



به نام خدا  
دانشکده‌ی مهندسی برق و کامپیوتر  
تمرین سری پنجم یادگیری ماشین



سلام بر دانشجویان عزیز، چند نکته مهم:

1. کدهای ارسال شده بدون گزارش صوتی فاقد نمره می‌باشند.
2. برای سوالات شبیه سازی، فقط از دیتاست داده شده استفاده کنید.
3. فایل نهایی خود را در یک فایل زیپ شامل، فایل‌های صوتی و فایل کدها آپلود کنید. نام فایل زیپ  
ارسالی الگوی ML\_HW#\_StudentNumber داشته باشد.
4. از بین سوالات شبیه سازی حتما به هر دو مورد پاسخ داده شود.
5. نمره تمرین ۱۰۰ نمره می‌باشد و حداکثر تا نمره ۱۱۰ ( ۱۰ نمره امتیازی ) می‌توانید کسب کنید.
6. هرگونه شباهت در مطالب ارائه شده و کد مربوط به شبیه سازی، به منزله تقلب می‌باشد و کل  
تمرین برای طرفین صفر خواهد شد.
7. در صورت داشتن سوال، از طریق ایمیل ta.mobin.roohi@gmail.com سوال خود را مطرح کنید.

## سوال ۱: ( ۱۵ نمره)

(الف) مزایا و معایب درخت‌های تصمیم و شبکه‌های عصبی را مقایسه کنید. همچنین توضیح دهید که هر یک از این مدل‌ها برای چه نوع مجموعه داده‌هایی مناسب‌تر هستند.

