Budapesti Műszaki és Gazdaságtudományi Egyetem
Természettudományi Kar

Számítástechnikai és Automatizálási Kutató Intézet (SZTAKI),
Mérnöki és Üzleti Intelligencia Kutatólaboratórium (EMI),
Intelligens Folyamatok Kutatócsoport (IP)

Statisztikát és információelméletet integráló mértéket alkalmazó dinamikus algoritmus mesterséges neurális hálózat tanítására

TDK DOLGOZAT

Szűcs Ágnes

Témavezető:

Dr. Viharos Zsolt János

Budapest, 2022

Köszönet

Köszönöm Dr. Viharos Zsolt Jánosnak a lelkes és érdeklődő témavezetést, az induló, továbblendítő ötleteket és az eddigi közös munkát.

Hálás vagyok a családomnak és a barátaimnak, hogy támogatnak a döntéseimben. Köszönöm kollégáimnak, az Intelligens Folyamatok Kutatócsoport tagjainak a motiváló társaságot.

Továbbá köszönöm a Budapesti Műszaki és Gazdaságtudományi Egyetemnek és a Számítástechnikai és Automatizálási Kutatóintézetnek, hogy intézményi hátteret biztosít a szakmai fejlődésemhez.

A kutatás az Európai Unió támogatásával valósult meg, az RRF-2.3.1-21-2022-00004 azonosítójú, Mesterséges Intelligencia Nemzeti Laboratórium projekt keretében, valamint a kutatást a "Kooperatív gyártó- és logisztikai rendszerek kutatása a versenyképes és fenntartható gazdaság támogatására" (COPROLOGS) című TKP2021-NKTA-01 NKFIH támogatás is lehetővé tette.

Tartalomjegyzék

Bevezetés

A Mesterséges Intelligencia és az Adattudomány széles körben elterjed módszerei között tartják számon a mesterséges neurális hálózatok használatát.

Ezen modell felépítésének fontos kulcseleme a hálózat súlyainak tanítása, beállítása algoritmikus módszerekkel. Jelen TDK dolgozat egy új tanítómérték és egy új algoritmus részletes elemzését tűzi ki célul. Ez a módszer a tanítás, mint optimalizálási problémának a megoldására a Levenberg-Marquardt (LM) algoritmust használja egy fontos újítással módosítva. A hiba, eltérés méréséreaz adattudományban elterjedten használt (átlagos) négyzetes hiba (MSE) helyett a kutatás Silva et al. (2008) által bemutatott integrált exponenciális hiba továbbfejlesztett változatát, az újonnan bevezetett abszolútértékes exponenciális hibát alkalmazza, valamint az LM elgoritmust is kiegészíti új elemekkel.

Ez a többszörösen továbbfejlesztett algoritmus nagyszámú benchmark adathalmazon került tesztelésre. A dolgozat ennek a tesztelésnek több fő szempontjai szerint, mint pontosság, sebesség, stabilitás kiértékelését mutatja be, ezeken kívül beszámol a témához kapcsolódó releváns tudományos irodalomban fellelhető state-of-the-art eredményhez képest elért jobb algoritmikus teljesítményről is.

(A szerző az alapötlet korábbi változatát és más aspektusait BSc szakdolgozatában (Szűcs (2020)) és korábbi TDK (Szűcs (2021)) dolgozatában is vizsgálta.)

1. fejezet

Irodalmi áttekintés

1.1. Levenberg-Marquardt algoritmus

Ez az alfejezet bemutatja a Levenberg-Marquardt algoritmust, mint optimalizálási feladatot, valamint említést tesz arról, hogy ez hogyan alkalmazható neurális háló tanításban.

1.1.1. Optimalizálási feladatként

A Levenberg-Marquardt algoritmust két kutató Kenneth Levenberg és Donald Marquardt fejlesztette ki egymástól függetlenül, az algoritmus nem-lineáris függvények minimalizálására ad numerikus megoldást (Yu & Wilamowski, 2011). A következő bekezdésekben az olvasó végigkövetheti a Levenberg-Marquardt (LM) algoritmus "evolúcióját" a gradiens módszertől kezdve, ez az LM módszer előnyös és előnytelen tulajdonságainak jobb megértését szolgálja.

1.1.1.1. Gradiens módszer

A gradiens módszer esetén az $f(\mathbf{x})$ függvény minimumhelyének keresése egy adott pontból ($\mathbf{x_k}$) indul és a legnagyobb csökkenés irányába halad lépésről lépésre. A csökkenés a gradiens vektor ellentettjeként határozható meg, ami alapján az újabb (minimumhelyhez "közelebbi") pont ($\mathbf{x_{k+1}}$) a következő képlet szerint kapható:

$$\mathbf{x_{k+1}} = \mathbf{x_k} - \alpha_k \nabla f(\mathbf{x_k}) \tag{1.1}$$

Vagyis az $\mathbf{x_k}$ érték eltolódik a csökkenés irányába a gradiensnek valamilyen α-szorosával ($\alpha_k \geq 0$). A gradiens módszer hátránya, hogy nem találja meg biztosan a függvény globális minimumát. Ha az algoritmus által meghatározott $\mathbf{x_k}$ pontok sorozata egy lokális minimum-

helyhez konvergált, akkor a továbbiakban is ennek a pontnak a környezetében fog maradni, az algoritmus ebben az esetben nem találja meg a globális minimumot.

1.1.1.2. Newton módszer

A Newton-módszerben a minimalizálandó függvényt közelítjük annak elsőfokú Taylor-polinomjával az $\mathbf{x_k}$ pontban.

$$f(\mathbf{x}) \approx f(\mathbf{x_k}) + f'(\mathbf{x_k})(\mathbf{x} - \mathbf{x_k}) =: \hat{f}(\mathbf{x}) \tag{1.2}$$

Majd ennek az $\hat{f}$ függvénynek kell keresni a minimum helyét, vagyis ahol:

$$\hat{f}'(\mathbf{x}) = f'(\mathbf{x_k}) + f''(\mathbf{x_k})(\mathbf{x} - \mathbf{x_k}) = 0 \tag{1.3}$$

Az $\hat{f}'$ zérushelye fogja megadni az újabb pontot ($\mathbf{x_{k+1}}$):

$$\hat{f}'(\mathbf{x_{k+1}}) = f'(\mathbf{x_k}) + f''(\mathbf{x_k})(\mathbf{x_{k+1}} - \mathbf{x_k}) = 0 \tag{1.4}$$

Ezt átrendezve adódik:

$$\mathbf{x_{k+1}} = \mathbf{x_k} - \frac{f'(\mathbf{x_k})}{f''(\mathbf{x_k})} \tag{1.5}$$

Az gradiens vektor és a Hesse-mátrix jelöléseinek behelyettesítésével:

$$\mathbf{x_{k+1}} = \mathbf{x_k} - \mathbf{H}^{-1}(\mathbf{x_k})\nabla f(\mathbf{x_k}) \tag{1.6}$$

A Newton módszerben szükséges az f függvény kvadratikusságának feltételezése. Az algoritmus hátránya, hogy nem csak az első, hanem a második deriváltak ismerete (kiszámítása) is szükséges hozzá.

1.1.1.3. Gauss-Newton módszer

A Gauss-Newton módszer nem-lineáris legkisebb négyzetes problémákra alkalmazható. Itt a cél az elsőfokú Taylor-polinomjával becsült minimalizálandó f függvény ($\hat{f}$) norma négyzetének a minimalizálása(Boyd & Vandenberghe, 2018).

$$\min \left\| \hat{f}(\mathbf{x}) \right\|^2 \tag{1.7}$$

Az első és második deriváltak a következőképpen alakulnak:

$$\left(\left\|\hat{f}(\mathbf{x})\right\|^2\right)'\Big|_{\mathbf{x_k}} = \sum_i f_i(\mathbf{x_k})\nabla f_i(\mathbf{x_k}) = \mathbf{J}^T(\mathbf{x_k})f(\mathbf{x_k}) \Rightarrow \nabla f \tag{1.8}$$

$$\left(\left\|\hat{f}(\mathbf{x})\right\|^2\right)''\Big|_{\mathbf{x_k}} = \mathbf{J}^T(\mathbf{x_k})\mathbf{J}(\mathbf{x_k}) + \sum_i f_i(\mathbf{x_k})\nabla^2 f_i(\mathbf{x_k}) = \mathbf{J}^T(\mathbf{x_k})\mathbf{J}(\mathbf{x_k}) \Rightarrow \mathbf{H} \tag{1.9}$$

A második deriváltban a szummás tag elhanyagolható, mivel a szorzandók valamelyike mindenképpen nagyon kicsi lesz (Ranganathan, 2004). Ezeket összevetve a 1.6 egyenlettel, a következő képlet adja meg az $\mathbf{x_{k+1}}$-et.

$$\mathbf{x_{k+1}} = \mathbf{x_k} - \left(\mathbf{J}^T(\mathbf{x_k})\mathbf{J}(\mathbf{x_k})\right)^{-1}\mathbf{J}^T(\mathbf{x_k})f(\mathbf{x_k}) \tag{1.10}$$

A Gauss-Newton módszerrel való közelítésben nem szükséges kiszámolni a második deriváltakat, viszont a $\mathbf{J}^T\mathbf{J}$ invertálhatósága már okozhat problémát. Ezenkívül az így kapott minimumhely messze is eshet a valódi minimumhelytől.

1.1.1.4. Levenberg-Marquardt módszer

A Levenberg-Marquardt algoritmusban nem csak az f függvény norma négyzetének minimalizálása a cél, hanem egy regularizáló tag is bevezetésre kerül, ami szabályozza, hogy mennyire eshet távol a közelített szélsőérték a valóditól ($\mu_k > 0$) (Ez a Gauss-Newton algoritmusnak hátránya volt). Ami képlettel az alábbiak szerint írható fel:

$$\min\left(\left\|\hat{f}(\mathbf{x})\right\|^2 + \mu_k\|\mathbf{x} - \mathbf{x_k}\|\right) \tag{1.11}$$

Ez alapján a Hesse-mátrixra ezt a közelítést kapjuk:

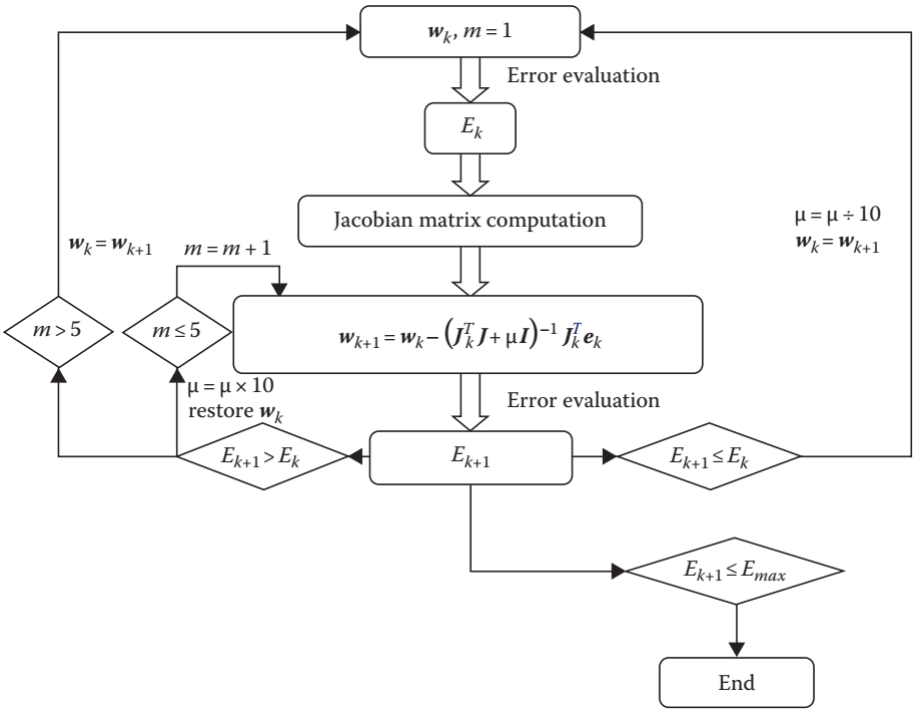
$$\mathbf{H}(\mathbf{x_k}) \approx \mathbf{J}^T(\mathbf{x_k})\mathbf{J}(\mathbf{x_k}) + \mu_k\mathbf{I} \tag{1.12}$$

Az $\mathbf{x_{k+1}}$ kiszámításának a képlete pedig:

$$\mathbf{x_{k+1}} = \mathbf{x_k} - \left(\mathbf{J}^T(\mathbf{x_k})\mathbf{J}(\mathbf{x_k}) + \mu_k\mathbf{I}\right)^{-1}\mathbf{J}^T(\mathbf{x_k})f(\mathbf{x_k}) \tag{1.13}$$

Ezzel a Levenberg-Marquardt módszer az invertálhatóság problémáját is kiküszöböli (Yu & Wilamowski (2011)), viszont a $\mathbf{J}$ mátrix esetlegesen nagy mérete még szintén okozhat számítási nehézségeket.

Látható, hogy az LM algoritmusban a μ_k adaptív megválasztásával egy olyan algoritmus valósul meg, ami $\mu \to 0$ esetén a Gauss-Newton módszer szerint viselkedik, $\mu \to \infty$ esetén pedig a gradiens módszerhez hasonlóan tesz lépéseket.



1.1. ábra. A Levenberg-Marquardt algoritmus folyamatábrája. Forrás: Yu & Wilamowski (2011)

Célszerű tehát az optimalizálás során a μ_k értékét bizonyos feltételektől függően variálni például mint 1.1 ábrán, így mindkét algoritmus (gradiens, Gauss-Newton) előnyös tulajdonságai egyidejűleg kihasználhatóvá válnak. Ha a feltételek engedik, akkor begyorsul az algoritmus, viszont ha "óvatosságra" van szükség, akkor gradiens módszer szerűen lépeget a minimumhely felé.

1.1.2. Neurális hálók tanítása

A neurális háló tanítása során a modellen képzett output és ez elvárt kimenet közötti különbség négyzetének minimalizálása a cél, ha az MSE a hibamérték az adott feladatban. Az output leírható a háló súlyainak (mint változóknak, 1.1.1 részben $\mathbf{x_k}$) a függvényeként, majd ezzel négyzetes hibát képezve kapható a minimalizálandó célfüggvény (1.1.1 részben az f). Így tehát a *neurális háló tanítása optimalizálási feladatként* állt elő.

Ugyanez lesz a kulcs más hibamértékek használata esetén is. Ennek a megoldásához most már csak az optimalizáló módszer elemeinek kiszámítása (képletének meghatározása), LM esetén ez a Jacobi mátrix, és az algoritmus futtatása szükséges. Erre az irodalomkutatás a TDK dolgozat szempontjából fontos, két lehetséges módszert is bemutat, melyek alapján kétféle új algoritmus is levezethető.

