6.0

WYZWANIA ZWIĄZANE Z DOSZKALANIEM DUŻYCH MODELI JĘZYKOWYCH – PERSPEKTYWA EKONOMETRII

Igor Kapiczyński
Szkoła Główna Handlowa w Warszawie

Dawid Zdanowicz
Szkoła Główna Handlowa w Warszawie

Daniel Kaszyński
Szkoła Główna Handlowa w Warszawie

Petro Vavryk
Taras Shevchenko National University of Kyiv

Streszczenie

Celem rozdziału jest przedstawienie konsekwencji, jakie mogą powstać w wyniku doszkalania dużych modeli językowych, oraz ich przykładowych miar ewaluacji. Fenomen, który zidentyfikowano w rozdziale na przykładzie rzeczywistych danych, znany jest w literaturze jako problem katastrofalnego zapominania (*catastrophic forgetting*). Opisane zjawisko prowadzi do spadku jakości modelu w obszarach niezależnych od obszaru dotrenowanego. Celem tekstu jest wskazanie na rozwój modeli ekonometrycznych, skutkujący obecnie powstaniem modeli klasy LLM oraz zwrócenie uwagi na konsekwencje doszkalania tych modeli, które wpływają na ich użyteczność.

Słowa kluczowe: sztuczna inteligencja, duże modele językowe, doszkalanie dużych modeli językowych, katastrofalne zapominanie

Rozdział z monografii: Doligalski T., Kaszyński D. (red. nauk.), Sztuczna inteligencja w przedsiębiorstwach i gospodarce (AI Spring 2024), Warszawa 2025.

Monografia oferowana jest w swobodnym dostępie do pobrania na stronie AI Lab SGH.

Wprowadzenie

Pierwsze modele ekonometryczne – współcześnie często zaliczane do modeli uczenia maszynowego – są znane od połowy XX w. dzięki pracom Ragnara Frischa, laureata Nagrody Nobla z 1969 r. W swoim artykule pt. *O problemie czystej ekonomii* wskazuje on na ekonometrię jako nową dyscyplinę naukową [Frisch, 1926, s. 1]: „Na pograniczu matematyki, statystyki i ekonomii znajduje się nowa dyscyplina, którą z braku lepszej nazwy można nazwać ekonometrią. Celem ekonometrii jest poddanie abstrakcyjnych praw teoretycznej ekonomii politycznej lub «czystej» ekonomii eksperymentalnej i numerycznej weryfikacji, a tym samym przekształcenie czystej ekonomii, w miarę możliwości, w naukę w ścisłym tego słowa znaczeniu".

Pierwotnie narzędziami ekonometrii były modele parametryczne, takie jak: regresja liniowa [Galton, 1877], model autoregresyjny [Yule, 1926], model autoregresyjny z ruchomą średnią [Kolmogorov, 1941; Wiener, 1949], regresja logistyczna [Cox, 1958], model autoregresyjny z wektorem błędu – VAR [Sims, 1980], model korekty błędu – ECM [Granger, 1981], model autoregresji z heteroskedastycznością warunkową – ARCH [Engle, 1982], uogólniony model autoregresji z heteroskedastycznością warunkową – GARCH [Bollerslev, 1986] lub model wektorowej autoregresji z wektorem błędu – VECM [Granger, Newbold, 1974].

Wskazane przykłady modeli ekonometrycznych istotnie różnią się od algorytmów i podejść wykorzystywanych w informatyce. Klasyczne[1] algorytmy IT działają według z góry określonych zasad, które są wprowadzane do algorytmu przez programistę na etapie tworzenia rozwiązania; każde działanie, operacja i decyzja uzyskane za pomocą rozwiązań IT są określone na podstawie instrukcji uwzględnionych w kodzie takiego rozwiązania [Kamiński i in., 2024]. W przypadku rozwiązań uczenia maszynowego działanie, operacje i decyzje, uzyskiwane za pomocą modelu, są wypadkową wzorców stosowanych w procesie uczenia z wykorzystaniem danych; model posiada cechy adaptacyjne zależne od danych zasilających proces uczenia takiego modelu. Oznacza to, że narzędzia ekonometryczne uczenia maszynowego były strukturalnie innymi podejściami od klasycznych programów lub algorytmów IT: w przypadku modeli ekonometrycznych pojedynczy model był uczony na podstawie wzorców danych, a nie bezpośrednio definiowany przez programistę (klasyczne IT).

[1] Wskazujemy na algorytmy, które są tworzone przez programistów i nie posiadają cech adaptacyjnych. Obecnie coraz więcej rozwiązań informatycznych posiada wbudowane mechanizmy modelu ekonometrycznych umożliwiających adaptacyjny charakter – te rozwiązania nie są rozumiane jako „klasyczne" IT.

