Bidirectional Self-Normalizing Neural Networks

Thalaiyasingam Ajanthan

Australian National University

August 2020
Collaborators

Yao Lu  Stephen Gould  Thalaiyasingam Ajanthan
Signal Propagation in Neural Networks

Forward:
\[ h^l = W^l x^l, \quad x^{l+1} = \phi(h^l), \]

Backward:
\[ d^l = D^l \left( \prod_{k=l+1}^{L-1} W^k D^k \right) \frac{\partial E}{\partial x^l}, \quad D^l_{ij} = \phi'(h^l_i). \]
Signal Propagation in Neural Networks

Forward:
\[ h^l = W^l x^l, \quad x^{l+1} = \phi(h^l), \]

Backward:
\[ d^l = D^l \left( \prod_{k=l+1}^{L-1} W^k D^k \right) \frac{\partial E}{\partial x^L}, \quad D^l_{ii} = \phi'(h_i^l). \]
Signal Propagation in Neural Networks

Forward:
\[ h^l = W^l x^l, \quad x^{l+1} = \phi(h^l), \]

Backward:
\[ d^l = D^l \left( \prod_{k=l+1}^{L-1} W^k D^k \right) \frac{\partial E}{\partial x^L}, \quad D_{ii}^l = \phi'(h_i^l). \]
Signal Propagation in Neural Networks

Objective

▶ Preserve signal norm in both forward and backward directions.
▶ Maintain nonlinear functionality.

Why?

▶ Very deep CNNs and RNNs. [Pennington-2018, Chen-2018]
▶ Improved robustness? [Lin-2019]
▶ Improved generalization?

What if signals are not preserved?

▶ Signals can saturate ⇒ vanishing/exploding gradients.
Signal Propagation in Neural Networks

Objective

▶ Preserve signal norm in both forward and backward directions.
▶ Maintain nonlinear functionality.

Why?

▶ Very deep CNNs and RNNs. [Pennington-2018, Chen-2018]
▶ Improved robustness? [Lin-2019]
▶ Improved generalization?

What if signals are not preserved?

▶ Signals can saturate ⇒ vanishing/exploding gradients.
Signal Propagation in Neural Networks

Objective

▶ Preserve signal norm in both forward and backward directions.
▶ Maintain nonlinear functionality.

Why?

▶ Very deep CNNs and RNNs. [Pennington-2018, Chen-2018]
▶ Improved robustness? [Lin-2019]
▶ Improved generalization?

What if signals are not preserved?

▶ Signals can saturate ⇒ **vanishing/exploding gradients.**
Existing Approaches to Preserve Signal Propagation

Initialization

► Variance scaling initialization. [Glorot-2010, Mishkin-2016]
► Dynamical isometry and mean-field theory. [Pennington-2017]

Normalization

► Batch normalization and its variants. [Ioffe-2015]
► \{Layer, group, spectral, weight, ...\} normalization. [...]

Architecture

► Residual connections. [He-2016]

Drawbacks

► No rigorous proofs.
► Do not solve gradient vanishing/explosion. [Philipp-2018]
Existing Approaches to Preserve Signal Propagation

Initialization

- Variance scaling initialization. [Glorot-2010, Mishkin-2016]
- Dynamical isometry and mean-field theory. [Pennington-2017]

Normalization

- Batch normalization and its variants. [Ioffe-2015]
- \{Layer, group, spectral, weight, ...\} normalization. [...] 

Architecture

- Residual connections. [He-2016]

Drawbacks

- No rigorous proofs.
- Do not solve gradient vanishing/explosion. [Philipp-2018]
Existing Approaches to Preserve Signal Propagation

Initialization

- Variance scaling initialization. [Glorot-2010, Mishkin-2016]
- Dynamical isometry and mean-field theory. [Pennington-2017]

Normalization

- Batch normalization and its variants. [Ioffe-2015]
- {Layer, group, spectral, weight, ...} normalization. [...]

Architecture

- Residual connections. [He-2016]

Drawbacks

- No rigorous proofs.
- Do not solve gradient vanishing/explosion. [Philipp-2018]
Existing Approaches to Preserve Signal Propagation

Initialization

▶ Variance scaling initialization. [Glorot-2010, Mishkin-2016]
▶ Dynamical isometry and mean-field theory. [Pennington-2017]

Normalization

▶ Batch normalization and its variants. [Ioffe-2015]
▶ {Layer, group, spectral, weight, ...} normalization. [...] 