1.1.2.1. Yu & Wilamowski-féle Nagymátrix alapú tanítás

Yu & Wilamowski (2011) a négyzetes hibaösszeg (Sum of Squared Error, SSE) minimalizálására törekszik. A Jacobi mátrixot úgy definiálják, hogy a sorokat az adott patternek esetén adott output csúcsok határozzák meg, az oszlopokat pedig a súlyok. A mátrix elemei az adott patternre, adott csúcson kijövő output elvárttól való eltérésének az adott súly szerinti deriváltja lesz. A Jacobi mátrix **sorainak száma pattern szám x output szám, oszlopainak számát pedig a súlyok száma** adja.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \frac{\partial e_{1,1}}{\partial w_2} & \cdots & \frac{\partial e_{1,1}}{\partial w_W} \\ \frac{\partial e_{1,2}}{\partial w_1} & \frac{\partial e_{1,2}}{\partial w_2} & \cdots & \frac{\partial e_{1,2}}{\partial w_W} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{1,N_L}}{\partial w_1} & \frac{\partial e_{1,N_L}}{\partial w_2} & \cdots & \frac{\partial e_{1,N_L}}{\partial w_W} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{P,1}}{\partial w_1} & \frac{\partial e_{P,1}}{\partial w_2} & \cdots & \frac{\partial e_{P,1}}{\partial w_W} \\ \frac{\partial e_{P,2}}{\partial w_1} & \frac{\partial e_{P,2}}{\partial w_2} & \cdots & \frac{\partial e_{P,2}}{\partial w_W} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{P,N_L}}{\partial w_1} & \frac{\partial e_{P,N_L}}{\partial w_2} & \cdots & \frac{\partial e_{P,N_L}}{\partial w_W} \end{bmatrix}_{((\text{RawNr.:\#Patterns(P)} \times \text{\#Outputs}(N_L)) \times \text{ColumnNr.:\#Weights(W)})} \tag{1.14}$$

Az algoritmus futtatása során a μ értékét úgy változtatja, hogy egy nagyságrenddel növeli, ha az algoritmus éppen távolodik az optimum ponttól, és csökkenti, ha jó irányban halad. Ez a fajta "jóság" a frissített súlyokon kiszámított hiba előzőleg mért hibával való összehasonlításából derül ki. A Yu & Wilamowski-féle Nagymátrix módszer részletes levezetését és különböző szempontok szerinti értékelését tartalmazza a szerző korábbi TDK dolgozata (Szűcs (2021)).

1.1.2.2. Heravi & Hodtani-féle Kismátrix alapú tanítás

Heravi & Hodtani (2016) a LM algoritmusban a Correntrópiából (CorrE) Gauss kernel segítségével származtatott mérték felhasználását célozza meg. Ez a mértéket mintaelemenként definiálja, majd ebből építi fel a Jacobi mátrixot a következő képpen. A különböző súlyok adják a sorokat, az oszlopokat pedig az egyes mintaelemek (pattern), a mátrixban pedig a származtatott mérték súly szerinti deriváltja szerepel.

$$J_i = \begin{bmatrix} \frac{\partial e(1)}{\partial w_1} & \frac{\partial e(2)}{\partial w_1} & \cdots & \frac{\partial e(P)}{\partial w_1} \\ \frac{\partial e(1)}{\partial w_2} & \frac{\partial e(2)}{\partial w_2} & \cdots & \frac{\partial e(P)}{\partial w_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e(1)}{\partial w_j} & \frac{\partial e(2)}{\partial w_j} & \cdots & \frac{\partial e(P)}{\partial w_j} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e(1)}{\partial w_W} & \frac{\partial e(2)}{\partial w_W} & \cdots & \frac{\partial e(P)}{\partial w_W} \end{bmatrix}_{\text{(RawNr.:\#Weights}(W)\times\text{ColumnNr.:\#Patterns(P))}} \tag{1.15}$$

Majd ezzel a Jacobi mátrixszal a megfelelően definiált súlyfrissítési képlet segítségével hajtották végre az optimalizációt. 1.1.2.1-től ez abban is különbözik (a hibamértéken túl), hogy a szerzők nem vették külön-külön output csúcsonként a hibát. Ezzel a mátrix dimenziója az outputszámszorossal kisebb lett, vagyis **pattern szám x a súlyok száma**.

1.2. Lehetséges hibamértékek

Ebben az alfejezetben elsősorban a neurális háló tanításra korábban, az irodalomban használt információelméleti mértékek és azok egymáshoz való viszonya kerül bemutatásra.

1.2.1. Neurális háló tanításban használt mértékek

Korunkban a mesterséges intelligencia és a gépi tanulás az élet számos területén fontos szerepet játszik, rengeteg kutató foglalkozik a módszerek tökéletesítésével és alkalmazásával. Már az előző század vége felé megindult ezeknek az elméleti fejlesztése Prieto et al. (2016). A modellezési feladatokban -, mint a mesterséges neurális hálók (ANN) - az klasszikusan használt (átlagos) négyzetes hiba (MSE) mellet már újabb hibamértékeket (például az információelmélet területéről) is elkezdtek használni a tanító algoritmusokban. Azóta is többeket foglalkoztat ez a felvetés és különböző aspektusai, ennek a fejlődésnek a - jelen dolgozat szempontjából is lényeges - kulcs momentumairól ad áttekintést a következő néhány bekezdés.

A témakörrel foglalkozó egyik első kutatásban Hopfield (1987) a valószínűségi függvények összehasonlítátsára jól használható entrópiát alkalmazza optimalizálandó költségfüggvényként kétféle neurális háló típusnál: a három rétegű analóg perceptronnál és a Boltzmann hálónál. Hopfield cikkében összeveti ezt a két modellt, de más hibafüggvényt nem vizsgál. Watrous (1992) már elvégzi a relatív entrópia négyzetes hibával való összehasonlítást különböző optimzáló algoritmusokban, viszont nem állapít meg egyértelmű hasonlóságot vagy

különbséget a teljesítményük között.

Park et al. (1995) vizsgálja a gradiens módszerben a négyzetes hiba (mint MSE) és a relatív entrópia (mint CE) más alternatíváit, mint Kullback-Leibler (KL), Jensen (J) és Jensen-Shannon (JS) divergencia. Részletezi a neurális háló működését és elemzi a mértékek viselkedését különböző hálóparaméter beállítások mellett. Szemlélteti az alternatív mértékek előnyeit, viszont az is kitűnik, hogy ezek nagyban függenek a paraméterbeállításoktól.

Az ANN tanító algoritmusok egyfajta optimalizáló módszernek is tekinthetők. Erdogmus & Principe (2002) cikkükben elméleti megállapításokat bizonyítanak az információelméleti mértéket használó optimalizáló algoritmusokra. Kutatásukban a Shannon-entrópia (SE) neurális-háló tanításban való alkalmazhatóságát vizsgálják. Belátják, hogy az α-rendű Rényi entrópia (RE) minimalizálja a Csiszár-távolságot, ezért egy speciális esetben, a SE-ra (ami a RE $\alpha \rightarrow 1$) igaz, hogy minimalizálja a távolságot a bemenő-kijövő illetve a bemenő-elvárt kimenetű adatpárok sűrűségfüggvényei között. Bebizonyították azt is, hogy az *approximált sűrűségfüggvénnyel számolt entrópia globális minimuma megegyezik a tényleges entrópiáéval, ezért a közelített entrópia használható az optimalizálás során*. Ezek a megállapítások a jelen TDK kutatás szempontjából is fontosak, mert megteremtik az elméleti hátterét a lehetséges megoldásoknak.

Ez az alapvető információelméleti mérték (SE) volt Silva et al. (2005b) jelöltje az általánosan használt négyzetes hiba lecserélésére, amit gradiens módszert alkalmazó tanításba illesztettek bele adaptív tanulórátát alkalmazva. A tapasztalatok azt mutatták, hogy az algoritmus stabilitása különösen érzékeny volt a nemparaméteres kernelbecslő simasági paraméterére. Az SE-t az MSE-hez és a CE-hez hasonlítva azt a következtetést vonták le, hogy az SE képes bizonyos esetekben jobb teljesítmény elérésére a tesztelési hibát és annak szórását tekintve.

Rady (2011b) részletesen összehasonlította az MSE-t és SE-t. Leírta, hogy az MSE jó választás a tanításra lineáris rendszer és Gauss-zaj esetén, viszont nemlineáris rendszerben és nem-Gaussos zaj esetén hibázik. A SE egy olyan alternatívája lehet az MSE-nek, ami képes ezt a problémát kiküszöbölni. A kutatásban különböző feltételek mellett (különféle aktivációs függvények és tanulórátáák) tesztelték az MSE-t és a SE-t, ami alapján mindkettőről megállapíthatók előnyös tulajdonságok. Például, hogy az MSE gyorsítja a konvergencia sebességét, viszont a konvergencia foka SE esetén magasabb. (A cikkben még más szempontú összehasonlítások is szerepelnek, de azok jelen kutatás szempontjából kevésbé fontosak.)

Az RE (Rényi Entrópia) egy általánosított információelméleti mérték α (alpha) paramé-

terrel, aminek másodrendű változatát ($\alpha = 2$) használják általában a tanítóalgoritmusokban. Rady (2011a) cikke a MSE és a RE tapasztalati alapú összehasonlítását mutatja be. Hasonlóan a Rady (2011b) munkájához a tanító algoritmust különböző szempontokból vette górcső alá és azt a megállapítást tette, hogy az MSE jobban gyorsítja a konvergenciát, mint az RE, viszont az RE esetén magasabb a konvergencia foka.

A kölcsönös információ (mutual information - MI) egy KL-hez hasonló információelméleti mérték. Olyan gyakorlati AI problémák megoldásánál nagyon hasznos, ahol valószínűségi sűrűségfüggvények összehasonlítása a feladat. Ez több módszerrel is lehetséges, amit Wen et al. (2019) cikke is bemutat, köztük a saját kutatási eredményükként létrejött többváltozós permutált feltételes kölcsönös információ használatát. Egy használható újszerű megoldást fejlesztettek, ami robusztus és hatékony, valamint kevésbé számításigényes, mint más hasonló módszerek.

Az irodalomban nem csak a jól ismert információ elméleti mértékek lelehetők fel, hanem újonnan fejlesztett mértékek is, amik többféle adattudományi problémára alkalmazhatók. Silva et al. (2008) például egy olyan hibafüggvényt konstruáltak neurális háló tanításhoz, ami két matematikai területet integrál, a statisztikát és az információelméletet. Ez az exponenciális hiba (E_{Exp}), ami ígéretes kombinációja különböző mértékeknek egy paraméter függvényében. Ez az új mérték különösen is fontos a TDK kutatás szempontjából, részletesebb taglalása alább olvasható, a 2.2.1 részben.

Amaral et al. (2013) különböző költségfüggvényekkel tanítva elemezte az auto-encoder működését. A CE és az összesített négyzetes hiba -SSE (MSE-hez hasonló) hiba mellett a Silva et al.-féle exponenciális hibát (E_{Exp} - csak $\tau > 0$-ra) is alkalmazta az előtanító és finomhangoló fázisokban. Az osztályozási feladatokon szerzett tapasztalatok azt mutatják, hogy a SSE a legjobb az előtanításra, a CE és E_{Exp} pedig az elvártnál korábban megállt a beépített korai leállási feltétel (early stopping) miatt. A különböző költségfüggvény párosítások nem voltak nagy hatással a klasszifikáció eredményére, bár az megfigyelhető volt, hogy az E_{Exp} függvény a finomhangolás során jól kezeli a kiegyensúlyozatlan adatot. Ehhez hasonló viselkedés volt megfigyelhető a CE-re is a kísérleti eredmények alapján.

A gyakorlatban az E_{Exp} többek között ózonepizódok forrásainak kutatásában használták, ami a troposzférában előforduló ózon egészségre káros hatása miatt fontos. Fontes et al. (2014) konstruált és optimalizált egy Multilayer Perceptront (MLP) az ózonepizódok bináris predikciójára. A tanítás során a hiba mérésére CE-t és E_{Exp}-t használtak, valamint keresték az optimális modellparamétereket (τ értéke, és a rejtett rétegen lévő csúcsok száma). Cikkükben

csak pozitív τ értékeket használtak, valamit azt is megemlítették, hogy **az E_{Exp} rendkívül érzékeny a τ paraméter értékének megválasztásra**.

Silva et al. (2014) cikkükben részletesen értékelték az MSE és néhány információelméleti mérték (CE, SE, kvadratikus RE - RE2, zero-error sűrűség - ZED) teljesítményét. Multilayer perceptront tanítottak ezen hibafüggvények felhasználásával osztályozási problémákon. 35 valós adathalmazon végzett tesztelés és ennek statisztikai elemzése után azt találták, hogy a széleskörűen használt MSE-nek jobb alternatívája lehet az CE és az E_{Exp} , illetve azt is megállapították, hogy a tesztelésben használt adathalmazok esetén az SE és RE2 használata kevésbé eredményes.

A használt hibamértékek mellett az algoritmus egyéb részletei és paraméterei is meghatározóak lehetnek az ANN tanítás szempontjából. Rimer & Martinez (2006) egy új osztályozási feladatokra specializált modell mértéket vezetett be, az MSE-nél és CE-nél jobb teljesítmény elérése érdekében. Az új módszer egyik kulcspontja, hogy a tanítás során módosítja a modellhiba mértéket, mivel dinamikus állítja a kimeneti célértéket minden tanító adatpontra. Azt is megfogalmazták, hogy az általános differenciálható mértékek, például az MSE arra a feltételezésre támaszkodnak, mely szerint minták kimenetei az inherens Gauss-zaj által ellensúlyozottak, normális eloszlásúak a klaszter átlag körül. Megállapították, hogy más hibamértékek (mint CE) megfelelőbbek osztályozási probléma esetén. A jólismert benchmark adathalmazokon történő validáció alapján megfigyelhető volt, hogy az MSE és CE optimalizáló hálómodell közel azonos eredményt ér el, ezért az új modellnek csak a CE-vel való összehasonlítást prezentálták a rövidség kedvéért. Az eredmény, hogy a bevezetett módszer a 11-ből 10 adathalmazon nagyobb tesztelési pontosságot és szűkebb konfidencia intervallumot eredményez, mint a CE esetén, anélkül, hogy súlyfelejtés történne a hiba-visszaterjesztés (Back-Propagation - BP) során.

Ahogy fentebb látható, **sokféle információelméleti mérték létezik és az algoritmus részleteinek és paramétereinek módosítására is sok lehetőség** áll rendelkezésre. Sőt ezek még össze is vegyíthetőek, mint ahogy Heravi és Hodtani cikkeiben (Heravi & Hodtani, 2018a,b,c, 2019) is olvasható. Meghatározó és fontos eredményeket publikált Heravi & Hodtani (2018c) mind elméleti mind szimulációs szempontból. Tisztázták a kapcsolatot az MSE, Minimum Error Entropy (MEE, RE2-vel kapcsolatos) és a Maximális Correntrópia között (CorrE), kiemelve két fő sorrendet, hierarchiát befolyásoló tényezőt: a jel-zaj viszont (Signal-to-Noise Ratio (SNR)) és a hozzáadott zaj (incorporated noise) Gauss-eloszlástól való távolságát. Ennek a látens faktornak a mérésésre a Kullback-Leibler divergencia (KL) alkalmazható.

Rövidítések	
AI	Mesterséges Intelligencia (Artificial intelligence)
ML	Gépi Tanulás (Machine Learning)
ANN	Mesterséges Neurális Háló (Artificial Neural Networks)
BP	Back-Propagation
ITM	Információelmélet (Information Theory Measure)
MSE	(Átlagos) Négyzetes Hiba (Mean Squared Error)
CE	Cross-Entrópia (Cross-Entropy)
KL	Kullback-Leibler divergencia (Kullback-Leibler divergence)
J	Jensen divergencia (Jensen divergence)
JS	Jensen-Shannon divergencia (Jensen-Shannon divergence)
SE	Shannon entrópia (Shannon Entropy)
RE	Rényi entrópia (Rényi Entropy)
RE2	Másodrendű RE (quadratic RE, $\alpha = 2$)
MI	Kölcsönös Információ (Mutual Information)
E_{Exp}	(Silva-féle) Exponenciális Hiba (Exponential Error)
CorrE	Correntrópia (Correntropy)
MEE	Minimum Error Entropy
EXP	E_{Exp} for $\tau > 0$
ZED(M)	Zero Error Density (Minimisation)
RR	Osztályozási hibaráta (Recognition Rate)

1.1. táblázat. Az itt felsorolt rövidítések használatosak a dolgozat többi részében is.

