A running example

Training Data → Food Image Classifier → Mobile App
Basic steps to build an ML system
The steps overview

• Step 1: collect data
• Step 2: look at your data
• Step 3: Create train/dev/test splits
• Step 4: build model
• Step 5: Evaluate your model
• Step 6: Diagnose error and repeat
Acquire and annotate data
Data should be **diverse**

(annotation can be expensive)
Data should be **realistic**

Ideal data sampled from the distribution your product will be run on.

Real photo taken by users  Professional ads photo
Look at your data.
Look at your data.

• You have some food images, take a closer look at them!

• Food from Europe different than from Africa? from Asia?

• Any potential bias in your data?

• Have the right people look at your data.

• Do this at every stage!
Expertise sometimes can be required

- Biomedical imaging annotation can be expensive
- Professionally trained radiologists
- Domain knowledge
Train/Dev/Test Split
Partitioning Data: Train, Test, and Validation

(1) Fit model to the training dataset

(2) Fit hyperparameters to the validation (or development) dataset

(3) Test model performance on the test set

Slides credit: Chris Ré, Stanford CS229
What makes a good split?

- **Ideal**: Train, test, & dev randomly sampled
  - Allows us to say train quality is approximately test quality

- **Test** is a **proxy** for the real world!
  - We’ll talk more about this later...

- **Challenge**: Leakage.
  - (Nearly) same example in train and dev.
  - Causes performance to be overstated!
    - Eg., same senders in train and test?
Build your model.
Build your model.

- A bag of learning algorithms learned from class.
- Simple model vs. deep models
Underfitting

Overfitting

Image credit: hackernoon.com
Model Capacity

• The ability to fit variety of functions
• Low capacity models struggles to fit training set
  • Underfitting
• High capacity models can memorize the training set
  • Overfitting
# Underfitting and Overfitting

<table>
<thead>
<tr>
<th>Model capacity</th>
<th>Data complexity</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Simple</td>
<td>Normal</td>
</tr>
<tr>
<td>High</td>
<td>Overfitting</td>
<td>Normal</td>
</tr>
</tbody>
</table>
Data Complexity

• Multiple factors matters
  • # of examples
  • # of features in each example
  • time/space structure
  • # of labels
Ablation studies.

• You’ve built up a model, it has many different components.
  • Which matter?
  • which are stable?

• Remove one feature at a time!
  • Adding features + baseline could overestimate overlap. How?

• Measure performance.
  • Critical for research!
Diagnose the error

(inspect the data where the model makes mistakes)
Open-world Machine Learning
**Closed-world**: Training and testing distributions match

**Open-world**: Training and testing distributions differ, unknowns can emerge
Deep Networks Do Not Necessarily Know What They Don’t Know…

Model trained on BDD dataset produces overconfident predictions for unknown object “helicopter”

Out-of-distribution Detection

[Diagram showing nodes connected to 'Car' and 'Truck' categories]
Out-of-distribution Detection: A Simple View

Closed-world

Input space: $\mathcal{X} = \mathbb{R}^d$
Label space: $y = \{1, -1\}$
Out-of-distribution Detection: A Simple View

**Closed-world**

\[ y = 1 \]

\[ y = -1 \]

\[ p^{in}(x) \]

**Open-world**

\[ y \notin \{+1, -1\} \]

Unknown class from out-of-distribution data

Out-of-distribution Detection: A Simple View
Out-of-distribution Detection: A Simple View

CIFAR-10 (in-distribution)

SVHN (OOD)
Out-of-distribution Detection: A Simple View

CIFAR-10

The Internet

Slide from OpenAI
The steps overview

- Step 1: collect data
- Step 2: look at your data
- Step 3: Create train/dev/test splits
- Step 4: build model
- Step 5: Evaluate your model
- Step 6: Diagnose error and repeat
Industry-scale Machine Learning
Model Complexity Keeps Increasing

LeNet (Lecun et al. 1998)

ResNet (He et al. 2016)
[Sun et al. 2017]
Challenge: Limited labeled data

ImageNet, 1M images \times 1000 \approx \text{thousand annotation hours} \quad 1B images \approx \text{million annotation hours}

[Deng et al. 2009]
Levels of Supervision

- Fully Supervised
- Weakly Supervised
- Un-supervised

TRAINING AT SCALE

- ImageNet
- Instagram/Flickr
- Crawled web images

A CUTE CAT COUPLE

FLOOR
Noisy Data

TRAINING AT SCALE
Weakly Supervised Training

- 3.5B PUBLIC INSTAGRAM IMAGES
- LARGE CAPACITY MODEL (RESNEXT101-32X48)
- 17K UNIQUE LABELS
- DISTRIBUTED TRAINING (350 GPUS)

[Mahajan et al. 2018]
Self-supervised Learning (no label)
What if we can get labels for free from unlabelled data and train unsupervised dataset in a supervised manner?
Pretext Tasks

- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the future from the recent past.
- Predict the past from the present.
- Predict the top from the bottom.
- Predict the occluded from the visible.
- Pretend there is a part of the input you don’t know and predict that.
Rotation

- $g(X, y=0)$
  - Rotate 0 degrees

- $g(X, y=1)$
  - Rotate 90 degrees

- $g(X, y=2)$
  - Rotate 180 degrees

- $g(X, y=3)$
  - Rotate 270 degrees

[Gidaris et al. 2018]
Rotation

Image $X$

$g(X, y=0)$
Rotate 0 degrees
Rotated image: $X^0$
ConvNet model $F(\cdot)$

$g(X, y=1)$
Rotate 90 degrees
Rotated image: $X^1$
ConvNet model $F(\cdot)$

$g(X, y=2)$
Rotate 180 degrees
Rotated image: $X^2$
ConvNet model $F(\cdot)$

$g(X, y=3)$
Rotate 270 degrees
Rotated image: $X^3$
ConvNet model $F(\cdot)$

Gidaris et al. 2018
Rotation

Image $X$

- $g(X, y=0)$
  - Rotate 0 degrees
  - Rotated image: $X^0$
  - ConvNet model $F(.)$
  - Maximize prob. $F^0(X^0)$
  - Predict 0 degrees rotation ($y=0$)

- $g(X, y=1)$
  - Rotate 90 degrees
  - Rotated image: $X^1$
  - ConvNet model $F(.)$
  - Maximize prob. $F^1(X^1)$
  - Predict 90 degrees rotation ($y=1$)

- $g(X, y=2)$
  - Rotate 180 degrees
  - Rotated image: $X^2$
  - ConvNet model $F(.)$
  - Maximize prob. $F^2(X^2)$
  - Predict 180 degrees rotation ($y=2$)

- $g(X, y=3)$
  - Rotate 270 degrees
  - Rotated image: $X^3$
  - ConvNet model $F(.)$
  - Maximize prob. $F^3(X^3)$
  - Predict 270 degrees rotation ($y=3$)

Gidaris et al. 2018
Patches

$X = (\text{cat face}, \text{ear})$; $Y = 3$

Example:

Question 1:

Question 2:

[Doersch et al., 2015]
Summary

• Basic steps to build an ML system
• Open-world machine learning
• Industry-scale machine learning
Thank you!