Architecture

▶ Residual connections. [He-2016]

Drawbacks

▶ No rigorous proofs.
▶ Do not solve gradient vanishing/explosion. [Philipp-2018]
## Summary of Existing Approaches

<table>
<thead>
<tr>
<th></th>
<th>Dynamical Isometry</th>
<th>Self normalization</th>
<th>Batch normalization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Weights (W)</strong></td>
<td>Orthogonal</td>
<td>VS-init.</td>
<td>Unconstrained</td>
</tr>
<tr>
<td><strong>Activations (φ)</strong></td>
<td>Most</td>
<td>SELU</td>
<td>All</td>
</tr>
<tr>
<td><strong>Preactivations (h)</strong></td>
<td>Linear region</td>
<td>Unconstrained</td>
<td>$E \approx 0, V \approx 1$</td>
</tr>
<tr>
<td><strong>Forward signal (x)</strong></td>
<td>Constrained</td>
<td>$E \approx 0, V \approx 1$</td>
<td>Constrained</td>
</tr>
<tr>
<td><strong>Backward signal (d)</strong></td>
<td>Constrained</td>
<td>Unconstrained</td>
<td>Unconstrained</td>
</tr>
<tr>
<td><strong>Functionality</strong></td>
<td>Pseudo-linear</td>
<td>Nonlinear</td>
<td>Nonlinear</td>
</tr>
</tbody>
</table>

[Philipp-2018]
## Summary of Existing Approaches

<table>
<thead>
<tr>
<th></th>
<th>Dynamical Isometry</th>
<th>Self normalization</th>
<th>Batch normalization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Weights</strong> ((\mathbf{W}))</td>
<td>Orthogonal</td>
<td>VS-init.</td>
<td>Unconstrained</td>
</tr>
<tr>
<td><strong>Activations</strong> ((\phi))</td>
<td>Most</td>
<td>SELU</td>
<td></td>
</tr>
<tr>
<td><strong>Preactivations</strong> ((\mathbf{h}))</td>
<td>Linear region</td>
<td>Unconstrained</td>
<td>(E \approx 0, \nabla \approx 1)</td>
</tr>
<tr>
<td><strong>Forward signal</strong> ((\mathbf{x}))</td>
<td>Constrained</td>
<td>(E \approx 0, \nabla \approx 1)</td>
<td>Constrained</td>
</tr>
<tr>
<td><strong>Backward signal</strong> ((\mathbf{d}))</td>
<td>Constrained</td>
<td>Unconstrained</td>
<td>Unconstrained</td>
</tr>
<tr>
<td><strong>Functionality</strong></td>
<td>Pseudo-linear</td>
<td>Nonlinear</td>
<td>Nonlinear</td>
</tr>
</tbody>
</table>

[Philipp-2018]

### Key missing points

- Do not preserve signal in both forward and backward directions.
- Do not maintain **nonlinear** functionality.
**Key idea:** New class of activation functions: **Gaussian-Poincaré Normalized** activations.

<table>
<thead>
<tr>
<th></th>
<th>Dynamical Isometry</th>
<th>Self normalization</th>
<th>BSNN (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weights ($\mathbf{W}$)</td>
<td>Orthogonal</td>
<td>VS-init.</td>
<td>Orthogonal</td>
</tr>
<tr>
<td>Activations ($\phi$)</td>
<td>Most</td>
<td>SELU</td>
<td>GPN-versions</td>
</tr>
<tr>
<td>Preactivations ($\mathbf{h}$)</td>
<td>Linear region</td>
<td>Unconstrained</td>
<td>Unconstrained</td>
</tr>
<tr>
<td>Forward signal ($\mathbf{x}$)</td>
<td>Constrained</td>
<td>$E \approx 0$, $V \approx 1$</td>
<td>$V + E^2 \approx 1$</td>
</tr>
<tr>
<td>Backward signal ($\mathbf{d}$)</td>
<td>Constrained</td>
<td>Unconstrained</td>
<td>$V + E^2 \approx 1$</td>
</tr>
<tr>
<td>Functionality</td>
<td>Pseudo-linear</td>
<td>Nonlinear</td>
<td>Nonlinear</td>
</tr>
</tbody>
</table>
**Bidirectional Self-Normalizing Neural Networks**