"Egyszerű" Gaussos zajeloszlás esetén, alacsony SNR érték mellett, az MSE felülmúlja az MEE (és a CorrE-t), de magas SNR értéknél a teljesítményük hasonló. Cauchy zaj esetén az információelméleti mértékek (MEE és CorrE) szignifikánsan jobban teljesítenek az MSE-nél. Hasonlóan Laplace zajeloszlás esetén, ha a Laplace eloszlás úgynevezett skálázó paramétere nagy, akkor az információelméleti mértékek szignifikánsan túlteljesítik az MSE-t, viszont, ha ez a paraméter igazán kicsi, teljesítményük hasonló lesz. Ez a kutatás egyértelműen igazolja, hogy különböző mértékek fölénye változó lehet az elemzés körülményeitől, az adatterülettől, annak gyűjtési módszereitől, reprezentativitásától, a beépített zajtól, stb függően.

1.3. Információelméleti mértékek és a Levenberg-Marquardt algoritmus

Az irodalomkutatás egyik fontos szempontja volt, hogy milyen mértékeket használtak már korábban a Levenberg-Marquardt (LM) algoritmusban a neurális háló tanítása során. Képfelismerési feladatra hozott létre egy új módszert Thévenaz & Unser (2000), amiben a KL és LM algoritmus kombinációját valósítják meg. A cikkben leírják az ehhez szükséges algoritmikus

elemeket, mint a KL megfelelő deriváltjai és a Hesse mátrixbeli elhanyagolások. Tesztelési eredmények alapján azt a következtetést vonják le, hogy a az új módszer gyorsabb és pontosabb, mint az akkoriban (2000) ismert egyéb képfelismerő módszerek.

A képpárok rendszerezése vagy felismerése sok különböző gyakorlati alkalmazásban elvárás. Dowson & Bowden (2007) tudományos munkája az egyik népszerű információelméleti mértéket MI-t használja a LM algoritmusban neurális háló tanításra. Az MI méri a megosztott információt két jel között, ami a cikkbeli alkalmazási terület szempontjából az jelenti, hogy két kép (jelek) intenzitásának (amplitúdó) együttes eloszlásfüggvényét számítják ki. A szerzők ennek motivációjaként a konkrét alkalmazási területet említik meg. Az MI használata csak kevéssel több számítást igényel, mint az MSE, viszont számos előnye van az adott feladat szempontjából. Nevezetesen az MI tolerálja a nemlineáris kapcsolatokat is a képek intenzitása között, és a zajra nézve robusztus. A szerzők bebizonyították, hogy az ő képfeldolgozási feladatukra a bevezetett inverz-kompozíciójú, MI alapú mérték szignifikánsan túlteljesíti az MSE alapú technikákat, a modellpontosságot sebességet és stabilitást tekintve a tanítási folyamat során.

Thévenaz & Unser (2000) kutatása alapján, Panin & Knoll (2008) fejlesztett egy sablon követő (template tracking) eljárást. A cikkükben MI-t LM-ben alkalmazva és az SSE-hez (mint MSE) hasonlítva arra jutottak, hogy az új algoritmus robusztus és alkalmazható 3D textúrájú objektumokon.

Heravi & Hodtani (2016) egyik első cikkében LM algoritmust tanítottak, ahol hibafüggvényként a correntrópiát (Correntropy-CorrE) használták, a sűrűségfüggvényt Parzen-féle ablakmódszerrel közelítve. A cikkben más CorrE alapú algoritmusokkal összehasonlítva azt találták, hogy az újonnan létrehozott CorrE alapú LM gyorsabb és robusztusabb.

1.3.1. A mértékek "hierarchiája"

Az neurális hálózatok tanításában használt mértékek irodalmának áttekintése alapján **megállapítható, amit Heravi & Hodtani (2018c) is kimondott, valamit az alkalmazással kapcsolatos cikkek sugalltak (pl. Rimer & Martinez (2006); Silva et al. (2014)), hogy nem állapítható meg egyértelmű sorrend a mértékek performanciája között**.

Ezt szemlélteti a 1.2 táblázat is, ahol az irodalomkutatás során áttekintett irodalmakban leírt, a mértékek egymáshoz való viszonyáról szóló megállapítások kerültek összegzésre. A '+' jel szemlélteti, ha volt olyan jelentősebb eredmény egy cikkben, ami arra vonatkozott, hogy az oszlop fejlécében található mérték modellpontosságban felülmúlta az adott sorhoz tartozó

Modellpontosság	MSE	SE	CE	CorrE	RE(2)	MI	J	JS	E_{Exp}
MSE		++	∘ +++ ∘	++ ∘	+	+	∘	∘	+
CE	∘ + ∘	+					∘	∘	
CorrE	+ ∘								
RE(2)				+					
ZED(M)		+			+				
J	∘		∘						
JS	∘		∘						
E_{Exp}	+		+						
Összegzés	8	4	8	4	2	1	2	2	1

1.2. táblázat. A különböző mértékek fölénye a többihez viszonyítva a végső modellpontosság tekintetében a ma ismert irodalmak alapján. Az oszlopok fejléce mutatja a fölényben lévő mértéket, a sorban lévő mérték felett. '+' jel azt jelenti, ha valamely cikkben fontos eredmény mutatta az oszlop tetején lévő mérték jobb teljesítményét a sorban lévő mérték felett. '∘' jelzi, ha egy kutatásban hasonló eredményt értek el.

mértéket. A '∘' jelöli, ha az adott szerzők a hasonló teljesítményre vonatkozó konklúziókat vontak le. Ha egy cellába több '+' vagy '∘' jel került, az azt jelenti jelen esetben, hogy az adott mértékek viszonyával frekventáltabban foglalkoztak az irodalomban.

1.3.2. Különböző mértékek a tanító algoritmusokban

A korábbi bekezdések részletesen bemutatják azon tudományos műveket, amikben a neurális hálózat tanítására különböző mértékeket alkalmaztak. Jelen kutatás szempontjából viszont nemcsak a tanítómérték a lényeges, hanem maga a tanítóalgoritmus is. Hiszen ezen technikák széles tárháza áll a kutatók rendelkezésére akár csak a Backpropagation módszereket tekintve (pl. original, ADAM, sztochasztikus, stb.). A 1.3 táblázat bemutatja az egyes mértékek alkalmazását Backpropagation algoritmusokban, valamint külön kiemelten a LM algoritmusban, szemlélteti, hogy számos olyan mérték létezik, amit nem integráltak a LM algoritmusba, ami lehetséges tudományos rés.

Mértékek	BP	LM
MSE	Watrous (1992), Park et al. (1995) Erdogmus & Principe (2002) Silva et al. (2005b, 2008, 2014), Kline & Berardi (2005) Rimer & Martinez (2006), Dowson & Bowden (2007) Panin & Knoll (2008), Rady (2011a,b) Heravi & Hodtani (2018a,c,b, 2019)	Marquardt (1963)
SE	Erdogmus & Principe (2002), Silva et al. (2005b, 2014) Rady (2011b)	
CE	Watrous (1992); Park et al. (1995) Silva et al. (2005b, 2008, 2014) Kline & Berardi (2005); Rimer & Martinez (2006) Fontes et al. (2014), Nilsaz-Dezfouli et al. (2016)	
CorrE	Heravi & Hodtani (2016, 2018a,c,b, 2019)	Heravi & Hodtani (2016) Heravi & Hodtani (2018a)
RE	Rady (2011a), Silva et al. (2014)	
ZEDM	Silva et al. (2008, 2014)	
E_{exp}	Silva et al. (2008, 2014), Fontes et al. (2014)	
KL	Park et al. (1995)	
MI		Thévenaz & Unser (2000) Dowson & Bowden (2007) Panin & Knoll (2008)
J	Park et al. (1995)	
JS	Park et al. (1995)	

1.3. táblázat. Különböző mértékek és BP illetve LM-beli alkalmazásuk a szakirodalomban.

A LM-ban korábban nem használt mértékek közül az egyik a Silva et al. (2008)-féle E_{exp}, ami ígéretes választás lehet az általánosító képessége miatt.

2. fejezet

Új algoritmus egyidejű, statisztikai és információelméleti mértékek alapján történő tanulásra

Az irodalmi áttekintés 1 rávilágít arra, hogy a Levenberg-Marquardt algoritmus számos előnyös tulajdonsággal rendelkezik a gradiens módszerhez képest és alkalmas a neurális háló tanítására. Az is kitűnik, hogy a kutatók már a korábbiakban is kisebb-nagyobb sikerrel próbálkoztak különböző mértékek használatával. Vajon melyik lehet az a hibafüggvény, amit érdemes lenne az LM algoritmusban is használni? Erre egy ígéretes jelölt az Silva et al.- féle E_{Exp} egy lényegi módosítással. (Ennek az újonnan definiált mértékenek a LM-ba történő integrációját a szerző korábbi művei Szűcs (2020, 2021) más aspektusokból is vizsgálják.)

2.1. Jelölés

A 2.1 táblázatban bemutatott jelölések érvényesek a következőkben a kutatásban használt algoritmusok bemutatása során.

Jelölések		
$\mathbf{o}^l$		Az output vektor az l -dik rétegen
	l	$0, \ldots, L$
	$L-1$	Rejtett rétegek száma
	$\mathbf{o}^0$	Az MLP input vektora
	$\mathbf{o}^L$	Az MLP output vektora
$o^l_{n_l}$		A kimenet n_l-dik kimeneti csúcson, az l-dik rétegen
	n_l	$0, \ldots, N_l$
	N_l	Az output csúcsok száma az l-dik rétegen
$w^l_{n_{l-1},n_l}$		$o^{l-1}_{n_{l-1}}$-ből $o^l_{n_l}$-be vezető és súlya
	l	$1, \ldots, L$ (a 0-dik rétegbe nem mutatnak súlyok)
p		A pattern indexe
	p	$1, \ldots, P$
	P	A patternek száma
i		Az iterációk indexe
	i	$1, \ldots, I$
	I	Az iterációk száma
τ		Valós paraméter (tau)
e_{n_L}		$t_{n_L} - o^L_{n_L}$
	t_{n_L}	Elvárt kimenet (target) értéke
	$o^L_{n_L}$	Valódi kimenet (output) a kimeneti rétegen
σ		Szigmoid aktivációs függvény

2.1. táblázat. Jelölések a matematikai háttér bemutatásában.

2.1.1. Neurális háló (Multilayer Perceptron)

A dolgozatban a későbbiekben bemutatott tanító algoritmusok olyan MLP-kre definiáltak, amelyek szigmoidot használnak aktivációs függvényként a csúcsokban, és rétegenként bias taggal rendelkeznek.

- $o^l_0 = 1$: Bias-csúcs az l-dik rétegen,

- $w^l_{n_{l-1},0} = 0$: A bias nem kapcsolódik az előző réteghez. Jelen kutatásban használt neurális háló minden rétegére van külön bias érték, amik függetlenek egymástól.

A neurális háló tanításának első lépése az inicializált hálón output képzése adott bemeneti vektorra. Az l-dik réteg n_l indexű csúcsának kimeneti értéke:

$$o^l_{n_l} = \sigma\left(\sum_{n=0}^{N_{l-1}} w^l_{n_{l-1},n_l} \cdot o^{l-1}_{n_{l-1}}\right) \tag{2.1}$$

A kapott kimenet az MLP kimeneti csúcsain:

$$o^L_{n_L} = \sigma\left(\sum_{n=0}^{N_{L-1}} w^L_{n_{L-1},n_L} \cdot o^{L-1}_{n_{L-1}}\right) \tag{2.2}$$

A továbbiakban az fentebbi képletek eredményét tekintjük a hálón egy adott bemeneti vektorra kapott kimenetként az adott rétegen lévő adott indexű csúcson.

2.2. Abszolút értékes exponenciális hiba (E_{ExpAbs})

Ez az alfejezet részletesen bemutatja az E_{Exp}-et és ismerteti az optimalizálhatósági lehetőségit. Majd ezek ismeretében újradefiniálja azt az E_{ExpAbs} formájában.

2.2.1. Motiváció - Silva et al.-féle Exponenciális hiba (E_{Exp})

Ahogy már korábban is említésre került (1.2) az exponenciális hiba (E_{Exp}) egy olyan függvény, amely több más hibafüggvényt általánosít (Silva et al. (2008)). A következő bekezdések bemutatnak egy olyan úttörő tudományos eredményt, ami a TDK dolgozatnak is az egyik alappontját jelenti. Silva et al. megvizsgálták több hibamérésre alkalmas függvény hasznos tulajdonságait és ezek alapján létrehoztak egy integrált mértéket. A továbbiakban a fejezet áttekinti, hogy milyen mértékeket és hogyan intergál az E_{Exp}.

2.2.1.1. Átlagos négyzetes hiba (MSE)

Az adattudományi algoritmusokban, mint a ANN tanításban általánosan használt hibamérték az átlagos négyzetes hiba (MSE). Az elvárt ($\mathbf{t}$) és az aktuálisan kapott ($\mathbf{o}_L$) kimenet különbségének a négyzete.

$$MSE = \frac{1}{N}\sum_{p=1}^{P}\sum_{n_L=1}^{N_L} e^2_{n_L,p} \tag{2.3}$$

Marquardt (1963) már egy ehhez hasonló mérték minimalizálására építette fel az algoritmusát. Ő csak a négyzetes hibát használta a patternszámmal való normalizálás nélkül.

2.2.1.2. Cross-entrópia (CE)

Az itt látható Cross-entrópia képlet a Kullback-Leibler távolságból származtatható (Cover Thomas & Thomas Joy, 1991).

$$CE = \sum_{p=1}^{P} \left(\sum_{n_L=1}^{N_L} \left[-t_{p,n_L} \cdot log\left(o_{p,n_L}^{L} \right) \right] \right) \tag{2.4}$$

Silva et al. (2008) több irodalmi forrással is alátámasztja, hogy a "CE várhatóan pontosabban közelít alacsony valószínűségeket", mint az MSE.

2.2.1.3. Zero-error sűrűség minimalizálás (ZEDM)

A zero-error sűrűségfüggvény szintén Silva et al. (2005a)-hoz fűződő eredmény, amit az entrópia kritérium inspirált. Képlete a következő:

$$ZEDM = \sum_{p=1}^{P} \left(\sum_{n_L=1}^{N_L} \left[h^2 \cdot exp\left(-\frac{1}{2h^2} \cdot (t_{p,n_L} - o_{p,n_L}^{L})^2 \right) \right] \right) \tag{2.5}$$

ahol h a Gauss kernelfüggvény simasági paramétere. Fontos kiemelni, hogy a tanítási folyamat során a ZEDM maximalizálása a cél.

