Few Is Enough: Task-Augmented Active Meta-Learning for Brain Cell Classification

Pengyu Yuan[1], Aryan Mobiny[1], Jahandar Jahanipour[1,2], Xiaoyang Li[1], Pietro Antonio Cicalese[1], Badrinath Roysam[1], Vishal Patel[3], Maric Dragan[2], and Hien Van Nguyen[1]

[1] University of Houston, Houston, Texas, USA
`pyuan2@uh.edu`
[2] National Institutes of Health, Bethesda, Maryland, USA
[3] Johns Hopkins University, Baltimore, Maryland, USA

Abstract. Deep Neural Networks (or DNNs) must constantly cope with distribution changes in the input data when the task of interest or the data collection protocol changes. Retraining a network from scratch to combat this issue poses a significant cost. Meta-learning aims to deliver an adaptive model that is sensitive to these underlying distribution changes, but requires many tasks during the meta-training process. In this paper, we propose a tAsk-auGmented actIve meta-LEarning (AGILE) method to efficiently adapt DNNs to new tasks by using a small number of training examples. AGILE combines a meta-learning algorithm with a novel task augmentation technique which we use to generate an initial adaptive model. It then uses Bayesian dropout uncertainty estimates to actively select the most difficult samples when updating the model to a new task. This allows AGILE to learn with fewer tasks and a few informative samples, achieving high performance with a limited dataset. We perform our experiments using the brain cell classification task and compare the results to a plain meta-learning model trained from scratch. We show that the proposed task-augmented meta-learning framework can learn to classify new cell types after a single gradient step with a limited number of training samples. We show that active learning with Bayesian uncertainty can further improve the performance when the number of training samples is extremely small. Using only 1% of the training data and a single update step, we achieved 90% accuracy on the new cell type classification task, a 50% points improvement over a state-of-the-art meta-learning algorithm.

Keywords: Meta learning · Active · Brain cell

1 Introduction

The ability of Deep Neural Networks (or DNNs) to generalize to a given target concept is dependent on the amount of training data used to generate the model. This is a significant limitation, as many real-world classification tasks

depend on a limited number of training samples for accurate classification. Various research groups have developed techniques that utilize different underlying principles to address this issue. These techniques include data augmentation [30] and generative models [12, 17, 38], which try to directly increase the number of training samples. Active learning is another technique which aims to select the most valuable samples to include in the training set [28, 29]. Finally, transfer learning allows the model to adapt to new and unseen tasks by using its own previous knowledge, often reducing the amount of data needed for generalization [4, 13, 25, 25].

Medical data is characterized by significant distribution shifts and small samples sets that negatively affect the quality of the generated model. In the cell classification task [37], there are several contributing factors for poor generalization; different biomarkers, unique cell morphologies, variations in stain intensity and image quality all contribute to the variability of the data. Each of these factors could be considered a unique parameter with which to create unique classification tasks. Traditional transfer learning methods pre-trained on various source tasks may not perform well; poor model initialization parameters coupled with unadjusted hyperparameters may cause the model to fall into a bad local minima. Mainstream transfer learning methods also require a time-consuming model retraining process [23]. In recent years, a more advanced model architecture called meta-learning has been developed to address these adaptability issues [34]. Meta-learning approaches try to generate a more robust model that can learn to quickly adapt to new tasks with minimal labeled samples. Although meta-learning does not require many labeled samples for each task, it requires many different tasks to effectively learn how to adapt; this may be a problem when the number of tasks is limited. This technique also selects training samples randomly for each new task, which may negatively impact the strength of the model if it is trained on easier samples.

In this paper, we utilize various strategies to create an adaptive framework called tAsk-auGmented actIve meta-LEarning (or AGILE) which allows the classifier to achieve high performance with few training samples and gradient updates for each new task. The experiments we perform on the brain cell type classification task show that the AGILE classifier can quickly adapt to new cell types by utilizing very few labeled training samples.

