Raport z dodatkowego projektu realizowanego w ramach przedmiotu Sztuczna Inteligencja i Sztuczne Życie

Wykonujący: Łukasz Abramowicz

Nr albumu: 454028

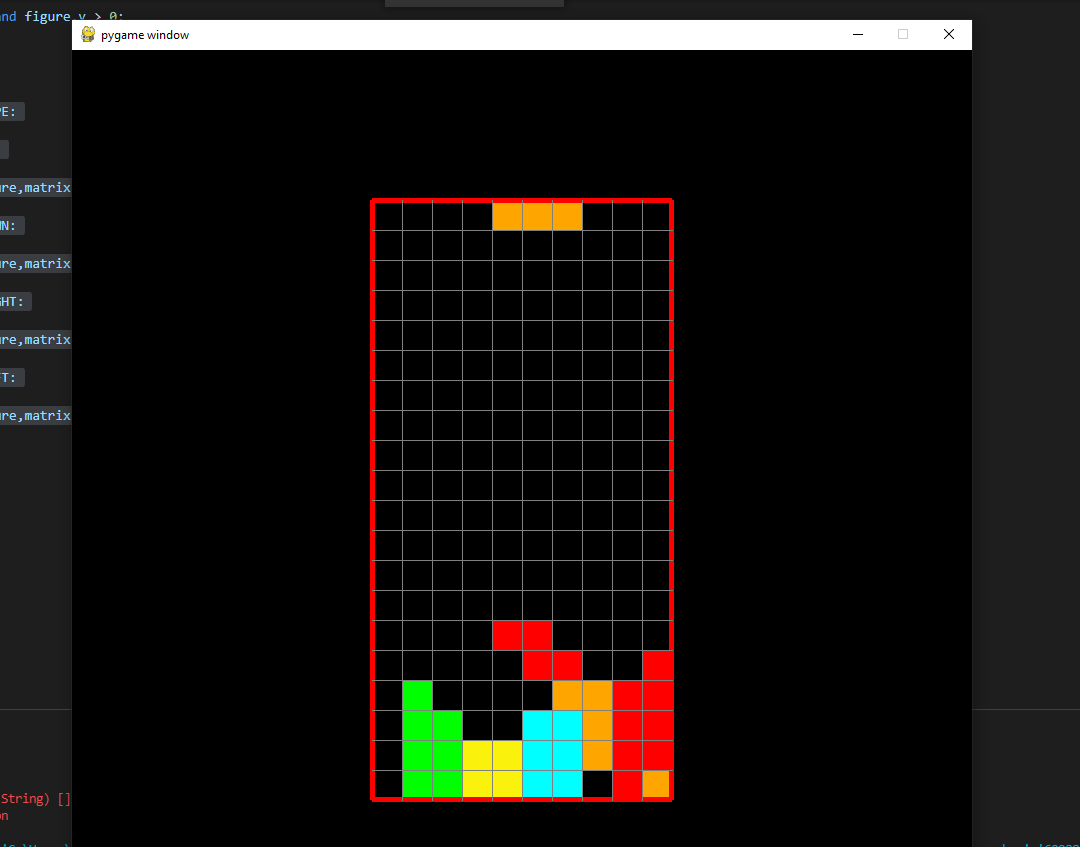
Cel: konstrukcja sztucznej sieci neuronowej i nauczenie jej gry w napisany przeze mnie Tetris przy pomocy języka Python i biblioteki pygame

Prowadzący projekt: Dr Andrzej Gajda

Przedmiot: Sztuczna inteligencja i Sztuczne Życie, Kognitywistyka, semestr 4.

Przed przystąpieniem do projektu:

* Przy pomocy wiedzy na temat biblioteki pygame zdobytej w trakcie pracy przy projekcie z przedmiotu Programowanie na semestrze 2. napisałem grę Tetris, aby zgłębić lepiej jej zasady i parametry. Plansza ma postać matrycy, w której komórkach znajdują się tuple 3-elementowe kodujące kolor w formacie RGB

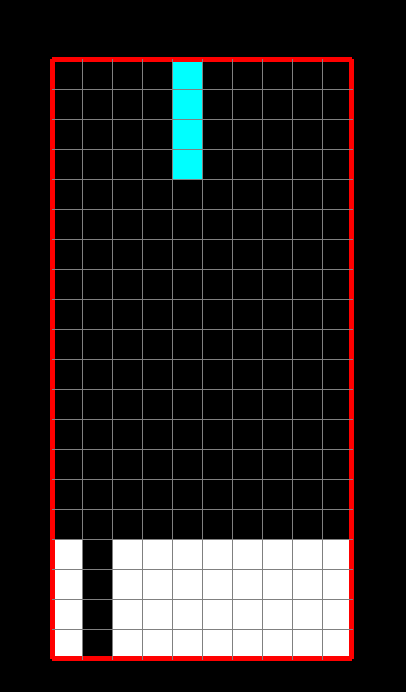


Pierwsze 2 tygodnie:

* Przegląd bibliotek odpowiedzialnych za tworzenie sztucznych sieci neuronowych
* Czytanie dokumentacji i oglądanie poradników do bibliotek Keras, Keras-RL, OpenAi Gym, TensorFlow oraz poradników dotyczących teorii sztucznych sieci neuronowych
* Poznałem zasady pracy przy użyciu wspomnianych wyżej bibliotek i zdobyłem intuicje jak odnośnie ich działania oraz poznałem zasady teoretyczne działania sztucznych sieci neuronowych

Tydzień 1 pracy:

* Zmodyfikowałem kod Tetrisa tak, aby generował proste matryce dla figury podłużnej z zapełnionymi już poziomami dolnymi, w których znajdowała się luka dopasowana do figury.



