

# Strojno učenje Hamiltoniana na konačnoj 2D rešetki

Antun Magdić\*

Fizički odsjek, Prirodoslovno-matematički fakultet, Sveučilište u Zagrebu

mentor: akademik Hrvoje Buljan

Opisana je metoda određivanja Hamiltoniana u aproksimaciji čvrste veze na kvadratnoj dvodimenzionalnoj rešetki iz zadanog spektra. Predložena metoda koristi strojno učenje za određivanje elemenata preskoka u Hamiltonianu. U slučajevima u kojima postoji rješenje za zadani spektar metoda radi vrlo dobro. Ako rješenja nema, metoda će dati Hamiltonian čiji spektar aproksimira zadani spektar. Hamiltonian je moguće parametrizirati po želji manjim brojem parametara od svih elemenata preskoka.

## I. UVOD

Uobičajeni problem određivanja spektra Hamiltoniana konačno dimenzionalnog kvantnog sustava ima jednostavno rješenje. Hamiltonian se napiše u obliku matrice u nekoj bazi i tu je matricu onda lako dijagonalizirati (pogotovo za sustave malih dimenzija). Inverzni problem, odnosno pronalaženje Hamiltoniana iz danog spektra, je zahtjevniji. U ovom radu bavimo se upravo tim problemom, i to za Hamiltoniane u aproksimaciji čvrste veze [1] na konačnim kvadratnim rešetkama u dvije prostorne dimenzije. Za rješavanje problema koristimo strojno učenje, odnosno neuronske mreže. Sličan problem obrađen je u [2], s razlikom da su tamo konačne rešetke imale jednu prostornu dimenziju.

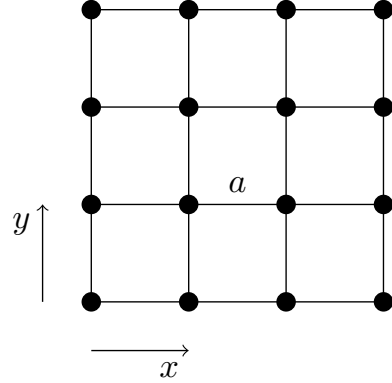
Problem je u jednoj dimenziji nešto jednostavniji i predstavlja tipičan primjer problema koji se može riješiti putem nadziranog strojnog učenja [3]. U dvije dimenzije problem je složeniji pa naivni pristup kao u jednoj dimenziji ne daje dobro rješenje. Stoga je bilo potrebno razviti složeniji pristup arhitekturi modela (neuronske mreže) i postupku učenja, što je i glavni doprinos opisan u ovom radu.

Kako vrijeme odmiče i postupci strojnog učenja napreduju, strojno učenje se sve češće pojavljuje u fizici. Dobar pregled raznih primjena dubokog učenja u području fotonike može se pronaći u [4].

U ovom radu u odjeljku II precizno je opisan problem koji rješavamo, u odjeljku III opisano je predloženo rješenje problema, a u odjeljku IV prikazani su rezultati predloženog rješenja na oglednom primjeru problema ( $4 \times 4$  rešetka).

## II. OPIS PROBLEMA

U ovom radu bavimo se konačnim dvodimenzionalnim kvadratnim rešetkama  $\{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$  (slika 1). Svakoj točki rešetke  $(x, y)$  pridruženo je stanje  $|x, y\rangle$ . Na tom skupu stanja definiran je Hamiltonian  $H$



Slika 1: Primjer dvodimenzionalne rešetke veličine  $4 \times 4$ .

