We present a new architecture for implementing an Efficient Unitary Neural Network (EUNNs)

1) The representation capacity of the unitary space in an EUNN is fully tunable, ranging from a subspace of SU(N) to the entire unitary space.
2) The computational complexity for training an EUNN is merely O(1) per parameter.
3) We find that our architecture significantly outperforms both other state-of-the-art unitary RNNs and the LSTM architecture, in terms of the final performance and/or the wall-clock training speed.

In mathematics, a complex square matrix $U$ is unitary if its conjugate transpose $U^*$ is also its inverse $U^* U = U U^* = I$.

Keep the norm of vectors:

$\|Ux\| = \|x\|$

**Advantages**

1) **Efficient**: O(1) operation per parameter
2) **Tunable**: span from small subspace to full unitary space
3) **Easy implementation**: element-wise functions
4) **FFT approximation** provides further speed-up

**Model**

Tunable Efficient Unitary Parametrization

**Implementation Algorithm**

Algorithm 1 Efficient implementation for $F$ with parameter $\theta$ and $\phi$:

**Input:** input $x$, size $N$; parameters $\theta$ and $\phi$, size $N/2$; constant permutation index list $\text{ind}_1$ and $\text{ind}_2$.

**Output:** output $y$, size $N$. $v_1 \leftarrow \text{concatenate}(\cos \theta, \cos \phi \ast \exp(i \theta))$

$v_2 \leftarrow \text{concatenate}(\sin \theta, \sin \phi \ast \exp(i \phi))$

$v_3 \leftarrow \text{permute}(v_1, \text{ind}_1)$

$v_4 \leftarrow \text{permute}(v_2, \text{ind}_1)$

$y \leftarrow v_1 \ast x \ast v_2 \ast \text{permute}(x, \text{ind}_2)$

$W = DF_A^{F_1} F_2 F_3 \ldots F_k$.

$W = DF_A^{F_1} F_2 F_3 \ldots F_k$

$F_{A}^{(i)} = R_{A}^{(i)} R_{A}^{(i-1)} \ldots R_{A}^{(0)} X_{/2-1, \_/2}$

capacity

**Contribution**

**Background**

**Gradient Vanishing/Explosion Problem**

**New Solution: Unitary RNN**

In mathematics, a complex square matrix $U$ is unitary if its conjugate transpose $U^*$ is also its inverse $U^* U = U U^* = I$.

Keep the norm of vectors:

$\|Ux\| = \|x\|$

**Related works**

- Restricted space Unitary Matrix Parametrization [1]
- Full-capacity Unitary Matrix by projection [2]
- Orthogonal parametrization by reflection[3]
- Orthogonal matrix by regularization[4]

**Reference & Code**


Tensorflow: https://github.com/jingli9111/EUNN-tensorflow
PyTorch: https://github.com/jingli9111/EUNN-PyTorch
Theano: https://github.com/jingli9111/EUNN-theano