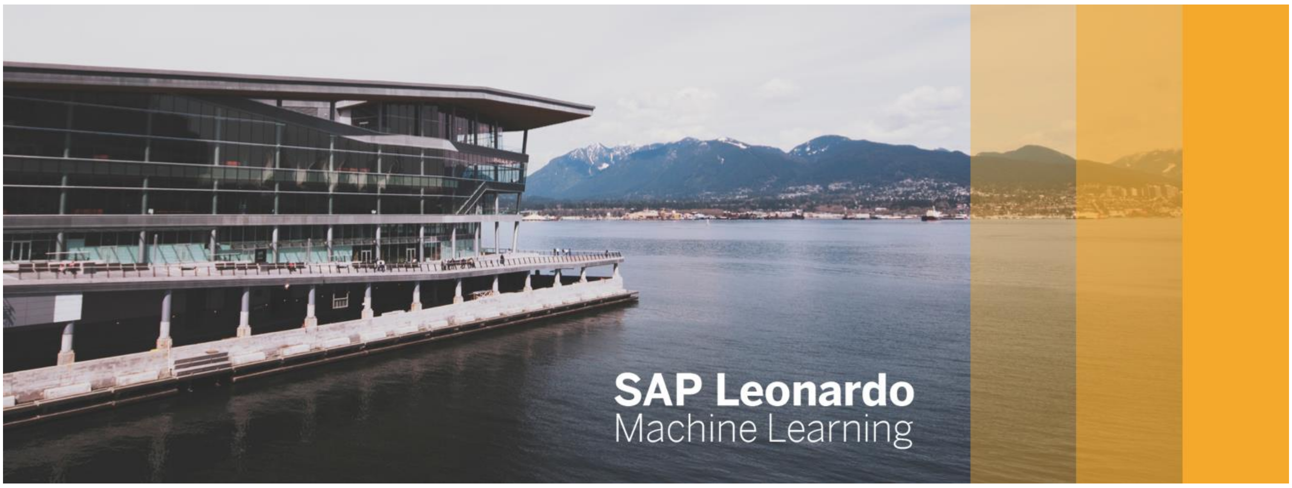
Conference Report

ICLR 2018

Vancouver, Canada, April 30 – May 03



Written by: Tassilo Klein, Shachar Klaiman, Konrad Schenk, Marius Lehne, Steven Jaeger

THE BEST RUN SAP

For its sixth consecutive year, the International Conference on Learning Representations (ICLR), one of the main conferences in the field of deep learning, took place at the Vancouver Convention Centre in (city), Canada between April 30th - May 3rd, 2018.

ICLR enables those in the field to review recent techniques and trends in learning representations and addresses an international community of researchers, students and industry partners. Much like the AI industry as a whole, ICLR is experiencing exponential growth. This year's conference received 935 submissions compared to 430 in 2017, from which 23 (2%) were accepted for oral workshops and 314 (34%) for posters. Similarly, attendance to ICLR has nearly doubled from around 1100 visitors to roughly 2000 compared to the previous year.

As official sponsor of ICLR’18, the SAP team showcased our newest research projects and participated in the large spectrum of talks and workshop sessions offered at the conference. In this report, we will summarize general trends and paper highlights we discovered that are worth revisiting.

MAIN TOPICS AND TRENDS

Generative Adversarial Networks (GANs)

With over 40 conference contributions in this area, Generative Adversarial Networks (GANs) remain one of the hottest topics in current machine learning research; a trend which can also be observed in other conferences. There were numerous contributions that focused on making the training procedure more stable through improved loss functions and architectures. Further, there is ongoing research on more fundamental and theoretical questions.

Improvements were made on the side of applying GANs, such as in a conference contribution that used GANs to break cyphers in an unsupervised way. Also in the vision domain, improvements have been made such that GANs are now able to generate realistic high resolution images.

Adversarial Attacks and Defenses

Deep neural networks can be manipulated into predicting incorrect classes through malicious input data. Those adversarial examples are characterized through the application of minimal perturbations to the original data such that a previously correct classification leads to an arbitrary classification result with a high confidence.

Adversarial attacks pose a severe risk to the safety of various types of systems using state-of-the art machine learning methods. For example an attacker could fool an autonomous driving system through a manipulation of a traffic sign. Depending on the intention of the attacker, this could lead to catastrophic results. Thus, increasing the robustness against adversarial attacks is an active subject of research. The topic was present through various talks and poster presentations. Continuous research is performed on finding and improving methods to generate adversarial examples, on strategies to defend against such attacks as well as getting a more in-depth understanding of adversarial attacks in general.

Compressing Neural Networks / Deep Learning on the Edge / Architecting resource efficient DNN

The notion of memory-efficient models as well as models that allow fast inference is getting more and more attention. This can be attributed to the fact that current focus was put on pushing for accuracy by devising ever deeper architectures that have large memory footprint and are slow. Therefore, operating these architectures in a cost-efficient manner or on portable devices is hardly possible. To alleviate the problem without sacrificing accuracy, the community is working on a series of approaches. Among the most prominent ones are quantization or binarization of network weights, selective removal of neurons to reduce over-capacity of networks and knowledge distillation, i.e. emulating the a deep network by a shallow network and smart architectures that allow skipping computations, i.e. for easy data points.

Deep Reinforcement Learning

While the first ideas for Reinforcement Learning (RL) dates back several decades, a lot of progress has been made with the advent of deep learning. The main concept of an RL system is to have an agent which learns a policy to change a systems state towards a target state through actions. The agent initially does not know which action is the most feasible for the currently given state but gets occasional rewards which are leveraged to optimize the policy of the agent.

The first ideas of leveraging the powers of deep learning for RL were to estimate the Q-values, which represent the total expected reward for the agent after taking a particular action, with neural networks. Soon, deep learning networks were used to replace other parts of conventional RL like policies or even to extend the RL framework, e.g. with actor-critic networks.

Even though RL still does not generalize well and shows difficulties in reproducibility, as stated in one of the invited talks, more than 70 related papers were published at ICLR in 2018. The scope ranges from applying Deep RL to solve various tasks to leveraging RL in DL methods. The more theoretical papers focus on generalizing learned policies or on improving RL with new approaches.

BEST PAPERS

The committee of ICLR’2018 has chosen three best papers, which are listed below along with the reasons of why we think they are of particular interest.

- **On the Convergence of ADAM and Beyond**

Reddi et al., 2018

Abstract:
“Several recently proposed stochastic optimization methods that have been successfully used in training deep networks such as RMSProp, Adam, Adadelta, Nadam are based on using gradient updates scaled by square roots of exponential moving averages of squared past gradients. In many applications, e.g. learning with large output spaces, it has been empirically observed that these algorithms fail to converge to an optimal solution (or a critical point in nonconvex settings). We show that one cause for such failures is the exponential moving average used in the algorithms. We provide an explicit example of a simple convex optimization setting where Adam does not converge to the optimal solution, and describe the precise problems with the previous analysis of Adam algorithm. Our analysis suggests that the convergence issues can be fixed by endowing such algorithms with ``long-term memory'' of past gradients, and propose new variants of the Adam algorithm which not only fix the convergence issues but often also lead to improved empirical performance.”

Why we think it’s interesting:

In this paper, the convergence of popular optimization algorithms like Adam and RMSProp are examined. Since they frequently fail to converge to an optimal solution due to their exponential moving average, new variants of these methods are presented. These will probably converge to an optimal solution in convex settings.

- **Spherical CNN**

Cohen et al., 2018

Abstract:
“Convolutional Neural Networks (CNNs) have become the method of choice for learning problems involving 2D planar images. However, a number of problems of recent interest have created a demand for models that can analyze spherical images. Examples include omnidirectional vision for drones, robots, and autonomous cars, molecular regression problems, and global weather and climate modelling. A naive application of convolutional networks to a planar projection of the spherical signal is destined to fail, because the space-varying distortions introduced by such a projection will make translational weight sharing ineffective. In this paper we introduce the building blocks for constructing spherical CNNs. We propose a definition for the spherical cross-correlation that is both expressive and rotation-equivariant. The spherical correlation satisfies a generalized Fourier theorem, which allows us to compute it efficiently using a generalized (non-commutative) Fast Fourier Transform (FFT) algorithm. We demonstrate the computational efficiency, numerical accuracy, and effectiveness of spherical CNNs applied to 3D model recognition and atomization energy regression.”

