1. Stručná história konekcionizmu, základné časti biologických neurónov a ich vlastnosti, typické vlastnosti konekcionistického modelu, klasický konekcionizmus model s prahovou logikou, implementácia Booleových funkcií.

Konekcionizmus je inšpirovaný biológiou. Je založený na umelých neurónových sietiach. Modeluje mentálne procesy, aplikácie v praktických problémoch. Mozog má 10^11 neurónov a 10^14 synáps.
História: klasický do 1940 – filozofia, psychológia, starý do 70^{th} – začiatok počítačov, umelých NN, spája sa s kognitívnou vedou, nový od 1986 – paralelné distribuované subsymbolické, multi-layer, rekurentné, najnovšie koncom 90^{th} – pravdepodobnostné metódy.
Knowledge reprezentation – podobné vstupy by mali mať podobnú reprezentáciu, na dôležitý príznak by malo reagovať viac neurónov. Subsymbolická reprezentácia, induktívny tréning(s/bez učiteľa), deduktívny testing.
Neurón vie simulovať každú lineárne separabilnú boolovskú fciu. Na ostatné treba 2vrstvovú sieť.

2. Binárny perceptrón: pojem učenia s učiteľom, učiace pravidlo, algoritmus trénovania, deliaca nadrovina, klasifikácia vzorov, lineárne separovateľné problémy, definícia a príklad.

Zo vstupu sa zráta suma, aktivačnou fciou(sgn,lin,sigmoid) výstup, porovná sa s učiteľom, upravia sa váhy podľa delta pravidla: w_i += α(d-y)x_i. Chyba E=1/2(d-y)^2. Koniec, ak E==0.
Rieši len lineárne separabilné problémy. Posúva sa deliaca nadrovina.

3. Spojitý perceptrón: Rôzne aktivačné funkcie perceptrónu, chybová funkcia a spôsob jej minimalizácie, pojem učenia s učiteľom, učiace pravidlo, algoritmus trénovania perceptrónu.

Aktivačné fcie: unipolárna, bipolárna, lin, sigmoid, hypertangens (e^-net -1)/(1-e^-net). E=1/2(d-y)^2.
Minimalizácia chyby, hľadáme minimum fcie. Zderivujeme a ideme podľa strmosti(gradientu). A to buď po každom vstupe (online).w_i += α(d-y)f`*x_i, alebo po epoche (batch) w_i += $\alpha\Sigma$_p{d_p-y_p}f`*x_i, kde f` je y*(1-y)

4. Viacvrstvové dopredné neurónové siete: architektúra a aktivačné vzorce, odvodenie metódy učenia pomocou spätného šírenia chýb (BP) pre dvojvrstvovú doprednú NS, modifikácie BP, typy úloh pre použitie doprednej NS.

Cieľ – aproximovať fciu. Ak je graf siete acyklický, dá sa použiť dopredné šírenie. Aktivačná fcia je sigmoida 1/(1+e^-net), alebo hyperbolický tangens (e^-net -1)/(1-e^-net). Adaptácia – hľadanie lok. miním gradientovou metódou.
Aktivačné vzorce: y_i = fΣ_k{w_ik*h_k}, h_k = fΣ_j{v_kj*x_j}.
Pre každý výstupný neurón sa zráta chyba: delta_i =(d_i – y_i)*f`. Pre každú váhu hidden-output sa w_ik += $\alpha$delta_i*h_k. Chyba sa propaguje, každý z hidden schytá: delta_k = $\Sigma$_i{w_ik*delta_i}f`.
A updatnú sa váhy input-hidden: v_kj += αdelta_k*x_j
Mod: momentum na preskočenie lok. miním. XOR ako zloženie boolfcií. Weight decay – váhy sa vynásobia konštantou. Reorder inputs.

5. Viacvrstvová dopredná NS ako univerzálny aproximátor funkcií (formulácia teorému), trénovacia a testovacia množina, zovšeobecňovanie, preučenie, skoré zastavenie učenia, selekcia modelu, prekrížená validácia.

Trojvrstvová sieť je schopná s danou presnosťou aproximovať ľubovoľnú fciu. Dvojvrstvová vie len konvexne ohraničiť triedy. Jednovrstvová len rovinou. Pri cross-validácii jeden cluster validačný.

6. Lineárne modely NS: vzťah pre riešenie systému lin. rovníc v jednovrstvovej sieti, pojem pseudoinverzie matice, autoasociatívna pamäť: lineárny obal, princíp funkcie modelu, detektor novosti.