(ب) چرا استفاده از تابع فعال سازی

$$f(x) = x$$

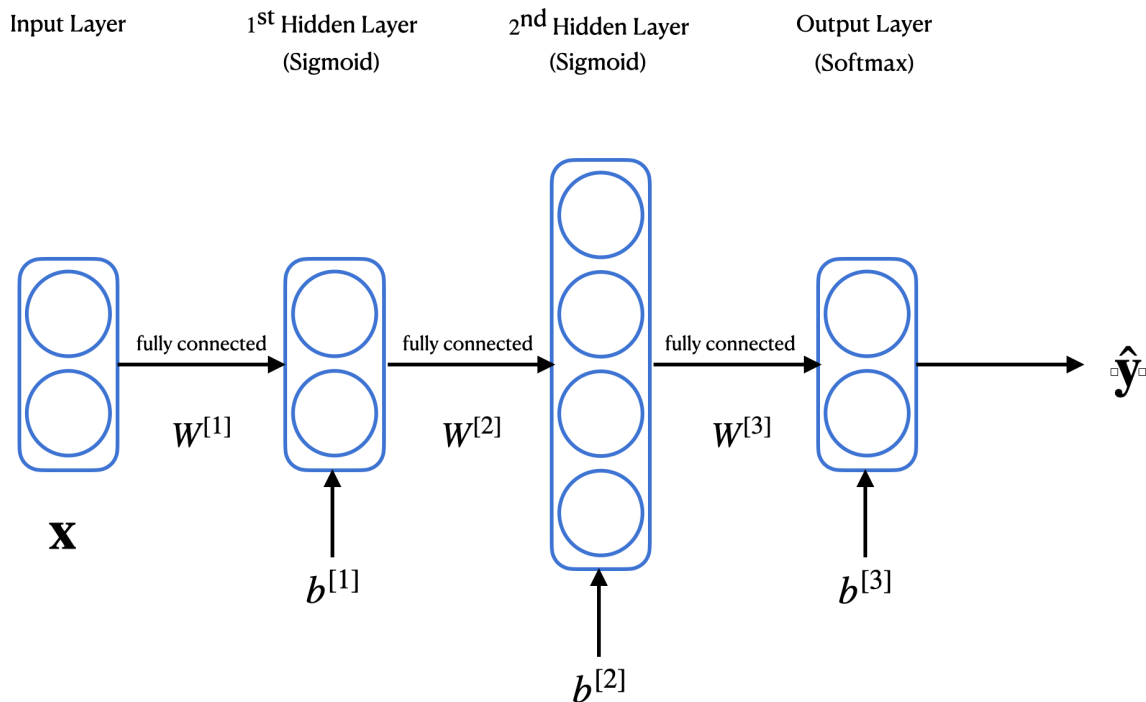
در لایه‌های مخفی یک MLP مناسب نیست؟

(ج) یکی از دلایل استفاده از توابع فعال‌سازی غیرخطی مشتق‌پذیر در شبکه‌های عصبی، به جای تابع آستانه (Threshold) استفاده‌شده در مدل پرسپترون Rosenblatt، امکان مشتق‌پذیر کردن کل شبکه است. مشتق‌پذیری نسبت به پارامترها این امکان را فراهم می‌کند که از روش‌های بهینه‌سازی مبتنی بر گرادیان، مانند روش کاهش گرادیان تصادفی (Stochastic Gradient Descent)، برای یادگیری پارامترهای شبکه استفاده شود. در این میان، تابع فعال‌سازی Rectified Linear Unit (ReLU) یکی از پرطرفدارترین توابع فعال‌سازی در شبکه‌های عصبی عمیق به شمار می‌آید. با این حال، این تابع به طور کامل مشتق‌پذیر نیست. توضیح دهید که چرا این تابع مشتق‌پذیر نیست و چگونه است که می‌توان همچنان از آن و روش‌های بهینه‌سازی مبتنی بر گرادیان برای آموزش شبکه‌های عصبی استفاده کرد؟ یک توضیح خلاصه کافی است.

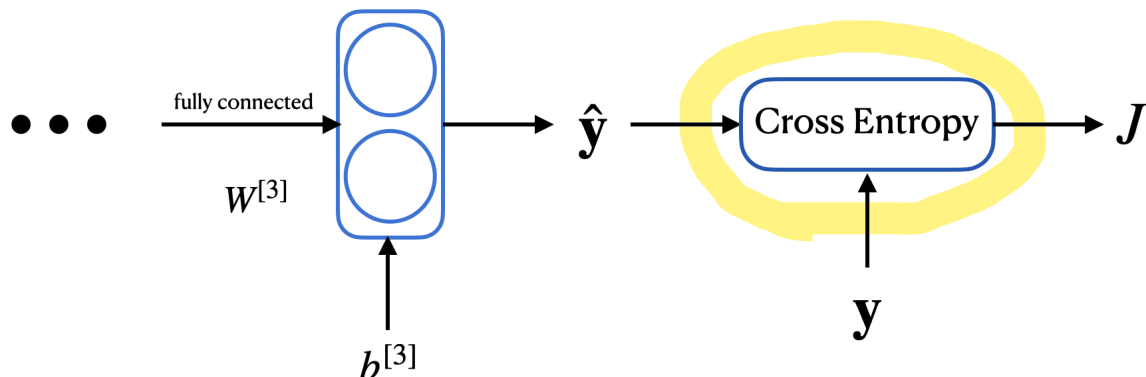
(د) در فرآیند آموزش شبکه‌های عصبی عمیق روی داده‌های با ویژگی‌های متعدد و پیچیده، تابع Loss ممکن است بسیار پیچیده باشد. توضیح دهید که در فرآیند یادگیری شبکه با استفاده از الگوریتم گرادیان کاهشی (Gradient Descent)، چه نقاطی در Loss Function Landscape می‌توانند باعث توقف زود هنگام یا کاهش سرعت یادگیری شوند؟ کدام یک از این نقاط چالش بزرگتری برای یادگیری ایجاد می‌کنند؟ چرا؟ دو مورد راهکار برای جلوگیری از گیر افتادن در این نقاط در حین یادگیری پیشنهاد دهید.

## سوال ۲: ( ۱۵ نمره )

MLP سه لایه‌ای زیر را در نظر بگیرید.



