



## گزارش پروژه اول درس شبکه های عصبی مصنوعی

محمدرضا یادگاری

### ۱ مقدمه

در این پروژه قصد داریم با استفاده از شبکه های عصبی پیچشی<sup>۱</sup> بر روی دیتاست های اعداد دست نویس فارسی<sup>۲</sup> و دیتاست Fashion Mnist<sup>۳</sup> به نتایج خوبی در دسته بندی داده ها برسیم. ساختار کلی یک شبکه عصبی پیچشی اینگونه است که یک تصویر را در ورودی می گیرد و با انجام عملیات هایی به وسیله عمل گر کانولوشنی (فیلترها) ، ویژگی هایی را از تصویر ورودی بدست می آورد . سپس این ویژگی ها (پس از تبدیل شدن به بردار) به یک شبکه Fully Connected Layer داده می شود که عمل دسته بندی را انجام بدهد. در این پروژه ابتدا با مدل های ساده سعی کردم میزان قدرت شبکه را بسنجم و رفته رفته مدل ها را پیچیده تر کردم و با این کار به جواب های بهتری رسیدم.

### ۲ داده ها

#### ۱.۲ دیتاست اعداد دست نویس فارسی:

مجموعه ما متشکل از ۸۰۰۰۰ تصویر رنگی اعداد دست نویس فارسی از رقم ۰ تا ۹ است. از این تعداد ، ۶۰۰۰۰ تای آن داده های ترین و ۲۰۰۰۰ تای آن داده های تست است.

<sup>۱</sup>Convolutional Neural Networks

<sup>۲</sup><http://farsiocr.ir/>

<sup>۳</sup><https://www.kaggle.com/datasets/zalando-research/fashionmnist>

سایز تصاویر 32 x 32 پیکسل می‌باشد. برای کم کردن حجم محاسبات تصاویر را Grayscale کردم در نتیجه هر تصویر به جای سه بعد دارای یک بعد است. با توجه به قابل قبول بودن تعداد تصاویر و ساده بودن دیتاست ، Data Augmentation و همچنین هیچگونه Normalization صورت نگرفت.

## ۲.۲ دیتاست Fashion Mnist:

مجموعه داده ما متشکل از ۷۰۰۰۰ تصویر Grayscale مربوط به ۱۰ کلاس دسته بندی است. از این ۷۰۰۰۰ تا ۶۰۰۰۰ تای آن داده ترین و ۱۰۰۰۰ تای آن داده تست است. سایز تصاویر ورودی 28 x 28 پیکسل است. با توجه اینکه تعداد تصاویر قابل قبول است و ساده بودن دیتاست ، Data Augmentation و همچنین هیچگونه Normalization صورت نگرفت.

## ۳ مدل ها

### ۱.۳ مدل برای دیتاست اعداد دست نویس فارسی:

مدل به کار گرفته شده ، شبکه عصبی پیچشی ساده ایی است که از دو لایه کانولوشنی و دولایه Fully Connected درست شده است. با مدل ساده ایی شروع به تست کردن شبکه کردم و با عوض کردن هایپرپارامترها<sup>۴</sup> به جواب های خوبی رسیدیم؛ در نتیجه مدل را تغییر ندادم و برای همه آزمایش ها (نتایج آنها در بخش بعدی قابل مشاهده است) از یک مدل استفاده شده است.

### ۲.۳ مدل برای دیتاست Fashion Mnist:

با مدل ساده ایی که برای دیتاست قبلی استفاده کرده بودم شروع کردم و هر بار مدل را پیچیده تر کردم تا به جواب های مطلوبی برسم. آخرین مدل به کار گرفته شده دارای پنج لایه کانولوشنی به همراه لایه های Batch Normalization برای استخراج ویژگی و سه لایه Fully Connected به همراه لایه های Batch Normalization برای دسته بندی است. لایه های فعال ساز هم بعد از هر لایه استفاده شده است و در اخر لایه های کانولوشنی ، یک لایه Dropout به کار گرفته شده است. تمام آزمایش های مختلف با مدل ها و هایپرپارامتر های مختلف در بخش بعدی آورده شده است.

## ۴ نتایج

### ۱.۴ نتایج روی دیتاست اعداد دست نویس فارسی:

نتایج متفاوت در ادامه و در جدول های زیر آمده است. جدول ها به ترتیب آورده شده اند، یعنی هر بار با تغییر کوچکی در شبکه به نتایج جدیدی رسیدم. آزمایش اول:

---

<sup>4</sup>Hyperparameters

Learning Rate	Epoch	Optimizer	Dropout	Batch Size
0.01	7	SGD	0.5	256

Number Of Accurate Predictions = 19098

Accuracy =  $19098/20000 = 0.9549$

آزمایش دوم:

Learning Rate	Epoch	Optimizer	Dropout	Batch Size
0.001	15	Adam	0.5	256

Number Of Accurate Predictions = 19637

Accuracy =  $19637/20000 = 0.98185$

آزمایش سوم:

Learning Rate	Lr-scheduler	Epoch	Optimizer	Dropout	Batch Size
0.01	StepLR	15	Adam	0.5	256

Number Of Accurate Predictions = 19766

Accuracy =  $19766/20000 = 0.9883$

آزمایش های من تمام شد.

