On the Blindspots of Convolutional Networks

Elad Hoffer
Intel Collaborative Research Institute
for Computational Intelligence (ICRI-CI)
Intel Labs
Technion Israel Institute Of Technology
elad.hoffer@gmail.com

Shai Fine
Intel Collaborative Research Institute
for Computational Intelligence (ICRI-CI)
Intel Labs
Shai.Fine@intel.com

ABSTRACT

Deep convolutional network have been the state-of-the-art approach for a wide variety of tasks over the last few years. Its successes have, in many cases, turned it into the default model in quite a few domains. In this work we will demonstrate that convolutional networks have limitations that may, in some cases, hinder it from learning properties of the data, which are easily recognizable by traditional, less demanding, models. To this end, we present a series of competitive analysis studies on image recognition and text analysis tasks, for which convolutional networks are known to provide state-of-the-art results. In our studies, we injected a truth-reveling signal, indiscernible for the network, thus hitting time and again the network's blind spots. The signal doesn't impair the network's existing performances, but it does provide an opportunity for a significant performance boost by models that can capture it. The various forms of the carefully designed signals shed a light on the strengths and weaknesses of convolutional network, which may provide insights for both theoreticians that study the power of deep architectures, and for practitioners that consider to apply convolutional networks to the task at hand.

CCS Concepts

•**Computing methodologies** → **Neural networks;** *Supervised learning by classification;* Machine learning approaches;

Keywords

Convolutional Network, ConvNet, CNN, Deep Learning

1. INTRODUCTION

Over the last few years, deep convolutional networks (ConvNet) demonstrated exceptional performances, many a time near human-level, in a wide range of tasks. These successes have made ConvNet the model of choice in quite a few domains. What makes ConvNets so successful is less obvious and a considerable effort is directed to study this question. This work is a modest contribution to this effort.

We will show that convolutional networks have limitations that may, in some cases, impair their ability to learn critical properties of the data, properties which are easily recognizable by traditional, less demanding, models.

Two recent studies have addressed the question what may cause deep neural networks (DNN) to fail measurably compering to human vision: In [14] it was shown that changing an image in a way imperceptible to humans can cause a DNN to label the image as something else entirely (i.e. False Negative), and in [10] it was shown that it is easy to produce images that are completely unrecognizable to humans, but state-of-the-art DNNs believe that these are recognizable objects with 99.99 confidence (i.e False Positive).

In this study we take a different approach and rather than making the network completely fail, we look at cases where the network fail to capture significant information embedded in the data. We do that by injecting a highly relevant information in objects which are known to be recognizable by ConvNets - The additional information does not derogate the network's existing performances, but if captured it may boost the performance significantly. In all cases, the ConvNets, which achieved best performances beforehand, were not able to improve. On the other hand, commonly used models, such as decision trees, random forest, Naive Bayes, logistic regression, and even shallow neural networks, which were all significantly inferior to ConvNets beforehand, were able to surpass the ConvNet's best performances by utilizing the embedded information. Comparing model properties enables to gain insights for the strengths and limitation of convolutional networks, and their projection over data characteristics and use cases. We will argue that these limitations should be taken into account, and that properties of the data can greatly affect the applicability of convolutional networks.

The rest of the paper is organized as follows: In section 3 we review ConvNets, discuss their properties, and explain the design of truth revealing signals that hit ConvNets blindspots. Section 4 detail the comparative studies conducted in image classification (c.f. 4.1) and text classification (c.f 4.2) tasks. Section 5 conclude with a discussion on the implications of the empirical results, lesson learned, and future research directions.

2. PREVIOUS WORKS

In a previous study, Szegedy et al. [14] were able to show that it is easy to modify images in a way imperceptible to

SIGKDD 16' San-Francisco
© 2016 ACM. ISBN 123-4567-24-567/08/06. . . $15.00
DOI: 10.475/123_4

human vision, such that a trained network will fail to classify them. It was also shown that certain types of noise, learned through optimization in the image space, are causing miss-classification on different network and different datasets as well. This indicates that convolutional networks are reliant on certain aspects of natural-image data, that when modified, causes the ConvNets fail.

In a followup work by Goodfellow et al. [4] it was argued that these phenomenon are related to quantization properties of natural images - The quantization used for pixel values creates "gaps", which when optimized against classification objective, causes the network to fail. Adversarial examples in essence exploit these "gaps" in the input space, leveraging unperceived (by humans) artifacts that are not characteristic traits of natural data. Generating adversarial examples and incorporating them in the training procedure can improve the generalization on the entire test set, and reduce the error on the adversarial examples. However, additional examples can be created using the subsequent model.

