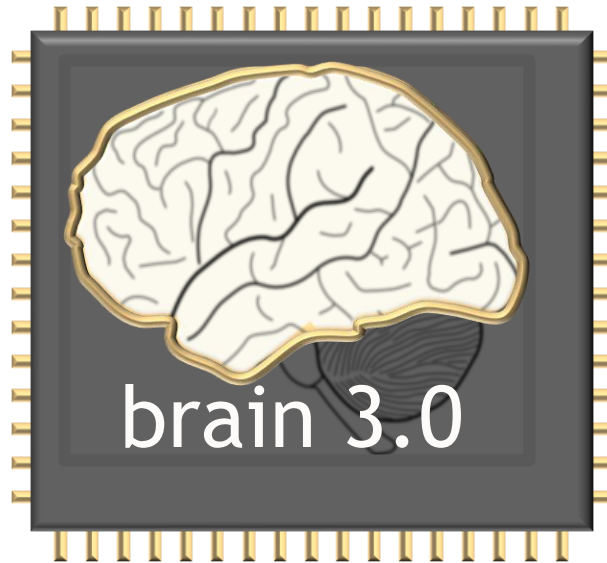


Nevromorfno računalništvo

Neuromorphic computing



Naši možgani so čudoviti stroj

Možgani in računalnik

NEVRONI: prednost v prostoru

Nevroni v človeških možganih tvorijo do 10^5 povezav s svojimi sosedi.

El. moč: 10 W,
 10^{10} nevronov, 10^{13}
sinaps,
50000 nevronov/ mm^3

EL. VEZJA: prednost v času

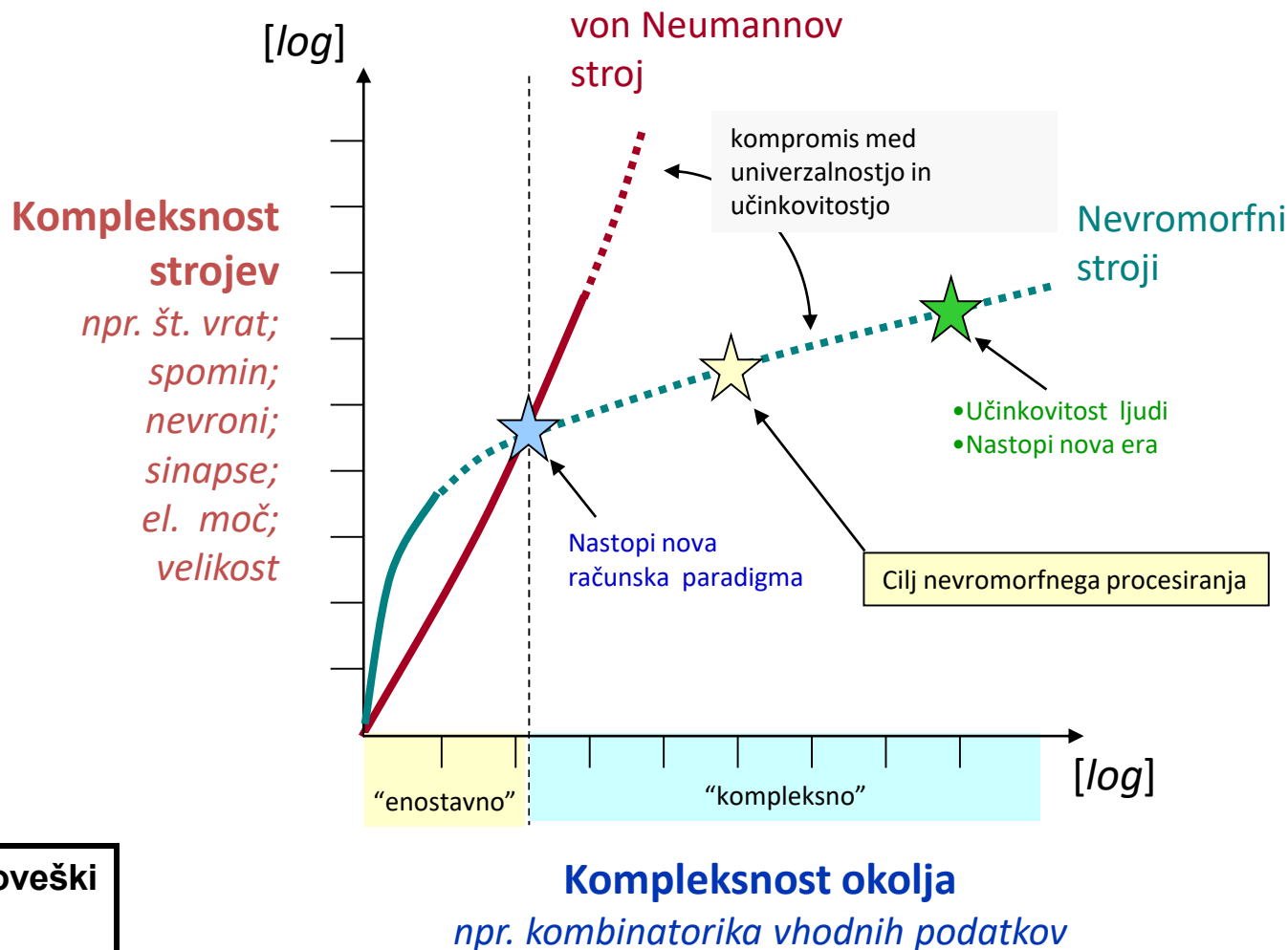
Integrirana vezja opravijo komunikacijski cikel 10^6 x hitreje kot nevroni.

Pentium 4:
El. moč: 40 W,
3 GHz CPE,
 42×10^6 tranzistorjev

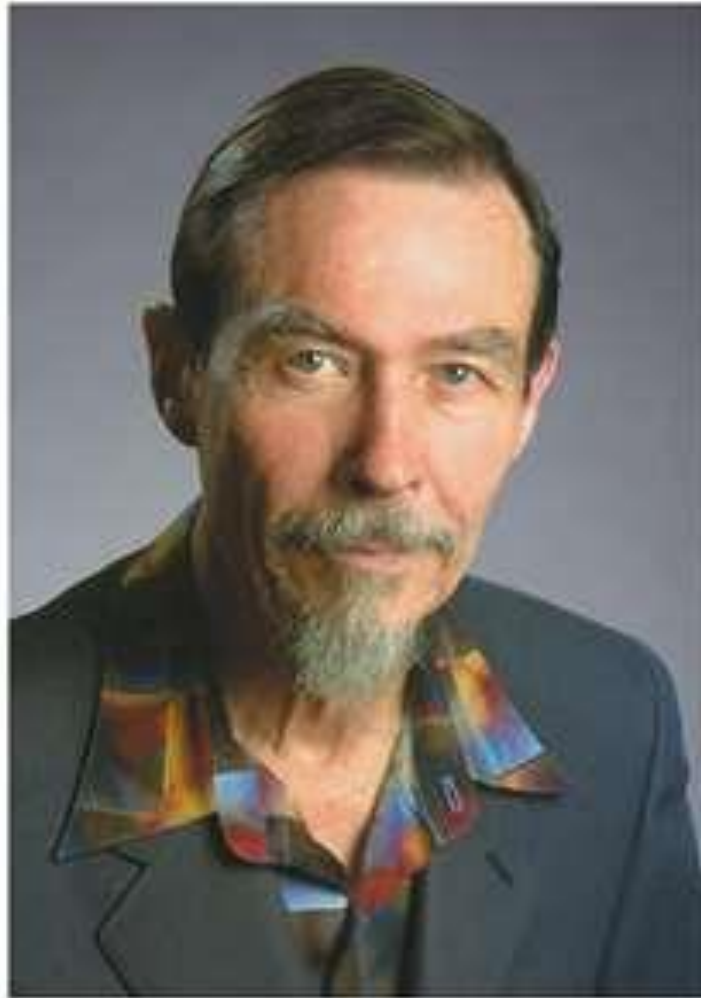
Možgani in računalnik

Izzivi:

- V primerjavi z biološkimi sistemi so današnji inteligentni stroji več kot milijonkrat manj učinkoviti v kompleksnih okoljih.
- Inteligentni stroji bodo resnično uporabni šele ko bodo tekmovali s človekom.



