Rotated Grid Search for Hyperparameter Optimization

Altan Allawala, Killian Rutherford, and Pavan Wadhwa

***Abstract*—This paper proposes a new hyperparameter search method involving elliptical grid transformations and rotations of a grid of probe points. This technique is termed "rotated grid search". We begin by motivating the method by discussing the limitations of random search. A new formalism for more efficiently probing a hyperparameter search space is then proposed. Next, we build a theoretical framework to compare hyperparameter optimization performance of rotated grid search against random search. We then evaluate both search methods empirically to quantify the marginal benefit of using one over the other. Monte-Carlo simulations on various synthetic objective functions show that rotated grid search outperforms random search over the full range of anisotropy explored in this study. Finally, we conduct a case study on a real dataset, rectangles-images, and show that rotated grid search outperforms random search in a high dimensional space.**

***Index Terms*—Random search, grid search, global optimization, model selection, rotated grid search, neural networks, deep learning.**

I Introduction

The aim of statistical learning is to find a function that maps some input features to an output such that an objective function is optimized. Statistical learning can thus be reduced to an optimization problem where the objective function is a function of parameters and hyperparameters (see, for instance, [1]). While parameters can often be optimized through gradient descent, hyperparameters are more difficult to optimize because the gradient of the objective function with respect to the hyperparameters has no analytic expression. Therefore, hyperparameters need to be specified before the parameters can be optimized [2].

As a result, most hyperparameter optimization techniques are based on trying multiple sets of hyperparameters and selecting the set that gives the best objective function value [3]. This is a computationally expensive step, since for each choice of hyperparameters, the parameters need to be refit (for example via gradient descent or tree-based learning). It is therefore important that the candidate hyperparameter sets be chosen to maximize the chances of finding the best objective function value [4], [5].

Two of the most common methods of choosing these candidate sets of hyperparameters are grid search and random search (as studied in [6]). Grid search benefits from being simple to implement whilst being reliable in low dimensional spaces [7], [8]. Conversely, random search tends to be more efficient in high-dimensional spaces because objective functions in real world Machine Learning problems often have low effective dimensionality (see [9] for a detailed discussion). In addition, it is slightly more practical than grid search since the resolution of a search can be changed "on the fly" by simply adding more random search points, unlike grid search where the resolution needs to be pre-defined before the grid points can be created [8].

However, both grid search and random search have certain drawbacks. Grid search suffers from the curse of dimensionality [10] whereby for every additional value of a hyperparameter that the user wishes to probe, the number of requisite trials increases exponentially. Random search, on the other hand, is not deterministic and this can complicate result replication when the number of random points probed is not sufficiently large [11].

In this study, we introduce a new search method, rotated grid search, which combines the attractive features of both grid search and random search. We compare the performance of rotated grid search against random search on synthetic data, and on the same real-world dataset used by [8].

II Rotated Grid Search

An appealing trait of random search, as claimed by [8], is that it probes the objective function with $N = n^2$ distinct points along any given dimension (for a square grid). This is in contrast to grid search which probes the objective function with only n distinct points when projected along any given dimension, since the points are axis-aligned. This becomes especially important for objective functions with low effective dimensionality, as shown in Fig. 1.

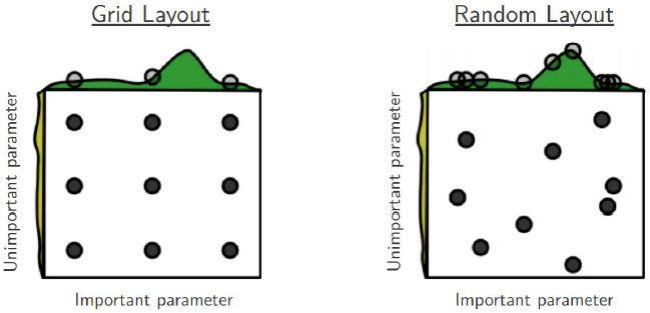


Fig. 1. In a square grid, grid search probes n distinct points along each axis whereas random search probes n^2 distinct points along each axis. Therefore for objective functions with low effective dimensionality (shown in green), random search offers a higher probability of being close to the optimal solution. Image credit [8].

