به نام خدا

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تمرین درس شبکه‌های عصبی–سری دوم

فردین آیار

شماره دانشجویی: ۹۹۱۳۱۰۴۰

استاد: دکتر صفابخش

دانشکده کامپیوتر– زمستان ۹۹

۱) شبکه عصبی پرسپترونی چند لایه، همانطور که در شکل ۱ نشان داده شده است، از یک لایه ورودی، یک لایه خروجی و یک یا چند لایه مخفی تشکیل شده است. هر لایه شامل تعدادی واحد پرسپترون(نرون) با تابع فعالسازی مشخصی است.خروجی هر نرون به تمام نرون‌های لایه بعدی متصل است و از این رو به آن شبکه کاملا متصل گفته می‌شود. به طور کلی آموزش این شبکه ها بر اساس انتشار خطای لایه خروجی(که مقدار آن به سادگی محاسبه می‌شود) به لایه‌های پیشین است. به طور دقیق، الگوریتم آموزش شبکه‌های پرسپترونی چند لایه به این صورت است:(از اثبات روابط صرف‌نظر می‌شود)

۱) تعیین مقادیر اولیه وزن‌ها بین همه نرون‌ها.

۲) مراحل بعدی تا موقعی که شرط توقف برقرار نیست تکرار شود.

۳) برای تمام ورودی‌ها S_p وخروجی متناظر با آن T_p مراحل بعدی تکرار شود.

۴) خروجی همه نرون‌های لایه مخفی و خروجی محاسبه شود. $y_{pj}^{(l)}$ خروجی نرون j از لایه l برای p–امین الگوی ورودی و $w_{ij}^{(l)}$ وزن بین نرون i از لایه l-1 و نرون j از لایه l است.

$$y_{pj}^{(l)} = f_j\left(\sum_i w_{ij}^{(l)} y_{pi}^{(l-1)}\right) \qquad y_{pi}^{(l-1)} = S_{pi} \;\; for \;\; l = 1$$

۵) مقدار دلتا برای نرون‌های لایه خروجی محاسبه شود. I_{pj} خروجی خالص j–امین نرون خروجی و f تابع فعالسازی است.

$$\delta_{pj} = (t_{pj} - y_{pj}) f'_j(I_{pj})$$

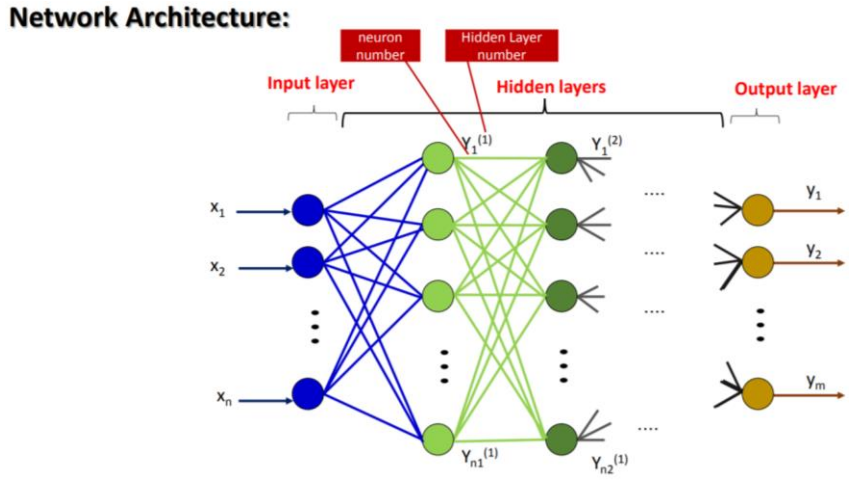
۶) دلتا برای نرون‌های لایه مخفی محاسبه شود.

$$\delta_{pj}^{(l)} = \left(\sum_k \delta_{pk}^{(l+1)} w_{jk}^{(l+1)}\right) f'_j\left(I_{pj}^{(l)}\right)$$

۷) همه وزن‌ها طبق فرمول زیر بروز شود. γ ضریب یادگیری است.

$$w_{ij}^{new(l)} = w_{ij}^{old(l)} + \gamma \delta_{pj}^{(l)} y_{pi}^{(l-1)}$$

۸) بررسی شرط توقف



شکل ۱

۲) داده‌ها یک بار شافل و در فایل s_data.csv ذخیره شده‌اند. تقسیم‌بندی داده‌ها به مجموعه‌های آموزش، آزمون و اعتبار سنجی در کد مربوط به هر روش انجام شده است.

