On Complex-Valued Equivariant Neural Networks for Radio Frequency Fingerprinting

Carter Brown[1], Enrico Mattei[1]

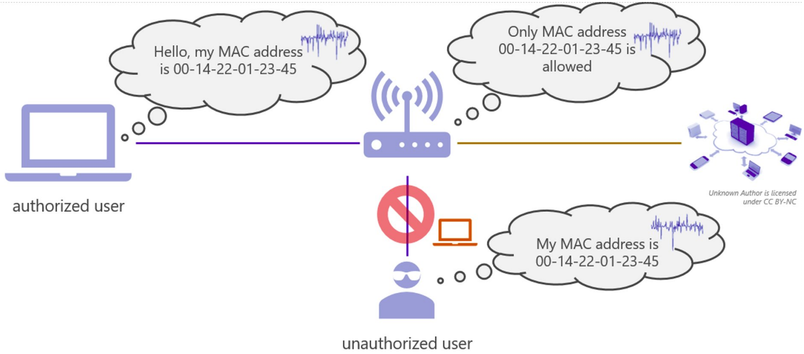
Camila A. Ramirez

[1] Equal Contribution

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Overview

Why finger print?



Why equivariance?

Test data often contains perturbations not present in training data.

Complex scalar multiplication:

- Attenuation (magnitude)
- Phase rotation

Complex vector multiplication:

- Channel and receiver effects

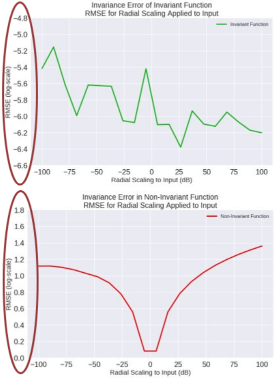
Performance of learning algorithms degrades.

Can we learn representations of RF signals that are invariant with respect to perturbations of the input?

[1] R. Chakraborty, Y. Xing, S. Yu, SurReal: complex-valued deep learning as principled transformations on a rotational Lie group, arXiv preprint arXiv:1910.11334

Approach

- Use polar coordinates (r, θ) to represent complex data $z \in \mathbb{C} \setminus 0 + 0i$ from signals
- Create pseudo-metric that is invariant to perturbations.
- Create convolutional layers that are equivariant perturbations.

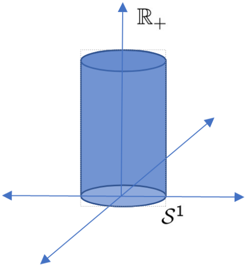


[2] S. Mallat, "Understanding deep convolutional networks, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. Vol. 374, Issue 2065, 2016.

Signals

Complex signals can be represented as $\mathcal{M} = \mathbb{R}_+ \times \mathcal{S}^1$

- $\mathcal{S}^1$ is the unit circle
- $p_i \in \mathcal{S}^1$ is identified with $u(\theta_i) = [\cos(\theta_i) \;\; \sin(\theta_i)]'$



Degradation

Attenuation and phase rotation can be defined as $\mathcal{G} = \mathbb{R}_+ \times \mathcal{SO}(2)$ with operation

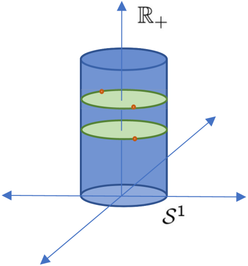
$$\bullet : g_1 = (r_1, R_1), g_2 = (r_2, R_2) \mapsto g_3 = (r_1 r_2, R_1 R_2)$$

and $R_i = \begin{bmatrix} \cos(\phi_i) & -\sin(\phi_i) \\ \sin(\phi_i) & \cos(\phi_i) \end{bmatrix}$

Group Action

Define a group action $* : \mathcal{G} \times \mathcal{M} \to \mathcal{M}$
$(r, R) * (x, u(\theta)) \mapsto (rx, Ru(\theta))$

- Attenuation: scalar multiplication in the real component x
- Phase rotation: matrix multiplication R on the unitary vector $u(\theta)$



Pseudo-Metric

Define a function $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ as:

$$d^2(m_1, m_2) = \log^2\left(\frac{x_2}{x_1}\right) + \arccos^2\left(u(\theta_2)' u(\theta_1)\right)$$

We use this semi-metric $d(\cdot, \cdot)$ to define a convolutional operation and show it's invariance to the group action $(\mathcal{G}, *)$.

