Coarse-to-Fine Semi-Structured Pruning of Graph Convolutional Networks for Skeleton-based Recognition

Hichem Sahbi [1]

Abstract

Deep neural networks (DNNs) are nowadays witnessing a major success in solving many pattern recognition tasks including skeleton-based classification. The deployment of DNNs on edge-devices, endowed with limited time and memory resources, requires designing lightweight and efficient variants of these networks. Pruning is one of the lightweight network design techniques that operate by removing unnecessary network parts, in a structured or an unstructured manner, including individual weights, neurons or even entire channels. Nonetheless, structured and unstructured pruning methods, when applied separately, may either be inefficient or ineffective.

In this paper, we devise a novel semi-structured method that discards the downsides of structured and unstructured pruning while gathering their upsides to some extent. The proposed solution is based on a differentiable cascaded parametrization which combines (i) a band-stop mechanism that prunes weights depending on their magnitudes, (ii) a weight-sharing parametrization that prunes connections either individually or group-wise, and (iii) a gating mechanism which arbitrates between different group-wise and entry-wise pruning. All these cascaded parametrizations are built upon a common latent tensor which is trained end-to-end by minimizing a classification loss and a surrogate tensor rank regularizer. Extensive experiments, conducted on the challenging tasks of action and hand-gesture recognition, show the clear advantage of our proposed semi-structured pruning approach against both structured and unstructured pruning, when taken separately, as well as the related work.

1. Introduction

Deep neural networks (DNNs) are nowadays becoming a hotspot in machine learning with increasingly performant models used to approach eclectic pattern recognition tasks (Krizhevsky et al., 2017; He et al., 2016; 2017; Jian et al., 2020; Ronneberger et al., 2015; Jiu & Sahbi, 2017; 2019). These models are also steadily oversized and this makes their deployment on cheap devices, endowed with limited hardware resources, very challenging. In particular, hand-gesture recognition and human computer interaction tasks require fast and lightweight DNNs with high recognition performances. However, DNNs are currently showing some saturated improvement in accuracy while their computational efficiency remains a major issue. Among these DNN models, graph convolutional networks (GCNs) are deemed effective especially on non-euclidean domains including skeleton-data (Zhu et al., 2016b). Two families of GCNs exist in the literature: spectral and spatial. Spectral methods project graph signals from the input to the Fourier domain, achieve convolution, prior to back-project the convolved signals in the input domain (Kipf & Welling, 2016; Levie et al., 2018; Li et al., 2018b; Defferrard et al., 2016; Bruna et al., 2013; Henaff et al., 2015; Chung, 1997; Sahbi, 2021c; Mazari & Sahbi, 2019b). Spatial methods proceed differently by aggregating signals through neighboring nodes, using multi-head attention, prior to achieve convolutions (as inner products) on the resulting node aggregates (Gori et al., 2005; Micheli, 2009; Wu et al., 2020; Hamilton et al., 2017; Knyazev et al., 2019; Sahbi et al., 2011; Sahbi, 2021b;a). Spatial GCNs are known to be more effective compared to spectral ones. Nonetheless, with multi-head attention, spatial GCNs become oversized, computationally overwhelming, and their deployment of cheap devices requires making them lightweight and still effective (Huang et al., 2018a; Sandler et al., 2018; Howard et al., 2017; Tan & Le, 2019; Cai et al., 2019; He et al., 2018a;b; Sahbi, 2021d; 2023).

Several existing works address the issue of lightweight network design, including tensor decomposition (Howard et al., 2019), quantization (Han et al., 2015a), distillation (Hinton et al., 2015; Mirzadeh et al., 2020; Zhang et al., 2018; Ahn et al., 2019; Sahbi & Geman, 2006), neural architecture search (Li et al., 2022) and pruning (LeCun et al., 1989; Has-

[1] Sorbonne University, CNRS, LIP6, F-75005, Paris, France. Correspondence to: <hichem.sahbi@sorbonne-universite.fr>.

Accepted to the Workshop on Advancing Neural Network Training at International Conference on Machine Learning (WANT@ICML 2024).

sibi & Stork, 1992; Han et al., 2015b; Sahbi, 2022). Pruning methods are particularly effective, and their general recipe consists in removing connections in order to enable reduced storage and faster inference with a minimal impact on classification performances. One of the mainstream methods is magnitude pruning (MP) (Han et al., 2015a) which removes the smallest weight connections before retraining the pruned networks. Two categories of MP techniques exist in the literature: unstructured (Han et al., 2015b;a) and structured (Li et al., 2016; Liu et al., 2017d). Unstructured methods remove weights individually by ranking them according to the importance of their magnitudes whilst structured approaches zero-out groups of weights (belonging to entire rows, columns, filters or channels) according to the importance of their *aggregated* magnitudes. Unstructured MP results into more flexible, accurate networks, and allows reaching any fine-grained targeted pruning rate but requires dedicated hardware to actually achieve efficient computation. In contrast, structured MP offers a more practical advantage by making the resulting DNNs compatible with standard hardware for efficient computation. However, this comes at the expense of a reduced classification performance and coarse-grained pruning rates. *In order to fully exhibit the potential of these two pruning categories, a more suitable framework should gather the upsides of both structured and unstructured pruning while discarding their downsides to some extent.*

In this paper, we introduce a novel variational MP approach that leverages both structured and unstructured pruning. This method dubbed as *semi-structured* is based on a differentiable cascaded weight parametrization composed of (i) a band-stop mechanism enforcing the prior that the smallest weights should be removed, (ii) a weight-sharing that groups mask entries belonging to the same rows, columns, or channels in a given tensor, and (iii) a gating mechanism which arbitrates between different mask group assignments while maximizing the accuracy of the trained lightweight networks. We also consider a budget loss that allows implementing any targeted fine-grained pruning rate and reducing further the rank of the pruned tensors, resulting into more efficient networks while being closely accurate as shown later in experiments.

2. Related work

The following review discusses the related work in variational pruning and skeleton-based recognition, highlighting the limitations that motivate our contributions.

Variational Pruning. The general concept behind variational pruning is to learn weights and binary masks that capture the topology of pruned networks. This is achieved by minimizing a global loss that combines a classification error and a regularizer that controls the sparsity (or the cost) of the resulting networks (Liu et al., 2017d; Wen et al., 2016; Louizos et al., 2017). However, these approaches are powerless to implement any given targeted pruning rate without overtrying multiple weighting of the regularizers. Alternative methods explicitly model the network cost using ℓ_0-based criteria (Louizos et al., 2017; Pan et al., 2016) in order to minimize the discrepancy between the observed and the targeted costs. Existing solutions rely on sampling heuristics or relaxation, which promote sparsity — using different regularizers (ℓ_1/ℓ_2-based, entropy, etc.) (Gordon et al., 2018; Carreira-Perpinán & Idelbayev, 2018; Koneru & Vasudevan, 2019; Wiedemann et al., 2019) — but are powerless to implement any given targeted cost exactly and result in overpruning effects leading to disconnected subnetworks. Furthermore, most of the existing solutions, including magnitude pruning (Han et al., 2015a), decouple the training of network topology from weights, making the learning of pruned networks suboptimal. On another hand, existing pruning methods are either structured (Li et al., 2016; Liu et al., 2017d) or unstructured (Han et al., 2015b;a) so their benefit is not fully explored. In contrast to the aforementioned related work, our contribution in this paper seeks to leverage the advantage of both structured and unstructured pruning where the training of masks and weights are coupled on top of shared latent parameters.