**Key idea:** New class of activation functions: **Gaussian-Poincaré Normalized** activations.

<table>
<thead>
<tr>
<th></th>
<th>Dynamical Isometry</th>
<th>Self normalization</th>
<th>BSNN (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weights ( (W) )</td>
<td>Orthogonal</td>
<td>VS-init.</td>
<td>Orthogonal</td>
</tr>
<tr>
<td>Activations ( (\phi) )</td>
<td>Most Unconstrained</td>
<td>SELU</td>
<td>GPN-versions</td>
</tr>
<tr>
<td>Preactivations ( (h) )</td>
<td>Linear region Unconstrained</td>
<td>Unconstrained</td>
<td>Unconstrained</td>
</tr>
<tr>
<td>Forward signal ( (x) )</td>
<td>Constrained Unconstrained</td>
<td>( \mathbb{E} \approx 0, \mathbb{V} \approx 1 )</td>
<td>( \mathbb{V} + \mathbb{E}^2 \approx 1 )</td>
</tr>
<tr>
<td>Backward signal ( (d) )</td>
<td>Constrained Unconstrained</td>
<td>Nonlinear</td>
<td>Nonlinear</td>
</tr>
<tr>
<td>Functionality</td>
<td>Pseudo-linear</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bidirectional Self-Normalization

**Setting:** Deep fully-connected networks with hidden layers of same width and no bias.

$W^l \in \mathbb{R}^{n \times n}$, $l = \{1, \ldots, L - 1\}$ and $\phi : \mathbb{R} \rightarrow \mathbb{R}$. 

No vanishing/exploding gradients.
Bidirectional Self-Normalization

**Setting:** Deep fully-connected networks with hidden layers of same width and no bias.

\[ W^l \in \mathbb{R}^{n \times n}, \ l = \{1, \ldots, L - 1\} \]
and \( \phi : \mathbb{R} \rightarrow \mathbb{R} \).

**Forward:** \( h^l = W^l x^l \), \( x^{l+1} = \phi(h^l) \),

**Backward:** \( d^l = D^l \left( \prod_{k=l+1}^{L-1} W^k D^k \right) \frac{\partial E}{\partial x^L}, \ D^l_{ii} = \phi'(h^l_i) \),

**Gradient:** \( \frac{\partial E}{\partial W^l} = d^l(x^l)^T \).
Bidirectional Self-Normalization

Setting: Deep fully-connected networks with hidden layers of same width and no bias.

\[ \mathbf{W}^l \in \mathbb{R}^{n \times n}, \; l = \{1, \ldots, L - 1\} \]
and \( \phi : \mathbb{R} \to \mathbb{R} \).

Require:
\[ \| \mathbf{x}^1 \|_2 = \| \mathbf{x}^2 \|_2 = \ldots = \| \mathbf{x}^L \|_2, \quad \{ \mathbf{W}^l \}, \phi \ \text{constrained}, \]
\[ \| \mathbf{d}^1 \|_2 = \| \mathbf{d}^2 \|_2 = \ldots = \| \mathbf{d}^L \|_2, \quad \{ \mathbf{W}^l \}, \phi' \ \text{constrained}. \]
Bidirectional Self-Normalization

**Setting:** Deep fully-connected networks with hidden layers of same width and no bias.

$W^l \in \mathbb{R}^{n \times n}$, $l = \{1, \ldots, L - 1\}$ and $\phi : \mathbb{R} \to \mathbb{R}$.

Require: $\|x^1\|_2 = \|x^2\|_2 = \ldots = \|x^L\|_2$, $\{W^l\}$, $\phi$ constrained,

$\|d^1\|_2 = \|d^2\|_2 = \ldots = \|d^L\|_2$, $\{W^l\}, \phi'$ constrained.

No vanishing/exploding gradients.
Orthogonal Weight Matrices

\[(W^l)^T W^l = W^l (W^l)^T = I_n .\]

Properties

- Linear networks: guarantees bidirectional self-normalization. [Saxe-2014]
- Nonlinear networks: improves trainability with appropriate scaling. [Pennington-2017]
- Widespread usage in GANs, training sparse networks, quantized networks, etc. [Brock-2017, Lee-2020, Lin-2019]
Orthogonal Weight Matrices

\[(W^l)^T W^l = W^l (W^l)^T = I_n.\]

Properties

- Linear networks: guarantees bidirectional self-normalization. [Saxe-2014]
- Nonlinear networks: improves trainability with appropriate scaling. [Pennington-2017]
- Widespread usage in GANs, training sparse networks, quantized networks, etc. [Brock-2017, Lee-2020, Lin-2019]
GPN Activations

\[ \mathbb{E}_{h \sim \mathcal{N}(0,1)} [\phi(h)^2] = \mathbb{E}_{h \sim \mathcal{N}(0,1)} [\phi'(h)^2] = 1. \]

