# Bidirectional Self-Normalizing Neural Networks

Thalaiyasingam Ajanthan

Australian National University

August 2020

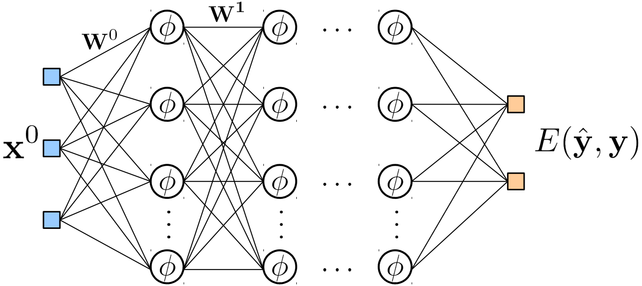
# Collaborators

Yao Lu

Stephen Gould

Thalaiyasingam Ajanthan

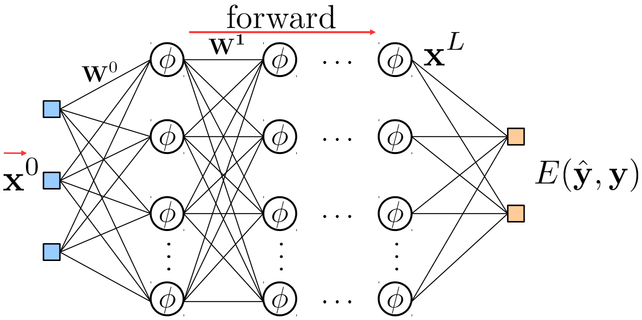
# Signal Propagation in Neural Networks




$$\text{Forward:} \quad \mathbf{h}^l = \mathbf{W}^l \mathbf{x}^l \, , \quad \mathbf{x}^{l+1} = \phi(\mathbf{h}^l) \, ,$$

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

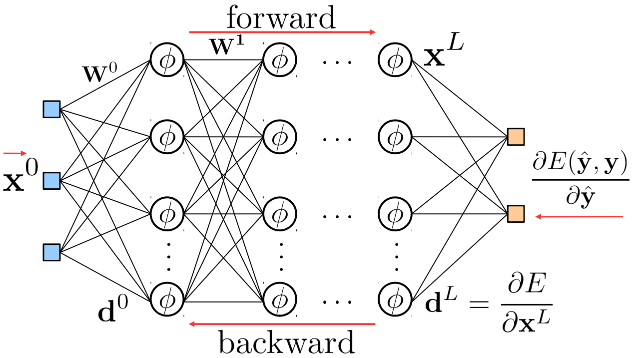
# Signal Propagation in Neural Networks




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

Backward: $\mathbf{d}^l = \mathbf{D}^l \left( \prod_{k=l+1}^{L-1} \mathbf{W}^k \mathbf{D}^k \right) \frac{\partial E}{\partial \mathbf{x}^L}$, $D_{ii}^l = \phi'(h_i^l)$ .

# Signal Propagation in Neural Networks




Forward: $\mathbf{h}^l = \mathbf{W}^l \mathbf{x}^l, \quad \mathbf{x}^{l+1} = \phi(\mathbf{h}^l) \;,$

Backward: $\mathbf{d}^l = \mathbf{D}^l \left( \prod_{k=l+1}^{L-1} \mathbf{W}^k \mathbf{D}^k \right) \dfrac{\partial E}{\partial \mathbf{x}^L}, \qquad D^l_{ii} = \phi'(h^l_i) \;.$

# Signal Propagation in Neural Networks

## Objective

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

## Why?

- Better trainability: faster convergence and stable training. [Glorot-2010, Klambauer-2017, Pennington-2017]
- Very deep CNNs and RNNs. [Pennington-2018, Chen-2018]
- Improved robustness? [Lin-2019]
- Improved generalization?

## What if signals are not preserved?

- Signals can saturate $\Rightarrow$ vanishing/exploding gradients.

