AR Models, VAEs, and GANs

Module 1.2, CV: Generative Models

Overview (ML Domains presentation last semester)

Autoregressive (AR) models

- Calculate the likelihood of each pixel given all the previous ones
- Generally uses language models

$$\underbrace{p(x)}_{\text{Likelihood of image } \boldsymbol{x}} = \prod_{i=1}^{n} \underbrace{p(x_i | x_1, \ldots, x_{i-1})}_{\text{Probability of i'th pixel value given all previous pixels}}$$

Variational Autoencoders (VAEs)

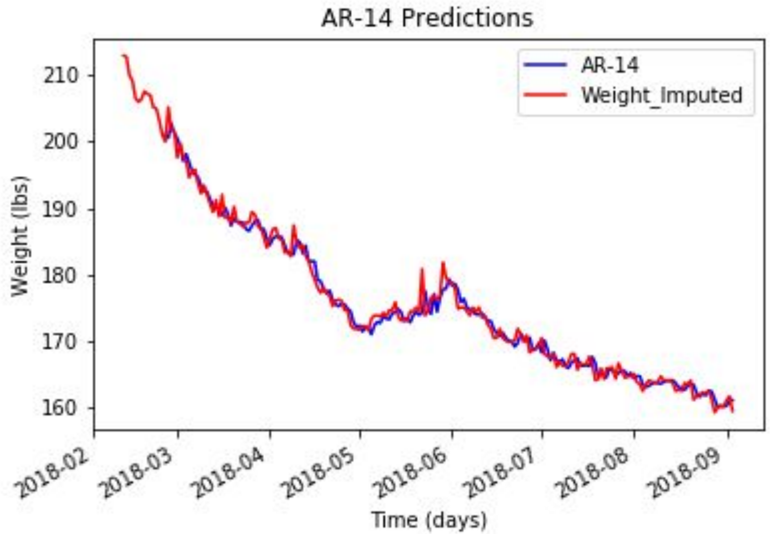
- Encoder and Decoder architecture
- Latent space
- **Intractable** probability density function

Generative Adversarial Networks (GANs)

- Generator and Discriminator
- Work against one other
- Notoriously hard to train
- Produces very realistic results

AR Models

- **Overview**: time series model that uses observations from previous time steps as input to a regression equation to predict the value at the next time step.
    - It is a very simple idea that can result in accurate forecasts on a range of time series problems.



Theory

- forecast the variable of interest using a linear combination of past values of the variable
  - feed-forward model which predicts future values from past values
  - linear model, where current period values are a sum of past outcomes multiplied by a numeric factor
    - using parameterized functions to predict next pixel given all the previous ones
      - Ex. logits
- remarkably flexible at handling a wide range of different time series patterns

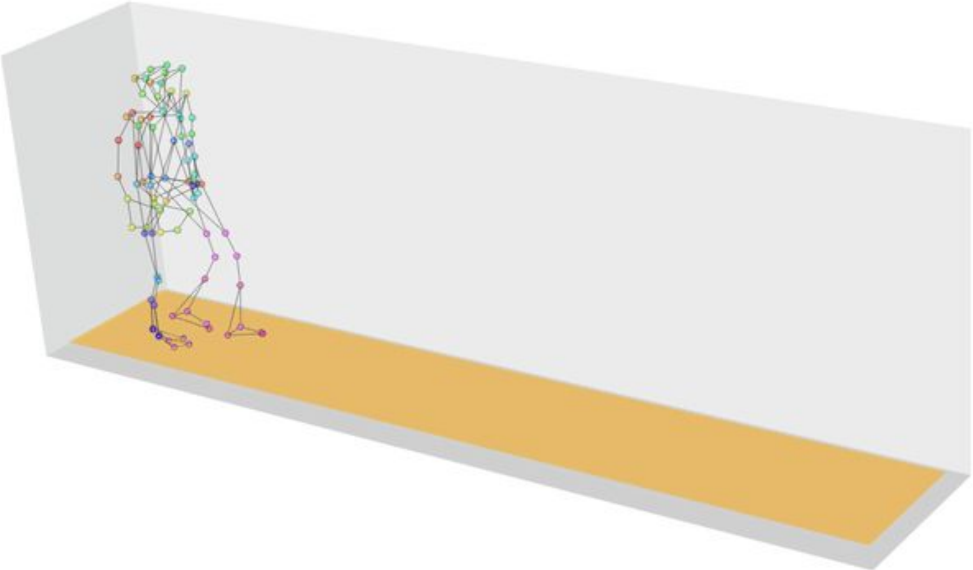
$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

$$\underbrace{p(x)}_{\text{Likelihood of image } \boldsymbol{x}} = \prod_{i=1}^{n} \underbrace{p(x_i | x_1, \ldots, x_{i-1})}_{\text{Probability of i'th pixel value given all previous pixels}}$$