Skeleton-based recognition. This task has gained increasing interest due to the emergence of sensors like Intel RealSense and Microsoft Kinect. Early methods for hand-gesture and action recognition used RGB (Liu & Yuan, 2018; Hu et al., 2015; Wang & Sahbi, 2013; Yuan et al., 2012; Wang & Sahbi, 2014), depth (Ohn-Bar & Trivedi, 2014; Wang et al., 2015), shape / normals (Oreifej & Liu, 2013; Rahmani & Mian, 2016; Yun et al., 2012; Ji et al., 2014; Li et al., 2015; Zanfir et al., 2013; Sahbi, 2007; Sahbi & Fleuret, 2004), and skeleton-based techniques (Wang et al., 2018). These methods were based on modeling human motions using handcrafted features (Xia et al., 2012; Yang & Tian, 2014), dynamic time warping (Vemulapalli et al., 2014), temporal information (Zhang et al., 2016; Garcia-Hernando & Kim, 2017), and temporal pyramids (Zhu et al., 2016a; Q. De Smedt & Vandeborre, 2016). However, with the resurgence of deep learning, these methods have been quickly overtaken by 2D/3D Convolutional Neural Networks (CNNs) (Feichtenhofer et al., 2016; Nunez et al., 2018a; Mazari & Sahbi, 2019a), Recurrent Neural Networks (RNNs) (Zhu et al., 2016b; Chen et al., 2017; Ke et al., 2017; Liu et al., 2017c; Liu & Yuan, 2018; Wang et al., 2016; Du et al., 2015; Wang & Wang, 2017; Liu et al., 2017b; Nunez et al., 2018b; Shahroudy et al., 2016; Zhang et al., 2017b; Lee et al., 2017; Liu et al., 2016; Maghoumi & LaViola, 2019; Zhang et al., 2017a; Liu et al., 2017a), manifold learning (Huang & Van Gool, 2017; Huang et al., 2018b; Kacem et al., 2018), attention-based networks (Liu et al., 2021; Weng et al., 2018; Hou et al., 2018; Chen et al., 2019; Song et al., 2017), and GCNs (Huang et al., 2017; Li et al., 2018a; Yan et al., 2018; Wen et al., 2019; Shi et al., 2018; Nguyen et al., 2019; Li et al., 2019; 2020). The recent emergence of GCNs, in particular, has led to their increased use in skeleton-based recognition (Li et al., 2018b). These models capture spatial and temporal attention among skeleton joints with better interpretability. However, when tasks involve relatively large input graphs, GCNs (particularly with multi-head attention) become computationally inefficient and require lightweight design techniques. In this paper, we design efficient GCNs that make skeleton-based recognition highly efficient while also being effective.

3. A Glimpse on Graph Convolutional Networks

Let $\mathcal{S} = \{\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)\}_i$ denote a collection of graphs with $\mathcal{V}_i$, $\mathcal{E}_i$ being respectively the nodes and the edges of $\mathcal{G}_i$. Each graph $\mathcal{G}_i$ (denoted for short as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$) is endowed with a signal $\{\phi(u) \in \mathbb{R}^s : \ u \in \mathcal{V}\}$ and associated with an adjacency matrix $\mathbf{A}$. GCNs aim at learning a set of C filters $\mathcal{F}$ that define convolution on n nodes of $\mathcal{G}$ (with $n = |\mathcal{V}|$) as $(\mathcal{G} \star \mathcal{F})_\mathcal{V} = f\big(\mathbf{A}\ \mathbf{U}^\top\ \mathbf{W}\big)$, here $^\top$ stands for transpose, $\mathbf{U} \in \mathbb{R}^{s\times n}$ is the graph signal, $\mathbf{W} \in \mathbb{R}^{s\times C}$ is the matrix of convolutional parameters corresponding to the C filters and $f(.)$ is a nonlinear activation applied entry-wise. In $(\mathcal{G} \star \mathcal{F})_\mathcal{V}$, the input signal $\mathbf{U}$ is projected using $\mathbf{A}$ and this provides for each node u, the aggregate set of its neighbors. Entries of $\mathbf{A}$ could be handcrafted or learned so $(\mathcal{G} \star \mathcal{F})_\mathcal{V}$ corresponds to a convolutional block with two layers; the first one aggregates signals in $\mathcal{N}(\mathcal{V})$ (sets of node neighbors) by multiplying $\mathbf{U}$ with $\mathbf{A}$ while the second layer achieves convolution by multiplying the resulting aggregates with the C filters in $\mathbf{W}$. Learning multiple adjacency (also referred to as attention) matrices (denoted as $\{\mathbf{A}^k\}_{k=1}^K$) allows us to capture different contexts and graph topologies when achieving aggregation and convolution. With multiple matrices $\{\mathbf{A}^k\}_k$ (and associated convolutional filter parameters $\{\mathbf{W}^k\}_k$), $(\mathcal{G}\star\mathcal{F})_\mathcal{V}$ is updated as $f\big(\sum_{k=1}^K \mathbf{A}^k\mathbf{U}^\top\mathbf{W}^k\big)$. Stacking aggregation and convolutional layers, with multiple matrices $\{\mathbf{A}^k\}_k$, makes GCNs accurate but heavy. We propose, in what follows, a method that makes our networks lightweight and still effective.

4. Proposed Method: Semi-Structured Magnitude Pruning

In what follows, we formally subsume a given GCN as a multi-layered neural network g_θ whose weights are defined as $\theta = \big\{\mathbf{W}^1, \dots, \mathbf{W}^L\big\}$, being L its depth, $\mathbf{W}^\ell \in \mathbb{R}^{d_{\ell-1}\times d_\ell}$ its ℓ^{th} layer weight tensor, and d_ℓ the dimension of ℓ. The output of a given layer ℓ is defined as $\phi^\ell = f_\ell({\mathbf{W}^\ell}^\top\ \phi^{\ell-1})$, $\ell \in \{2, \dots, L\}$, with f_ℓ an activation function; without a loss of generality, we omit the bias in the definition of ϕ^ℓ.

Pruning consists in zeroing-out a subset of weights in θ by multiplying $\mathbf{W}^\ell$ with a binary mask $\mathbf{M}^\ell \in \{0,1\}^{d_{\ell-1}\times d_\ell}$. The binary entries of $\mathbf{M}^\ell$ are set depending on whether the underlying layer connections are pruned, so $\phi^\ell = f_\ell((\mathbf{M}^\ell \odot \mathbf{W}^\ell)^\top\ \phi^{\ell-1})$, here $\odot$ stands for the element-wise matrix product. In our definition of semi-structured pruning, entries of the tensor $\{\mathbf{M}^\ell\}_\ell$ are set depending on the prominence and also on how the underlying connections in g_θ are grouped; pruning that removes the entire connections individually (resp. jointly) is referred to as *unstructured* (resp. *structured*) whereas pruning that removes some connections independently and others jointly is dubbed as *semi-structured.* However, such pruning (with $\{\mathbf{M}^\ell\}_\ell$) suffers from several drawbacks. In the one hand, optimizing the discrete set of variables $\{\mathbf{M}^\ell\}_\ell$ is deemed highly combinatorial and intractable especially on large networks. In the other hand, the total number of parameters $\{\mathbf{M}^\ell\}_\ell$, $\{\mathbf{W}^\ell\}_\ell$ is twice the number of connections in g_θ and this increases training complexity and may also lead to overfitting.