More recently, Nguyen et al. [10] released a complementary work, were they show that it is possible to create images which will be received with high confidence by a trained network, while containing content that is unrecognizable by humans. A simple procedure was introduced to alter existing images or create new content that holds this property, demonstrating that networks are especially fooled by a variety of geometrical shapes. These shapes are interpretable by humans, however they share little resemblance to the classified category. Nguyen et al.'s work highlights the limitations of deep convolutional networks arising from their generalization capability - It makes use of visual features that were learned to discriminate between training samples, to create pseudo images that fooled the trained network.

3. CONVNETS BLINDSPOTS

3.1 Convolutional networks

Deep convolutional neural network is the leading approach of deep learning in computer vision tasks. The main premise is that stationary properties of images allows the use of a reduced set of parameters that needs to be learned. Convolutional layers consists of a set of trainable weight kernels, that produce their output value by cross-correlating with input data. This way, every spatial patch in the image is weighted using the same shared kernels. In each layer, for every spatial location there are multiple values describing it. The different values for each location are known as the layer's *feature maps.*

Denoting the learnable kernel weights as w, the input pixels as $x_{i,j,k}$ and output pixels as $y_{i,j,k}$, where i, j is the spatial location and k is the corresponding feature map. The computed function for each feature map in a convolution layer ℓ is

$$y^{\ell}_{i,j,k} = \sum_{f=0}^{F-1} \sum_{u=0}^{M-1} \sum_{n=0}^{N-1} w_{n,m,f,k} \cdot x^{\ell-1}_{(i+n),(j+m),k} + b_k. \quad (1)$$

Since convolutional layers contain weights that are shared by multiple spatial regions, they naturally aggregate and average the gradients over large amounts of data. The same function can be applied as 1-dimensional convolution that can be applied on temporal data such as audio and text.

Another main layer of ConvNets is the pooling layer. Pooling layers are used to reduce the dimensionality of the input while gaining scale and shift invariance to small amounts. This is done by simply pooling (or down-sampling) spatial regions of the input, taking the average (Average-Pooling), max (Max-Pooling) or other variants.

Most ConvNets are comprised of layers of Convolution & Pooling followed by fully-connected layers (dot products) and a classifier. Between layers with learned parameters, a non-linearity function is applied. Modern deep learning models usually employ *ReLU* (rectified-linear-unit) $f(x) = \max(0, x)$ or some variant of it. Convolutional networks are used for different vision-related tasks such as classification[7], semantic segmentation[8], detection[13] and control[9].

Lately, ConvNets were also shown to provide competitive results in language domain on tasks requiring to classify segments of text by using convolutions on character-level [16].

Convolutional netwrok are trained by variants of stochastic-gradient-descent (SGD) on batches, and usually contain a number of parameters that exceeds the number of data samples by a large margin (over-specification). Another noteworthy aspect of ConvNets is that the input is processed over local regions (kernels are usually on scale of 3-11 pixels), while global information is gathered by stacking multiple layers, so that the final "spatial-region" of the model is at the scale of the whole image/sample.

3.2 Injecting truth-revealing signal

The use of convolutional networks is based on the assumption that the data fed into the network has stationary properties, with strong locality either in space (vision) or time (text, speech, etc). While providing a strong prior, which is helpful for many natural data source, this assumption may also lead to "blindspots" that should be reckoned with:

- Inability to identify *low-dimensional* signals in the input space, due to aggregation and "smoothing" by averaging of many data points
- Inability to identify *global* signals within the data, due to the focus on local features

A Third type of "blindspots", which we term *moving target*, is comprised by a combination of the former two

- Inability to identify low-dimensional signals which varies in space and time at a global scale, due to the non-stationary nature of that signal

These "blindspots" are not necessarily harmful in data such as natural images, meaning they may not interfere with the learning and scoring conducted by ConvNets (e.g. unlike the signals described in Section 2). However, failing to capture the information embedded in the "blindspots" may cause convolutional networks to miss out information which is crucial for the task at hand.

To study these limitations of convolutional nets, we designed truth-revealing signals that reside well within the "blindspots", and we embedded these signals in the data. The amount of the information injected to the data varied from incomplete to complete , i.e. from lossy to lossless encoding of the learning targets (the classification labels). Nevertheless, in all our case studies, the information conveyed by the embedded signals were significant, and models

Figure 1: Original images

that could capture and exploit these signals were able to substantially improve their accuracy.

