

şimdi bu ne diyeceksin..

veri seti Fashion MNIST, 10 class, 60K train 10K test image var 28x28'lik grayscale resimler..  
kl divergence ile sparse autoencoder eğittim..

bottleneck feature'ları ne kadar kullanılabilir - classification için diye baktım..

bottleneck 16 dimension..

kmeans yapıp cluster'ların center image'larına göre re-label yaptığımda %62 train ve %64 test accuracy elde ediyorum..

bu 500 epoch koşturduğum experimentin sonucu

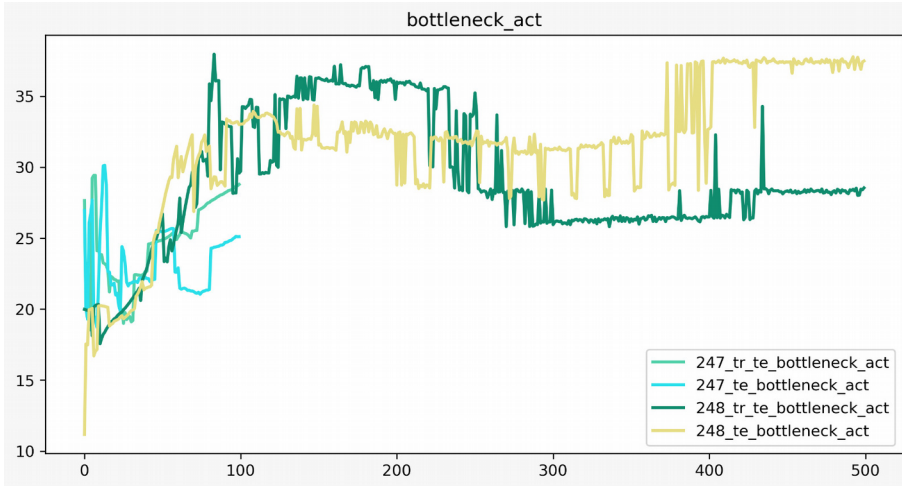
ama asıl hedefim olan - **sparsity** sayesinde **bottleneckdeki** aktivasyon bana **clustering** yapsın –  
bottleneck aktivasyonlarına bakarsak:

**pred\_vec = np.argmax(bottleneck\_vec, axis=1)**

orada çok ilginç bir durum var..

tr\_te\_ dediğim – train datasının test'miş gibi kullanılması.. yani no perturbations in transform..

te – zaten test :)



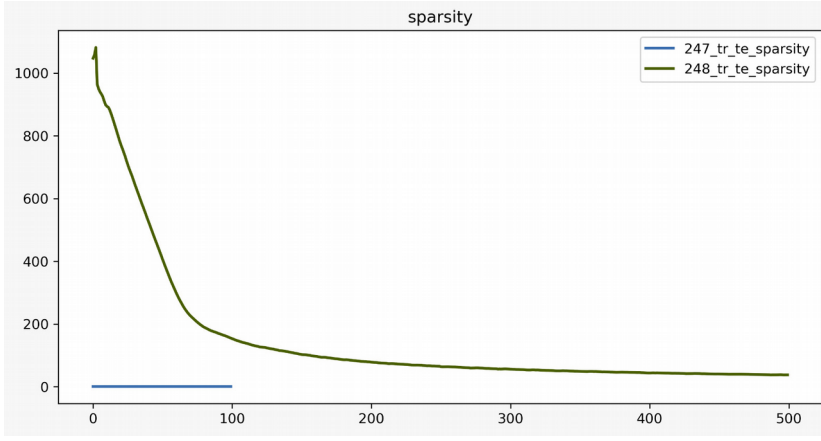
experiment 247 sadece 100 epoch ona bakma.. koyu yeşil ve bok sarısına odaklan :)

abi *test set accuracy* train'e göre 250'nci epoch'tan sonra daha accurate oluyo :)

%34 test accuracy, %28 filan da train accuracy 500 epoch sonra..

