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.

A number of reviewers were interested to understand in greater mathematical detail (i) How the Riemannian structure relates to the performance of the network and (ii) How 'geodesic paths' under our definition are 'high performance' paths in weight space. Here we provide mathematical clarification on the relationship between geodesic paths and functional performance.

Neural network representation: We represent a feed-forward neural network (NN) as a smooth, C^∞ function $f(\mathbf{x};\mathbf{w})$, that maps an input vector, $\mathbf{x} \in R^{\mathrm{k}}$, to an output vector, $f(\mathbf{x};\mathbf{w}) = \mathbf{y} \in R^{\mathrm{m}}$. The function, $f(\mathbf{x};\mathbf{w})$, is parameterized by a vector of weights, $\mathbf{w} \in R^{\mathrm{n}}$, that are typically set in training to solve a specific task. We refer to $W = R^n$ as the *weight space* (W) of the network, and we refer to $\mathcal{Y} = R^m$ as the *output space*. Therefore, the networks are represented in two spaces simultaneously: (1) the weights space (W) which is independent of the input (task) and (2) the output space ($\mathcal{Y}$) which is input (task) dependent.

In the specific numerical experiments within the paper (MNIST, CIFAR-10, Fashion-MNIST), we apply networks for classification tasks, and, in these cases, the number of elements in the output vector ($\mathbf{y}$) is equal to the number of classes in the dataset. In these examples, the *output space* is the softmaxed output of the feedforward neural network. However, our framework *is not* restricted to networks used for classification tasks or for supervised learning.

Objective: Broadly, our goal is to find a path in the weights space ($\gamma : [0,1] \to W$) from a trained network ($\mathbf{w_t}$) to a second network that encodes a secondary objective (like sparsity or performance on a second task) ($\mathbf{w_a}$) that *minimizes* the total movement of the output vector of the network as the networks' weights are being perturbed along the path (γ).

Construction of Metric Tensor: To formalize the notion of output change along a path in weight space, we construct a metric tensor to evaluate how infinitesimal perturbation in the weights space (W) impacts movement in the

output space ($\mathcal{Y}$), effectively measuring the similarity of the output of the network over all inputs fed to the network.

To construct the metric, we fix the input ($\mathbf{x}$) to the network and ask how the output changes on the outputs space ($\mathcal{Y}$) as the weights are infinitesimally changed from $\mathbf{w_t} \in W$ by $\mathbf{du}$.

$$f(\mathbf{x}, \mathbf{w_t} + \mathbf{du}) \approx f(\mathbf{x}, \mathbf{w_t}) + \mathbf{J_{w_t}}\ \mathbf{du}, \tag{1}$$

where $\mathbf{J_{w_t}}$ is the Jacobian of $f(\mathbf{x}, \mathbf{w_t})$ for a fixed $\mathbf{x}$, $J_{i,j} = \frac{\partial f_i}{\partial w^j}$, evaluated at $\mathbf{w_t}$.

In this work, we evaluate the distance moved in output space ($d_O(\mathbf{du}, \mathbf{g_{w_t}})$) by calculating the euclidean distance between the output vectors corresponding to $\mathbf{w_t}$ and $\mathbf{w_t} + \mathbf{du}$ for a single input ($\mathbf{x}$) fed to the network. (We note that the framework is not restricted to Euclidean distance and metrics can be constructed for general distance measures on the output space, but we focus on the Euclidean case for clarity).

$$d_O(\mathbf{du}, \mathbf{g_{w_t}}(\mathbf{x})) = \sqrt{|f(\mathbf{x}, \mathbf{w_t} + \mathbf{du}) - f(\mathbf{x}, \mathbf{w_t})|^2} \tag{2}$$

$$= \sqrt{\mathbf{du}^T\ (\mathbf{J_{w_t}}(\mathbf{x})^T\ \mathbf{J_{w_t}}(\mathbf{x}))\ \mathbf{du}} \tag{3}$$

$$= \sqrt{\mathbf{du}^T\ \mathbf{g_{w_t}}(\mathbf{x})\ \mathbf{du}} \tag{4}$$

$$d_O(\mathbf{du}, \mathbf{g_w}(\mathbf{x})) = \sqrt{\mathbf{du}^T \mathbf{g_w}(\mathbf{x}) \mathbf{du}} \tag{5}$$