4.1. Semi-Structured Weight Parametrization

In order to overcome the aforementioned issues, we consider an alternative *parametrization* that allows finding both the topology of the pruned networks together with their weights, without doubling the size of the training parameters, while making magnitude pruning semi-structured and learning still effective. This parametrization corresponds to the Hadamard product involving a weight tensor and a *cascaded* function applied to the same tensor as

$$\mathbf{W}^\ell = \hat{\mathbf{W}}^\ell \odot \big[\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)\big], \tag{1}$$

being $\hat{\mathbf{W}}^\ell$ a latent tensor and $\psi(\hat{\mathbf{W}}^\ell)$ (with $\psi = \psi_3\circ\psi_2\circ\psi_1$) a continuous relaxation of $\mathbf{M}^\ell$ which enforces the prior that (i) smallest weights $\hat{\mathbf{W}}^\ell$ should be removed from the network, (ii) the underlying mask entries $\psi(\hat{\mathbf{W}}^\ell)$ are shared (across tensor rows, columns, channels, etc.) when pruning is structured, and (iii) any given mask entry in $\psi(\hat{\mathbf{W}}^\ell)$ is either unstructurally or structurally pruned. In what follows, we detail the different parametrizations used to define $\psi(\hat{\mathbf{W}}^\ell)$; unless explicitly mentioned, we omit ℓ in the definition of $\hat{\mathbf{W}}^\ell$ and we rewrite it simply as $\hat{\mathbf{W}}$.

Band-stop Parametrization (ψ_1). This parametrization ψ_1 is entry-wise applied to the tensor $\hat{\mathbf{W}}$ and enforces the prior that smallest weights should be removed from the network. In order to achieve this goal, ψ_1 must be (i) bounded in $[0, 1]$,

(ii) differentiable, (iii) symmetric, and (iv) $\psi_1(\omega) \rightsquigarrow 1$ when $|\omega|$ is sufficiently large and $\psi_1(\omega) \rightsquigarrow 0$ otherwise. The first and the fourth properties ensure that the parametrization is neither acting as a scaling factor greater than one nor changing the sign of the latent weight, and also acts as the identity for sufficiently large weights, and as a contraction factor for small ones. The second property is necessary to ensure that ψ_1 has computable gradient while the third condition guarantees that only the magnitudes of the latent weights matter. A choice, used in practice, that satisfies these four conditions is

$$\psi_1(\omega) = 2\big(1+\exp(-\sigma\omega^2)\big)^{-1} - 1, \tag{2}$$

being σ a scaling factor that controls the crispness (binarization) of mask entries in $\psi_1(\hat{\mathbf{W}})$. According to Eq. 2, σ controls the smoothness of ψ_1 around the support of the latent weights. This allows implementing an annealed (soft) thresholding function that cuts-off all the connections in a smooth and differentiable manner as training of the latent parameters evolves. The asymptotic behavior of ψ_1 — that allows selecting the topology of the pruned subnetworks — is obtained as training reaches the latest epochs, and this makes mask entries, in $\psi_1(\hat{\mathbf{W}})$, crisp and (almost) binary. This mask $\psi_1(\hat{\mathbf{W}})$ (rewritten for short as ψ_1) is used as input to the subsequent parameterizations ψ_2 and ψ_3 as shown below.

Weight-sharing Parametrization (ψ_2). This parametrization ψ_2 implements semi-structured pruning by *tying* mask entries belonging to the same rows, columns or channels in the tensor ψ_1. More precisely, each mask entry in $\psi_2(\psi_1)$ will either be (i) entry-wise evaluated (dependent only on its underlying weight), or (ii) shared through multiple latent weights belonging to the same row, column or channel of ψ_1 resulting into the following multi-head parametrization (see Fig. 1)

$$\psi_2(\psi_1) = \begin{cases} \psi_2^u(\psi_1) & = \psi_1 \\ & \quad \text{unstructured (entry-wise)} \\ \psi_2^r(\psi_1) & = \mathbf{vec}^{-1}(\mathbf{P}_r\ \mathbf{vec}(\psi_1)) \\ & \quad \text{structured (row-wise)} \\ \psi_2^c(\psi_1) & = \mathbf{vec}^{-1}(\mathbf{vec}(\psi_1)^\top\ \mathbf{P}_c) \\ & \quad \text{structured (column-wise)} \\ \psi_2^b(\psi_1) & = \mathbf{vec}^{-1}(\mathbf{P}_r\mathbf{P}_c^\top\ \mathbf{vec}(\psi_1)) \\ & \quad \text{structured (block/channel-wise),} \end{cases} \tag{3}$$

here $\mathbf{vec}$ (resp. $\mathbf{vec}^{-1}$) reshapes a matrix into a vector (resp. vice-versa), and $\mathbf{P}_r \in \{0,1\}^{(d_{\ell-1}\times d_\ell)^2}$, $\mathbf{P}_c \in \{0,1\}^{(d_{\ell-1}\times d_\ell)^2}$ are two (layer-wise fixed) adjacency matrices that model the neighborhood system across respectively the rows and the columns of ψ_1 (in other words, $[\mathbf{P}_r]_{ij,pq} \neq 0$ iff the two connections (i,j), (p,q), in a given layer, share the same neuron, i.e., $i = p$) whilst the product $\mathbf{P}_r\mathbf{P}_c^\top \in \{0,1\}^{(d_{\ell-1}\times d_\ell)^2}$ models this neighborhood through blocks/channels of ψ_1. When composed (with ψ_1), the mask ψ_2 inherits all the aforementioned fourth properties: mask entries in $\psi_2(\psi_1)$ remain bounded in $[0,1]$, differentiable, symmetric, and close to 1 when entries of the latent tensor $\hat{\mathbf{W}}$ (i.e., inputs of ψ_1) are sufficiently large and 0 otherwise.

Gating Parametrization (ψ_3). As each connection in g_θ is endowed with a multi-head parametrization ψ_2, we define ψ_3 as a gating mechanism that selects only one of them. More precisely, each mask entry can either be (i) entry-wise pruned, i.e., untied, or (ii) tied to its row, column or block/channel. Again with ψ_3, the composed parametrization $\psi_3(\psi_2)$ is bounded in $[0,1]$, differentiable, symmetric and reaches 1 if the entries of the latent tensor $\hat{\mathbf{W}}$ are sufficiently large, and 0 otherwise. Formally, the gating mechanism ψ_3 is defined as

$$\begin{aligned}\psi_3(\psi_2) = \quad & \underbrace{\psi_2^b}_{\text{block-wise}} + \underbrace{(\bar{\psi}_2^b) \odot \psi_2^c}_{\text{column-wise}} + \underbrace{(\bar{\psi}_2^b \odot \bar{\psi}_2^c) \odot \psi_2^r}_{\text{row-wise}} \\ & + \underbrace{(\bar{\psi}_2^b \odot \bar{\psi}_2^c \odot \bar{\psi}_2^r) \odot \psi_2^u}_{\text{entry-wise}},\end{aligned} \tag{4}$$

being $\bar{\psi}_2^b = \mathbf{U} - \psi_2^b$, and $\mathbf{U}$ a tensor of ones with the same dimensions as ψ_2^b (and $\bar{\psi}_2^r$, $\bar{\psi}_2^c$, $\bar{\psi}_2^u$ are similarly defined). It is easy to see that when entries in ψ_1 (and hence ψ_2) are crisp, at most one of these four terms is activated (i.e., equal to one) for each connection in g_θ. From Eq. 4, block-wise pruning has the highest priority, followed by column-wise, row-wise and then entry-wise pruning respectively. This priority allows designing highly efficient lightweight networks with a coarse-granularity budget implementation for block/column/row-wise (structured) pruning, while entry-wise (unstructured) pruning is less computationally efficient but allows reaching any targeted budget with a finer granularity, and thereby with a better accuracy. Note that this parametrization acts as a weight regularizer which not only improves the lightweightness of the pruned networks but also their generalization performances (as shown later in experiments). Note also that ψ_1 and ψ_2 are commutable in the cascaded parameterization $\psi = \psi_3 \circ \psi_2 \circ \psi_1$ but ψ_3 should be applied at the end of the cascade.

