Transferability of Adversarial Attacks in Model-Agnostic Meta-Learning

Riley Edmunds
Machine Learning at Berkeley
edmunds@ml.berkeley.edu

Noah Golmant
Machine Learning at Berkeley
noah.golmant@ml.berkeley.edu

Vinay Ramasesh
Machine Learning at Berkeley
ramasesh@ml.berkeley.edu

Phillip Kuznetsov
Machine Learning at Berkeley
philkuz@ml.berkeley.edu

Piyush Patil
Machine Learning at Berkeley
piyush@ml.berkeley.edu

Raul Puri
Machine Learning at Berkeley
raul@ml.berkeley.edu

ABSTRACT

Model-Agnostic Meta-Learning (MAML) [3] has proven to be a powerful, lightweight framework for transfer of learned knowledge in task-specific model adaptation. However, recent work in adversarial machine learning has demonstrated that many deep neural networks are susceptible to adversarial examples – perturbed inputs that an attacker has designed to cause the model to make a mistake [17]. In this paper, we propose a series of experiments designed to test the susceptibility of MAML to adversarial attacks. Moreover, we test the hypothesis that an adversary can transfer an attack from MAML's shared representation of knowledge to a model tuned for performance on a particular task, and even between task-specific models fine-tuned from a common MAML initialization.

1 INTRODUCTION

Deep learning methods have achieved significant progress in many domains, occasionally surpassing human performance in specific, narrow problems. While the scope of abilities learned by such deep networks is, in general, tough to generalize, recent research has yielded methods for transferring learned knowledge between related tasks. In one approach, known as meta-learning, several related instances of a more general problem are tackled together, with an initial exploration seeking to uncover general principles underlying the related tasks. The result of this exploration is used to initialize a model which is then optimized to perform well on a single task.

One particularly simple yet effective framework, Model-Agnostic Meta-Learning (MAML), has been shown to perform well on few-shot learning tasks in domains such as character recognition and reinforcement learning [3] [15], opening the door for rapid adaptation by providing potential for task-level model specificity without loss of generalization. Meta-learning systems (particularly MAML) have huge potential for practical deployment, since they allow for fine-tuning of a general deep learning model to a user's specific needs.

The increasing popularity of meta-learning necessitates proactive investigation of its possible failure modes. Deep networks are often vulnerable to adversarially crafted, tiny perturbations to the input which, though imperceptible to humans, result in a misclassification by the network [17] [4]. These perturbed inputs are known as *adversarial examples*, and are often most effective against models that are trained for narrowly defined tasks.

The specificity inherent to fine-tuned MAML models suggests that they may be particularly susceptible to adversarial attacks. Further, the similarity between MAML models fine-tuned for related tasks suggests that adversarial attacks crafted to defeat one MAML fine-tuned model may transfer to MAML model fine-tuned for related tasks.

In this work, we study the susceptibility of MAML-trained classification models to adversarial attacks, and the transferability of such attacks across MAML models trained from a common initialization.

This paper's proposed contributions are as follows:

- We investigate the susceptibility of MAML-trained meta-models and fine-tuned models to adversarial examples.
- We investigate the transferability of adversarial attacks from MAML-trained meta-models to their fine-tuned descendants.
- We investigate the transferability of adversarial attacks between MAML-trained fine-tuned models which share a common meta-model.

2 BACKGROUND

2.1 Adversarial Examples

A model is a function f_θ with parameters θ. Given a particular model f_θ, an *adversarial attack* on the model applies a small perturbation to input examples to induce a misclassification. Concretely, the attack perturbs an input x by adding a small vector v, such that f_θ places the adversarial input $\tilde{x} = x + v$ in a different class than the original x. Usually, adversarial generation processes limit the size of the perturbation (often resulting in perturbations small enough to be unnoticeable to humans).

Non-targeted adversarial attacks generate perturbations with the objective of increasing the model's loss with respect to the true label y. A common approach for generating adversarial perturbations is to ascend the gradient of the loss function with respect to an example x. We use both of the following attack methods to generate non-targeted adversarial attacks in this work.

