

Towards a Continuous Hyperparameter Representation for Neural Networks

William Guss*

Other Author[†]

Other Author[‡]

July 5, 2017

Abstract

*Email: wguss@berkeley.edu

[†]Email: other@berkeley.edu

[‡]Email: other@berkeley.edu

Contents

1 Planning & Unorganized Results (*)

1.1 Motivation & Goal

In recent years, the success of deep learning has revolutionized machine learning in application to computer vision, reinforcement learning, and natural language processing¹. In addition to their optimal performance, neural networks require little to no feature engineering on the part of the practitioner. However in most recent approaches, considerable would-be feature engineering effort is placed on neural network design; that is, the price of automatically learning features in neural networks is that the practitioner must then specify with some inductive bias what parameterization and network capacity is optimal for learning those features. For example, the choice of convolutional or fully-connected layers in computer vision, or depth, width, and other topological properties of neural networks, all define the set of features which can be—or probabilistically will be—learned over the course of training. To paraphrase Andrew Ng, "coming up with features is difficult, time-consuming, and requires expert knowledge," and by analogy neural network design has become the new feature engineering of machine learning.

In an attempt to eliminate the dependence of deep learning on architecture engineering, two major approaches have been set forth: the first, neural architecture search, specifies architectures as existing in a unified hyperparameter space and then performs various search methods therein; the second, hyperparameter free deep learning, parameterizes the space of architectures so that the aforementioned hyperparameters are themselves learned using the same optimizer as the weights and parameters they specify. In essence neural architecture search is a learned—and sometimes brute force²—approximation of the practitioner, which shares striking similarities to the traditional feature selection methods of statistical machine learning³.

TODO: Discuss various solutions in aggregate i.e. neural architecture search, differentiable methods.

TODO: Suggest new method which does not rely on

1.2 Questions/Hypothesis

- How to do it?
- Is the optimization problem tractable (e.g. will it lead to getting stuck in local optima which are far from the global optimum)?
- Use DFMs to represent width
- What about depth? It might not matter

¹**TODO: cite**

²**TODO: Cite random search of architecture paper if this does exist**

³**TODO: Cite old ML automatic feature search methods**

- Graphons = DFMs (how many times you apply surface to itself determines how deep in the network you go)
- Takes approaches in fabrics and DAS and fully realizes a continuous optimization, how does performance compare in the continuous version?
- How do you scale continuous hyperparameter representation search? What search methods are optimal in this space (gradient)? <https://arxiv.org/abs/1609.01596>
- Is there any substantial benefit from separating the optimization of hyperparameters and parameters using direct feedback alignment?
- Gradient-based Hyperparameter Optimization through Reversible Learning

1.3 Theory

1.3.1 Desired Results

1.3.2 Some Exposition

1.4 Experiments

The following are a set of desired experiments to verify the newly proposed hyperparameter representation.

1.5 Reading List

1.6 Related Notes

- **Continuous Hidden Dimension**
- **Some Thoughts on Local Search on Hidden Units.** Let \mathcal{N} be the \mathfrak{n} -discrete instantiation of the following DFM

$$\mathcal{O} : \boxed{\mathbb{R}^n} \xrightarrow{\mathfrak{d}} \boxed{L^1(E(\gamma))} \xrightarrow{\mathfrak{f}} \boxed{\mathbb{R}}$$

where $E : \mathbb{R} \rightarrow \mathcal{L}(\mathbb{R})$ is a function which parameterizes the domain over which the \mathfrak{f} -functional integrates.

It was concluded in the last note that if $E(\gamma) = [0, \gamma] \in \mathcal{L}(\mathbb{R})$ then we have the following problem for the piecewise constant parameterization of weights on $\mathfrak{f}, \mathfrak{d}$. Let $F : \mathbb{R} \rightarrow \mathbb{R}$

be some loss function, and then computation of the local gradient ascent path gives

$$\begin{aligned}
\frac{\partial F}{\partial \gamma} &= \frac{dF}{dy^2} \frac{\partial y^2}{\partial \gamma} \\
&= \frac{dF}{dy^2} \cdot \left[\frac{\partial}{\partial \gamma} \int_{[0, \gamma]} \sum_{k=1}^{\infty} [\sigma \circ \mathfrak{d}(x)](u) \chi_{k \cdot [0, 1]}(u) W_k^1 d\mu(u) \right]_n \\
&= \frac{dF}{dy^2} \cdot \left[\sum_{k=1}^{\infty} [\sigma \circ \mathfrak{d}(x)](\gamma) \chi_{k \cdot [0, 1]}(\gamma) W_k^1 \right]_n \\
&= \frac{dF}{dy^2} \cdot y_{\lfloor \gamma \rfloor}^1 W_{\lfloor \gamma \rfloor}^1.
\end{aligned}$$

In otherwords, gradient ascent on F with respect to γ will increase γ if the error will decrease when the contribution of the last output neuron is increased (in magnitude); that is, if $\gamma' > \gamma$ then $(\gamma - \lfloor \gamma \rfloor)$ increases, and thus E decreases by virtue of the term

$$\int_{\lfloor \gamma \rfloor \cdot [0, 1]} y^1(u) W_{\lfloor \gamma \rfloor}^1 d\mu(u) = (\gamma - \lfloor \gamma \rfloor) y_{\lfloor \gamma \rfloor}^1 W_{\lfloor \gamma \rfloor}^1$$

increasing. Searching over γ is effectitvely the same as spending extra time changing the weight $W_{\lfloor \gamma \rfloor}^1$ using two linearly dependent parameters, $(\gamma - \lfloor \gamma \rfloor)$ and $W_{\lfloor \gamma \rfloor}^1$, itself⁴.

