

Wybrane zagadnienia analizy pomiarów konwergencji wyrobisk technikami maszynowego uczenia oraz rekurencyjnymi sieciami neuronowymi

dr inż. Dariusz JUSZYŃSKI
Okręgowy Urząd Górniczy we Wrocławiu

TREŚĆ: W artykule zawarto doświadczenia autora związane z budową modeli rekurencyjnych sieci neuronowych przeznaczonych do modelowania przebiegów konwergencji wyrobisk górniczych. We wstępie przedstawiono kwestię analizy danych typu sekwencyjnego, następnie omówiono analizy pomiarów konwergencji, które stanowią przykład sekwencji. W dalszej części zaprezentowano model sieci neuronowej wraz z procesem trenowania w oparciu o dane z pomiarów konwergencji wyrobisk górniczych. Na podstawie sformułowanego kryterium wykazano możliwość zastosowania sieci rekurencyjnych do modelowania przebiegów konwergencji.

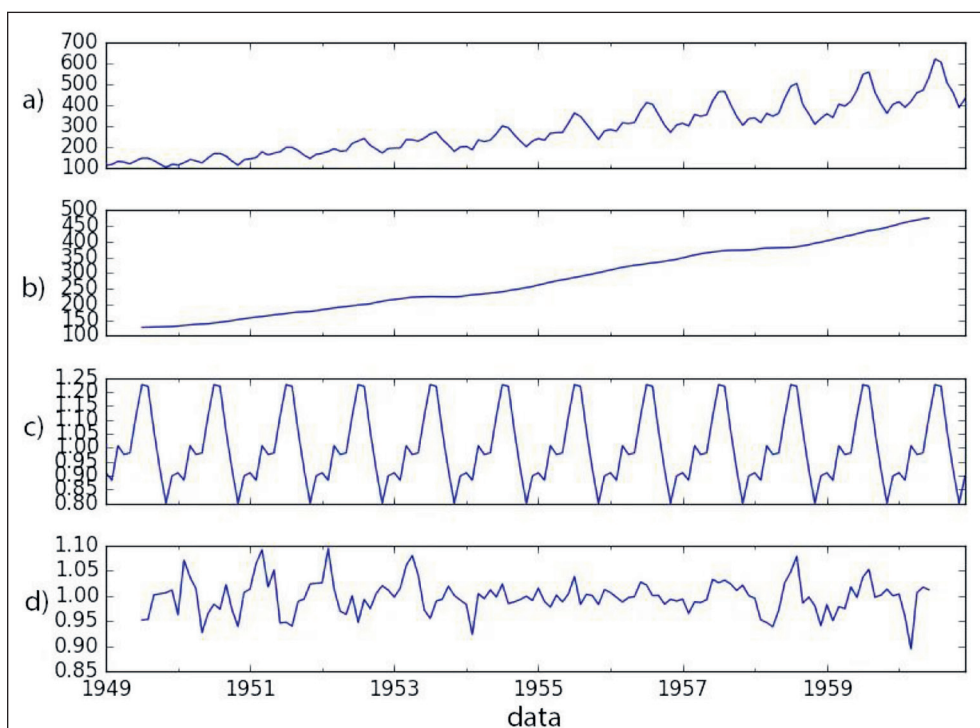
SŁOWA KLUCZOWE: sieci neuronowe, uczenie maszynowe, pomiary konwergencji wyrobisk, górnictwo

1. Wstęp

L iteratura zarówno polsko-, jak i obcojęzyczna, traktująca o analizach konwergencji wyrobisk górniczych za pomocą technik uczenia maszynowego oraz rekurencyjnych sieci neuronowych, jest uboga. Artykuł szerzej omawia te kwestie. Wybrany przykład ma charakter aplikacyjny, zaprezentowane analizy mogą być wykorzystane przez badaczy zajmujących się badaniem konwergencji wyrobisk w systemach rozcinań calizny systemem komór na filary technologiczne, jak również przez autora w kolejnych pracach nad zastosowaniem sieci neuronowych w geomechanice górniczej. Przedstawione spostrzeżenia mogą również rzucić nowe światło na dotychczasowe obserwacje procesów prowadzących do nagłego wyładowania energii górotworu w konsekwencji zaistnienia zjawiska tąpnięcia w wyrobiskach górniczych. Uogólniając, można stwierdzić, że niniejszy artykuł dotyczy analizy przeprowadzonych pomiarów in-situ, a o istotności problemu może świadczyć fragment publikacji prof. Małkowskiego [1], który stwierdził, że „monitoring w górnictwie ma zdecydowanie największe znaczenie w przypadku prognozowania zagrożeń trudno przewidywalnych, jak tąpnięcia, lub wyrzutów gazów i skał oraz zawałowego, które dodatkowo następuje w sposób nagły i narasta w krótkim okresie czasu”.

Uczenie maszynowe jest obecnie na szczycie zainteresowania medialnego, czego dowodem są liczne publikacje na temat ChatGPT, pojawiające się często w kontekście sztucznej inteligencji. W istocie wspomniana sztuczna inteligencja to głównie metody uczenia maszynowego, w tym sztucznych sieci neuronowych [2], które można rozłożyć na pojedyncze zadania, niezależnie rozwijane w ostatnich latach przez społeczność, ang. *machine learning* (uczenia maszynowego, dalej – ML). Najbardziej zaawansowane rozwiązania analizy tekstu, rozpoznawania obrazu lub jego elementów, analizy sentymentów, generowania obrazów, generowania tekstu są u podstaw oparte na zadaniach predykcji, klasyfikacji oraz metodach przygotowywania zbiorów uczących, metodach nauki głębokich sieci neuronowych i oceny jakości uczonego modelu.

Warto nadmienić, iż od lat publikowana jest krzywa *Hype Cycle for Artificial Intelligence* – Gartner [3], z której wynika dojrzałość danej koncepcji, może ona osiągnąć kolejno szczyt zainteresowania, dołek rozczarowania i płaskowyż zrozumienia. Obecnie koncepcje stojące za ChatGPT są na szczycie zainteresowania, co oznacza, że pełne wykorzystanie ich możliwości dopiero nastąpi. Tak też należy podejść do przedstawionej w dalszej części artykułu rekurencyjnej sieci neuronowej, której ograniczenia są dobrze znane (np. krótka pamięć w tył



► Rys. 1. Przykładowy szereg czasowy [4]: a – pomiar, b – trend, c – sezonowość, d – szum

► Fig. 2. Example time series: a – measurement, b – trend, c – season, d – residue

sekwencji, problemy zanikających lub wybuchających gradientów), a realne zastosowania w różnych dziedzinach ciągle poszukiwane (np. analiza szeregów czasowych, tj. wartości akcji, cen mieszkań, analiza tekstu itp.).

Przez tytułową analizę technikami ML autor rozumie wszystko, co jest związane z przygotowaniem surowych danych do właściwej pracy, dziedzina ta z angielskiego nazywana jest *data science*. Lista zaprezentowanych w artykule technik nie jest długa, obejmuje: obliczanie średniej wartości, sortowanie, grupowanie, uzupełnianie brakujących wartości, usuwanie elementów odstających oraz agregację danych. Niemniej przy obszernej bazie danych (zakład górniczy, rok, pole, data, odległość od wstrząsu, wpływ wstrząsu, miejsce pomiaru itp.) są one trudne do wykorzystania za pomocą wyłącznie popularnego arkusza kalkulacyjnego – programistycznie jest to możliwe w kilku liniach kodu.

Ostatnie słowa należy poświęcić środowiskom programistycznym użytym w dalszej części, w skład których wchodzi: Python (język programowania), Numpy i Pandas (przetwarzanie danych), TensorFlow (sieci neuronowe) oraz Matplotlib (wizualizacja danych). Wszystkie są darmowe i stanowią jeden z fundamentów współczesnego ML.

2. Sekwencje, szeregi czasowe, trend, sezonowość, zaszumienie

Przechodząc do pomiarów, które tworzą zbiór danych treningowych – sekwencji, należy wyjaśnić te pojęcia, czyli opisać, w jaki sposób najlepiej przygotowywać pomiary oraz z czego składają się sekwencje.

