CS 265
BIG DATA SYSTEMS
NoSQL | Neural Networks | SQL | Graph | Data Science
Talks at the lab this week:

11:30am today, **Yifan Wu from Berkeley**: Real-Time Interactive Data Analytic Interfaces

11:30am tmr, **K. Karanasos from Microsoft**: (big data/ML)

Discussion papers as of Monday

Review submission every class
Market share

Academic interest

Wealth of applications

deep learning
Some input

DATA  DATA  DATA

Some output
Image recognition

Cat

Dog
Table
Cat $\rightarrow$ Γάτα

Machine translation
What happens in Vegas...
How do we do this mapping?

\[ \text{cat} = f(\text{🐱}) \]
How do we do this mapping?

cat = f( )
Multi-layer perceptron

“can represent a wide variety of interesting functions”
Multi-layer perceptron
Deep neural networks

Some weird neural network!

Data to features

Features to labels
Multiple layers
Training phase: pass labeled data until we get to an acceptable accuracy

Inference: new data -> result
How do we train these networks?
How do we train these networks?
How do we train these networks?

Forward pass to compute a prediction

Labeled data
How do we train these networks?

Forward pass to compute a prediction

Labeled data

Fish
How do we train these networks?

Forward pass to compute a prediction

Backward pass to ‘slightly’ nudge the weights

Labeled data

Fish
How do we train these networks?

Forward pass to compute a prediction

Backward pass to ‘slightly’ nudge the weights

repeat until happy/convergence
PERFORMANCE (TRAINING/INFEERENCE)

Labeled data
PERFORMANCE (TRAINING/INFERENCE)

READING/Writing data + COMPUTATION
Labeled data

ACCURACY
Deep Residual Learning for Image Recognition [He et al., 2016]
Cat

DNN

Cat
Single Model

Ensemble Model
Single Model

Ensemble Model
Representationally richer  
Wisdom of the crowd

DNN  
DNN  
DNN
Cat  
Cat  ➔  CAT
Catastrophe
finding the first principles of neural networks
VERY EXPENSIVE TO TRAIN
100 variants of VGG-16 (different structures)

Dataset
- CIFAR-10

Training approaches
- Full-data
- Bagging

How expensive is it?
Training time (hrs.)

Number of neural networks

- Full-data
- Bagging
Time to add one neural network

- Bagging
- Full-Data

Number of neural networks vs. Training time (hrs.)
- Bagging
- Full-Data

27 min.  35 min.

CIFAR-10
Bagging

Full-Data

CIFAR-10

CIFAR-100

CIFAR-10 | ResNet

Complex data

Complex model

27 min. 35 min.

35 min. 67 min.

206 min.

516 min.
ensemble size

- ~10
- ~250

NN ensembles
other models ensembles
complex problems need larger ensembles
MotherNets
it is all about data movement and computation

Rapid Deep Ensemble Learning
MotherNets

DNN architectures
MotherNets

Capture structural similarity
MotherNets

Capture structural similarity
MotherNets

Capture structural similarity
MotherNets

Capture structural similarity
Train once
MotherNets

Same function as MotherNet

Capture structural similarity  Train once  Transfer learned function

i  

ii  

iii  

MotherNets

Capture structural similarity
Train once
Transfer learned function
Train incrementally
MotherNets

- Capture structural similarity
- Train once
- Transfer learned function
- Train incrementally
I. Capture structural similarity

*Find the largest network structure common amongst all networks*
I. Capture structural similarity

*Find the largest network structure common amongst all networks*

Neural Network Specifications

- **NN1**: L1=20, L2=13, L3=5
- **NN2**: L1=15, L2=8, L3=20
- **NN3**: L1=10, L2=18, L3=10, L4=22

→

MotherNet
I. Capture structural similarity

*Find the largest network structure common amongst all networks*

```
L1  L2  L3  L4
NN1 20  13  5
NN2 15  8  20
NN3 10  18 10  22
```

MotherNet
Find the largest network structure common amongst all networks
I. Capture structural similarity

*Find the largest network structure common amongst all networks*

**Neural Network Specifications**

- **NN1**: 
  - L1: 20
  - L2: 13
  - L3: 5

- **NN2**: 
  - L1: 15
  - L2: 8
  - L3: 20

- **NN3**: 
  - L1: 10
  - L2: 18
  - L3: 10
  - L4: 22

- **MotherNet**: 
  - L1: 10
  - L2: 8
  - L3: 5
I. Capture structural similarity