However, a drawback of random search is that its inherently random nature can, at times, result in large pockets of the hyperparameter space being probed too sparsely [12].

Manuscript received June 16, 2022; revised August 27, 2022. This work was supported by J. P. Morgan Chase & Co.

The authors are with the MRG Machine Learning Center of Excellence at J.P.Morgan Chase & Co. in Manhattan, NY 10016, USA (e-mail: altan.allawala@gmail.com, krr2125@columbia.edu, pavan.wadhwa@jpmorgan.com).

doi: 10.18178/ijmlc.2022.12.5.1110

Here we introduce a new hyperparameter search technique, rotated grid search, which preserves the advantage of random search by probing n^2 points along a given dimension, while employing a fully deterministic sampling technique to mitigate the risk of probing pockets of the hyperparameter space too sparsely. Indeed, the deterministic nature of the operation allows us to pick the optimal angle of rotation to ensure that points on the rotated grid are more spread out than points on a random grid, thereby allowing us to probe the hyperparameter space more efficiently. An example of such a rotated grid in two dimensions is shown in Fig. 2.

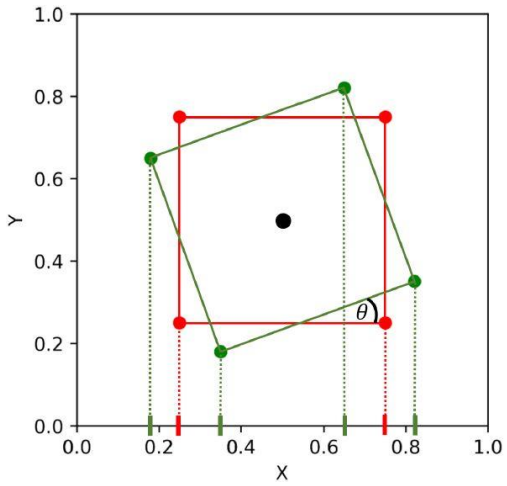


Fig. 2. A schematic showing projections of distinct hyperparameters probed along the x-axis with grid search (red dots) versus rotated grid search (green dots). Whereas grid search probes only 2 distinct x-values, rotated grid search probes 4 distinct x-values despite the same number of total search points, as shown by the green and red dashed lines projected onto the x-axis.

Rotated grid search allows for more distinct hyperparameters to be probed along each axis in a systematic way, overcoming the oft-quoted deficiency of both grid search (only n hyperparameters being probed along a given axis rather than n^2) and random search (pockets of the hyperparameter space may be probed sparsely just by random chance). In the case of a two dimensional grid, we apply an elliptical grid transformation to map from a square grid $(x, y) \in [0,1]$ to a circular grid $(u, v) \in [u^2 + v^2 < 1]$:

$$u = x\sqrt{1 - \frac{y^2}{2}},$$

$$v = y\sqrt{1 - \frac{x^2}{2}}.$$

This mapping ensures that the rotated points remain within the bounds of the grid. A counter-clockwise rotation by an angle θ is then applied to the grid, followed by an inverse elliptical grid transformation on the rotated points to map the circular grid back to a square grid, given by:

$$x = \frac{1}{2}\sqrt{2 + 2\sqrt{2}u + u^2 - v^2} - \frac{1}{2}\sqrt{2 - 2\sqrt{2}u + u^2 - v^2}$$

$$y = \frac{1}{2}\sqrt{2 + 2\sqrt{2}u - u^2 + v^2} - \frac{1}{2}\sqrt{2 - 2\sqrt{2}u - u^2 + v^2}$$

For grid dimensions $n > 2$, the above procedure is performed over all permutations of pairs of dimensions. Thus the general n-dimensional case will have $n(n-1)/2$ rotations.

