به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین اول

پرسش‌های ۱ و ۲	نام دستیار طراح	پویا صادقی
	رایانامه	pouyasadeghi2012@gmail.com
پرسش‌های ۳ و ۴	نام دستیار طراح	حدیثه مصباح
	رایانامه	hadisehmesbah@gmail.com
	مهلت ارسال پاسخ	۱۴۰۳.۰۱.۲۰

فهرست

شکل‌ها

جدول‌ها

قوانین

قبل از پاسخ دادن به پرسش‌ها، موارد زیر را با دقت مطالعه نمایید:

- از پاسخ‌های خود یک گزارش در قالبی که در صفحه‌ی درس در سامانه‌ی Elearn با نام ***REPORTS_TEMPLATE.docx*** قرار داده شده تهیه نمایید.
- پیشنهاد می‌شود تمرین‌ها را در قالب گروه‌های دو نفره انجام دهید. (بیش از دو نفر مجاز نیست و تحویل تک نفره نیز نمره‌ی اضافی ندارد) توجه نمایید الزامی در یکسان ماندن اعضای گروه تا انتهای ترم وجود ندارد. (یعنی، می‌توانید تمرین اول را با شخص A و تمرین دوم را با شخص B و ... انجام دهید)
- **کیفیت گزارش شما در فرآیند تصحیح از اهمیت ویژه‌ای برخوردار است**؛ بنابراین، لطفا تمامی نکات و فرض‌هایی را که در پیاده‌سازی‌ها و محاسبات خود در نظر می‌گیرید در گزارش ذکر کنید.
- در گزارش خود مطابق با آنچه در قالب نمونه قرار داده شده، برای شکل‌ها زیرنویس و برای جدول‌ها بالانویس در نظر بگیرید.
- الزامی به ارائه توضیح جزئیات کد در گزارش نیست، اما باید نتایج بدست آمده از آن را گزارش و تحلیل کنید.
- **تحلیل نتایج الزامی می‌باشد، حتی اگر در صورت پرسش اشاره‌ای به آن نشده باشد.**
- **دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند**؛ بنابراین، هرگونه نتیجه و یا تحلیلی که در صورت پرسش از شما خواسته شده را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می‌شود.
- **کدها حتما باید در قالب نوت‌بوک با پسوند .ipynb تهیه شوند، در پایان کار، تمامی کد اجرا شود و خروجی هر سلول حتما در این فایل ارسالی شما ذخیره شده باشد.** بنابراین برای مثال اگر خروجی سلولی یک نمودار است که در گزارش آورده‌اید، این نمودار باید هم در گزارش هم در نوت‌بوک کدها وجود داشته باشد.
- **در صورت مشاهده‌ی تقلب امتیاز تمامی افراد شرکت‌کننده در آن، 100- لحاظ می‌شود.**
- تنها زبان برنامه نویسی مجاز **Python** است.
- **استفاده از کدهای آماده برای تمرین‌ها به هیچ وجه مجاز نیست. در صورتی که دو گروه از یک منبع مشترک استفاده کنند و کدهای مشابه تحویل دهند، تقلب محسوب می‌شود.**

- نحوه محاسبه تاخیر به این شکل است: پس از پایان رسیدن مهلت ارسال گزارش، حداکثر تا یک هفته امکان ارسال با تاخیر وجود دارد، پس از این یک هفته نمره آن تکلیف برای شما صفر خواهد شد.
  - سه روز اول: بدون جریمه
  - روز چهارم: ۵ درصد
  - روز پنجم: ۱۰ درصد
  - روز ششم: ۱۵ درصد
  - روز هفتم: ۲۰ درصد
- حداکثر نمره‌ای که برای هر سوال می‌توان اخد کرد ۱۰۰ بوده و اگر مجموع بارم یک سوال بیشتر از ۱۰۰ باشد، در صورت اخد نمره بیشتر از ۱۰۰، اعمال نخواهد شد.
  - برای مثال: اگر نمره اخذ شده از سوال ۱ برابر ۱۰۵ و نمره سوال ۲ برابر ۹۵ باشد، نمره نهایی تمرین ۹۷.۵ خواهد بود و نه ۱۰۰.
- لطفا گزارش، کدها و سایر ضمایم را به در یک پوشه با نام زیر قرار داده و آن را فشرده سازید، سپس در سامانه‌ی Elearn بارگذاری نمایید:

HW[Number] _[Lastname]_[StudentNumber]_[Lastname]_[StudentNumber].zip

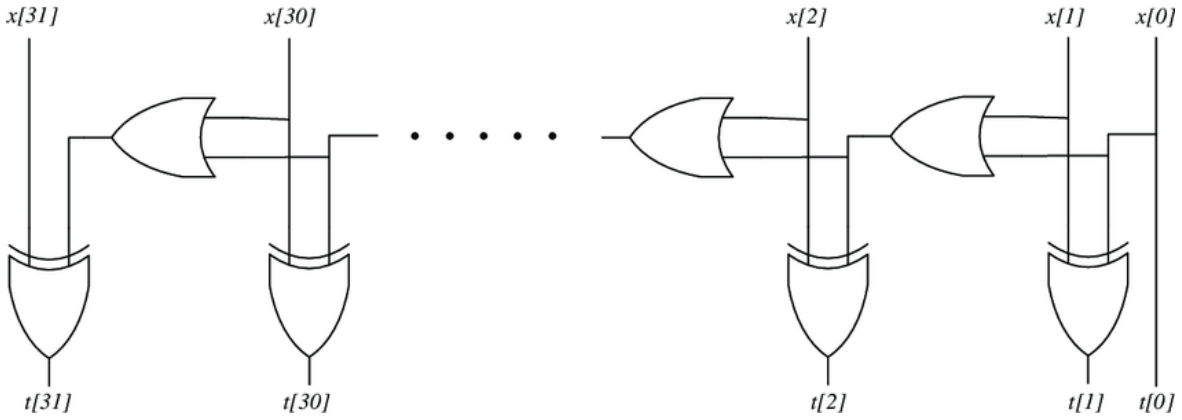
(مثال: HW1_Ahmadi_810199101_Bagheri_810199102.zip)

- برای گروه‌های دو نفره، بارگذاری تمرین از جانب یکی از اعضا کافی است ولی پیشنهاد می‌شود هر دو نفر بارگذاری نمایند.

پرسش ۱ – Mcculloch Pitts

۱-۱. شبکه محاسبه مکمل ۲

در این تمرین شما به کمک نورون توسعه یافته Mcculloch Pitts یک شبکه محاسبه کننده 2's complement خواهید ساخت که یک عدد ۴-بیتی را به عنوان ورودی گرفته و مکمل ۲ این عدد را در خروجی برمی‌گرداند. توجه شود که در این سوال تمامی نورون‌های ورودی و خروجی به صورت باینری می‌باشند. ساختار مرتبه گیت این مدار برای ورودی ۳۲-بیتی به صورت زیر است:



شکل ۱. تصویر نمونه از ساختار شبکه مدنظر

۲-۱. پیاده سازی تئوری شبکه

- این شبکه را به همراه نورون‌ها و ارتباطات بین آن‌ها رسم کنید. (۲۰نمره)
- وزن هر یال را محاسبه کنید. (۲۰نمره)

توجه شود همانگونه که در شکل داده شده دیده میشود، خروجی این مدار به صورت ترتیبی ایجاد می‌شود. شما در این مرحله نیازی نیست نگران این موضوع باشید و فقط ترسیم شبکه و محاسبه وزن‌ها را انجام دهید.