Serię numerycznych danych uporządkowanych w chronologiczny sposób nazywa się szeregiem czasowym, jest on sekwencją danych w stałym interwale cza-

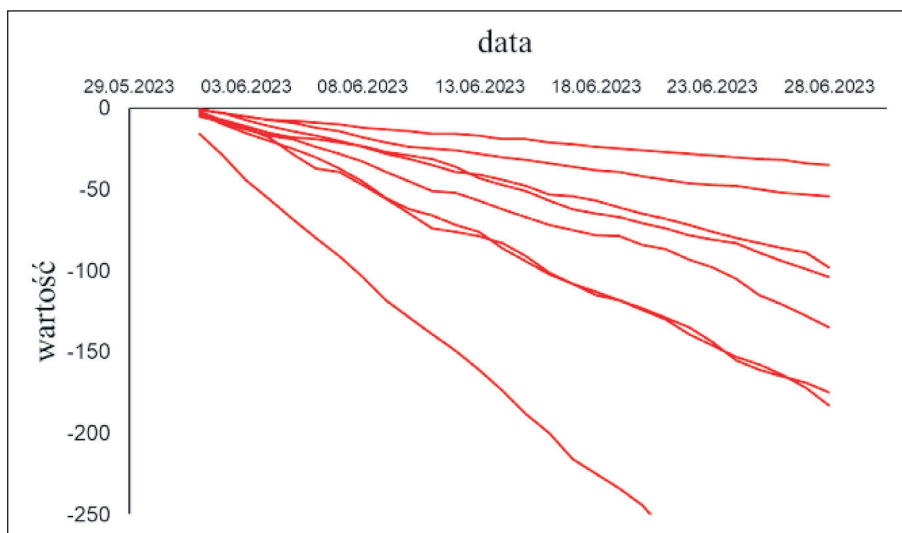
sowym [5]. Prognozowanie szeregu czasowego polega na użyciu modeli do przewidywania przyszłych wartości na podstawie obserwacji [6]. Szeregi czasowe składają się z czterech elementów: poziomu, trendu, sezonowości i szumu. Wszystkie posiadają poziom i szum, natomiast metody prognozowania polegają na rozpoznaniu wzorów sezonowych zmian oraz trendu obserwacji (rys. 1).

Na rysunku 2 przedstawiono przykładowe pomiary konwergencji. Wartość konwergencji wyrobiska górniczego zależy od umiejscowienia punktu pomiarowego, w obszarze roboczym zachodzi do momentu ustabilizowania wyrobiska (np. jego zaciśnięcia). Z uwagi na przebieg trendu oraz niecykliczne zaburzenia przebiegu implikuje nietrywialność zadania w dekompozycji.

3. Zasada braku darmowych obiadów, czyli POC

Istotą trenowania sztucznych sieci neuronowych (dalej SSN) jest podanie zbioru treningowego na wejście sieci oraz zakończenie procesu nauki w momencie, gdy sieć zaczyna uczyć się zbioru na pamięć. Metodą kontroli sieci na okoliczność zdolności do uogólniania (generalizacji) jest ocena modelu za pomocą zbioru, którego sieć nie widziała, nazywanego testowym. Jeżeli dane testowe pochodzą z tego samego rozkładu co dane treningowe, dobrze nauczona sieć powinna uzyskać wynik porównywalny z tym uzyskanym na etapie nauki. Niższy wynik świadczy o zjawisku przetrenowania, tj. nauczaniu się zbioru treningowego na pamięć bez zdolności do generalizacji.

Która sieć jest zatem najlepsza? W tym miejscu warto przywołać publikację Wolpera [7], twierdzi on, że jeżeli nie przyjmie się jakichkolwiek założeń dotyczących danych, to żaden model nie jest lepszy od pozostałych. Zamiennie przytacza się sentencję o braku darmowych



► Rys. 2. Przykładowe sekwencje konwergencji wyrobiska górniczego

► Fig. 2. Examples of mining excavation convergence sequences

obiadów, tj. dla jednych danych najlepiej sprawdzają się modele liniowe, dla innych SSN – nie ma więc uniwersalnie najlepszych modeli.

Zatem do analiz sekwencji, takich jak pomiary konwergencji, odpowiednie są rekurencyjne sieci neuronowe (RNN). Przedstawione w artykule rozwiązanie można uznać za dowód takiej koncepcji (ang. *proof of concept* – POC), w którym autor demonstruje możliwość uzyskania istotnych wyników. Potwierdza to zbieżność krzywych uczenia do swojego plateau, co stanowi przejaw konwergencji [8]. Aby uniknąć niejasności terminologicznych, w tekście zastosowano jasne rozgraniczenie tych pojęć. Na potrzeby publikacji pomija się udowodnienie, że modele proste, tj. regresji liniowej, drzew decyzyjnych itp., są niewystarczające dla przyjętego zbioru treningowego i postawionego zadania.

4. Pomiary konwergencji

– zastosowanie w górnictwie rud miedzi

Wprowadzając czytelnika w kwestię przydatności pomiarów konwergencji, definiowanej jako różnica długości bazy pomiarowej w czasie i wyrażonej w jednostkach długości, należy wspomnieć o przyczynie, co syntetycznie ujął Marczak [9], stwierdzając, że jest ona „mierzalnym wskaźnikiem zaciskania wyrobisk. Przytaczając kolejny fragment, „ocena konwergencji masywu skalnego i wyrobisk w obudowie służy kontroli procesów deformacji masywu skalnego i obudowy, a także pozwala na porównanie procesów zaciskania różnych grup wyrobisk” [9], poznajemy możliwości inżynierskiego zastosowania, jak również wszelkiego rodzaju naukowych analiz. Pomiary konwergencji należą do podstawowych metod obserwacji górotworu pod kątem oceny zagrożenia tąpnięciami [10].

5. Uczenie maszynowe, neuronowe sieci rekurencyjne

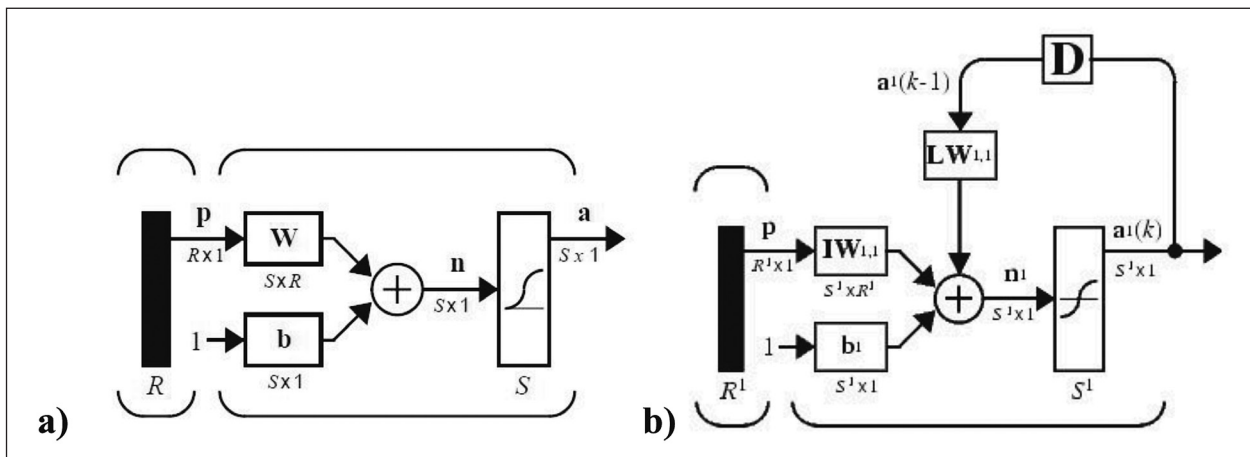
Co do zasady sieci neuronowe po utworzeniu traktuje się jak swego rodzaju czarne skrzynki [12], sieć

przetwarza sygnał wejściowy na wyjściowy, jednak nie ma (zazwyczaj) potrzeby zaglądać do jej wnętrza – macierzy wag, ponieważ dla człowieka jest to „obraz” niezrozumiały, nie jest możliwe wytłumaczenie, która z cech sygnału wejściowego zdecydowała o wyniku predykcji modelu. Zasadniczym elementem sieci jest neuron, którym w istocie jest funkcja matematyczna. Neurony w każdej warstwie są połączone z neuronami w sąsiednich warstwach, a połączenia te tworzą macierz losowo zainicjowanych wag, proces nauki polega na iteracyjnym poprawianiu wag, aby minimalizować funkcję kosztu. Wartość każdego neuronu oblicza się, sumując iloczyn sygnałów wejściowych do danego neuronu oraz ich wag (rys. 3a), wynik powiększa się o dodatkowe obciążenie (ang. *bias*).

Sieci przedstawione na rysunku 3 różnią się następującą cechą. W sieci jednokierunkowej wynik wspomnianego iloczynu skalarne stanowi argument funkcji aktywacji neuronu, której już obliczona wartość jest następnie propagowana do kolejnych neuronów zgodnie z opisem. W przypadku sieci rekurencyjnej argument funkcji aktywacji stanowi, podobnie jak wcześniej, sumę iloczynów wag i wejść wzbogaconą o wartość tego argumentu z poprzedniego kroku czasowego. Dopiero po przetworzeniu przez funkcję aktywacji wynik jest przekazywany dalej (rys. 3b). Zatem sieci rekurencyjne posiadają pewną pamięć wstecz [13].

