Solving hybrid machine learning tasks by traversingweight space geodesics

NeurIPS 21-Response

We would like to thank the reviewers for their detailed feedback. We are glad that each of the reviewers had positive comments about the submission.

The reviewers were also interested in understanding the connection between the algorithms we proposed in the paper and to how it could lead to geodesic paths.

Our algorithm finds geodesic paths (approximately)

In the earlier section we've shown that the problem of finding paths in the weight-space that minimize the total distance traversed in the output space is equivalent to minimizing the energy functional ($S(\gamma)$) resulting in a geodesic path between the two networks of interest (a trained network and another network that satisfies a secondary objective).
There are many computational strategies to solve for the geodesic path[1, 2]. Some of the predominant ones are: (i) Iterative midpoint method, (ii) Gradient descent, (iii) Gridding the space followed by Djikstra to find the shortest path, (iv) Shooting method[3] and (v) Iterative path straightening [3].

Strategies numbered (i, ii, v) require a seed path to calculate the geodesic, and in most cases generating such a seed path is challenging. The default seed path is usually the linear path connecting the two end points on the manifold which may not converge to a suitable geodesic. Although Strategy (iii) doesn't require a seed path, it requires gridding of the space which is feasible for R^3 but not for very high dimensional spaces (which are of interest to neural networks). Strategy (iv) adopts a unique shooting approach which works by shooting a discrete geodesic path γ_0 with finite number of steps starting from the start-point in the manifold along a random direction v_0, and iteratively updates the shooting direction v in order to converge to the target path.

We evaluated the geodesic between two trained networks using the iterative midpoint method ([1]) by choosing the linear path connecting the two networks as the seed path.

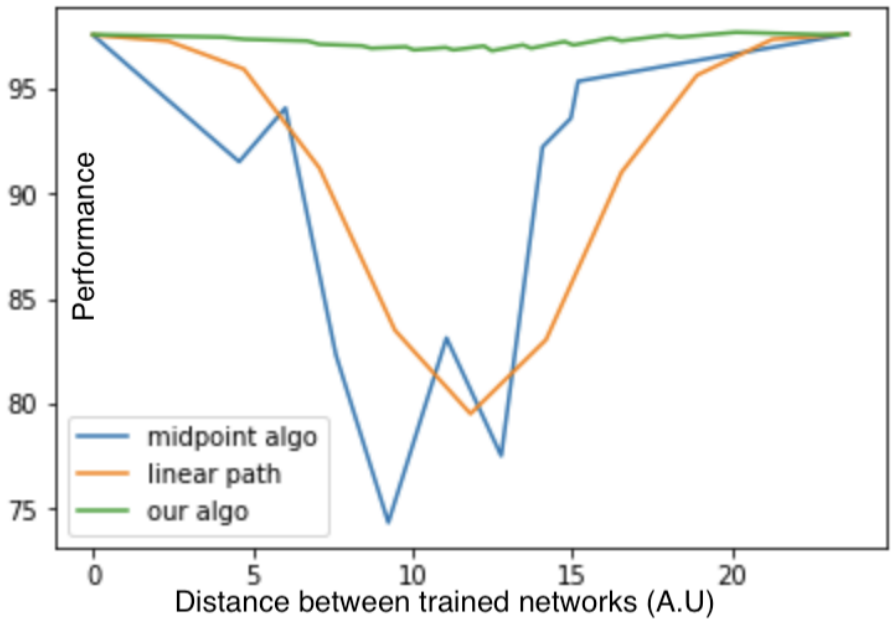

Figure 1: Iterative midpoint algorithm

We notice that the midpoint method is unable to explore the space when given a linear path as its seed (Figure-1) while the path generated by our algorithm (in green) is able to efficiently explore the space, find a low energy path between the two networks, resulting in a high-performance path.

We also applied strategy (v) to construct a geodesic between the two trained networks. The straightening method works by initializing γ_0 as an arbitrary path connecting the two trained networks, and iteratively straightening it using a gradient descent approach to minimize the energy functional ($S(\gamma)$) of the path.