۳) برای حل مسئله به روش رگرسیون، یکی از توابع میانگین مربعات خطا یا میانگین قدرمطلق‌های خطا برای تابع هزینه مناسب است. لایه آخر شبکه یک نرون خواهد داشت و می‌توان از تابع فعالیت خطی برای آن استفاده کرد. همچنین برای افزایش سرعت همگرایی می‌توان دامنه برچسب‌های خروجی را کاهش داد. در انتها برای بدست آوردن معیارهای دقت و ماتریس درهم‌ریختگی، مقدار خروجی به نزدیک‌ترین عدد صحیح گرد می‌شود. در طرف مقابل، حل مسئله به روش دسته‌بندی، مستلزم استفاده از تابع فعالیت softmax برای نرون‌های لایه آخر خواهد بود(تعداد نرون‌های خروجی برابر با تعداد کلاس‌ها است). همچنین اگر بخواهیم از روش one-hot برای خروجی شبکه استفاده کنیم، باید برچسب‌های خروجی را به دامنه صفر تا ((تعداد کلاس‌ها)) انتقال دهیم. در نهایت تابع، هزینه CategoricalCrossentropy برای آموزش شبکه دسته‌بند مناسب است.

هدف این مسئله پیش‌بینی سال انتشار موسیقی است. معمولا در این گونه مسائل، اگر پیش‌بینی دقیق ممکن نباشد، پیش‌بینی تقریبی نیز تا حد زیادی قابل قبول است؛ بنابراین حل مسئله به روش رگرسیون بهتر به نظر می‌رسد. به عنوان مثال اگر سال انتشار یک موسیقی ۲۰۰۹ باشد، پیش‌بینی ۲۰۱۰ برای آن نیز نسبتا قابل قبول است. رگرسیون به دلیل استفاده از خطای MSE(MAE) در اینگونه موارد نزدیک بودن پیش‌بینی را تشخیص داده و حتی اگر نتواند آن را بهبود ببخشد، سعی در نگه داشتن آن خواهد کرد. در طرف مقابل تابع هزینه دسته‌بند، پیش‌بینی ۲۰۱۰ یا ۱۹۹۰ را برای موسیقی که سال ۲۰۰۹ منتشر شده به یک اندازه نامناسب می‌بیند و هیچ امتیازی برای پیش‌بینی ۲۰۱۰ که به وضوح بهتر است، قائل نخواهد شد. به بیان بهتر، رگرسیون با توجه به ذات پیوسته بودن سال‌ها، مناسب‌تر است.

۴) در هر دو روش رگرسیون و دسته‌بندی، داده‌ها به سه مجموعه آموزش، آزمون و اعتبارسنجی با نسبت‌های داده شده در سوال تقسیم می‌شوند. سپس هر سه مجموعه به وسیله واریانس و میانگین مجموعه آموزش، نرمال سازی می‌شوند. همچنین برای یافتن پارامترهای مناسب، از صدهزار داده اول مجموعه آموزش استفاده می‌شود.

رگرسیون:

در جدول ۱ نتایج آزمایشات مختلف ارائه شده است. مجموع اعتبارسنجی در اینجا، ۲۰ درصد از صدهزار داده اول آموزش است. لایه خروجی شامل یک نرون با تابع فعالیت خطی و سایر لایه‌ها دارای تابع فعالیت relu هستند. همچنین از الگوریتم آدام برای بهینه‌سازی استفاده می‌شود.

جدول۱

	تعداد لایه‌های مخفی	تعداد نرون‌های لایه مخفی	ضریب یادگیری	تابع هزینه	تعداد ایپاک[۱]	mse_train	mse_vall
۱	۱	۱	۰.۰۱ ثابت	MSE	۷	151.0172	128.9282
۲	۱	۱	۰.۰۱۵ ثابت	MSE	۶	548.0469	101.6791
۳	۱	۱	۰.۰۰۵ ثابت	MSE	۱۴	107.9528	103.3172
۴	۱	۱۰	۰.۰۱ ثابت	MSE	۱۰	228.1830	119.3147
۵	۱	۱۰	۰.۰۰۱ ثابت	MSE	۳۵	96.1067	95.8543
۶	۱	۱۰	۰.۰۰۵ ثابت	MSE	۱۲	152.7461	123.8556
۷	۱	۱۰۰	۰.۰۰۵ ثابت	MSE	۱۵	302.7394	128.9227
۸	۱	۱۰۰	۰.۰۰۱ ثابت	MSE	۲۲	108.2803	110.7498
۹	۱	۱۰۰	۰.۰۰۰۵ ثابت	MSE	۴۲	95.2181	102.3608
۱۰	۲	۱۰–۱۰	۰.۰۰۰۵ ثابت	MSE	۳۱	100.3321	129.4816
۱۱	۲	۱۰–۱۰	۰.۰۰۰۱ ثابت	MSE	۹۵	91.6437	102.7306
۱۲	۲	۱۰–۱۰	۰.۰۴ متغیر	MSE	۱۸	88.7340	95.3480
۱۳	۲	۳۰۰–۱۰	۰.۰۴ متغیر	MSE	۲۵	81.8402	96.0496
۱۴	۲	۱۰–۳۰۰	۰.۰۴ متغیر	MSE	۱۵	85.3589	85.0344
۱۵	۲	۱۰–۳۰۰	۰.۰۱ متغیر	MSE	۲۴	86.6096	93.2714
۱۶	**۲**	**۱۰–۱۰۰۰**	**۰.۰۴ متغیر**	**MSE**	**۱۷**	**80.6562**	**81.7906**

[۱] شرط توقف الگوریتم، عدم بهبود loss_val با مقدار patience ۵ است.

