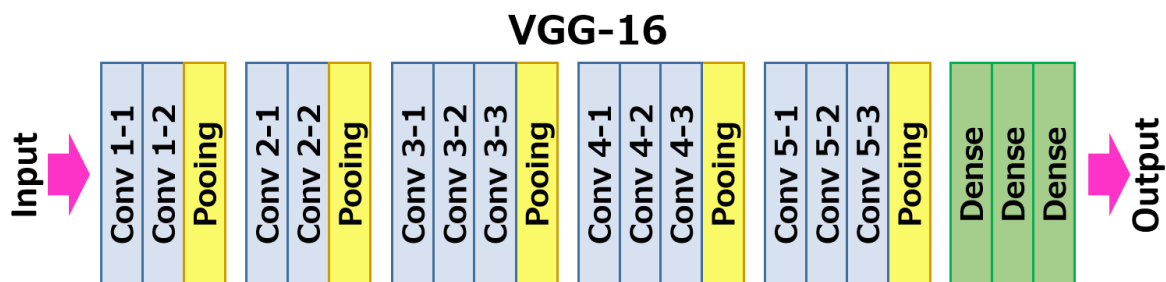


# Fundamentals of Computational Intelligence

Under Supervision of Dr. Shamee  
University of Kashan  
Spring 2024

توضیحاتی مختصر در رابطه با پروژه مبانی هوش محاسباتی:

- در این پروژه از کتابخانه Tensorflow استفاده شده است.
- مدل‌های استفاده شده در این پروژه شامل VGG16 , VGG19 می‌باشد.
- دیتاست استفاده شده در این پروژه CIFAR-10 می‌باشد که در کتابخانه Tensorflow موجود می‌باشد این دیتاست شامل ده کلاس از جمله : قورباغه، ماشین، هواپیما، اسب و .. ( می‌باشد
- در خط لوله پیش پردازش از اعمال نرمالیزیشن و Data Augumentation استفاده شده است.
- نرمالیزیشن به صورتی است که اعداد موجود در هر تانسور یا هر تصویر رو بین یک رنجی قرار می‌دهد و از آنجایی که در هر خانه از تانسور یا هر پیکسل از تصویر عددی بین 0 تا 255 می‌باشد با تقسیم کل دیتا به 255.0 این امر میسر می‌شود و هر خانه عددی بین 0 تا 1 خواهد بود.
- Data Augumentation یا افزایش داده که به صورت دلخواه در پروژه تعریف شده بود به این صورت است که با استفاده از اعمالی روی تصویر از جمله کراپ کردن روتیت کردن و ... باعث افزایش دیتا میشود و این یک امتیاز برای بهتر عمل کردن شبکه عصبی ماست هر چند که نیازی به این کار برای دیتاست CIFAR-10 نیست.
- از مدل vgg16 برای مدل از پیش ترین شده استفاده شد لازم به ذکر است که دیتای از پیش ترین شده برای این نوع معماری imagenet می‌باشد
- معماری vgg16 به شکل زیر است.



- در مدل VGG16 تمامی لایه ها به جز سه لایه آخر غیر قابل train شدند و حتی لایه ای برای بهتر عمل کردن و همچنین خروجی برای ده کلاس اضافه شد

- در مدل VGG19 تمامی لایه ها به جز دو لایه آخر غیر قابل train شدند و لایه هم برای عملکرد بهتر شبکه عصبی اضافه شد که با توجه به پیشرفته بودن این مدل نسبت به مدل قبلی بهتر عمل کرد

- با استفاده از گرید سرچ در کتابخانه scikit-learn مدل ها با مقادیر ابر پارامتر مختلف تنظیم شدند و در نهایت بهترین ترکیب خروجی گرفته شد.

- همانطور که تصویر زیر مشخص است برای این تعداد ابر پارامتر و تعداد ایپاک کم حدود 47 دقیقه زمان برد که به دلیل منابع محدود gpu تعداد ایپاک ها کم انتخاب شد



```
+ Code + Text
[25]
[26] grid = GridSearchCV(estimator=vgg16_model, param_grid=grid, n_jobs=-1)
[27] grid_result = grid.fit(X_train, y_train)
Epoch 1/3
704/704 [=====] - 15s 16ms/step - loss: 1.1647 - accuracy: 0.5934
Epoch 2/3
704/704 [=====] - 11s 15ms/step - loss: 0.9288 - accuracy: 0.6757
Epoch 3/3
704/704 [=====] - 11s 15ms/step - loss: 0.8307 - accuracy: 0.7087
[28] print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
Best: 0.682933 using {'batch_size': 64, 'epochs': 3, 'n_neurons': 512, 'optimizer': 'Adam'}
```

- سپس از دو مدل pretrain به صورت جدا گانه و این بار بدون تکنیک تنظیم استفاده شد.

- برای هر مدل confusion matrix یا ماتریس در هم ریختگی رسم شد که با توجه به color bar سمت راست هر نمودار هرچه خونه مربوط به دو کلاس مشابه پررنگ تر باشد نشان دهنده پیش بینی درست تر است که با توجه به پررنگ بودن روی قطر اصلی ماتریس نشان دهنده مدل های خوبی به وجود آورده ایم.

- برای هر مدل نمودار مربوطه به صحت در داده ترین و داده ولیدشن رسم شد هم چنین تابع لاس برای هر مدل برای داده ترین و داده ولیدشن رسم شده.

ممنون از توجهتون لطفا اگر ابهام یا سوالی در رابطه با پروژه هست ویا جایی واضح نیست اطلاع بدهید.

باسپاس فراوان.