این MLP دارای یک لایه ورودی است که برداری دوبعدی را به عنوان ورودی می‌پذیرد. سپس این بردار از دو لایه پنهان با فعالسازی Sigmoid که به ترتیب دارای دو و چهار نرون هستند گذر می‌کند. در نهایت، با گذر از لایه نهایی با فعالسازی Softmax به خروجی شبکه، یعنی یک بردار دوبعدی دیگر می‌رسد. تابع خطای شبکه را نیز خطای Cross Entropy در نظر بگیرید و آن را با  $J$  نمایش دهید.



مراحل forward propagation این MLP با استفاده از روابط زیر به دست می‌آید:

## Vectorized Forward Prop

### Input:

The input  $\mathbf{x}$ .

MLP with  $N$  layers.

### Initialize:

$$\mathbf{a}^{[0]} \leftarrow \mathbf{x}$$

### Loop:

— For layer  $k = 1, \dots, N$ :

$$\begin{aligned}\mathbf{z}^{[k]} &\leftarrow \mathbf{W}^{[k]} \mathbf{a}^{[k-1]} + \mathbf{b}^{[k]} \\ \mathbf{a}^{[k]} &\leftarrow \mathbf{f}_k(\mathbf{z}^{[k]})\end{aligned}$$

where  $\mathbf{f}_k(\cdot)$  is the activation function for the  $k$ -th layer.

### Output:

$$\hat{\mathbf{y}} = \mathbf{a}^{[N]}$$

که در اینجا درایه  $W_{ij}^{[k]}$  ماتریس وزن، همان وزن ارتباط میان نورون  $j$  ام لایه  $k - 1$  و نورون  $i$  ام لایه  $k$  است.  $b_i^{[k]}$  نیز bias مربوط به نورون  $i$  ام لایه  $k$  است.

(الف) سائیز و شکل ماتریس‌های  $W^{[1]}$ ،  $W^{[2]}$  و  $W^{[3]}$  به چه شکل است؟

(ب) با توجه به الگوریتم backpropagation، گرادیان تابع خطای  $J$  را نسبت به پارامترهای  $W_{11}^{[1]}$  و  $b_1^{[1]}$  به دست آورید. بهتر است که از صورت برداری روابط استفاده کنید.

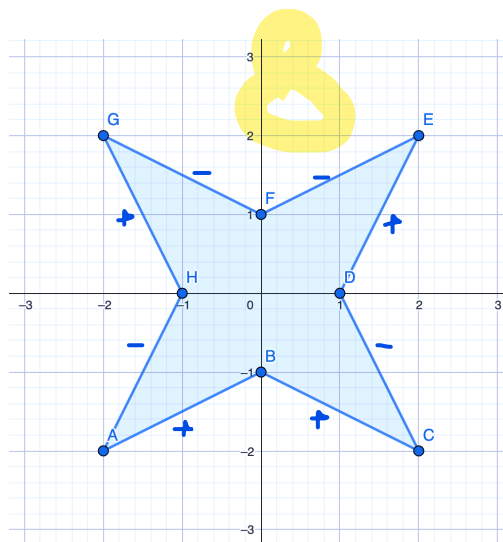
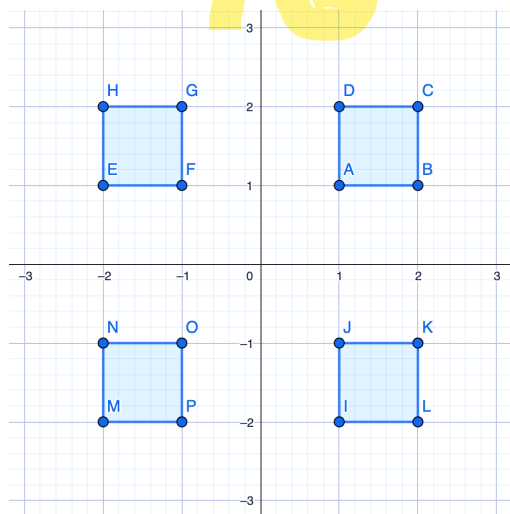
### سوال ۳: (۱۰ نمره)

(الف) یک شبکه پرسپترون چندلایه (MLP) طراحی کنید که از تابع فعال‌ساز Heaviside Step Function استفاده کند.

