دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارشکار پروژه اول

مبانی هوش محاسباتی

تمرین پیاده‌سازی شبکه عصبی

هلیاسادات هاشمی‌پور

۹۸۳۱۱۰۶

توضیحات تکمیلی مربوط به هر مرحله در کد پوشش داده شده و حال به یک توضیحات کلی در مورد هر مرحله می پردازیم:

قدم اول : دریافت دیتاست، مصورسازی داده‌ها و پیش‌پردازش

در این مرحله طبق خواسته های سوال در هر بخش عمل کرده ام یعنی در ابتدا چهار دسته اول دیتاست را در قالب یک آرایه خواندم و با توجه به ابعادی که دارد آن ها(هم داده های تست و هم داده های train) را ست کردم.(به صورت ماتریسی) همچنین برحسب برچسب هایشان هم آن ها را ذخیره کردم در نهایت ۴ تا ماتریس پدید آمد.

حال برای کاهش پیچیدگی محاسباتی هم از کدد آماده دستور کار استفاده کردم. سپس داده های خود را با تقسیم بر ۲۵۵ نرمال کردم و با استفاده از دستور reshape(-1,1024). داده ها را flat می کنیم. یک تابع هم برای شافل کردن داده ها در نظر گرفتم.

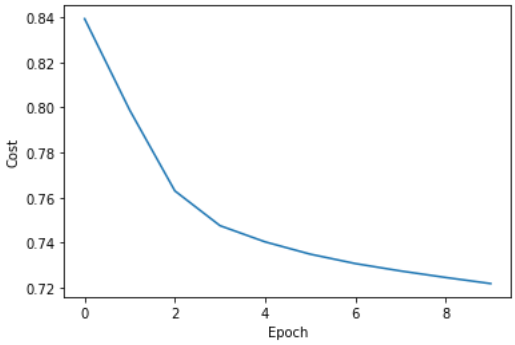
قدم دوم : محاسبه خروجی

با توجه به عملیات ذکر شده در دستور کار کد را پیاده سازی کردم و دقت حاصل تقریبا برابر با ۲۵ درصدشد.(کار با ماتریس هم با استفاده از Numpy صورت گرفت.)

```
Accuracy: 25.5%
```

قدم سوم : پیاده سازی Backpropagation

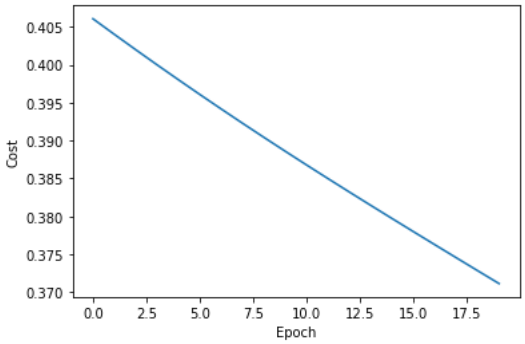
توضیحات مربوط به نحوه پیاده سازی این قسمت ضمیمه کد شده است. در نهایت دقت مدل و زمان اجرای فرایند یادگیری را چاپ کردم که به ترتیب به صورت : ۹۰ ثانیه و ۳۷ درصد هست که در دستور کار هم به شکل میانگین دقت را ۳۰ درصد فرض کرده است. .(با epoch، ۱۰)




```
Runtime of the program is 90.15610837936401 seconds
Accuracy:37.0%
```

قدم چهارم : پیاده سازی vectorization

نحوه پیاده سازی در این قسمت هم در دستورکار توضیح داده شده و فرق این قدم با قدم پیشین هم در استفاده نکردن از حلقه می باشد و در نهایت زمان اجرای بهتری حاصل می شود بانبراین به ترتیب دقت مدل و زمان اجرای فرایند به صورت: ۱ ثانیه و ۷۶ درصد است که همانطور که می بینیم زمان اجرای آن ۹۰ برابر زمان اجرای این قدم است.(با epoch، ۲۰)

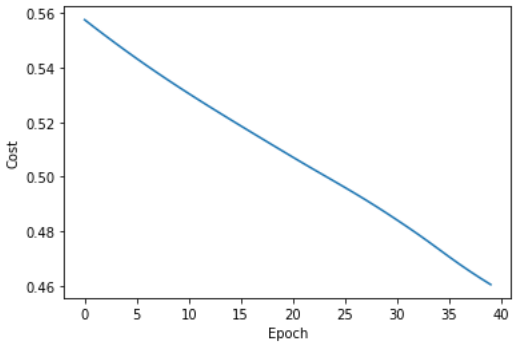



```
Runtime of the program is 1.2160513401031494 seconds
Accuracy:76.5%
```