۱۷	۲	۱۰۰۰–۱۰۰۰	۰.۰۴ متغیر	MSE	۲۹	64.3452	88.5496
۱۸	۲	۱۰۰۰–۱۰۰۰۰	۰.۰۴ متغیر	MSE	۲۶	64.8476	92.2482
۱۹	۴	۱۰–۱۰–۱۰–۱۰	۰.۰۴ متغیر	MSE	۲۲	85.1391	85.5339
۲۰	۴	۱۰–۱۰–۱۰–۱۰	۰.۱ متغیر	MSE	۲۶	81.6349	83.3764
۲۱	۴	۱۰–۱۰–۱۰–۱۰۰۰	۰.۰۴ متغیر	MSE	۱۶	78.9422	81.1356
۲۲	**۴**	**۱۰۰–۱۰۰–۱۰۰–۱۰۰۰**	**۰.۰۴ متغیر**	**MSE**	**۲۳**	**74.2097**	**80.3010**
۲۳	۴	۱۰۰۰–۱۰۰۰–۱۰۰۰–۱۰۰۰	۰.۰۴ متغیر	MSE	۳۴	68.5180	92.0512
۲۴	۴	۱۰۰۰–۱۰–۱۰–۱۰	۰.۰۴ متغیر	MSE	۲۶	80.1742	97.1564
۲۵	۴	۱۰–۱۰–۱۰–۱۰۰	۰.۰۴ متغیر	MSE	۱۹	81.1008	83.4398
۲۶	**۴**	**۱۰–۱۰–۹۰–۱۰۰**	**۰.۰۴ متغیر**	**MSE**	**۲۱**	**80.9029**	**81.8367**

در جدول فوق منظور از ((۰.۰۴ متغیر)) برای ضریب یادگیری، تعیین ضریب یادگیری در هر ایپاک از تابع زیر است. این رابطه با آزمون و خطا به گونه‌ای تعیین شده که در ایپاک‌های اول گام‌ها بزرگ و سپس گام‌ها به سرعت کوچک شوند.

```
def adapt_learning_rate(epoch):
    if epoch>5:
      return 0.04 / (epoch+1)**2
    else:
      return 0.04 / (epoch+1)
```

به طور کلی نکات زیر از جدول فوق قابل برداشت است:

۱) افزایش تعداد نرون‌ها و لایه‌های شبکه عصبی، موجب کاهش خطا در مجموعه آموزش خواهد شد. این امر در ابتدا خطای مجموعه اعتبارسنجی را کاهش و سپس به علت بیش‌برازش افزایش خواهد داد. (به عنوان مثال ردیف‌های ۱۶ و ۱۷ جدول فوق)

۲) در صورت استفاده از ضریب یادگیری ثابت، مقدار بهینه آن به شدت به ساختار شبکه وابسته خواهد بود و پیدا کردن آن مستلزم آزمون و خطای فراوان است. اما به عنوان یک قانون کلی، از نتایج جدول فوق این‌گونه استنباط می‌شود که ضریب یادگیری کوچکتر اگرچه سرعت یادگیری را کاهش می‌دهد اما در نهایت احتمالا به جواب‌های بهتری خواهد رسید.(به عنوان مثال ردیف‌های ۴، ۵ و ۶ جدول فوق)

۳) بهترین نتیجه در الگوریتم بهینه‌سازی زمانی حاصل می‌شود که گام‌های اولیه بسیار بزرگ باشد و گام‌های پس از آن به تدریج کوچک گردد تا نقطه بهینه با دقت بیشتری پیدا شود و در عین حال سرعت الگوریتم کاهش نیابد.در نتیجه استفاده از ضریب یادگیری متغیر، همان‌طور که توضیح داده شد، باعث بهبود دقت و سرعت الگوریتم بهینه‌سازی می‌شود.(به عنوان مثال ردیف ۱۱ و ۱۲ جدول فوق) مزیت بسیار مهم دیگر استفاده از ضریب یادگیری متغیر این است که نسبت به ساختار شبکه وابستگی کمتری دارد و بنابراین نیاز به تغییر آن در ساختارهای مختلف، کمتر است.(در جدول فوق تقریبا در تمام موارد متغیر، از مقدار اولیه ۰.۰۴ استفاده شده‌است. ردیف‌های ۲۰ و ۲۱ نشان می‌دهد که با تغییر مقدار اولیه ضریب یادگیری، سرعت و دقت الگوریتم تغییر چندانی نکرده است)