2   Related works

Brain cell type classification The brain is a highly complex organ made up of myriad different cell types, each with their own unique properties [7]. Gene expression experiments have highlighted cell type composition based on the expression value of markers for five major cell types: neurons, astrocytes, oligodendrocytes, microglia, and endothelial cells [6, 22]. Besides that, there are around 50-250 neuronal sub-cell types purported exist [2]. Different cell types may express different biomarkers or unique combinations of biomarkers, with some of then shared with other cell types. Correctly identifying the cell type using
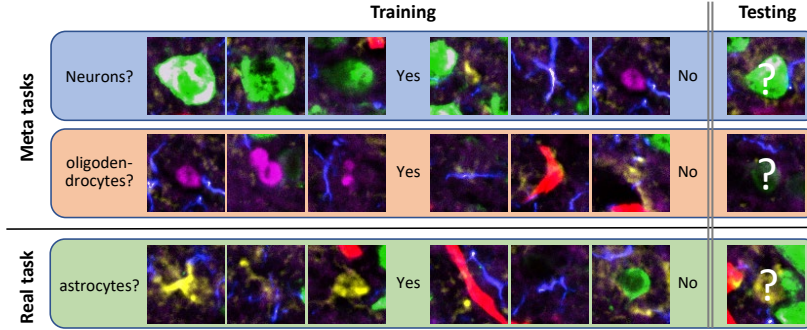
Fig. 1. Multi-task brain cell classification. Each task is a binary classification problem for a specific brain cell type. Cell to be classified is located in the center of the image. The model needs to be adapted to the unseen real task with few training samples.

these biomarkers is essential for many medical researches such as schizophrenia [31] and brain cell type specific gene expression [22]. It is easy to train a deep neural network to identify one cell type but it is not easy to scale the network for hundreds of new cell types. Our meta-learning framework aims to provide an adaptive model that can rapidly adjust itself to new classification tasks.

Meta-learning Meta-learning aims to study how meta-learning algorithms can acquire fast adaptation capability from a collection of tasks [3, 5, 26, 34]. Meta-learning often consists of a meta-learner and a learner that learn at two levels of different time scales [14, 27]. Santoro et al. Koch et al. [18], Vinyals et al. [35] and Snell et al. [33] proposed to learn a robust kernel function of feature embeddings to illustrate the similarities between different samples. Another popular approach is to directly optimize the meta-learner through the gradient descent [1,9,15,24]. Model agnostic meta-learning (MAML) [9] is proven to be one of the state-of-the-art approaches in the meta-learning field. These methods demonstrated human-level accuracy on many classification tasks. However most of them require a lot of tasks to train the meta-learner and it can not actively select training samples which might not be the optimal case in practice.

Active learning Active learning has been used to interactively and efficiently query information to achieve optimal performance for the task of interest. These methods select training samples based on information theory [20], ensemble approaches [21], and uncertainty measurements [16, 19]. However, these methods may not be effective for deep networks. Gal et al. [10, 11], Sener et al. [28] and Fang et al. [8] did a lot of studies in finding heuristics of annotating new samples for deep networks. However, few of them have been applied to domain adaptation or meta-learning problems. Woodward et al. [36] added an active part in one-shot learning but did not utilize the best meta-learning structure. In this
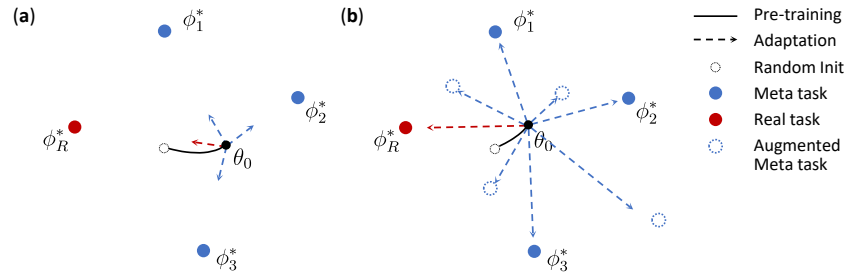
Fig. 2. Comparison of (**a**)transfer learning and (**b**)task-augmented meta-learning. (**a**) After pre-training on very few number of meta tasks, the model is pulled closer to the meta tasks and may be far from the real task. (**b**) By creating pseudo meta tasks to pre-train the meta-learning model, it gains the ability of adapting to the new tasks with only one gradient step updating.

paper, we combine the advantages of both meta-learning and active-learning to get a fast adaptive model which can use the fewest data to achieve a good performance.

