



**Vilniaus
universitetas**

Informatikos ir informatinio mąstymo mokomoji veikla

Pažintis su dirbtiniu intelektu



Kuriame
Lietuvos ateitį
2014–2020 metų
Europos Sąjungos
fondų investicijų
veiksmų programa



**Vilniaus
universitetas**

Pažintis su dirbtiniu intelektu

Informatikos ir informatinio mąstymo mokomosios veiklos sukurtos įgyvendinant projektą „Aukštųjų mokyklų tinklo optimizavimas ir studijų kokybės gerinimas Šiaulių universitetą prijungiant prie Vilniaus universiteto“ (Nr. 09.3.1-ESFA-V-738-03-0001), finansuojamą iš Europos socialinio fondo lėšų pagal 2014–2020 metų Europos Sąjungos fondų investicijų veiksmų programos 9 prioriteto „Visuomenės švietimas ir žmogiškųjų išteklių potencialo didinimas“ įgyvendinimo priemonę Nr. 09.3.1-ESFA-V-738 „Aukštųjų mokyklų tinklo tobulinimas“.

Metodinė medžiaga „Pažintis su dirbtiniu intelektu“ skirta Mokyklos pedagogikos studijų programos moduliui „Informatikos didaktika“. Tikslinė grupė – būsimi informatikos pagrindinio ar vidurinio ugdymo mokytojai. Medžiaga siejama su Bendrojo ugdymo informatikos programa, ypač algoritmų ir programavimo pasiekimų sritimi. Atlikdami veikloje numatytas užduotis studentai susipažįsta su dirbtinio intelekto bendrosiomis sąvokomis, metodais, mokomosiomis veiklomis, kurios puikiai tinka integruotam ugdymui, pavyzdžiui, atliekant realaus gyvenimo užduotis ar konstruojant sudėtingesnius įrenginius. Pateikiamos ir aptariamos įdomios žaidybinimo veiklos.

Šioje informatikos ir informatinio mąstymo veikloje dalis medžiagos panaudota iš tarptautinio „Erasmus+“ „TrainDL“ projekto, kurio ištekliai sukurti taikant *Creative Commons* licenciją (<https://train-dl.eu/lt/apie-traindl/medziaga>).

Šios veiklos rengėjai: Vaida Masiulionytė-Dagienė ir prof. dr. Valentina Dagienė

Visoms iliustracijoms yra taikoma *Creative Commons Attribution-ShareAlike 4.0 International license* (CC BY-NC-SA 4.0)

Redagavo: Viktoras Dagys

Vilnius, 2023

Išvadas

Dirbtinis intelektas (kartais vartojama santrumpa DI, anglų kalba – *artificial intelligence*, santrumpa AI) – kompiuterių mokslo sritis, siekianti sukurti tokias sistemas, kurios galėtų atlikti užduotis, reikalaujančias žmogaus intelekto savybių – suvokimo, samprotavimo, mokymosi, sprendimų priėmimo, natūralios kalbos apdorojimo. Kitaip tariant, dirbtinis intelektas yra būdas išmokyti kompiuterines sistemas mokytis iš patirties, prisitaikyti prie naujų situacijų ir remiantis turimais duomenimis priimti sprendimus.

Šiuolaikiniame pasaulyje dirbtinis intelektas tampa vis svarbesniu, nes jis gali iš esmės pakeisti mūsų gyvenimo, darbo ir bendravimo būdus. Pavyzdžiui, dirbtinis intelektas jau naudojamas sveikatos priežiūroje, finansų, transporto, pramogų srityse, jį pasitelkus siekiama padidinti efektyvumą, tikslumą ir našumą.

Vienas iš įdomiausių dirbtinio intelekto aspektų yra jo gebėjimas mokytis ir tobulėti laikui bėgant. To pasiekama taikant mašininio mokymosi metodus, kai dirbtinio intelekto sistema mokoma naudojant didelius duomenų kiekius ir atrandant dėsningumus, taip pat darant prognozes remiantis šiais duomenimis. Mašininis mokymasis naudojamas įvairiose srityse – nuo vaizdų ir kalbos atpažinimo iki rekomendavimo programų ir sukčiavimo aptikimo.

Kitas svarbus dirbtinio intelekto aspektas yra natūralios kalbos apdorojimas (angl. *natural language processing*), kuris skirtas mašinų mokymuisi suprasti ir interpretuoti žmogaus kalbą. Tai reikalinga pokalbių robotams, virtualiesiems asistentams, kalbų vertimo sistemoms.

Dirbtinis intelektas atveria naujas galimybes robotikos, automatizavimo ir autonominių sistemų kūrimo srityse. Pavyzdžiui, savaeigiai automobiliai naudoja dirbtinį intelektą navigacijai keliuose ir priimti sprendimams, kaip reaguoti į besikeičiančias eismo sąlygas.

Tačiau, kaip ir bet kurios kitos naujos technologijos atveju, nerimą kelia dirbtinio intelekto poveikis visuomenei. Pavyzdžiui, vieni nerimauja, kad dėl dirbtinio intelekto gali sumažėti darbo vietų ir atsirasti ekonominė nelygybė, o kitiems nerimą kelia duomenų privatumo ir algoritmų neobjektyvumo klausimai. Apskritai dirbtinis intelektas yra sparčiai besivystanti sritis, galinti pakeisti daugelį mūsų gyvenimo aspektų.

Truputis istorijos

1996 metais dirbtinis intelektas pirmą kartą laimėjo prieš pasaulio šachmatų čempioną Garį Kasparovą. Tai buvo kompiuterinė sistema „Deep Blue“. Tačiau Garis Kasparovas kitus kelis mačus laimėjo arba sužaidė lygiosiomis. „Deep Blue“ kompiuterinės sistemos dirbtinis intelektas buvo tobulinamas ir 1997 metais kompiuteris žaidė puikiai, laimėjo visus mačus su Gariu Kasparovu.



1 pav. Šachmatų partija prieš dirbtinį intelektą

Vis tik pirmosios dirbtinio intelekto užuomazgos siekia daug senesnius laikus. Dirbtinio intelekto tyrinėjimų pradžia galima laikyti 1930–1940 metų laikotarpį. Dirbtinis intelektas domino ne tik technologinių sričių mokslininkus, bet ir psichologus, psichiatrus – jiems buvo įdomu tyrinėti, kaip galima sukurti dirbtines smegenis. Buvo parašyta įvairių kūrinių, kuriuose veikė robotai.

Vienas iš įkvėpimo šaltinių dirbtinio intelekto tyrimams buvo 1942 metais išspausdintas mokslinės fantastikos rašytojo Aizeko Azimovo (Isaac Asimov, 1920–1992) apsakymas „Karuselė“ (angl. „Runaround“)¹. Jame daug dėmesio skiriama robotų naudojimo ir jų bendravimo su žmonėmis etikos klausimams. Šio apsakymo apie inžinierių sukurtą robotą SPD-13 „Speedy“ siužetas paremtas trimis robotikos dėsniais:

1. Robotas negali sužeisti žmogaus arba savo neveiklumu leisti žmogui nukentėti.
2. Robotas turi paklusti žmogaus duodamiems įsakymams, išskyrus atvejus, kai tokie įsakymai prieštarauja pirmajam dėsniui.
3. Robotas turi apsaugoti savo egzistavimą tol, kol tokia apsauga neprieštarauja pirmajam ar antrajam dėsniui.

Dirbtinio intelekto tėvu laikomas Alanas Tiuringas (Alan Turing, 1912–1954). Pasaulyje žinomas Alano Tiuringo testas, remiantis kuriuo galima atskirti, ar bendraujama su žmogumi, ar su mašina (robotu). Šis principas naudojamas ir šiandien: kai norime prisiregistruoti ar prisijungti įvairiose interneto svetainėse ir turime patvirtinti, kad esame ne robotai, tenka suvesti skaičių ir raidžių derinį arba pažymėti atitinkamus paveikslėlius. Ši technologija vadinama CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*, lietuviškai: „Visiškai automatizuotas viešasis Tiuringo testas kompiuteriams nuo žmonių atskirti“).

Technologijų plėtros dėka tobulėja ir dirbtinis intelektas. Nuo 2010 iki 2019 metų publikacijų apie dirbtinį intelektą skaičius išaugo penkis kartus iki maždaug 20 tūkstančių per metus.

¹ Šį apsakymą galima perskaityti ir į lietuvių kalbą antrą sykį išverstoje A. Azimovo knygoje „Aš, robotas“, kurią 2022 metais išleido leidykla „Kitos knygos“. Tiesa, ten trys robotikos dėsniai vadinami Trimis fundamentaliosiomis robototechnikos taisyklėmis.

2017 m. „Microsoft“ korporacija pateikė informaciją, kad jos kalbos atpažinimo sistema pasiekė tokį lygį, jog lieka 5,1 proc. žodžių klaidų, o tai jau atitinka žmogaus našumą transkribuojant skambučių pokalbius į tekstą anglų kalba. Dirbtinio intelekto algoritmai dabar priylgsta gydytojų gebėjimams ar net yra pranašesni už juos nustatant daugelį ligų, ypač kai diagnozės nustatymas grindžiamas vaizdų analize (Alzheimerio ligos, vėžio metastazių, odos, akių ir kitų ligų atveju).

2022 metų pabaigoje išleista didžiulio populiarumo sulaukusi „ChatGPT“ pokalbių roboto paslauga. Nors pagrindinė pokalbių roboto funkcija – imituoti žmogaus pokalbį, „ChatGPT“ yra universalus. Pavyzdžiui, jis gali rašyti ir derinti kompiuterines programas, kurti muziką, televizijos spektaklius, pasakas ir mokinių rašinius, atsakinėti į testų klausimus, rašyti poeziją ir dainų tekstus, emuliuoti „Linux“ sistemą, imituoti visą pokalbių kambarį, žaisti žaidimus, imituoti bankomatą. „ChatGPT“ turi ir tam tikrų trūkumų. Jį sukūrusi laboratorija „OpenAI“ pripažįsta, kad šis robotas „kartais rašo tikėtinais skambančius, bet neteisingus ar beprasmiškus atsakymus“. Kol kas robotas turi nedaug žinių apie naujausius pastarųjų metų įvykius, atsakymai į užklausas, apimančias žmonių apibūdinimą kartais pasižymi šališkumu ir panašiai.