Wskazane wcześniej przykłady modeli ekonometrycznych należały do klasy modeli parametrycznych, tj. modeli, w których zdefiniowana była postać relacji między zmiennymi objaśniającymi a zmienną objaśnianą [Hastie, Tibshirani, Friedman, 2009]. Na przykład model regresji liniowej odnosi się do sytuacji, w której twórca modelu antycypuje, że relacja między zmiennymi wprowadzanymi do modelu jest liniowa i zadana formułą:

Tabela 6.1. Proces generujący dane a model regresji liniowej

Założenie procesu generującego dane	Przyjęta postać modelu regresji liniowej
$y = X\beta + \varepsilon$ gdzie: y to wektor wartości zmiennej objaśnianej, X to macierz wartości zmiennych objaśniających, β to wektor parametrów prawdziwych procesu generującego dane, a ε to składnik losowy	$\hat{y} = X\hat{\beta}$ gdzie: $\hat{y}$ to wektor oszacowań wartości zmiennej objaśnianej, X to macierz wartości zmiennych objaśniających, a $\hat{\beta}$ to wektor oszacowanych parametrów modelu

Źródło: opracowanie własne.

W przypadku modeli parametrycznych, zakłada się „wprost" relację między zmiennymi objaśniającymi a zmienną objaśnianą; założenie to jest uchylone w przypadku modeli nieparametrycznych [Wasserman, 2006]. Modele tej klasy nie wymagają określenia struktury relacji między zmiennymi; w przeciwieństwie do modeli parametrycznych, w których struktura jest już zdefiniowana, modele nieparametryczne pozwalają na elastyczność w dopasowywaniu danych. Przykładami modeli nieparametrycznych są: metoda jądrowa [Parzen, 1962], drzewa decyzyjne [Quinlan, 1986], metoda najbliższych sąsiadów [Fix, 1985], regresja splajnów [Boor, 1978], metody regresji lokalnej [Loader, 2006] oraz modele funkcji łączonych [Hastie, Tibshirani, 1986]. Jeden z przykładów modeli nieparametrycznych stanowił model sieci neuronowych, wprowadzony w 1958 r. przez Franka Rosenblatta; był to model typu perceptron jako jedna z pierwszych sztucznych sieci neuronowych. Kolejne prace, m.in. Rumelharta, McClellanda i Parkera [1986], wprowadziły algorytm wstecznej propagacji błędu, umożliwiający efektywne trenowanie wielowarstwowych sztucznych sieci neuronowych.

Współczesnym nurtem związanym z modelami ekonometrycznymi jest rozwój modeli opartych na sieciach neuronowych – ich przykładem są generatywne sieci neuronowe, umożliwiające tworzenie nowych danych na podstawie danych historycznych, na których taka sieć była trenowana [Goodfellow i in., 2014]. Innowacją w zastosowaniu tych podejść jest wyjście poza czysto liczbowy przedmiot generowania prognoz, tak jak w przypadku wskazanych wcześniej podejść historycznych.

Modele generatywnej sztucznej inteligencji umożliwiają tworzenie nowych treści, takich jak teksty, obrazy oraz inne formaty danych [Shanmugamani, 2018]. Obecnie występuje zróżnicowanie zastosowań, które dzisiaj można wpisać pod nazwę generatywnych narzędzi sztucznej inteligencji: są to zarówno generatory tekstu, takie jak ChatGPT, BERT, jak też mechanizmy do tworzenia kodu źródłowego np. GitHub Copilot lub generowania obrazu jak DALL-E. Szczególnym przypadkiem modeli generatywnych, których początek można wskazywać na lata 80., są duże modele językowe (*large language models*, LLM).

6.1. Duże modele językowe

LLM są modelami przetwarzania danych tekstowych, tzw. przetwarzania języka naturalnego, bazujące w dużym stopniu na strukturze transformera[2] [Vaswani i in., 2017]. Modele LLM są trenowane na szerokich zbiorach danych tekstowych, np. anglojęzyczna Wikipedia zawiera 7 mln artykułów [Rothman, 2021]. Zastosowanie tych modeli obejmuje obszary związane z przetwarzaniem tekstu, m.in. tworzenie nowych treści, rozpoznawanie mowy, generowanie streszczeń czy też przygotowanie określonych szablonów tekstowych, np. formatowanie danego tekstu na podstawie podanego wzoru [Vajjala, Majumder, Gupta, Surana, 2020]. Modele LLM dzieli się na małe, średnie i duże. Podział ten definiowany jest liczbą parametrów modelu: małe posiadają niespełna kilka miliardów parametrów, średnie nawet do kilkudziesięciu miliardów i duże opisywane są w setkach miliardów parametrów. Klasyfikację modeli LLM ze względu na ich wielkość przedstawiono w tabeli 6.2.

Tabela 6.2. Klasyfikacja modeli LLM ze względu na ich wielkość

Rozmiar	Liczba parametrów	Przykład	Wymagania sprzętowe
Małe	do kilku mld	DistilBERT, Phi 2, GPT-Neo	komputer osobisty, GPU
Średnie	10–50 mld	Falcon, BloombergGPT	małe serwery
Duże	od 50 do setek mld	Megatron-Turing NLG, GPT-4, Minerva	duże serwery

Źródło: opracowanie własne.

[2] Jest to typ sieci neuronowej, która wykrywa kontekst sekwencyjnych danych i na jego podstawie generuje odpowiedź. Transformer składa się z enkodera (*encoder*), który ma za zadanie zrozumieć dane wejściowe, oraz dekodera (*decoder*), którego zadaniem jest wygenerowanie danych wyjściowych na podstawie danych uzyskanych w enkoderze.