uobičajen u aproksimaciji čvrste veze

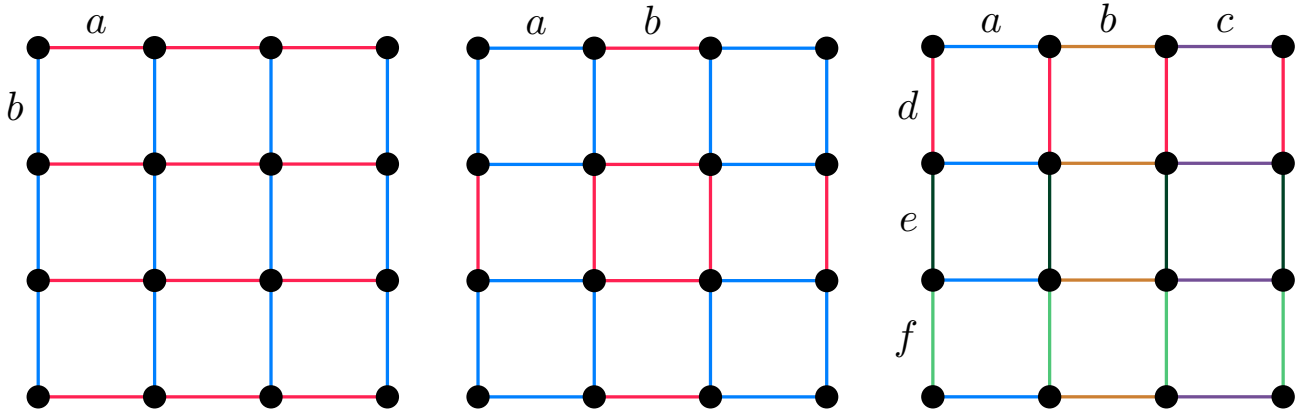
$$H = \sum_{\langle (x,y), (\xi,\eta) \rangle} t_{\{(x,y), (\xi,\eta)\}} (|\xi, \eta\rangle \langle x, y| + |x, y\rangle \langle \xi, \eta|) \quad (1)$$

gdje suma ide po najbližim susjedima (na slici 1 najbliži susjedi su povezani), a  $t_{\{(x,y), (\xi,\eta)\}}$  je element preskoka između stanja  $|x, y\rangle$  i  $|\xi, \eta\rangle$  (primjerice, na slici 1 s  $a$  je označen element preskoka  $t_{\{(2,2), (3,2)\}}$ ). Hamiltonian smo definirali tako da vrijedi  $\langle x, y| H |x, y\rangle = 0$ .

Označimo sa  $\sigma(H)$  spektar Hamiltoniana, uz napomenu da se radi o skupu  $m \cdot n$  realnih brojeva. U ovom radu rješavamo sljedeći problem. Za unaprijed zadani skup  $m \cdot n$  realnih brojeva  $S$  želimo pronaći Hamiltonian  $H$  tako da je  $\sigma(H) = S$ . U našem slučaju pronalaženje Hamiltoniana znači pronalaženje elemenata preskoka  $t$  u (1).

Ovakva definicija problema može izgledati problematično iz dva razloga. Prvo, može se dogoditi da postoje (i često je slučaj) dva različita Hamiltoniana  $H$  i  $H'$  tako da vrijedi  $\sigma(H) = \sigma(H') = S$ . To, međutim, ne predstavlja velik problem jer ćemo se zadovoljiti bilo kojim Hamiltonianom koji reproducira željeni spektar. Drugo, može se dogoditi da za zadani  $S$  ne postoji nijedan  $H$  tako da je  $\sigma(H) = S$ . Ovo predstavlja mnogo veći problem, međutim, u tom slučaju rješenja jednostavno nema i pokušat ćemo pronaći  $H$  tako da  $\sigma(H)$  aproksimira  $S$  što je bolje moguće (ovo nećemo definirati

\* amagdic.phy@pmf.hr



Slika 2: Različite parametrizacije Hamiltoniana na  $4 \times 4$  rešetki. Iste boje označavaju iste vrijednosti elemenata preskoka.

preciznije budući da se zapravo radi o slučaju kad nema rješenja).

Pronalaženje svih elemenata preskoka  $t$  često nije preterano zanimljivo jer se za  $m \times n$  rešetku radi o  $2mn - 2m - 2n$  parametara. Zanimljivije je parametrizirati Hamiltonian manjim brojem parametara, odnosno uzeti da su određeni elementi preskoka jednaki. Nekoliko takvih parametrizacija prikazano je na slici 2. Parametrizacije lijevo i u sredini su dvije različite parametrizacije s dva parametra, a desno je parametrizacija sa šest parametara.

### III. METODA

Opisani problem rješavamo metodama strojnog učenja [3, 5]. Konkretno, želimo naučiti neuronsku mrežu  $\hat{\tau}$  da za ulazne spektre  $s$  na izlazu daje vektor vrijednosti parametara Hamiltoniana  $\hat{t}$ . Budući da veličina spektra  $s$  (broj energija) ovisi o rešetki, za svaku rešetku moramo naučiti posebnu mrežu. Isto tako, broj parametara u vektoru  $\hat{t}$  ovisi o parametrizaciji Hamiltoniana (i, naravno, o rešetki) pa i za svaku parametrizaciju moramo naučiti posebnu mrežu.