2.2.1.4. Exponenciális hiba, az általánosított mérték

Az előbbi három hibafüggvény gradiens viselkedésének megfigyelése alapján, (Silva et al., 2008) megalkották az új hibafüggvényt az exponenciális hibát.

$$E_{Exp} = \sum_{p=1}^{P} \tau \exp\left(\frac{\mathbf{e_p}^2}{\tau} \right) = \sum_{p=1}^{P} \tau \exp\left(\frac{1}{\tau} \sum_{n_L=1}^{N_L} e_{n_L,p}^2 \right) \tag{2.6}$$

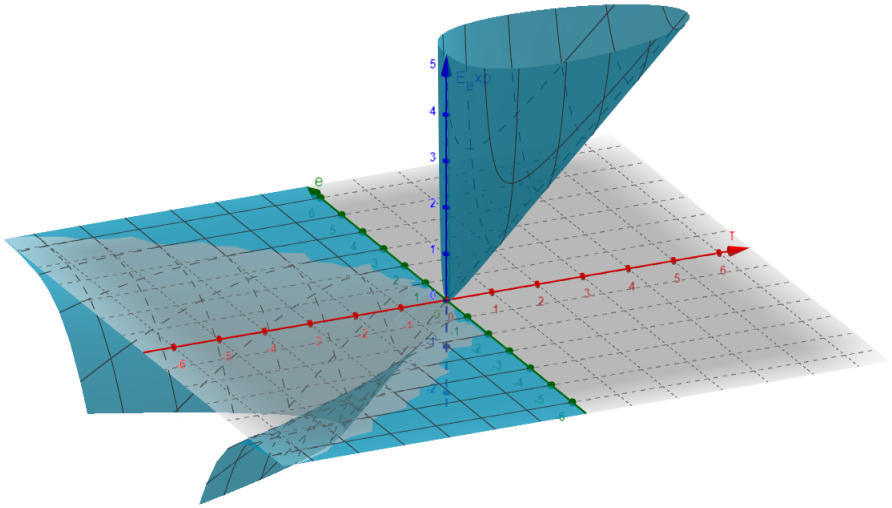
Megállapították, hogy a τ paraméter értékétől függően az E_{Exp}, MSE, ZEDM vagy CE szerint viselkedik, az alábbiak szerint:

- If $\tau > 0$ akkor CE,
- If $\tau < 0$ akkor ZEDM,
- If $\tau \to \infty$ akkor MSE.

A tesztelések során ezt a τ paramétert úgy választották ki, hogy vették a valós számok egy diszkrét, általában kifejezetten alacsony számosságú halmazát (pl. 5-10 db). Ezekre a lehetséges τ értékekre megvizsgálták az algoritmust, végül kiértékelve a legjobban teljesítőt választották az adott adathalmazon történő tanításhoz.

2.2.1.5. E_{Exp} az optimalizáló algoritmusban

Az E_{Exp} alakulásának megfigyelése egy pattern esetén a 2.1 ábrán látható felületet eredményezi, ami a τ-tól (piros tengely) és az e-től, mint a elvárt és a valódi kimenet közötti különbség (zöld tengely), függ.



2.1. ábra. err_p^{Exp} a τ (piros tengely) és az e, mint a elvárt és a valódi kimenet közötti különbség (zöld tengely)

Itt az látszik, hogy az E_{Exp} **minimalizálása nem feltétlenül minimalizálja a hibát** az adott adathalmazon. A problémát az okozza, hogy negatív τ esetén a függvény a végtelenbe tart, vagyis a minimalizáló algoritmusnak nem lesz érdeke minimalizálni az eltérést a valódi és az elvárt kimenet között (zöld, e tengely), elegendő, ha τ értékét folyamatosan csökkenti. Ha $\tau \to -\infty$, akkor a felületnek nem lesz optimum pontja, a függvény nem talál minimumhelyet, illetve a $-\infty$-be tart.

Az irodalomkutatás során áttekintett cikkekben (Amaral et al., 2013; Fontes et al., 2014; Silva et al., 2014) is, amiben később alkalmazták az E_{Exp}-et, az látható, hogy a τ paraméter negatív tartományát nem használták, vagy eleve csak pozitív tartományra definiálták.

2.2.2. Az abszolútértékes exponenciális hiba E_{ExpAbs}

A 2.2.1.5 bemutatta az E_{Exp} függvény modellhiba mérésben adódó problémáját. A TDK kutatás során ennek az akadálynak a megszüntetése a korábbiakban már bevált módszer alkalmazásával valósult meg Szűcs (2020). Az a következő képletekkel definiált abszolútértékes exponenciális hibafüggvényeket jelenti:

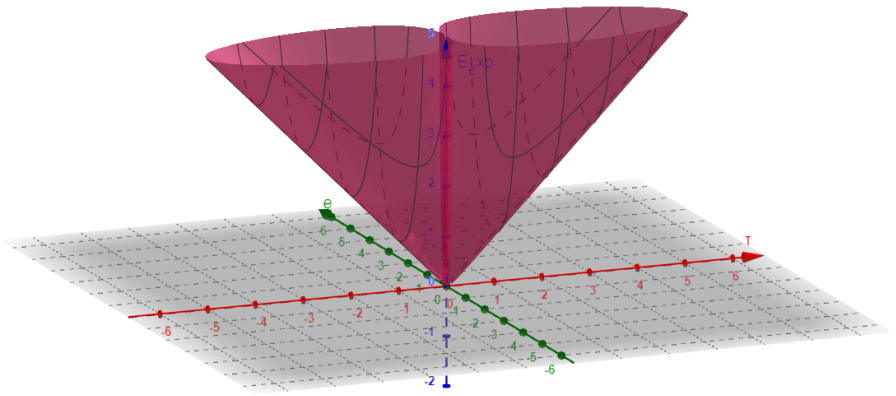
Egy mintaelemre:

$$err_p^{ExpAbs} := |\tau| \exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L} e_{n_L,p}^2\right) = |\tau| \exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L}\left(t_{n_L} - o_{n_L}^L\right)^2\right) \tag{2.7}$$

P darab mintaelemre:

$$E_{ExpAbs} := \sum_{p=1}^{P} |\tau| \exp\left(\frac{\mathbf{e_p}^2}{|\tau|}\right) = \sum_{p=1}^{P} |\tau| \exp\left(\frac{1}{|\tau|}\sum_{n_L=1}^{N_L} e_{n_L,p}^2\right) \tag{2.8}$$

A 2.2 ábrán szemléletesen is látszódik, hogy a felületen érdemes minimumhelyet keresni (vagyis az E_{ExpAbs}-ot minimalizálni).



2.2. ábra. E_{ExpAbs} függ a τ-tól (piros tengely) és az e-től mint az elvárt és a valódi output között (zöld tengely)

Ahogy a 2.2.1.4 szakaszban is be lett mutatva, $\tau < 0$ esetén volt az E_{Exp} hibafüggvény viselkedése a ZEDM-hez hasonló. Az E_{ExpAbs} használatával, viszont a τ nem tekinthető negatívnak, vagyis ezáltal a *képletből "elveszett" egy információelméleti mérték*. Viszont az **MSE (nagy τ érték esetén) és CE (kis τ érték esetén) szerű kettő viselkedés továbbra is meg-**

maradt, ezáltal továbbra is megőrizve a lehetőséget a statisztika és az információelmélet együttes figyelembevételére az E_{ExpAbs} használatával a tanítási algoritmusban.

2.3. Levenberg-Marquardt algoritmus és az E_{ExpAbs}

Jelen kutatásban az általánosan használt MSE helyett az E_{ExpAbs} került az LM algoritmusba hibafüggvényként. **Az algoritmusban egy különösen fontos kulcsmódosítás, hogy a neurális háló tanítása során adaptálódik a τ paraméter.** Mindez úgy valósul meg, hogy a Levenberg-Marquardt algoritmusban nem csak a súlyokat tekintjük változó paraméternek, hanem a τ értéket is, így a változtatása a háló súlyainak változtatásával együtt valósul meg. Ez az optimalizáláshoz használt Jacobi mátrixban definíciótól függően egy plusz sor vagy oszlop bevezetését jelenti.

2.3.1. Kismátrixos módszer

A feladatnak szerző szakdolgozatában megvalósított Szűcs (2020) lehetséges megoldása a Heravi & Hodtani (2016)-féle kismátrixos módszerre támaszkodik. Így jelen feladatban az alábbiak szerint épül fel a 1.1.2.2 szakaszban bemutatott Jacobi-mátrix.

$$J_i = \begin{bmatrix} \frac{\partial err_1^{ExpAbs}}{\partial w_{0,1}^1} & \frac{\partial err_2^{ExpAbs}}{\partial w_{0,1}^1} & \cdots & \frac{\partial err_P^{ExpAbs}}{\partial w_{0,1}^1} \\ \frac{\partial err_1^{ExpAbs}}{\partial w_{1,1}^1} & \frac{\partial err_2^{ExpAbs}}{\partial w_{1,1}^1} & \cdots & \frac{\partial err_P^{ExpAbs}}{\partial w_{1,1}^1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial err_1^{ExpAbs}}{\partial w_{N_1,1}^1} & \frac{\partial err_2^{ExpAbs}}{\partial w_{N_1,1}^1} & \cdots & \frac{\partial err_P^{ExpAbs}}{\partial w_{N_1,1}^1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial err_1^{ExpAbs}}{\partial w_{N_1,N_L}^{N_L}} & \frac{\partial err_2^{ExpAbs}}{\partial w_{N_1,N_L}^{N_L}} & \cdots & \frac{\partial err_P^{ExpAbs}}{\partial w_{N_1,N_L}^{N_L}} \\ \frac{\partial err_1^{ExpAbs}}{\partial \tau} & \frac{\partial err_2^{ExpAbs}}{\partial \tau} & \cdots & \frac{\partial err_P^{ExpAbs}}{\partial \tau} \end{bmatrix}_{\text{(RawNr.:(\#Weights}(W)\text{+1)}\times\text{ColumnNr.:\#Patterns(P))}} \tag{2.9}$$

A 2.9 elemei az egyes patternek esetén kapott output vektorokra számolt abszolút értékes exponenciális hibák (err_1^{ExpAbs}) súlyok és τ szerinti deriváltjai.

Például egy háromrétegű neurális hálón az $p = 1$-es indexű pattern esetén kapott hiba (err_1^{ExpAbs}) visszaterjesztése az 0. réteg 1-es indexű csúcsából (o_1^0) az 1. réteg 1-es indexű

csúcsába (o_1^1) mutató súlyra ($w_{1,1}^1$), vagyis a err_1^{ExpAbs} $w_{1,1}^1$ szerinti parciális deriváltja a következőképpen áll elő.

$$\frac{\partial err_1^{ExpAbs}}{\partial w_{1,1}^1} = \frac{\partial err_1^{ExpAbs}}{\partial o_1^2}\frac{\partial o_1^2}{\partial o_1^1}\frac{\partial o_1^1}{\partial w_{1,1}^1} + \frac{\partial err_1^{ExpAbs}}{\partial o_2^2}\frac{\partial o_2^2}{\partial o_1^1}\frac{\partial o_1^1}{\partial w_{1,1}^1} \tag{2.10}$$

Ezen példa alapján is látható, hogy a Jacobi mátrix elemeinek kiszámításához általánosságban (3-nál több rétegű háló esetén is) az alábbi parciális deriváltak képletének ismeretére van szükség a hiba-visszaterjesztéshez (back-propagation).

err_p^{ExpAbs} deriváltja a τ-ra nézve:

$$\frac{\partial err_p^{ExpAbs}}{\partial \tau} = sign(\tau) \cdot \exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L} e_{n_L,p}^2\right) \cdot \left(1 - \frac{1}{|\tau|}\sum_{n_L=0}^{N_L} e_{n_L,p}^2\right) \tag{2.11}$$

err_p^{ExpAbs} deriváltja az output (L-dik) réteg csúcsain kijövő outputok szerint:

$$\frac{\partial err_p^{Exp}}{\partial o_{n_L}^L} = -2 \cdot \left[\exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L} e_{n_L,p}^2\right)\right] \cdot e_{n_L,p} \tag{2.12}$$

A kimeneti réteg adott csúcsán kijövő output deriváltja az előző réteg adott outputja szerint.

$$\frac{\partial o_{n_L}^L}{\partial o_{n_{L-1}}^{L-1}} = \sigma\left(o_{n_L}^L\right) \cdot \left(1 - \sigma\left(o_{n_L}^L\right)\right) \cdot w_{n_{L-1},n_L}^L \tag{2.13}$$

Ugyanez általánosítva:

$$\frac{\partial o_{n_l}^l}{\partial o_{n_{l-1}}^{l-1}} = \sigma\left(o_{n_l}^l\right) \cdot \left(1 - \sigma\left(o_{n_l}^l\right)\right) \cdot w_{n_{l-1},n_l}^l \tag{2.14}$$

l-dik rétegen lévő kimenet(csúcs) deriváltja a csúcshoz közvetlenül kapcsolódó súly szerint.

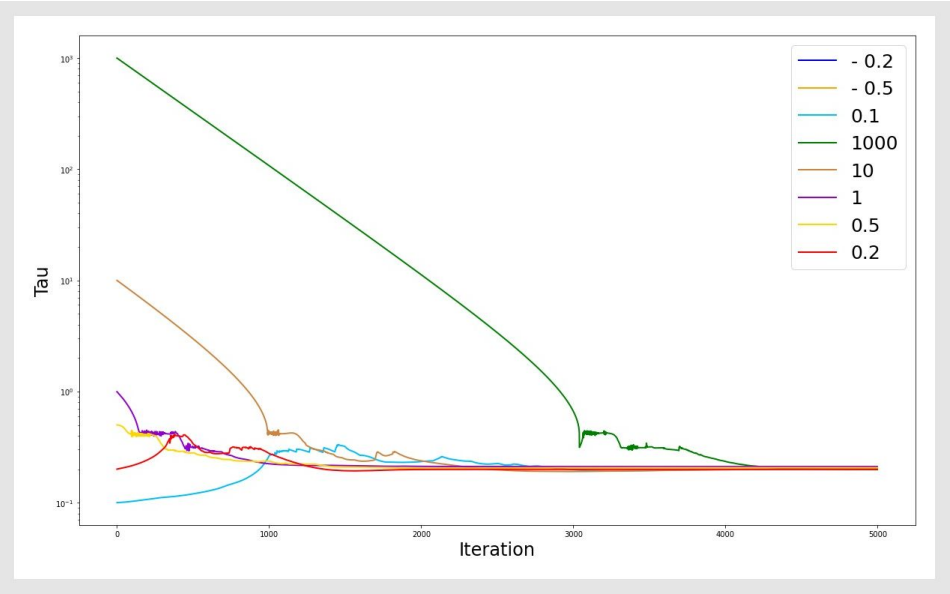
$$\frac{\partial o_{n_l}^l}{\partial w_{n_{l-1},n_l}^l} = \sigma\left(o_{n_l}^l\right) \cdot \left(1 - \sigma\left(o_{n_l}^l\right)\right) \cdot o_{n_{l-1}}^{l-1} \tag{2.15}$$

2.3.2. Módszerek a stabilabb és gyorsabb futás érdekében

A LM és az E_{ExpAbs} fentebb leírtak szerinti kombinációja konvergáló algoritmust eredményezett, ami sebességben és stabilitásban még fejlesztést igényelt. Erre a problémára két további módszer is integrálásra került: a momentum módszer (Rumelhart et al. (1986)) és a SuperSAB Tollenaere (1990). **Ezek kombinációjával módosított LM algoritmus ebből a szempontból is újdonságnak számít.**

2.3.2.1. Momentum módszer

A 2.3 ábra kiemeli a tipikus karakterisztikáját a τ változását ábrázoló görbének a tanítás alatt. Jól reprezentálja, hogy a tanítási folyamat végén a τ értékek (majdnem) ugyanazok adott adathalmazon (pl. az Írisz adathalmazon kb. 0.2).



2.3. ábra. τ érték változása a neurális hálózat tanítása alatt különböző kezdeti értékekről indítva. Az ábrán a függőleges tengely logaritmikusan skálázott.

A momentum módszer (Viharos (1999)) bevezetésének apropóját az adta, hogy a tanulás során a τ paraméter a kezdeti lépésekben egy hosszabb közel lineáris görbe (logaritmikus skálán) mentén csökkent. Ennek a lineáris szakasznak a rövidítése céljából merült fel a momentum módszer ötlete, ami ez alábbiak szerint került bevezetésre (csak) a τ paraméterre vonatkozóan.

$$\tau_{k+1} = \tau_k + \alpha\Delta\tau^{elozo} \tag{2.16}$$

A kifejezés szerint az τ értéke a következő iterációban az előző τ módosítás irányát is figyelembe veszi. A momentum módszer alkalmazásával módosított algoritmus így egy új α paraméterrel bővült.