4.2. Budget-Aware Variational Pruning

By considering Eq. 1, we define our semi-structured pruning loss as

$$\begin{aligned}&\mathcal{L}_e\big(\{\psi_3\circ\psi_2\circ\psi_1(\hat{\mathbf{W}}^\ell)\odot\hat{\mathbf{W}}^\ell\}_\ell\big) \\ &\quad +\lambda\bigg(\sum_{\ell=1}^{L-1} 1_{d_\ell}^\top\ [\psi_3\circ\psi_2\circ\psi_1(\hat{\mathbf{W}}^\ell)]\ 1_{d_{\ell+1}}\ - c\bigg)^2,\end{aligned} \tag{5}$$

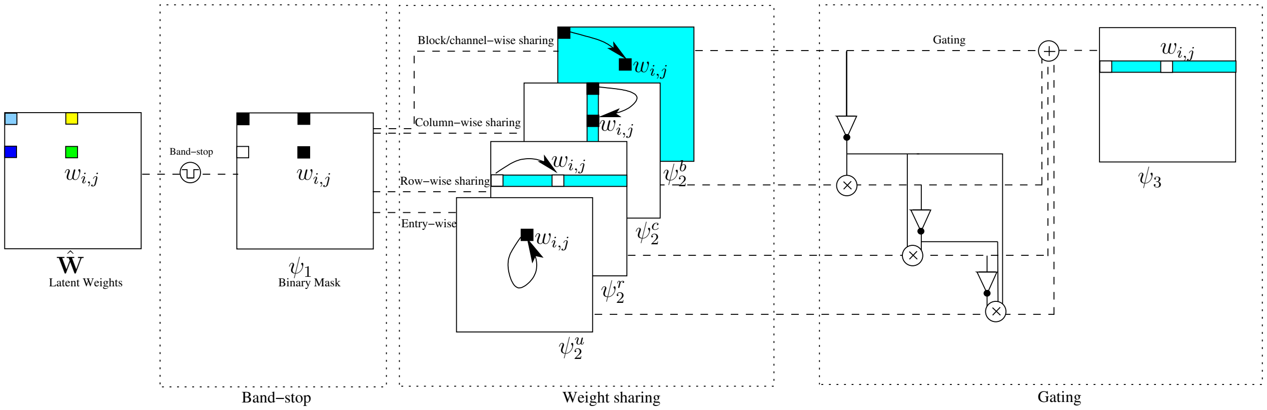

Figure 1: This figure shows the three stages of the cascaded parametrization including (i) band-stop, (ii) weight-sharing and (iii) gating. Cyan stands for shared connections, and the triangle for the "not gate" operator. For ease of visualization, only 4 connections are shown during the whole evaluation of the parameterization, and only the outcome (1 or 0) of $w_{i,j}$ is shown in the final mask tensor.

being 1_{d_ℓ} a vector of d_ℓ ones, and the left-hand side term is the cross entropy loss that measures the discrepancy between predicted and ground-truth labels. The right-hand side term is a budget loss that allows reaching any targeted pruning cost c. Nonetheless, it's worth noticing that actual efficiency is not only related to the pruning rate but also to the actual dimensionality of the tensors. In order to take full advantage of the semi-structured setting of our method, we complement the aforementioned budget function with another one that minimizes the rank of the pruned tensors $\{\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)\}_\ell$. However, as the rank is not differentiable, we consider a surrogate function (as an upper bound) of the rank[1]. Hence, Eq. 5 becomes

$$\begin{aligned}&\mathcal{L}_e\big(\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell) \odot \{\hat{\mathbf{W}}^\ell\}_\ell\big)\\ &\quad+\lambda\bigg(\sum_{\ell=1}^{L-1} 1_{d_\ell}^\top \left[\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)\right] 1_{d_{\ell+1}} - c\bigg)^2\\ &\quad+\beta \sum_{\ell=1}^{L-1} r[(\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)],\end{aligned} \tag{6}$$

being $r[\mathbf{W}]$ a surrogate differentiable rank function set in practice to

$$\begin{aligned} r[\mathbf{W}] \;\; = \;\; &\left[1_{d_{\ell+1}}^\top - \exp(-\gamma 1_{d_\ell}^\top \mathbf{W})\right] 1_{d_{\ell+1}}\\ &+1_{d_\ell}^\top\left[1_{d_\ell} - \exp(-\gamma \mathbf{W} 1_{d_{\ell+1}})\right], \end{aligned} \tag{7}$$

being γ an annealed temperature and $\exp(.)$ is entry-wise applied. Eq. 7 seeks to minimize the number of non-null rows/columns in a given tensor $\mathbf{W}$, and this allows achieving higher speedup compared to when *only* the budget loss is minimized (see experiments). In Eq. 6, β controls the "structureness" of pruning; large β favors stringent tensors first through blocks, columns and then through rows, while smaller β leads to *mixed* structured and unstructured pruning. Once the above loss optimized, actual rank minimization requires reordering dimensions layer-wise in order to fully benefit from compact tensors and eliminate fragmentation; this is achievable as only outward connections, from unpruned neurons in each layer, are actually pruned during optimization.

4.3. Optimization

Let $\mathcal{L}$ denote the global loss in Eq. 6, the update of $\{\hat{\mathbf{W}}^\ell\}_\ell$ is achieved using the gradient of $\mathcal{L}$ obtained by propagating the gradients through g_θ. More precisely, considering the parametrization in Eq. 1, the gradient of the global loss w.r.t. $\hat{\mathbf{W}}^\ell$ is obtained as

$$\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{W}}^\ell} = \frac{\partial \mathcal{L}}{\partial \psi(\hat{\mathbf{W}}^\ell)} \; \frac{\partial \psi(\hat{\mathbf{W}}^\ell)}{\partial \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)} \; \frac{\partial \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)}{\partial \psi_1(\hat{\mathbf{W}}^\ell)} \; \frac{\partial \psi_1(\hat{\mathbf{W}}^\ell)}{\partial \hat{\mathbf{W}}^\ell}, \tag{8}$$

here the original gradient $\partial\mathcal{L}/\partial\psi(\hat{\mathbf{W}}^\ell)$ is obtained from layer-wise backpropagation, and $\frac{\partial\mathcal{L}}{\partial\hat{\mathbf{W}}^\ell}$ is obtained by multiplying the original gradient by the three rightmost Jacobians (whose matrix forms are shown in Table 1).

In the above objective function, $\beta = 0.1$ and λ is overestimated (to 1000 in practice) in order to make Eq. 6 focusing on the implementation of the budget. As training reaches its final epochs, the budget loss reaches its minimum and the gradient of the global objective function will be dominated by the gradient of $\mathcal{L}_e$ (and to some extent by the gradient of the surrogate rank function); this allows improving both classification performances and efficiency as shown subsequently.