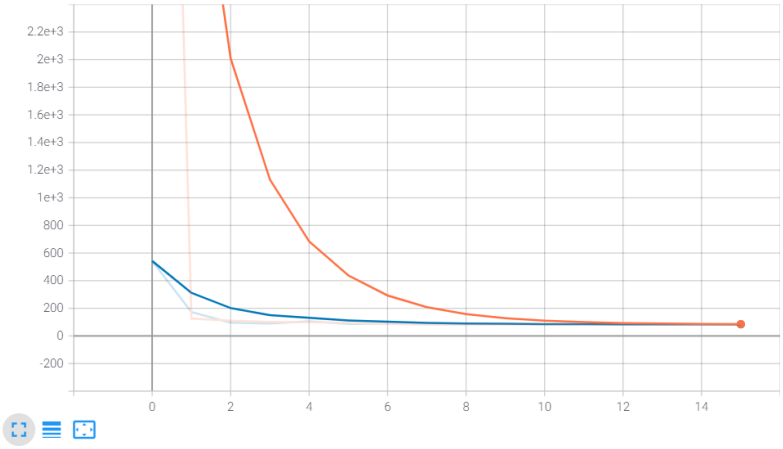
۴) افزایش تعداد نرون‌ها در لایه‌آخر نسبت به سایر لایه‌ها ارجح است. این مورد نشان می‌دهد که احتمالا ضرایب بالاتر از ویژگی‌ها در تعیین خروجی نقش بیشتری دارند. (به عنوان مثال ردیف‌های ۲۱، ۲۵ و ۲۶)همچنین در صورت افزایش نرون‌های لایه‌های ماقبل آخر، مدل به بیش‌برازش متمایل می‌شود. (به عنوان مثال ردیف‌های ۲۱ تا ۲۳)

سه مدل برتر در جدول فوق به صورت پررنگ مشخص شده‌اند. برای انتخاب مدل برتر، هم تعداد نرون‌ها و هم دقت مدل را در نظر می‌گیریم. با توجه به این معیارها به نظر می‌رسد مدل ۲۶ بهترین عملکرد را دارد. این مدل را روی کل داده‌ها آموزش می‌دهیم و علاوه بر معیارهای قبلی، MSE داده‌های آزمون را نیز محاسبه می‌کنیم. نتایج در جدول ۲ ارائه شده‌است. لازم به ذکر است ضریب یادگیری برای بهبود مدل، تغییر داده شده است.

	تعداد لایه‌های مخفی	تعداد نرون‌های لایه مخفی	ضریب یادگیری	تابع هزینه	تعداد ایپاک[۲]	mse_train	mse_vall	mse_test
۲۶	**۴**	**۱۰–۱۰–۹۰–۱۰۰**	**۰.۰۱ متغیر**	**MSE**	**۱۶**	**82.6877**	**83.1609**	**83.81086**

[۲] شرط توقف الگوریتم، عدم بهبود loss_val با مقدار patience ۵ است.