Invariance

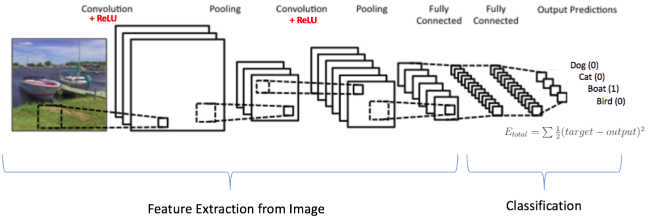
Consider a group element $g = (r, R) \in \mathcal{G}$ acting on $m_1, m_2 \in \mathcal{M}$:

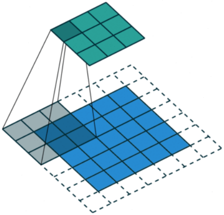
$$d^2(g * m_1, g * m_2) = \log^2\left(\frac{rx_2}{rx_1}\right) + \arccos^2\left((Ru(\theta_2))' Ru(\theta_1)\right)$$

$d^2(g * m_1, g * m_2) = d^2(m_1, m_2)$

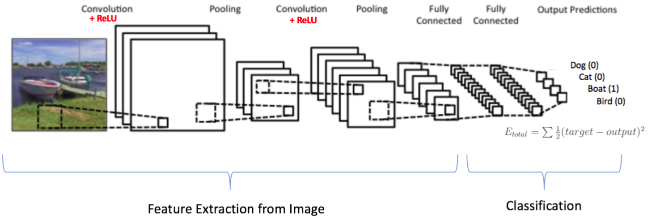
$$\begin{aligned}
&d^2(g * m_1, g * m_2) \\
&= \log^2\left(\tfrac{rx_2}{rx_1}\right) + \arccos^2\left((Ru(\theta_2))' Ru(\theta_1)\right) \\
&= \log^2\left(\tfrac{rx_2}{rx_1}\right) + \arccos^2\left(u(\theta_2)' R' R u(\theta_1)\right) \\
&= \log^2\left(\tfrac{rx_2}{rx_1}\right) + \arccos^2\left(u(\theta_2)' u(\theta_1)\right) \\
&= d^2(m_1, m_2)
\end{aligned}$$