Fast Gradient Sign Method (FGSM) FGSM applies a perturbation vector v (pointing in the direction of the sign of the loss-function gradient $\nabla_x \mathcal{L}$) element-wise to the input vector x, while fixing the size of the perturbation $|v|$ by some constant ϵ. The adversarial example is thus:

$$\tilde{x} = x + \epsilon \cdot \operatorname{sign}(\nabla_x \mathcal{L}(f_\theta(x), y)) \qquad (1)$$

Optimization-Based Attack This technique aims to jointly maximize the loss $\mathcal{L}$ and minimize the size of the perturbation $|v| =$

$d(x, \tilde{x})$ [9]. This can be formulated as the optimization problem:

$$\tilde{x} = \operatorname*{argmin}_{\tilde{x}} \lambda d(x, \tilde{x}) - \mathcal{L}(f_\theta(\tilde{x}), y) \tag{2}$$

2.2 Model-Agnostic Meta-Learning

In a meta-learning scenario, we consider the problem of learning a model that can adapt to multiple tasks. In this paper, these tasks are supervised classification problems. A supervised task $\mathcal{T} = \{\mathcal{L}(\hat{y}, y), P(x, y)\}$ consists of a loss function $\mathcal{L}(\hat{y}, y) \to \mathbb{R}$ and a distribution over samples $P(x, y)$. Tasks are distributed according to $p(\mathcal{T})$. For a task $\mathcal{T}_i = \{\mathcal{L}_i, P_i\} \sim p(\mathcal{T})$, the K-shot learning problem requires finding a good model using only K labeled samples drawn from P_i and the feedback generated by $\mathcal{L}_i$. In Meta-Agnostic Meta-Learning (MAML) [3], this task is accomplished by explicitly optimizing a model with information about the general distribution of tasks $p(\mathcal{T})$. This general model can be quickly trained to work for any $\mathcal{T}_i \sim p(\mathcal{T})$ using only K samples drawn from P_i. To compute parameters θ_i' that work well on a particular task $\mathcal{T}_i$, we perform the gradient descent update on the meta-model parameters θ given a step size α on $(x, y) \sim P_i$:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_i(f_\theta(x), y) \tag{3}$$

We would like to learn a good θ on a set of training tasks, so that we can later perform this update using K samples from a particular task $\mathcal{T}_i$ and achieve a low loss. This leads to the MAML meta-objective:

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_i(f_{\theta_i'}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_i(f_{\theta - \alpha \nabla_\theta \mathcal{L}_i(f_\theta)}) \tag{4}$$

That is, we optimize over the meta-model's parameters θ based on how each set of updated task-specific parameters θ_i' perform on their respective tasks $\mathcal{T}_i$. Once optimized, the meta-model is not likely to perform optimally on any of the individual tasks, but rather is close (in parameter space) to the optimal models for a variety of the individual tasks. Thus, going from the meta-model to a model optimized (*fine-tuned*) for a particular task consists of standard SGD optimization with respect to the task-specific loss, taking the meta-model parameters as the initial starting point.

There are many meta-learning models that can perform this process, however MAML is one of the simplest formulations and is also one of the most effective. MAML differs from other techniques [2, 16] in that it does not require extra parameters for the meta-objective. Despite this, it still achieves state-of-the-art empirical results [3].

2.3 Universal Perturbations

In general, adversarial perturbations for a specific task are input-specific; that is, a perturbation which causes a misclassification of a particular image will not work for the rest of the validation set. However, recent work has demonstrated the existence of so-called ‘universal adversarial perturbations,’ which can be identically applied across the entire dataset and cause a large number of misclassifications [11]. While a general perturbation calculated for a specific input is not universal, by sequentially applying an FGSM-like attack to a variety of input images and subsequently re-normalizing the size of the perturbation vector, such a universal perturbation can be constructed (for a single task). We will explore the construction of perturbations which are universal among a variety of related tasks.

2.4 Measuring Model Susceptibility to Adversarial Attacks

To our knowledge, no standard metric has been adopted to quantify the vulnerability of a particular model to adversarial attacks. In this work, we use a method which has been applied previously in a study of adversarial susceptibility [7]. Specifically, we apply perturbations of a variety of sizes to examples drawn from a labeled validation set and measure, as a function of the size, the fraction of perturbed inputs which result in a misclassification. This metric is known as the *top-1 inaccuracy* as it considers all cases in which the model output an incorrect classification; it is also possible to consider the *top-n inaccuracy*, in which the adversarial attack is considered successful only if the model's n most likely classifications do not contain the true label. To quantify the size of the perturbation, note that for the FGSM technique, this size is simply given by ϵ, the multiplier on the sign of the gradient vector (see Eq. 1); in the second method (Eq. 2), a proxy for the size of the perturbation is given by λ^{-1}.