2.3.2.2. 'SuperSAB'

A 'SuperSAB' egy már tradicionálisnak számtító gyorsítási módszer, ami a gradiens alapú technikák esetén használható. A Tollenaere (1990) alapötletének három komponense:

- Az úgynevezett η (eta) paramétert szorzóként rendeli a modell súlyok deriváltjaihoz, következésképpen a tanulási lépés nagysága az η és a deriváltak szorzatától függ.

- *Egyedi η értéket alkalmaz minden súlyhoz.* Tehát minden paraméternek saját szorzója van.

- *Ennek az újszerű η szorzónak az értéke dinamikusan adaptálódik az előző tanulási lépés iránya és a derivált iránya között fennálló reláció alapján.* Ha ezek az irányok megegyeznek (vagyis a tanítási lépések folyamatosan ugyanabba az irányba haladnak) az η szorzó enyhén növekszik, de amikor ez az irány ellentétes (vagyis a tanulás megváltoztatja az irányát), akkor az η szignifikánsan csökken. A kismértékű növelés és nagymértékű csökkentés fontos ebben a gyorsítási technikában. Így amikor a tanítás során a súlyfrissítések folyamatosan azonos irányban realizálódnak (folyamatos súlynövekedés vagy csökkenés) az η-val való szorzás növeli a tanítás során megtett lépést, ami az egyszerű derivált alapján volt számolva. Ha a súlyfrissítés iránya fluktuációt mutat, akkor az η szorzó csökkentésével az algoritmus kis, biztonságos lépéseket tesz az aktuális paraméterpont körül.

Ennek a gyorsítási trükknek a részleteit Tollenaere (1990) írja le, aminek az egyik legfőbb hatása, hogy 1-2 nagyságrenddel nagyobb tanulási sebességet eredményez, mint a momentum módszer.

2.3.2.3. A momentum módszer és a SupreSAB új kombinációja a Levenberg-Marquardt algoritmusban

Eme két gyorsítási technikának a Levenberg-Marquardt algoritmusba illesztése nem feltétlenül egyértelmű feladat, mivel a LM tanítás is egyfajta gyorsítási megoldás, valamint fennáll annak a kockázata, hogy az integrált megoldásban kioltják egymás gyorsító hatását a különböző módszerek.

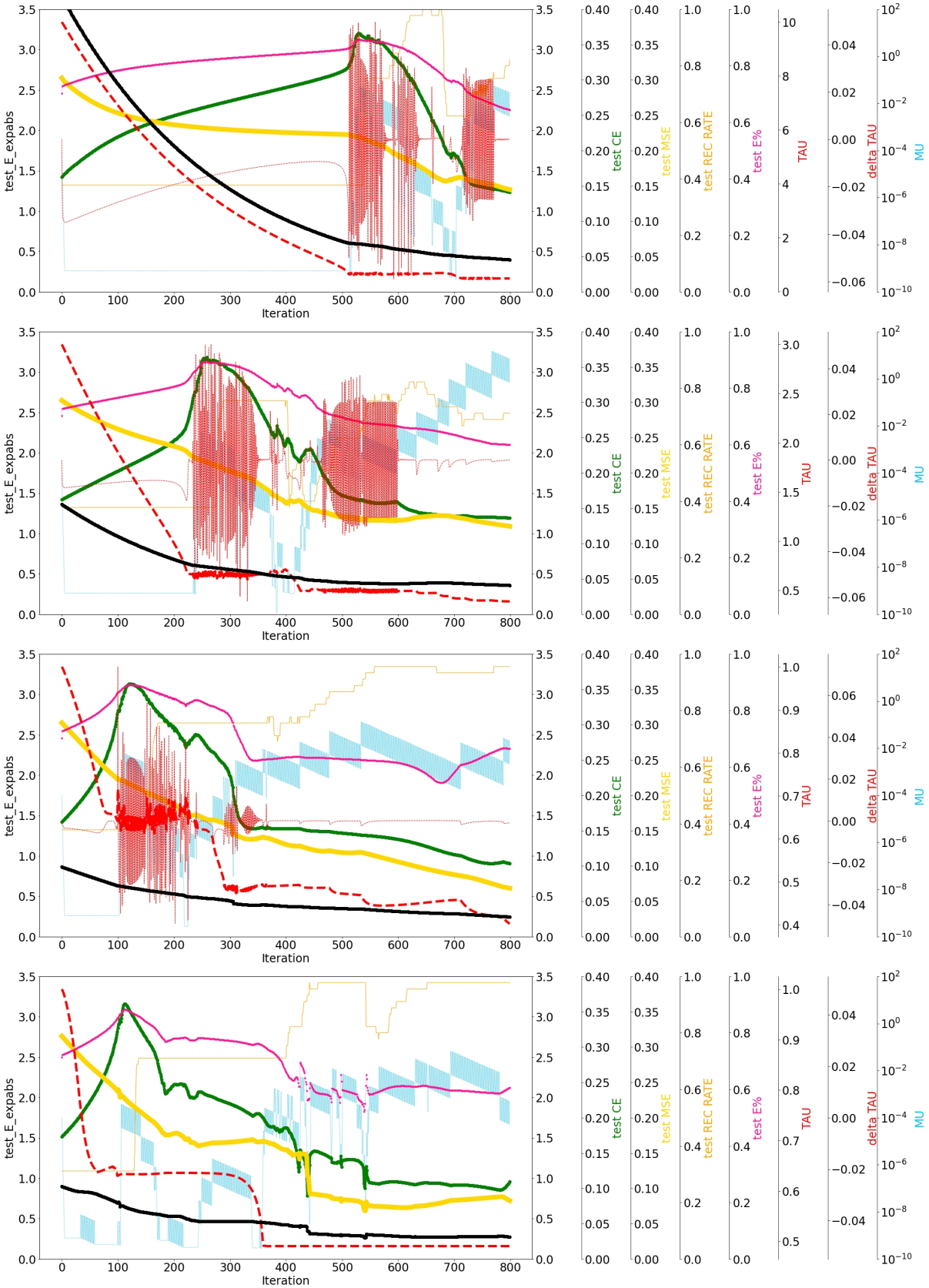
Az új algoritmusban a momentum és a SuperSAB módszer is csak a τ paraméter változását befolyásolja. Természetesen a módszereknek nem csak a sebességre nézve lehetnek pozitív hatásai, hanem a konvergenciára, stabilitásra, stb. is.

A végső algoritmusban tehát a következő technikák alkamazásával áll elő:

- Levenberg-Marquardt algoritmus a neurális hálózat minden súlyára és a τ paraméterre

- SuperSAB gyorsítási módszer a τ paraméterre

- momentum gyorsítási módszer a τ paraméterre



2.4. ábra. Az első három grafikon a csak momentum módszeres gyorsítást használó algoritmust szemlélteti ugyanazon a hálóstruktúrán futattva különböző kezdeti τ paraméterek (10, 3, 1) esetén.Az utolsó pedig a SuperSAB-bal is kombinált algortimus a $\tau = 1$ induló értékkel.

2.4 ábra szemlélteti az alkalmazott gyorsítási módszerek működését a tanítási folyamat során ugyanazon a hálóstruktúrán különböző τ értékekről indítva. Látható, hogy a különböző tanítási beállítások hasonló karakterisztikájú tanulási görbéket eredményeznek, de jelentősen gyorsabb lefutásúak alacsonyabb τ érték esetén. Az is megfigyelhető, hogy a módszerek kombinációjával (LM, momentum, SupreSAB) stabilabb és sokkal gyorsabb lesz a tanítási folyamat, kisebb oszcillációval, mint csak momentum módszer használata esetén.

3. fejezet

Eredmények

Ebben a fejezetben az előzőekben bemutatott új algoritmus tesztelésének körülményiről és a tesztelésből levont következtetésekről lesz szó.

3.1. Tesztelési körülmények

A kísérletek egy rejtett rétegű Multi Layer Perceptron-on futottak, kivéve a 3.2.5 fejezetben részletesen bemutatottat.

A kiértékelés a UCI Repository-ból származó 13 különböző banchmark adathalmazon történt a τ paramétert nem variáló és ezáltal gyorsítást sem tartalmazó, E_{ExpAbs}-ot használó Levenberg-Marquardt algoritmus (továbbiakban röviden *fix változat*) és a τ-t dinamikusan változtató kombinált módszerrel (momentum és SuperSAB) gyorsított LM algoritmus (továbbiakban röviden *dinamikus változat*) különböző szempontok szerinti összevetésével. Az adathalmazok részletes bemutatása 3.1 táblázatban olvasható.

Elsősorban *osztályozási feladatokra esett a választás, mert Silva et al. eredetileg osztályozási feladatokra fejlesztette az E_{Exp}-et (CE miatt).* Az adathalmazok tanító és teszthalmazra lettek felosztva 70:30 arányban. Mindkét adathalmazon legalább 30 különböző random inícializált hálón és újragenerált train-test felosztáson lett futtatva az új algoritmus mindkét verziója a 16 különböző induló τ érték mellett. A futtatás során korai leállási feltétel (early stopping) szerint állt meg az adott háló tanítása - ahol az early stopping paraméter 200 volt-, avagy maximálisan 5000-es tanító lépésszámig futott.

A momentum módszer α paramétere 0.1-nek lett választva, a SuperSab paraméterek pedig $\eta_+ = 1.05, \eta_- = 0.5$ (Viharos (1999)).

	Iris	Linnerud	Wine	Diabetes	Boston	Breast cancer	Ecoli	Glass
# Minta	150	20	178	442	506	569	327	214
# Input	4	3	13	10	13	30	5	9
Input típus	folytonos (4)	folytonos (3)	folytonos (13)	diszkrét (1) folytonos (9)	diszkrét (1) folytonos (12)	folytonos (4)	folytonos (5)	folytonos (9)
Output típus	diszkrét	folytonos	diszkrét	folytonos	folytonos	folytonos	diszkrét	diszkrét
# Osztály	3	-	3	-	-	2	5	6
Osztály eloszlás	50, 50, 50	-	59, 71, 48	-	-	217, 357	143, 77, 35 20, 52	70, 76, 17 13, 9, 29
# Output	1	3	1	1	1	1	1	1
Feldat típus	osztályozás	multi-output regresszió	osztályozás	regresszió	regresszió	osztályozás	osztályozás	osztályozás
# Csúcsok a rejtett rétegen	7	6	14	11	14	15	10	15
Feladat	Iris növények osztályozása a virágaik adatai alapján	Fitneszklubba járók fiziológiai értékeinek prediktálása	Borok osztályozása kémiai összetételük alapján	Betegség előrehaladásának prediktálása 1 év elteltével	Házak értékeinek előrejelzése	Rákos sejtek osztályozása képek alapján	Fehérje lokalizációs helyek prediktálása	Üveg típusok osztályozása kémiai összetételük alapján

3.1. táblázat. A kiértékelés során használt adathalmazok legfontosabb jellemzői Az adatok forrása a UCI Repository Dua & Graff (2017).

	Ionosphere	Blood	Parkinson	Thyroid	Vowel
# Minta	351	748	195	215	990
# Input	34	4	21	5	10
Input típus	folytonos (34)	folytonos (4)	folytonos (21)	folytonos (5)	folytonos (10)
Target típus	diszkrét	diszkrét	diszkrét	diszkrét	diszkrét
# Osztály	2	2	2	3	11
Osztály eloszlás	225, 126	178, 570	147, 48	150, 35, 30	90, ..., 90
# Output	1	1	1	1	1
Feladat típus	osztályozás	osztályozás	osztályozás	osztályozás	osztályozás
# Csúcsok a rejtett rétegen	15	6	15	8	15
Feladat	Ionoszférában lévő szabad elektronok osztályozása	Adott személy ad-e vért 2007 márciusában	Egészséges és parkinsonos emberek elkülönítése	Páciensek pajzsmirigy működésének előrejelzése	Különböző beszélőktől származó hangok felismerése

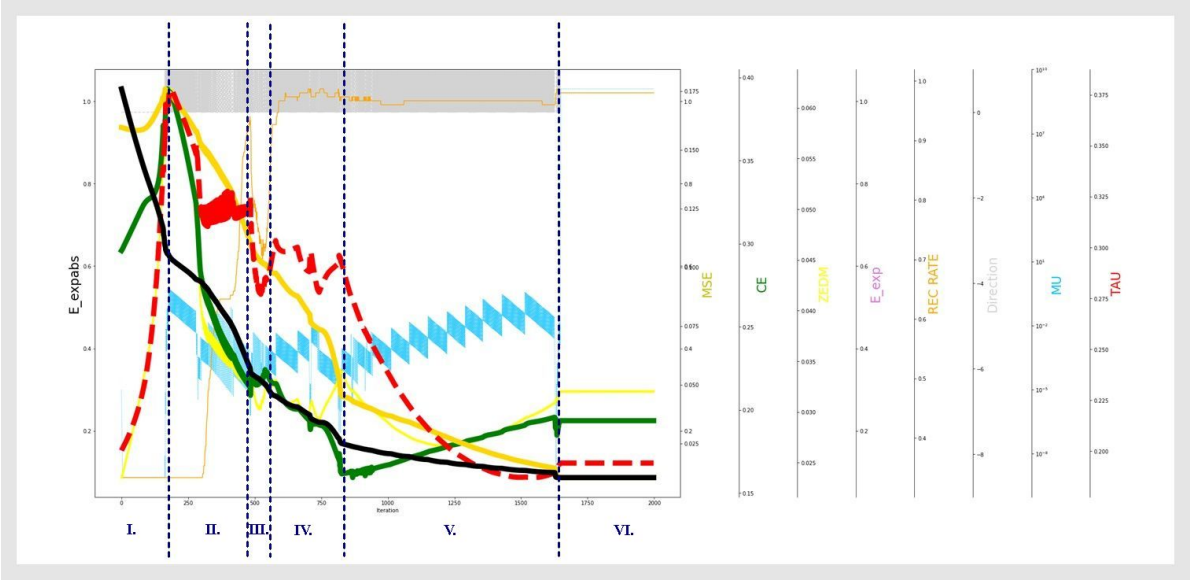
3.2. Futtatási eredmények

Az eredmények kétféle összehasonlítási módszerből származnak, egyrészt a tanulási görbék alakulásának összevetéséből, másrészt az egyes esetekben megtalált, vélhetően optimum pontban (early stopping pont) a futtatásban mért különböző mértékek boxplotos összehasonlításából.

A tanítás során különböző modellhibát mérő értékek kerültek rögzítésre a paraméterek pillanatnyi értékei mellett. A tanítási folyamatot ábrázoló grafikonokon a piros szaggatott vonal jelöli a τ értékét, a fekete az E_{ExpAbs}-ot, a zöld a CE-t, a vastag aranysárga az MSE-t, a vékony naracssárga az osztályozási hibarátát (Recognition Rate), a vékony kék pedig a μ paraméter változását.

3.2.1. Kismátrixos tanulási görbék

A kismátrixos módszer esetén 3.1 ábrán látható módon alakulnak a tanulás görbék és azok fázisai.



3.1. ábra. Tipikus tanulási fázisok kismátrixra az iris adathalmazon bemutatva. Más adathalmazra is hasonlóan néz ki.

A tanulási fázisok különbözőek, attól függően, hogy az algoritmus CE vagy MSE irányú, inkább információelméleti, vagy inkább statisztikai mérték szerint halad a tanulás.

I. Ebben a fázisban az MSE értéke és a CE együtt növekednek a τ paraméter növekedésével. Ez egy érdekes kezdőfázis, amiben mind a statisztikai, mind az információelméleti modellhiba romlik, az E_{ExpAbs} értéke pedig rendületlenül csökken (konvergál). Úgy

néz ki, hogy a modellnek egy olyan pozícióba kell magát "rontania", ahonnan sikeres tanítási folyamatot indíthat.

