|  |  |  |
| --- | --- | --- |
|  | به نام خدا |  |
| **دانشگاه تهران**  **دانشکده‌ مهندسی برق و کامپیوتر**  **شبکه های عصبی و یادگیری عمیق**  **مینی پروژه سری 2** | | |

|  |  |
| --- | --- |
| امید واهب – مهسا مسعود | نام و نام خانوادگی |
| 810196582 - 810196635 | شماره‌ دانشجویی |
| 05/04/1400 | تاریخ ارسال گزارش |

­

**فهرست گزارش سوالات**

[سوال 1 – آشنایی با کاربرد (شبکه های عصبی بازگشتی) در سری زمانی 3](#_Toc71025311)

[سوال 3 – آشنایی با کاربرد (شبکه های عصبی بازگشتی) در متن 3](#_Toc71025313)

# سوال 1 – آشنایی با کاربرد (شبکه های عصبی بازگشتی) در سری زمانی

/

# سوال 3 – آشنایی با کاربرد (شبکه های عصبی بازگشتی) در متن

در این سوال داده های فایل sentiment\_final که محتوای تعدادی توییت هستند را بررسی می کنیم و پس از پیش پردازش های لازم، مدلی روی این داده های متنی آموزش می دهیم که مثبت یا منفی بودن محتوای توییت را حدس بزند. کد این سوال در فایل Project2\_Q3.ipynb موجود است.

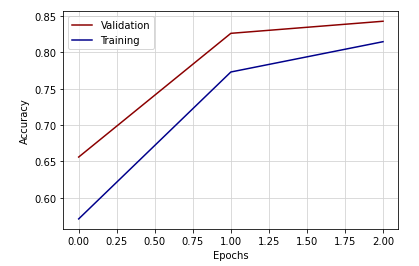
1- اکثر دیتاست های واقعی بالانس نیستند و تعداد یکی از کلاس ها بیشتر است که می تواند مشکل زا باشد. بزرگترین مشکلی که ایجاد می شود و از آن با پارادوکس دقت یاد می شود، این است که ممکن مدل ما به 90 درصد دقت دست یابد ولی این دقت صرفا حاصل از distribution داده ها باشد یعنی مدل ما درست آموزش ندیده است و فقط یک کلاس که تعداد بالایی دارد را خروجی می دهد و چون آن کلاس 90 درصد داده ها را تشکیل می دهد، دقت بالایی می گیریم. این اتفاق در داده هایی که بالانس نیستند زیاد اتفاق می افتد زیرا وقتی تعداد داده های یک کلاس بالا باشد روی وزن ها و پارامتر های مدل نیز بیشتر اثر می گذارد و مدل آموزش داده شده در اکثر مواقع لیبل پرتکرار را خروجی خواهد داد. از مسائلی که به خوبی این مشکل دیده می شود مسئله تشخیص سرطان و بیماری های دیگر است که معمولا کمتر از یک درصد از مردم دارن و وقتی یک دیتاست از مردم جامعه جمع می شود تا مدلی برای تشخیص آن آموزش داده شود اگر مدل ما روی تمامی افراد لیبل غیرسرطانی را بزند به دقت 99 درصد که خیلی بالا است می رسد و لی در عمل هیچ اطلاعاتی را به یاد نسپرده است. برای حل ایم مشکل روش ها و متدهای مختلفی پیش روی ما قرار دارد که به اختصار آنها را بحث می کنیم. اولین کار، حل مشکل از مبدا است یعنی داده های بیشتری از کلاس minority جمع کنیم و کل مشکل را حل کنیم ولی خوب معمولا این کار عملی نیست. راه دیگر زمانی که هیچ یک از متد ها جواب ندهد استفاده از متریک دیگری برای سنجش و آموزش مدل است یعنی به جای دقت که از داده ی imbalanced اثر می پذیرد، از معیارهای دیگری مثل f1 score، recall و precision استفاده کنیم. راه دیگر تغییر الگوریتم و یا اثر دادن پنالتی برای خروجی دادن لیبل اکثریت توسط مدل است. راه آخر که از بقیه موارد نیز بیشتر استفاده می شود resample کردن داده است که یا با undersample کلاس پرتعدادتر انجام می شود و یا با oversample کردن کلاس کم تعدادتر به عبارت دیگر یا تعدادی از داده های کلاس با مقادیر زیاد را حذف کرده و رندوم فقط بخشی از داده های آن را نگه می داریم که این کار وقتی تعداد داده های ما زیاد نیست اصلا توصیه نمی شود زیرا اطلاعاتی که با آنها می توانستیم مدل را بهبود دهیم را دور میریزیم و یا اینکه داده های کلاس با تعداد کمتر را چند بار کپی کرده و یا به کمک روش هایی آنها را augment کرده و یا داده های جدیدی از روی آنها می سازیم تا تعداد دو کلاس بالانس شود. این کار برای داده های عددی مناسب تر است ولی در این مساله چون جنس داده های ما از جنس متن هستند نمی توانیم داده هایی جدید generate کنیم که معنادار باشند پس چهار بار مقادیر کلاس positive را کپی می کنیم تا distribution این دو کلاس بالانس شود.

