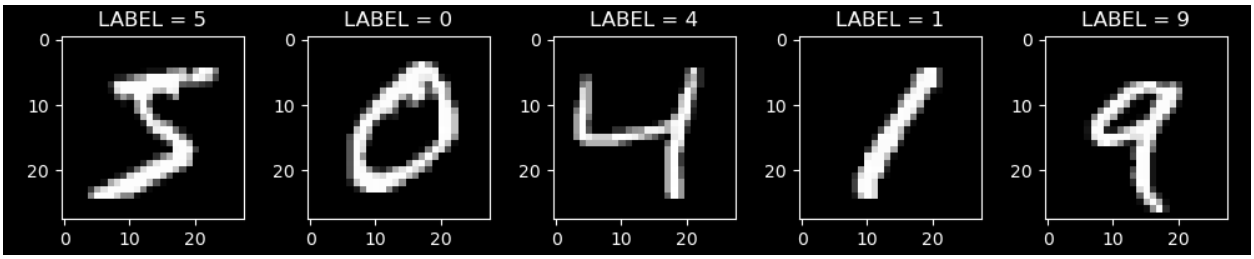
نام و نام خانوادگی: **امیرحسین علی‌بخشی**
شماره‌ی دانشجویی: **۹۷۳۱۰۹۶**

گزارش پروژه شبکه عصبی

استاد: **دکتر عبادزاده**
درس: **مبانی هوش محاسباتی**

گام اول – دریافت دیتاست

برای دریافت مجموعه داده، تابع get_data را تعریف میکنیم که ساختار کلی آن از لینک موجود در دستور کار وجود دارد و تغییراتی جزئی روی آن صورت گرفته است. در این تابع پس از استخراج داده ها به عنوان نمونه، ۵ داده اول را پلات میکنیم و لیبل آن را در پلات نمایش میدهیم:



گام دوم – محاسبه‌ی خروجی

برای پیاده‌سازی شبکه عصبی خود، کلاسی تحت عنوان NeuralNetwork ایجاد میکنیم و اطلاعاتی نظیر وزن ها، بایاس ها و سایر موارد مورد نیاز را به عنوان فیلد های آن تعریف میکنیم. یکی از متد های این کلاس feedforward است که به صورت زیر میباشد:

```
    def feedforward(self, img):
        z1 = (weights[0] @ img[0]) + biases[0]
        a1 = np.asarray([sigmoid(z[0]) for z in z1]).reshape((16, 1))
        z2 = (weights[1] @ a1) + biases[1]
        a2 = np.asarray([sigmoid(z[0]) for z in z2]).reshape((16, 1))
        z3 = (weights[2] @ a2) + biases[2]
        a3 = np.asarray([sigmoid(z[0]) for z in z3]).reshape((10, 1))
        return [a1, a2, a3], [z1, z2, z3]
```

که a3 خروجی شبکه میباشد. حال به کمک تابع calculate_accuracy دقت را محاسبه میکنیم:

```
def calculate_accuracy(self):
        number_of_correct_guesses = 0
        for image in range(self.number_of_samples):
            guess = np.argmax(self.feedforward(self.train_set[image])[0][-1])
            label = np.argmax(self.train_set[image][1])
            number_of_correct_guesses = number_of_correct_guesses + 1 if guess ==
 label else number_of_correct_guesses
        return number_of_correct_guesses / self.number_of_samples
```

برای ۱۰۰ داده اول، دقت نزدیک عدد ۱۰ درصد میباشد. در یک نمونه اجرا خروجی به صورت زیر بدست آمد:

```
        STEP 2: CALCULATING INITIAL ACCURACY
            initial accuracy: 12.0%
```