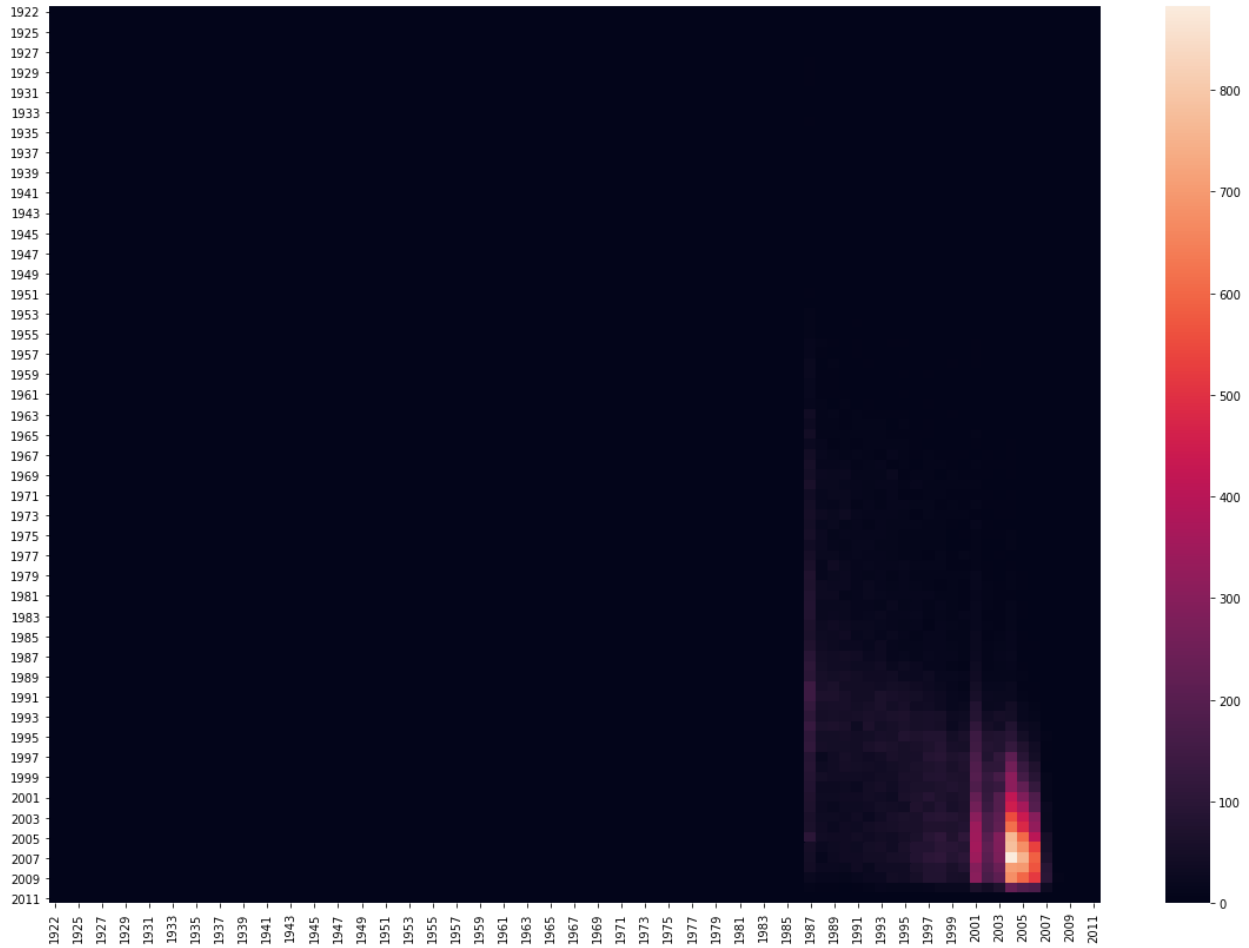
همانطور که مشاهده می‌شود، نتیجه روی داده‌های آزمون مشابه نتایج آموزش و اعتبارسنجی بوده و بنابراین، مدل تعمیم‌پذیری مناسبی دارد. اگرچه دقت بدست آمده حتی در بهترین مدل، نسبتا پایین است. خطای MSE ۸۰ به معنای این است که میانگین خطای پیش‌بینی هر داده احتمالا از ۶ سال بیشتر است که به هیچ‌وجه قابل قبول نیست. در ادامه، نمودار تابع هزینه MSE بر حسب تکرار را رسم می‌کنیم. با توجه به شکل ۲، هزینه مجموعه اعتبارسنجی پیش از توقف به یک مقدار حدی رسیده‌است اما به علت نوسانات زیاد، توقف آن به تاخیر افتاده است.



شکل ۲

در مرحله آخر، مجموعه آزمون به نزدیک‌ترین عدد صحیح گرد شده و دقت و ماتریس درهم‌ریختگی بدست آمده‌است. تحلیل نتایج این معیارها در بخش آخر ارائه خواهد شد.

```
Accuracy for test: 0.0636460657805375
```



شکل۳

دسته‌بندی:

برچسب داده‌ها را ب محدوده ۰ تا ۸۹ انتقال می‌دهیم. (روش one-hot) در این قسمت تابع هزینه CategoricalCrossentropy می‌باشد و در لایه خروجی از تابع فعالیت softmax استفاده می‌کنیم. سایز جزئیات مانند روش رگرسیون است. جزئیات آزمایش‌های مختلف در جدول ۲ ارائه شده است.

جدول۲

	تعداد لایه‌های مخفی	تعداد نرون‌های لایه مخفی	ضریب یادگیری	تابع هزینه	تعداد ایپاک[۳]	acc_train	acc_vall	Loss_val
۱	۱	۱	۰.۰۱ ثابت	CCE	۸	0.0751	0.0781	3.2875
۲	۱	۱	۰.۰۰۱ ثابت	CCE	۱۳	0.0794	0.0806	3.2568
۳	۱	۱	۰.۰۰۰۵ ثابت	CCE	۲۲	0.0803	0.0798	3.2567
۴	۱	۱۰	۰.۰۱ ثابت	CCE	۱۱	0.0797	0.0808	3.2118
۵	۱	۱۰	۰.۰۰۱ ثابت	CCE	۱۳	0.0862	0.0868	3.1534
۶	۱	۱۰	۰.۰۰۰۵ ثابت	CCE	۲۱	0.0891	0.0876	3.1483
۷	۱	۱۰۰	۰.۰۱ ثابت	CCE	۷	0.0802	0.0790	3.2405
۸	۱	۱۰۰	۰.۰۰۱ ثابت	CCE	۸	0.0977	0.0896	3.1456
۹	**۱**	**۱۰۰**	**۰.۰۰۰۵ ثابت**	**CCE**	**۱۱**	**0.1112**	**0.0915**	**3.1437**
۱۰	۱	۳۰۰۰	۰.۰۰۰۸ ثابت	CCE	۱۲	0.3489	0.0942	3.9778
۱۱	۲	۱۰–۱۰	۰.۰۰۱ ثابت	CCE	۲۷	0.0904	0.0891	3.1398

[۳] شرط توقف الگوریتم، عدم بهبود loss_val با مقدار patience ۵ است.

