



# گزارش بخش عملی تمرین هفتم درس یادگیری ماشین

تشخیص بیماری ذاتالریه (Pneumonia) با استفاده از عکس X-ray قفسه سینه و یادگیری عمیق

کیان باختری

bakhtari.kian@gmail.com

۱۴۰۰ مرداد ۳

## چکیده:

در این گزارش، مراحل طی شده برای حل مسئله‌ی تشخیص بیماری ذاتالریه با استفاده از شبکه‌ی عصبی و یادگیری عمیق به اختصار شرح داده می‌شود. **مجموعه دادگان** مورد استفاده شامل حدود ۶۰۰۰ عکس X-ray قفسه سینه از افراد سالم و مبتلا به بیماری ذاتالریه بوده و خود به سه بخش آموزش، ارزشیابی، و آزمون تقسیم شده است. در کنار این گزارش یک نوت بوک ژوپیتر به زبان پایتون وجود دارد؛ لطفاً برای مشاهده کدها به نوت بوک مراجعه کنید.

## ۱ مقدمه

سه کانال رنگی دارند به خاکستری (gray scale) تبدیل شوند تا از خطاهایی مرتبط با به روشنایی (illumination) جلوگیری شود. همچنین برای یکسان‌سازی ابعاد تصاویر، تمامی عکس‌ها به ابعاد پیشنهاد شده در صورت تمرین (یعنی ۲۱۶ در ۲۱۶) درآمدند. مجموعه اصلی آموزش، ارزشیابی، و آزمون با این دو تبدیل (به اضافه یک تبدیل به تنسور) آماده شدند.

### ۱-۳ داده افزوده

داده افزوده یا augmented data به داده‌هایی گفته می‌شود که از روی مجموعه داده موجود به شکل مصنوعی تولید شده و قرارگیری‌شان کنار داده‌های آموزش به افزایش قدرت تعمیم (generalization) مدل کمک می‌کند. در این تمرین، از اضافه کردن داده افزوده، هم به منظور افزایش قدرت تعمیم و هم به منظور افزایش توازن در دو کلاس استفاده شده است. بدین صورت که داده‌های آموزش علاوه بر شیوه‌ی توضیح داده شده در پاراگراف قبل، یک بار دیگر نیز لود شده و این بار پس از اعمال چندین تبدیل تصادفی (random transform) به صورت مجزا نگهداری می‌شوند. در این مجموعه‌ی جدید به شکل تصادفی فقط نصف داده‌های مربوط به کلاس بیمار آورده شده است تا کمی به توازن کلاس‌ها کمک شود. تبدیل‌های استفاده شده در بخش داده‌های افزوده هر کدام به احتمال ۷۰ درصد اعمال شده و بعضی‌هاشان حتماً اعمال می‌شوند. این تبدیل‌ها به ترتیب چرخش

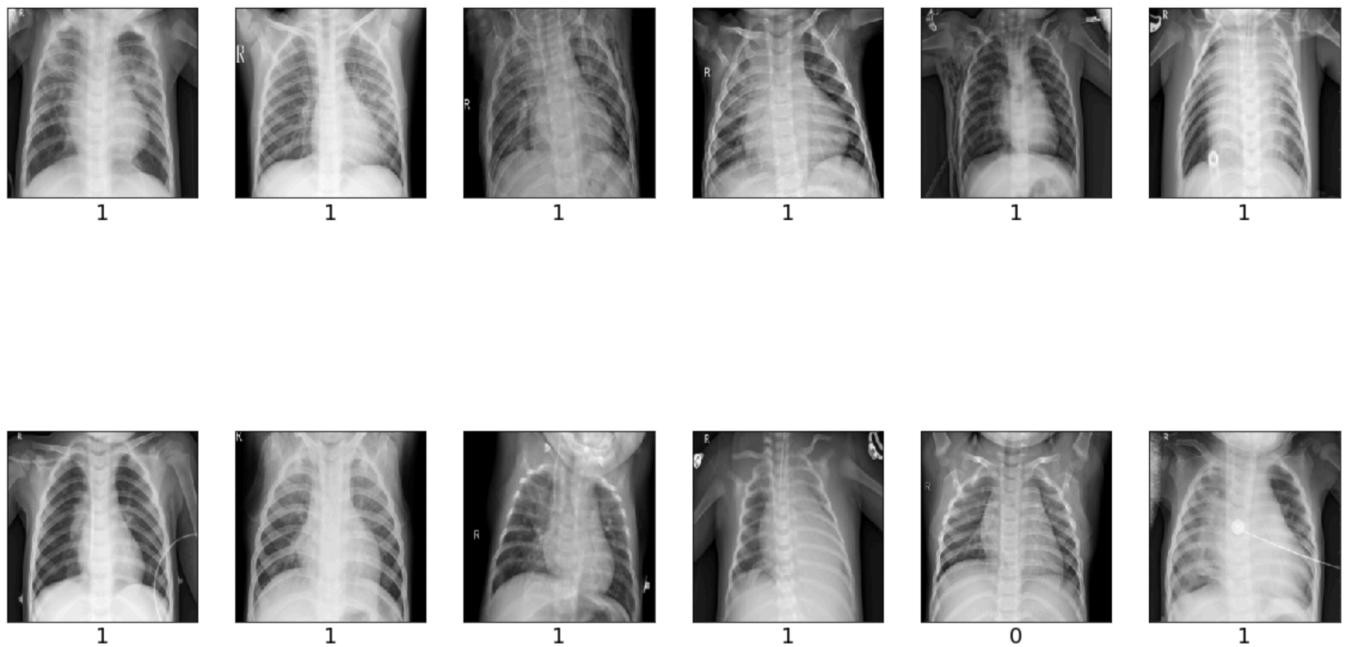
هدف این تمرین پیاده‌سازی یک شبکه‌ی عصبی با معماری residual network یا به اختصار ResNet بوده است که وظیفه‌ی طبقه‌بندی عکس‌های x-ray افراد مشکوک به بیماری ذاتالریه به دو کلاس سالم یا بیمار را انجام دهد. در کنار این گزارش یک نوت بوک ژوپیتر به زبان پایتون قرار دارد که حاوی کدهای مربوط به وظیفه‌ی خواسته شده می‌باشد. در این پیاده‌سازی از کتابخانه PyTorch استفاده شده است.

