

این گزارش به منظور ارائه نتایج توسعه و ارزیابی یک مدل پرسپترون چند لایه‌ای (MLP) برای طبقه‌بندی ارقام دست‌نویس مجموعه داده MNIST تهیه شده است. در این گزارش، تغییرات اعمال شده بر روند آموزش، بهینه‌سازی‌های انجام شده و محدودیت‌های عملکردی مدل، به ویژه در مواجهه با داده‌های جدید و دستی (Unseen Data)، به صورت صریح و صادقانه بیان شده است. [User Input]

گزارش دوره یادگیری عمیق: طبقه‌بندی ارقام MNIST با استفاده از MLP

۱. مقدمه و متدولوژی

هدف این پروژه، پیاده‌سازی و تنظیم یک مدل پایه پرسپترون چند لایه‌ای (MLP) برای تشخیص ۱۰ کلاس ارقام دست‌نویس MNIST بود. تمامی مراحل از جمله بارگذاری داده‌ها، نرمال‌سازی و طراحی معماری مدل با استفاده از کتابخانه‌های Keras و TensorFlow انجام شد.

۱.۱. آماده‌سازی داده‌ها

داده‌های آموزشی و آزمایشی (`x_train`, `y_train`, `x_test`, `y_test`) با استفاده از `keras.datasets.mnist.load_data()` بارگذاری شدند.

۱. نرمال‌سازی: (Normalization) مقادیر پیکسل‌ها در بازه `[0, 1]` قرار گرفتند (`x_train =`

`x_train.astype(float) / 255.`).

۲. یک‌بندی: (One-Hot Encoding) برای تطبیق با معماری طبقه‌بندی چندکلاسه، برچسب‌های هدف (`y_train`,

`y_test`) به ماتریس‌های دودویی (categorical) تبدیل شدند (۱۰ کلاس).

۱.۲. معماری پایه مدل

معماری مدل‌های مورد آزمایش، ساختار ترتیبی (Sequential) داشت و شامل دو لایه (Dense کاملاً متصل) بود:

- لایه ورودی که سپس مسطح (Flatten) می‌شود.
- لایه Dense پنهان با ۵۱۲ واحد و تابع فعال‌سازی `relu`.
- لایه Dense خروجی با ۱۰ واحد (یا ۱ واحد در آزمایش رگرسیون اولیه).

۲. تنظیمات تکراری و بهینه‌سازی مدل

برای یافتن بهترین ترکیب توابع زیان و فعال سازی، آزمایش های مختلفی انجام شد. مشاهده اصلی این بود که مدل ها حتی با کاهش Epoch ها به مقادیر پایین تر از مقدار اصلی (که ۱۰ بود [User Input])، سرعت همگرایی بالایی داشتند و دقت به سرعت تثبیت می شد. [User Input]

شکل ۱: نمونه ای از تنظیمات کاهش یافته Epoch در فرآیند آموزش (تصویر کد پایتون نمایش داده شده که متغیر epochs برابر با ۴ تنظیم شده است).

جدول ۱: خلاصه ای از تکرارهای مدل و نتایج عملکردی

شرح	دقت آزمون	تعداد Epoch	تابع زیان (Loss)	تابع فعال سازی خروجی	تکرار
عملکرد ضعیف به دلیل استفاده از رگرسیون برای طبقه بندی	۵۹.۹۹ %	۵	Mse	۱ واحد خروجی (None) اولیه	(۱) رگرسیون
بهبود قابل توجه پس از تغییر به ۱۰ واحد خروجی	۹۸.۰۹ %	۵	Mse	(None خطی) خام	(۲) طبقه بندی
کاهش Epoch ها	۹۷.۸۱ %	۴	Mse	sigmoid	۳
تست softmax با mse	۹۷.۷۶ %	۴	Mse	softmax	۴
بهترین ترکیب برای طبقه بندی	۹۷.۸۲ %	۴		softmax categorical_crossentropy	(۵) بهینه

۲.۱. مشاهده در مورد Epoch ها

همانطور که در جدول ۱ مشاهده می شود، کاهش تعداد Epoch ها به ۴ یا ۵، تأثیر کمی بر کاهش دقت نهایی داشت و مدل حتی در این تعداد کم Epoch ها، عملکردی معادل ۹۷ تا ۹۸ درصد از خود نشان داد. [User Input, 5, 10, 53, 69] این امر نشان دهنده سادگی نسبی مسئله (MNIST) برای معماری انتخاب شده و کارایی بالای مدل در فازهای اولیه آموزش است.