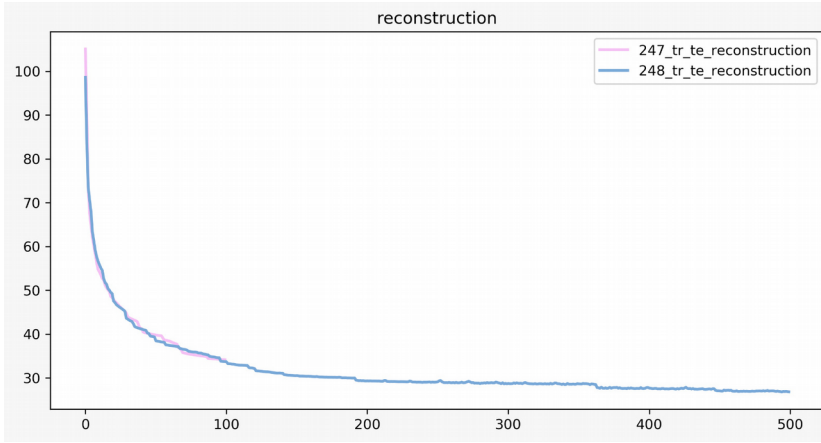
ama zıplamaları görüyorsun.. canımı sıkan da o zıplamalar..

tabi accuracy'de bunu görünce sparsity error'a bakmak lazım.. o ne oluyor..



mavi çizgiyi boşver brom..

sparsity error gayet güzel bir şekilde düşmeye devam ediyor..  
şu an evde 700 epoch'a ilerle dedim zaten hayvana..



bu da reconstruction error – MSE

amaç burada 3 farklı set edilebilir..

1. **bottleneck feature**'ları kullanışlı olsun - normal bir autoencoder'dan daha iyi olsun!!
2. **bottleneck**'de maximum aktivasyonların yaptığı **clustering** iyi olsun - icassp paper amacı buydu..
3. **sparsity enforce** ederek **MSE daha düşük** çıksın.. sparsity yokken elde edemediğim MSE'yi elde edeyim..

aslında bu küçük çaplı bir multi-task..

bottleneck'ten 2 şeyi backpropagate ediyorsun..

- 1 reconstruction loss,
- 2 sparsity loss..

şimdi bu modelde sanki bu verisetinde robust birşeyler elde ettim gibi..

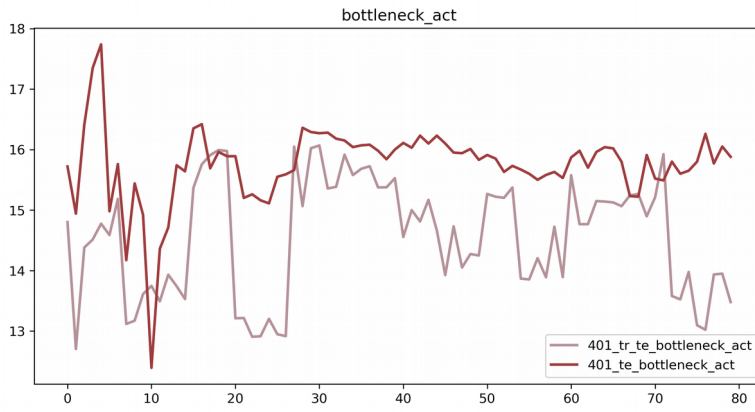
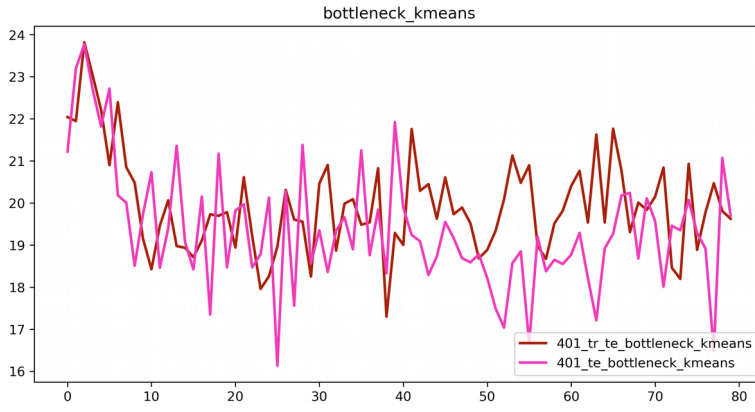
ama accuracy doesnt seem enough.. ama 16 cluster'da açıkçası way too less..