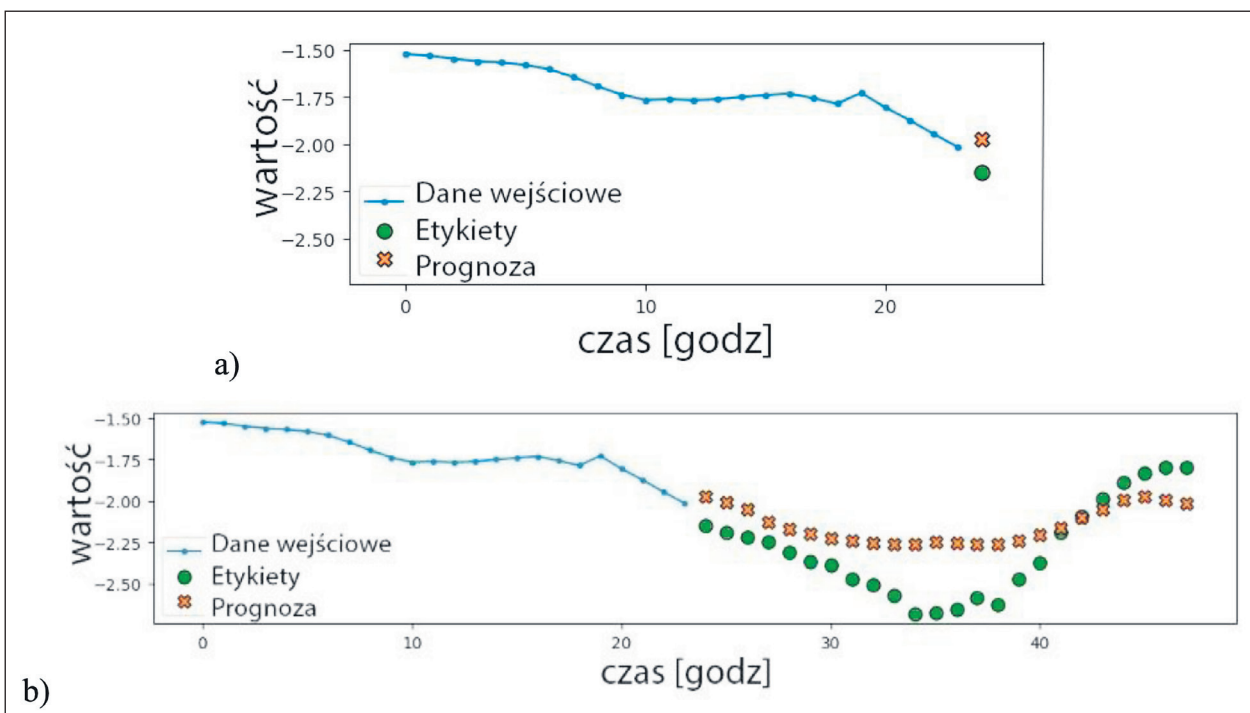
Rozwinięciem koncepcji neuronu rekurencyjnego są sieci LSTM i GRU, które posiadają zdolność do lepszego zapamiętywania sekwencji. Zaznaczenia wymaga pewna istotna kwestia, związana z funkcją aktywacji. Z przyczyn stabilności procesu nauki w przypadku RNN jest to tangens hiperboliczny, przyjmujący wartości od -1 do 1, stąd dane wejściowe należy przekształcić do przedziału albo przedziału -1,1 lub do zbioru o odchyleniu standardowym 1 i średniej 0. Proces ten nazywany jest normalizacją lub standaryzacją [14].

W modelach sieci neuronowych do czynienia mamy z takimi parametrami, jak: liczba warstw ukrytych, liczba neuronów, szybkość uczenia, typ optymalizatora itp., są to wartości podlegające strojeniu, nazywane hiperpa-



► Rys. 3: a – jednokierunkowa sztuczna sieć neuronowa, b – rekurencyjna sieć neuronowa [11]

► Fig. 3: a – one-direction artificial neural network, b – recurrent neural network



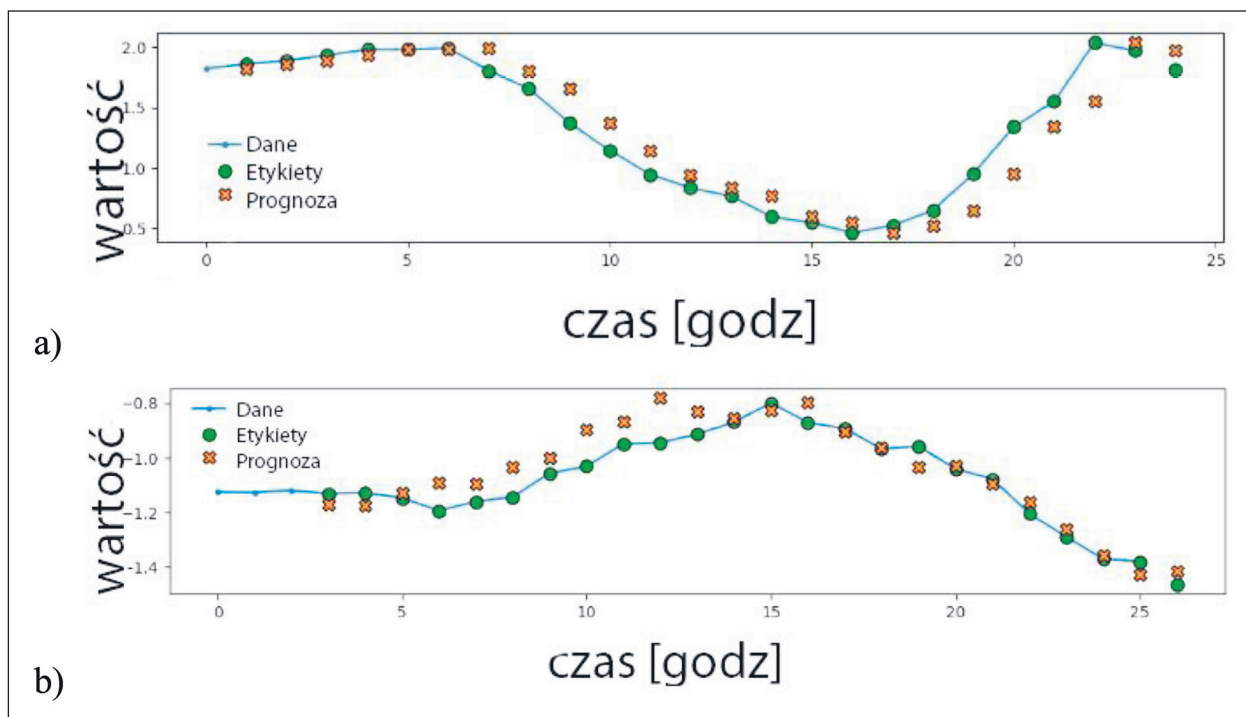
► Rys. 4. Wykres predykcji: a – pojedyncza, b – wiele prognoz [17]

► Fig. 4. Prediction charts: a – single, b – multiple predictions

rametrami. Warto wspomnieć o trudnościach, jakie są związane głównie z tymi parametrami [15], mianowicie: za mała liczba neuronów – brak możliwości odtworzenia nawet prostych funkcji, za duża – utrudnione uczenie z uwagi na dużą liczbę wag (głębokie sieci mogą efektywniej zarządzać parametrami, tj. kilkuwarstwowa sieć osiągnie ten sam wynik przy mniejszej liczbie wag); za mała liczba warstw ukrytych – ograniczone możliwości modelu, utrudnione uczenie się ogólnych wzorców (za słaba siła reprezentacyjna), za duża liczba warstw – wysoka wariancja modelu z uwagi na dużą liczbę stopni swobody (model jest za czuły, możliwość przetrenowania) lub niedotrenowanie modelu z uwagi na możliwość zaniku gradientu (najniższe warstwy nie są właściwie trenowane); za mały współczynnik uczenia – brak zbieżności do globalnego minimum funkcji kosztu,

osiągnięcie i utknięcie w lokalnym minimum, za duży współczynnik uczenia – brak możliwości osiągnięcia minimum ze względu na zbyt duże korekty wag; źle dobrany optymalizator może niepotrzebnie obciążać jednostkę obliczeniową i wydłużać czas obliczeń¹. Innymi słowy, przytaczając Gerona [16], zwiększenie złożoności modelu zazwyczaj zwiększa jego wariancję i zmniejsza

¹ W 2024 r. Królewska Szwedzka Akademia Nauk Nagrodę Nobla w dziedzinie fizyki przyznała Johnowi J. Hopfieldowi i Geoffreyowi E. Hintonowi za „fundamentalne odkrycia i wynalazki, które umożliwiają uczenie maszynowe za pomocą sztucznych sieci neuronowych”. Do 1986 r. bardzo trudno było prawidłowo wytrenować sieć neuronową. Hinton był współautorem publikacji dotyczącej algorytmu wstecznej propagacji błędów – metody uczenia, która zapoczątkowała nowy rozdział w rozwoju uczenia maszynowego i sztucznej inteligencji.