Why we think it's interesting:

Standard CNNs are tailored to a planar image space. In case of spherical images like omnidirectional pictures or global weather maps, a simple projection to a planar space is destined to fail due to distortions close to the poles. This paper introduces a convolutional operator that is both expressive and rotation-equivariant to solve the described problem by utilizing a generalized Fast Fourier Transformation.

- **Continuous adaptation via meta-learning in nonstationary and competitive environments**

Al-Shedivat et al., 2018

Abstract:

"Ability to continuously learn and adapt from limited experience in nonstationary environments is an important milestone on the path towards general intelligence. In this paper, we cast the problem of continuous adaptation into the learning-to-learn framework. We develop a simple gradient-based meta-learning algorithm suitable for adaptation in dynamically changing and adversarial scenarios. Additionally, we design a new multi-agent competitive environment, RoboSumo, and define iterated adaptation games for testing various aspects of continuous adaptation. We demonstrate that meta-learning enables significantly more efficient adaptation than reactive baselines in the few-shot regime. Our experiments with a population of agents that learn and compete suggest that meta-learners are the fittest."

Why we think it's interesting:

Agents that want to generalize to real world scenarios need to adapt to nonstationary environments. Meta-learning seems to be an interesting approach to tackle this issue and following the author's results, it is also an efficient way of few-shot adaption in competitive environments which are represented as Markov Chain of different tasks.

OUR PAPER HIGHLIGHTS

Due to the high number of papers at this year's ICLR, we can only present a selection of papers that we found specifically impressive. However, there were a lot more interesting papers presented and we therefore strongly recommend having a look at the list of papers and workshop sites here.

For illustrative purposes, selected figures were taken from the papers.

- **Learning and Memorization**

Chatterjee, 2018

Abstract:

"In the machine learning research community, it is generally believed that there is a tension between memorization and generalization. In this work we examine to what extent this tension exists, by exploring if it is possible to generalize through memorization alone. Although direct memorization with a lookup table obviously does not generalize, we find that introducing depth in the form of a network of support-limited lookup tables leads to generalization that is significantly above chance and closer to those obtained by standard learning algorithms on several tasks derived from MNIST and CIFAR-10. Furthermore, we demonstrate through a series of empirical results that our approach allows for a smooth tradeoff between memorization and generalization and exhibits some of the most salient characteristics of neural networks: depth improves performance; random data can be memorized and yet there is generalization on real data; and memorizing random data is harder in a certain sense than memorizing real data. The extreme simplicity of the algorithm and potential connections with stability provide important insights into the impact of depth on learning algorithms, and point to several interesting directions for future research."

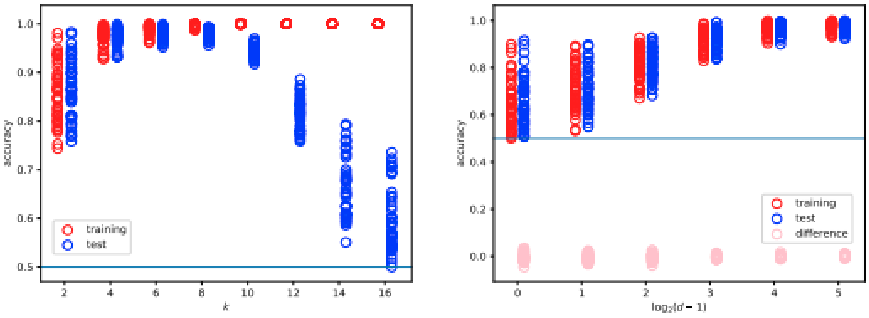


Figure 1: Generalization error (difference of training and test accuracy) goes up as k increases on the 45 pairwise separation tasks on 1-bit quantized MNIST (left). The large variation for $k = 2$ is due to insufficient mixing, and as we increase the depth of the network training this goes down (right).

Why we think it's interesting:

This paper is interesting because it proves that a network of simple lookup tables is capable of solving the MNIST task with surprisingly good test accuracies. It shows that memorization can lead to generalization.

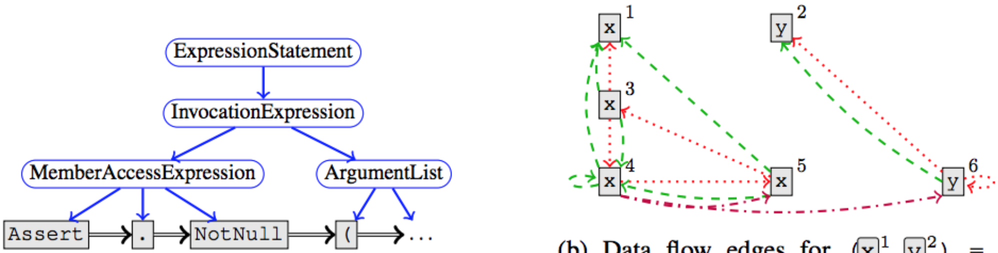
- **Learning to represent programs with graphs**

Allamanis et al., 2018

Abstract:

“Learning tasks on source code (i.e., formal languages) have been considered recently, but most work has tried to transfer natural language methods and does not capitalize on the unique opportunities offered by code's known syntax. For example, long-range dependencies induced by using the same variable or function in distant locations are often not considered. We propose to use graphs to represent both the syntactic and semantic structure of code and use graph-based deep learning methods to learn to reason over program structures.

In this work, we present how to construct graphs from source code and how to scale Gated Graph Neural Networks training to such large graphs. We evaluate our method on two tasks: VarNaming, in which a network attempts to predict the name of a variable given its usage, and VarMisuse, in which the network learns to reason about selecting the correct variable that should be used at a given program location. Our comparison to methods that use less structured program representations shows the advantages of modeling known structure, and suggests that our models learn to infer meaningful names and to solve the VarMisuse task in many cases. Additionally, our testing showed that VarMisuse identifies a number of bugs in mature open-source projects.”



(a) Simplified syntax graph for line 2 of Fig. 1, where blue rounded boxes are syntax nodes, black rectangular boxes syntax tokens, blue edges Child edges and double black edges NextToken edges.

(b) Data flow edges for ($\boxed{x}^1$,$\boxed{y}^2$) = Foo(); while ($\boxed{x}^3$ > 0) $\boxed{x}^4$ = $\boxed{x}^5$ + $\boxed{y}^6$ (indices added for clarity), with red dotted LastUse edges, green dashed LastWrite edges and dashdotted purple ComputedFrom edges.

Figure 2: Examples of graph edges used in program representation.

Why we think it’s interesting:

The authors show how it is possible to represent a program in a neural network. They observe that programming languages enforce a graph structure and therefore make direct use of graph-based neural network architectures. This work showcases some fascinating applications that can improve code quality and detect previously undetected errors in source code on real-world data sets.