[1] This function is an upper bound of the rank as it corresponds to the sum of the number of non-zero rows and non-zero columns.

Entry-wise	Row-wise	Column-wise	Block-wise
$[\mathbf{J}_1]_{ij,pq} = 1_{\{ij=pq\}}\psi_1'(\hat{\mathbf{W}}_{pq})$	NA	NA	NA
$[\mathbf{J}_2^u]_{ij,pq} = 1_{\{ij=pq\}}$	$[\mathbf{J}_2^r]_{ij,pq} = [\mathbf{P}_r]_{ij,pq}$	$[\mathbf{J}_2^c]_{ij,pq} = [\mathbf{P}_c']_{ij,pq}$	$[\mathbf{J}_2^b]_{ij,pq} = [\mathbf{P}_r\mathbf{P}_c']_{ij,pq}$
$[\mathbf{J}_3^u]_{ij,pq} = 1_{\{ij=pq\}}$ $\times[\bar{\psi}_2^b \odot \bar{\psi}_2^c \odot \bar{\psi}_2^r]_{pq}$	$[\mathbf{J}_3^r]_{ij,pq} = 1_{\{ij=pq\}}$ $\times[\bar{\psi}_2^b \odot \bar{\psi}_2^c \odot \bar{\psi}_2^u]_{pq}$	$[\mathbf{J}_3^c]_{ij,pq} = 1_{\{ij=pq\}}$ $\times[\bar{\psi}_2^b \odot \bar{\psi}_2^r \odot \bar{\psi}_2^u]_{pq}$	$[\mathbf{J}_3^b]_{ij,pq} = 1_{\{ij=pq\}}$ $\times[\bar{\psi}_2^c \odot \bar{\psi}_2^r \odot \bar{\psi}_2^u]_{pq}$

Table 1: Jacobians of different parametrizations w.r.t. different settings; here $[\mathbf{J}_1]_{ij,pq} = [\partial\psi_1/\partial\hat{\mathbf{W}}]_{ij,pq}$, $[\mathbf{J}_2^{\mathbf{x}}]_{ij,pq} = [\partial\psi_2^{\mathbf{x}}/\partial\psi_1]_{ij,pq}$ and $[\mathbf{J}_3^{\mathbf{x}}]_{ij,pq} = [\partial\psi_3/\partial\psi_2^{\mathbf{x}}]_{ij,pq}$ with $\mathbf{x} \in \{u, r, c, b\}$; here u, r, c and b stand for entry-wise, row-wise, column-wise and block-wise respectively. It is easy to see that all these Jacobians are extremely sparse and their evaluation is highly efficient. In this table, NA stands for not applicable as the Jacobian of the parametrization ψ_1 is necessarily entry-wise.

5. Experiments

This section assesses baseline and pruned GCNs' performance in skeleton-based recognition using SBU Interaction (Yun et al., 2012) and the First Person Hand Action (FPHA) (Garcia-Hernando et al., 2018) datasets, comparing our lightweight GCNs against related pruning techniques. SBU is an interaction dataset acquired using the Microsoft Kinect sensor, it contains 282 moving skeleton sequences performed by two interacting individuals and belonging to 8 categories. Each pair of interacting individuals corresponds to two 15 joint skeletons, each one encoded with a sequence of its 3D coordinates across video frames. The evaluation protocol follows the train-test split as in the original dataset release (Yun et al., 2012). The FPHA dataset includes 1175 skeletons belonging to 45 action categories performed by 6 different individuals in 3 scenarios. Action categories are highly variable, including various styles, speed, scale, and viewpoint. Each skeleton includes 21 hand joints, each one again encoded with a sequence of its 3D coordinates across video frames. The performances of different methods are evaluated using the 1:1 setting proposed in (Garcia-Hernando et al., 2018) with 600 action sequences for training and 575 for testing. The average accuracy over all classes of actions is reported in all experiments.

Method	Accuracy (%)
Raw Position (Yun et al., 2012)	49.7
Joint feature (Ji et al., 2014)	86.9
CHARM (Li et al., 2015)	86.9
H-RNN (Du et al., 2015)	80.4
ST-LSTM (Liu et al., 2016)	88.6
Co-occurrence-LSTM (Zhu et al., 2016b)	90.4
STA-LSTM (Song et al., 2017)	91.5
ST-LSTM + Trust Gate (Liu et al., 2016)	93.3
VA-LSTM (Zhang et al., 2017a)	97.6
GCA-LSTM (Liu et al., 2017a)	94.9
Riemannian manifold. traj (Kacem et al., 2018)	93.7
DeepGRU (Maghoumi & LaViola, 2019)	95.7
RHCN + ACSC + STUFE (Li et al., 2020)	98.7
Our baseline (unpruned) GCN	98.4

Table 2: Comparison of our baseline GCN against related work on the SBU database.

Method	Color	Depth	Pose	Accuracy (%)
2-stream-color (Feichtenhofer et al., 2016)	✓	✗	✗	61.56
2-stream-flow (Feichtenhofer et al., 2016)	✓	✗	✗	69.91
2-stream-all (Feichtenhofer et al., 2016)	✓	✗	✗	75.30
HOG2-dep (Ohn-Bar & Trivedi, 2014)	✗	✓	✗	59.83
HOG2-dep+pose (Ohn-Bar & Trivedi, 2014)	✗	✓	✓	66.78
HON4D (Oreifej & Liu, 2013)	✗	✓	✗	70.61
Novel View (Rahmani & Mian, 2016)	✗	✓	✗	69.21
1-layer LSTM (Zhu et al., 2016b)	✗	✗	✓	78.73
2-layer LSTM (Zhu et al., 2016b)	✗	✗	✓	80.14
Moving Pose (Zanfir et al., 2013)	✗	✗	✓	56.34
Lie Group (Vemulapalli et al., 2014)	✗	✗	✓	82.69
HBRNN (Du et al., 2015)	✗	✗	✓	77.40
Gram Matrix (Zhang et al., 2016)	✗	✗	✓	85.39
TF (Garcia-Hernando & Kim, 2017)	✗	✗	✓	80.69
JOULE-color (Hu et al., 2015)	✓	✗	✗	66.78
JOULE-depth (Hu et al., 2015)	✗	✓	✗	60.17
JOULE-pose (Hu et al., 2015)	✗	✗	✓	74.60
JOULE-all (Hu et al., 2015)	✓	✓	✓	78.78
Huang et al. (Huang & Van Gool, 2017)	✗	✗	✓	84.35
Huang et al. (Huang et al., 2018b)	✗	✗	✓	77.57
HAN (Liu et al., 2021)	✗	✗	✓	85.74
Our baseline (unpruned) GCN	✗	✗	✓	86.43

Table 3: Comparison of our baseline GCN against related work on the FPHA database.

