Reprogramming Language Models for Molecular Representation Learning

Ria Vinod∗
Department of Electrical Engineering and Computer Science
University of California, Berkeley
ria.vinod@berkeley.edu

Pin-Yu Chen
IBM Research
Yorktown Heights, NY
pin-yu.chen@ibm.com

Payel Das
IBM Research
Yorktown Heights, NY
daspa@us.ibm.com

Abstract

Recent advancements in transfer learning have made it a promising approach for domain adaptation via transfer of learned representations. This is especially relevant when alternate tasks have limited samples of well-defined and labeled data, which is common in the molecule data domain. This makes transfer learning an ideal approach to solve molecular learning tasks. While Adversarial Reprogramming has proven to be a successful method to repurpose neural networks for alternate tasks, most works consider source and alternate tasks within the same domain. In this work, we propose a new algorithm, Representation Reprogramming via Dictionary Learning (R2DL), for adversarially reprogramming pretrained language models for molecular learning tasks, motivated by leveraging learned representations in massive state of the art language models. The adversarial program learns a linear transformation between a dense source model input space (language data) and a sparse target model input space (e.g., chemical and biological molecule data) using a k-SVD solver to approximate a sparse representation of the encoded data, via dictionary learning. This method achieves the baseline established by state of the art toxicity prediction models trained on domain-specific data using only a standard well-trained text classifier, thereby establishing avenues for domain-agnostic transfer learning for tasks with molecule data.

1 Introduction

Deep learning has proven to be an extremely successful tool for various applications for the natural sciences. While works like MoleculeNet have made significant progress in publishing benchmarks on various molecular learning tasks, curating substantial, well-structured and labeled molecule datasets remains a critical constraint in training high performing models from scratch to establish baselines for a multitude of tasks [1]. A lack of substantive training datasets motivates transfer learning to be a natural approach to solve problems in the molecule data domain, as it has proven to be a successful technique to solve new tasks using learned representations from source model domains [2]. In this paper, we consider an empirically successful transfer learning method, adversarial reprogramming, where a learned function adversarially perturbs input samples to the model such that the model can perform a task chosen by the adversary [3]. The motivations of the transfer

∗Work done during internship at IBM Research
learning approach lie in the recent successes of powerful general language models [4], and being able to leverage these learned representations when applied to molecule data that can be treated as sequence data. However, the adversarial reprogramming method is well suited to models operating within a continuous input space (such as images), but discrete input spaces require that we learn a mapping between source-model space and target-model space. Reprogramming of text classifiers and language models has been explored [5], as well as knowledge injection for general language models [6], but such approaches do not investigate mappings between domains that require a very high representational capacity (from language data to molecule data). This shift in domain poses a significant challenge through the reduction of dimensionality of the input space. We end up with an overdetermined system, as we have more observations in the source-model input space than in the target-model input space. To address this, we use a dictionary learning approach to encode a sparse representation of the embeddings of the input data to the pre-trained language model (text classifier). Training the adversarial program (AP) via dictionary learning enables us to approximate a linear transformation (mapping) between the input spaces of the source and target models, it learns the optimal coefficients of the atoms in our molecule data to represent the dictionary. When compared to the baselines of training AMP models from scratch, repurposing a sentiment classifier via R2DL outperforms the baseline for AMP/non-AMP (90.01 % vs 88.0 %) and achieves approximately the same progress on toxicity prediction (89.34 % vs 93.7 %).

2 Related Work

2.1 Adversarial Machine Learning and Reprogramming

Adversarial attacks on ML models in discrete input spaces have been found to cause models to misclassify input samples by manipulating tokens in sequence data [7]. Adversarial Reprogramming (AR) is a technique [3] that is able to repurpose a model (without significant changes to the architecture and parameters) by training an adversary to optimally transform input data such that we can choose the output of the model. This method has been proven to be successful in both, white-box and black-box settings [8]. The depth and the size of large general language models make them ideal candidates to be adversarially reprogrammed, as modifying the internal architecture or finetuning over 1 billion parameters is infeasible for a traditional transfer learning approach. The work in [5] assumes an exposed model, where they use the learned parameters of the pre-trained source model to generate context-based vocabulary map. However, the hypothesis of comparable vocabulary size does not hold when the source data are English vocabularies (on the order of ∼10 million) and the target data are amino acids (on the order of ∼20). To that end, we introduce dictionary learning to train our AP.