۳-۱. پیاده سازی کد شبکه

در این مرحله با توجه به ترسیمات مرحله قبل، به پیاده سازی این شبکه با زبان پایتون خواهید پرداخت. در این بخش از پکیج‌های مناسب جهت پیاده سازی می‌توانید استفاده کنید اما توجه شود هدف این است که شبکه و نورون‌ها را خود شما پیاده سازی کنید. همچنین در این مرحله با توجه به اهمیت ترتیب محاسبات، در صورت نیاز در کد خود ترتیب را رعایت کنید، بدین صورت که در ابتدا t[۰] و سپس t[۱] و پس از آن t[۲] و در انتها t[۳] محاسبه خواهند شد. توجه شود که امکان موازی سازی نیز وجود دارد و در گزارش خود روش پیاده سازی را به صورت مختصر توضیح دهید. (۴۵ نمره)

با دادن چندین ورودی و خروجی متفاوت، صحت عملکرد شبکه را نمایش دهید. (۱۵ نمره)

پرسش ۲ – حملات خصمانه در شبکه‌های عصبی

۲-۱. حملات خصمانه

- همزمان با گسترش استفاده شبکه های عصبی در کاربردهای مختلف، سرفصل حملات به آن‌ها نیز گشوده شد. حملات به مدل‌های هوش مصنوعی را از نظر دسترسی به مدل می‌توان به دو دسته اصلی زیر تقسیم کرد:
- حملات white box: در اینگونه حملات به مدل دسترسی داریم و معمولا از بازگشت گرادیان جهت مهندسی سازی حمله استفاده می‌شود و از نوع ساده ترین حملات است. در این تمرین ما این مدل حمله را بررسی می‌کنیم.
- حملات black box: در این گونه حملات ما به مدل دسترسی نداریم.

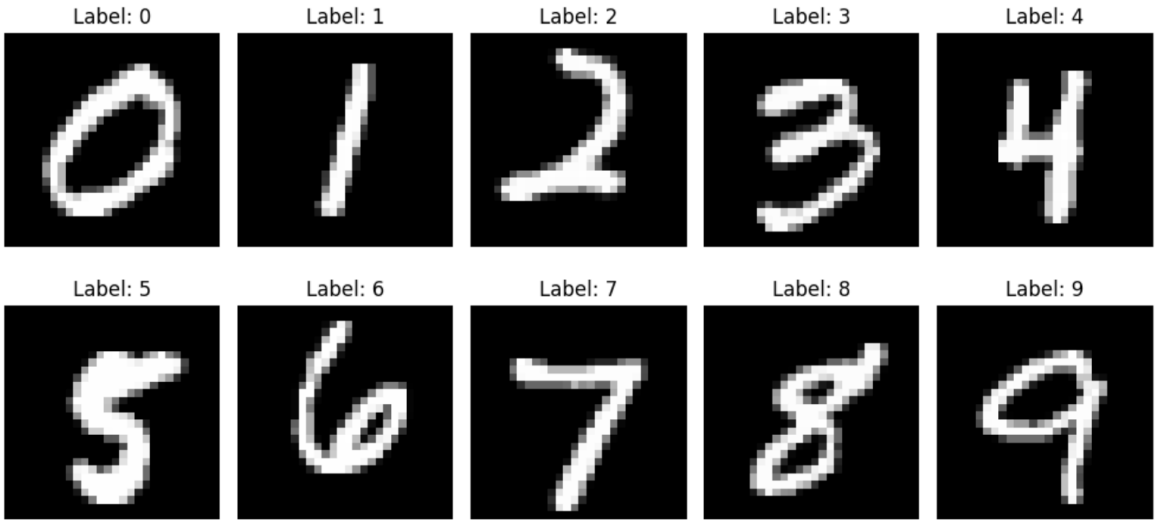
حملات gray box نیز وجود دارند که موضوع بررسی ما نمی‌باشند. همچنین از نظر هدف حمله نیز دو دسته می‌توانیم تعریف کنم:

- تغییر رفتار مدل به هدفی خاص، به عنوان مثال با دادن یک داده، مدل رفتار مشخصی را انجام دهد.
- کاهش دقت مدل، به این صورت که پیشبینی و خروجی مدل برای ما اهمیتی ندارد و هدف این است که مدل لیبل درست را تشخیص ندهد و دچار اشتباه شود. در اینجا اینگونه حملات مدنظر ما می‌باشند.

