

به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



درس یادگیری عمیق و کاربردها تمرین شماره ۳

امیر محمد کریمی

۸۱۰۱۹۴۳۸۳

اردیبهشت ماه ۱۳۹۸

فهرست

سوال ۱ ۳

سوال ۲ ۴

سوال ۳ ۵

سوال ۴ ۸

سوال ۵ ۱۱

سوال ۶ ۱۳

سوال ۷ ۱۳

سوال ۸ ۱۳

سوال ۹ ۱۳

پیوست ۲۳

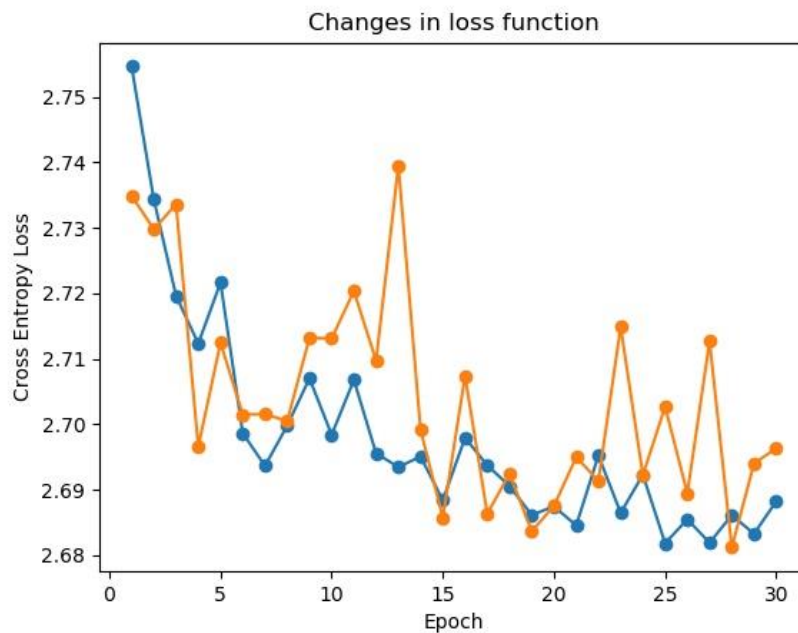
مراجع ۲۴

سوال ۱

در این قسمت، در ابتدا مطابق توضیحات گفته شده، شبکه‌ی مورد نظر را پیاده سازی می‌کنیم. این شبکه قرار است تا برای question answering به کار گرفته شود.

برای این تمرین از دیتاست DAQUAR استفاده می‌کنیم. ۸۰ درصد از داده‌ها را برای آموزش و ۲۰ درصد را برای validation به کار می‌بریم.

پس از طراحی شبکه مورد نظر، آن را آموزش می‌دهیم. میزان خطای شبکه در epoch ۳۰ به صورت زیر خواهد بود: (در تمامی تصاویر، نمودار آبی رنگ بیانگر خطای آموزش و نمودار نارنجی رنگ بیانگر خطای validation می‌باشد)

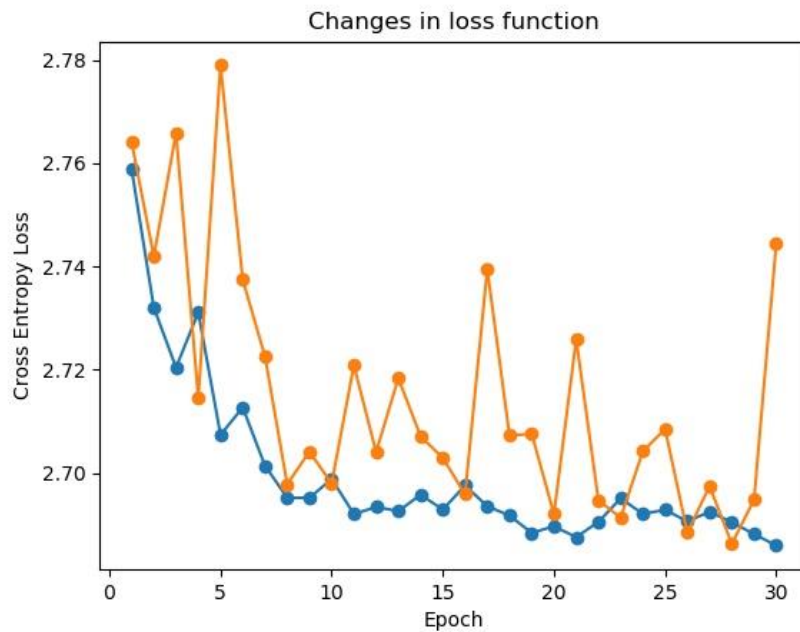


شکل ۱. میزان خطای شبکه در epoch ۳۰

با بررسی نمودار فوق مشاهده می‌کنیم که خطا با سرعت کمی در حال کاهش می‌باشد.

