

Zdeněk Buk Buk

Nobelova cena za zásadní objevy a inovace v oblasti umělých neuronových sítí. Od biologické inspirace k moderní umělé inteligenci

Pokroky matematiky, fyziky a astronomie, Vol. 69 (2024), No. 4, 193–219

Persistent URL: <http://dml.cz/dmlcz/152862>

Terms of use:

© Jednota českých matematiků a fyziků, 2024

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library*
<http://dml.cz>

Nobelova cena za zásadní objevy a inovace v oblasti umělých neuronových sítí. Od biologické inspirace k moderní umělé inteligenci

Zdeněk Buk

Abstrakt. V roce 2024 byla Nobelova cena za fyziku udělena Geoffreyovi Hintonovi a Johnu Hopfieldovi za zásadní objevy a inovace, které umožnily strojové učení s umělými neuronovými sítěmi. Tento článek se zaměřuje na historický vývoj neuronových sítí od jejich počátků, inspirovaných biologickými modely, až po moderní architektury, jako jsou hluboké sítě, rekurentní modely či konvoluční sítě a transformery. Popisuje klíčové milníky, teoretické základy a aplikace, které dnes ovlivňují širokou škálu oblastí od počítačového vidění po zpracování přirozeného jazyka.

1. Úvod

Tento článek má za cíl přiblížit problematiku neuronových sítí a umělé inteligence širší odborné veřejnosti. Motivací je nejen bouřlivý vývoj v této oblasti v posledních letech, ale samozřejmě také mimořádné ocenění, kterého se dostalo dvěma osobnostem v tomto oboru – Geoffreyovi Hintonovi a Johnu Hopfieldovi – v podobě Nobelových cen za fyziku v roce 2024.

Geoffrey Hinton (*6. prosince 1947, Londýn) je britsko-kanadský vědec, známý jako průkopník umělých neuronových sítí. Vystudoval experimentální psychologii na Univerzitě v Cambridge a doktorát získal na Univerzitě v Edinburghu, kde se věnoval umělé inteligenci. Během své kariéry působil na univerzitách v Sussexu, San Diegu a Carnegie-Mellon v Pittsburghu, od roku 1987 pak na Univerzitě v Torontu.

Hinton se proslavil svou prací na algoritmu zpětného šíření chyb [19] a na Boltzmannových strojích [2], [3]. Je považován za jednoho z „kmotrů hlubokého učení“ a v roce 2018 obdržel prestižní Turingovu cenu společně s Yoshuou Bengiem a Yannem LeCunem za technické průlomky, jež z neuronových sítí učinily klíčovou technologii výpočetní techniky.

Kromě akademického působení byl také spojen s Googlem, odkud odešel v roce 2023, aby mohl upozorňovat na rizika spojená s rozvojem umělé inteligence (AI). Hinton věří, že umělá inteligence bude mít na společnost podobný dopad jako průmyslová revoluce, ale zároveň varuje před možnými nebezpečími, která by mohla ohrozit lidstvo.

Mezi další Hintonova vyznamenání patří členství v britské Královské společnosti, Rumelhartova cena (2011) a Řád Kanady (2018).

Ing. ZDENĚK BUK, Ph.D., Katedra teoretické informatiky, Fakulta informačních technologií ČVUT v Praze, Tháškova 9, 160 00 Praha 6, e-mail: zdenek.buk@fit.cvut.cz

John Hopfield (*15. července 1933, Chicago, Illinois) je americký fyzik, biofyzik a neurovědec, známý svou průkopnickou prací v oblasti umělých neuronových sítí, molekulární biologie a statistické fyziky. Po bakalářském studiu fyziky na Swarthmore College (1954) získal doktorát na Cornellově univerzitě (1958) pod vedením Alberta Overhausera. Během své kariéry působil na prestižních institucích, jako jsou Bellovy laboratoře, Kalifornská univerzita v Berkeley, Kalifornský technologický institut v Pasadeně a Princetonská univerzita, kde je emeritním profesorem molekulární biologie.

Hopfieldova vědecká činnost pokrývá širokou škálu disciplín. Jeho přínos zahrnuje objev polaritonů, návrh Hopfieldovy dielektrické teorie, koncept kinetického proofreadingu v biomolekulární syntéze a slavnou Hopfieldovu síť (1982), která položila základy moderního strojového učení a výpočetní neurovědy. Je také spoluautorem pseudopotenciálů zachovávajících normu v chemické fyzice a průkopníkem aplikací kolektivních dynamik neuronových sítí.

Za svou práci obdržel mnoho významných ocenění, včetně Guggenheimova stipendia (1968), ceny Olivera E. Buckleyho (1969), ceny Maxe Delbrücka (1985), Diracovy medaile ICTP (2001), Harold Pender Award (2002), Světové ceny Alberta Einsteina za vědu (2005), Benjamin Franklin Medal (2019) a Boltzmannovy medaile (2022). V roce 1997 mu IEEE udělila Neural Networks Pioneer Award, a v roce 2009 získal Frank Rosenblatt Award za přínos k porozumění zpracování informací v biologických systémech.

Hopfield je členem Národní akademie věd USA, Americké akademie umění a věd a Americké filozofické společnosti. Jako prezident Americké fyzikální společnosti (2006) a pedagog inspiroval generace vědců. Jeho průkopnická práce zůstává klíčovou inspirací pro fyziku, biologii a umělou inteligenci.



