Neural Data Augmentation

Afonso Eduardo (s1740192@sms.ed.ac.uk), Parvez Alam Kazi (s178089@sms.ed.ac.uk)

Abstract

Deep neural networks achieve state-of-art performance in a wide range of applications, but typically require a large amount of data. Traditional methods mitigate the problem of small training sets by augmenting the data using known transformations. However, it is believed that plausible transformations can be learned automatically. We analyze the problem of training deep neural network classifiers on comparatively small data sets, exploring neural generative models as means to learn these transformations (neural data augmentation). In particular, we train conditional deep convolutional variants of Generative Adversarial Networks (CDCGAN) and Variational Auto-encoders (CDCVAE) on data sets based on CIFAR10 and MNIST. We qualitatively and quantitatively analyze the resulting data generation processes, finding that quality tends to improve as the training set size increases. Better architectures and models are however required for the generation of natural images. We then evaluate the performance of a VGG-Net classifier with and without data augmentation and find that neural data augmentation does not seem to be better than traditional methods, but it is likely due to the relatively simple architectures we consider.

1. Introduction

Deep learning has revolutionized several disciplines (LeCun et al., 2015), ranging from computer vision to speech recognition. Indeed, deep neural networks seem to excel at perception tasks, and the emergence of open-source deep learning frameworks, such as Theano (Theano Development Team, 2016), TensorFlow (Abadi et al., 2015) or Keras (Chollet et al., 2015), has brought the field into the mainstream. However, in order to achieve state-of-art performance, the architectures of these neural networks are typically complex and a recent trend is to consider increasingly deeper designs (Gu et al., 2017). This poses a number of challenges as not only the training procedure becomes computationally expensive, but also requires the collection of large data sets. The former is typically solved by GPU-accelerated computing, allowing significant speedups when compared to standard computing. The latter is however exceptionally difficult or, in some cases, impossible (e.g. medical data). One possible alternative is to consider data augmentation, where the original data set is enlarged using new artificial observations. Traditional strategies involve the use of transformations specified a priori, but it is believed that plausible transformations can instead be learned from the data without relying on human intervention.

In this work, we draw inspiration from neural generative models, specifically variants of Generative Adversarial Network (GAN) (Goodfellow et al., 2014a) and Variational Auto-encoder (VAE) (Kingma & Welling, 2013), in order to automatically learn possible transformations and to generate synthetic observations. This approach is henceforth referred to as neural data augmentation. Indeed, we believe that these black-box generative models contain enough representational power as to learn plausible transformations while, in principle, requiring minimal assumptions from the practitioner regarding the data. However, since the general procedure is not limited to learning the parameters of the generative distribution, but also its implicit and explicit functional form (GAN and VAE respectively), their applicability might be limited, especially in the low data regime. To this end, we investigate to which extent data augmentation using variants of GANs and VAEs can be an effective strategy, providing a qualitative and quantitative analysis of the quality and robustness of the data generation processes. In order to evaluate their suitability, we also compare them to traditional methods. Our primary aim is then to measure the consequences in terms of classification performance (accuracy) under different data regimes. For this purpose, we evaluate the behavior of a deep neural network classifier, a simpler VGG-Net (Simonyan & Zisserman, 2014), on increasingly smaller data sets with and without data augmentation. Note that in previous work, we had also considered a different classifier based on capsule layers (Sabour et al., 2017), but we do not explore this classifier further, mostly due to computational constraints. Similarly, we had considered as an optional objective the improvement of neural generative models using, for instance, architectures based on capsule layers, but, again, due to time and computational constraints we do not pursue this direction, leaving it instead as future work.

The rest of this report is organized as follows: in Section 2, we present the data sets, followed by the specification of the classifier and the traditional and neural data augmentation strategies. In addition, we describe the method we use to quantitatively assess the quality of the artificial data. Section 3 provides a description of the experiments and a detailed analysis of the results. Section 4 reviews related approaches in the literature and, finally, Section 5 summarizes our findings, relating the outcome to the research questions while also pointing to possible future directions.

2. Methodology

2.1. Data sets

Focusing on image recognition, we explore data sets with the same number of classes (10): MNIST (LeCun et al., 1998) and CIFAR10 (Krizhevsky & Hinton, 2009). In MNIST, each sample is a 28×28 gray-scale image of a handwritten digit; whereas, in CIFAR10, each observation is a 32×32 color image of a vehicle (airplane, automobile, ship, truck) or an animal (bird, cat, deer, dog, frog, horse). Both contain 10,000 images as test data, but while the classes in the test set of CIFAR10 are balanced, on MNIST they are only approximately so. In an attempt to balance the training/validation sets and make MNIST and CIFAR10 comparable with respect to the training set size, we consider a different split. In particular, we consider stratified sampling, generating training sets of size 42,500 and using the remainder observations for validation purposes (17,500 and 7,500 for MNIST and CIFAR10 respectively). We achieve a perfect class balance in CIFAR10, but not in MNIST. However, compared to the given split, this constitutes an improvement. More importantly, we consider training subsets of different sizes: we build versions with 90%, 80%, ..., 10%, 5%, 2% and 1% of the full training set (100%), referring to these as MNIST-α (CIFAR10-α), where α denotes a relative percentage. These versions are generated by successive stratified sampling. For instance, MNIST-90 is a subset of MNIST-100, MNIST-80 is a subset of MNIST-90, and so forth. Ideally, experiments should be performed over multiple random splits so that the results are not influenced by specific splits, but we are unable to follow this route due to computational constraints.

