Appendices

A. Existence of a solution to Problem 1 under Assumption 2

Under Assumption 2, Problem 1 is equivalent to

$$\underset{(W,b)\in C}{\text{minimize}} \;\; h(W) \tag{18}$$

with

$$C = \big\{(W,b) \in \mathbb{R}^{N\times M} \times \mathbb{R}^N \;\big|\; \max_{j\in\{1,\dots,J\}} c_j(W,b) \leqslant 0\big\}, \tag{19}$$

where the functions $(c_j)_{1\leqslant j\leqslant J}$ are defined in Eq. (11). These functions being convex, $\Phi = \max_{j\in\{1,\dots,J\}} c_j$ is convex (Bauschke & Combettes, 2019, Proposition 8.16). We deduce that $\Psi = \inf_{b\in\mathbb{R}^N} \Phi(\cdot, b)$ is also a convex function (Bauschke & Combettes, 2019, Proposition 8.35). Since $\Phi \geqslant -\eta T$, Ψ is finite valued. It is thus continuous on $\mathbb{R}^{N\times M}$ (Bauschke & Combettes, 2019, Corollary 8.40). Let us now consider the problem:

$$\underset{W\in \mathrm{lev}_{\leqslant 0}\,\Psi}{\text{minimize}} \;\; h(W) \tag{20}$$

where $\mathrm{lev}_{\leqslant 0}\,\Psi$ is the 0-lower level set of Ψ defined as

$$\mathrm{lev}_{\leqslant 0}\,\Psi = \big\{W \in \mathbb{R}^{N\times M} \;\big|\; \Psi(W) \leqslant 0\big\}, \tag{21}$$

Ψ being both convex and continuous, $\mathrm{lev}_{\leqslant 0}\,\Psi$ is closed and convex. According to Assumption 2, there exists $(\overline{W}, \overline{b}) \in \mathbb{R}^{N\times M} \times \mathbb{R}^N$ such that $h(\overline{W}) < +\infty$ and $\Phi(\overline{W}, \overline{b}) \leqslant 0$, which implies that $\Psi(\overline{W}) \leqslant 0$. This shows that $\mathrm{lev}_{\leqslant 0}\,\Psi$ has a nonempty intersection with the domain of h. By invoking now the coercivity property of h, the existence of a solution $\widehat{W}$ to Problem (20) is guaranteed by standard convex analysis results (Bauschke & Combettes, 2019, Theorem 11.10).

To show that $(\widehat{W}, \widehat{b})$ is a solution to (18), it is sufficient to show that there exists $\widehat{b} \in \mathbb{R}^N$ such that $\Phi(\widehat{W}, \widehat{b}) = \Psi(\widehat{W})$. This is equivalent to prove that there exists a solution $\widehat{b}$ to the problem:

$$\underset{b\in\mathbb{R}^N}{\text{minimize}} \;\; \Phi(\widehat{W}, b). \tag{22}$$

We know that $\Phi(\widehat{W}, \cdot)$ is a continuous function. In addition, we have assumed that there exists $(j^*, t^*) \in \{1,\dots,J\} \times \{1,\dots,T\}$ such that y_{j^*,t^*} is an interior point of $R(\mathbb{R}^N)$, which is also equal to the domain of ∂f and thus a subset of the domain of f. Since f is continuous on the interior of its domain, $\partial f(y_{j^*,t^*})$ is bounded (Bauschke & Combettes, 2019, Proposition 16.17(ii)). Then $d_{\partial f(y_{j^*,t^*})}$ is coercive, hence $c_{j^*}(\widehat{W}, \cdot)$ is coercive, and so is $\Phi(\widehat{W}, \cdot) \geqslant c_{j^*}(\widehat{W}, \cdot)$. The existence of $\widehat{b}$ thus follows from the Weierstrass theorem.

B. Results in Table 1

The results are derived from the expression of the convex function φ associated with each activation function ρ (Combettes & Pesquet, 2020a, Section 2.1) (Combettes & Pesquet, 2020b, Section 3.2).