Obr. 1. Laureáti Nobelovy ceny za fyziku 2024: John Hopfield a Geoffrey Hinton. © Nobel Prize Outreach. Autor: Nanaka Adachi

1.1. Neuronové sítě

Neuronové sítě původně inspirované svými biologickými vzory jsou dnes základem moderní umělé inteligence a významně tak ovlivňují další technologie, vědu i společnost. Popíšeme si základní principy fungování neuronových sítí a jejich aplikace. Zaměříme se na historický vývoj a klíčové milníky. V dalších částech se pak zaměříme na personální přínosy oboru umělých neuronových sítí a popisy vybraných architektur a typů sítí.

Problematika je natolik rozsáhlá, že není snadné zvolit jediný způsob její prezentace. V první části se proto zaměříme na vývoj a jednotlivé historické epochy. Cílem je ukázat, jak bouřlivý tento vývoj byl – od původních, dnes již triviálních modelů a algoritmů, které oslovovaly jen hrstku nadšenců, až po moderní obor. Dnes jde o disciplínu, která zaměstnává celé laboratoře, katedry i fakulty a proniká do všech oblastí lidské činnosti.

Aniž bychom si to explicitně uvědomovali, neuronové sítě a další prvky AI dnes zasahují do mnoha běžných digitálních procesů. Každá fotografie pořízená mobilním telefonem, každý text upravený automatickou korekturou či přeložený do jiného jazyka prochází procesy, které tyto technologie využívají. Úlohy, jako je rozpoznávání dopravních značek v asistenčních systémech moderních automobilů, automatické třídění fotografií ve vašich digitálních albech, sledování bezpečnosti v robotických linkách, defektoskopie ve výrobě, řízení, optimalizace, detekce anomálií, extrakce znalostí a pravidel z dat či doporučovací systémy, jsou příklady, kde se neuronové sítě uplatňují. Patří sem také jazykové modely a dnes velmi populární generativní modely. Tento výčet by mohl pokračovat – vše zmíněné se však opírá o metody, které jsou neuronovými sítěmi buď přímo tvořeny, nebo alespoň inspirovány.

2. Klíčové epochy vývoje neuronových sítí a umělé inteligence

Historie neuronových sítí a umělé inteligence je příběhem plným dramatických zvratů, vizionářských průlomů a nečekaných krizí. Je to cesta odvážných nápadů, které často předběhly svou dobu, a jejich autorů, kteří navzdory skeptikům věřili ve svou vizi. Tato historie je charakterizována obdobími euforického pokroku, kdy se zdálo, že hranice možností lidského poznání se neustále posouvají, ale také chvílemi hluboké nejistoty, kdy naděje na revoluční změny střídala deziluze z nenaplněných očekávání. Každá epocha odráží klíčové objevy, technologický pokrok a zároveň výzvy, které vědci museli překonat, aby tento fascinující obor posunuli k současnému významu. Je to příběh nezdolné lidské touhy porozumět světu, simulovat myšlení a přetvořit stroje na nástroje schopné skutečně „učit se“ a přemýšlet.

1. Počátky a biologická inspirace (1940–1957)

Tato éra se soustředila na první teoretické základy neuronových sítí inspirované fungováním biologických mozků.

- 1943: McCulloch a Pitts publikovali práci *A logical calculus of ideas immanent in nervous activity* [15], která představila matematický model neuronu.

- 1949: Donald Hebb formuloval Hebbovo pravidlo [7], biologicky inspirovaný mechanismus učení, popisující, jak se váhy mezi neurony mění na základě jejich společné aktivace.
- Zkoumání se zaměřovalo především na abstraktní modely, které položily teoretické základy neuronových sítí.

2. Vzestup perceptronu a první vlna optimismu (1958–1969)

Neuronové sítě získaly pozornost díky praktickým aplikacím prvních modelů.

- 1958: Frank Rosenblatt představil perceptron (viz kapitolu 3), první trénovatelnou neuronovou síť. Jeho systém Mark I demonstroval schopnost rozpoznávat jednoduché vizuální vzory.
- Perceptron ukázal potenciál učení z dat, což vedlo k nárůstu zájmu o AI.
- Nicméně omezení perceptronu (např. neschopnost řešit nelineární problémy) zůstávala nevyřešena.

3. Kritika perceptronů a první AI zima (1969–1980)

Optimismus poháněný perceptronem opadl po zveřejnění jeho zásadních omezení.

- 1969: Marvin Minsky a Seymour Papert publikovali knihu *Perceptrons*, která formálně ukázala, že jednovrstvé perceptrony nemohou řešit nelineárně separabilní úlohy, jako je např. triviální XOR problém (viz obrázek 3).
- Tato kritika vedla ke ztrátě zájmu o neuronové sítě, což způsobilo tzv. první AI zimu, během níž byla pozornost přesunuta k symbolické AI, například expertním systémům.
- Financování výzkumu neuronových sítí kleslo, což zpomalilo vývoj dalších modelů.

4. Renesance neuronových sítí a pokroky v učení (1980–1990)

Neuronové sítě zažily obnovený zájem díky novým teoretickým a praktickým objevům.