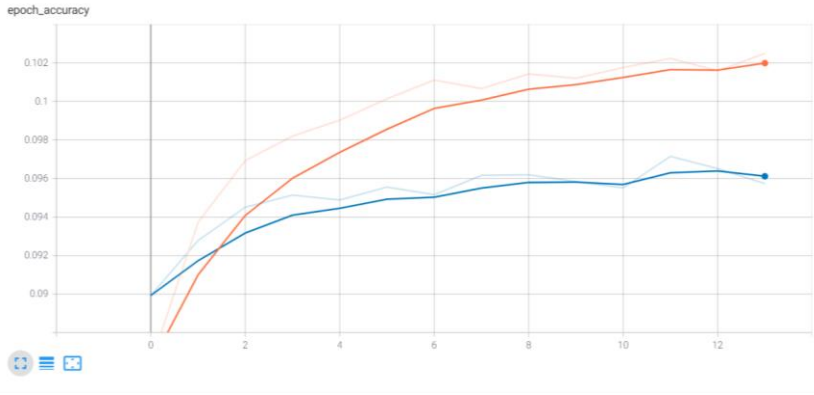
۱۲	۲	۱۰–۱۰	۰.۰۰۰۵ ثابت	CCE	۴۲	0.0902	0.0895	3.1364
۱۳	۲	۱۰–۱۰۰	۰.۰۰۱ ثابت	CCE	۱۲	0.0951	0.0890	3.1372
۱۴	۲	۱۰–۳۰۰	۰.۰۰۱ ثابت	CCE	۱۳	0.0942	0.0866	3.1424
۱۵	۲	۱۰–۳۰۰	۰.۰۰۰۵ ثابت	CCE	۱۹	0.0973	0.0892	3.1377
۱۶	۲	۳۰۰–۱۰	۰.۰۰۱ ثابت	CCE	۱۱	0.1029	0.0905	3.1280
۱۷	۴	۱۰–۱۰–۱۰–۱۰۰	۰.۰۰۱ ثابت	CCE	۲۴	0.0918	0.0887	3.1381
۱۸	۴	۱۰۰–۱۰۰–۱۰۰–۱۰۰	۰.۰۰۱ ثابت	CCE	۱۸	0.0977	0.0924	3.1056
۱۹	۴	۱۰۰۰–۱۰۰۰–۱۰۰۰–۱۰۰۰	۰.۰۰۱ ثابت	CCE	۱۰	0.0984	0.0927	3.1036
۲۰	۶	۱۰۰ * ۶	۰.۰۰۱ ثابت	CCE	۱۱	0.0969	0.0938	3.1041

به طور کلی نکات گفته شده در مورد رگرسیون در اینجا نیز صادق است، یعنی با کاهش ضریب یادگیری و یا افزایش تعداد ولایه‌های نرون‌ها، مقدار loss بیشتر کاهش می‌یابد.(ستون val_loss جدول فوق) اگرچه کاهش تابع loss لزوماً باعث افزایش دقت نمی‌شود و در نتیجه همانطور که در جدول مشاهده می‌شود، شبکه‌های عمیق‌تر، برتری خاصی نسبت به شبکه‌های تک لایه ندارند. همچنین به نظر می‌رسد در مدل دسته‌بند، مقدار ضریب یادگیری به ساختار شبکه وابستگی کمتری دارد و با تغییر ساختار شبکه، مقدار بهینه آن تغییر زیادی نکرده است، بنابراین در این مدل از ضریب یادگیری متغیر استفاده نمی‌کنیم. نکته مهم تر اینکه در این مدل، تعداد نرون‌های لایه اول بیشترین تاثیر را در بهبود دقت دارند. به عنوان مثال ردیف شماره ۱۰ نشان می‌دهد که با ۳۰۰۰ نرون در یک لایه مخفی، می‌توان به دقت ۳۴ درصد و بیشتر در آموزش رسید، در حالی که تابع loss همچنان مقدار نسبتاً بزرگی دارد. اگرچه فارغ از دقت داده‌های آموزش، دقت داده‌های اعتبارسنجی، به علت بیش برازش، هیچگاه از حدود ۱۰ درصد بالاتر نمی‌رود. این موضوع نشان می‌دهد که احتمالا الگوی دقیقی بین متغیرها و تابع خروجی وجود ندارد و مدل برای یادگیری، هیچ‌راهی به جز حفظ کردن ندارد!