Človeški korteks	Simuliran človeški korteks
15 W	10^{10} W
1 liter	4×10^{10} litrov



Carver Andress Mead

Neuromorphic Electronic Systems

CARVER MEAD

Invited Paper

“The fact that we can build devices that implement the same basic operations as those the nervous system uses leads to the inevitable conclusion that we should be able to build entire systems based on the organizing principles used by the nervous system. I will refer to these systems generically as *neuromorphic systems*”.

Carver Mead, 1990

Nevromorfno računanje

Uporaba bioloških principov

=> konstrukcija bolj učinkovitih strojev

Konstrukcija nevromorfnih strojev

=> pridobljeno znanje o bioloških principih

Projekt FACETS



MoNETA: A Mind Made from Memristors

<https://spectrum.ieee.org/robotics/artificial-intelligence/moneta-a-mind-made-from-memristors>

Nevromorfne arhitekture:

“The great BRAIN race”

From BrainScales to Human Brain Project: Neuromorphic Computing Coming of Age

<https://www.youtube.com/watch?v=g-ybKtY1quU>

Growing Number of NM Projects in the EU and the US

The Five Complementary Approaches to Neuromorphic Computing

- Commodity microprocessors (SpiNNaker, HBP)
- Custom fully digital (IBM Almaden)
- Custom Mixed-Signal (BrainScaleS, HBP)
- Custom subthreshold analog cells (Stanford, ETHZ)
- Custom Hybrid (Qualcomm)

Soft-binary-model
Hard-binary-model
Physical model (accelerated)
Physical model (real time)
Hybrid NM-traditional

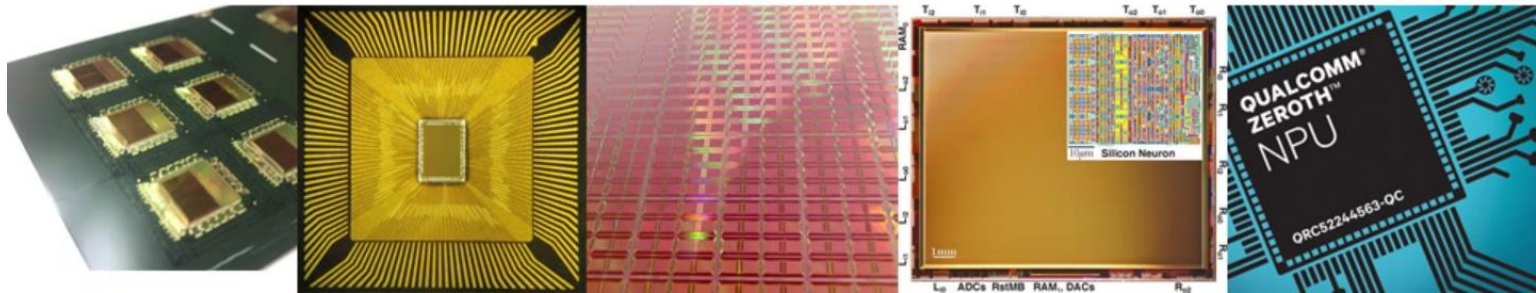
Anything in common ?

Massively parallel

Asynchronous communication

Configurability

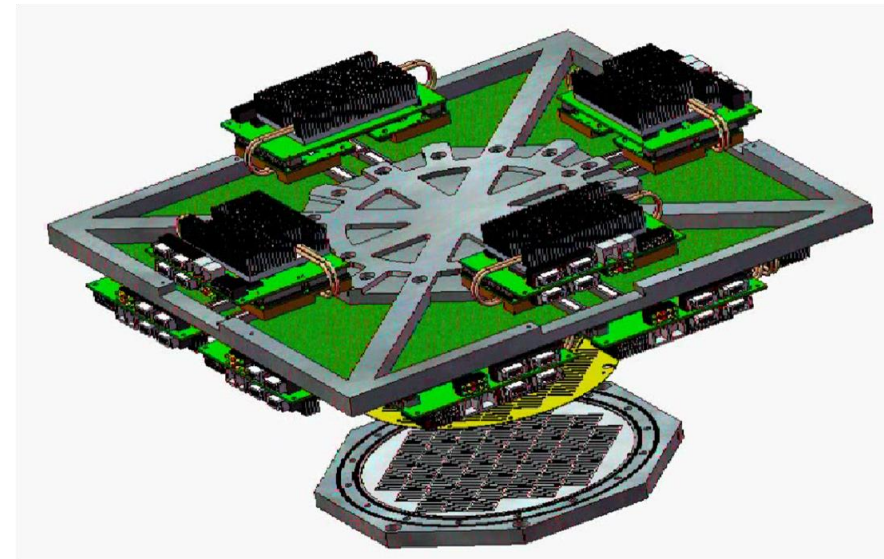
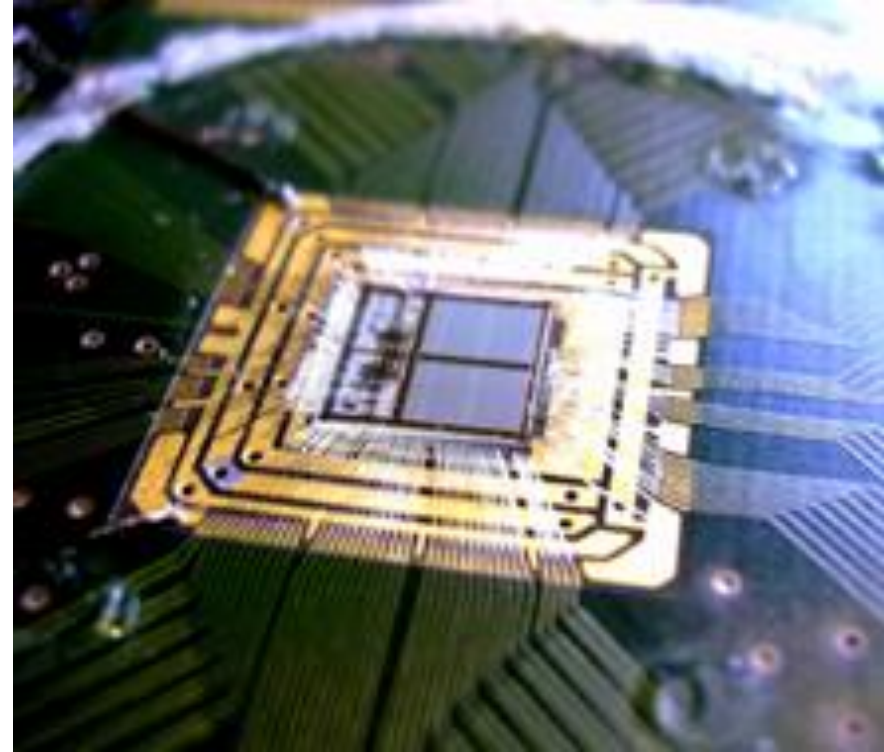
COMPLEMENTARITY OF APPROACHES ESSENTIAL !



Projekt FACETS

Fast Analog Computing with Emergent Transient States (FACETS)

- mednarodni projekt znanstvenikov in inženirjev, financiran s strani EU.
- Zadnji razvit čip vsebuje 200,000 nevronov (t.j. nevronske vezij), ki so med sabo povezani s 50 milijoni sinaps.
- Projekt zaključen 2010
- Delo se nadaljuje v okviru projekta BrainScale



Projekt FACETS

<http://facets.kip.uni-heidelberg.de/index.html>

Razvita strojna oprema: zelo veliko **VLSI nevronske vezje**, ki emulira signifikanten predel človeškega korteksa. Vezje temelji na novi računski paradigmi in izkorišča masovne **paralelne, kompleksne in dinamične povezave**, takšne kot jih zasledimo v človeških možganih.

Omenjena računska paradigma temelji na **podrobnih teoretičnih modelih strukture** (vzorci povezovanja, raznolikost nevronov) in **funkcionalnosti** (dinamična stanja aktivnosti, plastičnost) **človeškega korteksa**.