Mašininis mokymasis ir jo tipai

Mašininis mokymasis – tai dirbtinio intelekto tipas, leidžiantis kompiuterių programoms mokytis iš duomenų ir tobulinti savąjį užduočių atlikimą be jokio specialaus užprogramavimo. Tai tarsi mokymas kompiuterį mokytis iš pavyzdžių, o ne konkrečių instrukcijų teikimas atsižvelgiant į kiekvieną situaciją.

Pagal tai, ar mašininio mokymosi sistemos mokosi pačios, ar joms mokymosi procese reikia žmogaus pagalbos, jos skirstomos į keletą tipų. Pagrindiniai mašininio mokymosi tipai yra šie:

- Mokymasis su mokytoju, kitaip – prižiūrimasis mokymasis (angl. supervised learning);
- Mokymasis be mokytojo, kitaip – neprižiūrimasis mokymasis (angl. unsupervised learning);
- Sustiprintasis mokymasis (angl. reinforcement learning).

Prižiūrimasis mokymasis (angl. *supervised learning*) – tai kompiuterinių sistemų mokymas atpažinti įvairius duomenų šablonus. Norėdami tai padaryti, turime pateikti kompiuteriui pavyzdžių su žymėmis, t. y., pateikti jam dalykų, kuriuos norime, kad jis atpažintų, pavyzdžių kartu su žyme, kuri kompiuteriui nurodo, kas tai yra.

Tarkime, norime išmokyti kompiuterį atpažinti šunų nuotraukas. Pradėsime nuo to, kad surinksime šunų paveikslėlių rinkinį ir kiekvieną paveikslėlį pažymėsime etikete „šuo“. Taip pat galėtume surinkti kitų paveikslėlių, pavyzdžiui, kačių, paukščių ir medžių, ir kiekvieną paveikslėlį pažymėti atitinkama kategorija.

Tada šiuos paženklintus duomenis perduotume mašininio mokymosi algoritmui. Algoritmas – tai instrukcijų rinkinys, nurodantis kompiuteriui, kaip analizuoti duomenis ir nustatyti dėsningumus. Mašininio mokymosi su mokytoju atveju algoritmas pradeda analizuoti

paženklintus pavyzdžius ir mokosi susieti kiekvieno pavyzdžio požymius su atitinkama etikete.

Pavyzdžiui, algoritmas gali išmokti, kad šunys paprastai turi kailį, keturias kojas ir uodegą, o katės – kailį, keturias kojas ir smailias ausis. Analizuodamas daug paženklintų pavyzdžių, algoritmas gali pradėti atpažinti savybes, pagal kurias viena kategorija skiriasi nuo kitos.

Kai algoritmas išmoks atpažinti žymetuose duomenyse esančius dėsningumus, galime patikrinti jo tikslumą pateikdami klasifikuoti naujus, nežymėtus pavyzdžius. Pavyzdžiui, algoritmui galime pateikti anksčiau nematytą šuns nuotrauką ir paprašyti, kad jis ją klasifikuotų kaip „šuo“. Algoritmas, naudodamasis išmoktais požymiais, atpažintų paveikslėlį kaip šunį ir priskirtų paveikslėliui etiketę „šuo“.



2 pav. Vaizdų atpažinimo pavyzdys

Šiai nežinomo autoriaus [nuotraukai](#) suteikta [CC BY-SA](#) licencija

Jei algoritmas suklysta, galime suteikti jam grįžtamąjį ryšį ir panaudoti jį tikslumui pagerinti. Pavyzdžiui, jei algoritmas klaidingai pavadina katės paveikslėlį šunimi, galėtume pataisyti etiketę ir panaudoti šią pataisą mokydami algoritmą ateityje tiksliau atpažinti kačių ir šunų skirtumus.

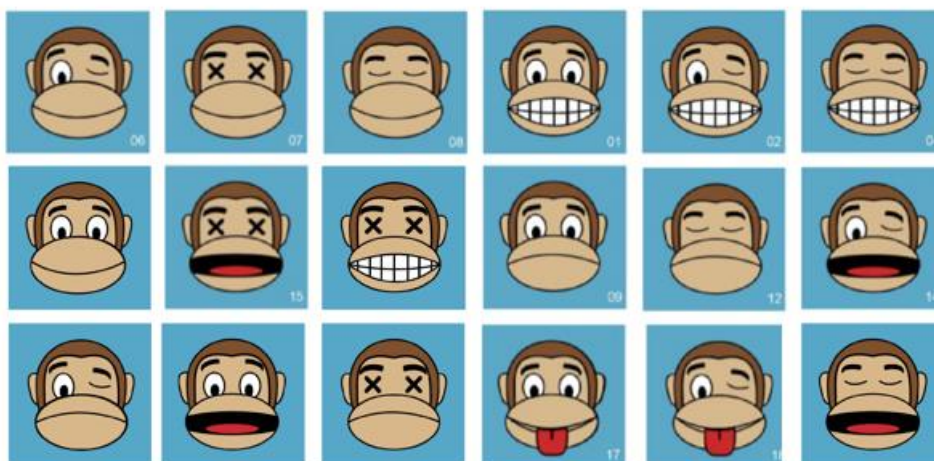
Prižiūrimo mokymosi algoritmas yra galinga priemonė, leidžianti išmokyti kompiuterius atpažinti duomenų modelius ir tiksliai prognozuoti naujus, neženklintus duomenis. Pateikdami pažymėtus pavyzdžius ir atsiliepimus, galime padėti mašininio mokymosi algoritmams pagerinti jų tikslumą ir laikui bėgant dar geriau atpažinti modelius.

1 praktinė užduotis.

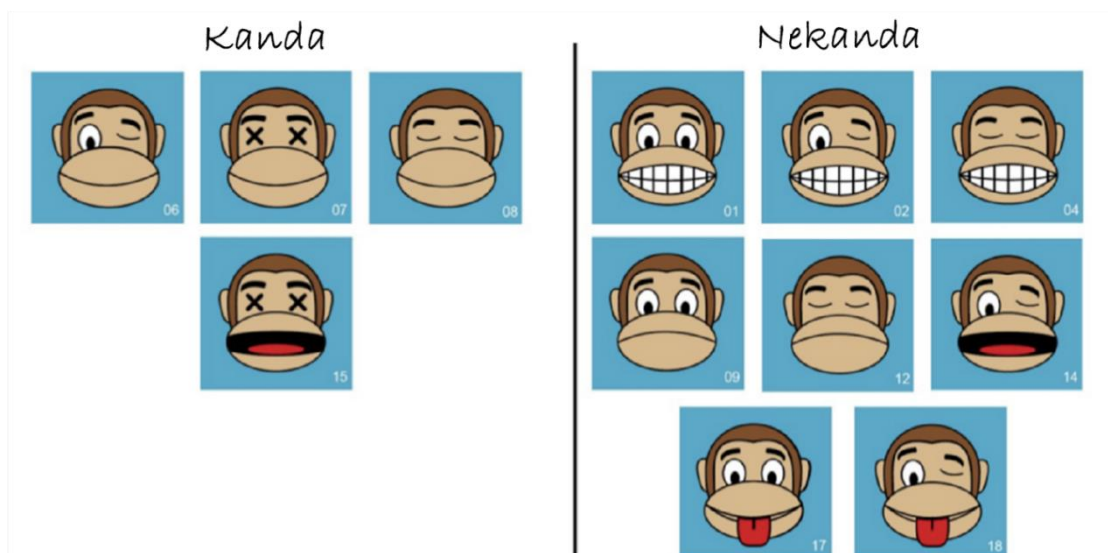
Žaidimas „geroji beždžionė, blogoji beždžionė“

Šioje praktinėje užduotyje pateikiamas mokymosi su mokytoju arba prižiūrimo mokymosi pavyzdys. Pradiniai duomenys užduotyje yra suskirstomi į grupes pagal tam tikrus požymius, tokiu būdu mes duomenims priskiriame etiketes.

Zoologijos sode gyvena įvairios beždžionės, vienos jų geros, mėgsta bendrauti su žmonėmis, kitos – piktesnės, kandžiojasi.

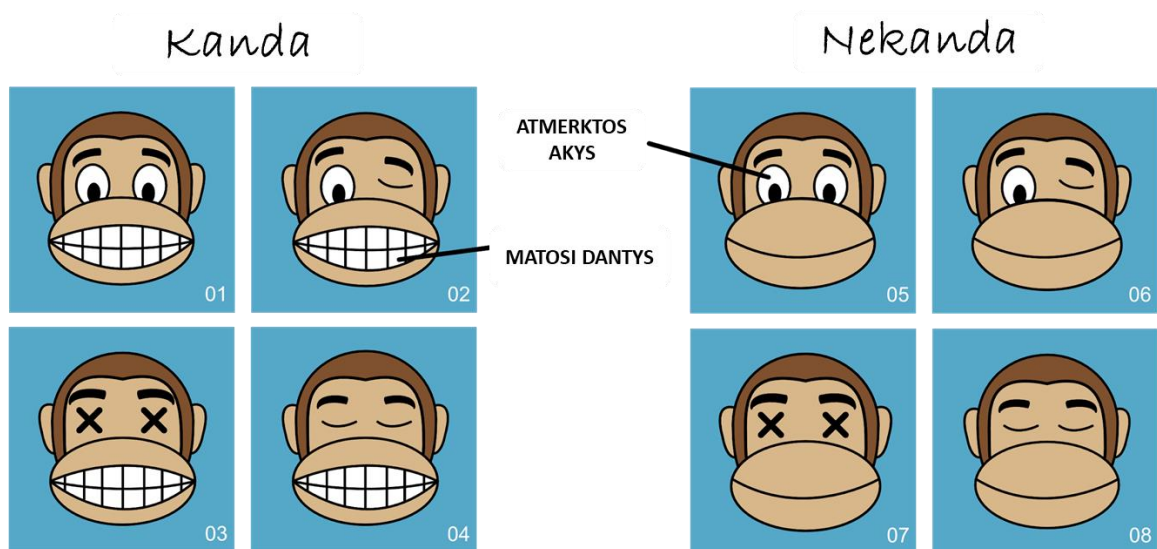


Beždžionės suskirstysime į dvi grupes: vienos kandžiojasi, kitos – ne. Į šias grupes beždžionės buvo suskirstytos pagal įvairius požymius – akis ir burną. Priskyrimas grupei reiškia tam tikros etiketės uždėjimą.