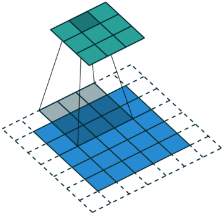
Convolutional Neural Network



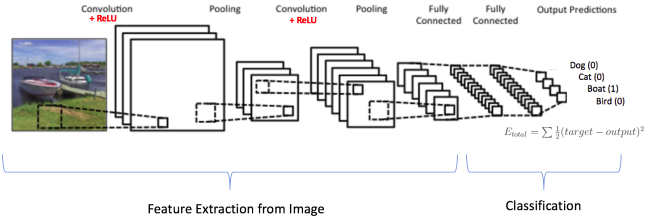


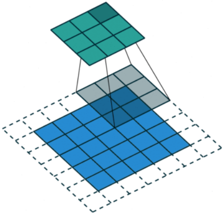
Convolutional Neural Network



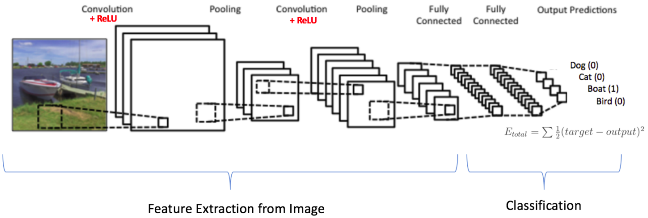


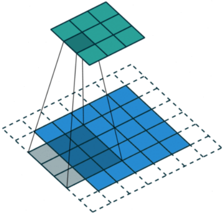
Convolutional Neural Network



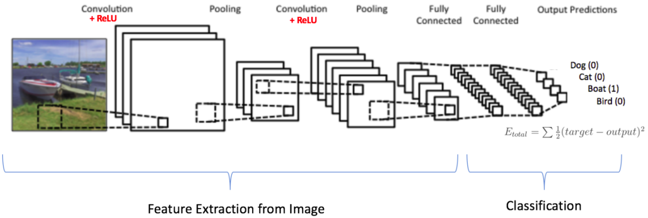


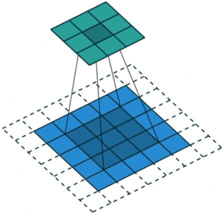
Convolutional Neural Network



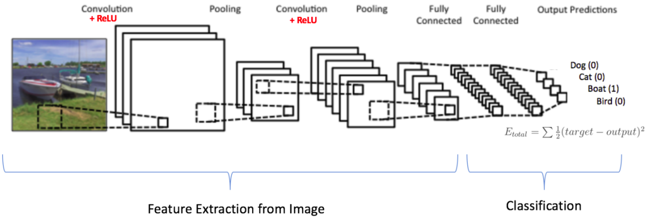


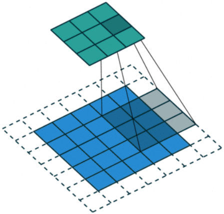
Convolutional Neural Network



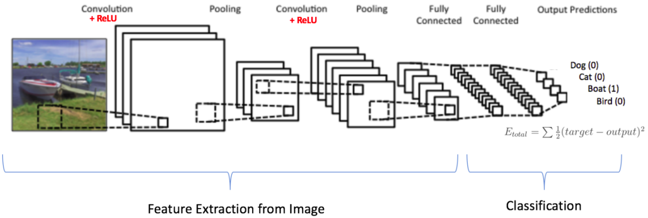


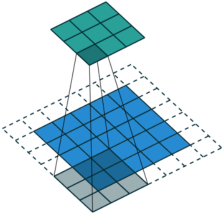
Convolutional Neural Network



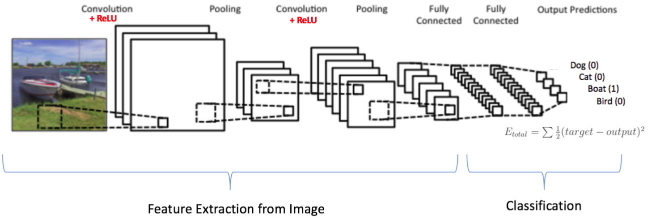


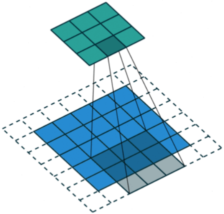
Convolutional Neural Network



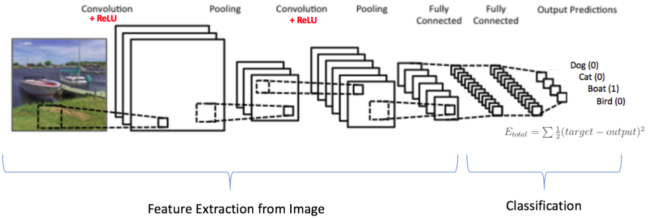


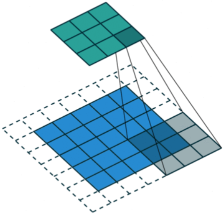
Convolutional Neural Network





Convolutional Neural Network





Convolution

Define a convolutional map

$$\mathcal{F}\{w_i, m_i\} := \underset{m \in \mathcal{M}}{\arg\min} \sum_{i=1}^{N} w_i d^2(m_i, m)$$

$$\underset{m \in \mathcal{M}}{\arg\min} \sum_{i=1}^{N} w_i [\log^2\left(\frac{x}{x_i}\right) + \arccos^2\left(u(\theta)' u(\theta_i)\right)]$$

for $\{m_i\} \in \mathcal{M}$ some signal window and $\sum_{i=1}^{N} w_i = 1$.

Convolution

$\mathcal{F}$ must be unique to show equivariance to the group action $(\mathcal{G}, *)$

$$g * \mathcal{F}(\{wi\}, \{m_i\}) = \mathcal{F}(\{wi\}, \{g * m_i\})$$

- $\mathcal{F}$ is not properly defined for equally-spaced sets of points $\{m_i\} \in \mathcal{M}$. Given some ordering, $\{m_i\}$ are pairwise ordered equidistant. We call these sets *equidistant*.
- $\{m_i\}$ are closed under $(\mathcal{G}, *)$: rotating and scaling equidistant points remain equidistant.