Strojna oprema izkazuje visoko stopnjo **skalabilnosti in nastavljenosti** in dovoljuje eksperimentalne študije nevrnskega obnašanja v različnih časovnih in prostorskih merilih (od posameznih proženj nevrona do plastičnosti nevrnskih vezij in dolgoročnega učenja). **Namenjena je predvsem raziskovanju v nevroznanosti in razvoju umetnih sistemov, ki bi odpravili potrebo po invazivnih študijah in vivo.**

http://facets.kip.uni-heidelberg.de/images/4/48/Public--FACETS_15879_Summary-flyer.pdf

Human Brain Project

<https://www.humanbrainproject.eu>

BrainScaleS

- Fizična analogno-digitalna emulacija nevronov, sinaps in plastičnosti živčevja z digitalnimi povezavami (ASIC implementacija)
- Vsak nevron ima adaptivno eksponentno dinamiko ([AdEx neuron dynamics](#))
- Sinapsa: 4-bitna ločljivost in princip delovanja STDP ([spike-timing-dependent plasticity](#))
- Prilagodljiva topologija mreže
- 10.000x hitrejša od bioloških nevronske mreže

SpiNNaker

- Večjederna ARM arhitektura
- 30.000 čipov z 18 jedri in 128 MB pomnilnika: skupaj 500.000 jeder

BrainScaleS

<https://electronicvisions.github.io/hbp-sp9-guidebook/pm/pm.html>

Vsak modul ima 384 medsebojno povezanih čipov **HICANNs**. Vsak izmed njih ima 114.688 programabilnih sinaps in 512 nevronov (skupaj 44×10^6 sinaps in 196.608 nevronov na modul).

Programiranje v Python-u, z orodjem za mapiranje nevronov [Marocco](#)

HBP Neuromorphic Computing Platform Guidebook

<https://electronicvisions.github.io/hbp-sp9-guidebook/index.html>

BrainScaleS

https://electronicvisions.github.io/hbp-sp9-guidebook/pm/using_pm_newflow.html#details-of-the-software-stack

```
neuron_count = 1 # size of the Population we will create

# Set the neuron model class
neuron_model = pynn EIF_cond_exp_isfa_ista # an Adaptive Exponential I&F Neuron

neuron_parameters = {
    'a'          : 4.0,      # adaptation variable a in nS
    'b'          : 0.0805,   # adaptation variable b in pA
    'cm'         : 0.281,    # membrane capacitance nF
    'delta_T'    : 1.0,      # delta_T fom Adex mod in mV, determines the sharpness of spike initiation
    'e_rev_E'    : 0.0,      # excitatory reversal potential in mV
    'e_rev_I'    : -80.0,    # inhibitory reversal potential in mV
    'i_offset'   : 0.0,      # offset current
    'tau_m'      : 9.3667,   # membrane time constant
    'tau_refrac' : 0.2,      # absolute refractory period
    'tau_syn_E'  : 20.0,     # excitatory synaptic time constant
    'tau_syn_I'  : 20.0,     # inhibitory synaptic time constant
    'tau_w'      : 144.0,    # adaptation time constant
    'v_reset'    : -70.6,    # reset potential in mV
    'v_rest'     : -70.6,    # resting potential in mV
    'v_spike'    : -40.0,    # spike detection voltage in mV
    'v_thresh'   : -50.4,    # spike initiaton threshold voltage in mV
}

# We create a Population with 1 neuron of our neuron model
N1 = pynn.Population(size=neuron_count, cellclass=neuron_model, cellparams=neuron_parameters)
```

```
#!/usr/bin/env python
```

```
import pyhmf as pynn
import Coordinate as C
from pymarocco import PyMarocco, Defects
from pymarocco.results import Marocco

import pylogging
for domain in ["Calibtic", "marocco"]:
    pylogging.set_loglevel(pylogging.get(domain), pylogging.LogLevel.INFO)
```

```
marocco = PyMarocco()
marocco.calib_backend = PyMarocco.CalibBackend.Default
marocco.defects.backend = Defects.Backend.None
marocco.persist = "results.bin"
pynn.setup(marocco = marocco)
```

```
pop = pynn.Population(1, pynn.IF_cond_exp)

marocco.manual_placement.on_hicann(pop, C.HICANNOnWafer(C.X(5), C.Y(5)), 4)
```

```
pynn.run(10)
pynn.end()
```

```
results = Marocco.from_file(marocco.persist)
```

```
for neuron in pop:
    for item in results.placement.find(neuron):
        for denmem in item.logical_neuron():
            print denmem
```


DARPA

SYNAPSE:

<https://www.darpa.mil/news-events/2014-08-07>

μBRAIN:

<https://www.darpa.mil/program/microbrain>

AI Next Campaign:

<https://www.darpa.mil/work-with-us/ai-next-campaign>

Strojna podpora umetni inteligenci:

Loihi – Intel

Loihi 1 - <https://en.wikichip.org/wiki/intel/loihi>

Loihi 2 - <https://www.intel.com/content/www/us/en/newsroom/news/intel-unveils-neuromorphic-loihi-2-lava-software.html>

<https://download.intel.com/newsroom/2021/new-technologies/neuromorphic-computing-loihi-2-brief.pdf>

Lava – <https://lava-nc.org/>
<https://github.com/lava-nc/lava>

Neural Fields: http://www.scholarpedia.org/article/Neural_fields

Strojna podpora umetni inteligenco na mobilnih platformah:

Apple - Neural engine: https://apple.fandom.com/wiki/Neural_Engine

„...a group of specialized cores functioning as a [neural processing unit](#) (NPU) dedicated to the acceleration of artificial intelligence operations and machine learning tasks...”

Google - Tensor Processing Unit: <https://cloud.google.com/tpu/docs/tpus>

„...A tensor processing unit (TPU) is an AI accelerator application-specific integrated circuit (ASIC) developed by Google specifically for neural network machine learning...”

Samsung - Neural Processing Unit (NPU):

<https://semiconductor.samsung.com/emea/insights/topic/ai/>

„...NPU is a processor that is optimized for deep learning algorithm computation, designed to efficiently process thousands of these computations simultaneously...”

Literatura

- [1] Neuromorphic, <<http://en.wikipedia.org/wiki/Neuromorphic>>.
- [2] Hammerstrom, D. "A Survey of Bio-Inspired and Other Alternative Architectures," in Waser, Rainer (ed.) Nanotechnology. Volume 4: Information technology II. Weinheim: Wiley-VCH, pp. 251-282, 2006.
- [3] Carver Mead, <http://en.wikipedia.org/wiki/Carver_Mead>
- [4] Holler, M., et al. "*An Electrically Trainable Artificial Neural Network (ETANN) with 10240 "Floating Gate" Synapses,*" *International Joint Conference on Neural Networks, 1989.*
- [5] Nestor, I., *Ni1000 Recognition Accelerator - Data Sheet, 1-7, 1996.*
- [6] Ramacher, U. et al. "SYNAPSE-1: a high-speed general purpose parallel neurocomputer system, " IPPS (774-781). 1995.
- [7] R. Serrano-Gotarredona, T. et al. "A Neuromorphic Cortical Layer Microchip for Spike Based Event Processing Vision Systems," IEEE Trans. on Circuits and Systems, Part-I. Vol. 53, No. 12, pp. 2548-2566, December 2006.
- [8] Serrano-Gotarredona, R., et al. "AER Building Blocks for Multi-Layer Multi-Chip Neuromorphic Vision Systems," , *Advances in Neural Information Processing Systems (NIPS)*, 18: 1217-1224, Dec, Y. Weiss and B. Schölkopf and J. Platt (Eds.), MIT Press, 2005
- [9] Brains in Silicon,<<http://www.stanford.edu/group/brainsinsilicon/index.html> >.
- [10] FACETS: Fast Analog Computing with Emergent Transient States, <<http://facets.kip.uni-heidelberg.de/index.html>>.
- [11] Graham-Rowe, D. "Building a Brain on a Silicon Chip," in Technology Review, March 25, 2009. [Online]. Available: <<http://www.technologyreview.com/computing/22339/page1/>>. [Accessed March 28, 2009].
- [12] C. Torres-Huitzil, et. al. "On-chip Visual Perception of Motion: A Bio-inspired Connectionist Model on FPGA, " *Neural Networks Journal*, 18(5-6):557-565, 2005.

