



# بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## گزارش کار تمرین ۴-۳ - یادگیری عمیق

عنوان تمرین :

طراحی و پیاده‌سازی شبکه عصبی تمام متصل (FCN)

برای طبقه‌بندی تصاویر CIFAR-۱۰

دانشجو : محمد داود وهاب رجائی

شماره دانشجویی : ۴۰۴۱۴۱۹۰۴۱

استاد محترم : دکتر کیوان راد

حل تمرین : دکتر گلی زاده

پاییز ۱۴۰۴

## فهرست مطالب

چکیده	۱
مقدمه و بیان مسئله	۱
روش‌شناسی (Methodology)	۲
۱. آماده‌سازی داده‌ها	۲
۲. معماری مدل پیشنهادی	۳
۳. پیکربندی آموزش	۳
نتایج تجربی (Experimental Results)	۴
۱. روند آموزش و اعتبارسنجی	۴
۲. ارزیابی روی داده‌های تست	۴
۳. تحلیل جزئیات کلاس‌ها	۵
بحث و تحلیل (Discussion)	۶
۱. تحلیل ماتریس درهم‌ریختگی	۶
۲. محدودیت‌های معماری FCN	۶
نتیجه‌گیری و کارهای آتی	۷

## چکیده

در این پروژه، یک شبکه عصبی تمام متصل (Fully Connected Neural Network) برای طبقه‌بندی تصاویر رنگی مجموعه داده CIFAR-۱۰ طراحی و آموزش داده شد. هدف اصلی، بررسی کارایی معماری‌های MLP بر روی داده‌های تصویری و تأثیر تکنیک‌های تنظیم (Regularization) بود. مدل نهایی با استفاده از لایه‌های Batch Normalization، Dropout و تابع فعال‌ساز LeakyReLU پیاده‌سازی شد و پس از ۲۵ دور آموزش (Epoch)، به دقت نهایی ۵۵.۵۵٪ روی داده‌های تست دست یافت. نتایج نشان داد که اگرچه شبکه‌های تمام متصل به دلیل نادیده گرفتن اطلاعات مکانی تصاویر دارای محدودیت هستند، اما با تنظیم دقیق هایپرپارامترها می‌توانند عملکردی بسیار بالاتر از حدس تصادفی (۱۰٪) ارائه دهند.

## مقدمه و بیان مسئله

طبقه‌بندی تصاویر یکی از مسائل بنیادین در بینایی ماشین است. در حالی که شبکه‌های کانولوشنی (CNN) استاندارد طلایی برای این وظیفه محسوب می‌شوند، درک عملکرد شبکه‌های تمام متصل (FCN) به عنوان یک پایه (Baseline) ضروری است. در این تمرین، چالش اصلی تبدیل داده‌های تصویری دوبعدی به بردارهای یک‌بعدی و آموزش یک شبکه عمیق برای یادگیری الگوهای بصری بدون استفاده از فیلترهای کانولوشنی است. ما از دیتاست CIFAR-۱۰ شامل ۱۰ کلاس (هواپیما، خودرو، پرنده، گربه، گوزن، سگ، قورباغه، اسب، کشتی و کامیون) استفاده کردیم که شامل ۶۰,۰۰۰ تصویر رنگی ۳۲×۳۲ پیکسل است.

## روش‌شناسی (Methodology)

### ۱. آماده‌سازی داده‌ها

- پیش‌پردازش (Preprocessing): تمامی تصاویر ابتدا به تانسورهای PyTorch تبدیل شدند. جهت کمک به همگرایی سریع‌تر مدل و پایداری گرادیان‌ها، نرمال‌سازی (Normalization) با میانگین ۰.۵ و انحراف معیار ۰.۵ روی کانال‌های رنگی اعمال شد تا مقادیر پیکسل‌ها در بازه  $[-1, 1]$  قرار گیرند.
- تقسیم‌بندی داده‌ها:
  - داده‌های آموزش (Training Set): ۴۲,۵۰۰ تصویر (۸۵٪)
  - داده‌های اعتبارسنجی (Validation Set): ۷,۵۰۰ تصویر (۱۵٪) جهت تنظیم هایپرپارامترها.
  - داده‌های تست (Test Set): ۱۰,۰۰۰ تصویر جهت ارزیابی نهایی.
- بارگذاری: از DataLoader با اندازه دسته (Batch Size) برابر با ۱۲۸ استفاده شد.

Sample Images from Each Class



شکل ۱: نمونه تصاویر مجموعه داده CIFAR-۱۰ پس از نرمال‌سازی

## ۲. معماری مدل پیشنهادی

مدل طراحی شده یک شبکه چندلایه پرسپترون (MLP) با مشخصات زیر است :

۱. لایه ورودی: تصویر  $32 \times 32 \times 3$  باز شده (Flatten) و به یک بردار با ابعاد ۳۰۷۲ تبدیل می‌شود.

