

進捗報告

1 今週やったこと

- NAS の実装

2 NAS

DARTS 系の NAS の実験をする。ネットワークにはまずエッジとセルが必要となる。コード 1, 2 に Pytorch で実装したクラスを示した。逆伝播ができることも確認した。

3 今後の予定

実装したクラスでネットワークを構築して実験する。対象とする問題は Cifar-10 を考えている。

4 ソースコード

動作することを第一に実装した。operators で演算子の候補を受け取る。セルは 3 つのノードで 1 入力 1 出力と暫定的に定めた。

Listing 1: Edge

```

1 class Edge(nn.Module):
2     def __init__(self, operators):
3         super(Edge, self).__init__()
4         self.operators = operators
5
6         rand = torch.randn(len(operators),
7                               requires_grad=True)
8         self.theta = rand / torch.sum(rand)
9
10    def forward(self, input: Tensor) ->
11        Tensor:
12        output = torch.zeros(input.shape,
13                               requires_grad=True)
14        for (theta_i, operator) in zip(self.
15            theta, self.operators):
16            if operator == None:
17                continue
18            output = output + theta_i *
19                operator(input)
20
21
```

```

16     return output

```

Listing 2: Cell

```

1 class Cell(nn.Module):
2     def __init__(self, operators):
3         super(Cell, self).__init__()
4
5         self.node_num = 3
6
7         self.ref = [(0, 1), (0, 2), (1, 2)]
8         self.edges = [Edge(operators) for _
9             in self.ref]
10
11    def forward(self, input) -> Tensor:
12        nodes = [torch.zeros(*list(input.
13            shape), requires_grad=True) for _
14            in range(self.node_num)]
15        nodes[0] = input
16
17        for idx, (inref, outref) in enumerate
18            (self.ref):
19            nodes[outref] = nodes[outref] +
20                self.edges[idx](nodes[inref])
21
22    return nodes[-1]

```