ECE59500CV Lecture 1: Course Overview

Jeffrey Mark Siskind

School of Electrical and Computer Engineering

Fall 2020

PURDUE UNIVERSITY

Course Overview—I

In the event of a major campus emergency, course requirements, deadlines and grading percentages are subject to changes that may be necessitated by a revised semester calendar or other circumstances. Such changes will be announced to the course email mailing list.

Course Overview—II

- What course this is.
- When and where it meets.
- The course staff.
- Who am I.
- My office hours, office, email, phone.
- This is a new course; this is the first time it is being taught.

Course Overview—III

- deep learning
- segmentation
- object classification and localization
- activity classification and localization
- semantic segmentation
- depth reconstruction
- 3D reconstruction
- generative adversarial networks
- image and video captioning
- image and video retrieval

Course Overview—IV

- Course texts
- `http://engineering.purdue.edu/ece595cv`
- Attend every lecture
- Prerequisites
- Computer accounts
- Course software
- Problem sets
- Grading
- Collaboration

Course Overview—V

- Communication
  1. Class
  2. `http://engineering.purdue.edu/ece595cv`
  3. Email
     - `ece59500cv-staff-list@ecn.purdue.edu`,
     - `ece59500cv-students-list@ecn.purdue.edu`
  4. Office hours
  5. Phone
  6. Openness Policy

Much of Machine Learning is Function Optimization—I

- You have a dataset that pairs samples x_i with labels l_i.

Much of Machine Learning is Function Optimization—I

- You have a dataset that pairs samples x_i with labels l_i.
- Each sample is vector of reals.

Much of Machine Learning is Function Optimization—I

- You have a dataset that pairs samples x_i with labels l_i.
- Each sample is vector of reals.
- The label is a real.

Much of Machine Learning is Function Optimization—I

- You have a dataset that pairs samples x_i with labels l_i.
- Each sample is vector of reals.
- The label is a real.
- You have a model $l = f(x; w)$ that maps a sample x to a label l.

Much of Machine Learning is Function Optimization—I

- You have a dataset that pairs samples x_i with labels l_i.
- Each sample is vector of reals.
- The label is a real.
- You have a model $l = f(x; w)$ that maps a sample x to a label l.
- The models has a parameter vector w, sometimes called weights.

Much of Machine Learning is Function Optimization—I

- You have a dataset that pairs samples x_i with labels l_i.
- Each sample is vector of reals.
- The label is a real.
- You have a model $l = f(x; w)$ that maps a sample x to a label l.
- The models has a parameter vector w, sometimes called weights.
- You train by finding $w = \arg\min_w \sum_i L(l_i, f(x_i; w))$.

Much of Machine Learning is Function Optimization—I

- You have a dataset that pairs samples x_i with labels l_i.
- Each sample is vector of reals.
- The label is a real.
- You have a model $l = f(x; w)$ that maps a sample x to a label l.
- The models has a parameter vector w, sometimes called weights.
- You train by finding $w = \arg\min_{w} \sum_{i} L(l_i, f(x_i; w))$.
- L is called the loss (function).

Much of Machine Learning is Function Optimization—I

- You have a dataset that pairs samples x_i with labels l_i.
- Each sample is vector of reals.
- The label is a real.
- You have a model $l = f(x; w)$ that maps a sample x to a label l.
- The models has a parameter vector w, sometimes called weights.
- You train by finding $w = \arg\min_w \sum_i L(l_i, f(x_i; w))$.
- L is called the loss (function).
- You then use the trained weights for new data x by computing $f(x; w)$.

Much of Machine Learning is Function Optimization—II

- One way to train is by doing naive gradient descent.

Much of Machine Learning is Function Optimization—II

- One way to train is by doing naive gradient descent.
- Start with w_0.

Much of Machine Learning is Function Optimization—II

- One way to train is by doing naive gradient descent.
- Start with w_0.
- Iterate with $w_{i+1} = w_i - \eta \nabla_w \sum_i L(l_i, f(x_i; w))$.

Much of Machine Learning is Function Optimization—II

- One way to train is by doing naive gradient descent.
- Start with w_0.
- Iterate with $w_{i+1} = w_i - \eta\nabla_w \sum_i L(l_i, f(x_i; w))$.
- η is called the learning rate.

Much of Machine Learning is Function Optimization—II

- One way to train is by doing naive gradient descent.
- Start with w_0.
- Iterate with $w_{i+1} = w_i - \eta\nabla_w \sum_i L(l_i, f(x_i; w))$.
- η is called the learning rate.
- $\nabla_w \sum_i L(l_i, f(x_i; w)) = \sum_i \nabla_w L(l_i, f(x_i; w))$.

Much of Machine Learning is Function Optimization—II

- One way to train is by doing naive gradient descent.
- Start with w_0.
- Iterate with $w_{i+1} = w_i - \eta \nabla_w \sum_i L(l_i, f(x_i; w))$.
- η is called the learning rate.
- $\nabla_w \sum_i L(l_i, f(x_i; w)) = \sum_i \nabla_w L(l_i, f(x_i; w))$.
- Slow. $O(n)$ to compute $\sum_i \nabla_w L(l_i, f(x_i; w))$.

Much of Machine Learning is Function Optimization—II

- One way to train is by doing naive gradient descent.
- Start with w_0.
- Iterate with $w_{i+1} = w_i - \eta\nabla_w \sum_i L(l_i, f(x_i; w))$.
- η is called the learning rate.
- $\nabla_w \sum_i L(l_i, f(x_i; w)) = \sum_i \nabla_w L(l_i, f(x_i; w))$.
- Slow. $O(n)$ to compute $\sum_i \nabla_w L(l_i, f(x_i; w))$.
- Instead use stochastic gradient descent (SGD).

