به نام خدا

دانشکده مهندسی مکانیک

نام درس: هوش مصنوعی

تمرین ۴ (شبکه بازگشتی)

استاد درس: دکتر شریعت‌پناهی

دانشجو:

مهدی نوذری

۸۱۰۶۰۱۱۳۹

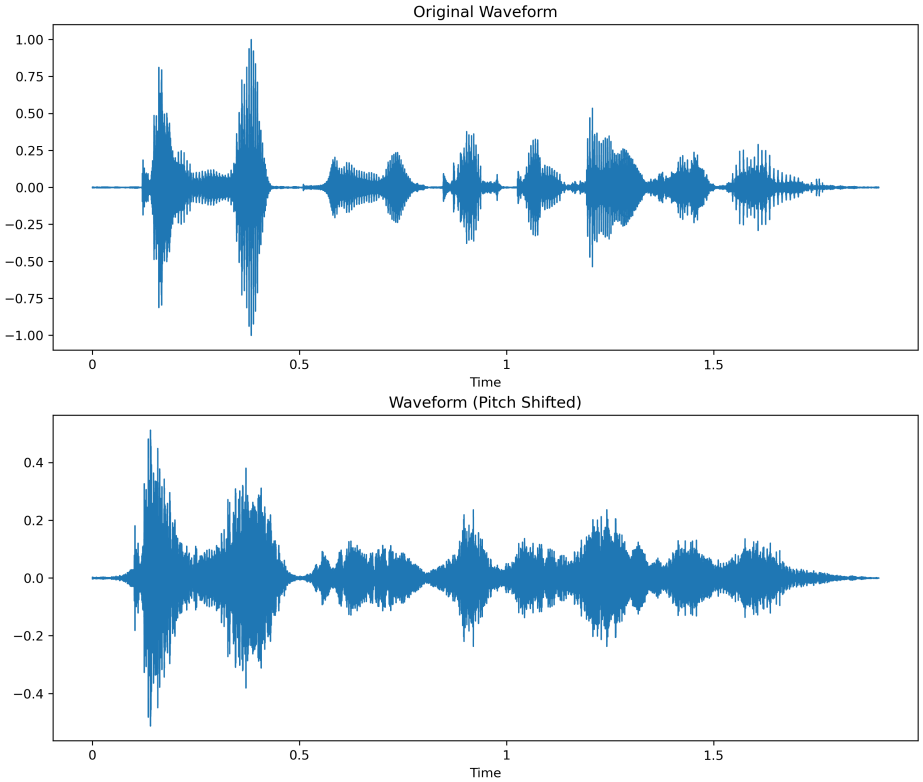
بهار ۱۴۰۳

تمامی فایل‌ها در Github موجود هستند: https://github.com/Morphit/UT_AI_1403

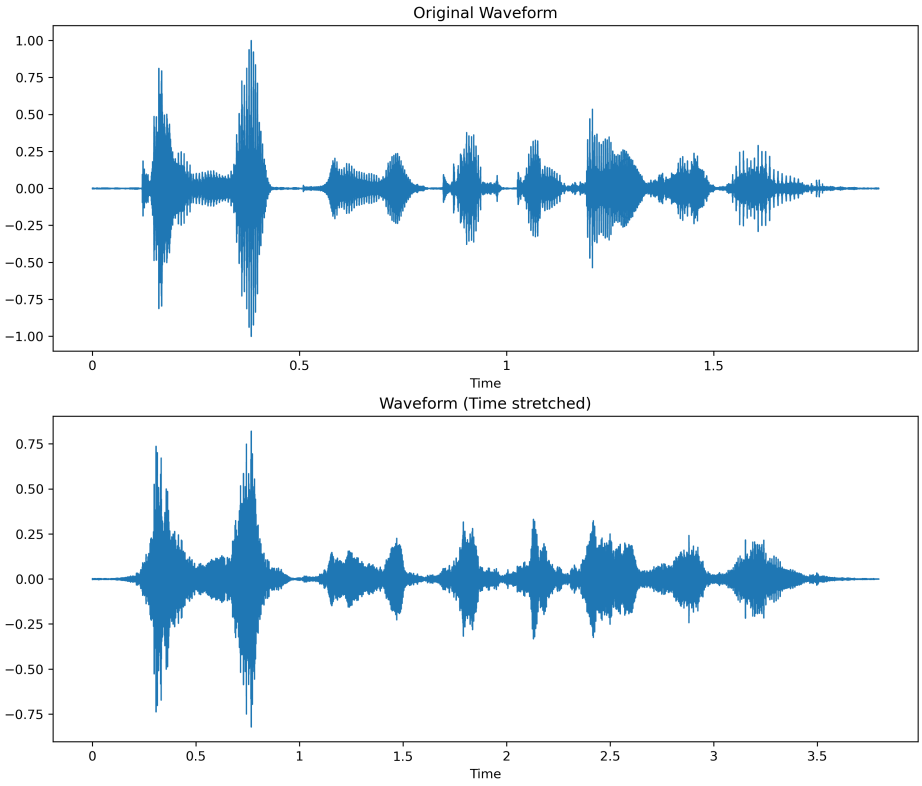
در این تمرین، هدف استفاده از تعدادی فایل صوتی شامل جملاتی به زبان آلمانی برای تربیت یک شبکه است که بتواند احساسات این جملات را تشخیص دهد. ابتدا داده‌ها باید گردآوری شده و در صورت نیاز داده‌افزایی صورت بگیرد، سپس با پیش‌پردازش داده‌ها ویژگی‌های سیگنال‌ها در طول زمان استخراج می‌شوند.

۱ گرداوری داده‌ها

در این بخش باید داده‌ها را گردآوری کرده و Data frame را تشکیل داد. چنانچه نام فایل‌ها را بررسی کنیم می‌توان دید که حرف ششم هر نام، برچسب آن را مشخص می‌کند و باید از آن برای برچسب‌زنی داده‌ها استفاده کرد. به این منظور در یک حلقه برای تمامی فایل‌ها ابتدا برچسب آن را پیدا کرده و داده‌های آن را به شکل یک سری زمانی استخراج می‌کنیم. با استفاده از این سری زمانی و دستورهای `pitch_shift` و `time_stretch` ۶ داده جدید تولید می‌شود که به ترتیب تن صدا و سرعت متفاوتی دارند اما برچسب یکسانی با گفتار اولیه دارند. می‌توان یک نمونه از هر تغییر را در شکل ۱ و شکل ۲ مشاهده کرد.