2.2. VGG-Net Classifier

Convolutional neural networks (CNN) have been highly successful (Schmidhuber, 2015) and, currently, the best performing architectures consider advanced layer designs, including shortcut connections to allow more effective training (Gu et al., 2017). Historically, a major development was proposed by (Simonyan & Zisserman, 2014), where state-of-art results were achieved using a homogeneous architecture containing small-sized filters (VGG-Net). This aspect revealed to be particularly effective, allowing the incorporation of more non-linearities, while reducing the number of weights. In this work, we consider simpler variants of the VGG-Net. The architectures have 3 convolutional stages with 64, 128 and 256 filters respectively. Each block consists of two convolutional layers with 3×3 kernels and weights are initialized with Glorot Uniform (Glorot & Bengio, 2010). We use Leaky ReLU and, after each convolutional operation, we optionally apply batch normalization (Ioffe & Szegedy, 2015). In particular, based on previous findings, we enable batch normalization on CIFAR10 data, but not on MNIST. Dimensionality reduction at each stage is carried out using max-pooling (2×2, stride 2). After each block, we apply dropout (Srivastava et al., 2014) with a rate of 0.6. Finally, for classification, a single fully-connected layer with softmax is added.

2.3. Traditional Data Augmentation

In computer vision, the use of data augmentation includes the application of rotation, translation, blurring and other transformations to existing images, often allowing a model to generalize better. Traditional (manual) augmentation techniques have been studied extensively in the context of neural networks and their use usually leads to better performance (Simard et al., 2003; Krizhevsky et al., 2012; Chatfield et al., 2014). In this work, for both MNIST and CIFAR10, we consider three possible transformations: random rotations (ROT) uniformly chosen between $-25°$ to $25°$, Gaussian blur (BLUR) injected into images with standard deviation chosen randomly between 0.25 and 2.25 and the combination of the two previous transformations (ROT-BLUR). Note that rotations beyond the interval chosen for ROT can cause problems during classification on MNIST: consider, for instance, the digits 6 and 9. In BLUR, higher standard deviations yield images that are too blurry and hence deemed not suitable for data augmentation.

2.4. Neural Data Augmentation

Adopting the idea of neural generative models as black box data augmentation strategies, we explore variants of Generative Adversarial Network (GAN) and Variational Auto-Encoder (VAE). In particular, as we aim to augment the existing data sets with labelled artificial data, we consider conditional versions, where synthetic observations are generated conditioned on a particular class. Additionally, we adopt deep convolutional versions because our experiments involve improving the classification performance on images, for which convolutional architectures are more suitable. That said, our generative models are Conditional Deep Convolutional (CDC) variants of GAN and VAE, and are henceforth referred to as CDCGAN and CDCVAE.

2.4.1. CDCGAN

In the adversarial framework (Goodfellow et al., 2014a), a generative model attempts to learn the true distribution of the data and, simultaneously, generates samples, feeding them to an adversarial discriminative model. This model, in turn, learns to determine whether the data being fed is fake or real, i.e. whether the data belongs to the true distribution. In game theory, this type of adversarial setting can be modelled as a minimax game. Generative Adversarial Networks (GAN) are a special case of this framework, where both models are given by neural networks. The first network corresponds to the generator $G(\boldsymbol{z})$, which learns a mapping from input noise $\boldsymbol{z}$ to the data space. The second network is the discriminator $D(\boldsymbol{x})$, a classifier that outputs the probability of incoming data $\boldsymbol{x}$ belonging to the true distribution and not the generator. The resulting interaction is described by the value function $V(D, G)$ (Goodfellow et al., 2014a):

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))], \quad (1)$$

where $p_{data}(\boldsymbol{x})$ denotes the empirical data distribution defined over input samples $\boldsymbol{x}$ and $p(\boldsymbol{z})$ is the prior over input noise variables, e.g. uniform over the interval $(-1, 1)$. In practice, G is trained by maximization of $\log(D(G(\boldsymbol{z})))$ because it leads to a more stable training procedure. At the equilibrium (Nash equilibrium), G approximately models the true data distribution and D outputs the same probability (0.5) for real and generated data. Furthermore, conditional generative models can be used to learn multimodal representations, which are useful for generating descriptive image labels and, in our case, for data augmentation using labelled data. A conditional model is obtained if both the generator and discriminator are conditioned on some extra information $\boldsymbol{y}$, e.g. class label. This results in a small change to the original objective (Mirza & Osindero, 2014):

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x} \mid \boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z} \mid \boldsymbol{y})))]. \tag{2}$$

Note that, in practice, the conditioning is implemented by concatenating the original input with the extra information. In terms of implementation and by definition of a GAN, the specification of two networks is required (G and D). For this purpose, we adopt CNNs as they lead to models that are good at identifying the spatial structure in images. However, due to several constraints, rather than adopting a more sophisticated architecture as in (Radford et al., 2015) or (He et al., 2016), we use the simpler architectures found in InfoGAN (Chen et al., 2016a). In particular, for MNIST, we use the architecture described in appendix C.1. of (Chen et al., 2016a) and, for CIFAR10, we adapt the architecture only in terms of input and output dimensions of each layer. In fact, we take as reference the implementation found in (hwalsuklee, 2018), using a latent representation $\boldsymbol{z}$ of size 62 for MNIST and 100 for CIFAR10.

