



مهلت تحویل : ۹ تیرماه

پروژه‌ی نهایی درس شبکه‌های عصبی

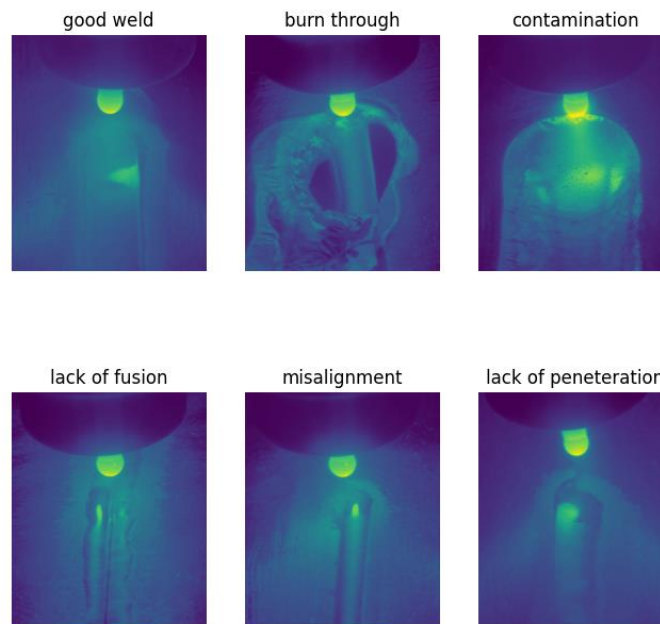
بخش ۱: پیاده‌سازی یک شبکه‌ی عصبی کانولوشنی

- رعایت تمامی قوانین موجود در آداب نامه و پروژه‌های پیشین الزامی می‌باشد.
- از آنجایی که مولفه‌های آزاد بسیاری در پیاده‌سازی و پاسخ به پروژه‌ها وجود دارد بنابراین هرگونه شباهت غیر متعارف در پاسخ‌نامه‌ها تقلب محسوب شده و نمره‌ی طرفین صفر لحاظ خواهد شد.
- توصیه‌ی اکید می‌شود طبق توضیحات دستور کار از گوگل کولب استفاده فرمایید.

**صورت مسئله :** در درس شبکه‌های عصبی و ارائه‌های انجام شده با مبانی و ساختار شبکه‌های عصبی مصنوعی و شبکه‌های عصبی کانولوشنی<sup>۱</sup> آشنا شدیم. در این پروژه قصد داریم به صورت عملی یک شبکه‌ی عصبی کانولوشنی را برای دسته‌بندی تصاویر فرآیند جوشکاری با هدف پایش سلامت جوش پیاده کنیم. مقاله‌ی مرتبط با این پیاده‌سازی همراه دستور کار پروژه قرار داده شده است.

در این پیاده‌سازی از مجموعه داده تصاویر فرآیند جوشکاری [TIG آلیاژ آلومینیوم ۵۰۸۳ \(A15083\)](#) موجود در سایت کگل استفاده خواهد شد. مجموعه داده حاضر حاوی ۶ کلاس می‌باشد. نمونه‌ای از تصاویر این مجموعه داده در شکل ۱ آورده شده است.

<sup>1</sup> Convolutional Neural Network (CNN)



شکل ۱ - نمونه‌ای از داده‌های مجموعه‌ی داده A15083

برای این دسته‌بندی، یک شبکه‌ی عصبی کانولوشنی طراحی کنید که بتواند تصاویر را به درستی در دسته‌ها قرار دهد. این شبکه نباید از شبکه‌های معروف موجود در ادبیات این حوزه باشد. لایه‌ها را تک به تک پشت سرهم قرار داده و شبکه را تشکیل دهید. محدودیتی که دارید آن است که تعداد لایه‌های شبکه از ۳۰ لایه بیشتر نباشد. استفاده از مازول‌های دارای skip connection و سایر عملگرهای کانولوشنی (Normal, Depthwise, Pointwise) مجاز است.

در این پیاده‌سازی مجاز هستید تا از پکیج‌های آماده و فریمورک‌های مشهور هوش مصنوعی مانند پایتورچ و scikit learn در کنار پکیج‌های نامپای، Pandas، Matplotlib، Pillow، و OpenCV استفاده نمایید. همچنین، برای نمایش Progress Bar می‌توانید از پکیج tqdm استفاده کنید.

## پیکربندی یک شبکه‌ی عصبی

در منابع هوش مصنوعی و یادگیری عمیق ساختار آموزش شبکه‌های عصبی به چهار قسمت تقسیم می‌شود:

- **تشکیل دیتاست و دیتالودر**: برای فراخوانی داده‌ها و ساخت دیتاست و دیتالودر<sup>۱</sup> مورد نیاز در پایتورچ نیاز است توابع کاستوم شده خودتان را کدزنی کنید. وظیفه‌ی دیتالودر تغذیه‌ی حلقه‌ی آموزش حین

<sup>۱</sup> DataLoader

آموزش شبکه است که با توجه به متدهای `__len__` و `__getitem__` که شما شخصا پیاده کرده‌اید داده‌های را گردآوری کرده و به صورت تکرار شونده به حلقه آموزش می‌رساند. برای یادگیری بیشتر در این مورد می‌توانید به [این آموزش](#) مراجعه فرمایید. در صورت نیاز می‌توانید از `transform`ها برای پردازش روی تصاویر استفاده نمایید.

