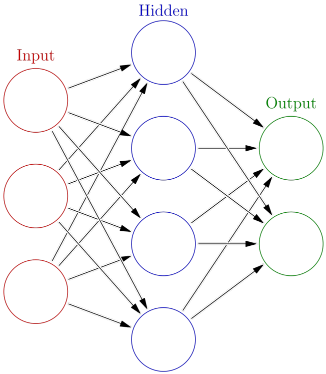
On the Power and Limitations of Random Features for Understanding Neural Networks

Gilad Yehudai Ohad Shamir

WEIZMANN INSTITUTE OF SCIENCE

NeurIPSi
November 2019

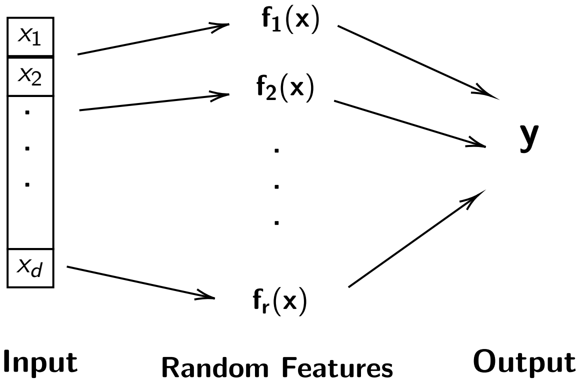
Why neural networks are so successful?



Random Features

The Main Question

Can Random Features help us understand neural networks?



Previous Works

- Learning polynomials with neural networks, Andoni et al. (2014)
- SGD learns the conjugate kernel class of the network, Daniely (2017)
- Gradient descent finds global minima of deep neural networks, Du et al. (2018)
- Gradient descent provably optimizes over-parameterized neural networks, Du et al. (2018)
- Random ReLU features: Universality, approximation, and composition, Sun et al. (2018)
- Learning and generalization in overparameterized neural networks, going beyond two layers, Allen-Zhu et al. (2018)

Previous Works

- **Neural tangent kernel: Convergence and generalization in neural networks**, Jacot et al. (2018)
- Learning overparameterized neural networks via stochastic gradient descent on structured data, Li et al. (2018)
- A generalization theory of gradient descent for learning over-parameterized deep ReLU networks, Cao et al. (2019)
- Can SGD learn recurrent neural networks with provable generalization? Allen-Zhu et al. (2019)

Random Features

Random Features Model

- $\mathcal{F} \subseteq \{f : \mathbb{R}^d \to \mathbb{R}\}$ a family of functions
- $\mathcal{D}$ a distribution over $\mathcal{F}$
- **Random features:** linear predictor over functions from $\mathcal{F}$

$$f(\mathbf{x}) = \sum_{i=1}^{r} u_i f_i(\mathbf{x})$$

Random Features

Random Features Model

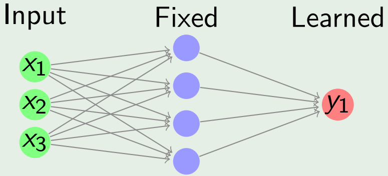
- $\mathcal{F} \subseteq \{f : \mathbb{R}^d \to \mathbb{R}\}$ a family of functions
- $\mathcal{D}$ a distribution over $\mathcal{F}$
- **Random features:** linear predictor over functions from $\mathcal{F}$

$$f(\mathbf{x}) = \sum_{i=1}^{r} u_i f_i(\mathbf{x})$$

Examples

1. Two-layer neural network with fixed first-layer weights:

$$\sum_{i=1}^{r} u_i \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle)$$



2. Any random or deterministic kernel, including neural tangent kernel (NTK)

Our Contribution

- Power of random features:
  - Neural networks can learn as well as polynomial predictors, as long as there are enough neurons (proof use random features)

Our Contribution

- Power of random features:
  - Neural networks can learn as well as polynomial predictors, as long as there are enough neurons (proof use random features)

- Limitations of Random features
  - **The random features model cannot even efficiently approximate a single ReLU neuron**

Learning Polynomials - Main Theorem

Theorem (Neural networks learn polynomials)

Given any data distribution $\mathcal{D}$ on $\mathbb{R}^d$, running SGD on a two-layer neural network with r neurons w.h.p will have better generalization capabilities over data from $\mathcal{D}$ than any polynomial predictor with degree at most k and coefficients at most α, as long as:

$$r > poly\left(\alpha, d^k\right)$$

Learning Polynomials - Main Theorem

Theorem (Neural networks learn polynomials)

Given any data distribution $\mathcal{D}$ on $\mathbb{R}^d$, running SGD on a two-layer neural network with r neurons w.h.p will have better generalization capabilities over data from $\mathcal{D}$ than any polynomial predictor with degree at most k and coefficients at most α, as long as:

$$r > poly\left(\alpha, d^k\right)$$

Remark - Neurons lower bound

We can lower bound r by the number of polynomials, $r > \Omega(d^k)$

Learning Polynomials - Proof Intuition

Reduction to random features

Let $u_i^{(t)}$, $\mathbf{w}_i^{(t)}$ be the weights of $N(\mathbf{x})$ at iteration t.