Sigmoid

$$(\forall \zeta \in \mathbb{R}) \;\; \varphi(\zeta) = \begin{cases} (\zeta+1/2)\ln(\zeta+1/2) + (1/2-\zeta) \\ \qquad \ln(1/2-\zeta) - \dfrac{1}{2}(\zeta^2+1/4) & \text{if } \; |\zeta| < 1/2 \\ -1/4 & \text{if } \; |\zeta| = 1/2 \\ +\infty & \text{if } \; |\zeta| > 1/2. \end{cases}$$

The range of the Sigmoid function is $]-1/2, 1/2[$ and the above function is differentiable on this interval and its derivative at every $\upsilon \in]-1/2, 1/2[$ is

$$\varphi'(\upsilon) = \ln(\upsilon+1/2) - \ln(\upsilon-1/2) - \upsilon. \tag{23}$$

We deduce that, for every $\zeta \in \mathbb{R}$, $\mathrm{proj}_{\partial\varphi(\upsilon)}(\zeta) = \varphi'(\upsilon)$.

Arctangent

$$(\forall \zeta \in \mathbb{R}) \quad \varphi(\zeta) = \begin{cases} -\frac{2}{\pi}\ln\left(\cos\left(\frac{\pi\zeta}{2}\right)\right) - \frac{1}{2}\zeta^2, & \text{if } \; |\zeta| < 1 \\ +\infty, & \text{if } \; |\zeta| \geqslant 1. \end{cases} \tag{24}$$

By proceeding for this function similarly to the Sigmoid function, we have, for every $\upsilon \in \rho(\mathbb{R}) =]-1, 1[$,

$$(\forall \zeta \in \mathbb{R}) \;\; \mathrm{proj}_{\partial\varphi(\upsilon)}(\zeta) = \varphi'(\upsilon) = \tan(\pi\upsilon/2) - \upsilon. \tag{25}$$

ReLU

$$(\forall \zeta \in \mathbb{R}) \quad \varphi(\zeta) = \begin{cases} 0 & \text{if } \zeta \geqslant 0 \\ +\infty & \text{otherwise.} \end{cases} \tag{26}$$

For every $\upsilon \in \rho(\mathbb{R}) = [0, +\infty[$, we have

$$\partial\varphi(\upsilon) = \begin{cases} \{0\} & \text{if } \upsilon > 0 \\ ]-\infty, 0] & \text{if } \upsilon = 0. \end{cases} \tag{27}$$

We deduce that

$$(\forall \zeta \in \mathbb{R}) \quad \mathrm{proj}_{\partial\varphi(\upsilon)}(\zeta) = \begin{cases} 0 & \text{if } \; \upsilon > 0 \text{ or } \zeta \geqslant 0 \\ \zeta & \text{otherwise.} \end{cases} \tag{28}$$

Leaky ReLU

$$(\forall \zeta \in \mathbb{R}) \qquad \varphi(\zeta) = \begin{cases} 0, & \text{if } \zeta > 0 \\ (1/\alpha - 1)\zeta^2/2 & \text{if } \zeta \leqslant 0. \end{cases} \tag{29}$$

Since this function is differentiable on $\mathbb{R}$, for every $v \in \mathbb{R}$,

$$(\forall \zeta \in \mathbb{R}) \quad \operatorname{proj}_{\partial\varphi(v)}(\zeta) = \varphi'(v) = \begin{cases} 0 & \text{if } v > 0 \\ (1/\alpha - 1)v & \text{otherwise.} \end{cases} \tag{30}$$

Capped ReLU

$$(\forall \zeta \in \mathbb{R}) \qquad \varphi(\zeta) = \begin{cases} 0 & \text{if } \zeta \in [0, \alpha] \\ +\infty & \text{otherwise.} \end{cases} \tag{31}$$

We have thus, for every $v \in [0, \alpha]$,

$$\partial\varphi(v) = \begin{cases} \{0\} & \text{if } v \in]0, \alpha[\\ ]-\infty, 0] & \text{if } v = 0 \\ [0, +\infty[& \text{if } v = \alpha. \end{cases} \tag{32}$$

This leads to

$$(\forall \zeta \in \mathbb{R}) \quad \operatorname{proj}_{\partial\varphi(v)}(\zeta) = \begin{cases} \zeta & \text{if } (v = 0 \text{ and } \zeta < 0) \\ & \text{or } (v = \alpha \text{ and } \zeta > 0) \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

ELU

$$(\forall \zeta \in \mathbb{R}) \qquad \varphi(\zeta) = \begin{cases} 0 & \text{if } \zeta \geqslant 0; \\ (\zeta + \alpha)\ln\left(\frac{\zeta+\alpha}{\alpha}\right) - \zeta - \frac{\zeta^2}{2}, & \text{if } -\alpha < \zeta < 0 \\ \alpha - \frac{\alpha^2}{2}, & \text{if } \zeta = -\alpha \\ +\infty, & \text{if } \zeta < -\alpha. \end{cases} \tag{34}$$

