Kernel Representation for Dynamic Networks

Yichen ZHOU Teck Khim NG
National University of Singapore

Introduction & Motivation

Dynamic Neural Networks:

- Standard convolutions in trained neural networks apply same kernel on different input samples.
- Dynamic convolutions boosts model capability by generating sample-adaptive kernels based on input features, improving model performance.

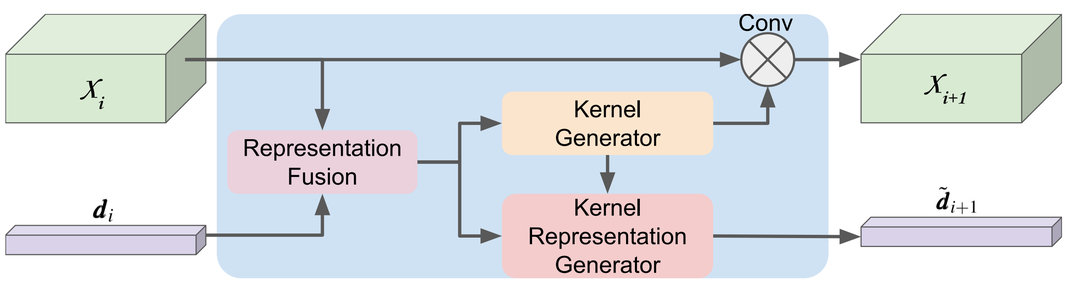
Limitations of Existing Methods and Motivation:

- Dynamic convolution methods generate kernels based solely on input features of the current layer, generation process of previous layers in the network are ignored.
- Sub-optimal kernel generation limits the representational power of dynamic networks.
- We propose Kernel Representation (KR) that maintains the information about features and kernels of prior layers to guide the dynamic kernel generation process

Kernel Representation (KR)

Preliminary on Dynamic Convolution

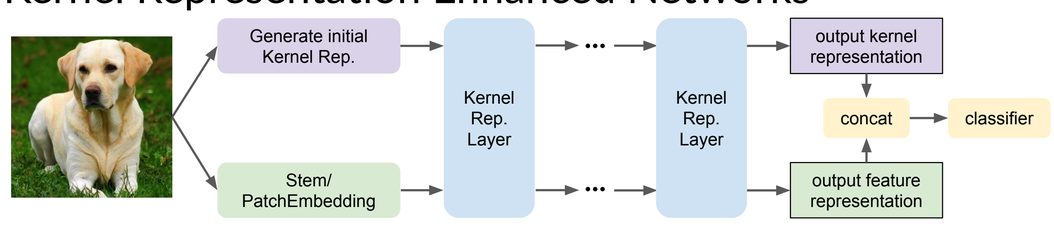
- Standard convolution: $\mathcal{X}_{i+1} = \text{Conv}(\mathcal{X}_i; \mathcal{W}_i)$, where $\mathcal{W}_i$ is kernel parameters that is fixed after training.
- Dynamic convolution generates sample-specific kernel $\mathcal{W}_i$ based on current input features $\mathcal{X}_i$: $\mathcal{W}_i = G(\mathcal{X}_i; \alpha_i)$, where α_i are parameters to generate $\mathcal{W}_i$, normally obtained with $\alpha_i = S(h_i)$, where S is a small MLP and $h_i = \text{GlobalAveragePool}(\mathcal{X}_i) \in \mathbb{R}^{c_{\text{in}}}$, is the global feature representation of $\mathcal{X}_i$.
- After training, weights in S are fixed, the only independent variable for $\mathcal{W}_i$ is the input features $\mathcal{X}_i$.



Kernel Representation (KR) Layer:

- Introduces an additional Kernel Representation vector d_i during the kernel generation process.
- **Representation Fusion (RF)**: Combines feature (h_i) and kernel (d_i) representations into $\tilde{h}_i = F(\text{cat}[d_i, h_i])$.
- **Kernel Generator (KG)**: Uses $\tilde{h}_i$ to predict parameters α_i, to be used to generate kernels $\mathcal{W}_i$. The variables for $\mathcal{W}_i$ are input features $\mathcal{X}_i$ and KR d_i.
- **Kernel Representation Generator (KRG)**: Generate output KR d_{i+1} for it to be used by the next layer with a small MLP K, i.e: $\tilde{d}_{i+1} = K(\text{cat}[\tilde{h}_i, \bar{\alpha}_i])$.
- Through a recursive process, d_{i+1} contains information on kernel generation and features from all prior layers.

Kernel Representation Enhanced Networks



- Layers are replaced Kernel Representation (KR) layers, except the first layer (stem or patch embedding).
- The initial kernel representation d_0 is projected from low-frequency components of the input image's 2D FFT, capturing high-level semantic information.
- Combines global feature activation with kernel representation for final classification, optimizing both branches together during training.
- Little computation overhead incurred by KR.

