Randomized neural networks are immortalized in this well-known AI Koan:

In the days when Sussman was a novice, Minsky once came to him as he sat hacking at the PDP-6.

“What are you doing?” asked Minsky. “I am training a randomly wired neural net to play tic-tac-toe,” Sussman replied. “Why is the net wired randomly?” asked Minsky. Sussman replied, “I do not want it to have any preconceptions of how to play.”

Minsky then shut his eyes. “Why do you close your eyes?” Sussman asked his teacher. “So that the room will be empty,” replied Minsky. At that moment, Sussman was enlightened.

We analyze shallow random networks with the help of concentration of measure inequalities. Specifically, we consider architectures that compute a weighted sum of their inputs after passing them through a bank of arbitrary randomized nonlinearities. We identify conditions under which these networks exhibit good classification performance, and bound their test error in terms of the size of the dataset and the number of random nonlinearities.

1 Introduction

In the earliest days of artificial intelligence, the bottom-most layer of neural networks consisted of randomly connected “associator units” that computed random binary functions of their inputs [1]. These randomized shallow networks have largely been superceded by optimally, or nearly optimally, tuned shallow architectures such as weighted sums of positive definite kernels (as in Support Vector Machines), or weighted sums of weak classifiers (as in Adaboost). But recently, architectures that randomly transform their inputs have been resurfacing in the machine learning community [2, 3, 4, 5], largely motivated by the fact that randomization is computationally cheaper than optimization. With the help of concentration of measure inequalities on function spaces, we show that training a shallow architecture by randomly choosing the nonlinearities in the first layer results in a classifier that is not much worse than one constructed by optimally tuning the nonlinearities. The main technical contributions of the paper are an approximation error bound (Lemma 1), and a synthesis of known techniques from learning theory to analyze random shallow networks.

Consider the problem of fitting a function \( f : \mathcal{X} \to \mathbb{R} \) to a training data set of \( m \) input-output pairs \( \{x_i, y_i\}_{i=1}^m \), drawn iid from some unknown distribution \( P(x, y) \), with \( x_i \in \mathcal{X} \) and \( y_i \in \pm 1 \). The fitting problem consists of finding an \( f \) that minimizes the empirical risk

\[
R_{\text{emp}}[f] = \frac{1}{m} \sum_{i=1}^{m} c(f(x_i), y_i). \tag{1}
\]

The loss \( c(y, y') \) penalizes the deviation between the prediction \( f(x) \) and the label \( y \). Popular choices for \( c \) are the hinge loss, \( \max(0, 1 - yy') \), used in the Support Vector Machine [6], the exponential
Formally, we show that Algorithm 1 returns a function that has low true risk. The true risk, \( R\|f\| \), of a function \( f \) is its expected loss on as-yet-unseen test points generated from the training distribution:

\[
R[f] \equiv \mathbb{E}_{(x,y) \sim P} c(f(x),y),
\]

The following theorem states that with very high probability, Algorithm 1 returns a function whose true risk is near the lowest true risk attainable by an infinite-dimensional class of functions \( \mathcal{F}_p \) defined below:

**Theorem 1 (Main result).** Let \( p \) be a distribution on \( \Omega \), and let \( \phi \) satisfy \( \sup_{x,w} |\phi(x; w)| \leq 1 \). Define the set

\[
\mathcal{F}_p \equiv \left\{ f(x) = \int_{\Omega} \alpha(w)\phi(x; w) \, dw \mid |\alpha(w)| \leq Cp(w) \right\}.
\]
Suppose \( c(y, y') = c(yy') \), with \( c(yy') \) \( L \)-Lipschitz. Then for any \( \delta > 0 \), if the training data \( \{x_i, y_i\}_{i=1,...,m} \) are drawn iid from some distribution \( P \), Algorithm 1 returns a function \( \hat{f} \) that satisfies

\[
\mathbb{R}[\hat{f}] - \min_{f \in \mathcal{F}_p} \mathbb{R}[f] \leq O \left( \left( \frac{1}{\sqrt{m}} + \frac{1}{\sqrt{K}} \right) LC \sqrt{\frac{\log \frac{1}{\delta}}{m}} \right)
\]

(7)

with probability at least \( 1 - 2\delta \) over the training dataset and the choice of the parameters \( w_1, \ldots, w_K \).