- 1982: John Hopfield představil Hopfieldovy sítě (viz kapitolu 4), které fungovaly jako asociační paměti a inspirovaly vývoj rekurentních sítí.
- 1986: Geoffrey Hinton, David Rumelhart a Ronald Williams formalizovali a popularizovali algoritmus zpětného šíření chyby (backpropagation) pro vícevrstvé perceptrony.
- 1989: Yann LeCun použil konvoluční neuronové sítě (CNN, viz kapitolu 4) pro rozpoznávání rukopisu.
- Tyto pokroky vedly k oživení zájmu o neuronové sítě, přestože technologická omezení, jako problémy mizení gradientů (vanishing gradient)¹, stále ztěžovala trénink hlubších modelů.

¹Tento termín se používá k popisu problému, kdy gradienty při zpětném šíření chyb (backpropagation) v hlubokých neuronových sítích postupně slábnou, což ztěžuje nebo znemožňuje efektivní trénink vrstev blízko vstupu sítě.

5. Druhá AI zima (1987–1993)

Nerealistická očekávání spojená s expertními systémy a omezení neuronových sítí vedly k další krizi.

- Selhání symbolické AI, zejména expertních systémů, způsobilo zklamání investorů a snížení financování.
- Investice do AI byly vnímány jako rizikové kvůli nenaplněným slibům. Očekávání, že AI rychle změní průmysl, se ukázala jako přehnaná.
- Neuronové sítě stále trpěly problémy, jako je mizení gradientů, což znemožňovalo efektivní trénink hlubokých modelů.
- Výpočetní technika té doby nebyla dostatečně výkonná pro složitější AI modely.
- AI byla vnímána jako „příliš ambiciózní“ obor. Akademická komunita se více zaměřila na alternativní oblasti, jako je tradiční počítačová věda nebo teorie algoritmů.
- Tato zima však motivovala hledání nových přístupů a zlepšení technologií.

6. Překonání problémů a příchod hlubokého učení (1990–2010)

Tato éra je charakterizována průlomem v řešení klíčových problémů neuronových sítí.

- 1997: Sepp Hochreiter a Jürgen Schmidhuber navrhli Long Short-Term Memory (LSTM, viz kapitolu 4) [8], která vyřešila problém mizení gradientů u rekurentních sítí (RNN). LSTM umožnila efektivní modelování dlouhodobých závislostí v sekvenčních datech.
- 2006: Geoffrey Hinton a jeho tým představili metody předtrénování hlubokých neuronových sítí, což položilo základy pro moderní hluboké učení.

Díky těmto inovacím se neuronové sítě staly schopnými modelovat složité vztahy a řešit problémy, které byly dříve neřešitelné.

7. Éra hlubokého učení a průmyslové dominance (2010–2020)

Neuronové sítě se staly dominantní technologií díky masivnímu rozvoji aplikací.

- 2012: AlexNet (Krizhevsky, Sutskever, Hinton) vyhrál soutěž ImageNet [11] a ukázal sílu konvolučních neuronových sítí (CNN).
- 2014: Ian Goodfellow představil generativní kontradiktorní sítě (GANs) [4], které umožnily generování realistických dat.
- 2017: Transformerové modely [23] přinesly revoluci ve zpracování přirozeného jazyka a vedly k vývoji modelů, jako jsou GPT a BERT.
- Tato éra přinesla masivní komerční využití neuronových sítí v oblastech, jako je počítačové vidění, zpracování jazyka a medicína.

8. Současná éra generativních modelů a etických výzev (2020–současnost)

Neuronové sítě se staly nedílnou součástí moderní společnosti.

- Velké jazykové modely (např. GPT-3, ChatGPT) a generativní modely (Stable Diffusion, DALL-E) získaly široké uplatnění.
- Vzrůstá zájem o otázky regulace, vysvětlitelnosti a etiky spojené s AI. Roste potřeba zajištění, že AI bude sloužit lidskému dobru a nebude zneužita například k šíření dezinformací či manipulaci.
- Výzkum směřuje k obecné umělé inteligenci (AGI) a kombinaci neuronových sítí s interdisciplinárními přístupy – kvantovým počítáním, biologií a teorií komplexních systémů. Toto naznačuje, že můžeme v ne příliš vzdálené budoucnosti vyhlížet vznik nové éry AI.
- Výzva pro lidstvo: Kam až nechat AI zajít? Současná éra přináší nejen technologický triumf, ale i bezprecedentní odpovědnost.

Toto rozdělení zohledňuje chronologickou návaznost klíčových objevů, období krizí (AI zim) i moderních průlomů, včetně řešení problémů, jako je mizení gradientů, díky architekturám, jako jsou LSTM. AI zimy přinesly důležitá ponaučení pro výzkum:

- Nutnost realistických očekávání a diskuse o schopnostech AI.
- Potřebu výkonnějšího hardware a efektivních algoritmů.
- Význam praktických aplikací pro získání financování a důvěry veřejnosti.

3. Stručný úvod do fungování neuronových sítí

Neuronové sítě (Artificial Neural Networks, ANN) jsou výpočetní modely inspirované biologickými neuronovými systémy. Fungují na principu propojených výpočetních jednotek – tzv. neuronů, které spolupracují při zpracování a transformaci dat. Cílem neuronových sítí je především schopnost modelovat složité vztahy v datech, objevovat skryté vzory a řešit úlohy, které jsou pro klasické algoritmy obtížně zvládnutelné.

