SUPPLEMENTARY MATERIAL

For more details regarding notation, architectures, experiments, and proofs, please see (Balestriero & Baraniuk, 2018).

A. Notation

For concreteness, we focus here on processing multichannel images x, such as RGB color digital photographs. However, our analysis and techniques apply to signals of any index-dimensionality, including speech and audio signals, video signals, etc., simply by adjusting the appropriate dimensionalities. We will use two equivalent representations for the signal and feature maps, one based on tensors and one based on flattened vectors. In the *tensor* representation, the input image x contains $C^{(0)}$ channels of size $\left(I^{(0)} \times J^{(0)}\right)$ pixels, and the feature map $z^{(\ell)}$ contains $C^{(\ell)}$ channels of size $\left(I^{(\ell)} \times J^{(\ell)}\right)$ pixels. In the *vector* representation, $[\boldsymbol{x}]_k$ represents the entry of the k^{th} dimension of the flattened, vector version $\boldsymbol{x}$ of x. Hence, $D^{(\ell)} = C^{(\ell)} I^{(\ell)} J^{(\ell)}$, $C^{(L)} = C$, $I^{(L)} = 1$, and $J^{(L)} = 1$.

x	An input/observation tensor of shape (K, I, J)
$\boldsymbol{x}$	A vectorized input with length D
$\hat{y}(x)$	An output/prediction associated to input x
x_n	Observation n
y_n	Target variable associated to x_n, for classification $y_n \in \{1, \dots, C\},\ C > 1$, for regression $y_n \in \mathbb{R}^C, C \geq 1$
$\mathcal{D}$ (resp. $\mathcal{D}_s$)	Labeled training set with N (resp. N_s) samples $\mathcal{D} = \{(X_n, Y_n)_{n=1}^{N}\}$
$\mathcal{D}_u$	Unlabeled training set with N_u samples $\mathcal{D}_u = \{(X_n)_{n=1}^{N_u}\}$
$f^{(\ell)}_{\theta^{(\ell)}}$	Layer at level ℓ with internal parameters $\theta^{(\ell)}, \ell = 1, \dots, L$
Θ	Collection of all parameters $\Theta = \{\theta^{(\ell)}, \ell = 1, \dots, L\}$
f_Θ	Deep Neural Network (DNN) mapping with $f_\Theta : \mathbb{R}^D \to \mathbb{R}^C$
$(C^{(\ell)}, I^{(\ell)}, J^{(\ell)})$	Shape of the representation at layer ℓ with $(C^{(0)}, I^{(0)}, J^{(0)}) = (K, I, J)$ and $(C^{(L)}, I^{(L)}, J^{(L)}) = (C, 1, 1)$
$D^{(\ell)}$	Dimension of the flattened representation at layer ℓ with $D^{(\ell)} = C^{(\ell)} I^{(\ell)} J^{(\ell)}$, $D^{(0)} = D$ and $D^{(L)} = C$
$z^{(\ell)}(x)$	Representation of x at layer ℓ in an unflattened format of shape $(C^{(\ell)}, I^{(\ell)}, J^{(\ell)})$, with $z^{(0)}(x) = x$
$[z^{(\ell)}(x)]_{k,i,j}$	Value of the representation of x at layer ℓ, channel c, and spatial position (i, j)
$\boldsymbol{z}^{(\ell)}(x)$	Representation of x at layer ℓ in a flattened format of dimension $D^{(\ell)}$
$[\boldsymbol{z}^{(\ell)}(x)]_k$	Value at dimension k. One has $[z^{(\ell)}]_{c,i,j} = [\boldsymbol{z}^{(\ell)}]_k$ with $k = c \times I^{(\ell)} \times J^{(\ell)} + i \times J^{(\ell)} + j$

B. Deep Network Topologies and Datasets

We first present the topologies used in the experiments except for the notation ResNetD-W which is the standard wide ResNet based topology with depth D and width W. We thus have the following network architectures for smallCNN and largeCNN:

smallCNN

```
Conv2DLayer(layers[-1],32,3,pad='valid')
Pool2DLayer(layers[-1],2)
Conv2DLayer(layers[-1],64,3,pad='valid')
Pool2DLayer(layers[-1],2)
Conv2DLayer(layers[-1],128,1,pad='valid')
Pool2DLayer(layers[-1],2)
```

largeCNN

```
Conv2DLayer(layers[-1],96,3,pad='same')
Conv2DLayer(layers[-1],96,3,pad='full')
Conv2DLayer(layers[-1],96,3,pad='full')
Pool2DLayer(layers[-1],2)
Conv2DLayer(layers[-1],192,3,pad='valid')
Conv2DLayer(layers[-1],192,3,pad='full')
Conv2DLayer(layers[-1],192,3,pad='valid')
Pool2DLayer(layers[-1],2)
Conv2DLayer(layers[-1],192,3,pad='valid')
Conv2DLayer(layers[-1],192,1)
Pool2DLayer(layers[-1],2)
```

Figure 5. Description of the smallCNN and large CNN topologies.

where the Conv2DLayer(layers[-1],192,3,pad='valid') denotes a standard 2D convolution with 192 filters of spatial size $(3, 3)$ and with valid padding (no padding).

