15 November 2020

Scalable Machine Learning and Deep Learning - Review Questions 2

By

Akhil Yerrapragada (akhily@kth.se)

Gibson Chikafa (chikafa@kth.se)

Group Name – gibson_akhil

15 November 2020

1. **0.5 point.** Which of the following is/are true about individual tree in Random Forest?
   (a) Individual tree is built on a subset of the features.
   (b) Individual tree is built on all the features.
   (c) Individual tree is built on a subset of instances.
   (d) Individual tree is built on full set of instances.

Ans: a and c

---

2. **0.5 point.** Ensemble model estimators (such as Random Forest) in Spark have a parameter called featureSubsetStrategy. What does it do?

Ans: It is a hyper parameter. It specifies the features used for splitting a tree as a fraction of total number of features.

Example strategies include - auto, all, onethird etc.

---

3. **1 point.** Explain why the entropy becomes zero when all class partitions are pure?

Ans: Entropy means randomness. In Pure Partition all the instances belong to same class, meaning, randomness is zero. Therefore, when all class partitions are pure, the randomness is zero.

Proof:

$$\mathtt{entropy}(\mathtt{D}) = -\sum_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p_i} \log_2(\mathtt{p_i})$$

If m = 1, then for 5 instances belonging to same class

Entropy(D) = –(5/5)log$_2$(5/5)

Entropy(D) = 0

---

4. **1 point.** Explain why the Gini impurity becomes zero when all class partitions are pure?

Ans: If all instances in the partition belong to same class, we say it's a pure partition. Below is the proof depicting why Gini Impurity becomes zero when class partitions are pure.

$$\mathtt{Gini}(\mathtt{D}) = 1 - \sum_{\mathtt{i}=1}^{\mathtt{m}} \mathtt{p_i^2}$$

If m = 1, then for 5 instances belonging to same class

Gini (D) = 1 – (5/5)^2

Gini(D) = 1 – 1 = 0

---

5. **0.5 point.** Assume a feedforward neural network with one hidden layer, in which the output of the hidden units and output units are computed by functions **h** = f(**x**) and **out** = g(**h**), respectively. Show that if we use linear functions in fand g, e.g., **h** = f(**x**) = $\mathbf{w}_1^{\top}\mathbf{x}$ and **out** = g(**h**) = $\mathbf{w}_2^{\top}\mathbf{h}$, then the feedforward network as a whole would remain a linear function of its input.

Ans:

Let us assume that we have one input layer, one hidden layer and one output layer. Which can be assumed as below:

Input layer -> Hidden layer -> Output layer

x -> f(x) -> g(h)

where h = f(x)

Feedforward network on whole as a linear function of input => y^ = g(f(x))

Similarly,

x -> $w_1^T x$ -> $w_2^T h$

where h = $w_1^T x$

Feedforward network on whole as a linear function of input => y^ = $w_2^T(w_1^T x)$

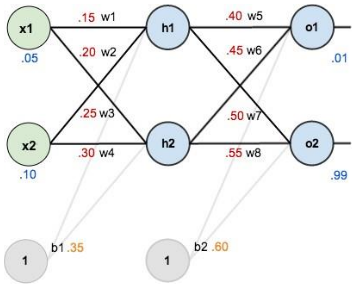
---

6. **0.5 point.** What's the problem of using step function as an activation function in deep feedforward neural networks?

Ans: It is not possible to compute gradient of the step function, meaning, it is non differentiable. Deep forward neural networks using gradient descent require a differentiable function for training. Therefore, it is not suitable to be used in deep forward neural networks.

---

7. **1 point.** Compute the value of w_2 and w_8 after the first iteration of the backpropagation in the following figure. Assume all the neurons use the ReLU activation function and we use squared error function as the cost function. In this figure, red and orange colors indicate the initial values of the weights and biases, while the numbers in blue show the input and true output values.