- **[Learning to Learn Without Labels]**

Metz et al., 2018

Abstract:

„A major goal of unsupervised learning is for algorithms to learn representations of data, useful for subsequent tasks, without access to supervised labels or other high-level attributes. Typically, these algorithms minimize a surrogate objective, such as reconstruction error or likelihood of a generative model, with the hope that representations useful for subsequent tasks will arise as a side effect (e.g. semi-supervised classification). In this work, we propose using meta-learning to learn an unsupervised learning rule, and meta-optimize the learning rule directly to produce good representations for a desired task. Here, our desired task (meta-objective) is the performance of the representation on semi-supervised classification, and we meta-learn an algorithm -- an unsupervised weight update rule -- that produces representations that perform well under this meta-objective. We examine the performance of the learned algorithm on several datasets and show that it learns useful features, generalizes across both network architectures and a wide array of datasets, and outperforms existing unsupervised learning techniques.“

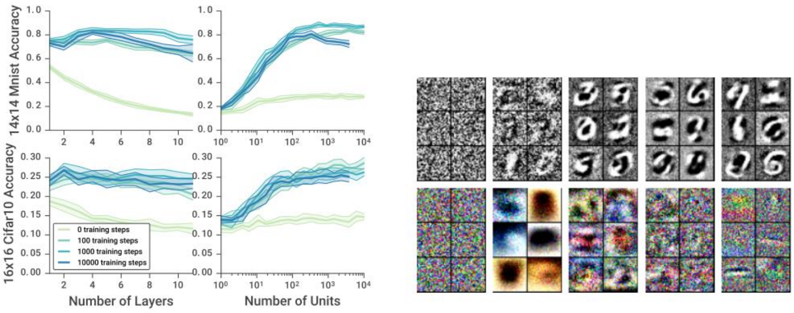


Figure 2: **Left:**The learned optimizer is capable of optimizing base models with hidden sizes and depths outside the meta-training regime. As we increase the number of units per layer, the learned model can make use of this additional capacity despite never having experienced it during meta-training. **Right:** From left to right we show first layer base-model ϕ produced by our learned optimizer over the course of meta-training. Each pane consists of first layer filters extracted from ϕ after 10k applications of the learned update rule on MNIST (top) and CIFAR10 (bottom). For MNIST, the optimizer learns image template like features. For CIFAR, low frequency features evolve into high frequencies and local edge detectors.

Why we think it’s interesting:

Instead of learning transferable features, the authors learn a transferable learning rule which does not require access to labels and generalizes across both data domains and neural network architectures. While they focus on the meta-objective of semi-supervised classification here, in principle a learning rule could be optimized to generate representations for any subsequent task.

- **Polar Transformer Networks**

Esteves et al., 2018

Abstract:

“Convolutional neural networks (CNNs) are inherently equivariant to translation. Efforts to embed other forms of equivariance have concentrated solely on rotation. We expand the notion of equivariance in CNNs through the Polar Transformer Network (PTN). PTN combines ideas from the Spatial Transformer Network (STN) and canonical coordinate representations. The result is a network invariant to translation and equivariant to both rotation and scale. PTN is trained end-to-end and composed of three distinct stages: a polar origin predictor, the newly introduced polar transformer module and a classifier. PTN achieves state-of-the-art on rotated MNIST and the newly introduced SIM2MNIST dataset, an MNIST variation obtained by adding clutter and perturbing digits with translation, rotation and scaling. The ideas of PTN are extensible to 3D which we demonstrate through the Cylindrical Transformer Network.”

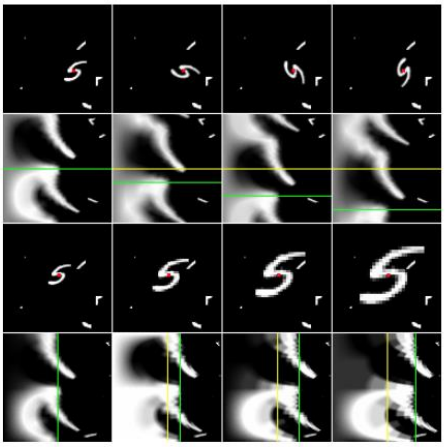


Figure 1: In the log-polar representation, rotations around the origin become vertical shifts, and dilations around the origin become horizontal shifts. The distance between the yellow and green lines is proportional to the rotation angle/scale factor. Top rows: sequence of rotations, and the corresponding polar images. Bottom rows: sequence of dilations, and the corresponding polar images.

Why we think it’s interesting:

The authors propose a novel network whose output is invariant to translations and equivariant to the group of dilations/rotations. By combining a polar origin predictor network, they are able to transform the input image into log-polar coordinates in which rotations and scaling manifest as translations. In this way, they are able to use normal convolutional neural networks (with a small adaptation to the padding) while retaining robustness to rotations and scale.

- **Wasserstein Auto-Encoders**

Tolstikhin et al., 2018

Abstract:

"We propose the Wasserstein Auto-Encoder (WAE)—a new algorithm for building a generative model of the data distribution. WAE minimizes a penalized form of the Wasserstein distance between the model distribution and the target distribution, which leads to a different regularizer than the one used by the Variational Auto-Encoder (VAE) (Kingma & Welling, 2014). This regularizer encourages the encoded training distribution to match the prior. We compare our algorithm with several other techniques and show that it is a generalization of adversarial auto-encoders (AAE) (Makhzani et al., 2016). Our experiments show that WAE shares many of the properties of VAEs (stable training, encoder-decoder architecture, nice latent manifold structure) while generating samples of better quality, as measured by the FID score."

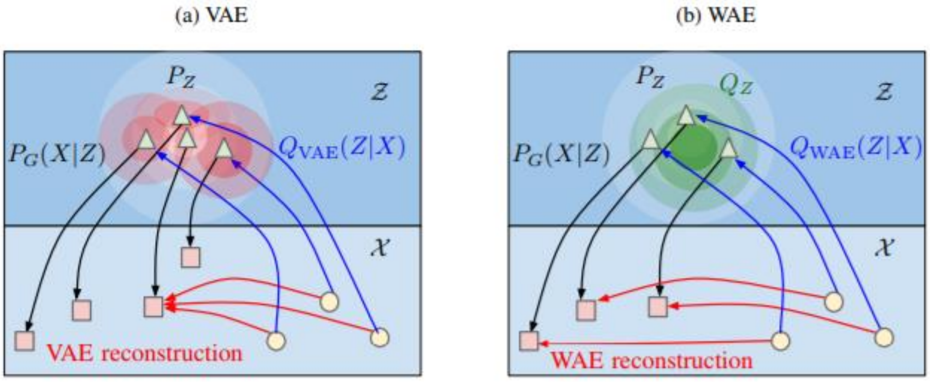


Figure 1: Both VAE and WAE minimize two terms: the reconstruction cost and the regularizer penalizing discrepancy between P_Z and distribution induced by the encoder Q. VAE forces $Q(Z|X=x)$ to match P_Z for all the different input examples x drawn from P_X. This is illustrated on picture (a), where every single red ball is forced to match P_Z depicted as the white shape. Red balls start intersecting, which leads to problems with reconstruction. In contrast, WAE forces the continuous mixture $Q_Z := \int Q(Z|X)dP_X$ to match P_Z, as depicted with the green ball in picture (b). As a result latent codes of different examples get a chance to stay far away from each other, promoting a better reconstruction.

Why we think it's interesting:

By reformulating the regularizer of the auto-encoder, the Wasserstein auto-encoder learns the encoder's distribution from the continuous mixture of the input examples. This stands in contrast to variational auto-encoders where each input example is forced to match the expected distribution.

- **On the importance of Single Directions for Generalization**

Morcos et al., 2018

Abstract:

“Despite their ability to memorize large datasets, deep neural networks often achieve good generalization performance. However, the differences between the learned solutions of networks which generalize and those which do not remain unclear. Additionally, the tuning properties of single directions (defined as the activation of a single unit or some linear combination of units in response to some input) have been highlighted, but their importance has not been evaluated. Here, we connect these lines of inquiry to demonstrate that a network's reliance on single directions is a good predictor of its generalization performance, across networks trained on datasets with different fractions of corrupted labels, across ensembles of networks trained on datasets with unmodified labels, across different hyperparameters, and over the course of training. While dropout only regularizes this quantity up to a point, batch normalization implicitly discourages single direction reliance, in part by decreasing the class selectivity of individual units. Finally, we find that class selectivity is a poor predictor of task importance, suggesting not only that networks which generalize well minimize their dependence on individual units by reducing their selectivity, but also that individually selective units may not be necessary for strong network performance.”

