MDPI

Article

Why Dilated Convolutional Neural Networks: A Proof of Their Optimality

Jonatan Contreras, Martine Ceberio, and Vladik Kreinovich

University of Texas at El Paso, El Paso TX 79968, USA; jmcontreras2@utep.edu, mceberio@utep.edu, vladik@utep.edu
* Correspondence: vladik@utep.edu (V.K.)

Abstract: One of the most effective image processing techniques is the use of convolutional neural networks, where we combine intensity values at grid points in the vicinity of each point. To speed up computations, researchers have developed a dilated version of this technique, in which only some points are processed. It turns out that the most efficient case is when we select points from a sub-grid. In this paper, we explain this empirical efficiency proving that the sub-grid is indeed optimal – in some reasonable sense. To be more precise, we prove that all reasonable optimality criteria, the optimal subset of the original grid is either a sub-grid, or a sub-grid-like set.

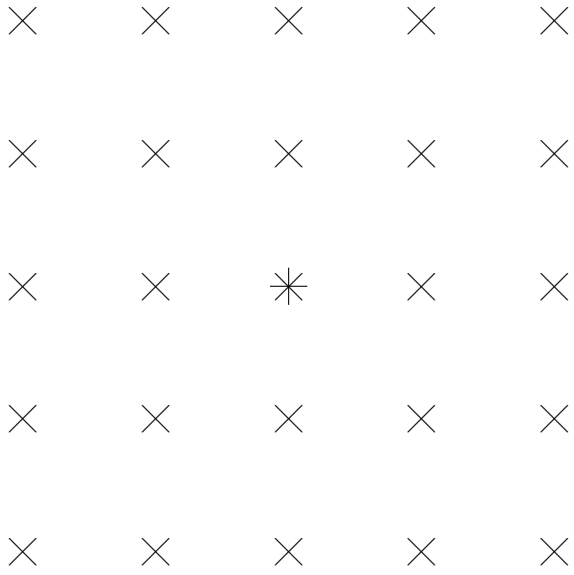
Keywords: convolutional neural networks; dilated neural networks; optimality

Citation: Contreras, J.; Ceberio, M.; Kreinovich, V. Why Dilated Convolutional Neural Networks: A Proof of Their Optimality. *Entropy* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

1. Formulation of the Problem

Convolutional neural networks: a brief reminder. At present, one of the most efficient image processing techniques is a *convolutional neural network*; see, e.g., [1]. In each step of the corresponding data processing, we combine intensity values corresponding to several neighboring points (pixels) – i.e., to the values at a rectangular grid restricted by some neighborhood of a given point:



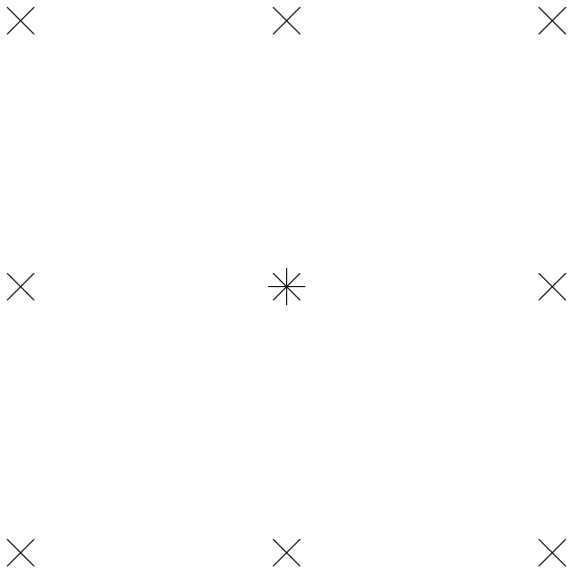
Usually, the maximum of these intensities is computed and assigned to the central point.

Dilated convolutional neural networks: main idea. One of the features of neural networks – as well as of many other machine learning tools – is that they often take

a long time to train. This is not surprising: we humans – whose brain processes are, in effect, simulated in neural networks – are not that fast to learn either. From this viewpoint, it is desirable to decrease the computational time needed for each training cycle.