Input graphs. Let's consider a sequence of skeletons $\{S^t\}_t$ with $S^t = \{\hat{p}_1^t, \dots, \hat{p}_n^t\}$ being the 3D skeleton coordinates at frame t, and $\{\hat{p}_j^t\}_t$ a joint trajectory through successive frames. We define an input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as a finite collection of trajectories, with each node $v_j \in \mathcal{V}$ in $\mathcal{G}$ being a trajectory $\{\hat{p}_j^t\}_t$, and an edge $(v_j, v_i) \in \mathcal{E}$ exists between two nodes if the underlying trajectories are spatially neighbors. Each trajectory is processed using *temporal chunking*, which splits the total duration of a sequence into M evenly-sized temporal chunks (with $M = 4$ in practice). Then, joint coordinates $\{\hat{p}_j^t\}_t$ of the trajectory are assigned to these chunks, based on their time stamps. The averages of these chunks are concatenated in order to create the raw description of the trajectory (denoted as $\phi(v_j) \in \mathbb{R}^s$ with $s = 3 \times M$). This process preserves the temporal structure of trajectories while being frame-rate and duration agnostic.

Implementation details & baseline GCNs. All the GCNs have been trained using the Adam optimizer for $2,700$ epochs with a batch size of 200 for SBU and 600 for FPHA, a momentum of 0.9, and a global learning rate (denoted as $\nu(t)$) inversely proportional to the speed of change of the loss used to train the networks; with $\nu(t)$ decreasing as $\nu(t) \leftarrow \nu(t-1) \times 0.99$ (resp. increasing as $\nu(t) \leftarrow \nu(t-1)/0.99$) when the speed of change of the loss in Eq. 6 increases (resp. decreases). Experiments were run on a GeForce GTX 1070 GPU device with 8 GB memory, without dropout or data augmentation. The baseline GCN architecture for SBU includes an attention layer of 8 heads, a convolutional layer of 16 filters, a dense fully connected layer, and a softmax layer. The baseline GCN architecture for FPHA is heavier and includes 16 heads, a convolutional layer of 32 filters, a dense fully connected layer, and a softmax layer. Both the baseline GCN architectures, on the SBU and the FPHA benchmarks, are accurate (see tables. 2 and 3), and our goal is to make them lightweight while maintaining their high accuracy.

Pruning rates	Accuracy (%)	SpeedUp	Observation
0%	98.40	none	Baseline (unpruned) GCN
70%	93.84	none	Band-stop Weight Param.
90%	87.69	426×	Structured
	89.23	487×	Structured (+ rank optimization)
	93.84	none	Unstructured
	93.84	16×	Unstructured (+ rank optimization)
	90.76	40×	Semi-structured
	89.23	52×	Semi-structured (+ rank optimization)
95%	87.69	678×	Structured
	87.69	787×	Structured (+ rank optimization)
	92.30	none	Unstructured
	92.30	16×	Unstructured (+ rank optimization)
	92.30	109×	Semi-structured
	93.84	106×	Semi-structured (+ rank optimization)
98%	81.53	797×	Structured
	81.53	2195×	Structured (+ rank optimization)
	89.23	none	Unstructured
	89.23	106×	Unstructured (+ rank optimization)
	83.07	135×	Semi-structured
	86.15	607×	Semi-structured (+ rank optimization)
Comparative (regularization-based) pruning			
98%	55.38	none	MP+ℓ_0-reg.
	73.84	none	MP+ℓ_1-reg.
	61.53	none	MP+Entropy-reg.
	75.38	none	MP+Cost-aware-reg.

Table 4: This table shows detailed performances and ablation study on SBU for different pruning rates. "none" stands for no-actual speedup is observed as the underlying tensors/architecture remain shaped identically to the unpruned network (despite having pruned connections). For structured, unstructured and semi-structured settings, when "rank optimization" is not used, only pruning rate is considered in the loss together with cross entropy. When "rank optimization" is used, all the three terms are combined in the loss.

Pruning rates	Accuracy (%)	SpeedUp	Observation
0%	86.43	none	Baseline (unpruned) GCN
50%	85.56	none	Band-stop Weight Param.
90%	68.00	274×	Structured
	71.30	547×	Structured (+ rank optimization)
	83.82	none	Unstructured
	84.17	16×	Unstructured (+ rank optimization)
	78.60	33×	Semi-structured
	80.52	38×	Semi-structured (+ rank optimization)
95%	56.69	759×	Structured
	62.60	931×	Structured (+ rank optimization)
	78.78	none	Unstructured
	80.17	29×	Unstructured (+ rank optimization)
	72.17	197×	Semi-structured
	74.60	214×	Semi-structured (+ rank optimization)
98%	47.47	1479×	Structured
	49.04	1399×	Structured (+ rank optimization)
	78.08	none	Unstructured
	77.56	126×	Unstructured (+ rank optimization)
	75.13	33×	Semi-structured
	73.91	278×	Semi-structured (+ rank optimization)
Comparative (regularization-based) pruning			
98%	64.69	none	MP+ℓ_0-reg.
	70.78	none	MP+ℓ_1-reg.
	67.47	none	MP+Entropy-reg.
	69.91	none	MP+Cost-aware-reg.

Table 5: This table shows detailed performances and ablation study on FPHA for different pruning rates. "none" stands for no-actual speedup is observed as the underlying tensors/architecture remain shaped identically to the unpruned network (despite having pruned connections).

Performances, Comparison & Ablation. Tables 4-5 show a comparison and an ablation study of our method both on the SBU and the FPHA datasets. First, according to the observed results, when only the cross entropy loss is used without budget (i.e., $\lambda = \beta = 0$ in Eq. 6), performances are close to the initial heavy GCNs (particularly on FPHA), with less parameters[2] as this produces a regularization effect similar to (Wan et al., 2013). Then, when pruning is structured, the accuracy is relatively low but the speedup is important particularly for high pruning regimes. When pruning is unstructured, the accuracy reaches its highest value, but no actual speedup is observed as the architecture of the pruned networks remains unchanged (i.e., not compact). When pruning is semi-structured, we observe the best trade-off between accuracy and speedup; in other words, *coarsely* pruned parts of the network (related to entire block/column/row connections) lead to high speedup and efficient computation, whereas *finely* pruned parts (related to individual connections) lead to a better accuracy with a contained marginal impact on computation, so speedup is still globally observed with a significant amount. Extra comparison of our method against other regularizers shows a substantial gain. Indeed, our method is compared against different variational pruning with regularizers plugged in Eq. 6 (instead of our proposed budget and rank

[2] Pruning rate does not exceed 70% and no control on this rate is achievable when $\lambda = 0$.