II. Az MSE és a CE is csökken, míg a τ oszcillál. Ebben a fázisban a statisztikai és információelméleti modell szerint is javul a teljesítmény. A τ oszcillációja azt jelenti, hogy az algoritmus egy aktív, felfedező szakaszban van és gyorsan konvergál.

III. A CE növekszik, az MSE csőkken, habár a minimalizálandó függvény inkább az MSE mint a CE irányát preferálja. Ebben a fázisban τ elkezdi erősíteni a CE-t, de egy rövid meredek növekedés után újra csökken.

IV. Az MSE és a CE is csökken, a τ kb 200 iteráción keresztül oszcillál, ami nagyon hasonlónak tűnik a II. fázisbelihez. Ennek a fázisnak a végén a μ átlagban csökkenést mutat, ami az jelenti, hogy az algoritmus Gauss-Newton irányba tolódik el, vagyis a gyorsabb konvergencia felé.

V. Ez ismét egy különösen érdekes része a tanulásnak, mivel az MSE és a CE ellentétes irányba halad. A τ paraméter folyamatosan csökken, ami arra utal, hogy az optimalizálódó hibafüggvény inkább CE-hez hasonló viselkedésű.

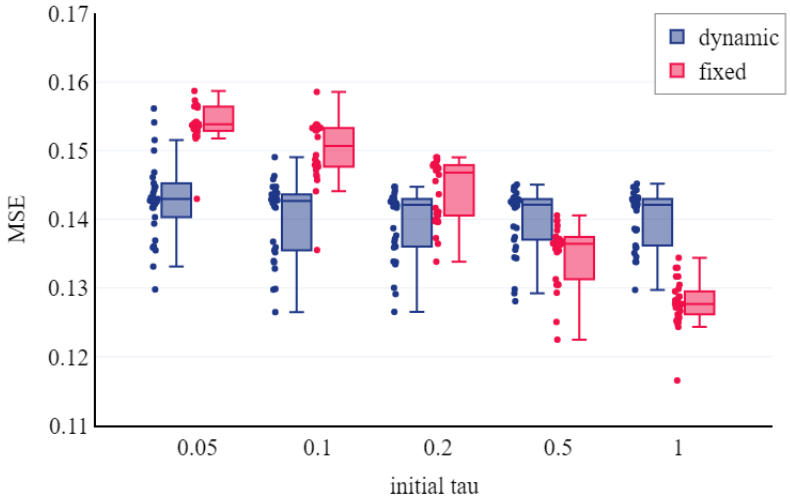
VI. Mindegyik hiba és paraméter konstanssá válik. Az algoritmus csak gradiens irányba megy, ami az tipikus végső fázis az LM használata esetén. Az E_{ExpAbs} nem csökken tovább, mivel elérte a minimumot.

3.2.2. Az új algoritmus teljesítménye modellpontosság szempontjából

A kisebbb és a nagyobb kezdeti τ értékekere is a dinamikus algoritmus ért jobb eredményeket, mint a fix. Az alábbi bekezdésekben a modellpontosságot a végső (optimális) MSE és CE mértékek szemléltetik. Fontos, az alábbi ábrákon az x tengely nem linárisan, hanem diszkréten skálázódik a τ értékek szerint, mivel a különbség az egyes esetek között ilyen módon jól reprezentálható.

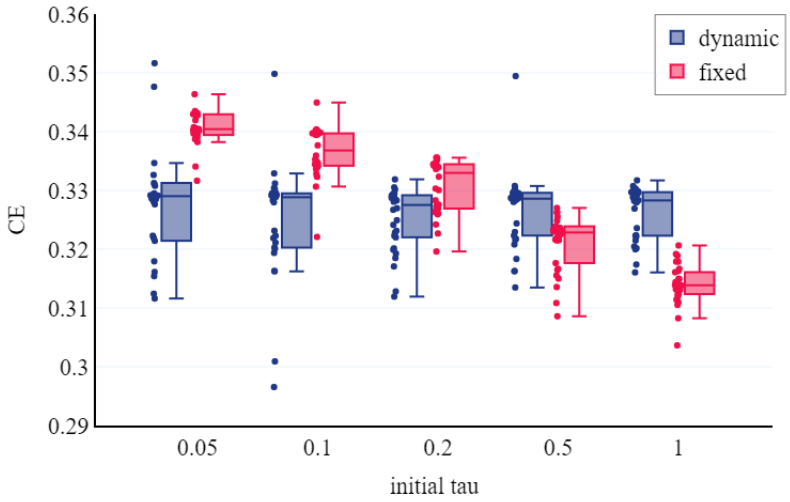
3.2.2.1. Átlagos modellpontosság kis τ értékekre

MSE: A 3.2 ábra azt mutatja, hogy az egyes kis τ-nkénti medián MSE értékek sokkal kisebbek az újonnan definiált gyorsított, dinamikus algorimtus esetén, mint a fix verzióra. Ez tehát az mutatja, hogy *kis τ értékekre az új dinamikus algoritmus pontosabb modellt ér el, mint a fix τ alapú, E_{ExpAbs}-val tanított LM*.



3.2. ábra. Kis τ értékekre a dinamikus algoritmus (kék) sokkal kisebb medián MSE értékeket eredményez, mint a fix (piros) változat. Az a grafikon a Blood adathalmazon szemlélteti az eredményt, de a többi tesztelt adathalmazra is hasonló ábra adódik.

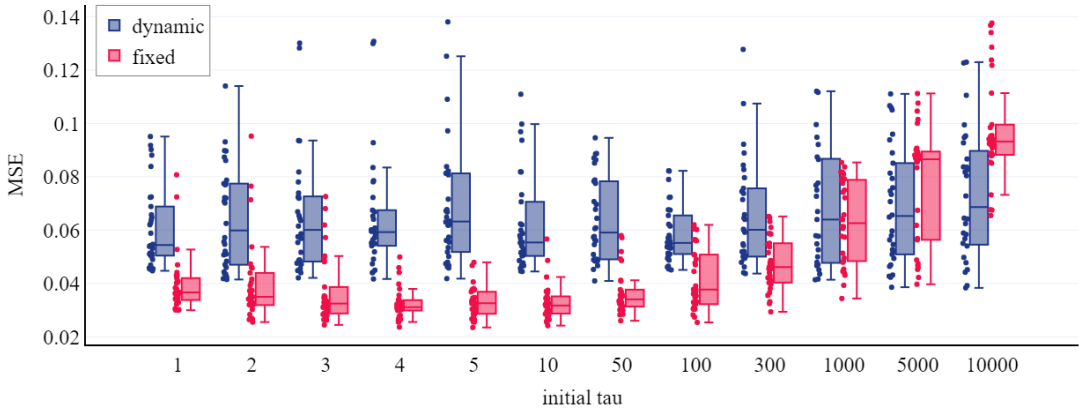
CE: Hasonló viselkedés figyelhető meg a CE esetén a 3.3 ábrán. A medián értékek a dinamikus tanulási folyamat végén sokkal kisebbek, mint a fix változat esetén alacsony τ értékekre. Silva et al. (2008) cikkükben említik, hogy $\tau > 0$ estén az E_{Exp} CE-hez hasonlóan viselkedik, habár az aktuális eredmények azt mutatják, hogy ez határ nem 0 (hanem kissé magasabb). A $\tau > 0$ futtatásokat tekintve az eredmények két részre oszthatók. Az látszik, hogy *a fix τ-s algoritmus csak a 0.1-100 τ érték esetén működik jól, a dinamikus változat viszont minden kezdeti τ érték (kissebbekre is) megfelelő megoldást talál.*



3.3. ábra. Kis τ értékekre a dinamikus gyorsított algoritmus (kék) alacsonyabb medián értékek elérésére képes, mint a fix τ-s változat(piros).Az a grafikon a Blood adathalmazon szemlélteti az eredményt, de a többi tesztelt osztályozási adathalmazra is hasonló ábra adódik.

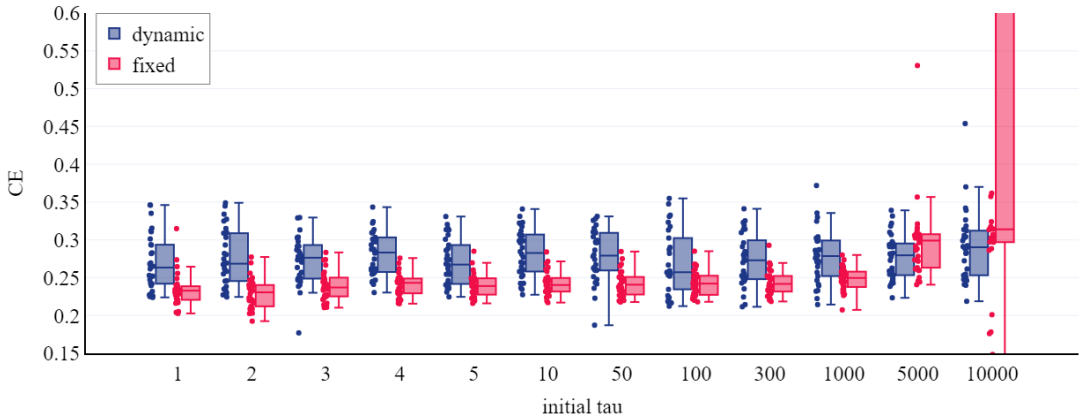
3.2.2.2. Átlagos végső modell pontosság nagy τ értékekre

MSE: A 3.4 képen látható, hogy magasabb innduló τ értékekre (nagy τ érték azt jelenti, hogy főként MSE-hez hasonló tanulást végez a modell) a dinamikus algoritmus jobb teljesítményt tud elérni, mint a fix változat. Látható, hogy egy bizonyos szakaszon a τ értékekre a fix τ-s változat alacsonyabb MSE értékeket képes elérni, de nagyobb τ értékek esetén ez a medián romlik. Silva et al. (2008) leírják, hogy amikor a $\tau \to \infty$ az E_{Exp} hiba mérték hasonlóan viselkedik az MSE alapú tanuláshoz. Jelen kutatás eredményeit figyelve ez nem teljesen így működik, az ellenkezője tapasztalható. Ez felveti annak a lehetőségét, hogy létezik egy optimális τ érték/intevallum az adott adathalmazra. Ez viszont a tanítási folyamatot megelőzően nem ismert, így a korábbi irodalomban bemutatott fix τ-s megoldás ezt nem találja meg, míg a TDK-ban bemutatott viszont igen.



3.4. ábra. Nagy τ értékekre a dinamikus algoritmus (kék) alacsonyabb medián MSE értékeket ér el, mint a fix (piros) változat. Az ábra az Ecoli adathalmazon elért eredményeket szemlélteti, de a többi tesztelt osztályozási adathalmazra is hasonló eredmény adódik.

CE: Kevésbé látványos ez az összefüggés CE-t vizsgálva, de szintén megfigyelhető a magasabb τ értékekesetén. A neurális háló végső állapotában alacsonyabb CE értéket ér al a dinamikus algoritmus esetén, ami azt jelenti, hogy sokkal pontosabb ebben a τ tartományban. (3.5 ábra)



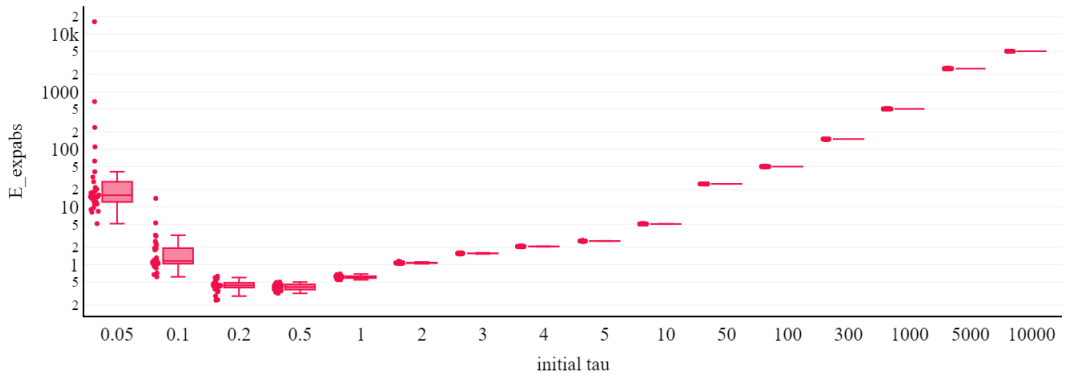
3.5. ábra. Magas τ értékekre a dinamikus algoritmus (kék), alacsonyabb medián CE értékeket ér el, mint a fix változat. Az ábra a Parkinson adathalmazon elért eredményeket szemlélteti, de a többi tesztelt osztályozási adathalmazra is hasonló eredmény adódik.

3.2.3. Az új algoritmus teljesítménye stabilitás/robusztusság szempontjából

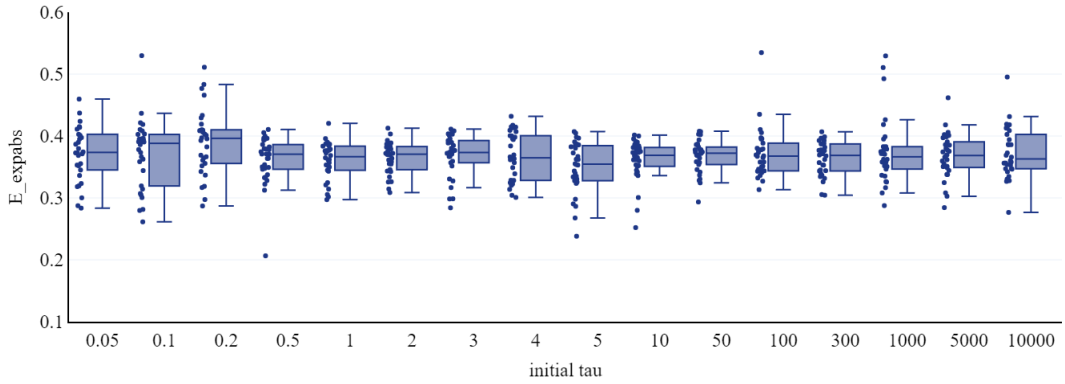
A dinamikus algoritmus független az induló τ értékétől a különböző modellpontosságot mérő mértékek esetén, míg ez a fix τ-s vátozatra nem igaz.

3.2.3.1. A modellpontosságok átlaga a különböző kezdeti τ értékek esetén is ugyanazok

E_{ExpAbs}: A 3.7 bemutatja, hogy a végső modell pontosságát mérő E_{ExpAbs} értékek majdnem ugyanazok a tanítási folyamat végén, következésképpen függetlenek a kezdeti τ értéktől, ami egy fontos jellemzője a dinamikus algoritmusnak. (Belátható a fix változat τ függősége (3.6ábra), ahol a tanulás során ez a paraméter nem változik.)

