Trening i ocena skuteczności przykładowej sieci neuronowej do gry w „Prezencik”

Leon Suliga, Mateusz Bogusławski

Celem niniejszego projektu jest stworzenie modelu sztucznej inteligencji, który będzie potrafił podejmować decyzje w grze "Prezencik" (znanej również jako "No Thanks!"), opartej na prostych zasadach, lecz z dużą głębią strategiczną.

1. Zasady gry „Prezencik”

- Cel gry

  Zdobyć jak najmniej punktów, w tej grze niski wynik jest lepszym wynikiem.

- Zawartość
  - Karty z numerami: od 3 do 35
    Przed rozpoczęciem gry losowo usuwa się 9 kart z talii.
  - Żetony, po 11 dla każdego gracza

- Przebieg rozgrywki
  - Na początku każdej rundy odkrywana jest nowa karta z talii.
  - Gracz na ruchu ma 2 opcje:
    - PASS (odmówić wzięcia karty, jeżeli posiada co najmniej jeden żeton): gracz kładzie 1 żeton na karcie, decyzje podejmuje kolejny w kolejności gracz.
    - TAKE (wziąć kartę): gracz bierze kartę oraz wszystkie żetony, które się na niej znajdują, zaczyna się nowa runda.

- Znaczenie żetonów
  - Umożliwiają odmowę wzięcia karty (PASS). Gracz bez żetonów może podjąć jedynie decyzję TAKE.

- Punktacja na koniec gry
  - Każda karta daje punkty równe swojej wartości. Przypadkiem szczególnym są ciągi kart o wartościach, których różnica między sąsiednimi kartami w ciągu wynosi 1, w tym przypadku pod uwagę brana jest karta o najmniejszej wartości w ciągu.
  - Każdy żeton posiadany w chwili zakończenia rozgrywki zmniejsza końcowy wynik o 1.

2. Generowanie zbioru danych wraz z etykietami do treningu sieci neuronowej oraz do jej oceny

Stany gry wykorzystywane do trenowania modelu generowane są losowo przez zaimplementowany generator, co pozwala na osiągniecię różnych, niezależnych od siebie stanów gry. Każdy z tych wygenerowanych stanów jest następnie oceniany przez eksperta, czyli programową funkcję heurystyczną, która na podstawie reguł decyduje, czy w danej sytuacji lepiej wziąć kartę, czy ją odrzucić. Decyzja ekspera reprezentowana jest przez liczbę zmiennoprzecinkową z przedziału [0;1], gdzie 0 oznacza silną decyzję TAKE, a 1 oznacza silną decyzję PASS, dzięki czemu możliwa jest późniejsza ocena decyzji modelu również z uwględnieniem tego, jak blisko porawnej decyzji była wytrenowana sieć neuronów. Ekspert może podjąc następujące decyzje:

- 0.0, gdy gracz nie posiada żetonów, jest zmuszony wziąć kartę
- 1.0, gdy karta pasuje do ciągu kart gracza, co pozwoli na osiągnięcie lepszego wyniku
- 1.0, gdy liczba żetonów na karcie jest większa od wartości karty
- 0.95, gdy liczba żetonów na karcie jest równa lub większa połowy wartości karty i obecnie gracz posiada mniej niż 10 żetonów
- 0.75, gdy liczba żetonów na karcie jest równa lub większa połowy wartości karty i obecnie gracz posiada nie mniej niż 10 żetonów
- 0.9, gdy któryś z przeciwników nie posiada żetonów i wzięcie przez niego karty będzie dla niego korzystne
- 0.75, gdy gracz posiada mniej niż 5 żetonów, a liczba żetonów na karcie jest niemniejsza niż 40% jej wartości
- 0.9, gdy gracz ma więcej niż 18 żetonów, i liczba żetonów na karcie wynosi co najmniej 60% jej wartości
- 0.9, gdy mój wynik jest o co najmniej 15 punktów gorszy od drugiego w kolejności gracza z najgorszym wynikiem i liczba żetonów na karcie wynosi co najmniej 30% jej wartości
- 0.2, gdy gracz ma przewagę co najmniej 20 punktów nad dowolnym z graczy i liczba żetonów na karcie wynosi mniej niż 40%
- 0.1, gdy liczba żetonów na karcie jest mnniejsza niż 20% jej wartości
- 0.3 – 0.7, w zależności od stosunku liczby żetonów do wartości karty

Model uczy się na podstawie tych danych, starając się naśladować eksperta i przewidywać podobne decyzje w nowych, wcześniej nieznanych sytuacjach.