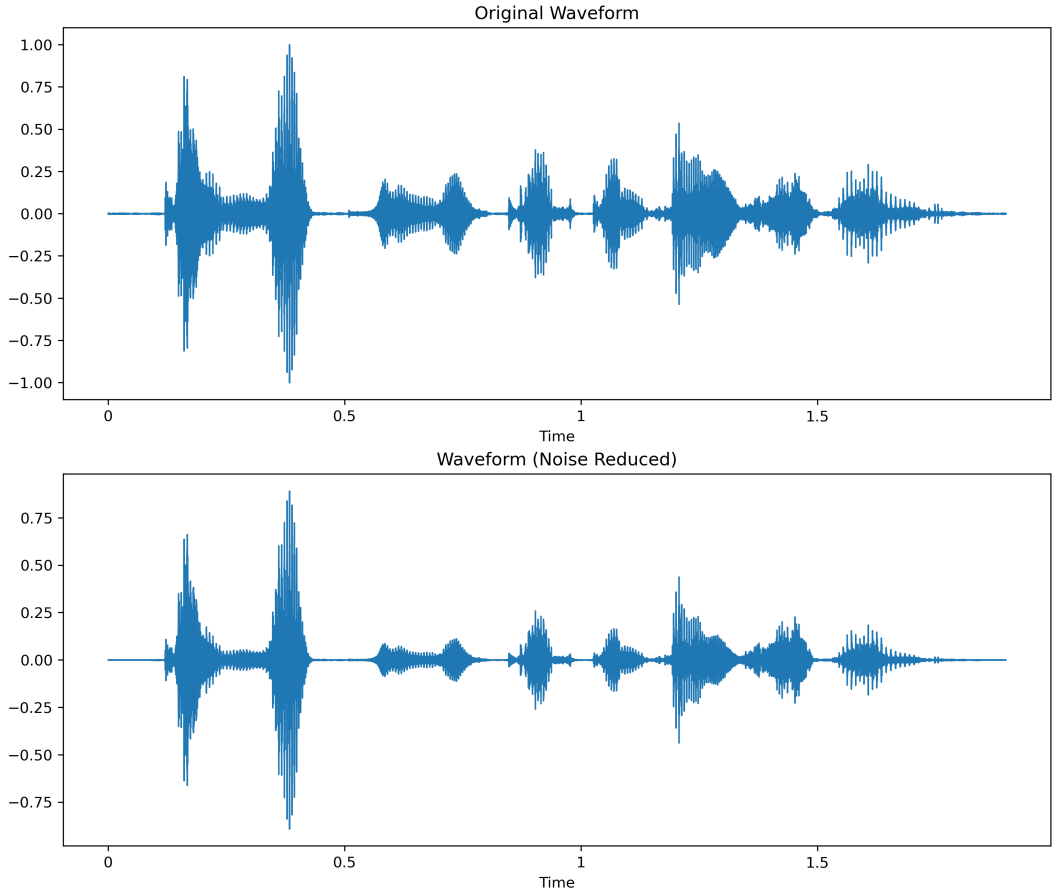
شکل ۱: نمونه از تغییر تن صدا



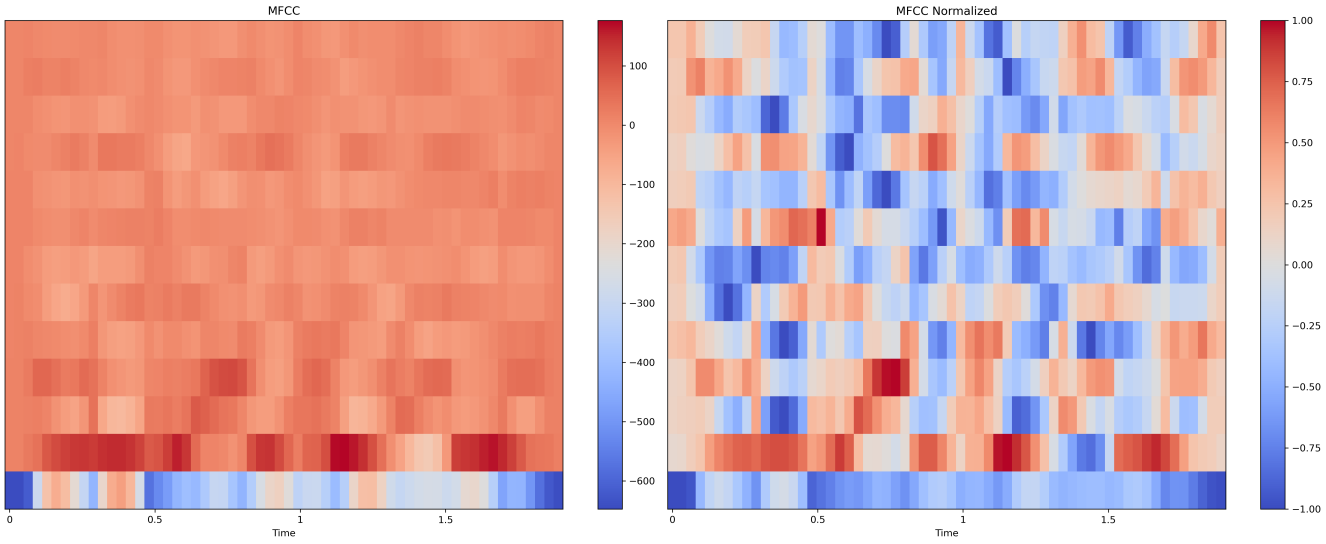
شکل ۲: نمونه از تغییر سرعت گفتار

۲ پیش‌پردازش داده‌ها

در این قسمت با استفاده از داده‌های سری زمانی فایل‌های صوتی، ویژگی‌های MFCC (Mel-Frequency Cepstral Coefficients) استخراج می‌شوند که بتوان آن‌ها را توسط شبکه پردازش کرد. این ویژگی‌ها شامل ۱۳ ویژگی هستند که در یک آرایه دو بعدی ذخیره می‌شوند. برای تولید این داده‌ها ابتدا نویز داده‌ها گرفته می‌شوند که می‌توان نمونه آن را در شکل ۳ مشاهده کرد. سپس به دو راهی که در بخش گذشته توضیح داده شد داده‌ها افزوده شده و نهایتا MFCC آن‌ها استخراج می‌شود. این ویژگی‌ها در مرحله بعدی نرمال‌سازی می‌شوند که نتیجه آن در شکل ۴ آمده است.



