

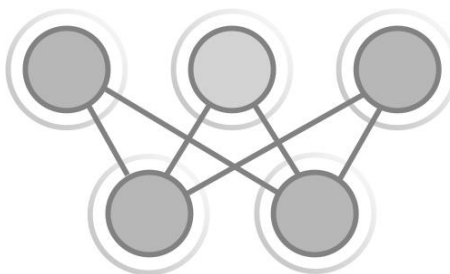
Prof.dr.sc. Bojana Dalbello Bašić


Fakultet elektrotehnike i računarstva
Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

www.zemris.fer.hr/~bojana
bojana.dalbello@fer.hr

Strojno učenje

Oblikovanje sustava koji uči



- 
- Postavljanje problema
 - Oblikovanje sustava
 - Način stjecanja iskustva
 - Ciljna funkcija
 - Reprezentiranje ciljne funkcije
 - Algoritam aproksimiranja ciljne funkcije
 - Procjena vrijednosti primjera za učenje
 - Završno oblikovanje
 - Slijed koraka u dizajnu sustava strojnog učenja
 - Teme u strojnom učenju
 - Zadatak

- **Problem učenja definiramo kroz:**
 - **Zadatak T** –
 - **Mjera P** – mjera uspješnosti koju povećavamo
 - **Iskustvo E** – izvor za stjecanje znanja
- **Definicija:**
 - Za računalski program (sustav) kaže se da **uči** kroz iskustvo **E** u odnosu na neki skup zadataka **T** i s obzirom na neku mjeru uspješnosti **P**, ako se povećava uspješnost obavljanja zadataka **T**, kroz iskustvo **E**, mjerena mjerom uspješnosti **P**.

Postavljanje problema (2)

- *Primjer:*

Zadatak T: raspoznavanje i klasificiranje rukom pisanih znamenaka

Mjera P: postotak ispravno klasificiranih znamenaka

Iskustvo E: primjeri za učenje - baza podataka rukom napisanih znamenaka s pripadnom klasifikacijom

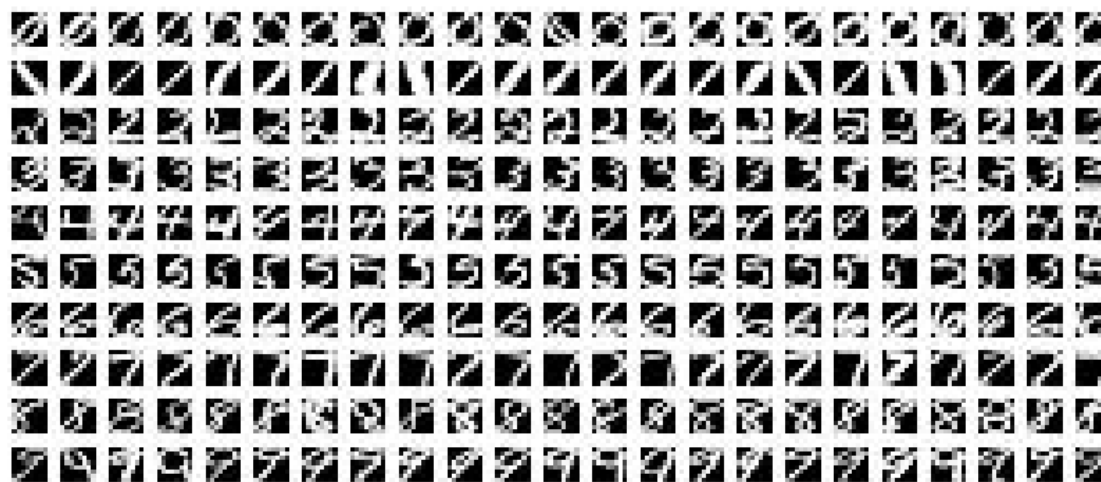


Fig. 3. Images of handwritten digits, normalized for horizontal and vertical scale and translation and sampled on an 8×8 pixel grid. Different writing angles introduce different levels of shearing in each image.