3   Methodology

Consider a dataset consisting Q samples: $\mathcal{D} = \{(\mathbf{x}_q, \mathbf{y}_q)\}_{q=1}^{Q}$, where $(\mathbf{x}_q, \mathbf{y}_q)$ is an input-output pair sampled from the joint distribution $P(\mathcal{X}, \mathcal{Y})$. A task can be specifically defined by learning a model $f_\phi(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$ which is parameterized by ϕ to maximize the conditional probability $P_\phi(\mathcal{Y}|\mathcal{X})$. Thus whenever there is a change in the conditional distribution which is mainly caused by distribution shifts in $\mathcal{X}$ or $\mathcal{Y}$, it can be viewed as a new task. When dealing with mutiple tasks drawn from $P(\mathcal{T})$, each task $\mathcal{T}_i$ is associated with a unique dataset $\mathcal{D}_i$. We split these tasks into two parts: meta tasks $\mathcal{T}_{\mathrm{meta}}$ and real tasks $\mathcal{T}_{\mathrm{real}}$. Real tasks mean the model performance on these tasks is what we really care about and Meta tasks are what we used to pre-train the model. If there is no model adaptation, then the meta tasks are not needed. For each task $\mathcal{T}_i$, we have a train/test split $\mathcal{D}_i^{\mathrm{train}}/\mathcal{D}_i^{\mathrm{test}} \subset \mathcal{D}_i$. The goal for task $\mathcal{T}_i$ is to learn a set of parameters ϕ_i from the $\mathcal{D}_i^{\mathrm{train}}$ to get the minimal loss on the test data, i.e. $\mathcal{L}(\phi_i, D_i^{test})$.

Our proposed framework AGILE has two phases, a task-augmented meta-learning method which learns to generate a strong model initialization which is sensitive to data distribution changes by using $\mathcal{T}_{\mathrm{meta}}$, and an active learning process which selects the most informative samples using Bayesian dropout uncertainties when apply the adaptive model on $\mathcal{T}_{\mathrm{real}}$.

3.1   Task-augmented meta-learning

The phase I of our AGILE framework is the task-augmented meta-learning module. For the meta-learning setting, we have a learner which operates at fast timescale and parameterized by $\phi \in \Phi$, and a meta-learner at a slower time-scale

parameterized by $\theta \in \Theta$. The goal of meta-learning is to learn meta-parameters θ that can produce good task-specific parameters ϕ for all M tasks after the fast adaptation:

$$\theta^* = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}\left(\mathcal{A}dapt(\theta, \mathcal{D}_i^{\text{train}}), \mathcal{D}_i^{\text{test}}\right) \qquad (1)$$

where $\mathcal{A}dapt()$ function is an adaptation step completed by the learner. We employ model agnostic meta-learning (MAML) [9] which initializes the model at θ then updates it using training data for each task $\mathcal{D}_i^{\text{train}}$ as follows:

$$\phi_i \equiv \mathcal{A}dapt(\theta, \mathcal{D}_i^{\text{train}}) = \theta \overset{Z}{-} \alpha \nabla_\theta \mathcal{L}(\theta, \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{K})), \quad (\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_i^{\text{train}} \qquad (2)$$

where $\mathcal{L}$ is the loss function, α the learning rate for the learner, and $\overset{Z}{-}$ a shorthand notation for running a Z-step gradient descent which is relatively fast. A fixed number of K class-balanced samples $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{K}$ randomly sampled from $\mathcal{D}_i^{\text{train}}$ are used to update the learner for every iteration.