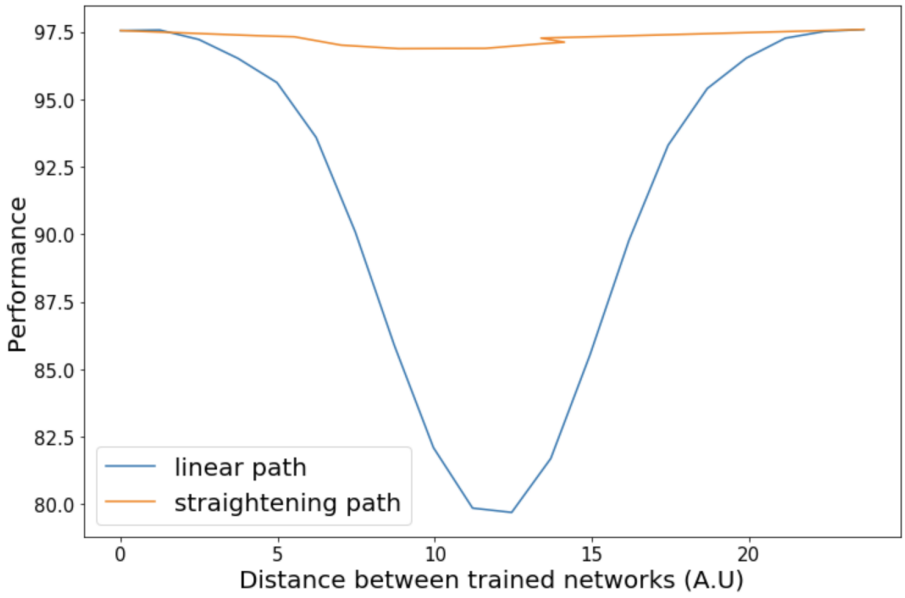

Figure 2: Straightening Path algorithm

Figure-2 clearly shows that the straightening path strategy that iteratively minimizes the energy functional is able to explore the space efficiently and converge to a geodesic path, which is also a high performance path.

We also compare the path obtained by our algorithm with that obtained by iteratively 'straightening' an arbitrary path by minimizing the energy functional:

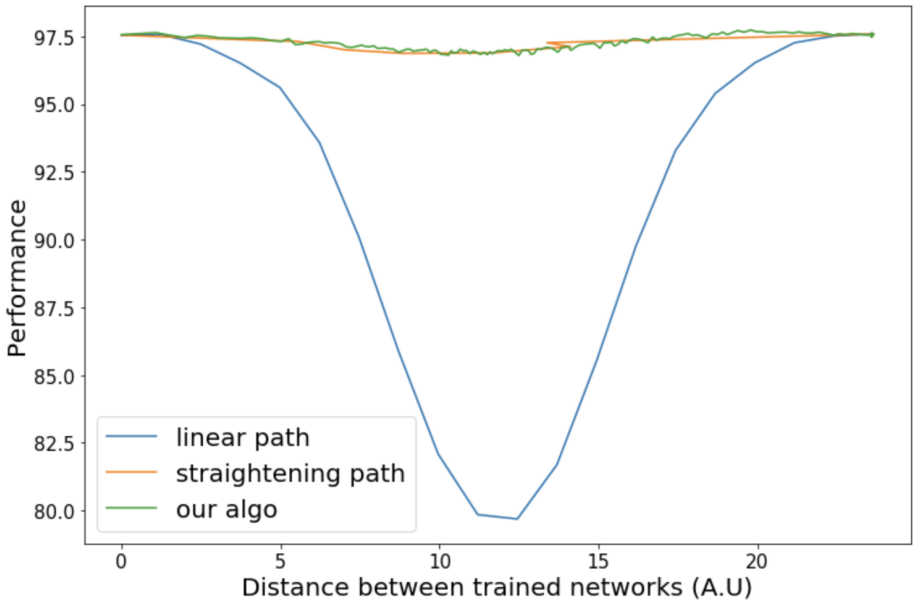

Figure 3: Comparing Linear, path through minimizing energy functional and our algorithm

