Score!

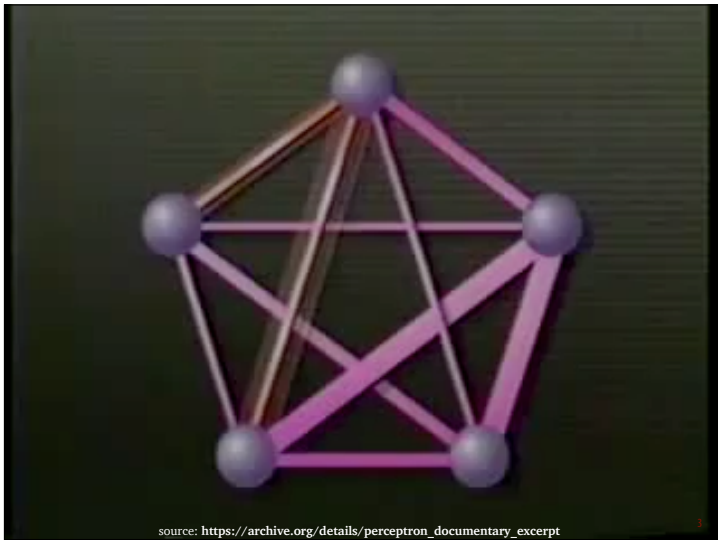
Can automatic music generation assist audio-visual artists?

Peter Bloem, Gregory Markus, Mailin Chen, Victor de Boer

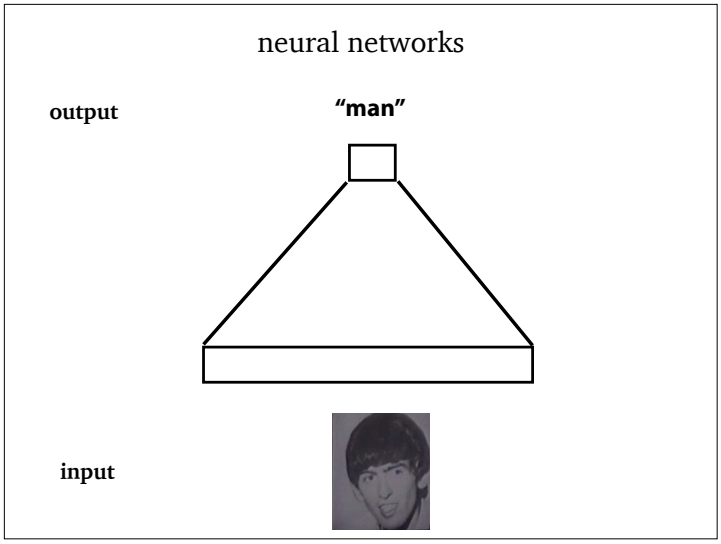
preliminaries

- Neural networks
- Generator networks
- Variational autoencoders

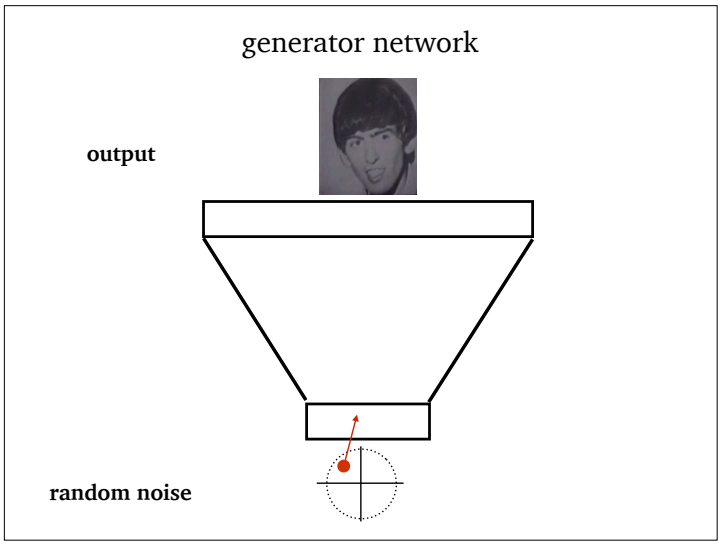
To explain the basics of SCORE!, we'll need a few preliminaries. We'll go though them as quickly as possible.



