Discovering Neural Wirings
Mitchell Wortsman1,2, Ali Farhadi1,2,3, Mohammad Rastegari1,3
1 PRIOR @ Allen Institute for AI 2 University of Washington 3XNOR.AI

**MOTIVATION**
- Traditionally, the connectivity patterns of neural networks are manually defined or largely constrained [even with methods of Neural Architecture Search (NAS)] [1].
- We allow for a much larger space of possible networks by relaxing the typical notion of layers and enabling channels to form connections independent of each other.
- The wiring of our network is not fixed during training – as we learn the network parameters we also learn the connectivity.

**CONTRIBUTIONS**
- We present an algorithm, Discovering Neural Wirings (DNW), to efficiently learn the connectivity of a neural network.
- By learning the connectivity of MobileNetV1 (x0.25) [2] we boost the accuracy on ImageNet by 10%.
- We demonstrate that DNW can also be used to effectively train sparse neural networks in a single training run: We only ever train with 10% of weights (in the forward pass) and only lose 2.5% accuracy on ImageNet compared to our dense baseline.

**KEY TAKEAWAYS**
- It is possible to realize the benefits of overparameterization during training, even when the resulting model is sparse.
- As NAS becomes more fine grained, finding a good architecture is akin to finding a sparse subnetwork of the complete graph.

**Algorithm**
- Intuition
  - If the gradient is pushing \( u \) in a direction which aligns with \( Z_u \), then we strengthen the magnitude of the weight \( w_{u,v} \).
  - If this alignment happens consistently then \( |w_{u,v}| \) will be eventually be strong enough to enter the edge set \( E \).
  - If \( (u,v) \) enters \( E \), another edge will be removed. We show that when this swapping does occur, it is beneficial (i.e. it decreases the loss on the mini-batch under both conditions).
- If the edge is already in the graph, the update rule is no different than standard SGD.

**Results**

<table>
<thead>
<tr>
<th>Model</th>
<th>ImageNet Top-1 Test Accuracy [%]</th>
<th>Model Accuracy [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobileNetV1-DNW</td>
<td>75.00</td>
<td>93.70</td>
</tr>
<tr>
<td>MobileNetV1-Random Graph</td>
<td>72.93</td>
<td>93.70</td>
</tr>
<tr>
<td>MobileNetV1-DNW-Small</td>
<td>65.74</td>
<td>93.70</td>
</tr>
<tr>
<td>ShuffleNetV1</td>
<td>80.93</td>
<td>93.70</td>
</tr>
<tr>
<td>Xception</td>
<td>79.23</td>
<td>93.70</td>
</tr>
</tbody>
</table>

**Sparsely Connected Networks**

We apply our algorithm to the task of training a sparse ResNet152. The sparsity is maintained throughout training, as motivated by Sparse Networks From Scratch [4].

---