قدم پنجم : تست کردن مدل

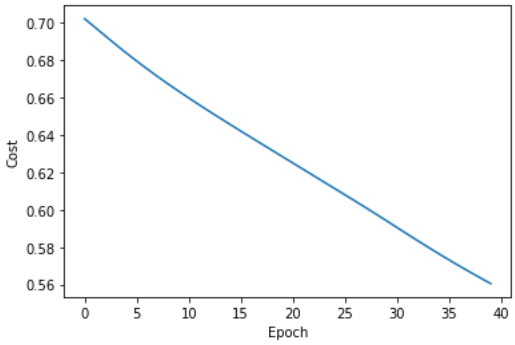
در این بخش هم با توجه به مقادیر داده شده در دستور کار به تست مدل خود می پردازیم و در آخر هم دقت مدل برای trainو test را چاپ می کنیم.

Bachpropagation:




```
Runtime of the program is 319.1710398197174 seconds
Accuracy of train:69.0%
Accuracy of test:41.0%
```

Vectorization:




```
Runtime of the program is 2.7594285011291504 seconds
Accuracy of train:56.00000000000001%
Accuracy of test:42.0%
```

امتیازی

بخش اول

۲-

Batch Normalization (BN)**تکنیکی برای بهبود عملکرد و پایداری شبکه های عصبی است. در واقع کار آن نرمال کردن توزیع دیتا می باشد.این کار برای نرمال سازی ورودی هر لایه با کمک میانگین و واریانس است. به شکلی که میانگین اعداد بردار نهایی برابر با صفر و واریانس آن ها برابر با ۱ میشود. (در اصل دیتای ما پس از نرمال شدن به شکلی می شوند که میانگین آن صفر شده و واریانس آن برابر یک می شود)**

۳-

Dropout **یعنی کنار گذاشتن بخش‌هایی از یک شبکه عصبی (برای کاهش** overfitting**) در واقع همانطور که می دانیم یک شبکه عصبی که شامل تعدادی نورون است و** Dropout **یعنی اینکه که در حین آموزش این نورون‌ها، از تعدادی از آن‌ها به صورت تصادفی چشم‌پوشی کنیم. چشم‌پوشی در انجا به این معنی است که آن نورون‌های خاص، در مسیر رفت یا برگشت در نظر گرفته نمی‌شوند. در این روش به ازای هر نورونی که داریم (به جز دو نورون آخر) یک عدد تصادفی بین صفر و یک تولید می‌کنیم حال مثلا نورون هایی که این عدد تصادفی آن ها از ۰.۵ کم تر باشد آن ها را علامت گذاری کرده و تمامی وزن های ورودی و خروجی را حذف می کنیم با این کار نقش نورون ها بلا استفاده شده است و یک شبکه سبکتری را داریم(بنابراین این شبکه ما قابلیت ایجاد اشکال پیچیده را ندارد)**

۴-

- AdaMax

همانطور که از نام آن پیداست AdaMax اقتباسی از بهینه ساز Adam است.

- SGD

الگوریتم بهینه سازی شیب نزولی تصادفی (SGD) در مقابل، یک به روز رسانی پارامتر را برای هر آموزش انجام می دهد.

SGD محاسبات اضافی را برای مجموعه داده های بزرگتر انجام می دهد، زیرا قبل از هر به روز رسانی پارامتر، گرادیان ها را برای همان مثال دوباره محاسبه می کند. به روز رسانی های مکرر را با واریانس بالا انجام می دهد که باعث می شود تابع هدف به شدت نوسان کند.

- Adagrad

Adagrad نرخ یادگیری را به طور خاص با ویژگی‌های فردی تطبیق می‌دهد: به این معنی است که برخی از وزن‌های موجود در مجموعه داده‌های شما نسبت به سایرین نرخ‌های یادگیری متفاوتی دارند. همیشه در یک مجموعه داده پراکنده که در آن ورودی های زیادی وجود ندارد بهترین کار را انجام می دهد.

- Adadelta

در اینجا دلتا به تفاوت بین وزن فعلی و وزن جدید به‌روز شده اشاره دارد. Adadelta استفاده از پارامتر نرخ یادگیری را به طور کامل حذف کرد و آن را با میانگین متحرک نمایی از دلتاهای مربع جایگزین کرد.