With respect to convolutional neural networks, a natural speed-up idea is to take into account that the more values we process, the longer this processing takes. Thus, to speed up computations, a reasonable idea is not to process *all* the values from the grid – within the given neighborhood of a central point – but only *some* of these values. This indeed has indeed been successful. The resulting networks are known as *dilated* convolutional neural networks, since skipping some points is kind of equivalent to extending (dilating) the distance between the remaining points; see, e.g., [3,5,6].

Dilated convolutional neural networks: specifics. In principle, we could skip different points – e.g., points are at odd squared distance to the center, or points selected by some other criterion. Interestingly, the most effective results are obtained if we select a *sub-grid* – e.g., if we denote the central point by $(0,0)$, the set of all the points (x,y) for which both x and y divisible by 2:



or the set of all the points for which both x and y are divisible by 3, or by 4, etc. [5].

But why? A natural questions are:

- Why a sub-grid works the best? Why not some other subset of the original grid?
- And maybe there are other subsets that we missed which are of equal quality – or even better than the sub-grid?

What we do in this paper. In this paper, we show that the sub-grids are – in some reasonable sense – optimal. We also show that some other sets (similar to sub-grids) may also be optimal.

We will not just prove that they are optimal with respect to *one* possible optimality criterion, we will show that they are optimal with respect to *all* reasonable optimality criteria.

Comment. Similar idea were used to explain other empirically successful features of neural networks [2] and of other algorithms and heuristics [4].

2. Towards Formulating the Problem in Precise Terms

What are we looking for? We start with the original grid, i.e., with the set $\mathbb{Z} \times \mathbb{Z}$ of all the points (x,y) for which both coordinates x and y are integers: $x, y \in \mathbb{Z}$.

We are looking for a non-empty subset of this set $\mathbb{Z} \times \mathbb{Z}$.

There should be convolution. We want to make sure that there is some convolution, i.e., that this subset contains more than one point.

We need to select a family of subsets. There may be several such subsets. For example, if we select the central point $(0, 0)$, we will get a set that skips some points.

If we select one of the skipped points as a central point, we will need a different subset.

So, in general, we are looking for not for a *single* subset, but for a *family* a of subsets.

What does "optimal" mean? Out of all possible families a, we want to select an *optimal* one. What does "optimal" mean?

In many cases, "optimal" is easy to describe:

- we have an objective function $f(a)$ that assigns a numerical value to each alternative a – e.g., gain in economic situations, and
- optimal means we select an alternative for which the value of this objective function is the largest possible.

However, this is not the only possible way to describe optimality.

For example, if, in an economic situation, we are maximizing the expected gain, and there are several different alternatives with the exact same largest value of expected gain, we can use this non-uniqueness to select, e.g., the alternative with the smallest value of risk $g(a)$ – as described, e.g., by the variance. In this case, we have, in effect, a more complex preference relation between alternatives – than in the case when decision is made based only on the value of the objective function. Specifically, we say that an alterative b is better than the alternative a – and we will denote it by $a < b$ – if:

- either we have $f(a) < f(b)$,
- or we have $f(a) = f(b)$ and $g(a) > g(b)$.

If this still leaves several alternatives equally good, we can have even more complex criteria.

In general, having an optimality criterion means that we are able to compare pairs of alternatives – at least some such pairs – and conclude that:

- for some of these pairs, we have $a < b$,
- for some of these pairs, we have $b < a$, and
- for some others pairs, we conclude that alternatives a and b are, from our viewpoint, of equal value; we will denote this by $a \sim b$.

Of course, these relations must be consistent: e.g., if b is better than a, and c is better than b, then c should be better than a.

What we *must* have is some alternative which is better than or equivalent to all others – otherwise, the optimization problem has no solutions. It also makes sense to require that there is only one such optimal alternative – indeed, as we have mentioned, if there are several equally good optimal alternatives, this means that the original optimality criterion is not final, we can use this non-uniqueness to optimize something else, i.e., in effect, to modify the original criterion into a more final one.

