COSC 522 – Machine Learning

Lecture 14 – From Machine Learning to Deep Learning

Hairong Qi, Gonzalez Family Professor
Electrical Engineering and Computer Science
University of Tennessee, Knoxville
http://www.eecs.utk.edu/faculty/qi
Email: hqi@utk.edu

Course Website: http://web.eecs.utk.edu/~hqi/cosc522/
# COSC 522 - Machine Learning (Fall 2020) Syllabus

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Date</th>
<th>Content</th>
<th>Tests</th>
<th>Assignment</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8/20</td>
<td>Introduction</td>
<td></td>
<td>HW0</td>
<td>8/25</td>
</tr>
<tr>
<td>2</td>
<td>8/25</td>
<td>Bayesian Decision Theory - MPP</td>
<td>HW1</td>
<td></td>
<td>9/1</td>
</tr>
<tr>
<td>3</td>
<td>8/27</td>
<td>Discriminant Function - MD</td>
<td>Proj1 - Supervised Learning</td>
<td>9/10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9/1</td>
<td>Parametric Learning - MLE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9/3</td>
<td>Non-parametric Learning - kNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9/8</td>
<td>Unsupervised Learning - kmeans</td>
<td>HW2</td>
<td></td>
<td>9/15</td>
</tr>
<tr>
<td>7</td>
<td>9/10</td>
<td>Dimensionality Reduction - FLD</td>
<td>Proj2 - Unsupervised Learning and DR</td>
<td>9/24</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9/15</td>
<td>Dimensionality Reduction - PCA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9/17</td>
<td>Linear Regression</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>9/22</td>
<td>Performance Evaluation</td>
<td>HW3</td>
<td></td>
<td>9/29</td>
</tr>
<tr>
<td>11</td>
<td>9/24</td>
<td>Fusion</td>
<td>Proj3 - Regression</td>
<td>10/8</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>9/29</td>
<td>Midterm Exam</td>
<td>Final Project - Milestone 1: Forming Team</td>
<td>9/29</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>10/1</td>
<td>Gradient Descent</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>10/6</td>
<td>Neural Network - Perceptron</td>
<td>HW4</td>
<td></td>
<td>10/13</td>
</tr>
<tr>
<td>15</td>
<td>10/8</td>
<td>Neural Network - BPNN</td>
<td>Proj4 - BPNN</td>
<td>10/22</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>10/13</td>
<td>Neural Network - Practices</td>
<td>Final Project - Milestone 2: Choosing Topic</td>
<td>10/13</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>10/15</td>
<td>Kernel Methods - SVM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>10/20</td>
<td>Kernel Methods - SVM</td>
<td>HW5</td>
<td></td>
<td>10/27</td>
</tr>
<tr>
<td>19</td>
<td>10/22</td>
<td>Kernel Methods - SVM</td>
<td>Proj5 - SVM &amp; DT</td>
<td>11/5</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>10/27</td>
<td>Decision Tree</td>
<td>Final Project - Milestone 3: Literature Survey</td>
<td>10/27</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>10/29</td>
<td>Random Forest</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>11/3</td>
<td></td>
<td>HW6</td>
<td></td>
<td>11/10</td>
</tr>
<tr>
<td>23</td>
<td>11/5</td>
<td>From PCA to t-SNE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>11/10</td>
<td>From Gaussian to Mixture and EM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>11/12</td>
<td>From Supervised/Unsupervised to RL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>11/17</td>
<td>From Classification/Regression to Generation</td>
<td>Final Project - Milestone 4: Prototype</td>
<td>11/17</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>11/19</td>
<td>From NN to CNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>11/24</td>
<td>Final Exam</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/3</td>
<td>8:00-10:15</td>
<td>Final Presentation</td>
<td>Final Project - Report</td>
<td>12/4</td>
<td></td>
</tr>
</tbody>
</table>

**Final Exam**

8:00-10:15
A list of misconceptions

• Is deep learning merely deeper?
  – The two unique features of convolutional neural network (CNN)

• Is deep learning a classifier?
  – Engineered features vs. automatic features

• Supervised vs. Unsupervised

• Model-based approach vs. Data-driven approach
  – the two extremes?