This function being differentiable on $\rho(\mathbb{R}) =]-\alpha, +\infty[$, we have for every $v \in]-\alpha, +\infty[$,

$$(\forall \zeta \in \mathbb{R}) \quad \operatorname{proj}_{\partial\varphi(v)}(\zeta) = \varphi'(v) = \begin{cases} 0 & \text{if } v > 0 \\ \ln\left(\frac{v+\alpha}{\alpha}\right) - v & \text{otherwise.} \end{cases} \tag{35}$$

QuadReLU Unlike the previous ones, this function does not seem to have been investigated before. It can be seen as a surrogate to the hard swish activation function, which is not a proximal activation function. Let us define

$$(\forall \zeta \in \mathbb{R}) \qquad \varphi(\zeta) = \begin{cases} +\infty & \text{if } \zeta < 0 \\ -\frac{\zeta^2}{2} + \frac{4}{3}\sqrt{\alpha}\zeta^{3/2} - \alpha\zeta & \text{if } \zeta \in [0, \alpha] \\ \frac{\zeta^2}{2} - \alpha\zeta + \frac{\alpha^2}{3} & \text{if } \zeta > \alpha. \end{cases} \tag{36}$$

φ is a lower-semicontinuous convex function whose subdifferential is

$$(\forall v \in [0, +\infty[) \quad \partial\varphi(v) = \begin{cases}]-\infty, -\alpha] & \text{if } v = 0 \\ \{-v + 2\sqrt{\alpha v} - \alpha\} & \text{if } v \in]0, \alpha] \\ \{v - \alpha\} & \text{if } v > \alpha. \end{cases} \tag{37}$$

From the definition of the proximity operator, for every $(v, \zeta) \in \mathbb{R}^2$, we have $v = \operatorname{prox}_\varphi(\zeta)$ if and only if

$$\begin{aligned} \zeta \in v + \partial\varphi(v) \quad &\Leftrightarrow \quad \begin{cases} \zeta \in]-\infty, -\alpha] & \text{if } v = 0 \\ \zeta = 2\sqrt{\alpha v} - \alpha & \text{if } v \in]0, \alpha] \\ \zeta = 2v - \alpha & \text{if } v > \alpha. \end{cases} \\ &\Leftrightarrow \quad v = \begin{cases} 0 & \text{if } \zeta \in]-\infty, -\alpha] \\ \frac{(\zeta+\alpha)^2}{4\alpha} & \text{if } \zeta \in]-\alpha, \alpha] \\ \frac{\zeta+\alpha}{2} & \text{if } \zeta > \alpha. \end{cases} \end{aligned} \tag{38}$$

This shows that

$$\operatorname{prox}_\varphi(\zeta) = (4\alpha)^{-1}(\zeta + \alpha)\operatorname{ReLU}_{2\alpha}(\zeta + \alpha). \tag{39}$$

In addition, for every $v \in [0, +\infty[$, it follows from Eq. (37) that the projection onto $\partial f(v)$ is

$$(\forall \zeta \in \mathbb{R}) \quad \operatorname{proj}_{\partial f(v)}(\zeta) = \begin{cases} v & \text{if } v = 0 \text{ and } \zeta \leqslant -\alpha \\ -v + 2\sqrt{\alpha v} - \alpha & \text{if } v \in]0, \alpha] \\ & \text{or } (v = 0 \text{ and } \zeta > -\alpha) \\ v - \alpha & \text{if } v > \alpha. \end{cases} \tag{40}$$

C. Softmax Activation

Let C denote the closed hypercube $[0, 1]^N$, let V be the vector hyperplane defined as

$$V = \big\{z = (\zeta^{(k)})_{1 \leqslant k \leqslant N} \in \mathbb{R}^N \;\big|\; \sum_{k=1}^N \zeta^{(k)} = 0\big\}, \tag{41}$$

and let A be the affine hyperplane defined as

$$A = \big\{z = (\zeta^{(k)})_{1 \leqslant k \leqslant N} \in \mathbb{R}^N \;\big|\; \sum_{k=1}^N \zeta^{(k)} = 1\big\} = V + u, \tag{42}$$

where $u = [1, \dots, 1]^\top / N = \mathbf{1}/N \in \mathbb{R}^N$. If R is the Softmax activation operator, the convex function f such that $\operatorname{prox}_f = R$ is (Combettes & Pesquet, 2020a, Example 2.23):

$$(\forall z = (\zeta^{(k)})_{1 \leqslant k \leqslant N}) \quad f(z) = \begin{cases} \sum_{k=1}^N \varphi(\zeta^{(k)}) & \text{if } z \in C \cap A \\ +\infty & \text{otherwise,} \end{cases} \tag{43}$$

where

$$(\forall \zeta \in [0, +\infty[) \quad \varphi(\zeta) = \zeta \ln \zeta - \frac{\zeta^2}{2} \tag{44}$$