Výstupy siete y=(y_1..y_m)^T, vstupy x=(x_1..x_n), matica váh W=(w_11..w_1n.. ..w_m1..w_mn). y=Wx. Pridávanie vrstiev je zbytočné, lebo lin. zobr. je uz. na skladanie. Trénovacia množinamá vstupy X a výstupy Y. Y=WX. Ak je X regulárna, W=YX^-1. Inak W=YX^+, X^+ je pseudoinverzia X. Ak je počet trénovacích vzoriek N väčší ako dimenzia vstupov n (X je širšia) a hodnosť X=n, potom X^+ = X^T*(XX^T)^-1. Inak X^+ = (X^T*X)^-1*X^T. Platí: X*X^T*X = X. Autoasociácia: Y=X. W=XX^+. Predpokl. že N<n. Vstupy definujú L podpriestor R^n ako svoj lineárny obal. Doplnok L k R^n je podpriestor R^n kolmý na L. Každý vektor sa dá jednoznačne rozložiť na zložky z L a R^n-L. Z odchýliek sa spraví ortogonálna projekcia(odstráni sa šum) vynásobením zľava XX^+. Keby sa to zľava vynásobilo (I-XX^+), máme bázový vektor z R^n – L, kolmý na L – detektor novosti je model s tým operátorom.

7. Lineárne modely NS: účel GrammovhoSchmidtovho ortogonalizačného procesu, GI model. Pamäť korelačnej matice ako autoasociatívna pamäť, vzťah pre výpočet váh, presluch, porovnanie s GI.

Pridanie novej vzorky nemusí byť až tak výpočtovo náročné ako rátať znovu pseudoinverziu.
GI: Ortogonalizačný proces: máme bázu u^1..u^k. Chceme ortogonálnu bázu v^1..v^k. v^1:=u^1. Keď už máme p vektorov určených, ďalší bude z priestoru gen. v^1..v^p,u^(p+1).
v^(p+1) = u^(p+1) - Σ_i=1^p{(v^i^T * u^(p+1) / ||v^i||^2) * v^i}
Keď príde nový vektor x, čo nepatrí do priestoru, týmto vzorcom získame zložku kolmú na priestor.
W_0 = 0.   W_(N+1) = W_N +(x_(N+1) * x_(N+1)^T / |x_(N+1)|^2).
x_(N+1) = x – W_N*x.
Korelačná matica: Váha z j-teho vstupu do i-teho výstupu w_ij je úmerna sume x_j*y_i pre všetky x,y z trénovacej množiny. W=YX^T. Ak sú vstupy X ortonormálne, X^T=X^-1=X^+. Potom CMM=GI. Autoasociácia: W=XX^T. Pre vstup x_p: Wx_p = x_p|x_p|^2 + C(p) – presluch z iných zložiek. Ak by bol nulový, x_1..x_N sú ortogonálne. Vo všeobecnosti teda nie je. Dá sa ale znížiť posunutím strednej hodnoty zložiek vstupov do nuly. x_k := x_k – 1/N Σx_i. (obr.)
Porovnanie: Ak vstupy nie sú ortogonálne, GI s detektorom novosti je lepší. (obr. rozp. tvárí).

8. Samoorganizácia v NS, základné princípy, pojem učenia bez učiteľa, typy úloh použitia, Ojovo pravidlo učenia – vzťah pre adaptáciu váh a vysvetlenie konvergencie váhového vektora, pojem vlastného vektora a vlastného čísla.

Učenie bez učiteľa – algoritmus učenia nemá informáciu o požadovaných aktivitách výstupných neurónov v priebehu trénovania. Súperenie medzi váhami, update váh blízkych víťazovi. Tendencia samozosilňovania – Hebbovo pravidlo. Využíva sa redundancia vo vstupných dátach (využíva sa klastrovanie – vektorová kvantizácia). Topologické zobrazenie príznakov. ()Ak netreba hlavné komponenty, ale stačí podpriestor nimi generovaný, je tu Ojovo pravidlo: w_ij += αy_i(x_j - Σ{y_k*w_kj}), máme n vstupov a p výstupov, i=1..p, k=1..p
výstupy nie sú usporiadané, variancia je cca rovnaká. Výsledky závisia od počiatočných podmienok a poradiu vstupov. Váhový vektor konverguje k vlastnému vektoru korelačnej matice vstupov xx^T. Matica A má vlastné číslo v, ak je determinant A-vI nulový. Vlastné čísla trojuholníkovej a diagonálnej matice sú na diagonále. X je vlastný vektor prislúchajúci vlastnému číslu v, ak Ax=vx.
Architektúry – dopredné (+ laterálne spojenia)

9. Metóda hlavných komponentov pomocou algoritmu GHA a APEX, architektúra modelu, vzťah pre adaptáciu váh, pojem vlastných vektorov a vlastných čísel, redukcia dimenzie, aplikácia na kompresiu obrazu.

