Data Mining & Machine Learning

CS37300
Purdue University

November 6, 2017

Kaggle Competition Update (extra credit)

Students with > 0.6 accuracy in **Public Leaderboard***

21 students so far

*Extra credit based on ***Private* Leaderboard**

Public Leaderboard | **Private Leaderboard**

This leaderboard is calculated with approximately 30% of the test data.
The final results will be based on the other 70%, so the final standings may be different.

Raw Data | Refresh

#	△1w	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲ 9	**General Grievous**			0.87980	6	4d
2	new	**Captain Rex**			0.83447	4	4d
3	▼ 2	**Revan**			0.83407	12	9d
4	▼ 2	**Luke Skywalker**			0.83286	9	4d
5	▼ 2	**Cad Bane**			0.81991	18	13d
6	▲ 6	**Count Dooku**			0.81262	3	2d
7	new	**Dengar**			0.81222	4	2d
8	new	**Darth Vader**			0.81019	10	13h
9	▼ 5	**Yoda**			0.80979	17	6d
10	▼ 2	**Bossk**			0.80291	7	1d
11	new	**Anakin Solo**			0.76851	1	2d
12	▼ 7	**Ki-Adi-Mundi**			0.76811	2	14d
13	▼ 7	**Kyp Durron**			0.73613	4	1mo
14	▼ 7	**Shaak Ti**			0.70133	2	1mo
15	▼ 6	**Admiral Thrawn**			0.65520	2	14d
16	new	**Clone Commander Cody**			0.63334	4	2d

Neural Networks - Generative Models

Overview

- Classification Tasks
- Generative Tasks
  - Boltzmann Machine
  - Restricted Boltzmann Machine (RBM)
  - Generative Adversarial Network (GANs)

Classification Task

is rich?

$y \in \{\text{True}, \text{False}\}$

is tanned

employs a captain

given to us

pays high property taxes

employs a cook

x

classifier (we learn)

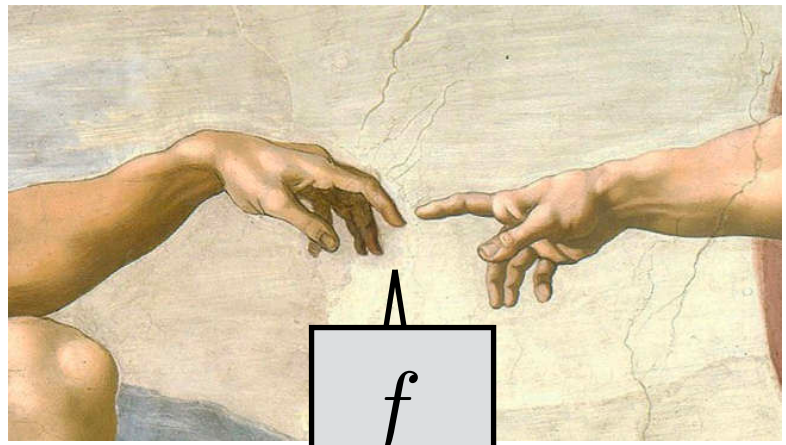
f

$\hat{y} = f(x)$

our prediction

- Classifier takes known input: x
- And outputs the most likely label: $f(x)$

Generative Task



by fiat
(out of the blue)

Learns to generate examples: (x, y)

(is tanned
employs a captain
pays high property taxes
employs a cook , is rich)

Easier and Harder Tasks

- Classification/regression/discriminative tasks are easier
  - Input is given
  - Just need to learn function (classifier) that maps input to desired (given) output
  - A.k.a. **supervised learning** (because we are given part of the answer)

- Generative tasks are harder (and closer to what we consider “intelligence”)
  - Must learn how to generate examples not given in the training data
  - A.k.a. **unsupervised learning** (because we are NOT given part of the answer)
  - Create a movie script, develop a new drug, **generate an image**, create new software,…
  - Harder to evaluate quality: how good is a generated example?
    - E.g.: are the images really generated or just rememebered from the training data?

Generative example (anonymous submission to ICLR 2018)

https://www.youtube.com/watch?v=XOxxPcy5Gr4

What is the Difference Between Classification and Generation Tasks?