Umetne nevronske mreže

(Artificial Neural Networks)

Knjiga: Martin T. Hagan, Howard B. Demuth, Mark H. Beale: *Neural Network Design*, 2002 <https://hagan.okstate.edu/NNDesign.pdf>

Algoritmi učenje NM (*learning rule*)

postopek spreminjanja uteži (*weights*) in praga (*bias*).

Nadzorovano učenje

(supervised
learning)

Učno pravilo je določeno z učno množico vhodnih in izhodnih vrednosti :

$$\{\mathbf{p}_1, \mathbf{t}_1\} \{\mathbf{p}_2, \mathbf{t}_2\} \dots \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

kjer je \mathbf{p} vektor vseh vhodov v nevronske mreže in \mathbf{t} vektor ustreznih pravih izhodov (**target**).

Vhodni vektorji postopno vstopajo v nevronske mreže, njeni izhodi pa se primerjajo s pričakovanimi izhodi \mathbf{t} . Uteži in pragovi se nastavijo tako da se minimizira napaka med izhodi nevronske mreže in pričakovanimi izhodi \mathbf{t} .

Učenje z ojačitvijo

(reinforcement
learning)

je podobno nadzorovanemu učenju, le da nimamo podanih pričakovanih izhodnih vrednosti \mathbf{t} , temveč je dana cenitvena funkcija delovanja nevronske mreže. Ta cenitvena funkcija tipično meri delovanje nevronske mreže preko več različnih vhodov nevronske mreže (mehanizem nagrajevanja in kaznovanja skozi daljše časovno obdobje).

Nenadzorovano učenje

(unsupervised
learning)

Uteži in pragovi se spreminjajo samo glede na dane vhode nevronske mreže. Pričakovane izhodne vrednosti niso podane. Večina nevronske mreže s tem učenjem izvaja neke vrste gručenje (*clustering*). Naučijo se kategorizirati vhodne vzorce v končno število razredov.

Perceptron

- Odločitvena meja je premica (hiper-ravnina v večdimenzionalnem vhodnem prostoru, če je število vhodov p večje od 2):

$$\mathbf{w}^T \mathbf{p} + b = 0 \quad (1)$$

- Vse točke (vhodi \mathbf{p}) na odločitveni meji imajo isto vrednost skalarnega produkta (1), t.j. 0. To pomeni, da imajo vsi ti vhodi isto projekcijo na vektor uteži \mathbf{w} in ležijo na ravnini, ki je **ortogonalna na vektor uteži \mathbf{w}** .
- Katerikoli vhodni vektor nad odločitveno mejo ima skalarni produkt (1) večji od 0, vektorji pod odločitveno mejo pa imajo skalarni produkt (1) manjši od 0.
- Torej bo vektor uteži \mathbf{w} , narisan v prostoru vhodnih vektorjev \mathbf{p} , vedno **kazal proti vhodnemu področju, kjer je izhod nevronske mreže pozitiven**.
- Vrednost praga se običajno nastavi za tem, ko so se nastavile uteži (t.j. za rotacijo oz. poravnavo vektorja uteži \mathbf{w}), izbrana pa je tako, da je izpolnjena enačba meje odločitve (1).

Perceptron: povzetek učnega algoritma

- napaka $e = t - a$
- **splošno učno pravilo** (neglede na dimenzijo vektorja \mathbf{p}):

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} + e \mathbf{p}_i = \mathbf{w}^{\text{old}} + (t - a) \mathbf{p}_i$$

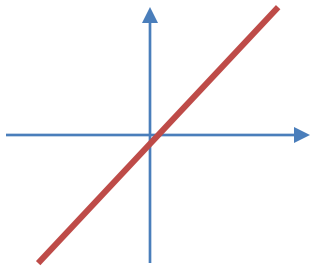
$$b^{\text{new}} = b^{\text{old}} + e$$

- To pravilo skonvergira v končnem številu korakov, če je le zastavljen problem linearno rešljiv (t.j. če so dani vhodni vektorji \mathbf{p} učne množice linearno ločljivi).

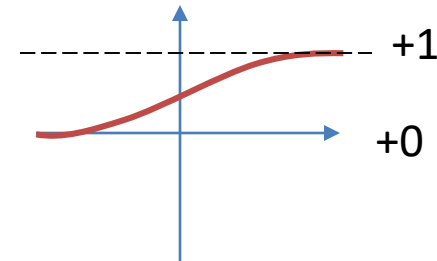
Dokaz: glej **Poglavje 4** knjige M.T. Hagan, H.B. Demuth, M.H. Beale: *Neural Network Design*, 2002.

Splošen nevron

odvodljiva aktivacijska funkcija



linearna aktivacijska funkcija
(žurka za osamelce – „daj muzko
ful na glas!“)



sigmoidna aktivacijska funkcija
(osamelci ne morejo bistveno
preglasiti ostalih)

Zamisel: Če je aktivacijska funkcija odvedljiva, lahko nevron učimo s pomočjo **gradientne optimizacije** njegovih uteži in praga.

Vse kar potrebujemo je **kriterijska oz. cenitvena funkcija!**

Postopek najmanjših kvadratičnih pogreškov

LMS ali Widrow-Hoff-ov učni algoritem

1. Učna množica

$$\{\mathbf{p}_1, \mathbf{t}_1\} \{\mathbf{p}_2, \mathbf{t}_2\} \dots \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

2. Cenitvena funkcija: kvadratični pogreški (Mean-Square-Error - MSE):

$$F(e) = E(e^2) = E((\mathbf{t}_Q - a)^2) = E((\mathbf{t}_Q - \mathbf{w}^T \mathbf{p}_i - b)^2)$$

kjer je E matematično upanje.

3. Gradientna optimizacija vsake uteži w_{ji} :

$$w_{ji}^{\text{new}} = w_{ji}^{\text{old}} - \alpha \Delta F(e)$$

α – stopnja učenja

$\Delta F(e)$ – gradient cenitvene funkcije

$$\Delta F(e) = \partial F(e) / \partial w_{ji} = \partial F(e) / \partial e \cdot \partial e / \partial w_{ji} = 2E(t_i - a_i) \cdot -p_{ji}$$

4. Gradientna optimizacija celotnega vektorja uteži \mathbf{w} :

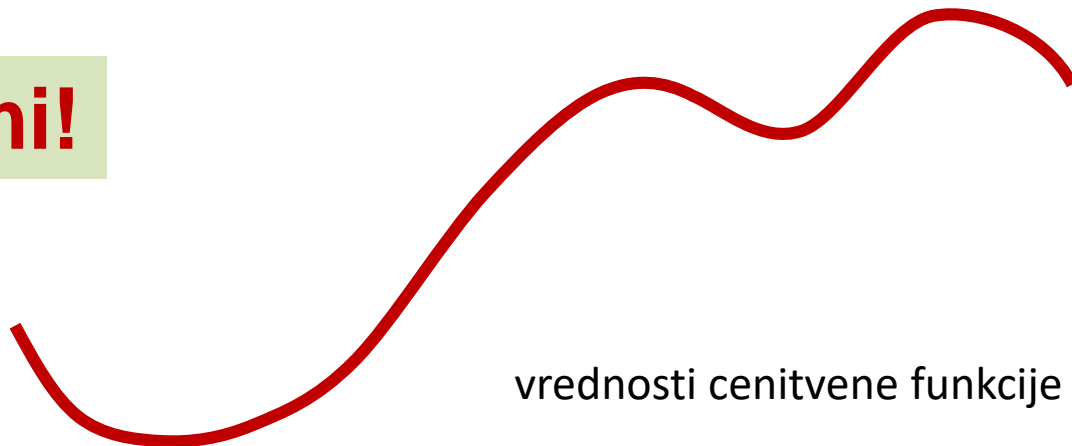
$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} + \alpha E(t_i - a_i) \cdot \mathbf{p}_i$$