PCA = lineárna transformácia do priestoru príznakov. S tým súvisí redukcia dimenzie. Slúži na predspracovanie dát, ktoré majú cca Gaussovské rozloženie.
Príznakový priestor je n-dimenzionálny priestor, kde každý vzor je bod s n súradnicami. n je počet príznakov vzoru. Podobné vzory sú blízko pri sebe.
Matica A má vlastné číslo v, ak je determinant $A-vI$ nulový. Vlastné čísla trojuholníkovej a diagonálnej matice sú na diagonále. X je vlastný vektor prislúchajúci vlastnému číslu v, ak $Ax=vx$.
Majme vstupý vektor x, premietneme ho do priestoru príznakov na vektor u. Variancia je u^T*Ru, kde $R=xx^T$ je korelačná matica vstupu. Keď variancia nadobúda extrém, $Ru = vu$, kde v sú vlastné čísla R a u sú vlastné vektory R. Pre n rôznych vlastných vektorov dostaneme n rôznych projekcií $a_i = x^T*u_i$. a_i sú hlavné komponenty(príznaky). Rekonštrukcia: $x = \Sigma\{a_i*u_i\}$. Keď to zosumujeme len pre p najväčších vlastných čísel, dostaneme akceptovateľnú aproximáciu s redukovanou dimenziou.
GHA:
Majme n vstupov, p výstupov. Pre i=1..p $w_{ij} \mathrel{+}= \alpha y_i(x_j - \Sigma_{k=1}^{i}\{y_k*w_{kj}\})$. Takto sa extrahuje p hlavných komponentov korelačnej matice vstupov usporiadaných podľa veľkosti. Netreba počítať korelačnú maticu. Výpočtovo nenáročný, ak $p<<n$. Je to reestimačný algoritmus. Použitie: kódovanie dát, kompresia.
APEX:
Má aj laterálne spojenia, ktoré majú inhibičný účinok, ale konvergujú k 0. Váhy konvergujú k vlastným vektorom. Je to dekorelačný algoritmus.

10. Učenie so súťažením (typu "winnertakeall"), nevýhody. Neurobiologická motivácia algoritmu SOM, laterálna interakcia a jej náhrada v SOM, sumarizácia algoritmu, voľba parametrov modelu, DP verzia algoritmu.

Oko: sietnica je prepojená s mozgovou kôrou – každý s každým, vplyv v závislosti od vzdialenosti má tvar mexického klobúka. Šírka mexického klobúka má byť dostatočne široká v porovnaní s šírkou lokálnych vstupov. TODO: laterálne spojenia_
$y(t+1) = S(W(x)+Ly(t))$ – y je vektor výstupov, S je sigmoida, L je matica laterálnej spätnej väzby, x vektor vstupov, W matica váh.
Výstupné neuróny majú vstup súčet výstupov zo vstupnej vrstvy a spätnoväzbovej zložky z výstupov okolitých neurónov v mape.
Víťazom súťaže je neurón, ktorý najviac reaguje na daný vstup(jeho váhový vektor je najbližšie vstupnému vektoru). Učenie: $w_i(t+1) = alpha*h(i^*,i)*(x(t)-w_i(t))$ – alpha je rýchlosť učenia (môže aj klesať), $h(i^*,i)$ vyjadruje kooperáciu (manhattanovská/gaussovská), časom sa zmenšuje. Fáza usporiadania/doladenia.
DP verzia: (dot product, skalárny súčin) $d_i = w_i^t*x$, spočíva to v rotácii váhového vektora smerom k vstupnému vektoru.Víťaz má najmenší uhol. Váhy ležia na povrchu hypergule.

11. SOM: vektorová kvantizácia, topografické zobrazenie príznakov, redukcia dimenzie, magnifikačný faktor, náčrt matematických problémov analýzy algoritmu, príklad použitia.

SOM umožňuje realizovať zobrazenie zachovávajúce topológiu a zobraziť tak charakteristické príznaky vstupných dát. Neuróny sú usporiadané v mriežke/v reťazi. Odozvy sú lokálne. Predpokladá sa redundancia vstupu – simultánna aktivita viacerých vstupných neurónov. Použitie: rozpoznávanie reči, kompresia obrazu, robotika. Magnifikačný faktor – počet vstupných dát na jednotku plochy.
Problémy:
vektorová kvantizácia: nahradenie množiny vstupných vektorov za menšiu množinu prototypov. Vstupný priestor sa rozdelí na Voronoiov diagram tak, aby sa minimalizovala chyba rekonštrukcie.