This video illustrates the basic idea of a neural network feed the computer the examples one by one, and tell it the target value for each.

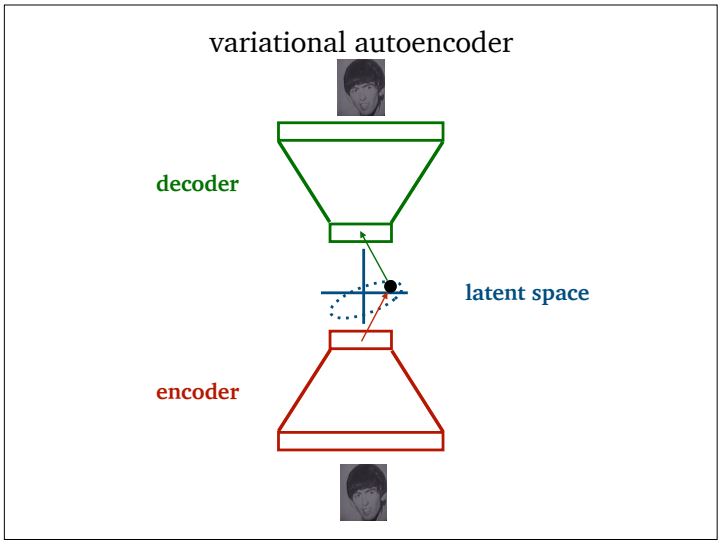


That’s all we need to know about neural networks. They are functions from an input (like an image) to an output (like a classification), and if we have a lot of examples of particular inputs that should lead to particular outputs, we can *train* the network to fit the data.



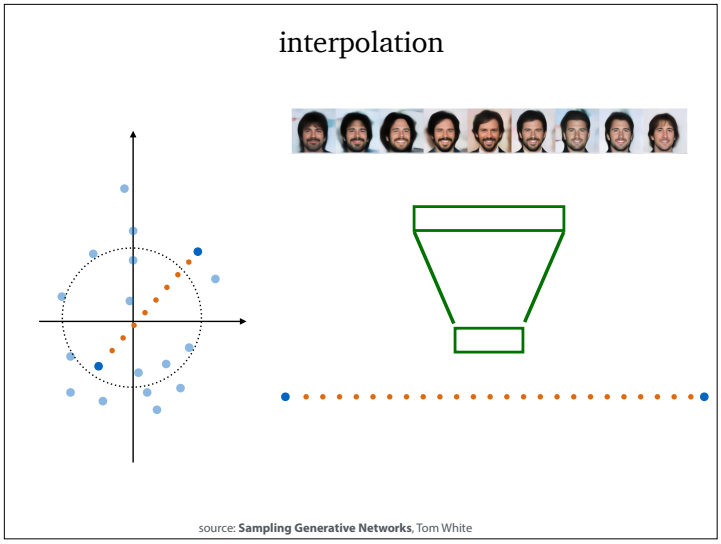
If we want to use the neural network to generate data, that is, to behave like a probability distribution, we can feed it random noise. This gives us a very power probability model: with the right parameters, the network can potentially generate anything from faces to buildings to landscapes.

But now we have a problem: we may have some outputs that we want to reproduce, but we have no idea what inputs should map to which outputs.

