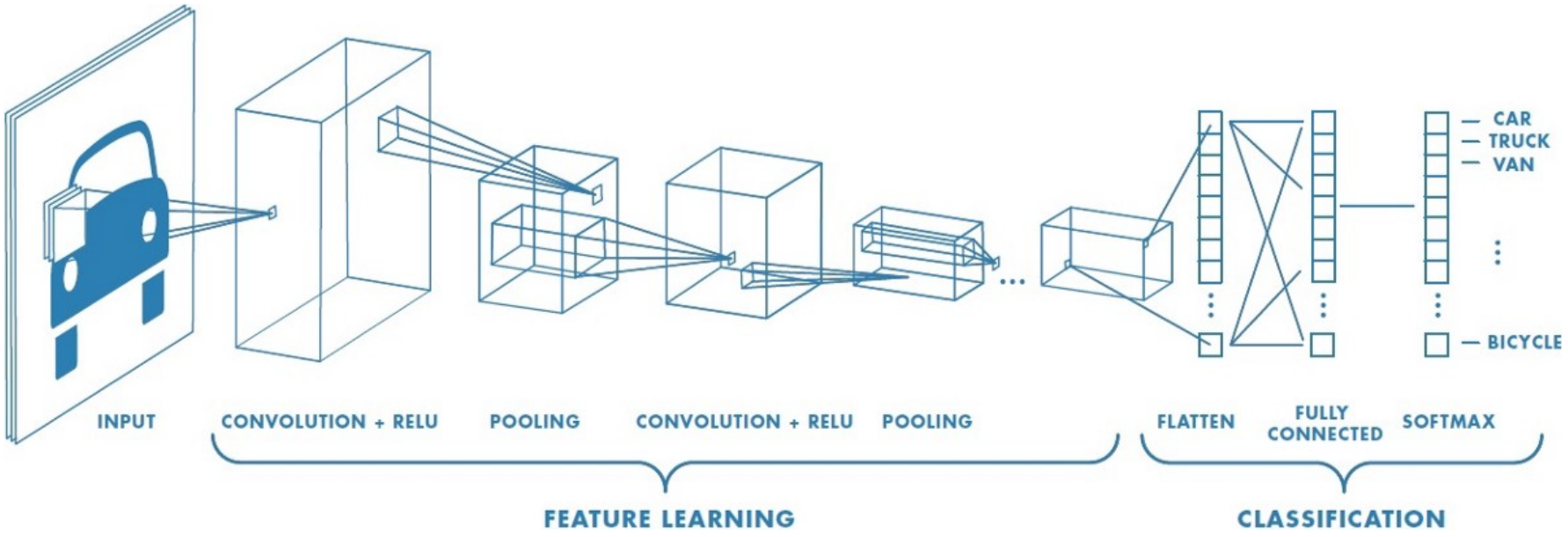
Efficiency of Convolutional Neural Networks

He Jie & Peter Racioppo



Introduction

- Convolutional neural networks (CNNs) apply convolutional filters in each layer to extract hierarchical patterns in input data.
- Local neurons act on local subsets of the input tensor. This makes CNNs well-suited to hierarchical representation of data and is biologically motivated by studies of the visual cortex.
- CNNs have sparse connectivity and shared weights, which reduces the network's computational requirements and prevents overfitting.
- Like fully-connected neural networks, CCNs are known to be universal approximators.