Task transformation Meta-learner is trained on the meta tasks. Our experiments show that when the number of meta tasks is not big enough, it can not obtain the general fast adaptation ability on $\mathcal{T}_{\text{real}} \sim P(\mathcal{T})$. Similar as data augmentation, we applied a set of task-augmentation functions $\{\mathcal{G}_l\}_{l=1}^{L} : \mathcal{T} \to \mathcal{T}'$ on existing tasks to create new tasks. Because the task is specifically determined by the conditional probability $P(\mathcal{Y}|\mathcal{X})$, either change $\mathcal{X}$ or $\mathcal{Y}$ will lead to a new task. The task-augmentation functions we applied are (1) Flipping the label (2) Shuffling the order of input channels (3) Rotating the images. Considering the input image with c biomarkers $\mathbf{x} \in \mathbb{R}^{w \times h \times c}$ and the binary label $y \in \mathbb{R}$, flipping labels is achieved by:

$$y' = z(1 - y) + (1 - z)y, \quad \text{where } z \sim \text{Bernoulli}(p_f) \qquad (3)$$

where p_f is the probability of flipping the label. Shuffling the input channels is selected with a probability of p_s. By Constructing c different one-by-one kernels $\{\mathbf{s}_{ij}\}_{i=1}^{c} \in \mathbb{R}^{1 \times 1 \times c}$ where $\mathbf{s}_{ij} = 1$ only if the jth bio-marker is placed at i^{th} channel after the random shuffling, the shuffled images are obtained by the convolution:

$$\mathbf{x}' = \mathbf{x} * \mathbf{s}_{ij}, \quad i, j = 1, 2, 3 \dots c \qquad (4)$$

The third task augmentation method we used is to rotate the images for $90°, 180°$ or $270°$ with the probability of p_r. In the experiment, we set $p_f = p_s = p_r$. Theoretically any data augmentation techniques can be applied here to slightly change the distribution of $\mathcal{X}$. By increasing the diversity of the meta tasks, the meta-learner can be well trained to extract useful features after adapting to training data in any tasks. The comparison between task augmented meta-learning and plain transfer learning is illustrated in Figure 2.

3.2   Active learning with Bayesian uncertainty

The phase II of AGILE is to apply the pre-trained model with active learning. Previous meta-learning methods consider training and testing the meta-learning algorithm in the exactly same manner, which might not be practical. They fixed the number of training samples for each task. But for the real task $\{\mathcal{T}_{\text{real}}\}_{j=1}^{N}$, this number may vary, especially when users want to annotate some new samples to improve the performance of the model. When the number of training samples is extremely small, how to unsupervisedly select the most informative samples to annotate is an important question. First, we use a random number of samples to update the learner during the meta-learning training process, so that the model is forced to learn with a different training size.

$$\phi_i = \theta \overset{Z}{-} \alpha \nabla_\theta \mathcal{L}(\theta, \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{\widetilde{K}})), \quad (\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_i^{\text{train}} \qquad (5)$$

where $\widetilde{K}$ is varied between 1 sample from each class to the maximum number of samples K allowed by the label budget. Second, we applied Bayesian dropout [10, 32] to get the uncertainties of the predictions over all unlabeled samples, which represents the current belief of the learner on the predictions:

$$H\left(\mathbf{y}|\mathbf{x}, \mathcal{D}_j^{\text{train}}\right) = -\sum_{\mathbf{y} \in \mathcal{Y}} p_{\text{MC}}\left(\mathbf{y}|\mathbf{x}\right) \log p_{\text{MC}}\left(\mathbf{y}|\mathbf{x}\right), \quad \text{where } (\mathbf{x}, \mathbf{y}) \in \mathcal{D}_j^{\text{test}} \qquad (6)$$

$P_{\text{MC}}\left(\mathbf{y}|\mathbf{x}\right)$ is the approximation of Bayesian inference with Monte Carlo integration $\frac{1}{T}\sum_{t=1}^{T} p_{\phi_j}\left(\mathbf{y}|\mathbf{x}\right)$, where $p_{\phi_j}\left(\mathbf{y}|\mathbf{x}\right)$ is the conditional probability of predicted class for input $\mathbf{x}$ in task j and T is the number of Monte-Carlo experiments. The higher entropy H is, the higher uncertainty is in this prediction. For this Bayesian deep learning network, dropout is turned on both for the training and test time on $\mathcal{T}_{\text{real}}$ with a drop rate of 0.1. Based on the uncertainties, the model will only select hard samples for training which significantly improves the performance.