2.2 Dictionary Learning

Typical amino acid/nucleotide representations of biological molecules or widely used SMILES representation of chemical molecules have fewer distinct tokens than that of language data. The significant reduction in dimensionality of the embedded space of English data to molecule data makes finding a mapping non-trivial. Results in [9] demonstrate that that representation learning algorithms have an advantage in transfer learning methods as they capture features relevant to alternate tasks. We require a mapping with high representational capacity. Work in [10] demonstrates that a sparse encoding is distributed and highly expressive: we can represent \( O(2^k) \) input regions with only \( O(N) \) parameters. Distributed representations can be clustered to extract relevant features where component extraction algorithms can find the optimal representation (a dictionary). We use a k-SVD approximation algorithm [11] to mitigate computational expenses.

3 Representation Reprogramming via Dictionary Learning (R2DL):

Algorithm and Method

3.1 Problem Formulation

We are given a pretrained classifier, \( X \), a source-task dataset \( \{V_S\}_{i=1}^n \) and target-task dataset \( \{V_T\}_{j=1}^m \). The embedded matrices are \( V_T \) and \( V_S \) respectively. We can encode an output label mapping function \( h : l_{V_S} \mapsto l_{V_T} \). We then aim to train an adversarial program (AP) \( f_\theta \) that finds
the optimal coefficients θ of our atoms in VT to represent a sparse encoding of the dictionary, VS such that VT = VSθ. The AP fθ is used to perform the target task through the transformation h(X(fθ(t))) where ti is a molecule data sample. While we do not make any modification the parameters or architecture of X, we assume access to the gradient ∇θX for loss evaluation during training.

3.2 Adversarial Program

To reprogram the pretrained classifier, we use a similar structure as introduced in [5], fθ : ti → si where ti ∈ VT and si ∈ VS. Dimension of the input space of VS and VT is |VS|, and |VT| respectively, where |VT| ≪ |VS|. The AP fθ is parametrized by θ ∈ R1×|VS|, which represents the coefficients of the atoms in VT such that VT = VSθ and VS is our dictionary. The observation of |VT| ≪ |VS| requires that our mapping has high representational capacity, so we encode a sparse representation of VS, to extract relevant features from the embeddings of the source-model vocabulary {VS}∞i=1 for the alternate task. To that end, approximate the dictionary, we use a k-SVD solver to optimize over the cross entropy loss for updates to θ.

3.3 k-SVD

We define θ ∈ R|VS|×|VT| and use θt to denote its t-th column, where a signal (embedding vector) vt, can be represented as a sparse linear combination of the signal atoms of columns of VS, vt = VSθt. vt is the representation of the AMP input sample in the dictionary space and satisfies ||vt − VSθt|| ≤ ϵ. An exact solution θ∗t such that vt = VSθ∗t is computationally expensive to find, and is subject to various convergence traps, so for the purpose of our efficient AP approach we approximate vt ≈ VSθt by limiting the number of iterations to converge to a solution for vt. Our optimization problem for finding the optimal sparse representation of input data samples in VT is then minimize ||θ||0 subject to ||vt − VSθt|| ≤ ϵ to enforce a sparse solution [11], where || · ||0 denotes the l0 norm. While algorithms exist to choose an optimal dictionary (an exact solution to k-SVD) that can be continually updated [11], we penalize computational expense over performance for the purpose of maintaining an efficient solution (at the cost of statistically insignificant improvements in accuracy) by using a predetermined number of steps for an approximate solution, VS that encodes the atoms of VT. θ is then used to evaluate the cross entropy loss on h(X(fθ(VT))), which will be updated in the AP fθ.