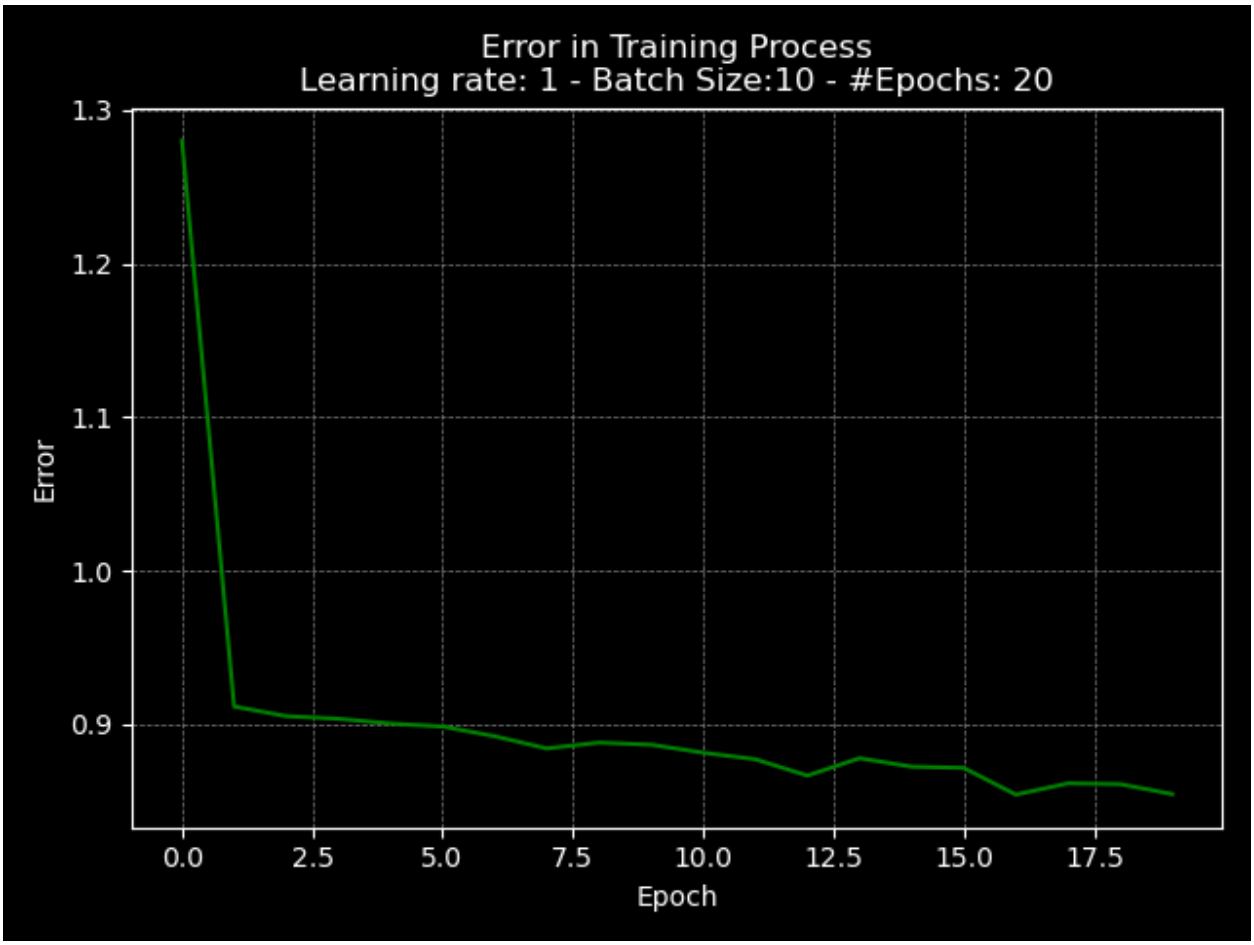
گام ۳ – پیاده‌سازی backpropagation

به کمک متد هایی که داخل کلاس NeuralNetwork تعریف شدند (مانند feedforward و back_propagation) ، شبه کد موجود در دستور کار را داخل train_network پیاده سازی کردیم. در هر لحظه شماره epoch، شماره batch و همچنین شماره عکسی که در حال پردازش آن هستیم در خروجی نمایش داده میشود و در پایان زمان، دقت، و همچنین نمودار میزان خطا برای epoch ها مختلف نمایش داده میشود.

```
TRAINING THE NETWORK:

    EPOCH: 01/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 02/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 03/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 04/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 05/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 06/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 07/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 08/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 09/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 10/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 11/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 12/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 13/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 14/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 15/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 16/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 17/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 18/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 19/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!
    EPOCH: 20/20     BATCH: 010/010   IMAGE: 0010/0010    EPOCH COMPLETED!

    TRAINING PROCESS COMPLETED IN 144S
    THE ACCURACY OF THE NETWORK IS 36.0%
```

