Johns Hopkins Engineering

Applied Machine Learning for Mechanical Engineers

Unsupervised Machine Learning Techniques, Part 1, C

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Encoders and Generative Networks

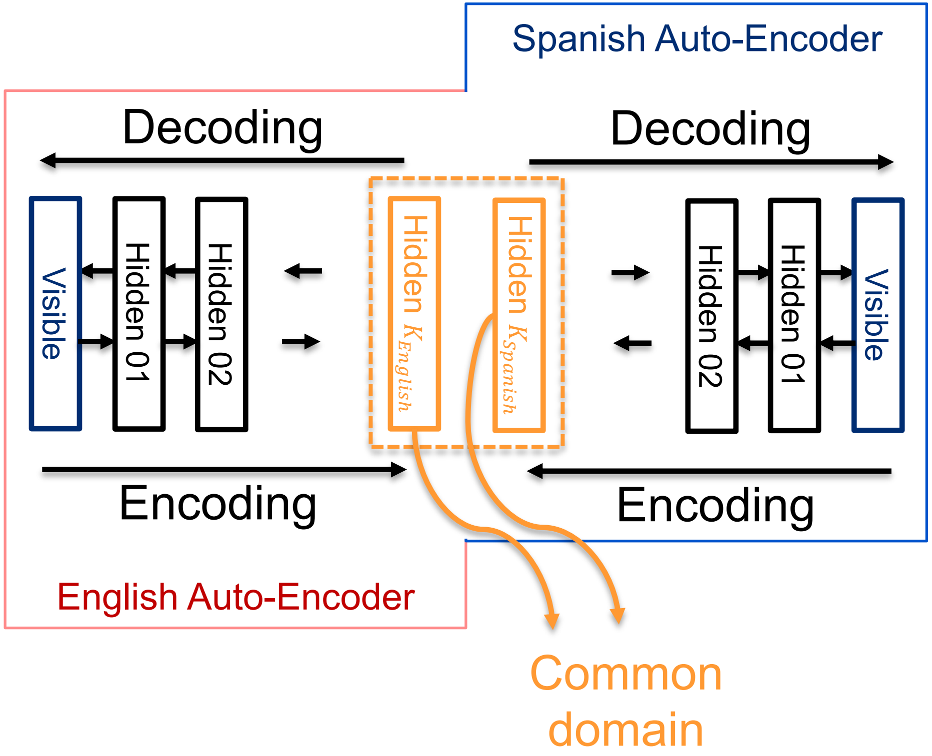
- By the end of this lecture you will be able to:
  - Describe Dual-Encoders
  - Describe Multi-Encoders
  - Describe Generative Adversarial Neural Networks (GAN)

Encoders and Generative Networks

- Dual Encoders
  - Transform data from one “domain” to another “domain” (feature space)
    - Example: English to Spanish
  - Identify a common feature space that addresses the common “concept”
    - Example: “Hello” is in English domain and “Hola” is in Spanish domain and the common concept is greeting or saluting
    - Example: A domain has a low-resolution image or video, and the other domain has a high-resolution version of that image or video. Look into the two videos available at https://arstechnica.com/science/2020/02/someone-used-neural-networks-to-upscale-a-famous-1896-video-to-4k-quality/
    - Example: A domain with noisy data and another one with denoised data

Encoders and Generative Networks

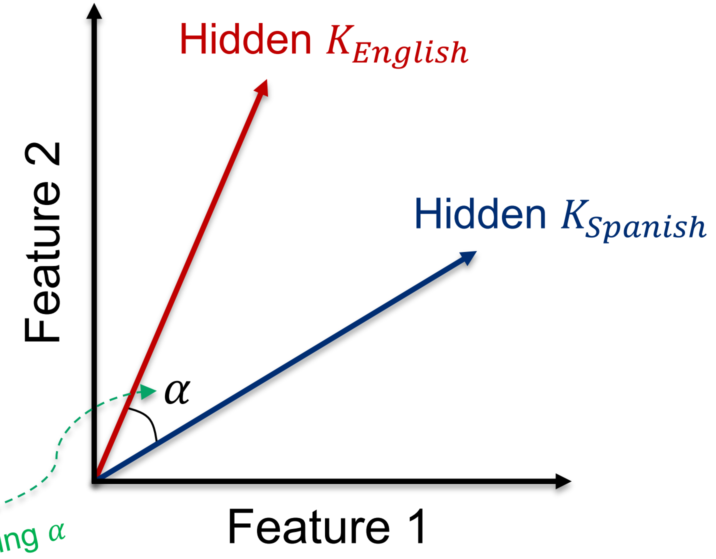
- Dual Encoders
  - Last hidden layer of domain 1 has the same number of neurons as the last hidden layer of domain 2
  - The loss function is addressing 3 things
    - Reconstruction error of domain 1
    - Reconstruction error of domain 2
    - The error between the last hidden layer of domain 1 and the last hidden layer of domain 2