Key facts

- If \( W^l \) is orthogonal, \( h^l \) can be shown to be approximately Gaussian.
- Function \( \phi \) is GPN and \( \mathbb{E}_{h \sim \mathcal{N}(0,1)}[\phi(h)] = 0 \), if and only if \( \phi \) is linear.
- A differentiable function \( \phi \) can be transformed into its GPN version by \( \mu \phi(h) + X \).
GPN Activations

\[ \mathbb{E}_{h \sim \mathcal{N}(0,1)} [\phi(h)^2] = \mathbb{E}_{h \sim \mathcal{N}(0,1)} [\phi'(h)^2] = 1. \]

Key facts

- If \( W^l \) is orthogonal, \( h^l \) can be shown to be approximately Gaussian.
- Function \( \phi \) is GPN and \( \mathbb{E}_{h \sim \mathcal{N}(0,1)} [\phi(h)] = 0 \), if and only if \( \phi \) is linear.
- A differentiable function \( \phi \) can be transformed into its GPN version by \( a\phi(h) + b \).
GPN Activations

\[ \mathbb{E}_{h \sim \mathcal{N}(0,1)} \left[ \phi(h)^2 \right] = \mathbb{E}_{h \sim \mathcal{N}(0,1)} \left[ \phi'(h)^2 \right] = 1. \]

Key facts

- If \( W^l \) is orthogonal, \( h^l \) can be shown to be approximately Gaussian.
- Function \( \phi \) is GPN and \( \mathbb{E}_{h \sim \mathcal{N}(0,1)}[\phi(h)] = 0 \), if and only if \( \phi \) is linear.
- A differentiable function \( \phi \) can be transformed into its GPN version by \( a\phi(h) + b \).
GPN Activations

\[ \mathbb{E}_{h \sim \mathcal{N}(0,1)} [\phi(h)^2] = \mathbb{E}_{h \sim \mathcal{N}(0,1)} [\phi'(h)^2] = 1. \]

Key facts

► If \( W^l \) is orthogonal, \( h^l \) can be shown to be approximately Gaussian.

► Function \( \phi \) is GPN and \( \mathbb{E}_{h \sim \mathcal{N}(0,1)}[\phi(h)] = 0 \), if and only if \( \phi \) is linear.

► A differentiable function \( \phi \) can be transformed into its GPN version by \( a\phi(h) + b \).
GPN Activations

Common activation functions and their GPN versions.
Thin-Shell Concentration

▶ Probabilistic version of the vector norm constraint.

A random vector $\mathbf{x} \in \mathbb{R}^n$ is TSC if for any $\epsilon > 0$

$$
\mathbb{P}\left\{ \left| \frac{1}{n} \| \mathbf{x} \|^2_2 - 1 \right| \geq \epsilon \right\} \to 0, \quad \text{as } n \to \infty. \quad \text{[Bobkov-2003]}
$$

Examples: Multivariate Gaussian

Any distribution on $n$-unit-sphere scaled by $\sqrt{n}$.

Intuitive visualization of Gaussian. \text{[Vershynin-2018]}
Thin-Shell Concentration

- Probabilistic version of the vector norm constraint. A random vector $\mathbf{x} \in \mathbb{R}^n$ is TSC if for any $\epsilon > 0$

$$\mathbb{P}\left\{ \left| \frac{1}{n} \|\mathbf{x}\|_2^2 - 1 \right| \geq \epsilon \right\} \to 0, \quad \text{as } n \to \infty. \ [\text{Bobkov-2003}]$$

Examples: Multivariate Gaussian

Any distribution on $n$-unit-sphere scaled by $\sqrt{n}$.

Intuitive visualization of Gaussian. [Vershynin-2018]
Thin-Shell Concentration

- Probabilistic version of the vector norm constraint.

A random vector \( \mathbf{x} \in \mathbb{R}^n \) is TSC if for any \( \epsilon > 0 \)

\[
\mathbb{P} \left\{ \left| \frac{1}{n} \| \mathbf{x} \|_2^2 - 1 \right| \geq \epsilon \right\} \rightarrow 0, \quad \text{as} \ n \rightarrow \infty. \quad \text{[Bobkov-2003]}
\]

Examples: Multivariate Gaussian

Any distribution on \( n \)-unit-sphere scaled by \( \sqrt{n} \).