در نهایت مدل ردیف ۹ را به عنوان مدل برتر انتخاب می‌کنیم. زیرا اگرچه دقت اعتبارسنجی آن از مدل شماره ۱۰ کمتر است، اما با توجه به کمتر بودن تعداد نرون‌ها، احتمالا بیش برازش در آن کمتر است. همچنین شبکه‌های عمیق‌تر، برتری چشم‌گیری (در دقت) نسبت به نرون‌های تک لایه ندارند، بنابراین از انتخاب آن‌ها صرف نظر می‌کنیم. لازم به ذکر است ضریب یادگیری برای بهبود مدل، تغییر داده شده است.

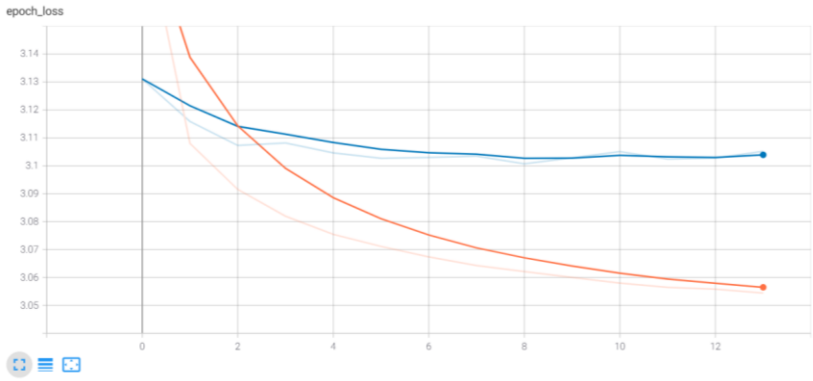
	تعداد لایه‌های مخفی	تعداد نرون‌های لایه مخفی	ضریب یادگیری	تابع هزینه	تعداد ایپاک[۴]	acc_train	acc_vall	acc_test
۹	۱	۱۰۰	۰.۰۰۰۸ ثابت	CCE	۱۴	0.1049	0.0957	0.0947

در شکل‌های ۴ و ۵ نمودارهای دقت و تابع هزینه برحسب تکرار برای این مدل ارائه شده است. با توجه به شکل ۵، هزینه مجموعه اعتبارسنجی قبل از توقف، به یک مقدار حدی رسیده است و بنابراین نقطه توقف الگوریتم نسبتا مناسب است. (اگرچه توقف الگوریتم می‌توانست بسیار زودتر صورت بگیرد)



شکل۴

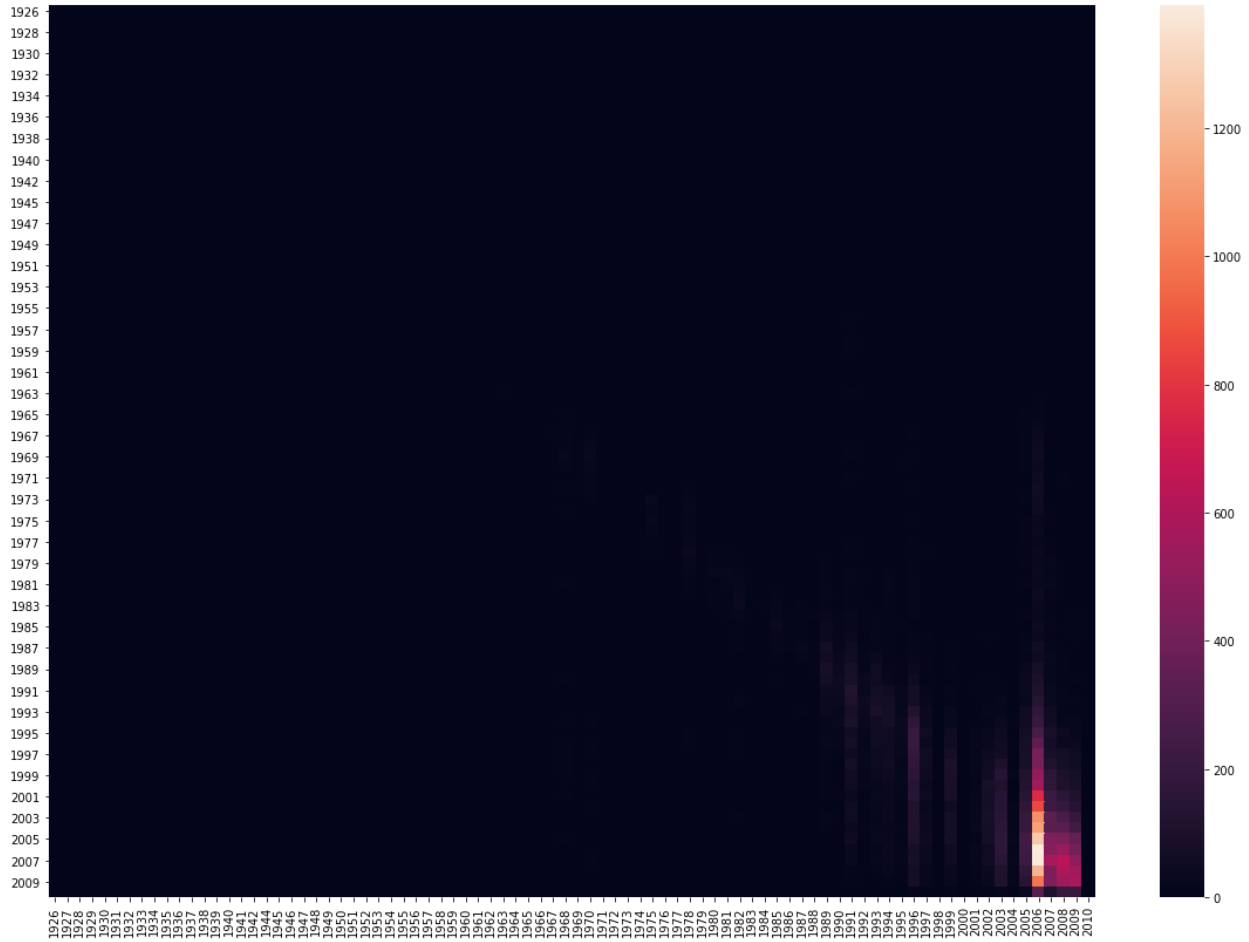
[۴] شرط توقف الگوریتم، عدم بهبود loss_val با مقدار patience ۵ است.



