CS7643: Deep Learning Problem Set 1

Student: Xiaohu Zhang

September 2, 2024

1 Proofs

(1)Gradient of softmax
Find the gradient of softmax with respect to the logits:

$$s_i = \frac{e^{Z_i}}{\sum_k e^{Z_k}} \tag{1}$$

We need to find:

$$\frac{\partial s_i}{\partial Z_j} = \frac{\partial \left(\frac{e^{Z_i}}{\sum_k e^{Z_k}} \right)}{\partial Z_j} \tag{2}$$

Let's break it down into two cases: when $i = j$ and when $i \neq j$.
For $i \neq j$: Consider the following example:

$$\frac{\partial}{\partial Z_3} \left(\frac{e^{Z_1}}{e^{Z_2} + e^{Z_3}} \right) = \frac{0 - e^{Z_1} \cdot e^{Z_3}}{\left(e^{Z_2} + e^{Z_3}\right)^2} \tag{3}$$

The general case for $i \neq j$ becomes:

$$\begin{aligned} \frac{\partial s_i}{\partial Z_j} &= \frac{\partial}{\partial Z_j} \left(\frac{e^{Z_i}}{\sum_k e^{Z_k}} \right) \\ &= \frac{0 \cdot \sum_k e^{Z_k} - e^{Z_i} \cdot e^{Z_j}}{\left(\sum_k e^{Z_k}\right)^2} \\ &= -s_i \cdot s_j \end{aligned}$$

For $i = j$:

$$\begin{aligned} \frac{\partial s_i}{\partial Z_i} &= \frac{\partial}{\partial Z_i} \left(\frac{e^{Z_i}}{\sum_k e^{Z_k}} \right) \\ &= \frac{e^{Z_i} \cdot \sum_k e^{Z_k} - e^{Z_i} \cdot e^{Z_i}}{\left(\sum_k e^{Z_k}\right)^2} \\ &= s_i \left(1 - s_i\right) \end{aligned}$$

Using the indicator function:

$$\mathbb{1}_{i,j} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial S_i}{\partial Z_j} = S_i(\mathbb{1}_{i,j} - S_j)$$

and in matrix form:

$$\frac{\partial S}{\partial Z} = S \cdot (I - S^T)$$

where I is the identity matrix.

(2)AND OR Operation

Find W_{AND} and b_{AND} satisfy the conditions,assume if b_{AND} is -1.1 and $W_{AND} = [1, 1]$

$$
\begin{aligned}
f(x_1) &= 0 + 0 - 1.1 < 0, 0 \\
f(x_2) &= 0 + 1 - 1.1 < 0, 0 \\
f(x_3) &= 0 + 0 - 1.1 < 0, 0 \\
f(x_4) &= 1 + 1 - 1.1 \geq 0, 1
\end{aligned}
$$

Find W_{OR} and b_{OR} satisfy the conditions, $W_{OR} = [1, 1]$ and $b_{OR} = 1$

$$
\begin{aligned}
f(x_1) &= 0 + 0 - 1 < 0, 0 \\
f(x_2) &= 0 + 1 - 1.1 = 0, 1 \\
f(x_3) &= 1 + 0 - 1 = 0, 1 \\
f(x_4) &= 1 + 1 - 1 \geq 0, 1
\end{aligned}
$$

(3) XOR Operation

If I plot the X_1 and X_2 as well as the XOR function in Figure 1, with X_1 being the class 0 and X_2 being the class 1, it would be impossible to draw a straight line to separate the two classes.

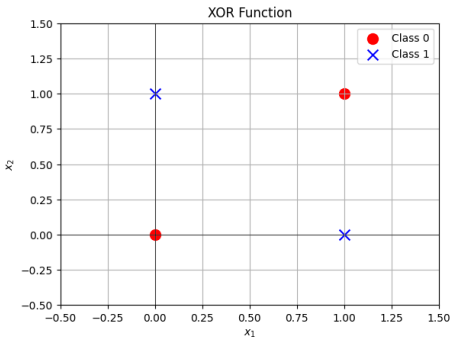

Figure 1: XOR Function with linear splitting

Paper Reading and Review

Key contribution

Paper: 'Weight Agnostic Neural Networks' by Adam Gaier and David Ha at Google Brain

1. What is the main contribution of this paper? In other words, briefly summarize its key insights. What are some strengths and weaknesses of this paper?