Intuitive visualization of Gaussian. [Vershynin-2018]
Thin-Shell Concentration

- Probabilistic version of the vector norm constraint.

A random vector \( x \in \mathbb{R}^n \) is TSC if for any \( \epsilon > 0 \)

\[
\mathbb{P} \left\{ \left| \frac{1}{n} \|x\|_2^2 - 1 \right| \geq \epsilon \right\} \to 0, \quad \text{as } n \to \infty. \quad \text{[Bobkov-2003]}
\]

Examples: Multivariate Gaussian

Any distribution on \( n \)-unit-sphere scaled by \( \sqrt{n} \).

Intuitive visualization of Gaussian. [Vershynin-2018]
Norm Preservation Theorems

Assumptions

1. Random vector \( \mathbf{x} \in \mathbb{R}^n \) is TSC. 
   - Normalize the input vector.
2. Random orthogonal weight matrix \( \mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)^T \) is uniformly distributed.
3. Activation function \( \phi : \mathbb{R} \to \mathbb{R} \) is GPN.
4. Activation function \( \phi \) and its derivative are Lipschitz continuous.
   - Most common activation functions satisfy 3 and 4.
Norm Preservation Theorems

Assumptions

1. Random vector $\mathbf{x} \in \mathbb{R}^n$ is TSC.
   - Normalize the input vector.

2. Random orthogonal weight matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)^T$ is uniformly distributed.

3. Activation function $\phi : \mathbb{R} \to \mathbb{R}$ is GPN.

4. Activation function $\phi$ and its derivative are Lipschitz continuous.
   - Most common activation functions satisfy 3 and 4.
Norm Preservation Theorems

Assumptions

1. Random vector $\mathbf{x} \in \mathbb{R}^n$ is TSC.
   - Normalize the input vector.

2. Random orthogonal weight matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)^T$ is uniformly distributed.

3. Activation function $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is GPN.

4. Activation function $\phi$ and its derivative are Lipschitz continuous.
   - Most common activation functions satisfy 3 and 4.
Norm Preservation Theorems

Assumptions

1. Random vector $\mathbf{x} \in \mathbb{R}^n$ is TSC.
   - Normalize the input vector.

2. Random orthogonal weight matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)^T$ is uniformly distributed.

3. Activation function $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is GPN.

4. Activation function $\phi$ and its derivative are Lipschitz continuous.
   - Most common activation functions satisfy 3 and 4.
Norm Preservation Theorems

Assumptions

1. Random vector $\mathbf{x} \in \mathbb{R}^n$ is TSC.
   ▶ Normalize the input vector.

2. Random orthogonal weight matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)^T$ is uniformly distributed.

3. Activation function $\phi : \mathbb{R} \to \mathbb{R}$ is GPN.

4. Activation function $\phi$ and its derivative are Lipschitz continuous.
   ▶ Most common activation functions satisfy 3 and 4.
Norm Preservation Theorems

Assumptions

1. Random vector \( \mathbf{x} \in \mathbb{R}^n \) is TSC.
   ▶ Normalize the input vector.

2. Random orthogonal weight matrix \( \mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)^T \)
   is uniformly distributed.

3. Activation function \( \phi : \mathbb{R} \to \mathbb{R} \) is GPN.

4. Activation function \( \phi \) and its derivative are Lipschitz continuous.
   ▶ Most common activation functions satisfy 3 and 4.
Norm Preservation Theorems

Theorem 1: Forward norm preservation
Random vector
\[
\left( \phi \left( \mathbf{w}_1^T \mathbf{x} \right), \phi \left( \mathbf{w}_2^T \mathbf{x} \right), \ldots, \phi \left( \mathbf{w}_n^T \mathbf{x} \right) \right)^T,
\]
is TSC.

▶ Multiplication by \( \mathbf{W} \) followed by \( \phi \) preserves the norm with high probability.

Theorem 2: Backward norm preservation
Let \( \mathbf{y} \in \mathbb{R}^n \) with \( \| \mathbf{y} \|_\infty \) and \( D_{ii} = \phi'(\mathbf{w}_i^T \mathbf{x}) \). Then for any \( \epsilon > 0 \)
\[
\mathbb{P} \left\{ \left| \frac{1}{n} \| \mathbf{Dy} \|_2^2 - \| \mathbf{y} \|_2^2 \right| \geq \epsilon \right\} \to 0,
\]
as \( n \to \infty \).