Učenje nevronske mreže

- **Vzvratno učenje (Backpropagation)**

- potrebuje učno množico (pare vhodov in izhodov)
 - prične z majhnimi naključnimi utežmi
 - uporablja cenitveno funkcijo (npr. MSE) za prilagajanje posameznih uteži (nadzorovano učenje)
- Gradientna optimizacija je iskanje optimuma po površju cenitvene funkcije

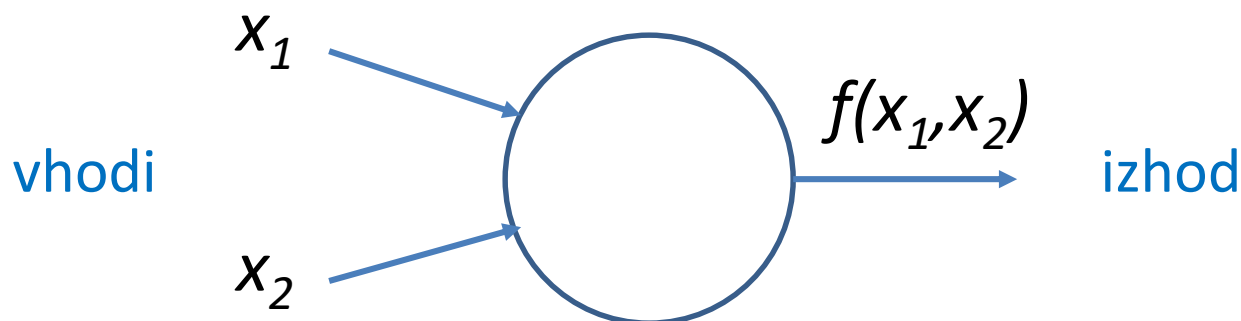
Lokalni optimumi!



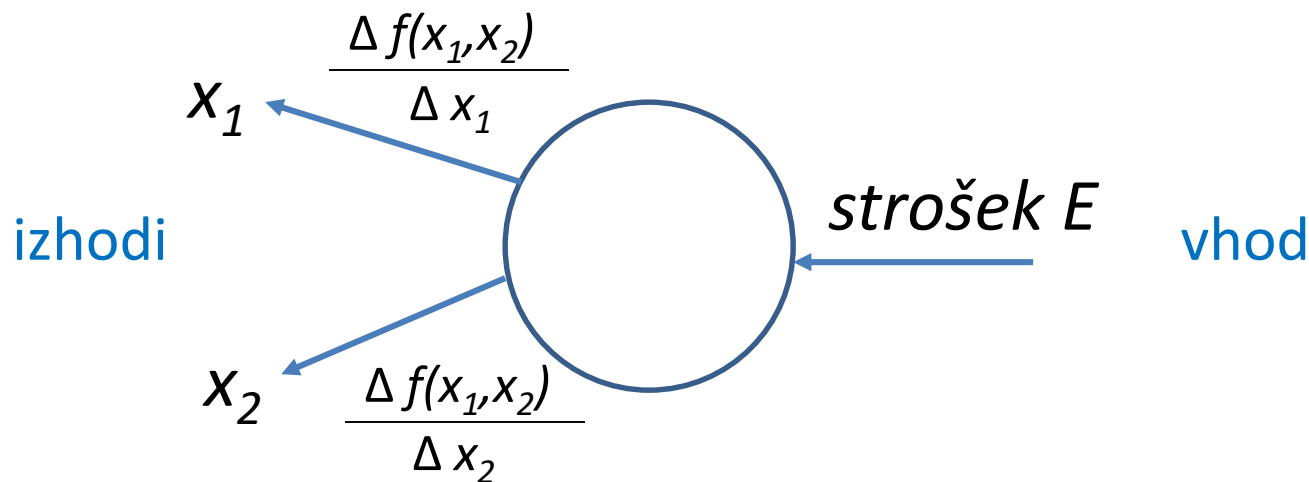
Vzvratno učenje – grafična predstavitev

nevron z aktivacijsko funkcijo f in njenim odvodom Δf

računanje naprej (feed forward)



in vzvratno učenje (backpropagation)



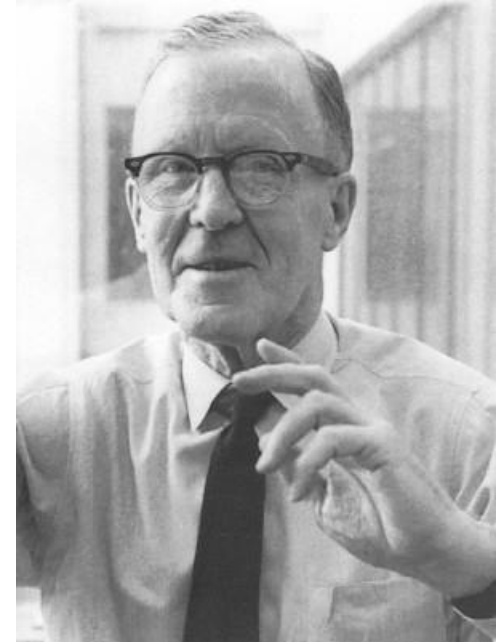
Parcialni odvodi v vozlišču

Hebbova teorija

"cells that fire together, wire together"

Hebbovo učenje: asociativno učenje, pri katerem sočasna aktivacija nevronov vodi v povečano sinaptično aktivnost (komunikacijo) med temi nevroni.

S stališča umetnih nevronov in umetnih nevronske mreže je Hebbov princip mogoče uporabiti kot metodo za nastavljanje uteži povezav med posameznimi nevroni. **Utež povezave med dvema nevronoma se poveča, če oba nevrona prožita istočasno** (t.j. pri istem vhodnem vektorju oz. vzorcu). Utež povezave se zmanjša, če nevrona prožita ob različnih časih. Tisti nevroni, ki dajejo isti odziv na vhodne vzorce pridobijo močne pozitivne povezave (eksitacija). **Tisti, ki prožijo asinhrono pridobijo negativno obtežene povezave (inhibicija).** Opisani mehanizmi omogočajo **asociacije** (delni vhodni vzorec bo vzbudil odziv vseh močno povezanih nevronov).



Donald O. Hebb

Hebbova teorija

"cells that fire together, wire together"

$$w_{ij} = x_i x_j$$

kjer je w_{ij} utež povezave med nevronom j in nevronom i in x_i vhod nevrona i

V primeru več vhodnih vrednosti:

$$w_{ij} = \frac{1}{p} \sum_{k=1}^p x_i^k x_j^k$$

kjer je p število učnih vzorcev.

Unsupervised
learning!

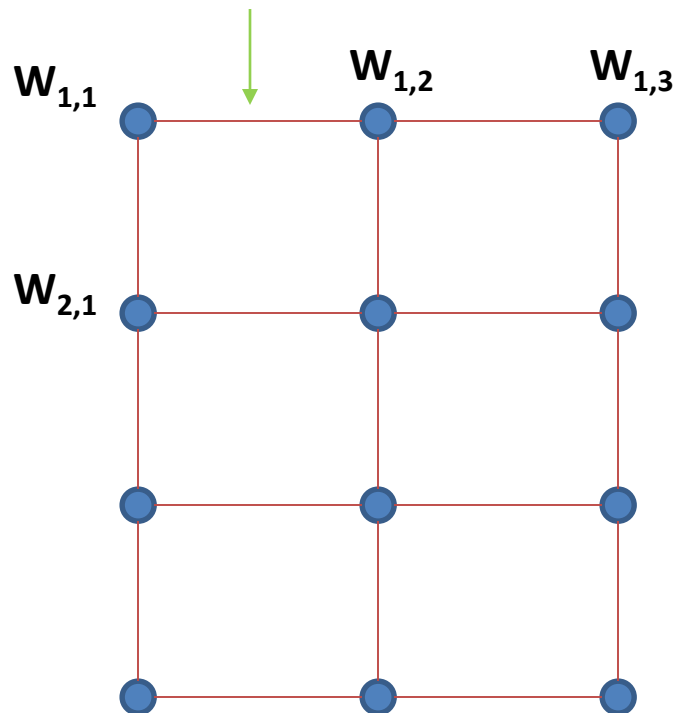
Pogojno učenje ([pogojni refleksi](#)):

- **Ivan Petrovič Pavlov** & njegovi psi: pozvoni z zvoncem kadarkoli hraniš pse. Čez čas se bodo psi slinili kadarkoli pozvoniš z zvoncem (z ali brez hrane).
- 2. svetovna vojna: ruski psi z bombami & hrana pod nemškimi tanki.