۲. لایه مخفی اول: لایه خطی با ۵۱۲ نورون + Batch Normalization + LeakyReLU (۰,۱) + Dropout (۰,۵).

۳. لایه مخفی دوم: لایه خطی با ۲۵۶ نورون + Batch Normalization + LeakyReLU (۰,۱) + Dropout (۰,۵)

۴. لایه خروجی: لایه خطی با ۱۰ نورون (متناظر با ۱۰ کلاس).

- تعداد کل پارامترهای قابل آموزش: ۱,۷۰۸,۸۱۰ پارامتر.
- دلیل انتخاب: استفاده از LeakyReLU برای جلوگیری از مشکل "مرگ نورون‌ها" و استفاده از Dropout (۰,۵) برای مقابله با بیش‌برازش شدید در شبکه‌های تمام متصل.

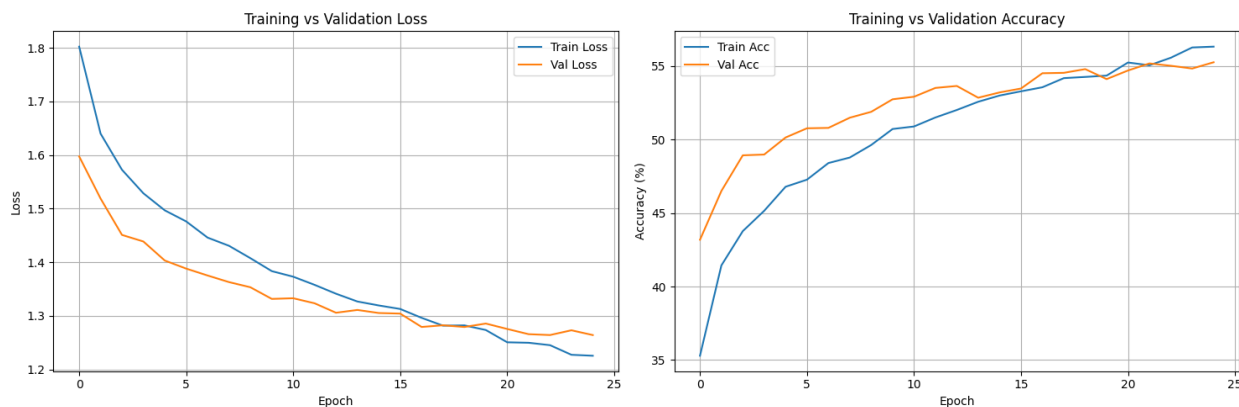
## ۳. پیکربندی آموزش

- تابع هزینه: CrossEntropyLoss
- بهینه‌ساز: Adam: با نرخ یادگیری اولیه ۰.۰۰۱
- زمان‌بندی (Scheduler): استفاده از ReduceLROnPlateau برای کاهش نرخ یادگیری در صورت عدم بهبود دقت اعتبارسنجی.
- تعداد اپک: ۲۵ دور کامل.

## نتایج تجربی (Experimental Results)

### ۱. روند آموزش و اعتبارسنجی

فرآیند آموزش مدل نشان‌دهنده همگرایی مناسب بود. همانطور که در نمودارهای زیر مشاهده می‌شود، نرخ خطا (Loss) روی داده‌های اعتبارسنجی همگام با داده‌های آموزشی کاهش یافت که نشان‌دهنده عملکرد مؤثر Dropout در جلوگیری از Overfitting است. مدل در اپک ۲۵ به بهترین دقت اعتبارسنجی ۵۵.۲۷٪ دست یافت و مدل در این نقطه ذخیره شد.