3 TRANSFERABILITY OF ADVERSARIAL ATTACKS ON MAML

Consider a trained meta-model f_θ and two models fine-tuned from f_θ, $f_{\theta_i'}$ and $f_{\theta_j'}$ for tasks $\mathcal{T}_i$ and $\mathcal{T}_j$. What is the relationship between these models in terms of their susceptibility to adversarial examples? For example, if one can construct adversarial examples for f_θ, is it easy to construct adversarial examples for $f_{\theta_i'}$? Can one easily perturb an adversarial example for $\mathcal{T}_i$ to attack the model for $\mathcal{T}_j$? We will explore these questions through a series of experiments.

In realistic scenarios, an attacker will likely not have full access to the model f or its parameters θ, but will be forced to construct some model which approximates the behavior of the original model. We call this a *pseudo-model*, $\hat{f}$. Attacks on $\hat{f}$ often translate well to attacks on the original model [12]. Our aim here is more to understand the vulnerabilities of the MAML framework to attack; thus, we assume the most favorable scenario to the attacker in which the attacker can use the entire model and has access to the model parameters. Despite this simplification, in a more realistic scenario, an attacker could create a sufficient-quality pseudo-model and apply the attacks laid out in this work.

3.1 Theoretical Justification for Adversarial Attacks on MAML

We consider the question of how easily one can adversarially attack a meta-model trained with MAML, and study constraints on the size of the adversarial perturbations necessary. In particular, we study the dimensionality of the subspace generated by adversarial examples at an input point, with the motivation that, as elaborated on by Goodfellow et al. [18], it is fruitful to treat the dimensionality of the adversarial subspace at a point as a proxy to measure both the prevalence of adversarial examples and the transferability of those examples to other models. First, recall that MAML-trained meta-models are trained not with accuracy or loss optimization in mind but rather trained to maximize the model's capacity to generalize quickly (i.e. through K-shot training, for low K) across a distribution of diverse tasks. Because such models are optimized to learn data representations suitable for high potential to rapidly generalize on a number of tasks, and not for straightforward loss reduction, a trained model will necessarily reside, in parameter space, at a point θ where various tasks' loss functions on the model are highly sensitive to small shifts in the parameters, behaving with a high degree of elasticity. Indeed, it is this high degree of elasticity which allows models to improve their loss in just a few training steps on a new task. Assuming that the fine-tuning step of the MAML algorithm is both effective and necessary, it follows that θ must be near local minima of many of the loss functions $\mathcal{L}_i$ corresponding to different tasks. Hence, for a well-trained meta-model, we can expect the degree of smoothness [1] of the loss function to be low.

Goodfellow et al. [18] shows that the smoothness of the loss function in input space inversely correlates with the number of independent adversarial directions at that point in input space. Further, the loss function $\mathcal{L}$ used to train the meta-model is formulated as a weighted average (over the task distribution) of tasks' loss functions $\mathcal{L}_i$, and hence low smoothness in each of the $\mathcal{L}_i$ indicates low smoothness in $\mathcal{L}$. Furthermore, if we assume that the parameterized set of models is constrained to itself contain only smooth models [2], then it follows that, viewing $\mathcal{L} : X \times \mathbb{R}^n \to \mathbb{R}^+$ as a joint function of the inputs (from input space X) and of parameters (from parameter space $\mathbb{R}^n$), the low smoothness of $\mathcal{L}(\theta, \cdot)$ extends to a similarly low degree of smoothness in $\mathcal{L}(\cdot, x)$, for any $x \in X$. Thus, the above provides us with an argument for the low degree of smoothness of the meta-model's loss function in input space, which, as discussed above in reference to the result by Goodfellow et al. [18], indicates the relatively high dimensionality of the adversarial subspace at points in the input space of the meta-model. Thus, given our assumed premises, we have strong reason to suspect both a prevalence of adversarial examples in a MAML-trained meta-model and ease of transferability of those adversarial examples to similar, and therefore, fine-tuned, models.