۲-۲. آشنایی با مجموعه دادگان

هدف از این قسمت، آشنایی و کار کردن با مجموعه داده مورد نظر است.

- ابتدا مجموعه دادگان MNIST را که مجموعه‌ای از ارقام دست‌نویس می‌باشد را فراخوانی کنید. سپس تعداد و ابعاد دادگان آموزش و آزمون را گزارش کنید. (۵ نمره)
- یک نمونه تصویر از هر کلاس را نمایش دهید. (۵ نمره)
- نمودار histogram مربوط به تعداد نمونه های هر کلاس را برای دادگان آموزش و آزمون رسم کنید و توزیع کلاس‌ها را بررسی کنید. آیا پردازشی جهت بالانس کردن نیاز است؟ (۵ نمره)
- با استفاده از min-max normalization، داده‌ها را به بازه $[0,1]$ اسکیل کنید. علت این کار را توضیح دهید. (۵ نمره)



شکل ۲: نمونه تصاویر از کلاس‌های مجموعه دادگان **MNIST**

۲-۳. ایجاد و آموزش مدل

در این قسمت، یک مدل با ساختار داده شده را برروی این مجموعه دادگان، با پارامترهای گفته شده آموزش دهید. (۲۰ نمره)

MODEL STRUCTURE

LAYER	Input -> Output	Activation
FC_1	512 <- 784	ReLU
FC_2	128 <- 512	ReLU
FC_3	32 <- 128	ReLU
FC_4	classes# <- 32	SoftMax

جدول ۱. مشخصات شبکه عصبی مدنظر برای پیاده سازی

MODEL'S TRAINING PARAMETERS

OPTIMIZER	Adam
LEARNING RATE	6e-5
EPOCHS	25
CRITERION	CrossEntropyLoss

جدول ۲. پارامترهای آموزش شبکه عصبی پیاده سازی شده

- توضیح دهید چرا ورودی این شبکه به سایز ۷۸۴ انتخاب شده است؟ (۵نمره)
- توضیح دهید چرا در لایه آخر تابع فعال سازی متفاوت است و دلیل استفاده از آن چیست؟ (می‌توانید درباره مفهوم logitها تحقیق کنید) (۵ نمره)

۴-۴. پیاده سازی حمله FGSM

در این بخش هدف پیاده سازی حمله FGSM بر روی شبکه آموزش داده مرحله قبل می‌باشد. برای آشنایی بیشتر با این نوع حمله، می‌توانید این مقاله را مطالعه کنید. برای انجام این حمله بصورت خلاصه، باید گرادیان را محاسبه کنید و با backpropagation آن، نویز مهندسی شده مدنظر را ایجاد کنید و سپس نویز را به تصویر اصلی اضافه کنید.