4. EXPERIMENTS

4.1 Vision - image classification

All the experiments were conducted on the CIFAR-10 dataset[6], which consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images (c.f Fig 1). Convolutinal network models were trained using “Torch7” environment [3].

In our studies, we employed five different coding schemes of the ground-truth labels in the images:

- One pixel encoding - Changing one pixel (same location for all images) to hold the label (class number). An example of a bird’s image with the encoded pixel (circled) is depicted in Figure 2.
- Pattern pixels encoding - Encoding the label by changing the pixels at specific fixed locations to hold the class number (label). In our study we used the four corners of the image for a binary encoding scheme, such that 10 labels are encoded with 4 pixels (1 in each corner), each of which represent a bit assignment, resp.
- Random pixel encoding - For each instance (image), encode the label (class number) in a randomly chosen pixel
- Multiple locations encoding - For each instance (image), encode the label with a pair of pixels, at distant locations, constrained to hold the exact same value. Thus, for every class, the label is encoded by the same location of the pair pixels that hold the same value, but the value itself changes from image to image. An example of the location encoding, where the pair pixels are circled, is depicted in Fig. 3.
- Noise encoding - The label is encoded by adding a random noise vector for each class. Those random vectors are dense and orthogonal to each other, and with a large enough norm. Thus, the noisy vectors (cf. Fig. 4) are in essence a lossless encoding of the labels, while the image itself (which is only mildly distorted, cf. Fig. 5) holds only partial information[1].

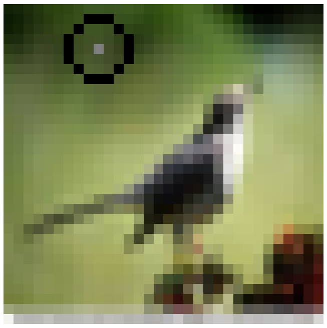

Figure 2: One pixel encoding

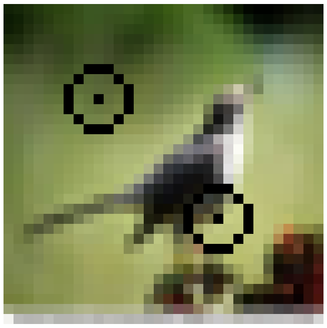

Figure 3: Multiple locations encoding

All of the above signals provide a complete information, where the first and second are *low dimensional* signals, the third and fourth are *moving target* signals, and the fifth is a *global* signal.

In all the experiments we used a simple 4-layered convolutional network with $600K$ parameters (Fig. 6). A $ReLU$ non-linearity is applied between two consecutive layers. Network configuration (ordered from input to output) consists of filter sizes {5,3,3,2}, and feature map dimensions {3,64,128,256,128}. A subsequent fully-connected layer is used to classify the 10 classes available in the dataset.

The network achieved 84.7% accuracy on the clean data, with no embedded signal, and with no data augmentation. This result, although behind state-of-the-art results ($\tilde{9}4\%$) [5] is a solid representative of a small convolutional net, and suffice to demonstrate our points. Applying the network in all five truth-revealing signal didn’t yield any change in the network performance, for better or for worse.

For traditional modeling we used Python’s scikit-learn package implementation[11] in a straightforward manner, namely no special data pre-processing, no code enhancements, nor any tailor made augmented functionality. The models that were tested are - GaussianNB, Decision Tree, Extreme Random Forest, Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analy-

[1] From a geometric standpoint, the encoding is a translation of the instances of each class along a different orthogonal axis. The translation preserve local geometric properties, thus making the scale and shift invariance properties less appealing, as the information is in the direction of translation

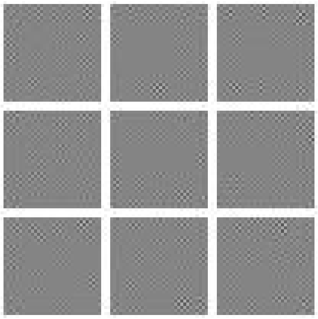

Figure 4: Orthogonal noise

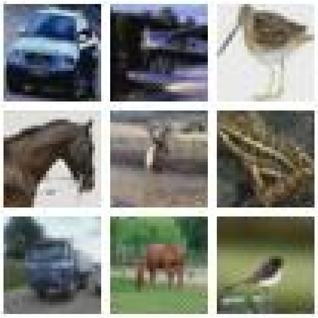

Figure 5: Noisy images

sis (QDA). In addition, we used Torch implementation of Shallow Linear NN (Perceptron), and Shallow Max NN.