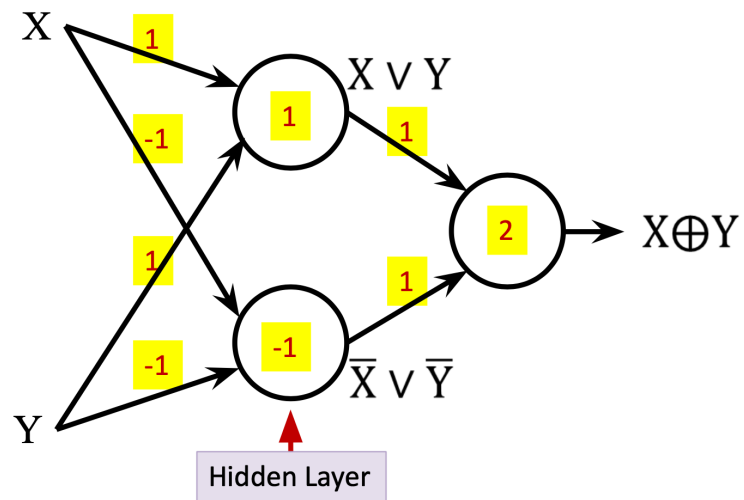
$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

نمودار آن را رسم کنید.

(معماری طراحی شده باید به گونه‌ای باشد که بتواند به ازای پارامترهای صحیح هر دو تصویر زیر را به درستی طبقه‌بندی کند. یعنی مدل باید بتواند برای ورودی‌های داخل ناحیه آبی مقدار ۱ و خارج آن مقدار ۰ را خروجی بدهد.)

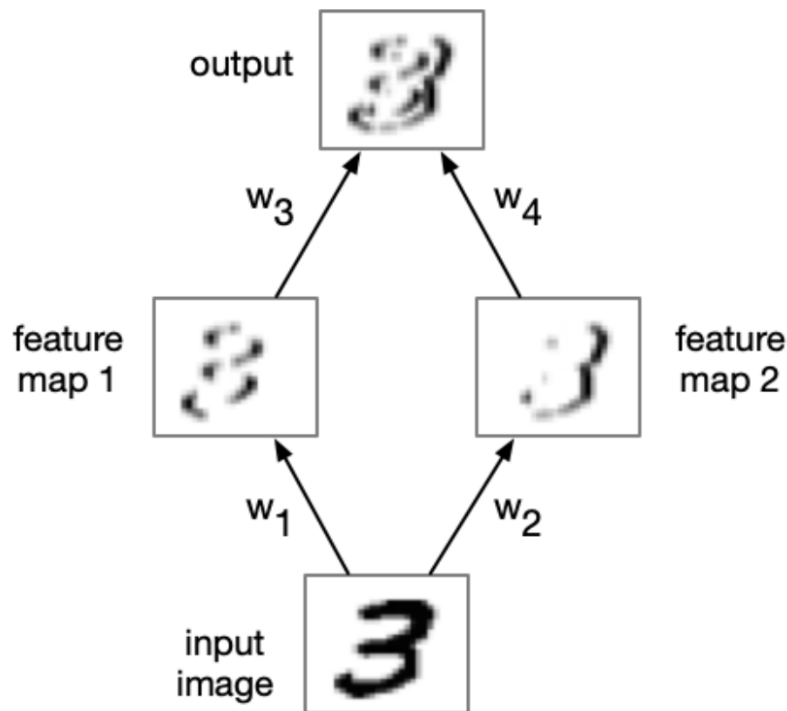


(ب) پس از طراحی معماری شبکه‌ای که بتواند هر دو ناحیه فوق را به درستی طبقه‌بندی کند، وزن‌ها و بایاس‌های این شبکه را محاسبه کرده و روی نمودار شبکه نمایش دهید. برای این کار می‌توانید مقدار آستانه یا Threshold مربوط به هر نورون را که همان قرینه مقدار بایاس نورون است را داخل نورون نمایش دهید. برای نمونه، نمودار زیر شبکه عصبی مربوط به XOR همراه با مقدار وزن‌ها و آستانه‌ها می‌باشد.