(with the convention $0 \ln 0 = 0$). The latter function is differentiable on $]0, +\infty[$. It then follows from standard subdifferential calculus rules that, for every $y = (\upsilon^{(k)})_{1 \leqslant k \leqslant N} \in]0, +\infty[^N$,

$$\partial f(y) = (\varphi'(\upsilon^{(k)}))_{1 \leqslant k \leqslant N} + \partial \iota_{C \cap A}(y), \tag{45}$$

where φ' is the derivative of φ on $]0, +\infty[$ and $\iota_{C \cap A}$ denotes the indicator function of the intersection of C and A (equal to 0 on this set and $+\infty$ elsewhere). It can be deduced from Eq. (45) that, for every $y = (\upsilon^{(k)})_{1 \leqslant k \leqslant N} \in]0, +\infty[^N$,

$$\partial f(y) = (\varphi'(\upsilon^{(k)}))_{1 \leqslant k \leqslant N} + N_C(y) + N_A(y), \tag{46}$$

where N_D denotes the normal cone to a nonempty closed convex set D, which is defined as

$$(\forall y \in D) \;\; N_D(y) = \big\{ t \in \mathbb{R}^N \;\big|\; (\forall z \in D) \; \langle t \mid z - y \rangle \leqslant 0 \big\}. \tag{47}$$

Thus, for every $y \in A$, $N_A(y) = N_V(y - u)$ is the orthogonal space $V^\perp$ of V.
Let us now assume that $y \in R(\mathbb{R}^N) =]0, 1[^N \cap A$. Then, since y is an interior point of C, $N_C(y) = \{0\}$. We then deduce from Eq. (46) that

$$\partial f(y) = Q(y) + V^\perp, \tag{48}$$

where

$$Q(y) = (\varphi'(\upsilon^{(k)}))_{1 \leqslant k \leqslant N} = (\ln \upsilon^{(k)} + 1 - \upsilon^{(k)})_{1 \leqslant k \leqslant N}. \tag{49}$$

It follows that, for every $z \in \mathbb{R}^N$,

$$\operatorname{proj}_{\partial f(y)}(z) = Q(y) + \operatorname{proj}_{V^\perp}(z - Q(y)). \tag{50}$$

By using the expression of the projection $\operatorname{proj}_V = \mathrm{Id} - \operatorname{proj}_{V^\perp}$ onto hyperplane V, we finally obtain

$$\operatorname{proj}_{\partial f(y)}(z) = Q(y) + \frac{\mathbf{1}^\top (z - Q(y))}{N} \mathbf{1}. \tag{51}$$

D. Experimental Setup

PyTorch is employed to implement our method. We use and extend SNIP and RigL code available here[2], LRR[3], GraSP[4], SynFlow[5], STR[6], and FORCE[7]. In order to manage our experiments we use Polyaxon[8] on a Kubernetes[9] cluster and use five computing nodes with eight V100 GPUs each. Floating point operations per second (FLOPs) is calculated as equal to one multiply-add accumulator using the code[10].

SIS has the following parameters: number of iterations of Algorithm 1, number of iterations of Algorithm 2, step size parameter γ in Algorithm 1, constraint bound parameter η used to control the sparsity, and relaxation parameter $\lambda_n \equiv \lambda$ of Algorithm 1. In our experiments, the maximum numbers of iterations of Algorithms 1 and 2 are set to 2000 and 1000, respectively. λ is set to 1.5 and γ is set to 0.1 for all the SIS experiments. η value depends on the network and dataset. With few experiments, we search for a good η value that gives suitable sparsity and accuracy.

VGG19 and ResNet50 on CIFAR-10/100. We train VGG19 on CIFAR-10 for 160 epochs with a batch size of 128, learning rate of 0.1 and weight decay of 5×10^{-4} applied at epochs 81 and 122. A momentum of 0.9 is used with stochastic gradient descent (SGD). We make use of 1000 images per training class when using SIS. We fine-tune the identified sparse subnetwork for 10 epochs at a learning rate of 10^{-3}. For CIFAR-100 we keep the same training hyperparameters as for CIFAR-10. When applying SIS to the dense network, we use 300 images per class from the training samples. We fine-tune the identified sparse subnetwork for 40 epochs on the training set with a learning rate of 10^{-3}. ResNet50 employs the same hyperparameters as VGG19, except the weight decay that we set to 10^{-4}. When applying SIS to train dense ResNet50, we use the same partial training set and the same hyperparameters during fine-tuning. In case of VGG19 for CIFAR-10 and CIFAR-100, we found that η values in range $(1.5, 2)$ works best for sparsity range $(90\%, 98\%)$. In case of ResNet50, η values in range $(1, 2)$ is used.