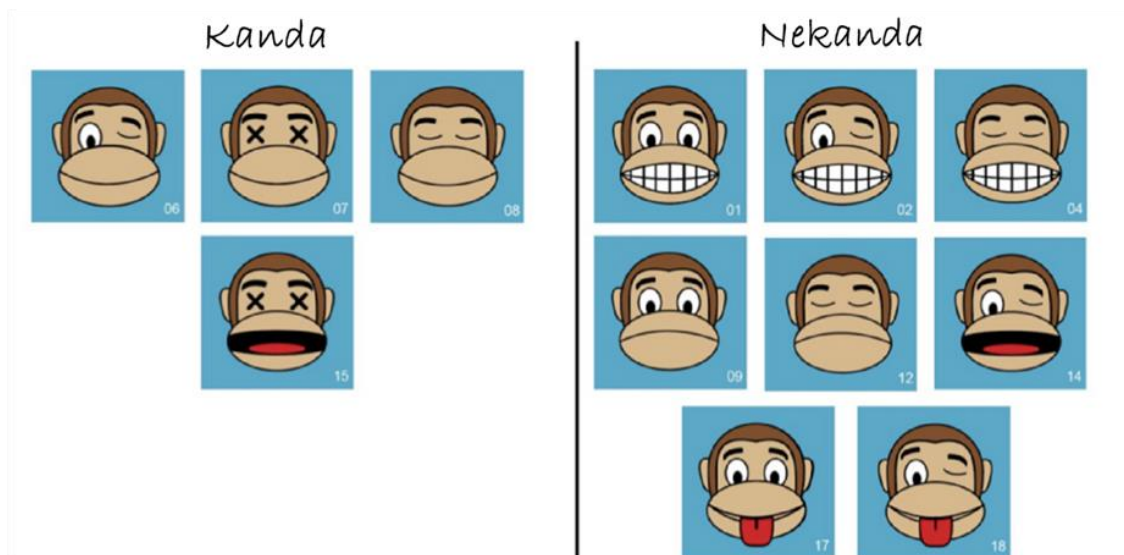
Priskiriant beždžionę kuriai nors grupei kai kurie požymiai gali būti stipresni, kiti – silpnesni. Pavyzdžiui, matome, kad kandančių beždžionių grupėje yra beždžionės, kurių akyse yra „X“, o prie nekandančių tokių nėra, taigi galime teigti, kad šis požymis mums jau nusako, kad beždžionė kanda.

Panagrinėkime pavyzdį su kitaip paskirtomis beždžionių grupėmis, atkreipdami dėmesį į požymių stiprumą:

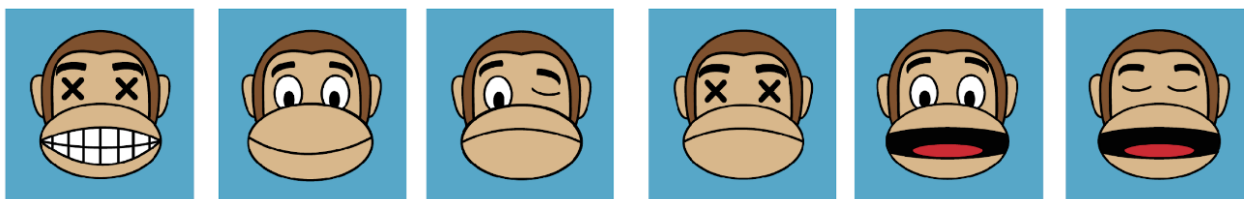


Iš šio pavyzdžio matome, kad jei beždžionė turi apnuogintus dantis, tai ji kanda. Kadangi visų kandančių beždžionių dantys matosi, o nekandančių nesimato, tai turime gan stiprų požymį, pagal kurį galime nuspręsti, kaip elgiasi beždžionė. Pažiūrėkime ir į akis: tai dar vienas požymis, kurį turi beždžionės ir kuris skiriasi priklausomai nuo to, ar beždžionė kandžiojasi, ar ne. Tačiau iš pateikto pavyzdžio matome, kad šioje situacijoje remtis akimis negalime, nes tiek kandančių, tiek nekandančių beždžionių yra tie patys akių požymiai, pavyzdžiui, atmerktos akys.

Grįžkime prie pirmo pavyzdžio, kuriame beždžionės pagal kitus požymius suskirstytos į grupes (kanda arba nekanda). Pabandykite nustatyti požymius, pagal kuriuos turėsite identifikuoti, ar beždžionė kanda, ar ne.

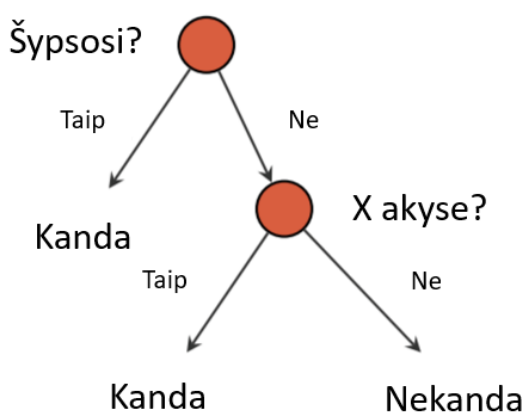


O dabar nustatykite, kuri iš čia pateiktų beždžionių kanda, o kuri ne:



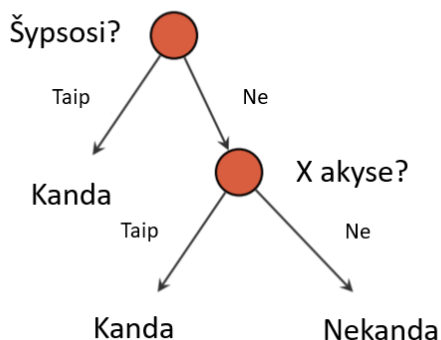
Atsakymai su paaiškinimais – užduoties pabaigoje, skaitykite atidžiai.

Vienas iš dirbtinio intelekto naudojamų algoritmų yra sprendimo (kartais vadinamo pasirinkimu) medžiai (angl. *decission tree*). Apibrėžiamos taisyklės ir tuomet, remiantis medžio struktūra, aprašoma, kaip kas atitinka taisyklę arba ne. Paveikslėlyje pateikiama, kaip tai atrodo:

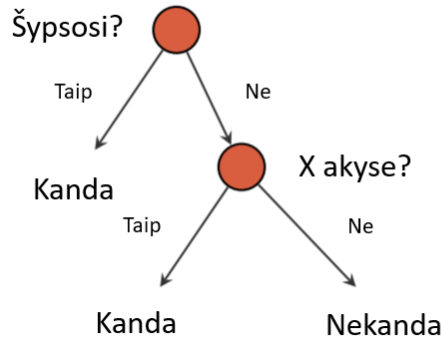


Panagrinėkime pavyzdį. Tarkime, turime tokį krokodiliuką:

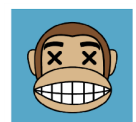
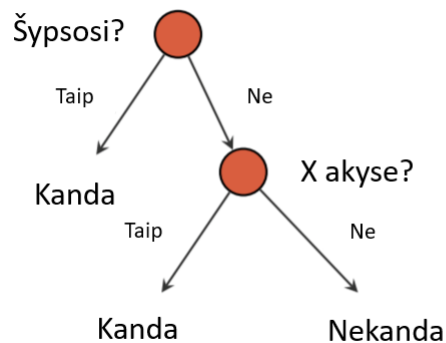
Jam pritaikome tas pačias taisykles, kurias taikėme beždžionėms. Atsižvelgiame į du požymius: akis ir burną.



Pirmąją taisyklę tikrinama, ar šypsosi, t. y., ar plačiai atverta burna (kaip beždžionių atveju), – matome, kad netinka. Todėl keliaujame medžiu žemyn (rodykle „Ne“) iki kitos taisyklės:

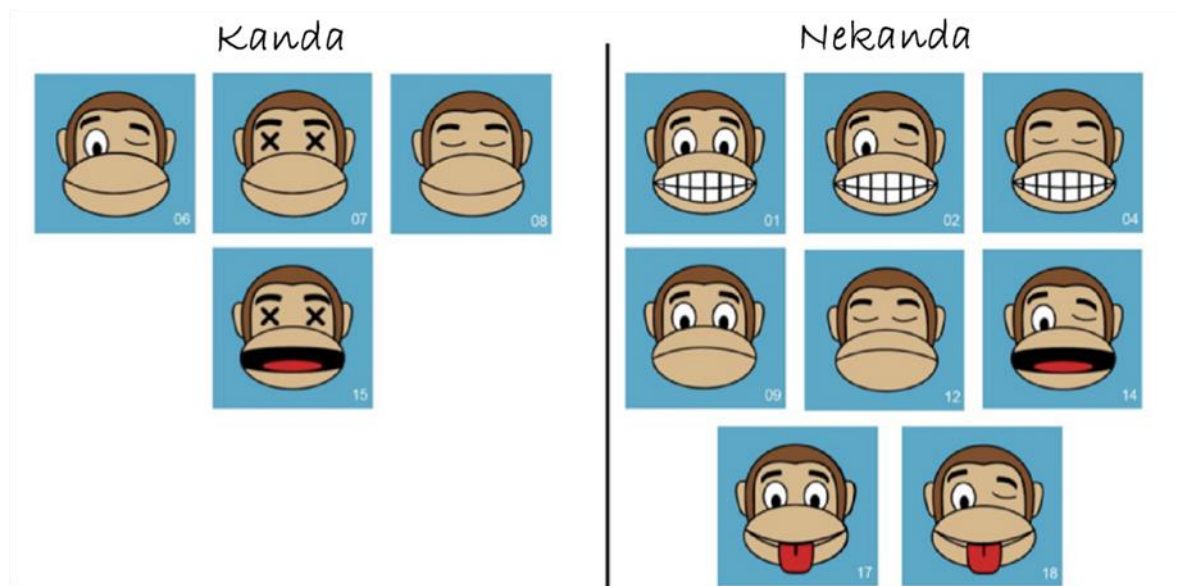


Tollesnė taisyklė klausia, ar akyse pažymėti kryžiuukai. Matome, kad irgi ne, akys yra atmerktos, taigi vėl keliaujame „Ne“ rodykle žemyn ir gauname atsakymą į klausimą, kad krokodilas nekanda:

