

УДК 004.89

ИСПОЛЬЗОВАНИЕ ГЕНЕРАТИВНЫХ ТЕСТОВ ДЛЯ ПРОВЕРКИ УСТОЙЧИВОСТИ ПРЕДОБУЧЕННЫХ МОДЕЛЕЙ

Жигальцов Д.А., Оператор научной роты, III курс
Военная академия связи имени Маршала Советского Союза С. М. Буденного,
г. Санкт-Петербург

По мере расширения сферы применения предобученных моделей в реальных задачах, критически важным становится не только обеспечение высокой точности предсказаний на стандартных выборках, но и способность моделей сохранять корректное поведение в условиях отклонения от обучающего распределения, нестандартных входов и потенциально атакующих воздействий. Предобученные модели – будь то языковые трансформеры, визуальные сверточные сети или мультимодальные архитектуры – зачастую оказываются уязвимыми перед входами, не предусмотренными в тренировочном корпусе, а значит – перед неожиданными сдвигами, редкими шаблонами, синтетическими или злокачественными вариациями. Для выявления таких сценариев и оценки устойчивости моделей к ним всё большую роль играют генеративные тесты, т.е. автоматизированные методы построения входов с целевой степенью отклонения от нормы, используемые для стресс-тестирования ИИ-систем.

Генеративные тесты (generative tests) представляют собой формализованный подход, при котором входные данные для модели не отбираются из статического набора, а создаются динамически, часто с учётом определённых гипотез о потенциальных уязвимостях или с заданием условий, к которым модель ранее не была адаптирована. В отличие от традиционного тестирования, где метрика качества вычисляется на фиксированном тестовом корпусе, генеративное тестирование позволяет моделировать поведение модели в гипотетических, но правдоподобных ситуациях, а также выявлять уязвимости, которые не проявляются при обычной эксплуатации. Это особенно важно для крупных предобученных моделей, обученных на обобщённых, но часто не репрезентативных данных [1].

Существует несколько направлений генерации тестов. Одним из наиболее используемых является генерация на основе мутаций. В языковых задачах это может быть переформулировка входа, замена синонимов, перестановка слов, добавление опечаток, шумов или контекста, способного сместить интерпретацию. Визуально – это трансформация изображения по освещённости, цветовой гамме, фону, а также добавление синтетических объектов или

фрагментов. Модели, устойчивые к таким вариациям, считаются более пригодными к использованию в условиях естественного шума и разнообразия реального мира. Для автоматизации мутаций используются библиотеки вроде TextAttack, Checklist, Robustness Gym, а также фреймворки визуальных и аудио-трансформаций, например, Albumentations, DeepAugment и др.

Другой подход – генерация на основе обучаемых генераторов, когда в связке с тестируемой моделью используется генеративная модель, обученная генерировать такие входы, которые приводят к максимальной неопределённости, ошибке или деградации качества. Например, в задаче классификации это может быть генератор, обученный создавать изображения, вызывающие частую смену класса у модели, в языковом моделировании – формулировки, которые заставляют модель выдавать неуверенные, токсичные или противоречивые ответы. Такая постановка задачи часто формализуется как двухуровневая оптимизация: генератор ищет такие входы, которые максимизируют ошибку тестируемой модели при сохранении правдоподобия входа. Это приближает генеративное тестирование к адверсариальному, но делает его более гибким и прикладным.

Также активно развиваются методы семантически контролируемой генерации, где тесты генерируются не случайным образом, а под управлением логических или онтологических ограничений. Например, в задаче интерпретации правовых документов можно генерировать ситуации, в которых изменяется только один параметр (сумма, дата, имя), а смысл должен сохраняться, либо, наоборот, смысл меняется при сохранении формы. Устойчивость модели к таким преобразованиям – маркер её семантической зрелости. Визуальные модели, работающие с медицинскими снимками, могут быть протестированы на изображениях с варьирующимися, но контролируемыми признаками, например, размером опухоли, интенсивностью тени, наличием артефактов. Такие тесты часто требуют совместной работы с экспертами и используются как дополнение к классической валидации.

Ещё одно направление – итеративное генеративное тестирование, когда тестовые входы строятся на основе реакции модели на предыдущие тесты. Например, если модель ошибается на тексте определённой структуры, создаются новые входы, расширяющие этот шаблон. Таким образом, происходит постепенное раскрытие уязвимых зон, а тестовый корпус обогащается критическими случаями. Это приближает процесс к активному тестированию (active testing), в котором система сама обучается находить слабые места.

Ключевым преимуществом генеративных тестов является то, что они не ограничены фиксированным набором эталонов и способны выявлять ошибки в динамике, а также моделировать редкие или опасные ситуации, которые трудно представить в обычных условиях. Особенно важно то, что такие тесты

позволяют формировать обоснованные доверительные границы применимости модели: если генератор не может найти входы, на которых модель нарушает логические или смысловые ограничения, это повышает степень доверия к системе [2].

Однако эффективность генеративного тестирования зависит от способности объяснять, почему тот или иной вход считается «уязвимым». Поэтому генерация тестов должна сопровождаться методами интерпретации, локализации ошибки и семантической категоризации. Современные инструменты – от SHAP и LIME до attention visualizers и gradient-based interpretability – могут быть встроены в процесс тестирования, позволяя одновременно генерировать вход, оценивать реакцию модели и объяснять её поведение [3].

Кроме выявления ошибок, генеративные тесты могут быть использованы и в процессе адаптации и усиления устойчивости модели. Они формируют основу для синтетического дообучения, расширяют обучающий корпус, помогают избегать переобучения на шаблонах. Более того, они могут быть внедрены в контур мониторинга модели в продуктивной среде, когда система периодически проверяет себя с помощью генераторов и сигнализирует о потенциальной деградации.

Таким образом, генеративные тесты – это не вспомогательный инструмент, а полноценный элемент инженерии доверенного ИИ, позволяющий сделать поведение модели более прозрачным, прогнозируемым и устойчивым. В условиях роста требований к верификации, сертификации и социальной приемлемости моделей, генеративное тестирование становится основой как технической надёжности, так и этической обоснованности внедряемых систем. В перспективе такие методы станут обязательным элементом стандартизированных пайплайнов валидации и лицензирования предобученных моделей [4, 5].

Список литературы:

1. Goodfellow, I. Generative Adversarial Nets / I. Goodfellow, J. Pouget-Abadie, M. Mirza [et al.] // Advances in Neural Information Processing Systems. – 2014. – Vol. 27.
2. Kingma, D. P. Auto-Encoding Variational Bayes / D. P. Kingma, M. Welling // International Conference on Learning Representations. – 2014.
3. Odena, A. TensorFuzz : Debugging Neural Networks with Coverage-Guided Fuzzing / A. Odena, C. Olsson, D. Andersen, I. Goodfellow // Proceedings of the 36th International Conference on Machine Learning. – 2019. – Vol. 97. – P. 4901–4911.
4. Tian, Y. DeepTest : Automated Testing of Deep-Neural-Network-driven Autonomous Cars / Y. Tian, K. Pei, S. Jana, B. Ray // Proceedings of the 40th

International Conference on Software Engineering. – New York : ACM, 2018. – P. 303–314. – DOI 10.1145/3180155.3180220.

5. Xie, X. DeepHunter : A Coverage-Guided Fuzz Testing Framework for Deep Neural Networks / X. Xie, L. Ma, F. Juefei-Xu [et al.] // Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis. – New York : ACM, 2019. – P. 146–157. – DOI 10.1145/3293882.3330579.