MATLAB의 새로운 딥러닝 기술: 객체 인식부터 GAN까지

송완빈, MathWorks
Artificial Intelligence is Transforming Engineering

- Robotics & Autonomous
- Industrial Automation
- Predictive Maintenance
- Patient Monitoring
- Electricity Use Forecasting
- Automated Driving
Artificial Intelligence

- In computer science, artificial intelligence (AI), sometimes called *machine intelligence*, is *intelligence demonstrated by machines*.
Integrating AI is a priority for companies today

Average number of AI projects expected

10x increase in AI projects in three years!


n = 57 to 63
Gartner Research Circle members with AI/ML projects deployed/in use today, excluding “unsure”
Source: Gartner AI and ML Development Strategies Survey
Q. How many projects are deployed/in use today? How many projects do you estimate in zero to 12 months, 12 to 24 months, and 24 to 36 months?
ID: 390794
AI skills and data quality are major concerns

**Top Three Challenges to AI and ML Adoption**

<table>
<thead>
<tr>
<th>Enterprise Maturity</th>
<th>Skills of Staff</th>
<th>Data Scope or Quality</th>
<th>Governance Issues or Concerns</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 106 * Gartner Research Circle members, excluding “unsure”</td>
<td>56%</td>
<td>34%</td>
<td>13%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fear of Unknown</th>
<th>Understanding AI Benefits and Uses</th>
<th>Security or Privacy Concerns</th>
<th>Measuring the Value</th>
<th>Risk or Liabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 106 * Gartner Research Circle members, excluding “unsure”</td>
<td>42%</td>
<td>20%</td>
<td>17%</td>
<td>6%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Finding a Starting Point</th>
<th>Finding Use Cases</th>
<th>Defining the Strategy</th>
<th>Finding Funding</th>
<th>Integration Complexity</th>
<th>Confusion Over Vendor Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 106 * Gartner Research Circle members, excluding “unsure”</td>
<td>26%</td>
<td>25%</td>
<td>12%</td>
<td>26%</td>
<td>10%</td>
</tr>
</tbody>
</table>

**Top barriers to successful adoption of AI**

1. Skills of your team
2. Data quality

But, Deep Learning with **MATLAB** is Growing Rapidly

**Shell:**
Machinery Identification

**Genentech:**
Pathology Analysis

**Ritsumeikan University:**
Reduce Exposure in CT Imaging

**Airbus:**
Aircraft inspection

**Musashi Seimitsu Industry Co.:**
Detect Abnormalities in Auto Parts

**Veoneer:**
Lidar Object Detection
Why MATLAB & MathWorks for Deep Learning?

- **Domain-specialized workflows** for engineering and science
- **Multi-platform deployment** of full applications and systems
- **Platform productivity**
- **Interoperability** with TensorFlow and PyTorch
AI-driven system design workflow
You Should Consider the Entire Process for AI Project
What is AI Modeling?

- Designing the Neural Network topology
- Training and validating the model with dataset
- Experimenting with and tuning different parameters

**Neural Network = Model**

- Input Layer
- Hidden Layers (n)
- Output Layer
Apps for AI Modeling

Deep Network Designer

• Choose from a comprehensive library of pre-trained models
• Easily design, analyze, and train networks graphically
• Monitor training with plots of accuracy, loss, and validation metrics.
• Generate equivalent MATLAB code to recreate design
Apps for AI Modeling

Experiment Manager

- Saves time during trial-and-error model selection
- Sweep over hyperparameter combinations
- Sort, filter, monitor training plot, confusion matrix
- Allows you to replicate research and track results
Get benefits from Apps

- Designing the Neural Network topology
- Training and validating the model with dataset
- Experimenting with and tuning different parameters
How about advanced deep learning model training?

- Generate an image similar to real images
- Image to Image Translation Using GAN
Answer is “You can now train advanced models with MATLAB”

- Variational Autoencoder (VAE)
- One shot learning Using Siamese Networks
- Neural Style Transfer
- Image Captioning using Attention
You now have 2 options to train Deep Learning model

• For a Simple Deep Learning model
  • Use Apps or High-Level API

```
net = trainNetwork();
```

• For a Advanced Deep Learning model
  • Use Low-Level API

```
for i = 1:epoch
    [loss, Grad] = dlfeval(@itloss, ...
      images, labels, net);
    [net.Learnables, ~] = adamupdate(...
      net.Learnables, grads);
end
```
You now have 2 options to train Deep Learning model

• For a Simple Deep Learning model
  • Use Apps or High-Level API

```
net = trainNetwork()
```

• When to Use?
  • Relatively Simple Deep Learning model
    • Object Recognition / Detection
    • Semantic Segmentation
    • Sequence Classification
    • Time Series Forecasting
  
  • Single Command to train Network
  
  • Leverage Apps for training, validating and tuning parameters
You now have 2 options to train Deep Learning model

• When to Use?
  • Advanced Deep Learning model training
    • Generative Models
    • Networks needs custom loss function, custom training rules
    • Multiple Network training
  • Low-level coding required for network training
  • Automatic differentiation for compute gradients

