



---

# تشخيص عواطف

---

مقالات و مستندات



AUGUST , 2021

EXA LAB

Mohammad Hossein Forouhesh Tehrani

## فهرست موضوعات

2	تشخیص عواطف چیست؟
3	چرا تشخیص عواطف و طبقه بندی متن مهم است؟
3	مقیاس پذیری
3	تجزیه و تحلیل زمان واقعی
4	معیارهای سازگار
4	طبقه بندی متن چگونه کار می کند؟
4	سیستم های مبتنی بر قانون
5	سیستم های مبتنی بر یادگیری ماشین
6	الگوریتم های طبقه بندی متون یادگیری ماشین
6	دسته بندی کننده بیز ساده
7	ماشین های بردار پشتیبان
7	یادگیری عمیق
8	یادگیری عمیق در برابر الگوریتم های یادگیری ماشین سنتی
9	سیستم های ترکیبی
9	معیارها و ارزیابی
10	تشخیص و طبقه بندی عواطف
12	References

## تشخیص عواطف چیست؟

تشخیص عواطف<sup>۱</sup> از زیرشاخه های طبقه بندی متن<sup>۲</sup> میباشد که یک تکنیک یادگیری ماشین است که در آن مجموعه ای از دسته بندی های از پیش تعریف شده را به متن اختصاص می دهد. طبقه بندی متن را می توان برای سازماندهی، ساختاردهی و دسته بندی تقریباً هر نوع متنی - از اسناد، مطالعات پزشکی و پرونده ها، و مستندات در سراسر وب استفاده کرد.

به عنوان مثال، مقالات جدید را می توان بر اساس موضوعات و بلیط های پشتیبانی را می توان با فوریت سازماندهی کرد. گفتگوهای چت را می توان با زبان سازماندهی کرد. ذکر نام تجاری را می توان با احساس<sup>۳</sup> سازماندهی کرد و غیره.

طبقه بندی متن یکی از وظایف اساسی در پردازش زبان طبیعی<sup>۴</sup> با کاربردهای وسیع مانند تجزیه و تحلیل احساسات<sup>۵</sup>، برچسب گذاری موضوع<sup>۶</sup>، تشخیص هرزنامه ها<sup>۷</sup>، تشخیص قصد<sup>۸</sup> و تشخیص عواطف<sup>۹</sup> است.

در اینجا نمونه ای از نحوه عملکرد آن آمده است:

"رابط کاربری بسیار ساده و آسان برای استفاده است".

طبقه بندی کننده متن می تواند این عبارت را به عنوان ورودی در نظر بگیرد، محتوای آن را تجزیه و تحلیل کرده و سپس به طور خودکار برچسب های مربوطه مانند UI و Easy to Use را اختصاص دهد. متن ورودی توسط یک مدل طبقه بندی متن پردازش می شود و برچسب های خروجی را ارائه می دهد.

---

<sup>1</sup> Emotion detection

<sup>2</sup> Text classification

<sup>3</sup> Sentiment

<sup>4</sup> Natural Language Processing

<sup>5</sup> Sentiment Analysis

<sup>6</sup> Topic Labeling

<sup>7</sup> Spam Detection

<sup>8</sup> Intent Detection

<sup>9</sup> Emotion Recognition

## چرا تشخیص عواطف و طبقه بندی متن مهم است؟

تخمین زده می شود که حدود 80 درصد از کل اطلاعات (watson 2016) بدون ساختار است و متن یکی از رایج ترین انواع داده های بدون ساختار است. به دلیل ماهیت کثیف بودن متن، تجزیه و تحلیل، درک، سازماندهی و مرتب سازی داده های متنی سخت و زمان بر است، بنابراین اکثر شرکت ها در استفاده کامل از آن کوتاهی می کنند.

اینجاست که طبقه بندی متن با یادگیری ماشینی مطرح می شود. با استفاده از طبقه بندی کننده های متنی، شرکت ها می توانند به طور خودکار همه نوع متن مربوطه را از طریق ایمیل ها، اسناد قانونی، رسانه های اجتماعی، چت بات ها، نظرسنجی ها و موارد دیگر به شیوه ای سریع و مقرون به صرفه ساختاردهی کنند. این به شرکت ها اجازه می دهد تا در تجزیه و تحلیل داده های متنی صرفه جویی کنند، فرآیندهای تجاری را خودکار کرده و تصمیمات تجاری مبتنی بر داده اتخاذ کنند.