We redefine $\mathcal{F}$ piecewise and thus satisfy uniqueness, allowing equivariance to hold.

Piecewise Convolution

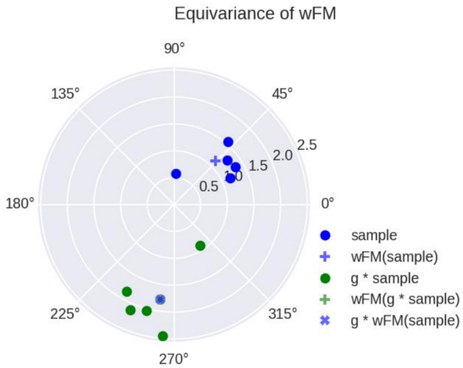
$$\mathcal{F}(\{w_i\},\{m_i\}) = \begin{cases} (\prod_{i=1}^{N} x_i^{w_i}, u(\min\{w_i\theta_i\}) & R = 0 \\ (\prod_{i=1}^{N} x_i^{w_i}, u(\hat{\theta})) & R > 0 \end{cases}$$

- $R = \sqrt{(\sum_{i=1}^{N} w_i \cos(\theta_i))^2 + (\sum_{i=1}^{N} w_i \sin(\theta_i))^2}$
- $\theta = \arctan \frac{\sum_{i=1}^{N} w_i \sin(\theta_i)}{\sum_{i=1}^{N} w_i \cos(\theta_i)}$
-

$$\hat{\theta} = \begin{cases} \theta & \sum_{i=1}^{N} w_i \cos(\theta_i) > 0 \text{ and } \sum_{i=1}^{N} w_i \sin(\theta_i) > 0 \\ \pi + \theta & \sum_{i=1}^{N} w_i \cos(\theta_i) < 0 \\ 2\pi + \theta & \sum_{i=1}^{N} w_i \cos(\theta_i) > 0 \text{ and } \sum_{i=1}^{N} w_i \sin(\theta_i) < 0 \end{cases}$$

Equivariance of Convolution

$$g * \mathcal{F}(\{wi\}, \{m_i\}) = \mathcal{F}(\{wi\}, \{g * m_i\})$$



Equivariance to Group Action

$$g * \mathcal{F}(\{w_i\}, \{m_i\}) = \mathcal{F}(\{w_i\}, \{g * m_i\})$$

Setup:

Let $m^* = \mathcal{F}(\{w_i\}, \{m_i\}) = \arg\min_{m \in \mathcal{M}} \sum w_i d^2(m_i, m)$

Let $\tilde{m} = \mathcal{F}(\{w_i\}, g * \{m_i\}) = \arg\min_{m \in \mathcal{M}} \sum w_i d^2(g * m_i, m)$

Part 1

$\sum_i w_i d^2(g * m_i, \tilde{m}) = \sum_i w_i d^2(m_i, m^*)$

$\sum_i w_i d^2(g * m_i, \tilde{m})$
$= \min_{\tilde{\tilde{m}} \in g * \mathcal{M}} \sum_i w_i d^2(g * m_i, \tilde{\tilde{m}})$
$= \min_{g^{-1} * \tilde{\tilde{m}} \in g^{-1} g * \mathcal{M}} \sum_i w_i d^2(g^{-1} g * m_i, g^{-1} * \tilde{\tilde{m}})$
by invariance of distance metric
$= \min_{m^{**} \in \mathcal{M}} \sum_i w_i d^2(m_i, m^{**})$
$= \sum_i w_i d^2(m_i, m^*)$
by uniqueness of $\mathcal{F}$.