4 Experiments

Algorithm 1: Representation Reprogramming via Dictionary Learning (R2DL)

Input: Pretrained sentiment classifier X, source model training data {VS}n i=1, target model training data {VT}m j=1, maximum number of iterations T1 for updates to θ, number of iterations T2 for k-SVD, output-label mapping function h(·), step size {αi}T1 i=1

Output: Optimal adversarial program parameters θ

1. Random initialization of θ;
2. Define objective function for k-SVD from (1);
3. while t1 ≤ T1 do
   4. while t2 ≤ T2 do
      use approximate k-SVD to solve VT ≈ VSθ;
      t2 ← t2 + 1;
   end
   #Loss evaluation for θ :
   θ ← θ − αt1 · ∇θh(X(fθ(VT)));
   t1 ← t1 + 1;
end

4.1 Baseline

To benchmark the performance of R2DL we compare it with the current established benchmark of a trained classifier using the same AMP training data set [12].
### 4.2 Restricted Training Data Setting

To further investigate the efficacy of the transfer learning approach, we compare the performance of R2DL versus the model trained from scratch with AMP data, with a restricted training data set. The test accuracies indicate that R2DL performs better when fewer labeled training data samples are available. Tables 2 and 3 show that when trained in a setting with fewer labeled data samples, R2DL outperforms the train from scratch method after the threshold of approximately 5000 samples. Below 5000 samples, both methods approximate random prediction, with R2DL not successfully transferring any learned representations.

#### Table 1: Baseline AMP Classification (Train from Scratch)

| Attribute          | Data-Split | Accuracy | Screening Threshold |
|--------------------|------------|----------|---------------------|
|                    | Train | Valid | Test | Majority Class | Test |
| {Toxic, non-Toxic} | 8153  | 1019  | 1020 | 82.73       | 93.7   | -1.573 |
| {AMP, non-AMP}     | 6489  | 811   | 812  | 82.68       | 88.0   | 7.944  |

#### 4.3 Repurposing Sentiment Classifiers

BERT and GPT-3 are 2 of the most common powerful language models, that generalize well across various NLP tasks, dealing with language (sequence) data. In this experiment, we use BERT, a bidirectional transformer, tuned for the sentiment classification task on the IMDB movie review dataset. As R2DL assumed access to the gradients of the source model ($\nabla X$), this approach works in a semi-black box setting given that we access but do not modify the internal architecture. In this task, we use sentiment classification as the source task, in which there are 2 output classes (positive, negative), and AMP toxicity classification as the target task (toxic, non-toxic). The output-label mapping $h$ is then a simple 1-1 correspondence between (positive, toxic) and (negative, non-toxic). The input data of the source model (BERT) is tokenized on a word-level which form the atoms for our dictionary representation of $V_s$. The input data to the target task, AMP sequences, are tokenized on a character level with only 7 distinct tokens. The embeddings of a source vocabulary token, $v_s$, is then represented as a weighted combination of the atoms of the AMP sequence tokens. Using the $l^0$ norm in our objective function, 100 k-SVD iterations and $\epsilon = 0.045$, we are able to achieve accuracy

\[ \text{accuracy} = 25000 \text{ AMP training samples and below were excluded from the results below as they showed statistically insignificant test accuracy.} \]

#### Table 2: Restricted Data Setting: Toxicity Prediction

| Task               | AMP Sequences Training Samples | R2DL Test Accuracy | Bi-LSTM Test Accuracy (train from scratch) |
|--------------------|--------------------------------|--------------------|-------------------------------------------|
| Toxicity Prediction| 5000                           | 42.12              | 37.34                                     |
| Toxicity Prediction| 6000                           | 62.98              | 49.62                                     |
| Toxicity Prediction| 7000                           | 86.23              | 82.78                                     |
| Toxicity Prediction| 8153                           | 89.34              | 93.7                                      |

