Inference and Applications
Recap: Recurrent Neural Networks

• Take five minutes to draw
Important Dates

• Project rough drafts are due next Tuesday
  • These should be nearly complete (coding, analysis, writing, etc.)

• No questionnaire for this week, but I’ve posted next week’s

• Final peer reviews are due Tue May 2

• Final drafts are due **Tue May 2 for seniors** and **Mon May 8 for everyone else**
  • Your group can get creative in your submissions if you have seniors and non-seniors

• All assignments, questionnaires, milestones, etc. must be completed by Tue May 2 (I will no longer accept late submissions after that date)
SpeakUp

Join by URL: https://web.speakup.info/room/join/80740

Join by key: web.speakup.info → use key 80740

Join by saved session: web.speakup.info

Join by QR:
Outline

• The machine learning pipeline

• Some terminology

• Web applications

• Demo
Machine Learning Pipeline

Software Engineering for Machine Learning: A Case Study

Saleema Amershi  
Microsoft Research  
Redmond, WA USA  
samershi@microsoft.com andrew.begel@microsoft.com

Andrew Begel  
Microsoft Research  
Redmond, WA USA  
bird@microsoft.com

Christian Bird  
Microsoft Research  
Redmond, WA USA  
cbird@microsoft.com

Robert DeLine  
Microsoft Research  
Redmond, WA USA  
rdelee@microsoft.com

Harald Gall  
University of Zurich  
Zurich, Switzerland  
gall@ifi.uzh.ch

Ece Kamar  
Microsoft Research  
Redmond, WA USA  
eckamar@microsoft.com

Nachiappan Nagappan  
Microsoft Research  
Redmond, WA USA  
nachin@microsoft.com

Besmira Nushi  
Microsoft Research  
Redmond, WA USA  
besmira.nushi@microsoft.com

Thomas Zimmermann  
Microsoft Research  
Redmond, WA USA  
tzimmer@microsoft.com
Machine Learning Pipeline

Ethical sweeps...

https://fullstackdeeplearning.com/
Terminology

• Data cleaning: removing bad data; replacing missing data, etc.
• Feature engineering: manually creating new features
• Training: learning good model parameters
• Hyperparameter tuning: learning good training and architectural parameters
• Inference: using a trained model
• Evaluation: evaluating a trained model
• Application: software enabling easier use of a model
• Deploying/production: making a trained model accessible
• Monitoring: saving and analyzing model usage
• ML Operations: making the above things easier, faster, cheaper
import gradio as gr
from fastai.vision.all import *

# Load the trained model
path = Path(sys.argv[1])
model = load_learner(path)

def classify(img):
    prediction = model.predict(img)
labeled, label_index, probabilities = prediction
label_prob = probabilities[label_index].item()

    return {
        "Frank": label_prob if labeled == "Frank" else 1 - label_prob,
        "Frary": label_prob if labeled == "Frary" else 1 - label_prob,
    }

title = "Frank or Frary? I'll Decide!"
website = "A demo for [CS 152](https://cs.pomona.edu/classes/cs152/)

iface = gr.Interface(
    fn=classify,
    inputs="image",
    outputs="label",
    title=title,
    article=website,
    ).launch()