# Signal Propagation in Neural Networks

## Objective

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

## Why?

- Better trainability: faster convergence and stable training. [Glorot-2010, Klambauer-2017, Pennington-2017]
- Very deep CNNs and RNNs. [Pennington-2018, Chen-2018]
- Improved robustness? [Lin-2019]
- Improved generalization?

## What if signals are not preserved?

- Signals can saturate $\Rightarrow$ vanishing/exploding gradients.

# Signal Propagation in Neural Networks

## Objective

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

## Why?

- Better trainability: faster convergence and stable training. [Glorot-2010, Klambauer-2017, Pennington-2017]
- Very deep CNNs and RNNs. [Pennington-2018, Chen-2018]
- Improved robustness? [Lin-2019]
- Improved generalization?

## What if signals are not preserved?

- Signals can saturate $\Rightarrow$ 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

- Self-normalizing neural networks. [Klambauer-2017]
- 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

- Self-normalizing neural networks. [Klambauer-2017]
- 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

- Self-normalizing neural networks. [Klambauer-2017]
- 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

- Self-normalizing neural networks. [Klambauer-2017]
- 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

| | **Dynamical Isometry** | **Self normalization** | **Batch normalization** |
|---|---|---|---|
| Weights ($\mathbf{W}$) | Orthogonal | VS-init. | Unconstrained |
| Activations ($\phi$) | Most | SELU | All |
| Preactivations ($\mathbf{h}$) | Linear region | Unconstrained | $\mathbb{E} \approx 0, \mathbb{V} \approx 1$ |
| Forward signal ($\mathbf{x}$) | Constrained | $\mathbb{E} \approx 0, \mathbb{V} \approx 1$ | Constrained |
| Backward signal ($\mathbf{d}$) | Constrained | Unconstrained | Unconstrained |
| Functionality | Pseudo-linear | Nonlinear | Nonlinear |

[Philipp-2018]

# Summary of Existing Approaches

| | **Dynamical Isometry** | **Self normalization** | **Batch normalization** |
|---|---|---|---|
| Weights ($\mathbf{W}$) | Orthogonal | VS-init. | Unconstrained |
| Activations ($\phi$) | Most | SELU | All |
| Preactivations ($\mathbf{h}$) | Linear region | Unconstrained | $\mathbb{E} \approx 0, \mathbb{V} \approx 1$ |
| Forward signal ($\mathbf{x}$) | Constrained | $\mathbb{E} \approx 0, \mathbb{V} \approx 1$ | Constrained |
| Backward signal ($\mathbf{d}$) | Constrained | Unconstrained | Unconstrained |
| Functionality | Pseudo-linear | Nonlinear | Nonlinear |

[Philipp-2018]

### Key missing points

- Do not preserve signal in both forward and backward directions.
- Do not maintain nonlinear functionality.

# Bidirectional Self-Normalizing Neural Networks

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

| | Dynamical Isometry | Self normalization | BSNN (Ours) |
|---|---|---|---|
| Weights ($\mathbf{W}$) | Orthogonal | VS-init. | Orthogonal |
| Activations ($\phi$) | Most | SELU | GPN-versions |
| Preactivations ($\mathbf{h}$) | Linear region | Unconstrained | Unconstrained |
| Forward signal ($\mathbf{x}$) | Constrained | $\mathbb{E} \approx 0, \mathbb{V} \approx 1$ | $\mathbb{V} + \mathbb{E}^2 \approx 1$ |
| Backward signal ($\mathbf{d}$) | Constrained | Unconstrained | $\mathbb{V} + \mathbb{E}^2 \approx 1$ |
| Functionality | Pseudo-linear | Nonlinear | Nonlinear |

