Building Real Applications

With Deep Learning
Plan

What is Deep Learning?

Who is using it?

How do we use it?
Deep Learning is in Everything!

Photos: Suggest Tags

This helps your friends label and share their photos, and makes it easier to find out when photos of you are posted.

Suggest photos of me to friends
When photos look like me, suggest tagging me

This feature uses a comparison of photos you're tagged in to suggest that friends tag you in new photos.

[Image of a smartphone with text: What can I help you with?]

[Image of a Facebook screen showing photos and a suggestion feature]

[Image of a Google self-driving car]
Building Intuition

Given a Problem

How to build a deep learning model to solve the problem?
Problem

What does it mean when someone gives you a problem?
Problem

Data (Features + Label)

<table>
<thead>
<tr>
<th>Comedy?</th>
<th>Length (m)</th>
<th>Female Lead?</th>
<th>Super Hero?</th>
<th>Like?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>90</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>120</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Goal (Given features, predict label)
Deep Learning vs. Other Machine Learning

Deep Learning
Learn a representation for the data

Decision Trees, KNN, etc.
Implement an algorithm, given features
Why Deep Learning

- Powerful
- Feature Learning
- Software availability
- Sounds good
Plan

What is Deep Learning?

Who is using it?

How do we use it?
  Architecture (Layer Types)
  Loss Functions
  Hyper Parameters
  Challenges
  Tricks
Convolutional Neural Networks (CNN/ConvNet)

Architecture
- Number of layers
- Number of neurons in each layer
Layer Types

Convolutional
Fully Connected
Pooling
Dropout
Activation
Layer Types: Convolutional

# Filters
Size Filters
Stride
Padding
In practice, most use Max Pooling.

\[ f(x,y) = \max(x,y) \]

\[ \frac{\delta f}{\delta x} = 1 \ (x \geq y) \]

\[ \frac{\delta f}{\delta y} = 1 \ (y \geq x) \]
Layer Types: Pooling

\[ f(x, y) = \max(x, y) \]

\[ \frac{\delta f}{\delta x} = 1 \quad (x \geq y) \]

\[ \frac{\delta f}{\delta y} = 1 \quad (y \geq x) \]

Intuition: \( x = 4, y = 2 \)

\( f(x, y) = 4 \), and the function is not sensitive to the setting of \( y \).

Increase \( y \) by small amount \( h \), \( f(x, y + h) = 4 \), so there is no effect.
Layer Types: Dropout

Randomly turn off activations in a layer
Layer Types: Activation

- **tanh**:
  \[ \tanh(x) \]

- **ReLU**:
  \[ R(z) = \max(0, z) \]
Vanishing/Exploding Gradient

Activation Function: Large input regions mapped to extremely small range.

Large change in input produces small change in output $\rightarrow$ small gradient
Plan

What is Deep Learning?

Who is using it?

How do we use it?
  Architecture (Layer Types)
  Loss Functions
  Hyper Parameters
  Challenges
  Tricks
Loss Function

Common Choices:

- Softmax
- Sigmoid Cross Entropy
- Euclidean
Plan

What is Deep Learning?

Who is using it?

How do we use it?
  - Architecture (Layer Types)
  - Loss Functions
  - Hyper Parameters
  - Challenges
  - Tricks

Challenges

Tricks
Hyper Parameters

Batch Size
   - Updating network parameters with each batch

Learning Rate

Step Size
   - When to reduce learning rate

Momentum
   - Add fraction of previous weight update to current update

Weight Decay
   - Reduce the value of weights
Plan

What is Deep Learning?

Who is using it?

How do we use it?
- Architecture (Layer Types)
- Loss Functions
- Hyper Parameters

Challenges

Tricks
Overfitting

HUGE Problem

What can we do to avoid it?
- Early Stopping
- Reduce # Parameters
- Dropout
- Data Augmentation
Training Time/Memory

GPUs have limited memory (6GB)

Training can take hours, days, or weeks!

Biggest factors in deciding your architecture, and hyper parameters
Plan

What is Deep Learning?

Who is using it?

How do we use it?
   Architecture (Layer Types)
   Loss Functions
   Hyper Parameters
   Challenges
   Tricks
Common Architectures

Alexnet (8 layers) 60M parameters
VGG-Net (19 layers) 138M parameters
Res-Net (50+ layers) 25M parameters
Fine-Tuning

Typically what is done in practice

Most problems do not have enough data to train these networks from scratch.
Deeper = Better?

Professor Heff Ginton takes a deeper approach to deep learning!
Debugging (What is going wrong?)

Start with a small amount of data and make sure you can overfit it

Looking at activations

Visualizers

- Mouth Open
- High Cheekbones
- 5 o’clock Shadow
- Arched Eyebrows
- Bushy Eyebrows
- Earrings
Activations

Class activation maps of top 5 predictions

Class activation maps for one object class
Deep Learning Fails

Happens A LOT

Training/Validation/Testing Distribution
Automatic Caption Generation

- "man in black shirt is playing guitar."
- "construction worker in orange safety vest is working on road."
- "two young girls are playing with lego toy."
- "girl in pink dress is jumping in air."
- "black and white dog jumps over bar."
- "young girl in pink shirt is swinging on swing."
Deep Learning Fails

A cat is sitting on a toilet in a bathroom.
Deep Learning Fails

a man is riding a skateboard on a ramp
When a user takes a photo, the app should check whether they're in a national park...

Sure, easy GIS lookup. Gimme a few hours.

...and check whether the photo is of a bird.

I'll need a research team and five years.

In CS, it can be hard to explain the difference between the easy and the virtually impossible.
Personal Experience

Getting familiar with software (Caffe, Torch, TensorFlow, MatConvNet)

Most Time:
  Creating/Choosing Loss Function
  Augmenting Data

Less Time:
  Picking Architecture
  Choosing Hyperparameters
Review

Deep Learning is a tool

Be aware of challenges

Combine different ML methods