Finally, it should be mentioned that one of the most notorious problems in GANs is related to unstable training and non-convergence. While most deep models are trained by minimizing a cost function using gradient descent, GANs are trained by solving a minimax game. The equilibrium is not guaranteed as, in some cases, the two adversaries tend to undo the progress of each other. In this context, (Goodfellow et al., 2014b) shows that simultaneous gradient descent converges if the updates are made in function space, but in practice, the updates are made in parameter space. Not surprisingly, improving the training dynamics for GANs is an active research area. Some recent contributions include (Srivastava et al., 2017) proposing a GAN with a reconstructor network and (Arjovsky et al., 2017) proposing the use of Wasserstein-1 distance. Also refer to (Salimans et al., 2016) for additional techniques.

2.4.2. CDCVAE

An auto-encoder network can be viewed as pair of two connected networks, an encoder and a decoder. The encoder takes an input and converts it into a more compact representation, which the decoder uses to reconstruct the original representation. The entire network is trained as a whole and the loss function can be for instance the mean-squared error or the cross-entropy between the output and the input. This is known as the reconstruction loss, penalizing the network for generating outputs different from the input. As the encoder is constrained to learn a lower-dimensional representation, it learns to preserve the most relevant information. However, the fundamental problem with standard auto-encoders is that the latent space is learned from the direct mapping of observed inputs, which for generative purposes (generalization) is problematic.

On the other hand, Variational Auto-encoders (VAE) (Kingma & Welling, 2013) are full probabilistic auto-encoders that include an encoder $q(\boldsymbol{z}|\boldsymbol{x}; f(\boldsymbol{x}; \boldsymbol{\theta}))$ and a decoder (also seen as generator) $p(\boldsymbol{x}|\boldsymbol{z}; u(\boldsymbol{h}; \boldsymbol{\zeta})$, where $\boldsymbol{x}$ corresponds to the observation, $\boldsymbol{z}$ to the latent representation and f and u are neural networks, with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$, whose outputs paramaterize the distributions q and p respectively. The traditional VAE further assumes normally and independently distributed latent variables, given by the prior $p(\boldsymbol{z})$, and a fully-factorized likelihood given by the decoder. Similarly, the authors assume a fully-factorized distribution for the encoder. Using these assumptions, for a N-sized observation, the generative model (prior and probabilistic decoder) and the probabilistic encoder can be written as:

$$p(\boldsymbol{x}, \boldsymbol{z}; u(\cdot; \boldsymbol{\zeta})) = \prod_{n=1}^{N} \mathcal{N}(z_n; 0, 1)\, p(x_n|z_n; u(z_n; \boldsymbol{\zeta})), \tag{3}$$

$$q(\boldsymbol{z}|\boldsymbol{x}; f(\boldsymbol{x}; \boldsymbol{\theta})) = \prod_{n=1}^{N} q(z_n|x_n; f(x_n; \boldsymbol{\theta})), \tag{4}$$

where in Equation 4 each factor is a Gaussian with mean and variance given by the outputs of network f. Note also that, for images (with pixel intensities between 0 and 1), p can be considered Bernoulli and, consequently, the output of u corresponds to the probability parameter. The parameters of the networks $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$ can then be learned by minimizing the evidence lower bound (ELBO) (Kingma & Welling, 2013), a lower bound on the marginal likelihood of the data:

$$\mathcal{L}_{ELBO} = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[-KL(q(\boldsymbol{z} \mid \boldsymbol{x}; f) \parallel p(\boldsymbol{z}))] + \mathbb{E}_{\boldsymbol{z},\boldsymbol{x}\sim q(\boldsymbol{z}|\boldsymbol{x})p_{data}(\boldsymbol{x})}[\log\ p(\boldsymbol{x} \mid \boldsymbol{z}; u)], \tag{5}$$

where $p_{data}(\boldsymbol{x})$ is the empirical data distribution and KL denotes the Kullback-Leibler divergence, a measure that determines the degree of dissimilarity between distributions. Hence, the first term forces the distribution given by the probabilistic decoder to be close to the prior and the second term quantifies the reconstruction loss. However, note that we are interested in a conditional version which, as before, consists in further conditioning the encoder and decoder on additional information (class label) (Sohn et al., 2015). In terms of implementation, this involves again concatenating the original input with the extra information. Furthermore, in order to allow a fair comparison, in our Conditional Deep Convolutional VAE (CDCVAE), the decoder uses the same

architecture as the generator network in CDCGAN and the encoder uses an architecture that is similar to that of the discriminator network in CDCGAN. Latent representation dimensions are also the same. The fundamental difference is that the encoder, by design, has two outputs: the first is the unconstrained mean (linear activation) and the second is the positive variance (softplus activation).

Finally, notice that, as in CDCGAN, we consider relatively simple architectures and model, mostly due to computational constraints. Recent work has however proposed several improvements over traditional VAE. For instance, the assumption of continuous and independent latent variables can be relaxed (Rolfe, 2016; Johnson et al., 2016) and more expressive distributions for the encoder can be found by normalizing flows (Rezende & Mohamed, 2015; Kingma et al., 2016), which consist in the the successive application of a series of invertible deterministic transformations.

2.5. Quantitative Assessment

Many advances in deep generative models have been driven based on the qualitative assessment of images. However, this assessment is clearly sub-optimal due to its subjective nature and because it requires human intervention. In general, for models where the direct evaluation of the marginal likelihood $p(\boldsymbol{x})$ is possible, a common strategy is to compute this quantity on held-out data and choose the model that leads to the highest likelihood (Barratt & Sharma, 2018). However, in recent generative models this may not be possible: VAEs only provide a lower bound approximation (ELBO), which tends not to be tight, and GANs are models where the likelihood is defined implicitly. In these circumstances, especially GANs, the most practical approach is to compute a metric based on samples from the model.