Before deploying rotated grid search, the optimal angle of rotation must be determined. A rotation of $\theta = 90^o$ recovers a non-rotated grid when the starting grid is equally spaced along both axes. On the other extreme, although an infinitesimally small rotation does indeed yield n^2 distinct points, these points are so close to each other that they effectively probe the same region of the projected hyperparameter space as the underlying grid search. Therefore, the optimum rotation angle lies somewhere between these two extremes. We set a rotation angle of $\theta = 20^o$ for the rest of this study. For a discussion on the choice of θ, see Appendix A.

III Synthetic Dataset

A. Methodology

We begin, for simplicity reasons, by restricting our analysis to two-dimensional hyperparameter spaces. Although classical (negative) loss functions are used as the objective function of learnable parameters, hyperparameters are usually not learnable (i.e. the objective function is not differentiable with respect to hyperparameters and hence gradient descent cannot be leveraged). In fact there is less constraint on how the objective function may vary with respect to hyperparameters because they emerge from the machine learning model formalism itself. To account for this arbitrary relationship between the objective function and model hyperparameters, we model the objective function in hyperparameter space, $J(x, y)$, as a superposition of q independent bivariate normal distributions:

$$J(x,y) = c\sum_{i=1}^{q} \quad exp\left[-\left(\frac{x-\mu_x^{(i)}}{\sqrt{2}\sigma_x^{(i)}}\right)^2 - -\left(\frac{y-\mu_y^{(i)}}{\sqrt{2}\sigma_y^{(i)}}\right)^2\right]$$

where c is a normalization factor and μ_x, μ_y are the means of each of the q independent bivariate normal distributions along the x and y directions respectively (where x and y correspond to the directions of the hyperparameter grid). These means are random variables sampled from random uniform distributions between zero and one. The standard deviation of the first variate, σ_x, is drawn from a random uniform distribution between zero and σ_{max} (treated here as a parameter). In other words,

$$\mu_x \sim Uniform(0,1),$$
$$\mu_y \sim Uniform(0,1),$$
$$\sigma_x \sim Uniform(0, \sigma_{max}).$$

As for σ_y, we define it as,

$$\sigma_y = \sigma_x/\beta,$$

where β is an anisotropy factor. The anisotropy of the objective function in hyperparameter space is therefore defined as the ratio of the standard deviations of the bivariate distributions along the two axes. An isotropic bivariate distribution is denoted by $\beta = 1$ whereas deviations from this provides a measure of anisotropy. For example, when $\beta > 1$, the multivariate distributions are compressed along the second (y) axis, causing the objective function to change more abruptly in that direction (see Fig. 3). Similarly when $\beta < 1$, the distributions are expanded along the second axis which is equivalent to compression along the first axis (after adjusting for the new scale), hence yielding the same level of anisotropy. Due to this symmetry, we only show results for

$\beta > 1$ in this study.

Increasing q, which denotes the number of superimposed bivariate normal distributions, increases the number of peaks (local maxima) of the objective function in hyperparameter space. Fig. 3 illustrates the impact of β and q on the objective function for different (q, β) combinations for $\sigma_{max} = 0.3$. This choice of σ_{max}, used throughout this study, prevents the objective function's constituent Gaussian distributions from becoming too flat over the domain. This allows for a rich landscape over which to perform hyperparameter optimization, thereby allowing us to perform meaningful comparisons between different search techniques.

For example, the top right plot of Fig. 3 is produced by generating $q = 10$ different bivariate normal distributions (each with a randomly drawn μ_x, μ_y and σ_x) and adding them together. Each local maximum visible on the chart (shown by yellower regions) corresponds to the peak of each of the constituent bivariate normal distributions. The final objective function is normalized such that its maximum value across all peaks is unity without loss of generality.

In this study, we perform Monte-Carlo simulations [13] on such objective functions to quantify the comparative performance of random search and rotated grid search. Table I describes each of the free parameters in this study. Of the five free parameters in this study, the first three (β, q and σ_{max}) control the shape of the objective function, while the last three (n, N and θ) control the specifics of the search method.