2- حال به سراغ پیش پردازش داده ها می رویم. ابتدا همه ی حروف را Lowercase می کنیم زیرا بزرگ و کوچکی کلمات اطلاعات اضافه تری نمی دهند و فقط پیچیدگی بیش از حد به مدل اضافه می کنند. سپس علایم punctuation را حذف می کنیم زیرا اطلاعاتی ندارند و قابل یادگیری نیستند. در مرحله بعد پیش پردازشی مناسب این تایپ داده یعنی توییت می کنیم و emoji ها را با کلمات معادلشان جایگزین می کنیم زیرا این نوع داده مدل را به مشکل می اندازد. در مرحله بعد URL ها و تگ های html را نیز حذف می کنیم. سپس جملات را به کلمات می شکنیم و stop words زبان انگلیسی را از آن حذف می کنیم یعنی کلماتی که در جملات خیلی تکرار می شوند و معنا و مفهومی برای فرایند یادگیری ما ندارند و باقی گذاشتن آنها موجب اختلال در عملکرد مدل است. در انتها نیز به کمک spell check اشتباهات تایپی را نیز برطرف می کنیم. از پیش پردازش های دیگر که انجام شد حذف اعداد بود همچنین lemmatize کردن را نیز قصد داشتیم بکنیم ولی چون متن ها توییت بودند و ممکن بود این پیش پردازش موثر نباشد و بدون آن نیز به دقت نسبتا خوبی رسیدیم از آن صرف نظر شد.

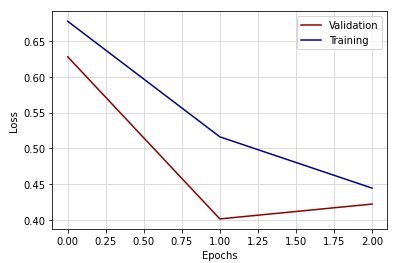
3- حال word embedding را با ساز 10000 کلمه پرتکرار انجام می دهیم.

4- سپس شبکه deep را پیاده سازی می کنیم که شامل یک لایه embedding و یک لایه lstm با 100 سلول و یک لایه Dense برای تولید خروجی مدنظر است. از optimizer معمول یعنی adam استفاده شد و چون شبکه recurrent است از activation function معمول این نوع شبکه یعنی sigmoid استفاده شد. معیار خا نیز binary\_crossentropy است و در چند ایپاک آموزش انجام می شود که خاصیت شبکه های lstm است.

5- نمودار خطا و دقت مدل به شرح زیر است:



شکل 2 - 1 : نمودار دقت مدل در ایپاک ها



شکل 2 - 2 : نمودار خطا مدل در ایپاک ها

می توان دید که در چند ایپاک مدل آموزش دیده است و سریع Overfit می شود که در شبکه های LSTM زیاد دیده می شود و نیاز به early stopping دارند.

6- حال توسط مدل آموزش دیده predict می کنیم و نتایج به شرح زیر خواهد بود:

