

به نام خدا



دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده مهندسی برق و کامپیوتر

هوش مصنوعی

پروژه پنجم (فاز اول)

پیاده‌سازی و آموزش شبکه‌های عصبی Forward Feed

مهلت تحویل: خرداد ۱۴۰۰

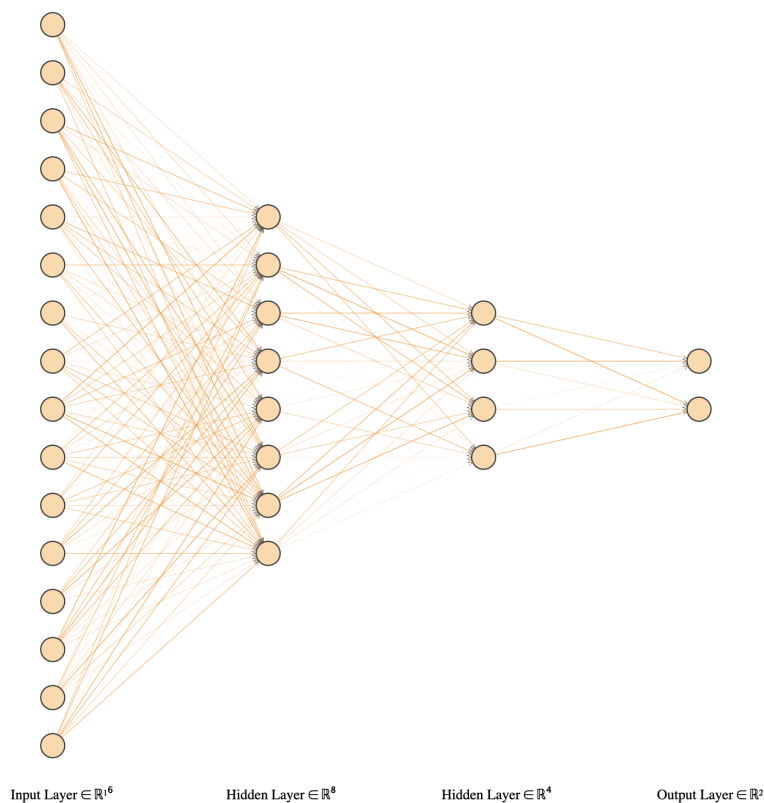
طراحان: آرش هاتفی - حمید خدادادی - سینا نگارنده

بهار ۱۴۰۰

## مقدمه:

### شبکه عصبی Feed Forward

در فاز اول پروژه پنجم به پیاده سازی شبکه‌های عصبی Feed Forward جهت طبقه‌بندی تصاویر می‌پردازیم.



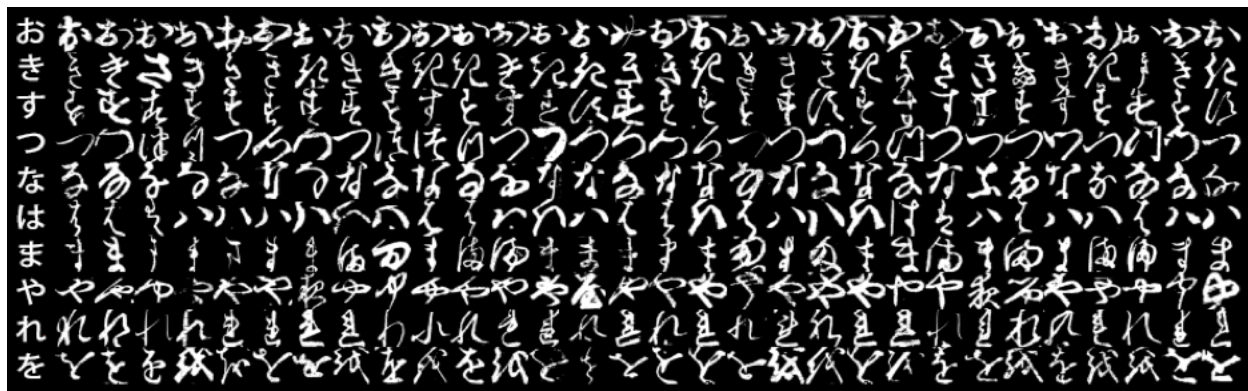
در شبکه‌های عصبی Feed Forward که در درس نیز با آن آشنا شدید، هر تصویر ابتدا مسطح شده و به صورت بردار به عنوان ورودی شبکه داده می‌شود. هر درایه این بردار (معادل با یک پیکسل تصویر) یک ویژگی برای آن محسوب می‌شود. شبکه قرار است بر اساس این ویژگی‌ها و با ساختن ترکیبات غیرخطی از آنها، وزن اتصالات بین لایه‌هایش را طوری تنظیم کند، که خروجی آن ضمن داشتن کمترین خطا، کلاس تصویر ورودی متناظر را به درستی پیش بینی کند.

## تعریف مسئله:

در این تمرین، در بخش اول به پیاده‌سازی یک شبکه‌ی عصبی Feed Forward از پایه و با استفاده از کتابخانه NumPy می‌پردازید. جهت تسریع این فرایند، یک Notebook ناقص از پیاده‌سازی شبکه نیز در اختیار شما قرار می‌گیرید که لازم است از آن استفاده نمایید. در بخش دوم، به کمک کدهای بخش اول، چند شبکه‌ی عصبی را روی داده‌های داده شده آموزش خواهید داد و تاثیر برخی از پارامترها را در فرایند یادگیری بررسی خواهید کرد.

## معرفی مجموعه داده:

در این تمرین شما با یک مجموعه داده KMnist (کوتاه شده Kuzushiji-MNIST) شامل تصاویری در مقیاس 28 در 28 پیکسل به صورت grayscale و 70,000 تصویر از 10 نویسه یا کاراکتر مختلف، معرف یک هجا در خط هیراگانا (Hiragana) که کلمات با استفاده از آن‌ها ساخته می‌شوند (شبيه به یک الفبا)، کار خواهید کرد. فایل داده‌ها در صفحه درس قرار داده شده است.



kmnist\_dataset

└─ test\_data.csv

└─ test\_labels.csv

└─ train\_data.csv

└─ train\_labels.csv

مجموعه به دو قسمت train و test تقسیم شده که به تعداد 60,000 آن برای آموزش و 10,000 برای تست قرار گرفته اند. هر سطر از فایل های داده ها، به صورت یک بردار از یک تصویر مسطح به اندازه ی 784 (28 در 28) است. فایل های برچسب ها شامل برچسب داده ها است. هر برچسب می تواند با یکی از اعداد 0 تا 9 نمایش داده شود که به نمایندگی از یک هجا می باشد.

## فاز اول: بررسی و پیش پردازش داده

اولین تصویر را در مجموعه داده train را بررسی کنید (از کتابخانه matplotlib استفاده کنید)، مقادیر هر پیکسل در چه محدوده ای قرار می گیرد؟

مقدار داده ها را به گونه scale کنید که قبل از اطلاعات دادن به شبکه عصبی، مقدار هر pixel بین 0 تا 1 باشد. در صورت انجام ندادن این کار چه مشکلی ممکن است رخ دهد؟

در مجموعه داده train، از هر کلاس، به دلخواه یک تصویر انتخاب کرده و نمایش دهید. برای هر تصویر، نوع آن را نیز به همراه تصویر نمایش دهید.

تعداد تصاویر هر دسته را برای مجموعه داده train و test محاسبه کنید و برای آن ها نمودار میله ای رسم کنید.

## فاز دوم: تکمیل بخشهای ناقص شبکه عصبی

یک فایل Notebook شامل کدهای ناقص مورد نیاز برای پیاده سازی شبکه عصبی Feed Forward آپلود شده و در این قسمت با تکمیل بخشهای مختلف این فایل، نهایتاً یک کلاس FeedForwardNN خواهید داشت که به کمک آن می توانید شبکه های عصبی Feed Forward با معماری های مختلف پیاده کنید و آموزش دهید. پارامترهای شبکه مورد نظر از طریق روش Stochastic Gradient Discent در طی فرایند آموزش به روز رسانی خواهد شد. بخش های حذف شده از کد که لازم است آنها را کامل کنید، با #TODO مشخص شده اند.

معرفی کلاس ها به همراه method های آنها و توضیحات تکمیلی راجب فایل NoteBook در فایل pdf کنار آن قرار گرفته است.

## فاز سوم: طبقه بندی داده ها

در این بخش به پیاده سازی و آموزش شبکه‌های عصبی Feed Forward با پارامترهای مختلف به کمک کلاس FeedForwardNN می‌پردازیم. در بخش Training Sample از NoteBook ناقص، یک مثال از شیوه‌ی استفاده از کلاس FeedForwardNN جهت پیاده سازی شبکه‌های عصبی آورده شده است.

یک شبکه‌ی عصبی طراحی کنید و آن را با شرط‌های زیر آموزش دهید:

- مقدار learning rate باید 0.01 باشد.
- تعداد epochها باید 10 باشد.
- اندازه batch\_size باید 32 باشد.
- تابع فعالساز relu در تمام لایه‌ها استفاده شود. (به جز لایه آخر که Identical است)
- از وزن‌دهی اولیه unifrom و یا normal استفاده کنید.
- شبکه‌ی عصبی حداکثر باید سه لایه‌ی پنهان داشته باشد.

در تمام بخش‌های آینده، باید نتیجه خود را به صورت پیشفرض با شرط‌های گفته شده بالا بدست آورید و فقط پارامترهای خواسته شده در هر مرحله را تغییر دهید.

### - قسمت اول) آموزش شبکه

یک شبکه‌ی عصبی با شرط‌های گفته شده را طراحی کنید و آموزش دهید. (می‌توانید از شبکه‌ای که در بخش Training Sample قرار گرفته استفاده کنید.)

### - قسمت دوم) وزن‌دهی شبکه

مقدار اولیه وزن‌ها در آموزش شبکه اهمیت دارد.

