Winged Horses with an Autoencoder

Anonymous author

Abstract

This paper proposes using an autoencoder to generate images that look like a Pegasus. I train an autoencoder, and condition it on birds and horses, identified by the annotated class labels for each image. This gives two latent codes via the encoder network which I linearly interpolate between to give the final outputs via the decoder network. This abstract should be short and concise, about 8-10 lines long.

1 Methodology

The method is to train an autoencoder [1], by minimising the squared L2 loss:

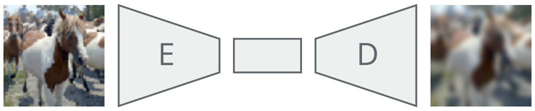
$$\mathcal{L}_{\text{AE}} = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\,\|\boldsymbol{x} - D(E(\boldsymbol{x}))\|^2] \tag{1}$$

The encoder function $E : \mathbb{R}^n \to \mathbb{R}^m$ compresses the dimensionality of the data $n << m$ to a latent encoding $z = E(x)$, which is then recovered by the decoder $D : \mathbb{R}^m \to \mathbb{R}^n$, where $\hat{x} = D(z)$.

The autoencoder is conditioned on birds and horses which are identified by the annotated class labels available for the images. This gives two latent codes via the encoder network which are linearly interpolated between to give the final outputs via the decoder network.

I use 3 channels and a latent size of 512 and a learning rate of 0.001 in the optimiser with 20 epochs.

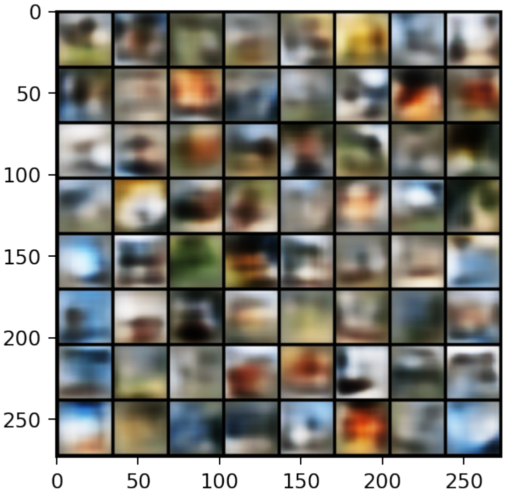
The autoencoder can be visualised as below:



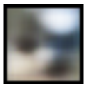
Where the neural network is trained to predict its inputs, thus learning an identity which it can reproduce as its output. The architectural diagram above was created using Inkscape and exported to a PDF.

2 Results

The results look very blurry, where the best batch of images looks like this:



From this batch, the most Pegasus-like image (with quite a stretch of the imagination) is:



3 Limitations

It's very difficult to see anything that looks like a Pegasus. In the future, this could be improved by training for more than 10 epochs, although this was not possible due to the time constraints.

Bonuses

This submission has a total bonus of +3 marks, as it is trained with STL-10 at the full 96x96 pixels, and the Pegasus has a dark body colour.

References

[1] Mark A Kramer. "Nonlinear principal component analysis using autoassociative neural networks". In: *AIChE journal* 37.2 (1991), pp. 233–243.