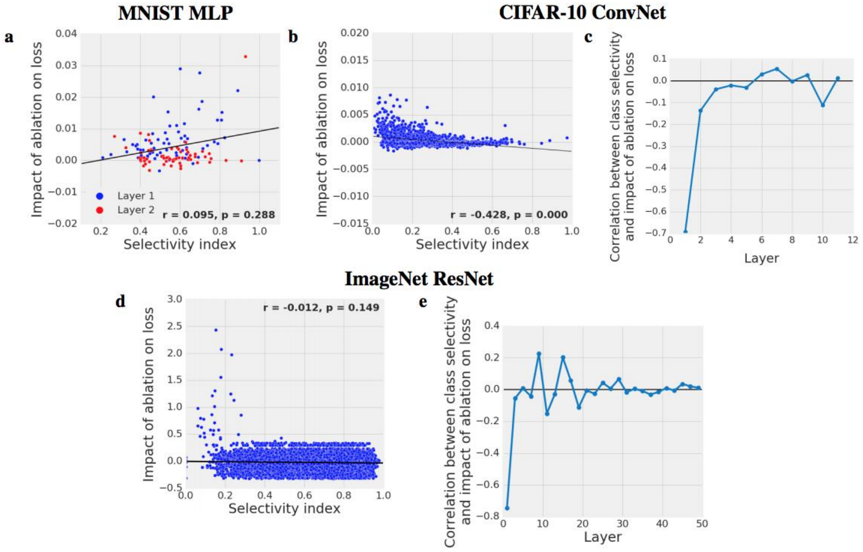


Figure 7: Selective and non-selective directions are similarly important. Impact of ablation as a function of class selectivity for MNIST MLP (**a**), CIFAR-10 convolutional network (**b-c**), and ImageNet ResNet (**d-e**). **c** and **e** show regression lines for each layer separately.

Why we think it’s interesting:

The paper shows that class selectivity of neurons is a poor predictor of task importance. It also reveals that networks that generalize well and show good performance have little dependence on individual units, overall reducing selectivity. Another interesting finding is that the commonly used batch normalization seems to implicitly discourage reliance on single unit neural selectivity. Besides, although dropout is known to serve as an effective regularizer to prevent memorization of randomized labels, the authors point out that it is unable to prevent over-reliance on single activations past the dropout fraction. This can be attributed to dropout leading to an implicit creation of redundant representation copies within the network in order to compensate loss of units.

- **Multi-Scale Dense Networks for Resource Efficient Image Classification**

Huang et al. 2017

Abstract:

“In this paper we investigate image classification with computational resource limits at test time. Two such settings are: 1. anytime classification, where the network's prediction for a test example is progressively updated, facilitating the output of a prediction at any time; and 2. budgeted batch classification, where a fixed amount of computation is available to classify a set of examples that can be spent unevenly across "easier" and "harder" inputs. In contrast to most prior work, such as the popular Viola and Jones algorithm, our approach is based on convolutional neural networks. We train multiple classifiers with varying resource demands, which we adaptively apply during test time. To maximally re-use computation between the classifiers, we incorporate them as early-exits into a single deep convolutional neural network and inter-connect them with dense connectivity. To facilitate high quality classification early on, we use a two-dimensional multi-scale network architecture that maintains coarse and fine level features all-throughout the network. Experiments on three image-classification tasks demonstrate that our framework substantially improves the existing state-of-the-art in both settings.”

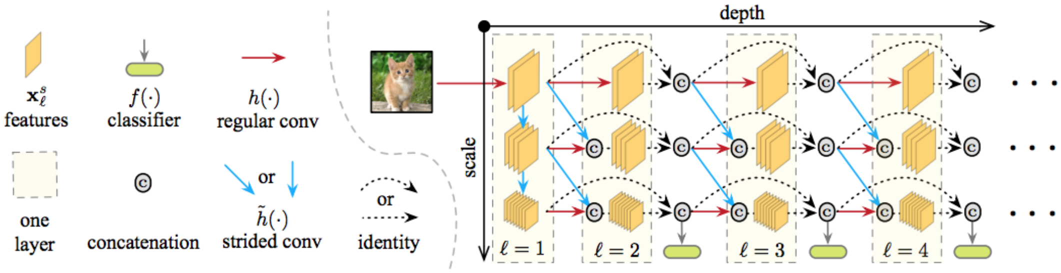


Figure 2: Illustration of the first four layers of an MSDNet with three scales. The horizontal direction corresponds to the layer direction (depth) of the network. The vertical direction corresponds to the scale of the feature maps. Horizontal arrows indicate a regular convolution operation, whereas diagonal and vertical arrows indicate a strided convolution operation. Classifiers only operate on feature maps at the coarsest scale. Connections across more than one layer are not drawn explicitly: they are implicit through recursive concatenations.

Why we think it’s interesting:

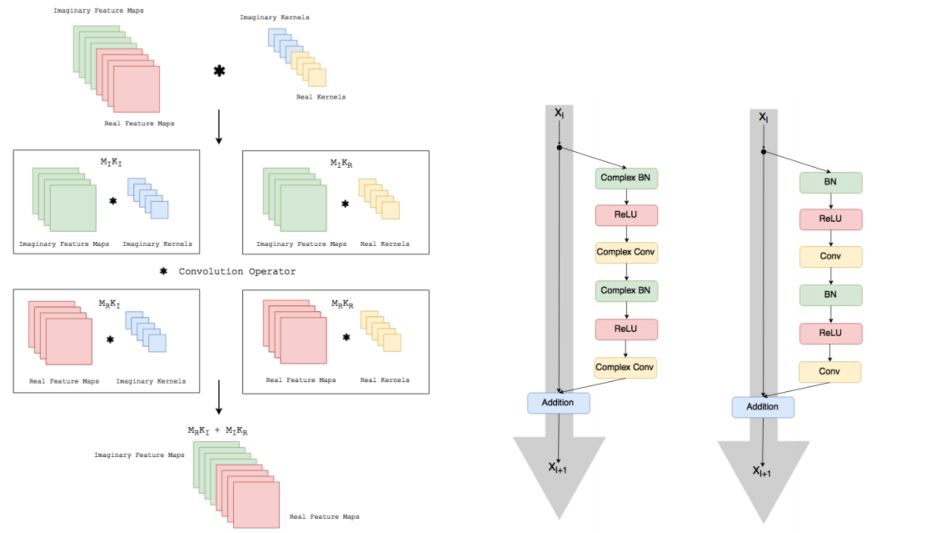
When classifying data, for many and often the majority of the samples, the classifier already at an early stage has sufficient confidence about associated class. Only for hard samples, the entire processing pipeline needs to be exercised. The paper addresses this issue by proposing escape gates that allow early termination of the pipeline. In order to be efficient and avoiding computational redundancy, a cascaded architecture is proposed.

- **Deep Complex Networks**

Trabelsi et al., 2018

Abstract:

“At present, the vast majority of building blocks, techniques, and architectures for deep learning are based on real-valued operations and representations. However, recent work on recurrent neural networks and older fundamental theoretical analysis suggests that complex numbers could have a richer representational capacity and could also facilitate noise-robust memory retrieval mechanisms. Despite their attractive properties and potential for opening up entirely new neural architectures, complex-valued deep neural networks have been marginalized due to the absence of the building blocks required to design such models. In this work, we provide the key atomic components for complex-valued deep neural networks and apply them to convolutional feed-forward networks and convolutional LSTMs. More precisely, we rely on complex convolutions and present algorithms for complex batch-normalization, complex weight initialization strategies for complex-valued neural nets and we use them in experiments with end-to-end training schemes. We demonstrate that such complex-valued models are competitive with their real-valued counterparts. We test deep complex models on several computer vision tasks, on music transcription using the MusicNet dataset and on Speech Spectrum Prediction using the TIMIT dataset. We achieve state-of-the-art performance on these audio-related tasks.”



(a) An illustration of the complex convolution operator.

(b) A complex convolutional residual network (left) and an equivalent real-valued residual network (right).

Figure 1: Complex convolution and residual network implementation details.

Why we think it’s interesting:

By expanding the available neural-networks tool-kit to the complex domain, the authors open the door to developing new models utilizing the larger representation space provided by complex-valued networks. Furthermore, they provide a more natural way to consume complex-valued input into neural-networks, e.g. audio signals.