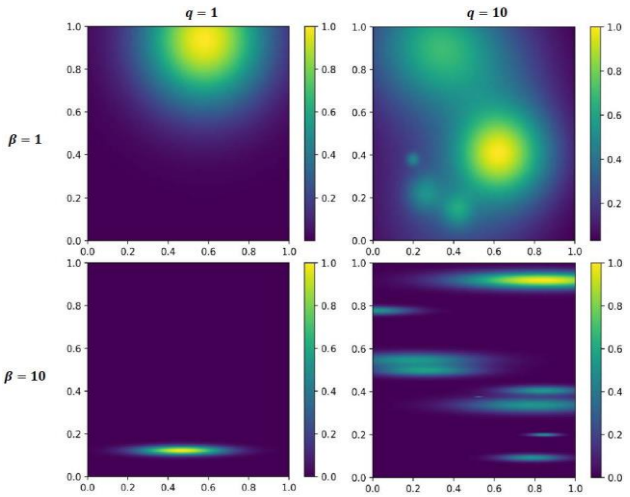


Fig. 3. Heat maps of random initializations of 2-dimensional objective functions for four separate (q, β) combinations. Peaks in the objective functions are denoted by yellow. q denotes the number of peaks (local optima) of the objective function, while β is a measure of the anisotropy of the objective function. Larger values of β imply higher anisotropy.

TABLE I: A DESCRIPTION OF THE FREE PARAMETERS USED IN THIS STUDY

Parameter	Description
β	Anisotropy of the objective function in two dimensions; $\sigma_y = \sigma_x/\beta$
q	Number of local maxima (peaks) of the objective function (number of superimposed Gaussians).
σ_{max}	The standard deviation along the x-direction of each peak, σ_x, is sampled from a uniform distribution within the interval [0, σ_{max}]. In this study we set $\sigma_{max} = 0.3$ (to prevent the objective function's constituent Gaussian distributions from becoming too flat over the domain).
n, N	Number of points, n, over which the objective function is evaluated in the x-direction. Using a square grid, the total number of points over which the objective function is evaluated is $N = n^2$. In this experiment we set $n = 6$ (justified below).
θ	Angle by which the square grid is rotated in "rotated grid search". The rotation produces the points over which the objective function is subsequently evaluated. In this study we set $\theta = 20^o$ (justified in Appendix A).

More precisely, we say that Search Method A beats Search Method B in a trial when the maximum value of the objective function found (over all $N = n^2$ evaluation points) by Search Method A is larger than the maximum value of the objective function found (over all $N = n^2$ evaluation points) by Search Method B. The probability of Search Method A beating Search Method B is then calculated as the expected number of times that the above condition is met over all Monte-Carlo trials. Note that each Monte-Carlo simulation consists of 10,000,000 trials to ensure sufficient convergence.

The search methods are evaluated on unit grids (as shown before) for a fixed number of evaluation points N and a fixed $\sigma_{max} = 0.3$ (to prevent the objective function's constituent Gaussian distributions from becoming too flat over the domain). In all search method cases, the N evaluation points are determined at the start. For grid search, n points are selected along both unit axes such that these points are equidistant from each other. For random search, points are uniformly randomly sampled along each axis to obtain the grid coordinates. Finally, the proposed rotated grid search methodology starts with the generated grid search points and rotates them as described in Section II.

B. Results

We wish to compare rotated grid search against random search for varying anisotropy factors. However, in order to simplify the problem and confirm literature results, we first investigate the effect of n and q on random and grid search.

As mentioned previously, random search probes the objective function with more distinct points along a given dimension than grid search. While this provides many benefits, its random nature can at times result in large pockets of hyperparameter space being probed too sparsely. However, this drawback becomes less important if the objective function has high anisotropy since the objective function varies along one dimension much less than along the other dimension. Therefore we expect the probability of random search beating grid search to increase as the anisotropy of the objective function (β) increases, which is confirmed in Fig. 4.

