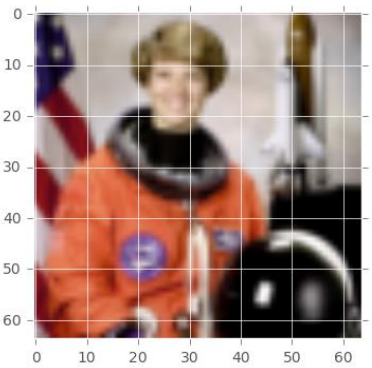
تمرین دوم: حل یک مسئله رگرسیون با استفاده از شبکه‌های عصبی

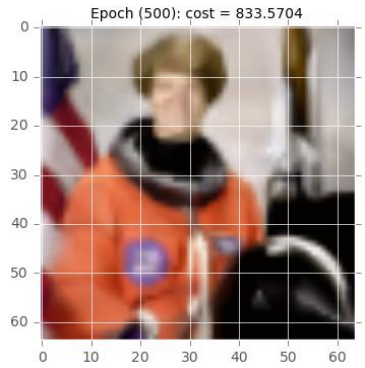
[مهلت تحویل: ۱۰ آذر ۱۳۹۶]

مسئله. بازسازی تصویر ورودی به وسیله رگرسیون

- ورودی: یک تصویر با ابعاد ۶۴ × ۶۴
- خروجی: تصویر تقریب زده شده با ابعاد ۶۴ × ۶۴



تصویر ورودی



تصویر خروجی

مجموعه آموزشی

- ورودی در هر نمونه آموزشی: مختصات یک پیکسل به صورت زوج مرتب $(x.y)$
- خروجی در هر نمونه آموزشی: رنگ پیکسل در فرمت RGB به صورت سه‌تایی مرتب $(r.g.b)$ به گونه‌ای که هر یک از این سه مقدار یک عدد حقیقی بین صفر و یک هستند.
- با توجه به اندازه تصویر ورودی که باید برابر با ۶۴ × ۶۴ باشد، در نتیجه تعداد کل پیکسل‌ها برابر با ۴۰۹۶ است. به عبارت دیگر اندازه مجموعه آموزشی برابر با ۴۰۹۶ است (هر پیکسل بیانگر یک نمونه آموزشی است).

ساختار شبکه عصبی

- لایه ورودی: مختصات پیکسل (دو عدد)
- لایه خروجی: رنگ پیکسل به صورت RGB (سه عدد حقیقی دربازه صفر و یک)
- لایه یا لایه‌های مخفی (تعداد و اندازه هر یک از لایه‌های مخفی را شما باید تعیین نمایید)

مراحل توسعه برنامه

۱) وارد کردن کتابخانه‌های مورد نیاز

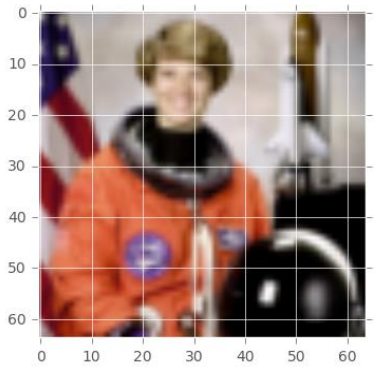
```
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('ggplot')

%matplotlib inline
```

۲) خواندن تصویر ورودی، تغییر اندازه و ترسیم آن

```
# Enter your code here to read input image
input_image = # …
```

```
from scipy.misc import imresize
img = imresize(input_image, (64, 64))
plt.imshow(img);
```



۳) ایجاد مجموعه آموزشی

```
xs = []  # locations of pixels
ys = []  # colors of pixels

for row in range(img.shape[0]):
    for col in range(img.shape[1]):
        # store pixel location as input
        xs.append([row, col])
        # store pixel color as output
        ys.append(img[row, col])

# convert input and output lists to numpy arrays
xs = np.array(xs)
ys = np.array(ys)
```

```
print("Input shape: ", xs.shape)
print("Output shape: ", ys.shape)
```