سوال ۲

در این بخش، به جای استفاده از مدل Resnet18، از Resnet34 استفاده می‌کنیم. نمودار میزان خطای شبکه به صورت زیر در می‌آید:

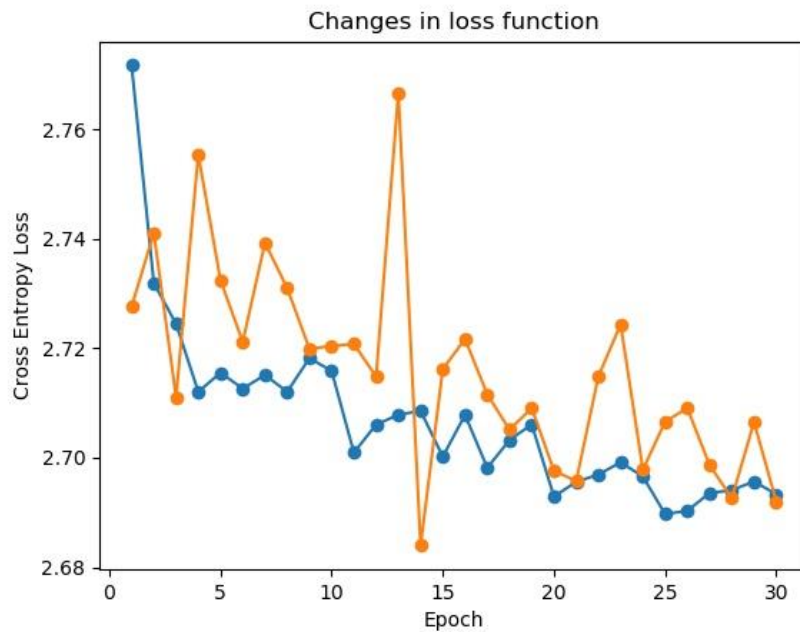


شکل ۲. میزان خطای شبکه در ۳۰ epoch

با مقایسه‌ی این نمودار با بخش اول مشاهده می‌کنیم که تفاوت زیادی در آموزش آن‌ها وجود ندارد اما در این بخش سرعت همگرایی (شیب نزولی نمودار) اندکی بهتر از حالت قبل خواهد بود (به دلیل وجود featureهای دقیق‌تر و بهتری در خروجی بخش تصویر)

سوال ۳

در این بخش، از شبکه‌ی densenet121 استفاده می‌کنیم. خروجی خطا به صورت زیر خواهد بود:

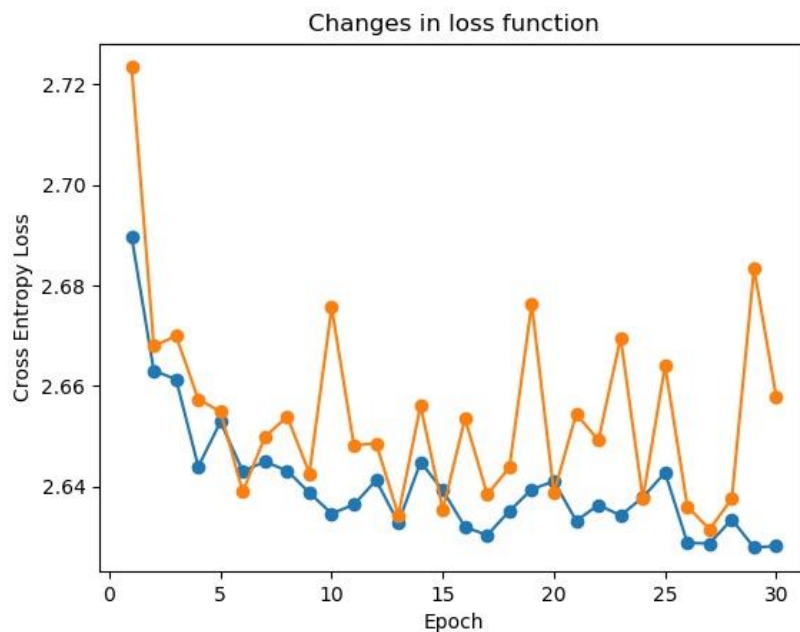


شکل ۳. میزان خطای شبکه در ۳۰ epoch

با بررسی این خطا با خطای قسمت اول، مشاهده می‌کنیم که روند آموزش کندتر و میزان خطا به طور کلی اندکی بیشتر است.