Podczas trenowania modeli LLM mogą wystąpić zjawiska niedostatecznego albo nadmiernego dopasowania danych. W przypadku dużych modeli językowych zjawiska te spowodowane są rozmiarem (tj. liczbą parametrów modelu) oraz długością trenowania modelu (tj. liczbą iteracji na zbiorze treningowym). Oceniając jakość modelu LLM na innym zbiorze danych niż zbiór danych treningowych, można się spodziewać, że zależność między poziomem błędu a iteracjami będzie miała kształt litery U; na początku trenowania błąd modelu LLM spada, przy czym od pewnego momentu następuje przeuczenie modelu i błąd zaczyna ponownie rosnąć. Kiedy znajdzie się w punkcie będącym minimum błędu, zwiększenie liczby parametrów bądź iteracji będzie równoznaczne z wystąpieniem zjawiska nadmiernego dopasowania do danych testowych oraz zmniejszeniem dopasowania do danych populacji ogólnej. W rzeczywistości natomiast, zwiększając te wartości, można przekroczyć pewne maksimum, do którego wartość miary błędu predykcji rośnie. Następnie ta wartość zaczyna spadać poniżej punktu wcześniej określanego jako optymalny [Nakkiran i in., 2021], będącego minimum wcześniejszego wykresu. W konsekwencji nowy wykres byłby w kształcie odwróconej litery N. Zjawisko to nosi nazwę podwójnego spadku (*double descent*).

Moc obliczeniowa potrzebna do trenowania dużych modeli językowych w dużym stopniu wynika z liczby parametrów, jakie posiadają te modele [Kaplan i in., 2020]. Modele LLM, które kategoryzuje się jako małe, można trenować nawet na komputerach osobistych. W przypadku średnich modeli LLM, które zawierają już kilkadziesiąt miliardów parametrów, trenowanie powinno odbywać się na mniejszych serwerach. Trenowanie dużych modeli LLM wymaga natomiast ogromnych nakładów finansowych, mocy obliczeniowej, którą można uzyskać tylko w dużych centrach danych. Na przykład wytrenowanie modelu GPT-3 kosztowało około 12 mln USD [Jordan, 2020].

Duże modele LLM wykorzystuje się w różnych obszarach, kontekstach, problemach, takich jak: tłumaczenie tekstu, generowanie kodu źródłowego, odpowiadanie na pytania, podsumowanie tekstu, analiza semantyczna [Zharovskikh, 2023]. Modele klasy LLM wykorzystywane są m.in. w finansach do wykrywania oszustw lub analizy dokumentów finansowych, w prawie do sporządzania dokumentów oraz znajdowania luk prawnych, a w medycynie do wykrywania chorób i tworzenia planów leczenia.

W zależności od zadania, do którego modele będą wykorzystywane, oceniane są one przy pomocy różnych metryk ewaluacji, tj. miar oceny działania danego modelu. W kolejnej sekcji tekstu przedstawiono standardowe miary oceny działania modeli LLM.

6.2. Pomiar jakości działania LLM

Nie istnieje jeden sposób pomiaru jakości działania modeli LLM [Chang i in., 2024]. Miary te definiowane są na podstawie obszaru działania modelu; inne miary oceny jakości mogą być wykorzystywane np. dla streszczeń tekstu, a inne dla generowania kodu źródłowego bądź tłumaczeń. Poniżej przedstawiono zestawienie najpopularniejszych miar oceny jakości działania modeli LLM.

6.3. Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

Jednym z przykładów miar jakości działania modeli LLM jest miara Recall-Oriented Understudy for Gisting Evaluation, zwana również krócej ROUGE [Lin, Chin-Yew, 2004]. Jest to metryka służąca do oceny jakości streszczeń lub tłumaczeń wytworzonych przez model LLM względem tekstu utworzonego przez człowieka [Lin, Chin-Yew, 2004]. Wartości, jakie miara ROUGE może przyjmować, obejmują zakres: 0–1, gdzie 0 wskazuje na całkowity brak podobieństwa tekstu, a 1 – na identyczność względem tekstu referencyjnego.

ROUGE jest w istocie zbiorem miar, różniących się od siebie rozmiarem n-gramów[3], które będą do siebie porównywane. Na przykład w przypadku ROUGE-2 porównuje się bigramy (pary wyrazów). Porównując zdania: „Jan jest studentem ekonomii" i „Kuba jest studentem fizyki", najpierw należy podzielić zdania na bigramy. W pierwszym przypadku są to: „Jan jest", „jest studentem", „studentem ekonomii". Natomiast w drugim są to: „Kuba jest", „jest studentem", „studentem fizyki". Następnie liczy się powtarzające się w obu przypadkach pary i dzieli przez liczbę bigramów występujących w tekście referencyjnym, w tym przypadku wartość ta wynosi 3. ROUGE-2 przyjmuje więc wartość 1/3 = 0,33. Inne miary ROUGE to ROUGE-1 i ROUGE-L, gdzie dzielimy najdłuższy wspólny szereg słów przez całkowitą liczbę słów w tekście referencyjnym.