For the same reason, for each point (x, y) on the plane, there should be exactly one set from the desired family a that contains this point. Indeed:

- if there are no such sets, it is not clear what to choose, and
- if there are several such sets, we could be able to use this non-uniqueness to optimize something else, i.e., in effect, to narrow down the family.

Thus, two different subsets from a family cannot have a common point – i.e., in mathematical terms, these sets must be *disjoint*.

In the next section, we will describe all this in precise terms.

Invariance. There is an additional natural requirement for possible optimality criteria, which is related to the fact that the original grid has lots of *symmetries*, i.e., transformations that transform this grid into itself.

For example, if we change the starting point of the coordinate system to a new point (x_0, y_0), then a point that originally had coordinates (x, y) now has coordinates $(x - x_0, y - y_0)$. It makes sense to require that the relative quality of two different families a and b will not change if we simply change the starting point.

Similarly, we can change the direction of the x-axis, then a point (x, y) becomes $(-x, y)$. If we change the direction of the y-axis, we get a transformation $(x, y) \to (x, -y)$. Finally, we can rename the coordinates: what was x will become y and vice versa; this corresponds to the transformation $(x, y) \to (y, x)$. Such transformations should also not affect the relative quality of different families.

Now, we are ready for the precise formulation of the problem.

3. Definitions and the Main Result

Definition.

- *By an* alternative, *we mean a family a of non-empty subsets of the grid $\mathbb{Z} \times \mathbb{Z}$ in which all sets are disjoint and at least one set has more than one element.*
- *By an* optimality criterion, *we mean a pair of relations $(<, \sim)$ on the set of all possible alternatives that satisfy the following conditions:*
  - *if $a < b$ and $b < c$, then $a < c$;*
  - *if $a < b$ and $b \sim c$, then $a < c$;*
  - *if $a \sim b$ and $b < c$, then $a < c$;*
  - *if $a \sim b$ and $b \sim c$, then $a \sim c$;*
  - *we have $a \sim a$ for all a; and*
  - *if $a < b$, then we cannot have $a \sim b$.*
- *We say that an alternative a is* optimal *with respect to the optimality criterion $(<, \sim)$ if for every other alternative b, we have $b < a$ or $b \sim a$.*
- *We say that the optimality criterion is* final *if there exists exactly one alternative which is optimal with respect to this criterion.*
- *By a* transformation T, *we mean one of the following transformations: $T_{x_0,y_0}(x, y) = (x - x_0, y - y_0)$, $T_x(x, y) = (-x, y)$, $T_y(x, y) = (x, -y)$, and $T_{x,y}(x, y) = (y, x)$.*
- *For each alternative a and for each transformation T, by the* result $T(a)$ *of applying the transformation T to the family a, we mean the family $T(a) = \{T(S) : S \in a\}$, where, for any set S, $T(S) \stackrel{\text{def}}{=} \{T(x, y) : (x, y) \in S\}$.*
- *We say that the optimality criterion is* invariant *if $a < b$ implies that $T(a) < T(b)$, and $a \sim b$ implies that $T(a) \sim T(b)$.*

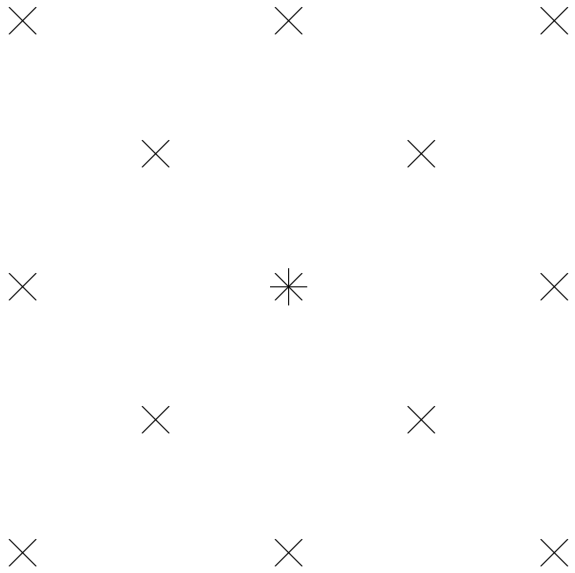
Proposition. *For every final invariant optimality criterion, the optimal alternative is equal, for some integer $k \geq 1$, to one of the following families:*