3.2 Experimental Setup

In experiments from Sections 3.3-3.6, both meta-models and fine-tuned models are CNNs containing 4 convolutional layers with ReLU activations and finally a fully connected layer followed by a softmax activation. For these experiments, we use the Omniglot dataset [8]. Each classification task consists of a random sample of twenty classes of characters from the entire dataset. Prior to fine-tuning, we train the Meta-Model for 60,000 iterations, replicating the results on 20-way classification in the original paper [3]. To create a fine-tuned model for a task, we sample 20 points from the task (1 test point per class), and 1-shot train the meta-model to obtain the weights of the fine-tuned model. We then sample another 20 points from the task to create an attack, ensuring that each point is correctly classified by the fine-tuned model. We then measure the quality of an attack via the fooling rate on the fine-tuned model: the fraction of examples which were originally classified correctly, yet were misclassified after perturbation. Following the experiments from [9], we use the ADAM solver [6] to construct our adversarial examples $\tilde{x}$.

3.3 Robustness of MAML to Adversarial Attack

As a baseline, we test the susceptibility of the MAML meta-model f_θ and a fine-tuned model $f_{\theta'_i}$ to adversarial attacks, both by direct FGSM and optimization-based attacks, and by construction of universal perturbation vectors.

To attack a meta-model f_θ, we draw a task $\mathcal{T}_j \sim p(\mathcal{T})$ and a data point $(x_j, y_j) \sim P_j$. $\mathcal{T}_j$ should be a task on which the meta-model was trained. We first use FGSM to construct a perturbed example $\tilde{x_j}$ by ascending the gradient $\nabla_x \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i}(x_j), y_j)$ from equation (1). Then, we construct universal adversarial perturbations for the given task. That is, we seek a perturbation which causes $f_{\theta'_i}$ to yield errors across its K task-specific input examples.

To attack fine-tuned model $f_{\theta'_i}$, we will draw a sample $(x_i, y_i) \sim P_i$ and ascend the gradient $\nabla_x \mathcal{L}_i(f_{\theta'_i}(x_i), y_i)$ to construct a perturbed example $\tilde{x_i}$. Then, we construct universal adversarial perturbations to all inputs to task $\mathcal{T}_i$.

3.4 Transferability of Adversarial Attacks from Meta-Model to Task-Specific-Model

We evaluate the ability to zero-shot transfer adversarial attacks from the meta-model f_θ to $f_{\theta'_i}$ which is fine-tuned for task $\mathcal{T}_i$.

To transfer a direct FGSM attack from f_θ, we draw an input x_i from task $\mathcal{T}_i$, to which we would like to transfer the attack, and adversarially perturb it by ascending the gradient of the *task's* loss using the *meta-model's* parameters: $\nabla_x \mathcal{L}_i(f_\theta(x_i), y_i)$. Since we only take K gradient steps to produce θ'_i from θ, we hypothesize that the parameters are sufficiently close to allow to ascend this gradient in a reasonable manner.

We previously discussed constructing a universal perturbation that works well on many samples for a particular task. In this experiment, we look for a perturbation which works well on many samples for many related tasks; that is, given a meta-model f_θ, we seek a perturbation which causes f_θ to yield errors on a variety of different inputs and subtasks. Here we attempt to find such a task-universal perturbation by applying a method similar to that used to find universal perturbations for a single task. We draw tasks and inputs repeatedly from their respective distributions, ascend the gradient of the relevant loss function and normalize the perturbation vector at each step. After a sufficiently dense sampling of tasks and inputs, the resulting perturbation vector should be universal for both tasks and inputs. Given the existence of such a universal perturbation with respect to the MAML meta-model, we will evaluate the ability to zero-shot transfer this adversarial attack to the resulting fine-tuned models. Specifically, beginning with the MAML meta-model, we fine-tune the model for each of its learned tasks (the ones for which the meta-model was vulnerable to attack) and evaluate whether the adversarial perturbation is still successful in attacking the task-specific models. Then, we fine-tune the meta-model for related tasks outside of this training set, and attempt to attack these models with the previously-found perturbation.

[1] Though glanced over here, there do exist rigorous metrics of how continuous a function is, e.g. the δ to ϵ ratio in the definition of uniform continuity.

[2] This is not a very restrictive assumption, seeing as it is necessary to perform gradient descent during training in the first place.