- **معماری شبکه:** در این بخش بدون نوشتن هیچ لایه‌ی جدیدی و تنها با استفاده از لایه‌های موجود در پایتورچ (با ارث گرفتن از `nn.Module` در پایتورچ) باید یک شبکه‌ی عصبی کانولوشنی بسازید تا فرآیند آموزش بر اساس این ساختار شبکه صورت پذیرد و پارامترهای شبکه به‌روز شوند. برای یادگیری بیشتر در این مورد می‌توانید به [این آموزش](#) مراجعه فرمایید.

- **ساخت Trainer:** در این بخش به انتخاب بهینه‌ساز، نوع متریک، تابع هزینه و چگونگی انتخاب و به‌روزرسانی نرخ یادگیری شبکه می‌پردازیم. تا با استفاده از آن‌ها در حلقه‌ی آموزش، شبکه را به صورت هدفمند آموزش دهیم.

- **ساخت حلقه‌ی آموزش:** مرحله‌ی نهایی در آموزش شبکه‌های عصبی مصنوعی، حلقه‌ی آموزش است. در این حلقه با مشخص کردن `epoch`، `step` و تکرار روی دیتالودر ساخته شده عمل بک‌پراپگیشن<sup>۱</sup> صورت گرفته و وزن‌ها به‌روز می‌شوند. در همین حین شبکه تست می‌شود و عملکرد شبکه حین آموزش گزارش می‌شود. نحوه‌ی گزارش چگونگی آموزش در حلقه‌ی آموزش بسیار مهم می‌باشد.

پس از آموزش شبکه نیاز است عملکرد شبکه را روی داده‌های تست ارزیابی کنید. نمره بر اساس دقت<sup>۲</sup> شبکه، سرعت عملکرد شبکه (`fps`<sup>۳</sup>)، نوع و تعداد لایه‌ها خواهد بود. بدین منظور کدهای خود را با استفاده از `T4-gpu` موجود در گوگل کولب آموزش و تست کنید. چگونه `fps` را محاسبه می‌کنید؟ با چه تعداد تصویر؟

در پایان نیاز است ساختار شبکه را رسم نمایید. (استفاده از هر پکیج یا نرم‌افزاری برای رسم ساختار شبکه‌ی شما آزاد می‌باشد).

---

<sup>۱</sup> Backpropagation

<sup>۲</sup> Accuracy

<sup>۳</sup> Frame per second

## پرسش‌ها

- تصاویر در این مجموعه داده دارای سطوح خاکستری می‌باشد، ابعاد تصاویر به چه صورت است؟ آیا تغییری در ابعاد تصاویر نیاز است؟ در مورد تعداد کانال‌ها چگونه؟ آیا تعداد کانال‌های تصویر خاکستری به ۳ یا ۱ امکان‌پذیر است؟ برای شبکه‌ی خودتان با تغییر کانال تصویر به ۳ و ۱، تعداد پارامترهای شبکه چه تغییری می‌کند؟ تعداد feature map چگونه؟
- از چه لایه‌هایی در این شبکه استفاده کرده‌اید؟ کاربرد هر یک از این لایه‌ها چیست؟
- معمولاً مقداردهی اولیه برای وزن‌ها صورت می‌گیرد. در پیاده‌سازی شما در کدام مرحله این اتفاق افتاده است؟
- تعداد کل پارامترهای شبکه چند تا است؟ چه لایه‌هایی بیشترین تعداد پارامتر را دارند؟
- از چه تابع هزینه‌ای برای برازش استفاده کرده‌اید؟ آیا با وجود تابع هزینه استفاده از معیارهای ارزیابی ضروری است؟ تابع هزینه با مقدار دقت چه تفاوتی دارد؟
- از چه تابع بهینه‌سازی استفاده کرده‌اید؟
- برای یکی از لایه‌های ابتدایی شبکه و همچنین یکی از لایه‌های پایانی شبکه مقادیر کانولوشن را رسم ([Visualize](#)) کنید. آیا در این شکل‌ها مفاهیم به خصوصی مشاهده می‌کنید؟
- برای کلاس‌های این مجموعه داده ماتریس درهم‌ریختگی<sup>۱</sup> را رسم نمایید. چه تحلیلی از عملکرد شبکه دارید؟ آیا مجموعه داده برای تمامی کلاس‌ها خوب عمل کرده است؟ چرا؟ چه راه حلی وجود دارد؟

## بخش دوم: بک‌پراپگیشن در شبکه‌های کانولوشنی

آموزش شبکه‌های عصبی مصنوعی با استفاده از روش گرادیان نزولی و بک‌پراپگیشن صورت می‌گیرد. در این بخش قصد داریم به نحوه‌ی پردازش این بک‌پراپگیشن در یک شبکه‌ی ساده بپردازیم. از این رو به پرسش‌های زیر پاسخ دهید.