Postavljanje problema (3)

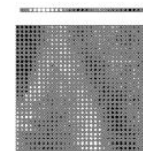
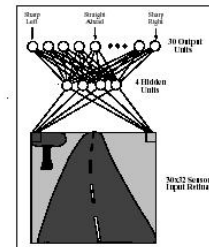
- *Primjer:*

Zadatak T: vožnja 4-tračnom auto-cestom uz uporabu senzora vida

Mjera P: prijeđena prosječna udaljenost prije nastanka pogreške

Iskustvo E: niz slika i upravljačkih komandi snimljenih za vrijeme vožnje čovjeka

ALVINN [Pomerleau] drives 70 mph on highways



Oblikovanje sustava koji uči



Odabir načina stjecanja iskustva



Određivanje ciljne funkcije



**Odabir predstavljanja
ciljne funkcije za učenje**



**Odabir algoritma
za aproksimaciju funkcije**

Oblikovanje sustava koji uči



Odabir načina stjecanja iskustva



Određivanje ciljne funkcije



**Odabir predstavljanja
ciljne funkcije za učenje**



**Odabir algoritma
za aproksimaciju funkcije**

Oblikovanje sustava koji uči



Odabir vrste primjera za učenje



Određivanje ciljne funkcije



**Odabir predstavljanja
ciljne funkcije za učenje**



**Odabir algoritma
za aproksimaciju funkcije**

Odabir načina stjecanja iskustva

- Iskustvo = primjeri za učenje

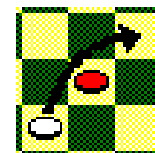
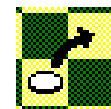
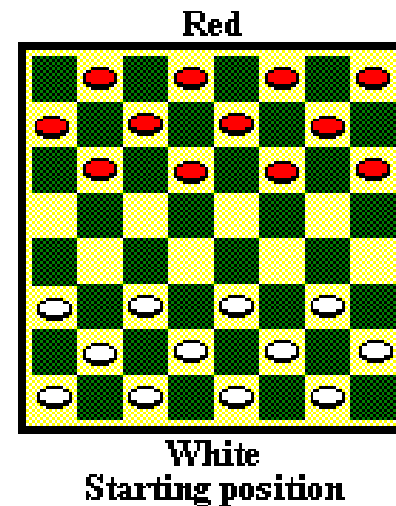
- Primjer:*

Zadatak T: igra DAME

Mjera P: postotak pobjeda

Iskustvo E: igranje protiv samog sebe

- Tri važna **svojstva iskustva**:
 - direktni ili indirektni povratni utjecaj?
 - učenik kontrolira slijed primjera za učenje?
 - reprezentativnost ?



Odabir načina stjecanja iskustva (2)

- **1. svojstvo iskustva** - utjecaj iskustva na učenje

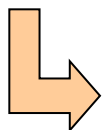
Igra Dame – sustav uči:

- **Direktno:**

- primjeri za učenje = pojedinačna stanja na ploči + pridružen ispravan korak

- **Indirektno:**

- primjeri za učenje = niz poteza + konačni ishod različitih igara



Ispravnost određenog poteza **izvodi se** (u ranoj fazi igre) indirektno iz činjenice je li takva igra dobivena ili izgubljena.

Odabir načina stjecanja iskustva (4)

2. svojstvo iskustva (engl. training experience) je stupanj u kojem učenik kontrolira slijed primjera za učenje.

Učitelj izabire informativne primjere (*npr. stanje na ploči*) i daje korektan potez.

ILI

Učenik (program) upravlja stanjima na ploči i indirektnom klasifikacijom primjera za učenje (*npr. program igra protiv samog sebe*)

Odabir načina stjecanja iskustva (5)

- **3. svojstvo skupa primjera za učenje**

Reprezentativnost primjera za učenje.

- Primjer: izvor iskustva: sustav igra protiv samog sebe.

- U teoriji strojnog učenja – važna pretpostavka:

▪
**Raspodjela primjera za učenje = Raspodjeli primjera
za testiranje.**

- **Odluka u oblikovanju našeg sustava igre DAME**: naš sistem će učiti igrajući sam protiv sebe.
- Kakav je to tip iskustva? Prednost takvog izbora? Nedostatak?