## ۲ تحلیل اکتشافی داده

مجموعه دادگانی که در اختیار داریم شامل ۱۵۸۳ عکس از افراد سالم و ۴۲۷۳ عکس از افراد مبتلا به بیماری می‌باشد. توازن میان حجم کلاس‌ها وجود ندارد که در ادامه به این مشکل رسیدگی خواهیم کرد. عکس‌ها همگی دارای سه کانال هستند و ابعادشان با هم متفاوت است. ۱۲ نمونه از عکس‌های حاضر در مجموعه داده را در شکل ۱ مشاهده می‌کنید.

## ۳ آماده‌سازی و تبدیل‌ها

تصاویر با استفاده از ImageFolder تبدیل شده و در سه مجموعه‌ی مجزا (آموزش، ارزشیابی، آزمون) قرار می‌گیرد. برای تصاویر پزشکی مانند x-ray که بنا نیست رنگی باشند، توصیه می‌شود که اگر عکس‌ها



شکل ۱: چند نمونه از عکس‌های حاضر در مجموعه دادگان. برچسب صفر به معنای سالم و برچسب یک به معنای بیمار می‌باشد.

در قسمت نهایی دو لایه تمام-متصل (fully connected) دارد، تابع فعال‌ساز نهایی softMax است و  $\text{dropout}=0.4$  تنظیم شده است. پس از ۱۰ پیمایش (epoch) آموزش، دقت این مدل روی داده‌ی آموزش به ۹۱ و روی داده‌ی آزمون به ۸۴ رسید. نتایج این مدل در پیمایش یازدهم را در شکل ۲ مشاهده می‌کنید. همچنین در شکل ۳ تعدادی از داده‌های افزوده‌ی استفاده شده در این مدل به نمایش درآمده‌اند.

تصادفی (randomRotation) بین ۲ تا ۳۰ درجه، تارسازی تصادفی گاوی (randomGaussianBlur) و جابجایی تصادفی حول محور عمودی هستند. در صورت تمرين آورده شده که هر احتمال اعمال این تبدیل‌ها بر هر داده برابر ۵۰ درصد باشد. با توجه به این که آموزش مدل هم بر بروی داده‌های آموزش اصلی و هم داده‌های آموزش افزوده انجام می‌شود، در عمل این نسبت رعایت شده است.

## ۲-۴ مدل شماره دو

در این مدل تعداد کanal‌ها در لایه‌های میانی ابتدا به ۲۲ و بعد به ۴۲ می‌رسد که نسبت به مدل قبلی افزایش داشته است. تفاوت دیگری با مدل اول داده نشده است. دقت مدل بر روی داده‌ی یادگیری افزایش داشت و به ۹۵ رسید اما بر داده‌ی آزمون و ارزشیابی پیشرفته مشاهده نشد. در شکل ۴ نتایج این مدل پس از پیمایش سوم و در شکل ۵ نمودار نتایج مدل قابل مشاهده است.