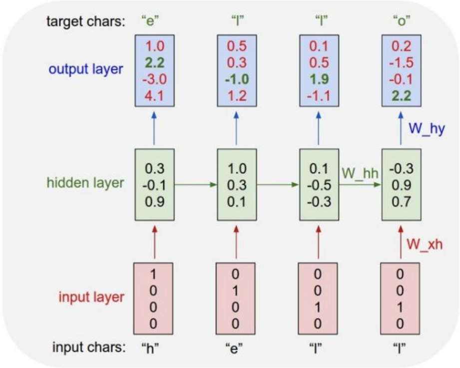
$$\operatorname{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

Applications

- Modeling video
- Dynamic time warping
- Image prediction/generation
- Image restoration
- Neural machine translation in linear time



Example: Character RNN (from Andrej Karpathy)

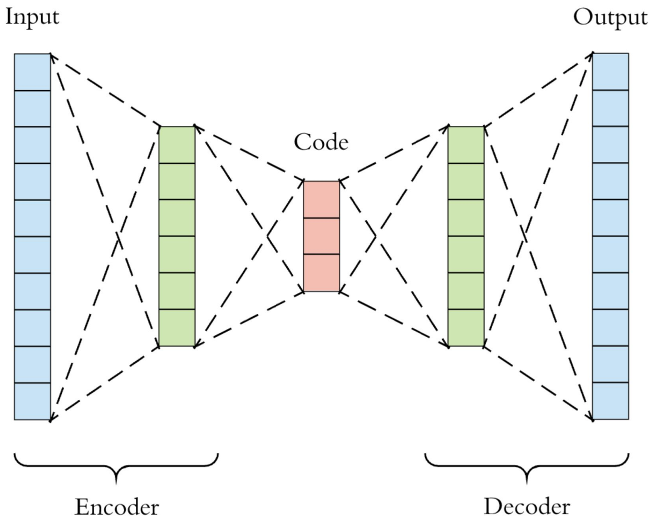


1. Suppose $x_i \in \{h, e, l, o\}$. Use one-hot encoding:
   - h encoded as $[1, 0, 0, 0]$, e encoded as $[0, 1, 0, 0]$, etc.
2. **Autoregressive**: $p(x = hello) = p(x_1 = h)p(x_2 = e|x_1 = h)p(x_3 = l|x_1 = h, x_2 = e) \cdots p(x_5 = o|x_1 = h, x_2 = e, x_3 = l, x_4 = l)$
3. For example,

$$\begin{aligned} p(x_2 = e|x_1 = h) &= softmax(o_1) = \frac{\exp(2.2)}{\exp(1.0) + \cdots + \exp(4.1)} \\ o_1 &= W_{hy} h_1 \\ h_1 &= tanh(W_{hh} h_0 + W_{xh} x_1) \end{aligned}$$

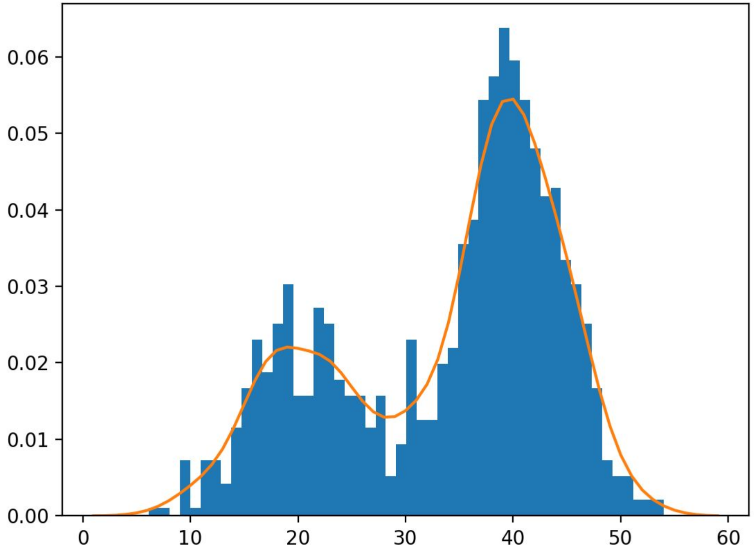
Encoder-Decoder Architectures

- Autoencoders **unsupervisedly learn efficient data encodings**
- Encoder: **learns a new, lower-dimension representation** of the input, which can then undergo modification
    - Latent space
    - Where algorithms operate
- Decoder: **reconstructs** an object of the same type as the input from that encoded representation



