قدم اول :

این مرحله در فایل first_step.py می باشد .

در این مرحله از کد Loading_Datasets که در اختیارمان قرار گرفته بود استفاده می کنیم و داده ها را به شکل مناسب load می کنیم .

قدم دوم :

این مرحله در فایل second_step.py می باشد .

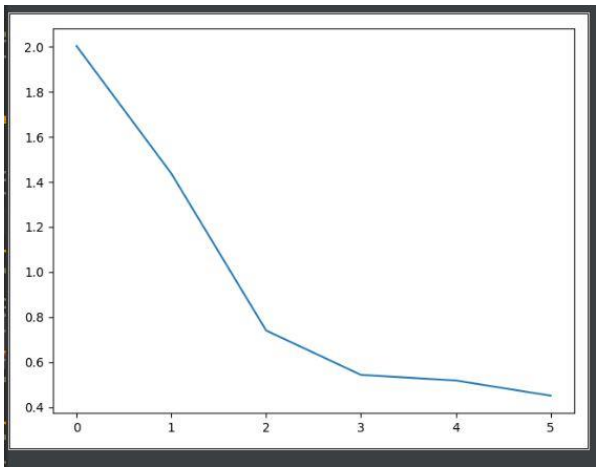
در این مرحله پس از لود کردن داده ها یک کلاس شبکه عصبی تعریف می کنیم و وزن ها و ... را برایش تعریف می کنیم و سپس تابع feed_forward را پیاده سازی می کنیم . همچنین 200 داده را به شبکه عصبی می دهیم و دقت خروجی را اندازه گیری می کنیم که به طور میانگین دقتش حدود 25 درصد شد . علت این است که ما شبکه را هنوز train نکرده ایم و صرفا با همان مقادیر تصادفی اولیه وزن ها پیشبینی را انجام داده ایم .

قدم سوم :

این مرحله در فایل third_step.py می باشد .

در این مرحله backpropagation را به صورت معمولی پیاده سازی می کنیم و برای به دست آوردن گرادیان وزن ها روی تک تک آنها لوپ میزنیم . به صور میانگین حدود 7 دقیقه زمان لرن کردن طول می کشد و دقت هم به صورت میانگین حدود 62 درصد می باشد .

نمودار میانگین cost در هر epoch :

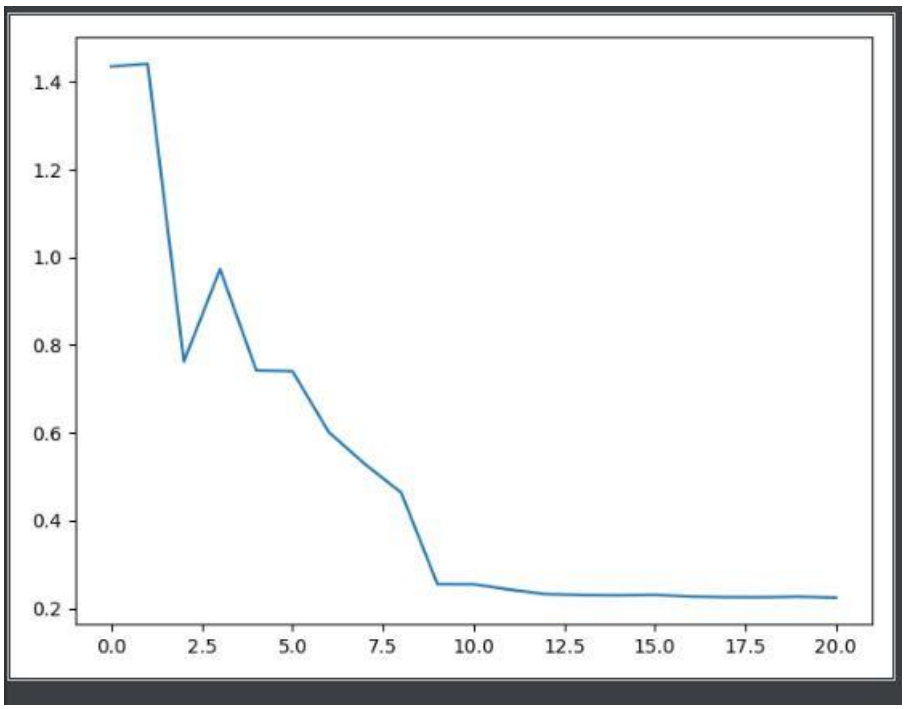


قدم چهارم :

این مرحله در فایل fourth_step.py می باشد .

در این مرحله به دست آوردن گرادیان وزن ها را به صورت Vectorized انجام می دهیم و با اینکار سرعت یادگیری به شدت افزایش می یابد و زمان یادگیری برای 20 ایپاک ، به طور میانگین 3.5 ثانیه می شود!

نمودار میانگین cost در هر epoch :



همچنین با میانگین گرفتن از 10 بار اجرای کد (با ایپاک 20) ، دقت به طور میانگین حدود 91 درصد می باشد .

قدم پنجم

این مرحله در فایل fifth_step.py می باشد .

اکنون روی کل داده های train آموزش را با ایپاک 10 انجام می دهیم (که حدوداً 18 ثانیه طول می کشد) و سپس دقت را برای داده های test نیز می سنجیم . میانگین 10 بار اجرای کد به صورت زیر است :

دقت Train : 95 درصد

دقت Test : 94 درصد

نمودار میانگین cost در هر epoch :