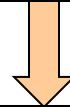
Oblikovanje sustava koji uči



Odabir načina stjecanja iskustva



Određivanje ciljne funkcije



**Odabir predstavljanja
ciljne funkcije za učenje**



**Odabir algoritma
za aproksimaciju funkcije**

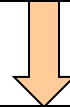
Oblikovanje sustava koji uči



Odabir načina stjecanja iskustva



Što će program učiti?



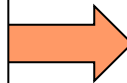
**Odabir predstavljanja
ciljne funkcije za učenje**



**Odabir algoritma
za aproksimaciju funkcije**

- Koji tip znanja će se usvajati i kako će ga koristiti sustav za igru?
- Igra DAME:

stanje na ploči



skup dozvoljenih stanja

- Program mora naučiti kako izabrati najbolji potez iz skupa dozvoljenih poteza.

- Što će program učiti?
- Funkciju koja izabire najbolji potez iz skupa dozvoljenih poteza, nazovimo tu funkciju *ChooseMove*:
 - ***ChooseMove*: $B \rightarrow M$**
B – skup stanja (situacija na ploči)
M – skup dozvoljenih poteza
- *ChooseMove* - dobar izbor, ali ju je teško učiti kada je na raspolaganju indirektno iskustvo

- Alternativna ciljna funkcija je *evaluacijska funkcija* **V** koja pridružuje numeričku vrijednost svakom pojedinačnom stanju.
 - **$V : B \rightarrow \mathbb{R}$**
B – skup svih dozvoljenih stanja
 \mathbb{R} – skup realnih brojeva

Odabir ciljne funkcije (4)

- Ako sustav može uspješno naučiti funkciju **V** tada lako može izabrati najbolji potez u bilo kojoj situaciji na ploči
- Što treba biti vrijednost funkcije **V(b)** za neko stanje na ploči b?
 - Načelno, svaka funkcija koja pridružuje veću vrijednost boljim stanjima.

Odabir ciljne funkcije (5)

- Definicija evaluacijske funkcije V za igru DAME (jedna od mnogo mogućih):

Ako je b završno stanje na ploči i pobjeda
onda **$V(b)=100$**

Ako je b završno stanje na ploči i poraz
onda **$V(b)=-100$**

Ako je b završno stanje na ploči i neriješeno
onda **$V(b)=0$**

Ako b nije završno stanje u igri onda **$V(b) = V(b')$** ,
gdje je b' najbolje završno stanje koje se može
dosegnuti iz trenutnog stanja b , igrajući optimalno do
kraja igre (uz pretpostavku da to čini i protivnik).

Definicija V – ispravna, ali neupotrebljiva!

Odabir ciljne funkcije (6)

- U stvarnosti algoritam učenja radi samo s aproksimacijom ciljne funkcije (označavamo je s \hat{V}).
- Proces učenja ciljne funkcije naziva se aproksimacija funkcije

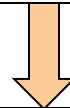
Oblikovanje sustava koji uči



Odabir načina stjecanja iskustva



Određivanje ciljne funkcije



**Odabir predstavljanja
ciljne funkcije za učenje**



**Odabir algoritma
za aproksimaciju funkcije**

Odabir reprezentacije za ciljnu funkciju

- Nakon odabira idealne ciljne funkcije V – treba izabrati predstavljanje funkcije \hat{V} za implementaciju i učenje.
- \hat{V} može biti:
 - tablica s vrijednostima za svako stanje na ploči
 - skup produkcijskih pravila koja se podudaraju sa značajkama stanja na ploči
 - kvadratna polinomijalna funkcija predefiniranih značajki stanja na ploči
 - umjetna neuronska mreža

Odabir reprezentacije za ciljnu funkciju (2)

Ekspresivnost funkcije \hat{V}
(što bliža V) **vs.**

Količina podataka za učenje koja je
potrebna za izbor između alternativnih
hipoteza

\hat{V} će se računati kao linearna kombinacija slijedećih značajki:

x_1 – broj crnih pločica na ploči

x_2 – broj crvenih pločica na ploči

x_3 – broj crnih kraljeva na ploči

x_4 – broj crvenih kraljeva na ploči

x_5 – broj crnih pločica koje su ugrožene crvenima

x_6 – broj crvenih pločica koje su ugrožene crnima

$$\hat{V}(b) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6$$

- Težinski faktori $w_0 \dots w_6$ podešavaju se u postupku učenja i određuju važnost pojedinih značajki.