► Rys. 5. Wykres predykcji: a – sekwencyjna prognoza co takt², b – sekwencyjna prognoza w oparciu o wiele taktów [9]
 ► Fig. 5. Prediction graph: a – sequential prediction every cycle, b – sequential prediction based on many cycle

obciążenie. Odwrotny proces następuje w przypadku uproszczenia modelu.

Postęp procesu uczenia monitoruje się za pomocą krzywej uczenia, która przedstawia błąd przyjętej funkcji (zazwyczaj dobrze spisuje się błąd średniokwadratowy, ang. *MSE*) w każdej iteracji osobno dla zbiorów treninowego i walidacyjnego (testowy pozostawiany jest do ostatecznej oceny modelu po zakończonej nauce).

Reasumując, koncepcję pracy, czasami nazywaną potokiem przetwarzania, można podzielić na trzy etapy: *przygotowanie zbioru do nauki* → *praca modelu sieci neuronowej* → *ocena wyniku*

Etap przygotowania zbioru do nauki sieci obejmuje elementarne czynności na nieprzetworzonych danych, tj. odczyt danych, konwersję do postaci szeregu czasowego, przetwarzanie oraz podział na zbiory treninowy, walidacyjny i testowy. Kolejnym etapem jest praca nad modelem (podczas nauki sieci), jego konstruowanie oraz strojenie hiperparametrów. Po wyuczeniu sieć działa na zasadzie czarnej skrzynki, przetwarzając dane wejściowe i generując wyniki. Ostatnim etapem jest ocena uzyskanego wyniku. Podczas nauki zazwyczaj jest to błąd średniokwadratowy bądź średni błąd bezwzględny. Takie podejście umożliwia potokowe przetwarzanie danych.

6. Podstawowe zadania uczenia maszynowego

Pośród głównych zadań ML, tj. predykcji czy klasyfikacji, lub zadań złożonych, czyli: rekonstrukcji wejść,

detekcji anomalii czy odsumowania, w dalszej części rozwinęte zostanie zagadnienie predykcji sekwencji, która może być wykonana w dwojaki sposób, jako:

- pojedyncza prognoza (pojedyncze wyjście – rys. 4a) lub wiele prognoz (wiele wyjść naraz – rys. 4b),
- sekwencyjna prognoza co takt (rys. 5a) lub sekwencyjna prognoza w oparciu o wiele taktów (rys. 5b).

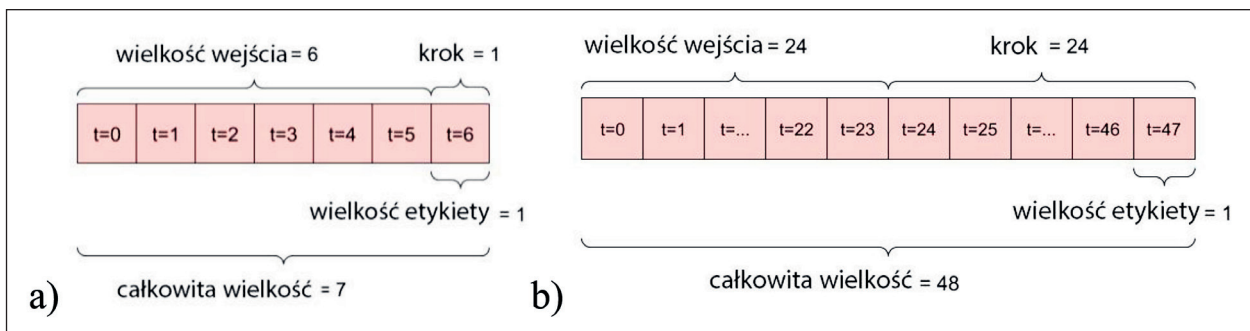
Rysunek 6a przedstawia prognozę godzinę do przodu na podstawie sześciu poprzednich pomiarów (pojedyncza prognoza), natomiast rysunek 6b przedstawia prognozę na 24 godziny do przodu na podstawie poprzednich dwudziestu czterech pomiarów (wiele wyjść naraz). Uzyskanie różnych wyników, jak na rys. 4 i 5, spowodowane jest odpowiednim podziałem zbioru treninowego oraz danych wejściowych. Dobór konkretnego wariantu zależy od rodzaju zadania i jego warunków brzegowych.

7. Analiza sekwencji

Pracę rozpoczęto od importu wszystkich potrzebnych pakietów środowiska programistycznego. Pomiaru pierwotnie zawarte w plikach *.csv* zostały wczytane do obiektu *Pandas Dataframe* (zoptymalizowany do przechowywania i przetwarzania danych w pamięci komputera). Użycie argumentów *parse_dates* oraz *dayfirst* skutkowało konwersją dat do obiektu *DatetimeIndex* umożliwiającego przetwarzanie szeregów czasowych. Jeżeli pewne cechy wymagały obliczenia (przetworzenia), na tym etapie można je było wykonać na każdym pliku źródłowym z osobna.

Z uwagi na brak wartości skumulowanej konwergencji oraz pożądaną wartość ujemną wyniku obliczono dzienną skumulowaną wartość konwergencji, którą

² Takt *t*, ang. *time step* – wskazuje porządek, w jakim pojawiają się dane w danych sekwencyjnych. W każdym takcie *t* neuron w sieciach RNN otrzymuje sygnał wejściowy oraz wynik wygenerowany przez siebie w takcie poprzednim – *t-1*.



► Rys. 6. Rodzaje predykcji: a – pojedyncza, b – wiele prognoz [17]

► Fig. 6. Types of prediction: a – single, b – multiple predictions

zapisano w nowej kolumnie (tab. 1). Ramkę danych zapisano w formacie .csv do dalszej analizy. Pierwszym krokiem oczyszczenia danych była zbiorcza wizualizacja.

Dane na tym etapie posiadały różną długość, załamania mogły wskazywać na wystąpienie określonych zdarzeń w trakcie pomiarów. Wizualizacja (rys. 7) umożliwiła potwierdzenie, że dane należało oczyścić, np. wybierając szeregi o stałej długości, pozbawione gwałtownych załamania. Sprawdzono również, czy załamanie nie było wywołane wstrząsem górotworu. Po filtracji danych otrzymano największą wartość dla pola I, wynoszącą 61 mm. Następnie odczytano datę tego pomiaru. Znając daty wstrząsów wysokoenergetycznych, po ich porównaniu potwierdzono, że dla pola w tym dniu do niego doszło. Następnie sprawdzono, ile razy przekroczona została średnia wartość konwergencji dla danego punktu pomiarowego po wstrząsie. Oblicza się to jako iloczyn wartości dla dnia pomiaru po wstrząsie oraz średniej konwergencji dla punktów pomiarowych. Stwierdzono, że wstrząs wywołał 7-krotny przyrost średniej konwergencji. Wykonano zatem wizualizację danych do dnia poprzedzającego omawiany wstrząs.

Wykres, który prezentujemy na rysunku 8, wyglądał obiecująco, jednak długość szeregów czasowych była zbyt duża. Komentarza wymaga pewien dylemat natury prognozy – prognoza dzień do przodu na bazie 26 pomiarów z pewnością osiągałaby wysokie wartości metryk pomiarowych, jednak w praktyce byłaby mało użyteczna z uwagi na uzyskanie pierwszej prognozy po 26 dniach. Zasadne staje się pytanie, jak długo możemy czekać na wynik i jaką dokładność (niepewność) możemy osiągnąć? Dla dalszej części przyjęty został okres 15 dni.

Na tym etapie skorzystano z metod przetwarzania szeregów czasowych, uzyskując pożądany zakres dat, który obejmuje okres 15 dni przed wystąpieniem

niepożądanego wstrząsu. Na tak uzyskanych danych dokonano odwzorowania lokalizacji konwergometru przy użyciu odpowiednich kolorów: niebieski oznacza linię rozczinki, zielony linię likwidacji, czerwony filary wielkogabarytowe, zaś czarny filary o przekroju prostokątnym. Wizualizacja tych danych została przedstawiona na rysunku 9. Pierwszym wnioskiem było wyraźne rozdzielanie grup krzywych konwergencji w zależności od lokalizacji (z wyjątkiem jednej krzywej), co potwierdziło ogólnie znaną wiedzę. Średnie wartości dla punktów pomiarowych w okresie 15 dni przed wstrząsem wynosiły od 1,7 mm do 12 mm na dobę, przy średniej dla całego pola wynoszącej 4,4 mm na dobę.