The set \( \mathcal{F}_p \) is quite rich. It consists of functions whose weights \( \alpha(w) \) decays more rapidly than the given sampling distribution \( p \). For example, when \( \phi(x; w) \) are sinusoids with frequency \( w \), \( \mathcal{F}_p \) is the set of all functions whose Fourier transforms decay faster than \( C p(w) \). The proof of the theorem provides explicit values for the constants in the big \( O \) notation.

2 Proof of the Main Theorem

Algorithm 1 returns a function that lies in the random set

\[
\hat{F}_w = \left\{ f(x) = \sum_{k=1}^{K} \alpha_k \phi(x; w_k) \mid |\alpha_k| \leq \frac{C}{\sqrt{K}} \right\}.
\]

(8)

The bound in the main theorem can be decomposed in a standard way into two bounds:

1. An approximation error bound that shows that the lowest true risk attainable by a function in \( \hat{F}_w \) is not much larger than the lowest true risk attainable in \( \mathcal{F}_p \) (Lemma 2).

2. An estimation error bound that shows that the true risk of every function in \( \hat{F}_w \) is close to its empirical risk (Lemma 3).

The following Lemma is helpful in bounding the approximation error:

**Lemma 1.** Let \( \mu \) be a measure on \( \mathcal{X} \), and \( f^* \) a function in \( \mathcal{F}_p \). If \( w_1, \ldots, w_K \) are drawn iid from \( p \), then for any \( \delta > 0 \), with probability at least \( 1 - \delta \) over \( w_1, \ldots, w_K \), there exists a function \( \hat{f} \in \hat{F}_w \) so that

\[
\sqrt{\int_{\mathcal{X}} (\hat{f}(x) - f^*(x))^2 \, d\mu(x)} \leq \frac{C}{\sqrt{K}} \left( 1 + \sqrt{2 \log \frac{1}{\delta}} \right).
\]

(9)

The proof relies on Lemma 4 of the Appendix, which states that the average of bounded vectors in a Hilbert space concentrates towards its expectation in the Hilbert norm exponentially fast.

**Proof.** Since \( f^* \in \mathcal{F}_p \), we can write \( f^*(x) = \int_{\mathcal{X}} \alpha(w) \phi(x; w) \, dw \). Construct the functions \( f_k = \beta_k \phi(\cdot; w_k) \), \( k = 1 \ldots K \), with \( \beta_k = \frac{\alpha(w_k)}{p(w_k)} \), so that \( \mathbb{E} f_k = f^* \). Let \( \hat{f}(x) = \sum_{k=1}^{K} \beta_k \phi(x; w_k) \) be the sample average of these functions. Then \( \hat{f} \in \hat{F}_w \) because \( |\beta_k/K| \leq C/K \). Also, under the inner product \( \langle f, g \rangle = \int f(x)g(x) \, d\mu(x) \), \( \|\beta_k \phi(\cdot; w_k)\| \leq C \). The claim follows by applying Lemma 4 to \( f_1, \ldots, f_K \) under this inner product. \( \square \)

**Lemma 2** (Bound on the approximation error). Suppose \( c(y, y') \) is \( L \)-Lipschitz in its first argument. Let \( f^* \) be a fixed function in \( \mathcal{F}_p \). If \( w_1, \ldots, w_K \) are drawn iid from \( p \), then for any \( \delta > 0 \), with probability at least \( 1 - \delta \) over \( w_1, \ldots, w_K \), there exists a function \( \hat{f} \in \hat{F}_w \) that satisfies

\[
\mathbb{R}[\hat{f}] \leq \mathbb{R}[f^*] + \frac{LC}{\sqrt{K}} \left( 1 + \sqrt{2 \log \frac{1}{\delta}} \right).
\]