Odabir reprezentacije za ciljnu funkciju (3)

Specifikacija
zadaće učenja

Zadatak T – igra DAME

Mjera uspješnosti P – postotak igara dobivenih na svjetskom turniru

Iskustvo E – igre igrane protiv samog sebe.

Implementacija
skup odabranih
rješenja za model

Ciljna funkcija: $V : B \rightarrow \mathbb{R}$

Predstavljanje ciljne funkcije:

$$V' = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

- Učinak oblikovanja modela je reduciranje problema sa učenja strategije igre na učenje vrijednosti koeficijenata $w_0 \dots w_6$

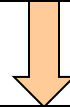
Oblikovanje sustava koji uči



Odabir načina stjecanja iskustva



Određivanje ciljne funkcije



**Odabir predstavljanja
ciljne funkcije za učenje**



**Odabir algoritma
za aproksimaciju funkcije**

Odabir algoritma za aproksimaciju funkcije

- Učenje funkcije \hat{V} zahtijeva skup primjera za učenje, svaki primjer je par $(b, V_{\text{train}}(b))$
- *Primjer*
 - $((x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0), +100)$

Procjena vrijednosti primjera za učenje

- Dostupna informacija:
 - Je li igra dobivena ili ne (**indirektno iskustvo**).
- Potrebno:
 - Primjeri za učenje koji pridružuju numeričke vrijednosti pojedinim stanjima na ploči.
- Završno stanje – lako pridjeljivanje vrijednosti
- Međustanja - ?
 - Dobivena ili izgubljena igra ne znači nužno da su svi potezi kroz igru loši ili dobri.

Procjena vrijednosti primjera za učenje (2)

- Pravilo procjene vrijednosti V za učenje

$$V_{\text{train}}(b) \leftarrow \hat{V}(\text{sljedbenik}(b))$$

Učeničeva tekuća
aproksimacija
funkcije V .

$\text{sljedbenik}(b)$ – sljedeće stanje
koje slijedi b i za kojeg je opet na
potezu program.

- Tekuća verzija aproksimacije \hat{V} koristi se za procjenu vrijednosti primjera za učenje $V_{\text{train}}(b)$, a zatim će ta vrijednost biti korištena za podešavanje same te funkcije.

- Traži se algoritam za određivanje w_i da bi najbolje aproksimirao skup primjera za učenje $\{(b, V_{\text{train}}(b))\}$.
- Jedan pristup određivanju **najbolje hipoteze** (skup težina $w_0 \dots w_6$) jest minimiziranje kvadrata pogreške između vrijednosti za učenje i vrijednosti dobivene na temelju hipoteze.

$$E = \sum_{(b, V_{\text{train}}(b)) \in \text{Skup primjera za učenje}} (V_{\text{train}}(b) - \hat{V}(b))^2$$

- Minimizacija $E \equiv$ nalaženje najvjerojatnije hipoteze za dani skup uzoraka za učenje.