The used datasets are MNIST, CIFAR10-100 and SVHN. The standard datasets were used with standard train test set except for MNIST for which it was drawn at random unless specified in the experiment section.

C. Proofs

C.1. Proof of Theorem 1:

A composition of affine spline operators is an affine spline operator.

C.2. Proof of Theorem 2:

See (Amos et al., 2016) and adapt the result in the context of MASOs in the spirit of (Boyd & Vandenberghe, 2004).

C.3. Proof of Propositions 1-3

These are trivial by the definitions and the uniqueness of a piecewise affine convex mapping. By definition of the mappings, the correspondence is direct.

C.4. Proof Proposition 4: Examples of DN Layers as MASOs

We provide some examples to better illustrate this layer reparametrization.

Affine Transform+Activation: Any MASO following an affine transformation can be computed via definition of a new reparametrized MASO with

$$
\begin{aligned}
S\left[A_\sigma^{(\ell)}, b_\sigma^{(\ell)}\right](W^{(\ell)}x + b_{\boldsymbol{W}}^{(\ell)}) &= \begin{bmatrix} \max_{r=1,\dots,R_1} [A_\sigma^{(\ell)}]_{1,r,.}^T (W^{(\ell)}x + b_{\boldsymbol{W}}^{(\ell)}) + [b_\sigma^{(\ell)}]_{1,r} \\ \dots \\ \max_{r=1,\dots,R_K} [A_\sigma^{(\ell)}]_{K,r,.}^T (W^{(\ell)}x + b_{\boldsymbol{W}}^{(\ell)}) + [b_\sigma^{(\ell)}]_{K,r} \end{bmatrix} \\
&= \begin{bmatrix} \max_{r=1,\dots,R_1} (W^{(\ell)T}[A_\sigma^{(\ell)}]_{1,r})^T x + [A_\sigma^{(\ell)}]_{1,r}^T b_{\boldsymbol{W}}^{(\ell)} + [b_\sigma^{(\ell)}]_{1,r} \\ \dots \\ \max_{r=1,\dots,R_K} (W^{(\ell)T}[A_\sigma^{(\ell)}]_{K,r})^T x + [A_\sigma^{(\ell)}]_{K,r}^T b_{\boldsymbol{W}}^{(\ell)} + [b_\sigma^{(\ell)}]_{K,r} \end{bmatrix} \\
&= S[A, b](x)
\end{aligned}
$$

with $[A^{(\ell)}]_{k,r,\cdot} = W^{(\ell)T}[A_\sigma^{(\ell)}]_{k,r,\cdot}$ and $[b^{(\ell)}]_{k,r} = [b_\sigma^{(\ell)}]_{k,r} + b_{\boldsymbol{W}}^T[A_\sigma^{(\ell)}]_{k,r,\cdot}$.

Linear skip-connection: Any MASO with a skip-connection can be computed via definition of a new reparametrized MASO with

$$
\begin{aligned}
S\Big[A^{(\ell)}, b^{(\ell)}\Big](x) + x =& \begin{bmatrix} \max_{r=1,\dots,R_1}[A^{(\ell)}]_{1,r,\cdot}^T x + [b^{(\ell)}]_{1,r} \\ \dots \\ \max_{r=1,\dots,R_K}[A^{(\ell)}]_{K,r,\cdot}^T x + [b^{(\ell)}]_{K,r} \end{bmatrix} + x \\
=& \begin{bmatrix} \max_{r=1,\dots,R_1}([A^{(\ell)}]_{1,r,\cdot} + \boldsymbol{e}_1)^T x + [b^{(\ell)}]_{1,r} \\ \dots \\ \max_{r=1,\dots,R_K}([A^{(\ell)}]_{K,r,\cdot} + \boldsymbol{e}_D)^T x + [b^{(\ell)}]_{K,r} \end{bmatrix} \\
=& S[A', b'](x)
\end{aligned}
$$

with $[A']_{k,r,\cdot} = [A^{(\ell)}]_{k,r,\cdot} + \boldsymbol{e}_r$ and $[b']_{k,r} = [b^{(\ell)}]_{k,r} + b^{(\ell)T}[A^{(\ell)}]_{k,r,\cdot}$.