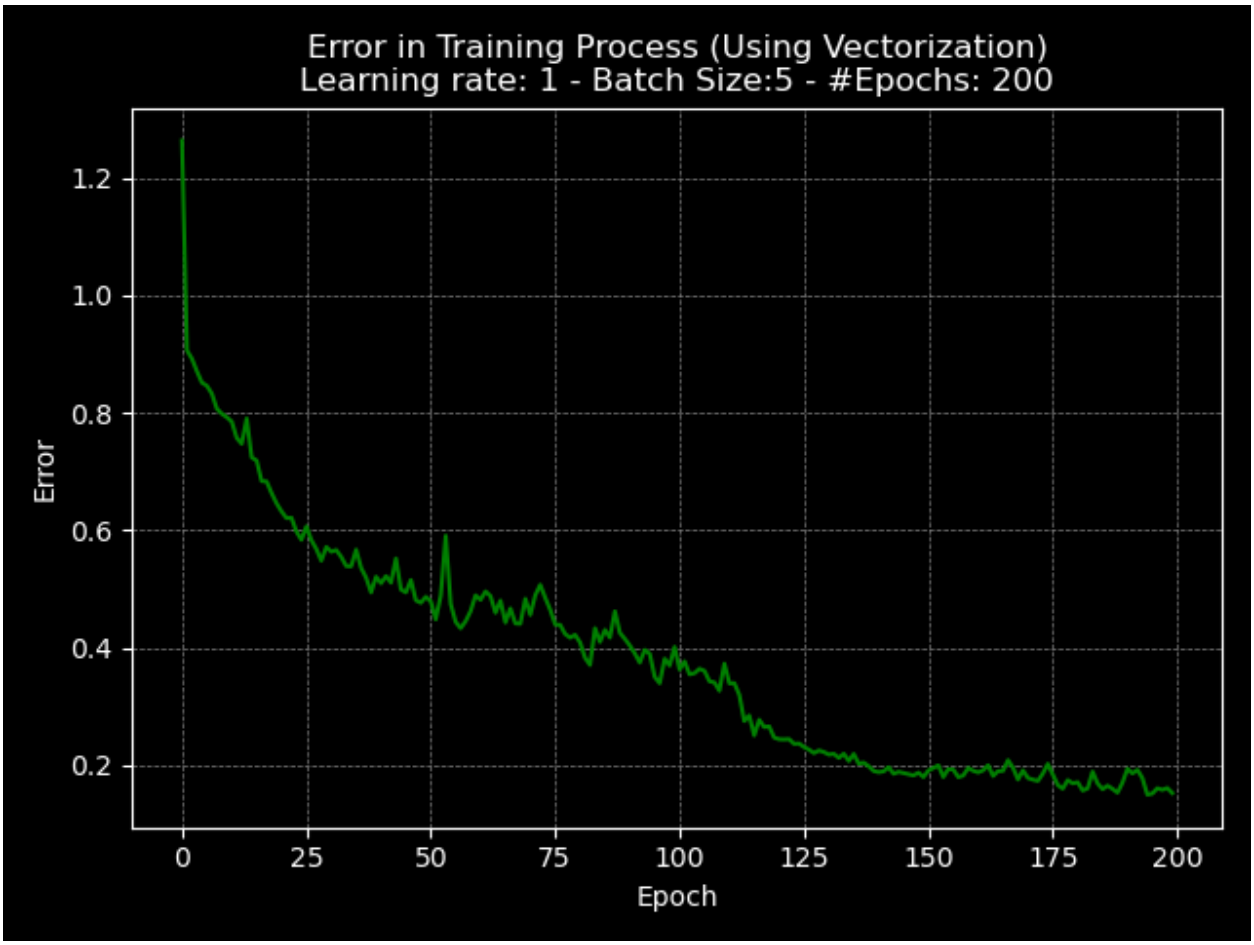


گام ۴ – vectorization

کلاس NeuralNetworkVectorized را میسازیم و متدهای کلاس NeuralNetwork را بازنویسی میکنیم. تعداد epoch ها را برابر ۲۰۰ قرار میدهیم و آموزش را شروع میکنیم:

```
EPOCH: 001/200     BATCH: 020/020   IMAGE: 0005/0005    EPOCH COMPLETED!
EPOCH: 002/200     BATCH: 020/020   IMAGE: 0005/0005    EPOCH COMPLETED!
EPOCH: 003/200     BATCH: 020/020   IMAGE: 0005/0005    EPOCH COMPLETED!
...
EPOCH: 199/200     BATCH: 020/020   IMAGE: 0005/0005    EPOCH COMPLETED!
EPOCH: 200/200     BATCH: 020/020   IMAGE: 0005/0005    EPOCH COMPLETED!

TRAINING PROCESS COMPLETED IN 15S
THE ACCURACY OF THE NETWORK IS 91.0%
```