[2] https://github.com/google-research/rigl
[3] https://github.com/lottery-ticket/rewinding-iclr20-public/tree/master/vision/gpu-src/official
[4] https://github.com/alecwangcq/GraSP
[5] https://github.com/ganguli-lab/Synaptic-Flow
[6] https://github.com/RAIVNLab/STR
[7] https://github.com/naver/force
[8] https://github.com/polyaxon/polyaxon
[9] https://kubernetes.io/
[10] https://github.com/Lyken17/pytorch-OpCounter

ResNet50 on ImageNet We use the weights of ResNet50 pre-trained on ImageNet available at PyTorch hub[11]. When applying SIS to the dense pre-trained network we use 20% samples per class from the training set. We fine-tune the identified sparse subnetwork for 40 epochs on the training set with a learning rate of 10^{-4}. We use different η values in range $(0.7, 1.5)$ for sparsity range $(60\%, 90\%)$. We found that $\eta = 2.3$ achieves 96.5% sparsity.

MobileNets on ImageNet We use MobileNetV1 dense pre-trained model from here[12] and MobileNetV2 from PyTorch hub[13]. In case of MobileNetV3, we replace the hard swish activation function used in the original paper (Howard et al., 2019) with our QuadReLU function (see the last row of Table 1). We use hyperparameters provided in the original paper to train MobileNetV3. When applying SIS to the dense pre-trained MobileNets, we use 20% samples per class from the training set. We fine-tune the identified sparse subnetwork for 30 epochs on the training set with a learning rate of 10^{-4}. For MobileNets, we search η values in range $(0.6, 1.75)$ for sparsity range $(75, 90)$.

Jasper on LibriSpeech A BxR Jasper network has B blocks, each consisting of R repeating sub-blocks. Each sub-block consists of 1D-Convolution, Batch Normalization, ReLU activation, and Dropout. The kernel size of convolutions increases with depth. The network has one convolution block at the beginning and three at the end. We train a network of 13 encoding blocks and one decoding block, having 54 1D-Convolution layers on the LibriSpeech dataset. The total number of parameters in our trained network is 333 million. Jasper network is trained on train-clean-100, train-clean-360, and train-other-500 splits of the LibriSpeech dataset (Panayotov et al., 2015). The training configuration can be found here[14]. We use train-clean-100 when using SIS. We fine-tune the identified sparse sub-network on the completed training set for ten epochs with a learning rate of 10^{-4}. We use η values in range $(0.6, 1.75)$ for sparsity range $(70, 90)$.

Transformer-XL on WikiText-103 We train the Transformer-XL network (Dai et al., 2019b) on the base version of WikiText-103 (Merity et al., 2017). We use the training configuration available here[15]. We use 10% of the training set articles when using SIS. We use η values in range $(0.5, 0.75)$ for sparsity range $(40, 70)$.

N-BEATS on M4 We train the interpretable architecture network of N-BEATS on the M4 dataset. The trained network has six residual blocks. Each block consists of four fully-connected layers and two linear projection layers. With 24 fully-connected layers, this network has 14 million trainable parameters. To compare different methods, we only train a single network on a 48-hour window instead of 180 networks on different timescales. We use the training configuration available here[16]. The training set has 50K time-series samples. We use 10K training samples to generate a sparse sub-network using SIS. We use η values in range $(0.75, 1.5)$ for sparsity range $(70, 90)$.

[11] https://pytorch.org/hub/pytorch_vision_resnet/

[12] https://github.com/RAIVNLab/STR

[13] https://pytorch.org/hub/pytorch_vision_mobilenet_v2/

[14] https://github.com/NVIDIA/DeepLearningExamples/blob/master/PyTorch/SpeechRecognition/Jasper/configs/jasper10x5dr_sp_offline_specaugment.toml

[15] https://github.com/NVIDIA/DeepLearningExamples/blob/master/PyTorch/LanguageModeling/Transformer-XL/pytorch/wt103_base.yaml

[16] https://github.com/ElementAI/N-BEATS/blob/master/experiments/m4/interpretable.gin