سوال ۴

در این قسمت، از glove42b به جای glove6b استفاده می‌کنیم (که تعداد کلمات بیشتر و دقت بیشتری دارد). نمودار خطا در این حالت به صورت زیر است:

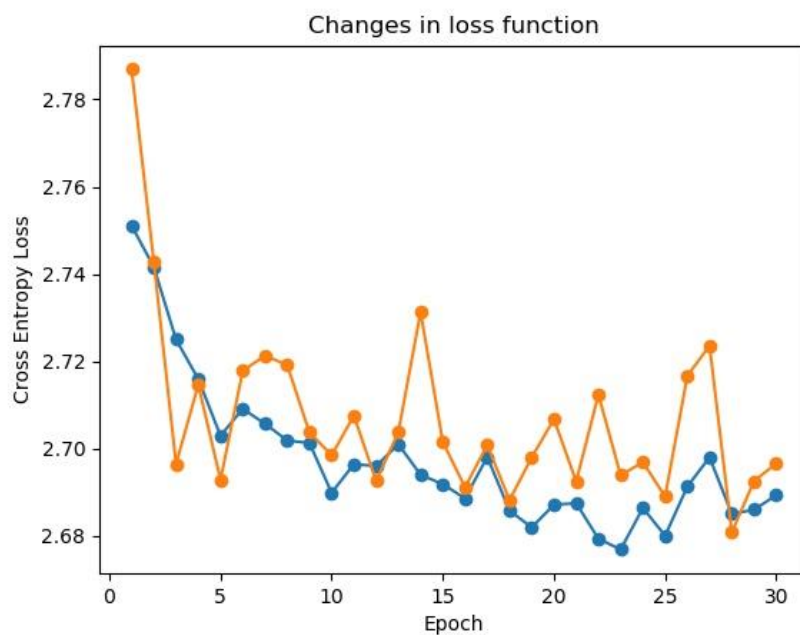


شکل ۴. میزان خطای شبکه پس از ۳۰ epoch

با مقایسه‌ی این شبکه با شبکه‌ی بخش اول مشاهده می‌کنیم که میزان خطا کمتر و همچنین سرعت همگرایی نیز بیشتر می‌باشد.

سوال ۵

در این بخش، به جای استفاده از GRU از سلول‌های LSTM استفاده می‌کنیم (طبق نتایج موجود در کاربردهای مختلف از این دو نوع سلول، توقع تغییر چندانی در خطا نخواهیم داشت). تغییرات خطا به صورت زیر در می‌آید:

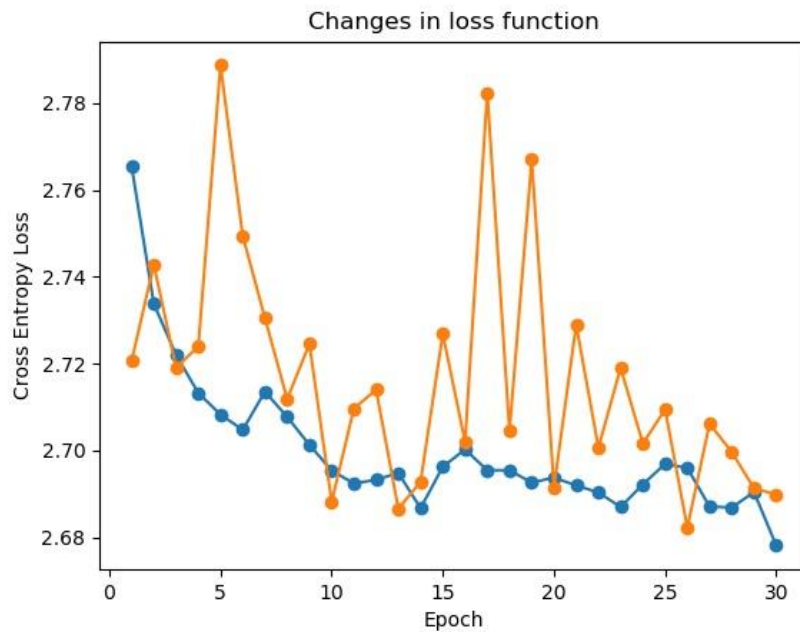


شکل ۵. میزان خطای شبکه در ۳۰ epoch

مطابق انتظار، خطا در این حالت تفاوت چندانی با بخش اول نخواهد داشت.