## ۳-۴ مدل شماره سه

در مدل سوم تعداد لایه‌های تمام متصل در آخرین مرحله از دو به یک کاهش پیدا کرد که مطابق با پیشنهاد صورت تمرين Max-Pooling است. همچنین در این مدل پس از هر پیمایش ضریب یادگیری با فاکتور ۹۰ درصد کاهش پیدا کرده است. نسبت به مدل دوم تغییری دیگری اعمال نشد و نتایج مطلوب نبود. در شکل ۶ و ۷ به ترتیب نتایج این مدل پس از آخرین پیمایش و نمودار نتایج کلی آورده شده است. در شکل ۸ نیز نمودار امتیازات چهارگانه‌ی این مدل بر داده‌ی آزمون قابل مشاهده است.

## ۴ شبکه عصبی

معاری شبکه‌ی پیاده‌سازی شده مطابق با خواسته‌ی صورت تمرين است. ابتدا برای کم کردن ابعاد ورودی لایه‌های residual از Max-AvgPooling و Pooling استفاده شد که با توجه ابعاد نسبتاً بزرگ انتخاب شده برای کرنل‌ها نتیجه‌ی مطلوبی در بر نداشت. سپس مطابق پیشنهاد صورت تمرين برای down sampling از لایه‌های convolutional استفاده شد که مجدداً به علت اندازه‌ی نامناسب کرنل، حاصل مورد قبولی به دست نیامد. پس از کمی جست‌وجو، با افزایش تعداد کanal‌های میانی در شبکه و کاهش ابعاد کرنل‌های لایه‌های مذکور، نتایج قابل قبول تری حاصل شد.

## ۴-۱ مدل شماره یک

مدل‌ها به صورت کاملاً پارامتریک نوشته شده‌اند که به راحتی قابل تغییر باشند. اصلی‌ترین پارامترهای متغیر در مدل‌ها، تعداد کanal‌ها در لایه‌های میانی شبکه است. مدل‌هایی که تعداد کanal‌های میانی شان از ۶ یا ۱۲ تجاوز نمی‌کرد آزموده شده و نتایج مطلوبی به دست ندادند. سپس مدلی موردنی آزمون قرار گرفت (مدل شماره یک) که تعداد کanal‌هایی از ۱ در ورودی به ۱۶ و در لایه‌ی میانی به ۳۲ می‌رسد.

شکل ۱۰ به نمایش درآمده است.

#### ۶-۴ مدل شماره شش

تعداد نورون‌ها در لایه‌ی تمام متصل دوم به ۵۱۲ افزایش داده شد. همچنین تعداد کانال‌ها در لایه‌های میانی در مدل قبل به ۲۲ و بعد به ۴۲ می‌رسید که در این مدل به ۲۴ و ۴۸ افزایش داده شد. همچنین برای آزمودن کارایی داده‌های افزوده، در این مدل از داده‌های افزوده استفاده نشد. نتایج برای داده‌ی یادگیری بدون تغییر چشمگیری تکرار شد و همچنین بر داده‌ی ارزشیابی و آزمون نیز بهبودی مشاهده نشد. نمودار نتایج کلی این مدل را می‌توانید در شکل ۱۱ و نمودار امتیازات چهارگانه را در شکل ۱۲ مشاهده کنید.

#### ۷-۴ مدل شماره هفت

در این مدل شدت تبدیل‌های داده‌های افزوده را بیشتر و با حذف مقدار بیشتری از داده‌های کلاس بیمار از این مجموعه، توازن داده‌های افزوده را بیشتر کردیم. نتیجه‌ی کلی حاصل شده در شکل ۱۳ قابل مشاهده بوده و تناوب دقت روی داده‌ی ارزشیابی جلب نظر می‌کند. قله‌ها بالا فاصله پس از آموزش با داده‌های افزوده و دره‌ها بالا فاصله پس از آموزش با داده‌های معمولی هستند. دقت بر داده‌ی آزمون و ارزشیابی در این مدل به ۸۲ رسید که در شکل ۱۳ هم مشخص است.