One possibility is given by the Inception Score (IS) (Salimans et al., 2016). Summarily, this metric is determined by applying a pre-trained neural network, specifically an Inception v3 network trained on ImageNet (Szegedy et al., 2015), to generated images and computing statistics of its output. The obtained densities are the conditional label distribution $p(y \mid \boldsymbol{x})$, which measures meaningful objects when entropy is low (informative distribution), and the marginal label distribution $p(y)$, quantifying the ability of the generative model to generate varied images when entropy is high (least informative, close to uniform). The metric is then determined according to $\exp(\mathbb{E}_{\boldsymbol{x}} KL(p(y \mid \boldsymbol{x}) \parallel p(y)))$. However, IS has been shown in (Heusel et al., 2017) to suffer from a number of pathological cases, where the images become subjectively worse, but the score remains flat. As a result, the authors propose the use of Fréchet Inception Distance (FID), measuring the difference between two Gaussians:

$$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + Tr(\boldsymbol{C} + \boldsymbol{C}_w - 2\sqrt{\boldsymbol{C}\boldsymbol{C}_W}), \quad (6)$$

where the mean and covariance $(\boldsymbol{m}_w, \boldsymbol{C}_w)$ are found by computing the respective statistics over the activations of the coding layer pool_3 of the Inception v3 network when fed real images, and $(\boldsymbol{m}, \boldsymbol{C})$ are found by feeding synthetic images to the network. A reference implementation can be found in (Heusel, 2018).

It is also worth mentioning that quantitative analysis of generative models has been a fraught topic. For instance, (Theis et al., 2015) demonstrated that commonly used evaluation criteria tend to be uncorrelated when the data is high-dimensional. Good performance using one measure does not necessarily imply good performance using another. As a result, this topic deserves more attention from the research community and the discussion in (Barratt & Sharma, 2018) can serve as an informative guideline for future work.

3. Experiments

Before proceeding with the description and analysis of the experiments, we note that, as in previous work, the networks (neural generative models and classifiers) are optimized using Adam (Kingma & Ba, 2014). The neural generative models use a $\beta_1 = 0.5$ and the discriminator in CDCGAN is trained with a learning rate of 0.0002. These are the provided hyperparameters in the implementation we take as reference (hwalsuklee, 2018). All other hyperparameters, including those that are used to train the VGG-Net classifiers, are kept as default. It is possible that these hyperparameters do not correspond to optimal configurations, but due to computational and time constraints we do not test other profiles. Furthermore, and as briefly noted in previous work, we believe that despite its popularity, Adam appears to have a few shortcomings, in part due to exponential moving averages. Recent stochastic optimization methods, such as AMSGrad (Reddi et al., 2018), aim to fix these problems and, consequently, this research direction might be worth pursuing in future work.

We begin our experiments by training both CDCGAN and CDCVAE, followed by an analysis of the training process. This is a fundamental step in our work as these trained models generate the images that we use to augment the original data sets. In particular, for each possible training set, i.e. CIFAR-α (or MNIST-α) with α taking 13 possible values from $1, 2, 5, 10, ...90, 100$, we train CDCGAN and CDCVAE independently for 500 epochs as this seems to be a good compromise between training time and quality of the generated data. Due to space constraints, we do not include the plots of the training losses. However, it is worth pointing that the loss of the CDCVAE, negative evidence lower bound (ELBO), tends in most cases to converge to local minima within 200 to 300 epochs for MNIST and 300 to 400 epochs for CIFAR10. It is not surprising that convergence takes longer for CIFAR10 because it has more complex data. Additionally, convergence is reached within fewer epochs for larger training sets (epochs take longer) and the minimum value of the ELBO becomes smaller. This latter observation is also unsurprising since the ELBO corresponds to a lower bound on the log marginal likelihood and

Figure 1. Synthetic image samples from neural generative models, CDCGAN (left) and CDCVAE (right). The models have been trained independently on both CIFAR10 (top) and MNIST (bottom) using multiple training sets.

larger training sets tend to exhibit more variability, some of which the generative model is not able to explain. Other noteworthy observations include that the optimization process becomes noisier in the presence of less training data and this effect becomes more pronounced as the data complexity increases (CIFAR10). On the other hand, the losses exhibited by the CDCGAN (discriminator and generator) are less straightforward to analyze. The training process is significantly noisier than that of CDCVAE, exhibiting particularly wild fluctuations in the low data regime. The training process tends to be more stable (less noisy) as the training sets become larger, but even in these cases convergence is typically not reached due to the difficulty in finding equilibrium in the minimax GAN game (Nash equilibrium). In fact, unlike VAEs, generative models based on GANs are known to suffer from training stability problems and, as previously mentioned, recent work tries to address some of these shortcomings, using techniques such as feature matching and minibatch features (Salimans et al., 2016), alternative objectives that are continuous and differentiable with respect to the parameters of the generator such as the Wasserstein-1 distance (Arjovsky et al., 2017; Gulrajani et al., 2017) or using an additional reconstructor network (Srivastava et al., 2017). All these approaches have been shown to lead to better convergence properties and, consequently, generate images that are of higher quality. Indeed, some of these modifications have been alluded in previous work as being completely optional objectives, but due to several constraints we have not been able to test them thoroughly. Instead, we leave them as a possible avenue for future experimental work.