از جمله فواید استفاده از یادگیری ماشین در تشخیص عواطف میتوان به موارد زیر اشاره کرد:

### مقیاس پذیری

تجزیه و تحلیل و سازماندهی دستی آهسته و بسیار کم دقت است. یادگیری ماشین می تواند میلیون ها نظرسنجی، نظرات، ایمیل ها و غیره را به طور خودکار با کسری از هزینه، اغلب در چند دقیقه تجزیه و تحلیل کند. ابزارهای طبقه بندی متن برای هر گونه نیاز تجاری، بزرگ یا کوچک مقیاس پذیر است.

### تجزیه و تحلیل زمان واقعی

شرایط بحرانی وجود دارد که شرکت ها باید در اسرع وقت آنها را شناسایی کرده و اقدام فوری انجام دهند (به عنوان مثال، بحران روابط عمومی در رسانه های اجتماعی). طبقه بندی متون یادگیری ماشین می تواند به طور مداوم و در زمان واقعی از نام تجاری شما پیروی کند، بنابراین اطلاعات مهم را شناسایی کرده و قادر خواهید بود فوراً اقدام کنید.

## معیارهای سازگار

حاشیه نویسان انسانی هنگام طبقه بندی داده های متنی به دلیل حواس پرتی، خستگی و ملال دچار اشتباه می شوند و ذهنیت افراد معیارهای ناسازگار ایجاد می کند. از سوی دیگر، یادگیری ماشین از لنز و معیارهای یکسانی برای همه داده ها و نتایج استفاده می کند. هنگامی که یک مدل طبقه بندی متن به درستی آموزش می بیند، با دقت بی نظیری عمل می کند.

## طبقه بندی متن چگونه کار می کند؟

شما می توانید طبقه بندی متن را به دو صورت دستی یا خودکار انجام دهید.

طبقه بندی متنی دستی شامل حاشیه نویسی انسانی است که محتوای متن را تفسیر کرده و متناسب با آن طبقه بندی می کند. این روش می تواند نتایج خوبی را به همراه داشته باشد اما زمان بر و گران است.

طبقه بندی خودکار متن شامل یادگیری ماشینی، پردازش زبان طبیعی (NLP) و سایر تکنیک های هدایت شده توسط هوش مصنوعی است تا به طور خودکار متن را به شیوه ای سریعتر، مقرون به صرفه و دقیق تر طبقه بندی کند.

روشهای زیادی برای طبقه بندی خودکار متن وجود دارد ، اما همه آنها تحت سه نوع سیستم قرار می گیرند:

### سیستم های مبتنی بر قانون

رویکردهای مبتنی بر قوانین با استفاده از مجموعه ای از قوانین زبانی دست ساز متن را به گروه های سازمان یافته طبقه بندی می کند. این قوانین به سیستم دستور می دهد تا از عناصر معنایی مرتبط یک متن برای شناسایی دسته های مربوطه بر اساس محتوای آن استفاده کند. هر قانون شامل یک پیش زمینه یا الگو و یک طبقه پیش بینی شده است.

بگویید که می خواهید مقالات خبری را به دو گروه طبقه بندی کنید: ورزش و سیاست. ابتدا، شما باید دو لیست از کلمات را مشخص کنید که هر گروه را مشخص می کند (به عنوان مثال، کلمات مربوط به ورزش مانند فوتبال، بسکتبال، لبرون جیمز و غیره، و کلمات مربوط به سیاست، مانند دونالد ترامپ، هیلاری کلینتون، پوتین، و غیره).

در مرحله بعد، هنگامی که می خواهید یک متن ورودی جدید را طبقه بندی کنید، باید تعداد کلمات مربوط به ورزش را که در متن ظاهر می شوند بشمارید و همین کار را برای کلمات مربوط به سیاست انجام دهید. اگر تعداد واژه های مربوط به ورزش بیشتر از تعداد کلمات مربوط به سیاست باشد، متن به عنوان ورزش طبقه بندی می شود و بالعکس.