6.4. Perpleksja – *perplexity*

Innym przykładem miary oceny jakości działania modeli LLM jest miara perpleksji (*perplexity*). Jest ona oparta na funkcji wiarygodności, umożliwiającej określenie prawdopodobieństwa pojawiania się danego wyrazu w kontekście danego zdania. Im

[3] Jako *n*-gram rozumie się *n*-elementowy szereg słów ustawionych w danej kolejności, występujących w określonej formie odmiany.

większa wartość, tym mniejsze prawdopodobieństwo pojawienia się danego tekstu (tzn. tekst jest mniej prawdopodobny).

$$\log(PPL) = -\frac{1}{N}\sum_{i=1}^{N}\log(P(t_i \mid c_{t_i})),$$

gdzie: *PPL* to miara perpleksji, *N* oznacza rozmiar zbioru testowego, *P* reprezentuje prawdopodobieństwo, t_i oznacza token „i", dla którego obliczana jest wartość miary pod warunkiem poprzedzających go tokenów c_{t_i}.

Poprzez token rozumie się formalną (matematyczną) reprezentację danego wyrazu[4].

Poniżej przedstawiono przykład działania modelu LLM przy różnych stopniach jego wytrenowania. Demonstruje on, jak model językowy GPT-2 odpowiada na identyczne zapytanie przy trzech różnych poziomach wytrenowania. Doszkalanie modelu zostanie dokładniej omówione w kolejnej sekcji tekstu.

Pierwszym z trzech jest model surowy, którego parametry przyjmują losowe wartości (model bez wytrenowania). Tekst generowany przez taki model składa się z losowych tokenów, tj. słów lub znaków. Jego metryka, średnia perpleksja, jest stosunkowo wysoka, co oznacza, że niepewność odnośnie do generowanego tekstu jest wysoka.

Następnie to samo zapytanie zostało przesłane do wytrenowanego, ogólnego modelu. Wygenerowany tekst składa się już z poprawnych zdań, jednak pozbawiony jest spójności tematycznej oraz głębszego zrozumienia kontekstu. Miara perpleksji, czyli miara niepewności, jest znacząco niższa, co przekłada się na tekst bardziej zgodny z oczekiwaniami opartymi na danych wejściowych.

Tabela 6.3. Przykładowe treści wygenerowane przez model GPT-2 dla modelu surowego, ogólnego oraz wyspecjalizowanego

Model	Wyjście	Średnia perpleksja
Surowy model	**Economics is** afterlifeç gayFine Alas Alas Alasjadjadjadjadjadjad Madurojadjadjad	17 406
Wytrenowany ogólny model	**Economics is** a very important field for the study of economics	3522
Wyspecjalizowany model	**Economics is** a rich and diverse region. Inclusion of diverse economic and social issues in a diverse region is essential to succeed	854

Źródło: opracowanie własne.

[4] Dla przykładu słowo *reading* w zależności od przyjętej reprezentacji może oznaczać jeden lub kilka tokenów ([„read", „ing"] lub [„reading"]). Słowniki tokenów różnią się pomiędzy modelami językowymi, stąd różnice w reprezentacji słów lub ich części.

Ostatnie zapytanie zostało skierowane do modelu wyspecjalizowanego i doszkolonego (*fine-tuned*). W tym przypadku został on wytrenowany na ekonomicznej bazie danych. Tekst wygenerowany przez taki model jest bardziej złożony i spójny, a niepewność w zakresie pojawiających się słów w zdaniu – mniejsza względem całego kontekstu.

6.5. *Cross entropy loss*

Jednym ze sposobów oceny modeli oraz monitorowania przebiegu ich treningu jest wykorzystanie funkcji straty (*loss function*). Funkcje te służą do obliczania wartości, zwanej stratą walidacyjną (*validation loss*), która mierzy utratę skuteczności w generowaniu odpowiedzi względem tekstów walidacyjnych. Mogą się one różnić w zależności od modelu czy zadania, do którego model jest trenowany. Jedna z tych funkcji to strata entropii krzyżowej (*cross entropy loss*). Funkcja wykorzystywana do obliczenia wartości straty ma postać:

$$L = \frac{1}{N}\sum_{i=1}^{N} -\log \hat{y}^{(i)}_{x_{i+1}},$$

gdzie: N to rozmiar zbioru testowego, a L to wartość funkcji straty.

Funkcja ta dla każdego i liczy entropię krzyżową pomiędzy przewidzianym rozkładem prawdopodobieństwa $\hat{y}^{(i)}$ a kolejnym słowem zaobserwowanym w tekście referencyjnym. Następnie liczona jest średnia entropia dla całego zbioru walidacyjnego, ale funkcja ta może także zostać użyta do obliczenia straty w zbiorze testowym czy treningowym.

Obliczenie wartości funkcji straty wygląda w następujący sposób: na początku niezbędny jest tekst, do którego model będzie porównywany. Przyjmijmy, że jest nim zdanie: „Jan gra w piłkę nożną". Następnie generowany jest tekst. Podczas generowania powstają rozkłady prawdopodobieństwa dla każdego kolejnego słowa, z pominięciem pierwszego, które zostało uwzględnione w komendzie. Zakładając, że wygenerowany tekst brzmi: „Jan grał w piłkę ręczną", wartość funkcji straty jest liczona, jeśli wygenerowane słowo nie jest tym samym co słowo w tekście referencyjnym i przyjmuje ona wtedy wartość ujemnego logarytmu prawdopodobieństwa wystąpienia tego słowa. Jeśli słowo to odpowiada słowu na tej samej pozycji w tekście referencyjnym, strata jest równa 0. W wygenerowanym zdaniu słowa, dla których wartość funkcji straty będzie różna od 0, to „Jan" i „piłkę", ponieważ następne słowa nie odpowiadają tym referencyjnym. Na koniec liczona jest średnia ze wszystkich wartości i otrzymany wynik stanowi wartość funkcji straty dla całego tekstu.