The next step is to generate images from the trained CDCGAN and CDCVAE models (2×13 trained models). Before measuring classification performance, it is also important to analyze the data generation process itself as this provides additional insight into how robust these generative models may be under different scenarios, namely different data regimes. In particular, we generate a number of images that is equal to the size of the corresponding training sets, forcing the number of samples per class to be the same, and provide a qualitative analysis of the generated samples. Figure 1 provides some examples of the images that have been generated using CDCGAN (left) and CDCVAE (right) at different training regimes, specifically 10%, 30%, 50%, 80% and 100%. In both cases, the overall quality tends to improve as the training set size increases, e.g. the images become less noisier. This applies to both CIFAR10 and MNIST, but admittedly this trend might be more difficult to identify in CIFAR10 because all generated images, including those trained on the 100% version, are not able to replicate the global structure and perspective found in natural images. Presumably, these defects can be solved by adopting better model architectures, using for instance a combination of UNet, ResNet and DenseNet as done in (Antoniou et al., 2017), and, as previously discussed, in the case of CDCGAN by also introducing the necessary modifications to mitigate unstable training. Moreover, Figure 1 shows that, while CDCVAE appears to be able to generate samples of digits that are of higher quality than those generated by CDCGAN, the corresponding CIFAR10 images seem qualitatively worse (blurrier). This effect has also been identified in the literature and it is known that GAN approaches typically generate sharper images (Goodfellow, 2016). The effect arises from the use of prior and variational (approximate posterior) distributions that are not sufficiently expressive, leading to models that are not asymptotically consistent. More expressive distributions can be found by leveraging for instance autoregressive structures (Chen et al., 2016b) and normalizing flows (Kingma

	CIFAR10					MNIST				
	ROT	BLUR	ROTBLUR	CDCVAE	CDCGAN	ROT	BLUR	ROTBLUR	CDCVAE	CDCGAN
1	152.68	0.00	152.68	217.57	322.86	48.17	0.00	48.17	59.40	87.66
2	128.72	0.00	128.72	189.78	324.95	41.63	0.00	41.63	50.31	117.85
5	98.84	0.00	98.84	164.43	289.15	33.77	0.00	33.77	34.13	92.32
10	90.27	0.00	90.27	145.92	308.80	31.85	0.00	31.85	24.81	67.63
20	86.65	0.00	86.65	142.74	198.34	31.26	0.00	31.26	17.78	72.83
30	84.97	0.00	84.97	138.41	168.91	30.75	0.00	30.75	14.57	63.51
40	84.04	0.00	84.04	135.45	206.53	30.58	0.00	30.58	13.10	68.61
50	84.15	0.00	84.15	139.24	147.05	30.68	0.00	30.68	11.81	37.68
60	84.15	0.00	84.15	138.93	179.15	30.66	0.00	30.66	11.99	28.69
70	83.87	0.00	83.87	139.51	157.47	30.69	0.00	30.69	11.64	23.60
80	83.36	0.00	83.36	139.32	137.68	30.54	0.00	30.54	11.73	14.19
90	83.15	0.00	83.15	140.35	123.05	30.48	0.00	30.48	11.96	16.18
100	83.19	0.00	83.19	141.30	164.50	30.61	0.00	30.61	11.96	15.94

Table 1. Fréchet Inception Distance (FID) computed on images that have been generated using traditional data augmentation techniques (ROT, BLUR and ROTBLUR) and neural data augmentation methods (CDCVAE and CDCGAN). For each training set size (%), the neural data augmentation method that yields the lowest FID is underlined. Training set sizes below 30% do not meet the recommended criteria of a sample size greater or equal than 10,000 and might not reflect the "true" FID.

et al., 2016), but, again, these modifications are left as future work. Perhaps due to the blur effect, the set of images from Figure 1 also seems to suggest that CDCVAE is able to generate qualitatively better images in the low data regime.

As mentioned in Section 2.5, we do not limit ourselves to a qualitative analysis. Indeed, in order to quantitatively assess the quality of the data generation processes, we compute the Fréchet Inception Distance (FID) (Heusel et al., 2017) for each set of images. We also extend this analysis to traditional techniques, namely those based on rotations (ROT), Gaussian blur (BLUR) and the combined effect of the two previous transformations (ROTBLUR). Consistency between neural and traditional approaches is ensured by applying a specific transformation with random parameters (recall Section 2.3) to each image in the corresponding training set, obtaining sets of images of equal size. All corresponding results are shown in Table 1, but note that as recommended in (Heusel, 2018) the minimum size to calculate the FID should be of 10,000 samples, which according to our experimental setup does not hold for training set sizes below 30%. The results reveal some of the limitations of applying a pre-trained neural network to generated images and computing statistics at a particular hidden layer, as done in FID. In particular, we observe that the FID of images generated by BLUR is always roughly equal to 0, suggesting that these synthetic samples should be as realistic as the original images and contain a high diversity of images with clear objects. However, since Gaussian blur has been injected into the original images, the metric should assign higher FID values, reflecting that the objects are blurrier. We hypothesize that this problem is caused by the pre-trained neural network itself, which appears to yield to a certain degree blur-invariant features and, consequently, the activations and the resulting statistics are roughly the same. Another problem is that the Inception v3 network has been pre-trained on a different data set (ImageNet) and, as discussed in (Barratt & Sharma, 2018), this might yield misleading results, but, again, better quantitative metrics are currently lacking. This metric shows, nevertheless, a good consistency property: the quality of the data generation process in traditional methods is not affected by the training set size and the FID values should reflect this characteristic by being approximately equal, which indeed holds for the recommended training set sizes (greater or equal than 30%). Regarding the neural generative approaches, we observe that while the FID values on MNIST tend to become increasingly smaller as the training set size increases (especially so for CDCGAN), on CIFAR10 and specifically for CDCVAE the values fluctuate around 140. This, in turn, suggests that, despite the smaller FID values when compared to CDCGAN, CDCVAE might be a poor generative model of natural images in the sense that the quality of the generation process does not seem to improve in the presence of more training data. CDCVAE is however more robust to low data regimes. Finally, it is also worth pointing that, when trained on MNIST, CDCVAE (and CDCGAN when trained on sufficiently large data sets) appears to be a good generative model of digits. This is in contrast to CIFAR10, where traditional approaches (simple rotations) achieve consistently smaller FID values.