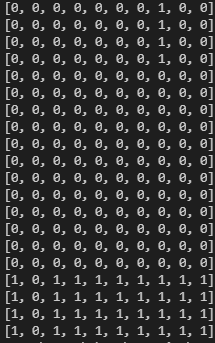
* Wstępnie zastanawiałem się nad architekturą sieci

Tydzień 2 pracy:

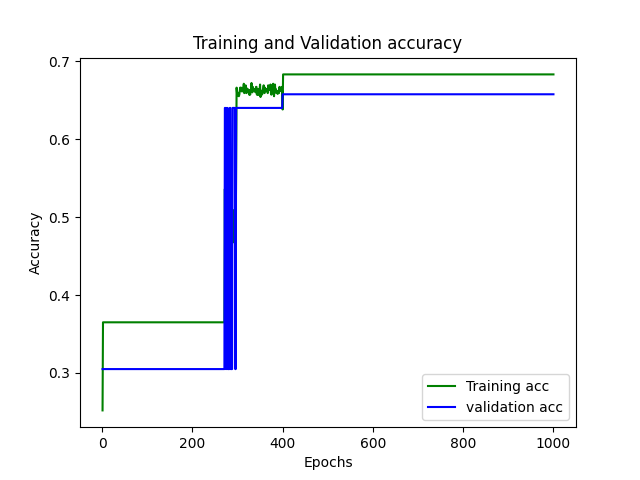
* Zaimplementowałem pierwszą sieć i próbowałem wprowadzić do niej całą matrycę Tetrisa, jednak pojawiały się błędy w przyjmowaniu przez sieć inputu, próbowałem różnych kształtów matrycy i zapisywania jej w różnych typach jednak to nie pomogło
* Szukałem informacji jak zaimplementować właśne środowisko w OpenAi Gym, aby z Tetrisa stworzyć własne środowisko i zacząłem jego podstawową konstrukcję
* Nauczyłem się jak funkcjonuje proste środowisko biblioteki OpenAi Gym

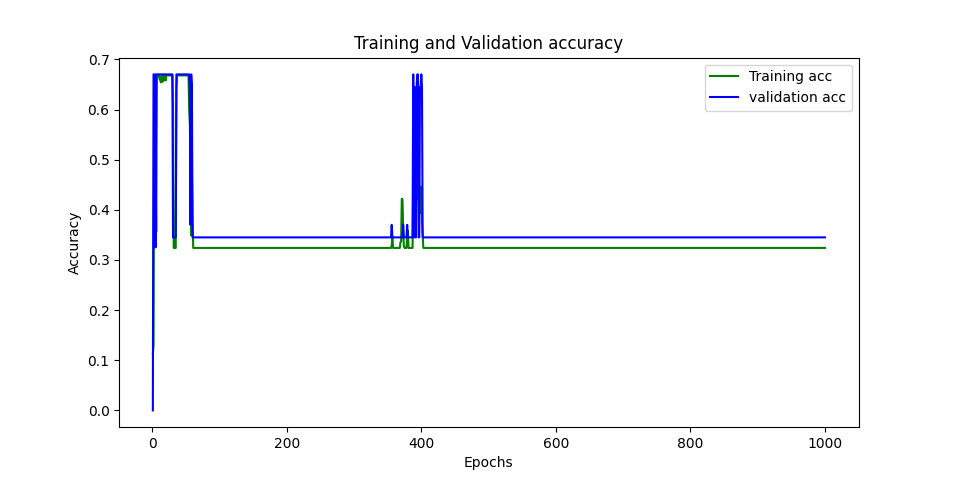
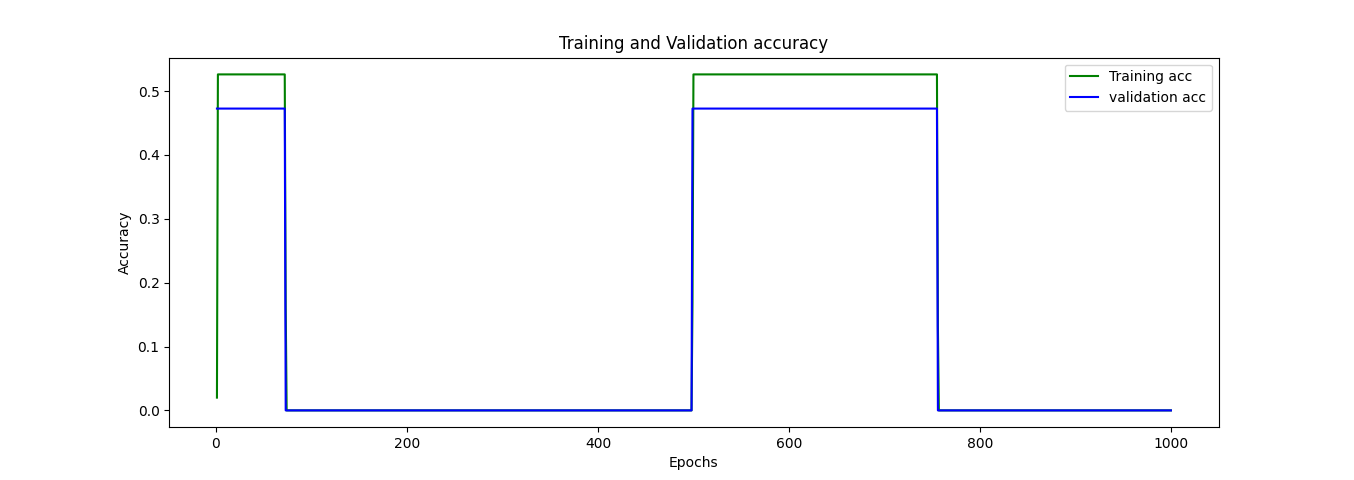
Tydzień 3 pracy:

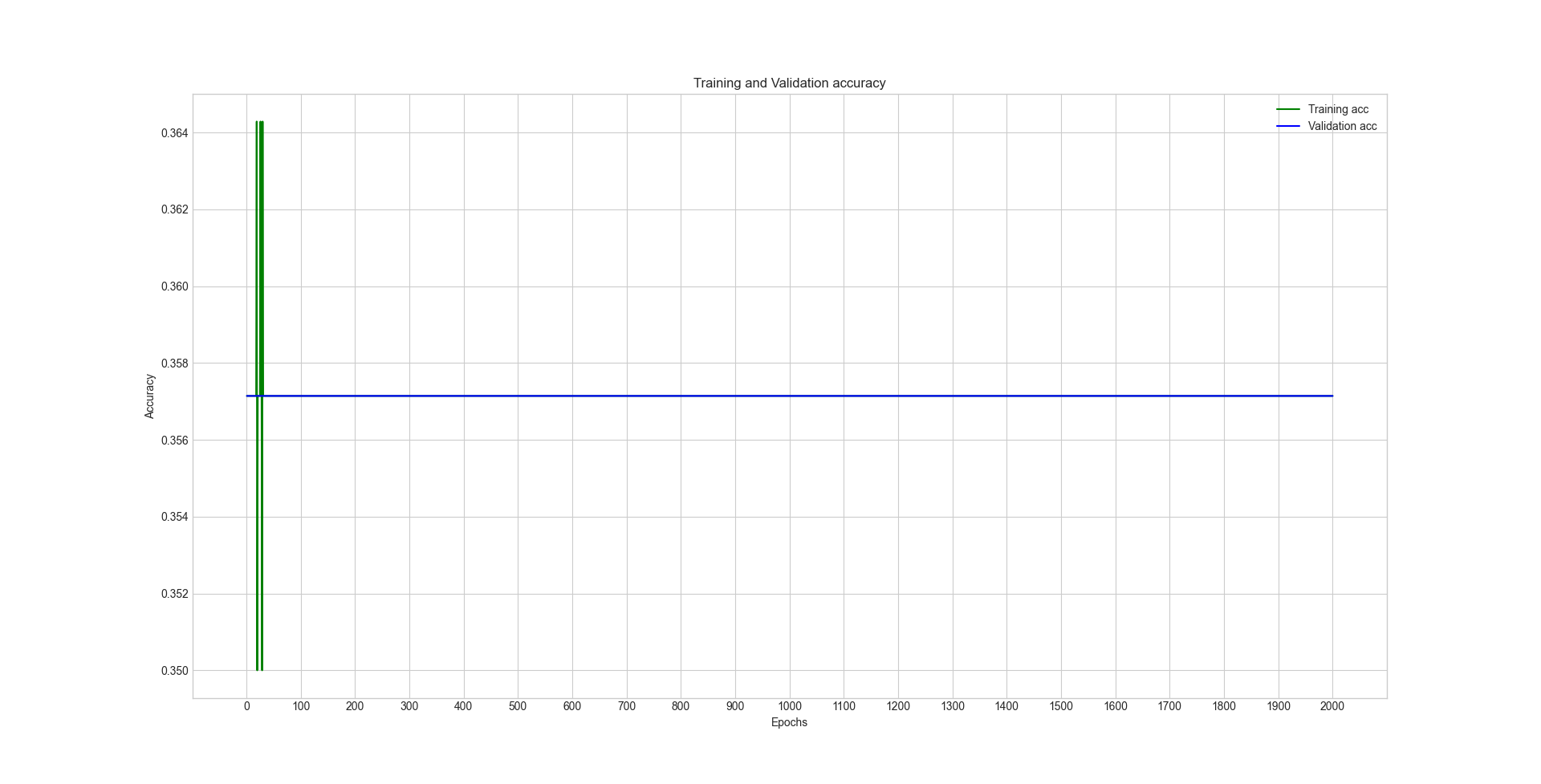
* Rozwiązywałem problem z inputem z tygodnia drugiego poprzez używanie typu tensor charakterystycznego dla biblioteki tensorflow, jednak to generowało kolejne błędy za każdym razem jak poprzedni rozwiązałem. Ostatecznie, po rozmowie z Grzesiem Ziółkowskim, udało mi się wprowadzić input pod postacią array’i biblioteki numpy i zacząłem testować sieć. Matryce, przed wprowadzeniem do sieci, przekształcałem z postaci listy list tupli RGB do postaci listy list zawierających 0 jeśli dana komórka była wolna i 1 jeśli była zajęta, odpowiedzi natomiast kodowałem jako wektor przesunięcia opadającej figury po osi OX.