Take an appropriate learning rate and number of iterations (depend on r):

$$N^{(t)}(\mathbf{x}) = \sum_{i=1}^{r} u_i^{(t)} \sigma\left(\left\langle \mathbf{w}_i^{(t)}, \mathbf{x} \right\rangle\right) \approx \sum_{i=1}^{r} u_i^{(t)} \sigma\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x} \right\rangle\right)$$

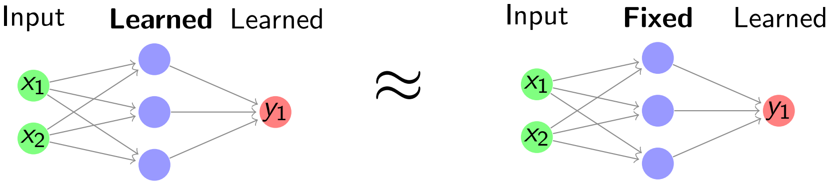
Learning Polynomials - Proof Intuition

Reduction to random features

Let $u_i^{(t)}$, $\mathbf{w}_i^{(t)}$ be the weights of $N(\mathbf{x})$ at iteration t.

Take an appropriate learning rate and number of iterations (depend on r):

$$N^{(t)}(\mathbf{x}) = \sum_{i=1}^{r} u_i^{(t)} \sigma\left(\left\langle \mathbf{w}_i^{(t)}, \mathbf{x} \right\rangle\right) \approx \sum_{i=1}^{r} u_i^{(t)} \sigma\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x} \right\rangle\right)$$



Learning Polynomials - Proof Intuition

Reduction to random features

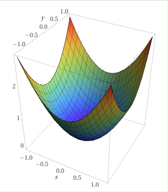
Let $u_i^{(t)}$, $\mathbf{w}_i^{(t)}$ be the weights of $N(\mathbf{x})$ at iteration t.

Take an appropriate learning rate and number of iterations (depend on r):

$$N^{(t)}(\mathbf{x}) = \sum_{i=1}^{r} u_i^{(t)} \sigma\left(\left\langle \mathbf{w}_i^{(t)}, \mathbf{x} \right\rangle\right) \approx \sum_{i=1}^{r} u_i^{(t)} \sigma\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x} \right\rangle\right)$$

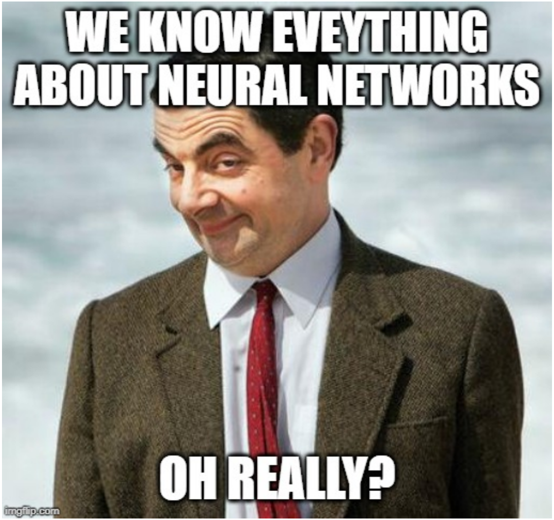
Convex optimization over random features

$$\sum_{i=1}^{r} u_i^{(t)} \sigma\left(\left\langle \mathbf{w}_i^{(0)}, \mathbf{x} \right\rangle\right)$$



Is This All There is About Neural Networks?

Is This All There is About Neural Networks?



Limitations of Random Features

Setting

(1) $\mathbf{x} \sim \mathcal{N}(0, I_d)$
(2) $y = [\langle \mathbf{w}^*, \mathbf{x} \rangle + b^*]_+$ where $[\cdot]_+$ denote ReLU

Limitations of Random Features

Setting

(1) $\mathbf{x} \sim \mathcal{N}(0, I_d)$
(2) $y = [\langle \mathbf{w}^*, \mathbf{x} \rangle + b^*]_+$ where $[\cdot]_+$ denote ReLU

Theorem (General random features)

For every distribution $\mathcal{D}$ of functions from $\mathcal{F}$ there exist $\mathbf{w}^* \in \mathbb{R}^d$, $b^* \in \mathbb{R}$ such that w.h.p over sampling of $f_1, \ldots f_r$ if:

$$\mathbb{E}_{\mathbf{x} \sim \mathcal{N}(0,I)} \left[\left(\sum_{i=1}^{r} u_i f_i(\mathbf{x}) - [\langle \mathbf{w}^*, \mathbf{x} \rangle + b^*]_+ \right)^2 \right] \leq \frac{1}{50}$$

then:

$$r \cdot \max_i |u_i| \geq \Omega(2^d)$$

Limitations of Random Features - Corollary

Theorem (Symmetric random features)

If the random features have a symmetric structure, then the previous theorem applies for **any** $\mathbf{w}^*$

Limitations of Random Features - Corollary

Theorem (Symmetric random features)

If the random features have a symmetric structure, then the previous theorem applies for **any** $\mathbf{w}^*$

Corollary

Using the random features model, it is not possible to explain how two-layer neural networks can learn a single ReLU neuron.