سوال ۶

در این قسمت، پارامتر `hidden_size` را به ۱۰۰ می‌رسانیم و یادگیری شبکه را انجام می‌دهیم. خروجی خطای شبکه به صورت زیر خواهد بود:

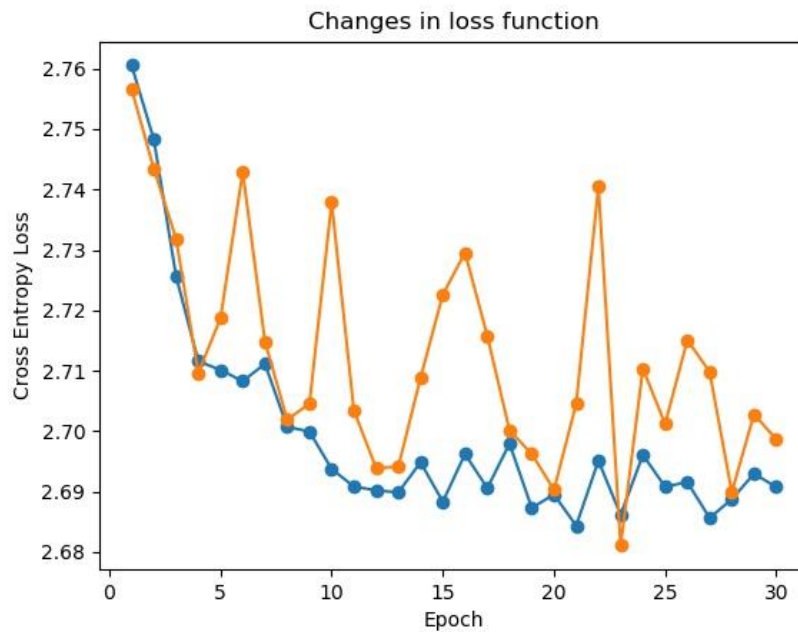


شکل ۶. میزان خطای شبکه در ۳۰ epoch

در این حالت نسبت به بخش اول، شبکه نوسانات بیشتری خواهد داشت اما در سرعت همگرایی تغییر بزرگی رخ نداده است.

سوال ۷

در این بخش، تعداد لایه‌های بخش Recurrent شبکه را به ۲ افزایش می‌دهیم. خروجی خطای شبکه به صورت زیر خواهد بود:

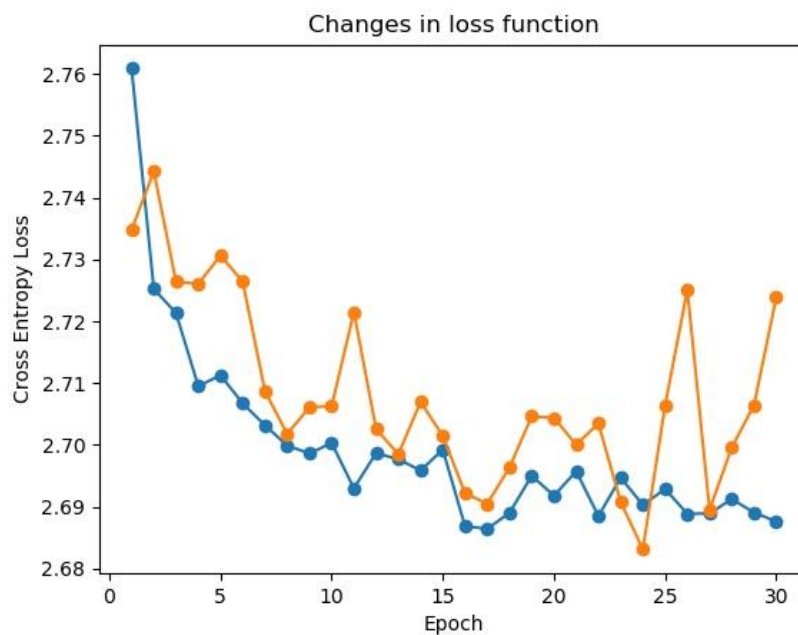


شکل ۷. میزان خطای شبکه در epoch ۳۰

در این بخش نیز نوسانات خطا افزایش یافته است اما سرعت همگرایی افزایش یافته است و میزان خطا نیز به میزان کمی کاهش یافته است.