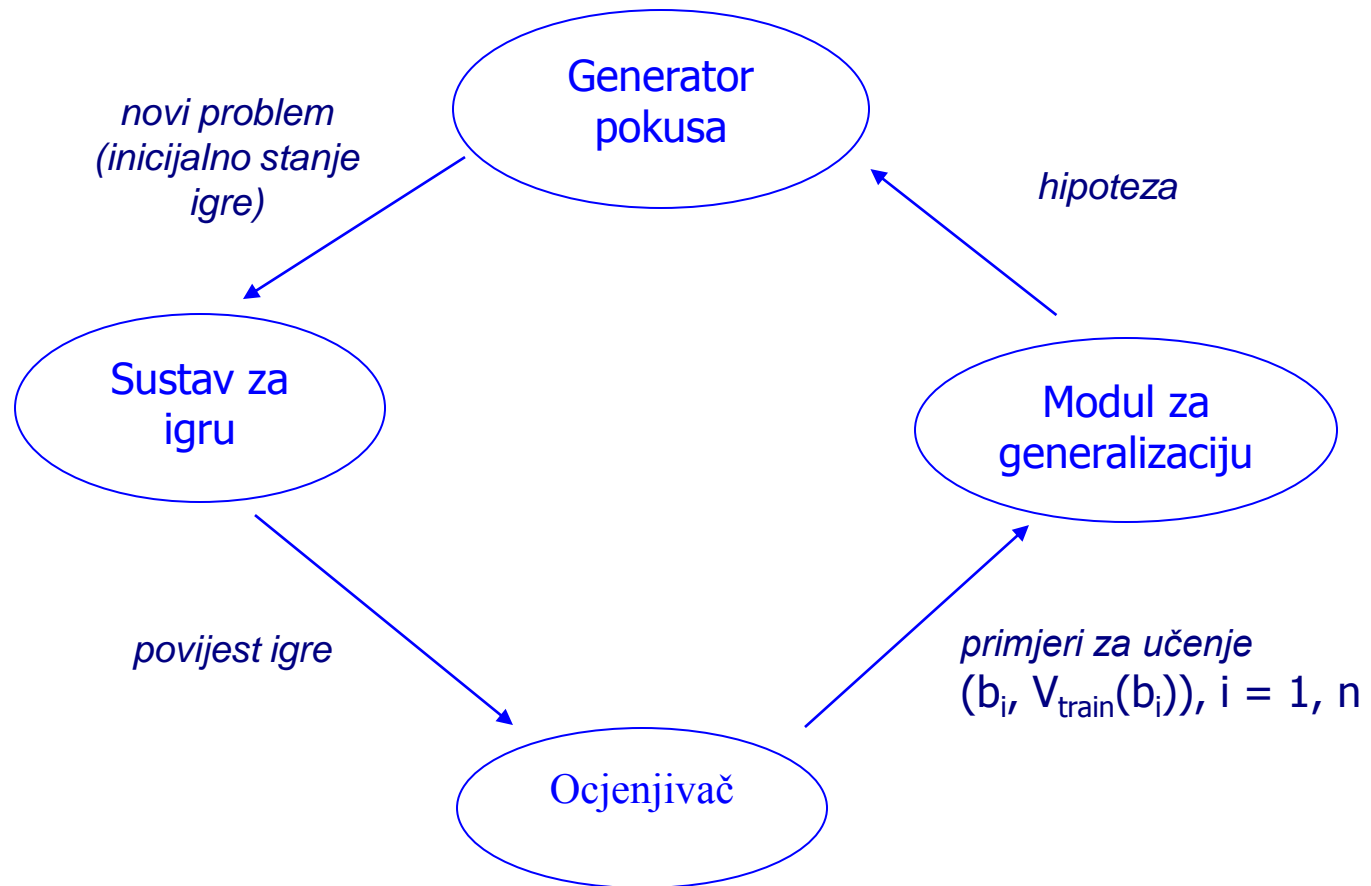
Podešavanje težina (2)

- Zahtjevi na algoritam za učenje:
 - postupno podešavanje težina s novim primjerima i
 - robusnost na pogreške u primjerima za učenje.
- Jedan od takvih algoritama je **LMS** (**L**east **M**ean **S**quares). Za svaki pojedini primjer korigiraju se težine u nekom manjem iznosu, u smjeru koji smanjuje pogrešku.
- LMS se može shvatiti kao pretraživanje prostora mogućih hipoteza (težinskih koeficijenata) uz stohastički gradijentni silazak tako da se minimizira pogreška E .

- Za svaki primjer za učenje (b , $V_{\text{train}}(b)$)
- Izračunaj $\hat{V}(b)$ uz trenutne vrijednosti težinskih koeficijenata.
- Za svaki težinu w_i , podesi:
$$w_i = w_i + \eta (V_{\text{train}}(b) - \hat{V}(b)) x_i$$
- η konstanta (npr. 0.1) oblikuje veličinu korekcije težina.
- Ako je $(V_{\text{train}}(b) - \hat{V}(b)) = 0$ tada nema korekcije.