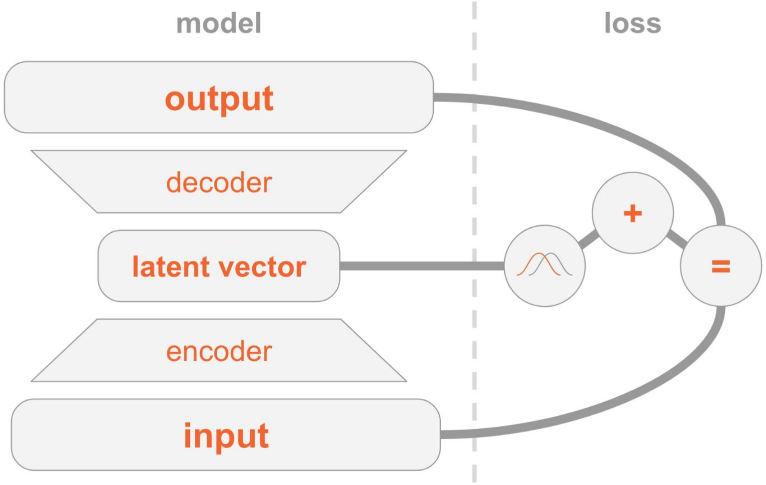
VAEs: Probability Density Function

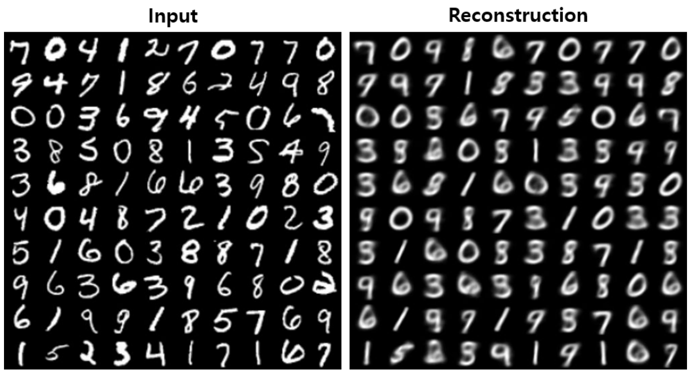
- Tool used by ML algorithms trained to **calculate probabilities from continuous random variables**
  - relationship between observation/model outcome and its probability
  - know whether a given observation is unlikely
- **Intractable**
  - problems for which there exist no efficient algorithms to solve them
  - Estimated (i.e. Monte Carlo function)



Theory

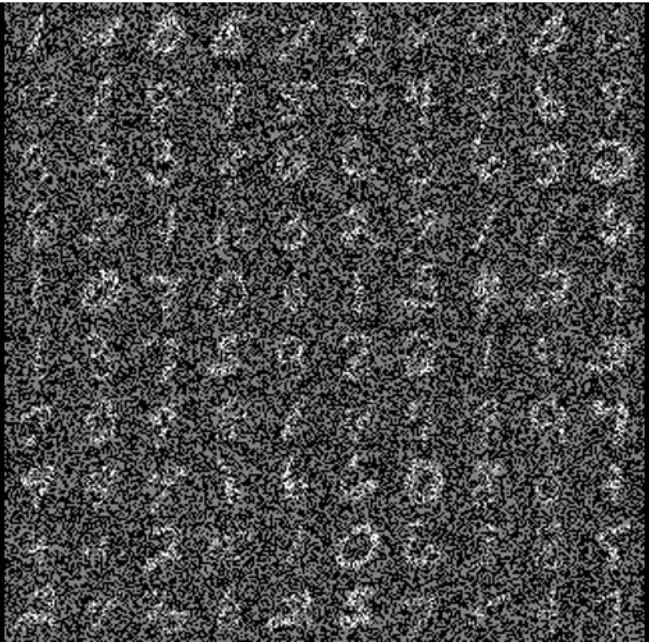
- **Learn the PDF of the training data**
  - i.e. high probability value assigned to image of a human, low probability value assigned to random/Gaussian noise
- Overcomes hurdle of pixel dependency
  - Creates the latent space for each image rather than sampling each pixel independently
  - R[k] where each vector contains k features needed to draw an image
- **Sample examples from the learned PDF**
  - Generate new examples that look similar to the original dataset





Original input image

Input image with noise



Restored image via VAE

Why not just autoencoders?

