

مرتضی حاجی تبار فیروزجائی

morteza.mhf@gmail.com

در این کد شبکه عصبی بازگشتی را پیاده سازی خواهیم کرد. سه لایه RNN در Keras موجود است: SimpleRNN, LSTM و GRU. برای بیشتر اعداد عملی، از GRU یا LSTM استفاده می شود. LSTM از آن دو قدرتمند تر است اما هزینه محاسباتی بیشتری دارد. GRU به عنوان گزینه ساده تر با بار محاسباتی کمتر کاربرد دارد.

❖ شبکه عصبی بازگشتی در کلاسه بندی (classification)

- از مجموعه داده imdb در کتابخانه keras استفاده می کنیم. این مجموعه داده، نظرات درباره ی فیلم، از IMDB است که با احساس (مثبت/منفی) برچسب گذاری شده است. نظرات از قبل پردازش شده اند و هر نظر بصورت لیستی از کلمات (عدد صحیح) رمزگذاری شده است. برای سهولت، کلمات با توجه به پرتکرار بودن در کل مجموعه داده کد شده اند. بعنوان مثال عدد صحیح "۳" سومین کلمه ی پرتکرار را کد می کند.
 - در هنگام load کردن داده num_words را برابر ۶۰۰۰ قرار می دهیم به این معنی که ۶۰۰۰ از پرتکرار ترین کلمات را حفظ کرده و باقی مانده را حذف کند و همچنین بخش آموزش و تست را جدا می کنیم.
 - دنباله های ورودی را کوتاه کرده تا دارای طول یکسان باشند. از تابع sequence.pad_sequences استفاده می کنیم.
 - لایه ی Embedding (به عنوان لایه ی اول) و LSTM را برای ساخت شبکه به کار می بریم.
 - لایه ی Embedding را در ادامه توضیح می دهیم:
- پاسخ: ما روشی می خواهیم که نه تنها محتوای ی جمله بلکه متن آن را نیز مشخص کند. یک لایه embedding تلاش می کند معنی هر کلمه در جمله را با نگاشت هر کلمه به موقعیتی در فضای بردار تشخیص دهد. در واقع نوعی گروه بندی کردن کلمات مشابه در کنار هم است. برای مثال ممکن است good و great نزدیک هم قرار گیرند و bad و horrible نیز نزدیک هم قرار گیرند. و کلمات این دو مجموعه جدا از هم قرار گیرند که یعنی معنای بسیار متفاوتی دارند. همچنین اگر داشته باشیم:

Have a great day

Have a good day

کدگذاری آن اگر به صورت زیر باشد:

[0, 1, 2, 3]

[0, 1, 4, 3]

نگاشت ما به صورت $\{ \text{"Have"}, 1: \text{"a"}, 2: \text{"great"}, 3: \text{"day"}, 4: \text{"good"} \}$ خواهد بود. با ملاحظه این دو جمله انسان متوجه می شود دو جمله شباهت زیادی دارند اما با ملاحظه کدگذاری آن متوجه می شوید دو کلمه با هم متفاوت اند. مفهوم لایه Embedding به این صورت است. کدها نیز به پیوست است.