برای مسائل دسته بندی مرسوم است که Precision و Recall و F-measure یا به جای این ها Confusion Matrix گزارش شود ، اما متأسفانه من یادم رفته بود که در هر مرحله این موارد را از مدل بگیرم و چون الان نمیتوان دوباره آزمایش ها را تکرار کنم ، به همین "تعداد پیشبینی های درست روی داده های تست" اکتفا کردم.

## ۲.۴ نتایج روی دیتاست Fashion Mnist:

در ادامه مدل های به کار گرفته شده به همراه هایپرپارامترهای مختلف در جدول هایی آورده شده است. مدل ها و جدول ها به ترتیب گزارش شده اند ، یعنی تغییرات به ترتیب اتفاق افتاده اند. نتایج مقایسه بهترین مدل روی این دیتاست با مدل Resnet18 در بخش دیگری ۵ آورده شده است.

در تمام آزمایش های پایین  $\text{Batch Size} = 256$  است.

آزمایش اول:

یک شبکه ساده شامل دو لایه کانولوشنی  $(1, 10)$  ،  $(10, 20)$  و دولایه Fully Connected.

Learning Rate	Epoch	Optimizer	Momentum	Dropout
0.01	7	SGD	0.5	0.5

Number Of Accurate Predictions = 8225

Accuracy =  $8225/10000 = 0.8225$

آزمایش دوم:

همان شبکه آزمایش اول:

Learning Rate	Epoch	Optimizer	weight-decay	Dropout
0.01	15	Adam	0.1	0.5

Number Of Accurate Predictions = 7714

Accuracy =  $7714/10000 = 0.7714$

آزمایش سوم:

شبکه با دولایه کانولوشنی  $(1, 32)$  ،  $(32, 64)$  و سه لایه Fully Connected:

Learning Rate	Epoch	Optimizer	Momentum	Dropout
0.01	15	SGD	0.5	0.5

Number Of Accurate Predictions = 8610

Accuracy =  $8610/10000 = 0.861$

آزمایش چهارم:  
همان شبکه آزمایش سوم ۲.۴:

Learning Rate	Epoch	Optimizer	weight-decay	Dropout
0.01	15	Adam	0	0.5

Number Of Accurate Predictions = 7862  
Accuracy =  $7862/10000 = 0.7862$

آزمایش پنجم:  
همان شبکه آزمایش سوم ۲.۴:

Learning Rate	Lr-scheduler	Epoch	Optimizer	weight-decay	Dropout
0.01	StepLR	15	Adam	0	0.5

Number Of Accurate Predictions = 8473  
Accuracy =  $8473/10000 = 0.8473$

آزمایش ششم:  
همان شبکه آزمایش سوم ۲.۴:

Learning Rate	Lr-scheduler	Epoch	Optimizer	momentum	Dropout
0.01	StepLR	15	SGD	0.5	0.5

Number Of Accurate Predictions = 8483  
 Accuracy =  $8483/10000 = 0.8483$

آزمایش هفتم:  
 مدل با دولایه کانولوشنی (۱،۳۲) ، (۳۲،۶۴) و سه لایه Fully Connected تفاوت این مدل با مدل قبلی ۲.۴ این است که تعداد فیلتر های لایه های کانولوشنی بیشتر شده اند.

Learning Rate	Lr-scheduler	Epoch	Optimizer	momentum	Dropout
0.01	StepLR	15	SGD	0.5	0.5

Number Of Accurate Predictions = 8687  
 Accuracy =  $8687/10000 = 0.8687$

آزمایش هشتم:  
 مدل به کار گرفته شده شامل سه لایه کانولوشنی با فیلتر های (۱،۶۴) ، (۶۴،۱۲۸) ، (۱۲۸،۲۵۶) و سه لایه Fully Connected است:

Learning Rate	Lr-scheduler	Epoch	Optimizer	momentum	Dropout
0.01	StepLR	15	SGD	0.5	0.5

Number Of Accurate Predictions = 9025  
 Accuracy =  $9025/10000 = 0.9025$

آزمایش نهم:  
 مدل به کار گرفته شده شامل پنج لایه کانولوشنی و سه لایه Fully Connected است. هدف از انتخاب این مدل و تغییرات آن نسبت به مدل های قبلی ، افزایش لایه های کانولوشنی برای بیرون کشیدن ویژگی های بیشتر از تصویر ورودی است و بهترین عملکرد را این مدل (که نتایج آن در پایین آمده است) نسبت به بقیه مدل ها داشت. البته بعد از این آزمایش ، آزمایش های دیگری نیز انجام گرفت که نتایجشان بهتر نبود و در اینجا آورده نشده است (در کدی که همراه با گزارش ارسال شده یک شبکه دیگر را هم می بینید که نتایج تا حدودی متفاوت تری دارد):