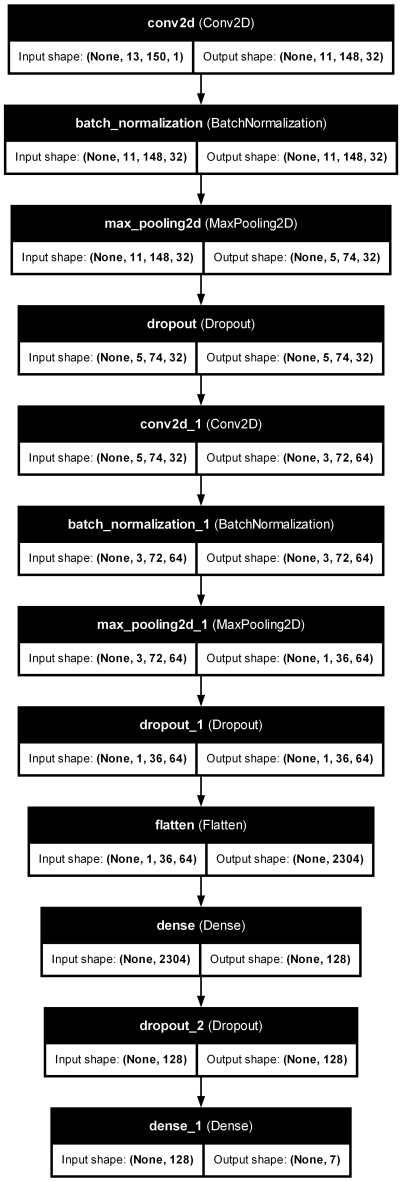
شکل ۳: نمونه از کم کردن نویز



شکل ۴: نمونه از ویژگی‌های MFCC و نرمال‌سازی آن‌ها

۳ شبکه CNN

برای پیش‌بینی در این قسمت از یک شبکه CNN به ساختار شکل ۵ استفاده می‌شود. این شبکه شامل لایه‌های پیچشی، انبایش، Dropout برای جلوگیری از اورفیت شدن و دارای Batch Normalization می‌باشد. ورودی شبکه به اندازه ویژگی‌های MFCC به طول ۱۵۰ داده می‌باشد (۱۳ در ۱۵۰). تمامی لایه‌های پنهان دارای تابع فعال‌سازی Relu بوده و لایه آخر از Softmaxاستفاده می‌کند. برای تربیت مدل از الگوریتم Adam با نرخ یادگیری 0.001 استفاده می‌شود. حین تربیت از دو تابع هزینه مختلف استفاده شده که نتایج هریک در ادامه آمده است.