6.6. Doszkalanie LLM

Doszkalanie modelu (*fine-tuning*) to proces, który polega na dostosowaniu już wcześniej wytrenowanego modelu do specyficznego kontekstu dziedzinowego [Devlin i in., 2019], w którym to kontekście model będzie wykorzystywany (np. wcześniej wskazywane przykłady z prawa lub medycyny). W przypadku LLM doszkalanie modelu polega na zmianie części jego parametrów w celu dostosowania wydajności do konkretnego kontekstu bez konieczności trenowania całego modelu od początku [Khandare, 2023]. Jest to szczególnie użyteczne, gdy zadanie, w ramach którego model ma być wykorzystywany, różni się od oryginalnego zadania, do którego został przeznaczony. Istnieje natomiast kilka rodzajów technik doszkalania modeli, wśród których można wyróżnić:

- **całkowite doszkalanie modelu** – polegające na trenowaniu całego modelu na podstawie nowego zbioru danych; taki typ doszkalania wymaga znacznych nakładów mocy obliczeniowej oraz wystarczająco dużego zbioru danych, na podstawie którego trenowane będą wszystkie warstwy modelu;
- **selektywne doszkalanie modelu** – technika polegająca na dostrojeniu tylko wybranych warstw modelu, zwykle górnych lub dolnych (inaczej: początkowych bądź końcowych); wymaga znacznie mniejszej mocy obliczeniowej w porównaniu z całkowitym doszkalaniem modelu i jest używana w przypadku, gdy zadanie docelowe jest zbieżne z pierwotnym zadaniem ogólnego modelu;
- **efektywne doszkalanie modelu** (*parameter-efficient fine-tuning*, PEFT) – charakteryzujące się dodaniem małej warstwy nowych parametrów w celu dostosowania modelu pod konkretne zadanie; pozostałe parametry pozostają niezmienne, co zmniejsza potrzebną moc obliczeniową;
- **hybrydowe doszkalanie modelu** – polegające na połączeniu różnych podejść doszkalania w celu zachowania części oryginalnych cech modelu z jednoczesnym dodaniem nowych; proces doszkalania dzieli się tutaj na kilka etapów, w których raz trenowana jest większa część parametrów, a raz mniejsza; proces ten pozwala na dostrojenie modelu do bardzo specyficznych zadań, nie pozbawiając go jednocześnie jego podstawowych możliwości.

6.7. Przykład doszkalania LLM

W ramach badania przeprowadzony został eksperyment numeryczny, polegający na wykorzystaniu modelu BART opracowanego przez firmę Facebook [Lewis i in., 2019].

Celem eksperymentu było sprawdzenie, w jaki sposób doszkalanie modelu LLM na wyspecjalizowanym zbiorze danych (artykuły medyczne) wpływa na jakość generowanych treści na innym ogólnym zbiorze danych (tu: zawierającym artykuły z serwisu internetowego telewizji CNN).

Dane medyczne zawierały 20 tys. obserwacji w zbiorze treningowym, użytych do dotrenowania modelu i dodatkowych 1000 w zbiorze walidacyjnym. Składały się one z abstraktów biomedycznych i odpowiadających im streszczeń.

Tabela 6.4. Przykład obserwacji ze zbioru danych medycznych

Artykuł	Streszczenie
„Cardiovascular disease is the leading cause of death globally. While pharmacological advancements have improved the morbidity and mortality associated with cardiovascular disease, non-adherence to prescribed treatment remains a significant barrier to improved..."	"Medication adherence in cardiovascular medicine"

Źródło: opracowanie własne.

Dane nazwane jako CNN zawierały 1000 obserwacji użytych do walidacji modelu. Zbiór składał się z artykułów ukazanych w CNN i odpowiadających im streszczeń.

Tabela 6.5. Przykład obserwacji ze zbioru artykułów CNN

Artykuł	Streszczenie
„The Palestinian Authority officially became the 123 rd member of the International Criminal Court on Wednesday, a step that gives the court jurisdiction over alleged crimes in Palestinian territories. The formal accession was marked with a ceremony at The Hague, in the Netherlands..."	"Membership gives the ICC jurisdiction over alleged crimes committed in Palestinian territories since last June..."

