CS 446 MJT — Homework 2

junzhew3

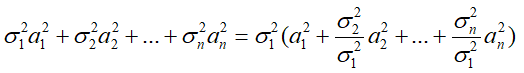
haoxuan3

1. **Singular Value Decomposition and norms**

∵ U is orthonormal ∴ ∴

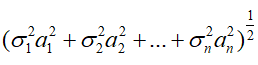
Same ∴

∴ and S ,  is the largest singular value.

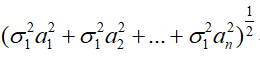
Assuming, where.

∴ <=, and only when x=(1,0,...0)T, have the max value .

∴ 



1. As proved in question (a), ，and

S , is the largest singular value. Assuming 

∴

∵ is the largest singular value

∴

1. ∵ and

∴

∵

∴

∵

∴

∵ is proved in the above question

∴

∵

∴

∵ U and V are orthonormal

∴

∴ same

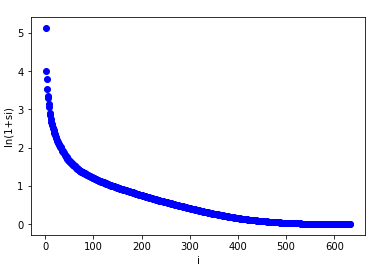
∴ same

1. Assuming , ∴ ,

Assuming , ∴

1. **Singular Value Decomposition and image reconstruction**

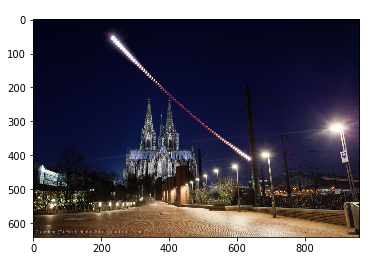
(b)



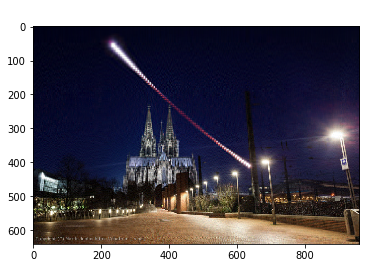
The plot is nonlinear, we can find that at first the value is very large, then decrease sharply, and finally level off to 0. Most of this values are contributed by the front.

(c)

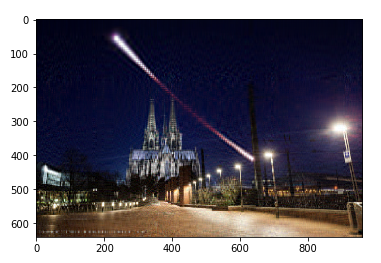
Origin image:



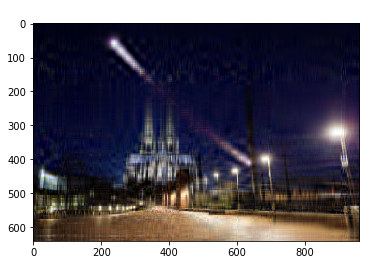
The 100-reconstruction of the image:



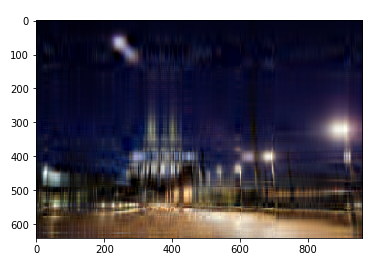
The 50-reconstruction of the image:



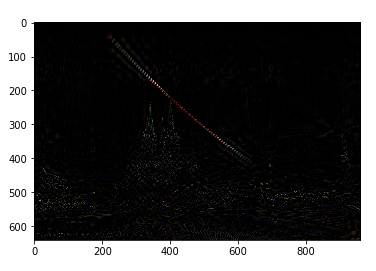
The 20-reconstruction of the image:



The 10-reconstruction of the image:

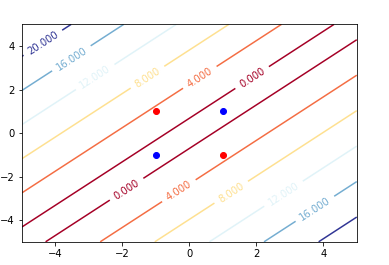


The min-600-reconstruction of the image:



1. **Neural Networks on XOR**

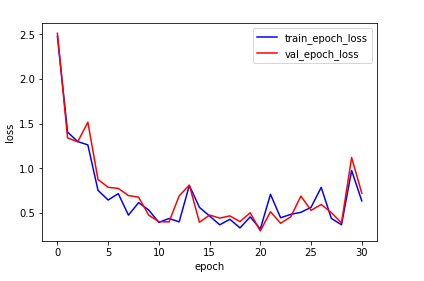
(c)



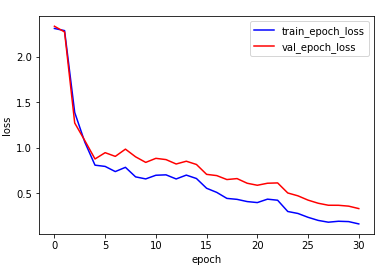
1. **Convolutional Neural Networks.**

(e)

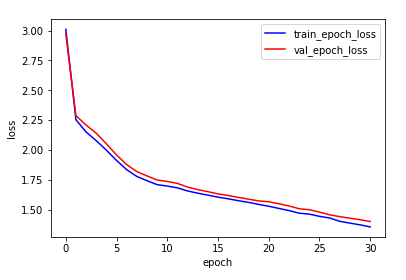
Plot of (c):



Plot of (d):



Plot of (e):

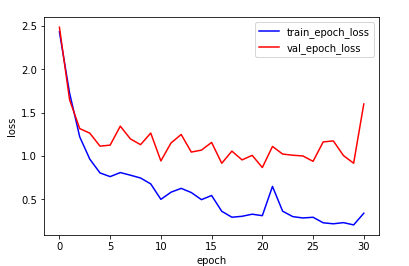


Exponentially decaying learning rate will make the computation a little heavier. But when comparing plot of (c) and Plot of (d), we can find the loss will decrease more smoothly and get better results. We can see in the Plot of (d), the val\_epoch\_loss is really closed to 0 after 30 epoch and the train\_epoch\_loss is a little bigger but also closed to 0.

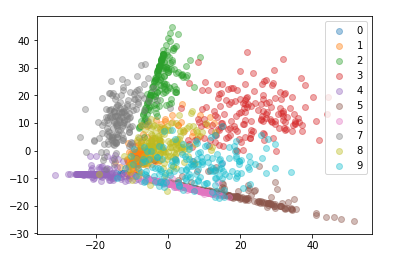
Increasing the batch size will make the calculation faster. And when comparing plot of (c) and Plot of (e), we can find the curves of loss change are smoother. But the value of loss is maximum in the three plots.

(f)

Loss plot:



PCA plot:



(g)

The accuracy of KDTree on the validation set is 0.8555555555555556