Spline-based Compatible Reduced Basis Methods for Flow Problems

A. Rasheed[1,2] E. Fonn[2] E. H. van Brummelen[3] T. Kvamsdal[4]

[1]adil.rasheed@sintef.no

[2]Mathematics and Cybernetics, SINTEF Digital

[3]Department of Mechanical Engineering, TU/e

[4]Department of Mathematical Sciences, NTNU

Logisitc Regression

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}) - y^{(i)})$$
$$Cost(h_\theta(x) - y) = -log(h_\theta) \text{ if } y = 1$$
$$Cost(h_\theta(x) - y) = -log(1 - h_\theta)$$
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(h_\theta(x^{(i)}))(1 - y^{(i)})log(1 - h_\theta(x^{(i)}))]$$

Decision Tree

$$Entropy = \sum_i -p_i log_2 p_i$$

p_i is the probability of class i

Information Gain $=$ $Entropy_{parent}$-Average $Entropy_{children}$

Batch Gradient Descent vs Stochastic Gradient Descent

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Repeat {

$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

for every $j = 0, ..., n$

}

$$cost(\theta, (x^{(i)}, y^{(i)}) = \frac{1}{2}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m} cost(\theta, (x^{(i)}, y^{(i)})$$

Repeat {

for i =1,....m {

$$\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

}

}

Deep Neural Network

$$
\begin{aligned}
&\mathbf{h}^{(1)} = \sigma(\boldsymbol{\Theta}^{\mathbf{1}}\mathbf{X} + \mathbf{b}^{1})\\
&\mathbf{h}^{(2)} = \sigma(\boldsymbol{\Theta}^{\mathbf{2}}\mathbf{h}^{(1)} + \mathbf{b}^{2})\\
&\vdots\\
&\mathbf{h}^{(n-1)} = \sigma(\boldsymbol{\Theta}^{\mathbf{n-1}}\mathbf{h}^{(n-2)} + \mathbf{b}^{n-1})\\
&\mathbf{H}_{\Theta}(\mathbf{X}) = \sigma(\boldsymbol{\Theta}^{\mathbf{n}}\mathbf{h}^{(n-1)} + \mathbf{b}^{n})
\end{aligned}
$$

Deep Neural Network

$$
\begin{aligned}
&\mathbf{X}(m,q), \mathbf{\Theta^1}(q+1, p^{(2)}) \\
&\mathbf{h}^{(1)}(m, p^{(2)}), \mathbf{\Theta^2}(p^{(2)}+1, p^{(3)}) \\
&\mathbf{h}^{(2)}(m, p^{(3)}), \mathbf{\Theta^3}(p^{(3)}+1, p^{(4)}) \\
&\vdots \\
&\mathbf{h}^{(n-1)}(m, p^n), \mathbf{\Theta^{n-1}}(p^{n-1}+1, p^n), b_{n-1}(m,1)
\end{aligned}
$$

Deep Neural Network

$$\mathbf{X} = [\mathbf{1}\ \mathbf{X}]$$
$$\mathbf{h}^{(1)} = \sigma(\mathbf{X}\boldsymbol{\Theta}^{\mathbf{1}})$$
$$\mathbf{h}^{(1)} = [\mathbf{1}\ \ \mathbf{h}^{(\mathbf{1})}]$$
$$\mathbf{h}^{(2)} = \sigma(\mathbf{h}^{(1)}\boldsymbol{\Theta}^{\mathbf{2}})$$
$$\mathbf{h}^{(2)} = [\mathbf{1}\ \ \mathbf{h}^{(\mathbf{2})}]$$
$$\mathbf{h}^{(3)} = \sigma(\mathbf{h}^{(2)}\boldsymbol{\Theta}^{\mathbf{3}})$$
$$\vdots$$
$$\mathbf{h}^{(n-2)} = \sigma(\mathbf{h}^{(n-3)}\boldsymbol{\Theta}^{\mathbf{n-2}})$$
$$\mathbf{h}^{(n-2)} = [\mathbf{1}\ \ \mathbf{h}^{(\mathbf{n-2})}]$$
$$\mathbf{H}_{\boldsymbol{\Theta}} = \sigma(\mathbf{h}^{(n-2)}\boldsymbol{\Theta}^{\mathbf{n-1}})$$

$p^{(i)}$ is the number of nodes in the i^{th} layer. Thus $p^{(n)}$ is the number of output nodes, n is the number of layers in the network (input+hidden+output) J

Batch Gradient Descent vs Stochastic Gradient Descent

$\mathbf{X} = [\mathbf{1} \ \mathbf{X}]$
$\mathbf{h}^{(1)} = \sigma(\mathbf{X}\boldsymbol{\Theta}^{\mathbf{1}})$
$\mathbf{h}^{(1)} = [\mathbf{1} \ \ \mathbf{h}^{(\mathbf{1})}]$
$\mathbf{h}^{(2)} = \sigma(\mathbf{h}^{(1)}\boldsymbol{\Theta}^{\mathbf{2}})$
$\mathbf{h}^{(2)} = [\mathbf{1} \ \ \mathbf{h}^{(\mathbf{2})}]$
$\mathbf{h}^{(3)} = \sigma(\mathbf{h}^{(2)}\boldsymbol{\Theta}^{\mathbf{3}})$
$\vdots$
$\mathbf{h}^{(n-2)} = \sigma(\mathbf{h}^{(n-3)}\boldsymbol{\Theta}^{\mathbf{n-2}})$
$\mathbf{h}^{(n-2)} = [\mathbf{1} \ \ \mathbf{h}^{(\mathbf{n-2})}]$
$\mathbf{H}_{\boldsymbol{\Theta}} = \sigma(\mathbf{h}^{(n-2)}\boldsymbol{\Theta}^{\mathbf{n-1}})$
$p^{(i)}$ is the number of nodes in the i^{th} layer. Thus $p^{(n)}$ is the number of output nodes, n is the number of layers in the network (input+hidden+output)

Batch Gradient Descent vs Stochastic Gradient Descent

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[y_k^{(i)} log(H_\Theta(x^{(i)})_k)(1-y_k^{(i)}) log(1-H_\Theta(x^{(i)})_k)\right]$$

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[y_k^{(i)} - (H_\Theta(x^{(i)})_k)^2\right]$$

$$J(\Theta) = J(\Theta) + \frac{\lambda}{2m}\sum_{i=1}^{n}\sum_{j=1}^{p^{(i)}-1}\sum_{q=1}^{p^{(i)}}(\Theta_{q,j}^{i})^2$$