Thus we are led to the question: *Is hyperparameter search a matter of immediate model accuracy or expected capacity for accuracy, and in that distinction, does optimizing hyperparameters with respect to model accuracy coorespond to optimization on model capactiy and visa versa?* Let us examine this question in the following context.

Above, we noted that a local search on γ decreased error in exactly the same fashion as standard gradient descent, but a step in γ of more than integral amount can increase error. To see this let $k = \lfloor \gamma \rfloor$. When $\Delta\gamma > 1$ then the $(k+1)$ th neuron is then "enabled" so-to-speak. However, this $(k+1)$ th neuron may perform a computation that increases error and so in the next step of gradient descent $\Delta\gamma$ would be negative, retreating away from the added model capacity of a randomly intiialized $(k+1)$ th neuron. That is not to say that γ might not increase again, repeating the process, or in the limit of such oscilations the update $W_{k+1}^1 - \alpha \partial E / \partial W_{k+1}^1 \rightarrow W_{k+1}^1$, will eventually contribute to model accuracy, but relying on these dynamics as a result with no guarentees of convergence is questionable. Despite the fact that \mathcal{N} may need additional model capacity⁵, local search on capacity with respect to accuracy may not yield the required capacity to increase accuracy in the limit.

Baring that local search doesn't necisarrily yield the desired properties, we might now consider a global search. The more general setting is of course considering E as a

⁴An additonal conclusion is, at least by analogy, that local search on $E(\gamma)$ at any one place assumes that adjacent neurons have similar values

⁵There are functions which are unlearnable without a sufficient number of neurons for example.

function of variable support geometry. In particular let $E : C_*^\infty(\mathbb{R}) \rightarrow \mathcal{L}(\mathbb{R})$ so that $f \mapsto \text{supp}(|f|/2 + f/2)$. Note that $C_*^\infty(\mathbb{R})$ is the set of infinitely differentiable functions which vanish on at most a μ -null set. Then computation of a gradient descent step becomes a variational problem

$$\begin{aligned} \frac{\delta F}{\delta f} &= \frac{\partial F}{\partial y^2} \frac{\delta y^2}{\delta f} \\ &= \frac{\partial F}{\partial y^2} \left[\frac{\delta}{\delta f} \int_{E(f)} y^1 \omega_1 \, d\mu \right]_{\mathbf{n}} \\ &= \frac{\partial F}{\partial y^2} \left[\frac{\delta}{\delta f} \lim_{\kappa \rightarrow \infty} \int_{\mathbb{R}} \sigma(\kappa f) y^1 \omega_1 \, d\mu \right]_{\mathbf{n}} \end{aligned}$$

Now we attempt to compute the functional derivative of integration; that is, let $J_\kappa[f] = \int_{\mathbb{R}} \sigma(\kappa f) y^1 \omega_1 \, d\mu$. Then

$$\begin{aligned} \frac{\delta J}{\delta f} &= \lim_{\epsilon \rightarrow 0} \lim_{\kappa \rightarrow \infty} \frac{J_\kappa[f + \epsilon \phi] - J_\kappa[f]}{\epsilon} \\ &= \lim_{\kappa \rightarrow \infty} \left[\frac{d}{d\epsilon} J_\kappa[f + \epsilon \phi] \right]_{\epsilon=0} \\ &= \lim_{\kappa \rightarrow \infty} \left[\int_{\mathbb{R}} \frac{d}{d\epsilon} \sigma(\kappa(f + \epsilon \phi)) y^1 \omega_1 \, d\mu \right]_{\epsilon=0} \\ &= \lim_{\kappa \rightarrow \infty} \left[\int_{\mathbb{R}} \sigma'(\kappa(f + \epsilon \phi)) \kappa \phi y^1 \omega_1 \, d\mu \right]_{\epsilon=0} \\ &= \int_{\mathbb{R}} \lim_{\kappa \rightarrow \infty} \sigma'(\kappa f) \kappa \phi y^1 \omega_1 \, d\mu. \\ &= \int_{\mathbb{R}} \delta(f(u)) \phi(u) y^1(u) \omega_1(u) \, d\mu(u) = \sum_{z \in Z(f)} \phi(z) y^1(z) \omega_1(z) \end{aligned}$$

where $Z(f)$ is the set of zeroes of f . Thus we yield a functional gradient ascent step via the linear approximation

$$\begin{aligned} J[\phi] &= J[f] + \frac{\delta J}{\delta f} [\phi - f] + \frac{1}{2t} \|\phi - f\|^2 \\ 0 &= 0 + \frac{\delta J}{\delta f} + \frac{\phi - f}{t} \\ \phi &= f - t \left(\psi \mapsto \sum_{z \in Z(f)} \psi(z) y^1(z) \omega_1(z) \right). \end{aligned}$$

In addition to the previous update rule, we come to the redundant conclusion that $\delta J / \delta f|_{f=\Gamma} = 0$ for Γ with no zeros, and therefore when $\Gamma < 0$ we have only found a minimum for J . In any case, the gradient in the hard limit of k is zero at every point

at which $f(u)$ is non-zero. Using a soft limit $\kappa \nearrow \infty$ we get an ascent direction in the continuum of our initial search on γ_1 that is

$$\phi = f - \sigma'(\kappa f) \kappa \phi \sum_{k=1}^{\infty} \chi_{k \cdot [0,1]} [\sigma \circ \delta(x)] W_k^1.$$

However, we face the same question as to whether optimization on the loss function would yield ascent in the direction of random vectors for the purpose of capacity and not by their similarity actions to values which decrease error.

- **Fuzzy Vector Spaces as a Solution to Continuous Parameterization of Hidden Dimension.** As we saw in the previous o

1.7 Timeline

T