- Program koji uči igrati DAME može biti opisan sa 4 programska modula koji predstavljaju centralnu komponentu u većini sustava strojnog učenja:
 - 1. Sustav za igru
 - 2. Ocjenjivač
 - 3. Modul za generalizaciju
 - 4. Generator pokusa

Završno oblikovanje (2)



- **Sustav za igru** (*engl. Performance System*)
 - Modul koji rješava zadanu zadaću igre.
- *Ulaz:* Instanca nove situacije (nove igre)
- *Izlaz:* trag odluka (povijest igre).
- U našem slučaju, *sustav za igru* izabire slijedeći potez u svakom koraku na temelju naučene evaluacijske funkcije \hat{V}
Očekujemo poboljšanje ponašanja s povećanjem točnosti funkcije V .

- **Ocjenjivač** (*engl. Critic*)
- *Ulaz:* povijest (trag) igre.
- *Izlaz:* primjeri za učenje ciljne funkcije.

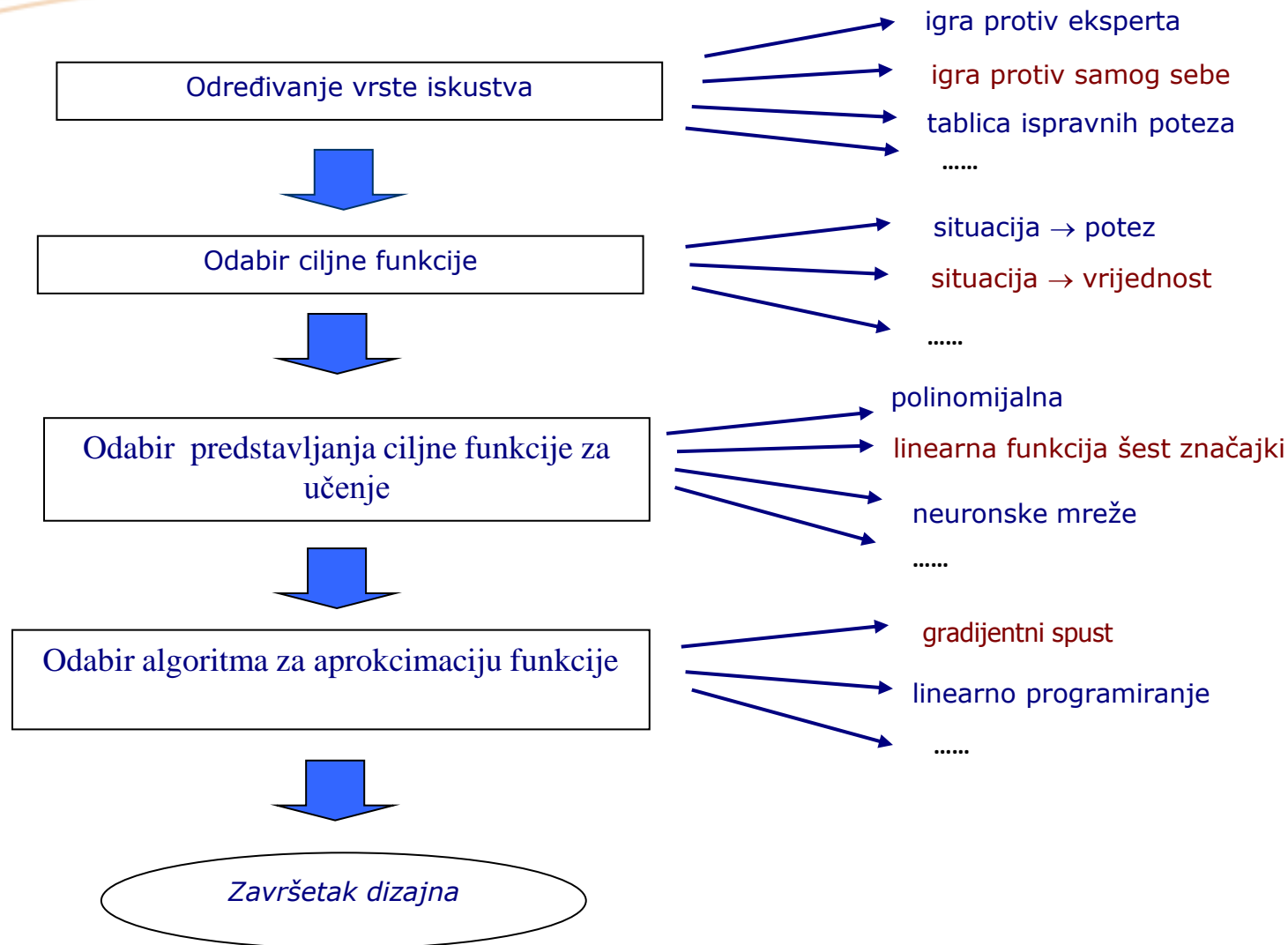
$$\mathbf{V}_{\text{train}}(\mathbf{b}) \leftarrow \hat{\mathbf{V}}(\text{sljedbenik}(\mathbf{b}))$$