- در فرآیند بک‌پراپگیشن منظور از Chain Rule چیست؟ برای یک نرون یا شبکه‌ی بسیار ساده آن را بیان کنید. برای یادگیری چگونگی اعمال بک‌پراپگیشن به عملگر کانولوشن می‌توانید به این [لینک](#) و این [لینک](#) مراجعه کنید.

---

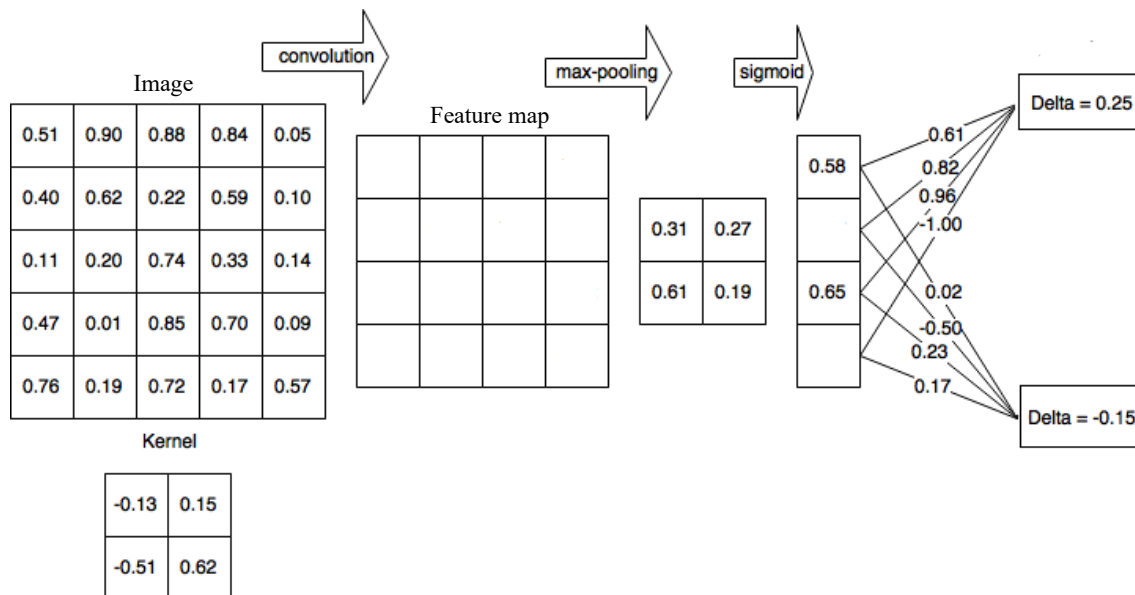
<sup>1</sup> Confusion Matrix

- شبکه‌ی زیر را در نظر بگیرید. در این شبکه لایه‌های مختلفی اعم از کانولوشن، تماما متصل، Maxpooling ، Sigmoid ، flatten و تمام متصل وجود دارد. تحقیق کنید که برای هر کدام از این لایه‌ها بک‌پراپگیشن به چه صورت انجام می‌پذیرد. برای سادگی سهم خطا از خروجی‌ها را با دلتا نشان داده‌ایم.

$$Delta = \frac{\partial E}{\partial x_{o1}}$$

- ابتدا با توجه به جزییاتی که در ادامه می‌آید جاهای خالی را در شکل ۲ پر کنید. کانولوشن بین Image و Kernel با  $stride = (1,1)$  و بدون padding. Maxpooling با  $stride = (2,2)$ .
- با استفاده از کدزنی (فقط پایتون و نامپای و ابزار نمایش گر) میزان تغییرات هر پارامتر شبکه را محاسبه کنید. در پایان با استفاده از نرخ یادگیری  $0.5$  وزن‌ها را به‌روز کرده و در گزارش خود بیاورید. توجه فرمایید کد شما درباره با یکسری وزن‌های دیگر چک خواهند شد و باید کد شما این قابلیت را داشته باشد تا با کرنل دیگری هم جواب درستی دهد.

$$w^{(t+1)} = w^{(t)} - \eta \frac{\partial E}{\partial w_{mn}}$$



شکل ۲ - نمای شبکه برای محاسبه‌ی به روزرسانی‌های وزن‌ها

- در کتابخانه‌های مطرح هوش مصنوعی برای عملگرهایی مانند کانولوشن از ترفند [unfold](#) و [fold](#) استفاده می‌شود. در مورد این ترفند توضیح دهید. سپس آن را برای مثال گفته شده پیاده کنید.