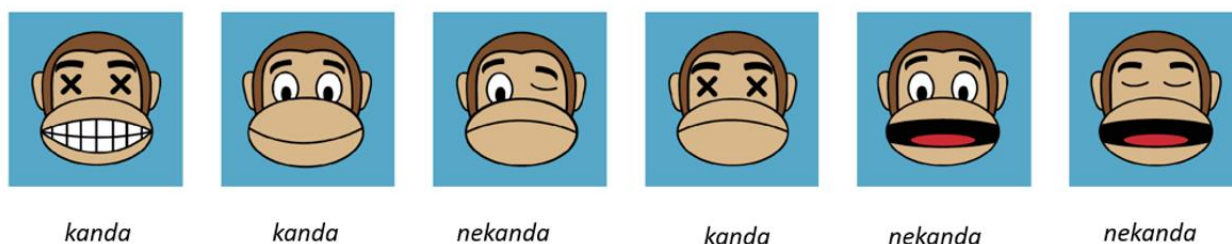


Dabar panagrinėkime, kaip sektųsi nustatyti, kanda ar nekanda ši beždžionė:

Iš jau nagrinėto pavyzdžio (žr. pakartotą paveikslą) matome, kad kryžiuokus akyse turi dvi kandančios beždžionės iš keturių, tai yra 50 proc., o šypsosi trys beždžionės iš aštuonių nekandančių beždžionių. Taigi turime 37,5 proc. Šiuo atveju matome, kad turime tvirtesnį požymį, daugiau procentų, kad beždžionė kanda.



Tokiu pačiu principu pasitikrinkite savo atsakymus nagrinėdami ir kitas beždžiones. Pateiktame paveiksle prie kiekvienos beždžionės parašyta, ar ji kanda, ar ne:



Pagal teisingų atsakymų skaičių galima nustatyti dirbtiniu intelektu grindžiamos sistemos tikslumą:

$$\frac{\text{Teisingi atvejai}}{\text{visi atvejai}} = \boxed{} \quad \text{Sistemos tikslumas}$$

Neprižiūrimasis mokymasis arba mokymasis be mokytojo (angl. *unsupervised learning*) – tai dar viena mašininio mokymosi rūšis, kurią taikant randami dėsningumai duomenyse neduodant pažymėtų pavyzdžių. (Prižiūrimojo mokymosi atveju kompiuteriui pateikiamas duomenų rinkinys ir jo prašoma pačiam nustatyti dėsningumus ir ryšius.)

Įsivaizduokite neprižiūrimąjį mokymąsi kaip naujo miesto tyrinėjimą be žemėlapiu ar gido. Galite klaidžioti, ieškodami įdomių lankytinų objektų, pastatų ar rajonų, neturėdami jokio išankstinio supratimo apie tai, ką rasite. Panašiai ir neprižiūrimo mokymosi atveju kompiuteris tyrinėja duomenis, ieškodamas įdomių dėsningumų ar sąsajų be jokių išankstinių etikečių ar kategorijų.

Vienas iš labiausiai paplitusių neprižiūravimo mokymosi tipų yra **klasterizavimas**. Klasterizavimas apima duomenų elementų rinkinio suskirstymą į grupes pagal jų panašumus ar skirtumus. Pavyzdžiui, įsivaizduokite, kad turite duomenų rinkinį apie įvairių rūšių vaisius, bet neturite jokių etikečių, nurodančių, kuris vaisius yra kuris. Galite naudoti klasterizavimą ir sugrupuoti duomenis į skirtingus klasterius pagal vaisių savybių, pavyzdžiui, spalvos, formos ar dydžio, panašumus.

Naujienose vyraujančių temų nustatymas apima pagrindinių temų ar motyvų nustatymą dokumentų rinkinyje. Pavyzdžiui, įsivaizduokite, kad turite naujienų straipsnių rinkinį ir norite nustatyti pagrindines aptariamas temas. Galėtumėte naudoti neprižiūrimą mokymąsi, kad sugrupuotumėte duomenis į skirtingas grupes pagal žodžių dažnumą ir panašumą, o tada nustatyti, kurie žodžiai linkę pasirodyti kartu kiekviename klasteryje ir išvesti pagrindines temas.

Prekių krepšelio analizė apima vartotojų elgsenos modelių nustatymą analizuojant kartu perkamas prekes. Pavyzdžiui, įsivaizduokite, kad turite duomenų apie klientų pirkimus maisto prekių parduotuvėje ir norite nustatyti, kurios prekės dažnai perkamos kartu. Galite naudoti neprižiūrimą mokymąsi, kad sugrupuotumėte duomenis į skirtingas grupes pagal prekių panašumą, ir tada nustatyti, kurios prekės dažniausiai perkamos kartu kiekvienoje grupėje.

Neprižiūrimas mokymasis gali būti veiksminga priemonė naujoms įžvalgoms ir ryšiams duomenyse atrasti, ypač kai duomenys yra sudėtingi ir sunkiai suprantami. Tačiau, kadangi neprižiūrimas mokymasis nesiremia pažymėtais pavyzdžiais, gali būti sudėtingiau įvertinti rezultatų tikslumą. Todėl neprižiūrimas mokymasis dažnai naudojamas kartu su prižiūrimu mokymusi, kad būtų galima gauti geriausius abiejų pusių rezultatus: gebėjimą savarankiškai rasti dėsningumus ir gebėjimą mokytis iš pažymėtų pavyzdžių.

2 praktinė užduotis

Šioje užduotyje pateikiamas neprižiūravimo mokymosi pavyzdys, tiksliau neprižiūravimo mokymosi suvokimo principas per pačių mokinių patirtį.

Užduočiai reikės: 20–30 įvairių dydžių, formų ir spalvų sagų (arba kitų objektų, kurie turi kelis požymius), popieriaus lapo arba lentos ir rašiklių.

Instrukcija

Iš pradžių paaiškinkite mokiniams neprižiūravimo mokymosi sąvoką, remdamiesi anksčiau aprašytais pavyzdžiais ir samprotavimais.

Pristatykite sagų rūšiavimo užduotį.

Paaiškinkite, kad kiekvienam mokiniui duosite po saują sagų ir paprašysite jų suskirstyti sagas į grupes pagal panašumus ar skirtumus. Tačiau jiems nesakykite, kaip rūšiuoti sagas – tai jie turės patys išsiaiškinti.



Išdalykite mokiniams sagas, pasirinkdami, kad kiekvienas mokinys turėtų panašų skirtingų dydžių, formų ir spalvų derinį.

Leiskite mokiniams kelias minutes rūšiuoti sagas į grupes. Paskatinkite juos aptarti savo minčių procesą ir samprotavimus tarpusavyje.

Kai mokiniai surūšiuos sagas, paprašykite jų paaiškinti, kaip jie nusprendė sugrupuoti sagas. Ar jie grupavo jas pagal spalvą, formą, dydį ar kitą požymį? Ar skirtingi mokiniai skirtingai grupavo sagas?

Nubraižykite lentoje ar popieriuje lentelę ar schemą, kurioje būtų pavaizduotos skirtingos sagų grupės. Paprašykite mokinių, kad jie padėtų jums paženklinti kiekvieną grupę pagal jų nustatytus panašumus.

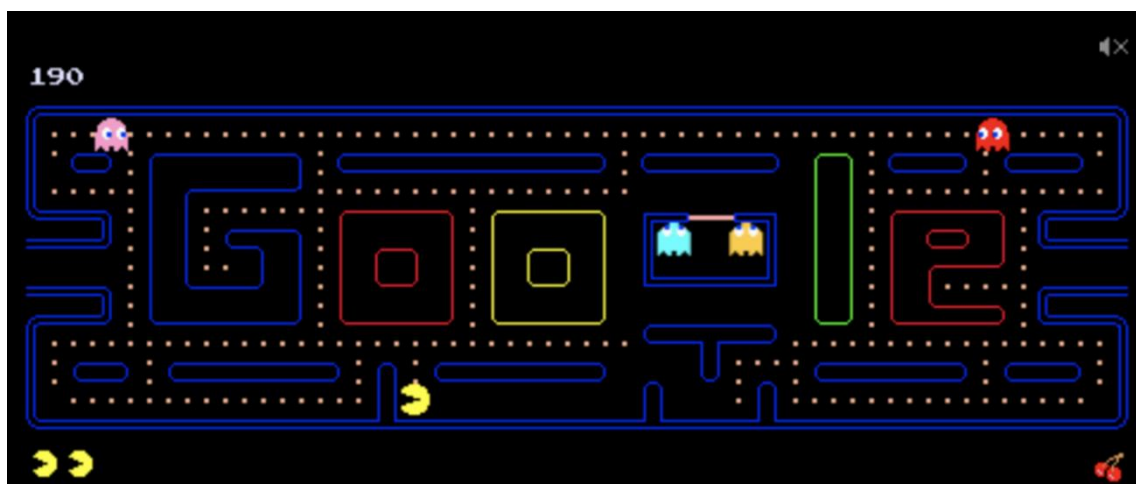
Galiausiai aptarkite su mokiniams, kaip ši veikla susijusi su neprižiūrimu mokymusi. Paaiškinkite, kad sagos simbolizuoja neapdorotus duomenis, o mokinių rūšiavimas – neprižiūrimo mokymosi algoritmą, bandantį nustatyti duomenų dėsningumus ir ryšius. Kaip skirtingi mokiniai gali skirtingai rūšiuoti sagas, taip ir skirtingi neprižiūrimo mokymosi algoritmai gali nustatyti skirtingus duomenų dėsningumus. Tačiau tikslas visada yra rasti prasmingas duomenų grupes ar ryšius, nenaudojant išankstinių etikečių ar kategorijų.