Background: CNNs

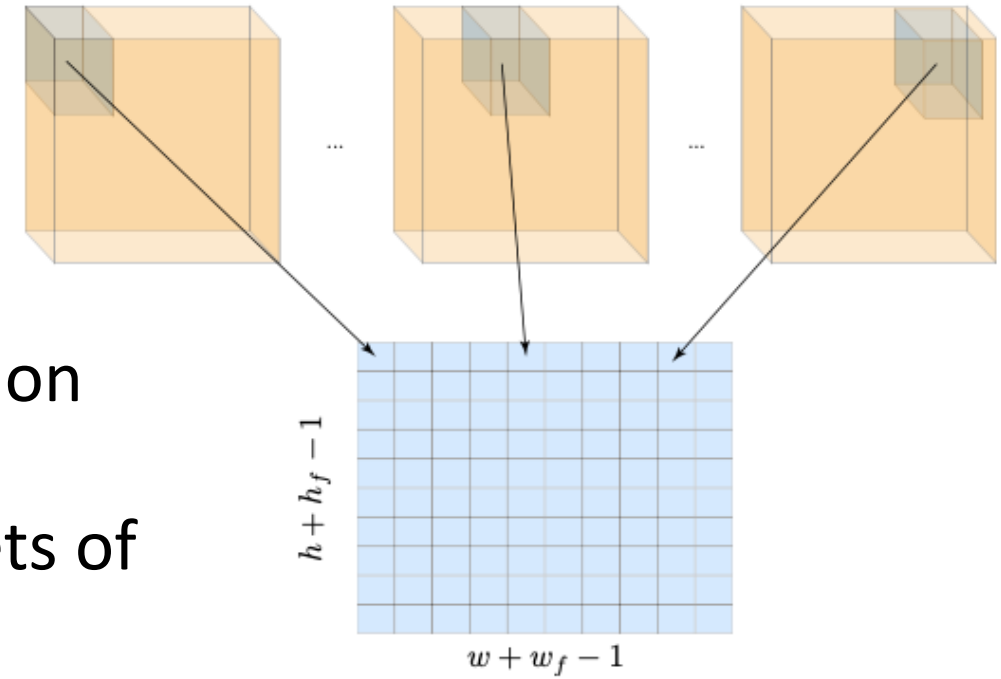
- The convolution (actually cross-correlation) of two real-valued discrete-time functions f and g is

$$f * g = \sum_{-\infty}^{\infty} f(m)g(n+m)$$

- The output of the jth layer is then

$$h^{(j)}(x) = \sigma(T^{(j)}h^{(j-1)}(x) - b^{(j)})$$

- where T is a Toeplitz matrix and σ is an activation function (often the ReLU activation $\sigma(x) = \max(0, x)$).
- "Pooling" operation often applied to subsets of the input to introduce spatial invariances.



Expressiveness of ConvNets

- (Bolcskei, et al, 2018) and (Petersen and Voigtlaender, 2018) showed that, for some function classes, sparsely connected neural networks can achieve approximation error bounds of the same order as those obtained by fully-connected networks. However, these works do not give any conditions on the patterns of sparse connections.
- Universal approximation results for CNNs were proved in (Zhou, 2018), (Yarotsky, 2018), and (Petersen and Voigtlaender, 2018).
- (Zhou, 2018): any continuous function on a compact subset of $\mathbb{R}^d$ can be approximated by a sufficiently deep CNN.
- Given the hypothesis space $H_J^{w,b}$ of CNNs of depth J, for any compact subset Ω of $\mathbb{R}^d$ and function f $(\text{continuous on } \Omega)$, there exist weight vectors w and b and $f_J^{w,b} \in H_J^{w,b}$ s.t. $\lim_{J\to\infty} \left\| f - f_J^{w,b} \right\| = 0$.

Open Questions

- CNNs are the state-of-the-art technique in image recognition and classification.
- Evidently, CNNs are well-suited to images encountered in practice (due to spatial/translational invariance, local representation, etc.).
- But limited theoretical results: For what kinds of data are CNNs superior? What, quantitively, are their advantages over dense nets?
- On which types of training data would a fully-connected network converge to a CNN?
- A promising research direction: study models of data generation which reflect the invariant or hierarchical properties of real-world data.