به عنوان مثال، این سیستم مبتنی بر قانون، عنوان "اولین بازی لبرون جیمز با لیکرز کی است؟" به عنوان ورزش زیرا یک اصطلاح مربوط به ورزش (لبرون جیمز) را در نظر می گرفت و هیچ اصطلاح مربوط به سیاست را در نظر نمی گرفت.

سیستم های مبتنی بر قانون برای انسان قابل درک هستند و می توانند در طول زمان بهبود یابند. اما این رویکرد معایبی دارد. برای شروع، این سیستم ها نیاز به دانش عمیق از حوزه دارند. آنها همچنین زمان بر هستند، زیرا ایجاد قوانین برای یک سیستم پیچیده می تواند بسیار چالش برانگیز باشد و معمولاً نیاز به تجزیه و تحلیل و آزمایش زیادی دارد. نگهداری سیستم های مبتنی بر قانون نیز دشوار است و مقیاس خوبی ندارد زیرا افزودن قوانین جدید می تواند بر نتایج قوانین قبلی تأثیر بگذارد.

### سیستم های مبتنی بر یادگیری ماشین

به جای تکیه بر قوانین دستی، طبقه بندی متن یادگیری ماشین یاد می گیرد که بر اساس مشاهدات گذشته طبقه بندی کند. الگوریتم های یادگیری ماشین با استفاده از مثال های از پیش تعیین شده به عنوان داده های آموزشی، می توانند ارتباطات مختلف بین قطعات متن را بیاموزند و خروجی خاصی (یعنی برچسب ها) برای یک ورودی خاص (یعنی متن) مورد انتظار است. "برچسب" طبقه بندی یا دسته ای از پیش تعیین شده است که هر متن ممکن است در آن قرار گیرد.

اولین قدم برای آموزش طبقه بندی کننده NLP یادگیری ماشین استخراج ویژگی است: از روشی برای تبدیل هر متن به یک نمایش عددی در قالب یک بردار استفاده می شود. یکی از روشهای متداول استفاده از کیسه کلمات (Brownlee 2019) است، جایی که بردار فرکانس یک کلمه را در فرهنگ لغت از پیش تعریف شده کلمات نشان می دهد.

برای مثال، اگر فرهنگ لغت ما کلمات {این، است، نه، عالی، بد، بسکتبال} باشد، برداری متن "این عالی است،" به صورت زیر خواهد بود:

(1, 1, 0, 0, 1, 0, 0, 0).

سپس، الگوریتم یادگیری ماشین با داده های آموزشی که شامل جفت مجموعه ویژگی ها (بردارهای هر مثال متنی) و برچسب ها (به عنوان مثال ورزش، سیاست) است، تهیه می شود تا یک مدل طبقه بندی ایجاد شود.

پس از آموزش با نمونه های آموزشی کافی، مدل یادگیری ماشین می تواند شروع به پیش بینی دقیق کند. از همان استخراج کننده ویژگی برای تبدیل متن دیده نشده به مجموعه ویژگی ها استفاده می شود، که میتواند در مدل طبقه بندی وارد شود تا پیش بینی هایی در مورد برچسب ها (به عنوان مثال، ورزش، سیاست) انجام شود.

طبقه بندی متون با یادگیری ماشین معمولاً بسیار دقیق تر از سیستم های قانون ساخته شده توسط انسان است، به ویژه در کارهای طبقه بندی پیچیده. همچنین، طبقه بندی با یادگیری ماشین آسان تر است و شما همیشه می توانید نمونه های جدید را برای یادگیری کارهای جدید برچسب گذاری کنید.

الگوریتم های طبقه بندی متون یادگیری ماشین

برخی از محبوب ترین الگوریتم های طبقه بندی متن شامل خانواده الگوریتم های Naive Bayes، ماشین های بردار پشتیبانی (SVM) و یادگیری عمیق است.

دسته بندی کننده بیز ساده<sup>۱۰</sup>

به طور کلی خانواده الگوریتم های آماری Naive Bayes برخی از الگوریتم های پرکاربرد در طبقه بندی متن و تجزیه و تحلیل متن هستند.