#### ۸-۴ مدل شماره هشت

تناوبی که در نمودار مدل قبل مشاهده شد، بررسی شده و نتیجه آن شد که گویا مدل پس از دیدن داده‌های افزوده بهتر می‌شود و با شروع پیمایش‌های بعدی و دیدار مجدد با داده‌های آموزش، دقت خود بر داده‌ی ارزشیابی را از دست می‌دهد. سپس، چندین حالت مختلف آزمایش شد و نتیجه‌ای که به دست آمد این بود که هرچه مقدار و تنوع داده‌های افزوده افزایش یابد، عملکرد مدل نیز بهتر می‌شود. به همین منظور حجم و تنوع داده‌های افزوده افزایش داده شد. مدل هشت در تمامی پارامترها مشابه مدل هفت است؛ تنها داده‌های افزوده را بیشتر کردیم. دقت این مدل بر داده‌ی آزمون به ۸۴ رسید. نمودار کلی نتایج در شکل ۱۴ و confusion matrix این مدل در شکل ۱۵ آمده است. از حالا به بعد، با توجه به این که پارامترهای مدل تغییر داده نشده‌اند، نام مدل همان شماره هشت باقی می‌ماند. مولفه‌های مدل در شکل ۱۸ قابل مشاهده است. با آزمایش مذکور در پاراگراف بالا، مشاهده شد که مجدداً مدل رفتار تناوبی توضیح داده شده را تکرار می‌کند. یعنی عملکرد با داده‌های افزوده بهتر می‌شود. مرحله به مرحله، به حجم و تنوع تبدیل‌ها در داده‌های افزوده اضافه کردیم و سرانجام با استفاده از شش data loader که هر کدام تبدیل‌های متفاوتی را بر تصاویر اعمال می‌کردند، دقت (accuracy) بر داده‌ی ارزشیابی ۹۳ درصد و بر داده‌ی آزمون ۸۸ درصد به دست آمد. نمودار نتایج کلی مدل (حالت نهایی) را در شکل ۱۶ مشاهده می‌کنید. همچنین confusion matrix نهایی مدل در شکل ۱۷ به نمایش درآمده است.

Starting epoch 11

Batch 0/41, Loss=1.6992

Accuracy on training: 87.94%

Accuracy on test: 81.25%

Batch 10/41, Loss=0.4170

Accuracy on training: 85.35%

Accuracy on test: 80.93%

Batch 20/41, Loss=0.3411

Accuracy on training: 91.26%

Accuracy on test: 83.01%

Batch 30/41, Loss=0.3491

Accuracy on training: 90.18%

Accuracy on test: 77.40%

Batch 40/41, Loss=0.3026

Accuracy on training: 88.69%

Accuracy on test: 75.00%

Batch 0/41, Loss=0.2527

Accuracy on training: 88.23%

Accuracy on test: 74.20%

Batch 10/41, Loss=0.4641

Accuracy on training: 90.07%

Accuracy on test: 83.33%

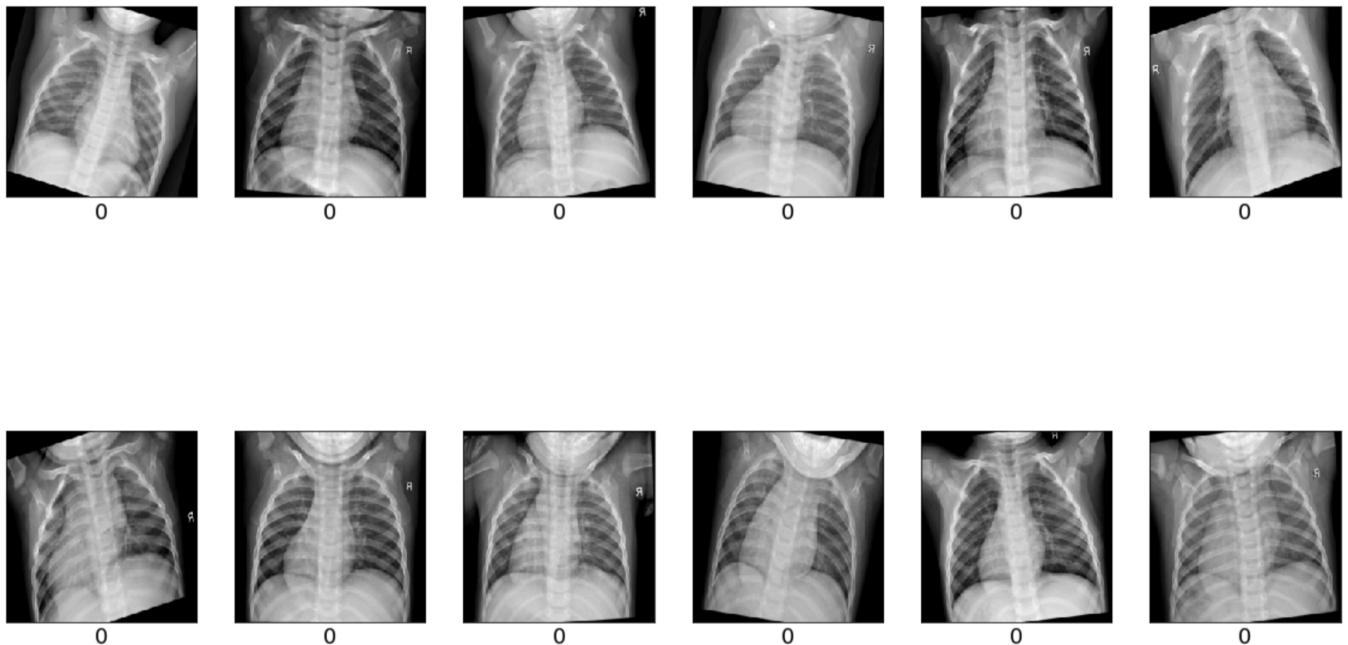
شکل ۲: نتایج به دست آمده از مدل یک در آخرین پیمایش