Limitations of Random Features - Corollary

Theorem (Symmetric random features)

If the random features have a symmetric structure, then the previous theorem applies for **any** $\mathbf{w}^*$

Corollary

Using the random features model, it is not possible to explain how two-layer neural networks can learn a single ReLU neuron.

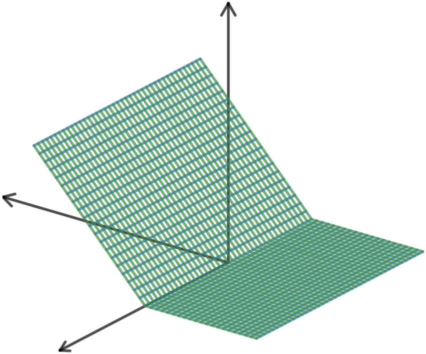
Learning a single neuron

The following optimization problem:

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,I)}\left[\left([\langle \mathbf{w},\mathbf{x}\rangle]_+ - [\langle \mathbf{w}^*,\mathbf{x}\rangle]_+\right)^2\right]$$

where $\mathbf{w}$ is optimized, i.e. learning a single neuron with a single neuron, is tractable with gradient based methods (e.g. Soltanolkotabi (2017))

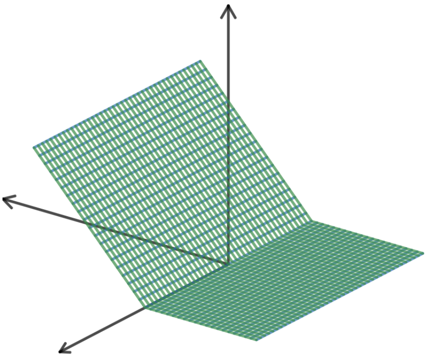
(Short) Proof Intuition



Learning ReLU with ReLU

- Our random features are $f_i(\mathbf{x}) = [\langle \mathbf{w}_i, \mathbf{x} \rangle]_+$ for random $\mathbf{w}_i$
- Our target neuron is $[\langle \mathbf{w}^*, \mathbf{x} \rangle]_+$

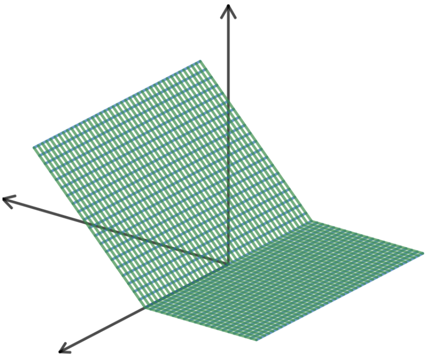
(Short) Proof Intuition



Learning ReLU with ReLU

- Our random features are $f_i(\mathbf{x}) = [\langle \mathbf{w}_i, \mathbf{x} \rangle]_+$ for random $\mathbf{w}_i$
- Our target neuron is $[\langle \mathbf{w}^*, \mathbf{x} \rangle]_+$
- By picking $\mathbf{w}_i$ spherically at random, they are almost orthogonal to $\mathbf{w}^*$ (concentration of measure)

(Short) Proof Intuition



Learning ReLU with ReLU

- Our random features are $f_i(\mathbf{x}) = [\langle \mathbf{w}_i, \mathbf{x} \rangle]_+$ for random $\mathbf{w}_i$
- Our target neuron is $[\langle \mathbf{w}^*, \mathbf{x} \rangle]_+$
- By picking $\mathbf{w}_i$ spherically at random, they are almost orthogonal to $\mathbf{w}^*$ (concentration of measure)
- Will need many $\mathbf{w}_i$ in order to have high correlation with the direction $\mathbf{w}^*$

Take Home Message

- Random features are very limited in explaining neural networks
  - In particular, they cannot even explain learnability of a single ReLU neuron (which is quite learnable)

Take Home Message

- Random features are very limited in explaining neural networks
  - In particular, they cannot even explain learnability of a single ReLU neuron (which is quite learnable)
- Random features do not capture representation learning, which seem to be a significant part of neural networks

Take Home Message

- Random features are very limited in explaining neural networks
  - In particular, they cannot even explain learnability of a single ReLU neuron (which is quite learnable)
- Random features do not capture representation learning, which seem to be a significant part of neural networks
- There are still many open question (despite all the positive results) as to why neural networks successfully learn with gradient based methods

Take Home Message

- Random features are very limited in explaining neural networks
  - In particular, they cannot even explain learnability of a single ReLU neuron (which is quite learnable)
- Random features do not capture representation learning, which seem to be a significant part of neural networks
- There are still many open question (despite all the positive results) as to why neural networks successfully learn with gradient based methods

Thank You!

gilad.yehudai@weizmann.ac.il