Part 2

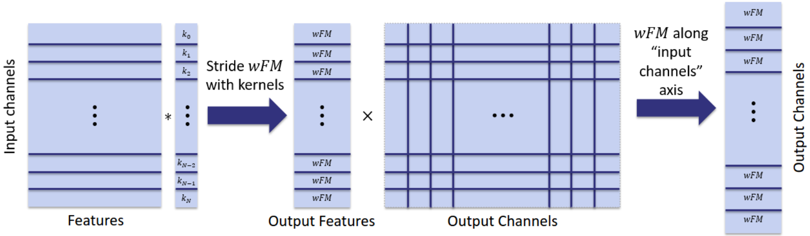
$\tilde{m} = g * m^*$

$\sum_i w_i d^2(g * m_i, \tilde{m})$
$= \sum_i w_i d^2(m_i, m^*)$
$= \sum_i w_i d^2(g * m_i, g * m^*)$
by invariance of distance metric.
$\Rightarrow \tilde{m} = g * m^*$, by uniqueness of $\mathcal{F}$.

$\therefore \mathcal{F}(\{w_i\}, \{g * m_i\}) = \tilde{m} = g * m^* = g * \mathcal{F}(\{w_i\}, \{m_i\}) \quad \square$

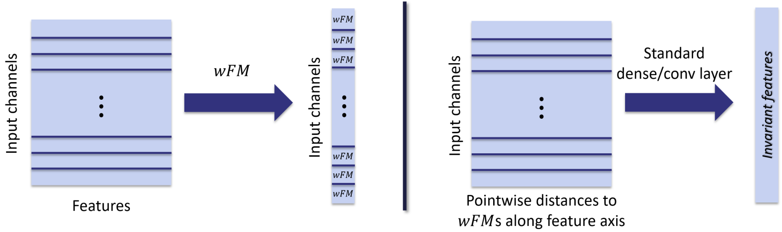
PolarNet

Along each input channel axis, replace the dot product in a convolutional layer by $\mathcal{F}$.



PolarNet

Invariance of $d^2(\cdot, \cdot)$ and equivariance of $\mathcal{F}(\cdot, \cdot)$ allows for the construction of an invariant layer as the distance from each input feature m_i to $\mathcal{F}(\{w_i\}, \{m_i\})$.

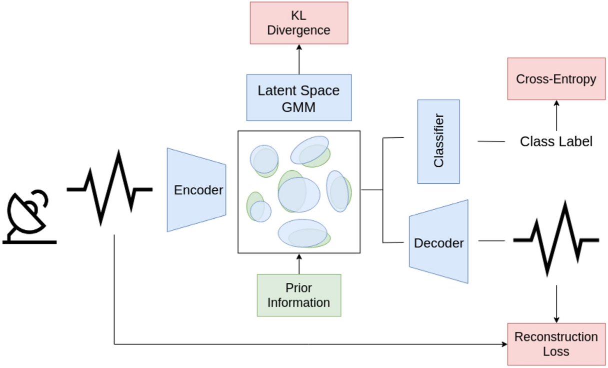


Model Comparison

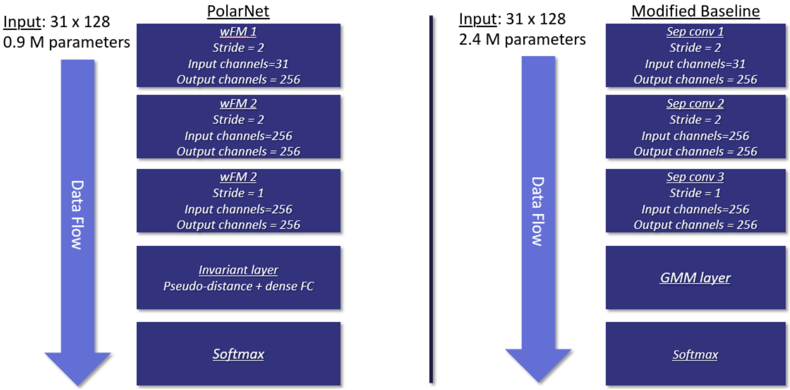
- PolarNet: polar complex representation (r, θ)
- CartesianNet: Cartesian complex representation (x, y)
- Baseline model: our baseline model with default configuration
- Modified baseline model: baseline model with modifications to make the processing closer to PolarNet