# Bidirectional Self-Normalizing Neural Networks

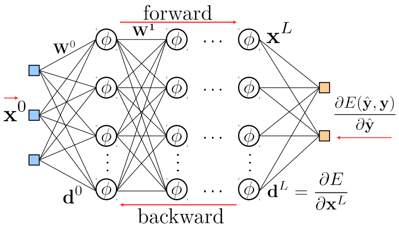
Key idea: New class of activation functions: Gaussian-Poincaré Normalized activations.

| | **Dynamical Isometry** | **Self normalization** | **BSNN (Ours)** |
|---|---|---|---|
| Weights ($\mathbf{W}$) | Orthogonal | VS-init. | Orthogonal |
| Activations ($\phi$) | Most | SELU | GPN-versions |
| Preactivations ($\mathbf{h}$) | Linear region | Unconstrained | Unconstrained |
| Forward signal ($\mathbf{x}$) | Constrained | $\mathbb{E} \approx 0, \mathbb{V} \approx 1$ | $\mathbb{V} + \mathbb{E}^2 \approx 1$ |
| Backward signal ($\mathbf{d}$) | Constrained | Unconstrained | $\mathbb{V} + \mathbb{E}^2 \approx 1$ |
| Functionality | Pseudo-linear | Nonlinear | Nonlinear |

# Bidirectional Self-Normalization

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

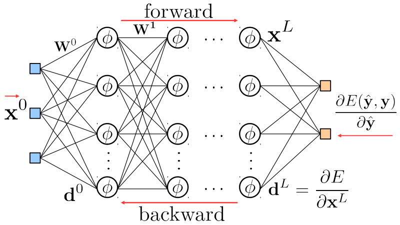
$\mathbf{W}^l \in \mathbb{R}^{n\times n}$, $l = \{1, \dots, L-1\}$ and $\phi : \mathbb{R} \to \mathbb{R}$.



# Bidirectional Self-Normalization

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

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




$$\text{Forward:} \quad \mathbf{h}^l = \mathbf{W}^l \mathbf{x}^l, \quad \mathbf{x}^{l+1} = \phi(\mathbf{h}^l),$$

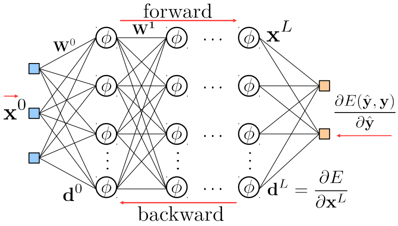
$$\text{Backward:} \quad \mathbf{d}^l = \mathbf{D}^l \left( \prod_{k=l+1}^{L-1} \mathbf{W}^k \mathbf{D}^k \right) \frac{\partial E}{\partial \mathbf{x}^L}, \qquad D^l_{ii} = \phi'(h^l_i),$$

$$\text{Gradient:} \quad \frac{\partial E}{\partial \mathbf{W}^l} = \mathbf{d}^l (\mathbf{x}^l)^T .$$

# 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 $\phi : \mathbb{R} \to \mathbb{R}$.

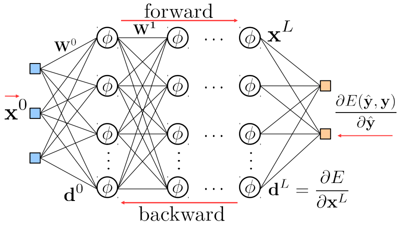



$$\text{Require:} \quad \|\mathbf{x}^1\|_2 = \|\mathbf{x}^2\|_2 = \ldots = \|\mathbf{x}^L\|_2, \quad \{\mathbf{W}^l\}, \phi \text{ constrained },$$
$$\|\mathbf{d}^1\|_2 = \|\mathbf{d}^2\|_2 = \ldots = \|\mathbf{d}^L\|_2, \quad \{\mathbf{W}^l\}, \phi' \text{ constrained }.$$

# Bidirectional Self-Normalization

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

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




$$\text{Require:} \quad \|\mathbf{x}^1\|_2 = \|\mathbf{x}^2\|_2 = \ldots = \|\mathbf{x}^L\|_2, \quad \{\mathbf{W}^l\}, \phi \text{ constrained },$$
$$\|\mathbf{d}^1\|_2 = \|\mathbf{d}^2\|_2 = \ldots = \|\mathbf{d}^L\|_2, \quad \{\mathbf{W}^l\}, \phi' \text{ constrained }.$$

No vanishing/exploding gradients.

