Summary

1.a. Image representation

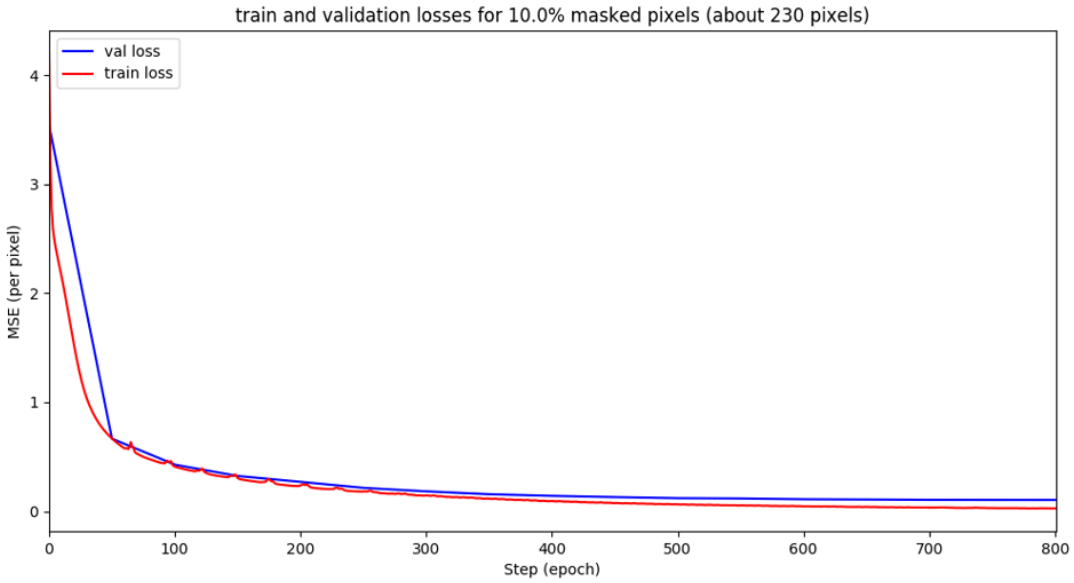
a. Network architecture and input
   Input – Add additional channels to encode the serial number of each image
   Network – Naïve larger hidden layer to enable the network to memorize more data.

II. Generalization –

a. The first 'generalization' that came to mind, was that of common learning task – e.g. generalization to unseen examples. In our case, since for every input grid of an image we have specific output (so every pixel coordinate / rgb output is a learning example), we can create a test set by masking-out some portion of the image pixels, using them for validation purpose. In this specific case 10% of the pixels of each image where masked during training, and used for validation.
   Those pixels should be fixed for each given image and the relevant predicted values during training should not reflect on the training loss (e.g. masked)

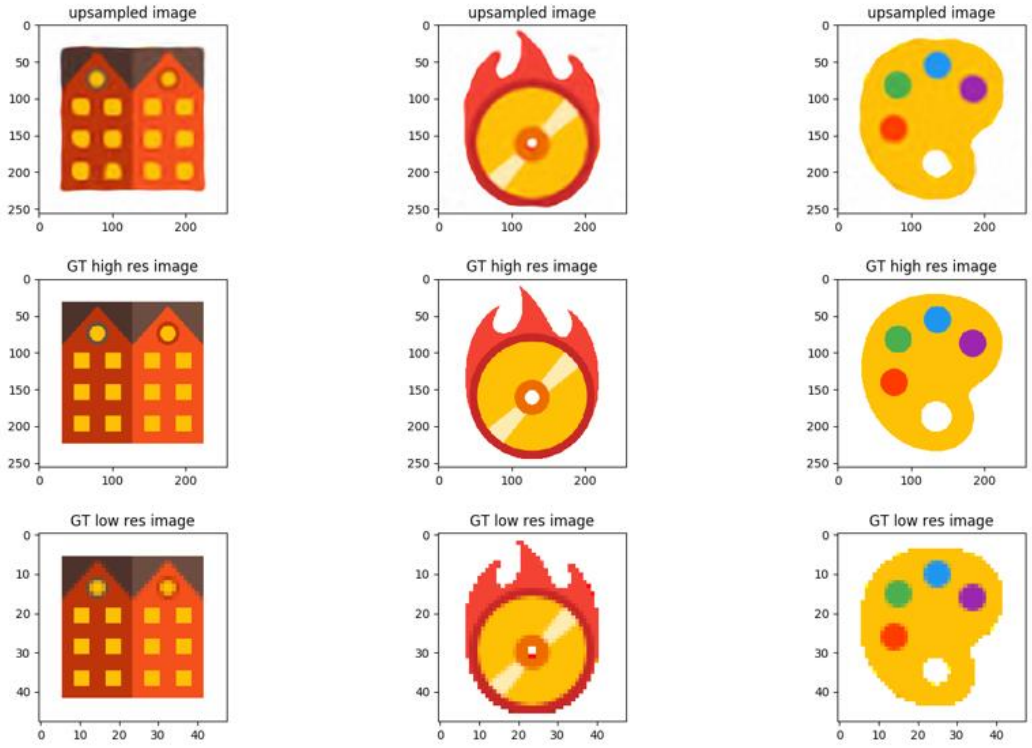
b. Another possible interpretation in the context of memorization might be the simple error between the represented dataset and the produced one. In some sense we expect the net to accurately represent/memorize the full data set, so any error can be viewed as generalization error of sort. I must admit this interpretation feels a bit less natural to me, and I need to convince myself this is a legitimate use of the term generalization.