- **Modul za generalizaciju** (*engl. Generalizer*)
 - *Ulaz:* primjeri za učenje
 - *Izlaz:* hipoteza – procjena ciljne funkcije \hat{V} .
- Modul poopćuje sa pojedinačnih slučajeva (primjeri za učenje) postavljajući hipotezu o općenitoj funkciji koja pokriva cijeli skup primjera za učenje i iznad njih.
- U našem slučaju to je LMS algoritam, a izlazna hipoteza je funkcija \hat{V} zadana sa $w_0..w_6$.

- **Generator pokusa** (*engl. **Experiment Generator***)
- *Ulaz:* trenutna hipoteza (trenutno naučena funkcija cilja)
- *Izlaz:* novi problem (npr. novo početno stanje ploče) koji će maksimizirati učenje cijelog sustava.
- U našem primjeru: *početno stanje na ploči.*

- Izbori koje smo donijeli tijekom oblikovanja sustava za
- DAME čine poseban slučaj za:
 - modul igre,
 - modul ocjenjivač,
 - modul za generalizaciju i
 - generator pokusa.
- **Mnogi sustavi za strojno učenje mogu biti opisani na ovaj način.**

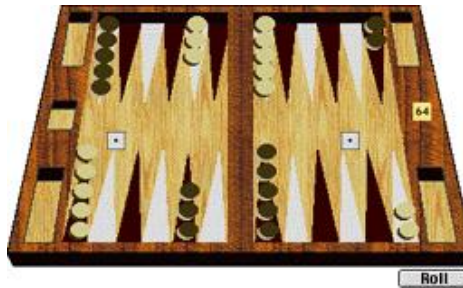
Slijed koraka u dizajnu sustava strojnog učenja



- Ako se stvarna ciljna funkcija može predstaviti kao lin. kombinacija ovih značajki, tada će ih naš program naučiti ili se barem možemo nadati dobroj aproksimaciji.
- Ako pretpostavimo da dobra aproksimacija prave funkcije V može biti predstavljena line. kombinacija, pitanje je da li ta tehnika garantira nalaženje funkcije V ?
- Da li program koji smo modelirali, može pobijediti svjetskog šampiona u igri DAME?

Problemi previše složeni da bi se riješili na uobičajeni način

- Učenje se primjenjuje svuda gdje je potrebno efikasno pretraživati veliki prostor stanja.
- **Učenje igranja backgammon-a na razini svjetskog prvaka**
- Najuspješnije implementacije igara na računalu temelje se na učenju.