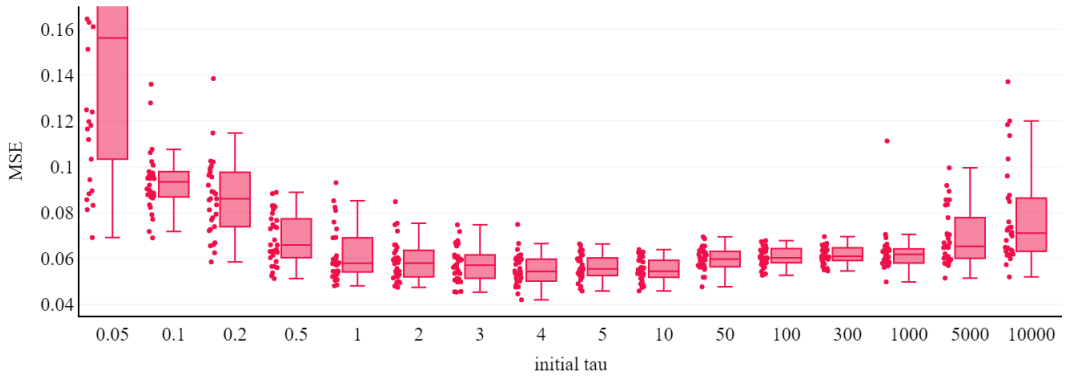


3.6. ábra. Végső E_{ExpAbs} medián értékei τ-nként, logaritmikus y tengellyel ábrázolva. Az ábra reprezentálja, hogy a fix verzió nagyban függ a tau értékétől. Ez a kép az Ionosphere adathalmaz eredményi alapján készült, de hasonló viselkedés figyelhető meg a többi osztályozási feladatra is.

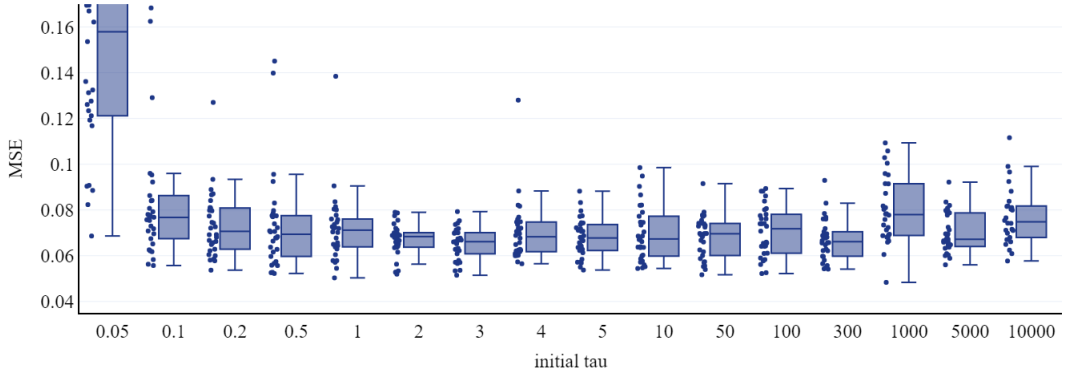


3.7. ábra. A végső E_{ExpAbs} értékek mediánjai minden *tau* értékre szinte ugyanazok, ami mutatja a dinamikus algoritmus τ függetlenségét. Ez a kép az Ionosphere adathalmaz eredményi alapján készült, de hasonló viselkedés figyelhető meg a többi osztályozási feladatra is.

MSE: A fix (3.8 ábra) és dinamikus (3.9 ábra) algoritmus összehasonlítása a végső MSE értékek mediánjait tekintve azt mutatja, hogy míg a dinamikus algoritmus körülbelül hasonló értékeket ér el, addig a fix változat eléggé τ érzékeny. Az érzékenységre vonatkozó megállapítást Silva et al. (2008, 2014); Fontes et al. (2014); Amaral et al. (2013) is megtették, ők teszteléssel választották ki a megfelelő értéket, amelyikre a legjobban teljesítő modellt kapták. A későbbiekben látható, hogy a dolgozatban bemutatott algoritmus dinamikus verziója sokkal előnyösebb és pontosabb modellt ér el.



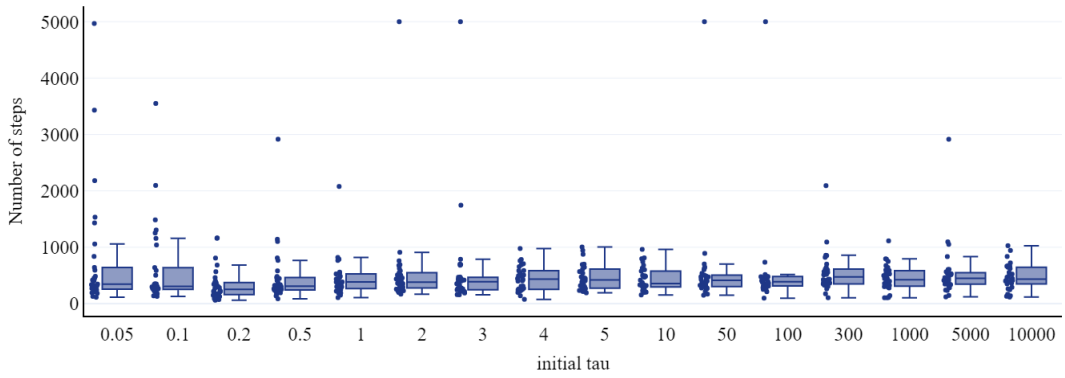
3.8. ábra. A végső MSE értékek mediánjai τ-nként ábrázolva. Az ábra azt mutatja, hogy a fix változat erősen τ függő. Ez a kép az Glass adathalmaz eredményi alapján készült, de hasonló viselkedés figyelhető meg a többi tesztelt osztályozási feladatra is.



3.9. ábra. A végső MSE értékek mediánja hasonlóak a különböző tesztelt induló τ értékekre, ami mutatja a dinamikus algoritmus kezdeti τ-tól való függetlenségét. Az ábra azt mutatja, hogy a fix változat erősen τ függő. Ez a kép az Glass adathalmaz eredményi alapján készült, de hasonló viselkedés figyelhető meg a többi tesztelt osztályozási feladatra is. (Ezen az ábrán a 0.05 érték kivételt képez, de a tapasztalatok alapján ezen a induló τ értékre a fix és dinamikus algoritmus is meglehetősen instabil.)

3.2.3.2. Az átlagos lépésszám különböző kezdő τ értékek esetén ugyanaz

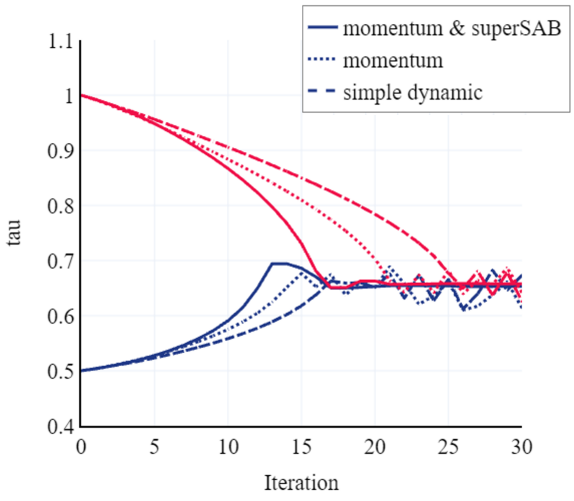
Tanító lépések száma: A τ-függetlenség nemcsak a modellpontossági mértékek esetén látható, hanem az elvárt iterációs lépésszámban is (3.10 ábra). Ez azt jelenti, hogy a dinamikus algoritmus futási ideje nem függ a dinamikusan változó τ paraméter kezdeti értékétől, ez igen előnyös tulajdonság.



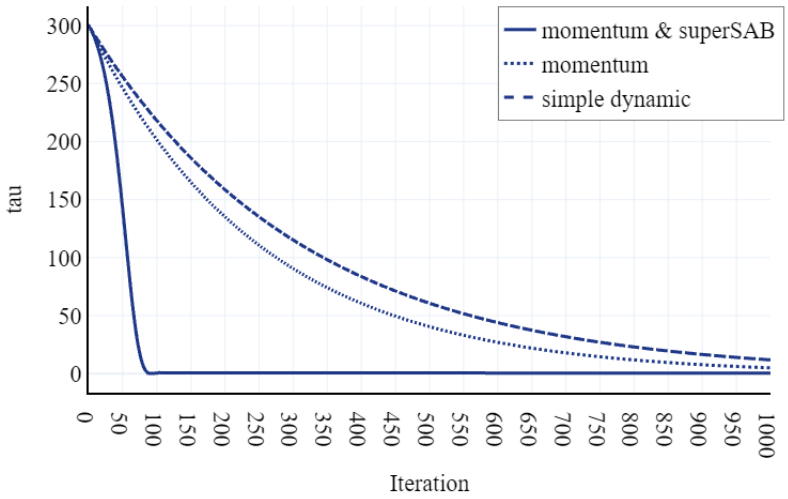
3.10. ábra. Az egyes τ értékenként végzett 30-30 futtatás lépésszámának boxai nagyon hasonlóak egymáshoz. Ez a kép az Ionosphere adathalmaz eredményi alapján készült, de hasonló viselkedés figyelhető meg a többi tesztelt osztályozási feladatra is.

3.2.4. Sikeres gyorsítás

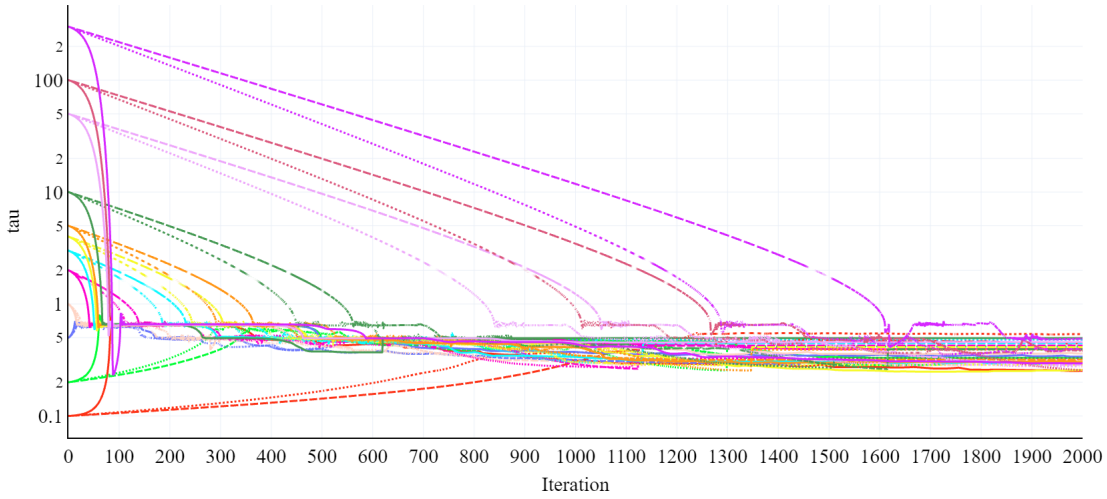
Az 3.11 ábra a 2.3.2 fejezetben ismertetett 3 különböző verzióját mutatja a dinamikus E_{ExpAbs} alapú LM algoritmusnak. Ha a kezdeti τ érték közel van a végsőhöz (optimálishoz) akkor a sebesség növekedés kisebb, viszont a kombinált megoldás (momentum és SuperSAB) tanulási sebessége sokkal jelentősebb növekedést mutat, ha a végső τ értéktől távolabb esik a kezdeti (3.12 ábra). Ez *a sebesség növekedés konzekvensen megfigyelhető a tesztelt kezdeti τ értékek teljes tartományára* (3.13 ábra).



3.11. ábra. Különböző gyorsítási módszerek a dinamikus τ-s verzióra az Iris adathalmazon (két különböző tanítást szemléltetve): folytonos vonal a momentummal és SuperSAB-bal gyorsított, a pontozott vonal a csak momentummal gyorsított, a szaggatot vonal pedig a szimpla dinamikus esetet jelöli. Az egyszerű dinamikus és a gyorsított változatok is ugyanazt a közelítőleg optimális τ értéket találják meg, de különböző sebességgel.



3.12. ábra. Óriási gyorsulás figyelhető meg nagyobb τ értékek esetén. A legjobb gyorsítási módszer kevesebb, mint 100 iteráció alatt megtalálja a megfelelő τ értéket, míg a többi módszer viszonylag lassan tanul.



3.13. ábra. τ paraméterre vonatkozó különböző tanulási görbék az Iris adathalmazon (több tanulás, eltérő induló τ értekek esetén): folytonos vonal a momentummal és SuperSAB-bal gyorsított, a pontozott vonal a csak momentummal gyorsított, a szaggatot vonal pedig a szimpla dinamikus esetet jelöli. Fontos, itt az y tengely logaritmikus.

3.2.5. Az új algoritmus összehasonlítása a az irodalomban fellelhető, aktuálisan legjobb megoldással

Az E_{Exp} alkalmazásával kapcsolatos szakirodalom elég ritka, az egyik legkorszerűbb megoldást és eredmény Amaral et al. (2013) írják le. Ők különböző költségfüggvény kombinációkat teszteltek auto-encoder tanítására. A bevezetett előtanítási fázisban az volt a cél, hogy

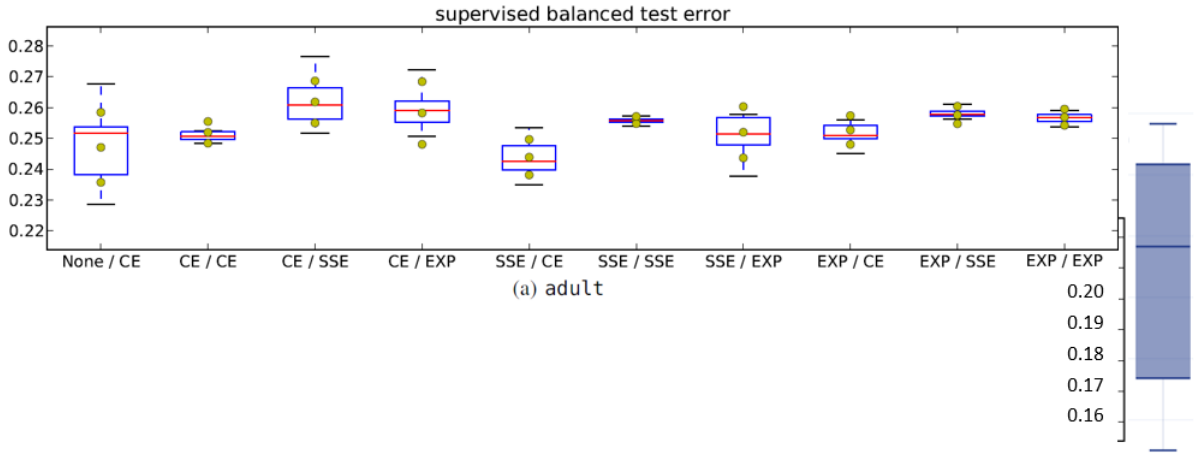
a neurális hálózathoz találjanak egy olyan kezdeti beállítást egy kiválasztott hibamérték segítségével, ami a random inícializálásnál jobb. Majd ezt az előtanított hálót egy másik hibamértékkel tovább tanították a finomhangolási fázisban. Ebben a két fázisban 3-3 különbözö költségfüggvény lehetséges páros kombinációit alkalmazták: CE, SSE (MSE), E_{Exp} (csak pozitív, fix τ értékkel).

A UCI gépitanulási adattárházából származó Adult adathalmaz megfelelő (és a koárbbi cikk leírás miatt egyetlen) jelölt a jelen TDK dolgozat témájául szolgáló algoritmus és a korábbi, ismert legkorszerűbb megoldás összehasonlítására. Ez azért lehetséges, mert Amaral et al. (2013) ugyanezeket a hibamértékeket alkalmazták a tanítás során, és cikkükben a kapcsolódó eredmények megfelelően dokumentáltak.

A fentebb említett kutatásban a háló struktúra az Adult adathalmaz tanításánál 2 rejtett réteget tartalmazott és mindkettőn 2-2 neuront. Leállási feltételként az early stopping módszert alkalmazták rögzített tanító (5000 adatpont) és validációs (1414 adatpont) halmazra, a modelleket pedig 26147 adatponton tesztelték.