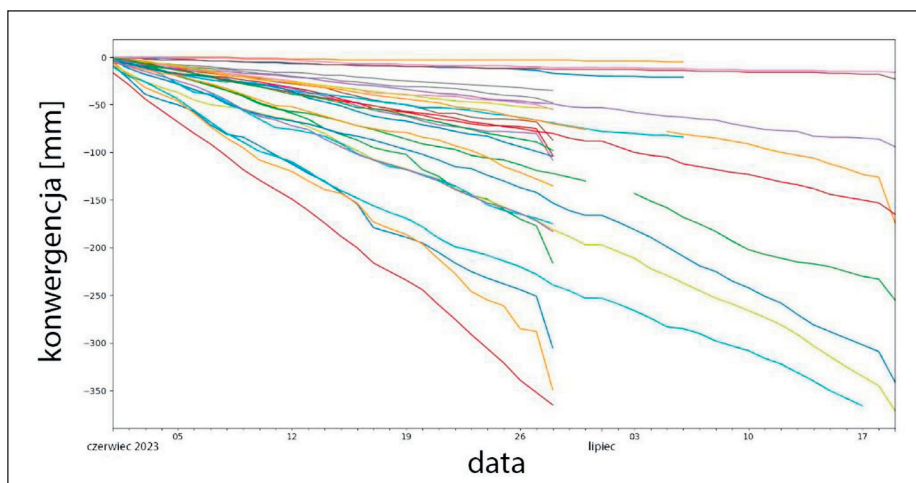
Jak zatem kształtowała się średnia konwergencja w zależności od lokalizacji punktu pomiarowego? Na rysunku 10a przedstawiono wyniki wskazujące na najwyższą konwergencję na linii frontu (F). Stosunkowo wysokie wartości, ale zarazem najniższe z całej grupy, osiągnęły filary wielkogabarytowe (FW) oraz o wydłużonym przekroju (FP). Niemniej są to wartości konwergencji cztery razy niższe niż na linii frontu. Konwergencja na linii likwidacji (L) osiąga połowę wartości konwergencji na linii frontu. Na rysunku 10b rozdzielono wykres ze względu na pola, zaznaczono na nim obliczoną średnią wartość konwergencji.

Ostatni wykres (rys. 11) przedstawia najwyższą wartość konwergencji z rozróżnieniem miejsca prowadzonego pomiaru. Najwyższa wartość (po wstrząsie) została stwierdzona w punkcie znajdującym się na froncie eksploatacji. Zgodnie z założeniami projektowymi punkty w rejonie filarów o zwiększonych wymiarach (FP i FW) nie uległy wpływowi wstrząsu. Kolejny punkt przy zrobach (L) uległ gwałtownemu przemieszczeniu. W sposób naturalny nasuwa się pytanie o odległość poszczególnych punktów od epicentrum wstrząsu oraz

► Tab. 1. Przykład ramki danych – obliczenie sumy konwergencji

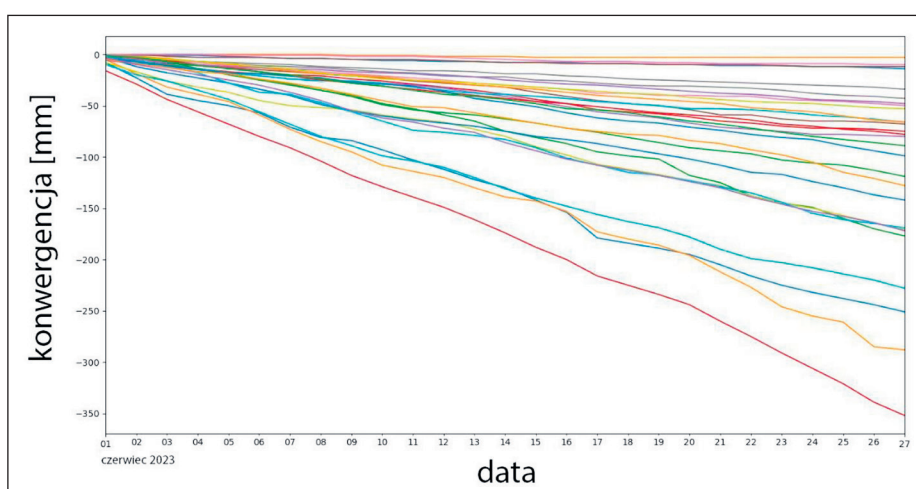
► Table 1. Example data frame – convergence sum calculation

		Pole-I	Pole-I	Pole-I
nr	data	delta	lokalizacja	suma
1	2023-06-01	8	F	-8
	2023-06-02	9	F	-17
...



► Rys. 7. Zbiorczy wykres danych

► Fig. 7. Summary chart



► Rys. 8. Wykres ograniczony do dnia wstrząsu

► Fig. 8. Chart limited to the day of the mining shock

o stabilność filarów w pobliżu zrobów – są to jednak kwestie wymagające odrębnych analiz. Na tym zakończymy część poświęconą wstępnym etapom pracy i wizualizacji danych. Kolejnym krokiem była normalizacja danych do przedziału wartości od 0 do 1.

8. Normalizacja zbiorów uczących

Normalizacja przeprowadzana jest na zbiorze treningowym zgodnie z zasadą nieuwzględniania pozostałych zbiorów. Z uwagi na wiedzę, że dane pochodzą z tego samego procesu i że zostały już sprawdzone i oczyszczone (np. na okoliczność elementów odstających czy brakujących), przeprowadzono ją raz na całym zbiorze przed podzieleniem na zbiory treningowe i testowe³. Przed normalizacją zbioru opisywała statystyka – maks.: -2, min.: -352, śr.: -81,2, std.: 67,2, natomiast po przeprowadzeniu normalizacji zgodnie z wzorem (*wartość* –

³ W uczeniu maszynowym normalizacja powinna być przeprowadzana wyłącznie na zbiorze treningowym. Pozwala to uniknąć „przecieku danych”, ang. *data leakage* (przyp. rec.).

minimum)/(*maximum* – *minimum*) uzyskano statystykę zbioru: maks.: 1, min.: 0, śr.: 0,77, std.: 0,192.

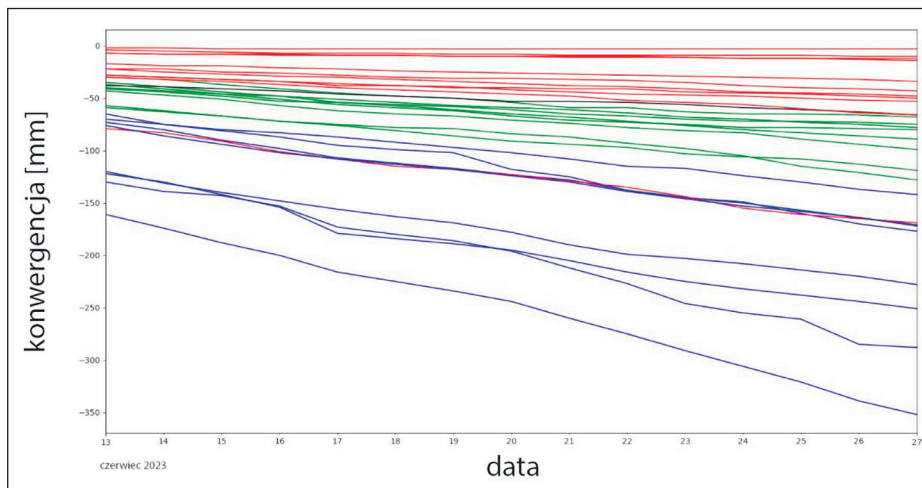
Warto zauważyć, że w dalszej części wykorzystana została funkcja aktywacji, która zwracała wartości w przedziale od 0 do 1, co uzasadniało zastosowanie normalizacji min–max.

9. Charakterystyka modeli sieci neuronowych

Wykonane zostały cztery zbiory uczące dla czterech zadań prognostycznych, a mianowicie:

- 1) prognoza jednej wartości do przodu – SF (ang. *single-output, forward*),
- 2) prognoza wielu wartości do przodu – MF (ang. *multi-output, forward*),
- 3) prognoza sekwencyjna jednej wartości do przodu – SS (ang. *sequence, single-time-step*),
- 4) prognoza sekwencyjna na podstawie trzech poprzednich kroków – jeden krok do przodu – SM (ang. *sequence, multi-time-step*).