- *the family of all the sets $G_{k,x_0,y_0} \stackrel{\text{def}}{=} \{(x_0 + k \cdot n_x, y_0 + k \cdot n_y) : n_x, n_y \in \mathbb{Z}\}$ corresponding to all possible pairs of integers (x_0, y_0);*
- *the family of all the sets*

$$N_{k,x_0,y_0} \stackrel{\text{def}}{=} \{(x_0 + k \cdot n_x, y_0 + k \cdot n_y) : n_x, n_y \in \mathbb{Z} \text{ and } n_x + n_y \text{ is even}\}$$

  corresponding to all possible pairs of integers (x_0, y_0).

Comment. In the first case, we have a sub-grid – as was described in Section 1. In the second case, we have a different grid-type set:



Thus, this result explains the effectiveness of sub-grids – and also provides us with a new alternative worth trying.

Proof.

1°. Since the optimality criterion is final, there exists exactly one optimal family a_{opt}. Let us first prove that this family is itself invariant, i.e., that $T(a_{\text{opt}}) = a_{\text{opt}}$ for all transformations T.

Indeed, the fact that the family a_{opt} is optimal means that for every family a, we have $a < a_{\text{opt}}$ or $a \sim a_{\text{opt}}$. Since this is true for every family a, it is also true for every family $T^{-1}(a)$, where T^{-1} denotes inverse transformation (i.e., a transformation for which $T(T^{-1}(x,y)) = (x,y)$). Thus, for every family a, we have either $T^{-1}(a) < a_{\text{opt}}$ or $T^{-1}(a) \sim a_{\text{opt}}$. Due to invariance, we have $a = T(T^{-1}(a)) < T(a_{\text{opt}})$ or $a \sim T(a_{\text{opt}})$. By definition of optimality, this means that the alternative $T(a_{\text{opt}})$ is also optimal. However, since the optimality criterion is final, there exists exactly one optimal family, so $T(a_{\text{opt}}) = a_{\text{opt}}$. The statement is proven.

2°. By definition of an alternative, every family – including the optimal family – contains at least one set S that has at least two points. Let S be any such set, and let (x_0, y_0) be any of its points. Then, due to Part 1 of this proof, the set $S_0 \stackrel{\text{def}}{=} T_{x_0,y_0}(S)$ also belongs to the optimal family, and this set contains the point

$$T_{x_0,y_0}(x_0, y_0) = (x_0 - x_0, y_0 - y_0) = (0,0).$$

Since the set S had at least two different points, the set $S_0 = T_{x_0,y_0}(S)$ also contains at least two different points. Thus, the set S_0 must contain a point (x,y) which is different from $(0,0)$.

3°. Let us prove that if the set S_0 contains a point (x,y), then it also contains the points $(x,-y)$, $(-x,y)$, and (y,x).

Indeed, due to Part 1 of this proof, with the set S_0 the optimal family contains the set $T_y(S_0)$. This set contains the point $T_y(0,0) = (0,0)$. Thus, the sets S_0 and $T_y(S_0)$ have a common element $(0,0)$. Since different sets from the optimal family must be disjoint, it follows that the sets S_0 and $T_y(S_0)$ must coincide. The set $T_y(S_0)$ contains the points $(x,-y)$ for each point $(x,y) \in S$. Since $T_y(S_0) = S_0$, this implies that for each point $(x,y) \in S_0$, we have $(x,-y) \in T_y(S_0) = S_0$.

Similarly, we can prove that $(-x,y) \in S_0$ and $(y,x) \in S_0$.