- **Unsupervised Representation Learning by Predicting Image Rotations**

Gidaris et al., 2018

Abstract:

“Over the last years, deep convolutional neural networks (ConvNets) have transformed the field of computer vision thanks to their unparalleled capacity to learn high level semantic image features. However, in order to successfully learn those features, they usually require massive amounts of manually labeled data, which is both expensive and impractical to scale. Therefore, unsupervised semantic feature learning, i.e., learning without requiring manual annotation effort, is of crucial importance in order to successfully harvest the vast amount of visual data that are available today. In our work we propose to learn image features by training ConvNets to recognize the 2d rotation that is applied to the image that it gets as input. We demonstrate both qualitatively and quantitatively that this apparently simple task actually provides a very powerful supervisory signal for semantic feature learning. We exhaustively evaluate our method in various unsupervised feature learning benchmarks and we exhibit in all of them state-of-the-art performance. Specifically, our results on those benchmarks demonstrate dramatic improvements w.r.t. prior state-of-the-art approaches in unsupervised representation learning and thus significantly close the gap with supervised feature learning. For instance, in PASCAL VOC 2007 detection task our unsupervised pre-trained AlexNet model achieves the state-of-the-art (among unsupervised methods) mAP of 54.4% that is only 2.4 points lower from the supervised case. We get similarly striking results when we transfer our unsupervised learned features on various other tasks, such as ImageNet classification, PASCAL classification, PASCAL segmentation, and CIFAR-10 classification.“

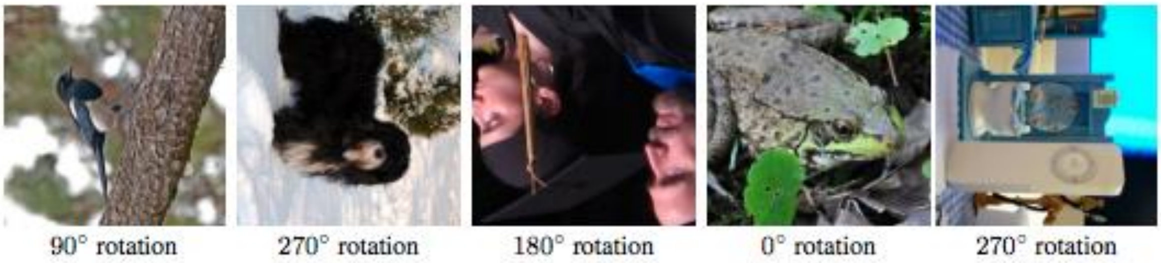


Figure 1: Images rotated by random multiples of 90 degrees (e.g., 0, 90, 180, or 270 degrees). The core intuition of our self-supervised feature learning approach is that if someone is not aware of the concepts of the objects depicted in the images, he cannot recognize the rotation that was applied to them.

Why we think it’s interesting:

In self-supervised learning, a surrogate task is learned for which big data is available. The learned feature representation is then used for the actual task, for which there is insufficient amounts of training data available. Surprisingly enough, the feature representation derived at first sight shows that meaningless tasks often turn out to be quite powerful. In the paper the authors show their approach using the task of predicting the rotation of the image (from 90 degree image rotations).

- **Twin Networks: Matching the Future for Sequence Generation**

Serdyuk et al., 2018

Abstract:

“We propose a simple technique for encouraging generative RNNs to plan ahead. We train a “backward" recurrent network to generate a given sequence in reverse order, and we encourage states of the forward model to predict cotemporal states of the backward model. The backward network is used only during training, and plays no role during sampling or inference. We hypothesize that our approach eases modeling of long-term dependencies by implicitly forcing the forward states to hold information about the longer-term future (as contained in the backward states). We show empirically that our approach achieves 9% relative improvement for a speech recognition task, and achieves significant improvement on a COCO caption generation task.”

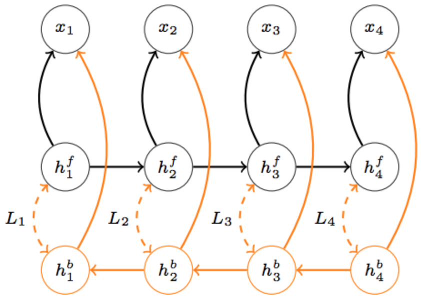


Figure 1: The forward and the backward networks predict the sequence $s = \{x_1, ..., x_4\}$ independently. The penalty matches the forward (or a parametric function of the forward) and the backward hidden states. The forward network receives the gradient signal from the log-likelihood objective as well as L_t between states that predict the same token. The backward network is trained only by maximizing the data log-likelihood. During the evaluation part of the network colored with orange is discarded. The cost L_t is either a Euclidean distance or a learned metric $||g(h_t^f) - h_t^b||_2$ with an affine transformation g. Best viewed in color.

Why we think it’s interesting:

In order to enable a generative RNN to plan ahead, a second backward RNN is applied during training in reverse direction. The authors introduce a loss based on the difference of the cotemporal hidden states, to encourage the forward RNN to anticipate future signals.

Generating Wikipedia by Summarizing Long Sequences

Liu et al., 2018

Abstract:
“We show that generating English Wikipedia articles can be approached as a multi- document summarization of source documents. We use extractive summarization to coarsely identify salient information and a neural abstractive model to generate the article. For the abstractive model, we introduce a decoder-only architecture that can scalably attend to very long sequences, much longer than typical encoder- decoder architectures used in sequence transduction. We show that this model can generate fluent, coherent multi-sentence paragraphs and even whole Wikipedia articles. When given reference documents, we show it can extract relevant factual information as reflected in perplexity, ROUGE scores and human evaluations.”

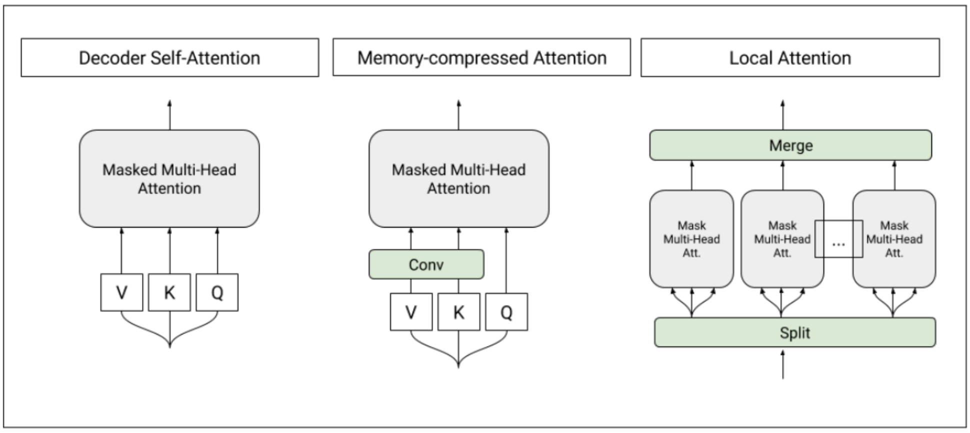


Figure 1: The architecture of the self-attention layers used in the T-DMCA model. Every attention layer takes a sequence of tokens as input and produces a sequence of similar length as the output. **Left:** Original self-attention as used in the transformer-decoder. **Middle:** Memory-compressed attention which reduce the number of keys/values. **Right:** Local attention which splits the sequence into individual smaller sub-sequences. The sub-sequences are then merged together to get the final output sequence.