Sustiprintasis mokymasis (angl. *reinforced learning*) – tai mašininio mokymosi rūšis, kai robotas sąveikauja su aplinka ir mokosi iš grįžtamojo ryšio – apdovanojimų ir bausmių. Roboto tikslas – išmokyti veiksmų seką, kuri laikui bėgant maksimizuotų jo atlygį.

Sustiprintojo mokymosi atveju robotas gauna grįžtamąjį ryšį apdovanojimų arba bausmių pavidalu, priklausomai nuo jo veiksmų. Apdovanojimai arba bausmės paprastai apibrėžiami atlygio funkcija, kuri nurodo atlygį, kurį agentas gauna už kiekvieną veiksmą, kurį jis atlieka tam tikroje būsenoje.

Agento tikslas – išmokyti taisykles, t. y., taisyklių rinkinį, nustatantį, kokį veiksmą atlikti kiekvienoje būsenoje, kad laikui bėgant jo bendras atlygis būtų kuo didesnis. Kad tai pasiektų, agentas bandymų ir klaidų metodu mokosi, už kuriuos veiksmus gauna didžiausią atlygį.

Štai paprastas pavyzdys, iliustruojantis, kaip veikia sustiprintas mokymasis.



3 pav. Sustiprinto mokymosi pavyzdys

Įsivaizduokite robotą, kuriam pavesta įveikti labirintą ir pasiekti tikslą. Robotas pradeda nuo įėjimo į labirintą ir turi rasti kelią iki tikslo neatsitrenkdamas į jokią sieną. Kiekvienoje labirinto sankryžoje robotas gali pasirinkti, ar pasukti į kairę, į dešinę, ar eiti tiesiai. Jei robotas pasiekia tikslą, jis gauna +10 atlygį. Jei jis atsitrenkia į sieną, gauna nuobaudą -1. Roboto tikslas – rasti greičiausią kelią iki tikslo neatsitrenkus į jokią sieną.

Iš pradžių robotas savo veiksmus gali pasirinkti atsitiktinai. Tačiau tyrinėdamas labirintą jis pradės mokytis, už kokius veiksmus gaunamas didžiausias atlygis. Pavyzdžiui, jei robotas sankryžoje pasuks į kairę ir pasieks tikslą, jis išmoks, kad pasukti į kairę yra gera strategija. Kita vertus, jei robotas pasuks į dešinę ir atsitrenks į sieną, jis sužinos, kad pasukti į dešinę yra bloga strategija.

Laikui bėgant robotas patobulins savo strategiją ir pradės priimti protingesnius sprendimus, kokius veiksmus atlikti kiekvienoje sankryžoje. Galiausiai jis gali rasti optimalų kelią į tikslą ir gauti didžiausią įmanomą atlygį.

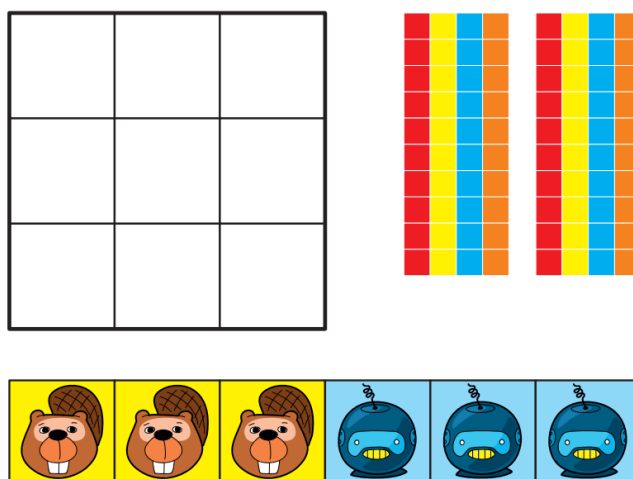
Sustiprintasis mokymasis naudojamas įvairiose srityse – nuo robotikos ir žaidimų iki finansų ir sveikatos priežiūros. Tai veiksmingas metodas, leidžiantis robotams mokytis iš patirties ir priimti protingus sprendimus sudėtingoje ir dinamiškoje aplinkoje.

3 praktinė užduotis

Žaidimas „Robotas ir bebras“

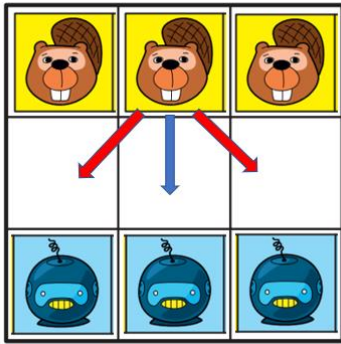
Ši užduotis reikalauja šiek tiek kantrybės. Tačiau sužaidus šį žaidimą keletą kartų, imama suprasti, kaip veikia sustiprintasis mašininis mokymasis.

Reikia 3×3 langelių žaidimo lentos ir po tris bebrų ir robotų figūrėles:



(Šio dokumento pabaigoje pateikiami žaidimo lapai.)

Išsikerpame figūrėles ir išdėliojame, tris prieš tris, kaip šaškėse:



Žaidimo taisyklės

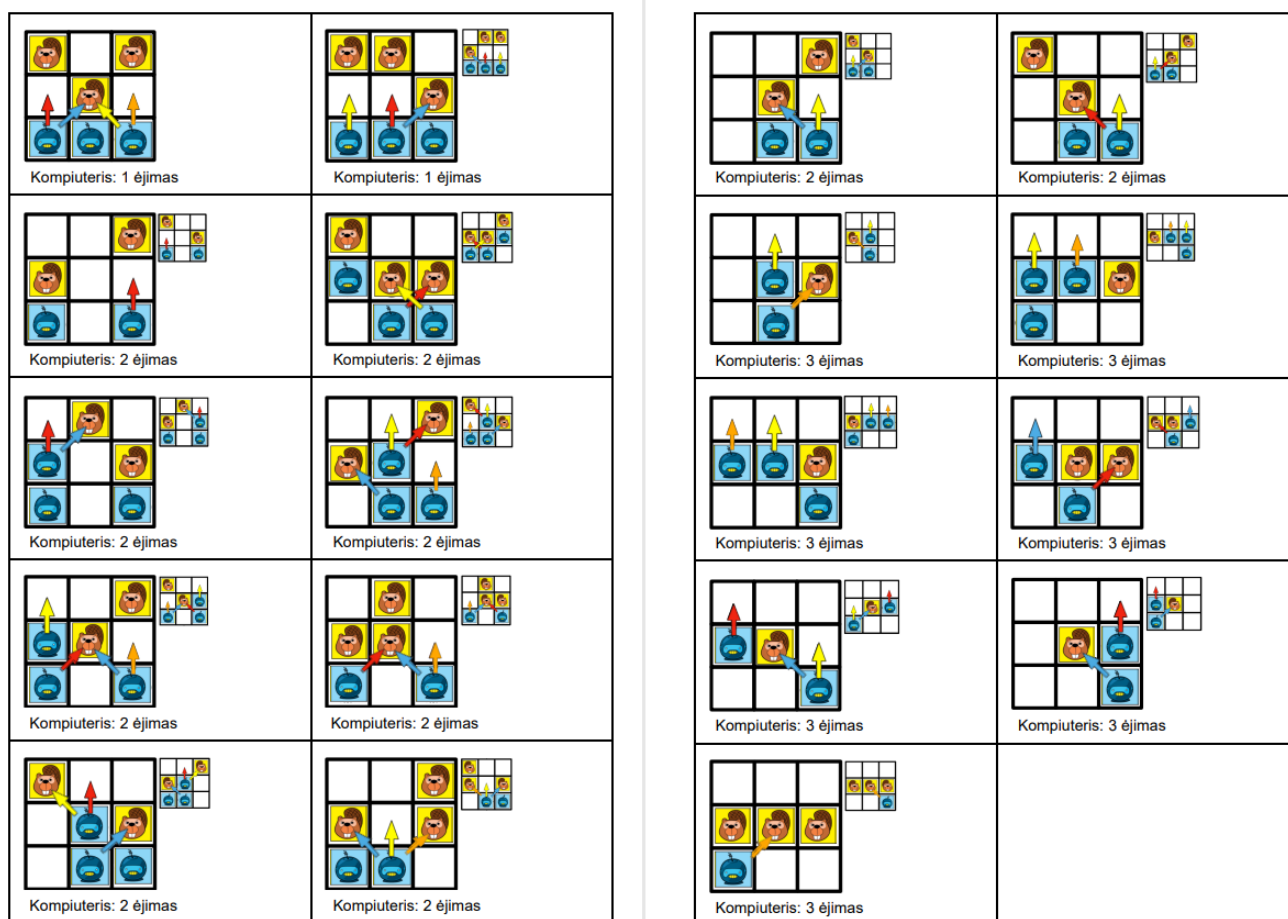
Pradedą bebras, jis gali eiti vienu langeliu tiesiai (pažymėta mėlyna rodykle). Jei kažkas stovi priešais esančiame langelyje, eiti negalima. Kirsti galima tik įstrižai (raudonos rodyklės), bet nėra privaloma. Kertama ne peršokant, o atsistoiant ant tos figūros (kaip šachmatuose), nukirsta figūra nuimama.

Robotai vaikšto ir kerta pagal tokias pat taisykles: eina tiesiai, kerta įstrižai.

Žaidimas laimimas, kai: 1) viena iš figūrų pasiekia priešingą pusę, 2) numušą visas priešininko figūras arba 3) užtikrina, kad priešininkas nebegalėtų judėti, t. y., užstato priešininką savo figūromis taip, kad jo ėjimo metu jis neturėtų kur eiti ir negalėtų nieko kirsti.

Sužaiskite dviese keletą kartų, kad perprastumėte žaidimo taisykles.

