1:

1. الگوریتم سنکرون می‌تواند در حلقه بی‌نهایت بیافتد ولی الگوریتم ترتیبی هیچگاه در حلقه‌ی بی‌نهایت نمی‌افتد. در الگوریتم ترتیبی گام‌های کوچک همیشه یا وکتور کنونی را تنها می‌گذارند یا به یک همسایه اکیدا نزولی تغییر انرژی می‌روند. چون شبکه محدود است این کاهش انرژی تا ابد ادامه پیدا نمی‌کند پس یک مینیمم محلی دارد ولی در الگوریتم سنکرون اینگونه نیست و امکان دارد در حلقه بی‌نهایت بیافتد.
2. خیر. دو تا از الگوهای هدف عکس می‌شوند تا مینیمم محلی انرژی نباشند.
3. ساده سازی اطلاعات خروجی از لایه‌ی کانولوشن. به بیانی دیگر این لایه هر feature map خروجی از لایه کانولوشن را می‌گیرد و یک feature map فشرده تولید می‌کند.
   1. Weights Number = (5\*5\*6 + 1)\*8 = 1208 (+1 for bias)
   2. Weights Number = (20\*20\*6)\*(10\*10\*8) = 1920000

2:

batch\_size = 128 , validation\_data=(x\_test, y\_test)

model.add(Conv2D(32, kernel\_size=(5, 5),

strides=(1, 1),

activation='relu',

input\_shape=input\_shape))

model.add(MaxPooling2D(pool\_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(64, (5, 5), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

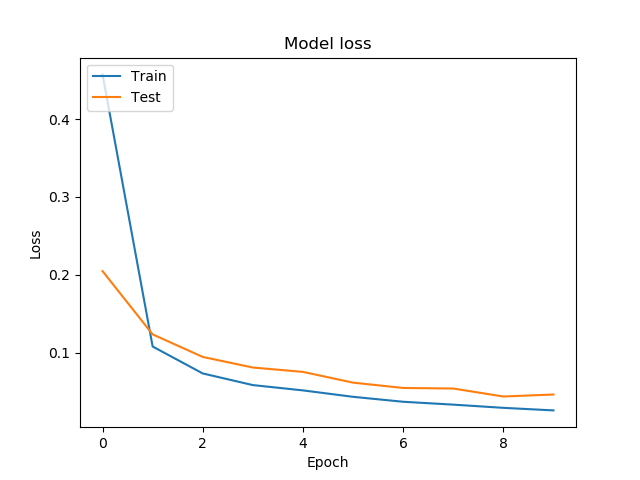
model.add(Dense(1000, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(num\_classes, activation='softmax'))

دو فیلتر با سایز (۵ و ۵) و دو لایه MaxPooling که در بالا مشخص است.





3:

20%

30%

30%

0%

20%

30%

20%

20%

20%

**% noise**

**10**

**% noise**

**30**

**% noise**

**60**

**Font size 16**

**Font size 32**

**Font size 64**

افزایش نویز تشخیص حروف را برای شبکه سخت تر می‌کند و باعث عدم تشخیص(ترکیب ۲ یا چند حروف) یا تشخیص اشتباه می‌شود. افزایش فونت کمی تشخیص را بهتر می‌کند.

در این تمرین از شبکه هاپفیلد سنکرون استفاده شده است. نتایج در پوشه Result ذخیره شده‌اند.