#### سوال ۴: ( ۱۰ نمره )

معماری شبکه کانولوشنال زیر را برای تشخیص **مرزهای عمودی** در یک تصویر در نظر بگیرید.



تابع فعال سازی **ReLU** در اولین لایه پیش‌بینی اعمال می‌شود. لایه خروجی از تابع فعال سازی **خطی** استفاده می‌کند.

در تصاویر از سفید برای نشان دادن ۰ و مقادیر تیره‌تر برای نشان دادن مقادیر بزرگ‌تر (مثبت‌تر) استفاده می‌کنیم.

الف) دو فیلتر کانولوشن برای لایه اول، یعنی همان  $w_1$  و  $w_2$ ، به اندازه  $3 \times 3$  طراحی کنید. یکی از آنها باید مرزهای تاریک/روشن را تشخیص دهد و دیگری باید مرزهای روشن/تاریک را تشخیص دهد. پاسخ خود را توضیح دهید.

ب) فیلترهای کانولوشن با اندازه  $3 \times 3$  برای لایه خروجی طراحی کنید، که خروجی مورد نظر در تصویر را محاسبه می‌کند. پاسخ خود را توضیح دهید.

## سوال ۵: (شبیه سازی، ۳۰ نمره)

در این سؤال، شما باید یک پرسپترون چندلایه (Multilayer Perceptron) را پیاده‌سازی کرده و از آن برای پیش‌بینی ارزش خانه‌ها (SalePrice) استفاده کنید. برای انجام این کار، تنها از داده‌های آموزشی و آزمایشی پیوست شده به تمرین استفاده کنید.

(الف) داده‌های این سوال، حاوی داده‌های گمشده هستند. ابتدا دربارهٔ راه‌های متداول رفع مشکل داده‌های گمشده تحقیق کرده و حداقل چهار مورد از این روش‌ها را به صورت خلاصه توضیح دهید. مزایا و معایب هر روش را نیز ذکر کنید.

(ب) درصد گمشدگی داده‌ها برای هر کدام از ویژگی‌های مجموعه داده را اعلام کنید. سپس با استفاده از یک یا چند روش مناسب، مشکل داده‌های گمشده را رفع کنید. در صورتی که قصد حذف رکوردهای حاوی داده‌های گمشده را دارید، اطمینان حاصل کنید که این حذف بیشتر از ۱۰ درصد از کل داده‌های آموزشی + آزمایشی را شامل نشود.

(ج) آیا شما ترجیح می‌دهید از تمام ویژگی‌های موجود در مجموعه داده استفاده کنید، یا فکر می‌کنید ممکن است برخی از ویژگی‌ها مدل را به اشتباه بیندازند، مفید نباشند، یا اضافه و غیرضروری باشند؟ اگر چنین ویژگی‌هایی وجود دارند، کدام‌اند و چرا باید حذف یا نادیده گرفته شوند؟

(د) داده‌های کیفی را با استفاده از روشی مانند one-hot encoding به حالت مناسبی برای ورودی دادن به مدل تبدیل کنید. همچنین داده‌های کمی پیش‌بینی کننده را نرمال‌سازی کنید.

(ه) پرسپترون چندلایه را از صفر پیاده‌سازی کرده و بر روی داده‌های آموزشی آموزش دهید. توجه کنید که استفاده از مدل‌های شبکه عصبی آماده موجود در کتابخانه‌هایی مانند Scikit-Learn مجاز نیست. برای پیاده‌سازی می‌توانید از فریم‌ورک‌های یادگیری عمیق مانند PyTorch و یا TensorFlow استفاده کنید.