• The world beyond CNN
  – GAN, AE, RNN, RL

• Implementation
Core idea 1: Receptive field (RF) and shared weight

Fully-connected

Feature maps
Core idea 2: Hierarchical vision - Max pooling

hidden neurons (output from feature map)

max-pooling units

28 x 28 input neurons

3 x 24 x 24 neurons

3 x 12 x 12 neurons
A simple CNN framework

Derivation of Backpropagation in Convolutional Neural Network (CNN)

Zhifei Zhang
University of Tennessee, Knoxville, TN
October 18, 2016

Abstract—Derivation of backpropagation in convolutional neural network (CNN) is conducted based on an example with two convolutional layers. The step-by-step derivation is helpful for beginners. First, the feedforward procedure is claimed, and then the backpropagation is derived based on the example.

1 Feedforward

The structure of CNN example that will be discussed in this paper is exactly the same to the structure used in the demo of Matlab DeepLearnToolbox [1]. All later derivation will use the same notations in this figure.

1.1 Initialization of Parameters

The parameters are:

- C1 layer, \( k_{1,1} \) (size 5x5) and \( b_1 \) (size 1x1), \( p = 1, 2, \ldots, 6 \)
- C2 layer, \( k_{2,p,q} \) (size 5x5) and \( b_2 \) (size 1x1), \( q = 1, 2, \ldots, 12 \)
- FC layer, \( W \) (size 10x192) and \( b \) (size 10x1)

In \( k_{p,q} \) and \( b_q \), \( l \) indicates the layer, \( p \) and \( q \) denote the map indices of current and next layers, respectively.
A list of misconceptions

• Is deep learning merely deeper?
  – The two unique features of convolutional neural network (CNN)

• Is deep learning a classifier?
  – Engineered features vs. automatic features

• Supervised vs. Unsupervised

• Model-based approach vs. Data-driven approach
  – the two extremes?

• The world beyond CNN
  – GAN, AE, RNN, RL

• Implementation
Engineered features vs. automatic features

Need domain knowledge

Input media → Feature extraction → Feature vector → Pattern classification → Recognition result

1 Feedforward

28x28 → 24x24 → ... → 𝒃₁

Kernel: 𝒌₁,₁ = 5x5

Average pooling: 8x8 → 4x4

Vectorization: 16x12 = 192

10x1

Fully connection layer: FC

Output layer

1.1 Initialization of Parameters

• C₁ layer, 𝒌₁,₁ (size 5◊5) and 𝒃₁,₁ (size 1◊1), 𝑝 = 1, 2, ..., 6

• C₂ layer, 𝒌₂,𝑝,𝑞 (size 5◊5) and 𝒃₂,𝑞 (size 1◊1), 𝑞 = 1, 2, ..., 12

• FC layer, 𝑷 (size 10◊192) and 𝒃 (size 10◊1)

Figure 1: The structure of CNN example that will be discussed in this paper. It is exactly the same to the structure used in the demo of Matlab DeepLearnToolbox [1]. All later derivation will use the same notations in this figure.

In 𝒌ₚ,ₜ and 𝒃ₜ, 𝑩 indicates the layer, 𝑝 and 𝑞 denote the map indices of current and next layers, respectively.

