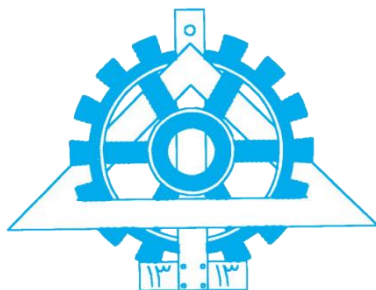


به نام خدا

دانشکده مهندسی برق و کامپیوتر

درس یادگیری عمیق

تمرین سری اول



در این تمرین هدف پیاده‌سازی یک شبکه عصبی با یک یا دو لایه مخفی به منظور طبقه‌بندی و آموزش آن می‌باشد. مجموعه داده‌ای که برای آموزش مورد استفاده قرار می‌گیرد دیتاست <sup>1</sup>MNIST می‌باشد. توجه داشته باشید که برای انجام این تمرین مجاز به استفاده از هیچ کد آماده و ابزارهای مرتبط با شبکه عصبی و یادگیری عمیق (نظیر pytorch و tensorflow) نمی‌باشید و باید تمامی مراحل بروزرسانی وزنها و پارامترهای شبکه را خودتان پیاده‌سازی نمایید.

۱. در ابتدا باید یک شبکه عصبی با یک لایه مخفی ۲۰۰ نورونی پیاده‌سازی کنید که activation function آن relu و loss آن hinge loss (svm loss) می‌باشد و batch size را برابر ۱۲۸ در نظر بگیرید. دقت داشته باشید که در این مرحله باید پیاده‌سازی شما مستقل از تعداد لایه‌ها و تعداد نورون‌های آن‌ها باشد.

۲. در این بخش یک Gaussian-RBF-Network با ساختار سوال قبل پیاده‌سازی نمایید و تابع loss آن را مجموع مربعات خطا (MSE) در نظر بگیرید.

۳. شبکه‌های مطرح شده در قسمت‌های ۱ و ۲ را با استفاده از مقداردهی مناسب به وزنها و با استفاده از روش Stochastic Gradient Descent (SGD) آموزش دهید. مقدار loss و accuracy را در طول یادگیری رسم نمایید و این نتایج این دو شبکه را با هم مقایسه نمایید.

۴. تابع فعال‌سازی را در شبکه اول از relu به softplus و سپس به leaky relu با  $\alpha=0.2$  تغییر دهید و تغییر روند یادگیری را توضیح دهید.

۵. در شبکه اول با تغییر تعداد لایه‌ها و یا تعداد نورون‌ها، باعث over fit شدن شبکه شوید. سپس با افزودن L2 Regularization تا حدی جلوی این اتفاق را بگیرید. (دقت کنید در حل این سوال نیاز به سعی و خطا می‌باشد) این کار را برای شبکه دوم نیز انجام دهید و نتایج را مقایسه کنید.

۶. در این مرحله به تاثیر weight initialization می‌پردازیم. در یک نمودار دقت نهایی شبکه اول (بعد از یک epoch) را بسته به واریانس وزن‌های اولیه (از ۰ تا ۱) رسم کنید. سپس، بهترین واریانس را انتخاب کنید و یک شبکه تک لایه را برای ۲ epoch آموزش دهید. حال یک لایه دیگر به همین شبکه اضافه کنید (یعنی وزنه‌های ورودی به لایه پنهان را همان که به دست آمده نگاه دارید) [به این سبک از layer wise initialization می‌گویند] و شبکه حاصل را برای یک epoch دیگر آموزش دهید حال نتیجه را با یک شبکه ۲ لایه که از ابتدا آموزش می‌بیند و وزن‌های آن همه از همان واریانس بهینه آمده و برای ۳ epoch آموزش می‌بیند مقایسه کنید.

۷. در این بخش پیش‌پردازش داده را مورد بررسی قرار خواهیم داد. برای این منظور ورودی را به چهار روش متفاوت به شبکه اول وارد می‌کنیم. در روش اول ورودی به شکل خام اعمال خواهد شد. در روش دوم با اعمال PCA بعد داده را به ۱۲۸ تبدیل کرده و سپس به شبکه وارد می‌کنیم. در روش سوم با استفاده از PCA بعد را به ۶۴ رسانده و سپس به شبکه می‌دهیم و در روش آخر میانگین داده را صفر و واریانس آن را یک می‌سازیم. دقت نهایی و سرعت همگرایی این روش‌ها را با هم مقایسه کنید و تفاوت‌ها را توضیح دهید.

۸. سوال قبل را برای شبکه دوم نیز انجام داده و نتایج حاصل را با بخش قبل مقایسه کنید.

۹. حال به optimizer خود momentum اضافه کنید و در چهار نمودار ۳ بعدی تاثیر مقدار batch size و learning rate را با مومنتم‌های مختلف (0.99, 0.9, 0.5, 0) بر دقت نهایی برای شبکه اول را نشان داده و آن را توضیح دهید.

۱۰. سوال قبل را برای شبکه دوم نیز انجام داده و نتایج را با نتایج سوال قبل مقایسه کنید.

<sup>1</sup> <http://yann.lecun.com/exdb/mnist/>

نکات:

- توجه کنید که نیمی از نمره تمرین مربوط به گزارش می‌باشد. لازم به ذکر است که رعایت اصول نگارشی حائز اهمیت می‌باشد.
- گزارش تمرین را حتما به صورت PDF ارسال نمایید.
- کدهای تمرین را به همراه گزارش تمرین در سایت درس آپلود نمایید.
- نحوه نام‌گذاری تمرین براساس studentnumber\_homeworknumber.pdf باشد.
- زبان پیاده‌سازی این تمرین python می‌باشد.
- هر گونه پرسش پیرامون تمرین را با ایمیل‌های [sepehr.sameni@gmail.com](mailto:sepehr.sameni@gmail.com) و [r.ebrahimnakhli@gmail.com](mailto:r.ebrahimnakhli@gmail.com) و [esmaeilfarahng@gmail.com](mailto:esmaeilfarahng@gmail.com) مکاتبه فرمایید.