ResNet layer: Any ResNet layer can be computed via definition of a new reparametrized MASO with

$$
\begin{aligned}
S\Big[A_\sigma^{(\ell)}, b_\sigma^{(\ell)}\Big](\boldsymbol{C}x + b_{\boldsymbol{C}}) + \boldsymbol{C}_{\text{skip}}x + b_{\text{skip}} =& \begin{bmatrix} [\boldsymbol{C}_{\text{skip}}]_{1,\cdot}^T.x + [b_{\text{skip}}]_1 + \max_{r=1,\dots,R_1}[A_\sigma^{(\ell)}]_{1,r,\cdot}^T(\boldsymbol{C}x + b_{\boldsymbol{C}}) + [b_\sigma^{(\ell)}]_{1,r} \\ \dots \\ [\boldsymbol{C}_{\text{skip}}]_{K,\cdot}^T.x + [b_{\text{skip}}]_K + \max_{r=1,\dots,R_K}[A_\sigma^{(\ell)}]_{K,r,\cdot}^T(\boldsymbol{C}x + b_{\boldsymbol{C}}) + [b_\sigma^{(\ell)}]_{K,r} \end{bmatrix} \\
=& \begin{bmatrix} \max_{r=1,\dots,R_1}(\boldsymbol{C}^T[A_\sigma^{(\ell)}]_{1,r,\cdot} + [\boldsymbol{C}_{\text{skip}}]_{1,\cdot})^T x + [A_\sigma^{(\ell)}]_{1,r,\cdot}^T.b_{\boldsymbol{C}} + [b_\sigma^{(\ell)}]_{1,r} + [b_{\text{skip}}]_1 \\ \dots \\ \max_{r=1,\dots,R_K}(\boldsymbol{C}^T[A_\sigma^{(\ell)}]_{K,r,\cdot} + [\boldsymbol{C}_{\text{skip}}]_{K,\cdot})^T x + [A_\sigma^{(\ell)}]_{K,r,\cdot}^T.b_{\boldsymbol{C}} + [b_\sigma^{(\ell)}]_{K,r} + [b_{\text{skip}}]_K \end{bmatrix} \\
=& S\Big[A^{(\ell)}, b^{(\ell)}\Big](x)
\end{aligned}
$$

with $[A^{(\ell)}]_{k,r,\cdot} = \boldsymbol{C}^T[A_\sigma^{(\ell)}]_{k,r,\cdot} + [\boldsymbol{C}_{\text{skip}}]_{k,\cdot}$ and $[b^{(\ell)}]_{k,r} = [A_\sigma^{(\ell)}]_{k,r,\cdot}^T.b_{\boldsymbol{C}} + [b_\sigma^{(\ell)}]_{k,r} + [b_{\text{skip}}]_k$.

C.5. Proof of Proposition 5

We aim at minimizing the cross-entropy loss function for a given input X_n belonging to class Y_n. We also have the constraint $\sum_{c=1}^C ||A[X_n]_c||^2 \leq K$. The loss function is thus convex on a convex set. It is thus sufficient to find a extremum point. We denote the augmented loss function with the Lagrange multiplier as

$$
l(A[X_n]_1, \dots, A[X_n]_C, \lambda) = -\langle A[X_n]_{Y_n}, X_n\rangle + \log\left(\sum_{c=1}^C e^{\langle A[X_n]_c, X_n\rangle}\right) - \lambda\left(\sum_{c=1}^C ||A[X_n]_c||^2 - K\right).
$$

The sufficient KKT conditions are thus

$$
\begin{aligned}
&\frac{dl}{dA[X_n]_1} = -1_{\{Y_n=1\}}x + \frac{e^{\langle A[X_n]_1, x\rangle}}{\sum_{c=1}^C e^{\langle A[X_n]_c, x\rangle}}x - 2\lambda A[X_n]_1 = \underline{0} \\
&\vdots \\
&\frac{dl}{dA[X_n]_C} = -1_{\{Y_n=1\}}x + \frac{e^{\langle A[X_n]_C, x\rangle}}{\sum_{c=1}^C e^{\langle A[X_n]_c, x\rangle}}x - 2\lambda A[X_n]_C = \underline{0} \\
&\frac{\partial l}{\partial \lambda} = K - \sum_{c=1}^C ||A[X_n]_c||^2 = 0.
\end{aligned}
$$

We first proceed by identifying λ as follows

$$
\begin{aligned}
\left.\begin{array}{l}
\frac{dl}{dA[X_n]_1}=\underline{0}\\
\vdots\\
\frac{dl}{dA[X_n]_C}=\underline{0}
\end{array}\right\} &\implies \sum_{c=1}^{C} A[X_n]_c^T \frac{dl}{dA[X_n]_c}=0\\
&\implies -\langle A[X_n]_{Y_n},x\rangle+\sum_{c=1}^{C}\frac{e^{\langle A[X_n]_c,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}\langle A[X_n]_c,x\rangle-2\lambda\sum_{c=1}^{C}||A[X_n]_c||^2=0\\
&\implies \lambda=\frac{1}{2K}\left(\sum_{c=1}^{C}\frac{e^{\langle A[X_n]_c,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}\langle A[X_n]_c,x\rangle-\langle A[X_n]_{Y_n},x\rangle\right).
\end{aligned}
$$

