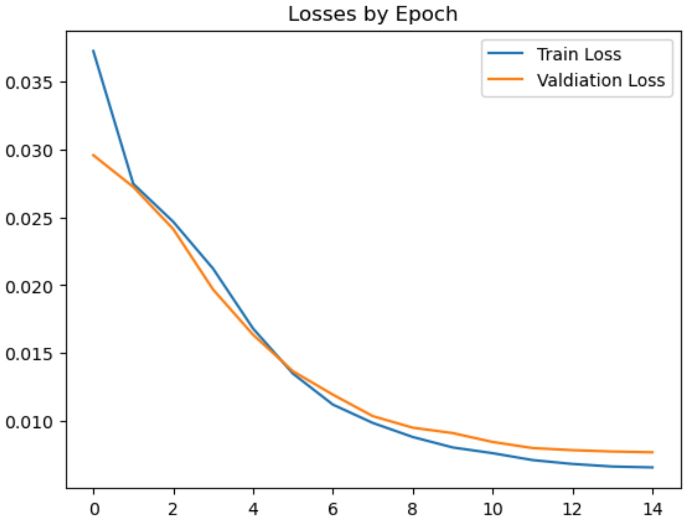
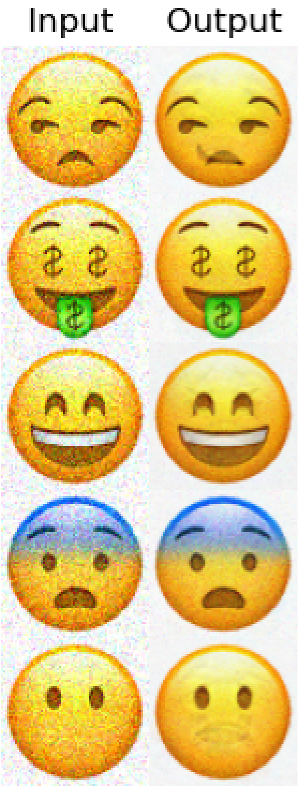
Deep Learning Homework #2

Team: Jack Hanke, Dan Plotkin, Nicole Birova

1. **Question 1.**
   a. The dataset was created by filtering the emoji dataset to all emojis that have the label ‘face’, and then further filtering this dataset to remove all emojis that did not fit the “circular face style”. For example, the facepalming emoji was in a different style than the smiling face emoji, and so were manually filtered out. This resulted in 90 distinct emojis. These 3x256x256 images were then reduced to 3x64x64, and then augmented with 10 copies of each emoji perturbed with Gaussian Noise (default options provided by Pytorch transforms).
   b. Our final network contained an encoder with two convolution blocks and a fully connected layer into a 512 dimensional latent space. The decoder was a two layer feed forward neural network.
   c. We first created a simple feed-forward autoencoder with a symmetrical encoder and decoder, filtering from 3*64*64 neurons to 64*64 neurons to 4*64 neurons. We used LeakyReLUs, except for the final output being a sigmoid. This was slow to train and mostly just predicted the average of the dataset. We transitioned to using a convolutional encoder to reduce training time and improve performance. We chose small kernel and pooling sizes, as we already reduced the image size substantially. Our initial experiments with the FFNN showed we needed a somewhat large latent space to get the network to not predict the average emoji.
   d. We used the ADAM optimizer with a learning rate of 0.001, and trained for 15 epochs. This is the default learning rate from PyTorch, and we chose 15 epochs as the validation started to bottom out.
   e. See plot below.

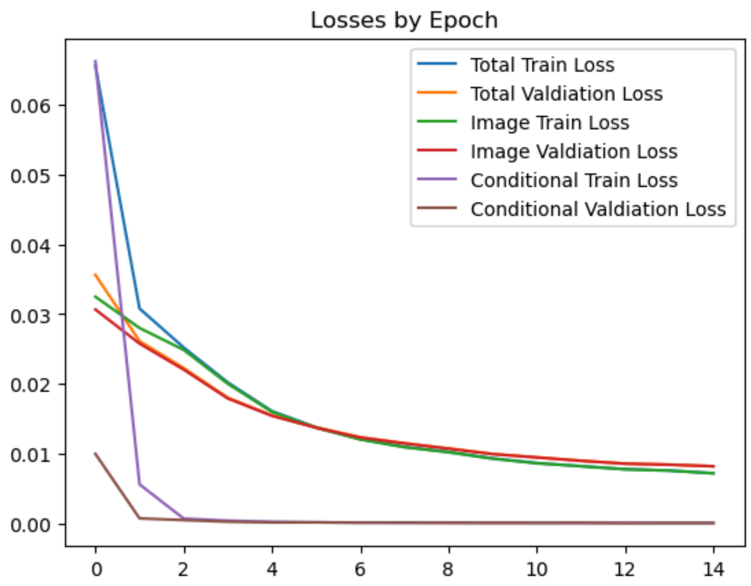


   f. The final average test mean squared error was 0.00777.
   g. See the input/output of the final network below.