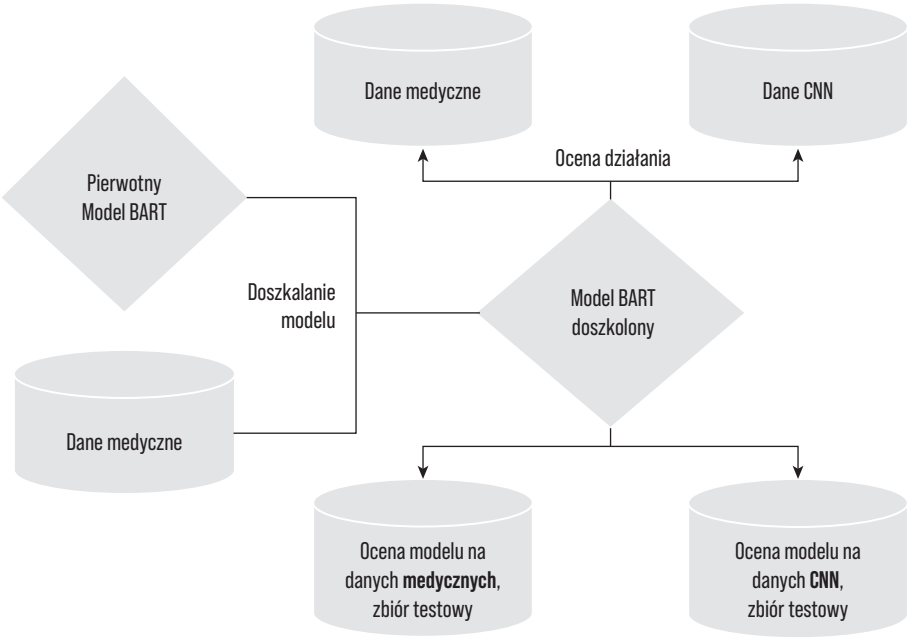
Źródło: opracowanie własne.

Schemat przeprowadzonego eksperymentu numerycznego przedstawiono na rysunku 6.1.

W pracy zastosowano efektywne doszkalanie modelu – PEFT, przeprowadzone w sześciu iteracjach (*epoch*). Liczba parametrów modelu użytego w eksperymencie wynosi 139 641 600. Następnie zostało dodane 221 184 parametry, co stanowi 0,16% wszystkich parametrów modelu początkowego.

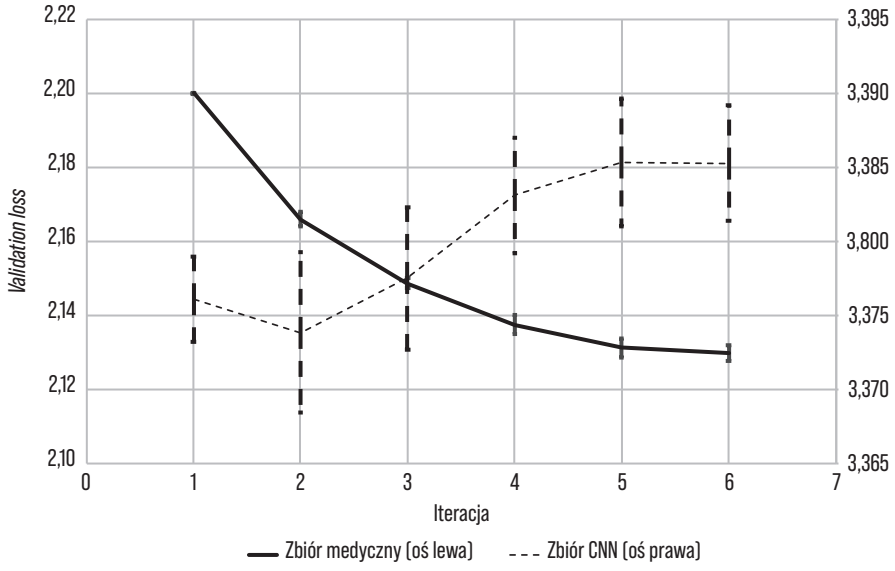
Średnie wartości błędów na zbiorze walidacyjnym, odpowiednio dla danych medycznych oraz danych CNN, uzyskane w oparciu na modelu doszkalanym na zbiorze medycznym, przedstawiono na rysunku 6.2 (wartości liczbowe w załączniku A).

Rysunek 6.1. Schemat eksperymentu numerycznego



Źródło: opracowanie własne.

Rysunek 6.2. Schemat eksperymentu numerycznego



Źródło: opracowanie własne.

Wraz z każdą iteracją doszkalania uzyskiwano statystycznie lepszą jakość modelu na danych medycznych (linia ciągła na rysunku 6.2). Oznacza to, że model generował lepszej jakości tekst w kontekście medycznym, co jest zjawiskiem oczekiwanym: doszkalając model w oparciu na specyficznych danych medycznych, oczekujemy, że będzie on lepiej działał w tym konkretnym kontekście dziedzinowym.

Interesujący jest jednak profil jakości działania modelu dla danych zbioru CNN (linia przerywana na rysunku 6.2). Obszar ten jest w dużej mierze niezależny od danych medycznych, obejmując kwestie społeczne, socjologiczne, prawne i ekonomiczne, które co do zasady są niezależnym obszarem względem medycyny[5]. Z każdą iteracją doszkalania modelu BART, w oparciu na danych medycznych, ten sam model wykorzystywany do analizy danych ze zbioru CNN uzyskiwał coraz wyższy błąd *validation loss*. Obserwowany jest statystyczny spadek jakości działania modelu w oparciu na danych CNN.