cluster sayısını 256'ya çıkarıp experimenti tekrar run edeceğim..

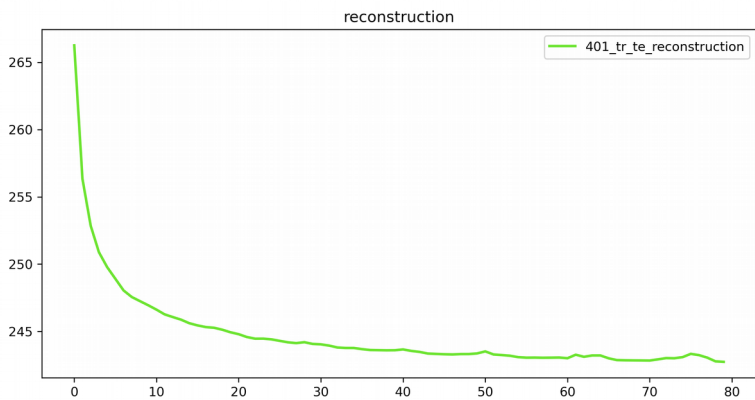
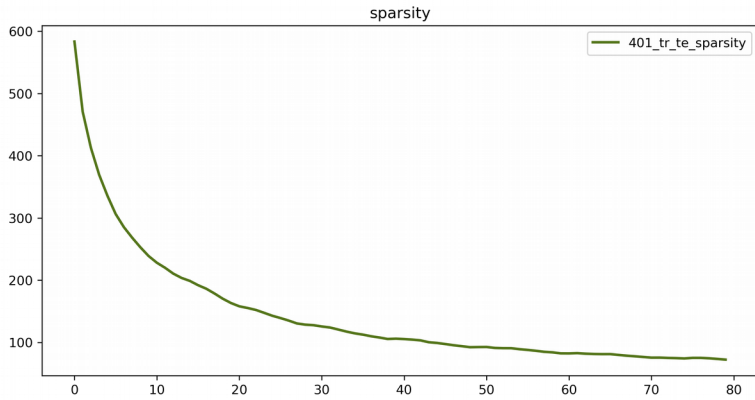
bizim keyhandshape'lerde de 64e64 veya 128e128'lik croplarda deneyeceğim..

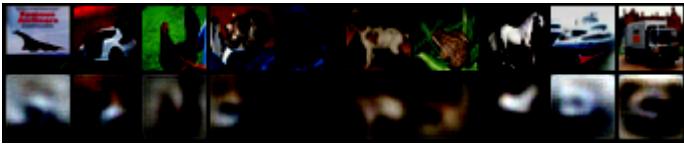
sonra senle bu feature'ların kmeans ve bottleneck activation accuracy'leri yüksek çıkmasına kasıyorum..

Aynı sparsity fonksiyonunu bir de RGB veride denedim.. Cifar10 - experiment401

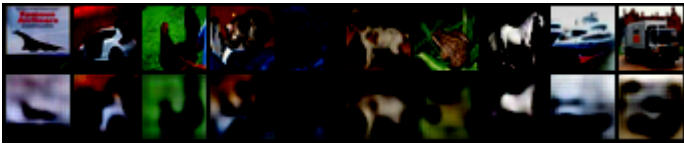


şu zıplamalar delirtiyo.. bi de test genelde train'den yüksek..





epoch 1



epoch 30



epoch 79