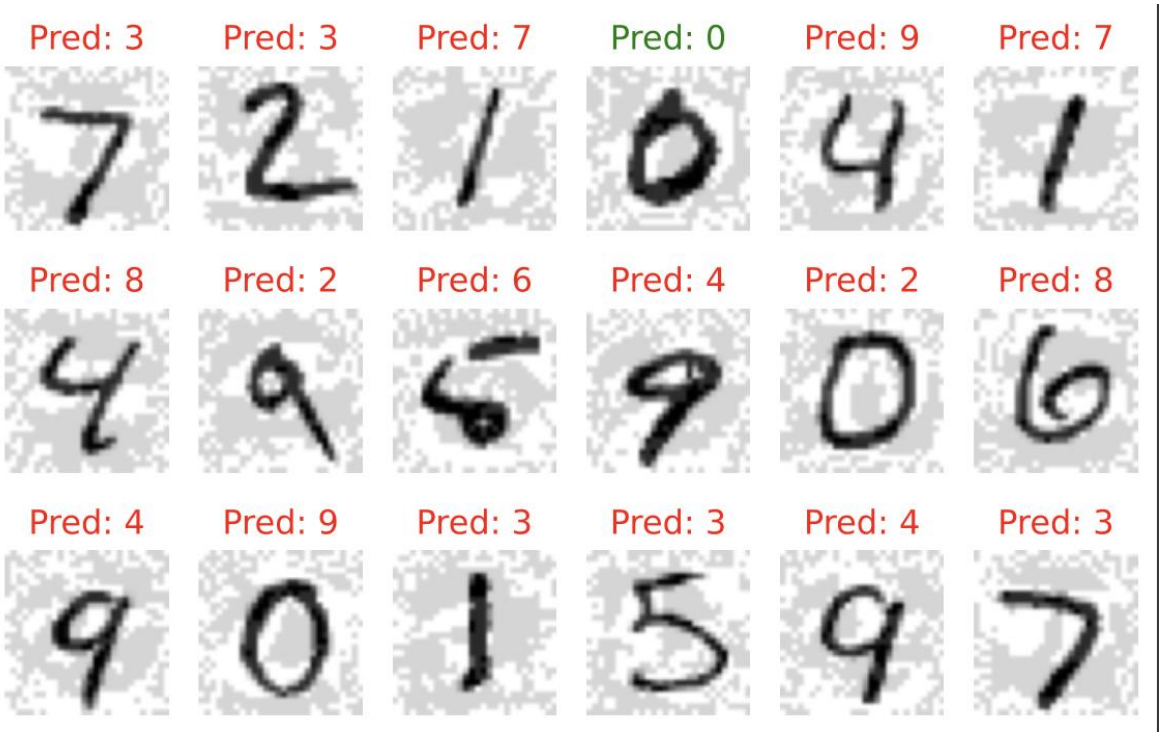
برای این منظور، یک لایه ماسک نویز با ابعاد مشابه تصویر اصلی و مقدار صفر ایجاد کنید. این ماسک را به تصویر ورودی اضافه کرده و تصویر را به مدل ورودی بدهید. در ادامه با تابع هزینه CrossEntropyLoss می‌توانید گرادیان را محاسبه کرده و سپس هزینه را برای نویز محاسبه کنید. سپس با delta = epsilon*SignOf(loss) می‌توانید نویز مهندسی شده را بدست آورید. در انتها با ورودی دادن تصویر به همراه این نویز به شبکه، این حمله صورت خواهد پذیرفت. مقدار پارامتر اپسیلون را به دلخواه می‌توانید قرار بدید. مقدار این پارامتر در بازه ۰,۱ می‌باشد.

```python
def fgsm(model, X, y, epsilon):
    """
    Construct FGSM adversarial examples on the examples
    :parameter
    model: Your DNN model
    X: Input example(s)
    y: Ground truth labels
    epsilon: A float number
    """
    # delta = a placeholder to aggregate gradient and then calc sign
    # you should calculate models loss in here and backward gradient on delta
    return epsilon * delta.grad.detach().sign()

## How to use adversarial samples
# delta = fgsm(..., 0.1)
# yp = your_dnn_model(X+delta)
```

شکل ۳ : نمونه کد از نحوه پیاده سازی تابع مورد نیاز برای حمله **fgsm** و نحوه انجام حمله

- حمله را پیاده سازی کنید. (۱۵ نمره)
- بررسی کنید در چند درصد مواقع این حمله موثر واقع شده است و چقدر دقت مدل را تحت تاثیر قرار داده است؟ (۵ نمره)
- برای هر کلاس، حداقل یک نمونه از تصاویر به همراه نویز ایجاد شده به همراه برچسب اصلی و برچسب پیشبینی شده توسط مدل را نمایش دهید. (۵ نمره)



شکل ۴: نمونه تصاویر به همراه نویز مهندسی شده و نتیجه پیشبینی مدل برای آن‌ها

۴-۵. پیاده سازی حمله PGD

در این بخش هدف پیاده سازی حمله PGD بر روی شبکه آموزش دیده می‌باشد. این مقاله به شما در درک این حمله کمک خواهد کرد. این حمله را پیاده سازی کرده و کارآیایی آن را برروی شبکه تحلیل کنید. پارامترهای این حمله را به دلخواه و به صورت منطقی می‌توانید مقدار دهی کنید.