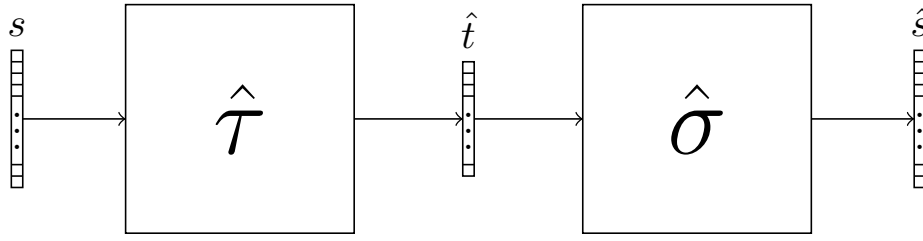
Kao što je već spomenuto, više različitih Hamiltoniana može imati isti spektar. Zbog toga ne možemo mrežu  $\hat{\tau}$  učiti jednostavnim postupkom nadziranog učenja, gdje bismo generirali skup podataka  $(s_i, t_i)$ . Moglo bi se dogoditi da u skupu podataka za učenje imamo parove  $(s, t_1)$  i  $(s, t_2)$  te da vrijedi  $t_1 \neq t_2$ . To bi bila kontradiktorna informacija u skupu za učenje i učenje ne bi uspjelo. Ovdje se ne radi o situaciji koja je česta u strojnom učenju gdje se u skupu za učenje mogu pojaviti pogrešno označeni podaci gdje mreža može učiti dokle god je krivo označenih podataka dovoljno malo. I  $(s, t_1)$  i  $(s, t_2)$  su ispravni podaci i prilikom učenja smanjenjem pogreške na jednom primjeru, povećala bi se pogreška na drugom i zbog toga mreža ne bi mogla učiti.

Ovaj problem možemo riješiti na sljedeći način. Cilj cijelog postupka je pronaći parametre Hamiltoniana čiji

je spektar što sličniji ulaznom spektru. Stoga za funkciju pogreške nećemo uzeti funkciju koja uspoređuje predviđeni  $\hat{t}_i = \hat{\tau}(s_i)$  s nekim unaprijed zadanim  $t_i$ , već funkciju koja uspoređuje  $\hat{s}_i = \sigma(\hat{t}_i) = \sigma(\hat{\tau}(s_i))$  sa  $s_i$ , gdje  $\sigma(\hat{t}_i)$  označava spektar Hamiltoniana s parametrima  $\hat{t}_i$ . Međutim, ovo je problematično iz dva razloga. Funkcija  $\sigma$  (određivanje spektra matrice) se u većini slučajeva na računaru računa iterativnim metodama i nije ju lako derivirati, a u strojnom učenju želimo da se cijela funkcija pogreške može derivirati u zatvorenoj formi. S praktične strane, funkcija  $\sigma$  u većini slučajeva nije dovoljno brza da bi bila korisna prilikom učenja mreže. Da bismo riješili ova dva problema, možemo učiniti sljedeće: prvo naučimo drugu mrežu  $\hat{\sigma}$  da računa funkciju  $\sigma$  i onda tu mrežu iskoristimo u funkciji pogreške prilikom učenja mreže  $\hat{\tau}$ .

Cijela arhitektura je shematski prikazana na slici 3. U prvom koraku generiramo skup podataka za učenje  $\mathcal{D}_\sigma = \{(t_i, s_i)\}$  koji koristimo za učenje mreže  $\hat{\sigma}$ . Ovo se može učiniti klasičnim pristupom nadziranog učenja, a za funkciju pogreške može se uzeti srednja kvadratna pogreška između  $s_i$  i  $\hat{\sigma}(t_i)$  (da bi ovo radilo, svaki spektar u skupu za učenje bi trebalo sortirati prilikom generiranja skupa podataka za učenje). U drugom koraku učimo mrežu  $\hat{\tau}$  na skupu podataka  $\mathcal{D}_\tau = \{s'_i\}$  u kojem se nalaze samo spektri. Funkcija pogreške (srednja kvadratna pogreška ili neka druga slična funkcija) uspoređuje  $s'_i$  sa  $\hat{\sigma}(\hat{\tau}(s'_i))$ .