(statistically)

Statistical Difference Between Classification and Generation Tasks

With probabilistic models it is easy to explain:

- **Classification task**
  - Given an example (x_i, y_i)
  - Wants to learn conditional probability $p(y_i|x_i)$
  - To use it, we need x_i and the output is the predicted class: $\hat{y}_i = \arg\max_y p(y|x_i)$

e.g.: $x_i =$  , $y_i = \text{dog}$

- **Generation task**
  - Given an example (x_i, y_i)
  - Wants to learn joint probability $p(y_i, x_i)$
  - To use it, we sample another example from $p(y, x)$, the output is an entirely new example

e.g.: $x =$  , $y = \text{dog}$

Are Generative Models More Powerful?

- Suppose we have $p(y, x)$
- Can we perform classification $\hat{y}_i = \arg\max_y p(y|x_i)$ of example x_i ?

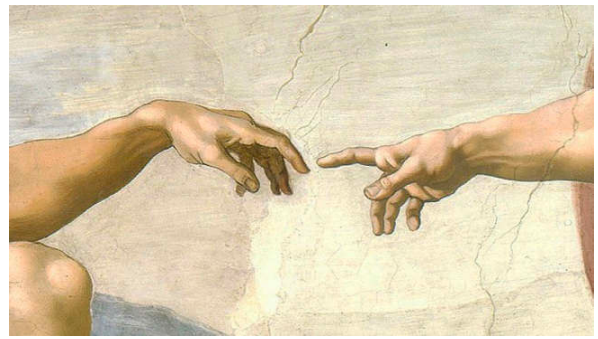
e.g.: $x_i =$  $y_i = \text{dog}$

- Yes! Using Bayes rule

$$p(y|x_i) = \frac{p(y, x_i)}{\sum_y p(y, x_i)}$$

- Generative models are more powerful in this sense… but also harder to learn
  - Because generative models are harder to learn, more specialized models (classifiers) will do better in real-world classification tasks

How Generative Models Work in Practice



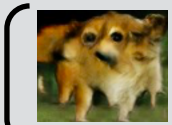
$z = \text{rand}()$

f

$f(z) = \mathbf{x}$

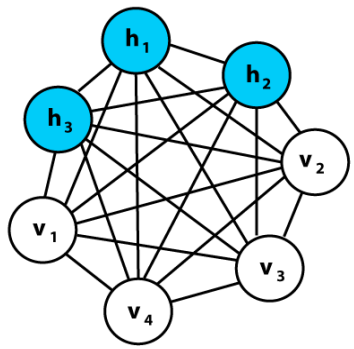
x is a vector of
whatever you want
(in our past examples it was **x**=(x,y))

- Function takes one or more pseudo-random number as input
- Outputs an example

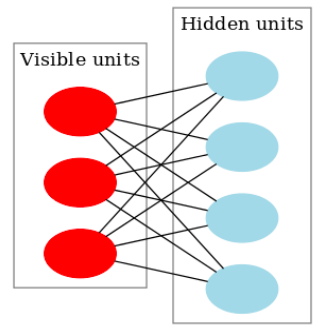
e.g.: $\mathbf{x} = \left(\text{  }, \text{dog} \right)$

Examples of Generative Neural Network Models

- This week we will see two types of generative models:
  - Boltzmann machine-type models