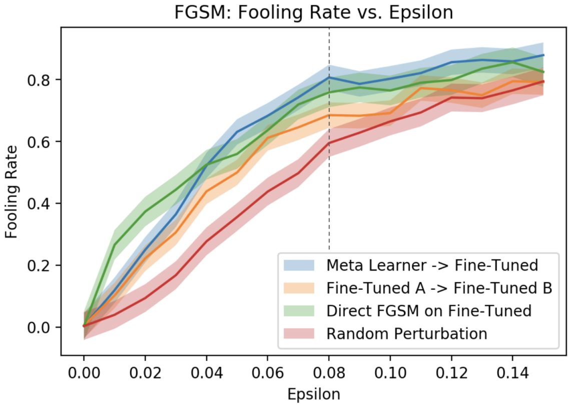

Figure 1: Fooling rate of the attack methods as a function of the size of the perturbation. Although the random method shows that the Omniglot classifier is already fragile, the transfer techniques still work effectively for low ϵ.

Table 1: Attack Performance ($\epsilon = 0.8$)

Attack Method	Fooling Rate
Random	.572
UPERT FT → FT	.619
FGSM FT → FT	.685
UPERT ML → FT	.701
FGSM Direct	.759
FGSM ML → FT	**.810**

3.5 Transferability of Adversarial Attacks between Task-Specific-Models

Next, we evaluate the transferability of adversarial attacks from some MAML fine-tuned model $f_{\theta'_i}$ to some other MAML fine-tuned model $f_{\theta'_j}$ by the optimization attack, direct FGSM and universal perturbation vectors.

We expect that between two fine-tuned models derived from a common MAML model, the parameters are sufficiently similar to let us use an attack successful on one model to attack the other. That is, the representations between two fine-tuned models are similar enough that it is feasible to come up with an adversarial perturbation that exploits the shared representation captured in one pseudo-model.

Here, we don't even need the original meta-model: having both fine-tuned models and the knowledge that they were fine-tuned from the same attack is sufficient. We evaluate the ability to zero-shot transfer adversarial attacks from fine-tuned model f_{θ_i} to fine-tuned model f_{θ_j}. To transfer a direct FGSM attack from f_{θ_i}, we draw an input from the task $\mathcal{T}_j$ to which we would like to transfer the attack, and adversarially perturb it by ascending the gradient of the initial task $\nabla_x \mathcal{L}_i(f_{\theta'_i}(x), y)$.

To transfer a universal adversarial perturbation from f_{θ_i}, we first construct an adversarial perturbation universal to all inputs to task $\mathcal{T}_i$; that is to say, we seek a perturbation p which causes $f_{\theta'_i}$ to yield misclassification for all available input examples. We then adversarially perturb input x to task $\mathcal{T}_j$ by p.

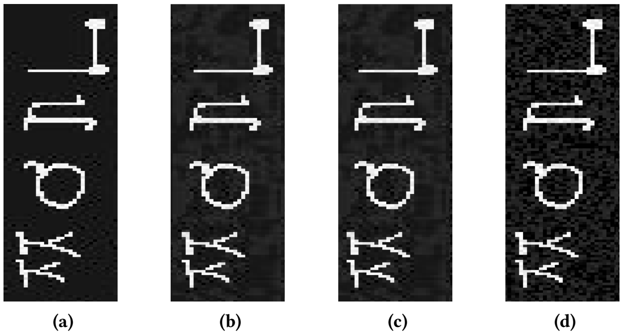
(a) (b) (c) (d)

Figure 2: Characters from the Omniglot language classification task (a) without perturbations, (b) perturbed with FGSM, (c) randomly perturbed, (d) with a universal peturbation

4 PRELIMINARY RESULTS

We show preliminary results in Figure 1 and Table 1, and sample adversarial images in Figure 2. The setup for this test is as described in section 3.2; briefly, a 4-layer CNN Meta-Model is trained on the Omniglot data set for 60,000 iterations, after which single-shot training is used to create child fine-tuned models. As shown in Figure 1, four types of attacks on a fine-tuned model were explored: a random perturbation, a direct attack using FGSM, an attack using FGSM on the meta-model parameters, and an attack using FGSM on a sibling fine-tuned model parameters.

As expected, the lowest fooling rate for all epsilon was achieved by the random perturbation; however, the difference between the fooling rates for the random perturbation and the gradient-based attacks is not as large as one might anticipate. Indeed, even for perturbations of size between 0.06 and 0.08, the susceptibility of our classifier to random perturbations is roughly 50%. Further work is needed to understand whether this high susceptibility to random attack is a general characteristic of MAML fine-tuned models or an artifact of either the Omniglot dataset or our particular classifier. For example, one could train the exact same CNN architecture we used without the MAML initialization, or with more than single-shot training on the fine tune task, and see whether this susceptibility persists.