Ostaje još pitanje kako generirati skupove za učenje  $\mathcal{D}_\sigma$  i  $\mathcal{D}_\tau$ . Skup  $\mathcal{D}_\sigma$  generiramo tako da generiramo nasumične vrijednosti parametara Hamiltoniana i zatim dijagonalizacijom dobijemo odgovarajuće spektre. Skup  $\mathcal{D}_\tau$  mogli bismo dobiti tako da iz skupa  $\mathcal{D}_\sigma$  zadržimo samo spektre. Na taj način mreža bi odlično naučila određivati parametre za spektre koji se stvarno mogu dobiti koristeći određenu parametrizaciju. U slučaju da se spektar ne može dobiti u parametrizaciji za koju je mreža naučena, parametri koje određuje mreža dat će spektar koji ne mora uopće biti sličan ulaznom spektru. To je i očekivano jer mreža takve spektre tijekom učenja nikad nije vidjela. Zato je u skup  $\mathcal{D}_\tau$  najbolje jednostavno staviti bilo koje



Slika 3: Shematski prikaz neuronske mreže prilikom učenja.

nasumične spektre. Naravno, za neke od njih mreža neće moći odrediti dobre parametre (jer ne postoje), ali za sve će naučiti određivati parametre tako da se spektar dobiven iz parametara koje odredi mreža poklapa s ulaznim spektrom što je bolje moguće.

Neke dodatne korisne informacije vezane uz učenje mreže i generiranje podataka mogu se pronaći u dodatku A.

Neuronska mreža opisana u ovom odjeljku može se također promatrati i kao vrsta autoenkodera [5]. Više o tome može se pronaći u dodatku B.

#### IV. REZULTATI

Opisana metoda primijenjena je na Hamiltonian na  $4 \times 4$  rešetki s parametrizacijom kao na slici 2 desno. Za dani spektar  $s$  mreža treba odrediti 6 parametara  $\hat{t}$  (na slici  $a, b, c, d, e$  i  $f$ ). Iz dobivenih parametara, odnosno Hamiltoniana, možemo odrediti spektar  $\sigma(H(\hat{t}))$ . Da bismo ispitali uspješnost opisane metode, generiramo nasumične spektre i uspoređujemo ih sa spektrima Hamiltoniana prilagođenog ulaznim spektrima. Neki od rezultata prikazani su na slici 4. Gornja dva spektra daju odlične rezultate iz čega zaključujemo da je odabrana parametrizacija za njih dobra.

Donja dva rezultata su nešto lošija, u slučaju na slici lijevo odstupanja nisu prevelika, ali kod desnog su za neke energije odstupanja velika, u apsolutnom i relativnom iznosu. Očito je da spektar pronađenog Hamiltoniana ne odgovara dobro zadanom spektru, međutim, teško je točno odrediti koji je razlog i kako to popraviti, ako je uopće moguće. Naravno, moguće je da mreža jednostavno nije dobro naučena. Ipak, mreža za spektre iz

skupa  $\mathcal{D}_\sigma$  radi mnogo bolje nego za one iz skupa  $\mathcal{D}_\tau$  pa je izgledno da je razlog odstupanja ulazni spektar koji se ne može reproducirati parametrizacijom koju smo izabrali (a pitanje je može li se reproducirati bilo kojom parametrizacijom).

Na skupu podataka za testiranje u kojem su spektri koje se može reproducirati korištenom parametrizacijom postiže se srednja pogreška 0.006 (korištena je srednja kvadratna pogreška). Na skupu u kojem se nalaze proizvoljni spektri pogreška je 0.05. Nažalost, kao što smo vidjeli u primjerima, nije moguće odrediti je li uzrok pogreške zadani spektar za koji nema dobrog rješenja ili neuronska mreža koja ne može pronaći dobro rješenje.