یکی از اعضای آن خانواده Multinomial Naive Bayes (MNB) با یک مزیت بزرگ است، که می توانید نتایج بسیار خوبی را به دست آورید حتی زمانی که مجموعه داده شما بسیار بزرگ نیست و منابع محاسباتی کمیاب هستند.

بیس بیز بر اساس قضیه بیز است که به ما کمک می کند احتمالات مشروط وقوع دو رویداد را بر اساس احتمال وقوع هر رویداد جداگانه محاسبه کنیم. بنابراین ما احتمال هر تگ را برای یک متن مشخص محاسبه می کنیم و سپس تگ را با بیشترین احتمال خروجی می دهیم.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

---

<sup>10</sup> Naive Bayes

احتمال  $A$ ، اگر  $B$  درست باشد، برابر با احتمال  $B$  است، اگر  $A$  درست باشد، برابر احتمال درست بودن  $A$ ، تقسیم بر احتمال درست بودن  $B$ . این بدان معناست که هر بردار که یک متن را نشان می دهد باید حاوی اطلاعاتی در مورد احتمال ظاهر کلمات خاص در متون یک دسته معین باشد، به طوری که الگوریتم می تواند احتمال آن متن متعلق به دسته را محاسبه کند.

### ماشین های بردار پشتیبان<sup>۱۱</sup>

یکی دیگر از الگوریتم های یادگیری ماشین برای طبقه بندی متن است، زیرا مانند Naive Bayes، برای شروع نتایج دقیق به داده های آموزشی زیادی نیاز ندارد. با این حال، ماشین های بردار پشتیبان به منابع محاسباتی بیشتری نسبت به Naive Bayes نیاز دارد، اما نتایج حتی سریعتر و دقیق تر است.

به طور خلاصه، ماشین های بردار پشتیبان یک خط یا ابرصفحه<sup>۱۲</sup> ترسیم می کند که فضا را به دو زیرفضا تقسیم می کند. یک زیرفضا حاوی بردارهایی (برچسب هایی) است که به یک گروه تعلق دارند و یک زیرفضای دیگر شامل بردارهایی است که به آن گروه تعلق ندارند. این بردارها نمایانگر متون آموزشی شما هستند و گروه، برچسبی است که متون خود را با آنها برچسب گذاری کرده اید. با پیچیده تر شدن داده ها، ممکن نیست که بردارها / برچسب ها را فقط در دو دسته طبقه بندی کنیم.

### یادگیری عمیق<sup>۱۳</sup>

یادگیری عمیق (Dair-ai 2018) مجموعه ای از الگوریتم ها و تکنیک ها است که با الهام از نحوه عملکرد مغز انسان، شبکه های عصبی نامیده می شود. معماری های یادگیری عمیق مزایای زیادی برای طبقه بندی متن ارائه می دهند زیرا با دقت بسیار بالا با ریاضیات و محاسبات سطح پایین عمل می کنند.

دو معماری اصلی یادگیری عمیق برای طبقه بندی متن عبارتند از:

شبکه های عصبی متحد<sup>۱۴</sup> (Brownlee 2019) و شبکه های عصبی مکرر<sup>۱۵</sup>. (Jason 2019)

<sup>11</sup> Support Vector Machines (SVM)

