PROJECT BESZÁMOLÓ

Bevezetés a Természetes Nyelvű Szövegfeldolgozásba

Tartalom

[Project bemutatása 2](#_Toc68182587)

[Csapat tagok 2](#_Toc68182588)

[Fejlesztési idő 2](#_Toc68182589)

[Adathalmaz 2](#_Toc68182590)

[ELőfeldolgozás 2](#_Toc68182591)

[Modell 2](#_Toc68182592)

[Kiértékelés 3](#_Toc68182593)

[További fejlesztési lehetősége, Tapasztalatok 3](#_Toc68182594)

# Project bemutatása

*Sentiment140 elemzése. BertWordPieceTokenizer-t használunk, a projekt elején megnéztük milyen sentimentek találhatóak a dataset-ben, és kiderült csak negatív és pozitív sentimentű tweet-ek találhatóak benne, ezt egy histogrammal néztük meg az adatok beolvasása után.*

# Fejlesztési idő

*82 óra*

# Adathalmaz

* *A példa project adathalmaza 1,600,000 rekordot tartalmaz.*
* *A rekordokhoz tartozó mezők:* 
  + ***target****: polaritása az adott tweet-nek:*
    - *0 -> negatív*
    - *4 -> pozitív*
  + ***ids****: A tweet azonosítója*
  + ***date****: A tweet dátuma (’Sat May 16 23:58:44 UTC 2009’)*
  + ***flag****: Query adatot tárol (nagyrésze NO QUERY)*
  + ***user****: Az adott felhasználó, aki tweetelt (’robotickilldozr’)*
  + ***text****: A tweet szövege*
* *Az adatok címkézettek*
* *A címkék többes osztályúak*

# Előfeldolgozás

* *Erőforrás hiánya miatt csökkentettük az adatok számát (pontosabban 1/8-adát hagytuk meg), persze előtte megkeverve az adatok sorrendjét.*
* *Átneveztük az oszlopokat az egyszerűbb kezelhetőség miatt.*
* *A target oszlopban a 4-es értéket átírtuk 1-esekre. (innentől a pozitív sentimentek az 1-eseknek felelnek meg). A sigmoid aktivációs függvény miatt tettük meg ezt.*
* *’Length’ oszlop felvétele a sentimentek hossza alapján.*
* *Pár ábrát is létrehoztunk hoztunk az adatok egyszerűbb átláthatóságához (pl negatív-pozitív sentimentek aránya, sentiment hossz átlaga és eloszlása, top10 negatív-pozitív felhasználók)*
* *A nem használt oszlopokat elhagytuk, ezek a következők: id, date, query, username*
* *Kiszedtük a stopword-ökat, és stemmeltük az adatokat és eltávolítottuk a linkeket és az említéseket.*
* *A szöveget tokenizáltuk BertWordPieceTokenizer technikával.*
* *Az adathalmazt felbontottuk Train/val/test, 60/20/20 arányban tanításhoz és kiértékeléshez.*

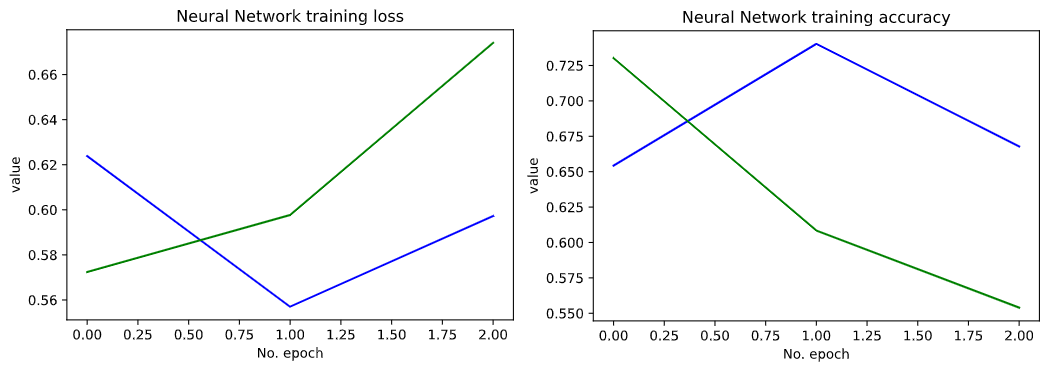
# Modell

* *A modellt több neurális hálózat használatával is kipróbáltuk.*
  + Használtunk Bidirectional-t a teljesítmény növelés érdekében, azonban ez a módszer nagyon lassúnak bizonyult.
  + *LSTM-et is kipróbáltunk, hogy össze tudjuk hasonlítani az eredményt a Bidirectional által adott eredménnyel.*
* *A pontosságot Accuracy-val mértük.*
* *Aktivációs függvénynek sigmoid-ot használtunk.*
* *A tanítást alapvetően 5 cikluson keresztül kezdtük azonban erőforráshiányában a ciklus számon csökkentenünk kellett. Ezen felül még a batch\_size-ot is csökkentettük 32-re.*

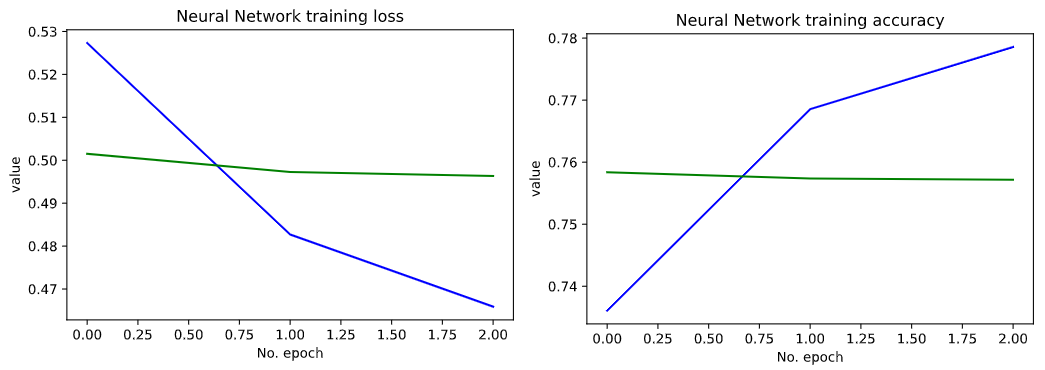
# Kiértékelés

A kiértékelés során a kék egyenes volt tanító (training) adathalmaz, zöld pedig a validációs adathalmaz volt. Ezeket a következő ábrákon láthatjuk majd.

* *Bidirectional-t használva kapott eredmények:*
  + *Futási idő:240-270 sec/epoch*



* *LSTM-et használva kapott eredmények:*
  + *Futási idő: 176-213 sec/epoch*



# További fejlesztési lehetősége, tapasztalatok

*Azt tapasztaltuk, hogy a prepocessing lépések növelésével jobb eredményt kaptunk, azonban túl sok preprocessing lépés használata negatív hatást fejtett ki, mivel az adathalmazunk rövid tweeteket tartalmaz, emiatt veszíthetünk adatokat. Tapasztalatunk alapján a Bidirectional-t ilyenfajta adathalmaz esetén nem teljesen éri meg alkalmazni. Ezzel szemben az LTSM gyorsabb lefutást és pontosabb eredményt adott vissza nekünk.*

*Ezek összehasonlítása itt látható:*

*A képen szöveg látható

Automatikusan generált leírás*