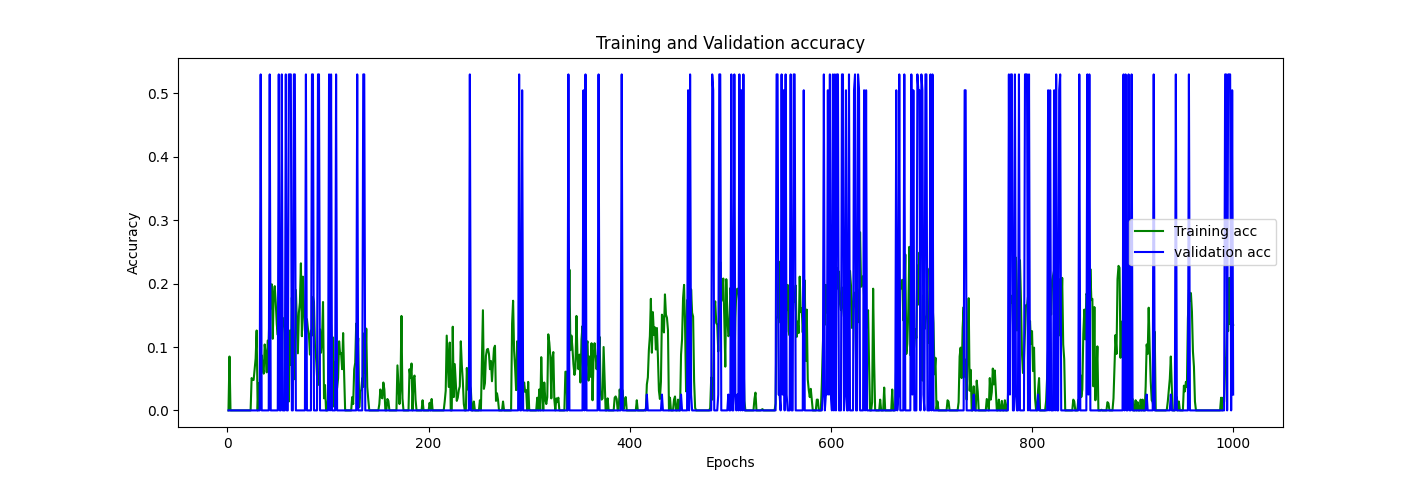


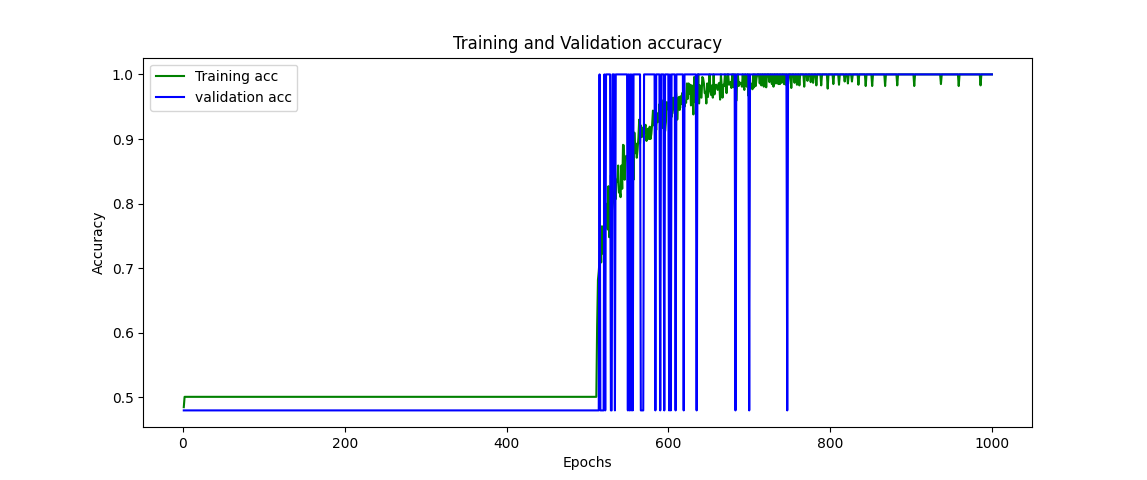
* Bawiłem się ustawieniami sieci w poszukiwaniu najlepszych parametrów
* Pogłębiałem wiedzę na temat funkcjonowania środowisk gymowskich i rozpisałem je teoretycznie
* Napotkałem na problemy w uczeniu sieci, mianowicie nie uczyła się na tyle dobrze, aby za każdym razem dochodziła ona do zadowalających wyników:





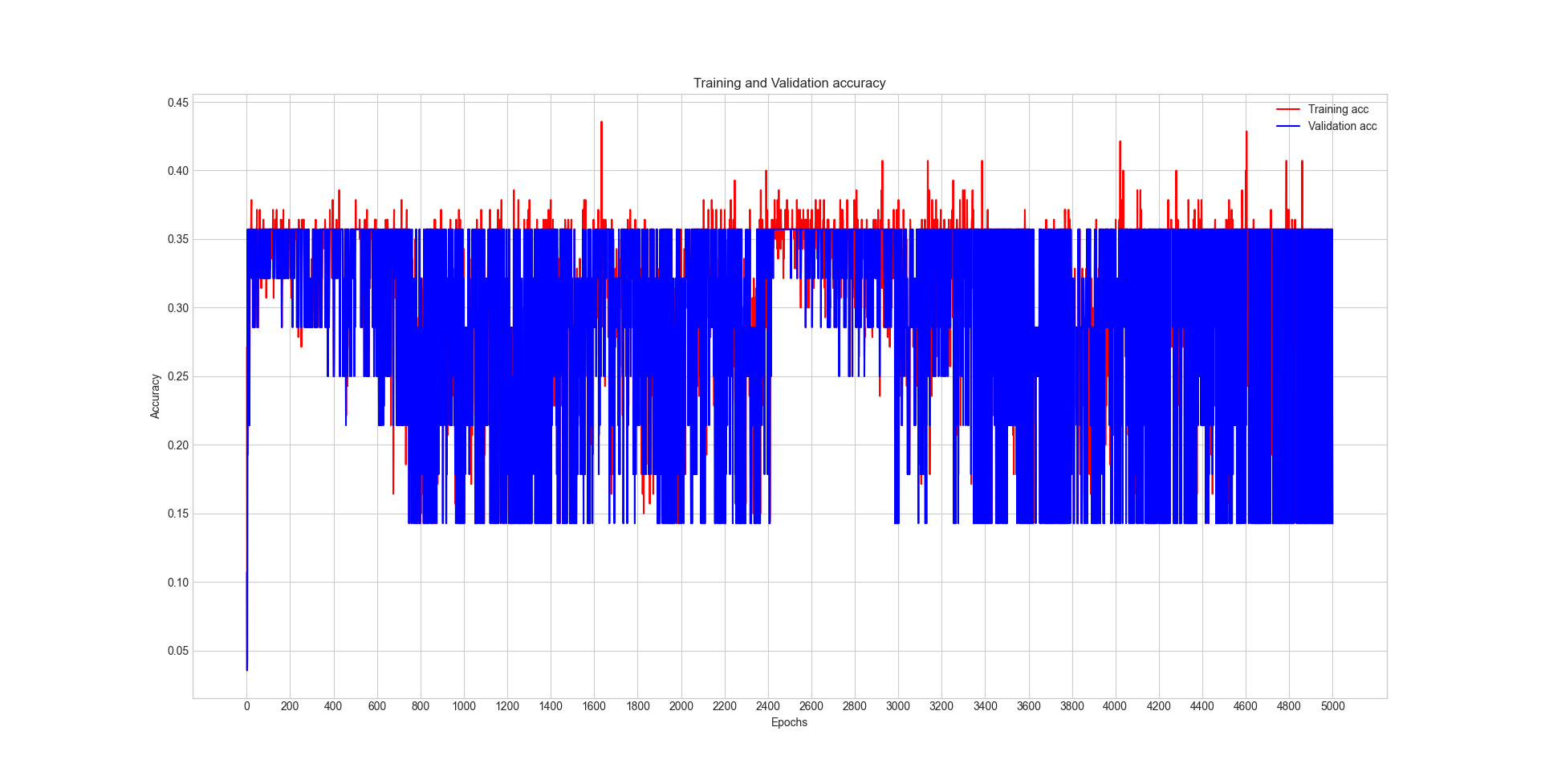


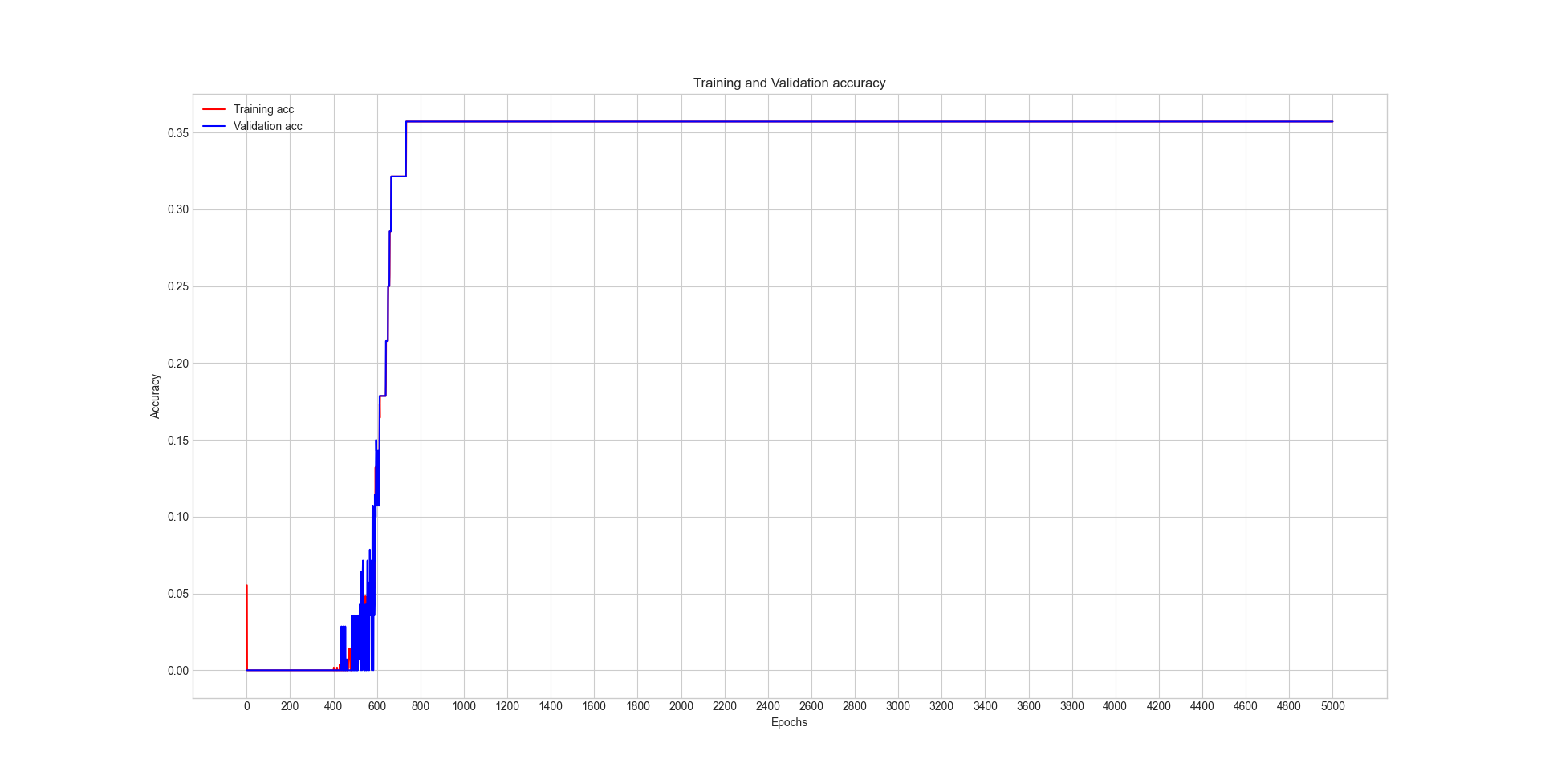


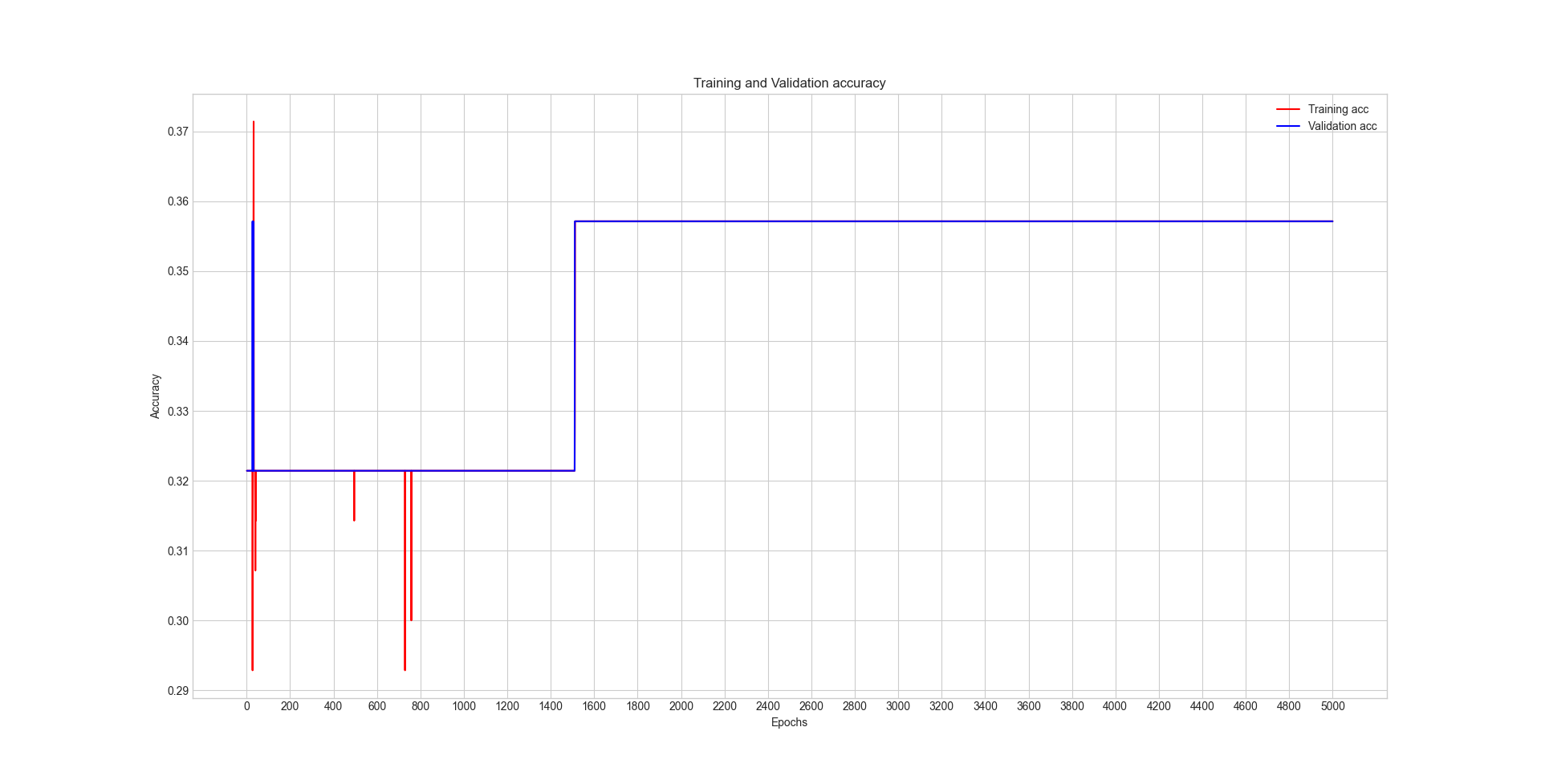


Tydzień 4 pracy:

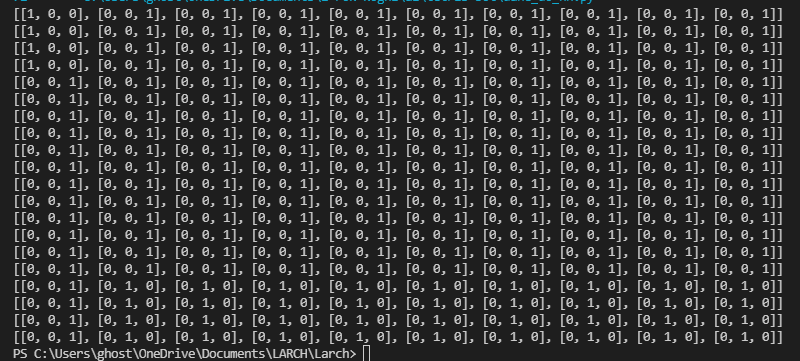
* Zgodnie z zaleceniem Doktora Gajdy bawiłem się architekturą sieci dodając warstwy o różnych liczbach neuronów. Do 10 warstw z liczbami neuronów odpowiadającymi kolejnym potęgom liczby 2 nie zauważyłem postępów rozwiązujących problem z dobrym uczeniem sieci, zatem postanowiłem spróbować z siecią konwolucyjną, co zalecił Doktor Gajda
* Jeśli z sieciami konwolucyjnymi nie wyjdzie to będę pracował nad wyciąganiem informacji kluczowych do ustawiania figury z matrycy i dawaniem tych informacji jako inputu do sieci
* Dodałem do sieci warstwy konwolucyjne jednak na razie nie widzę znacznych postępów