12. Hybridné modely NS, RBF model: aktivačné vzorce, bázové funkcie, príznakový priestor, problém interpolácie, aproximačné vlastnosti RBF siete.

RBF aktivácia: y_i = Σ_k{w_ik*h_k(x)}+w_i0, x je vektor vstupov, h radiálna aktivačná fcia.
radiálna fcia: h_k = e^(-|x-v_k|^2/σ_k^2), v_k je centrum, σ_k šírka.
Podľa wikipédie to má len jeden výstup y = Σ{w_i*h(|x-c_i|)}, kde c_i je centrum i-tého neurónu.
RBF je ako MLP tiež univerzálny aproximátor.
Bázové fcie: Gaussová e^(-r^2/σ^2), Multiquadrics sqrt(r^2+c^2), Inverse multiq. 1/sqrt(r^2+c^2), Cauchy 1/(1+r^2).
Príznakový priestor je n-dimenzionálny priestor, kde každý vzor je bod s n súradnicami. n je počet príznakov vzoru. Podobné vzory sú blízko pri sebe.
Problém interpolácie: RBF umožňuje interpolovať zobrazenie R^n->R. Interpolácia je metóda umožujúca konštruovať nové body z konečnej množiny známych bodov. Napríklad danými bodmi preložiť polynóm. Váhy sa dajú vypočítať pomocou lin. algebry.
Problém: zobrazenie nemusí byť všadedefinované ani jednoznačné ani spojité. (šum..)

13. Hybridné modely NS, RBF model: spôsoby trénovania váh. Základné vlastnosti dynamických modelov NS na online aproximáciu dátových množín (TRN, DCS). Porovnanie RBF a MLP.
RBF aktivácia: y_i = Σ_k{w_ik*h_k(x)}+w_i0, x je vektor vstupov, h radiálna aktivačná fcia.
MLP aktivácia: y_i = f(Σ_k=1^(q+1){w_ik*h_k}), h_k = f(Σ_j=1^(n+1){v_kj*x_j}).
radiálna fcia: h_k = e^(-|x-v_k|^2/σ_k^2), v_k je centrum, σ_k šírka.
RBF sa trénuje dvojfázovo: najprv sa upravujú centrá a šírky, potom váhy.
Centrá: vyberú sa náhodne zo vstupov alebo klasterizáciou vstupov (samoorganizáciou) alebo metódou najmenších štvorcov (supervised with error BP).
Šírky: sa vyberú podľa najväčšej vzdialenosti centier.
Váhy: zráta sa matica G (g_11..g_1n.. ..g_m1..g_mn), g_ij = h(|x_i-x_j|). w=G^+*y
Dá sa to, lebo na rozdiel od MLP majú RBF siete jedno lokálne minimum, keď sú centrá fixnuté.
Ale tiež sa to dá aj gredient descentom.
TRN = topology representing network, neorientovaný graf, po určení víťaza sa zvýši vek hrán z neho idúcich, staré hrany sa rozpoja.
DCS = dynamic cell structures, neorientovaný graf, môžu sa vkladať vrcholy. Odstraňujú sa vrcholy bez spojenia.
RBF vs. MLP: obe sú nelineárne dopredné vrstvové univerzálne aproximátory, RBF dvojfázové, curve-fitting in high-dim. space, local, rýchlejšie konverguje, MLP jednofázové, stochastické, globálne.

14. Rekurentné NS: spôsoby reprezentácie času, typy úloh pre rekurentné NS. Modely s časovým oknom do minulosti, výhody a nedostatky, príklad použitia.
V trénovacej množine môžu byť pre rovnaké vstupy rôzne výstupy. Treba rozlišovať podľa času.
Viacvrstová sieť s časovým posunom(okno do minulosti). Stačí obyčajné BP, treba zachovať poradie trénovacích vzoriek. Ťažko určiť veľkosť okna. Niekedy treba aj nekonečné.
Mealyho automat, prechody sú dvojice(vstup, výstup). Niečo ako a-prekladač s resetovacím prechodom.
Dva typy skrytých neurónov – rekurentné(kontextové) a nerekurentné. Nakresliť rôzne architektúry.
Treba vstupnú, výstupnú vrstvu, skrytú a rekurentnú.
Použitie: rozpoznávanie, predikcia, generácia.