Model Comparison

Baseline Model



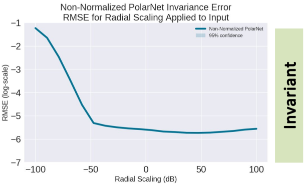
Model Comparison



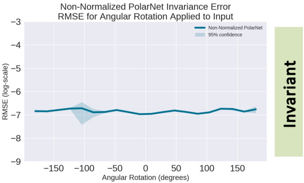
Invariance Error: Non-Normalized

PolarNet with Pseudo Metric

Radial Scaling

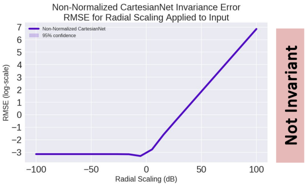


Angular Rotation

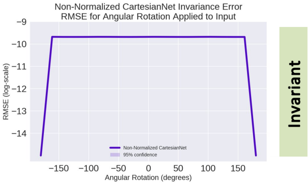


CartesianNet with Euclidean Distance

Radial Scaling



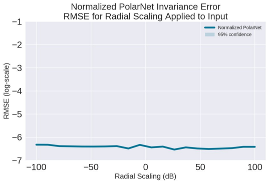
Angular Rotation



Invariance Error: Normalized

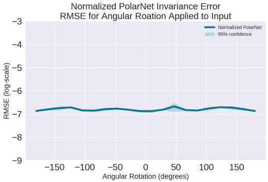
PolarNet with Pseudo Metric

Radial Scaling



Invariant

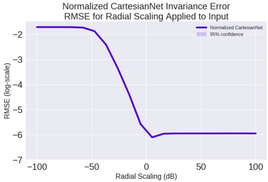
Angular Rotation



Invariant

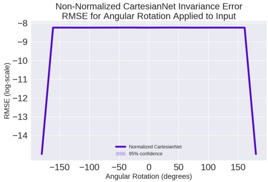
CartesianNet with Euclidean Distance

Radial Scaling



Invariant

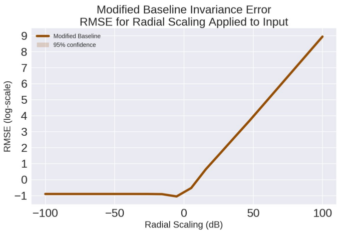
Angular Rotation



Invariant

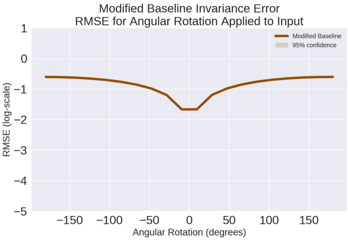
Invariance Error: Non-Normalized

Radial Scaling



Not Invariant

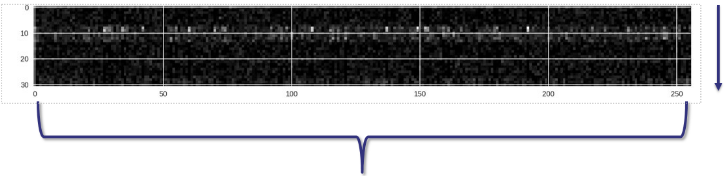
Angular Rotation



Not Invariant

What is PolarNet learning?

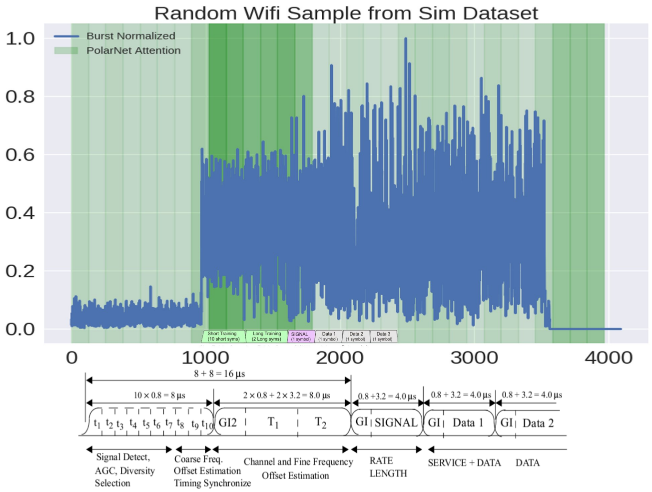
wFM kernel in first layer of PolarNet



Weighted mean of 31 incoming windows

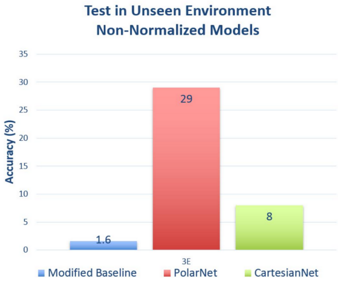
256 weighted-mean features

What is PolarNet learning?