Why we think it’s interesting:

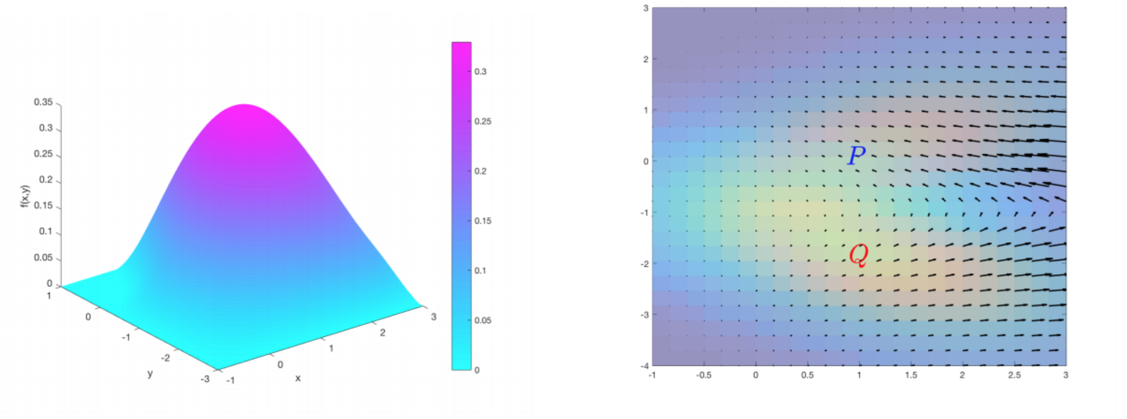
The authors of this paper have found a creative way of leveraging Wikipedia as a dataset for multi-document summarization tasks. Furthermore, they were able to train a transformer-decoder model to generate long, coherent, and meaningful texts.

- **Sobolev GAN**

Mroueh et al., 2017

Abstract:

“We propose a new Integral Probability Metric (IPM) between distributions: the Sobolev IPM. The Sobolev IPM compares the mean discrepancy of two distributions for functions (critic) restricted to a Sobolev ball defined with respect to a dominant measure µ. We show that the Sobolev IPM compares two distributions in high dimensions based on weighted conditional Cumulative Distribution Functions (CDF) of each coordinate on a leave one out basis. The Dominant measure µ plays a crucial role as it defines the support on which conditional CDFs are compared. Sobolev IPM can be seen as an extension of the one dimensional Von Mises Cramer statistics to high dimensional distributions. We show how Sobolev ´ IPM can be used to train Generative Adversarial Networks (GANs). We then exploit the intrinsic conditioning implied by Sobolev IPM in text generation. Finally we show that a variant of Sobolev GAN achieves competitive results in semi-supervised learning on CIFAR-10, thanks to the smoothness enforced on the critic by Sobolev GAN which relates to Laplacian regularization.”



(a) Numerical solution of the PDE satisfied by the optimal Sobolev critic.

(b) Optimal Sobolev Transport Vector Field $\nabla_x f^*(x)$ (arrows are the vector field $\nabla_x f^*(x)$ evaluated on the 2D grid. Magnitude of arrows was rescaled for visualization.)

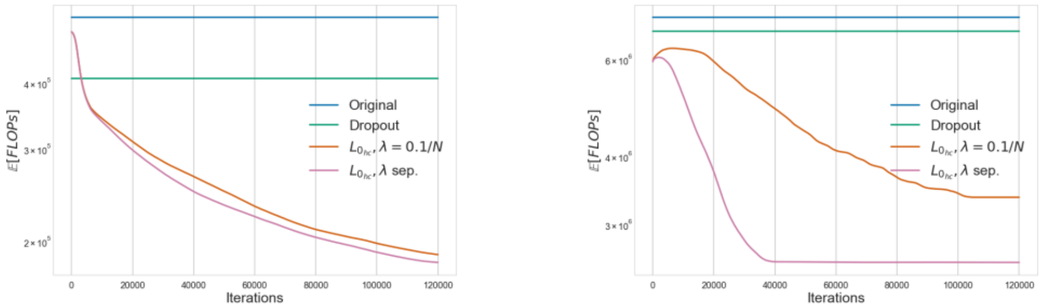
Why we think it’s interesting:

Compared to other GAN modifications such as Wasserstein that operate on the space of Lipschitz functions, Sobolov GANs are restrained to the data-dependent constraints, which has intrinsic smoothness properties. Its probability measure is based comparing weighted (coordinate-wise) conditional Cumulative Distribution Functions (CDF). It’s relation to manifold regularization in the Laplacian framework makes it potentially interesting to semi-supervised learning. Generation of discrete entities such as text still poses a challenge. Finally, the intrinsic conditioning and the CDF matching make Sobolev IPM suitable for discrete sequence matching.

- **Learning Sparse Neural Networks through L_0 Regularization**

Louizos et al., 2018

Abstract:
“We propose a practical method for L_0 norm regularization for neural networks: pruning the network during training by encouraging weights to become exactly zero. Such regularization is interesting since (1) it can greatly speed up training and inference, and (2) it can improve generalization. AIC and BIC, well-known model selection criteria, are special cases of L_0 regularization. However, since the L_0 norm of weights is non-differentiable, we cannot incorporate it directly as a regularization term in the objective function. We propose a solution through the inclusion of a collection of non-negative stochastic gates, which collectively determine which weights to set to zero. We show that, somewhat surprisingly, for certain distributions over the gates, the expected L_0 regularized objective is differentiable with respect to the distribution parameters. We further propose the \emph{hard concrete} distribution for the gates, which is obtained by ``stretching'' a binary concrete distribution and then transforming its samples with a hard-sigmoid. The parameters of the distribution over the gates can then be jointly optimized with the original network parameters. As a result our method allows for straightforward and efficient learning of model structures with stochastic gradient descent and allows for conditional computation in a principled way. We perform various experiments to demonstrate the effectiveness of the resulting approach and regularizer.”



(a) Expected FLOPs at the MLP.

(b) Expected FLOPs at LeNet5.

Figure 3: Expected number of floating point operations (FLOPs) during training for the original, dropout and L_0 regularized networks. These were computed by assuming one flop for multiplication and one flop for addition.

Why we think it's interesting:

The authors show an interesting way of reducing the model complexity by introducing a differentiable L0 norm on the weights using stochastic gates.

- **Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples**

Lee et al., 2018

Abstract:
"The problem of detecting whether a test sample is from in-distribution (i.e., training distribution by a classifier) or out-of-distribution sufficiently different from it arises in many real-world machine learning applications. However, the state-of-art deep neural networks are known to be highly overconfident in their predictions, i.e., do not distinguish in- and out-of-distributions. Recently, to handle this issue, several threshold-based detectors have been proposed given pre-trained neural classifiers. However, the performance of prior works highly depends on how to train the classifiers since they only focus on improving inference procedures. In this paper, we develop a novel training method for classifiers so that such inference algorithms can work better. In particular, we suggest two additional terms added to the original loss (e.g., cross entropy). The first one forces samples from out-of-distribution less confident by the classifier and the second one is for (implicitly) generating most effective training samples for the first one. In essence, our method jointly trains both classification and generative neural networks for out-of-distribution. We demonstrate its effectiveness using deep convolutional neural networks on various popular image datasets."

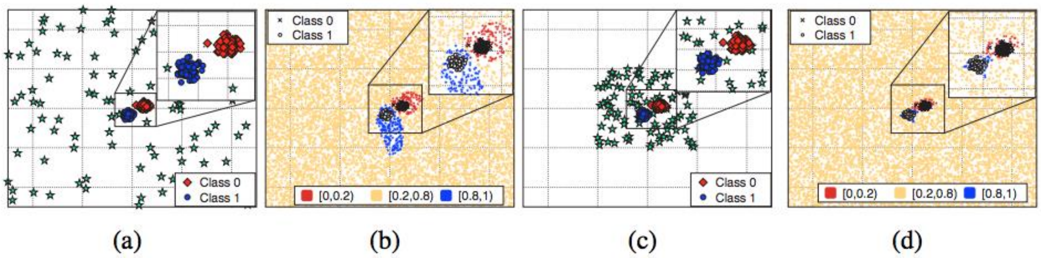


Figure 1: Illustrating the behavior of classifier under different out-of-distribution training datasets. We generate the out-of-distribution samples from (a) 2D box $[-50, 50]^2$, and show (b) the corresponding decision boundary of classifier. We also generate the out-of-distribution samples from (c) 2D box $[-20, 20]^2$, and show (d) the corresponding decision boundary of classifier.

Why we think it’s interesting:

DNNs are often mis-calibrated and overly confident in their classification. This can be attributed to mis-capturing of the out-of-distribution space, which is potentially infinitely large and thus intractable. The paper proposes a GAN that generates ‘boundary’ samples in the low-density area close to the in-distribution space (hard negatives) as well as the predictive distribution close to uniformity (max entropy).