۴) پیش‌پردازش داده‌های ورودی

```
# your code goes here
# ...
```

۵) ایجاد شبکه عصبی برای رگرسیون

```
# your code goes here
# ...
```

۶) آموزش

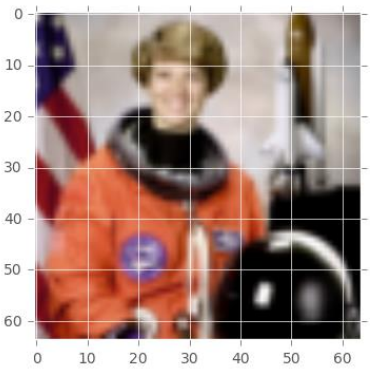
```
# your code goes here
# ...
```

۷) ترسیم نمودار خطا برحسب تعداد تکرارهای آموزش

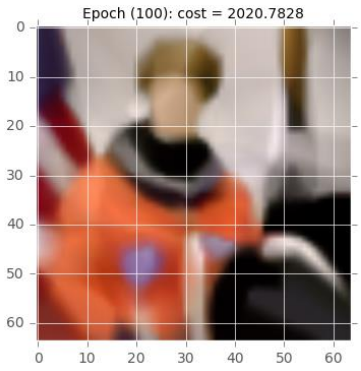
```
plt.figure(figsize=(12, 4))
plt.plot(training_costs)
plt.xlabel('Epoch')
plt.ylabel('Cost')
plt.title('Training Cost Per Epoch');
```

نمونه اجرا

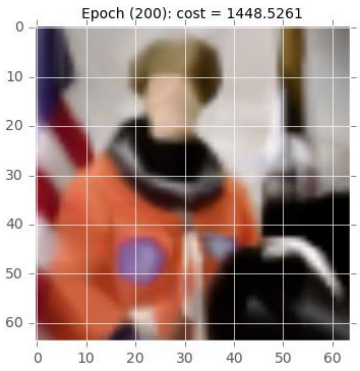
یک مثال از مراحل اجرای شبکه عصبی برای این مسئله و تصاویر تولید شده پس از هر ۱۰۰ تکرار.



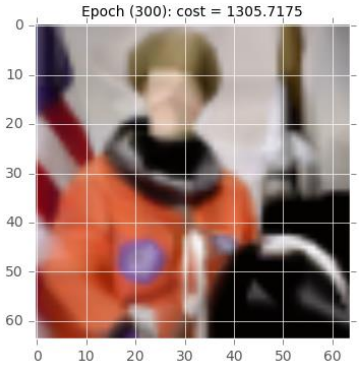
تصویر ورودی



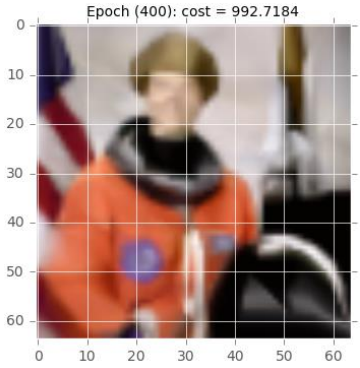
تصویر حاصل پس از ۱۰۰ تکرار



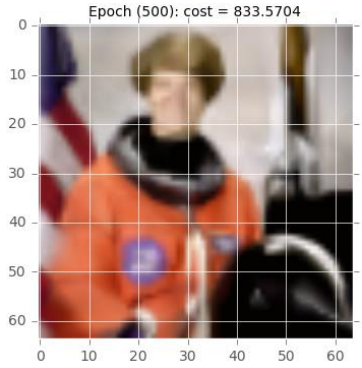
تصویر حاصل پس از ۲۰۰ تکرار



تصویر حاصل پس از ۳۰۰ تکرار



تصویر حاصل پس از ۴۰۰ تکرار



تصویر حاصل پس از ۵۰۰ تکرار

خروجی‌های مورد نیاز

- ساختار شبکه عصبی [تعداد و اندازه هر یک از لایه‌های مخفی]
- مقدار نهایی تابع هزینه
- نمودار مقدار تابع هزینه به ازای هر تکرار
- کل مدت زمان مورد نیاز برای آموزش

توجه: به منظور محاسبه مدت زمان اجرای یک تکه از کد در پایتون می‌توانید به صورت زیر عمل کنید:

```
import time
start_time = time.time()
# the code you want to compute its execution time
# ...
end_time = time.time()
execution_time = end_time - start_time
print(“Total Training time = %.2f seconds” % execution_time)
```

توجه: لطفا کدها را از این فایل کپی و پیست نکنید و به جای این کار خودتان کدها را در محیط ژوپیتر نوت‌بوک تایپ نمایید.

مراجع و منابع

[۱] اسلایدهای مربوط به رگرسیون، درس یادگیری ماشین، سید ناصر رضوی، دانشکده مهندسی برق و کامپیوتر، دانشگاه تبریز. این اسلایدها را می‌توانید از اینجا دانلود نمایید.

[۲] ویدیوی مربوط به هفته چهارم از «کارگاه یادگیری ماشین با پایتون». (لینک مشاهده ویدیو در یوتیوب)