۲.۲. عملکرد نهایی در مجموعه داده آزمون MNIST

پیکربندی نهایی (تکرار ۵) که از تابع زیان categorical_crossentropy به همراه تابع فعال سازی softmax استفاده می کرد، به دقت ۹۷.۸۲٪ دست یافت.

شکل ۲: ماتریس اغتشاش (Confusion Matrix) برای بهترین مدل (دقت ۹۷.۸۲٪) (تصویر ماتریس اغتشاش نمایش داده شده که یک قطر پررنگ (نشان دهنده طبقه بندی صحیح) دارد و خطاهای کمی در خارج از قطر دیده می شود). (این ماتریس نشان دهنده یک عملکرد بسیار قوی در مجموعه داده ای است که مدل با آن آموزش دیده و آزموده شده است.

۳. ارزیابی مدل بر روی داده‌های دست‌ساز و جدید (Unseen Data)

مدل نهایی با دقت بالا (۹۷.۸۲٪) ذخیره‌سازی و سریالایز (serialize) شد (model.json) و (model.keras) سپس با استفاده از یک اسکریپت پایتون جدید، بارگذاری مجدد گردید.

هدف از این مرحله، آزمایش قابلیت تعمیم (Generalization) مدل بر روی داده‌های کاملاً جدید و خارج از توزیع MNIST بود که به صورت دستی طراحی و تولید شده بودند (ارقام ۰ تا ۹).

۳.۱. مراحل پیش‌پردازش برای داده‌های دست‌ساز

تصاویر دست‌ساز ۱۰ گانه قبل از ورود به مدل، دقیقاً مانند داده‌های MNIST پیش‌پردازش شدند:

۱. تبدیل به مقیاس خاکستری (Grayscale).

۲. تغییر اندازه به ۲۸×۲۸ پیکسل.

۳. نرمال‌سازی به [۰, ۱].

۴. معکوس‌سازی رنگ (img - 1)، چرا که ارقام MNIST به صورت سفید روی پس‌زمینه سیاه هستند.

شکل ۳: نمونه‌هایی از ارقام دست‌ساز بارگذاری شده برای تست تعمیم مدل (تصاویری با ابعاد ۲۸×۲۸ پیکسل که ارقام ۰ تا ۹ را نشان می‌دهند و در کد بارگذاری شده‌اند).



۳.۲. نتایج ارزیابی و شکست در تعمیم (Generalization Failure)

نتایج پیش‌بینی کلاس‌ها بر روی این داده‌های جدید نشان داد که مدل در طبقه‌بندی صحیح ارقام دست‌ساز عملکرد ضعیفی دارد و نتوانست تعمیم یابد.

نکته محوری: این مشکل تأیید می‌کند که مدل بیش‌برازش (Overfitting) داشته یا حداقل کاملاً برای داده‌های خارج از توزیع MNIST تنظیم نشده است. [User Input]

این شکست در تعمیم نشان‌دهنده این است که مدل، به جای یادگیری ویژگی‌های کلی و انتزاعی ارقام، ویژگی‌های بسیار خاص مربوط به سبک، اندازه خطوط، و نرمال‌سازی‌های داخلی مجموعه داده آموزشی (MNIST) را حفظ کرده است. هرگونه تغییر در استایل نگارشی یا کیفیت تصویر (مانند تصاویر دست‌ساز) منجر به از دست رفتن دقت می‌شود.

۴. نتیجه‌گیری و پیشنهادات

مدل MLP توسعه یافته، هدف اولیه خود را با کسب دقت ۹۷.۸۲٪ بر روی مجموعه داده آزمون MNIST برآورده کرد. همچنین، فرآیند آموزش سریع بود و مدل به سرعت در ۴ یا ۵ Epoch به همگرایی می‌رسید. [User Input]

اعتراف به نقص: با وجود این دقت بالا، **ارزیابی صادقانه** بر روی داده‌های دست‌ساز و جدید، یک ضعف جدی در قابلیت تعمیم مدل را آشکار کرد. [User Input] نتیجه حاکی از آن است که یا مدل دچار **بیش‌برازش** شده و یا نیازمند تنظیمات دقیق‌تری است تا بتواند با نوسانات (variance) دنیای واقعی کنار بیاید. [User Input]

پیشنهاد برای بهبود: برای رفع مشکل بیش‌برازش و افزایش قابلیت تعمیم، اجرای تکنیک‌های منظم‌سازی (Regularization) مانند Dropout در لایه‌های Dense یا انتقال به یک معماری شبکه‌های عصبی پیچیده‌تر و قوی‌تر مانند شبکه‌های عصبی کانولوشنی (CNN) ضروری است.