In 𝒌ₚ,ₜ and 𝒃ₜ, 𝑩 indicates the layer, 𝑝 and 𝑞 denote the map indices of current and next layers, respectively.
The flowchart comparison

- Raw image
  - Low-level IP
    - Enhanced image
      - Segmentation
        - Objects & regions
          - Feature Extraction
  - End-to-End
    - Deep Learning
      - Recognition
        - Understanding, Decision, Knowledge
      - Features
        - Engineered vs. Automatic
A list of misconceptions

• Is deep learning merely deeper?
  – The two unique features of convolutional neural network (CNN)

• Is deep learning a classifier?
  – Engineered features vs. automatic features

• Supervised vs. Unsupervised

• Model-based approach vs. Data-driven approach
  – the two extremes?

• The world beyond CNN
  – GAN, AE, RNN, RL

• Implementation
Revisit: A bit of history

• 1956-1976
  – 1956, The Dartmouth Summer Research Project on Artificial Intelligence, organized by John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon

    We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College ... The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

  – The rise of symbolic methods, systems focused on limited domains, deductive vs. inductive systems
  – 1976, the AI Winter

• 1976-2006
  – 1986, BP algorithm

• 2006-???
  – 2006, Hinton (U. of Toronto), Bingio (U. of Montreal), LeCun (NYU)
  – 2012, ImageNet by Fei-Fei Li (2010-2017) and AlexNet
Unsupervised learning – Autoencoder (AE)

\[
\theta_1^*, \theta_2^* = \arg \min_{\theta_1, \theta_2} \frac{1}{n} \sum_{i=1}^{n} L(x^{(i)}, z^{(i)}) = \arg \min_{\theta_1, \theta_2} \frac{1}{n} \sum_{i=1}^{n} L(x^{(i)}, g_{\theta_2}(f_{\theta_1}(x^{(i)})))
\]

\[
L(x, z) = \frac{1}{2} \| x - z \|^2_2
\]

\[
y = f_{\theta_1}(W_1x + b_1)
\]

\[
z = g_{\theta_2}(W_2y + b_2)
\]

\[
\theta_1 = \{W_1, b_1\}, \theta_2 = \{W_2, b_2\}
\]
Discrimination vs. Representation of Data

- **Best discriminating the data**
  - Fisher’s linear discriminant (FLD)
  - NN
  - CNN

  \[ J(w) = \frac{||\tilde{m}_1 - \tilde{m}_2||^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{||w^T(m_1 - m_2)||^2}{w^T S_1 w + w^T S_2 w} = \frac{w^T S_B w}{w^T S_W w} \]

- **Best representing the data**
  - Principal component analysis (PCA)

  \[ x = \sum_{i=1}^{m} y_i b_i + \sum_{i=m+1}^{d} y_i b_i \approx \sum_{i=1}^{m} y_i b_i + \sum_{i=m+1}^{d} \alpha_i b_i \]

  Error: \[ \Delta x = \sum_{i=m+1}^{d} (y_i - \alpha_i) b_i \]
Raw data \( (X_{nxd}) \) → covariance matrix \( (\Sigma_X) \) →
eigenvalue decomposition \( (\lambda_{dx1} \text{ and } E_{dxd}) \) →
principal component \( (P_{dxm}) \) →
\[ Y_{nxm} = X_{nxd} \ast P_{dxm} \]
Denoising Autoencoder (DAE)

\[ L_H(x, z) \]

[DAE:2008]
The two papers in 2006

Techniques to avoid overfitting

• Regularization
  – Weight decay or L1/L2 normalization
  – Use dropout
  – Data augmentation

• Use unlabeled data to train a different network and then use the weight to initialize our network
  – Deep belief networks (based on restricted Boltzmann Machine or RBM)
  – Deep autoencoders (based on autoencoder)
AE as pretraining methods

- **Pretraining step**
  - Train a sequence of shallow autoencoders, greedily one layer at a time, using unsupervised data

- **Fine-tuning step 1**
  - Train the last layer using supervised data

- **Fine-tuning step 2**
  - Use backpropagation to fine-tune the entire network using supervised data
A list of misconceptions

• Is deep learning merely deeper?
  – The two unique features of convolutional neural network (CNN)

• Is deep learning a classifier?
  – Engineered features vs. automatic features

• Supervised vs. Unsupervised

• Model-based approach vs. Data-driven approach
  – the two extremes?