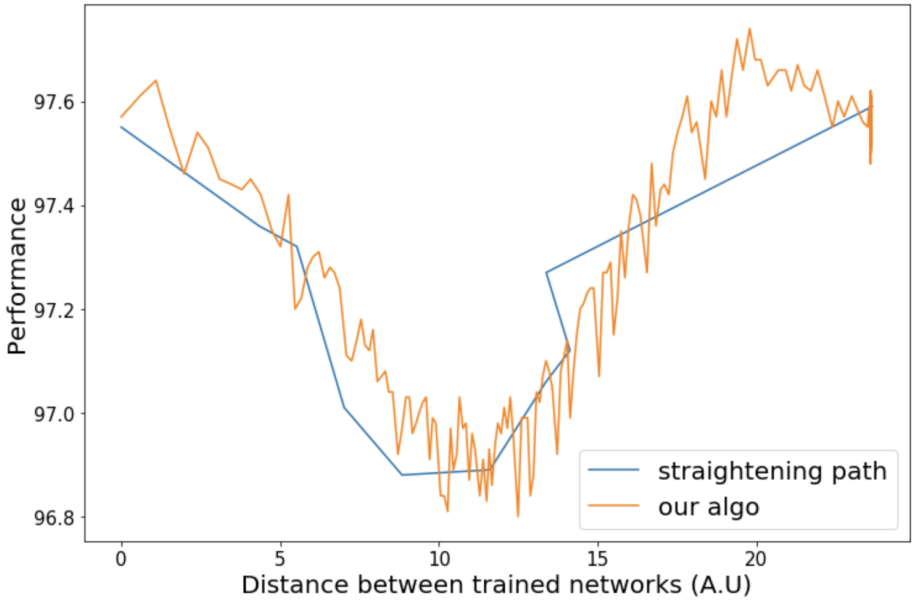

Figure 4: Comparing Path through minimizing energy functional and our algorithm

Figure-3,4 shows that our algorithm finds a path similar to that obtained by conventional approaches (such as straightening an arbitrary path between two networks).

In the earlier section, we concluded that minimizing the energy functional (S(γ)) is equivalent to minimizing the total distance traversed in the output space when the networks weights are being perturbed along the path (γ).

Our algorithm was designed to find a path that fulfills two objectives: (i) minimize the energy functional in order to minimize the total distance traversed by the network in the output space and (ii) discover novel networks that satisfy certain secondary properties (eg weight sparsity or alleviating catastrophic forgetting).

For applications that require us to discover novel networks on the manifold, we have no access to the terminal network, therefore current strategies that either use an arbitrary seed path between two points on the manifold or those that grid the entire space wouldn't be sufficient.

Recall equation-4 in the paper, the energy functional S(γ) is:

$$S(\gamma) = \int_0^1 \frac{d\gamma(t)}{dt}^T \mathbf{g}_{\gamma(t)}(\mathbf{x}) \frac{d\gamma(t)}{dt} \; dt \tag{1}$$

$$= \int_{\mathbf{w_t}}^{\mathbf{w_a}} d_O^2(\mathbf{du}, \mathbf{g_w}(\mathbf{x})) \tag{2}$$

We can rewrite this integral as a Riemannian sum by dividing the domain of the path ($\gamma : [0,1] \in W$ - $[0,1]$ into K subintervals
$(t_0 t_1)$, (t_1, t_2), ..., (t_{K-1}, t_K)
wherein $0 = t_0 <$, $t_1 <$, ..., $< t_K$=1.

The energy of the entire path in terms of the discrete energies of the subintervals is:

$$S(\gamma) = \lim_{\Delta t_k \to 0} \sum_{k=0}^{K} S(\gamma(t_k)) \Delta t_k \tag{3}$$

wherein energy of each subinterval is:

$$S(\gamma(t_k)) = d_O^2(\mathbf{du}, \mathbf{g}_{\gamma(t_k)}) \tag{4}$$

$$= \mathbf{du}^T \mathbf{g}_{\gamma(t_k)} \mathbf{du} \tag{5}$$

As our goal was to find a path from a trained network ($\mathbf{w_t}$) to a hyperplane that satisfy certain secondary objectives (like sparsity), we designed a hybrid approach that incorporates both the objectives:

- Minimizing the energy for every small step taken (min: $\mathbf{du}^T g_w \mathbf{du}$), while
- Maximizing the motion towards the network of interest in the euclidean weights space. (max: $\mathbf{du}^T(\mathbf{w_a} - \mathbf{w})$)

The linear path connecting the two networks on the manifold provides the euclidean geodesic (min-energy) path: $\gamma(t) = (1-t)\mathbf{w} + t\mathbf{w_a}$ between $\mathbf{w}$ and $\mathbf{w_a}$. The path has zero acceleration $\frac{d^2\gamma}{dt^2} = 0$. Therefore the tangent vector to this path is $\frac{d\gamma}{dt} = \mathbf{w_a} - \mathbf{w}$ at all points $\mathbf{w}$.

Therefore there is a constant explore-exploit tussle between moving towards the target network while minimizing the energy for every step taken.

Therefore **our algorithm** is composed of the two terms described above:

$$\theta(\mathbf{w})^T < \mathbf{g_w}(\mathbf{x}) > \theta(\mathbf{w}) - \beta\ \theta(\mathbf{w})^T(\mathbf{w_a} - \mathbf{w})$$

The first term minimizes the kinetic energy and maximizing the second term finds vectors that point along the euclidean geodesic.

β is a parameter that can be tuned to weigh the two. In figure-5, we demonstrate that by tuning β, we can generate multiple paths to go between the two trained networks (trained on MNIST). Please note that the linear path between the two networks is shown in figure-3 in blue, while our paths obtained after tuning β can still find paths of high performance.