W dalszej części omówione zostały wspomniane modele wraz z ich zbiorami, po czym zamieszczono zbiorcze wykresy wraz z podsumowaniem otrzymanych



► Rys. 9. Wykres ograniczony do 15 dni przed wystąpieniem wstrząsu

► Fig. 9. Graph limited to 15 days before shock onset

wyników. Należy zwrócić uwagę, że przytoczony podział ma charakter ogólny, a przyjęcie konkretnej liczby prognozowanych wartości i odczytywanych „wartości wstecz” będzie już zależać od założeń brzegowych.

9.1. Prognoza jednej wartości do przodu – SF

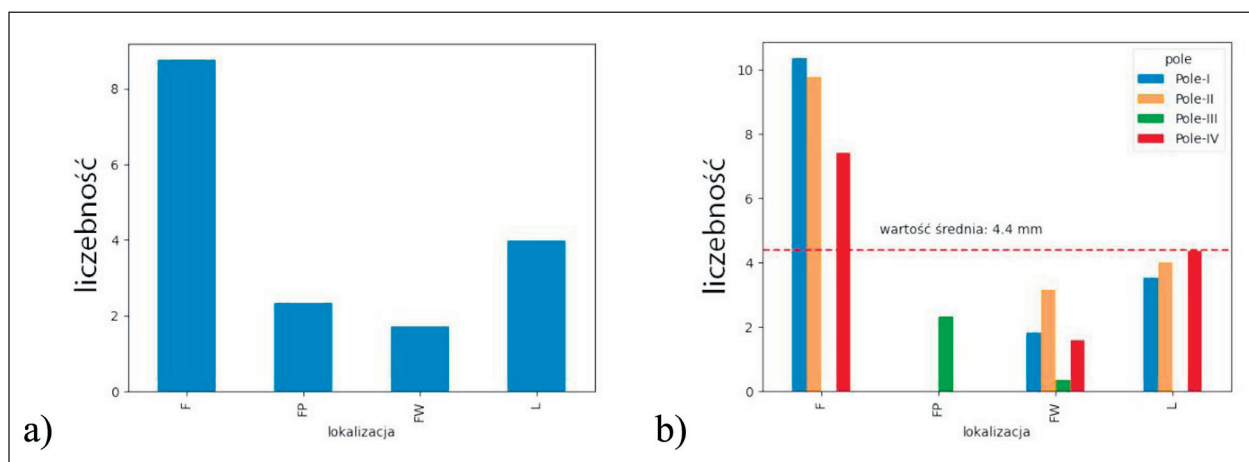
Ten typ zbiorów był najprostszy, ponieważ wartości docelowe (prognozowane) stanowiły ostatnią wartość sekwencji, stąd też zbiór danych wejściowych i wyjściowych otrzymany został poprzez indeksowanie ramki danych. W tego typu zadaniu należy ustawić odpowiednią wymiarowość danych, co wynika ze standardów uczenia modeli. Dane wejściowe mają strukturę obejmującą trzy wymiary: rozmiar wsadu, długość sekwencji i liczbę cech. Dla omawianych danych treningowych będzie to odpowiednio: liczebność punktów pomiarowych – 28, długość pomiaru – 14 dni oraz liczebność pomiarów – 1, z uwagi na wyłączny pomiar konwergencji. W środowisku programistycznym zwracany jest wymiar danych, potwierdzając tę strukturę (28, 14, 1). Zadanie polegało na prognozie wartości do przodu, dlatego dane wyj-

ściowe posiadały wymiarowość (28,1). Następnie dane były dzielone na zbiory treningowy i testowy z przeznaczeniem 20% danych do zbioru testowego, co jest powszechnie stosowaną proporcją.

Model zbudowany w pakiecie TensorFlow to model sekwencyjny (Sequential), składający się z czterech komórek rekurencyjnych klasy SimpleRNN oraz warstwy wyjściowej, którą stanowił neuron (klasy Dense) z liniową funkcją aktywacji. Funkcją straty modelu był błąd średniokwadratowy (MSE), natomiast metryką uczenia modelu – średni błąd bezwzględny (MAE). W tym przykładzie, w konstruktorze klasy SimpleRNN, ustalono krotkę (14,1) oznaczającą ładowanie kolejnych 14 pojedynczych wartości do modelu.

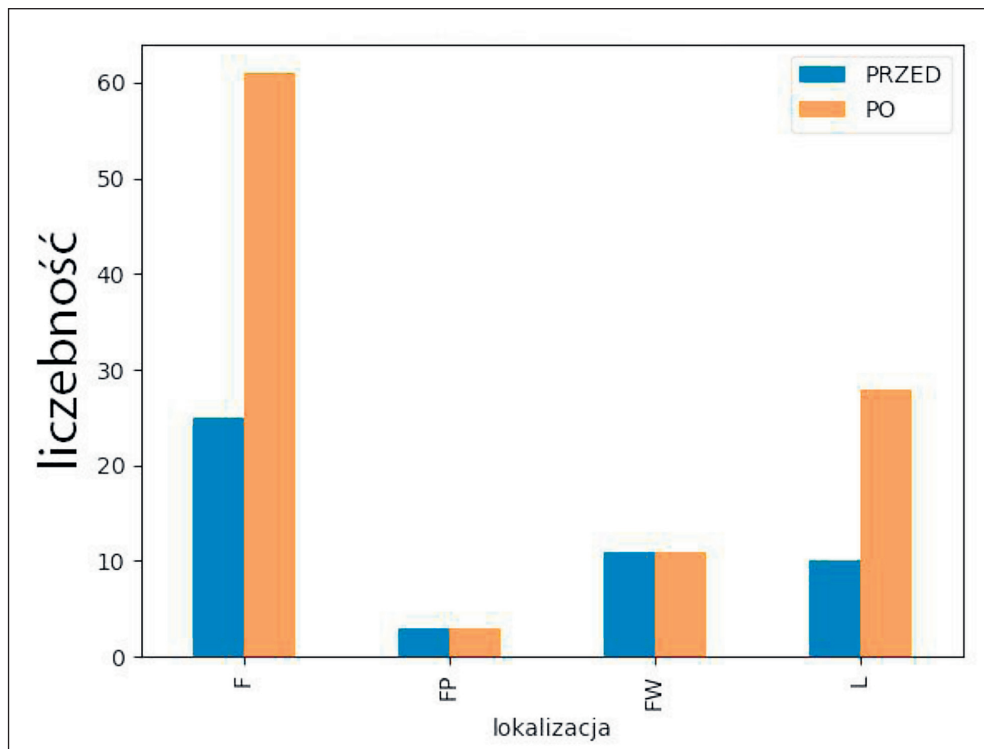
9.2. Prognoza wielu wartości do przodu – MF

W tym przypadku prognoza była wykonywana w pięciu kolejnych krokach, w oparciu o 10 poprzednich wartości. Analogicznie do modelu SF przeprowadzony został podział zbioru. Załóżmy, że interesuje nas prognoza na 5 kolejnych dni, wówczas podziału zbioru



► Rys. 10. Średnia wartości konwergencji dla punktów pomiarowych: a – zbiorczo, b – z podziałem na pola; F – front, FP – filar o zwiększonym wymiarze, FW – filary wielkogabarytowe, L – linia likwidacji

► Fig. 10. Average convergence values for measurement points: a – collectively, b – divided into fields; F – front, FP – pillar of increased size, FW – large-size pillars, L – liquidation line



► Rys. 11. Maksymalna wartość konwergencji dziennej przed i po wstrząsie; F – front, FP – filar o zwiększonym wymiarze, FW – filary wielkogabarytowe, L – likwidacja

► Fig. 11. Maximum daily convergence value before and after shock; F – face, FP – pillar of increased size, FW – large-sized pillars (stiffened), L – gob line

dokonujemy poprzez indeksowanie, z wydzieleniem odpowiednich 5 dni. Zmiana ta będzie widoczna na wykresach zawierających prognozy.

Model zbudowany był z dwóch warstw zawierających po cztery neurony rekurencyjne (klasy *SimpleRNN*). Pierwsza warstwa rekurencyjna zwracała pełne sekwencje (atrybut *return_sequences=True*), a atrybut *input_shape* ustawiono jako krotkę *(None,1)*, która symbolizowała dwa ostatnie wymiary zbioru. Ustawienie wartości *None* dla wymiaru *długość sekwencji* umożliwiło modelowanie sekwencji o dowolnej długości, co pozwoliło na większą elastyczność i odróżnia tę konstrukcję od modelu SF, który ogranicza się do stałej długości sekwencji. Wspomniana elastyczność polegała na tym, że wymiar zbioru (sekwencje o stałej długości 10 dni) był odczytywany samodzielnie przez model.