# Orthogonal Weight Matrices

$$(\mathbf{W}^l)^T\mathbf{W}^l = \mathbf{W}^l(\mathbf{W}^l)^T = \mathbf{I}_n \ .$$

### 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

$$(\mathbf{W}^l)^T \mathbf{W}^l = \mathbf{W}^l (\mathbf{W}^l)^T = \mathbf{I}_n \ .$$

## 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

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

## Key facts

- If $\mathbf{W}^l$ is orthogonal, $\mathbf{h}^l$ 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 a 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 $\mathbf{W}^l$ is orthogonal, $\mathbf{h}^l$ 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 $\mathbf{W}^l$ is orthogonal, $\mathbf{h}^l$ 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 $a\phi(h) + b$.

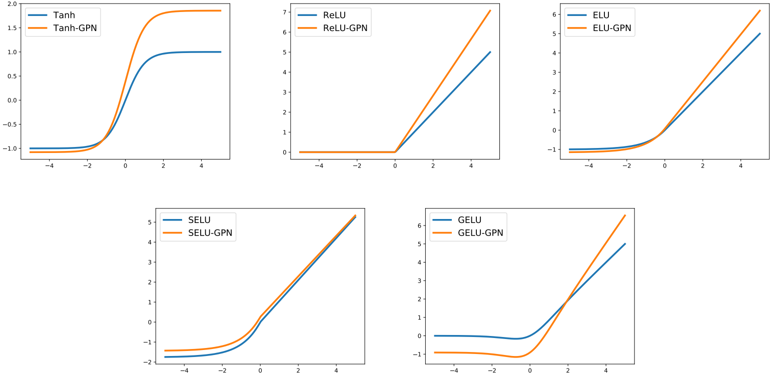
# GPN Activations

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

## Key facts

- If $\mathbf{W}^l$ is orthogonal, $\mathbf{h}^l$ 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 $\epsilon > 0$

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

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 $\epsilon > 0$

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

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 $\epsilon > 0$

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

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 $\epsilon > 0$

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

Examples: Multivariate Gaussian
Any distribution on $n$-unit-sphere scaled by $\sqrt{n}$.

Intuitive visualization of Gaussian. [Vershynin-2018]

# 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 $\phi : \mathbb{R} \to \mathbb{R}$ 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 $\phi : \mathbb{R} \to \mathbb{R}$ 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 $\epsilon > 0$

$$\mathbb{P}\left\{\left|\frac{1}{n}\|\mathbf{D}\mathbf{y}\|_2^2 - \|\mathbf{y}\|_2^2\right| \geq \epsilon\right\} \to 0 ,$$

as $n \to \infty$.

- Multiplication by $\mathbf{D}$ preserves the norm with high probability.

# Norm Preservation Theorems

**Theorem 1: Forward norm preservation**

Random vector

$$\left(\phi\left(\mathbf{w}_1^T\mathbf{x}\right), \phi\left(\mathbf{w}_2^T\mathbf{x}\right), \ldots, \phi\left(\mathbf{w}_n^T\mathbf{x}\right)\right)^T ,$$

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 $\epsilon > 0$

$$\mathbb{P}\left\{\left|\frac{1}{n}\|\mathbf{D}\mathbf{y}\|_2^2 - \|\mathbf{y}\|_2^2\right| \geq \epsilon\right\} \to 0 ,$$

as $n \to \infty$.

- Multiplication by $\mathbf{D}$ preserves the norm with high probability.