شکل ۵: ساختار شبکه CNN

- **Categorical Cross Entropy**

این تابع هزینه به صورت زیر می‌باشد:

$$L_{\text{CCE}} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{۱}$$

این تابع از پر استفاده‌ترین تابع هزینه برای شبکه‌های عصبی است و برای داده‌های One-Hot استفاده می‌شود. این تابع برای مسائلی مناسب است که بیشتر از دو کلاس دارند.

- **Kullback-Leibler Divergence**

این تابع هزینه به صورت زیر می‌باشد:

$$L_{\text{KL}} = \sum_{i=1}^{C} y_i \log\left(\frac{y_i}{\hat{y}_i}\right) \tag{۲}$$

مشابه تابع قبلی، این تابع نیز برای مسائل چند کلاسه استفاده می‌شود. اما تفاوت ویژه‌ای که با آن دارن این است که برچسب‌ها لازم نیست به صورت One-Hot باشند. این ویژگی به داده‌ها اجازه می‌دهد که عدم قطعیت داشته باشند. برای مثال در این مسئله چنانچه بتوان گفتارها را در بیشتر از یک احساس دسته‌بندی کرد (برای مثال ۷۰ درصد خشم و ۳۰ درصد اضطراب) این تابع هزینه مفید خواهد بود. به این ترتیب حدس زده می‌شود که برای این مسئله کمی ضعیف‌تر از تابع Categorical Cross Entropy عمل کند.

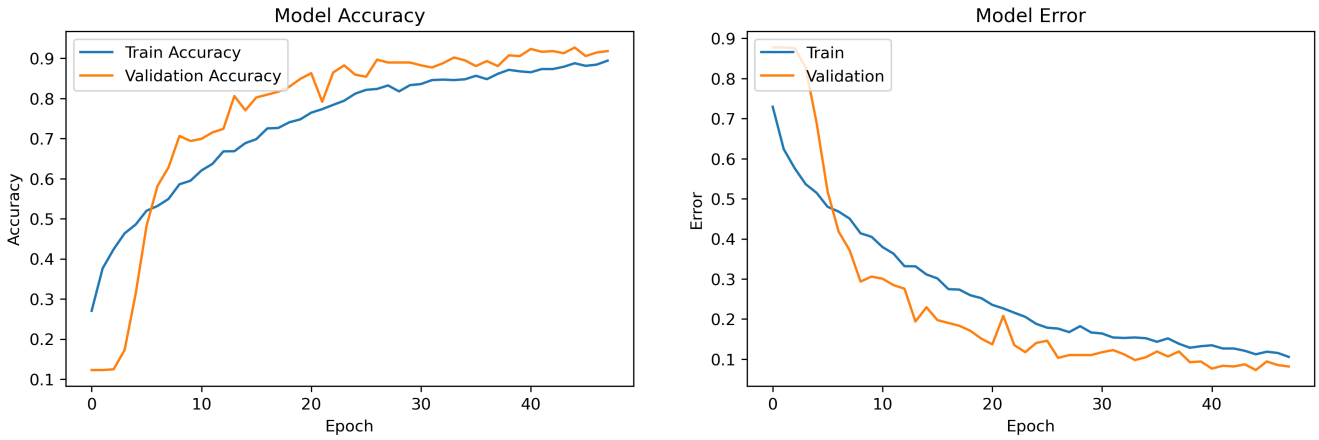
برای آموزش مدل از Early Stopping با تحمل ۶ ایپاک استفاده می‌شود. بعد از تربیت مدل‌ها، دقت روی داده‌های تست برای مدل‌ها با تابع هزینه در جدول زیر آمده است