<sup>12</sup> hyperplane

<sup>13</sup> Deep Learning

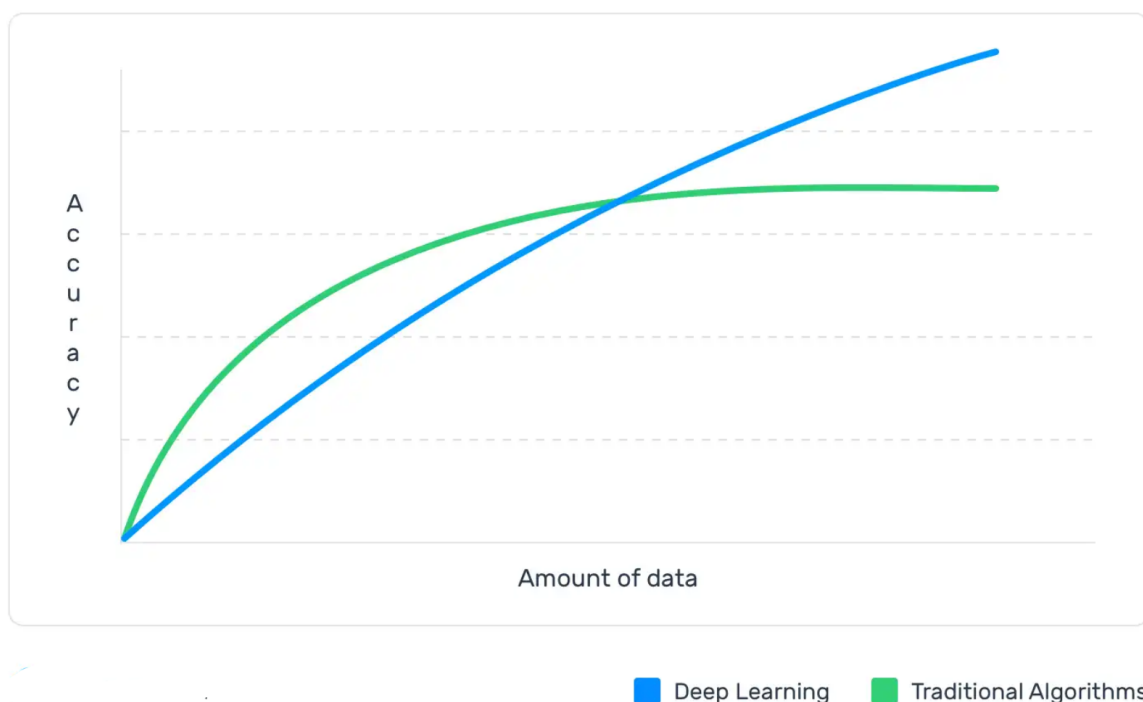
<sup>14</sup> Convolutional Neural Networks

<sup>15</sup> Recurrent Neural Networks



یادگیری عمیق عبارت است از یادگیری ماشین سلسله مراتبی، با استفاده از الگوریتم های متعدد در زنجیره مترقی رویدادها. این شبیه به نحوه عملکرد مغز انسان هنگام تصمیم گیری است و از تکنیک های مختلف به طور همزمان برای پردازش حجم عظیمی از داده ها استفاده می کند. همچنین شبکه های عصبی مکرر بسیار سریع تر از شبکه های عصبی متحد به جواب مسئله همگرا میشوند اما در داده های تست عملکرد ضعیف تری دارند. (Kim 2014)

الگوریتم های یادگیری عمیق به داده های آموزشی بسیار بیشتری نسبت به الگوریتم های سنتی یادگیری ماشین (حداقل میلیون ها نمونه برچسب زده شده) نیاز دارند. با این حال، آنها برای یادگیری از داده های آموزشی، مانند الگوریتم های یادگیری ماشین سنتی، مانند طبقه بندی کننده های SVM و NB، آستانه ای برای یادگیری از داده های آموزشی ندارند:



یادگیری عمیق در برابر الگوریتم های یادگیری ماشین سنتی  
الگوریتم های یادگیری عمیق، مانند Word2Vec یا GloVe نیز به منظور بدست آوردن نمایش بردار بهتر برای کلمات و بهبود دقت طبقه بندی کننده هایی که با الگوریتم های یادگیری ماشین سنتی آموزش دیده اند، استفاده می شود.

## سیستم های ترکیبی<sup>۱۶</sup>

سیستم های ترکیبی یک طبقه بندی پایه آموزش داده شده با یادگیری ماشین را با یک سیستم مبتنی بر قوانین ترکیب می کنند که برای بهبود بیشتر نتایج استفاده می شود. این سیستم های ترکیبی را می توان به راحتی با افزودن قوانین خاص برای آن دسته از برچسب های متناقض که توسط طبقه بندی کننده اصلی به درستی مدل نشده اند تنظیم کرد.