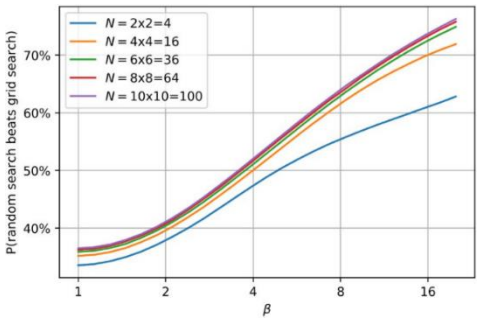


Fig. 4. Probability that random search beats grid search as a function of the anisotropy factor β (log 2 -scale) for varying number of evaluation points N for an objective function consisting of a single peak (q = 1) and σ_{max} = 0.3.

The results show that the probability of random search beating grid search increases when the anisotropy β of the objective function increases. The probability is higher than 50% when the anisotropy factor is larger than 4. The probability also increases as the number of evaluation points of the objective function, $N = n^2$, increases, converging around $N = 6^2 = 36$. Therefore for the remainder of this study we hold the total number of evaluation points fixed at 6 along each dimension in order to make the analysis more tractable.

Hence each Monte-Carlo trial for each search technique consists of 36 point evaluations of the objective function within the hyperparameter space.

It is also instructive to look at the same relationship (probability of random search beating grid search vs the anisotropy factor β) as the number of peaks q) of the objective function is varied. Fig. 5 shows that the performance spread between random search and grid search diminishes as the number of peaks of the objective function increases. Intuitively, this is because increasing the number of peaks (q) increases the probability of being close to a local maximum. Hence the search method itself starts to matter less.

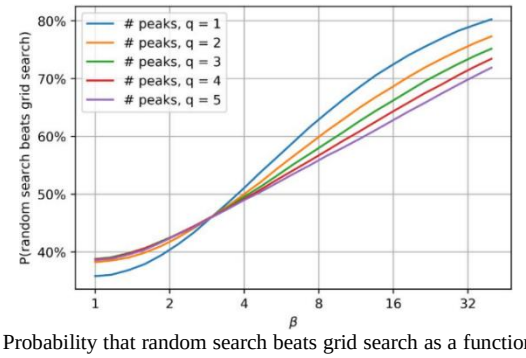


Fig. 5. Probability that random search beats grid search as a function of the anisotropy factor β (log2-scale) for different number of peaks of the objective function ($N = 6^2 = 36$ points) and $\sigma_{max} = 0.3$.

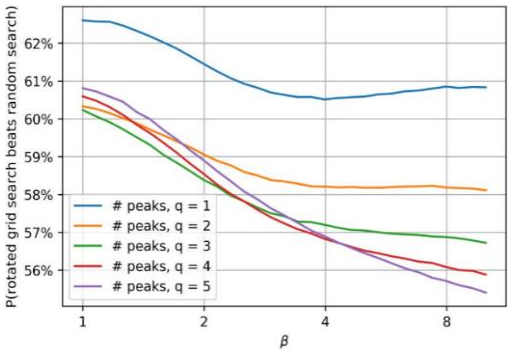


Fig. 6. Probability that rotated grid search ($\theta = 20^o$) beats random search as a function of the anisotropy factor β (log 2 -scale) for varying number of peaks q; $\sigma_{max} = 0.3$ and $N = 6^2 = 36$ search points.

These results were verified to hold over a wide range of $\sigma_{max} \in [0.1, 1]$, and are in line with results presented in the literature. Given these results, rotated grid search (with $\theta = 20^o$) was compared against random search for varying anisotropy factors, β, as shown in Fig. 6. Since both rotated grid search and random search contain non axis-aligned points, it is instructive to compare them against each other, as it may not be initially clear which would perform better. We find that rotated grid search beats random search over the full range of anisotropies explored as shown in Fig. 6. As before with random search over grid search, the outperformance of rotated grid search over random search declines as the number of peaks increases. This is again because increasing the number of peaks increases the probability of being close to a local maximum and thus the search method itself starts to matter less.