#### ۴-۴ مدل شماره چهار

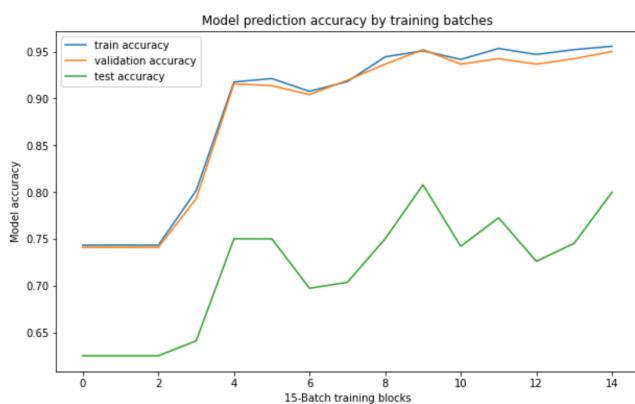
با توجه به این که در مدل سوم دقت بر داده‌ی یادگیری برخلاف مدل دوم به ۹۰ درصد و بیشتر نرسید، مجدداً در مدل چهارم تعداد لایه‌های تمام متصل به دو عدد افزایش یافت تا پیچیدگی بیشتر شود. تفاوت این مدل با مدل دوم در این است که برای آخرین لایه (خروجی) ازتابع فعال‌ساز sigmoid استفاده شده است و همچنین در بخش داده‌های افزوده از شدت تبدیل‌ها کاسته شده است. برای مثال بازی درجه‌ی چرخش تصادفی از ۳۰ درجه به ۱۰ درجه کاهش یافته و شدت تارسازی گاوسی نیز کم شده است. دقت بر داده‌ی آموزش مجدداً بالا رفت اما دقت بر داده‌های ارزشیابی و آزمون پیشرفته نداشتند. نمودار نتایج کلی مدل چهارم را در شکل ۹ مشاهده می‌کنید.

#### ۵-۴ مدل شماره پنج

همان‌طور که از نمودار مدل چهارم در شکل ۹ قابل برداشت است، به نظر می‌رسد که این مدل بیش‌برازش شده است. تعداد نورون‌ها در لایه‌ی تمام متصل دوم، ۱۰۲۴ تنظیم شده که در مدل پنجم به ۱۰ کاهش داده شد تا از پیچیدگی مدل کاسته شده و از بیش‌برازش جلوگیری شود که همین‌طور هم شد. اما باز هم نتایج بر داده‌های ارزشیابی و آزمون پیشرفته نکردند. در این مدل با استفاده از امکان ReduceLROnPlateau ضربی یادگیری در هر موقعی که پیشرفته در نتایج مشاهده نمی‌شده کاهش یافته است. نمودار نتایج این مدل در



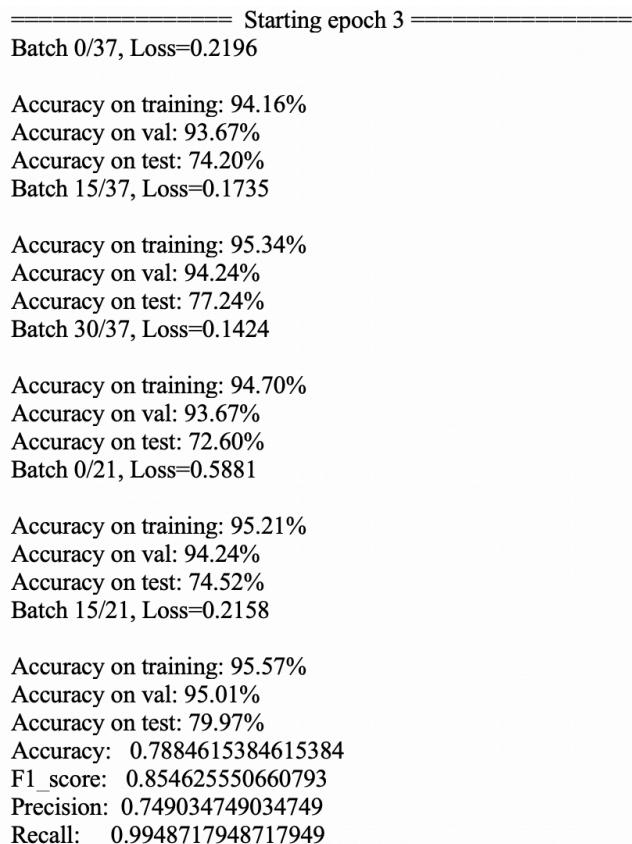
شکل ۳: نمونه هایی از داده های افزوده (استفاده شده در مدل یک)