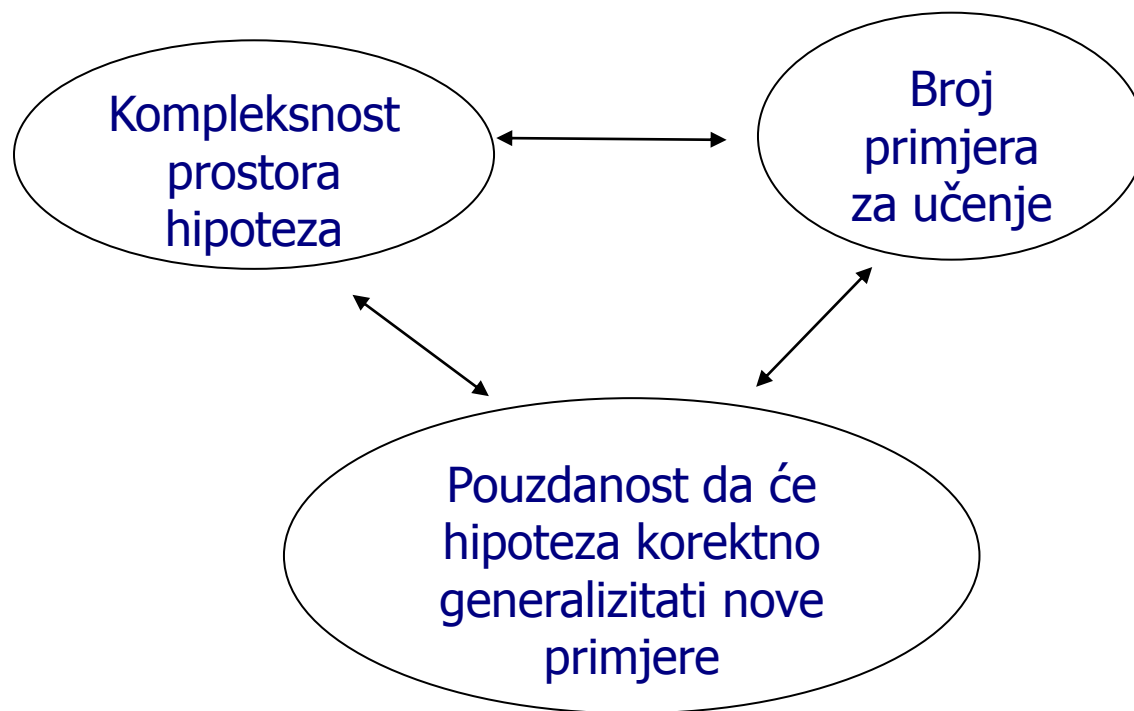


- Najbolji svjetski program TD-GAMMON (Tesauro, 1992, 1995) je naučio strategiju igranja (na svjetskoj razini) na temelju više od 10^6 odigranih partija protiv samog sebe.

- Pohranjivanje tipičnih situacija i poteza te nalaženje “najbliže” situacije koja odgovara novonastaloj →
algoritam k najbližih susjeda.
- Generiranje velikog broja programa i dozvoliti im da igraju jedni protiv drugih, zadržavajući samo one najbolje i pospješujući ih ili mutirajući ih →
evolucijski programi
- Oponašajući ljudski način igranja: analizirajući ili objašnjavajući sebi razloge određenih uspjeha ili neuspjeha →
učenje temeljeno na objašnjenima –
(*engl. explanation based learning*).

Jedan vrlo koristan pogled na strojno učenje

- Pretraživanje vrlo velikog prostora mogućih hipoteza i nalaženje one koja najbolje odgovara danim podacima i nekom apriornom znanju.
- Teorijska analize odnosa:



- Koji algoritmi postoje za učenje ciljnih funkcija na temelju primjera. U kojim okolnostima će dani algoritam konvergirati željenoj funkciji, ako je dano dovoljno podataka? Koji algoritmi odgovaraju najbolje kojim tipovima problema?
- Koliko podataka za učenje je dovoljno?
- Kada i kako može apriorno znanje koje ima učenik voditi proces generalizacije na temelju primjera?
- Koja je najbolja strategija odabira slijedećeg primjera za učenje i kako taj izbor utječe na kompleksnost problema učenja?

Teme u strojnom učenju (2)

- Koju funkciju treba sistem naučiti ? Može li se i taj problem automatizirati?
- Kako može učenik automatski promijeniti svoju reprezentaciju znanja da bi povećao učinak učenja?

- Implementirajte algoritam sličan ovome koji je iznesen, ali za jednostavniju igru TIC-TAC-TOE.
- Predstavite funkciju \hat{V} koja se uči, kao linearnu kombinaciju značajki stanja igre po vlastitom izboru.
- Da bi program naučio neka igra protiv kopije samog sebe uz uporabu evaluacijske funkcije koju ste kreirali.
- Prikažite grafički postotak dobivenih igara vašeg sustava naspram broja odigranih partija.