• For a Advanced Deep Learning model
  • Use Low-Level API
    ```matlab
    for i = 1:epoch
        [loss, Grad] = dlfeval(@iLoss, ...
                        images, labels, net);
        [net.Learnables, ~] = adamupdate(...
                        net.Learnables, grads);
    end
    ```
    ![Diagram of Generative Model](image)
    ![Diagram of Discriminative Model](image)
Structure of a Low-Level API - Custom training loop

Convert network and data

\[
\text{net} = \text{dlnetwork}(	ext{lgraph})
\]
\[
\text{data} = \text{dlarray}(	ext{data})
\]

Manual Training loop

Calculate forward processing, loss and gradient

\[
\text{function} \quad [\text{loss}, \text{Gradient}] = \text{myfunction}(\text{data}, \text{net}, \ldots)
\]
\[
\text{loss} = \ldots
\]
\[
\text{Gradient} = \text{dlgradient} (\text{loss}, \text{net.Learnables})
\]

\[
\text{for} \quad i = 1 : \text{epoch}
\]
\[
[\text{loss}, \text{Gradient}] = \text{dlfeval} (@\text{myfunction}, \text{data}, \text{net}, \ldots)
\]
\[
[\text{net.Learnables}, \sim] = \text{adamupdate} (\text{net.learnables}, \text{Gradient}, \ldots)
\]
\[
\text{end}
\]

- Custom training loop for the network training
- Can define custom loss function for gradient calculation
- Compute gradients using Automatic Differentiation
Let’s briefly work through with GAN!
Generative Adversarial Network

**Train to trick the Discriminator**

**Train to judge real / fake correctly**

Generate an image similar to real images
Generative Adversarial Network in Action – Networks

✓ Generate an image from random numbers using transposed Convolution.
✓ Common Network for classification
Generative Adversarial Network in Action – Loss Function

Generator loss function = 

To generate data that the discriminator classifies as "real"

Discriminator loss function = 

To judge the Real image as Real + To judge the Fake image as Fake

Follow the structure, then you can get GAN model!

Full code available
Extensions of GANs

- cGAN
  - A *conditional* generative adversarial network is a type of GAN that also takes advantage of labels during the training process.

- Visual representation of cGAN:
  - Generator: Takes a Sunflower image as input and outputs a Fake Images (Daisy flower).
  - Discriminator: Takes both Real Images (Sunflower) and Fake Images (Daisy) as input and predicts whether the images are Real or Fake.

- Generate Daisy flower!

- Full code available
Extensions of GANs

- **AnoGAN**
  - Anomaly detection using GAN, Unsupervised Learning
  - Train GAN with only Normal data (No Abnormal data needed)
  - Adjust latent vector $z$ that can generates similar images with real data

Images are from *Concrete Crack Images for Classification*
Extensions of GANs

- **Pix2Pix**
  - Image to Image Translation using GAN (Conditional GAN Model)
  - Using *pairs* of images of "before" and "after", generate "after" using "before"  
    
    ![Pix2Pix HD Example](image)

Full code available

Real Image A

Generated image B'

Real Image A

Generator

Discriminator

Label Image

Generated Image

Load Gen. Network

Load Sample Label

Paint Label Image

Generate Image
Extensions of GANs

- **CycleGAN**
  - Unpaired Image to Image Translation using GAN

**Domain transfer**

- Real Image in *domain A*
- *Generator A2B*
- Fake Image in *domain B*
- Discriminator for *domain B*
- Predicted Labels: Real or Fake
- *Generator B2A*
- Reconstructed Image

Full code available

---

MATLAB EXPO

MathWorks
Siamese Network for One-shot Learning

- Siamese Network
  - Neural networks containing two or more identical subnetwork components with shared weights

  $$\|F_1 - F_2\|^2$$ is small.

  $$\|F_1 - F_2\|^2$$ is large.

We'd Like to track this vehicle

And Many More!
Getting started with rich examples

- Train a Siamese network to compare images.
- Train a network to generate handwritten digits.
- Train a network to predict handwritten digit labels and angles or rotations of handwritten digits.
- Train a generative adversarial network (GAN) for sound synthesis.

Documentation

MathWorks®
MATLAB makes it easy to learn and automate workflow steps

**Access Data**
- MURING/LABELING
- FUSION
- DENOISING

**Preprocess**
- MURING/LABELING
- FUSION
- DENOISING

**Access Models**
- BUILD
- BORROW

**Train**
- FROM SCRATCH
- TRANSFER

**Deploy**

---

**Data Preparation**
- Datastores
  -预处理数据
  -数据存储
  -Image, video, audio, and other datatypes

**Preparing Data**
- Finding value
  -Apps for label, images, video, and ground truth data
  -State-of-the-art signal and image processing functions

**Pretrained Models**
- Frameworks
  -VGG-19
  -ResNet-50
  -ResNet-101
  -ONNX

**Interactive Apps**
- Simulation and Test
  -Advanced Deep Learning Model

**Deployment**
- Training Hardware
  -Speed and Scale
  -CUDA
  -TensorRT
  -ONNX

---

**MathWorks**
Thank You