Encoders and Generative Networks

- Dual Encoders
  - Last hidden layer of domain 1 has the same number of neurons as the last hidden layer of domain 2
  - The loss function is addressing 3 things
    - Reconstruction error of domain 1
    - Reconstruction error of domain 2
    - The error between the last hidden layer of domain 1 and the last hidden layer of domain 2

Minimizing α

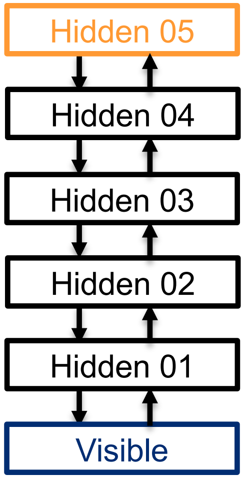


Encoders and Generative Networks

- Multi-Encoders Encoders
  - Multiple input domains (e.g., English, Spanish, Chinese, etc.)
  - Last hidden layer of each domain's auto-encoder has the same number of neurons as the last hidden layer of other domains' auto-encoder
  - The loss function is addressing 2 sets of things
    - Reconstruction error of each domain
    - The error between last hidden layer of each pairs of domains

Encoders and Generative Networks

- Generative Networks
  - Generate new input datapoints that has not been existed before
    - Benefit: enriching repository!
  - Trained auto-encoders have potential to generate new data
    - Feed the trained auto-encoder with a batch of inputs and compute the last hidden layer (encode)
    - Change the values of some neurons (like randomly or add some noise to them) and then decode to get a new reconstruction



Encoders and Generative Networks

- Generative Adversarial Networks (GAN)

  - Comprised of two machine learning parts, one is called a discriminator and the other one is called a generator

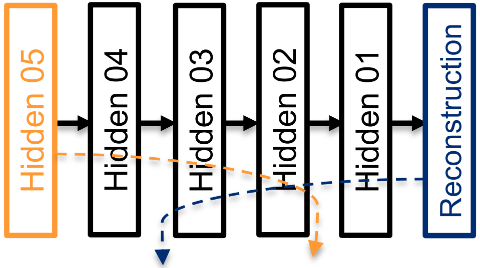
  - Discriminator, denoted as function $d(.)$, is like a binary classifier that does not classify but instead gives you the probability (a value between 0 and 1) of a set of inputs belonging to the same distribution as a repository. I mean it tells you the probability of a new unseen image belonging to an already existed repository.
  - Example: What is the probability of an image (input) to be a face (distribution of faces)?
    - Example: What is the probability of an image (input) to be a cat (distribution of cats)?
    - Example: What is the probability of a sound record (input) to belong to a dog (distribution of dogs)?
    - Example: What is the probability of a signal (input) to be noise free (distribution of noise free signals)?

  - Generator, denoted as function $g(.)$, is like a decoder generator, create new inputs by manipulating the values in the last hidden layer.

Encoders and Generative Networks

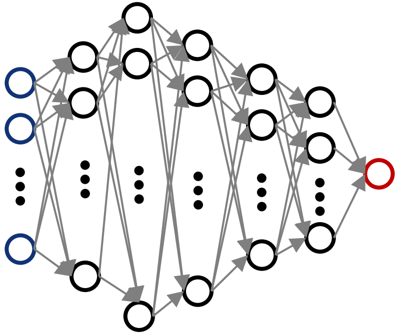
- Generative Adversarial Networks (GAN)
  - Example of cat problem:
    - Generator’s goal is to generate inputs that are as good as to be interpreted as cats by discriminator (goal is to fool the discriminator!)
    - Discriminator’s goal is not to be fooled by the generator!
    - So, the generator and discriminator are like enemies or adversaries, fighting with each other