Samooorganizirajoče nevronske mreže (SOM)

pred učenjem

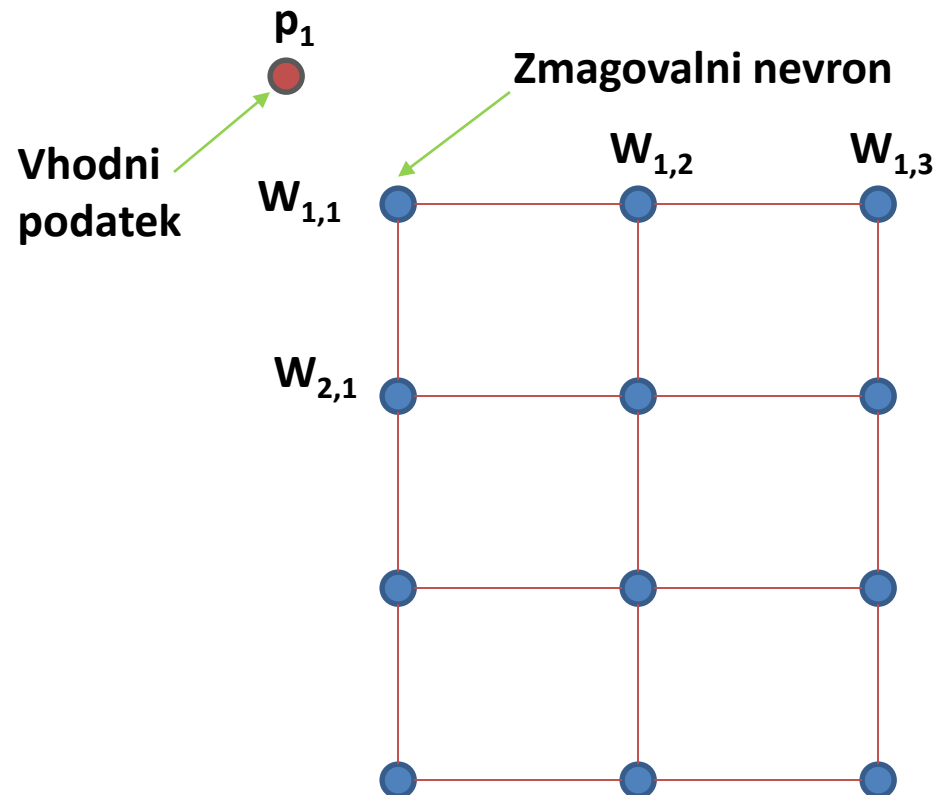
Sosedskost pred učenjem (to niso fizične povezave med nevroni)



Skupni vektorski prostor
uteži nevronov w in tudi
vhodnih vektorjev p

med učenjem

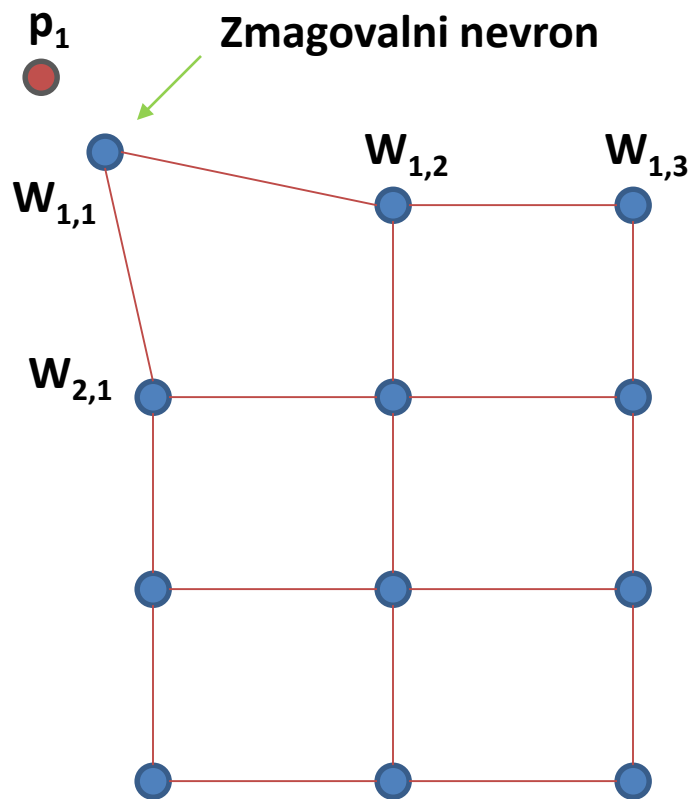
(zmagovalni nevron še ni posodobil svojih uteži)



Samoorganizirajoče nevronske mreže (SOM)

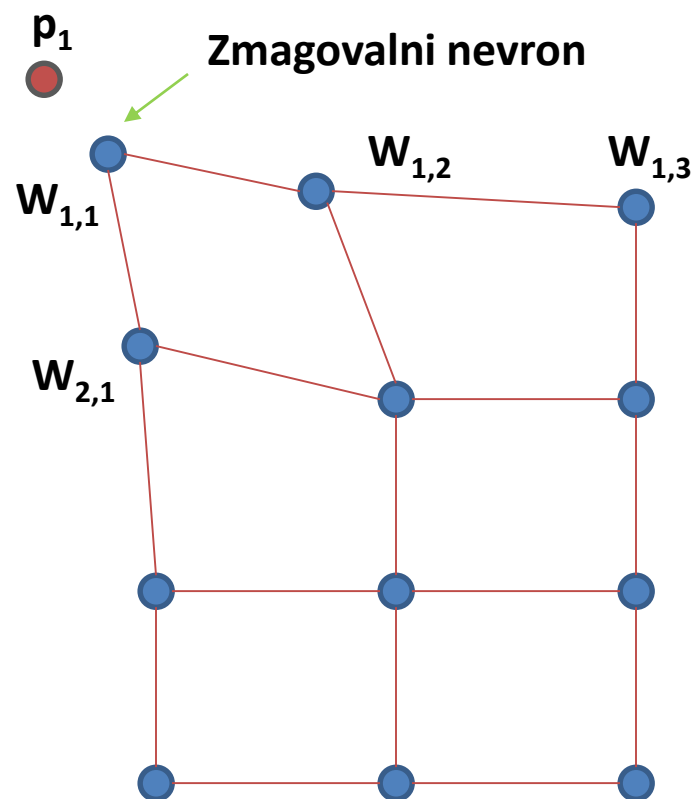
med učenjem

(zmagovalni nevron posodobi svoje uteži tekmovalne NN)



med učenjem

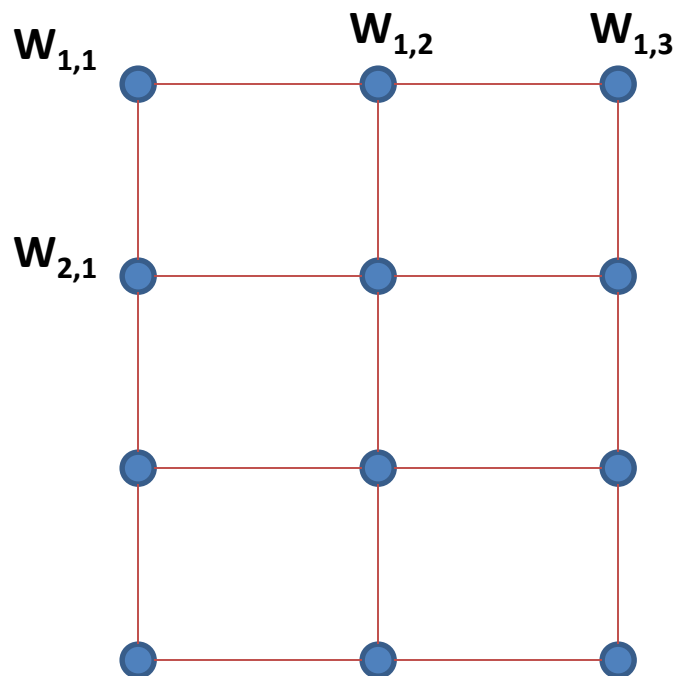
(zmagovalni nevron in njegovi sosedje posodobijo svoje uteži - SOM)