#### Table 3: Restricted Data Setting: AMP Prediction

| Task     | AMP Sequences Training Samples | R2DL Test Accuracy | Bi-LSTM Test Accuracy (train from scratch) |
|----------|--------------------------------|--------------------|-------------------------------------------|
| AMP Prediction | 3500                            | 59.82              | 64.52                                     |
| AMP Prediction | 4500                            | 72.76              | 68.41                                     |
| AMP Prediction | 5500                            | 84.17              | 74.34                                     |
| AMP Prediction | 6489                            | 90.01              | 88.0                                      |
on the order of the benchmark when trained from scratch in table 1. Table 4 shows accuracies for sentiment classification source models and the target task, and Table 5 shows accuracies as we increase the number of k-SVD iterations for a convergent solution.

| Source Model                        | AMP Sequence Samples | k-SVD Iterations | Training Accuracy | Test Accuracy |
|-------------------------------------|----------------------|------------------|-------------------|---------------|
| BERT (Bidirectional Transformer)    | 8153                 | 100              | 74.78             | 87.23         |
| BERT (Bidirectional Transformer)    | 8153                 | 250              | 76.23             | 86.93         |
| Bi-LSTM                             | 8153                 | 100              | 68.34             | 81.25         |

From Table 5, we see that R2DL achieves the performance (89.34 %) of the baseline (93.70 %) when the pretrained classifier is a bidirectional transformer. Intuitively, we can understand why more expressive models have an increased representational capacity that can be leveraged to transfer relevant features from the pretrained classifier. Additionally, more precise k-SVD solutions beyond 100 iterations are not correlated with increased performance, and only increase computational cost, making this approach less efficient.

5 Conclusion

This work formalizes the argument for finding sparse representations of dense input data for cross domain transfer learning. In a setting where access to well structured and labelled data is limited, we can leverage representations in deep models through adversarial reprogramming and approximating a sparse coding. This approach demonstrated a either higher or comparable performance across different tasks when compared to training from scratch, but at a much lower cost and also performed better on fewer target task training samples. Our results provide new insights for domain-agnostic transfer learning, and establishes avenues for several molecular learning tasks that have been constrained by a lack of access to well-defined datasets.

6 Future Work

Immediate plans for this work include evaluating cost effectiveness of this approach with analysis of computational complexity and time. Future plans for this work extend to representation learning tasks for molecules with multi-dimensional structures (Simplified molecular-input line-entry system - SMILES). We also plan to explore meta-learning capabilities for generalization with respect to various molecular learning tasks, variable length molecule sequences, and variable sequence structures. Planned target tasks in the molecule domain include graph generation and genetic code classification.

References

[1] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: A benchmark for molecular machine learning. CoRR, abs/1703.00564, 2017.

[2] S. J. Pan and Q. Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010.
[3] Gamaleldin F. Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. 2019.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805, 2018.

[5] Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov, and Farinaz Koushanfar. Adversarial reprogramming of sequence classification neural networks. 2019.

[6] Anne Lauscher, Olga Majewska, Leonardo F. R. Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš. Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers. arXiv preprint arXiv:2005.11787, 2020.

[7] Nicolas Papernot, Patrick D. McDaniel, Ananthram Swami, and Richard E. Harang. Crafting adversarial input sequences for recurrent neural networks. CoRR, abs/1604.08275, 2016.

[8] Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. 2020.

[9] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. CoRR, abs/1206.5538, 2012.

[10] Ian Goodfellow, Honglak Lee, Quoc V. Le, Andrew Saxe, and Andrew Y. Ng. Measuring invariances in deep networks. pages 646–654, 2009.

[11] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing, 54(11):4311–4322, 2006.

[12] Payel Das, Tom Sercu, Kahini Wadhawan, Inkit Padhi, Sebastian Gehrmann, Flaviu Cipcigan, Vijil Chenthamarakshan, Hendrik Strobelt, Cicero dos Santos, Pin-Yu Chen, et al. Accelerating antimicrobial discovery with controllable deep generative models and molecular dynamics. arXiv preprint arXiv:2005.11248, 2020.