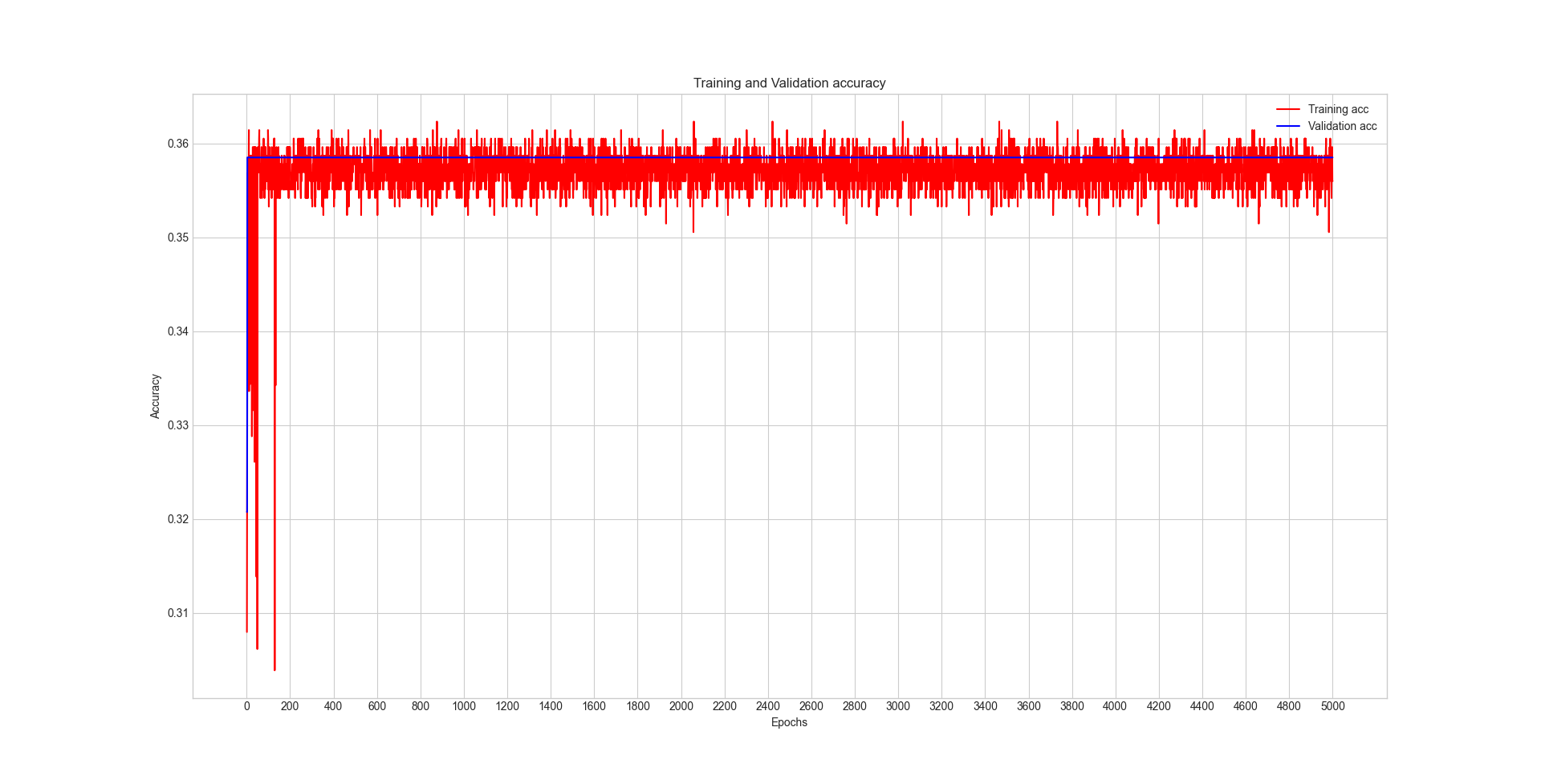


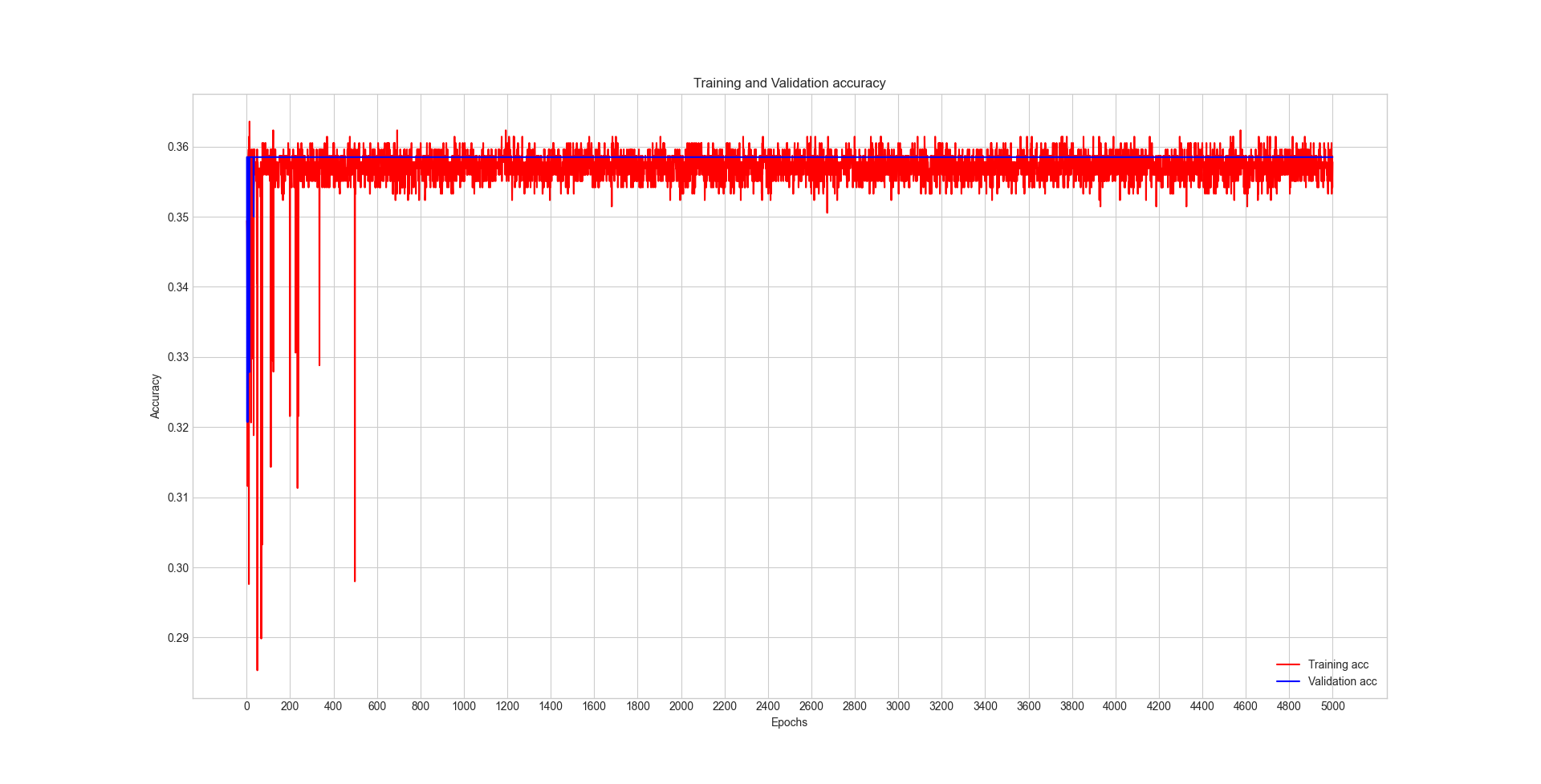




* Próbowałem zapisywać matryce w formie one-hot encoding – komórka matrycy miała postać listy boolowskiej [czy\_jest\_figurą, czy\_pozycja\_zajęta, czy\_pozycja\_wolna], a etykiety miały postać pojedynczej boolowskiej listy ruchu, który należy podjąć [lewo, nic, prawo]







Tydzień 5 pracy:

* Doktor Gajda miał pogorszone zdrowie przez co narada wojenna się nie odbyła
* Mimo okoliczności kontynuowałem pracę – przez brak efektów przy podejściu konwolucyjnym zacząłem ekstrahować informacje z matrycy i podawałem te informacje jako input zamiast matryc; odpowiedź dalej miała tą samą postać co po tygodniu 4.
* Zacząłem od zapisywania w array’u wysokości kolejnych kolumn planszy i na końcu dodawałem do tego współrzędne opadającej figury (kierunek na przyszłość – rozgałęziona sieć, jedna część analizuje wysokości kolumn, druga współrzędne figury i na końcu się łączą w jedną sieć, aby podjąć ostateczną decyzję)
* Próbowałem zapisywać różne informacje w różnych kombinacjach jednak niewiele to dało (może trzeba jeszcze raz przemyśleć jakie informacje mogą być najbardziej kluczowe?)
* Konsultowałem się z Bartkiem Swędrowskim i podsunął mi kilka pomysłów – część z nich sprawdziłem, ale dalej nic nie pomogło ustabilizować sposobu uczenia sieci
* Przez brak efektów poczułem się już lekko sfrustrowany, jednak się nie poddałem – zmieniłem koncepcję z supervised learningu na reinforcement learning i powróciłem do rozpoczętego w tygodniu 2. środowiska gymowskiego oraz zacząłem modyfikować kod samego Tetrisa tak, aby dobrze współpracował z agentami kerasa-rl (Bartek sugerował też użycie stable-baseline3 – zapamiętać)

