FELADATKIÍRÁS

Az autoencoder a deep learning egyik népszerű architektúrája nem felügyelt tanulásra, amit egyfajta tömörítési eljárásként is lehet értelmezni. A hallgató feladata magyar nyelvű szavak rekonstrukciója autoencoder, illetve variational autoencoder segítségével.



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Velkey Géza

AutoencoderEs kísérletek

Konzulens

Ács Judit

BUDAPEST, 2017

Tartalomjegyzék

[Összefoglaló 4](#_Toc482100767)

[Abstract 5](#_Toc482100768)

[1 Bevezetés 6](#_Toc482100769)

[2 Kapcsolódó Munkák 7](#_Toc482100770)

[3 Architektúra 8](#_Toc482100771)

[4 Adat és Előfeldolgozás 9](#_Toc482100772)

[4.1 Feature kinyerés 10](#_Toc482100773)

[5 Kísérleti Beállítások 11](#_Toc482100774)

[6 Eredmények kiértékelése 12](#_Toc482100775)

[Irodalomjegyzék 14](#_Toc482100776)

[Függelék 15](#_Toc482100777)

Összefoglaló

Ide jön a ½-1 oldalas magyar nyelvű összefoglaló, melynek szövege a Diplomaterv Portálra külön is feltöltésre kerül.

Abstract

Autoencoders are neural networks which aim to reconstruct their input with the least amount of distortion. Using autoencoders eliminates the need for – often expensive – labeled training samples, by turning an unsupervised learning setting into a supervised one. In the simplest case, an autoencoder is a feed forward neural network with one or more hidden layers which are typically much smaller than the network’s input layer, creating a compressed representation of the input. We present a series of autoencoder experiments using Hungarian words as their input. Our architectures include deep autoencoders with more than one hidden layer and variational autoencoders. We also experiment with diﬀerent preprocessing steps such as replacing di- and trigraphs with a single character. Our error analysis of the reconstruction errors gives insight into frequent morphological paradigms occuring in Hungarian.

# Bevezetés

A gépi tanuló algoritmusok alkalmazásai körében sok példa van arra, hogy kulcsfontosságú a bemeneti adatok megfelelő reprezentálása. A mély neurális hálózatok ebben bizonyítottan jól teljesítenek [1], ezek közül pedig nagy népszerűségnek örvendenek az autoencoderek, melyek felépítése igen egyszerű, a bemenetüket próbálják visszaadni a kimenetükön.

A számítógépekkel történő nyelvfeldolgozás során gyakran kerülünk olyan helyzetbe, hogy a bemenet dimenziója nagyon nagy, mint például egy nyelv összes szavát tartalmazó szótár esetén, amit általában úgy oldanak meg, hogy egy embedding réteg csökkenti a bemenet dimenzióját és word vectorokat készít a szavakból. Ebben az önálló laboratórium feladatban több fajta szó reprezentációval is kísérleteztünk magyar nyelven.

A dokumentáció felépítése a következő: Először ismertetjük a kapcsolódó munkákat, majd a neurális hálózatok architektúrájával foglalkozunk. A 4. részben a bemeneti adatokat, azok szűrését és előfeldolgozását taglaljuk. Az 5. fejezetben szerepelnek a kísérleti beállítások, a feltérképezett paramétertér leírása, valamint a kísérleteket koordináló genetikus algoritmus működése. A 6. fejezetben a kísérleti eredmények szerepelnek, majd a 7. rész lezárja a dokumentációt.

# Kapcsolódó Munkák

Az autoencoderek első alkalmazása [2] a Principal Component Analysis (PCA) nemlineáris általánosítása volt. Rájöttek, hogy az autoenkóderek teljesítménye nagyban függ az inicializásuktól is, és Restricted Boltzmann Machine-t (RBM) alkalmaztak a neurális hálózatok előtanítására.

Az autoencodereket sok különböző területen alkalmazták mind NLP, mind képfeldolgozás és számítógépes biológiai adatfeldolgozó alkalmazásokban is. [3] sentiment analízisre használta a hálót, [4] egy jóval több rétegből álló autoencoderrel angol- kínai fordítást valósított meg. [5] rekurrens autoencodert készített LSTM cellákból, mellyel bekezdéseket állított vissza word vectorokból. [6] sikeresen használta az architektúrát információ visszaállítására.

A legújabb eredmények között található [7], mely egy Split-Brain architektúrát használ, amivel mi is több kísérletet végeztünk magyar nyelven.

Mivel az eddig NLP-vel kapcsolatos eredmények főként bekezdések, mondatok visszaállításra és word vectorok alkalmazására voltak kialakítva, a magyarhoz hasonló nyelvek szavakon belüli struktúráját nem használták fel, mely fontos addicionális információt tartalmaz.

# Architektúra

Az autoenkóderek az előző fejezetben láthatóak alapján

## Fully Connected Autoencoder

## Variational Autoencoder

## Split-Brain Autoencoder

# Adatok és Előfeldolgozás

## Feature kinyerés

# Kísérleti Beállítások

# Eredmények kiértékelése

# Konklúzió

Irodalomjegyzék

1. Y.Bengio,A.Courville,andP.Vincent,“Representationlearning: Areview and new perspectives,” IEEE Trans. PAMI, vol. 35, no. 8, pp. 1798–1828, 2013.
2. G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” science, vol. 313, no. 5786, pp. 504–507, 2006.
3. W. Rong, Y. Nie, Y. Ouyang, B. Peng, and Z. Xiong, “Auto-encoder based bagging architecture for sentiment analysis,” Journal of Visual Languages & Computing, vol. 25, no. 6, pp. 840–849, 2014.
4. S. Lu, Z. Chen, B. Xu, et al., “Learning new semi-supervised deep autoencoder features for statistical machine translation.,” in ACL (1), pp. 122– 132, 2014.
5. J. Li, M.-T. Luong, and D. Jurafsky, “A hierarchical neural autoencoder for paragraphs and documents,” arXiv preprint arXiv:1506.01057, 2015.
6. P. Mirowski, M. Ranzato, and Y. LeCun, “Dynamic auto-encoders for semantic indexing,” in Proceedings of the NIPS 2010 Workshop on Deep Learning, pp. 1–9, 2010.
7. C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization.” 2017.
8. P. Baldi, “Autoencoders, unsupervised learning, and deep architectures.,” ICML unsupervised and transfer learning, vol. 27, no. 37-50, p. 1, 2012.
9. Cs. Oravecz, T. Váradi, and B. Sass, “The Hungarian Gigaword Corpus,” in Proceedings of LREC 2014, 2014.
10. L. Han, A. L. Kashyap, T. Finin, J. Mayﬁeld, and J. Weese, “Umbc\_ebiquity-core: Semantic textual similarity systems,” in Second Joint Conference on Lexical and Computational Semantics (\*SEM), (Atlanta, Georgia, USA), pp. 44–52, Association for Computational Linguistics, 2013.

Függelék