جدول ۱: جمع‌بندی دقت مدل CNN با دو تابع هزینه متفاوت

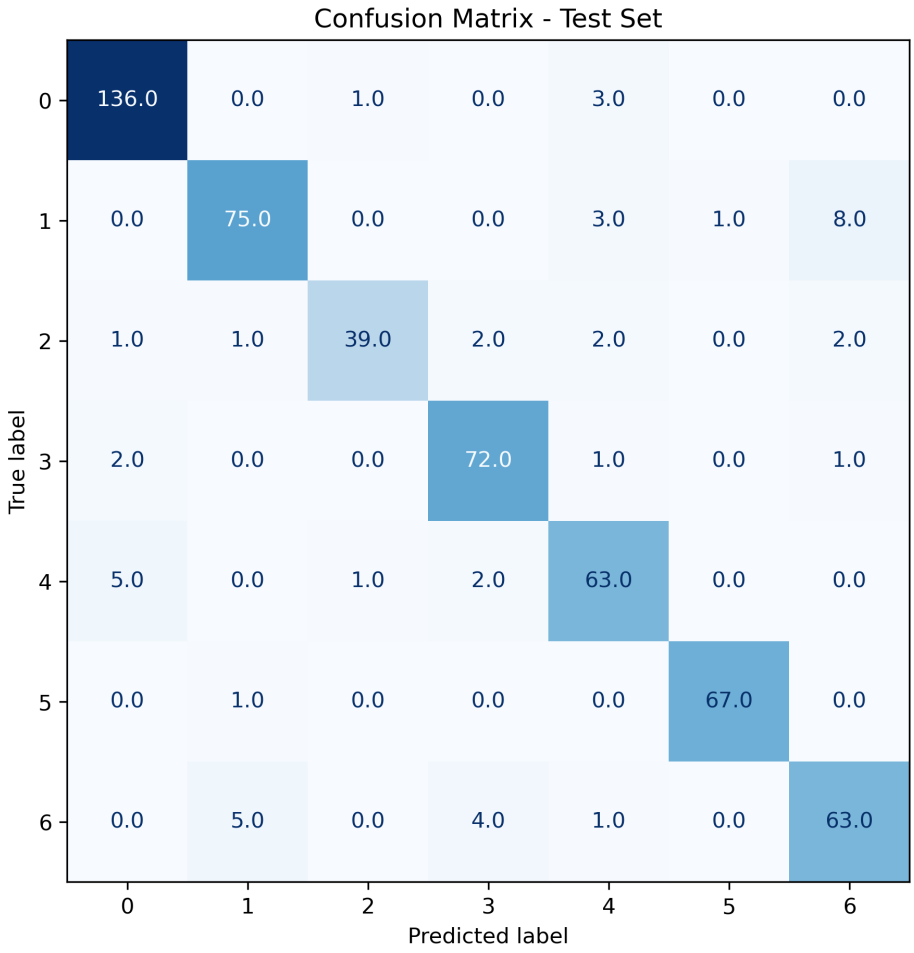
Loss function	**Epochs till convergence**	**Test accuracy**
Categorical Cross Entropy	47	0.9163
Kullback-Leibler Divergence	48	0.9003

به این ترتیب تابع Categorical Cross Entropy عملکرد بهتری هم در همگرایی و هم در دقت دارد. نمودار خطا و دقت مدل با این تابع هزینه در شکل ۶ و ماتریس آشفتگی آن در شکل ۷ آمده است.

می‌توان در ماتریس آشفتگی مشاهده کرد که مدل دقت نسبتا خوبی داشته و اکثر کلاس‌ها به درستی پیش‌بینی می‌شوند. طبیعتا درصد دقت در کلاس‌هایی که داده‌ها در آن کمتر است، هم به دلیل تربیت ضعیف‌تر و هم به علت نسبت بالاتر خطاها، پایین‌تر می‌آید. همچنین می‌توان دید که کلاس‌هایی که برای انسان نیز شبیه به هم هستند با هم بیشتر اشتباه گرفته می‌شوند. برای مثال بی‌حوصلگی (کلاس ۱) و بی‌تفاوتی (کلاس ۶) بیشتر از سایر داده‌ها به جای هم تشخیص داده شده‌اند.