IV Case Study

Given that the results in Section III.B show rotated grid search outperforming random search on simulated 2-dimensional objective functions, we conduct a case study to compare these two hyperparameter search methods on a real dataset.

A. Dataset

To compare rotated grid search over random search on a real dataset, we follow [8], [14] and [15] and use the rectangles-images dataset[1]. This is a dataset of outlines of rectangles where each image is labeled as either tall or wide. The rectangles are filled with a natural image patch, for example a section of a normal image. The background is another natural image patch (see Fig. 7 as an example). The image dimensions are 28×28 pixels. The height and width of the rectangles are sampled uniformly under the constraint that the area covered by the rectangles are between one to three quarters of the total image. Additionally, the length and width of each rectangle is constrained to be at least 10 pixels and the difference is forced to be at least 5 pixels. An example is displayed in Fig. 7.

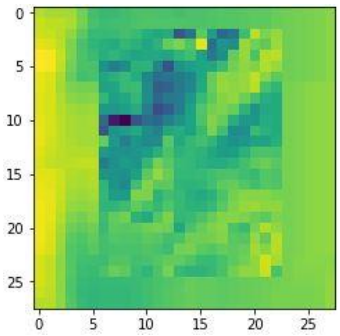


Fig. 7. An example of an image from the rectangles-images dataset. Each image is labeled as either tall or wide, depending on which is larger.

In this study we used 10,000 training examples, 2,000 validation examples, and 50,000 testing examples.

B. Neural Network

In [8], hyperparameter search was performed with a neural network trained on the rectangles-images dataset [16]. They showed that random search outperformed grid search, which confirms our results obtained in Section III.B. Following that work, we perform rotated grid search on this dataset and show that rotated grid search in turn outperforms random search. Table II lists the hyperparameters within the search space explored:

[1] Datasets can be found at http://www.dmi.usherb.ca/~larocheh/mlpython/datasets.html#module-datasets.rectangles_images

TABLE II: THE MOST IMPORTANT HYPERPARAMETERS IN DECREASING ORDER, WITH THEIR CORRESPONDING SEARCH RANGE

Hyperparameter	Search range
Initial learning rate	[0.01, 100]
t_0	[3e2, 3e4]
Lecun scaling factor	[0.2, 2.0]
Number of hidden units	[18, 1024]
Initial weight	[0, 1]

These hyperparameters are selected for the search because they contribute the largest variance to the objective function. In addition, we use a sigmoid activation function, a batch size of 20 and set the value of the L_2 coefficient to 3.1e-7. For a detailed description of all hyperparameters, see [8].

C. Results

We compare random search and rotated grid search over a wide range of anisotropies β (defined in Section III.A) by using "fake" hyperparameters [2] in the search space. By substituting important hyperparameters for these "fake" hyperparameters, we are lowering the rank structure of the objective function, which decreases its anisotropy. Five different experiments are performed corresponding to varying levels of anisotropy, as shown in Fig. 8.

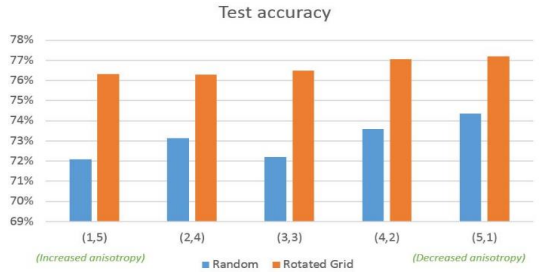


Fig. 8. Barchart comparing the test accuracy of rotated grid search and random search on the rectangles-images dataset. The horizontal axis denotes the number of ("important","fake") hyperparameters in the search space. In all five experiments, the two methods search over a total of six hyperparameters using 64 trials.