- این حمله را پیاده سازی کنید. (۱۰ نمره)
- تفاوت‌های این روش با FGSM را توضیح دهید. (۵ نمره)
- چرا از این روش استفاده می‌شود و نسبت به FGSM چه مزایایی دارد؟ در چند درصد مواقع این حمله موثر واقع شده است و چه بهبودهایی نسبت به روش قبل مشاهده می‌کنید؟ (۵ نمره)
- برای هر کلاس، حداقل یک نمونه از تصاویر به همراه نویز ایجاد شده به همراه برچسب اصلی و برچسب پیشبینی شده توسط مدل را نمایش دهید. (۵ نمره)

پرسش ۳ – Adaline و Madaline

در این پرسش به بررسی دو روش Adaline و Madaline پرداخته خواهد شد.

۳-۱. Adaline

در این بخش با استفاده از روش Adaline یک شبکه عصبی آموزش داده خواهد شد که در مجموعه داده Wine (که از ۳ نوع شراب مختلف به نام‌های Class 1، Class 2 و Class 3 تشکیل شده)، نوع Class 1 را از سایر دسته‌ها تشخیص دهد.

الف) ابتدا نمودار پراکندگی داده‌ها را در دو بعد رسم کنید (برای سادگی از دو ویژگی اول یعنی Alcohol و Malic Acid استفاده شود)، سپس یک شبکه Adaline روی این داده‌ها آموزش دهید. همچنین در این گام آموزش، نمودار تغییرات خطا، یعنی $\frac{1}{2}(\text{target} - \text{net})^2$ را رسم نمایید. (۲۰ نمره)

ب) حال این کار را برای نوع شراب Class 2 انجام دهید (بدین معنی که مجموعه داده را به دو بخش Class 2 و Non-Class 2 تقسیم کرده و آموزش بر روی این داده‌ها انجام دهید). سپس دلیل خوب یا بد جدا شدن داده‌ها را نسبت به بخش الف توضیح دهید. (۲۰ نمره)

۳-۲. Madaline

در این بخش به پیاده سازی شبکه Madaline بر روی یک مجموعه داده مصنوعی پرداخته میشود.

الف)ابتدا الگوریتم های MRI و MRII را که در کتاب مرجع موجود است، انتخاب کرده و توضیح مختصری بدهید. (۱۰ نمره)

ب) برای آموزش از مجموعه داده مصنوعی، مطابق با آ نچه در شکل زیر نشان داده شده استفاده کنید. با استفاده از یکی از الگوریتم هایی که در بخش الف مطالعه نمودید، شبکه را آموزش دهید. سپس با تعداد نورو نهای متفاوت(یک بار ۳ نورون، یک بار ۵ نورون و یک بار ۸ نورون) نقاط را از هم جدا کنید. نهایتاً دقت جداسازی را در همه حالات نمایش دهید. (۴۰ نمره)

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_gaussian_quantiles
from sklearn.model_selection import train_test_split

np.random.seed(42)

X, y = make_gaussian_quantiles(n_samples=300, n_features=2,
n_classes=2, random_state=42)

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired, marker='o',
edgecolors='k')
plt.title('Nonlinear Separable Data (make_gaussian_quantiles)')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```

پرسش ۴ – شبکه‌ی عصبی بهینه

۴-۱. رگرشن

شبکه‌های عصبی به ویژه در مسائل رگرسیون، ممکن است به مشکل overfitting (برازش بیش از حد) برخورد کنند. Overfitting به معنای این است که مدل به جای یادگیری، رفتار داده‌های آموزش را حفظ می‌کند و دیگر برای داده‌های جدید تعمیم‌پذیر نیست.