where $\mathbf{g_{w_t}}(\mathbf{x}) = \mathbf{J_{w_t}}(\mathbf{x})^T\ \mathbf{J_{w_t}}(\mathbf{x})$ is the metric tensor evaluated at the point $\mathbf{w_t} \in W$ for a single datapoint ($\mathbf{x}$) and $d_O(\mathbf{du}, \mathbf{g_w}(\mathbf{x}))$ is the distance moved in output space when the weights are perturbed by $\mathbf{du}$ at $\mathbf{w} \in W$ and the metric tensor ($\mathbf{g_w}(\mathbf{x})$) is evaluated for a single datapoint $\mathbf{x}$.

The key issue of how to incorporate multiple training datapoints in a common space is achieved by evaluating the root mean square (RMS) distance over the individual distances moved in output space for single datapoints ($\mathbf{x_i}$) fed to the network. That is:

$$d_O(\mathbf{du}, \mathbf{g_{w_t}}()) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_O^2(\mathbf{w_t} + \mathbf{du}, \mathbf{w_t}, \mathbf{x_i})} \tag{6}$$

$$= \sqrt{\frac{1}{N}\sum_{i=1}^{N} \mathbf{du^T} \mathbf{g_{w_t}}(\mathbf{x_i}) \mathbf{du}} \tag{7}$$

$$= \sqrt{\mathbf{du^T} < \mathbf{g_{w_t}}(\mathbf{x}) > \mathbf{du}} \tag{8}$$

$$\tag{9}$$

where, $< \mathbf{g_{w_t}}(\mathbf{x}) > = \frac{1}{N}\sum_{i=1}^{N} \mathbf{g_{w_t}}(\mathbf{x_i})$, ie the metric at every point in W is the average of metrics derived from individual data points. The metric tensor is an $n \times n$ symmetric matrix that defines an inner product and local distance metric, $\langle \mathbf{du}, \mathbf{du} \rangle_{\mathbf{w}} = \sqrt{\mathbf{du^T} < \mathbf{g_w}(\mathbf{x}) > \mathbf{du}}$, on the tangent space of the manifold, $T_w(W)$ at each $\mathbf{w} \in W$.

Thus, in this section, we have constructed a metric tensor by computing a euclidean distance between output vectors corresponding to a single input (ie $d_O(\mathbf{du}, \mathbf{g_w}(\mathbf{x})) = \sqrt{\mathbf{du}^T \mathbf{g_w}(\mathbf{x}) \mathbf{du}}$), while the distance between output vectors corresponding to all inputs fed to the network is the quadratic mean (RMS) distance across individual datapoints.

Note: The same result can be derived by constructing a geodesic within an energy (vs length framework). In the energy framework (which is a standard approach in the mathematical literature that avoids the complexities of the square root), $S(\gamma) = \int_0^1 \frac{d\gamma(t)}{dt}^T \mathbf{g}_{\gamma(t)}(\mathbf{x}) \frac{d\gamma(t)}{dt}\ dt$ where we construct a metric tensor that uses the square of the euclidean distance between output vectors corresponding to a single input (ie $d_O(\mathbf{du}, \mathbf{g_w}(\mathbf{x})) = \mathbf{du^T} \mathbf{g_w}(\mathbf{x}) \mathbf{du}$), while the distance between output vectors corresponding to all inputs fed to the network is the mean of the distance across individual datapoints.

Connection between Network output and Network performance on tasks Our formulation focuses on finding paths in weights space where change in network output ($f(\mathbf{x}; \mathbf{w})$) is minimized for changes in weight. For specific tasks, we evaluate the performance of the network using an explicit loss function. In our work we typically apply the cross entropy loss ($\mathbf{Z}$), which is a function of the output of the network, $f(\mathbf{x}; \mathbf{w})$ and a true label ($t_{\mathbf{x}}$) $\{1, 2, ..., m\}$ of the input fed to the neural network.

$$(\mathbf{y_s}(\mathbf{x_i}))_j = \frac{\exp(f(\mathbf{x_i};\mathbf{w})_j)}{\sum_{j=1}^{m}\exp(f(\mathbf{x_i};\mathbf{w})_j)} \quad \text{Softmax} \tag{10}$$

$$\mathbf{Z}(\mathbf{w}) = -\sum_{i=1}^{N}\log(\mathbf{y_s}(\mathbf{x_i}))_{\mathbf{t_{x_i}}} \quad \text{Cross entropy loss} \tag{11}$$

Constructing paths in weights space

Our objective is to find a path in the weights space (W) between a trained network ($\mathbf{w_t}$) and a second network that encodes a secondary objective (like sparsity) ($\mathbf{w_a}$) such that the total distance moved in the outputs space is minimized.