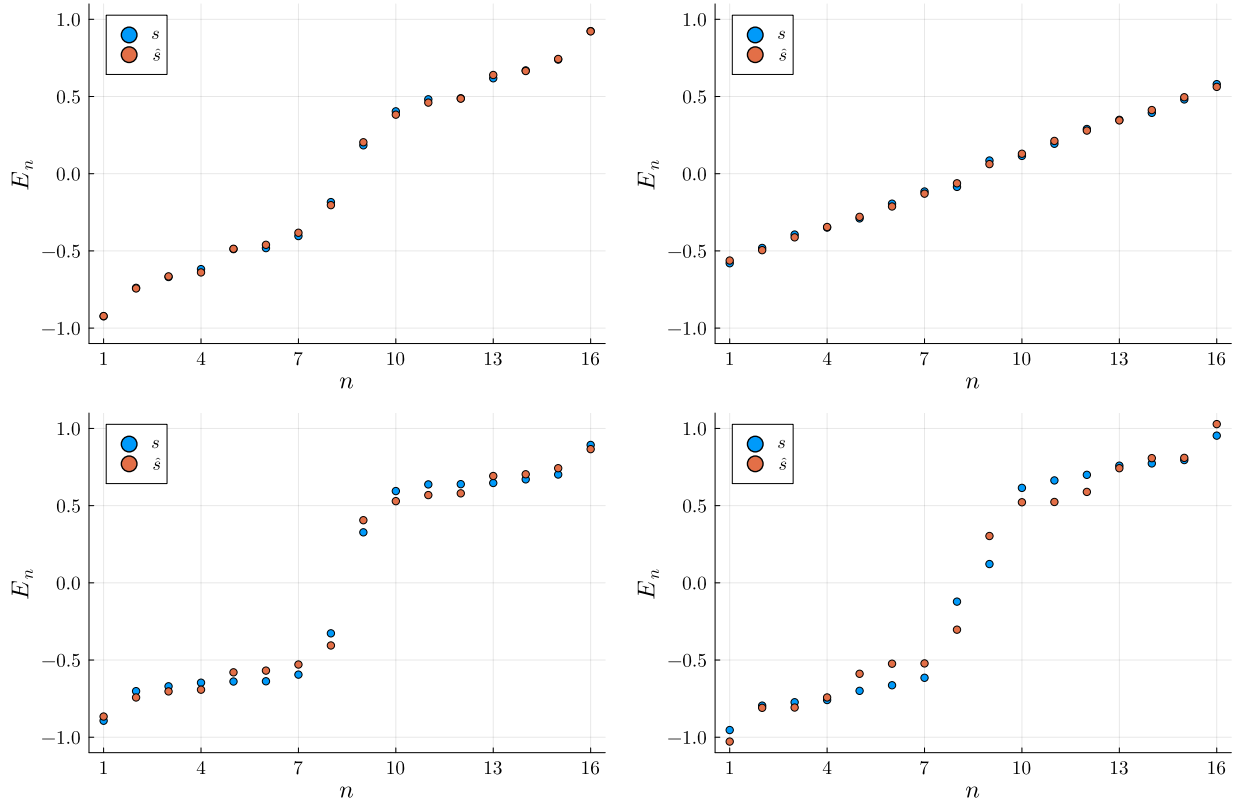
S drugim parametrizacijama dobivaju se slični rezultati, no jasno je da se parametrizacija s manje parametara može slabije prilagoditi zadanim spektrima. Odabir parametrizacije najviše ovisi o kontekstu u kojem bi se ova metoda određivanja Hamiltoniana koristila.

#### V. ZAKLJUČAK

U ovom radu opisana je metoda kojom se može odrediti Hamiltonian na temelju zadanog spektra. Naravno, taj Hamiltonian nije jedinstven s obzirom na to da više različitih Hamiltoniana može imati isti spektar. U situacijama u kojima rješenje postoji, metoda radi dobro. Kad nema rješenja, metoda ga naravno ne može pronaći, međutim, teško je procijeniti je li problem u odabranoj parametrizaciji Hamiltoniana ili u zadanom spektru koji se ne bi mogao reproducirati u bilo kojoj parametrizaciji. Isto tako, čak ako se spektar može reproducirati u nekoj složenoj parametrizaciji, takve složene parametrizacije s velikim brojem parametara uglavnom nisu zanimljive, a i teže su za učenje.

- 
- [1] N. W. Ashcroft and N. D. Mermin, *Solid State Physics* (Holt-Saunders, 1976).
  - [2] S. Xia, S. Lei, D. Song, L. D. Lauro, I. Alamgir, L. Tang, J. Xu, R. Morandotti, H. Buljan, and Z. Chen, Deep learning empowered synthetic dimension dynamics, in *CLEO 2024* (Optica Publishing Group, 2024) p. FW3Q.2.
  - [3] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. (Springer,

- 2007).
- [4] W. Ma, Z. Liu, Z. Kudyshev, A. Boltasseva, W. Cai, and Y. Liu, Deep learning for the design of photonic structures, *Nature Photonics* **15**, 1 (2020).
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) book in preparation for MIT Press.
- [6] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *CoRR* **abs/1412.6980** (2014).



