CS F425: Deep Learning

08 Neural Network

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Feb 07, 2023 ON-CAMPUS Campus @ BITS-Pilani [Jan-May 2023]

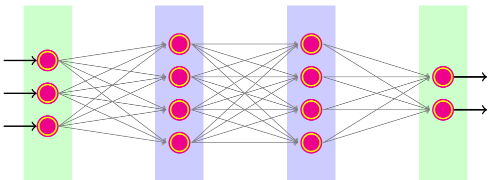
http://ktiwari.in/dl

Brief History

- 1872 Staining/Reticular Theory of Nervous Tissue
- 1943 McCulloch & Pitt (Neuron Model)
- 1947 Donald Hebb (Hebbian Learning)
- 1948 Nerbert (Cybernetics, optimal filter, feedback)
- 1954 Frank Rosenblatt (Perceptron)
- 1959 — MADALINE
- 1962 Hubel & Wiesel (Visual Cortex Model) — Checkers
- 1965 Frank Rosenblatt (MLP: Multi Layer Perceptron)
- 1969 Minsky (Limitations of Perceptron)
- 1986 (Backpropagation)
- 1989 Universal Approximation Theorem
- 1997 LSTM — Deep Blue
- 1998 LeCun (ConvNet for MNIST digit)
- 2006 Unsupervised Pre Training
- 2010 Dahl. (Speech Recognition)
- 2012 AlexNet (ImageNet: Computer Vision) 26→16 … →12% zfNet
- 2013 VGG →7.3
- 2014 Google LeNet →6.7
- 2015 ResNet →3.6 — Waymo, AlphaGo, Yolo
- 2017 DenseNet
- 2018 Transformers, U-Net segmentation — Deepfake, StyleGAN
- 2022 ChatGPT — ChatGPT, AlphaFold

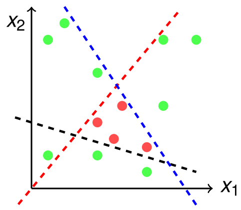
Neural Network

When neurons are interconnected in layers



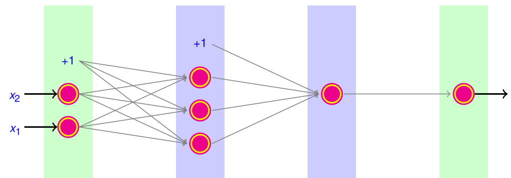
- Number of layers may differ
- Nodes in each intermediate layers may also differ
- Multiple output neurons are used for different class
- **Two levels deep** NN can represent any boolean function

More Example: Design NN for the following data



Whether is it green?

Red-line	Blue-line	Black-line	Color
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



Note: Weights and activation in subsequent layers add power to the model in terms of non linearity.

Neural Network Applications

NN is appropriate for problems with the following characteristics:

- Instances are provided by many attribute-value pairs (more data)
- The target function output may be discrete-valued, real-valued, or a vector of several real or discrete valued attributes
- The training examples may contain errors
- Long training times are acceptable
- Fast evaluation of the target function may be required
- The ability of humans to understand the learned target function is not important

Perceptron Training (delta rule)

When data is not linearly-separable; error fluctuates with *parameter training* updates. It is difficult to decide, when to stop?

- Delta rule converges to a best-fit approximation of the target
- Uses gradient descent
- Consider unthresholded perceptron, $o(\vec{x}) = \vec{w}.\vec{x}$
- Training error is defined as

$$E(\vec{w}) = \frac{1}{2}\sum_{d\in D}(t_d - o_d)^2$$

- Gradient would specify direction of steepest increase $\nabla E(\vec{w}) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots, \frac{\partial E}{\partial w_n}\right]$
- Weights can be learned as $w_i = w_i - \eta\frac{\partial E}{\partial w_i}$
- It can be seen that $\frac{\partial E}{\partial w_i} = \sum_{d\in D}(t_d - o_d)(-x_{id})$

Perceptron Training (delta rule)

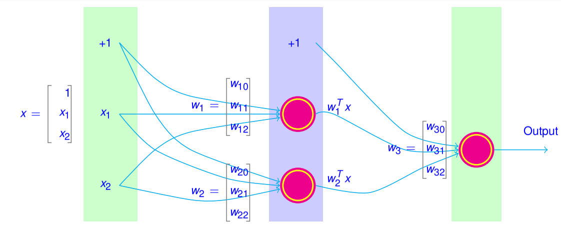
Algorithm 1: Gradient Descent (D, η)

```
Initialize w_i with random weights
repeat
    For each w_i, initialize △w_i = 0
    for each training example d ∈ D do
        Compute output o using model for d whose target is t
        For each w_i, update △w_i = △w_i + η(t−o)x_i
    For each w_i, set w_i = w_i + △w_i
until termination condition is met;
return w
```

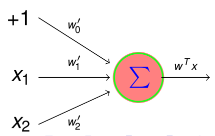
- A date item $d \in D$, is supposed to be multidimensional $d = (x_1, x_2, ..., x_n, t)$
- Algorithm converges toward the minimum error hypothesis.
- Linear programming can also be an approach

Linear Activation is Not Much Interesting

NN with perceptrons have limited capability, even with many layers



$$
\begin{aligned}
Output &= w_{30} \times 1 + w_{31} \times (w_1^T x) + w_{32} \times (w_2^T x) \\
&= w_{30} \times 1 + w_{31} \times [w_{10} \times 1 + w_{11} \times x_1 + w_{12} \times x_2] \\
&\quad + w_{32} \times [w_{20} \times 1 + w_{21} \times x_1 + w_{22} \times x_2] \\
&= (w_{30} + w_{31} w_{10} + w_{32} w_{20}) + (w_{31} w_{11} + w_{32} w_{21}) \times x_1 \\
&\quad + (w_{31} w_{12} + w_{32} w_{22}) \times x_2 \\
&= w'_0 + w'_1 \times x_1 + w'_2 \times x_2
\end{aligned}
$$



Expression of single perceptron

Thank You!

Thank you very much for your attention!