Transfer techniques seem to be effective in achieving high fooling rates for low values of epsilon (see Figure 1). For the majority of the epsilon values tested, it does not appear that there is much difference between attacking a model using its own gradient or attacking it using either the Meta-gradient or another fine-tuned gradient. A natural question arises as to whether the similar efficacy of these attacks is due to the high vulnerability of the classifier itself (as evinced by the effectiveness of the random perturbations) or due to some similarity in FGSM directions given by each gradient. Further work will explore this question.

Perhaps most surprisingly, for epsilons greater than roughly 0.04, the direct attack of a model using its own gradient was *outperformed* by attack using the meta-model gradient. Whether this is a robust feature of MAML-trained models or a statistical anomaly remains to be seen.

5 CONCLUSION

The impact of adversarial attacks on MAML trained meta-models and the task-specific models fine-tuned from these meta-models is far reaching. If it can be shown that adversarial attacks can transfer with ease between fine-tuned models, then future models taught via meta-learning may all be vulnerable to adversarial perturbations generated using a single fine-tuned model. On the other hand, if it can be shown that adversarial attacks have difficulty transferring between tasks, even with "fine-tuning", then this reveals that meta-learned models provide some sort of robustness against transfer of adversarial attacks.

In either case, the results of this study should motivate future work on adversarial robustness of meta-learning models. Recent work in deep learning security has aimed at finding ways to make models more robust to adversarial attacks, both by training phase defenses, such as adversarial distillation [5] [14], defensive distillation [1], and gradient blocking [13]; and by adversarial sample detection defenses, such as detectors [19] and reformers [10]. The steps we take towards understanding the vulnerabilities of MAML-based models to transferred attacks may lead to improved defenses methods against such attacks, potentially for both MAML and other meta-learning approaches.

REFERENCES

[1] N. Carlini and D. Wagner. 2016. Defensive Distillation is Not Robust to Adversarial Examples. *ArXiv e-prints* (July 2016). arXiv:cs.CR/1607.04311

[2] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. 2016. RL2: Fast Reinforcement Learning via Slow Reinforcement Learning. *arXiv preprint arXiv:1611.02779* (2016).

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv preprint arXiv:1703.03400* (2017).

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[5] G. Hinton, O. Vinyals, and J. Dean. 2015. Distilling the Knowledge in a Neural Network. *ArXiv e-prints* (March 2015). arXiv:stat.ML/1503.02531

[6] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[7] A. Kurakin, I. Goodfellow, and S. Bengio. 2016. Adversarial examples in the physical world. *ArXiv e-prints* (July 2016). arXiv:cs.CV/1607.02533

[8] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science* 350, 6266 (2015), 1332–1338.

[9] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into Transferable Adversarial Examples and Black-box Attacks. *CoRR* abs/1611.02770 (2016).

[10] D. Meng and H. Chen. 2017. MagNet: a Two-Pronged Defense against Adversarial Examples. *ArXiv e-prints* (May 2017). arXiv:cs.CR/1705.09064

[11] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2016. Universal adversarial perturbations. *arXiv preprint arXiv:1610.08401* (2016).

[12] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016).

[13] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman. 2016. Towards the Science of Security and Privacy in Machine Learning. *ArXiv e-prints* (Nov. 2016). arXiv:cs.CR/1611.03814

[14] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. 2015. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. *ArXiv e-prints* (Nov. 2015). arXiv:cs.CR/1511.04508

[15] S. Reed, Y. Chen, T. Paine, A. van den Oord, S. M. A. Eslami, D. Rezende, O. Vinyals, and N. de Freitas. 2017. Few-shot Autoregressive Density Estimation: Towards Learning to Learn Distributions. *ArXiv e-prints* (Oct. 2017). arXiv:1710.10304

[16] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*. 1842–1850.

[17] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *CoRR* abs/1312.6199 (2013).

[18] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. 2017. The Space of Transferable Adversarial Examples. *ArXiv e-prints* (April 2017). arXiv:stat.ML/1704.03453

[19] W. Xu, D. Evans, and Y. Qi. 2017. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. *ArXiv e-prints* (April 2017). arXiv:cs.CV/1704.01155