## معیارها و ارزیابی

اعتبار سنجی متقابل<sup>۱۷</sup> یک روش رایج برای ارزیابی عملکرد طبقه بندی کننده متن است. این کار با تقسیم مجموعه آموزشی به مجموعه های تصادفی و با طول مساوی (به عنوان مثال ، 4 مجموعه با 25 درصد از داده ها) کار می کند. برای هر مجموعه ، یک طبقه بندی متن با نمونه های باقی مانده (به عنوان مثال، 75 درصد نمونه ها) آموزش داده می شود. در مرحله بعد، طبقه بندی کننده ها مجموعه های مربوطه خود را پیش بینی می کنند و نتایج با برچسب های حاشیه نویسی شده انسان مقایسه می شوند. این امر تعیین می کند که یک پیش بینی درست بوده است (مثبت واقعی و منفی واقعی) و چه زمانی اشتباه کرده است (مثبت کاذب ، منفی کاذب). با این نتایج، می توانید معیارهای عملکردی را بسازید که برای ارزیابی سریع نحوه عملکرد خوب طبقه بندی کننده مفید هستند:

- صحت<sup>۱۸</sup>: درصد متونی که با برچسب صحیح طبقه بندی شده اند.
- دقت<sup>۱۹</sup>: درصد نمونه هایی که طبقه بندی کننده از تعداد کل نمونه هایی که برای یک برچسب پیش بینی کرده بود بدست آورد.
- بازیابی<sup>۲۰</sup>: درصد نمونه هایی که طبقه بندی کننده برای یک برچسب معین از تعداد کل نمونه هایی که باید برای آن تگ پیش بینی کرده بود ، پیش بینی کرد.
- امتیاز اف<sup>۲۱</sup>: میانگین هارمونیک دقت و فراخوانی.

---

<sup>16</sup> Hybrid Systems

<sup>17</sup> Cross Validation

<sup>18</sup> Accuracy

<sup>19</sup> Precision

<sup>20</sup> Recall

<sup>21</sup> F1-Score

## تشخیص و طبقه بندی عواطف

شاید متداول ترین مثال طبقه بندی متن، تشخیص و طبقه بندی عواطف باشد، که در آن منظور از تشخیص و طبقه بندی عواطف، فرایند خودکار خواندن یک متن برای عواطف نظر (خشم، شادی، وحشت، نفرت و فراتر از آن). شرکت ها از تشخیص عواطف در طیف گسترده ای از برنامه ها مانند تجزیه و تحلیل محصولات، نظارت بر نام تجاری، تحقیقات بازار، پشتیبانی مشتری، تجزیه و تحلیل نیروی کار و موارد دیگر استفاده می کنند.

تجزیه و تحلیل احساسات به شما امکان می دهد تمام اشکال متن را به طور خودکار برای احساس و عاطفه نویسنده تجزیه و تحلیل کنید.

تشخیص عواطف به ما کمک میکند تا بفهمیم که یک متن در مورد چه چیزی صحبت می کند. این شاخه یکی دیگر از موارد استفاده عالی برای طبقه بندی متن است که متن را برای درک دلیل بازخورد تجزیه و تحلیل می کند. شاید شکایت باشد، یا شاید مشتری قصد خرید یک محصول را دارد. از آن برای خدمات به مشتریان، بازاریابی پاسخ های ایمیل، تولید تجزیه و تحلیل محصولات و اتوماسیون شیوه های تجاری استفاده می شود. تشخیص عمیق با یادگیری ماشین می تواند ایمیل ها و مکالمات چت بات را بخواند و به طور خودکار آنها را به بخش صحیح هدایت کند.

در این پروژه ما از جایگزینی لغات فارسی مدل ترانسفورمی چند زبانه sentence-transformers استفاده کردیم. همچنین در این پروژه با پیاده سازی هر دو شبکه عصبی LSTM و متحد، ادعای مطرح شده در مقاله ((Kim 2014) را مبنی بر سرعت بالاتر همگرایی شبکه LSTM و دقت بالاتر شبکه متحد سنجیدیم. همانطور که در نتایجی که در فایل Exa CoLab.ipynb میتوانید ببینید، شبکه ی LSTM سریع تر همگرا میشود ولی نتایج آن برای داده های تست ضعیف تر از متحد است. شبکه متحد نیاز بیشتری به آموزش دارد. به نظر میرسد با فراهم آوردن منابع بیشتر برای شبکه عصبی LSTM دقت آن در قسمت تست به شکل چشم گیری افزایش یابد.