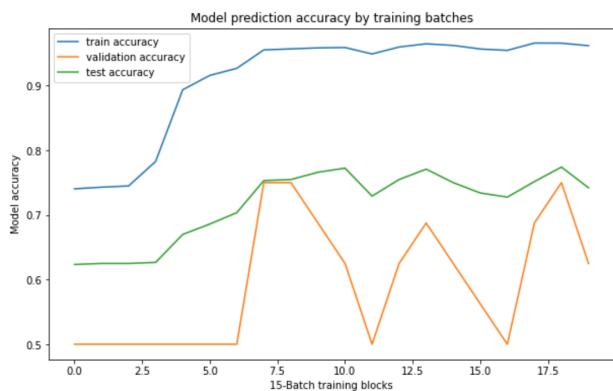
شکل ۵: نمودار نتایج مدل شماره دو - محور X بلوک های ۱۵ تابی از دسته (batch) ها را مشخص کرده است.

## Drop out ۵

آخرین حالت مدل بدون حضور لایه dropout آموزش داده شد و نتایج بهتری نسبت به حالت بدون حضور این لایه ها به دست آمد. البته که تفاوت آن ها چندان معنی دار نبوده و طبق مشاهدات در این تمرین، حضور یا عدم حضور شان تفاوت چندانی نمی کند. البته حضور این لایه ها با ضریب اعمال کمتر یا بیشتر از ۰.۴ که پیشنهاد شده بود آزموده نشد.



شکل ۴: نتایج مدل دوم پس از پیمایش سوم - نمودار نتایج کلی این مدل در شکل ۵ آورده شده است



شکل ۹: نمودار نتایج کلی مدل چهارم

Starting epoch 7  
Batch 0/37, Loss=0.6074

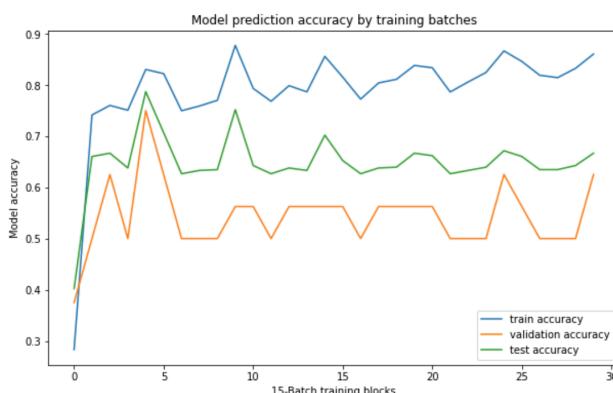
Accuracy on training: 85.11%  
Accuracy on val: 88.10%  
Accuracy on test: 66.51%  
Batch 15/37, Loss=0.3142

Accuracy on training: 82.36%  
Accuracy on val: 85.41%  
Accuracy on test: 64.26%  
Batch 30/37, Loss=0.4695

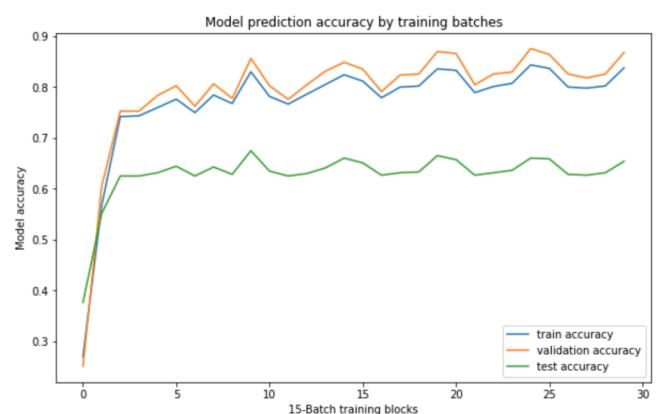
Accuracy on training: 80.34%  
Accuracy on val: 82.73%  
Accuracy on test: 63.14%  
Batch 0/26, Loss=0.9057

Accuracy on training: 80.45%  
Accuracy on val: 82.73%  
Accuracy on test: 63.30%

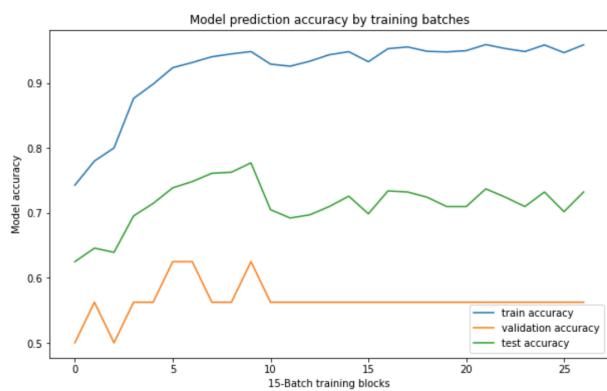
شکل ۶: نتایج مدل سوم پس از آخرین پیمایش - نمودار کلی نتایج در شکل بعدی قابل مشاهده است