The results obtained by the best models are depicted in Table 1. In all the cases, we used only 1/4 of the data and achieve best performances. In all cases except one, the best model was able to perfectly learn the exact coding scheme and to achieve perfect results. The only exception is the ”Multiple Location” scheme, and this is due to the inherent ambiguity in this encoding scheme[2]. We note in passing that in many cases some of the other models achieved comparable results. For example, although random forest is not in the table, in many of the cases it came very close to the winning model, thus making it a good universal model choice.

Truth Signal	ConvNet	Best Competing Model	
One Pixel	84.7%	100%	Dec. Tree
Pattern Pixels	84.7%	100%	Dec. Tree
Random Pixel	84.7%	100%	Shallow MaxNN
Multiple Locations	84.7%	98.4%	QDA
Noise	84.7%	100%	Perceptron

Table 1: CIFAR10 - Image Classification Accuracy

The results demonstrate the impact of the ”blindspots”. Each of the truth-revealing signals that were used, corresponds to either a local information that is being “smoothed out” (one pixel encoding, pattern pixels), a global signal that is being discarded (noise encoding) and a moving target (random pixel, multiple locations). Each of these signals, although sufficient to provide perfect classification, cannot be perceived by a convolutional network similar to the one we used. Each signal may certainly be addressed by changing the network architecture to meet the specific characteristics of that signal (e.g reducing kernel size to capture local information, or enlarging it to capture global signals), but this will require a prior knowledge of that signal, and will hamper the generic use of ConvNets. We conjecture that this set of signals will introduce the same ”blindspots” challenges for many of the popular and commonly used ConvNets.

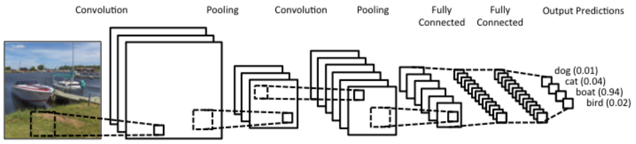

Figure 6: Spatial convolutional networks [1]

Some signals are naturally captured by a decision tree, leveraging the direct correspondence between the signal (regardless of position and locality) and the target. Other signals are captured by shallow NNs that focus attention more on the global properties of the data, thus revealing information encoded by a global truth-revealing noise. Most interesting observation occur in the case of orthogonal “noise encoding”, where it was demonstrated that while a deep convolutional network did not integrate the global information, a shallow 1-layered fully connected network achieved perfect prediction within a small number of iterations. This is due to the fact that a shallow NN is able to form a global view on the input space, and learn widely supported (global) properties. Specifically, in the ”noise encoding” case, the shallow network learned a weight vector corresponding to each of the noise vectors, which suffices to discriminate between the classes.

To further demonstrate this point, we devised images that hold no information except the label. Specifically, we used blank images with ”one pixel” signal. These images were easily classified by the ConvNet, since the truth signal had no interference with the image information. We view this as an additional evidence that it is difficult for the network to capture and use a low dimensional information, when aggregating over a larger part of the spatial space that holds information.

4.2 Text classification

To further evaluate and confirm our findings, we devised a set of experiments in the text domain. We used the framework of Zhang et al. [16], including the data set and the Torch implementation that they kindly shared. Zhang et al. designed two 1-dimensional convolutional network for text classification at the character level (c.f. Figure 7). The two ConvNets share the same architecture, and differ in their size (number of neurons) - They are both 9 layers deep with 6 convolutional layers and 3 fully-connected layers. The alphabet consists of 70 characters (including space). Hence, each input feature has 70 categorical values, which were numerically coded using one-hot-encoding scheme. Overall, there are 1014 different input features. The ConvNets also include 2 dropout modules in between the 3 fully-connected

[2] There is a chance of having the exact same value in distant pixels

layers to regularize. In our experiments we used the smaller network implemented by Zhang et al.