4   Experiments and Results

For this brain cell type classification problem, we collected 4000 cells from 5 major cell types imaged from rat brain tissue sections: neurons, astrocytes, oligodendrocytes, microglia, and endothelial cells. There are 800 cells for each cell type. 7 biomarkers are used as the feature channels: DAPI, Histones, NeuN, S100, Olig 2, Iba1 and RECA1. DAPI and Histones are used to indicate the location of the cells while others are biomarkers for classification of specific cell types. Each cell is located in the center of a patch with a size of 100×100 pixels. As shown in Figure 1, for each cell type, we have a binary classification task, 800 cells from this cell type and 800 from others. The model needs to correctly identify whether the given cell is this cell type. And it has no prior knowledge about the usage of each biomarker. Out of 1600 cells, 60% of them are used as potential training samples and the rest are used for the test. Binary classification

Table 1. Methods configuration comparison which differ mainly in the data they use and the training framework. Meta-learning methods are supposed to perform well with few training samples and little training time. (# means the number of)

Methods	Use data			in Real-train		# Meta tasks
	Meta-train	Meta-test	Real-train	# samples	# gradient updates	
Vanilla_limit	-	-	✓	16 (1%)	100	0
Vanilla_full	-	-	✓	960 (60%)	100	0
Transfer	✓	-	✓	16 (1%)	100	3
MAML	✓	✓	✓	16 (1%)	1	3
AGILE(phase I)	✓	✓	✓	16 (1%)	1	many
AGILE(phase II)	✓	✓	✓	16 (1%)	1	many
AGILE(phase II)	✓	✓	✓	160 (10%)	1	many

Table 2. Quantitative results of different methods. Original method use all available training data (60%) and act as the upper bound while task-augmented MAML method get the highest accuracy using very few training data (1%).

Methods (Size %)	Precision	Recall	F1-score	Accuracy($\pm$ Std)	CI$_{95}$
Vanilla_limit (1%)	0.642	0.622	0.632	0.637($\pm$0.062)	0.632 - 0.642
Vanilla_full (60%)	0.937	**0.965**	**0.951**	**0.950**($\pm$0.021)	0.948 - 0.952
Transfer (1%)	0.447	0.433	0.440	0.449($\pm$0.085)	0.449 - 0.456
MAML (1%)	0.408	0.402	0.405	0.409($\pm$0.030)	0.406 - 0.412
AGILE(phase I) (1%)	0.791	0.790	0.791	0.791($\pm$0.054)	0.786 - 0.796
AGILE(phase II) (1%)	0.883	0.926	0.904	0.902($\pm$0.048)	0.898 - 0.906
AGILE(phase II) (10%)	**0.950**	0.951	**0.951**	**0.950**($\pm$0.044)	0.946 - 0.954

for neurons, oligodendrocytes and microglia are considered as the meta tasks. And the real tasks are to identify the remaining two cell types.

We used three other methods for comparison. The first is the Vanilla method which is to train a network from scratch without any adaptation. We further split it into Vanilla_limit and Vanilla_full methods which use limited and all available training data respectively. They are acting like the lower bound and the upper bound of the performances. The second is the transfer learning model, which means the model is pre-trained on 3 meta tasks and then fine-tuned on 2 real takes. Both of these are trained with 100 gradient updates to ensure the model convergence. The third method is the plain MAML model [9] with limited data and only 1 gradient update for the purpose of fast adaptation with few training data. We chose MAML as our meta-learning baseline because it achieved highly competitive performances compared to other meta-learning methods [9]. All configuration differences are shown in Table 1. Note that all methods share the same neural network structure which has 4 convolution blocks. Each block consists of a convolutional layer with 32 filters of size 3*3, a batch normalization layer, and a max-pooling layer. Relu activation is used after each convolutional layer. After the last convolution block, a dense layer is added to project the 32 feature maps to only two classes. And all methods have the same learning rate $\alpha = 0.01$ for the classifier. The learning rate for meta-learner is 0.001. Adam
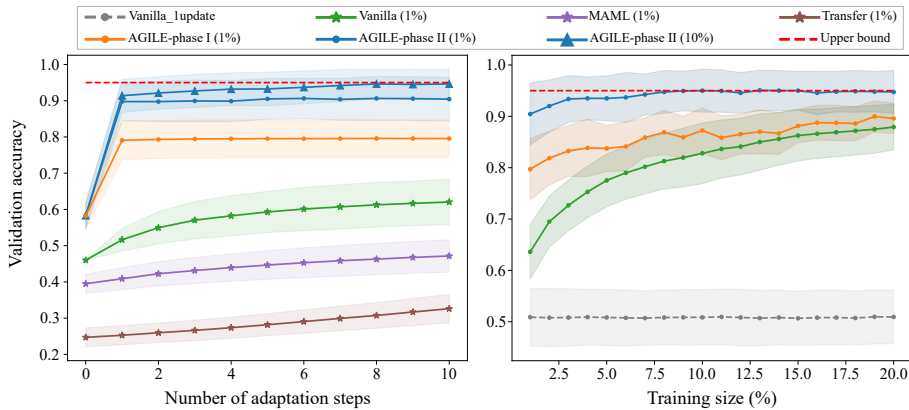
Fig. 3. (**a**) Comparison of different methods when adapting to new real task with only 1% labeled training data, AGILE method shows its fast adapting ability. Upper bound is obtained by training on all 60% training data with Vanilla method. (**b**) The impact of the training size on the AGILE method and the vanilla method.