9.3. Prognoza sekwencyjna jednej wartości do przodu – SS

Przygotowanie zbioru danych przy prognozie sekwencyjnej wymagało dodatkowego ich przetworzenia, jednak były to operacje typowe dla języka programowania, tj. iteracja, indeksowanie czy przypisanie. Cechą charakterystyczną tego typu prognozy jest następujący przepływ danych: prognoza jest wykonywana na pięć taktów do przodu w każdym kroku obliczeniowym, np. w taktie 0 generowany jest wynik dla taktów od 1 do 5 oraz obliczany błąd prognozy, kolejno w taktie 1 generowany jest wynik dla taktów od 2 do 6, tym samym w taktie 10 generowany jest wynik od taktu 11 do 15

i zwracany błąd z ostatniego kroku. Ostatni krok był o tyle istotny, że podczas wizualizacji szczególną uwagę należało zwrócić na dane z ostatniego taktu, który był kluczowy dla analizy wyników (wymiar predykcji odpowiada liczbie taktów). Kolejną konsekwencją była przyjęta metryka, która obliczała swoją wartość tylko dla danego taktu.

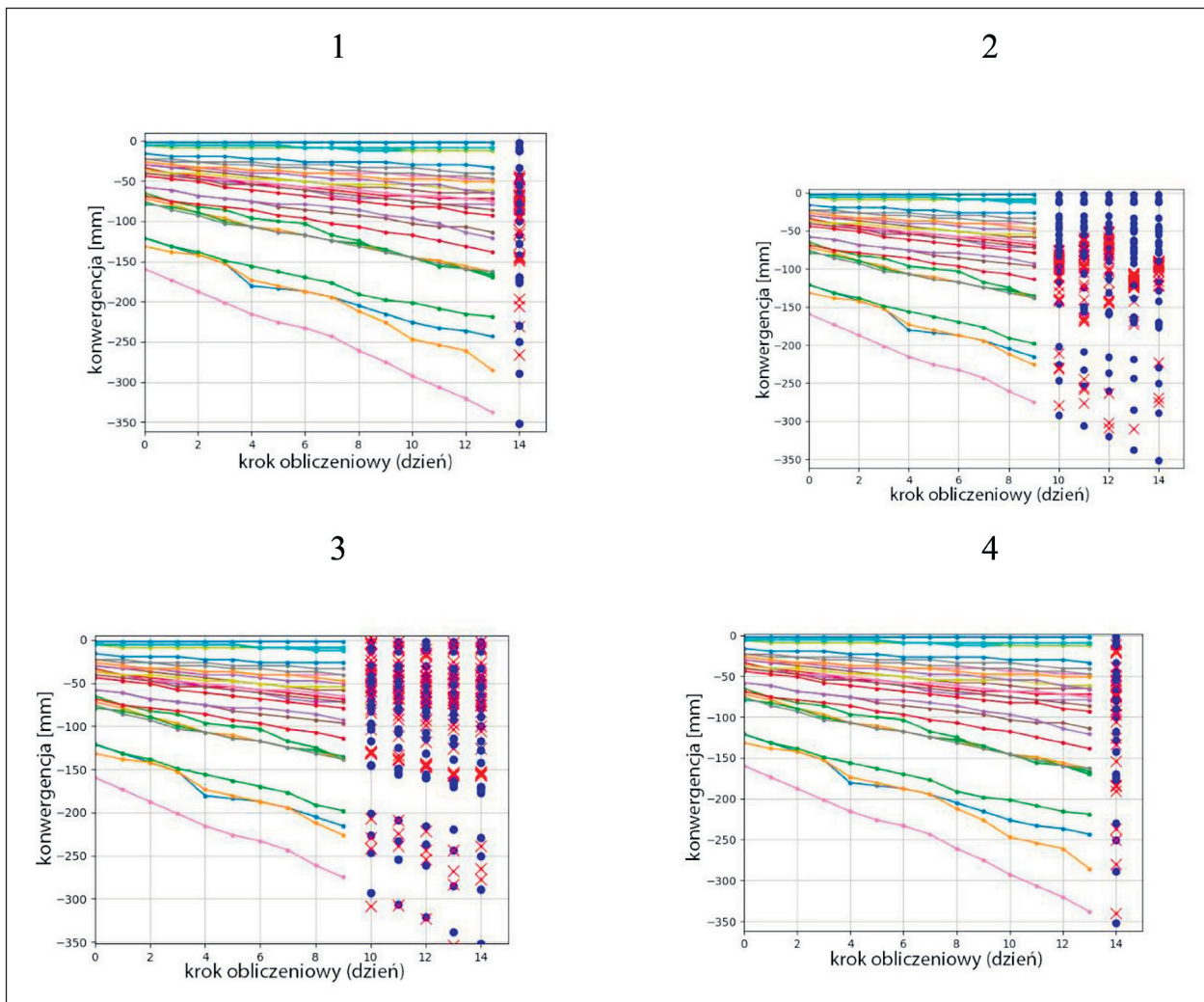
W przeciwieństwie do metryki funkcja błędu obliczała gradient dla wszystkich taktów, co mogło powodować niestabilność procesu uczenia. Uniknięcie lokalnych minimów funkcji błędu wymagało wyższych wartości współczynnika uczenia, co w konsekwencji prowadziło do lepszych wyników niż przy poprzednich metodach, jednak mogło powodować, że krzywa uczenia stawałaby się mniej stabilna i odbiegała od idealnego, gładkiego przebiegu. Istotną cechą wpływającą na lepsze wyniki takich sieci jest również fakt, że każdy takt jest kolejną

► Tab. 2. Podsumowanie wyników prognozowania – MAE

► Table 2. Summary of forecasting results – MAE

Nazwa modelu	Metryka – MAE
SF	0,0672
MF	0,106
SS	0,02*
SM	0,0239*

* z ostatniego taktu



► Rys. 12. Wynik prognoz poszczególnych modeli: 1 – SF, 2 – MF, 3 – SS, 4 – SM

► Fig. 12. Result of model predictions: 1 – SF, 2 – MF, 3 – SS, 4 – SM

jednostką treningową, dzięki czemu model jest trenowany intensywniej – jeśli przetwarzamy X taktów, oznacza to, że model przeszedł przez X dodatkowych kroków treningowych. Zwiększa to jego zdolność do rozpoznawania wzorców danych i pozwala osiągnąć lepsze wyniki. W dalszej pracy nad strojeniem sieci wprowadza się tzw. metody regularyzacji, jednak w tym przypadku nie była ona wymagana.