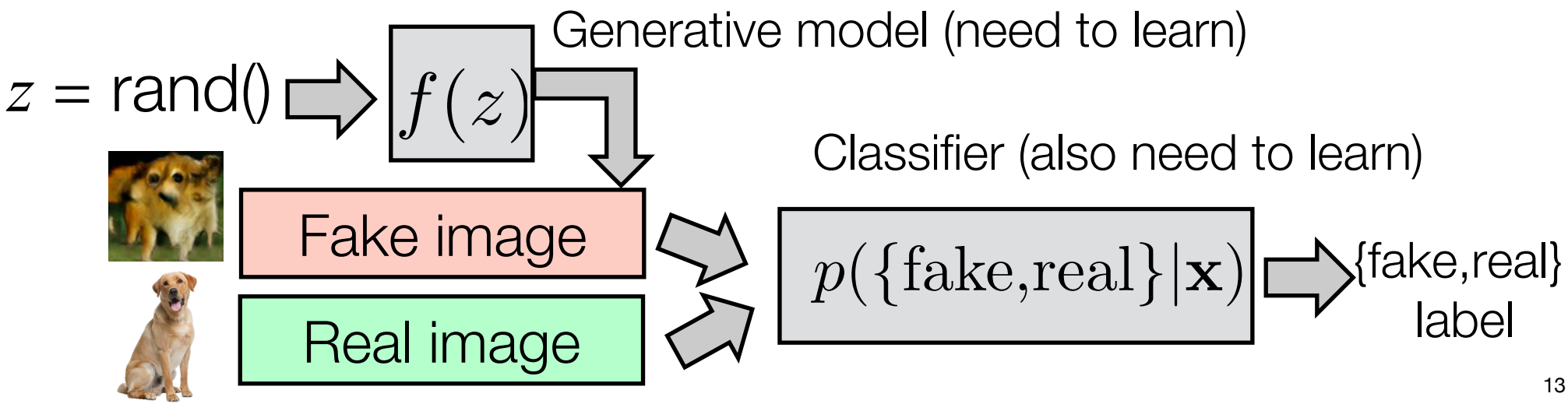


Boltzmann machine



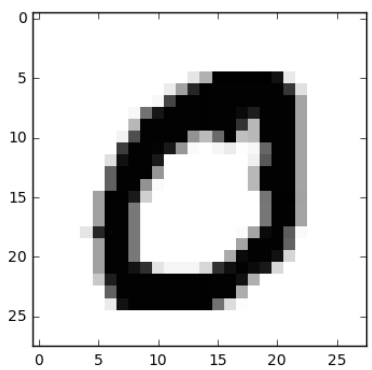
Restricted Boltzmann Machine (RBM)