Jelen kutatásban a leállási feltétel look-ahead paramétere 500 volt (25 helyett, amit Amaral et al. (2013) cikk használt). Az η_+ és η_- paraméterek értékei 1.02 és 0.3, az induló τ 10-nek lett választva a dinamikus τ-t használó változatban, a momentum módszer α paramétere pedig 0.1-nek. A neurális hálózat struktúrája az idézett kutatáshoz hasonlóan két rejtett rétegű hálózat volt 2-2 neuronnal.

Amaral et al. (2013) kutatásának tanulsága, hogy az osztályozási feladatokban az előtanítási fázisban a SSE teljesít a legjobban. Azt is megfigyelték, hogy a finomhangolásuói lépésben a CE költségfüggvény jól kezeli a kiegyensúlyozatlan adatot. Ezért a legjobb hibamérték párosítás az előtanító és finomhangoló fázisokra az SSE és a CE, amit a 3.14 ábra is szemléltet.



3.14. ábra. Az új algoritmus osztályozási hiba százaléka (1 - RecRate) (kék box a jobb oldalon) összehasonlítva a Amaral et al. (2013) cikkében található eredményekkel (fehér boxok). Az ábra azt mutatja, hogy a jelen dolgozatban bemutatott új, dinamikus algoritmus jelentősen jobban teljesít 2 rejtett réteg és azokon 2-2 rejtett csúcs esetén.

Ha összevetjük a jelen kutatás eredményivel, az összehasonlítás azt mutatja meg, hogy *a új, dinamikus módszer sokkal pontosabb modellt képes tanítani, mint az Amaral et al. (2013) által bemutatott két lépéses state-of-the-art megközelítés*.

3.3. Összegzés

A dolgozatban bemutatott irodalmak, az új algoritmus levezetése és valós adathalmazokon való tesztelése alapján a következő megállapítások tehetők. **A TDK kutatás eredménye, azaz, a bevezetett új dinamikus, statisztikai és egyidejűleg információmléleti mérték alkalmazása és a kapcsolódó, új tanító algoritmus azt eredményezték, hogy**

T1: a az új, dinamikus változat alacsony és magas induló τ értékek esetén jobb modellpontosságot ért el, mint a fix;

T2: a dinamikus algoritmus τ független, stabilabb a fix változatnál;

T3: a momentum módszer és a SuperSAB kombinációjának Levenberg-Marquardt algoritmusba ágyazása valóban sokkal gyorsítja tanítást dinamikusan változó τ esetén;

T4: a dinamikus τ-s változat használatával kiküszöbölhető az előtanítási lépés és pontosabb modell érhető el, mint a korábban ismert Amaral et al. (2013)-féle, state-of-the-art kutatásban.

Ezek alapján elmondható, hogy a TDK kutatás témájául szolgáló algoritmus jól teljesít és a benne megjelenő ötletek a továbbiakban is ígéretes kutatási területet jelentenek (a kutatásokat célszerű tovább folytatni).

3.4. Kitekintés

További kutatási ötletek között szerepel a módszer regressziós problémára történő továbbfejlesztése, a gyorsítási technikák kiterjesztése a háló súlyaira is, valamint más alkalmas mértékek kipróbálása különböző ígéretes tanító/optimalizáló algoritmusokban.

Egy másik következő kutatási irány ugyanezen dinamikus, integrált mérték beágyazása más tanító algoritmusba, pl. ADAM területén az első viszgálatok - pozitív eredménnyel - már meg is történtek.

A. függelék

A.1. E_{ExpAbs} deriváltjainak elemei - részletes levezetés

$$\begin{aligned}
\frac{\partial err_p^{ExpAbs}}{\partial \tau} &= sign(\tau) \cdot \exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L} e_{n_L}^2\right) \\
&+ |\tau| \cdot \exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L} e_{n_L}^2\right) \cdot (-1) \cdot \frac{1}{|\tau|^2}\ sign(\tau)\left(\sum_{n_L=0}^{N_L} e_{n_L}^2\right) \\
&= sign(\tau) \cdot \exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L} e_{n_L,p}^2\right) \cdot \left(1 - \frac{1}{|\tau|}\sum_{n_L=0}^{N_L} e_{n_L,p}^2\right) \\
&= sign(\tau) \exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L} \left(t_{n_L} - o_{n_L}^L\right)^2\right) \cdot \left(1 - \frac{1}{|\tau|}\sum_{n_L=0}^{N_L} \left(t_{n_L} - o_{n_L}^L\right)^2\right)
\end{aligned} \tag{A.1}$$

$$\begin{aligned}
\frac{\partial err_p^{Exp}}{\partial o_{n_L}^L} &= |\tau| \exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L} \left(t_{n_L} - o_{n_L}^L\right)^2\right) \cdot \frac{1}{|\tau|} \cdot 2 \cdot \left(t_{n_L} - o_{n_L}^L\right) \cdot (-1) \\
&= -2 \cdot \left[\exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L} \left(t_{n_L} - o_{n_L}^L\right)^2\right)\right] \cdot \left(t_{n_L} - o_{n_L}^L\right) \\
&= -2 \cdot \left[\exp\left(\frac{1}{|\tau|}\sum_{n_L=0}^{N_L} e_{n_L,p}^2\right)\right] \cdot e_{n_L,p}
\end{aligned} \tag{A.2}$$

$$
\begin{aligned}
\frac{\partial o^L_{n_L}}{\partial o^{L-1}_{n_{L-1}}} &= \frac{\partial}{\partial o^{L-1}_{n_{L-1}}}\left(\sigma\left(\sum_{n_{L-1}=0}^{N_{L-1}} w^L_{n_{L-1},n_L}\cdot o^{L-1}_{n_{L-1}}\right)\right)\\
&= \sigma'\left(\sum_{n_{L-1}=0}^{N_{L-1}} w^L_{n_{L-1},n_L}\cdot o^{L-1}_{n_{L-1}}\right)\cdot\frac{\partial}{\partial o^{L-1}_{n_{L-1}}}\left(\sum_{n_{L-1}=0}^{N_{L-1}} w^L_{n_{L-1},n_L}\cdot o^{L-1}_{n_{L-1}}\right)\\
&= \sigma'\left(\sum_{n_{L-1}=0}^{N_{L-1}} w^L_{n_{L-1},n_L}\cdot o^{L-1}_{n_{L-1}}\right)\cdot w^L_{n_{L-1},n_L}\\
&= \sigma\left(\sum_{n_{L-1}=0}^{N_{L-1}} w^L_{n_{L-1},n_L}\cdot o^{L-1}_{n_{L-1}}\right)\cdot\left(1-\sigma\left(\sum_{n_{L-1}=0}^{N_{L-1}} w^L_{n_{L-1},n_L}\cdot o^{L-1}_{n_{L-1}}\right)\right)\cdot w^L_{n_{L-1},n_L}\\
&= \sigma\left(o^L_{n_L}\right)\cdot\left(1-\sigma\left(o^L_{n_L}\right)\right)\cdot w^L_{n_{L-1},n_L}
\end{aligned}
\tag{A.3}
$$

$$
\begin{aligned}
\frac{\partial o^l_{n_l}}{\partial o^{l-1}_{n_{l-1}}} &= \frac{\partial}{\partial o^{l-1}_{n_{l-1}}}\left(\sigma\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\right)\\
&= \sigma'\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\cdot\frac{\partial}{\partial o^{l-1}_{n_{l-1}}}\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\\
&= \sigma'\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\cdot w^l_{n_{l-1},n_l}\\
&= \sigma\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\cdot\left(1-\sigma\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\right)\cdot w^l_{n_{l-1},n_l}\\
&= \sigma\left(o^l_{n_l}\right)\cdot\left(1-\sigma\left(o^l_{n_l}\right)\right)\cdot w^l_{n_{l-1},n_l}
\end{aligned}
\tag{A.4}
$$

$$
\begin{aligned}
\frac{\partial o^l_{n_l}}{\partial w^l_{n_{l-1},n_l}} &= \frac{\partial}{\partial w^l_{n_{l-1},n_l}}\left(\sigma\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\right)\\
&= \sigma'\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\cdot\frac{\partial}{\partial w^l_{n_{l-1},n_l}}\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\\
&= \sigma'\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\cdot o^{l-1}_{n_{l-1}}\\
&= \sigma\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\left(1-\sigma\left(\sum_{n_{l-1}=0}^{N_{l-1}} w^l_{n_{l-1},n_l}\cdot o^{l-1}_{n_{l-1}}\right)\right)\cdot o^{l-1}_{n_{l-1}}\\
&= \sigma\left(o^l_{n_l}\right)\cdot\left(1-\sigma\left(o^l_{n_l}\right)\right)\cdot o^{l-1}_{n_{l-1}}
\end{aligned}
\tag{A.5}
$$

Irodalomjegyzék

Amaral, T., Silva, L. M., Alexandre, L. A., Kandaswamy, C., Santos, J. M., & de Sá, J. M. (2013). Using different cost functions to train stacked auto-encoders. In *2013 12th Mexican international conference on artificial intelligence* (pp. 114–120). IEEE.

Boyd, S., & Vandenberghe, L. (2018). *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*. (1st ed.). Cambridge University Press. URL: https://www.cambridge.org/core/product/identifier/9781108583664/type/book. doi:10.1017/9781108583664.

Cover Thomas, M., & Thomas Joy, A. (1991). *Elements of information theory*. Wiley Series in Telecommunication. John Wiley and Sons, Inc.

Dowson, N., & Bowden, R. (2007). Mutual information for lucas-kanade tracking (milk): An inverse compositional formulation. *IEEE transactions on pattern analysis and machine intelligence*, *30*, 180–185.

Dua, D., & Graff, C. (2017). UCI machine learning repository. URL: http://archive.ics.uci.edu/ml.

Erdogmus, D., & Principe, J. C. (2002). An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems. *IEEE Transactions on Signal Processing*, *50*, 1780–1786.

Fontes, T., Silva, L., Silva, M., Barros, N., & Carvalho, A. (2014). Can artificial neural networks be used to predict the origin of ozone episodes? *Science of the total environment*, *488*, 197–207.

Heravi, A. R., & Hodtani, G. A. (2016). A new robust correntropy based levenberg-marquardt algorithm. In *2016 Iran Workshop on Communication and Information Theory (IWCIT)* (pp. 1–6). IEEE.

Heravi, A. R., & Hodtani, G. A. (2018a). Comparison of the convergence rates of the new correntropy-based levenberg–marquardt (clm) method and the fixed-point maximum correntropy (fp-mcc) algorithm. *Circuits, Systems, and Signal Processing*, *37*, 2884–2910.

Heravi, A. R., & Hodtani, G. A. (2018b). A new correntropy-based conjugate gradient backpropagation algorithm for improving training in neural networks. *IEEE transactions on neural networks and learning systems*, *29*, 6252–6263.

Heravi, A. R., & Hodtani, G. A. (2018c). Where does minimum error entropy outperform minimum mean square error? a new and closer look. *IEEE Access*, *6*, 5856–5864.

Heravi, A. R., & Hodtani, G. A. (2019). A new and fast correntropy-based method for system identification with exemplifications in low-snr communications regime. *Neural Computing and Applications*, *31*, 4407–4422.

Hopfield, J. J. (1987). Learning algorithms and probability distributions in feed-forward and feed-back networks. *Proceedings of the national academy of sciences*, *84*, 8429–8433.

Kline, D. M., & Berardi, V. L. (2005). Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing & Applications*, *14*, 310–318.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, *11*, 431–441.

Nilsaz-Dezfouli, H., Abu-Bakar, M., & Pourhoseingholi, M. (2016). An artificial neural network model for outcome prediction in gastric cancer patients. *JOURNAL OF FUNDAMENTAL AND APPLIED SCIENCES*, *8*, 1687–1698.

Panin, G., & Knoll, A. (2008). Mutual information-based 3d object tracking. *International Journal of Computer Vision*, *78*, 107–118.

Park, J. C., Neelakanta, P. S., Abusalah, S., De Groff, D. F., & Sudhakar, R. (1995). Information-theoretic based error-metrics for gradient descent learning in neural networks. *Complex Systems*, *9*.

Prieto, A., Prieto, B., Ortigosa, E. M., Ros, E., Pelayo, F., Ortega, J., & Rojas, I. (2016). Neural networks: An overview of early research, current frameworks and new challenges. *Neurocomputing*, *214*, 242–268.

Rady, H. (2011a). Reyni's entropy and mean square error for improving the convergence of multilayer backprobagation neural networks: a comparative study. *International Journal of Electrical & Computer Sciences IJECS-IJENS*, *11*, 68–79.

Rady, H. A. K. (2011b). Shannon entropy and mean square errors for speeding the convergence of multilayer neural networks: A comparative approach. *Egyptian Informatics Journal*, *12*, 197–209.

Ranganathan, A. (2004). The Levenberg-Marquardt Algorithm. URL: https://www.yumpu.com/en/document/read/17722281/the-levenberg-marquardt-algorithm.

Rimer, M., & Martinez, T. (2006). Cb3: an adaptive error function for backpropagation training. *Neural processing letters*, *24*, 81–92.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536. URL: https://www.nature.com/articles/323533a0. doi:10.1038/323533a0. Number: 6088 Publisher: Nature Publishing Group.

Silva, L. M., Alexandre, L. A., & de Sá, J. M. (2005a). Neural Network Classification: Maximizing Zero-Error Density, . *3686*, 127–135. URL: http://link.springer.com/10.1007/11551188_14. doi:10.1007/11551188_14. Series Title: Lecture Notes in Computer Science.

Silva, L. M., de Sá, J. M., & Alexandre, L. A. (2005b). Neural network classification using shannon's entropy. In *ESANN* (pp. 217–222). Citeseer.

Silva, L. M., de Sá, J. M., & Alexandre, L. A. (2008). Data classification with multilayer perceptrons using a generalized error function. *Neural Networks*, *21*, 1302–1310.

Silva, L. M., Santos, J. M., & Marques de Sá, J. (2014). Classification performance of multilayer perceptrons with different risk functionals. *International Journal of Pattern Recognition and Artificial Intelligence*, *28*, 1450013.

Szűcs, A. (2020). Integration of information theory measures into the Levenberg-Marquardt algorithm for adaptive training of artificial neural networks.

Szűcs, A. (2021). Tdk dolgozat: Mesterséges neurális hálózatok adaptív tanítása nagy mátrix technikával módosított Levenberg-Marquardt algoritmusba integrált információelméleti mértékekkel.

Thévenaz, P., & Unser, M. (2000). Optimization of mutual information for multiresolution image registration. *IEEE transactions on image processing*, *9*, 2083–2099.

Tollenaere, T. (1990). SuperSAB: Fast adaptive back propagation with good scaling properties. *Neural Networks*, *3*, 561–573. URL: https://www.sciencedirect.com/science/article/pii/0893608090900067. doi:10.1016/0893-6080(90)90006-7.

Viharos, Z. J. (1999). *Intelligens módszerek gyártási folyamatok modellezésében és optimalizálásában*. Ph.D. thesis.

Watrous, R. L. (1992). A comparison between squared error and relative entropy metrics using several optimization algorithms. *Complex Systems*, *6*, 495–506.

Wen, D., Jia, P., Hsu, S. H., Zhou, Y., Lan, X., Cui, D., Li, G., Yin, S., & Wang, L. (2019). Estimating coupling strength between multivariate neural series with multivariate permutation conditional mutual information. *Neural Networks*, *110*, 159–169. URL: https://doi.org/10.1016/j.neunet.2018.11.006. doi:10.1016/j.neunet.2018.11.006.

Yu, H., & Wilamowski, B. M. (2011). Levenberg-marquardt training. *Industrial electronics handbook*, *5*, 1–16.