▶ Multiplication by \( \mathbf{D} \) preserves the norm with high probability.
Norm Preservation Theorems

**Theorem 1: Forward norm preservation**

Random vector

\[
(\phi(w_1^T x), \phi(w_2^T x), \ldots, \phi(w_n^T x))^T,
\]

is TSC.

- Multiplication by \( W \) followed by \( \phi \) preserves the norm with high probability.

**Theorem 2: Backward norm preservation**

Let \( y \in \mathbb{R}^n \) with \( \| y \|_\infty \) and \( D_{ii} = \phi'(w_i^T x) \). Then for any \( \epsilon > 0 \)

\[
P \left\{ \left| \frac{1}{n} \| Dy \|_2^2 - \| y \|_2^2 \right| \geq \epsilon \right\} \to 0,
\]
as \( n \to \infty \).

- Multiplication by \( D \) preserves the norm with high probability.
Norm Preservation Theorems

Theorem 1: Forward norm preservation
Random vector
\[
(\phi (w_1^T x), \phi (w_2^T x), \ldots, \phi (w_n^T x))^T,
\]
is TSC.

- Multiplication by $W$ followed by $\phi$ preserves the norm with high probability.

Theorem 2: Backward norm preservation
Let $y \in \mathbb{R}^n$ with $\|y\|_\infty$ and $D_{ii} = \phi'(w_i^T x)$. Then for any $\epsilon > 0$
\[
P \left\{ \left| \frac{1}{n} \|Dy\|_2^2 - \|y\|_2^2 \right| \geq \epsilon \right\} \to 0,
\]
as $n \to \infty$.

- Multiplication by $D$ preserves the norm with high probability.
Norm Preservation Theorems

Theorem 1: Forward norm preservation
Random vector
\[(\phi (w_1^T x), \phi (w_2^T x), \ldots, \phi (w_n^T x))^T,\]
is TSC.

▶ Multiplication by \(W\) followed by \(\phi\) preserves the norm with high probability.

Theorem 2: Backward norm preservation
Let \(y \in \mathbb{R}^n\) with \(\|y\|_{\infty}\) and \(D_{ii} = \phi'(w_i^T x)\). Then for any \(\epsilon > 0\)
\[
\mathbb{P} \left\{ \left| \frac{1}{n} \|Dy\|_2^2 - \|y\|_2^2 \right| \geq \epsilon \right\} \rightarrow 0 ,
\]
as \(n \rightarrow \infty\).

▶ Multiplication by \(D\) preserves the norm with high probability.
Sketch of the Proofs

Key theory: Concentration of measure [Vershynin-2018]

- Most mass of some high-dimensional probability distributions is concentrated around a certain range.
- Lipschitz functions do not affect this property.
Sketch of the Proof: Theorem 1

\[ \frac{1}{n} \sum_i \phi(w_i^T x)^2 \]

Rows \( \{w_i\} \) of a random orthogonal matrix are approximately independent for large \( n \).
Sketch of the Proof: Theorem 1

Rows $\{w_i\}$ of a random orthogonal matrix are approximately independent for large $n$.

$\{\theta_i^T x\}$ is approximately Gaussian when $\{\theta_i\}$ are independent and $x$ is TSC.
Sketch of the Proof: Theorem 1

\[ \frac{1}{n} \sum_i \phi(\mathbf{w}_i^T \mathbf{x})^2 \]

\[ \frac{1}{n} \sum_i \phi(\mathbf{\theta}_i^T \mathbf{x})^2 \]

\[ \frac{1}{n} \sum_i \phi(z_i)^2 \]

Rows \(\{\mathbf{w}_i\}\) of a random orthogonal matrix are approximately independent for large \(n\).

\(\{\mathbf{\theta}_i^T \mathbf{x}\}\) is approximately Gaussian when \(\{\mathbf{\theta}_i\}\) are independent and \(\mathbf{x}\) is TSC.

Proof follows from Lipschitz continuous and GPN function \(\phi\).
Sketch of the Proof: Theorem 1

\[
\frac{1}{n} \sum_i \phi(w_i^T x)^2
\]

\[
\frac{1}{n} \sum_i \phi(\theta_i^T x)^2
\]

\[
\frac{1}{n} \sum_i \phi(z_i)^2
\]

1

Rows \(\{w_i\}\) of a random orthogonal matrix are approximately independent for large \(n\).