4°. By combining the two conclusions of Part 3 – that $(x, -y) \in S_0$ and that therefore $T_x(x, -y) = (-x, -y) \in S_0$, we conclude that for every point $(x, y) \in S_0$, the point

$$-(x, y) \stackrel{\text{def}}{=} (-x, -y)$$

is also contained in the set S_0.

5°. Let us prove that if the set S_0 contains two points (x_1, y_1) and (x_2, y_2), then it also contains the point

$$(x_1 y_1) + (x_2, y_2) \stackrel{\text{def}}{=} (x_1 + x_2, y_1 + y_2).$$

Indeed, due to Part 1 of this proof, the set $T_{-x_2,-y_2}(S_0)$ also belongs to the optimal family. This set shares an element

$$T_{-x_2,-y_2}(0, 0) = (0 - (-x_2), 0 - (-y_2)) = (x_2, y_2)$$

with the original set S_0. Thus, the set $T_{-x_2,-y_2}(S_0)$ must coincide with the set S_0. Due to the fact that $(x_1, y_1) \in S_0$, the element

$$T_{-x_2,-y_2}(x_1, y_1) = (x_1 - (-x_2), y_1 - (-y_2)) = (x_1 + x_2, y_1 + y_2)$$

belongs to the set $T_{x_1,y_1}(S_0) = S_0$. The statement is proven.

6°. Let us prove that if the set S_0 contains a point (x, y), then, for each integer c, this set also contains the point

$$c \cdot (x, y) = (c \cdot x, c \cdot y).$$

Indeed, if c is positive, this follows from the fact that

$$(c \cdot x, c \cdot y) = (x, y) + \ldots + (x, y) \ (c \text{ times}).$$

When c is negative, then we first use Part 4 and conclude that $(-x, -y) \in S_0$, and then conclude that the point $(|c| \cdot (-x), |c| \cdot (-y) (= (c \cdot x, c \cdot y)$ is in the set S_0.

7°. Let us prove that if the set S_0 contains points $(x_1, y_1), \ldots, (x_n, y_n)$, then for all integers $c_1, \ldots, c_n$, it also contains their linear combination

$$c_1 \cdot (x_1, y_1) + \ldots + c_n \cdot (x_n, y_n) = (c_1 \cdot x_1 + \ldots + c_n \cdot x_n, c_1 \cdot y_1 + \ldots + c_n \cdot y_n).$$

Indeed, this follows from Parts 5 and 6.

8°. The set S_0 contains some points which are different from $(0, 0)$, i.e., points for which at least one of the integer coordinates is non-zero. According to Parts 3 and 4, we can change the signs of both x and y coordinates and still get points from S_0. Thus, we can always consider points with non-negative coordinates.

Let d denote the greatest common divisor of all positive values of the coordinates of points from S_0.

If a value x appears as an x-coordinate of some point $(x, y) \in S_0$, then, due to Part 3, we have $(x, -y) \in S_0$ and thus, due to Part 4,

$$(x_y) + (x, -y) = (2x, 0) \in S_0.$$

Similarly, if a value y appears as a y-coordinate of some point $(x, y) \in S_0$, then we get $(0, 2y) \in S_0$ and thus, due to Part 3, $(2y, 0) \in S_0$.

It is a known that a common divisor d of the values $v_1, \ldots, v_n$ can be represented as a linear combination of these values:

$$d = c_1 \cdot v_1 + \ldots + c_n \cdot v_n.$$

For each value v_i, we have $(2v_i, 0) \in S_0$, thus, for

$$2d = c_1 \cdot (2v_1) + \ldots + c_n \cdot (2v_n),$$

by Part 7, we get $(2d, 0) \in S_0$. Due to Part 3, we thus have $(0, 2d) \in S_0$, and due to Parts 5 and 6, all points $(n_x \cdot (2d), n_y \cdot (2d))$ for integers n_x and n_y also belong to the set S_0.