# Norm Preservation Theorems

**Theorem 1: Forward norm preservation**

Random vector

$$\left(\phi\left(\mathbf{w}_1^T\mathbf{x}\right), \phi\left(\mathbf{w}_2^T\mathbf{x}\right), \ldots, \phi\left(\mathbf{w}_n^T\mathbf{x}\right)\right)^T ,$$

is TSC.

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

**Theorem 2: Backward norm preservation**

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

$$\mathbb{P}\left\{\left|\frac{1}{n}\|\mathbf{D}\mathbf{y}\|_2^2 - \|\mathbf{y}\|_2^2\right| \geq \epsilon\right\} \to 0 ,$$

as $n \to \infty$.

- Multiplication by $\mathbf{D}$ preserves the norm with high probability.

# Norm Preservation Theorems

**Theorem 1: Forward norm preservation**

Random vector

$$\left(\phi\left(\mathbf{w}_1^T\mathbf{x}\right), \phi\left(\mathbf{w}_2^T\mathbf{x}\right), \ldots, \phi\left(\mathbf{w}_n^T\mathbf{x}\right)\right)^T ,$$

is TSC.

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

**Theorem 2: Backward norm preservation**

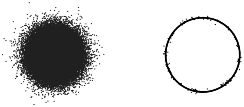
Let $\mathbf{y} \in \mathbb{R}^n$ with $\|\mathbf{y}\|_\infty$ and $D_{ii} = \phi'(\mathbf{w}_i^T\mathbf{x})$. Then for any $\epsilon > 0$

$$\mathbb{P}\left\{\left|\frac{1}{n}\|\mathbf{D}\mathbf{y}\|_2^2 - \|\mathbf{y}\|_2^2\right| \geq \epsilon\right\} \to 0 ,$$

as $n \to \infty$.

- Multiplication by $\mathbf{D}$ preserves the norm with high probability.

# Sketch of the Proofs



Key theory: Concentration of measure [Vershynin-2018]

- Most mass of some high-dimensional probability disctributions is concentrated around a certain range.
- Lipschitz functions do not affect this property.

# Sketch of the Proof: Theorem 1

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

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

# Sketch of the Proof: Theorem 1

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

$$\downarrow$$

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

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

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

# Sketch of the Proof: Theorem 1

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

$$\downarrow$$

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

$$\downarrow$$

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

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

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

Proof follows from Lipschitz continuous and GPN function $\phi$.

# Sketch of the Proof: Theorem 1

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

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

$$\downarrow$$

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

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

$$\downarrow$$

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

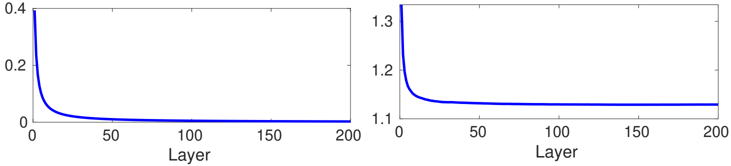
Proof follows from Lipschitz continuous and GPN function $\phi$.

$$\downarrow$$

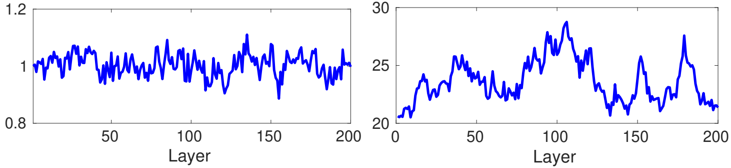
$$\mathbf{1}$$

Each of these steps are rigorously proved.