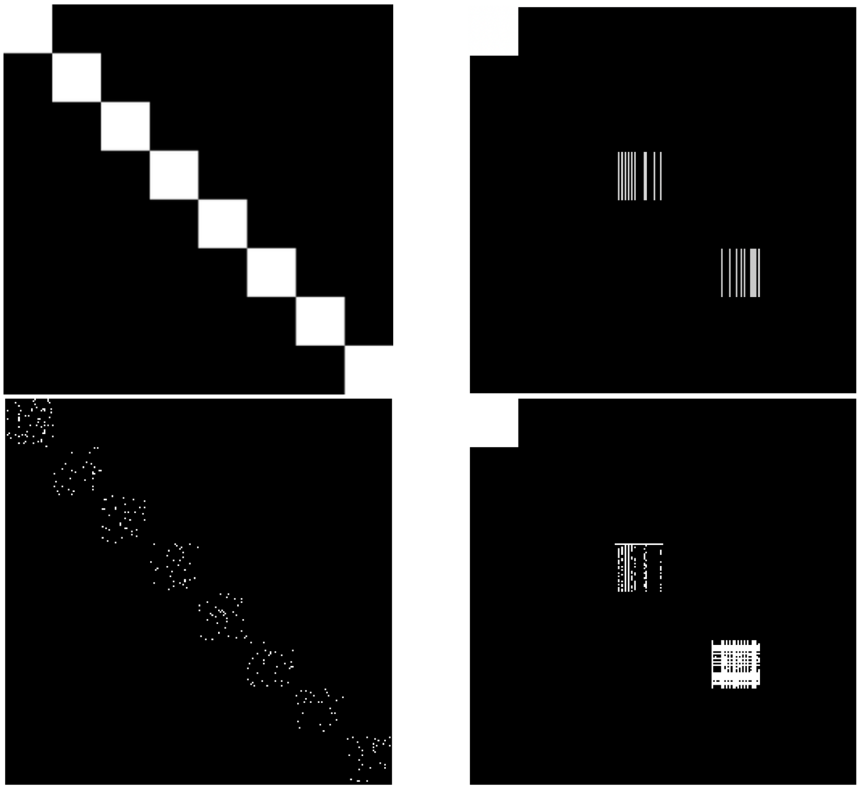

Figure 2: This figure shows a crop of the mask tensor obtained after the gating parametrization when trained on the FPHA dataset. Top-left corresponds to the original mask (without pruning) while the others correspond to masks obtained with structured, unstructured and semi-structured pruning respectively. In all these masks, each diagonal block corresponds to a channel. Better to zoom the PDF.

regularizers), namely ℓ_0 (Louizos et al., 2017), ℓ_1 (Koneru & Vasudevan, 2019), entropy (Wiedemann et al., 2019) and ℓ_2-based cost (Lemaire et al., 2019), all without our parametrization. From the observed results, the impact of our method is substantial for different settings and for equivalent pruning rate (namely 98%). Note that when alternative regularizers are used, multiple settings (trials) of the underlying mixing hyperparameters (in Eq. 6) are considered prior to reach the targeted pruning rate, and this makes the whole training and pruning process overwhelming. While cost-aware regularization makes training more affordable, its downside resides in the observed collapse of trained masks; this is a well known effect that affects performances at high pruning rates. Finally, Fig.2 shows examples of obtained mask tensors taken from the second (attention) layer of the pruned GCN. For semi-structured pruning, we observe a compact tensor layer with some individually pruned connections whereas structured and unstructured pruning — when applied separately — either produce *compact* or *spread* tensors, with a negative impact on respectively *accuracy* or *speed*. In sum, semi-structured pruning gathers the advantages of *both* while discarding their inconveniences.

6. Conclusion

This paper introduces a novel magnitude pruning approach that combines *both* the strengths of structured and unstructured pruning methods while discarding their drawbacks. The proposed method, dubbed as *semi-structured*, is based on a novel cascaded weight parametrization including band-stop, weight-sharing, and gating mechanisms. Our pruning method also relies on a budget loss that allows implementing fine-grained targeted pruning rates while also reducing the rank of the pruned tensors resulting in more efficient and still effective networks. Extensive experiments, conducted on the challenging task of skeleton-based recognition, corroborate all these findings.

References

Ahn, S., Hu, S. X., Damianou, A., Lawrence, N. D., and Dai, Z. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9163–

9171, 2019.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Cai, H., Gan, C., Wang, T., Zhang, Z., and Han, S. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

Carreira-Perpinán, M. A. and Idelbayev, Y. "learning-compression" algorithms for neural net pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8532–8541, 2018.

Chen, X., Guo, H., Wang, G., and Zhang, L. Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 2881–2885. IEEE, 2017.

Chen, Y., Zhao, L., Peng, X., Yuan, J., and Metaxas, D. N. Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. *arXiv preprint arXiv:1907.08871*, 2019.

Chung, F. R. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Du, Y., Wang, W., and Wang, L. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1110–1118, 2015.

Feichtenhofer, C., Pinz, A., and Zisserman, A. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1933–1941, 2016.

Garcia-Hernando, G. and Kim, T.-K. Transition forests: Learning discriminative temporal transitions for action recognition and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 432–440, 2017.

Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 409–419, 2018.

Gordon, A., Eban, E., Nachum, O., Chen, B., Wu, H., Yang, T.-J., and Choi, E. Morphnet: Fast & simple resource-constrained structure learning of deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1586–1595, 2018.

Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pp. 729–734. IEEE, 2005.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015a.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015b.

Hassibi, B. and Stork, D. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, 5, 1992.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

He, Y., Kang, G., Dong, X., Fu, Y., and Yang, Y. Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*, 2018a.

He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., and Han, S. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 784–800, 2018b.

Henaff, M., Bruna, J., and LeCun, Y. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hou, J., Wang, G., Chen, X., Xue, J.-H., Zhu, R., and Yang, H. Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition. In *Proceedings of* *the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Hu, J.-F., Zheng, W.-S., Lai, J., and Zhang, J. Jointly learning heterogeneous features for rgb-d activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5344–5352, 2015.

Huang, G., Liu, S., Van der Maaten, L., and Weinberger, K. Q. Condensenet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2752–2761, 2018a.

Huang, Z. and Van Gool, L. A riemannian network for spd matrix learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Huang, Z., Wan, C., Probst, T., and Van Gool, L. Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6099–6108, 2017.

Huang, Z., Wu, J., and Van Gool, L. Building deep networks on grassmann manifolds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018b.

Ji, Y., Ye, G., and Cheng, H. Interactive body part contrast mining for human interaction recognition. In *2014 IEEE international conference on multimedia and expo workshops (ICMEW)*, pp. 1–6. IEEE, 2014.

Jian, W., Zhou, Y., and Liu, H. Densely connected convolutional network optimized by genetic algorithm for fingerprint liveness detection. *IEEE Access*, 9:2229–2243, 2020.

Jiu, M. and Sahbi, H. Nonlinear deep kernel learning for image annotation. *IEEE Transactions on Image Processing*, 26(4):1820–1832, 2017.

Jiu, M. and Sahbi, H. Deep representation design from deep kernel networks. *Pattern Recognition*, 88:447–457, 2019.

Kacem, A., Daoudi, M., Amor, B. B., Berretti, S., and Alvarez-Paiva, J. C. A novel geometric framework on gram matrix trajectories for human behavior understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):1–14, 2018.

Ke, Q., Bennamoun, M., An, S., Sohel, F., and Boussaid, F. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3288–3297, 2017.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Knyazev, B., Taylor, G. W., and Amer, M. Understanding attention and generalization in graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Koneru, B. N. G. and Vasudevan, V. Sparse artificial neural networks using a novel smoothed lasso penalization. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(5):848–852, 2019.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Lee, I., Kim, D., Kang, S., and Lee, S. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1012–1020, 2017.

Lemaire, C., Achkar, A., and Jodoin, P.-M. Structured pruning of neural networks with budget-aware regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9108–9116, 2019.

Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

Li, B., Li, X., Zhang, Z., and Wu, F. Spatio-temporal graph routing for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8561–8568, 2019.

Li, C., Cui, Z., Zheng, W., Xu, C., and Yang, J. Spatio-temporal graph convolution for skeleton based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018a.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

Li, R., Wang, S., Zhu, F., and Huang, J. Adaptive graph convolutional neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018b.

Li, S., Jiang, T., Huang, T., and Tian, Y. Global co-occurrence feature learning and active coordinate system conversion for skeleton-based action recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 586–594, 2020.

Li, W., Wen, L., Chuah, M. C., and Lyu, S. Category-blind human action recognition: A practical recognition system. In *Proceedings of the IEEE international conference on computer vision*, pp. 4444–4452, 2015.