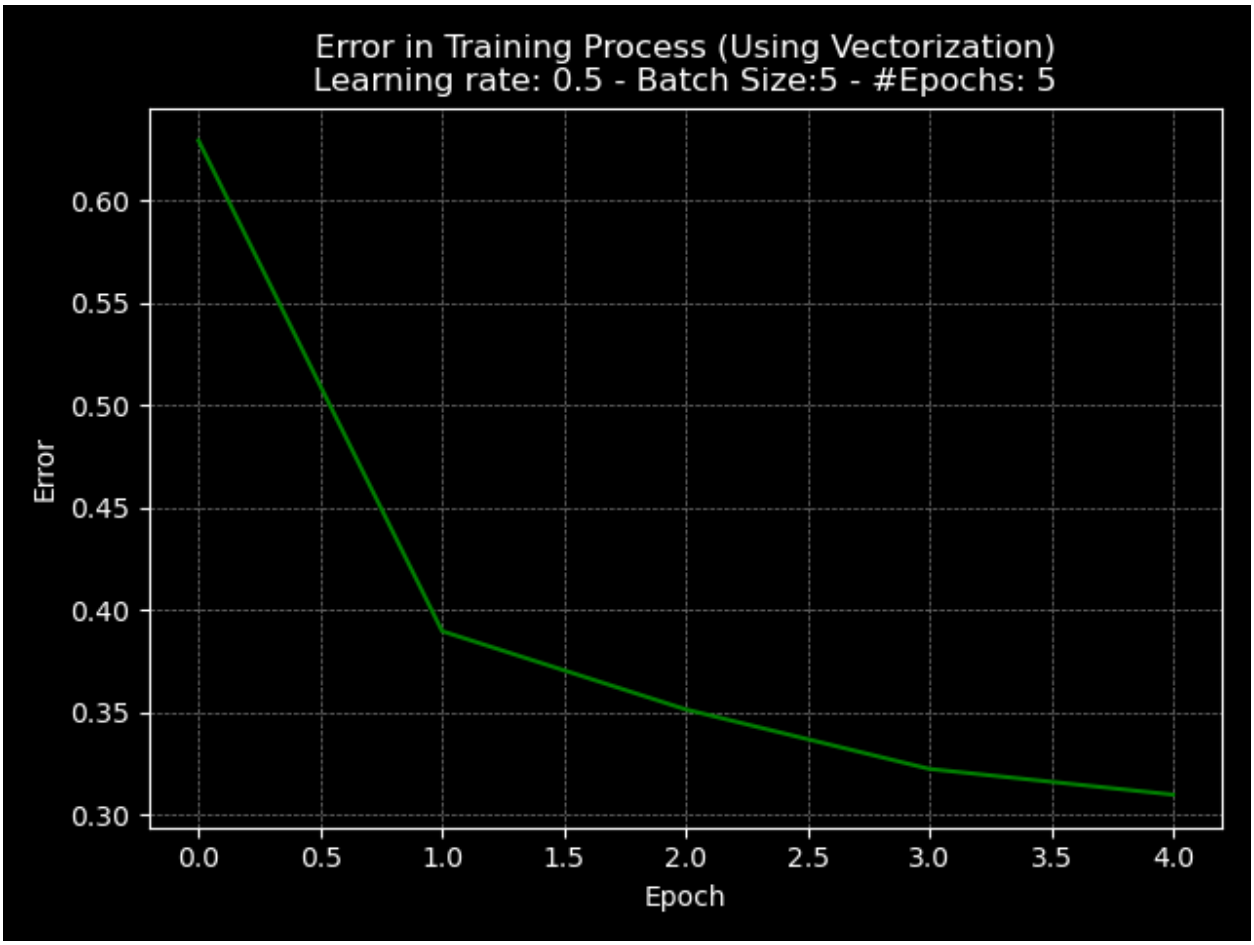


گام ۵ – تست مدل

به کمک ۶۰۰۰۰ داده موجود در train set مدل خود را آموزش میدهیم. سپس نمودار خطا را رسم کرده و دقت مدل را برای train set و test set گزارش میکنیم:

```
EPOCH: 01/05   BATCH: 12000/12000   IMAGE: 0005/0005    EPOCH COMPLETED!
EPOCH: 02/05   BATCH: 12000/12000   IMAGE: 0005/0005    EPOCH COMPLETED!
EPOCH: 03/05   BATCH: 12000/12000   IMAGE: 0005/0005    EPOCH COMPLETED!
EPOCH: 04/05   BATCH: 12000/12000   IMAGE: 0005/0005    EPOCH COMPLETED!
EPOCH: 05/05   BATCH: 12000/12000   IMAGE: 0005/0005    EPOCH COMPLETED!

TRAINING PROCESS COMPLETED IN 229s
THE ACCURACY OF THE NETWORK FOR TRAIN SET: 82.81666666666668%
THE ACCURACY OF THE NETWORK FOR TEST  SET: 82.86%
```



