هوش محاسباتی – پروژه اول

توضیح مراحل انجام شده

کد های مرتبط با پروژه در ۵ فایل جدا نوشته شده:

- Activations: در این فایل تعدادی توابع فعال سازی و مشتق آنها نوشته شده.
- Layers: در این فایل لایه Dense تعریف شده و توابع مرتبط با Feedforward و Backpropagation نوشته شده.
- Loses: در این فایل چندین تابع هزینه مختلف نوشته شده.
- Network: اینجا ساختار و کد های مربوط به تشکیل شبکه عصبی نوشته شده است.
- Main: فایل اصلی که نوت بوک جوپیتر بوده و در آن مدل ساخته شده، آموزش داده شده و تحلیل های مرتبط قرار داده شده.

به طول کلی سعی شده کد ها همانند keras بوده و مطابقت داشته باشند. همچنین بهینه سازی های مرحله Vectorization دستور کار، اعمال شده و کد های قبلی حذف شده اند.

بخش دوم: محاسبه خروجی (Feed Forward)

Defining Model Architecture

+ Code + Markdown

```
network = Network()                                  # Initialize network
network.add(Dense(28 * 28, 16, activation = Tanh()))  # Hidden layer #1
network.add(Dense(16, 16, activation = Tanh()))       # Hidden layer #2
network.add(Dense(16, 10, activation = Tanh()))       # Output layer
network.compile(MSE())                               # Compile network
```

[106] ✓ 0.4s

Prediction with no previous training

```
X, y_true = subsample(x_train, y_train, 100)    # Subsample the data
y_pred = network.predict(X)     # Make predictions

# Calculate the accuracy
accuracy = get_accuracy(y_true, y_pred)
print(f'Accuracy: {round(accuracy * 100, 2)}%')
```

[107] ✓ 0.7s

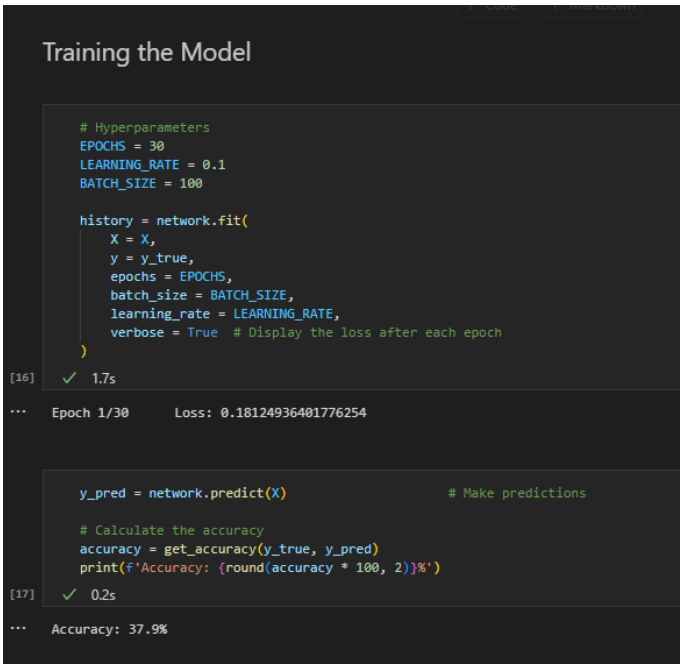
```
Accuracy: 12.0%
```

مطابق تصویر بالا، شبکه عصبی با دو لایه پنهان، تابع Tanh به عنوان تابع فعال ساز و با 100 نمونه که از پیش دیده نشده (مدل آموزش ندیده!) دقت برابر **12%** مشاهده میشود که مطابق انتظار است.