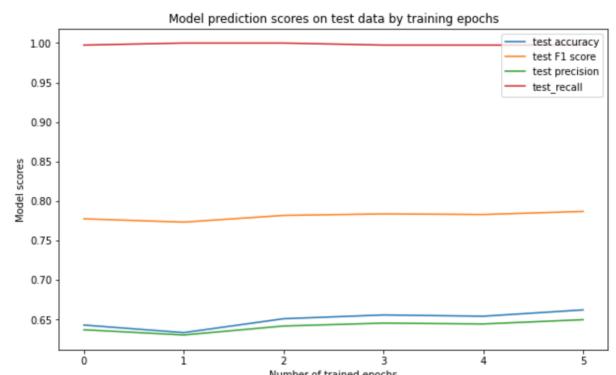
شکل ۱۰: نمودار نتایج کلی مدل پنجم



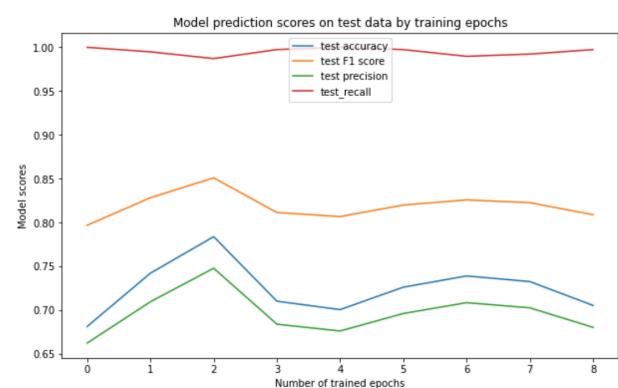
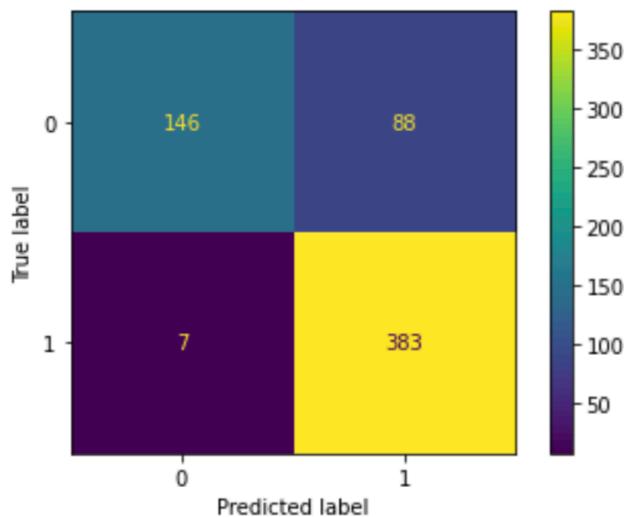
شکل ۷: نمودار نتایج کلی مدل سوم