*Find the largest network structure common amongst all networks*

<table>
<thead>
<tr>
<th>NN1</th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>13</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NN2</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>8</td>
<td>20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NN3</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>18</td>
<td>10</td>
<td>22</td>
<td></td>
</tr>
</tbody>
</table>

MotherNet

Neural Network Specifications
I. Capture structural similarity

*Find the largest network structure common amongst all networks*

Neural Network Specifications

NN1

<table>
<thead>
<tr>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>13</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

NN2

<table>
<thead>
<tr>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>8</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

NN3

<table>
<thead>
<tr>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>18</td>
<td>10</td>
<td>22</td>
</tr>
</tbody>
</table>

MotherNet

Smaller than any of the ensemble networks
MotherNets

Capture structural similarity

Train once

Transfer learned function

Train incrementally
MotherNets

Capture structural similarity | Train once | Transfer learned function | Train incrementally
II. Transfer learned function

Function Preserving Transformations

*Increase the capacity (expressivity) of the networks while preserving their function (also accuracy)*
II. Transfer learned function

Function Preserving Transformations

*Increase the capacity (expressivity) of the networks while preserving their function (also accuracy)*

Widen the network

Deepen the network
Function Preserving Transformations

*Increase the capacity (expressivity) of the networks while preserving their function (also accuracy)*

Morph MotherNets to ensemble networks
MotherNets

i. Capture structural similarity
ii. Train once
iii. Transfer learned function
iv. Train incrementally
Capture structural similarity

Train once

Transfer learned function

Train incrementally

full data

bagging
100 variants of VGG-16 (different structures)

CIFAR-10 Dataset

Training approaches:
- Full-data
- Bagging
Training time (hrs.)

Number of neural networks

- Full-data
- Bagging
- MotherNets
Time to add one neural network

- Bagging
- Full-Data
- MotherNets

CIFAR-10
Time to add one neural network

- Bagging
- Full-Data
- MotherNets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Bagging</th>
<th>Full-Data</th>
<th>MotherNets</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>27 min.</td>
<td>35 min.</td>
<td>7 min.</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>35 min.</td>
<td>67 min.</td>
<td>14 min.</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>ResNet</td>
<td>206 min.</td>
<td>155 min.</td>
</tr>
</tbody>
</table>

Complex data

Complex model
Accuracy vs Performance?
Accuracy vs Performance?

![Graph showing accuracy vs performance](image)

- **Error rate (%)**
- **Training time (min)**
- **Test error rate (%)**
- **Number of clusters**

Legend:
- Bagging
- Full-data
- KD
- MotherNets
- Vote
- EA
- SL
- O

**Graph Details**
- The graph compares different ensemble methods including Bagging, Full-data, KD, and MotherNets.
- It illustrates the trade-off between accuracy and training time for various ensemble configurations.
- The x-axis represents different classifiers or methods, and the y-axis represents error rate or training time.

**Experiments**

- **Step 1: Training the MotherNet.**
- **Step 2: Hatching Ensemble Networks.**
- **Step 3: Training Ensemble Networks.**

**Training Setup**

- While our focus is on reducing training costs, the MotherNets can be used to accelerate the training of diverse ensembles of neural networks.

**Optimizing for Inference**

- **2.5. Optimizing for Inference**
- We explain how MotherNets can be used to accelerate the training process and how these networks can be deployed compactly.
- This approach optimizes the training process, reducing both error rate and training time.

**Figure 3**

- MotherNets achieve comparable individual and ensemble accuracy to the full-data approach but in a fraction of the training time – striking a better time-accuracy tradeoff.
Accelerating inference in MotherNets

↑Inference cost
Accelerating inference in MotherNets

↑ Inference cost

IV — Incremental Training

Training

Maintain common MotherNets parameters

Inference cost

Training
Accelerating inference in MotherNets

Inference cost

IV — Incremental Training

Training

Inference

Maintain common MotherNets parameters

Full-pass

↑Inference cost

Maintain common MotherNets parameters
Accelerating inference in MotherNets

Inference cost

IV — Incremental Training

Maintain common MotherNets parameters
Accelerating inference in MotherNets

Initial Results

Ensemble of 5 VGGNets
Layers between 13 and 34

CIFAR-10
Accelerating inference in MotherNets

Initial results

- **Standard**
- **Share-MN**

Inference time:
- Standard: 3.71 min.
- Share-MN: 1.95 min.

Test error rate:
- Standard: 7.5%
- Share-MN: 7.9%
finding the first principles of neural networks