Generator



$$\boldsymbol{X} = g(\boldsymbol{H}_5)$$

Discriminator



$\boldsymbol{X}$ is a generated input and p is the probability of that input to be in a distribution (e.g., faces)

$$p = d(\boldsymbol{X}) = d\big(g(\boldsymbol{H}_5)\big)$$

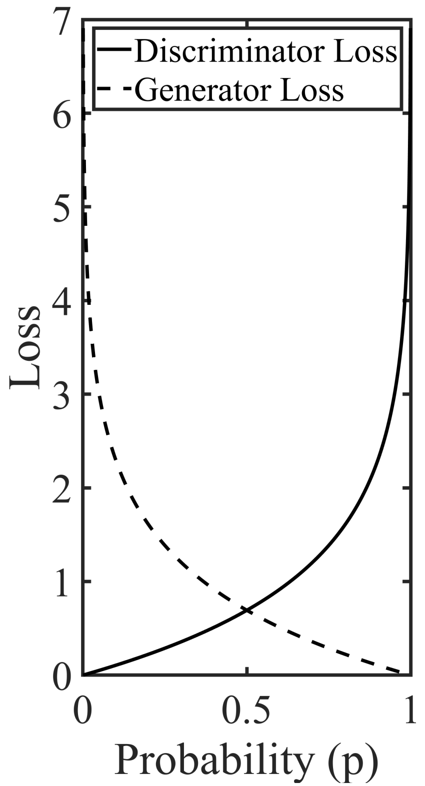
Encoders and Generative Networks

- Generative Adversarial Networks (GAN)
  - Loss functions
    - Discriminator wants to see the outcomes of the generator as fake inputs. So, it tends to produce low magnitude (0 to be ideal), probabilities (a large probability is a large error). The discriminator loss function is as follow:

$$l = -\ln(1 - p)$$

    - Generator wants the discriminator to be fooled and to see its generated inputs as right inputs (e.g., inputs that show cat faces). So, it requires the discriminator to produce high magnitude (1 to be ideal), probabilities (a low probability is a low error). The generator loss function is as follow:

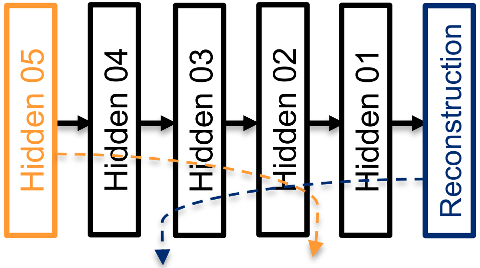
$$l = -\ln(p)$$



Encoders and Generative Networks

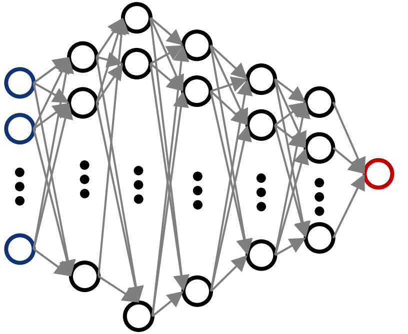
- Generative Adversarial Networks (GAN)
  - In each iteration:
    - Step 1: in the generator, generate random values of last hidden layer neurons and decode them to reconstruct an input ($\boldsymbol{X}$)
    - Step 2: feed the reconstructed input to the discriminator and compute a probability, p
    - Step 3: Compute the discriminator's and generator's loss values and update their parameters
    - Step 4: feed the discriminator with a real data (e.g., a cat's face), compute a probability p, and update the discriminator weights and biases.

Generator



$$\boldsymbol{X} = d(\boldsymbol{H}_5)$$

Discriminator



$$p = g(\boldsymbol{X}) = g\big(d(\boldsymbol{H}_5)\big)$$

Encoders and Generative Networks

- Generative Adversarial Networks (GAN)
  - These people do not exist! (generated by GAN):




https://thispersondoesnotexist.com/
https://en.wikipedia.org/wiki/StyleGAN
Must Watch!: https://www.youtube.com/watch?v=6E1_dgYlifc

Encoders and Generative Networks

- In this lecture, you learned about:
  - Dual Encoders
  - Multi-Encoders
  - Generative Adversarial Neural Networks (GAN)
- In the next Module, we will practice a number of unsupervised machine learning techniques in Python

JOHNS HOPKINS

WHITING SCHOOL *of* ENGINEERING