Arguably the most important part of this work corresponds to the analysis of the classification performance using different data augmentation techniques, contrasting traditional and neural methods. For this purpose, we augment the original training sets with synthetic images, obtaining training sets of double the size. As described in Section 2.2, the classifiers are VGG-Nets whose optimal configurations have been determined in previous work. In order to keep the notation and the visualization simple, we still refer to these augmented sets as 1, 2, 5, 10, ..., 100%. For control purposes, we further consider an additional method, AUGORIG, where we duplicate each original image. This allows us to focus directly on the quality of the augmentations by removing the effect of training the classifiers on more data (each epoch takes twice as long), which surprisingly is often neglected in the literature. The experimental process then consists in independently training the optimal VGG-Net classifiers for 200 epochs on each possible augmented training set and registering the best observed validation accuracy and its corresponding test accuracy. The

	CIFAR10							MNIST						
	ORIG	AUGORIG	BLUR	ROT	ROTBLUR	CDCVAE	CDCGAN	ORIG	AUGORIG	BLUR	ROT	ROTBLUR	CDCVAE	CDCGAN
1	41.81	44.68	**44.78**	44.61	44.53	43.48	43.06	95.52	96.29	96.17	96.66	**96.76**	95.90	95.64
2	51.64	**52.17**	50.22	51.45	51.54	49.20	48.47	97.43	97.61	97.55	**98.01**	97.96	97.57	96.74
5	64.19	63.74	64.18	**64.58**	64.05	59.50	61.87	98.46	98.50	98.53	**98.80**	98.51	98.19	98.23
10	70.60	**71.98**	71.30	71.30	70.63	67.13	69.34	98.93	98.94	98.94	98.89	**99.10**	98.82	98.87
20	77.83	**78.36**	78.32	77.38	78.14	74.07	76.78	99.10	99.17	99.10	99.08	**99.25**	99.21	99.11
30	80.38	**81.92**	81.46	81.01	81.43	78.31	80.76	99.17	99.29	99.27	**99.41**	99.33	99.22	99.25
40	83.48	**84.17**	83.36	83.44	83.88	80.36	82.62	**99.35**	99.26	99.26	99.29	99.31	99.32	99.21
50	84.91	84.72	84.57	**84.94**	84.40	82.47	83.91	99.33	99.31	**99.38**	99.32	99.36	99.28	99.24
60	85.42	**86.48**	86.45	86.42	85.86	83.57	85.45	99.36	99.45	99.46	99.46	99.42	**99.47**	99.26
70	86.18	87.13	**87.55**	86.56	86.79	83.70	85.94	99.35	99.43	99.39	**99.49**	99.46	99.38	99.38
80	87.16	87.78	**87.97**	87.38	87.74	85.18	86.68	99.34	99.36	99.43	99.43	99.44	**99.45**	99.31
90	87.96	**88.72**	88.29	88.40	88.36	85.91	87.42	99.37	**99.50**	99.35	**99.50**	99.45	99.43	99.36
100	88.62	**89.36**	89.01	89.10	89.20	86.28	87.85	99.43	99.40	**99.56**	99.40	99.47	99.34	99.40

Table 2. Test accuracy (%) after training on data sets with different augmentations using optimal VGG-Net classifiers. For each training set size (%), the method that yields the best performance is highlighted in bold. The best traditional data augmentation method (BLUR, ROT and ROTBLUR) is dash-underlined and the best neural data augmentation (CDCVAE, CDCGAN) is underlined.

resulting learning curves corresponding to test accuracy are shown in Figure 2. We note that all learning curves appear to follow a logarithmic trend with accuracy increasing as the training sets become larger, an observation that is further validated by high R^2 values (greater than 0.90). However, for legibility reasons, we do not show the fit to each curve, unlike previous work where we provided a fit to the curves corresponding to ORIG (original training sets without data augmentation). Surprisingly, we observe that while data augmentation techniques seem in general to slightly boost classification performance on MNIST, augmentations on CIFAR10 do not seem to have a positive effect, especially when compared to the control method AUGORIG. For instance, while CDCVAE appears to be among the best methods on MNIST, it clearly has a negative effect on CIFAR10, where the test accuracy is consistently lower than both the control method AUGORIG and the baseline ORIG. This negative effect is also observed for CDCGAN, although in some cases (e.g. CIFAR10-30) the performance is marginally better than ORIG. CDCGAN tends to be the worst method on MNIST (albeit only marginally) and it is worth recalling that the FID analysis suggested that the MNIST augmentations generated by CDCVAE are of higher quality than that of CDCGAN. Unfortunately, FID does not seem in general to correlate well with classification accuracy because the natural images (CIFAR10) generated by CDCVAE also showed smaller FID values, but the performance using CDCVAE augmentations is distinctly the worst.