(10)

**Proof.** For any two functions \( f \) and \( g \), the Lipschitz condition on \( c \) followed by the concavity of square root gives

\[
\begin{align*}
\mathbb{R}[f] - \mathbb{R}[g] &= \mathbb{E} c(f(x), y) - c(g(x), y) \\
&\leq \mathbb{E} |c(f(x), y) - c(g(x), y)| \\
&\leq L \mathbb{E} |f(x) - g(x)| \\
&\leq L \sqrt{\mathbb{E} (f(x) - g(x))^2}.
\end{align*}
\]

(11)

The lemma then follows from Lemma 1. \( \square \)
A standard result from statistical learning theory states that for a given choice of \( w_1, \ldots, w_K \) the empirical risk of every function in \( \hat{F}_w \) is close to its true risk.

**Lemma 3** (Bound on the estimation error). Suppose \( c(y, y') = c(yy') \), with \( c(yy') \) L-Lipschitz. Let \( w_1, \ldots, w_K \) be fixed. If \( \{x_i, y_i\}_{i=1}^{m} \) are drawn iid from a fixed distribution, for any \( \delta > 0 \), with probability at least \( 1 - \delta \) over the dataset, we have

\[
\forall f \in \hat{F}_w, \quad |R[f] - R_{emp}[f]| \leq \frac{1}{\sqrt{m}} \left( 4LC + 2|c(0)| + LC \sqrt{\frac{1}{2} \log \frac{1}{\delta}} \right). \tag{13}
\]

**Proof sketch.** By Hölder, the functions in \( \hat{F}_w \) are bounded above by \( C \). The Rademacher complexity of \( \hat{F}_w \) can be shown to be bounded above by \( C/\sqrt{m} \) (see the Appendix). The theorem follows by results from [12] which are summarized in Theorem 2 of the Appendix.

**Proof of Theorem 1.** Let \( f^* \) be a minimizer of \( R \) over \( F \), \( \hat{f} \) a minimizer of \( R_{emp} \) over \( \hat{F}_w \) (the output of the algorithm), and \( \hat{f}^* \) a minimizer of \( R \) over \( \hat{F}_w \). Then

\[
R[\hat{f}] - R[f^*] = R[\hat{f}] - R[\hat{f}^*] + R[\hat{f}^*] - R[f^*] \tag{14}
\]

\[
\leq |R[\hat{f}] - R[\hat{f}^*]| + |R[\hat{f}^*] - R[f^*]|. \tag{15}
\]

The first term in the right side is an estimation error: By Lemma 3, with probability at least \( 1 - \delta \), \( |R[\hat{f}^*] - R_{emp}[\hat{f}^*]| \leq \epsilon_{est} \) and simultaneously, \( |R[\hat{f}] - R_{emp}[\hat{f}]| \leq \epsilon_{est} \), where \( \epsilon_{est} \) is the right side of the bound in Lemma 3. By the optimality of \( \hat{f} \), \( R_{emp}[\hat{f}] \leq R_{emp}[\hat{f}^*] \). Combining these facts gives that with probability at least \( 1 - \delta \), \( |R[\hat{f}] - R[f^*]| \leq 2\epsilon_{est} = \frac{2}{\sqrt{m}} \left( 4LC + 2|c(0)| + LC \sqrt{\frac{1}{2} \log \frac{1}{\delta}} \right) \).

The second term in Equation (15) is the approximation error, and by Theorem 1, with probability at least \( 1 - \delta \), it is bounded above by \( \epsilon_{app} = \frac{LC}{\sqrt{m}} \left( 1 + \sqrt{2\log \frac{1}{\delta}} \right) \).