• The world beyond CNN
  – GAN, AE, RNN, RL

• Implementation
Case study: Hyperspectral Image (HSI) Super-Resolution (SR)

Hyperspectral images (HSI): Low spatial but high spectral resolution

Multispectral images (MSI): High spatial but low spectral resolution

HSI-SR: High spatial and high spectral resolution
The traditional formulation

\[ Y_h = S_h \Phi_h, \]

\[ X = S_m \Phi_h. \]

\[ Y_m = S_m \Phi_m, \]

The objective function:

\[ P(X|Y_h, Y_m) \]

The constraints on S:
- Sum-to-one and non-negative
- Sparse
The deep-learning approach using unsupervised learning

\[ Y_h = S_h \times \Phi_h \]

Hyperspectral encoder-decoder

LR HSI

\[ Y_h \]

\[ E_h(\theta_{he}) \]

\[ W_{he1} W_{he2} \ldots W_{heq} \]

\[ S_h \]

Reconstruction loss

\[ D_h(\theta_{hd}) \]

\[ W_{d1} W_{d2} \ldots W_{dk} \]

\[ \hat{Y}_h \]

Shared weights

Multispectral encoder-decoder

HR MSI

\[ Y_m \]

\[ E_m(\theta_{me}) \]

\[ W_{me1} W_{me2} \ldots W_{meg} \]

\[ S_m \]

Reconstruction loss

\[ D_m(\theta_{md}) = D_h(\theta_{hd}) \]

\[ X \rightarrow \hat{Y}_m \]

\[ Y_m = S_m \times \Phi_m \]
The deep-learning approach with two physical constraints on S (Sum-to-one and Non-negativity)

Encoder $E_h(\theta_{hc})$

Decoder $D_h(\theta_{hd})$

$$s_j = \begin{cases} 
v_1 & \text{for } j = 1 \\
v_j \prod_{o < j} (1 - v_o) & \text{for } j > 1
\end{cases}$$

Kumaraswamy $v_o \sim \left(1 - \left(1 - u \frac{1}{\beta}\right) \frac{1}{\alpha}\right)$. 

$$v_o \sim \text{Beta}(u, 1, \beta)$$

$$\text{kuma}(u, \alpha, \beta) = \alpha \beta u^{\alpha-1} (1 - u^\alpha)^{\beta-1}$$
From model-based to data-driven

Model-based  Physics-based Learning  Data-driven
A list of misconceptions

• Is deep learning merely deeper?
  – The two unique features of convolutional neural network (CNN)

• Is deep learning a classifier?
  – Engineered features vs. automatic features

• Supervised vs. Unsupervised

• Model-based approach vs. Data-driven approach
  – the two extremes?

• The world beyond CNN
  – GAN, AE, RNN, RL

• Implementation
GAN

- Two neural networks compete against each other
  - A generator network $G$: mimic training samples to fool the discriminator
  - A discriminator network $D$: discriminate training samples and generated samples

\[ \begin{align*}
\text{For } D: & \quad \max_D \mathbb{E}_{x \sim q(x)} [\log(D(x))] + \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))] \\
\text{For } G: & \quad \min_G \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))]
\end{align*} \]
GAN

The objective function of GANs:

\[
\min_G \max_D \mathbb{E}_{x \sim q(x)} \left[ \log(D(x)) \right] + \mathbb{E}_{z \sim p(z)} \left[ \log(1 - D(G(z))) \right]
\]