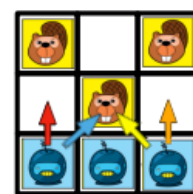
Tiek „bebrai“, tiek „robotai“ patys mąstė, kur eiti ir ką kirsti, kad laimėtų žaidimą. Dabar žaidimą truputį pakoreguosime. Bebrai atstovaus žmones, o robotai dirbtinį intelektą. Taigi dabar robotai turės visiškai apibrėžtus savo ėjimus ir patys nebemąstys, o ieškos žaidimo situacijos pateiktose lentelėse:



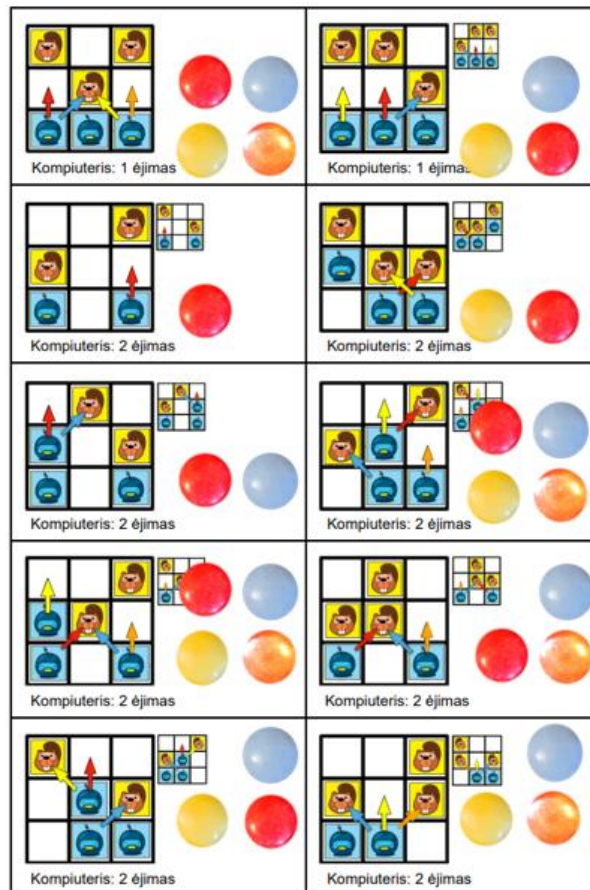
Pradedą bebrai ir, jei paeina vienu langeliu į vidurį, tai robotas turi situaciją, parodytą pirmosios lentelės pačiame viršutiniame kampe:

Robotas gali rinktis ėjimą pagal pateiktas rodykles. Eiti tiesiai (raudona arba oranžinė), kirsti bebrą (mėlyna arba geltona). Šioje situacijoje robotas negalvoja, kuris ėjimas jam būtų naudingesnis, jis yra kompiuteris ir renkasi ėjimą atsitiktine tvarka. Paėjus robotui, eina bebras. Tuomet robotas susidariusios situacijos vėl ieško žaidimo lentoje pateiktose ėjimų lentelėse. Pastebėsime, kad prie kiekvienos pagrindinės žaidimo situacijos yra pateikiama veidrodinė situacija, kurią taip pat reikia peržiūrėti. Taip žaidžiama tol, kol laimi viena arba kita pusė.

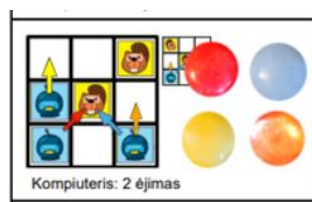
Štai taip sistema mokosi. Dabar turimas taisyklės papildysime požymiais. Tam galite nupirkti atitinkamų spalvų (raudonos, oranžinės, mėlynos, geltonos) mažų saldainiukų, sagų ar kokių kitų figūrėlių. Jas išdėliokite ant galimų ėjimų lentelių prie kiekvienos situacijos:



Kompiuteris: 1 ėjimas

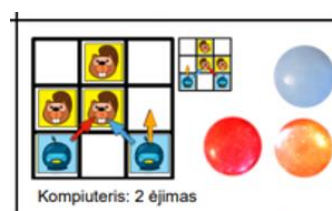


Kokios spalvos galimų ėjimų rodyklės yra prie kiekvienos situacijos, tokių spalvų saldinius ar figūrėles išdėliojame šalia. Galima pradėti žaidimą. Pirmi pradeda bebrai, tuomet robotai ieško situacijos lentelėse ir pagal nurodytus ėjimus atsitiktinai pasirinkę paeina. O kam reikalingi tie mūsų požymiai? Tarkime, robotas atsiduria tokioje situacijoje:



Jis gali rinktis iš 4 galimų ėjimų ir, tarkim, pasirenka eiti pagal geltoną rodyklę. Taip paėjęs robotas laimi. Kad kitą kartą, atsidūręs tokioje situacijoje, žinotų, jog geltonas ėjimas sėkmingas, jis šalia pasideda dar vieną geltoną saldainį ar figūrėlę. Taigi kitą kartą visų ėjimų saldainių bus po vieną, o geltonų jau du. Kaip prisimenate iš ankstesnės užduoties, geltono ėjimo požymis sustiprėjo. Tad kitą kartą robotas rinksis tik šį ėjimą.

Tarkime, žaidimo eigoje robotas atsiduria tokioje situacijoje:



Galima rinktis iš trijų ėjimų (raudono, oranžinio ir mėlyno) ir robotas pasirenka raudoną. Jį pasirinkęs nukerta bebrą. Tuomet eilė bebrui. Kairėje pusėje esantis bebras paeina tiesiai į atsilaisvinusią roboto vietą ir laimi. Tokiu atveju, kadangi robotas pralaimėjo, jis turi pašalinti nuo lentos nesėkmę lėmusį ėjimą, kad kitą kartą jo nebekartotų. Tad nuo šios situacijos langelio nuimamas raudonas saldainis. Kitą kartą robotas, papuolęs į šią situaciją, nebegalės rinktis raudono ėjimo.

Tokiu principu, nuimdami ir pastiprindami galimus ėjimus sužaiskite keletą kartų. Maždaug tiek, kiek matysite, kad bebrui jau nebepavyksta laimėti. Tokią situaciją turėtumėte pasiekti po maždaug 20 žaidimų raundų. Tiek maždaug reikia, kad dėliojant šiuos požymius dirbtinis intelektas apsimokytų pakankamai, jog sugebėtų visada laimėti.

(Žaidimo kortelės ir taisyklių lapai pateikiami šio dokumento pabaigoje.)

Dirbtiniai neuroniniai tinklai

Dirbtinis neuroninis tinklas (angl. *artificial neural network*) – tarpusavyje sujungtų dirbtinių neuronų grupė. Ši technologija mėgdžioja žmogaus galvos smegenų darbą – tiksliau, neuronų veikimą. Panašiai kaip ir smegenys, dirbtinių neuronų tinklai yra sudaryti iš tarpusavyje sujungtų mazgų arba neuronų, kurie veikia kartu apdorojami ir analizuodami duomenis.

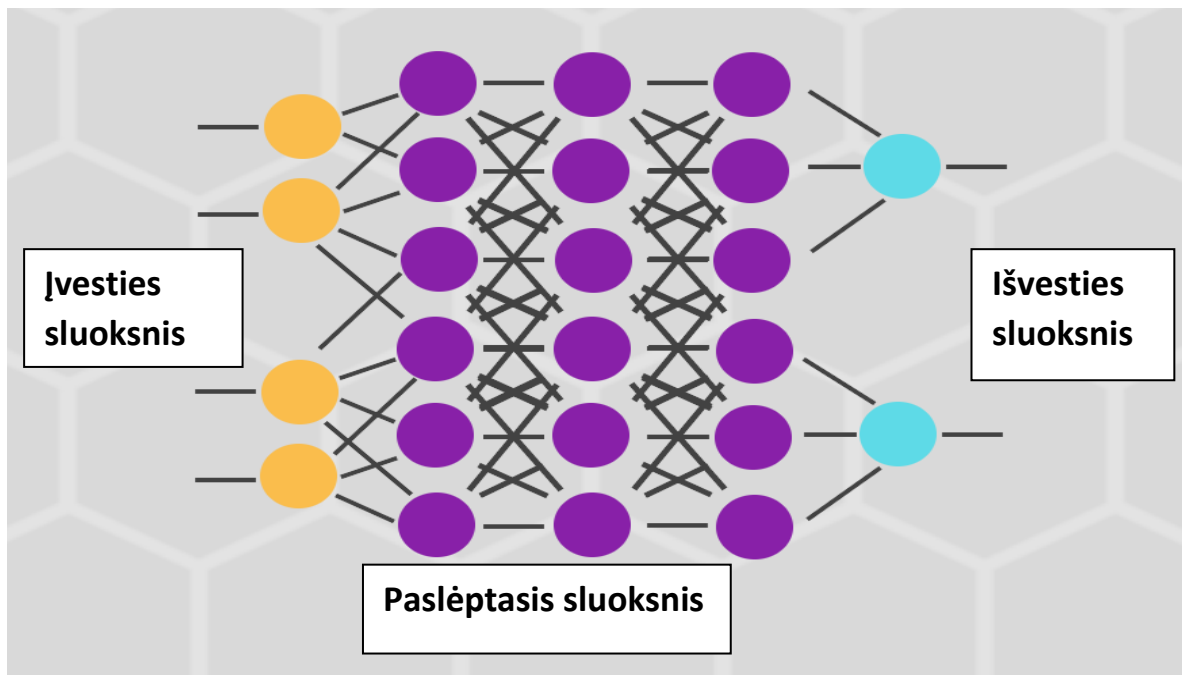
Pagrindinę dirbtinio neuroninio tinklo struktūrą sudaro trys sluoksniai: įvesties sluoksnis, paslėptasis sluoksnis ir išvesties sluoksnis (4 pav.). Įėjimo sluoksnyje į tinklą įvedami duomenys, paslėptajame sluoksnyje duomenys apdorojami ir analizuojami, o išvesties sluoksnyje rodomi analizės rezultatai.

Kiekvienas tinklo neuronas yra sujungtas su kitais neuronais keliais, vadinamais sinapsėmis. Šių jungčių stiprumas, arba svoriai, lemia, kaip tinklas apdoroja ir analizuoja informaciją. Tinklas, reaguodamas į grįžtamąjį ryšį, šių jungčių svorius koreguoja vadinamuoju grįžtamojo ryšio procesu, todėl laikui bėgant gali pagerinti savo tikslumą.