شکل ۶: نمودار خطا و دقت – مدل CNN



شکل ۷: ماتریس آشفتگی – مدل CNN

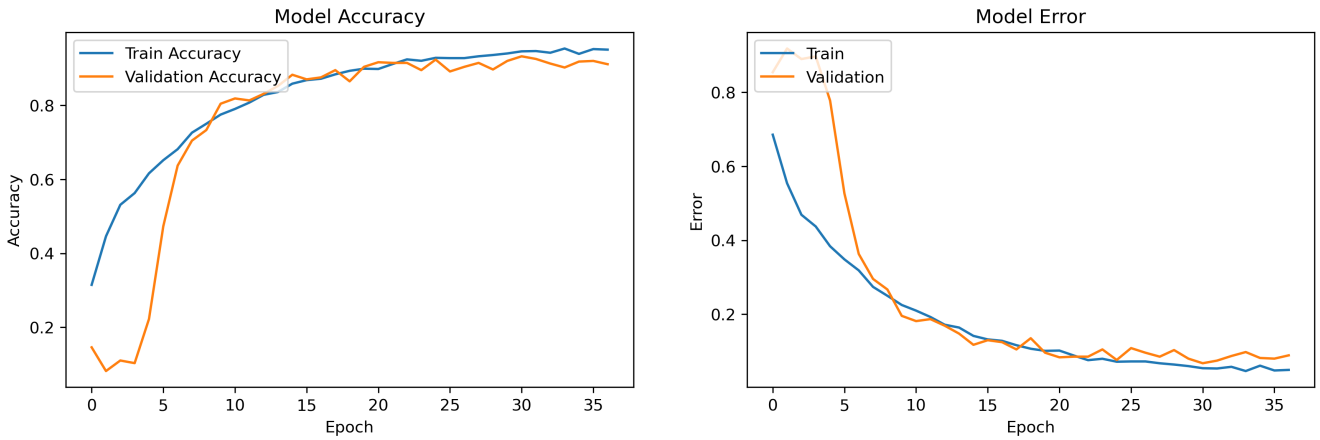
۴ شبکه CNN - LSTM

در این بخش از یک شبکه LSTM استفاده می‌شود که برای کاهش ابعادی، شبکه CNN بخش قبلی به آن اضافه شده است. به این ترتیب ساختار شبکه تقریبا همان شکل ۵ می‌باشد. با این تفاوت که بعد از آخرین لایه پیچشی و انبایشی، یک شبکه LSTM وجود دارد که در نهایت به سایر لایه‌های شبکه متصل می‌شود. دقت مدل به ازای دو تابع هزینه مختلف توضیح داده شده به صورت زیر می‌باشد که باز هم تابع اول بهتر عمل می‌کند.