Now we plug λ in $\frac{dl}{dA[X_n]_k},\forall k=1,\dots,C$

$$
\begin{aligned}
\frac{dl}{dA[X_n]_k}=&-1_{\{Y_n=k\}}x+\frac{e^{\langle A[X_n]_k,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}x-2\lambda A[X_n]_k\\
=&-1_{\{Y_n=k\}}x+\frac{e^{\langle A[X_n]_k,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}x-\frac{1}{K}\sum_{c=1}^{C}\frac{e^{\langle A[X_n]_c,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}\langle A[X_n]_c,x\rangle A[X_n]_k\\
&+\frac{1}{K}\langle A[X_n]_{Y_n},x\rangle A[X_n]_k.
\end{aligned}
$$

we now leverage the fact that $A[X_n]_i=A[X_n]_j,\forall i,j\neq Y_n$ to simplify notations

$$
\begin{aligned}
\frac{dl}{dA[X_n]_k}=&\left(\frac{e^{\langle A[X_n]_k,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}-1_{\{k=Y_n\}}\right)x-\frac{C-1}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}\langle A[X_n]_i,x\rangle A[X_n]_k\\
&+\frac{1}{K}\left(1-\frac{e^{\langle A[X_n]_{Y_n},x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}\right)\langle A[X_n]_{Y_n},x\rangle A[X_n]_k\\
=&\left(\frac{e^{\langle A[X_n]_k,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}-1_{\{k=Y_n\}}\right)x-\frac{C-1}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}\langle A[X_n]_i,x\rangle A[X_n]_k\\
&+\frac{C-1}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}\langle A[X_n]_{Y_n},x\rangle A[X_n]_k\\
=&\left(\frac{e^{\langle A[X_n]_k,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}-1_{\{k=Y_n\}}\right)x+\frac{C-1}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C}e^{\langle A[X_n]_c,x\rangle}}\langle A[X_n]_{Y_n}-A[X_n]_i,x\rangle A[X_n]_k.
\end{aligned}
$$

We now proceed by using the proposed optimal solutions $A^*[X_n]_c, c=1,\dots,C$ and demonstrate that it leads to an extremum point which by nature of the problem is the global optimum. We denote by i any index different from Y_n, first case $k = Y_n$:

$$\begin{aligned}
\frac{dl}{dA[X_n]_{Y_n}} &= -\left(1 - \frac{e^{\langle A[X_n]_{Y_n},x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}}\right) x + \frac{C-1}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}}\langle A[X_n]_{Y_n} - A[X_n]_i, x\rangle A[X_n]_{Y_n} \\
&= -\frac{C-1}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}} x + \frac{C-1}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}}\langle A[X_n]_{Y_n} - A[X_n]_i, x\rangle A[X_n]_{Y_n} \\
&= -\frac{C-1}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}}\left(-x + \langle A[X_n]_{Y_n} - A[X_n]_i, x\rangle A[X_n]_{Y_n}\right) \\
&= -\frac{C-1}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}}\left(-x + \langle\sqrt{\frac{C-1}{C}K}X_n + \sqrt{\frac{K}{C(C-1)}}X_n, x\rangle\sqrt{\frac{C-1}{C}K}X_n\right) \\
&= \frac{(C-1)}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}}\left(-X_n + ||X_n||^2 X_n\right) \\
&= 0.
\end{aligned}$$

Other cases $k \neq Y_n$

$$\begin{aligned}
\frac{dl}{dA[X_n]_i} =& \frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}} x + \frac{C-1}{K}\frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}}\langle A[X_n]_{Y_n} - A[X_n]_i, x\rangle A[X_n]_i \\
=& \frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}}\left(x + \frac{C-1}{K}\langle A[X_n]_{Y_n} - A[X_n]_i, x\rangle A[X_n]_i\right) \\
=& \frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}}\left(x - \frac{C-1}{K}\langle\sqrt{\frac{C-1}{C}K}X_n + \sqrt{\frac{K}{C(C-1)}}X_n, x\rangle\sqrt{\frac{K}{C(C-1)}}X_n\right) \\
=& \frac{e^{\langle A[X_n]_i,x\rangle}}{\sum_{c=1}^{C} e^{\langle A[X_n]_c,x\rangle}}\left(X_n - ||X_n||^2 X_n\right) \\
=& 0.
\end{aligned}$$

□