Much of Machine Learning is Function Optimization—II

- One way to train is by doing naive gradient descent.
- Start with w_0.
- Iterate with $w_{i+1} = w_i - \eta \nabla_w \sum_i L(l_i, f(x_i; w))$.
- η is called the learning rate.
- $\nabla_w \sum_i L(l_i, f(x_i; w)) = \sum_i \nabla_w L(l_i, f(x_i; w))$.
- Slow. $O(n)$ to compute $\sum_i \nabla_w L(l_i, f(x_i; w))$.
- Instead use stochastic gradient descent (SGD).
- Each iteration computes $\sum_i \nabla_w L(l_i, f(x_i; w))$ over only a portion of the dataset, called a batch or minibatch.

Much of Machine Learning is Function Optimization—II

- One way to train is by doing naive gradient descent.
- Start with w_0.
- Iterate with $w_{i+1} = w_i - \eta \nabla_w \sum_i L(l_i, f(x_i; w))$.
- η is called the learning rate.
- $\nabla_w \sum_i L(l_i, f(x_i; w)) = \sum_i \nabla_w L(l_i, f(x_i; w))$.
- Slow. $O(n)$ to compute $\sum_i \nabla_w L(l_i, f(x_i; w))$.
- Instead use stochastic gradient descent (SGD).
- Each iteration computes $\sum_i \nabla_w L(l_i, f(x_i; w))$ over only a portion of the dataset, called a batch or minibatch.
- Requires computing ∇.

Issues

- Not all machine learning is function optimization.

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).
- Sometimes the labels aren't reals.

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).
- Sometimes the labels aren't reals.
- What is a good model?

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).
- Sometimes the labels aren't reals.
- What is a good model?
- $f = f_k \circ \cdots \circ f_1$

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).
- Sometimes the labels aren't reals.
- What is a good model?
- $f = f_k \circ \cdots \circ f_1$
- What are the good components f_j?

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).
- Sometimes the labels aren't reals.
- What is a good model?
- $f = f_k \circ \cdots \circ f_1$
- What are the good components f_j?
- The weights might not be a vector.

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).
- Sometimes the labels aren't reals.
- What is a good model?
- $f = f_k \circ \cdots \circ f_1$
- What are the good components f_j?
- The weights might not be a vector.
- What is a good loss (function)?

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).
- Sometimes the labels aren't reals.
- What is a good model?
- $f = f_k \circ \cdots \circ f_1$
- What are the good components f_j?
- The weights might not be a vector.
- What is a good loss (function)?
- There are better optimization procedures than SGD.

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).
- Sometimes the labels aren't reals.
- What is a good model?
- $f = f_k \circ \cdots \circ f_1$
- What are the good components f_j?
- The weights might not be a vector.
- What is a good loss (function)?
- There are better optimization procedures than SGD.
- What is a good learning rate?

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).
- Sometimes the labels aren't reals.
- What is a good model?
- $f = f_k \circ \cdots \circ f_1$
- What are the good components f_j?
- The weights might not be a vector.
- What is a good loss (function)?
- There are better optimization procedures than SGD.
- What is a good learning rate?
- What is a good way to select (the size of) batches?

Issues

- Not all machine learning is function optimization.
- Sometimes the samples aren't vectors (of reals).
- Sometimes the labels aren't reals.
- What is a good model?
- $f = f_k \circ \cdots \circ f_1$
- What are the good components f_j?
- The weights might not be a vector.
- What is a good loss (function)?
- There are better optimization procedures than SGD.
- What is a good learning rate?
- What is a good way to select (the size of) batches?
- What is a good way to compute ∇?

Derivatives

- In Calculus 101 you learned how to differentiate functions.

Derivatives

- In Calculus 101 you learned how to differentiate functions.
- Sum rule.

Derivatives

- In Calculus 101 you learned how to differentiate functions.
- Sum rule.
- Product rule.

Derivatives

- In Calculus 101 you learned how to differentiate functions.
- Sum rule.
- Product rule.
- Quotient rule.

Derivatives

- In Calculus 101 you learned how to differentiate functions.
- Sum rule.
- Product rule.
- Quotient rule.
- Polynomial rule.

Derivatives

- In Calculus 101 you learned how to differentiate functions.
- Sum rule.
- Product rule.
- Quotient rule.
- Polynomial rule.
- Rules for $\sqrt{\cdot}$, $\exp$, $\log$, $\sin$, $\tan^{-1}$, $\ldots$

Derivatives

- In Calculus 101 you learned how to differentiate functions.
- Sum rule.
- Product rule.
- Quotient rule.
- Polynomial rule.
- Rules for $\sqrt{\cdot}$, $\exp$, $\log$, $\sin$, $\tan^{-1}$, $\ldots$
- Chain rule.

Derivatives

- In Calculus 101 you learned how to differentiate functions.
- Sum rule.
- Product rule.
- Quotient rule.
- Polynomial rule.
- Rules for $\sqrt{\cdot}$, $\exp$, $\log$, $\sin$, $\tan^{-1}$, $\ldots$
- Chain rule.
- That worked for expressions you wrote by hand.

Derivatives

- In Calculus 101 you learned how to differentiate functions.
- Sum rule.
- Product rule.
- Quotient rule.
- Polynomial rule.
- Rules for $\sqrt{\cdot}$, $\exp$, $\log$, $\sin$, $\tan^{-1}$, $\ldots$
- Chain rule.
- That worked for expressions you wrote by hand.
- But not for computer programs.