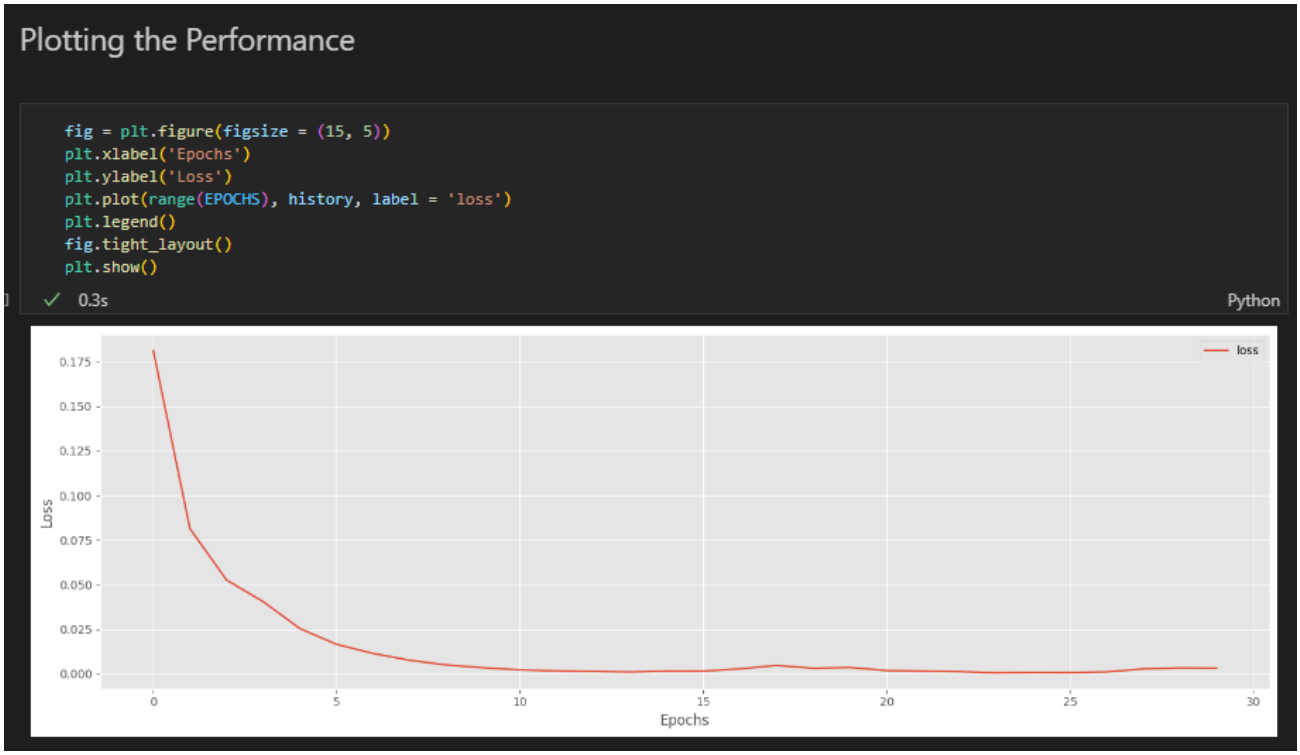
هوش محاسباتی – پروژه اول

بخش سوم: پیاده سازی Backpropagation

در این مرحله مدل با مشخصات داده شده برای 100 داده اجرا شد. زمان صرف شده برای آموزش مدل 35 ثانیه (زمان کم است، روی سخت افزار قوی اجرا شد..) به دلیل معقول بودن زمان آموزش، تعداد Epoch به 30 افزایش داده شد و ضریب یادگیری به 0.1 کاهش یافت و batch size معادل 100 قرار داده شد که باعث افزایش دقت به میزان **37%** شد. نتایج در تصاویر پایین قابل مشاهده هستند (برای جا شدن کد ها در تصویر، خروجی آموزش از تصویر بریده شده):



همانطور که مشاهده میشود روند کاهش خطا مطابق دستور کار است.



بخش چهارم: Vectorization

در این بخش، بهینه سازی های گفته شده انجام شده و تقریبا تمام حلقه های غیر ضروری حذف شدند.

بخش پنجم: تست کردن مدل

خروجی زیر برای Epoch معادل ۲۵ و ضریب یادگیری 0.1 است که دقت **87%** را داده است.