\(\{\theta_i^T x\}\) is approximately Gaussian when \(\{\theta_i\}\) are independent and \(x\) is TSC.

Proof follows from Lipschitz continuous and GPN function \(\phi\).

Each of these steps are rigorously proved.
Synthetic Experiments

Tanh, $\|x^l\|_2^2/n$

Tanh-GPN, $\|x^l\|_2^2/n$

Forward-backward signal propagation, $L = 200$, $n = 500$. 

Tanh-GPN, $\left\| \frac{\partial E}{\partial \mathbf{W}^l} \right\|_F$
Synthetic Experiments

Tanh, $\|x^l\|_2^2/n$

Tanh-GPN, $\|x^l\|_2^2/n$

Tanh, $\|\frac{\partial E}{\partial W_l}\|_F$

Tanh-GPN, $\|\frac{\partial E}{\partial W_l}\|_F$

Forward-backward signal propagation, $L = 200, n = 500$.

Tanh shows pseudo-linearity while Tanh-GPN is nonlinear.
Synthetic Experiments

Histogram of $\phi'(h_l^i)$, i.e., singular value distribution, $L = 200$, $n = 500$.

Tanh shows **pseudo-linearity** while Tanh-GPN is nonlinear.
Synthetic Experiments

SELU, $\|x^l\|_2^2/n$

SELU-GPN, $\|x^l\|_2^2/n$

SELU, $\|\frac{\partial E}{\partial W^l}\|_F$

SELU-GPN, $\|\frac{\partial E}{\partial W^l}\|_F$

Forward-backward signal propagation, $L = 200, n = 500$. 
Synthetic Experiments

SELU, $\|x^l\|_2^2/n$

SELU-GPN, $\|x^l\|_2^2/n$

SELU, $\|\frac{\partial E}{\partial W^l}\|_F$

SELU-GPN, $\|\frac{\partial E}{\partial W^l}\|_F$

Forward-backward signal propagation, $L = 200, n = 500.$

SELU suffers gradient explosion while SELU-GPN is stable.
Synthetic Experiments

Gradient norm ratio, \( i.e., \max_l \| \frac{\partial E}{\partial W_l} \|_F / \min_l \| \frac{\partial E}{\partial W_l} \|_F \), \( L = 200 \).
Synthetic Experiments

Gradient norm ratio, \( i.e., \max_l \| \frac{\partial E}{\partial \mathbf{W}_l} \|_F / \min_l \| \frac{\partial E}{\partial \mathbf{W}_l} \|_F, L = 200. \)

Larger width leads to more stable gradients.
### Experiments on Real-World Data

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-GPN</td>
<td>GPN (Ours)</td>
</tr>
<tr>
<td>Tanh</td>
<td>99.05 (87.39)</td>
<td><strong>99.81</strong> (84.93)</td>
</tr>
<tr>
<td>ReLU</td>
<td>11.24 (11.24)</td>
<td><strong>33.28</strong> (11.42)</td>
</tr>
<tr>
<td>LReLU</td>
<td>11.24 (11.24)</td>
<td><strong>43.17</strong> (11.19)</td>
</tr>
<tr>
<td>ELU</td>
<td>99.06 (98.24)</td>
<td><strong>100.0</strong> (97.86)</td>
</tr>
<tr>
<td>SELU</td>
<td>99.86 (97.82)</td>
<td><strong>99.92</strong> (97.91)</td>
</tr>
<tr>
<td>GELU</td>
<td>11.24 (12.70)</td>
<td><strong>97.67</strong> (11.22)</td>
</tr>
</tbody>
</table>

Training accuracy with various activation functions, $L = 200$, $n = 500$. 
Experiments on Real-World Data

