Neural Network

Background

Zhifei Zhang and Hairong Qi
Algorithms whose performance improve as they are exposed to more data over time

- Naive Bayes
- kNN
- Decision Tree
- SVM
- K-Means
- Dimensionality Reduction, e.g., FLD, PCA
For more detailed history of deep learning [https://www.import.io/post/history-of-deep-learning/](https://www.import.io/post/history-of-deep-learning/)
Intuition of Neural Network

Perceptron

Perceptron is usually used to classify the data into two parts. Therefore, it is also known as a Linear Binary Classifier.

- Weights shows the strength of the particular node.
- A bias value allows you to shift the activation function to the left or right.
- The activation function map the input between the required values, e.g., [0, 1].
Perceptron

\[
\begin{align*}
\begin{pmatrix}
1 \\
x_1 \\
\vdots \\
x_{n-1} \\
x_n
\end{pmatrix}^T \begin{pmatrix}
w_0 \\
w_1 \\
\vdots \\
w_{n-1} \\
w_n
\end{pmatrix} &= s \\
F(s) &= \hat{y}
\end{align*}
\]

Common activation functions:

- **Sigmoid**
  \[\sigma(x) = \frac{1}{1+e^{-x}}\]

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

- **ReLU**
  \[\text{max}(0, x)\]

- **Leaky ReLU**
  \[\text{max}(0.1x, x)\]

- **ELU**
  \[\begin{cases} 
  x & x \geq 0 \\
  \alpha(e^x - 1) & x < 0
  \end{cases}\]
Perceptron Learning Rule

Assume the input $x = (x_0, \ldots, x_n)$, ground truth $y$, and output of perceptron $\hat{y}$, the weight is updated as follow:

$$
\Delta w_i = w_i - \alpha (\hat{y} - y) x_i
$$

Where $\alpha$ is the learning rate that is a positive value in the range of $(0, 1)$.

Objective:

$$
E = \frac{1}{2} (y - \hat{y})^2
$$

Partial derivative of $E$ w.r.t. $w_i$:

$$
\Delta w_i = \frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial s} \frac{\partial s}{\partial w_i}
$$

$$
\Delta w_i = (\hat{y} - y) \cdot F'(s) \cdot x_i
$$

Assume 1, directly pass the error backward
XOR Problem
Multi-layer Perceptron --- Neural Network

Input layer --- Hidden layer --- Output layer

Neural Network in more practice
Multi-layer Perceptron --- Solve the XOR Problem

Online demo: [https://lecture-demo.ira.uka.de/neural-network-demo/](https://lecture-demo.ira.uka.de/neural-network-demo/)
Backpropagation

How the multi-layer perceptron (neural network) is learned?

Back propagate the error layer-by-layer

Output error:
\[ E = \frac{1}{2} (y - \hat{y})^2 \]
Backpropagation

How those multi-layer perceptron (neural network) is learned?

Back propagate the error layer-by-layer

Output error:

\[ E = \frac{1}{2} (y - \hat{y})^2 \]

\[ \Delta w_i = (\hat{y} - y) \cdot F'(s_h) \cdot h_i \]
Backpropagation

How those multi-layer perceptron (neural network) is learned?

Back propagate the error layer-by-layer, and

\[
\Delta w_j = \varepsilon \cdot F'(s_x) \cdot x_j
\]

Output error:

\[
E = \frac{1}{2} (y - \hat{y})^2
\]

Back propagated error:

\[
\varepsilon = \frac{\partial E}{\partial h_i} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial s_h} \cdot \frac{\partial s_h}{\partial h_i}
\]

\[
\varepsilon = (\hat{y} - y) \cdot F'(s_h) \cdot w_i
\]

Hidden nodes are independent from each other.
Convolutional Neural Network (CNN)

Handle images which need to preserve the special structure
Convolutional Neural Network (CNN)

LeNet-5 by Yann LeCun et al., 1998

A demo code in Matlab: https://github.com/ZZUTK/An-Example-of-CNN-on-MNIST-dataset
Convolutional Neural Network (CNN)

Appearance of CNN is early (back to 1998) but it booms around 2010s, WHY?

- Large dataset (over 10 million)
- Powerful computing unit

From 2009

From 2010
Convolutional Neural Network (CNN)

Appearance of CNN is early (back to 1998) but it booms around 2010s, WHY?

Deep convolutional neural networks with competitive performance to human

AlexNet, 2012
Convolutional Neural Network (CNN)

Since 2010s, the deep convolutional neural networks are mostly referred to as deep learning, and it flourishes in computer vision area until today.
Convolutional Neural Network (CNN)

Convolution and Pooling

Where is the activation function?
Convolutional Neural Network (CNN)

Why deep? What does each layer learn?
Convolutional Neural Network (CNN)

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) results show deep learning surpassing human levels of accuracy.
Convolutional Neural Network (CNN)

How to learn a CNN? Again, **Backpropagation**.

No parameters, do not need update.  
Directly pass the error backward.  
The same as multi-layer perceptron
Convolutional Neural Network (CNN)

How to learn a CNN? Again, **Backpropagation**.

Multi-layer Perceptron

\[ \varepsilon = E \cdot w \]

\[ \Delta w = E \cdot x \]

Convolutional Neural Network

\[ \varepsilon = E \cdot k_{rot180} \]

\[ \Delta k = x_{rot180} \ast E \]

For simplicity, assume \( F'(s) = 1 \) here.