Zidentyfikowana w pracy anomalia to szerzej znane w literaturze katastrofalne zapominanie (*catastrophic forgetting*), związane z sytuacją, w której model poprzez doszkalanie w jednym obszarze wykorzystania (tu: dane medyczne) traci dokładność działania w innym niezależnym obszarze (tu: dane CNN) [Kirkpatrick i in., 2017]. Zjawisko to jest obecne m.in. w przypadku architektur modeli opartych na sieciach neuronowych. Za główną przyczynę wspomnianej anomalii wskazuje się mechanizm dostosowywania wag modelu. Podczas doszkalania modelu w danym obszarze potencjalnie wszystkie wagi modelu mogą ulegać nieznacznym modyfikacjom, co niekorzystnie wpływa na jakość działania modelu na innych obszarach wykorzystania.

Podsumowanie

Rozwój ekonometrii – jak zaznaczono we wstępie, dziedziny leżącej pomiędzy matematyką, statystyką oraz informatyką – wiązał się na początku z paradygmatem modelowania parametrycznego (tj. modelami o zadanej postaci funkcyjnej), a później z przejściem na modele nieparametryczne. Przykładem klasy modeli nieparametrycznych są sztuczne sieci neuronowe. Początkowe wykorzystanie tych modeli, skutkujące poprawą względem wcześniejszych podejść w obszarze jakości generowanych pro-

[5] Autorzy dostrzegają związki między kwestiami społecznymi, socjologicznymi, prawnymi lub wskazaniami ekonomicznymi a medycyną; niemniej dane zbioru CNN odnoszą się do kwestii medycznych wyłącznie powierzchownie, nie traktując tego jako głównego obszaru zainteresowań. Dla uproszczenia wywodu przyjęto, że zagadnienia objęte w danych CNN są niezależne od zagadnień objętych danymi medycznymi.

gnoz z modeli, przyniosło dalszy rozwój i zaawansowanie wykorzystywanych podejść. Obecnie rozwój tego obszaru widać na przykładzie popularnych, dostępnych dla użytkowników nietechnicznych, dużych modeli językowych.

Ponadto obserwujemy coraz większą popularyzację wykorzystania modeli LLM w wymiarze zarówno horyzontalnym: różne konteksty dziedzinowe, jak również wertykalnym: szczegółowe i specjalistyczne zadania do wykonywania. Jednocześnie budowa modelu LLM od podstaw jest z reguły ekonomicznie nieuzasadniona z punktu widzenia pojedynczego przedsiębiorstwa.

Taki stan rzeczy powoduje, że przedsiębiorstwa coraz częściej wykorzystują modele językowe – wytrenowane i udostępnione przez duże firmy technologiczne. W celu dostosowania sposobu działania tych modeli z charakteru ogólnego na wyspecjalizowany, dziedzinowy wypracowano szereg metod ich doszkalania.

Proces doszkalania, w zależności od rodzaju, może dotyczyć wszystkich lub wybranych parametrów sieci neuronowej. Co więcej, występują podejścia (takie zastosowano w pracy), w których – zamiast modyfikacji istniejących parametrów – do sieci neuronowej wprowadza się dodatkowe warstwy i to na nich wykonywana jest modyfikacja.

Doszkalanie modeli powoduje poprawę dokładności działania modelu w dziedzinie, w której był doszkalany. W pracy wskazano jednak scenariusz, w którym obok poprawy dokładności działania modelu w konkretnej dziedzinie, w jakiej wykonywane było doszkalanie, zaobserwowano również pogorszenie oceny działania modeli w innej dziedzinie. Omawiany proces w literaturze nazywa się katastrofalnym zapominaniem i związany jest z naruszeniem wartości wag modelu, odpowiadających za obszar, w którym model nie był doszkalany (w przykładzie były to dane CNN).

Obserwacja poczyniona w pracy dotyczy faktu, że przejście z modeli ogólnych na modele domenowo wyspecjalizowane, poprzez procedurę doszkalania modelu, może powodować nieplanowane pogorszenie działania tego modelu w obszarze niezależnym. Oznaczać to może, że wraz z postępującym doszkalaniem modelu w konkretnym obszarze zastosowań, model ten powinien być również wyłączany z pozostałych niezależnych zastosowań.

Załącznik

Tabela Z6.1. Strata walidacyjna modelu dotrenowanego na danych medycznych i ocenionego na danych medycznych

Iteracja	Średnia	Odchylenie standardowe	Przedział ufności
1	2,20021	0,00005	0,00003
2	2,16601	0,00299	0,00185
3	2,14860	0,00220	0,00137
4	2,13749	0,00410	0,00254
5	2,13114	0,00419	0,00260
6	2,12980	0,00336	0,00208

Źródło: opracowanie własne.

Tabela Z6.2. Strata walidacyjna modelu dotrenowanego na danych medycznych i ocenionego na artykułach z CNN

Iteracja	Średnia	Odchylenie standardowe	Przedział ufności
1	3,37608	0,00459	0,00285
2	3,37384	0,00874	0,00542
3	3,37749	0,00773	0,00479
4	3,38311	0,00634	0,00393
5	3,38535	0,00701	0,00435
6	3,38530	0,00631	0,00391

Źródło: opracowanie własne.

Bibliografia

Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity, *Journal of Econometrics*, 31(3), s. 307–327.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., Xie, X. (2024). A Survey on Evaluation of Large Language Models, *ACM Transactions on Intelligent Systems and Technology*, 15(3), s. 1–45.