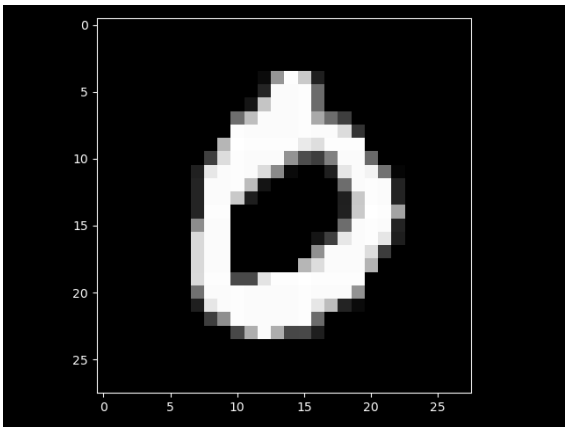
سوالات امتیازی

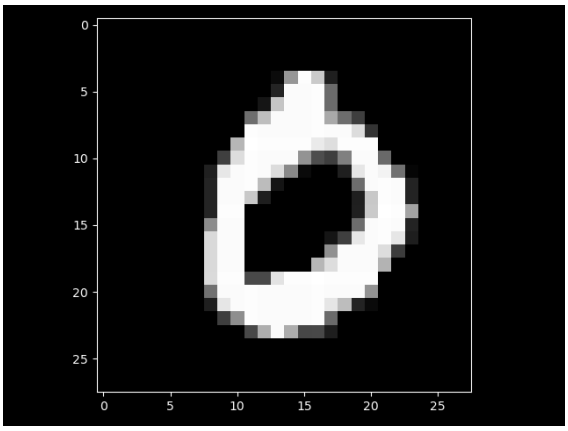
شیفت مجموعه تست به راست

به کمک تابع `shifted_test_set` یک کپی از مجموعه ورودی میسازیم و ورودی های محموعه را (ماتریس ۲۸×۲۸) چهار واحد به راست شیفت داده و از سمت چپ ضفر وارد مدار میکنیم.

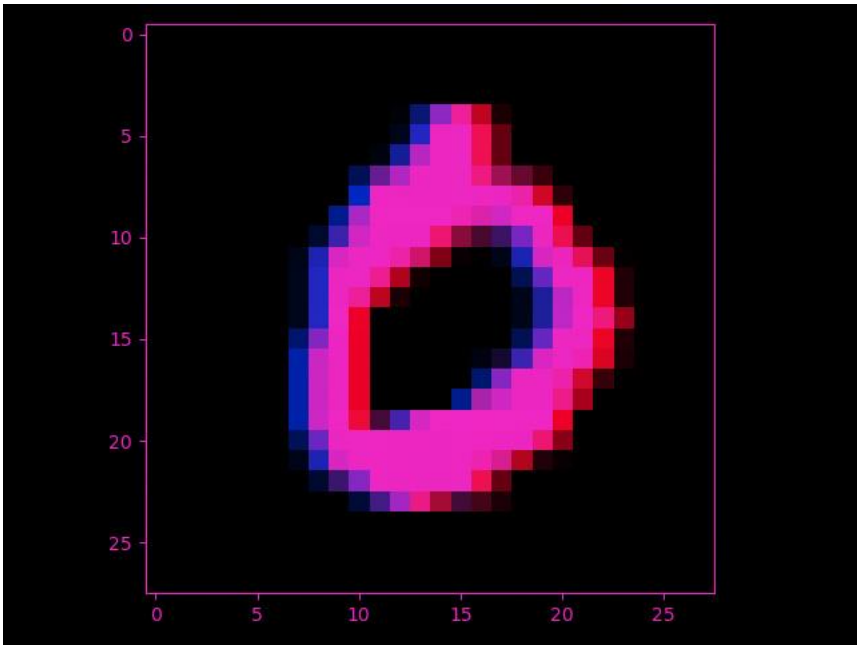
```
def shifted_test_set(test_set):
    res = copy.deepcopy(test_set)
    print('shifting pixels 4 units to the right'.upper())
    for i in range(len(res)):
        l = list(res[i])
        l[0] = l[0].reshape((28, 28))
        np.roll(l[0], 1, axis=1)
        for j in range(4):
            l[0][:, j] = np.zeros((28,))
        l[0] = np.matrix.flatten(l[0])
        res[i] = tuple(l)
    return res
```

بعنوان مثال یکی از داده ها را قبل و پس از این تابع مشاهده میکنیم:





اگر یکی از این عکس ها را آبی رنگ و دیگری را قرمز کنیم و آن‌ها را روی هم منطبق کنیم متوجه شیفت خواهیم شد:



در چنین حالتی ابتدا یکبار دیگر دقت را برای train_set و بار دیگر برای خروجی تابع shifted_test_set محاسبه میکنیم.

```
TRAINING PROCESS COMPLETED IN 223S
    THE ACCURACY OF THE NETWORK FOR TRAIN SET:  74.50666666666666%
    THE ACCURACY OF THE NETWORK FOR TEST  SET:  11.35%
```