شکل ۵

در ادامه معیار دقت و ماتریس درهم‌ریختگی را برای این مدل ارائه می‌کنیم. تحلیل این نتایج در قسمت بعدی ارائه می‌شود.

```
Accuracy for test: 0.09473173571359271
```

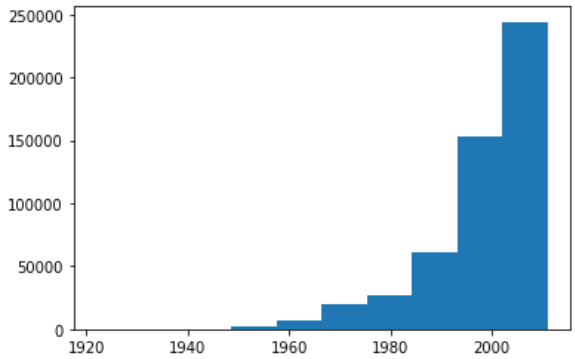


شکل ۶

مقایسه و نتیجه‌گیری:

نتایج مربوط به هر دو روش دسته‌بندی و رگرسیون بسیار ضعیف و غیرقابل قبول است. به هرحال روش دسته‌بندی دقت بیشتری دارد و از این نظر برتر است. همانطور که در سوال ۳ توضیح داده شد، در این‌گونه مسائل احتمالا پیش‌بینی تقریبی سال انتشار نیز ممکن است قابل قبول باشد.(کمینه بودن میانگین خطا) به عنوان مثال اگر سال انتشار موسیقی به جای ۲۰۱۰، اشتباها ۲۰۰۹ پیش‌بینی شود بهتر از پیش‌بینی ۱۹۹۰ برای آن است. تشخیص این مورد از روی مقدار دقت به تنهایی ممکن نیست و باید ماتریس درهم‌ریختگی را بررسی کنیم. هرچه عناصر اطراف قطر اصلی در ماتریس درهم‌ریختگی مقدار بیشتری داشته‌باشند یا اصطلاحا ماتریس قطری‌تر باشد، از این نظر بهتر است. در سوال ۲ پیش‌بینی می‌شد که مدل رگرسیون به دلیل استفاده از خطای MSE احتمالا از نظر میانگین خطا بهتر باشد، اما با توجه به ماتریس‌های درهم‌ریختگی، مدل دست‌بندی از این نظر هم بهتر است. (اصطلا قطری‌تر است)

در انتها بهتر است علت خالی بودن اکثر درایه‌های ماتریس‌درهم‌ریختگی را نیز بدانیم. با توجه به هیستوگرام برچسب‌های خروجی(شکل ۷)، علت این است که اکثر برچسب‌ها متعلق به ۲۰ سال آخر بوده‌اند و مدل‌ها به سمت این ۲۰ سال متمایل شده‌اند. در انتها مجددا یادآوری می‌شود که با توجه به نتایج، به نظر می‌رسد ویژگی‌های موجود برای پیش‌بینی سال انتشار موسیقی مناسب نیست. حتی بدون توجه به نتایج، انتظار می‌رود پیش‌بینی سال دقیق انتشار یک موسیقی، با توجه به شبیه‌بودن موسیقی‌هایی که در سال‌های متوالی منتشر می‌شود، تقریبا غیرممکن باشد.



شکل ۷