Since our path begins from a trained network ($\mathbf{w_t}$) that produces an output vector that evaluates to low loss value, we want to find a path of weight changes (beginning from $\mathbf{w_t}$ and terminating at $\mathbf{w_a}$) such that the integrated impact on the movement in the outputs space ($\mathcal{Y}$) is minimized.

Mathematically, we want to find a curve $\mathcal{C} \in W$, with start and end points $\mathbf{w_t}$ and $\mathbf{w_a}$ respectively, such that the integrated distance moved in the outputs space is minimized.

$$L(\mathcal{C}) = \int_{\mathcal{C}} d_O(\mathbf{du}, < \mathbf{g_w}(\mathbf{x}) >) \tag{12}$$

$$= \int_{\mathcal{C}} \sqrt{\mathbf{du^T} < \mathbf{g_w}(\mathbf{x}) > \mathbf{du}} \tag{13}$$

On parameterizing the curve traversed ($\mathcal{C} \in W$) by $\gamma : [0,1] \to \mathcal{C} \in W$, wherein $\gamma(0) = \mathbf{w_t}$, $\gamma(1) = \mathbf{w_a}$, the differential element along the path ($\mathbf{du}$) can be rewritten as:

$$\mathbf{du} = \frac{d\gamma}{dt}dt, \quad \mathbf{w} = \gamma(t)$$

The total length of the path parameterized by $\gamma(t)$ is:

$$L(\gamma) = \int_0^1 \sqrt{(\frac{d\gamma}{dt}(dt))^T < \mathbf{g}_{\gamma(t)}(\mathbf{x}) > (\frac{d\gamma}{dt}(dt))} \tag{14}$$

$$L(\gamma) = \int_0^1 \sqrt{\frac{d\gamma(t)}{dt}^T < \mathbf{g}_{\gamma(t)}(\mathbf{x}) > \frac{d\gamma(t)}{dt}}\ dt, \tag{15}$$

Let Ω be the set of all piecewise differentiable curves from $\mathbf{w_t}$ to $\mathbf{w_a}$ in the weights space (W), we want to find γ^* such that:

$$L(\gamma^*) = \min_\gamma L(\gamma) \quad \forall \gamma \in \Omega \tag{16}$$

Therefore, by minimizing L(γ), we are ensuring that the neural networks output is similar (in an L2 sense as defined in eq-2) to that of the trained network as we traverse the path(γ) from $\mathbf{w_t}$ to $\mathbf{w_a}$. We empirically observe that when the output vectors of the networks along the path are similar, the loss function of networks along the path remains similar to that of the (starting) trained network, resulting in a series of high performing networks.

Equivalent to geodesics: We also note that minimizing L(γ) is equivalent to determining the geodesic path in the pseudo-Riemannian manifold (W,$\mathbf{g_w}$) between $\gamma(0) = \mathbf{w_t}$, $\gamma(1) = \mathbf{w_a} \in W$ with the metric computing distance moved in the outputs space.

A minor annoyance with the total path-length formulation L(γ) is that it is invariant under reparameterization of γ, and so it does not admit unique solutions. Instead of length, we consider the energy functional S(γ) [Equation-4 in the paper].

$$S(\gamma) = \int_0^1 \frac{d\gamma(t)}{dt}^T \mathbf{g}_{\gamma(t)}(\mathbf{x}) \frac{d\gamma(t)}{dt}\, dt, \tag{17}$$