(و) مدل پرسپترون چندلایه آموزش دیده خود را بر روی داده‌های آزمایشی تست کرده و معیارهای زیر را برای داده‌های آموزشی و آزمایشی را در نوت‌بوک کد اعلام کنید.

1. Root Mean Squared Error (RMSE)
2. Mean Absolute Error (MAE)



3. Mean Absolute Percentage Error (MAPE)
4. Root Mean Squared Error between the logarithm of the predicted value and the logarithm of the observed sales price.

(ز) در پایان نتایج را تحلیل کنید و شبکه عصبی خود را از منظر **Underfitting** یا **Overfitting** بررسی نمایید. در مدل شما کدام رخ داده است؟ برای کاهش این موضوع در شبکه‌های عصبی چه کارهایی را پیشنهاد می‌کنید؟

## سوال ۴: (شبیه سازی، ۳۰ نمره)

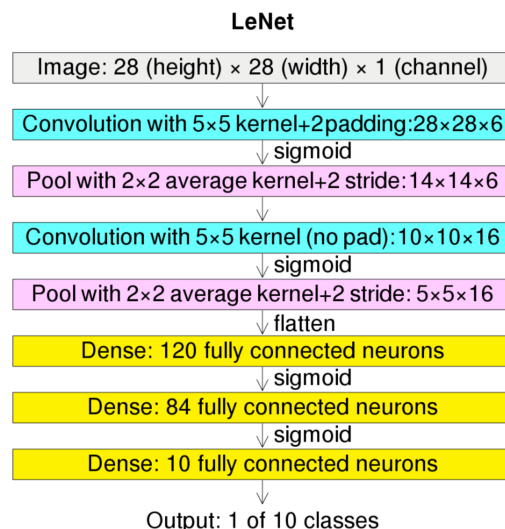
در این سوال، از پرسپترون چندلایه (Multilayer Perceptron)، شبکه عصبی کانولوشنی LeNet و ResNet برای بازشناسایی ارقام دست‌نویس دیتاست MNIST استفاده کرده خواهید کرد. همچنین یکی از ویژگی‌های مهم شبکه‌های عصبی کانولوشنی یعنی Translation Invariance را میان این مدل‌ها بررسی و مقایسه خواهید کرد.

(الف) در ابتدا، ویژگی Translation Invariance شبکه‌های کانولوشنی را توضیح دهید و بگویید به چه معنا است؟

(ب) توضیح دهید چه اجزایی از شبکه عصبی کانولوشنی باعث ایجاد این ویژگی می‌شود؟ (پاسخ دقیق به این سؤال مشخص نیست؛ با این حال، بهترین حدس‌ها و جدیدترین دیدگاه‌های موجود در این زمینه که در حال حاضر مورد قبول هستند ارائه شود.)

(ج) یک شبکه پرسترون چندلایه با معماری دلخواه طراحی کنید و روی مجموعه آموزشی مجموعه داده MNIST آموزش دهید. توجه کنید که شبکه طراحی شده شما حداقل به دقت آزمایشی ۹۰ درصد بر روی دادگان آزمایشی مجموعه داده MNIST برسد.

(د) سپس شبکه LeNet-5 را که معماری آن در شکل زیر نشان داده شده است را پیاده‌سازی کرده و بر روی مجموعه داده MNIST آموزش دهید. انتظار می‌رود که مدل شما به دقت آزمایشی حداقل ۹۵ درصد بر روی داده‌های تست MNIST برسد.



(ه) درباره شبکه‌های عصبی رزیدوال (Residual Neural Network) تحقیق کنید. مزایای این مدل‌ها چه هستند؟ این شبکه‌ها چه اجزای ویژه‌ای در مقایسه با شبکه‌های عصبی کاولوشنی دارند؟

(و) مدل رزیدوال ResNet18 را بر روی مجموعه داده MNIST آموزش داده و دقت آزمایشی را در نوتبوک اعلام کنید. انتظار می‌رود که مدل شما به دقت آزمایشی حداقل ۹۵ درصد روی مجموعه داده MNIST برسد.