If S_0 has no other points, then for sets containing $(0, 0)$, we indeed conclude that these sets form a desired sub-grid, with $k = 2d$.

9°. What if these are other points in the set S_0? Since d is the greatest common divisor of all the coordinate values, each of these points has the form $(c_x \cdot d, c_y \cdot d)$ for some integers c_x and c_y. Since this point is not of the form $(n_x \cdot (2d), n_y \cdot (2d))$, this means that either c_x, or c_y is an odd number – or both.

Let us first consider the case when exactly one of the values c_x and c_y is odd. Without losing generality, let us assume that c_x is odd, so $c_x = 2n_x + 1$ and $c_y = 2n_y$ for some integers n_x and n_y. Due to Part 9, we have $(2n_x \cdot d, 2n_y \cdot d) \in S_0$, to the difference

$$((2n_x + 1) \cdot d, 2n_y \cdot d) - (2n_x \cdot d, 2n_y \cdot d) = (d, 0)$$

also belongs to the set S_0. Thus, similarly to Part 8, we can conclude that for every two integers c_x and c_y, we have $(c_x \cdot d, c_y \cdot d) \in S_0$. So, in this case, S_0 coincides with the sub-grid for which $k = d$.

The only remaining case is when not all points $(c_x \cdot d, c_y \cdot d)$ belong to the set S_0. This means that for some such point both values c_x and c_y are odd: $c_x = 2n_x + 1$ and $c_y = 2n_y + 1$ for some integers n_x and n_y. Due to Part 9, we have $(2n_x \cdot d, 2n_y \cdot d) \in S_0$, to the difference

$$((2n_x + 1) \cdot d, (2n_y + 1) \cdot d) - (2n_x \cdot d, 2n_y \cdot d) = (d, d)$$

also belongs to the set S_0.

Since, due to Part 9, we have $(2n_x \cdot d, 2n_y \cdot d) \in S_0$ for all n_x and n_y, we conclude, by using Part 4, that $(2n_x + 1) \cdot d, (2n_y + 1) \cdot d) \in S_0$. So, all pairs for which both coordinates are odd multiples of d are in S_0. Thus, we get the new case described in Proposition 1.

10°. The previous results were about the sets containing the point $(0, 0)$.

For all other sets S containing some other point (x_0, y_0), we get the same result about the sub-grid if we take into account that the optimal family is invariant, and thus, with the set S, the optimal family also contains the set $T_{x_0,y_0}(S)$ that contains $(0, 0)$ and is, thus, equal either to the desired sub-grid or to the new sub-grid-type set.

The proposition is proven.

Author Contributions: Both authors contributed equally to this paper. Both authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and HRD-1834620 and HRD-2034030 (CAHSI Includes). It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Leaning*, MIT Press: Cambridge, Massachusetts, 2016.
2. Kreinovich, V.; Kosheleva, O. Optimization under uncertainty explains empirical success of deep learning heuristics", In: Pardalos, P.; Rasskazova, V.; Vrahatis, M.N. (eds.), *Black Box Optimization, Machine Learning and No-Free Lunch Theorems*, Springer: Cham, Switzerland, 2021, pp. 195–220.

3. Li, Y.; Zhang, X.; Chen, D. CSRNet: dilated convolutional neural networks for understanding the highly congested scenes, *Proceedings of the 2018 Conference on Computer Vision and Pattern Recognition CVPR'2018*, Salt Lake City, Utah, June 18–22, 2018, pp. 1091–1100.
4. Nguyen, H.T.; Kreinovich, V. *Applications of Continuous Mathematics to Computer Science*, Kluwer: Dordrecht, 1997.
5. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions, *Proceedings of the 4th International Conference on Learning Representations ICLR'2016*, San Juan, Puerto Rico, May 2–4, 2016.
6. Zhang, X.; Zou, Y.; Shi, W. Dilated convolution neural network with LeakyReLU for environmental sound classification, *Proceedings of the 2017 22nd International Conference on Digital Signal Processing DSP'2017*, London, U.K., August 23–25, 2017.