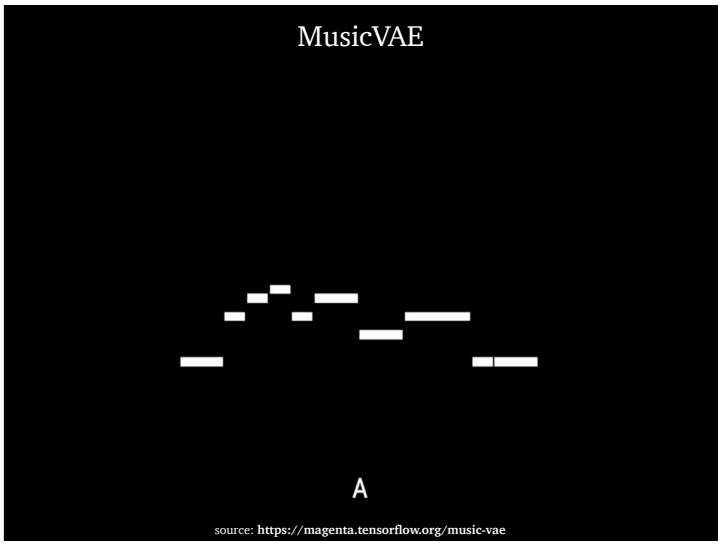


To fix this problem, we simply learn two functions. The first maps an input to random noise, and the second maps the random noise back to the input.

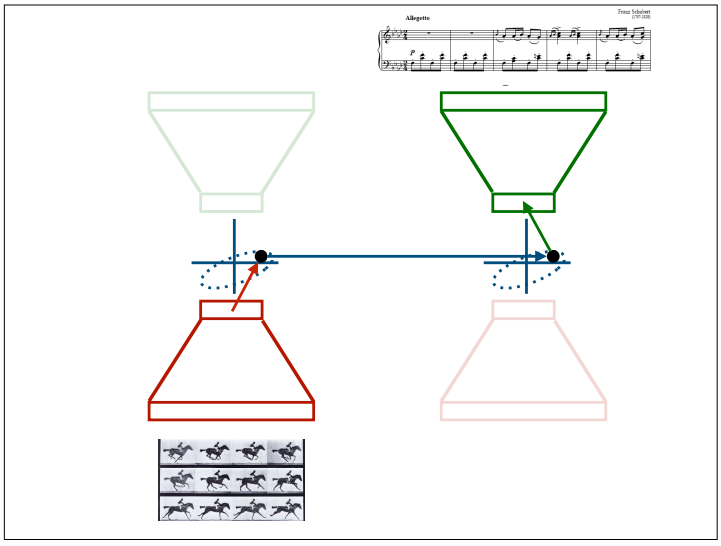
We call the first the encoder, and the second the decoder. The input and output are now the same, and we can just train the network to reproduce the input. The “random noise” now becomes a latent representation of the image. Once the model is trained, we can sample some new random noise, and feed to the decoder to generate a random face.



If we take two points in the latent space, and draw a line between them, we can pick evenly spaced points on that line and decode them. If the generator is good, this should give us a smooth transition from one point to the other, and each point should result in a convincing example of our output domain.



We can do this for music just as well as for images. Here is a model, made by Google Brain's magenta project. It's an autoencoderfor small two-bar segments of monophonic music. Segment A and B are taken from the data, and the video shows a smooth interpolation between the two.



The idea of score is to see what happens when we take two these latent space models, and combine them. Since both models are variational autoencoders, the distribution on the latent space is the same.

This mapping creates a high-level, semantic mapping between audio and video. We don't know what visual features will be mapped to what musical features, but changes in the video should be reflected in the music.

https://www.dropbox.com/s/lvv3mgkilg6h9f4/pencils.mp4?dl=0

https://www.dropbox.com/s/8xpdojz9q8crvut/racing.mp4?dl=0

https://www.dropbox.com/s/v65hc9vomy96voa/sjoelen.mp4?dl=0

future work

- Stronger musical modelling: transformer layers
- More long-term coherence: autoregressive sampling
- More short-term response: latent sequence instead of latent code.
- Better integration with AV tools (work in progress)
- Meaningful semantic annotation.