Deep Learning and Hierarchical Generative Models (Mossel, 2018)

- For a particular class of hierarchical generative models, deep learning is both efficient and necessary.
- Consider the class of Markov models on a directed tree $T = \langle V, E \rangle$ of degree d.
- Each vertex v has a set of labels $L(v)$ and a representation function $R : V \to [q]$ (where $[q]$ is some alphabet).
- A transition matrix determines the transition probabilities along the directed edges between any two representations $R(u)$ and $R(v)$.
- The classification problem is to infer the labels of T from the representation functions and some small subset of data with known labels.

- Data is generated from the root of the tree, cascading down to the leaves. In the simplest model, we assume each letter of $R(w)$ is copied with probability λ from its parent and is otherwise chosen at random. Each letter is sampled independently.
- More sophisticated model: “the wiring between different features at different levels is given by some known permutation that is level dependent.” (this is similar to a convolutional network)
- The authors show:
- 1.) There exists an efficient deep learning algorithm that reconstructs the tree T.
- 2.) The probability that a shallow algorithm labels a random leaf in T correctly is no better than random.

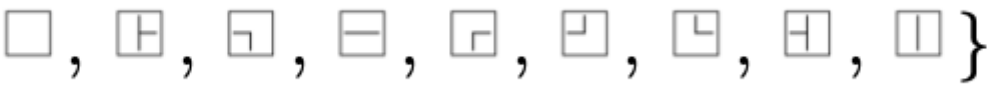
A Provably Correct Algorithm for Deep Learning that Actually Works (Malach and Shalev-Shwartz, 2018)

- Describes a method for training deep CNNs layer by layer.
- Applies gradient training for a two-layer network followed by a clustering algorithm.
- Based on a hierarchical generative model for natural images.
- Authors prove convergence of their algorithm for data produced by their generative model.
- Implementation on real-world data produces results comparable to standard CNNs.

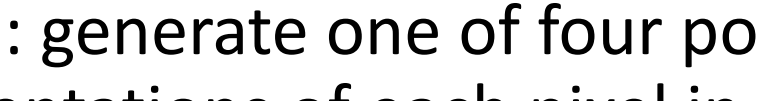
Generative Model

3-Level Synthetic Digits Example:

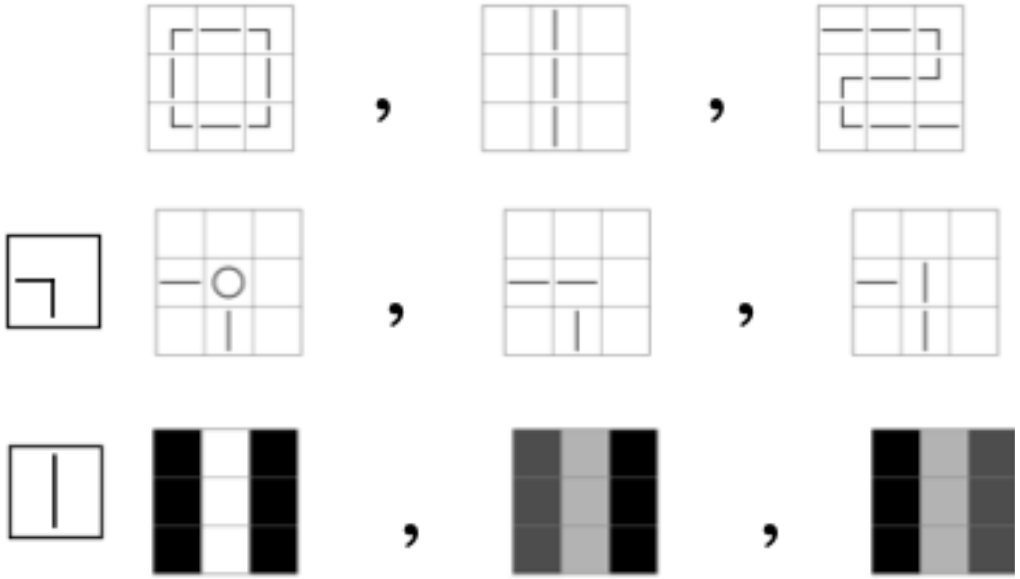
- Level 1: image “patches” are sampled over the “pixels” in C_0, which represent edges.