III. Networks generalization during training
   Train/val loss:



2.a. Up sampling demonstration

I. Up-sampling image from 48x48 to 256x256 with net trained on 48X48 images



2.b. Interpolation between selected pairs of images

I. **Similarity measure** –

considerations:

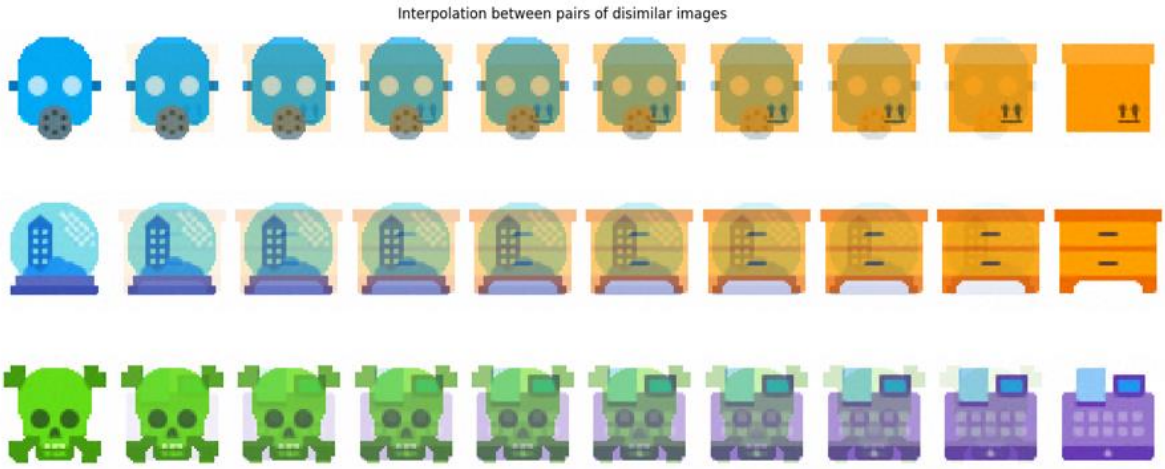
- *Activation sparsity* – Sparse
- *Encoding effective length*- Some images need more 'active' activation then others
- *Desired similarity result* – large latent space / vector. We want similar direction for similar representation. Amplitude should not skew the measure

Cosine similarity was chosen

$similarity = \cos(\theta) = \frac{x \cdot y}{\|x\|\|y\|}$.

II. **Demonstrate interpolation**

a. Dissimilar



III. **Interpolation shortcoming**

One can hypothesize the following about our 'blending' effect:

- 'Poor' representation of the imbedding space
- Since we use the last hidden layer as our embedding – we don't leave much room for interpretation

3.a+c. Design of improved solution and explain why this should produce refined solution:

Following up on the assumption of 'poor representation of embedding space', and use of last hidden layer (followed by a linear one), an improved solution would relate to several key points:

1. Compact embedding space, not at the end of the architecture – in other words a sort of encoder-decoder
2. Additional data – Augmentation and additional dataset
3. Generalization beyond memorization – As presented in Sitzmann et al. work, it is possible to learn a "prior of the space of function parametrized by SIREN"