Slika 4: Zadani spektri  $s$  i spektri  $\hat{s}$  Hamiltoniana  $H(\hat{\tau}(s))$ .

#### Dodatak A: Korisni savjeti za učenje i generiranje podataka

Prvo ćemo dokazati jedan koristan teorem o spektru Hamiltoniana u aproksimaciji čvrste veze na konačnoj rešetki za koji vrijedi  $\langle x, y | H | x, y \rangle = 0$ .

**Teorem 1.** *Neka je  $H$  Hamiltonian (1). Ako je  $E$  svojstvena vrijednost od  $H$ , onda je i  $-E$  svojstvena vrijednost od  $H$ .*

*Dokaz.* Podijelit ćemo vektore baze  $|x, y\rangle$  u dva skupa (vidi sliku 5):

$$A = \{|x, y\rangle : x + y \text{ paran}\},$$

$$B = \{|x, y\rangle : x + y \text{ neparan}\}.$$

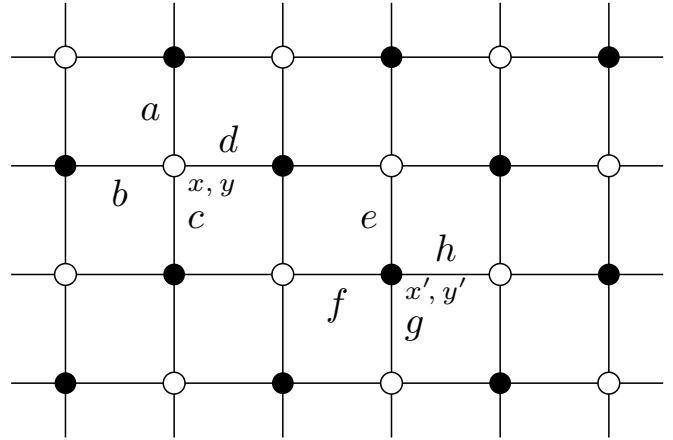
Vidimo da je  $\langle \alpha | H | \beta \rangle$  za sve  $|\alpha\rangle \in A$ ,  $|\beta\rangle \in B$ . Neka je  $|\psi\rangle$  svojstveni vektor od  $H$  sa svojstvenom vrijednosti  $E$ . Možemo ga napisati u bazi  $|x, y\rangle$  kao

$$|\psi\rangle = \sum_{x+y \text{ paran}} u_{x,y} |x, y\rangle + \sum_{x+y \text{ neparan}} v_{x,y} |x, y\rangle.$$

Jednadžba  $H|\psi\rangle = E|\psi\rangle$  vodi na jednadžbe za komponente  $u_{x,y}$  i  $v_{x,y}$  oblika (vidi sliku 5)

$$av_{x,y+1} + bv_{x-1,y} + cv_{x,y-1} + dv_{x+1,y} = Eu_{x,y}$$

$$eu_{x',y'+1} + fu_{x'-1,y'} + gu_{x',y'-1} + hu_{x'+1,y'} = Ev_{x',y'}$$



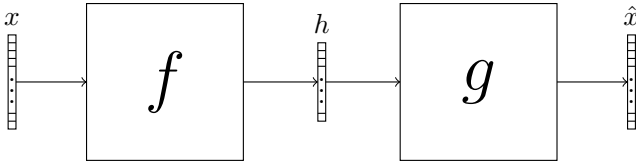
Slika 5: Točke rešetke podijeljene u dva skupa.

(za  $(x, y)$  ili  $(x', y')$  na rubu rešetke na lijevoj strani nekih članova nema). Promjenom  $E$  u  $-E$  i  $v_{x,y}$  u  $-v_{x,y}$ , jednadžbe ostaju zadovoljene pa je

$$|\phi\rangle = \sum_{x+y \text{ paran}} u_{x,y} |x, y\rangle - \sum_{x+y \text{ neparan}} v_{x,y} |x, y\rangle.$$

svojstveni vektor od  $H$  sa svojstvenom vrijednosti  $-E$ , odnosno i  $-E$  je svojstvena vrijednost od  $H$ .  $\square$

Ovaj teorem nam pokazuje da se spektar na  $m \times n$



Slika 6: Autoenkoder.

rešetki može zadati s  $\lfloor mn/2 \rfloor$  realnih brojeva. Zato spektar u neuronskim mrežama prikazujemo kao vektor s  $\lfloor mn/2 \rfloor$  elemenata. U njega stavljamo samo pozitivne energije, negativne onda automatski slijede. Da bismo izbjegli komplikacije prilikom uspoređivanja spektara, taj je vektor uvijek sortiran.