- RMSprop

یک نسخه انحصاری از Adagrad است که به جای اینکه اجازه دهد همه گرادیان ها برای حرکت انباشته شوند، فقط گرادیان ها را در یک پنجره تعمیر خاص جمع می کند.

- Adam

روش دیگری برای استفاده از گرادیان های گذشته برای محاسبه گرادیان های فعلی است، این روش با اضافه کردن کسری از مفهوم تکانه استفاده می کند. شیب قبلی نسبت به فعلی، عملاً در بسیاری از پروژه ها در طول آموزش شبکه های عصبی پذیرفته شده است.

۵-

تمرکز اصلی این معیار، بر روی درستی تشخیص‌های «بله» توسط الگوریتم است. در واقع معیار صحت (Precision)معیاری است که به ما می‌گوید الگوریتم چند درصد «بله»هایش درست بوداست فرمول آن به شکل زیر است:

$$Precision = \frac{TP}{TP + FP}$$

بر خلاف معیار Recall به دنبال محاسبه‌ی پوشش بر روی کل داده‌هاست. مرکز اصلی معیار recall بر روی داده‌هایی است که واقعا «بله» بوده‌اند.

$$\text{recall} = \frac{TP}{TP + FN}$$

معیار F۱ که در واقع ترکیب متعادلی بین معیارهای دقت و صحت است.

$$F - measure = \frac{2 \times (precision \times recall)}{precision + recall}$$

بخش دوم

۱-

data augmentation تکنیکی است که با استفاده از آن می توان به طور مصنوعی و بدون تصویربرداری جدید و با استفاده از ایجاد نسخه های دستکاری شده از تصاویر موجود اندازه دیتاست را افزایش داد. در واقع آموزش شبکه های عصبی مصنوعی عمیق با داده‌های بیشتر می تواند به داشتن شبکه های قوی تر منتهی شود. تکنیک های Image data augmentation می توانند گونه هایی از تصاویر را بسازند شبکه با یادگیری خصوصیات جدیدی از آن تصاویر علاوه بر تصاویر اصلی، قدرت درک گستره بیشتری از هر شی را پیدا می کند. این روش فقط در train استفاده می شود .از این روش برای افزایش اندازه مجموعه آموزشی و گرفتن تصاویر متفاوت بیشتر استفاده می شود. البته از نظر فنی، می‌توانیم از تقویت داده در مجموعه تست استفاده کنیم تا ببینیم مدل در چنین تصاویری چگونه رفتار می‌کند، اما معمولاً افراد این کار را انجام نمی‌دهند. (بنابراین این روش فقط در مجموعه آموزشی انجام می شود زیرا به تعمیم و استحکام مدل کمک می کند. بنابراین هیچ فایده ای برای افزایش مجموعه تست ندارد.)

انتخاب تکنیک data augmentation به کار رفته باید با دقت و مطابق با محتوای دیتاست و نوع مسئله انجام شود. می توان که بخش کوچکی از تصاویر آموزش را با شبکه ای کوچک آموزش داده و در این آموزش از تنظیم های مختلفی برای تولید تصاویر جدید استفاده کنیم و بهبود یا عدم بهبود شبکه را به طور مداوم بررسی کنیم.

کتابخانه یادگیری عمیق Keras قابلیت هایی را برای ساده کردن فرآیند آموزش شبکه ها با داده های این روش دارد.

تکنیک های موجود در این روش:

تکنیک شیفت عمودی و افقی تصاویر: این کار به معنای هل دادن تمامی پیکسل‌های تصویر در یک جهت مشخص می باشد.

برگرداندن تصاویر از چپ به راست و بالا به پایین: برگرداندن تصویر به معنای معکوس کردن جهت ردیف‌ها یا ستون‌های پیکسل‌های یک تصویر است که باعث می شود تصویر در جهت محورهای عمودی یا افقی تصویر معکوس شود.

تکنیک چرخش تصویر به مقدار تصادفی:این روش جهت تقویت دیتاست از چرخش و دوران تصاویر استفاده می‌کند و برای این کار هر تصویر را به مقداری تصادفی در جهت عقربه‌های ساعت می‌چرخاند.

تکنیک بزرگنمایی تصویر به اندازه تصادفی: با فعال‌سازی این مورد تصاویر مختلفی با بزرگنمایی‌های تصادفی در بازه‌ی مشخص شده از تصویر اصلی ایجاد می‌شود.