The first experiment (1,5) consists of the most important hyperparameter (initial learning rate) with 5 fake hyperparameters. Since objective function does not vary along the directions of these fake hyperparameters, this experiment is a proxy for a large anisotropy (β). The next experiment (2,4) replaces one of the fake hyperparameters with the second most important hyperparameter. Since the objective function will now vary substantially in two out of the six directions, it is expected to be less anisotropic. In this way, the last experiment (5,1) consists of all five hyperparameters with only a single fake hyperparameter and simulates the least anisotropic objective function (i.e. smallest β). Rotated grid search outperforms random search in all five experiments. This is in line with the theoretical expectations and numerical simulations of random objective functions in previous sections.

V CONCLUSION

In this paper, we show that rotated grid search outperforms random search over the full range of anisotropies explored in this study on both a synthetic dataset and a high dimensional real dataset (rectangles-images). Given that the anisotropy of an objective function is usually not known a priori, the results suggest that rotated grid search is a better default choice hyperparameter optimization scheme.

APPENDIX

As mentioned in Section II, it is important that the grid of any search method have distinct values along each dimension because the objective function can have low (or no) dependence on certain hyperparameters, which in turn reduces its effective dimensionality.

Additionally, we require these distinct points along each axis to be spaced out as evenly as possible whilst still maintaining their distinctness. To ensure this, we look at the distance between a randomly chosen optimum point in the domain, and each of the n^2 evaluation points, and take the minimum distance across all n^2 points. The expected minimum distance can then be calculated by averaging this minimum distance across a large number of simulations of the randomly chosen optimum points. A smaller expected minimum distance implies that points are more evenly spaced along that axis. Importantly, we calculate this distance after projecting the points onto an axis, so as to measure the expected distance from the optimum even for objective functions where only one of the two features are important. We therefore define the metric as,

$$M = E\left[\min_i\left(\left|\vec{P} - \vec{X}_i\right|_x\right)\right]$$

where $|\cdot\cdot\cdot|_x$ is the length of an arbitrary vector after projecting it onto the x-axis, P is a randomly chosen optimal point, and X_i is the i^{th} evaluation point in the search.

Based on Fig. 9, the rotation angle can then be chosen as the value that minimizes this expected distance between the projected optimum point and the closest projected evaluation point, which gives approximately $\theta = 20^o$.

Note that although $\theta = 20^o$ is one such candidate, the plot also shows that as the number of probe points N increases, this range expands. For example at $N = 64$, any $5^o < \theta < 40^o$ yield approximately the same expected minimum distance. In addition, due to mirror symmetry, $50^o < \theta < 85^o$ have identical expected minimum distances too. Therefore although $\theta = 20^o$ was used for the simulations in this study, there is in fact a large range of equally good choices of θ provided that the number probe points is not small. In fact, the only values of θ to avoid are in the vicinity of 0^o, 45^o and 90^o (in the two-dimensional case). This is because at 0^o or 90^o, the rotated grid collapses to a traditional grid and at 45^o, whilst not entirely collapsing to a traditional grid, many grid points do overlap (which reduces the effectiveness of the probe). As dimensionality increases, the probability of points overlapping becomes vanishingly small because of increased mixing between the dimensions (i.e. a rotated grid of dimension n is generated by $O(n^2)$ rotations between dimensions, as mentioned in Section II). Therefore the final choice of $\theta = 20^o$, although not unique, is a safe choice to prevent any symmetry collapse, hence maintaining reasonable uniqueness of probe points.

[2] A "fake" hyperparameter is one that does not affect the neural network performance, i.e. a dummy hyperparameter in the search that was not used as input to the neural network. Given constant real hyperparameters, any change in the fake hyperparameters will not affect the neural network architecture. Their only purpose is to artificially expand the search space.

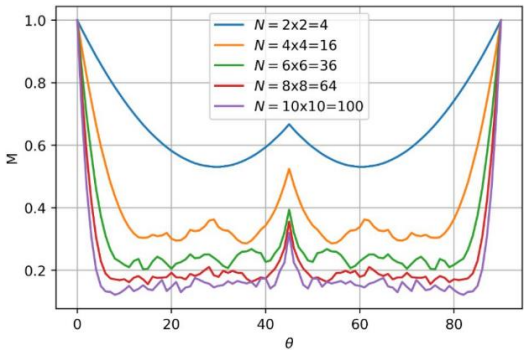


Fig. 9. Expected minimum distance, M, between grid points and a randomly generated optimum point as a function of rotation angle, θ, for various values of N. M has been normalized for each N.