Key Contribution

This paper fundamentally changed our understanding of neural networks. We used to believe that neural networks must tune their weights through the gradient descent algorithm. However, the paper proposes an approach that eliminates the need for weight learning by assigning shareable equal weights to each layer of the neural network and searching for optimal neural architectures. This method achieves significant performance on certain supervised learning tasks and generates high accuracy on the MNIST dataset. The method begins with shared weights across all layers and connections in the network. It then searches for the optimal network based on its complexity and performance.

Strengths

This approach is significantly different from Neural Architecture Search (NAS), which considers weight training as a key step. The Weight Agnostic Neural Network (WANN) approach minimizes the impact of weights and focuses on discovering the optimal network architecture. By not relying on weights, WANN avoids the drawbacks of traditional networks that require gradient descent methods, such as issues like gradient explosion or vanishing. The reduction in training requirements can further be applied to cases like few-shot learning.

Weaknesses

Based on the paper's performance, the weights still played a significant role in the study. The ensemble weights, as well as the trained and tuned weights, performed much better compared to the random weights. The weights will still play a role if we want to achieve more accurate results. Additionally, the author acknowledged that this method cannot outperform the well-known and standard CNN approach. Furthermore, the study in the paper included classification tasks but mainly focused on games with a rich amount of data available for training. It remains unclear whether the network can also work on more complex cases.

Personal Takeaway

2. What is your personal takeaway from this paper?
My first impression of this paper's approach to searching for the optimal network structure is that it is very similar to searching for a policy in reinforcement learning.
Additionally, the weight settings remind me of the Kaiming initialization algorithm used by PyTorch, as proposed in He (2015). In Kaiming's paper, he proposes a method for assigning initial weights to ensure gradient stability. This approach contrasts with the method in this paper, which uses randomly generated weights to make the neural network less dependent on the weights.

Last but not least, the same author, Kaiming He, published another paper in He 2019. arguing that a randomly generated network can also achieve significant accuracy on the ImageNet benchmark, which also seems to emphasize that a randomly generated network architecture could work, as opposed to the complex network search methods performed in the WANN paper. My key takeaway is that the weights and network architecture are both very important, and the results may highly depends on the tasks applied for the algorithm.

Specific Question 1

3. The traditional view of optimization in deep learning (and often in general) is that we are searching the space of weights to find the best ones. In other words, learning is a search problem. How would you view the above paper from the perspective of search?

In my view, the WANN paper shift the focus of searching from searching weights to searching the optimal architecture.This way reframing the learning problem as well as the traditional gradient descent approach as the topology search in order to identify the ideal network structure, similar to the policy in the reinforcement learning space. The optimization algorithm is also different as the loss function is no longer the target, performance and complexity is the key considerations in such searching process.

Specific Question 2

4. One of the key aspects of deep learning is that given a parameterized function, we can find weights to represent any function if it has sufficient depth and complexity. What does this paper say about the representational power of architectures given a fixed method for deter- mining weights? Does the method for determining the weights matter? Do you think these two have equal representational power? Why or why not?

Fixed method to determine weights:

The paper did some study using a range of weights or a single weights, random weights etc to search for the network architecture. It is found that the fixed value of the weight various significantly when the fixed value had marginal changes, which will failed the task.

Methods of determine the weights:

The method really matters, as mentioned in the paper, random weighs, ensemble weights, training and tuned weights will have significantly different accuracy with 10 percent difference.

Representational Power:

The network architecture and the weights setting are not equal. This paper demonstrated that even with a random weights, the accuracy can still remained around 80 percent, which indicates that the optimal network structure will be more impactful consider it is taking account the trade off between complexity and performance. The paper also indicates that the architecture will have more impact when the network is implemented for Specific purpose.