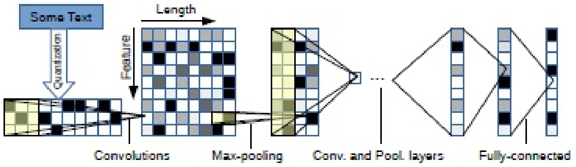

Figure 7: Zhang et al. Char ConvNet[16]

The convolutional network was trained to classify segments of written text, similarly to their spatial counterpart on images. As both kind of models (spatial and temporal ConvNets) suffer from the same limitations, we expected to observe the same phenomena - Truth-revealing signals with certain properties will be ignored by the network, while being extensively used by the other models.

In their experiments, when matched performance against traditional models, Zhang et al. used logistic regression at the word level: Bag-of-words/ngrams counts and their TFIDF (term-frequency inverse-document-frequency), resp. Since we were interested in analyzing ”blindspots”, we employed the competing models at the character level, thus enabling a fair and ”clean” competitive analysis. Zheng et al. conducted their experiments on 8 different data sets. In four out of eight (AG News, Sogou News, DBPedia, Yelp Review Polarity), logistic regression came first. In two others (Yahoo! answers, Amazon Review Polarity) logistic regression came second but with a small margin to the ConvNet. In the last two (Yelp Review Full, Amazon Review Full), the ConvNets won by a significant margin. These were the data sets that we chose to conduct our experiments (c.f. Table 2 for details of the selected data sets)

Data set	Classes	Train	Test
Yelp Review Full	5	650,000	50,000
Amazon Review Full	5	3,000,000	650,000

Table 2: Text classification data sets [16]

We employed three different coding schemes of the ground-truth labels in the text

- Mnemonic - The first character in each message was changed to reflect the category label, reminiscent of the short mnemonic form of the element name that is used in the machine-readable encoded document
- Length encoding - The label was encoded by setting a fixed length for the text in all the messages of the same class
- Pattern char - Encode the label by changing the characters at specific fixed locations to hold the class number (label). In our study we used three specific character locations (the 50th, 150th, and 200th) for a binary encoding scheme, such that 5 labels are encoded with the following ”bit” assignment - In each selected location either set the character to be 0, or keep it unchanged.

The *Mnemonic* signal is of the same nature as the *One Pixel* encoding in the images. It provides a complete information at the character level.

The *Length encoding* signal is a global signal that is being discarded by the network although it holds the necessary information. This is similar to the *Noise Encoding* signal at the image examples.

The *Pattern char* signal is a type of low dimensional signal that is being discarded because of the aggregation and smoothing done by the network. It is of the same nature as the *Pattern pixels* encoding, however it provides only partial information

The results attained with the small ConvNet on a clean data (without signal injection) were in the same ballpark as the ones reported at [16]. The network didn’t yield any significant improvement in performances when trained on any of the truth-revealing signal injected data sets.

Similarly to the image classification experiments, for traditional modeling we used scikit-learn implementation. However, unlike the image classification setting, it turned out that for text classification Decision Tree is the best model to handle all injected signals at the character level. The results are depicted at Table 3. It includes also the best quoted results of Zhang et al. using ConvNets at the character level and Logistic Regression at the word level.

We can see that the same observations seen in the spatial case are also visible when applying convolutional network on text. We can attribute once again the 3 types of truth-revealing signals to either global information being missed (Length encoding) or local information being smoothed and discarded (Mnemonic, Pattern char). We can also see the effect of providing partial vs. complete information - Mnemonic signal provide complete information, Length encoding is a little ambiguous (uses space padding to mark the end of the text, but spaces can be found in the body of the text as well), and Pattern char provides a partial information (the bit states are encoded with either 0 or the existing char unchanged, which might be 0 as well, since 0 represents the space char).

Truth-Signal	Model	Yelp F.	Amazon F.
Mnemonic	DT	100%	100%
Length encoding	DT	99.80%	99.97%
Pattern char	DT	68.66%	69.66%
Clean data	LR	59.86%	55.26%
Clean data	ConvNet	62.05%	62.05%

Table 3: Text Classification accuracy

5. DISCUSSION

In this paper we introduced the notion of ”blindspot” as a mean to study the strength and limitations of convolutional networks. Our method is based on injecting a truth-revealing signal to the data, indiscernible to the network, thus hitting time and again the network’s blind spots. The signal doesn’t impair the network’s existing performances. Rather, it provides an opportunity for a significant performance boost by models that can capture it.

The models we used to compare with are all standard, integral part of commonly used generic Machine Learning toolboxes. In all our experiments we used the Python’s Scikit-learn package[11] in a straightforward manner, namely no special data pre-processing, no code enhancements, nor any tailor made augmented functionality.