در ادامه خروجی هایی از کد LSTM را میبینیم:

```
Precision: tensor([0.9975, 0.9334, 0.9579, 0.9800, 0.9919, 1.0000, 1.0000, 1.0000, 1.0000,
0.9852, 1.0000, 0.9706, 1.0000, 1.0000, 0.9706, 1.0000, 0.9724, 1.0000,
0.9767, 1.0000, 0.9863, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 0.0000],
dtype=torch.float64)
tensor(0.9716, dtype=torch.float64)

Recall: tensor([0.9812, 0.9951, 0.9847, 0.8750, 0.9903, 0.9600, 0.9773, 0.9910, 1.0000,
0.9589, 0.9623, 0.8919, 0.9865, 1.0000, 0.8684, 0.9857, 0.9758, 0.9862,
0.9767, 0.9000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
0.9474, 1.0000, 0.8750, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 0.0000],
dtype=torch.float64)
tensor(0.9571, dtype=torch.float64)

Accuracy : 0.9764960826804467
```

[18:42]

=====TRAIN=lstm=====

```
Epoch 10, Time: 0 mins: 4 secs
    Train Loss 1.897, Train Acc 40.450
    Valid Loss 1.781, Valid Acc 44.829
Epoch 20, Time: 0 mins: 4 secs
    Train Loss 1.252, Train Acc 63.358
    Valid Loss 1.129, Valid Acc 67.967
Epoch 30, Time: 0 mins: 4 secs
    Train Loss 0.770, Train Acc 79.467
    Valid Loss 0.655, Valid Acc 83.179
Epoch 40, Time: 0 mins: 4 secs
    Train Loss 0.487, Train Acc 88.513
    Valid Loss 0.408, Valid Acc 91.583
Epoch 50, Time: 0 mins: 4 secs
    Train Loss 0.341, Train Acc 93.567
    Valid Loss 0.295, Valid Acc 94.300
Epoch 60, Time: 0 mins: 4 secs
    Train Loss 0.262, Train Acc 95.117
    Valid Loss 0.232, Valid Acc 96.017
```

=====Test=lstm=====

```
Precision: tensor([0.1379, 0.2317, 0.3629, 0.0556, 0.2128, 0.0000, 0.2105, 0.0000, 0.0000,
0.2053, 0.0000, 0.0000, 0.0000, 0.0000, 0.1667, 0.0400, 0.0426, 0.1935,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.3750, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000],
dtype=torch.float64)
tensor(0.0497, dtype=torch.float64)

Recall: tensor([0.3200, 0.3851, 0.1557, 0.0556, 0.2778, 0.0000, 0.2500, 0.0000, 0.0000,
0.2981, 0.0000, 0.0000, 0.0000, 0.0000, 0.0278, 0.0500, 0.0667, 0.2308,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.6000, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000],
dtype=torch.float64)
tensor(0.0604, dtype=torch.float64)

Accuracy : 0.195
```

## References

- Brownlee. 2019. *Crash Course in Convolutional Neural Networks for Machine Learning*.  
<https://machinelearningmastery.com/crash-course-convolutional-neural-networks/>.
- Brownlee, Jason. 2019. *gentle introduction bag of words model*.  
<https://machinelearningmastery.com/gentle-introduction-bag-words-model/>.
- Dair-ai. 2018. *Deep Learning for NLP: An Overview of Recent Trends*. <https://medium.com/dair-ai/deep-learning-for-nlp-an-overview-of-recent-trends-d0d8f40a776d>.
- Jason. 2019. *Crash Course in Recurrent Neural Networks for Deep Learning*.  
<https://machinelearningmastery.com/crash-course-recurrent-neural-networks-deep-learning/>.
- Kim, Yoon. 2014. "Convolutional Neural Networks for Sentence Classification."  
<https://arxiv.org/pdf/1408.5882.pdf>.
- watson. 2016. *Biggest data challenges might not even know*.  
<https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know/>.