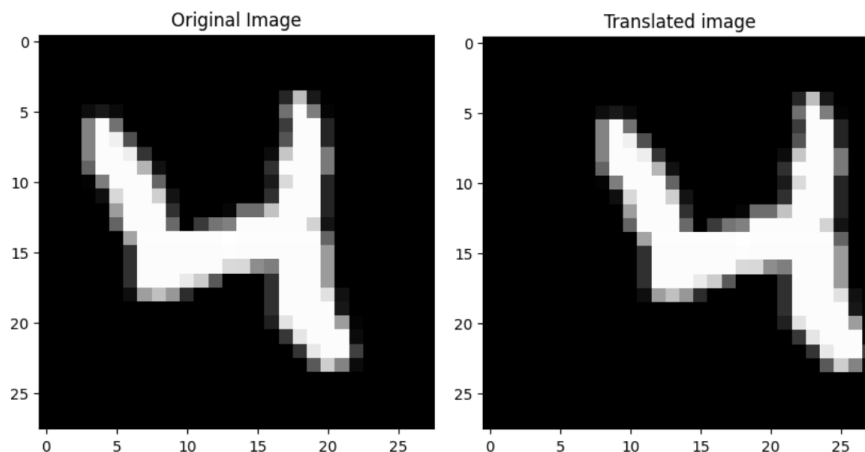
- معماری و جزئیات مدل به صورت پیاده سازی شده در PyTorch در فایل resnet18-specification.txt پیوست شده است.
- برای راحت‌تر شدن پیاده سازی (در حد چند خط کد ساده) تنها برای این مدل اجازه استفاده از مدل آماده resnet18 موجود در PyTorch می‌باشید. برای این کار کافی است مدل را با کد زیر بسازید:

```
model=torchvision.models.resnet18(pretrained=False)
```

دقت کنید که resnet18 در اصل بر روی دیتاست ImageNet آموزش داده شده است و در نتیجه لایه‌های ورودی و خروجی آن مناسب دیتاست MNIST نیستند و لازم است که این لایه‌ها را با توجه به معماری داده شده در فایل resnet18-specification.txt تغییر دهید.

(ز) در نهایت، یک تصویر به انتخاب خودتان را از مجموعه داده MNIST در نظر بگیرید و عکس را چند پیکسل شیفت دهید با این هدف که عدد عکس درون آن جا به جا شود. توجه کنید این جا به جایی نباید به صورتی باشد که عدد از کادر درون تصویر خارج شود.

سپس این عکس را به شبکه‌های MLP، LeNet و ResNet آموزش داده شده خود بدهید و کلاس‌های پیش بینی شده برای آن را مقایسه کنید. در صورتی که به نتیجه مورد نظر نرسیدید عکس با کلاس‌های متعدد و شدت اعمال Translation از نظر دوری و نزدیکی امتحان کنید.



### نکات نهایی:

- برای راحتی پیاده‌سازی و استفاده از ResNet18، ترجیحاً از PyTorch با توضیحات داده شده استفاده کنید.
- تنها برای ResNet می‌توانید از مدل آماده در فریم‌ورک استفاده کنید. مدل MLP را با معماری دلخواه و مدل LeNet-5 را با توجه به معماری داده شده باید از صفر با استفاده از یکی از فریم‌ورک‌های یادگیری عمیق مانند PyTorch یا TensorFlow پیاده‌سازی کنید.
- دقت کنید که استفاده از کارت گرافیک برای آموزش مدل باعث سریع‌تر شدن این فرایند خواهد شد. اگر به کارت گرافیک دسترسی مستقیم ندارید، می‌توانید از کارت گرافیک‌هایی که Google Colab و Kaggle به طور رایگان ارائه می‌دهند استفاده کنید.

با آرزوی سلامتی و موفقیت برای شما