optimizer is used for all optimization. Training iterations are set as 12000 for all methods. 12 replaced tasks are used in one iteration during the training on meta-tasks. The only differences between these methods are the data and the training/adaptation framework they use.

The quantitative results on the real tasks are summarized in Table 2. The upper bound we can get by using all training data is 95% classification accuracy while the lower bound is 63.7%. Plain transfer learning method and MAML methods are even worse than the lower bound because they were stuck in the local minimum after pre-trained on only 3 meta tasks. Our proposed task-augmented meta-learning is able to quickly adapt with few training samples and gradient updates. With only 16 training samples, the AGILE method can reach 90% accuracy. With 160 training samples, AGILE reaches the upper bound. The 95% confidence interval are presented in the Table 2.

The adaptation processes are shown in Figure 3 (**a**). The solid line is the mean value and the shaded area shows the variance. Without any prior knowledge, the original model starts with the classification accuracy around 50%. Transferring an existing model has the lowest starting point because images labeled as "1" in one task should be labeled as "0" in other tasks. The original MAML method also does not help for this brain cell type binary classification problem. Our results show that MAML is unable to adapt effectively due to the small number of training tasks. In contrast, AGILE model has a sharp increase in accuracy after 1 gradient update. Note that although AGILE model is trained for maximal performance after one gradient step, it continues to improve with additional gradient steps.

Figure 3 (**b**) illustrate the relationship between the training size and the performances of AGILE method and the baseline methods. AGILE performs ex-

tremely well when the training size is small. Since AGILE selects samples based on their uncertainties, the newly added training samples might not be class-balanced. This is more practical because difficult classes require more examples for training. Our strategy contrasts against current meta-learning setting where all the training set is class-balanced by default. AGILE method reachs validation accuracy of 95% with 160 samples. Meantime, the vanilla model with 100 gradient updates get the accuracy around 84% and the vanilla model with only 1 update basically cannot learn anything.

5   Conclusion

In this paper, we proposed a fast adaptation framework AGILE combining data augmentation, meta learning and active learning to deliver a model which is sensitive to the data distribution/task changes and able to adjust itself with few training samples and few updating steps. The results show that only 10% of training data and 1 gradient update are enough to get the best performance on identifying unseen brain cell type. AGILE can be used in many diagnose systems or detection algorithms which have to deal with various input data.