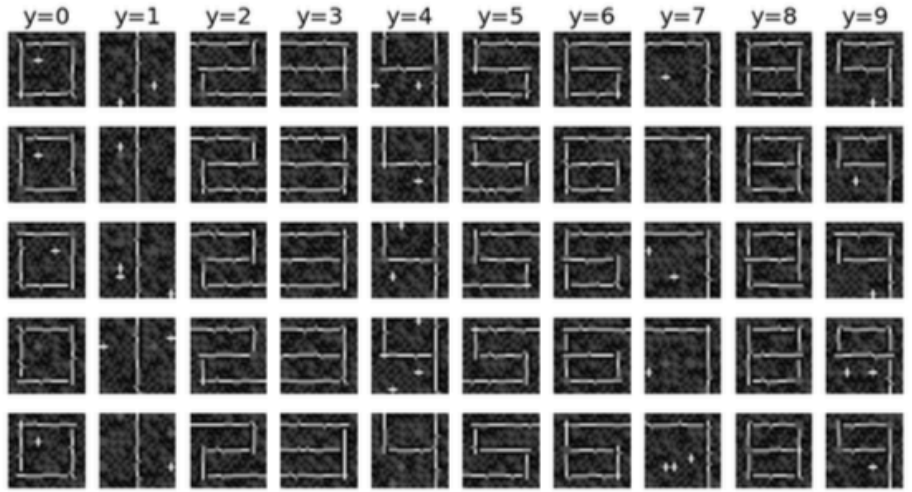
$\mathcal{C}_0 =$ 

- Level 2: generate one of four possible representations of each pixel in C_0, using the pixels in C_1.
- Level 3: generate one of four possible representations for each pixel in C_1, using the pixels in C_2.

$\mathcal{C}_1 =$  , $\mathcal{C}_2 = \mathbb{R}$

- Thus, images are generated from the top down. In general, we can iterate indefinitely.





Algorithm

- (1) Run a clustering algorithm on the input image patches.
- (2) Run a convolution operation on the cluster centroids, followed by a ReLU activation and pooling.
- (3) Run a 1x1 convolution, followed by a linear layer, for classification. Train each such two-layer network using gradient descent.
- (4) Throw away the linear layer. Use the Conv-Pool-ReLU-Conv to generate input to the next layer.
- (5) After repeating this process *k* times, feed the output of the (Conv-Pool-ReLU-Conv) x *k* network to a classifier.

Analysis

- The clustering step identifies geometrically similar regions, using e.g. the Euclidean distance between two patches.
- The CNN identifies “semantic” similarity between patches: the similarity between the distribution of occurrences of each patch across the image.
- The gradient of the output of the (first) convolutional layer depends only on the “mean image” of the semantic classes.
- It follows that the two-layer ConvNet learns an embedding of the observed patches into a space where patches from the same semantic class are close together, while patches from different classes are far apart.
- “The gradient descent algorithm maps semantically similar patches to geometrically similar vectors, allowing the clustering of the next iteration to perform clustering based again on the simple geometrical distance.”

Wavelet Scattering Networks

- Image classification algorithms must be invariant to translation and rotation, and continuous to deformation. (A deformed image should have a small distance from the original in the representation space.)
- Use of wavelets is inspired by translation-invariant representations based on Fourier transforms. (Wavelets are localized oscillations, which are stable to deformation).
- *Group Invariant Scattering* (Mallat, 2012) constructs a windowed scattering transform, a local integration of products of wavelet moduli. These wavelet-based filters are both translation-invariant and Lipschitz-continuous to diffeomorphisms.