By the Cauchy-schwarz inequality, we know that for each smooth curve γ,

$$L(\gamma)^2 = \left(\int_0^1 \sqrt{\frac{d\gamma(t)}{dt}^T < \mathbf{g}_{\gamma(t)}(\mathbf{x}) > \frac{d\gamma(t)}{dt}}\, dt \right)^2 \tag{18}$$

$$L(\gamma)^2 \leq \left(\int_0^1 1^2 dt \right) \left(\int_0^1 \frac{d\gamma(t)}{dt}^T \mathbf{g}_{\gamma(t)}() \frac{d\gamma(t)}{dt}\, dt \right) \tag{19}$$

$$L(\gamma)^2 \leq 2S(\gamma) \tag{20}$$

with equality ($\mathrm{L}(\gamma)^2 = 2\mathrm{S}(\gamma)$) if and only if $|\frac{d\gamma(t)}{dt}^T < \mathbf{g}_{\gamma(t)}(\mathbf{x}) > \frac{d\gamma(t)}{dt}|$ is a constant. Since any curve can be reparameterized to have constant $|\frac{d\gamma(t)}{dt}^T < \mathbf{g}_{\gamma(t)}(\mathbf{x}) > \frac{d\gamma(t)}{dt}|$, minimizing L($\gamma$) is equivalent to minimizing S(γ).

Conclusion:

$$\min_\gamma(S(\gamma)) = \min_\gamma(L(\gamma)) \tag{21}$$

$$= \min_\gamma \left(\int_0^1 \sqrt{\frac{d\gamma(t)}{dt}^T < \mathbf{g}_{\gamma(t)}(\mathbf{x}) > \frac{d\gamma(t)}{dt}} \; dt \right) \tag{22}$$

$$= \min \left(\int_{\mathbf{w_t}}^{\mathbf{w_a}} d_O(\mathbf{du}, < \mathbf{g_w}(\mathbf{x}) >) \right) \tag{23}$$

$$= \min \left(\int_{\mathbf{w_t}}^{\mathbf{w_a}} \sqrt{\frac{1}{N} \sum_{i=1}^{N} |f(\mathbf{x_i}; \mathbf{w} + \mathbf{du}) - \mathbf{f}(\mathbf{x_i}; \mathbf{w})|^2} \right) \tag{24}$$

Therefore, in this section we have shown that finding the geodesic path between $\mathbf{w_t}$, $\mathbf{w_a} \in W$ (eq-22) by minimizing L(γ) finds a series of networks along path γ that minimizes the total distance traversed in the networks output space (eq-24). We also find empirically that minimizing the distance moved in the networks output space results in a series of networks with an output close to that of the trained network ($\mathbf{w_t}$) resulting in a series of high-performance networks. The value of the minimum distance traversed in the output space will depend on the network and task.

Constructing geodesic paths

The construction of geodesics can also be achieved through a 'local approach' that leads to the derivation of the geodesic equation given in the paper (Equation-5 in the manuscript). In the local approach, we seek a minimization of the energy functional S(γ) by applying the Euler-Lagrange method.

In local coordinates, the curve ($\gamma(t)$) is (w$^1(\gamma(t))$, ..., w$^n(\gamma(t))$) and we use the abbreviation

$$\dot{w}^i(t) = \frac{d}{dt}(w^i(\gamma(t)))$$

Then,

$$S(\gamma) = \int_0^1 \sum_{i,j} g_{ij}(\mathbf{w}(\gamma(t)))\dot{w}^i(t)\dot{w}^j(t)$$

Applying the Euler-Lagrange equations for the energy functional (S), we get:

$$\ddot{w}^i + \Gamma^i_{jk}\dot{w}^j\dot{w}^k = 0 \quad \forall i, j, k = 1, 2, .., n \tag{25}$$

$$\Gamma^i_{jk} = \frac{1}{2} \sum_l (g_{il})^{-1} \left(\frac{\partial g_{jl}}{\partial w^k} + \frac{\partial g_{kl}}{\partial w^j} - \frac{\partial g_{jk}}{\partial w^l} \right) \tag{26}$$

The Euler-Lagrange approach finds that minimization of the energy functional (S(γ)) is achieved by paths that also satisfy Eq-25,26. The equation is

a second order ODE, also called the Geodesic equation, wherein the Christoffel symbols (Γ^i_{jk}) encode derivatives of the metric. All paths that solve the global minimization problem also satisfy the geodesic equation (paths generated by the geodesic equation can be thought of as 'zero acceleration paths').