Įsivaizduokime dirbtinį neuroninį tinklą kaip darbuotojų komandą, kuri bando išspręsti galvosūkį. Kiekvienas darbuotojas yra atsakingas už skirtingą dëlionės dalį ir visi jie bendrauja tarpusavyje, kad išsiaiškintų, kaip sujungti visas dalis.

Dirbtiniame neuroniniame tinkle kiekvieną darbuotoją atstovauja „neuronas“. Šie neuronai tarpusavyje sujungti sinapsėmis, kurios yra tarsi komunikacijos kanalai tarp darbininkų.

Pradedant dëlionės sprendimo procesą, komandai duodama neužbaigta dëlionė ir pasakoma, kaip turėtų atrodyti užbaigta dëlionė. Darbuotojai pradeda nagrinėti matomas dëlionės dalis ir bando išsiaiškinti, kaip jos dera tarpusavyje.



SEQ_ * ARABIC 4 pav. Dirbtinių neuroninių tinklų schema
Šiai nežinomo autoriaus [nuotraukai](#) suteikta [CC BY-SA](#) licencija

Kiekvienas darbuotojas gali atlikti paprastus skaičiavimus su dëlionės dalimis, už kurias yra atsakingas. Pavyzdžiui, jei darbuotojas atsakingas už kampinę dëlionės detalę, jis gali pažvelgti į gretimų detalių spalvą ir formą ir atspėti, kaip turėtų atrodyti trūkstamos detalės.

Kadangi darbuotojai bendrauja tarpusavyje per sinapses, jie gali sujungti savo atskirus spėjimus ir susidaryti išsamesnį vaizdą, kaip turėtų atrodyti užbaigta dëlionė. Laikui bėgant jie koreguoja savo spėjimus, remdamiesi grįžtamojo ryšiu, ir toliau dirba, kad pagerintų bendrą tikslumą.

Kai komanda įsitikina, kad išsprendė galvosūkį, ji gali pristatyti savo užbaigtą darbą asmeniui, kuris jiems davė šį galvosūkį. Jei jų išspręstas galvosūkis atitinka norimą rezultatą, jie sėkmingai atliko užduotį!

Dirbtiniame neuroniniame tinkle vyksta labai panašūs procesai. Neuronai dirba kartu, analizuodami įvesties duomenis ir stengdamiesi gauti tikslus rezultatus. Remdamiesi grįžtamojo ryšiu jie koreguoja savo svorius bei skaičiavimus ir toliau tobulina savo prognozes, kol įsitikina, kad išvedė teisingą rezultatą.

Vaizdų atpažinimas – viena iš sričių, kur yra naudojami dirbtiniai neuroniniai tinklai.. Pavyzdžiui, neuroninį tinklą galima išmokyti nuotraukose atpažinti kates. Šis metodas naudingas kuriant savavaldžius automobilius, nes toks automobilis turi gebėti atpažinti savo aplinkos objektus.

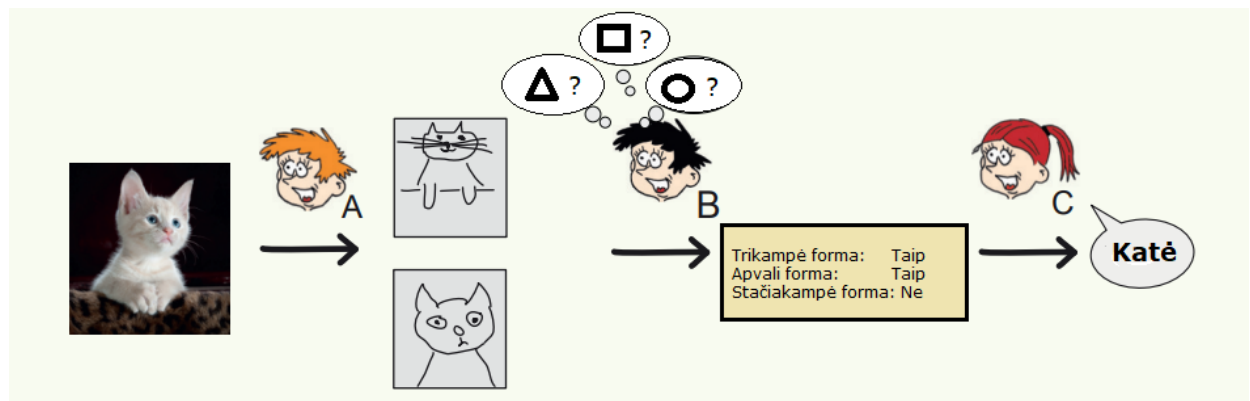
4 praktinė užduotis

Šioje užduotyje labai supaprastintai pateikiami dirbtinių neuroninių tinklų veikimo principai.

Užduočiai reikės: nuotraukų su kačių, automobilių ir namų vaizdais.

Instrukcija

Padalinkite mokinius į grupes po tris, kiekviena grupė gauna po krūvą nuotraukų. Kiekvienos grupės mokiniai pasidalina tris vaidmenis (A, B ir C), simbolizuojančius po vieną neuroninio tinklo sluoksnį.



Vaidmenis pasiskirstę mokiniai atlieka tokius veiksmus. A iš krūvos nuotraukų pasirenka paveikslėlį (B ir C neturi matyti jo vaizdo), sukuria du skirtingus eskizus (piešimo laikas – po 30 sekundžių kiekvienam) ir perduoda juos B vaidmenį pasirinkusiam mokiniui. Svarbu, kad C nematytų eskizų. B gauna eskizus iš A ir patikrina, ar gautuose eskizuose pavaizduotos stačiakampės, trikampės ar apvalios figūros. Tada B perduoda surinktą informaciją apie eskizuose esančias figūras C vaidmenį atliekančiam mokiniui. C įvertina gautą informaciją naudodamasis šia lentele:

	<i>Stačiakampės formos?</i>	<i>Trikampės formos?</i>	<i>Apvalios formos?</i>
Namas	Taip	Taip	Ne
Automobilis	Taip	Ne	Taip
Katė	Ne	Taip	Taip

Įvertinęs gautą informaciją C paskelbia, ar pirminis paveikslėlis yra namas, ar automobilis, ar katė. Po to A pasako, ar sprendimas teisingas.

Leiskite mokiniams išbandyti žaidimą atliekant skirtingus vaidmenis. Taip pat galima daryti, kad vieną vaidmenį atlieka du mokiniai (arba aiškiai paskirti kiekvieną vaidmenį trims mokiniams, kurių kiekvienas atstovaus po neuroną, t. y., sluoksnio mazgą, o ne visą sluoksnį).

Po kažkurio laiko išdalinkite mokiniams keletą paveikslėlių, kurie arba nepatenka į kategorijas, kurias tinklas gali atpažinti, arba turi požymių, kurie neleidžia aiškiai klasifikuoti. Pavyzdžiui, šuns paveikslėlis nėra teisingai atpažįstamas, nes tinklas nežino kategorijos „šuo“. Remiantis šia išvada, mokiniai turėtų pasvarstyti, kaip tinklas gali būti pakeistas ar išplėstas, kad ateityje galėtų atpažinti šios ar kitos kategorijos objektus. Pirmiausia turi būti įvesta nauja išvesties kategorija. Taip pat šiuo metu esančių charakteristikų kiekis pasidaro per mažas, kad galima būtų tinkamai atpažinti objekto kategoriją. Todėl reikia pridėti papildomų požymių, kurie leidžia atskirti kategorijas, arba kelios charakteristikos turi būti sujungtos, kad būtų sudarytas sudėtingesnės charakteristikos modelis. Šis sujungimas galiausiai apima papildomų sluoksnių pridėjimą į neuroninį tinklą.

Keletas nuorodų, kur galima išbandyti įvairias dirbtinio intelekto veiklas

<https://course.elementsofai.com/lt/> – „Elementas of AI“ („DI elementai“) sudaro keli nemokami internetiniai įmonės „MinnaLearn“ ir Helsinkio universiteto parengti kursai. Aptariama, kas yra dirbtinis intelektas, ką su juo galima (ir ko neįmanoma) nuveikti ir kaip pradėti kurti dirbtinio intelekto metodus. Šiuose kursuose teorija derinama su praktiniais pratimais ir suteikiama galimybė mokytis savo tempu.

<https://teachablemachine.withgoogle.com/> – „Google“ korporacijos sukurta internetinė priemonė, leidžianti naudotojams kurti mašininio mokymosi modelius naudojant savo vaizdus, garsus ir gestus. Įrankis sukurtas taip, kad juo būtų lengva naudotis ir nereikėtų jokių išankstinių mašininio mokymosi žinių.

<https://chat.openai.com/chat> – „OpenAI“ natūralios kalbos apdorojimo technologija paremtas internetinis pokalbių robotas. Pokalbių robotas sukurtas taip, kad su juo būtų galima kalbėtis ir gauti atsakymus įvairiomis temomis.

<https://www.deepl.com/translator> – internetinė vertimo paslauga, kurioje neuroniniai tinklai naudojami aukštos kokybės vertimams teikti. Paslauga palaiko vertimus iš įvairių kalbų.

<https://experiments.withgoogle.com/collection/ai> – „Google“ korporacijos interaktyvių eksperimentų rinkinys, kuriame pristatomos mašininio mokymosi ir dirbtinio intelekto galimybės.

<https://code.org/oceans> – edukacinė platforma, kurioje pateikiama mokomoji medžiaga ir veiklos, skirtos mokyti mokinius informatikos ir programavimo. Taip pat pateikiama medžiaga, skirta mokyti dirbtinio intelekto.