Základy pro studium umělých neuronových sítí položili Warren McCulloch a Walter Pitts v roce 1943 článkem *A logical calculus of ideas immanent in nervous activity* [15]. Vytvořili tu první matematický model neuronu, který se choval jako jednoduchá logická jednotka, a ukázali, jak lze jednoduché logické funkce realizovat sítí propojených neuronů. Touto publikací začíná první éra, kterou bychom mohli označit jako „Počátky“ (viz kapitolu 2) a která bude trvat do konce šedesátých let dvacátého století.

Frank Rosenblatt následně vyvinul perceptron [18], jednu z prvních praktických implementací neuronových sítí, které se dokázaly učit na základě trénovacích dat. Perceptron se stal milníkem ve vývoji umělé inteligence a neuronových sítí, protože šlo o první výpočetní model s praktickou implementací, který byl schopen řešit reálné úlohy. Rosenblatt demonstroval perceptron na úloze klasifikace jednoduchých vizuálních vzorů. Na experimentální perceptron nazvaný Mark I, postavený na IBM 704, byl připojen optický senzor, který převáděl obraz na binární vstupy. Systém byl schopen naučit se rozpoznávat jednoduché tvary, například čtverce nebo trojúhelníky [13].

3.1. Princip fungování perceptronu

Perceptron je jednoduchý výpočetní model, který přijímá několik vstupních hodnot $(x_1, x_2, \dots, x_n)$, představujících vlastnosti zpracovávaných dat (například intenzitu pixelů u obrázku) a jeden binární výstup. Každý vstup x_i je spojen s odpovídajícím váhovým faktorem w_i . Tzv. vnitřní potenciál neuronu je výsledkem lineární kombinace vstupů

$$z = \sum_{i=1}^n w_i x_i + b, \quad (1)$$

kde n je počet vstupů perceptronu, b představuje posun, nebo práh perceptronu (bias), x_i a w_i pak jednotlivé vstupy a jejich váhy.

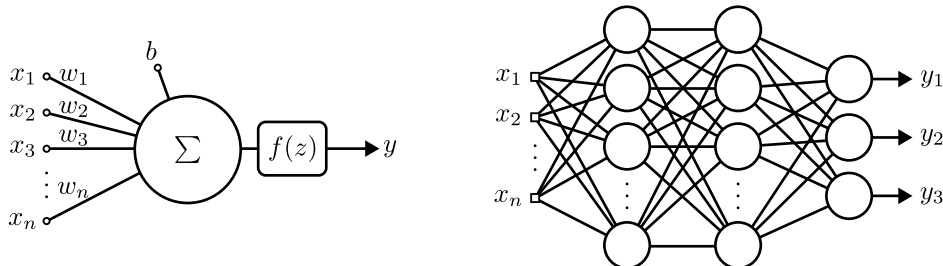
Tato lineární kombinace z je vstupem do aktivační funkce, která rozhoduje o výsledku perceptronu. V původním perceptronu byla použita jednoduchá prahová funkce

$$y = f(z) = \begin{cases} 1 & \text{pokud } z \geq 0, \\ 0 & \text{jinak.} \end{cases} \quad (2)$$

Výstupem perceptronu je tedy binární hodnota (0 nebo 1), což umožňuje klasifikaci do dvou kategorií.

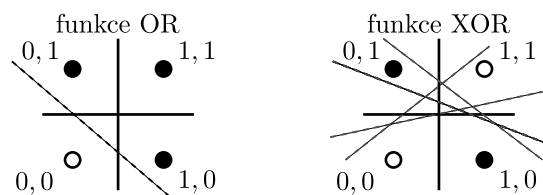
3.2. Omezení perceptronu a přechod k vícevrstevným sítím

Přestože perceptron znamenal významný krok vpřed, jeho schopnosti byly omezené. Hlavním problémem byla neschopnost řešit lineárně neseparabilní úlohy, jako je známý problém XOR. V roce 1969 tuto slabinu formálně dokázali Marvin Minsky a Seymour Papert ve své knize *Perceptrons* [16]. Kritika perceptronů vedla k útlumu zájmu o neuronové sítě na více než deset let (tzv. první AI zima, viz kapitolu 2).



Obr. 2. Ilustrace perceptronu a obecné vícevrstvé dopředné neuronové sítě se třemi výstupy. Taková síť je aplikovatelná např. v klasifikačních úlohách, kdy se na základě n vstupních příznaků klasifikuje do 3 tříd

Problém perceptronu spočíval v jeho jednovrstvé struktuře. Tento model dokáže rozdělit vstupní prostor pouze pomocí přímky (v obecném případě nadroviny), což je dostatečné pouze pro lineárně separabilní úlohy. Aby bylo možné řešit složitější nelineární problémy, bylo nutné přidat další vrstvu neuronů, tzv. skrytou vrstvu.