شکل ۲: منحنی‌های یادگیری؛ روند تغییرات Loss و Accuracy در طول ۲۵ اپک

### ۲. ارزیابی روی داده‌های تست

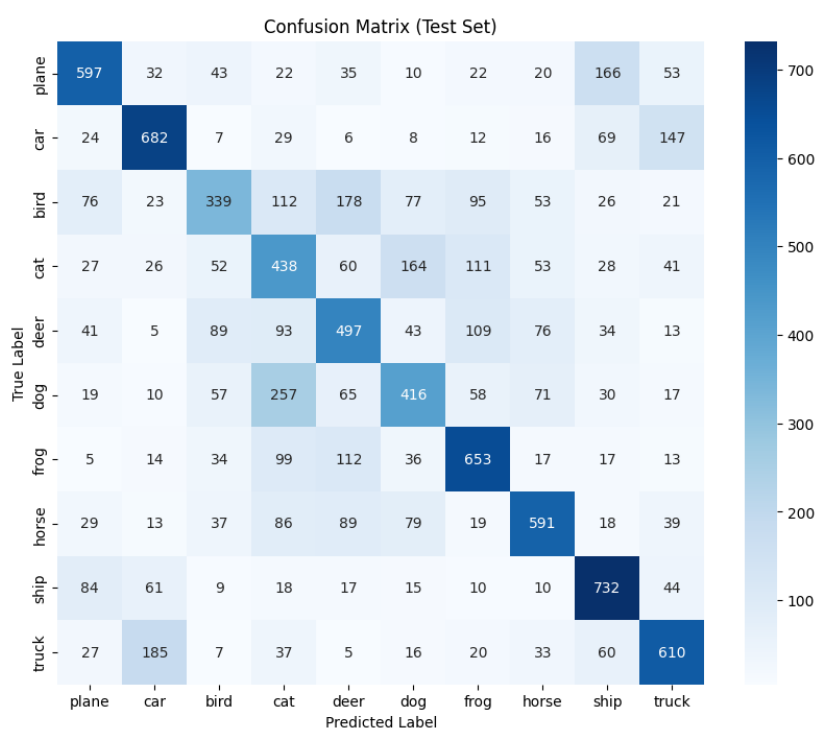
پس از پایان آموزش، مدل نهایی روی ۱۰,۰۰۰ تصویر دیده نشده تست ارزیابی شد:

- دقت نهایی (Test Accuracy): ۵۵.۵۵٪
- عملکرد نسبت به پایه: با توجه به وجود ۱۰ کلاس، حدس تصادفی دقتی برابر ۱۰٪ دارد. مدل ما ۴۵٪ بهبود نسبت به حالت تصادفی داشته است.

### ۳. تحلیل جزئیات کلاس‌ها

با بررسی گزارش طبقه‌بندی (Classification Report)، نتایج زیر حاصل شد :

- بهترین عملکرد : کلاس‌های "کشتی (Ship)" با  $F_1$ -Score برابر ۰.۶۷ و "خودرو (Car)" با ۰.۶۷.
- ضعیف‌ترین عملکرد : کلاس‌های "گربه (Cat)" با  $F_1$ -Score برابر ۰.۴۰ و "پرنده (Bird)" با ۰.۴۱.



شکل ۳ : ماتریس درهم‌ریختگی نتایج تست

## بحث و تحلیل (Discussion)

### ۱. تحلیل ماتریس درهم‌ریختگی

با دقت در ماتریس درهم‌ریختگی (شکل ۳) مشاهده می‌شود که بیشترین اشتباهات مدل بین کلاس‌های زیر رخ داده است:

۱. گربه و سگ : مدل اغلب این دو حیوان را با هم اشتباه می‌گیرد. دلیل این امر شباهت ساختاری (چهارپا بودن، داشتن گوش و دم) و عدم توانایی شبکه FCN در استخراج ویژگی‌های ظریف محلی است.

۲. پرنده و هواپیما : اشتباه بین این دو کلاس اغلب به دلیل رنگ پس‌زمینه (آسمان آبی) رخ می‌دهد. از آنجا که شبکه تمام متصل به شدت به مقادیر پیکسل‌ها و رنگ وابسته است، بافت کلی تصویر را ملاک قرار می‌دهد.

### ۲. محدودیت‌های معماری FCN

دستیابی به دقت ۵۵٪ برای یک شبکه تمام متصل روی دیتاست پیچیده‌ای مثل CIFAR-۱۰ نتیجه‌ای قابل قبول است.

دلیل عدم دستیابی به دقت‌های بالاتر (مانند ۹۰٪ در CNN ها)، از بین رفتن اطلاعات مکانی (Spatial Information) هنگام مسطح کردن (Flatten) تصویر است. شبکه FCN رابطه همسایگی بین پیکسل‌ها را درک نمی‌کند و نمی‌تواند ویژگی‌های مستقل از مکان (Translation Invariance) را یاد بگیرد.



## نتیجه‌گیری و کارهای آتی

در این تمرین، یک خط لوله کامل یادگیری عمیق با استفاده از شبکه تمام متصل پیاده‌سازی شد. استفاده از تکنیک‌های Batch Normalization و Dropout پایداری مدل را تضمین کرد و دقت ۵۵.۵۵٪ حاصل شد. نتایج نشان داد که مدل در تشخیص وسایل نقلیه موفق‌تر از حیوانات عمل می‌کند و محدودیت اصلی، نوع معماری شبکه است.

پیشنهادهای برای کارهای آتی :

استفاده از شبکه‌های کانولوشنی (CNN) برای حفظ اطلاعات مکانی و استخراج ویژگی‌های بصری عمیق‌تر. استفاده از Data Augmentation (مانند چرخش، برش و تغییر رنگ تصادفی) برای افزایش تنوع داده‌های آموزشی و بهبود تعمیم‌پذیری مدل.

با تشکر از نگاه پر مهر شما