References

1. Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., De Freitas, N.: Learning to learn by gradient descent by gradient descent. In: Advances in neural information processing systems. pp. 3981–3989 (2016)
2. Ascoli, G.A., Donohue, D.E., Halavi, M.: Neuromorpho. org: a central resource for neuronal morphologies. Journal of Neuroscience **27**(35), 9247–9251 (2007)
3. Bengio, S., Bengio, Y., Cloutier, J., Gecsei, J.: On the optimization of a synaptic learning rule. In: Preprints Conf. Optimality in Artificial and Biological Neural Networks. vol. 2. Univ. of Texas (1992)
4. Bengio, Y.: Deep learning of representations for unsupervised and transfer learning. In: Proceedings of ICML workshop on unsupervised and transfer learning. pp. 17–36 (2012)
5. Bengio, Y., Bengio, S., Cloutier, J.: Learning a synaptic learning rule. Université de Montréal, Département d'informatique et de recherche . . . (1990)
6. De Jager, P.L., Srivastava, G., Lunnon, K., Burgess, J., Schalkwyk, L.C., Yu, L., Eaton, M.L., Keenan, B.T., Ernst, J., McCabe, C., et al.: Alzheimer's disease: early alterations in brain dna methylation at ank1, bin1, rhbdf2 and other loci. Nature neuroscience **17**(9), 1156–1163 (2014)
7. Elizabeth A. Weaver II, H.H.D.: Cells of the brain. `https://www.dana.org/article/cells-of-the-brain/` (August 2019)

8. Fang, M., Li, Y., Cohn, T.: Learning how to active learn: A deep reinforcement learning approach. arXiv preprint arXiv:1708.02383 (2017)
9. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1126–1135. JMLR. org (2017)
10. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059 (2016)
11. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1183–1192. JMLR. org (2017)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
13. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: 2011 international conference on computer vision. pp. 999–1006. IEEE (2011)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
15. Hochreiter, S., Younger, A.S., Conwell, P.R.: Learning to learn using gradient descent. In: International Conference on Artificial Neural Networks. pp. 87–94. Springer (2001)
16. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2372–2379. IEEE (2009)
17. Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M.: Semi-supervised learning with deep generative models. In: Advances in neural information processing systems. pp. 3581–3589 (2014)
18. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop. vol. 2. Lille (2015)
19. Li, X., Guo, Y.: Adaptive active learning for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 859–866 (2013)
20. MacKay, D.J.: Information-based objective functions for active data selection. Neural computation **4**(4), 590–604 (1992)
21. McCallumzy, A.K., Nigamy, K.: Employing em and pool-based active learning for text classification. In: Proc. International Conference on Machine Learning (ICML). pp. 359–367. Citeseer (1998)
22. McKenzie, A.T., Wang, M., Hauberg, M.E., Fullard, J.F., Kozlenkov, A., Keenan, A., Hurd, Y.L., Dracheva, S., Casaccia, P., Roussos, P., et al.: Brain cell type specific gene expression and co-expression network architectures. Scientific reports **8**(1), 1–19 (2018)
23. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
24. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning (2016)
25. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: European conference on computer vision. pp. 213–226. Springer (2010)
26. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: One-shot learning with memory-augmented neural networks. arXiv preprint arXiv:1605.06065 (2016)

27. Schweighofer, N., Doya, K.: Meta-learning in reinforcement learning. Neural Networks **16**(1), 5–9 (2003)
28. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A coreset approach. arXiv preprint arXiv:1708.00489 (2017)
29. Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009)
30. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**(1), 60 (2019)
31. Skene, N.G., Bryois, J., Bakken, T.E., Breen, G., Crowley, J.J., Gaspar, H.A., Giusti-Rodriguez, P., Hodge, R.D., Miller, J.A., Muñoz-Manchado, A.B., et al.: Genetic identification of brain cell types underlying schizophrenia. Nature genetics **50**(6), 825–833 (2018)
32. Smith, L., Gal, Y.: Understanding measures of uncertainty for adversarial example detection. arXiv preprint arXiv:1803.08533 (2018)
33. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in neural information processing systems. pp. 4077–4087 (2017)
34. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artificial intelligence review **18**(2), 77–95 (2002)
35. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in neural information processing systems. pp. 3630–3638 (2016)
36. Woodward, M., Finn, C.: Active one-shot learning. arXiv preprint arXiv:1702.06559 (2017)
37. Zeng, H., Sanes, J.R.: Neuronal cell-type classification: challenges, opportunities and the path forward. Nature Reviews Neuroscience **18**(9), 530 (2017)
38. Zhu, X., Goldberg, A.B.: Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning **3**(1), 1–130 (2009)