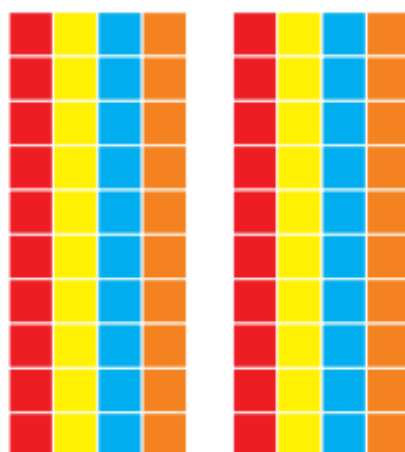
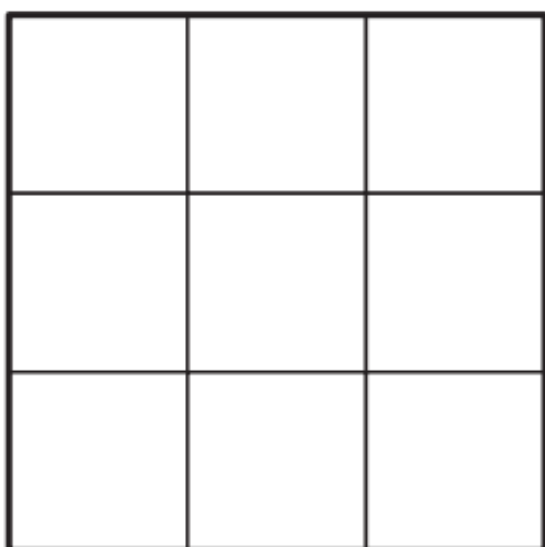
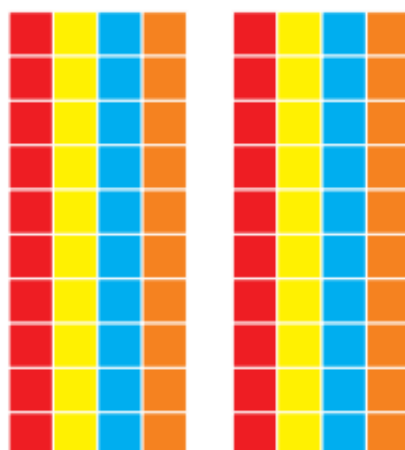
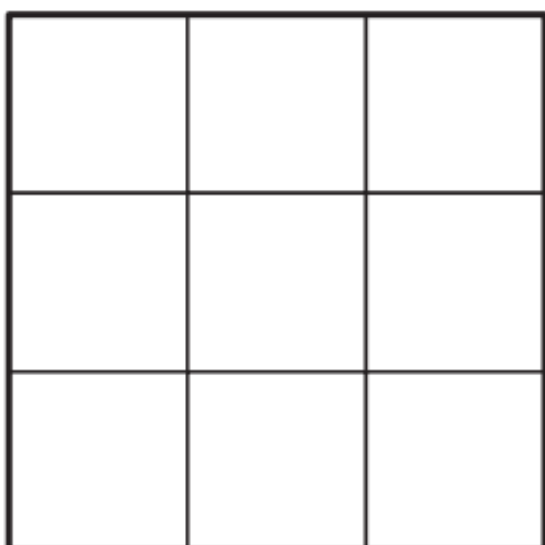
Šaltiniai

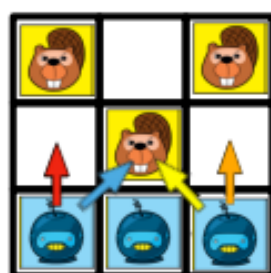
Lindner, A., Seegerer, S. AI unplugged: <https://www.aiunplugged.org/>

Kaplan, A., Haenlein, M. (2019) Siri, Siri in my Hand, who's the Fairest in the Land? On the Interpretations, Illustrations and Implications of Artificial Intelligence, *Business Horizons*, 62(1), 15-25.

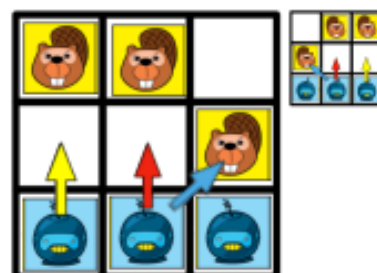
Haenlein, M., Kaplan, A. (2019). Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence. *California Management Review*. 61(4), 5-14. doi:10.1177/0008125619864925

Long, D., Magerko, B. (2020). What is AI literacy? Competencies and design considerations. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, 1-16.

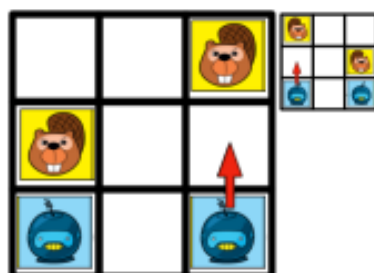




Kompiuteris: 1 ējimas



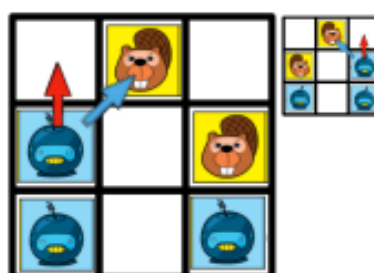
Kompiuteris: 1 ējimas



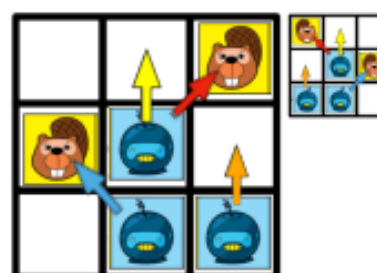
Kompiuteris: 2 ējimas



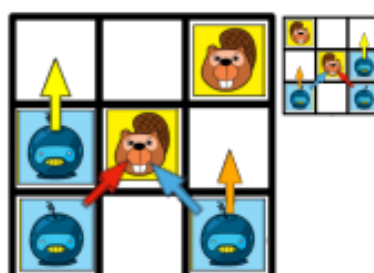
Kompiuteris: 2 ējimas



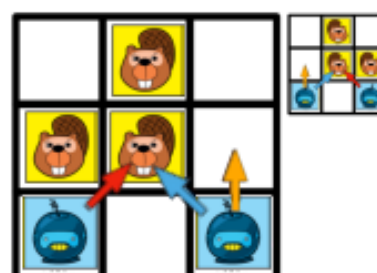
Kompiuteris: 2 ējimas



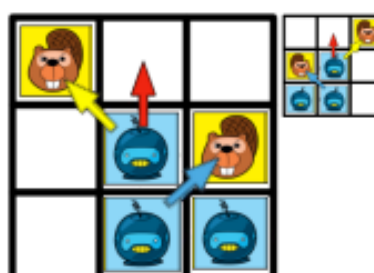
Kompiuteris: 2 ējimas



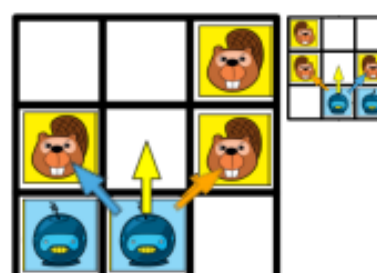
Kompiuteris: 2 ējimas



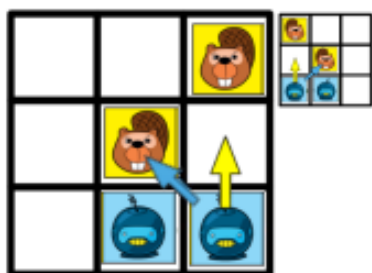
Kompiuteris: 2 ējimas



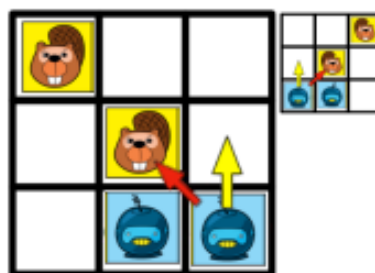
Kompiuteris: 2 ējimas



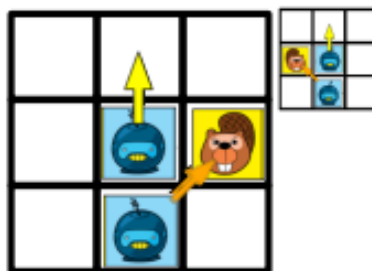
Kompiuteris: 2 ējimas



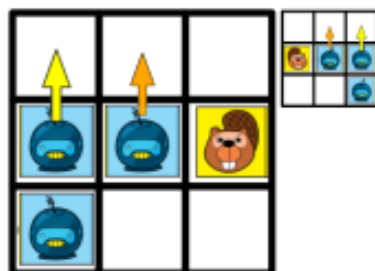
Kompiuteris: 2 ējimas



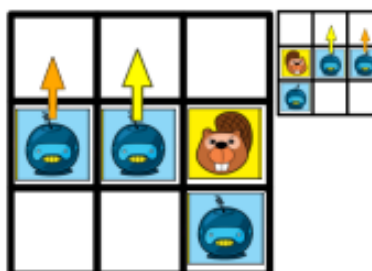
Kompiuteris: 2 ējimas



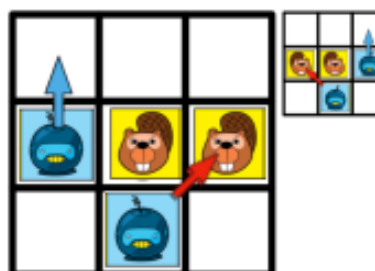
Kompiuteris: 3 ējimas



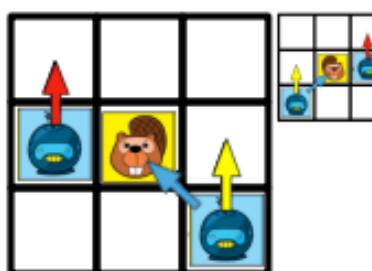
Kompiuteris: 3 ējimas



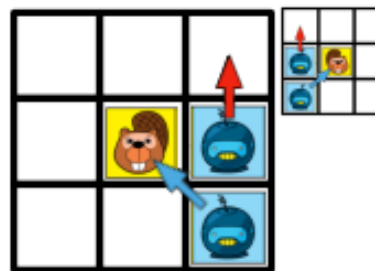
Kompiuteris: 3 ējimas



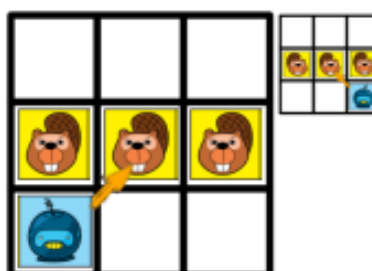
Kompiuteris: 3 ējimas



Kompiuteris: 3 ējimas



Kompiuteris: 3 ējimas



Kompiuteris: 3 ējimas