Obr. 3. Ilustrace rozdílu mezi lineárně a nelineárně separabilní úlohou. Zatímco u funkce OR je možné oddělit dvě třídy jednoduchou přímkou, funkce XOR takovou separaci neumožňuje. Řešení těchto nelineárně separabilních problémů vyžaduje vícevrstvé neuronové sítě s nelineárními aktivačními funkcemi

3.3. Zavedení nelineárních aktivačních funkcí a vícevrstvých sítí

Přechod k vícevrstvým neuronovým sítím (MLP, Multi-Layer Perceptrons) a zavedení nelineárních aktivačních funkcí představovaly klíčový průlom, který umožnil řešení lineárně neseparabilních úloh. Tento pokrok však nebyl možný bez dalších metod a inovací, které zajistily efektivní trénink hlubokých sítí.

Jedním z klíčových momentů bylo zavedení algoritmu zpětného šíření chyb (back-propagation), který v roce 1986 zásadně rozpracovali Geoffrey Hinton, David Rumelhart a Ronald Williams [19]. Tento algoritmus umožnil trénovat vícevrstvé sítě s nelineárními aktivačními funkcemi tím, že efektivně optimalizoval váhové parametry pomocí gradientního sestupu. Díky této metodě se neuronové sítě staly praktickým nástrojem pro modelování komplexních vztahů v datech, což položilo základy moderního hlubokého učení. Hintonův přínos byl tedy zásadní pro překonání bariér, které dříve limitovaly rozvoj neuronových sítí.

V rámci vícevrstvých perceptronů je výstup jednotlivých neuronů první (skryté) vrstvy vstupem pro neurony další vrstvy. Tato struktura umožňuje rozdělení vstupního prostoru na komplexnější rozhodovací hranice. Aby však bylo možné využít této výhody, musela být jednoduchá prahová funkce nahrazena spojitou nelineární aktivační funkcí. Klíčové příklady těchto funkcí zahrnují:

- **Sigmoidální funkce:**

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \quad (3)$$

Sigmoida mapuje vstupy do intervalu $(0, 1)$ a umožňuje plynulé přechody mezi aktivním a neaktivním stavem.

- **Hyperbolický tangens (tanh):**

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \quad (4)$$

Tanh je podobný sigmoidální funkci, ale jeho hodnoty se pohybují v rozmezí $(-1, 1)$, což usnadňuje práci se zápornými i kladnými hodnotami.

- **ReLU** (Rectified Linear Unit):

$$\text{ReLU}(z) = \max(0, z). \quad (5)$$

ReLU aktivuje pouze kladné hodnoty a díky své jednoduchosti umožňuje efektivní trénink hlubokých sítí.

Nelineární aktivační funkce umožňují, aby vícevrstvé neuronové sítě modelovaly složité vztahy ve vstupních datech, což byl zásadní předpoklad pro další rozvoj neuronových sítí.

3.4. Pokroky a nové výzvy

Zatímco vícevrstvé sítě přinesly možnost řešit mnohem širší spektrum problémů, jejich zavedení otevřelo nové otázky, například:

- Jak se vyhnout problémům mizení gradientů při trénování hlubokých sítí?
- Jak optimalizovat architektury sítí pro specifické úlohy?
- Jak zajistit, aby výsledné modely byly robustní a dobře se generalizovaly na nová data?

Tyto otázky se staly základem dalšího rozvoje neuronových sítí a vedly k objevům pokročilých architektur, jako jsou konvoluční neuronové sítě (CNN), rekurentní neuronové sítě (RNN) a modely typu Transformer. Tímto vývojem pokračuje dynamická historie neuronových sítí až do současnosti.

4. Pokročilejší architektury neuronových sítí

Neuronové sítě se od základního perceptronového modelu postupně rozvinuly do široké škály pokročilejších architektur, které umožňují řešení komplexních úloh v různých doménách. Tyto architektury přinesly nové způsoby zpracování dat, optimalizace a učení, čímž zásadně rozšířily možnosti umělé inteligence.

4.1. Hopfieldovy sítě: První průkopníci auto-asociačních pamětí

John Hopfield v roce 1982 představil novou třídu rekurentních sítí známou jako Hopfieldovy sítě. Tyto sítě fungují jako **auto-asociační paměti**, což znamená, že dokážou uložit (zapamatovat) a později si „vybavit“ vzory na základě neúplných nebo šumem narušených vstupů. Hopfieldovy sítě položily základy mnoha klíčových konceptů v oblasti rekurentních sítí a teorie dynamických systémů.

Fungování Hopfieldových sítí: Hopfieldovy sítě jsou plně propojené rekurentní neuronové sítě, kde každý neuron je spojen se všemi ostatními (s výjimkou sebe sama). Tyto sítě slouží jako asociativní paměť, která ukládá vzory a vrací stabilní stavy odpovídající těmto vzorům. Základní princip jejich fungování spočívá v iterativní aktualizaci stavů neuronů, dokud síť nedosáhne stacionárního stavu.

Počáteční stav Hopfieldovy sítě je zadán jako binární vektor $\mathbf{s} = [s_1, s_2, \dots, s_N]$, kde každý neuron s_i nabývá hodnot -1 nebo $+1$. Tento stav může být buď přesný vzor uložený během trénování, nebo jeho narušená verze obsahující šum či chyby.

Síť konverguje ke stabilnímu stavu, který odpovídá lokálnímu minimu energetické funkce (viz níže). Tento stabilní stav reprezentuje vzor, který je nejbližší zadanému vstupu podle schopností sítě.