Adding an AE
Case study: Age progression and regression

<table>
<thead>
<tr>
<th>Input</th>
<th>Others</th>
<th>Ours</th>
<th>Continuously bidirectional aging</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>31~40</td>
<td>31~40</td>
<td>0<del>5 6</del>10 11<del>15 16</del>20 21<del>30 31</del>40 41<del>50 51</del>60 61<del>70 71</del>80</td>
</tr>
<tr>
<td>5</td>
<td>69~80</td>
<td>71~80</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>16~20</td>
<td>16~20</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>60~80</td>
<td>61~70</td>
<td></td>
</tr>
</tbody>
</table>

Project page: [https://zzutk.github.io/Face-Aging-CAAE](https://zzutk.github.io/Face-Aging-CAAE)
Case study: Conditional Adversarial Autoencoder - CAAE

Encoder $E$
- Input face: $128 \times 128 \times 3$
- Conv_1: $64 \times 64 \times 64$
- Conv_2: $32 \times 32 \times 128$
- Conv_3: $16 \times 16 \times 256$
- Conv_4: $8 \times 8 \times 512$
- FC_1: $64x64x10$
- FC_2: $32x32x256$
- FC_3: $16x16x512$
- FC_4: $8x8x1024$
- Reshape: $8x8 \times 1024 \\ 

Discriminator on $z$ -- $D_z$
- z or $p(z)$
- Prior distribution of $z$ (uniform)

Discriminator on face image -- $D_{img}$
- Input/output face: $128 \times 128 \times 3$
- Conv_1: $64 \times 64 \times 128$
- Conv_2: $32 \times 32 \times 256$
- Conv_3: $16 \times 16 \times 512$
- Conv_4: $8 \times 8 \times 128$
- Reshape: $8x8 \times 128$
- FC_1: $64x64x1024$
- Deconv_1: $32x32x256$
- Deconv_2: $16x16x512$
- Deconv_3: $8x8x128$
- Deconv_4: $128 \times 128 \times 3$

Generator $G$
- Output face: $128 \times 128 \times 3$
- Deconv_1: $16x16x256$
- Deconv_2: $32x32x128$
- Deconv_3: $64 \times 64 \times 128$
- Deconv_4: $128 \times 128 \times 3$
- L2 loss
RNN: A friendly introduction to NN

https://www.youtube.com/watch?v=UNmqTiOnRfg&ab_channel=LuisSerrano
A friendly introduction to RNN

\[
\begin{pmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
1 \\
0 \\
0
\end{pmatrix}
= \begin{pmatrix}
0 \\
1 \\
0
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
0 \\
1 \\
0
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
1
\end{pmatrix}
\]
A more complicated case

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix}
= \begin{bmatrix}
1 \\
0 \\
0 \\
1
\end{bmatrix}
\]

\text{Same}

\[
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
0 & 1 \\
0 & 1
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
1 \\
1
\end{bmatrix}
\]

\text{Next day}

\[
\begin{bmatrix}
1 \\
0 \\
0 \\
1
\end{bmatrix}
= \begin{bmatrix}
1 \\
0 \\
1 \\
2
\end{bmatrix}
\]

\text{Next day}
A more complicated case (cont’d)
A more complicated case (cont’d)
Recurrent neural network (RNN)
The long-short term memory (LSTM) module

LSTMs are explicitly designed to avoid the long-term dependency problem.
Case study: The talking face

Goal: Given an arbitrary audio clip and a face image, automatically generate realistic and smooth face video with accurate lip sync.

Application: Face animation, entertainment, video bandwidth reduction, etc.

[Suwajanakorn et al., 2017]
The talking face

The proposed method: conditional video generation

http://web.eecs.utk.edu/~ysong18/projects/talkingface/talkingface.html
A list of misconceptions

- Is deep learning merely deeper?
  - The two unique features of convolutional neural network (CNN)
- Is deep learning a classifier?
  - Engineered features vs. automatic features
- Supervised vs. Unsupervised
- Model-based approach vs. Data-driven approach – the two extremes?
- The world beyond CNN
  - GAN, AE, RNN, RL
- Implementation
  - Matlab
  - TensorFlow
  - PyTorch
  - Keras