By the union bound, with probability at least \( 1 - 2\delta \), the right side of Equation (15) is bounded above by \( 2\epsilon_{est} + \epsilon_{app} \).

### 3 Related Work

Greedy algorithms for fitting networks of the form (2) have been analyzed, for example, in [7, 11, 9]. Zhang [13] analyzed greedy algorithms and a randomized algorithm similar to Algorithm 1 for fitting sparse Gaussian processes to data, a more narrow setting than we consider here. He derived bounds on the performance by casting these methods as stochastic descent.

Approximation error bounds such as that of Maurey [11][Lemma 1], Girois [14] and Gnecco and Sanguineti [15] rely on random sampling to guarantee the existence of good parameters \( w_1, \ldots, w_K \), but they require access to the representation of \( f^* \) to actually produce these parameters. These approximation bounds cannot be used to guarantee the performance of Algorithm 1 because Algorithm 1 is oblivious to the data, and therefore \( f^* \), when it generates the parameters. Lemma 2 differs from these bounds in that it relies on \( f^* \) only to generate the weights \( \alpha_1, \ldots, \alpha_K \), but it remains oblivious to \( f^* \) when generating the parameters, sampling them from \( p \) instead. Furthermore, because \( \hat{F}_w \) is smaller than the classes considered by [11, 15], the approximation error rate in Lemma 1 matches those of existing approximation error bounds.

### 4 Experiments

We empirically demonstrate the benefits of random featurization in this section. For more empirical examples, see [2, 3, 4, 5].

We compared a simplified version of Algorithm 1 where the constraint (4) was replaced by a quadratic penalty on \( \alpha \) against Adaboost on three classification problems: The adult dataset
Figure 1: Comparisons between Random Kitchen Sinks and Adaboosted decision stumps on adult (first row), activity (second row), and KDDCUP99 (third row). The first column plots test error of each classifier as a function of $K$. The accuracy of Random Kitchen Sinks catches up to that of Adaboost as $K$ grows. The second column plots the total training and testing time as a function of $K$. For a given $K$, Random Kitchen Sinks is between two and three orders of magnitude faster than Adaboost. The third column combines the previous two columns. It plots testing+training time required to achieve a desired error rate. For a given error rate, Random Kitchen Sinks is between one and three orders of magnitude faster than Adaboost.
has roughly 32,000 training instances. Each categorical variable was replaced by a binary indicator variable over the categories, resulting in 123 dimensions per instance. The test set consists of 15,000 instances. KDDCUP99 is a network intrusion detection problem with roughly 5,000,000 127-dimensional training instances, subsampled to 50,000 instances. The test set consists of 150,000 instances. activity is a human activity recognition dataset with 20,0000 223-dimensional instance, of which about 200 are irrelevant for classification. The test set consists of 50,000 instances. The datasets were preprocessed by zero-meaning and rescaling each dimension to unit variance. The feature functions in these experiments were decision stumps \( \phi(x; w) = \text{sign}(x_{w_d} - w_t) \), which simply determine whether the \( w_d \) th dimension of \( x \) is smaller or greater than the threshold \( w_t \). The sampling distribution \( p \) for Random Kitchen Sinks drew the threshold parameter \( w_t \) from a normal distribution and the coordinate \( w_d \) from a uniform distribution over the coordinates. For some experiments, we could afford to run Random Kitchen Sinks for larger \( K \) than Adaboost, and these runs are included in the plots. We used the quadratic loss, but find no substantial differences in quality under the hinge loss (though there is degradation in speed by a factor of 2-10). We used MATLAB optimized versions of Adaboost and Random Kitchen Sinks, and report wall clock time in seconds.

Figure 1 compares the results on these datasets. Adaboost expends considerable effort in choosing the decision stumps and achieves good test accuracy with a few of them. Random Kitchen Sinks requires more nonlinearities to achieve similar accuracies. But because it is faster than Adaboost, it can produce classifiers that are just as accurate as Adaboost’s with more nonlinearities in less total time. In these experiments, Random Kitchen Sinks is almost as accurate as Adaboost but faster by one to three orders of magnitude.