- **Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks**

Liang et al., 2018

Abstract:

“We consider the problem of detecting out-of-distribution images in neural networks. We propose ODIN, a simple and effective method that does not require any change to a pre-trained neural network. Our method is based on the observation that using temperature scaling and adding small perturbations to the input can separate the softmax score distributions between in- and out-of-distribution images, allowing for more effective detection. We show in a series of experiments that ODIN is compatible with diverse network architectures and datasets. It consistently outperforms the baseline approach by a large margin, establishing a new state-of-the-art performance on this task. For example, ODIN reduces the false positive rate from the baseline 34.7% to 4.3% on the DenseNet (applied to CIFAR-10) when the true positive rate is 95%.”

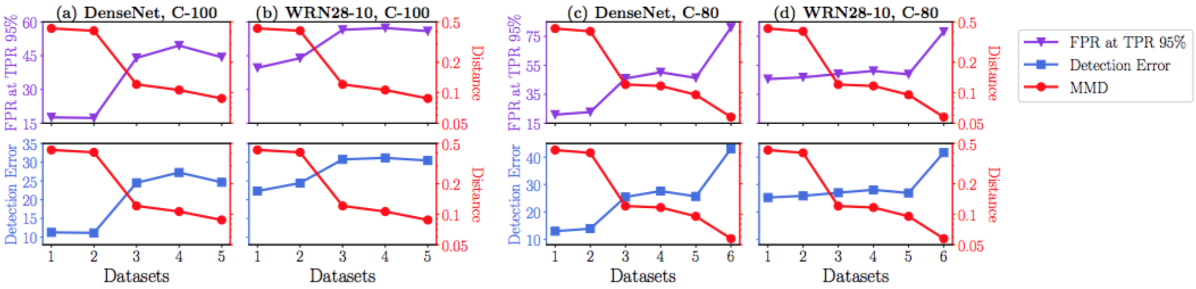


Figure 2: (a)-(d) Performance of our method vs. MMD between in- and out-of-distribution datasets. Neural networks are trained on CIFAR-100 and CIFAR-80, respectively. The out-of-distribution datasets are 1: LSUN (cop), 2: TinyImageNet (crop), 3: LSUN (resize), 4: is iSUN (resize), 5: TinyImageNet (resize) and 6: CIFAR-20.

Why we think it's interesting:

The problem of detecting out-of-distribution samples is strongly related to mis-calibration of DNNs and the resulting property of over-confidence. As the out-of-distribution space is typically infinitely large, it is hard to model without running into tractability issues. To tackle this problem, the paper proposes softmax temperature scaling as well as input data perturbation, leading to a better separation of distributions.

- **Measuring the Intrinsic Dimension of Objective Landscapes**

Li et al., 2018

Abstract:

"Many recently trained neural networks employ large numbers of parameters to achieve good performance. One may intuitively use the number of parameters required as a rough gauge of the difficulty of a problem. But how accurate are such notions? How many parameters are really needed? In this paper we attempt to answer this question by training networks not in their native parameter space, but instead in a smaller, randomly oriented subspace. We slowly increase the dimension of this subspace, note at which dimension solutions first appear, and define this to be the intrinsic dimension of the objective landscape. The approach is simple to implement, computationally tractable, and produces several suggestive conclusions. Many problems have smaller intrinsic dimensions than one might suspect, and the intrinsic dimension for a given dataset varies little across a family of models with vastly different sizes. This latter result has the profound implication that once a parameter space is large enough to solve a problem, extra parameters serve directly to increase the dimensionality of the solution manifold. Intrinsic dimension allows some quantitative comparison of problem difficulty across supervised, reinforcement, and other types of learning where we conclude, for example, that solving the inverted pendulum problem is 100 times easier than classifying digits from MNIST, and playing Atari Pong from pixels is about as hard as classifying

CIFAR-10. In addition to providing new cartography of the objective landscapes wandered by parameterized models, the method is a simple technique for constructively obtaining an upper bound on the minimum description length of a solution. A byproduct of this construction is a simple approach for compressing networks, in some cases by more than 100 times.”

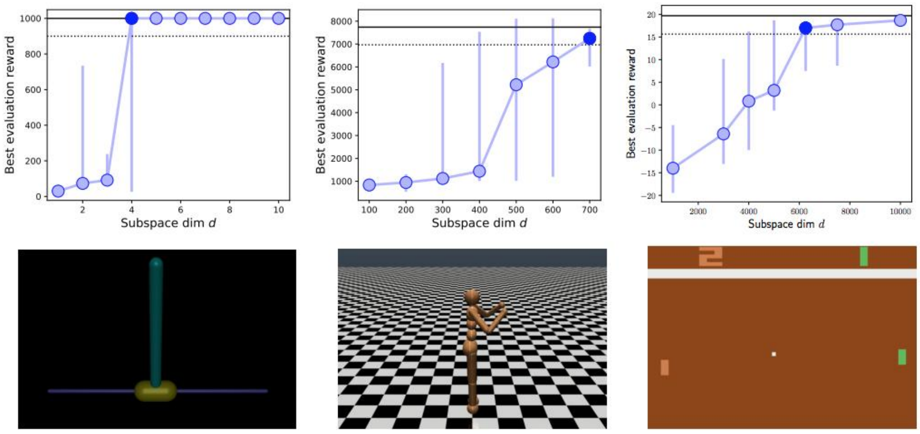


Figure 5: Results using the policy-based ES algorithm to train agents on **(left column)** `InvertedPendulum−v1`, **(middle column)** `Humanoid−v1`, and **(right column)** `Pong−v0`. The intrinsic dimensions found are 4, 700, and 6k. This places the walking humanoid task on a similar level of difficulty as modeling MNIST with a FC network (far less than modeling CIFAR-10 with a convnet), and Pong on the same order of modeling CIFAR-10.

Why we think it’s interesting:

The paper proposes a simple solution answering the challenge to assess the difficulty of a task. The intrinsic dimensionality of the problem is correlated to the minimum number of parameters of a DNN that provides a solution. The proposed approach is iterative and starts from a small number.

- **FearNet: Brain-Inspired Model for Incremental Learning**

Kemker & Kanan, 2018

Abstract:
“Incremental class learning involves sequentially learning classes in bursts of examples from the same class. This violates the assumptions that underlie methods for training standard deep neural networks, and will cause them to suffer from catastrophic forgetting. Arguably, the best method for incremental class learning is iCaRL, but it requires storing training examples for each class, making it challenging to scale. Here, we propose FearNet for incremental class learning. FearNet is a generative model that does not store previous examples, making it memory efficient. FearNet uses a brain-inspired dual-memory system in which new memories are consolidated from a network for recent memories inspired by the mammalian hippocampal complex to a network for long-term storage inspired by medial prefrontal cortex. Memory consolidation is inspired by mechanisms that occur during sleep. FearNet also uses a module inspired by the basolateral amygdala for determining which memory system to use for recall. FearNet achieves state-of-the-art performance at incremental class learning on image (CIFAR-100, CUB-200) and audio classification (AudioSet) benchmarks.”

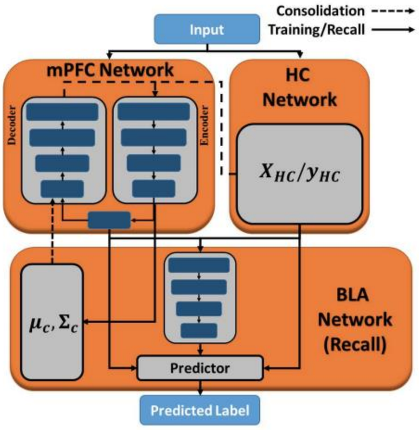


Figure 1: FearNet consists of three brain-inspired modules based on 1) mPFC (long-term storage), 2) HC (recent storage), and 3) BLA for determining whether to use mPFC or HC for recall.

Why we think it's interesting:

FearNet uses a brain-inspired dual-memory system for long and short-term storage. Motivated by memory replay during sleep, FearNet employs a generative autoencoder for pseudorehearsal, which mitigates catastrophic forgetting by generating previously learned examples that are replayed alongside novel information during consolidation.