Samoorganizirajoče nevronske mreže (SOM)

pred učenjem

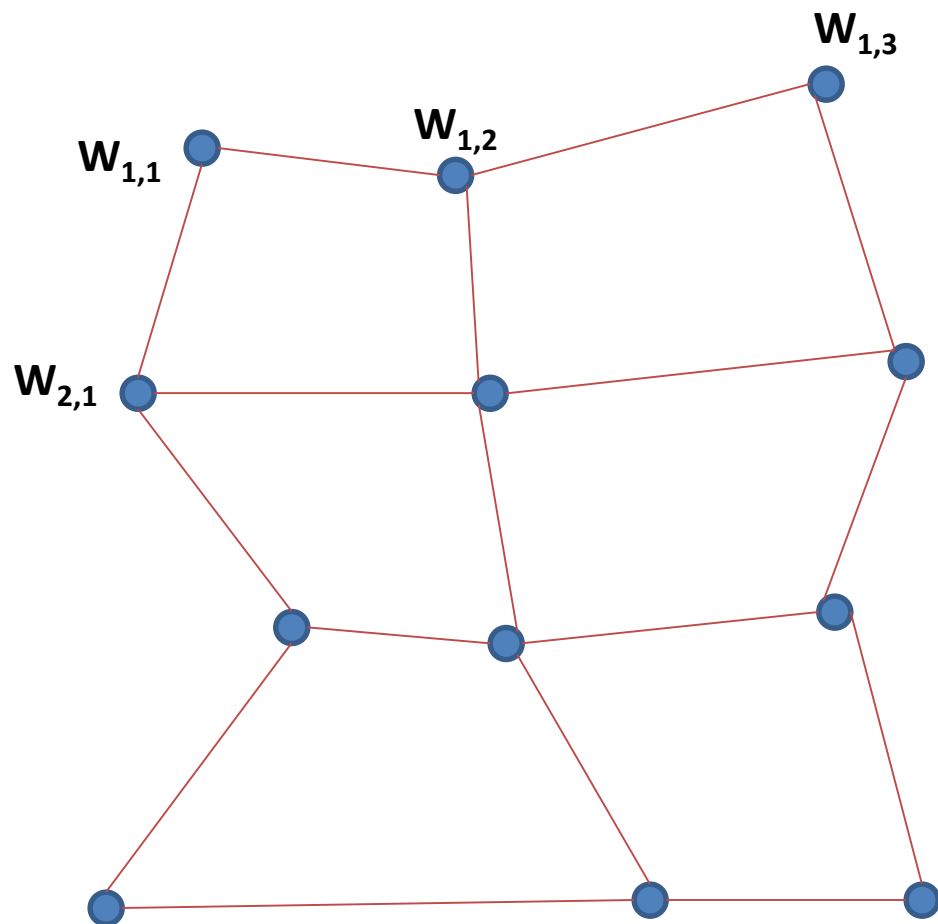
Sosedskost pred učenjem (to niso fizične povezave med nevroni)



Skupni vektorski prostor
uteži nevronov \mathbf{w} in tudi
vhodnih vektorjev \mathbf{p}

po učenju z veliko vhodnimi podatki

(zmagovalni nevron in njegovi sosedje posodobijo svoje uteži - SOM)

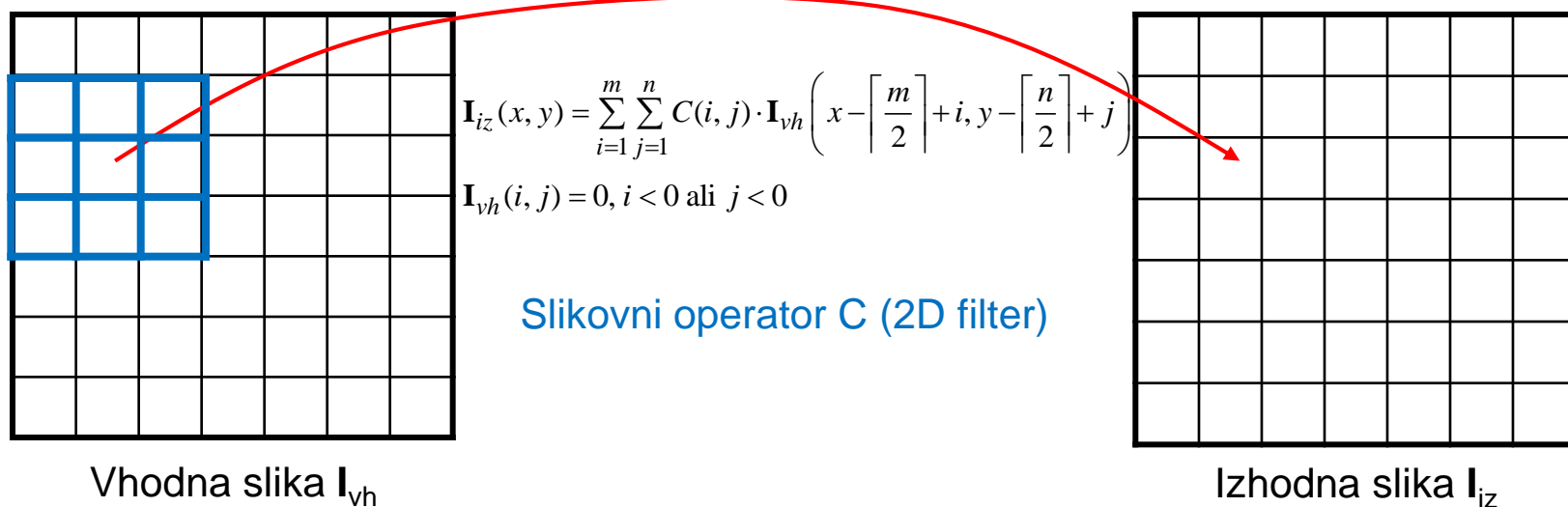


Konvolutivne nevronske mreže

(Convolutional Neural Networks) –

<http://cs231n.github.io/convolutional-networks/>

Konvolucija:



- Slikovni operator C (2D filter) se izračuna za vsako lokacijo na sliki.
- Izračuni za posamezne izhodne piksele so neodvisni in lahko tečejo paralelno.
- En nevron implementira en operator (2D filter) na izbrani lokaciji
- Uteži nevrona so uteži/koefficienti 2D filtra
- Vsi nevroni, ki implementirajo isti filter (samo na drugih lokacijah na sliki) se lahko učijo skupaj! To precej pospeši učenje nevronske mreže!