5 Discussion and Conclusions

These experiments show that by selecting many random nonlinearities, one can achieve the same accuracy as a greedy algorithm that carefully selects the nonlinearities in significantly less training time. This result is consistent with the interpretation that the role of the nonlinearities in a shallow network is to transform the inputs so that they are linearly separable by the upper layer of the network. Lemma 1 and the experiments point out that random featurization is an effective way to find such a transformation. Various hardness of approximation lower bounds for fixed basis functions exist (see, for example [11]). The guarantee in Lemma 1 avoids running afoul of these lower bounds because it does not seek to approximate every function in \( F_p \), simultaneously, but rather only the true risk minimizer with high probability.

It may be surprising that Theorem 1 holds even when the feature functions \( \phi \) are nearly orthogonal. The result works because the importance sampling constraint \( |\alpha(w)| \leq C p(w) \) ensures that a feature function does not receive a large weight if it is unlikely to be sampled by \( p \). When the feature functions are highly linearly dependent, better bounds can be obtained because any \( f(x) = \int \alpha(w) \phi(x; w) \) can be rewritten as \( f(x) = \int \alpha'(w) \phi(x; w) \) with \( |\alpha'|/p \leq |\alpha|/p \), improving the importance ratio \( C \). This intuition can be formalized via the the Rademacher complexity of \( \phi \), a result which we leave for future work.

The convergence rate for Adaboost [7] is exponentially fast in \( K \), which at first appears to be much faster than \( 1/\sqrt{K} \). However, the base of the exponent is the minimum weighted margin encountered by the algorithm through all iterations, a quantity that is difficult to bound a priori. This makes a direct comparison of the bounds difficult, though we have tried to provide empirical comparisons.

A Exponentially Fast Concentration of Averages towards the Mean in a Hilbert Space

Lemma 4. Let \( X = \{x_1, \cdots, x_K \} \) be iid random variables in a ball \( \mathcal{H} \) of radius \( M \) centered around the origin in a Hilbert space. Denote their average by \( \bar{X} = \frac{1}{K} \sum_{k=1}^{K} x_k \). Then for any \( \delta > 0 \), with probability at least \( 1 - \delta \),

\[
\|\bar{X} - \mathbb{E} \bar{X}\| \leq \frac{M}{\sqrt{K}} \left( 1 + \sqrt{2 \log \frac{1}{\delta}} \right).
\] (16)
Proof. We use McDiarmid’s inequality to show that the scalar function \( f(X) = \|X - E_X X\| \) is concentrated about its mean, which shrinks as \( O(1/\sqrt{K}) \).

The function \( f \) is stable under perturbation of its \( i \)th argument. Let \( \tilde{X} = \{x_1, \cdots, \tilde{x}_i, \cdots, x_K\} \) be a copy of \( X \) with the \( i \)th element replaced by an arbitrary element of \( H \). Applying the triangle inequality twice gives

\[
|f(X) - f(\tilde{X})| = \|X - \tilde{X}\| = \|\tilde{X} - E_X \tilde{X}\| \leq \|X - E_X \tilde{X}\| \leq \frac{\|x_i - \tilde{x}_i\|}{K} \leq \frac{2M}{K}.
\]  

(17)

To bound the expectation of \( f \), use the familiar identity about the variance of the average of iid random variables

\[
E \|X - E_X X\|^2 = \frac{1}{K} (E \|x\|^2 - E x)^2,
\]  

(18)

in conjunction with Jensen’s inequality and the fact that \( \|x\| \leq M \) to get

\[
E f(X) \leq \sqrt{E f^2(X)} = \sqrt{E \|X - E_X X\|^2} \leq \frac{M}{\sqrt{K}}.
\]  

(19)