15. Rekurentné NS: opis architektúry a princíp trénovania RNN pomocou algoritmu BPTT a pomocou RTRL. Príklad použitia.

Aktiváciu majú rovnakú ako MLP: S_i^(t+1) = g(Σw_ij*S_j^t + I_i^t) – I je vstup.
BPTT: Chyba sa šíri po časovo rozvinutej sieti takisto ako v BP. Veľké pamäťové nároky. Moc sa neujala.
RTRL: Netreba vedieť dopredu dĺžku postupnosti, môže sa meniť a netreba alokovať toľko pamäti. Výpočtovo náročnejšia. Požadovaná odpoveď v čase T+1 je O. Chyba k-teho neurónu v čase t=T+1 je E_k^t = O_k – S_k^t.
V čase t z {1..T} je E_k^t = 0. Celková chyba je E^t = 1/2{Σ(E_k^t)^2}. Váhy sa updatujú ako v MLP. delta w_ij^t = εΣ(E_k^t*deriv.S_k^t podla w_ij).

16. Elmanova sieť: interné reprezentácie pri symbolovej dynamike, markovovské správanie, architektúrny bias, vzťah k IFS, sieť s echo stavmi architektúra, trénovanie.

Elman: predikuje postupnosti z kontextu, rozpoznáva, dopĺňa, má hierarchickú reprezentáciu. Stavový priestor siete má fraktálovú reprezentáciu. Trénovať sa môže BP bez rekurentných váh, alebo BPTT alebo RTRL.
IFS – množina transformácií. Príklad – žaba v sierinskeho trojuholníku. Postupnosť transformácií je adresa v priestore. Dve postupnosti ak majú dlhý spoločný suffix, sú blízko seba.
Markovovská vlastnosť: budúce stavy sú nezávislé na minulých stavoch.
Architektúrny bias: je fenomén – štruktúra klastrov odráža históriu vstupov.

17. Hopfieldov model NS: deterministická verzia, typy dynamiky modelu, energia systému, relaxácia, možné typy atraktorov, autoasociatívna pamäť – nastavenie váh, kapacita pamäte.

Každý neurón je v jednom z dvoch stavov S_i ∈{-1,+1}. Spojené sú každý s každým. Váha synapsy J_ij. Postsynaptický potenciál h_i^int=Σ_j=1^N{J_ij*S_j}. Ak prekročí prah excitácie h_i^ext, neurón i sa aktivuje. S_i = sgn(h_i^int – h_i^ext).
Paralelná dynamika: Všetky menia svoj stav naraz. Najprs sa zrátajú zmeny, potom sa aplikujú.
Sekvenčná dynamika: Vždy sa mení len jeden náhodne vybratý neurón. Zodpovedá to pohybu po hranách hyperkocky.
Energia systému: E(S)=-1/2(ΣJ_ij*S_i*S_j) - ΣS_i*h_i^ext.
Atraktory: pravé/falošné
Správanie: chaotické trajektórie(synchrónna dyn., asymetrická J), limitné cykly(pri synch. dyn.), bodové atraktory.(energia klesá, kým nedosiahne minimum, pri async. dyn.)
Autoasociátor: nastavenie váh: J_ij=Σx_i^p*x_j^p pre jednotlivé pamäťové vzory p. Pri relaxácii keď sa dostane do nejakého vzoru (S_i=x_i^p) (recall), podmienka stability: 1+C_i^p = x_i^p*h_i^p>0. C_i^p je presluch, rovná sa nule, ak sú pamäťové vzory ortogonálne. Pamäťová kapacita sa potom blíži k počtu neurónov.

18. Hopfieldov model NS: stochastická verzia, opis dynamiky, parameter inverznej teploty, odstránenie falošných atraktorov. Porovnanie s deterministickou verziou. Typy úloh pre použitie Hopfieldovho modelu.

Každý neurón je v jednom z dvoch stavov S_i ∈{-1,+1}. Spojené sú každý s každým. Váha synapsy J_ij. Postsynaptický potenciál h_i^int=Σ_j=1^N{J_ij*S_j}. Ak prekročí prah excitácie h_i^int, neurón i sa aktivuje.
Sú tu reverzné atraktory, ale tie nás netrápia. Chceme odstrániť zmiešané atraktory. Pridáme šum(teplotu). Každý zmiešaný stav má kritickú teplotu do 0.46, od ktorej už nie je stabilný.
Použitie: rozoznávanie vzorov(písmo, tváre..), generovanie, rozpoznávanie postupností(reč, zvuky, video..), autoasociácia, klasifikácia, rekonštrukcia, optimalizačné problémy, obch. cestujúci, párovanie,