9.4. Prognoza sekwencyjna na podstawie trzech poprzednich taktów – jeden krok do przodu – SM

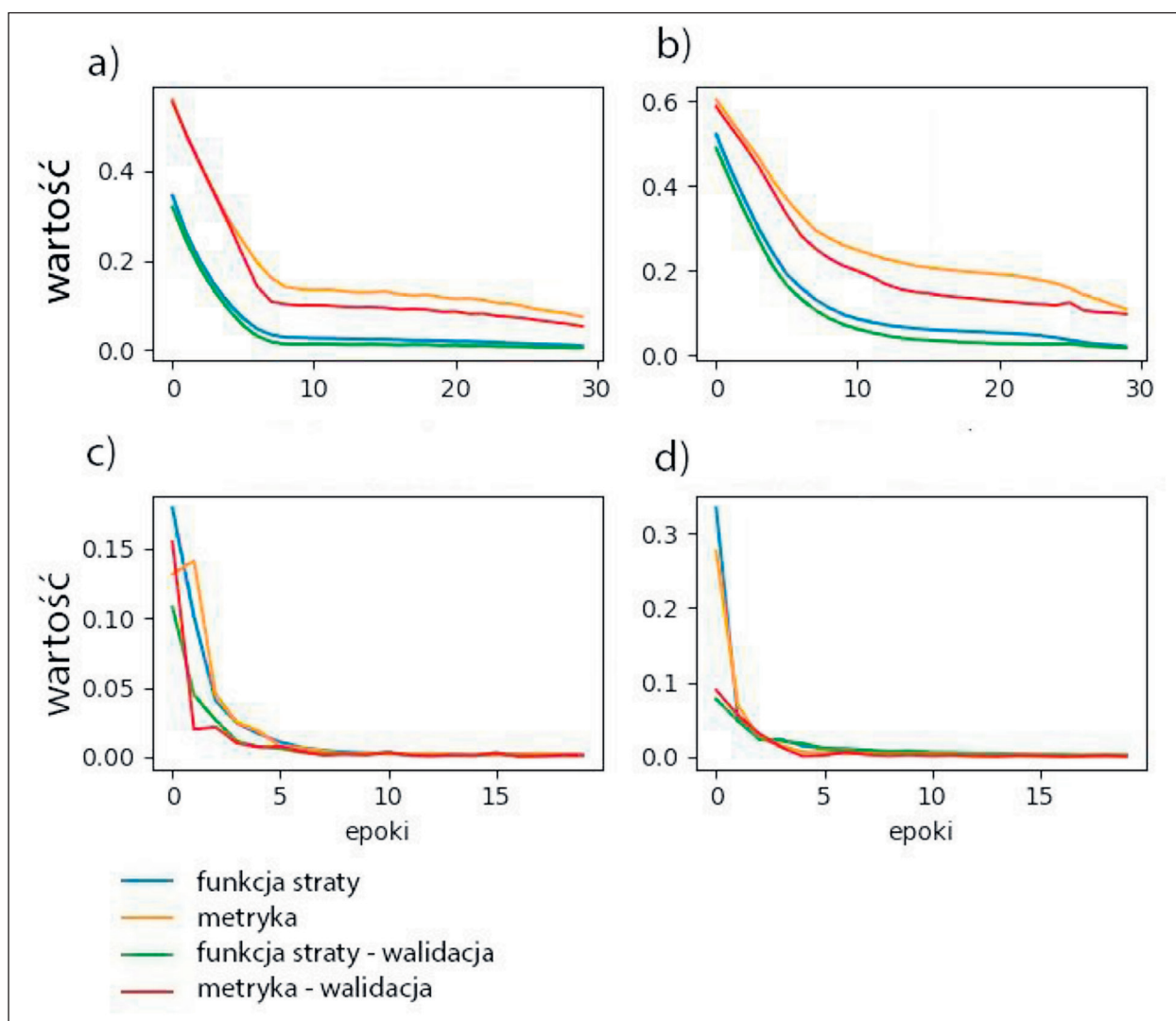
Ostatnia sieć także bazuje na zasadzie sekwencyjnego odczytywania taktów, jednak w tym przypadku przyjęto prognozowanie pojedynczej wartości w oparciu o trzy kolejne takty. Prognoza wykonywana jest jeden takt do przodu przy każdym kroku obliczeniowym w oparciu o takty -1, -2 oraz -3 względem prognozowanego taktu. Oznacza to, że w kroku 0 (zgodnie z numeracją 0, 1, 2) generowany jest wynik dla taktu 3 na podstawie danych z taktów od 0 do 2. Sieć ta powinna też osiągać najlepsze wyniki, ponieważ krótszy, np. trzydniowy, przedział danych wejściowych pozwala sieci lepiej reagować na

zmienność pomiarów i różne odchylenia. Zaleta ta nie została opisana w niniejszym artykule, ale warto podkreślić jej znaczenie w kontekście dokładności prognoz.

10. Modelowanie rekurencyjnymi sieciami neuronowymi

Cztery przedstawione zadania (rys. 12) stanowią w istocie kombinację różnych podejść do przygotowania zbiorów uczących, a sprowadzają się do podstawowego podziału, jaki zaprezentowano na wstępie artykułu. Pokazuje to elastyczność projektowania badań (modelu), a jednocześnie uniemożliwia porównanie wyników, jakie są do uzyskania.

Aby precyzyjnie określić dokładność modelu, konieczne jest przeprowadzenie badań z wykorzystaniem komórek LSTM lub GRU, regularyzacją modelu oraz analizą wpływu wielkości zbioru uczącego na metrykę modelu. Jest to jednak odrębne zagadnienie wymagające osobnego opracowania, z racji tego autor ograniczył się do stwierdzenia, że w analizowanym przypadku modele sekwencyjne generalnie osiągały o rząd wielkości lepsze wartości średniego błędu bezwzględnego – MAE (tab. 2) oraz że do osiągnięcia minimum wymagana



► Rys. 13. Krzywe uczenia: a – model SF, b – model MF, c – model SS, d – model SM

► Fig. 13. Learning curves: a – SF model, b – MF model, c – SS model, d – SM model

była mniejsza liczba epok (drugi wniosek byłby istotny przy skomplikowanych modelach, gdzie obliczenia są czasochłonne).

Na rysunku 13 przedstawiono krzywe uczenia dla czterech wymienionych modeli. Krzywe przedstawiają wartości funkcji błędu oraz wybranej metryki dla zbiorów treningowych i walidacyjnych. Dla modeli SS i SM przyjęto metrykę błędu średniokwadratowego ostatniego taktu. Dowodem zasadności wykorzystania sieci RNN do modelowania różnych zadań było wykazanie możliwości zbieżności tych modeli na podstawie krzywych uczenia. Pierwszą cechą krzywych potwierdzającą dobre dopasowanie modelu jest odległość wypłaszczenia krzywej walidacyjnej i treningowej – wartości zbliżone są pożądane. Drugą cechą jest osiągnięta wartość wskaźników. Wartości w tabeli 2 oraz kształt krzywych na

rysunku 13 są obiecujące, ponieważ w etapie uczenia nie zastosowano żadnych technik regularyzacji modelu, z racji tego można przypuszczać, że metryki modelu mogą ulec polepszeniu po ich wprowadzeniu.

11. Podsumowanie

Wykonane modele, reprezentujące cztery różne zadania prognostyczne, osiągnęły po etapie uczenia zbieżność krzywych uczenia na zbiorach walidacyjnych, wobec czego osiągnięto kryterium prawidłowego wytrenowania sieci. Umożliwi to kontynuację prac, w ramach których można skupić się na trenowaniu sieci na trudniejszych przykładach (np. o różnej długości), postawieniu bardziej złożonych zadań niż prognoza kilku wartości do przodu, np. obliczaniu błędu rekonstrukcji itd.

Issues of analysing convergence measurements using machine learning techniques and recurrent neural networks

Abstract: The article contains the author's experience with four recurrent neural network models made in the Tensorflow environment, designed to predict the convergence values of mine workings. It was shown that convergence measurements are sequential data and simple processing operations were performed on them. After formulating the criterion of convergence of learning curves, as a measure of correct training of the network, the construction of models was started. The models were made in four forecast variants, modelling both single and multiple outputs. Based on the learning curves, the fulfilment of the criterion was demonstrated and the possibility of using recurrent networks in modelling simple convergence courses was proven.

Literatura

1. P. Matkowski, Zarządzanie monitoringiem zagrożeń w górnictwie, Journal of the Polish Mineral Engineering Society, 2017.
2. R.L.A. Popova Zhuhadar, Comparative View of AI, Machine Learning, Deep Learning, and Generative AI, https://commons.wikimedia.org/wiki/File:Unraveling_AI_Complexity_-_A_Comparative_View_of_AI,_Machine_Learning,_Deep_Learning,_and_Generative_AI.jpg, dostęp: 6.10.2024.
3. www.gartner.com, dostęp: 6.10.2024.
4. A. Anish, Time series analysis, 2020, <https://medium.com/swlh/time-series-analysis>, 1.10.2023.
5. P.J. Brockwell, International Encyclopedia of Education (Third Edition), 2010.
6. R.J. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice, OTexts, 2018.
7. D. Wolper, The lack of a priori distinctions between learning algorithms, 1996.
8. R. Kurban, Boost your Network Performance. Hyperparameter Tuning and Training Optimization, 2021, <https://towardsdatascience.com/boost-your-network-performance-cc0a2a95c5ef>, dostęp: 2023.
9. H. Marczak, Ocena zaciskania wyrobisk chodnikowych na podstawie pomiarów konwergencji, Postępy Nauki i Techniki, 2010.
10. J. Butra, Eksploatacja złoże rud miedzi w warunkach zagrożenia tąpnięciami, Wrocław: KGHM Cuprum sp. z o.o. CBR, 2010.
11. www.mathworks.com, 6.10.2023.
12. P. Blazek, Why we will never open deep learning's black box," 2022, <https://towardsdatascience.com/why-we-will-never-open-deep-learning-black-box-4c27cd335118>, 1.10.2023.
13. H. Sak i in., Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. arxiv.org, 2014, <https://doi.org/10.48550/arXiv.1402.1128>, dostęp: 1.10.2023.
14. J. Sevilla, Importance of input data normalization for the application of neural networks to complex industrial problems, DOI:10.1109/23.589532, dostęp: 2023.
15. S. Sinha, Neural Networks: Problems & Solutions, <https://towardsdatascience.com/neural-networks-problems-solutions-fa86e2da3b22>, dostęp: 2023.
16. A. Geron, Uczenie maszynowe z użyciem Scikit-learn i Tensor Flow. Katowice, Helion, 2020.
17. www.tensorflow.org, data dostępu; 6.10.2023.