**بخش اول: دسته بندی داده های CIFAR10( 10 کلاسه)**

- بخش اول به صورت from_scrach هست.
- فایل Template مربوط به این بخش است.
- دیتاست CIFAR10 را خود کتابخونه ها دارند.
- CIFAR10 ده کلاسه هست ( هواپیما ، کشتی و ...)
- استفاده از scikit learn جهت اندازه گیری دقت برای بخش اول اشکالی ندارد.
- برای روابط back praction باید روابط مشتق پیاده سازی شود
- لایه ی آخر یعنی لایه ی خروجی باید به تعداد کلاس هایمان باشد پس == 10
- در بخش اول پروژه به دنبال پیاده سازی شکبه هستیم و ساختار معماری شکبه مشخص است و پارامتر ها مشخص است و دنبال train کردن آن شبکه هستیم
- (حساب کردن خطا) چون پروژه ما classification است نباید از loss function استفاده کرد و باید از -Entropy Cross Categorical
- نوعی از loss functionها MSE هست.
- فیچر اکسترکشن برابر Resnet3 است.
- ما میخواهیم تابه خطا را بهینه کنیم. چه MSE چه Entropy-Cross Categorical
- F1:

Computes F-1 Score.

Why TensorFlow

Inherits From: `FBetaScore`

```
tfa.metrics.F1Score(
    num_classes: tfa.types.FloatTensorLike,
    average: str = None,
    threshold: Optional[FloatTensorLike] = None,
    name: str = 'f1_score',
    dtype: tfa.types.AcceptableDTypes = None
)
```

It is the harmonic mean of precision and recall. Output range is `[0, 1]`. Works for both multi-class and multi-label classification.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

https://www.tensorflow.org/addons/api_docs/python/tfa/metrics/F1Score#:~:text=It%20is%20the%20harmonic%20mean,class%20and%20multi%2Dlabel%20classification.

**بخش دوم : دسته بندی داده های CIFAR10**

- اما در بخش دوم پروژه ، به دنبال هایپر پارمتر ها شکبه هستیم (فقط هایپرپارامترها بهینه سازی شوند). یعنی ساختار شبکه را میخواهیم بهینه سازی کنیم که با استفاده از الگوریتم تکاملی انجام می شود(Neuro-evolution)
- پارامتر ها ← تکاملی (back praction) آپدیت می شود
- ابرپارامتر ها(تعداد لایه ها تعداد نورن ها ، اکتیویشن فانکشن )← با استفاده از Neuro-evolution بدست می آید
- پارامتر ها در یک شبکه عصبی یعنی وزن ها و بایاس ها هستند و ابر پارامتر چیزهایی که از قبل مشخص می شود و learn نمی شود.

- در مسائل شبکه عصبی که تعداد ابرپارمترها زیاد است،از NAS استفاده می شود.(یکی از روش های NAS الگوریتم تکاملی هست.)
- برای تعیین ابرپارامتر ها باید ، کرومزومی تعیین کنیم که مولفه های پارمتر هایی که قابل بهینه سازی هستند باشند
- نحوه بهینه ساختار شکبه یا ابرپارامترها ← با استفاده از الگوریتم تکاملی
- نحوه بهینه سازی شبکه یا پارامترهایش ← استفاده از back propagation
- معیار برازندگی در مسئه طراحی شبکه صحت رویه مجموعه آزمایشی است.
  تعداد کرومزوم ها ؟
- فضای حالت های رندوم ابرپارامترها باید در محدوده ی زیر باشد :

*جدول ۱: ابرپارامترهای موجود در شبکه و مقادیر مجاز برای هر کدام*

| ابرپارامتر | مقادیر ممکن |
|---|---|
| شبکه استخراج‌کننده ویژگی | ResNet18 - ResNet34 – Vgg11 |
| تعداد لایه‌های مخفی MLP | ۰-۱-۲ |
| تعداد نورون‌ها در هر لایه مخفی | ۱۰-۲۰-۳۰ |
| تابع فعال‌سازی در هر لایه مخفی | ReLU-Sigmoid |

- فیچراکسترکتور هم باید انتخاب شد (جستوجو اعمال شود)(Resnet34, Resnet18,Vgg11) نیازی به بررسی هر فیچراکترکشن نیست.
- لزومی به یکسان بودن تعداد نورون ها در هر لایه مخفی نیست .
- تعداد اجراهای لازم برای ارزیابی با epochs ها فرق دارد. اینگونه است که 5 بار epoch کن و 1 بار تست کن ولی نتیجه ی این یک دقت accuracy است(مثلا 80%) حال به تعداد اجرای لازم برای ارزیابی این عملیات تکرارمی شود(5 بار) این کار 1 بار است و باید 5 تا accuracy مختلف بدست بیاوریم حال میانگین این 5 accuracy برابر فیتنس است .

*جدول ۲: تنظیمات لازم برای حل مسئله*

| | |
|---|---|
| تعداد دورها (epochs) | ۵ |
| تعداد نسل‌های الگوریتم تکاملی | ۱۰ |
| تعداد افراد جمعیت (popSize) | ۱۰ |
| تعداد اجرای لازم برای ارزیابی هر عضو از جمعیت | ۵ |

**بخش سوم :**

- بخش سوم به صورت from_scrach هست.
- با استفاده از شبکه SOM انجام می شود که شبیه k_means است .در k_means باید k را مشخص نماییم.
- در شکبه som لایه اخر آن تعدا خوشه ها را مشخص می کند.(لایه خروجی 10 تا نرون ← 10 تا خوشه)

Batch سایز باید توانی از 2 باشد . مثلا 16 و 32 و ... نمی توان گفت کدام خوب است . باید تست کرد .