Expressiviy of Overlapping Weights in CNNs Notes

Piyush Patil

September 22, 2017

1 Introduction and Background

The central idea behind this paper is to show that in convolutional layers, making use of overlapping weights between neighboring neurons is an exponential force multiplier in the expressivity of the network. Loosely speaking, expressivity is a term referring to some quantitative metric for the ability and ease with which a neural net is able to approximate a family of functions, and the richness, diversity, and generality of the largest such family.

Before moving on to considering overlapping fields and their effect on expressivity, we need t ofirst formally define expressivity. Below, a *neural network* is a function (usually between matrices and vectors) which can be expressed as the composition of several *layers*, each of which is a differentiable, parameterized function (usually between tensors). The *architecture* of a neural net refers a set of characterizing properties about the network, such as (1) the class of functions from which its layers are drawn from, (2) the length of the composition sequence, the dimensin

> **(Definition) Expressive P-Efficiency**: *Let $\mathcal{H}_1, \mathcal{H}_2$ be sets of neural nets, representing two different network architectures, where $\mathcal{H}_1$ has property P but $\mathcal{H}_2$ doesn't. Relative to some norm $|\cdot|$ defined on $\mathcal{H}_1, \mathcal{H}_2$, we say that $\mathcal{H}_1$ is **expressively P-efficient** with respect to $\mathcal{H}_2$ if the following conditions hold:*
>
> 1. $\forall f \in \mathcal{H}_2 : \exists f^* \in \mathcal{H}_1$ s.t. $|f^*| \in O(|f|)$ *and f^* approximates f to arbitrary precision.*
> 2. $\exists f \in \mathcal{H}_1$ s.t. *if $f^* \in \mathcal{H}_2$ arbitrarily approximates f then $|f^*| \in \Omega(\sigma(|f|))$ for superlinear σ. If this is true for every $f \in \mathcal{H}_1$, we say $\mathcal{H}_1$ is **completely more P-efficient** than $\mathcal{H}_2$.*

By *arbitrary approximation* we mean for any $\epsilon \in \mathbb{R}^+$ we can choose a function which differs in supremum norm from the target by less than ϵ. Essentially, the above definition simply states that if one architecture has a property P and another doesn't, we can use functions from the former to arbitrarily approximate functions from the latter, using only approximation functions that are linear in the size of the target; to go the other direction, and sacrifice the property P and try to approximate functions from the former space using functions from the latter, we can only succeed if the size of the functions needed grows superlinearly in the size of the target. Hence, any function that $\mathcal{H}_2$ can realize can also be realized with $\mathcal{H}_2$, using only a linear multiple of the number of parameters, and further the converse does not hold. Thus, it appears that having property P is crucial to the success of $\mathcal{H}_1$, which is able to easily realize functions that require superlinearly many parameters for $\mathcal{H}_2$ to realize.

The paper focuses on convolutional neural architectures, which are characterized by layers consisting of a set of n small (relative to the input dimensionality), square *kernel matrices*, whose behavior is to (1) convolve the input with each of its kernels, (2) stack the result in an order n tensor, (3) pass the tensor through a point-wise activation function, and (4) pass the activation through a downsampling function to keep dimensionality from getting out of hand as we continuously stack tensors. The *stride* of the convolution operator controls how much of the input is "skipped" as the kernel slides over. In particular, if the stride is smaller than the kernel matrix dimensionality, adjacent entries in the output matrix will depend on some (but not all) of the same input entries; hence, the output entries depend on overlapping input entries. For this reason, convolutional network architectures whose stride is smaller than the kernel dimension are said to be of *overlapping type*. The objective of the paper is to prove that architectures of the overlapping type are expressively more efficient than those of the non-overlapping type. To do so, they prove a lower bound on the complete expressive capacity of overlapping architectures.

2 Convolutional Arithmetic Circuits

The architecture in use, over which overlapping and non-overlapping variants are considered, is the *convolutional arithmetic circuit* architecture. In practice, convolutional layers consisting of rectified linear units (ReLU) as activations fo