# Synthetic Experiments



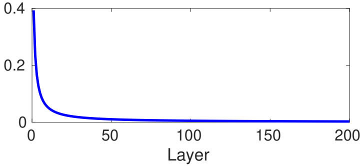

Tanh, $\|\mathbf{x}^l\|_2^2/n$ Tanh, $\|\frac{\partial E}{\partial \mathbf{W}^l}\|_F$



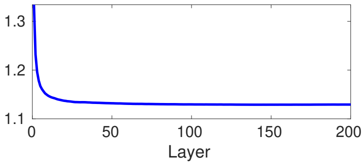

Tanh-GPN, $\|\mathbf{x}^l\|_2^2/n$ Tanh-GPN, $\|\frac{\partial E}{\partial \mathbf{W}^l}\|_F$

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

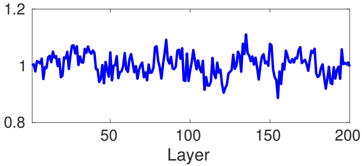
# Synthetic Experiments



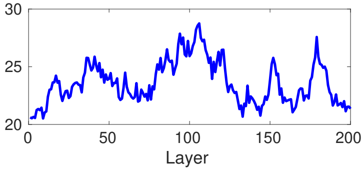

Tanh, $\|\mathbf{x}^l\|_2^2/n$




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




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

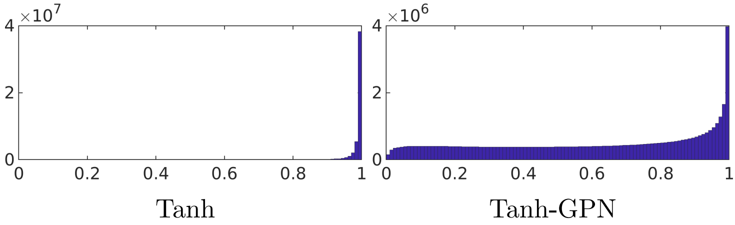



Tanh-GPN, $\|\frac{\partial E}{\partial \mathbf{W}^l}\|_F$

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

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

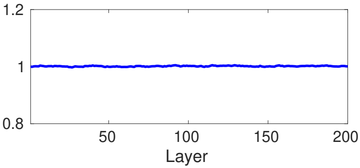
# Synthetic Experiments



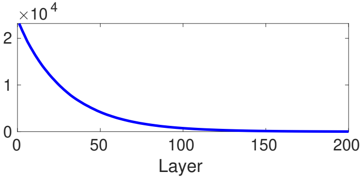

Histogram of $\phi'(h_i^l)$, *i.e.*, singular value distribution, $L = 200, n = 500$.