  - Adversarial network-type models



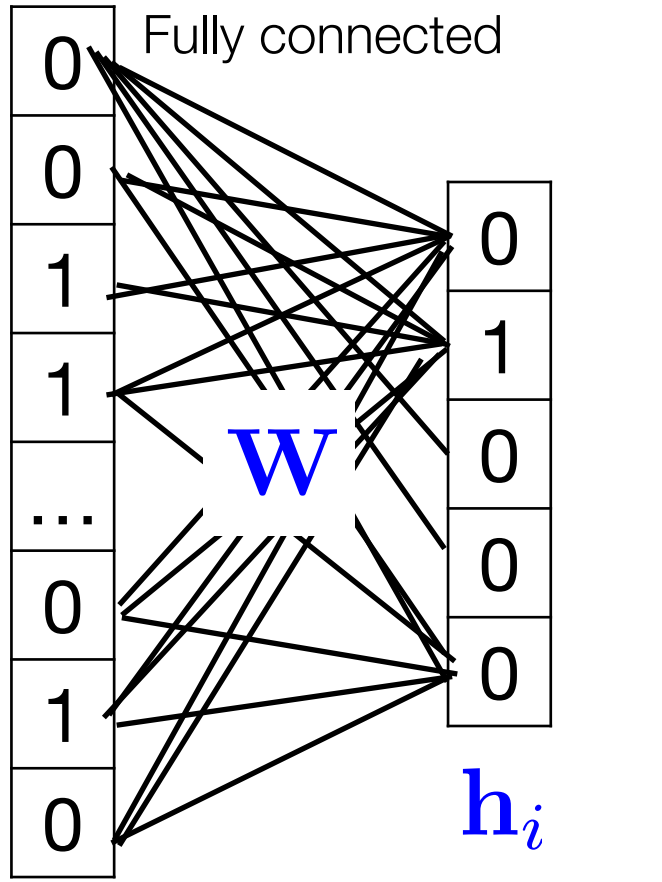
Restricted Boltzmann Machines

Restricted Boltzmann Machines



example i

vectorize



Hidden units

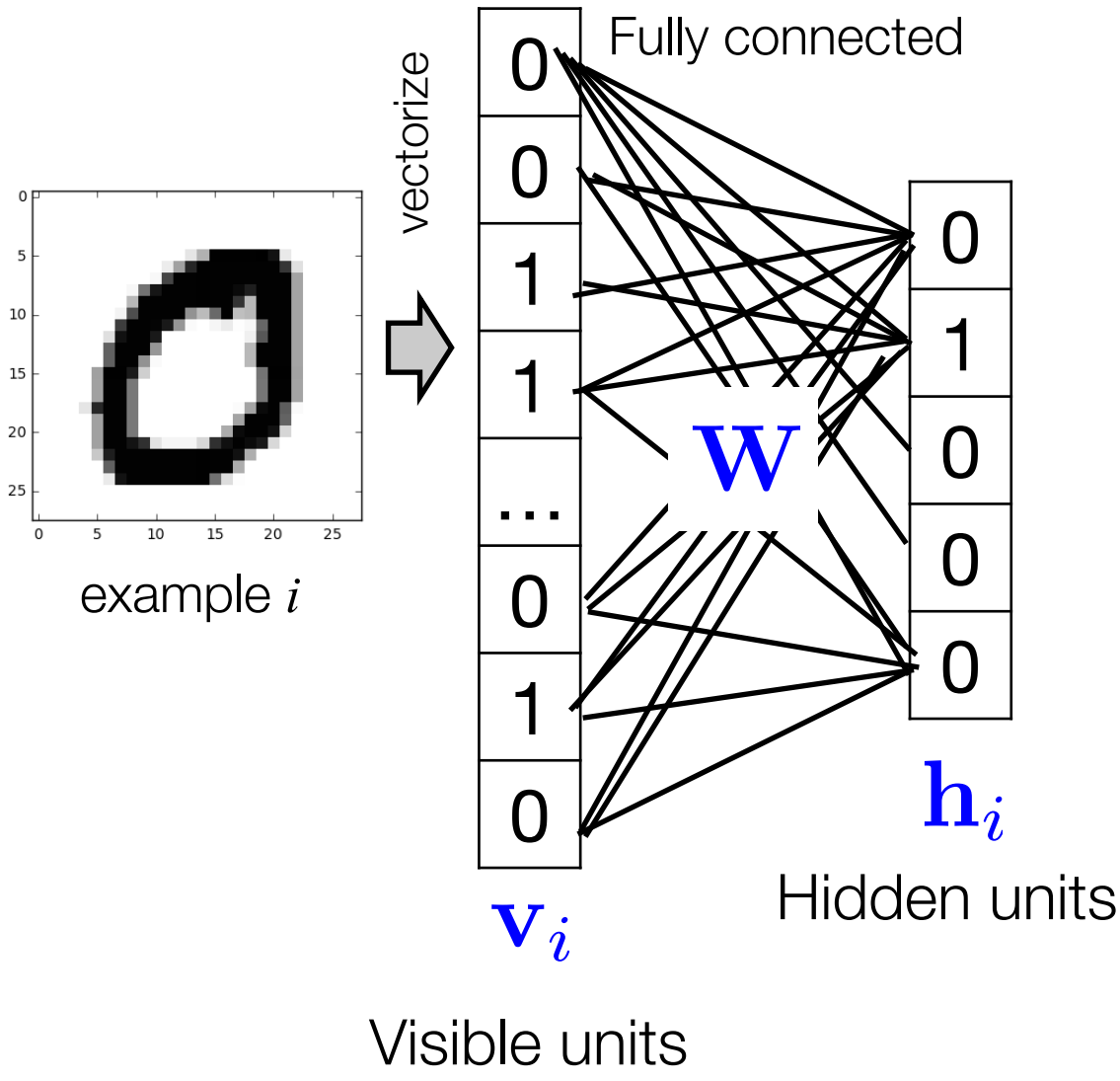
Visible units

- Need to learn a good joint probability $p(\mathbf{v}_i, \mathbf{h}_i)$
- We will define the joint probability as

$$p(\mathbf{v}_i, \mathbf{h}_i; \mathbf{W}) = \frac{\exp\left(\mathbf{v}_i \mathbf{W} \mathbf{h}_i\right)}{\sum_{\forall \mathbf{v}, \mathbf{h}} \exp\left(\mathbf{v} \mathbf{W} \mathbf{h}\right)}$$

- What is the model space?
  - i.e., what are we searching over?
- What is the score function?
- How can we do the search?

Restricted Boltzmann Machines



- Model space:
  The set of all possible joint probability distributions given by all possible weights $\mathbf{W}$

$$p(\mathbf{v}_i, \mathbf{h}_i; \mathbf{W}) = \frac{\exp\left(\mathbf{v}_i \mathbf{W} \mathbf{h}_i\right)}{\sum_{\forall \mathbf{v}, \mathbf{h}} \exp\left(\mathbf{v} \mathbf{W} \mathbf{h}\right)}$$

- What is the score function?
  - Not easy…. topic of Friday's class
- How can we do the search?
  - Not easy…. topic of Friday's class