اگر مقدار اولیه تمام وزن‌های شبکه برابر صفر قرار بود و شبکه را آموزش می‌دادید، نتیجه آن چه بود؟ خروجی را با حالت قبل (که در آن وزن‌ها به صورت مقدار تصادفی مقداردهی اولیه می‌شدند)، مقایسه کنید. نیازی به پیاده سازی نیست، تنها توضیح دهید.

#### - قسمت سوم) تاثیر learning rate

یکی از پارامترهای مهم در آموزش دادن شبکه های عصبی، learning rate می باشد. حال با کاهش و افزایش این پارامتر، شبکه را آموزش دهید و پس از یافتن مقدار بهینه برای شبکه خود، نتیجه را گزارش کنید. همچنین رفتار شبکه را برای learning rate با مقدار بالاتر و پایین تر نسبت به حالت بهینه به دست آمده توجیه کنید. برای تمام قسمتهای بعد، از learning rate بهینه‌ای که به دست آورده‌اید استفاده کنید.

#### - قسمت چهارم) تاثیر batch size

عملکرد شبکه را به ازای با batch size مقدار 16 و 64 و 128 نیز بسنجید و نتایج را توضیح دهید. علت استفاده از batch در فرایند آموزش چیست؟ مزایا و معایب احتمالی batch size بسیار کوچک و بسیار بزرگ را شرح دهید. به نظر شما با افزایش batch size آیا باید learning rate را افزایش دهیم یا کاهش؟ برای این کار در حالتی که batch size را 128 گذاشته اید learning rate را یک بار افزایش و یک بار کاهش دهید و نتیجه را تفسیر کنید. برای تمام قسمتهای بعد، از batch size بهینه‌ای که به دست آورده‌اید استفاده کنید.

#### - قسمت پنجم) تاثیر activation function

عملکرد شبکه‌ی طراحی شده در قسمت اول را به کمک Activation Function های زیر بسنجید و نتایج را مقایسه نمایید.

## Activation Function: Sigmoid - Hyperbolic tangent (Tanh) - Leaky ReLU

دلیل اینکه tanh و sigmoid عملکرد مناسبی برای این دست شبکه‌ها ندارند را بیان کنید.

برتری Leaky Relu نسبت به Relu چیست؟

توجه: در ادامه مراحل، از activation function با بهترین نتیجه را، در لایه‌های شبکه استفاده نمایید.

### - قسمت ششم) تاثیر epoch

تعداد epoch هایی که یک شبکه ی عصبی به آن اندازه آموزش داده می شود در نتیجه نهایی موثر است.

دلیل اینکه نیاز است تا شبکه‌های عصبی برای چند epoch آموزش داده شوند را توضیح دهید؟ حال تعداد epoch هایی که شبکه را آموزش می دهید را به 15 epoch تغییر دهید و نتایج را مقایسه و تفسیر کنید.

آیا همواره افزایش تعداد epoch های بیشتر برای تمرین مفید است؟

در صورت که این روش همواره مفید نیست، چه راه حلی برای پیش‌گیری از اتفاق نامطلوبی که روی می‌دهد دارید؟

برای تمام قسمت‌های بعد، از epoch بهینه‌ای که به دست آورده‌اید استفاده کنید.

### - قسمت هفتم) تاثیر momentum

پارامتر momentum در optimizer یکی دیگر از پارامترهایی می‌تواند روی عملکرد شبکه تاثیر بگذارد.

عملکرد این پارامتر را به اختصار توضیح دهید. مزایا و معایب این پارامتر را شرح دهید.

پارامتر momentum را در شبکه پیاده سازی کنید. مقدار پارامتر را به ترتیب 0.5، 0.9 و 0.99 قرار دهید. نتیجه را گزارش دهید.

آیا افزایش momentum لزوما باعث افزایش دقت و آموزش بهتر شبکه می شود؟

### نکات پایانی:

- دقت کنید که هدف پروژه تحلیل نتایج است بنابراین از ابزارهای تحلیل داده بطور مثال نمودارها استفاده کنید و توضیحات مربوط به هر بخش از پروژه را بطور خلاصه و در عین حال مفید در گزارش خود ذکر کنید.
- نتایج و گزارش خود را در یک فایل فشرده با عنوان `AI_CA5_<#SID>.zip` تحویل دهید. محتویات پوشه باید شامل فایل `jupyter-notebook`، خروجی `html` و فایل های مورد نیاز برای اجرای آن باشد. توضیح و نمایش خروجی های خواسته شده بخشی از نمره این تمرین را تشکیل می دهد. از نمایش درست خروجی های مورد نیاز در فایل `html` مطمئن شوید.
- در صورتی که سوالی در مورد پروژه داشتید بهتر است در فروم درس مطرح کنید تا بقیه از آن استفاده کنند؛ در غیر این صورت توسط ایمیل با طراحان در ارتباط باشید.
- هدف از تمرین، یادگیری شماسست. لطفا تمرین را خودتان انجام دهید.