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

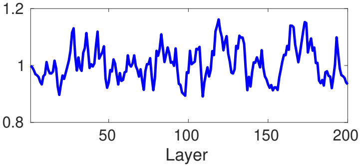
# Synthetic Experiments



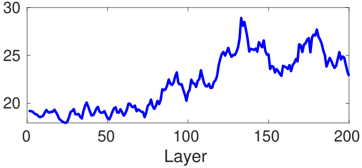

SELU, $\|\mathbf{x}^l\|_2^2/n$




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



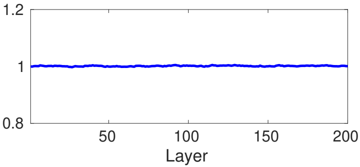

SELU-GPN, $\|\mathbf{x}^l\|_2^2/n$



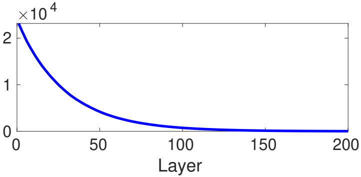

SELU-GPN, $\|\frac{\partial E}{\partial \mathbf{W}^l}\|_F$

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

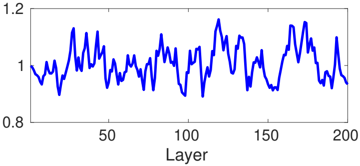
# Synthetic Experiments



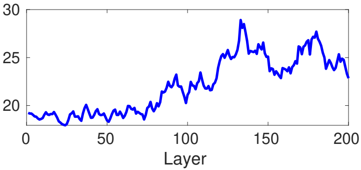

SELU, $\|\mathbf{x}^l\|_2^2/n$




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




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

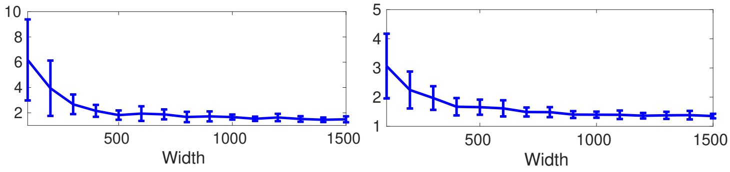



SELU-GPN, $\|\frac{\partial E}{\partial \mathbf{W}^l}\|_F$

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

SELU suffers gradient explosion while SELU-GPN is stable.

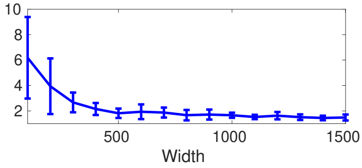
# Synthetic Experiments



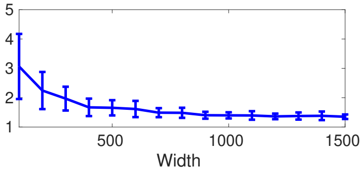

Tanh-GPN SELU-GPN

Gradient norm ratio, *i.e.*, $\max_l \| \frac{\partial E}{\partial \mathbf{W}^l} \|_F / \min_l \| \frac{\partial E}{\partial \mathbf{W}^l} \|_F$, $L = 200$.

# Synthetic Experiments




Tanh-GPN




SELU-GPN

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

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

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

# Experiments on Real-World Data

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

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

GPN yields significant improvements in training accuracy.

# Experiments on Real-World Data

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

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

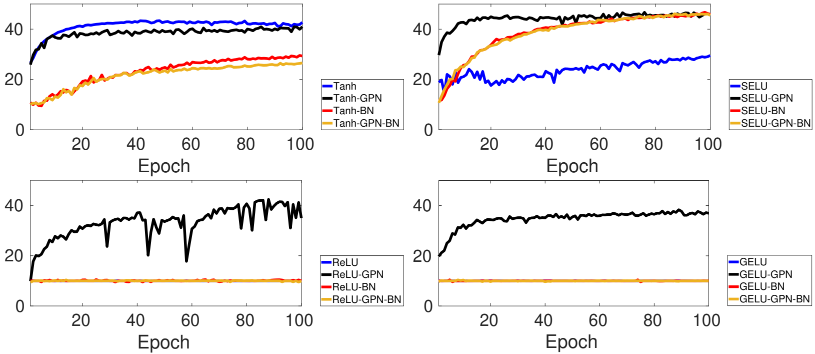
# Experiments on Real-World Data

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

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

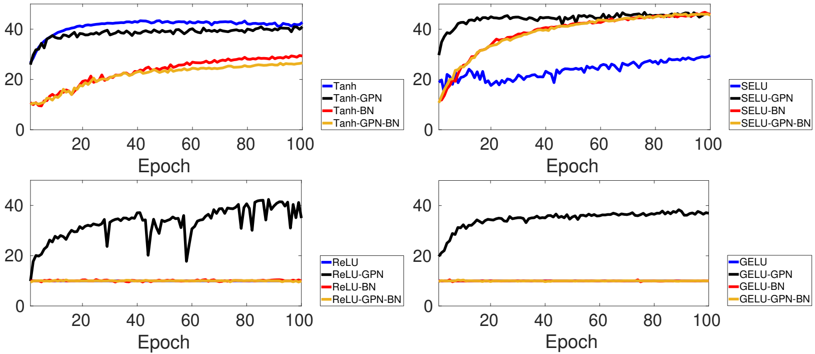
GPN yields improvements in many cases.

# Experiments on Real-World Data




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

# 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!