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th></th>
<th>CIFAR-10</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-GPN</td>
<td>GPN (Ours)</td>
<td>Non-GPN</td>
<td>GPN (Ours)</td>
</tr>
<tr>
<td>Tanh</td>
<td>99.05 (87.39)</td>
<td><strong>99.81</strong> (84.93)</td>
<td>80.84 (27.90)</td>
<td><strong>96.39</strong> (25.13)</td>
</tr>
<tr>
<td>ReLU</td>
<td>11.24 (11.24)</td>
<td><strong>33.28</strong> (11.42)</td>
<td>10.00 (10.00)</td>
<td><strong>46.60</strong> (10.09)</td>
</tr>
<tr>
<td>LReLU</td>
<td>11.24 (11.24)</td>
<td><strong>43.17</strong> (11.19)</td>
<td>10.00 (10.21)</td>
<td><strong>51.85</strong> (09.89)</td>
</tr>
<tr>
<td>ELU</td>
<td>99.06 (98.24)</td>
<td><strong>100.0</strong> (97.86)</td>
<td>80.73 (42.39)</td>
<td><strong>99.37</strong> (43.35)</td>
</tr>
<tr>
<td>SELU</td>
<td>99.86 (97.82)</td>
<td><strong>99.92</strong> (97.91)</td>
<td>29.23 (46.47)</td>
<td><strong>98.24</strong> (47.74)</td>
</tr>
<tr>
<td>GELU</td>
<td>11.24 (12.70)</td>
<td><strong>97.67</strong> (11.22)</td>
<td>10.00 (10.43)</td>
<td><strong>90.51</strong> (10.00)</td>
</tr>
</tbody>
</table>

Training accuracy with various activation functions, $L = 200, n = 500$.

GPN yields significant improvements in training accuracy.
Experiments on Real-World Data

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th></th>
<th>CIFAR-10</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-GPN</td>
<td>GPN (Ours)</td>
<td>Non-GPN</td>
<td>GPN (Ours)</td>
</tr>
<tr>
<td>Tanh</td>
<td>96.57 (89.32)</td>
<td>95.54 (87.11)</td>
<td><strong>42.71</strong> (29.32)</td>
<td>40.95 (26.58)</td>
</tr>
<tr>
<td>ReLU</td>
<td>11.35 (11.42)</td>
<td><strong>28.13</strong> (11.34)</td>
<td>10.00 (10.00)</td>
<td><strong>34.96</strong> (09.96)</td>
</tr>
<tr>
<td>LReLU</td>
<td>11.35 (11.63)</td>
<td><strong>49.28</strong> (11.66)</td>
<td>10.00 (10.06)</td>
<td><strong>39.38</strong> (10.00)</td>
</tr>
<tr>
<td>ELU</td>
<td>95.41 (97.48)</td>
<td>96.56 (96.69)</td>
<td><strong>45.76</strong> (44.16)</td>
<td>43.12 (44.36)</td>
</tr>
<tr>
<td>SELU</td>
<td>97.33 (97.38)</td>
<td>96.97 (97.39)</td>
<td>29.55 (45.88)</td>
<td><strong>45.90</strong> (45.52)</td>
</tr>
<tr>
<td>GELU</td>
<td>11.35 (10.28)</td>
<td><strong>95.82</strong> (09.74)</td>
<td>10.00 (10.00)</td>
<td><strong>36.94</strong> (10.00)</td>
</tr>
</tbody>
</table>

**Test accuracy** with various activation functions, $L = 200, n = 500$. 
## Experiments on Real-World Data

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-GPN</td>
<td>GPN (Ours)</td>
</tr>
<tr>
<td>Tanh</td>
<td>96.57 (89.32)</td>
<td>95.54 (87.11)</td>
</tr>
<tr>
<td>ReLU</td>
<td>11.35 (11.42)</td>
<td>28.13 (11.34)</td>
</tr>
<tr>
<td>LReLU</td>
<td>11.35 (11.63)</td>
<td>49.28 (11.66)</td>
</tr>
<tr>
<td>ELU</td>
<td>95.41 (97.48)</td>
<td>96.56 (96.69)</td>
</tr>
<tr>
<td>SELU</td>
<td>97.33 (97.38)</td>
<td>96.97 (97.39)</td>
</tr>
<tr>
<td>GELU</td>
<td>11.35 (10.28)</td>
<td>95.82 (09.74)</td>
</tr>
</tbody>
</table>

Test accuracy with various activation functions, $L = 200, n = 500$.

GPN yields improvements in many cases.
Test accuracy curves on CIFAR-10, $L = 200$, $n = 500$. 

GPN accelerates training in many cases.
Experiments on Real-World Data

Test accuracy curves on CIFAR-10, $L = 200, n = 500$.

GPN accelerates training in many cases.
Summary

- We introduced BSNN which constrains signal norm in both directions in nonlinear networks via orthogonal weights and GPN activation functions.
- Many common activations functions can be transformed into their respective GPN versions.
- Rigorously proved that gradient vanishing/exploding problem disappears with high probability if the width is sufficiently large.
Current Limitations

- Theoretical analysis is limited to same width, fully-connected networks.
- Generalization capabilities are unclear.
- Universality of BSNN is an open question.
Questions?
Thank you!