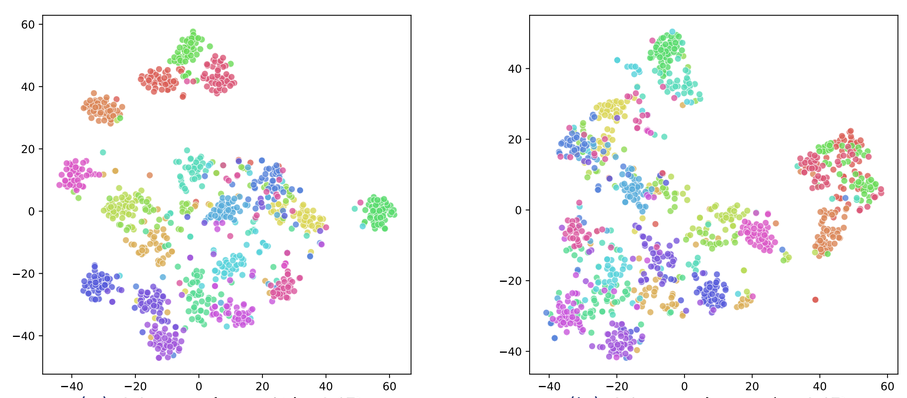
Experimental Results & Visualizations

- KR improves performance for ImageNet classification on different backbone architectures: ViT, PoolFormer, MobileNetV2 and ResNets:

Models	FLOPs	Top-1 Acc (%)
ViT-Nano	0.61G	67.5
+ODConv (4×)	0.62G	71.8 +4.3
+KR (ours) (1×)	0.61G	71.0 +3.5
+KR (ours) (4×)	0.62G	**73.8** +6.3
ViT-Tiny	1.26G	72.2
+ODConv (4×)	1.28G	75.5 +3.3
+KR (ours) (1×)	1.27G	75.8 +3.6
+KR (ours) (4×)	1.29G	**76.6** +4.4
MobileNetV2 (0.5×)	97M	64.3
+CondConv(8×)	110M	67.2 +2.9
+DyConv(4×)	103M	69.1 +4.8
+DCD	106M	69.3 +5.0
+ODConv(1×)	102M	68.3 +4.0
+ODConv(4×)	106M	70.0 +5.7
+KR (ours) (1×)	105M	68.5 +4.2
+KR (ours) (4×)	109M	**70.3** +6.0

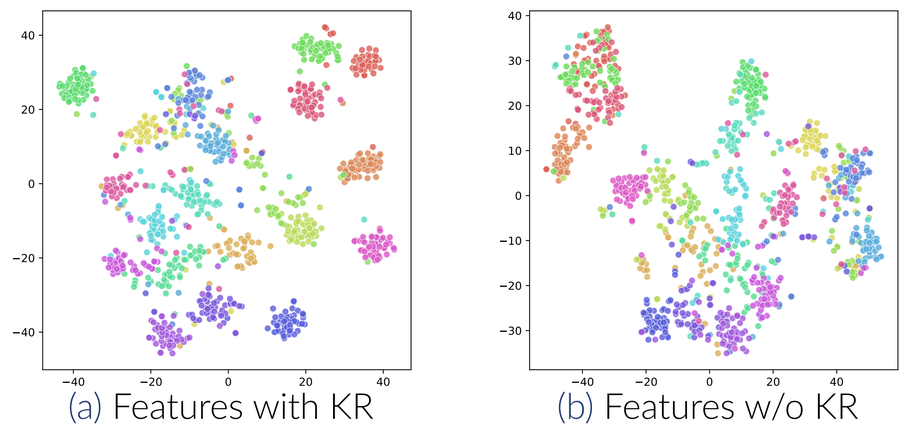
Models	FLOPs	Acc (%)
ResNet-18 slim	1.35G	66.3
ResNet-18 slim +ODConv (1×)	1.37G	69.6 +3.3
ResNet-18 slim +ODConv (4×)	1.44G	71.6 +5.3
ResNet-18 slim +KR (1×)	1.38G	70.7 +4.4
ResNet-18 slim +KR (4×)	1.45G	**72.6** +6.3
ResNet-18	1.81G	70.3
+DyConv (4×)	1.86G	72.8 +2.5
+WeightNet	1.83G	71.6 +1.3
+DCD	1.84G	72.3 +2.0
+ODConv (1×)	1.84G	73.1 +2.8
+ODConv (4×)	1.92G	73.9 +3.6
+KR (ours) (1×)	1.85G	73.3 +3.0
+KR (ours) (4×)	1.93G	**74.2** +3.9
PoolFormer-S12	1.82G	77.2
+KR (ours) (1×)	1.84G	78.2 +1.0
+KR (ours) (4×)	1.88G	**79.0** +1.8

T-SNE visualization of kernels and features on trained ViT-Nano models of validation samples from 20-classes:



(a) Kernels with KR (b) Kernels w/o KR

- Kernels generated with KR shows better clustering for samples within the same class and clearer separation for samples from different classes.



(a) Features with KR (b) Features w/o KR

- Features distribution shows similar improvement when obtained with more suitable kernels generated with KR.