3. Trening sieci neuronowej

Struktura sieci

- Warstwa wejściowa

  8 neuronów:

  1. Wartość aktualnej karty (znormalizowana)
  2. Liczba żetonów na karcie (znormalizowana)
  3. Liczba aktualnie posiadanych żetonów
  4. Liczba aktualnie posiadanych kart
  5. Czy co najmniej jedna obecenie posiadana karta jest o 1 mniejsza lub większa niż aktualna (0 lub 1)
  6. Czy przeciwnicy mają co najmniej jedną kartę, która jest o 1 mniejsza lub większa niż aktualna (0 lub 1)
  7. Czy co najmniej jeden przeciwnik nie posiada żetonów (0 lub 1)
  8. Maksymalna liczba żetonów u przeciwników

- Dwie warstwy ukryte

  32 neurony każda

- Warstwa wyjściowa

  1 neuron, który przyjmuje wartości z przedziału [0;1], gdzie:

  0 – silna decyzja TAKE

  1 – silna decyzja PASS

Działanie sieci

Proces uczenia sieci neuronowej składa się z dwóch głównych etapów, które powtarzają się dla każdego przykładu treningowego. W pierwszej fazie, tzw. forward pass, wektor wejściowy składający się z ośmiu cech (wartości zmiennoprzecinkowych) przechodzi przez kolejne warstwy ukryte sieci. Każdy neuron w warstwie oblicza wartość **z** według wzoru:

$$z=\sum_i w_i x_i+b$$

a następnie stosowana jest funkcja aktywacji ReLU, która przyjmuje wartość

$$ReLU(z)=max(0,z)$$

Na końcu, w warstwie wyjściowej (złożonej z jednego neuronu), wartości z ostatniej warstwy ukrytej są sumowane bez dodatkowej funkcji aktywacji, co daje końcową wartość przewidywaną (ciągłą).

Następnie w fazie obliczania straty porównuje się przewidywaną wartość z etykietą eksperta za pomocą błędu średniokwadratowego (MSE), zgodnie ze wzorem:

$$MSE = \frac{1}{n} \sum (y_{pred} - y_{true})^2.$$

Kolejnym krokiem jest propagacja wsteczna (backpropagation), w której obliczane są pochodne funkcji straty względem wag i biasów, a następnie aktualizowane są one za pomocą algorytmu optymalizacji **Adam**, który dostosowuje tempo uczenia (learning rate) dla każdej wagi osobno, na podstawie historii gradientów. Celem tych aktualizacji jest minimalizacja błędu predykcji. Proces ten powtarzany jest wielokrotnie (do 500 epok lub do momentu osiągnięcia zbieżności), co pozwala modelowi stopniowo dopasować się do danych treningowych.

4. Rezultaty

Z pomocą zaimplementowanego generatora pozyskano 1000 różnych stanów gry. Z których 800 zostało użyte do treningu sieci neuronowej, pozostałe 200 stanów posłużyło do oceny decyzji podejmowanych przez model.

174 decyzje z 200 pokrywało się z decyzjami TAKE/PASS eksperta.

Średni błąd kwadratowy wyniósł, w przypadku decyzji reprezentowanych przez liczby zmiennoprzecinkowe z przedziału [0;1] wyniósł 0.0195.

5. Wnioski

Przeprowadzony eksperyment pokazał, że prosta sieć neuronowa, oparta na danych wygenerowanych losowo i ocenianych przez eksperta heurystycznego, jest w stanie skutecznie nauczyć się podejmowania decyzji w grze „Prezencik”. Uzyskany średni błąd kwadratowy (MSE = 0.0195) wskazuje na wysoką zgodność modelu z ekspertem, a 87% trafności decyzji TAKE/PASS na zbiorze testowym potwierdza zdolność modelu do generalizacji i poprawnej interpretacji stanów gry, spoza zbioru treningowego. Mimo relatywnie prostej struktury, model dobrze odwzorowuje strategie eksperckie i może służyć jako skuteczny komponent decyzyjny w implementacji automatycznego gracza.

Wyniki pokazują, że podejście oparte na uczeniu nadzorowanym i heurystycznej etykiecie może być efektywnym rozwiązaniem w zadaniach związanych z grami strategicznymi o ograniczonej liczbie cech wejściowych.