In addition to Figure 2, we report the test accuracy values in Table 2 as it allows for a more thorough analysis of classification performance. Regarding CIFAR10, we notice that the control method AUGORIG consistently achieves one of the best performances. This observation fundamentally reveals that 200 epochs might not be sufficient for the classifier to learn effectively, because otherwise the observed accuracy values in AUGORIG would be lower than that of ORIG due to overfitting which, in turn, data augmentation strategies aim to mitigate – note that we do not show the validation accuracy table due to space constraints, but

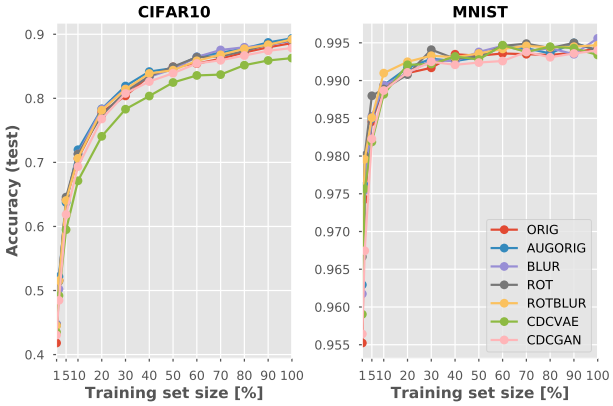

Figure 2. Test accuracy using optimal VGG-Nets on CIFAR10 and MNIST with varying training set sizes (learning curves) and for different data augmentation methods.

similar conditions are observed. Recall that, as in previous work, the reason the classifiers are trained for 200 epochs is mostly due to computational constraints and the need to carry out a high number of experiments. That said, we believe that as future work it would be interesting to repeatedly perform these experiments with an increased number of epochs in order to address the aforementioned problem and to extract statistically valid conclusions. Nevertheless, we still believe that it is possible to compare traditional and neural data augmentation methods using our current experimental setup. Unfortunately, the neural data augmentation methods do not seem better than traditional approaches, but, again, it is likely to be due to the relatively simple architectures that we have considered. Regarding the comparison between CDCVAE and CDCGAN, CDCVAE augmentations yield better results for training set sizes of 1% and 2%, suggesting that CDCVAE is indeed more robust to low data regimes. In general, however, CDCGAN shows comparatively better results on CIFAR10 and the largest difference 2.71% occurs when the training size is 20%, decreasing to around 1.5% as the training set size increases (apart from moderate fluctuations). On MNIST and excluding sizes 5%, 10%, 30% and 100%, CDCVAE shows marginally better results, with the largest difference being of 0.83% when

the training set size is 2%, but in general the difference is less than 0.3%. Finally, and as a side note, it is interesting to observe that combining transformations (ROTBLUR) does not necessarily yield the best results among traditional methods.

4. Related Work

Deep neural networks typically require a large amount of data in order to train them successfully. However, in many realistic scenarios only a limited amount of data is available. In these cases, deep neural networks usually overfit, generalizing poorly on new data. To mitigate this effect, techniques such as ℓ_1-, ℓ_2-regularization, dropout (Srivastava et al., 2014) or batch normalization (Ioffe & Szegedy, 2015) have been developed. Indeed, many of these procedures (and variants) are actively used in recent neural network architectures (Schmidhuber, 2015; Gu et al., 2017), but, when operating in a low data regime, these methods may not be able to mitigate overfitting to a sufficient degree, since the flexibility of the network may still be high. In this regime, the combination of deep neural networks with Bayesian theory has shown promising results (Gal & Ghahramani, 2015). Transfer learning schemes (Yosinski et al., 2014) appear to be equally relevant, where the network is, for instance, trained on a large-scale data set with a possibly different task and then fine-tuned on the (small-scale) target data set. It is also worth mentioning that some researchers have dedicated their efforts to the collection of large-scale data sets. Traditional collection of labeled data sets has become impracticable because it often involves human intervention. One approach has been to adopt crowdsourcing services that allow large-scale annotation (Russakovsky et al., 2015), but a set of new problems arises and it does not remove humans from the loop. Others have considered the use of external data from online search engines (Xie et al., 2014).

An alternative approach is to consider data augmentation, where the data set is enlarged by new synthetic observations. For this purpose, traditional methods use a combination of transformations that are specified a priori, which for image data includes rotations, translations and additive noise (Krizhevsky et al., 2012). More recent advances posit that data augmentation strategies can be automatically learned. In (Hauberg et al., 2016), the authors propose the random generation of diffeomorphisms on a per-class basis by learning the parameters of class-specific generative models and proceed to show the benefits of this approach in terms of classification performance when compared to traditional schemes. On the other hand, inspired by the development of deep neural generative models, such as Generative Adversarial Network (GAN) (Goodfellow et al., 2014a), the authors in (Antoniou et al., 2017) propose a GAN variant that is able to learn plausible transformations and can generalize such transformations to unseen classes, since the generation process is itself class-agnostic. Note that in this work we also draw inspiration from neural generative models, but we explore different models and challenges, providing an analysis of the quality of the data generation processes and a direct comparison between neural and traditional data augmentation strategies.

5. Conclusions

In this report, we have analyzed the problem of training deep neural network classifiers on comparatively small data sets, exploring the idea of using neural generative models as a means to automatically learn plausible transformations from data and to augment the original data sets with artificial observations (neural data augmentation). This is in stark contrast to traditional data augmentation, where these transformations are specified a priori.