We conducted our series of experiments in two different application domains, images and text, using two different convolutional networks that differ in their input space encoding (discrete symbols for text, continuous pixel values for image) and their processing over data (temporal vs spatial convolution). Still, we ended up with similar observations and similar ”blindspots”. This demonstrates the robustness of our findings, and it enable us to come up with a unified set of insights and recommendations.

In designing the studies, we started with analyzing the principle characteristics of ConvNets, hypothesize on their deficiencies, and then designed experiments to either validate or disprove these assumptions. Our leading theme is that ConvNets are less susceptible to complete and incomplete information with the following characterization

- Dimension - The information resides either at a very low dimension or spread over very many dimensions
- Global - When the information is in a form of a global property, such as max or some forms of summary statistics

Taking advantage over these deficiencies, we were able to layout simple design principles for the truth-revealing signals, which are manifested by the following groupings of the encoding schemes

1. Low dimensional signals - One pixel encoding, Pattern pixel encoding, Pattern char encoding, Mnemonic encoding
2. High dimensional signal - Noise encoding, Length encoding
3. Stochastic signal - Random pixel encoding, Multiple locations

The various signals injected to the data are artificially made and highly unlikely to appear in natural data. Still, they can be seen as extreme cases of naturally occurring situations. For example, the *orthogonal noise* we've added to the image dataset and shown to be missed by the convolutional network, can be seen as extreme case of global noise that is informative. A natural case of this kind of noise is when a global signal like physical aberrations from the photography machinery is added to images and is correlated with the conditions in which an image was taken. It is thus informative to the classification task, and ignoring it can hurt potential performance. A similar example can be seen in text, where the *length encoding* we have shown to be ignored, can represent a global structural attribute of the text lines. For example, the style a line was written.

An interesting observation is that properties, such as invariance to shift and scale, which are usually considered as an advantage of ConvNets, may also be harmful, e.g. when the information is actually codded in the shift and scale itself. This is a good reminder to the “No Free Lunch” Theorem [15], which state that there is no single machine learning model that is uniformly better then any other across all domains and all tasks.

In some cases, a shallow network was able to capture the (important) information that a deep network failed to notice. In another case, when we ”shutdown” all the information except the truth-revealing signal, the deep network was able to capture it and reached perfect performance. We believe that these two phenomena are rooted at the same characteristic of the deep network - The form of the interplay that deep convolutional networks conduct between feature construction (convolution layers) and feature selection (pooling layers). By shutting down obscuring information, and by using a wider fun in (using shallow networks), we manage to intervene this interplay and overcome the difficulties in some of the cases.

The comparison of shallow vs. deep network is of a particular interest, since in recent work it was argued that deep neural networks provide a universal advantage over shallow networks [2]. Our findings are not necessarily constructing, but it seems that there is a need for a finer characterization of the strength of deep architectures, and how it is manifested in various domains.

Using ConvNets on a new source of data should take into account the ”blindspots” presented in this work. This is especially important when the data is pre-processed to suit a convolutional processing, possibly incurring noise artifacts or ignoring crucial information. In our studies we witness this phenomena with the text example, where the one-hot-encoding fed into the networks allows us to insert the *length encoding* signal without being noticed. The same phenomena might occur for example, when learning from audio signals by running a ConvNets on spectrograms [12] and ignoring valuable information.

Last but not least is the utilization of traditional machine learning models - Compared with ConvNets, these models offer a significant computational advantage, require far less data to converge, do not involve a lot of parameter tuning, and yield simpler hypotheses, which according to Occam's Razor, should be favored when performance are comparable.

Acknowledgement

We would like to thank Lior Wolf for suggesting the ”information shut down” experiment in images, and the ”length encoding” experiment in text.

6. REFERENCES

[1] Clarifai. Convolutional neural networks in practice. http://www.clarifai.com/technology.

[2] N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning: a tensor analysis. *arXiv preprint arXiv:1509.05009*, 2015.

[3] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[5] B. Graham. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*, 2014.

[6] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[10] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 427–436. IEEE, 2015.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] J. Schluter and S. Bock. Improved musical onset detection with convolutional neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6979–6983. IEEE, 2014.

[13] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat : Integrated Recognition , Localization and Detection using Convolutional Networks. *arXiv preprint arXiv:1312.6229*, pages 1–15, 2013.

[14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[15] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 1996.

[16] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657, 2015.