Proces iterativní aktualizace:

1. **Inicializace:** Počáteční stavy neuronů jsou nastaveny podle vstupního vektoru $\mathbf{s}$. Síť je připravena iterativně upravovat stavy jednotlivých neuronů na základě jejich propojení a vstupní energie.
2. **Výpočet aktivace neuronů:** Pro každý neuron i se spočítá aktivace jako vážený součet stavů všech ostatních neuronů, na které je propojen,

$$a_i = \sum_j w_{ij} s_j - \theta_i, \quad (6)$$

kde w_{ij} je váha propojení mezi neurony i a j ($w_{ii} = 0$), θ_i je prahová hodnota neuronu i , s_j je aktuální stav neuronu j .

3. **Aktualizace stavu neuronů:** Stav neuronu i se aktualizuje podle pravidla

$$s_i = \begin{cases} +1 & \text{pokud } a_i \geq 0, \\ -1 & \text{jinak.} \end{cases} \quad (7)$$

Tato aktualizace probíhá opakovaně, přičemž se mohou používat dvě strategie. *Synchronní aktualizace*, kdy se stavy všech neuronů aktualizují současně, a *asynchronní (sekvenční) aktualizace*, kdy se stavy mění jeden po druhém, obvykle v náhodném pořadí. Tento přístup je stabilnější a častěji používaný v praxi.

4. **Iterace:** Proces aktualizace se opakuje, dokud se všechny stavy neuronů nepřestanou měnit, tj. dokud síť nedosáhne stacionárního stavu.

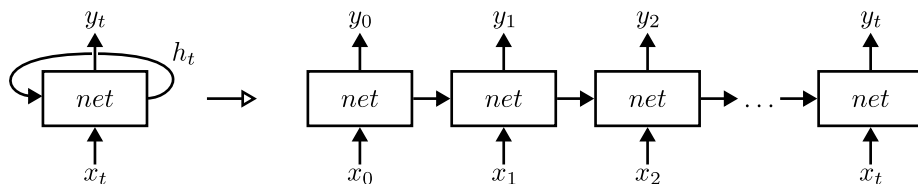
Dynamika Hopfieldových sítí je řízena minimalizací energetické funkce definované vztahem

$$E = -\frac{1}{2} \sum_{i,j} w_{ij} s_i s_j + \sum_i \theta_i s_i. \quad (8)$$

Každá aktualizace stavu neuronu s_i vede ke snížení hodnoty energetické funkce E . Tento proces zaručuje, že síť bude vždy konvergovat do stabilního stavu odpovídajícího lokálnímu minimu energie. Stabilní stav může být uložený vzor nebo jedno z „falešných“ minim, pokud je síť špatně trénovaná nebo vstupní vzor obsahuje příliš mnoho šumu.

4.2. Rekurentní neuronové sítě (RNN)

Rekurentní neuronové sítě (RNN) představují významný milník ve vývoji umělých neuronových sítí. Byly navrženy tak, aby dokázaly efektivně modelovat sekvenční data, jako jsou časové řady, řeč, přirozený jazyk nebo sekvence akcí/událostí. Na rozdíl od dopředných neuronových sítí (Feedforward Neural Networks) mají RNN zpětné vazby, což jim umožňuje přenášet informace mezi časovými kroky a vytvářet tak interní paměťovou strukturu.



Obr. 4. Rozvinutí obecné rekurentní sítě v čase. Obrázek znázorňuje obecnou strukturu, kdy skrytý stav h_t je zaveden zpět do sítě. Některé topologie místo skrytého stavu zavádějí zpětnou vazbu mezi výstupem y_t a vstupem x_t , což sice má vliv na funkcionalitu, ale nikoliv na princip rozvinutí v čase, který se využívá v některých učicích algoritmech jako např. backpropagation through time

Každý neuron v RNN zpracovává vstupní data postupně, časový krok za krokem, přičemž využívá informace ze současného vstupu i z předchozích skrytých stavů. Pro jeden časový krok t lze výpočet RNN popsat následujícími vztahy:

1. Aktualizace skrytého stavu

$$h_t = f(W_x x_t + W_h h_{t-1} + b_h), \quad (9)$$

kde:

- $x_t \in \mathbb{R}^d$ je vstupní vektor v čase t ,
- $h_{t-1} \in \mathbb{R}^m$ je skrytý stav z předchozího časového kroku,
- $W_x \in \mathbb{R}^{m \times d}$ a $W_h \in \mathbb{R}^{m \times m}$ jsou váhové matice pro vstupy a zpětné vazby,
- $b_h \in \mathbb{R}^m$ je vektor posunu (bias),
- $f: \mathbb{R}^m \rightarrow \mathbb{R}^m$ je nelineární aktivační funkce (např. tanh nebo ReLU aplikovaná po složkách),
- d je dimenze vstupního vektoru a m představuje dimenzi skrytého stavu (tj. počet skrytých neuronů).