الف) Overfitting چه دلایلی دارد؟ (۲ نمره)

ب) چه راه‌حل‌هایی برای رفع Overfitting وجود دارد؟ (۲ نمره)

پ) در شبکه‌های عصبی تعداد هایپرپارامتر چیست و نحوه‌ی مقداردهی بهینه‌ی آن‌ها چگونه است؟ فرق آن با هایپر پارامتر چیست ؟ (۶ نمره)

ت) از کد زیر استفاده کنید تا تعدادی نمونه‌ی سینوسی را بسازید:

```
import numpy as np
num_points = 1000
x_values = np.linspace(0, 4 * np.pi, num_points)
y_values = np.sin(x_values)
```

افزایش داده‌ها

- از داده‌های تولید شده ۱۰۰ نمونه به صورت تصادفی بردارید و بقیه را برای تست کنار بگذارید، حال شبکه‌ی عصبی دولایه با استفاده از این داده‌ها آموزش دهید و نتیجه را رسم کنید.
- هربار ۱۰۰ تا از داده‌های آموزش برداشته و به داده‌های تست اضافه کنید تا به نسبت ۹۰٪ به ۱۰٪ برسید و شبکه‌ی عصبی دو لایه را هربار آموزش دهید و نتایج را رسم کنید (تصاویر به دست آمده از شبکه عصبی را به صورت گیف نمایش دهید)
- چه تغییری در طول افزایش تعداد داده‌ها مشاهده می‌کنید؟ (۱۵ نمره)

افزایش لایه‌ها

- ٪90 نمونه‌ها را برای آموزش و ٪۱۰ آن‌ها را برای تست کنار بگذارید.
- از یک شبکه‌ی عصبی یک لایه شروع کرده و هربار آن آموزش دهید و نتایج را رسم کنید.
- هر مرحله تعداد لایه‌های شبکه‌ی عصبی را یکی زیاد کنید تا به 20 لایه برسید و نتایج را رسم کنید (تصاویر بدست آمده از شبکه‌ی عصبی را به صورت گیف نمایش دهید)
- چه تغییری با افزایش لایه‌ها مشاهده می‌کنید؟ (۲۰ نمره)

ث) حال سعی کنید با استفاده از GridSearchCV در کتابخانه‌ی sklearn تعداد لایه‌های بهینه برای مساله‌ی بالا را پیدا کنید. در چه مواردی این تکنیک مناسب نیست؟ (۱۵ نمره)

۴-۲. طبقه‌بندی

الف) حال که نحوه‌ی Overfitting در رگرشن را دیدید، این کار را در وظیفه طبقه‌بندی نیز امتحان می‌کنیم. مجموعه داده MNIST را در نظر بگیرید.

افزایش داده‌ها

- ٪۵ داده‌ها را برای آموزش کنار بگذارید (مجموعه داده‌ی تست برای MNIST جدا است) حال یک شبکه‌ی عصبی سه لایه را برای آن آموزش دهید و هربار نتایج را ذخیره کنید.
- هر مرحله ٪۱۵ بیشتر نسبت به مرحله‌ی قبل داده برای آموزش کنار گذاشته و شبکه را آموزش دهید، و نتایج را ذخیره کنید.
- چه تغییری در طول افزایش تعداد داده‌ها مشاهده می‌کنید؟ (۲۰ نمره)

افزایش لایه‌ها

- کل داده‌های آموزش را برای آموزش یک شبکه‌ی عصبی سه لایه استفاده کرده و هربار نتایج را ذخیره کنید.
- هر بار یک لایه به شبکه‌ی عصبی اضافه کنید تا به 20 لایه برسید. و پس از هر آموزش نتایج را ذخیره کنید.
- چه تغییری در طول افزایش تعداد داده‌ها مشاهده می‌کنید؟ (۲۰ نمره)

ب) آیا می‌توانید شبکه‌ی عصبی بهینه برای MNIST را پیدا کنید (اگر در اینترنت آن را پیدا کردید با ذکر مرجع آن را پیاده‌سازی کنید و نتایج را گزارش کنید.) (۱۰ نمره امتیازی)