Li, Y., Zhao, P., Yuan, G., Lin, X., Wang, Y., and Chen, X. Pruning-as-search: Efficient neural architecture search via channel pruning and structural reparameterization. *arXiv preprint arXiv:2206.01198*, 2022.

Liu, J., Shahroudy, A., Xu, D., and Wang, G. Spatio-temporal lstm with trust gates for 3d human action recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pp. 816–833. Springer, 2016.

Liu, J., Wang, G., Duan, L.-Y., Abdiyeva, K., and Kot, A. C. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing*, 27(4):1586–1599, 2017a.

Liu, J., Wang, G., Hu, P., Duan, L.-Y., and Kot, A. C. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1647–1656, 2017b.

Liu, J., Wang, Y., Xiang, S., and Pan, C. Han: An efficient hierarchical self-attention network for skeleton-based gesture recognition. *arXiv preprint arXiv:2106.13391*, 2021.

Liu, M. and Yuan, J. Recognizing human actions as the evolution of pose estimation maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1159–1168, 2018.

Liu, M., Liu, H., and Chen, C. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017c.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017d.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.

Maghoumi, M. and LaViola, J. J. Deepgru: Deep gesture recognition utility. In *Advances in Visual Computing: 14th International Symposium on Visual Computing, ISVC 2019, Lake Tahoe, NV, USA, October 7–9, 2019, Proceedings, Part I 14*, pp. 16–31. Springer, 2019.

Mazari, A. and Sahbi, H. Deep temporal pyramid design for action recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2077–2081. IEEE, 2019a.

Mazari, A. and Sahbi, H. Mlgcn: Multi-laplacian graph convolutional networks for human action recognition. In *The British Machine Vision Conference (BMVC)*, 2019b.

Micheli, A. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., and Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

Nguyen, X. S., Brun, L., Lézoray, O., and Bougleux, S. A neural network based on spd manifold learning for skeleton-based hand gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12036–12045, 2019.

Nunez, J. C., Cabido, R., Pantrigo, J. J., Montemayor, A. S., and Velez, J. F. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*, 76: 80–94, 2018a.

Nunez, J. C., Cabido, R., Pantrigo, J. J., Montemayor, A. S., and Velez, J. F. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*, 76: 80–94, 2018b.

Ohn-Bar, E. and Trivedi, M. M. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014.

Oreifej, O. and Liu, Z. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 716–723, 2013.

Pan, W., Dong, H., and Guo, Y. Dropneuron: Simplifying the structure of deep neural networks. *arXiv preprint arXiv:1606.07326*, 2016.

Q. De Smedt, H. W. and Vandeborre, J.-P. Skeleton-based dynamic hand gesture recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Las Vegas, NV, United states, june, pp 1206-1214*, 2016.

Rahmani, H. and Mian, A. 3d action recognition from novel viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1506–1515, 2016.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Sahbi, H. Kernel pca for similarity invariant shape recognition. *Neurocomputing*, 70(16-18):3034–3045, 2007.

Sahbi, H. Kernel-based graph convolutional networks. In *25th International Conference on Pattern Recognition (ICPR)*, pp. 4887–4894. IEEE, 2021a.

Sahbi, H. Learning connectivity with graph convolutional networks. In *25th International Conference on Pattern Recognition (ICPR)*, pp. 9996–10003. IEEE, 2021b.

Sahbi, H. Learning laplacians in chebyshev graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2064–2075, 2021c.

Sahbi, H. Lightweight connectivity in graph convolutional networks for skeleton-based recognition. In *IEEE International Conference on Image Processing (ICIP)*, pp. 2329–2333. IEEE, 2021d.

Sahbi, H. Topologically-consistent magnitude pruning for very lightweight graph convolutional networks. In *IEEE International Conference on Image Processing (ICIP)*, pp. 3495–3499. IEEE, 2022.

Sahbi, H. Phase-field models for lightweight graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4643–4649, 2023.

Sahbi, H. and Fleuret, F. Kernel methods and scale invariance using the triangular kernel. Technical report, INRIA, 2004.

Sahbi, H. and Geman, D. A hierarchy of support vector machines for pattern detection. *Journal of Machine Learning Research*, 7(10), 2006.

Sahbi, H., Audibert, J.-Y., and Keriven, R. Context-dependent kernels for object classification. *IEEE transactions on pattern analysis and machine intelligence*, 33 (4):699–708, 2011.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1010–1019, 2016.

Shi, L., Zhang, Y., Cheng, J., and Lu, H. Non-local graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1805.07694*, 1(2):3, 2018.

Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Vemulapalli, R., Arrate, F., and Chellappa, R. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 588–595, 2014.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pp. 1058–1066. PMLR, 2013.

Wang, H. and Wang, L. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 499–508, 2017.

Wang, L. and Sahbi, H. Directed acyclic graph kernels for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3168–3175, 2013.

Wang, L. and Sahbi, H. Bags-of-daglets for action recognition. In *IEEE International Conference on Image Processing (ICIP)*, pp. 1550–1554. IEEE, 2014.

Wang, P., Li, W., Gao, Z., Zhang, J., Tang, C., and Ogunbona, P. O. Action recognition from depth maps using deep convolutional neural networks. *IEEE Transactions on Human-Machine Systems*, 46(4):498–509, 2015.

Wang, P., Li, Z., Hou, Y., and Li, W. Action recognition based on joint trajectory maps using convolutional neural networks. In *Proceedings of the 24th ACM international conference on Multimedia*, pp. 102–106, 2016.

Wang, P., Li, W., Ogunbona, P., Wan, J., and Escalera, S. Rgb-d-based human motion recognition with deep learning: A survey. *Computer Vision and Image Understanding*, 171:118–139, 2018.

Wen, W., Wu, C., Wang, Y., Chen, Y., and Li, H. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Wen, Y.-H., Gao, L., Fu, H., Zhang, F.-L., and Xia, S. Graph cnns with motif and variable temporal block for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 8989–8996, 2019.

Weng, J., Liu, M., Jiang, X., and Yuan, J. Deformable pose traversal convolution for 3d action and gesture recognition. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 136–152, 2018.

Wiedemann, S., Marban, A., Müller, K.-R., and Samek, W. Entropy-constrained training of deep neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2019.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Xia, L., Chen, C.-C., and Aggarwal, J. K. View invariant human action recognition using histograms of 3d joints. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pp. 20–27. IEEE, 2012.

Yan, S., Xiong, Y., and Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Yang, X. and Tian, Y. Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, 25(1):2–11, 2014.

Yuan, F., Xia, G.-S., Sahbi, H., and Prinet, V. Mid-level features and spatio-temporal context for activity recognition. *Pattern Recognition*, 45(12):4182–4191, 2012.

Yun, K., Honorio, J., Chattopadhyay, D., Berg, T. L., and Samaras, D. Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pp. 28–35. IEEE, 2012.

Zanfir, M., Leordeanu, M., and Sminchisescu, C. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2752–2759, 2013.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., and Zheng, N. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE international conference on computer vision*, pp. 2117–2126, 2017a.

Zhang, S., Liu, X., and Xiao, J. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 148–157. IEEE, 2017b.

Zhang, X., Wang, Y., Gou, M., Sznaier, M., and Camps, O. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4498–4507, 2016.

Zhang, Y., Xiang, T., Hospedales, T. M., and Lu, H. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4320–4328, 2018.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016a.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016b.