سوال ۸

در این قسمت، سلول‌های GRU را به صورت دو طرفه تعریف می‌کنیم. شبکه‌های دو طرفه‌ی Recurrent بدین صورت هستند که از ۲ لایه‌ی hidden تشکیل شده‌اند که یکی از آن‌ها به صورت همگام با زمان و دیگری در خلاف جهت زمان به یکدیگر متصل هستند و بدین صورت، علاوه بر در نظر گرفتن ارتباط هر کلمه با کلمات قبلی، ارتباط آن با کلمات بعدی را نیز در نظر می‌گیرند. خروجی خطای شبکه به صورت زیر به دست آمده است:

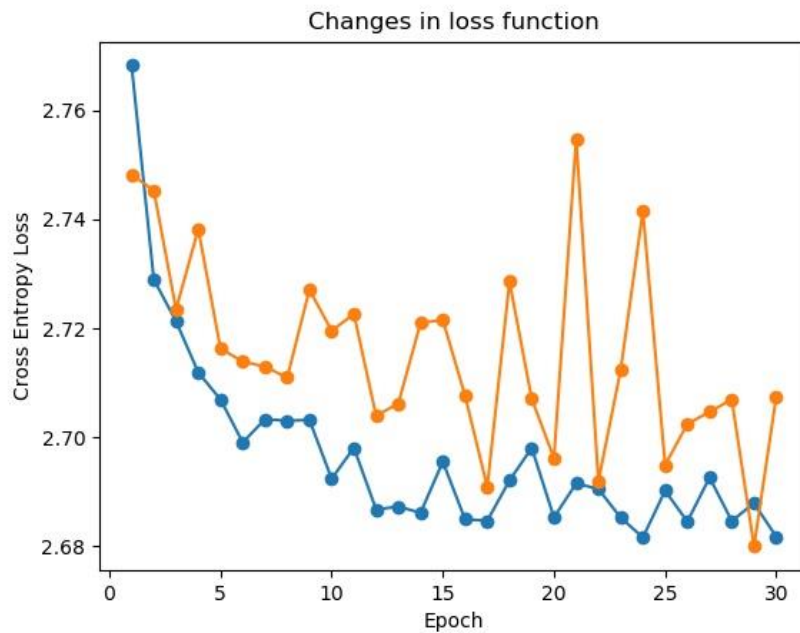


شکل ۸. میزان خطای شبکه در ۳۰ epoch

همانطور که قابل مشاهده است، در ای حالت سرعت همگرایی اندکی افزایش داشته است و همچنین میزان خطا نیز به صورت کلی نسبت به حالت اول کاهش یافته است.

سوال ۹

در این قسمت، به جای در نظر گرفتن ویژگی‌های تصویر به عنوان مقدار اولیه‌ی حالت‌های hidden شبکه‌ی Recurrent، آن را به همراه کلمات به ورودی شبکه می‌دهیم و مقدار اولیه‌ی حالت‌های hidden را ۰ در نظر می‌گیریم. خطای شبکه به صورت زیر در می‌آید:



شکل ۹. میزان خطای شبکه در ۳۰ epoch

با مقایسه‌ی نتایج این بخش با بخش اول، تفاوت چندانی نمی‌بینیم و تنها میزان خطای کلی شبکه اندکی افزایش داشته است. سرعت همگرایی تغییر خاصی نکرده است.

پیوست 1: روند اجرای برنامه

برای اجرای این پروژه، در ابتدا لازم است تا فایل `utils.py` را اجرا کنیم (این فایل، یک سری از آماده‌سازی‌های داده‌ها را انجام می‌دهد).

سپس، کفایت تا فایل `train.py` را اجرا کنیم تا شبکه شروع به آموزش کند.

تنها لازم است که فایل مربوط به `glove6b` و `glove42b` در پوشه‌ای به نام `glove` در کنار پوشه‌ی مربوط به کدها موجود باشد. همچنین دیتاست مورد نظر نیز باید در کنار پوشه‌ی مربوط به کدها باشد.

[1] Pytorch.org. (2019). *torch.nn — PyTorch master documentation*. [online] Available at: <https://pytorch.org/docs/stable/nn.html> [Accessed 20 May 2019].

[2] Medium. (2019). *How to use Pre-trained Word Embeddings in PyTorch*. [online] Available at: <https://medium.com/@martinpella/how-to-use-pre-trained-word-embeddings-in-pytorch-71ca59249f76> [Accessed 20 May 2019].

[3] Pennington, J. (2019). *GloVe: Global Vectors for Word Representation*. [online] Nlp.stanford.edu. Available at: <https://nlp.stanford.edu/projects/glove/> [Accessed 20 May 2019].