Ans:

Forward Pass:

net_{h1} = w1x1 +w2x2 +b1 = 0.15 x 0.05 + 0.2 x 0.1 +0.35= 0.3775
net_{h2} = w3x1+ w4x2 + b1= 0.25 x 0.05 + 0.3 x 0.1 +0.35= 0.3925

out_{h1} = max (0, net_{h1}) = 0.3775 [$net_{h1} > 0$]
out_{h2} = max (0, net_{h2}) = 0.3925 [$net_{h2} > 0$]

net_{o1}= w_5out_{h1}+w_6out_{h2} +b2 = 0.4 x 0.3775 + 0.45 x 0.3925 + 0.6 = 0.92762
net_{o2}= w_7out_{h1} + w_8out_{h2} + b2= 0.5 x 0.3775 + 0.55 x 0.3925 +0.6 = 1.00455

out_{o1} = max (0, net_{o1}) = 0.92762
out_{o2} = max (0, net_{o2}) = 1.00455

Error:

E_{o1} = ½ (target$_{o1}$ -output$_{o1}$)^2 = ½ (0.01 - 0.927625)^2 = 0.42101
E_{o2} = ½ (target$_{o2}$ -output$_{o2}$)^2 = ½ (0.99 -1.004625)^2 = 0.00010585

E_{total} = E_{o1} + E_{o2}= 0.4211158

Backward Pass:

To calculate w8:

$\partial E_{total} / \partial w_8 = \partial E_{total} / \partial out_{o2} \times \partial out_{o2} / \partial net_{o2} \times \partial net_{o2} / \partial w_8$

$\partial E_{total} / \partial out_{o2}$ = - (0.99 – 1.00455) = 0.01455 Equation 1

We know that ReLu(x) = max (0, x)

The first derivative of it would be

$$f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

Therefore, $\partial out_{o2} / \partial net_{o2}$ = 1 Equation 2

$\partial net_{o2} / \partial w_8$ = out_{h2} = 0.3925 Equation 3

By multiplying Equation 1,2,3, we get:

$\partial E_{total} / \partial w_8$ = 0.01455 * 1 * 0.3925 = 0.0057108

$w_8^{(next)}$ = w_8 – n * $\partial E_{total} / \partial w_8$ here, we assume learning rate n = 0.5

= 0.55 – 0.5 * 0.00571

= 0.547145

To calculate w2:

$\partial E_{total} / \partial w_2 = \partial E_{total} / \partial out_{h1} \times \partial out_{h1} / \partial net_{h1} \times \partial net_{h1} / \partial w_2$

$\partial E_{total} / \partial out_{h1} = \partial E_{o1} / \partial out_{h1} + \partial E_{o2} / \partial out_{h1}$

$\partial E_{o1} / \partial out_{h1} = \partial E_{o1} / \partial out_{o1} \times \partial out_{o1} / \partial net_{o1} \times \partial net_{o1} / \partial out_{h1}$

$\partial E_{o2} / \partial out_{h1} = \partial E_{o2} / \partial out_{o1} \times \partial out_{o1} / \partial net_{o1} \times \partial net_{o1} / \partial out_{h1}$

$\partial E_{o1} / \partial out_{o1} = -(0.01 – 0.9276) = 0.9176$

$\partial out_{o1} / \partial net_{o1} = 1$

$\partial net_{o1} / \partial out_{h1} = w_5 = 0.40$

$\partial E_{o2} / \partial out_{o1} = -(0.99 – 1.00455) = 0.01455$

$\partial out_{o1} / \partial net_{o1} = 1$

$\partial net_{o1} / \partial out_{h1} = w_5 = 0.40$

$\partial E_{o1} / \partial out_{h1} = 0.9176 * 1 * 0.40 = 0.3670$ Equation 4

$\partial E_{o2} / \partial out_{h1} = 0.01455 * 1 * 0.40 = 0.00582$ Equation 5

By substituting the values of 4 and 5 in the below equation, we get,

$\partial E_{total} / \partial out_{h1} = \partial E_{o1} / \partial out_{h1} + \partial E_{o2} / \partial out_{h1}$

$\partial E_{total} / \partial out_{h1} = 0.3670 + 0.00582 = 0.37282$

$\partial E_{total} / \partial w_2 = \partial E_{total} / \partial out_{h1} \times \partial out_{h1} / \partial net_{h1} \times \partial net_{h1} / \partial w_2$

$\partial E_{total} / \partial w_2 = 0.37282 * 1 * 0.1 = 0.037282$

$w_2^{(next)} = w_2 – n * \partial E_{total} / \partial w_2$ here, we assume learning rate n = 0.5

$= 0.2 – 0.5 * 0.03728$

$= 0.2 – 0.01864$

$= 0.18136$