To this end, we have identified image recognition as being an appropriate task and have built base data sets with training sets of varying size from CIFAR10 and MNIST. We have then considered neural data augmentation based on conditional deep convolutional variants of popular neural generative models, namely CDCGAN and CDCVAE. We have qualitatively analyzed the resulting data generation processes and found that, unsurprisingly, the overall quality tends to improve as the training set size increases. This trend also applied to the more complex CIFAR10 data set, but the generated data lacked the global structure and perspective found in natural images. Presumably, better architectures, based for instance on ResNet (He et al., 2016), can solve these defects, but these modifications are left as future work. Other possible modifications include feature matching and minibatch features (Salimans et al., 2016) or the use of Wasserstein-1 distance (Gulrajani et al., 2017) to mitigate the unstable training found in CDCGAN. Similarly, while CDCVAE seemed to generate qualitatively better images of digits than CDCGAN, natural images were blurrier. Supposedly, more expressive priors and approximate posteriors can mitigate this problem (Kingma et al., 2016). Nevertheless, CDCVAE seemed to generate better images in the low data regime. We have also analyzed the generated images quantitatively and FID suggested that, when compared to traditional approaches, both CDCVAE and CDCGAN were poor generative models of natural images. However, FID suffers from fundamental flaws, mostly because statistics are computed from the hidden layers of a network pre-trained on a different data set (ImageNet). Better approaches seem non-existent which leads us to believe that quantitative assessment of synthetic data deserves more attention. Finally, we have analyzed classification performance (accuracy) using different data augmentations. In all learning curves, accuracy increased as the training set became larger. Unfortunately, neural data augmentation did not seem better than traditional methods, but it is likely to be due to the relatively simple architectures. When compared to CDCGAN, CDCVAE yielded better results on MNIST and in the low data regime, but was generally worse on CIFAR10. Surprisingly, while data augmentation seemed to have a positive effect on MNIST, the same did not occur on CIFAR10. It would however be interesting to repeat the experiments with an increased number of epochs as to observe overfitting in the control method.

References

Abadi, Martín et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/.

Antoniou, Antreas, Storkey, Amos, and Edwards, Harrison. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Arjovsky, Martin, Chintala, Soumith, and Bottou, Léon. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Barratt, Shane and Sharma, Rishi. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

Chatfield, Ken, Simonyan, Karen, Vedaldi, Andrea, and Zisserman, Andrew. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

Chen, Xi, Duan, Yan, Houthooft, Rein, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2172–2180, 2016a.

Chen, Xi, Kingma, Diederik P, Salimans, Tim, Duan, Yan, Dhariwal, Prafulla, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016b.

Chollet, François et al. Keras. https://keras.io, 2015.

Gal, Yarin and Ghahramani, Zoubin. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

Goodfellow, Ian. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014a.

Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Gu, Jiuxiang, Wang, Zhenhua, Kuen, Jason, Ma, Lianyang, Shahroudy, Amir, Shuai, Bing, Liu, Ting, Wang, Xingxing, Wang, Gang, Cai, Jianfei, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 2017.

Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, Dumoulin, Vincent, and Courville, Aaron C. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5769–5779, 2017.

Hauberg, Søren, Freifeld, Oren, Larsen, Anders Boesen Lindbo, Fisher, John, and Hansen, Lars. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In *Artificial Intelligence and Statistics*, pp. 342–350, 2016.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Heusel, Martin. Ttur: Two time-scale update rule for training gans, 2018. URL https://github.com/bioinf-jku/TTUR.

Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6629–6640, 2017.

hwalsuklee. tensorflow-generative-model-collections: Tensorflow implementation of various gans and vaes., 2018. URL https://github.com/hwalsuklee/tensorflow-generative-model-collections.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456, 2015.

Johnson, Matthew, Duvenaud, David K., Wiltschko, Alex, Adams, Ryan P., and Datta, Sandeep R. Composing graphical models with neural networks for structured representations and fast inference. *arXiv preprint arXiv:1603.06277*, 2016.

Kingma, Diederik P and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, Diederik P, Salimans, Tim, Jozefowicz, Rafal, Chen, Xi, Sutskever, Ilya, and Welling, Max. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pp. 4743–4751, 2016.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Yann, Bengio, Yoshua, and Hinton, Geoffrey. Deep learning. *Nature*, 521(7553):436, 2015.

Mirza, Mehdi and Osindero, Simon. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Reddi, Sashank J., Kale, Satyen, and Kumar, Sanjiv. On the convergence of adam and beyond. *International Conference on Learning Representations (forthcoming)*, 2018.

Rezende, Danilo Jimenez and Mohamed, Shakir. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Rolfe, Jason Tyler. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Sabour, Sara, Frosst, Nicholas, and Hinton, Geoffrey E. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pp. 3859–3869, 2017.

Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Schmidhuber, Jürgen. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Simard, Patrice Y, Steinkraus, David, Platt, John C, et al. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pp. 958–962, 2003.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Sohn, Kihyuk, Lee, Honglak, and Yan, Xinchen. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pp. 3483–3491, 2015.

Srivastava, Akash, Valkoz, Lazar, Russell, Chris, Gutmann, Michael U, and Sutton, Charles. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pp. 3310–3320, 2017.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, Rabinovich, Andrew, et al. Going deeper with convolutions. Cvpr, 2015.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

Theis, Lucas, Oord, Aäron van den, and Bethge, Matthias. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Xie, Saining, Yang, Tianbao, Wang, Xiaoyu, and Lin, Yuanqing. Hyper-class augmented and regularized deep learning for fine-grained image classification. *CVPR2015*, 2014.

Yosinski, Jason, Clune, Jeff, Bengio, Yoshua, and Lipson, Hod. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.