شکل ۱۱: نمودار نتایج کلی مدل ششم

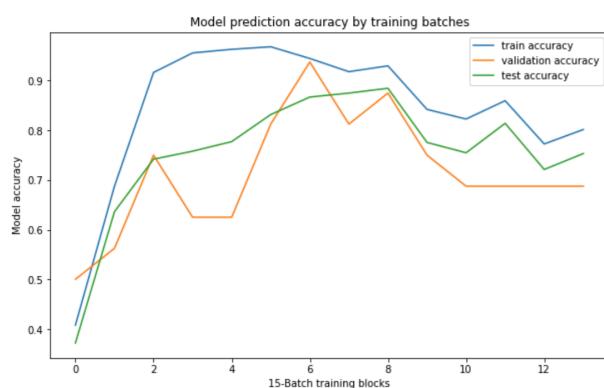


شکل ۸: نمودار امتیازات چهارگانه بر داده‌ی آزمون در مدل سوم

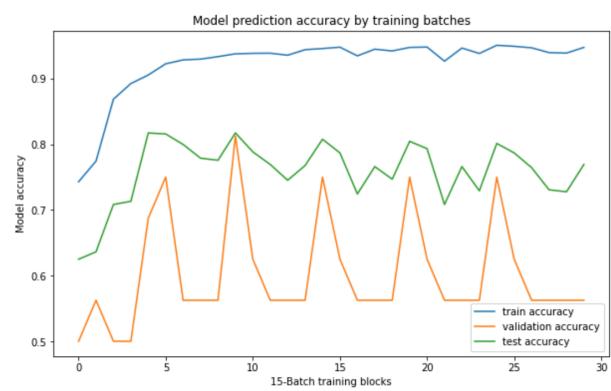


شکل ۱۲: نمودار امتیازات چهارگانه مدل ششم

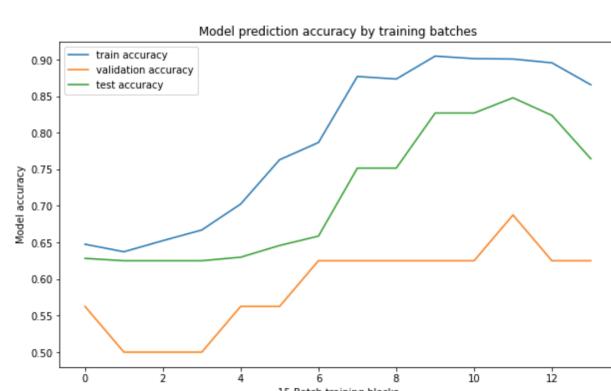
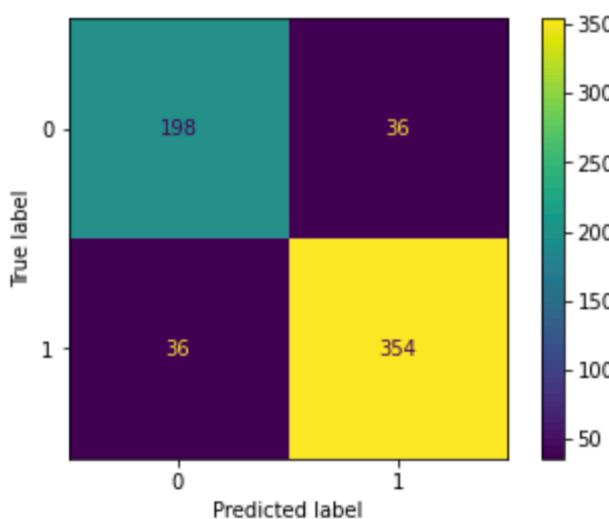
شکل ۱۵: مدل شماره هشت - (حالت نهایی نیست)



شکل ۱۶: نمودار نتایج نهایی مدل هشتم



شکل ۱۳: نمودار نتایج کلی مدل هفتم - به نتاوب دقت بر داده‌ی ارزشیابی توجه کنید



شکل ۱۴: نمودار نتایج نهایی مدل هشتم - پیش از حالت نهایی داده‌های افزوده

شکل ۱۷: مدل شماره هشت - (حالت نهایی)

```

ResNet(
    (resb1): ResBlock1(
        (zpad): ZeroPad2d(padding=(5, 5, 5, 5), value=0.0)
        (c1): Conv2d(1, 16, kernel_size=(7, 7), stride=(3, 3))
        (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (c2): Conv2d(16, 24, kernel_size=(7, 7), stride=(1, 1))
        (bnormc2): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (downConv): Conv2d(1, 24, kernel_size=(5, 5), stride=(3, 3))
        (dro): Dropout(p=0.4, inplace=False)
        (resize): Resize(size=(68, 68), interpolation=PIL.Image.BILINEAR)
        (bnorm2): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (resb2): ResBlock2(
        (zpad): ZeroPad2d(padding=(0, 0, 0, 0), value=0.0)
        (c1): Conv2d(24, 40, kernel_size=(5, 5), stride=(1, 1))
        (bnorm): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (c2): Conv2d(40, 40, kernel_size=(5, 5), stride=(1, 1))
        (bnormc2): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (downConv): Conv2d(24, 40, kernel_size=(3, 3), stride=(1, 1))
        (dro): Dropout(p=0.4, inplace=False)
        (resize): Resize(size=(60, 60), interpolation=PIL.Image.BILINEAR)
        (bnorm2): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (resb3): ResBlock2(
        (zpad): ZeroPad2d(padding=(0, 0, 0, 0), value=0.0)
        (c1): Conv2d(40, 40, kernel_size=(3, 3), stride=(1, 1))
        (bnorm): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (c2): Conv2d(40, 40, kernel_size=(3, 3), stride=(1, 1))
        (bnormc2): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (downConv): Conv2d(40, 40, kernel_size=(5, 5), stride=(1, 1))
        (dro): Dropout(p=0.4, inplace=False)
        (resize): Resize(size=(56, 56), interpolation=PIL.Image.BILINEAR)
        (bnorm2): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (f5): Linear(in_features=125440, out_features=600, bias=True)
    (f6): Linear(in_features=600, out_features=2, bias=True)
    (dro): Dropout(p=0.4, inplace=False)
)

```

شکل ۱۸: مولفه‌های ساختاری مدل نهایی