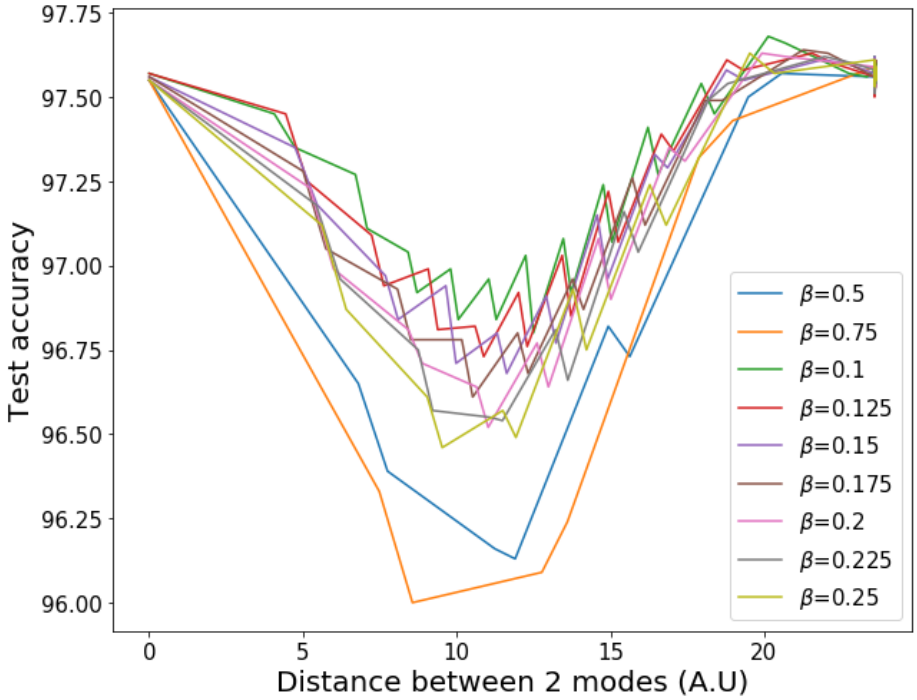

Figure 5: Tuning β

The paths obtained from our algorithm are very similar to that obtained by traditional methods (like path straightening by minimizing the energy functional as shown in fig-4) also have a close to zero acceleration typical of geodesic paths (figure-6).

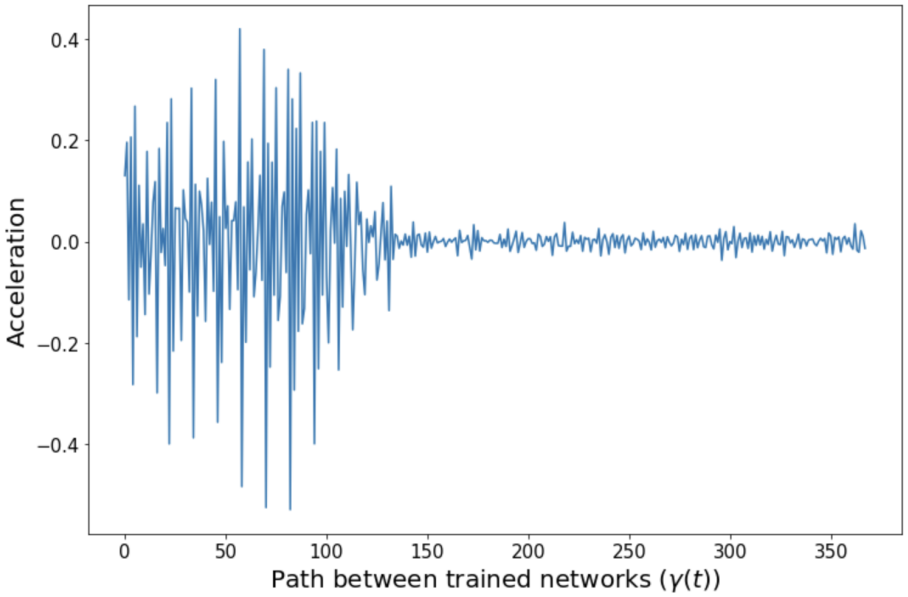

Figure 6: Acceleration of path from our algorithm

References

[1] Anand Deopurkar, Katherine Redfield, and Jongmin Baek. Finding geodesics on surfaces, 2007.

[2] Keenan Crane, Marco Livesu, Enrico Puppo, and Yipeng Qin. A survey of algorithms for geodesic paths and distances. *arXiv preprint arXiv:2007.10430*, 2020.

[3] Qian Xie, Sebastian Kurtek, Huiling Le, and Anuj Srivastava. Parallel transport of deformations in shape space of elastic surfaces. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 865–872, 2013.