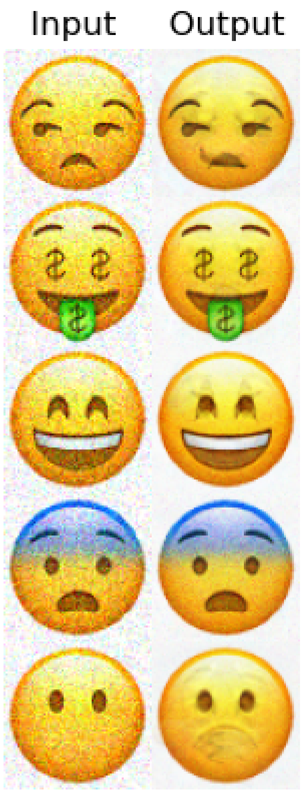


h. We found that Gaussian noise improves generalization, as neural networks trained without this augmentation tended to either predict the average, or a small class of averages. It's important to point out that the final AE performs denoising, as shown in the diagram below.

2. **Question 2.**
   a. We decided to split the 90 distinct circular face emojis into yellow and non-yellow face emojis. We included the scared/embarrassed emojis as part of the non-yellow emojis, as they have a portion of their face colored blue. There were 11 non-yellow emojis and the remaining 79 were yellow. We conducted data augmentation on both classes to the same degree, creating 10 new emojis each with Gaussian noise perturbations.
   b. We added a one-hot encoding for each class, and used binary cross entropy as our auxiliary loss function. The two output prediction neurons are the last two neurons on the final layer of the AE, and the loss is split between the image reconstruction loss and 0.5*the classification loss.
   c. View plot below.



d. See diagram below.

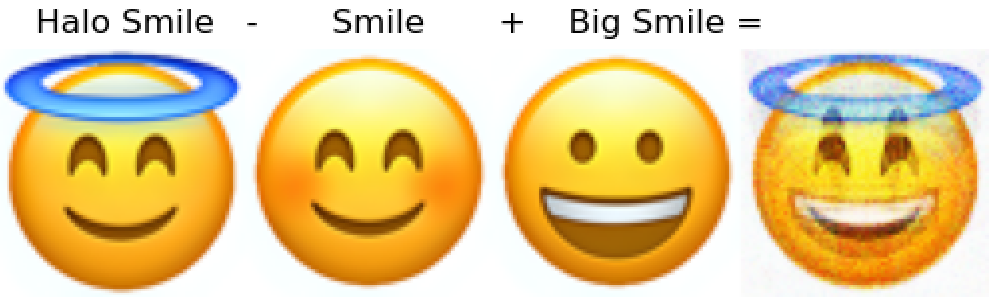


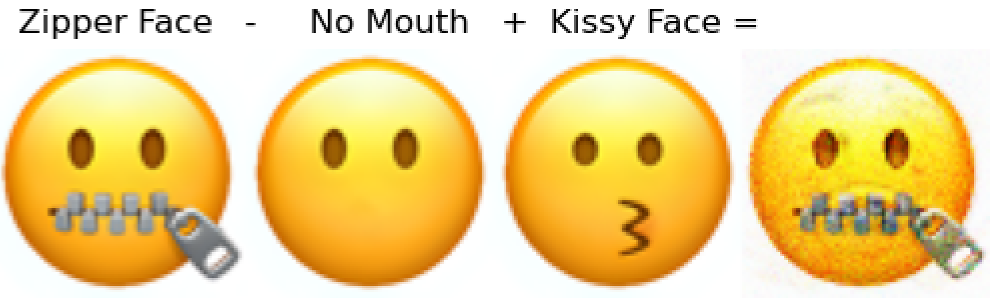
e. We arrive at a similar total loss to the non-conditional AE, as the AE learns quickly to predict yellow vs. non-yellow emojis perfectly before the second epoch. This leads to a slightly slower overall learning curve, as the network also seeks to classify the reconstructed images. Overall, the final performance is very similar under

f. Performance likely drops slightly because we only added a few extra neurons to complete an entirely different task. The neurons before the last layer (the unchanged ones compared to the Question 1 network) must work to both classify and reconstruct the image. An experiment to confirm this would include making a network that adds additional neurons to each layer in a comparable way, to see if this allows for classification and reconstruction to interfere less with each other.

3. **Question 3.**

   a. We chose to try two different attributes, adding a halo from the Halo Smile emoji, and adding a zipper mouth from the Zipper Face emoji. The vector math between the latent representations of the emojis are shown as the title of the image.





   b. Both generated emojis clearly exhibit the added attribute. Both generated emojis have a grayer background, which is an artifact from the data augmentation process. It also appears the saturation is higher in both generated images, and the team is unsure why this is. Finally, both generated examples show some level of blurriness, as the encoder confounds the direction with other learned features within the latent space.

   c. Including more augmented datapoints, as well as a larger latent space may improve the confounding between attribute directions.