Cox, D.R. (1958). The Regression Analysis of Binary Sequences, *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 20(2), s. 215–232.

De Boor, C. (1978). *A Practical Guide to Splines*. New York: Springer.

Devlin, J., Chang, M., Lee, K., Toutanova, K. (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. W: *North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (s. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics.

Engle, R.F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation, *Econometrica: Journal of the Econometric Society*, 50(4), s. 987–1007.

Fix, E. (1985). *Discriminatory Analysis: Nonparametric Discrimination, Consistency Properties* (vol. 1). Texas: USAF School of Aviation Medicine.

Frisch, R. (1926). *Sur un problème d'économie pure*. Grøndahl & søns boktrykkeri.

Galton, F. (1877). *Typical Laws of Heredity*. William Clowes and Sons.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative Adversarial Nets. W: *Advances in Neural Information Processing Systems* (vol. 3, s. 2672–2680). DOI: 10.1007/978-3-658-40442-0_9.

Granger, C.W. (1981). Some Properties of Time Series Data and Their Use in Econometric Model Specification, *Journal of Econometrics*, 16(1), s. 121–130.

Granger, C.W., Newbold, P. (1974). Spurious Regressions in Econometrics, *Journal of Econometrics*, 2(2), s. 111–120.

Hastie, T., Tibshirani, R. (1985). Generalized Additive Models; Some Applications. W: *Generalized Linear Models: Lecture Notes in Statistics* (vol. 32, s. 66–81), R. Gilchrist, B. Francis, J. Whittaker (Eds). New York: Springer.

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). New York: Springer.

Jordan, J. (2020). *The Cost of Training GPT-3*, https://towardsdatascience.com/the-cost-of-training-gpt-3-7-m-8b9b1c472fbf (dostęp: 30.04.2024).

Kamiński, B., Król-Całkowska, J., Raulinajtys-Grzybek, M., Więckowska, B., Trzebiński, W., Byszek, K., Jaśkowiak, D., Kaszyński, D. (2024). *Sztuczna inteligencja w zdrowiu. Bezpieczeństwo prawne i wykorzystanie w Polsce*. SGH Think Tank dla ochrony zdrowia. DOI: 10.33119/SZIWZBPIWWP-2024-1-81.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D. (2020). *Scaling Laws for Neural Language Models*. DOI: 10.48550/arXiv.2001.08361.

Khandare, S.S. (2023). *Mastering Large Language Models: Advanced Techniques, Applications, Cutting-Edge Methods, and Top LLMs* (English ed.). Independently published.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D. (2017). Overcoming Catastrophic Forgetting in Neural Networks, *Proceedings of the National Academy of Sciences*, 114(13), s. 3521–3526.

Kolmogorov, A. (1941). Interpolation and Extrapolation of Stationary Random Sequences. *Izvestiya the Academy of Sciences of the USSR*, 5, s. 3–14.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. (2019). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. DOI: 10.48550/arXiv.1910.13461.

Lin, C.Y. (2004, July). Rouge: A Package for Automatic Evaluation of Summaries. W: *Text Summarization Branches Out* (s. 74–81). Barcelona: Association for Computational Linguistics.

Loader, C. (2006). *Local Regression and Likelihood*. New York: Springer.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., Sutskever, I. (2021). Deep Double Ddescent: Where Bigger Models and More Data Hurt, *Journal of Statistical Mechanics: Theory and Experiment*, 12, 124003.

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode, *The Annals of Mathematical Statistics*, 33(3), s. 1065–1076.

Quinlan, J.R. (1986). Induction of Decision Trees, *Machine Learning*, 1, s. 81–106.

Rothman, D. (2021). *Transformers for Natural Language Processing: Build Innovative Deep Neural Network Architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and More*. Birmingham: Packt Publishing.

Rouge, L.C. (2004). A Package for Automatic Evaluation of Summaries. W: *Text Summarization Branches Out* (s. 74–81). Barcelona: Association for Computational Linguistics.

Shanmugamani, R. (2018). *Deep Learning for Computer Vision: Expert Techniques to Train Advanced Neural Networks Using TensorFlow and Keras*. Birmingham: Packt Publishing.

Sims, C.A. (1980). Macroeconomics and Reality, *Econometrica: Journal of the Econometric Society*, 48(1), s. 1–48.

Vajjala, S., Majumder, B., Gupta, A., Surana, H. (2020). *Practical Natural Language Processing: a Comprehensive Guide to Building Real-World NLP Systems*. Sebastopol: O'Reilly Media.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. (2017). *Attention Is All You Need*, *Neural Information Processing Systems*, 30, s. 5998–6008.

Wasserman, L. (2006). *All of Nonparametric Statistics*. New York: Springer.

Wiener, N. (1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. The MIT Press. DOI: 10.7551/mitpress/2946.001.0001.

Winston, P.H. (1977). *Artificial Intelligence*. Massachusetts: Addison-Wesley.

Yule, G.U. (1926). Why do We Sometimes Get Nonsense Correlations Between Time-Series?, *Journal of the Royal Statistical Society*, 89(1), s. 1–63.

Zharovskikh, A. (2023). *Best Applications of Large Language Models*, https://indatalabs.com/blog/large-language-model-apps (dostęp: 30.04.2024).