Learning Rate	Lr-scheduler	Epoch	Optimizer	weight-decay	Dropout
0.01	StepLR	15	Adam	0	0.5

Number Of Accurate Predictions = 9330

Accuracy =  $9025/10000 = 0.933$

## ۵ Transfer Learning

در این قسمت قصد داریم مدل Resnet18 را بر روی دیتاست Fashion Mnist تست بکنیم و نتایج آن را با بهترین مدل به کار گرفته شده در آزمایش های قبلی ۲.۴ مقایسه کنیم.

مدل Resnet18 شامل ۱۸ لایه کانولوشنی است که بر روی بر روی دیتاست ImageNet ترین شده است. ما در این آزمایش از همان وزن هایی که بر روی دیتاست ImageNet ترین شده است و بهینه نیز می باشد استفاده کرده ایم، بنابراین لازم است تغییراتی روی شبکه Resnet18 ایجاد کنیم که بتوان این مدل را روی Fashion Mnist پیاده سازی کرد.

اولین نکته این است که دیتاست ImageNet شامل مجموعه تصاویر رنگی است و تعداد کلاس های دسته بندی آن هزار عدد است. همچنین سایز تصاویر آن 224 X 224 است. دیتاست Fashion Mnist دارای تصاویر Grayscale و شامل ۱۰ عدد کلاس دسته بندی است و سایز تصاویر آن 28 X 28 است.

پس در ابتدا سایز تصاویر را به کمک تابع Transformers از کتابخانه Torchvision به 224 X 224 می رسانیم.

سپس اولین لایه کانولوشنی Resnet18 را که شامل یک لایه کانولوشنی (۳،۶۴) است را به (۱،۶۴) تبدیل می کنیم؛ زیرا تصاویر دیتاست Fashion Mnist ، Grayscale هستند.

در گام بعدی آخرین لایه Fully Connected را که دارای نرون های ورودی و خروجی (۵۱۲،۱۰۰۰) است را به (۵۱۲،۱۰) تبدیل می کنیم؛ زیرا تعداد کلاس های دسته بندی دیتاست Fashion Mnist ، ۱۰ عدد می باشد.

برای اینکه حجم و زمان محاسبات کمتر شود، پارامتر های داخلی شبکه را آپدیت نمی کنیم و فقط اولین لایه کانولوشنی و آخرین لایه Fully Connected را آپدیت می کنیم. با این حال بازهم زمان ترین شدن خیلی زیاد بود. متأسفانه Colab فقط Cpu در اختیار من قرار داد و بیشتر از ۷ ساعت زمان ترین شدن شبکه طول کشید و در آخر هم کامل ترین نشد و Colab کرش کرد. در نتیجه، نتایج تا آخرین Epoch قبل از کرش شدن آورده شده اند.

احتمالا اگر Gpu در اختیار می داشتم ، می توانستم نتایج بهتر و دقیق تری بگیرم و مقایسه بهتری با بهترین مدل خودم داشته باشم.

اسم بهترین مدل ۲.۴ خودم را Best می گذارم تا کار مقایسه آسان تر شود.

همچنین برای اینکه بتوانیم مقایسه خوبی در یک جدول داشته باشیم، عبارت Number Of Accurate Predictions را به اختصار NOAP می نامیم. و عبارت Accuracy را به اختصار Acc می نامیم. نتایج در جدول زیر قابل مشاهده است:

Model	Learning Rate	Lr-scheduler	Epoch	Batch Size	Optimizer	Dropout	NOAP	Acc
Resnet18	0.01	StepLR	5	64	SGD	0.5	8342	0.8342
Best	0.01	StepLR	15	256	Adam	0.5	9330	0.933

Table 1: Comparison Of Two Models

با توجه به داده های جدول می بینیم که بهترین مدل ما نتایج بهتری دارد.

## ۶ جمع بندی

در تمرین اول (دسته بندی روی اعداد دست نویس فارسی) متوجه شدیم که Adam Optimizer بهتر از SGD Optimizer عمل می کند. اما در تمرین دوم (دسته بندی روی دیتاست Fashion Mnist) تا حدودی مدل با SGD Optimizer جواب های بهتری داشت. در کل ران کردن مدل ها خیلی زمان می برد و در خیلی از مواقع از تست کردن مدل جدید به دلیل اینکه ممکن است زمان طولانی برای ران کردن نیاز داشته باشد، صرف نظر کردم. دیتاست Fashion Mnist دیتاست پیچیده تری بود و از روی دقت مدل ها هم می شد به این قضیه پی برد. ایده Transfer Learning به نظر خوب و هوشمندانه می رسد و احتمالا بشود کارهای زیادی با آن انجام داد.