2. Výpočet výstupu

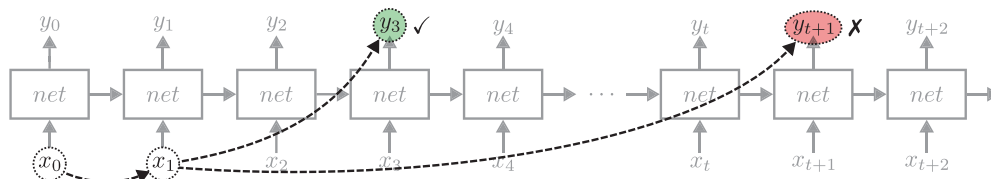
$$y_t = g(W_y h_t + b_y), \quad (10)$$

kde:

- $W_y \in \mathbb{R}^{m \times m}$ je váhová matice propojující skrytý stav s výstupem,
- $b_y \in \mathbb{R}^m$ je vektor posunu pro výstupní vrstvu,
- $g: \mathbb{R}^m \rightarrow \mathbb{R}^m$ je aktivační funkce pro výstup.

RNN tak vytváří sekvenci výstupů $y_1, y_2, \dots, y_T$ na základě sekvence vstupů $x_1, x_2, \dots, x_T$.

Pozn.: Výše uvedený popis se vztahuje k jedné konkrétní architektuře RNN, která však není jedinou možnou variantou. Některé typy RNN využívají skrytý stav a výstupní vrstvu, což znamená, že dimenze výstupního vektoru y_t nemusí odpovídat dimenzi skrytého stavu h_t . Jiné architektury naopak používají skrytý stav přímo jako výstup, a v takovém případě odpadá potřeba explicitní výstupní vrstvy aplikované na skrytý stav.



Obr. 5. Obrázek ilustruje problém dlouhodobých závislostí u rekurentních neuronových sítí. Základní RNN mají obtíže při učení spojení mezi vzdáleným kontextem a aktuální hodnotou, čímž ztrácejí schopnost efektivně pracovat s dlouhodobými závislostmi

Výhody RNN:

- Schopnost zpracovávat data libovolné délky.
- Udržování kontextu z předchozích časových kroků díky zpětným vazbám.
- Možnost modelovat sekvence s dynamickými a nelineárními vztahy.
- Úspora parametrů (počtu vah).

Problémy RNN:

- **Mizení gradientů:** Při trénování RNN pomocí algoritmu zpětného šíření v čase (*Backpropagation Through Time, BPTT*) gradienty během aktualizace váhových parametrů exponenciálně klesají (nebo rostou). To způsobuje problémy při učení dlouhodobých závislostí. Platí

$$\frac{\partial L}{\partial W_h} \sim \prod_{t=1}^T \frac{\partial h_t}{\partial h_{t-1}}, \quad (11)$$

kde L je ztrátová funkce (viz níže). Pokud derivace $\frac{\partial h_t}{\partial h_{t-1}}$ obsahuje hodnoty menší než 1, gradient rychle mizí s rostoucím T .

- **Explodování gradientů:** Pokud derivace přesahuje hodnotu 1, gradienty exponenciálně rostou, což destabilizuje trénink.

Tyto problémy byly částečně vyřešeny zavedením architektur, jako jsou Long Short-Term Memory (LSTM) a Gated Recurrent Unit (GRU).

4.2.1. Příklad definice ztrátové funkce

Ztrátová funkce rekurentní neuronové sítě

Rekurentní neuronová síť generuje sekvenci výstupů $\mathbf{y} = [y_1, y_2, \dots, y_T]$, která se porovnává s cílovou/očekávanou sekvencí (trénovacími daty) $\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T]$. K vyjádření rozdílu mezi předpověďmi sítě a skutečnými cílovými hodnotami se typicky používá ztrátová funkce přes všechny časové kroky $t = 1, 2, \dots, T$, např.

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^n (y_{t,i} - \hat{y}_{t,i})^2, \quad (12)$$

kde T je počet časových kroků v sekvenci, $y_{t,i}$ a $\hat{y}_{t,i}$ jsou jednotlivé složky vektorů $\mathbf{y}$ a $\hat{\mathbf{y}}$.

Backpropagation Through Time

BPTT je algoritmus pro optimalizaci parametrů (vah) rekurentních neuronových sítí. Rozkládá výpočet gradientu ztrátové funkce na jednotlivé časové kroky, aby bylo možné trénovat síť pomocí gradientního sestupu. Proces lze shrnout takto:

1. **Rozvinutí sítě v čase:** Rekurentní síť je „rozvinuta“ přes časové kroky t , což vytvoří sérii propojených vrstev. Stav skrytých neuronů $\mathbf{h}_t$ na každém kroku je definován jako

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t; \mathbf{W}), \quad (13)$$

kde $\mathbf{h}_t$ je stav na kroku t , $\mathbf{x}_t$ je vstup, $\mathbf{W}$ jsou váhové parametry a f je aktivační funkce.

2. **Ztrátová funkce:** Celková ztrátová funkce je suma přes časové kroky

$$L = \sum_{t=1}^T \ell(y_t, \hat{y}_t). \quad (14)$$

3. **Výpočet gradientu:** Gradient ztrátové funkce vzhledem k parametrům $\mathbf{W}$ v časovém kroku t ovlivňuje nejen stav $\mathbf{h}_t$, ale i stavy z předchozích kroků $\mathbf{h}_{t-1}, \mathbf{h}_{t-2}, \dots$,

$$\frac{\partial L}{\partial \mathbf{W}} = \sum_{t=1}^T \frac{\partial L}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \mathbf{W}}, \quad (15)$$

kde $\frac{\partial L}{\partial \mathbf{h}_t}$ zahrnuje součet zpětně šířených gradientů přes všechny následující časové kroky.