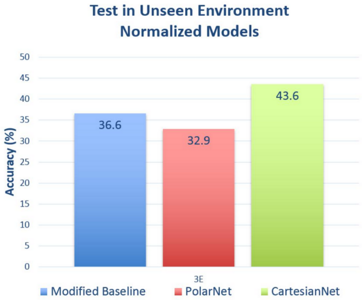
Performance: Non-normalized

- No scale normalization applied in preprocessing, only mean centering
- PolarNet outperforms the other two models
- Comparing PolarNet to un-normalized networks shows the inherent invariance



Performance: Normalized

- Normalize bursts by max value in preprocessing
- Now all models invariant to radial scaling
- Modified Baseline model still not invariant to rotations
- CartesianNet has same invariances as PolarNet but is numerically more stable



From Complex Scalar to Impulse Response

- The perturbation that we really want to be invariant to is convolution with an FIR filter.
- In frequency domain, convolution becomes element-wise multiplication.
- We can extend previous equations to filter perturbations.
- Thus construct a network invariant to filters (channels) in the frequency-domain.

FIR Filter

$\mathcal{G} = \mathbb{R}^n_+ \times \mathcal{T}^n$ with $\mathcal{T}^n \leq \mathcal{SO}(2n)$ the maximal torus subgroup, and operation $\bullet : g_1 = (\vec{r}_1, T_1), g_2 = (\vec{r}_2, T_2) \mapsto g_3 = (\vec{r}_1\vec{r}_2, T_1 T_2)$ where $\vec{r}_1\vec{r}$ is element-wise multiplication and $T \in \mathcal{T}^n$ is of the form:

$$T = \begin{bmatrix} R_1 & & & \\ & R_2 & & \\ & & \ddots & \\ & & & R_n \end{bmatrix}$$

with $R_i \in \mathcal{SO}(2)$.

Signal

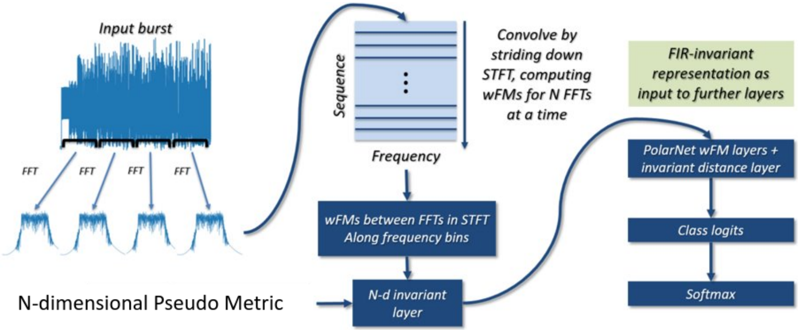
$\mathcal{M} = \mathbb{R}^n_+ \times \mathcal{S}^1_1 \times \cdots \times \mathcal{S}^1_n$ with
$S = [\vec{u}_1(\theta_1) \cdots \vec{u}_n(\theta_n)] \in \mathcal{S}^1_1 \times \cdots \times \mathcal{S}^1_n$ a block diagonal matrix:

$$S = \begin{bmatrix} \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix} & & & \\ & \begin{bmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{bmatrix} & & \\ & & \ddots & \\ & & & \begin{bmatrix} \cos(\theta_n) \\ \sin(\theta_n) \end{bmatrix} \end{bmatrix}$$

Degradation

Group action $*:\mathcal{G}\times\mathcal{M}\to\mathcal{M}$ as $(\vec{r},T)*(\vec{x},S)\mapsto(\vec{r}\vec{x},TS)$.
Element-wise multiplication in the real component and block rotation $R_i|_T$ at each unitary vector $\vec{u}_i(\theta_i)|_S$.

Channel-Invariant Network Architecture



The End