- VAEs are specifically equipped to handle **variational inference**
  - **Cast inference as an optimization problem**
  - Problem: (1) Given an input x, the probability distribution over outputs y is too complicated to work with. Or (2) Given a training corpus x, the probability distribution over parameters y is too complicated to work with.
  - Solution: Approximate that complicated p(y | x) with a simpler distribution q(y).
- vanilla autoencoder is **not a generative model:** it does not define a distribution we can sample from to generate new data points

Frameworks (open-source)

- VAE
- VQ-VAE
  - type of variational autoencoder that uses vector quantisation to obtain a discrete latent representation
  - differs from VAEs in that the encoder network outputs discrete, rather than continuous, codes
- VQ-VAE-2
  - Increased resolution (via hierarchical multi-scale latent maps)
    - Some points that contribute to blurriness are assigned high probability in learned PDF

Applications

- **Image generation, modification, and restoration**
- **Language models**
  - Sentence interpolation
- **Semi-supervised learning**
  - approach to machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training
- **Training data generation + augmentation**
- **Medical imaging**
  - Clinical prediction
  - Mesh construction (i.e. future brain 3D mesh construction)

GANs

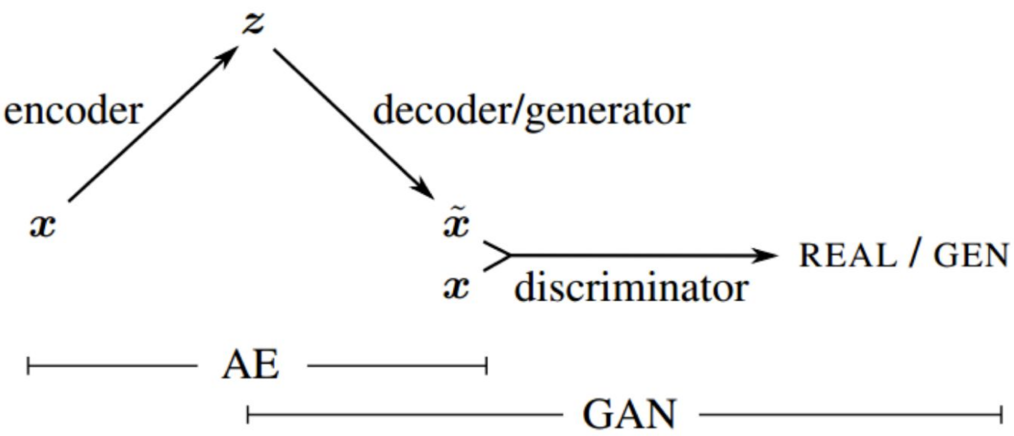
- unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset.
- Frames the problem as a supervised ML approach with two submodels, generator and discriminator
- Generator
  - generate new plausible examples from the problem domain
- Discriminator
  - classify examples as real (from the domain) or fake (generated).
- Trained adversarially - notoriously hard to train but produce very realistic results
  - whichfaceisreal.com

Generator-Discriminator architecture

- **Generator**
  - takes a fixed-length random vector as input and generates a sample in the domain.
  - vector drawn randomly from a Gaussian distribution, used to seed the generative process
  - After training, points in this multidimensional vector space will correspond to points in the problem domain, forming a compressed representation of the data distribution
  - This vector space is referred to as a latent space
- **Discriminator**
  - takes an example from the domain as input (real or generated) and predicts a binary class label of real or fake (generated)
  - Naive classification model

VAEs vs. GANs

- GANs are typically superior as deep generative models as compared to VAEs
- However, notoriously difficult to work with and require a lot of data and tuning
- Hybrid models: e.g. VAE-GAN



Next 2 weeks: Sat will be going over how to code a VAE!