplomaterv.vik.bme.hu/hu/Theses/A-szcintillacio-merese-es-modellezese-muholdas>, dipl., BME HVT, 2014. febr. cím: <https://diplomaterv.vik.bme.hu/hu/Theses/A-szcintillacio-merese-es-modellezese-muholdas>.
- [12] ITU, *Recommendation ITU-R P.618-13*, Radiowave propagation. cím: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.618-13-201712-I!!PDF-E.pdf.
- [13] K. P. Murphy, *Machine learning: a probabilistic perspective*, Adaptive computation and machine learning series sor. Cambridge, MA: MIT Press, 2012, 1067 old., ISBN: 978-0-262-01802-9.
- [14] I. Goodfellow, Y. Bengio és A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [15] M. T. Hagan, H. B. Demuth, M. H. Beale és O. De Jesús, *Neural network design*, 2nd edition. s.L: Martin T. Hagan, 2014, 1 old., ISBN: 978-0-9717321-1-7.
- [16] W. Feng, N. Guan, Y. Li, X. Zhang és Z. Luo, “Audio visual speech recognition with multimodal recurrent neural networks”, 2017. máj. DOI: 10.1109/IJCNN.2017.7965918.
- [17] S. Varsamopoulos, K. Bertels és C. Almudever, *Designing neural network based decoders for surface codes*, 2018. nov.
- [18] A. Amini. (2021. máj.). “MIT 6.S191Introduction toDeep Learning”. English. <http://introtodeeplearning.com/>, MIT, cím: <http://introtodeeplearning.com/>.
- [19] J. Ebert, W. Bailer, J. Flavio, K. Plimon és M. Winter, “A Method for ACM on Q/V-Band Satellite Links Based on Artificial Intelligence”, *2020 10th Advanced Satellite Multimedia Systems Conference and the 16th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, Graz, Austria: IEEE, 2020. okt. 20., 1–5. old., ISBN: 978-1-72815-794-8. DOI: 10.1109/ASMS/SPSC48805.2020.9268889. cím: <http://ieeexplore.ieee.org/document/9268889/> (elérés dátuma 2021. 10. 08.).
- [20] L. Torgo, “Regression trees”, *Encyclopedia of Machine Learning and Data Mining*, C. Sammut és G. I. Webb, szerk., Boston, MA: Springer US, 2017, 1080–1083. old., ISBN: 978-1-4899-7685-7 978-1-4899-7687-1. DOI: 10.1007/978-1-4899-7687-1_717. cím: http://link.springer.com/10.1007/978-1-4899-7687-1_717 (elérés dátuma 2021. 10. 03.).
- [21] L. Bai, C.-X. Wang, Q. Xu, S. Ventouras és G. Goussetis, “Prediction of Channel Excess Attenuation for Satellite Communication Systems at Q -Band Using Artificial Neural Network”, *IEEE Antennas and Wireless Propagation Letters*, 18. évf., 11. sz., 2235–2239. old., 2019. nov., ISSN: 1536-1225, 1548-5757. DOI: 10.1109/LAWP.2019.2932904. cím: <https://ieeexplore.ieee.org/document/8786170/> (elérés dátuma 2021. 10. 08.).

- [22] Y. Yu és L. Zhang, “Adaptive Modulation Scheme for Satellite Communication Channel Based on RLNN”, *Journal of Physics: Conference Series*, 1856. évf., 1. sz., 12053. old., 2021, Publisher: IOP Publishing. DOI: 10.1088/1742-6596/1856/1/012053. cím: <https://doi.org/10.1088/1742-6596/1856/1/012053>.
- [23] P. Kaur, K. Singh és H. Kaur, “Adaptive Modulation of OFDM by using Radial Basis Function Neural Network”, *International Journal of Advanced Research in Computer and Communication Engineering*, 3. évf., 2278–1021. old., 2014. júl.
- [24] M. P. Mota, D. C. Araujo, F. H. Costa Neto, A. L. F. de Almeida és F. R. Cavalcanti, “Adaptive Modulation and Coding Based on Reinforcement Learning for 5G Networks”, *2019 IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, USA: IEEE, 2019. dec., 1–6. old., ISBN: 978-1-72810-960-2. DOI: 10.1109/GCWkshps45667.2019.9024384. cím: <https://ieeexplore.ieee.org/document/9024384/> (elérés dátuma 2021. 10. 08.).
- [25] L.-S. Chen, W.-H. Chung, I.-Y. Chen és S.-Y. Kuo, “AMC With a BP-ANN Scheme for 5G Enhanced Mobile Broadband”, *IEEE Access*, 1–1. old., 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3024726. cím: <https://ieeexplore.ieee.org/document/9200321/> (elérés dátuma 2021. 10. 08.).
- [26] R. Daniels és R. W. Heath, “Online adaptive modulation and coding with support vector machines”, *2010 European Wireless Conference (EW)*, 2010, 718–724. old. DOI: 10.1109/EW.2010.5483527.
- [27] S. Kojima, K. Maruta és C.-J. Ahn, “High-precision SNR Estimation by CNN using PSD Image for Adaptive Modulation and Coding”, *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, 1–5. old. DOI: 10.1109/VTC2020-Spring48590.2020.9129413.
- [28] M. Árpád László, D. Tamás és C.-H. László, “AI-Supported Fading Prediction”, English, *26th Ka Band Communications Conference 2021 Propagation*, Propagation 4: sor., ISSN-2573-6124, Washington DC: Ka, Broadband Communications, Navigation és Earth Observation Conference, 2021. szept., 1–5. old. cím: https://proceedings.kaconf.com/papers/2021/ka4_2.pdf (elérés dátuma 2021. 10. 14.).