- **AmbientGAN: Generative models from lossy measurements**

Bora et al., 2018

Abstract:

"Generative models provide a way to model structure in complex distributions and have been shown to be useful for many tasks of practical interest. However, current techniques for training generative models require access to fully-observed samples. In many settings, it is expensive or even impossible to obtain fully-observed samples, but economical to obtain partial, noisy observations. We consider the task of learning an implicit generative model given only lossy measurements of samples from the distribution of interest. We show that the true underlying distribution can be provably recovered even in the presence of per-sample information loss for a class of measurement models. Based on this, we propose a new method of training Generative Adversarial Networks (GANs) which we call AmbientGAN. On three benchmark datasets, and for various measurement models, we demonstrate substantial qualitative and quantitative improvements. Generative models trained with our method can obtain \$2\$-\$4\$x higher inception scores than the baselines."

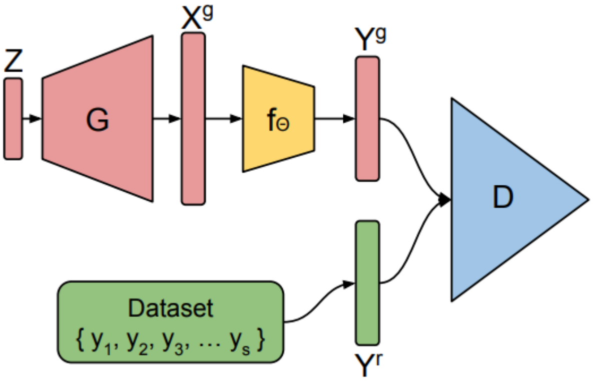


Figure 1: AmbientGAN training. The output of the generator is passed through a simulated random measurement function f_Θ. The discriminator must decide if a measurement is real or generated.

Why we think it’s interesting:

The proposed approach tackles the problem of learning with noisy data. Rather than distinguishing a real image from a generated image as in a traditional GAN, in the AmbientGAN approach the discriminator must distinguish a real measurement from a simulated measurement of a generated image. This permits working with very noisy data, however, requires knowledge of the noise process.

- **A moth brain learns to read MNIST**

Delahunt & Kutz, 2018

Abstract:
“We seek to characterize the learning tools (ie algorithmic components) used in biological neural networks, in order to port them to the machine learning context. In particular we address the regime of very few training samples. The Moth Olfactory Network is among the simplest biological neural systems that can learn. We assigned a computational model of the Moth Olfactory Network the task of classifying the MNIST digits. The moth brain successfully learned to read given very few training samples (1 to 20 samples per class). In this few-samples regime the moth brain substantially outperformed standard ML methods such as Nearest-neighbors, SVM, and CNN. Our experiments elucidate biological mechanisms for fast learning that rely on cascaded networks, competitive inhibition, sparsity, and Hebbian plasticity. These biological algorithmic components represent a novel, alternative toolkit for building neural nets that may offer a valuable complement to standard neural nets.”

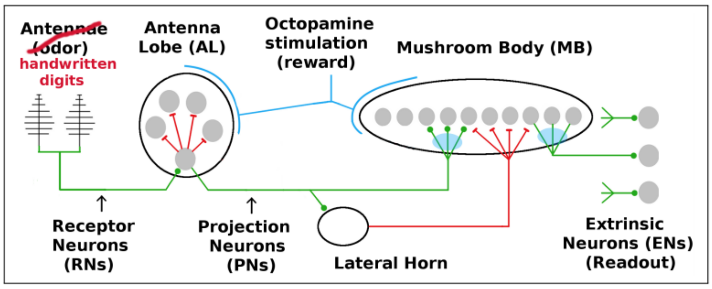

~~Antennae (odor)~~ handwritten digits
Antenna Lobe (AL)
Octopamine stimulation (reward)
Mushroom Body (MB)
Receptor Neurons (RNs)
Projection Neurons (PNs)
Lateral Horn
Extrinsic Neurons (ENs) (Readout)

Figure 1: **Network schematic.** Green lines show excitatory connections, red lines show inhibitory connections. Light blue ovals show plastic connections into and out of the MB. The processing units in the AL competitively inhibit each other. Global inhibition from the lateral horn induces sparsity on MB responses. The ENs give the final, actionable readouts of the system's response to a stimulus.

Why we think it's interesting:

This paper introduces a biologically inspired simple network, capable of few-shot and even single-shot learning.

- **Shifting Mean Activation Towards Zero with Bipolar Activation Functions**

Eidnes & Nøkland, 2018

Abstract:

"We propose a simple extension to the ReLU-family of activation functions that allows them to shift the mean activation across a layer towards zero. Combined with proper weight initialization, this alleviates the need for normalization layers. We explore the training of deep vanilla recurrent neural networks (RNNs) with up to 144 layers, and show that bipolar activation functions help learning in this setting. On the Penn Treebank and Text8 language modeling tasks we obtain competitive results, improving on the best reported results for non-gated networks. In experiments with convolutional neural networks without batch normalization, we find that bipolar activations produce a faster drop in training error, and results in a lower test error on the CIFAR-10 classification task."

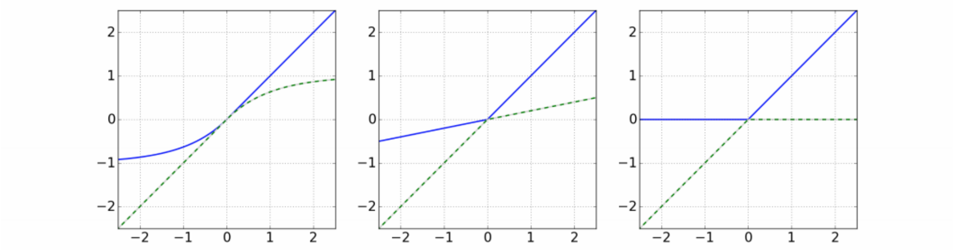


Figure 1: Bipolar versions of popular activation functions. From left: Bipolar ELU, Bipolar Leaky ReLU, Bipolar ReLU.

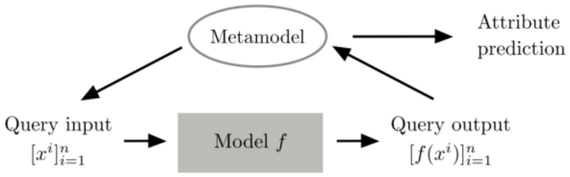
Why we think it’s interesting:

With a simple and easy to implement trick which shifts the mean activation across a layer to zero, the authors were able to improve the performance of neural networks using ReLu and/or ELU neurons.

- **Towards Reverse-Engineering Black-Box**

Joon Oh et al., 2018

Abstract:
“Many deployed learned models are black boxes: given input, returns output. Internal information about the model, such as the architecture, optimisation procedure, or training data, is not disclosed explicitly as it might contain proprietary information or make the system more vulnerable. This work shows that such attributes of neural networks can be exposed from a sequence of queries. This has multiple implications. On the one hand, our work exposes the vulnerability of black-box neural networks to different types of attacks -- we show that the revealed internal information helps generate more effective adversarial examples against the black box model. On the other hand, this technique can be used for better protection of private content from automatic recognition models using adversarial examples. Our paper suggests that it is actually hard to draw a line between white box and black box models.”



Why we think it’s interesting:

The new metamodel methods presented, called *kennen*, can successfully predict attributes of trained neural-networks (black-boxes) related not only to the architecture but also to training hyperparameters.

CONCLUSION

Overall, ICLR was a smaller and cozier counterpart of other top machine learning conferences such as NIPS or CVPR. During the five conference days, a broad range of topics were covered and we were introduced to various new research approaches as well as practical applications for industry. Besides the focus on reinforcement learning and generative models, there was an emphasis on new CNN architectures. Also, in view of the vast progress in computer vision, the relatively slow development in healthcare was basis for discussion along with the various reasons behind such as data privacy. Lastly, the conference reinforced the issue of reproducibility in machine learning uncovering a strong need for improvement in this area.

www.sap.com/contactsap

THE BEST RUN SAP®