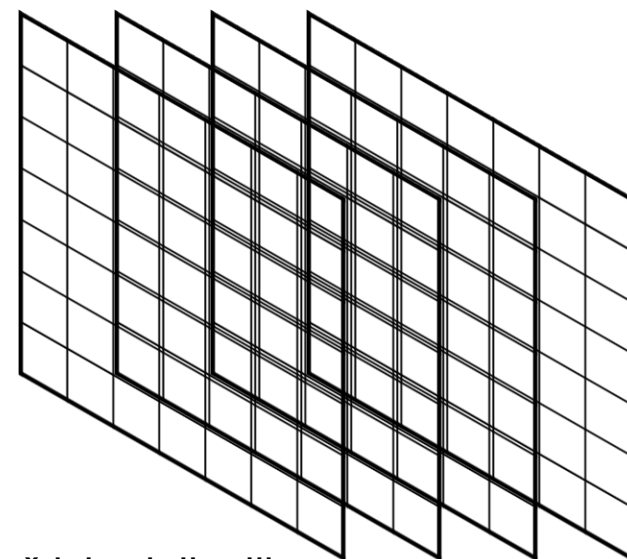
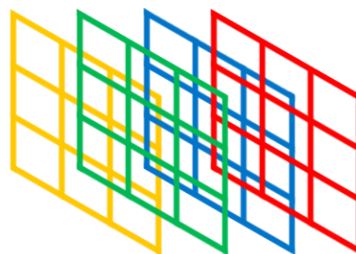
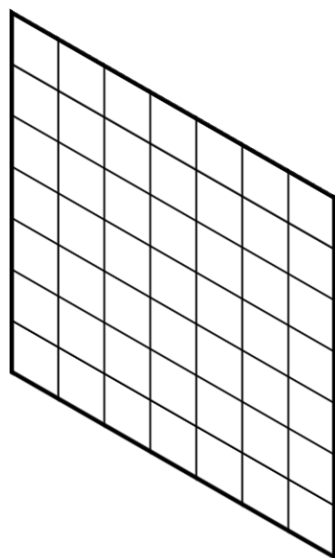
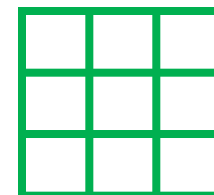
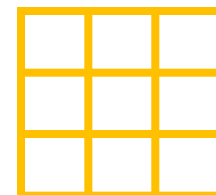
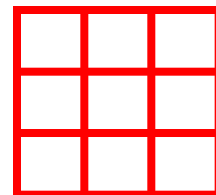
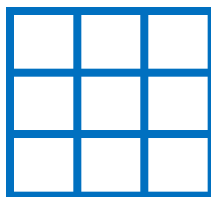
Konvolutivne nevronske mreže

(Convolutional Neural Networks) –

<http://cs231n.github.io/convolutional-networks/>

Možni 2D filtri:

- Detektor robov
- Detektor kotov
- Detektor homogenosti slikovne regije
- Detektor izbrane barve
- ...



Vhodna slika I_{vh}

Več
operatorjev

Več izhodnih slik

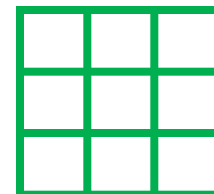
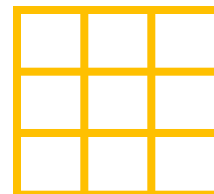
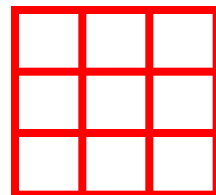
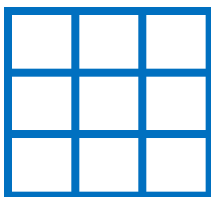
Konvolutivne nevronske mreže

(Convolutional Neural Networks) –

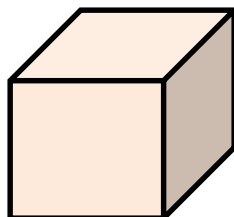
<http://cs231n.github.io/convolutional-networks/>

Možni 2D filtri:

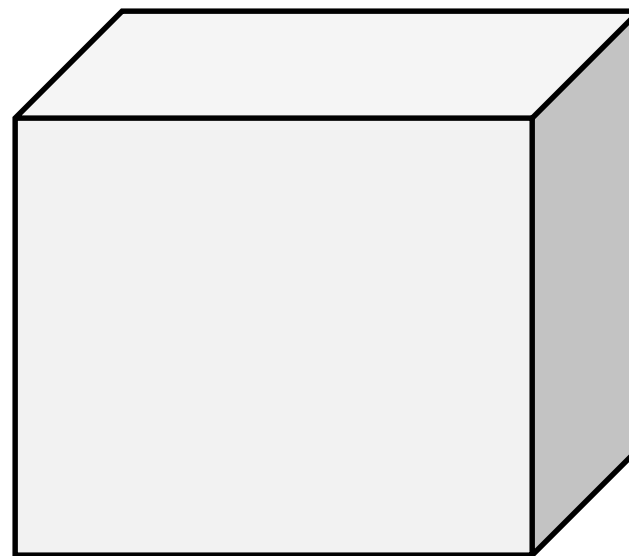
- Detektor robov
- Detektor kotov
- Detektor homogenosti slikovne regije
- Detektor izbrane barve
- ...



Vhodna slika I_{vh}



Več
operatorjev



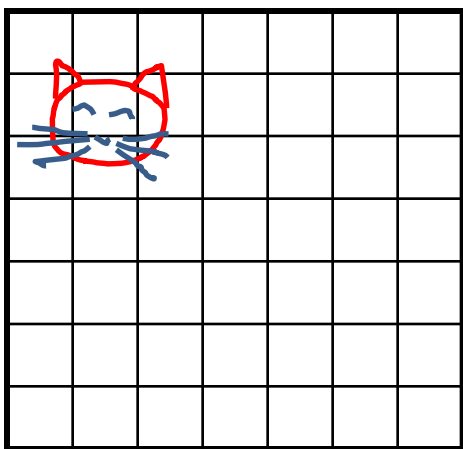
Več izhodnih slik

Konvolutivne nevronske mreže

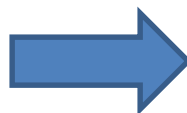
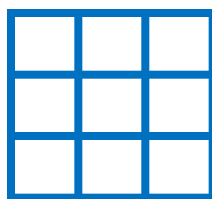
(Convolutional Neural Networks) –

<http://cs231n.github.io/convolutional-networks/>

Kje na sliki je mačka?



Detektor (2D filter) mačke

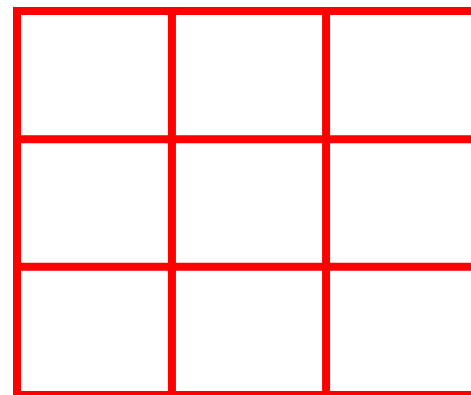
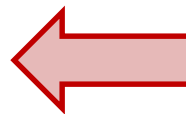


1	2	1	0	0	0	0
3	9	4	0	0	0	0
2	3	2	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Aha, mačka je v
zgornjem levem
kotu!

9	0	0
0	0	0
0	0	0

Max pool plast



Osnovni hiperparametri (poleg plasti in št. nevronov)

Kaj je vzorec (angl. sample)?

Vzorec je ena vrstica podatkov. Vsebuje vhode, ki se vnesejo v nevronske mreže in izhod, ki se uporablja za primerjavo z izhodom nevronske mreže (za izračun napake oz. stroškovne funkcije).

Kaj je serija (angl. batch)?

Velikost serije je hiperparameter, ki določa število vzorcev, ki jih je treba obdelati pred posodobitvijo notranjih parametrov nevronske mreže, torej prej posodobitvijo uteži nevronov.

Kaj je število ponovitev (angl. epoch)?

Število ponovitev je hiperparameter, ki določa, kolikokrat bo učni algoritem obdelal celoten nabor podatkov o v učni množici, preden proglasimo nevronske mreže za naučeno.

Kaj je učni algoritem (angl. optimizer)?

Učni algoritem določa strategijo spreminjanja uteži v nevronske mreže, vključno s strategijo izogibanja lokalnih minimumov.

Konvolucijske nevronske mreže trpijo za katastrofalnim pozabljanjem in imajo težava z dolgoročnim sklepanjem

J. Božic, D. Skočaj: Katastrofalno pozabljanje pri inkrementalnem učenju konvolucijske nevronske mreže, ERK2019

[https://erk.fe.uni-lj.si/2019/papers/bozic\(katastrofalno_pozabljanje\).pdf](https://erk.fe.uni-lj.si/2019/papers/bozic(katastrofalno_pozabljanje).pdf)

Michael Nguyen: Illustrated Guide to LSTM's and GRU's: A step by step explanation

<https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>

Christopher Olah: Understanding LSTM Networks

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>