- Wavelet Scattering Networks: Introduced in (Zhang, Benveniste, 1992): neurons are replaced by wavelet generators. These networks achieve translation-invariance by an average pooling of the wavelet modulus coefficients (modulus nonlinearity chosen for stability).
- Scattering operators are localized operations, so the values (and possibly sparsity) of network parameters are related to the geometry of the input image.
- (Bruna, Mallat, 2012): The modulus of the scattering transform is nearly zero everywhere but a subset of paths through the network. Reducing computations to these paths defines a convolution network with much fewer internal and output coefficients.

Dynamic Routing Between Capsules (Sabour, Frosst, and Hinton, 2017)

- CNNs “translate knowledge about good weight values acquired at one position in an image to other positions” but do not account for “important spatial hierarchies between simple and complex objects.”
- The authors suggest replacing the scalar-output feature detectors of CNNs with vector-outputs by groups of neurons termed “capsules.”
- The length of the activity vector is used to represent the probability of the existence of a certain object, edge, etc. in the image, while its orientation represents certain properties of the object.
- The use of vectors allows for the use of a dynamic routing algorithm.

- Max-pooling is replaced with routing-by-agreement, in which each capsule sends its output to capsules in the next layer using a softmax-like algorithm, which is input a log-likelihood proportional to the dot product of the capsule's activation vectors.
- A non-linear "squashing" function is used to shrink short vectors to near zero length, in order to encourage sparsity.
- Authors claim that CapsNets significantly improve viewpoint invariance, use fewer parameters, and improve generalization.
- A CapsNet trained on MNIST achieved state-of-the-art performance and outperforms CNNs at recognizing highly overlapping digits.

Possible Research Directions

- The methods in (Zhou, 2018) can perhaps be used to establish approximation bounds for CNNs with pooling.
- The methods in (Du, et al, 2019) for FNNs using Dynamical Systems Theory might be useful in extending the results in (Brutzkus and Globerson, 2017) to multilayer CNNs.
- How can the generative models in (Mossel, 2018) and (Malach and Shalev-Shwartz, 2018) be made more realistic?
- For example: extend the analysis in (Mossel, 2018) to random trees.

References

- H. Bolcskei, P. Grohs, G. Kutyniok, P. Petersen. "Optimal approximation with sparsely connected deep neural networks." 2018. arXiv:1705.01714v4.
- P. Petersen, V. Voigtlaender. "Optimal approximation of piecewise smooth functions using deep ReLU neural networks." 2018. arXiv:1709.05289v4.
- D.X. Zhou. "Universality of Deep Convolutional Neural Networks." arXiv:1805.10769. 28 May 2018.
- D. Yarotsky. "Universal approximations of invariant maps by neural networks." arXiv:1804.10306. 26 April 2018.
- E. Mossel. "Deep Learning and Hierarchal Generative Models." 2016. arXiv:1612.09057.
- E. Malach, S. Shalev-Shwartz. "A Provably Correct Algorithm for Deep Learning that Actually Works." School of Computer Science, The Hebrew University, Israel. arXiv:1803.09522v2 [cs.LG]. 24 June 2018.

- S. Mallat. "Group Invariant Scattering." 2011. arXiv:1101.2286.
- Q. Zhang, A. Benveniste. “Wavelet Networks,” IEEE Transactions on Neural Networks, Vol. 3, No. 6, 1992, pp. 889-898. doi:10.1109/72.165591.
- J. Bruna, S. Mallat. "Invariant Scattering Convolution Networks" 2012. arXiv:1203.1513v2.
- S. Sabour, N. Frosst, G. Hinton. "Dynamic Routing Between Capsules." 26 Oct. 2017. arXiv:1710.09829.
- S. Du, J. Lee, H. Li, L. Wang, X. Zhai. "Gradient Descent Finds Global Minima of Deep Neural Networks." 2019. arXiv:1811.03804.
- A. Brutzkus, A. Globerson. "Globally Optimal Gradient Descent for a ConvNet with Gaussian Inputs." arXiv:1702.07966. 26 Feb 2017.