Conflict of Interest

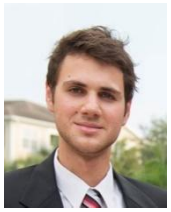
The authors declare no conflict of interest.

Author Contributions

A. Allawala, K. Rutherford and P. Wadhwa formulated the study. A. Allawala and K. Rutherford built the neural network models. All authors wrote the paper and have approved the final version.

References

[1] V. N. Vapnik. *Statistical Learning Theory*, Wiley-Interscience, 1998.
[2] I Czogiel, K Luebke, and C Weihs, "Response surface methodology for optimizing hyperparameters," Technical Report, *Universitat Dortmund Fachbereich Statistik*, 2005.
[3] F. Hutter, "Automated configuration of algorithms for solving hard computational problems," PhD thesis, University of British Columbia, 2009.
[4] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, London, UK, 1995.
[5] B. James, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proc. International Conference on Machine Learning*, PMLR, 2013.
[6] J. S Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyperparameter optimization," *Advances in Neural Information Processing Systems*, pp. 2546–2554, 2011.
[7] A. Hussain and S. A. Ludwig, "Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization," in *Proc. 2021 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2021.
[8] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
[9] L. Liam and A. Talwalkar, "Random search and reproducibility for neural architecture search," *Uncertainty in Artificial Intelligence*, PMLR, 2020.
[10] M.-L. Cauwet *et al*., "Fully parallel hyperparameter search: Reshaped space-filling," in *Proc. International Conference on Machine Learning*, PMLR, 2020.
[11] R. E. Caflisch, W. J. Morokoff, and A. B. Owen, "Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension," Department of Mathematics, University of California, Los Angeles, 1997.
[12] R. Bellman, "Curse of dimensionality," *Adaptive Control Processes: A Guided Tour*, Princeton, NJ, 3:2, 1961.
[13] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*, vol. 10, John Wiley & Sons, 2016.
[14] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, " An empirical evaluation of deep architectures on problems with many factors of variation," in *Proc. the 24th International Conference on Machine Learning*, 2007, pp. 473–480.
[15] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising auto encoders," in *Proc. International Conference on Machine Learning*, 2008.
[16] Y. Bengio and X. Glorot, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. the 13th International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.



Altan Allawala holds a PhD and MSc in theoretical physics from Brown University, USA and a BSc in physics from University of Melbourne, Australia.

He is a quant researcher at Polymer Capital, a market-neutral quantitative hedge fund, where he is building out the alpha capture arm of the firm.

Prior to his current role, Altan was vice president at J. P. Morgan's MRG Machine Learning Center of Excellence where was the project lead of a proprietary financial machine learning package and conducted fundamental research into machine learning topics relevant to the industry with a focus on anomaly detection.

Killian Rutherford holds an MSc in computer science, MSc in applied physics from Columbia University, USA and a BA in natural sciences from University of Cambridge, UK.

He is the vice president at J. P. Morgan's MRG Machine Learning Center of Excellence where he is involved in evaluating model risk for a variety of machine learning models in algorithmic trading and consumer banking domains. He also conducts research across various machine learning topics within the financial industry.

Pavan Wadhwa holds a PhD and MBA in finance from the University of Texas at Austin and a bachelor's degree in electrical engineering from the Indian Institute of Technology, Kanpur.

He is the managing director in the model risk group at J. P. Morgan where he is responsible for reviewing models related to deposits, fee/revenue and anti-money laundering. Additionally, he has established a Center of Excellence to validate Machine Learning models.

Pavan spent 15 years on various trading desks of J. P. Morgan in NY and London generating thematic and Relative Value trade ideas for traders and clients, eventually running global rates strategy. Prior to his current role, he was head of US interest rate strategy at Blackrock.