Greedy Layerwise Training Can Scale to ImageNet

Eugene Belilovsky, Michael Eickenberg, Edouard Oyallon. ICML 2019.

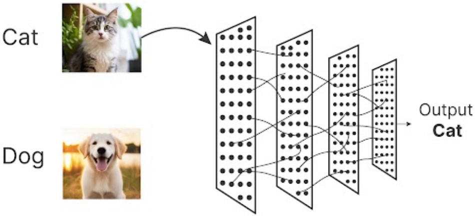
UBC MLRG

17-June-2024

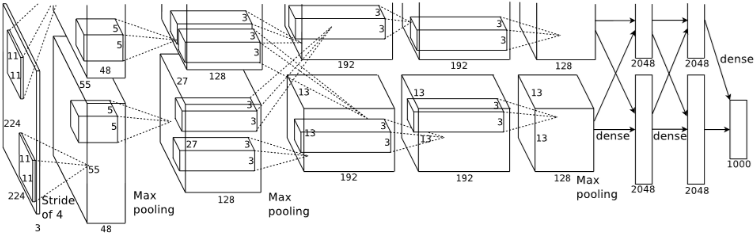
Layerwise vs end-to-end training

“do CNN layers need to be learned jointly to obtain high performance? We will show that even for the challenging ImageNet dataset the answer is no.”

Image recognition



AlexNet



5 convolutional and 3 fully-connected layers with about 62 million parameters.

Visual Geometry Group (VGG)

Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Layerwise CNN

Successively solving an auxiliary problem

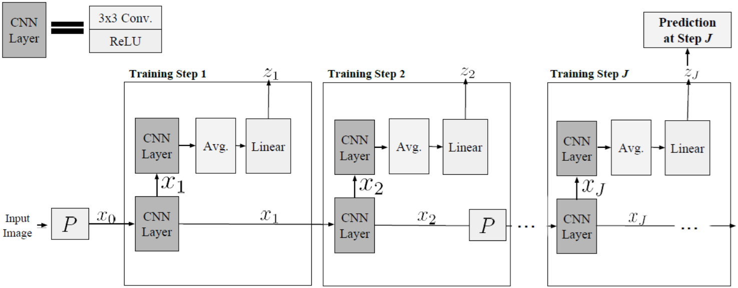


Figure 1. High level diagram of the layerwise CNN learning experimental framework using a $k = 2$-hidden layer. P, the down-sampling (see Figure 2 (Jacobsen et al., 2018)) , is applied at the input image as well as at $j = 2$.

Layerwise CNN

Transformation of sample x through some block $j = 1, \dots, J$

$$\begin{cases} x_{j+1} = \rho W_{\theta_j} P_j x_j \\ z_{j+1} = C_{\gamma_j} x_{j+1} \in \mathbb{R}^c \end{cases} \tag{1}$$

- Total number of blocks J. (e.g. $J = 4$ for CIFAR10, $J = 8$ for ImageNet)
- Classifier/ prediction z_j
- Non-linear activation ρ (e.g. ReLu)
- convolution operator W_{θ_j} with parameters θ_j (e.g. 3×3 convolution)
- Pooling operator P_j
- Auxiliary classifier C_{γ_j} with parameters γ_j to go from x_j to z_j

Layerwise CNN

CNN classifier C_{γ_j}

$$C_{\gamma_j} x_j = \begin{cases} LAx_j & k = 1 \\ LA\rho\tilde{W}_{k-2}\ldots\rho\tilde{W}_0 x_j & k > 1 \end{cases} \quad (2)$$

- convolutional layers $\tilde{W}_0, \ldots, \tilde{W}_{k-2}$
- c classes in classification problem
- k-hidden layer CNN auxiliary problem
- Linear operator L with output dimension c
- Spatial averaging operator A

Layerwise CNN

Minimize empirical risk $\hat{\mathcal{R}}$ greedily at every block.

Algorithm 1 Layer Wise CNN

Input: Training samples $\{x_0^n, y^n\}_{n \leq N}$
for $j \in 0..J-1$ **do**
  `Compute` $\{x_j^n\}_{n \leq N}$ `(via Eq.`(1)`)`
  $(\theta_j^*, \gamma_j^*) = \arg\min_{\theta_j, \gamma_j} \hat{\mathcal{R}}(z_{j+1}; \theta_j, \gamma_j)$
end for

Reason 1 for greedy layerwise training

Performance

- Computation and memory costs (does not require full gradients)
- Works just as well as end-to-end in experiments

CIFAR10 results

Layer-wise Trained	Acc. (Ens.)
SimCNN ($k = 1$ train)	88.3 (88.4)
SimCNN ($k = 2$ train)	90.4 (90.7)
SimCNN($k = 3$ train)	91.7 (**92.8**)
BoostResnet (Huang et al., 2017)	82.1
ProvableNN (Malach et al., 2018)	73.4
(Mosca et al., 2017)	81.6
Reference e2e	
AlexNet	89
VGG [1]	92.5
WRN 28-10 (Zagoruyko et al. 2016)	**96.0**
Alternatives	[Ref.]
Scattering + Linear	82.3
FeedbackAlign (Bartunov et al., 2018)	62.6 [67.6]

Table 2. Results on CIFAR-10. Compared to the few existing methods using *only* layerwise training schemes we report much more competitive results to well known benchmark models that like ours do not use skip connections.In brackets e2e trained version of the model is shown when available.