Tydzień 6:

* Szedłem dalej w stronę RL, Tetrisa wstawiłem w klasę, ustawiłem zwracanie informacji potrzebnych agentowi
* Udało mi się uruchomić środowisko przy podejmowaniu decyzji losowych zarówno z renderowaniem rozgrywki jak i bez niego, co będzie bardzo przydatne przy dłuższych testach, ponieważ rozgrywka pochłania trochę czasu.
* Skonstruowałem model sieci oraz agenta, który „gra” w Tetris (na razie jednak dość słabo mu to wychodzi ;\_; ) i za podejmowane działania dostaje nagrody w zależności od tego jak jego działanie wpłynęło na środowisko
* Na razie jeszcze się nie uczy, jednak Doktor Gajda poradził, aby ustawić nagrody cząstkowe za mniejsze wyniki akcji – wtedy droga do wysokiej nagrody będzie bardziej subtelna – oraz spróbować zmniejszyć rozmiary figur w Tetrisie, lub ograniczyć liczbę różnych figur

Tydzień 7:

* Otrzymałem informację, że projekt został zakończony i teraz powinienem sporządzić sprawozdanie z przebiegu pracy i oto właśnie je piszę :D
* Odpoczywam też od programowania, ponieważ przez ostatnie dwa miesiące całkiem sporo czasu na nie poświęciłem

Podsumowanie:

* Mam wrażenie, że nauczyłem się bardzo wiele na temat działania i implementacji sztucznych sieci neuronowych
* Sądzę również, że polepszyłem trochę moje zdolności programistyczne

Plany na przyszłość:

* Mimo, że to już koniec projektu jako takiego planuję rozwijać rozpoczętą pracę
* Zamierzam testować zaznaczone pomysły i sugestie w wolnych chwilach, a jeśli się uda to będę zachwycony ^<^