This bound for the expectation of \( f \) and McDiarmid’s inequality give

\[
Pr \left[ f(X) - \frac{M}{\sqrt{K}} \geq \epsilon \right] \leq Pr \left[ f(X) - Ef(X) \geq \epsilon \right] \leq \exp \left( -\frac{K\epsilon^2}{2M^2} \right)
\]  

(20)

To get the final result, set \( \delta \) to the right hand side, solve for \( \epsilon \), and rearrange. \( \square \)

B Generalization bounds that use Rademacher complexity

One measure of the size of a class \( \mathcal{F} \) of functions is its Rademacher complexity:

\[
\mathcal{R}_m[\mathcal{F}] \equiv \mathbb{E}_{\sigma_1, \cdots, \sigma_m} \left[ \sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^m \sigma_i f(x_i) \right].
\]

The variables \( \sigma_1, \cdots, \sigma_m \) are iid Bernouli random variables that take on the value -1 or +1 with equal probability and are independent of \( x_1, \cdots, x_m \).

The Rademacher complexity of \( \hat{\mathcal{F}}_w \) can be bounded as follows. Define \( \mathcal{S} \equiv \{ \alpha \in \mathbb{R}^K | \|\alpha\|_{\infty} \leq \frac{C}{\sqrt{K}} \} \):

\[
\mathcal{R}_m[\hat{\mathcal{F}}_w] = \mathbb{E}_{\sigma, X} \sup_{\alpha \in \mathcal{S}} \left| \frac{1}{m} \sum_{i=1}^m \sigma_i \left( \sum_{k=1}^K \alpha_k \phi(x_i; \omega_k) \right) \right| = \mathbb{E}_{\sigma, X} \sup_{\alpha \in \mathcal{S}} \sum_{k=1}^K \frac{1}{m} \sum_{i=1}^m \sigma_i \phi(x_i; \omega_k)
\]  

(21)

\[
\leq \mathbb{E}_{\sigma, X} \frac{C}{K} \sum_{k=1}^K \frac{1}{m} \sum_{i=1}^m \sigma_i \phi(x_i; \omega_k) \leq \mathbb{E}_{\sigma, X} \frac{C}{K} \sum_{k=1}^K \left( \mathbb{E}_{\sigma} \left( \frac{1}{m} \sum_{i=1}^m \sigma_i \phi(x_i; \omega_k) \right)^2 \right)
\]  

(22)

\[
\leq \frac{C}{K} \sum_{k=1}^K \sqrt{\frac{1}{m}} \sum_{k=1}^K \phi^2(x_i; \omega_k) \leq \frac{C}{K} \sum_{k=1}^K \sqrt{\frac{1}{m}}
\]  

(23)

\[
\leq \frac{C}{\sqrt{m}}.
\]  

(24)

where the first inequality follows by Hölder, the second by the concavity of square root, the third by the fact that conditioned on \( \omega \), \( \mathbb{E}_\sigma \sigma_i \phi(x_i; \omega) \sigma_j \phi(x_j; \omega) = 0 \) when \( i \neq j \), and the fourth follows by the boundedness of \( \phi \).

The following theorem is a summary of the results from [12]:
**Theorem 2.** Let $\mathcal{F}$ be a class of bounded functions so that $\sup_x |f(x)| \leq C$ for all $f \in \mathcal{F}$, and suppose $c(y, y') = c(yy')$, with $c(yy')$ $L$-Lipschitz. Then with probability at least $1 - \delta$ with respect to training samples $\{x_i, y_i\}_m$ drawn from a probability distribution $P$ on $X \times \{-1, +1\}$, every function in $\mathcal{F}$ satisfies

$$R[f] \leq R_{\text{emp}}[f] + 4LR_m[\mathcal{F}] + 2|c(0)| \sqrt{\frac{1}{m}} + LC \sqrt{\frac{1}{2m} \log \frac{1}{\delta}}. \quad (25)$$

**References**