ImageNet results

	Top-1 (Ens.)	Top-5 (Ens.)
SimCNN ($k = 1$ train)	58.1 (59.3)	79.7 (80.8)
SimCNN ($k = 2$ train)	65.7 (67.1)	86.3 (87.0)
SimCNN ($k = 3$ train)	69.7 (71.6)	88.7 (89.8)
VGG-11 ($k = 3$ train)	67.6 (70.1)	88.0 (89.2)
VGG-11 (e2e train)	67.9	88.0
Alternative	[Ref.]	[Ref.]
DTargetProp (Bartunov et al., 2018)	1.6 [28.6]	5.4 [51.0]
FeedbackAlign (Xiao et al., 2019)	6.6 [50.9]	16.7 [75.0]
Scat. + Linear (Oyallon et al., 2018)	17.4	N/A
Random CNN	12.9	N/A
FV + Linear (Sánchez et al., 2013)	54.3	74.3
Reference e2e CNN		
AlexNet	56.5	79.1
VGG-13	69.9	89.3
VGG-19	72.9	90.9
Resnet-152	78.3	94.1

Table 3. Single crop validation acc. on ImageNet. Our SimCNN models use $J = 8$. In parentheses see the ensemble prediction. Layer-wise models are competitive with well known ImageNet benchmarks that similarly don't use skip connections. $k = 3$ training can yield equal performance to end to end on VGG-11. We highlight many methods and alternative training do not work at all on ImageNet. In brackets, e2e acc. is shown when available.

Reason 2 for greedy layerwise training

Greedy layerwise approach allows building theory for deep networks

- Known properties of shallow networks (especially 1-hidden layer NNs)
- Show that progressively adding shallow networks give improvements

Progressive improvement at every block

Proposition 3.1 (Progressive improvement). *Assume that* $P_j = Id$. *Then there exists* $\tilde{\theta}$ *such that:*
$\hat{\mathcal{R}}(z_{j+1}; \hat{\theta}_j, \hat{\gamma}_j) \leq \hat{\mathcal{R}}(z_{j+1}; \tilde{\theta}, \hat{\gamma}_{j-1}) = \hat{\mathcal{R}}(z_j; \hat{\theta}_{j-1}, \hat{\gamma}_{j-1})$.

Progressive Improvement

Proposition 3.2. *Assume the parameters* $\{\theta_0^*, ..., \theta_{J-1}^*\}$ *are obtained via a optimal layerwise optimization procedure. We assume that* $W_{\theta_j^*}$ *is 1-lipschitz without loss of generality and that the biases are bounded uniformly by* B. *Given an input function* $g(x)$, *we consider functions of the type* $z_g(x) = C_\gamma \rho W_\theta g(x)$. *For* $\epsilon > 0$, *we call* $\theta_{\epsilon,g}$ *the parameter provided by a procedure to minimize* $\hat{\mathcal{R}}(z_g; \theta; \gamma)$ *which leads to a 1-lipschitz operator that satisfies:*

1. $\underbrace{\|\rho W_{\theta_{\epsilon,g}} g(x) - \rho W_{\theta_{\epsilon,\tilde{g}}} \tilde{g}(x)\| \leq \|g(x) - \tilde{g}(x)\|}_{\text{(stability)}}, \forall g, \tilde{g},$

2. $\underbrace{\|W_{\theta_j^*} x_j^* - W_{\theta_{\epsilon, x_j^*}} x_j^*\| \leq \epsilon(1 + \|x_j^*\|)}_{(\epsilon\text{-approximation})},$

with, $\hat{x}_{j+1} = \rho W_{\theta_{\epsilon,\hat{x}_j}} \hat{x}_j$ *and* $x_{j+1}^* = \rho W_{\theta_j^*} x_j^*$ *with* $x_0^* = \hat{x}_0 = x$, *then, we prove by induction:*

$$\|x_J^* - \hat{x}_J\| = \mathcal{O}(J^2 \epsilon) \qquad (3)$$

Questions/ Discussion

- Is end-to-end training necessary for image classification? For other tasks?
- Is greedy layerwise training more efficient?

Models	Number of Parameters
SimCNN $k = 3$, $M_f = 512$	46M
SimCNN $k = 3$	102M
SimCNN $k = 2$	64M
SimCNN $k = 1, J = 6$	96M
AlexNet	60M
VGG-16	138 M

Table 7. Overall parameter counts for SimCNN models trained in Sec. 4 and from literature.

- Does this approach further our understanding of deep networks?
- How many blocks? How many CNN-layers per block?
- How else can we use the auxiliary classifiers?
  Ensemble used in the paper

$$Z = \sum_{j=1}^{J} 2^j z_j$$

References

- Eugene Belilovsky, Michael Eickenberg and Edouard Oyallon. *Greedy Layerwise Learning Can Scale to ImageNet*. 2019.
- Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. *ImangeNet Classification with Deep Convolutional Neural Networks*. 2012.
- Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015.

Datasets

CIFAR-10 (170MB):

- 32×32 colour images
- 50,000 training images
- 10,000 validation images
- 10 categories

ImageNet1000 (150Gb):

- colour images of various sizes
- > 1.2 million training images
- 50,0000 validation images
- 1000 categories