Prilikom učenja za funkciju pogreške između zadanog spektra  $s$  i predviđenog spektra  $\hat{s}$  uzimamo

$$l(s, \hat{s}) = \frac{1}{(\max_i s_i)^2} \text{mse}(s, \hat{s}),$$

gdje je mse srednja kvadratna pogreška. Dijelimo s najvećom energijom, kako bismo izbjegli situaciju u kojoj spektri s većim apsolutnim vrijednostima energija imaju veću težinu pri izračunu pogreške (mogli smo dijeliti i sa srednjom vrijednosti energije ili nečim sličnim, ali ovako je brže i jednostavnije, a postiže se isti učinak). Iako možda imamo namjeru koristiti mrežu samo na jednoj skali energije i mislimo da je bitno da radi dobro samo na toj skali, prilikom učenja mreža  $\hat{\tau}$  je nasumično inicijalizirana i može na izlazu dati parametre Hamiltoniana čije su energije jako male, a  $\hat{\sigma}$  mora dobro raditi i za njih kako bi učenje moglo dobro teći. Stoga je bitno da mreža  $\hat{\sigma}$  uči dobro na svim skalama, a iz istog razloga je bitno da su u skupu za učenje zastupljeni spektri na svim skalama (naravno, na svim znači u širokom rasponu).

Za učenje je korišten Adam [6], sa stopama učenja između  $10^{-3}$  i  $10^{-5}$ . Stopa učenja mijenjana je ručno,

ovisno o tijeku učenja.

Veličine mreža određene su za pojedine rešetke na temelju pokušaja i pogrešaka. U primjeru iz odjeljka IV ( $4 \times 4$  rešetka) mreža  $\hat{\sigma}$  ima veličine skrivenih slojeva 32, 32, 32, 16, a veličine skrivenih slojeva mreže  $\hat{\tau}$  su 64, 128, 128, 128, 64, 32, 32.

Nije na odmet napomenuti da s prenaučenošću uglavnom nema problema, budući da možemo generirati podataka za učenje koliko god je potrebno. Konkretno u opisanom primjeru  $4 \times 4$  rešetke, skup za učenje imao je 100000 primjera, a učenje je trajalo nekoliko tisuća epoha.

#### Dodatak B: Opisana neuronska mreža kao autoenkoder

Autoenkoder [5] je neuronska mreža koja se uglavnom koristi za učenje nižedimenzionalnih reprezentacija podataka. Autoenkoder je shematski prikazan na slici 6.  $f$  i  $g$  su neuronske mreže.  $f$  na ulazu dobiva neki vektor  $x$  koji zatim preslikava u vektor  $h$  (uobičajeno manje dimenzije od  $x$ ). Zatim  $g$  treba rekonstruirati  $x$  iz  $h$ , odnosno, preslikava  $h$  u  $\hat{x}$ . Ako je ova rekonstrukcija dobra, onda je reprezentacija vektora  $x$  vektorom  $h$  dobra, odnosno, mreža je naučila nižedimenzionalnu reprezentaciju ulaznog vektora. Možemo očekivati da su u vektoru  $h$  sadržane sve bitne značajke vektora  $x$ . Uobičajeno se  $f$  naziva koder, a  $g$  dekoder.

Usporedbom sa slikom 3 vidimo da naša mreža  $\hat{\tau}$  odgovara mreži autoenkodera  $f$ , a mreža  $\hat{\sigma}$  odgovara mreži  $g$ . Međutim, naš model nije autoenkoder. U autoenkoderu se zajedno uče i  $f$  i  $g$  jer je cilj da model pronađe neku (bilo koju) reprezentaciju. Mi prvo učimo odvojeno  $\hat{\sigma}$ , a zatim  $\hat{\tau}$  s naučenim  $\hat{\sigma}$  i na taj način forsiramo određenu reprezentaciju – onu u kojoj  $h$ , odnosno u našoj mreži  $\hat{t}$ , predstavlja elemente preskoka u modelu čvrste veze. Tu reprezentaciju zadajemo skupom podataka na kojem učimo mrežu  $\hat{\sigma}$ .