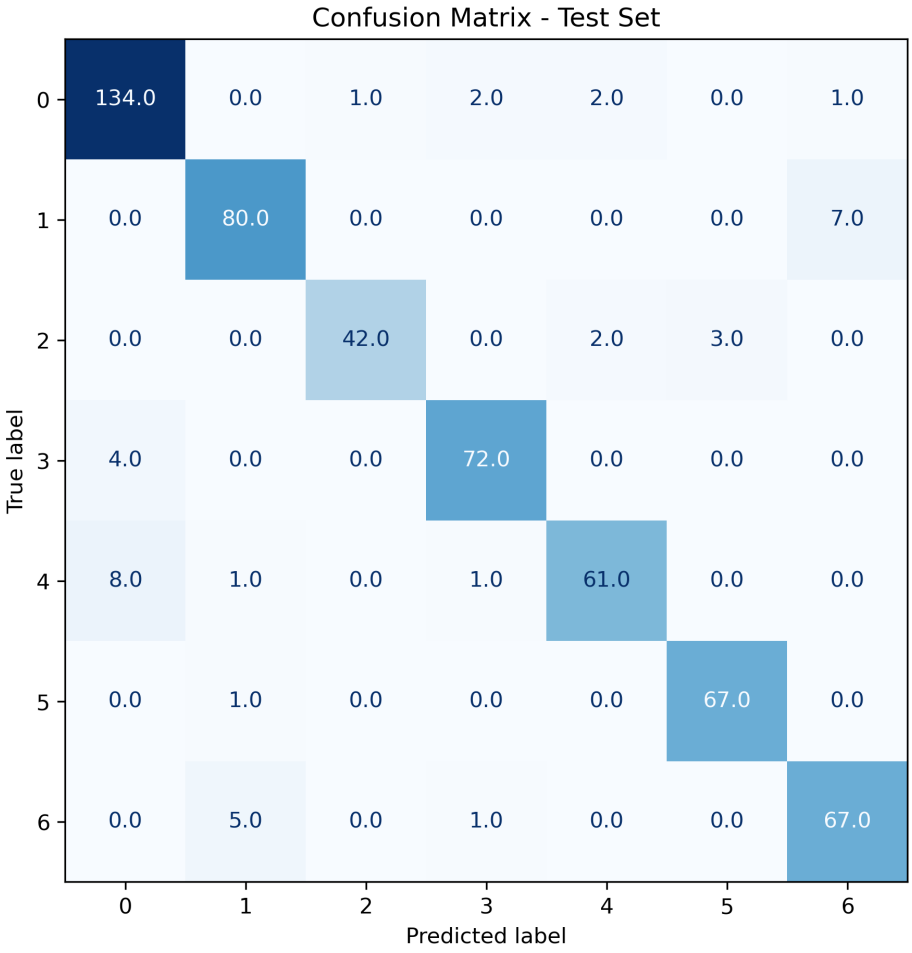
جدول ۲: جمع‌بندی دقت مدل CNN-LSTM با دو تابع هزینه متفاوت

Loss function	**Epochs till convergence**	**Test accuracy**
Categorical Cross Entropy	37	0.9306
Kullback-Leibler Divergence	35	0.9074

نمودار خطا و دقت در شکل ۶ و ماتریس آشفتگی آن در شکل ۷ آمده است.



شکل ۸: نمودار خطا و دقت - مدل CNN-LSTM



شکل ۹: ماتریس آشفتگی – مدل CNN-LSTM

۵ تحلیل نتایج

با مقایسه دقت نهایی و نمودار دقت در روند تربیت برای داده‌های تربیت و اعتبارسنجی می‌توان دو مدل CNN و CNN-LSTM را با هم مقایسه کرد. ابتدا از دقت که در جدول ۱ و جدول ۲ آمده است می‌توان دریافت که همانطور که انتظار می‌رفت شبکه CNN-LSTM به خاطر بررسی ترتیب داده‌ها که در داده‌هایی مانند صوت حائز اهمیت است بهتر عمل می‌کند. علاوه بر این موضوع، چنانچه به شکل شکل ۶ و شکل ۸ توجه کنیم، می‌توانیم متوجه شویم که در حین روند تربیت، نه تنها همگرایی سریع‌تر رخ داده، بلکه دقت داده‌های اعتبارسنجی نزدیک‌تر به داده‌های آموزش است. این موضوع در کنار دقت بالاتر مدل به تعمیم‌پذیری بهتر مدل CNN-LSTM اشاره دارد.

در کنار این‌ها، با بررسی ماتریس آشفتگی می‌توان دید که مدل دارای شبکه LSTM توانسته در ۶ کلاس از کل ۷ کلاس مقدار بیشتری کلاس صحبح را پیش‌بینی کند. همچنین می‌توان مشاهده کرد که این مدل برای برخی از کلاس‌ها که به جای هم تشخیص داده می‌شدند، رفتار بهتری دارد.