4. **Aktualizace parametrů:** Parametry $\mathbf{W}$ jsou aktualizovány pomocí gradientního sestupu

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial L}{\partial \mathbf{W}}, \quad (16)$$

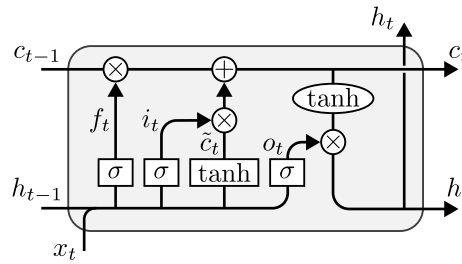
kde η je učicí konstanta (learning rate).

Klíčové body BPTT

- **Rozpad gradientu:** V důsledku opakovaného násobení gradientů přes mnoho časových kroků dochází k problémům, jako je mizení (vanishing gradients) nebo explodování gradientu.
- **Truncated BPTT:** Aby se snížila výpočetní náročnost, používá se často zkrácená verze BPTT, která zpětně šíří gradienty pouze přes omezený počet časových kroků.

4.2.2. LSTM a GRU

LSTM: Long Short-Term Memory (LSTM) je rozšíření RNN, které obsahuje speciální mechanismy nazývané *brány*, které umožňují řídit tok informací v síti.



Obr. 6. Neuron síť LSTM (taktéž LSTM cell). Je zde vidět vstupní vektor, stav neuronu, skrytý stav a rovněž vnitřní struktura v podobě bran (gates)

Výpočet bran v LSTM:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (\text{zapomínací brána}), \quad (17)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (\text{vstupní brána}), \quad (18)$$

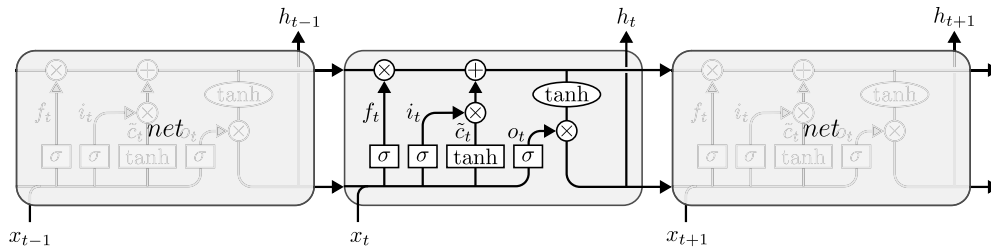
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (\text{výstupní brána}), \quad (19)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (\text{kandidátní stav}), \quad (20)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (\text{aktualizace stavu}), \quad (21)$$

$$h_t = o_t \odot \tanh(c_t) \quad (\text{skrytý stav}). \quad (22)$$

- $f_t, i_t, o_t \in \mathbb{R}^m$ jsou aktivační vektory pro brány řídící tok informací,
- $c_t \in \mathbb{R}^m$ je stav buňky uchovávající dlouhodobé informace,
- $\odot$ představuje násobení vektorů po složkách. Brány tak slouží k selekci nebo modulaci informací tím, že vynulují nebo váží určité prvky vektoru.



Obr. 7. Ukázka rozvinutí LSTM sítě v čase

GRU: Gated Recurrent Unit (GRU) je jednodušší varianta LSTM, která kombinuje některé brány a snižuje počet parametrů:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (\text{aktualizační brána}), \quad (23)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (\text{resetovací brána}), \quad (24)$$

$$\tilde{h}_t = \tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h) \quad (\text{kandidátní stav}), \quad (25)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (\text{skrytý stav}). \quad (26)$$

Aplikace RNN, LSTM a GRU:

- **Zpracování přirozeného jazyka:** automatický překlad, jazykové modelování, generování nebo sumarizace textů.
- **Analýza časových řad:** predikce akciových trhů, sledování senzorických dat, klasifikace sekvenčních dat.
- **Rozpoznávání řeči:** převod řeči na text.
- **Generování sekvencí:** generování textu, skladeb, řídicích signálů, atd.

Rekurentní sítě a jejich varianty, jako jsou LSTM a GRU, byly průkopnickými přístupy pro modelování časových závislostí a sekvencí. Díky těmto architekturám se staly jedním z klíčových nástrojů v oblasti umělé inteligence a hlubokého učení.

4.3. Radial Basis Function (RBF) sítě

Radial Basis Function (RBF) [5] jsou neuronové sítě založené na *radiálně symetrických aktivačních funkcích*, které reagují na vzdálenost vstupu od určitého centra. Jsou obzvláště vhodné pro úlohy klasifikace, interpolace, regrese a funkční aproximace.

Základní struktura: RBF sítě mají jednoduchou architekturu, která obvykle obsahuje tři vrstvy:

1. Vstupní vrstva:

- Každý neuron přijímá jednu dimenzi vstupního vektoru $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top$.
- Tato vrstva pouze předává vstupy do další vrstvy.

2. Skrytá vrstva:

- Každý neuron ve skryté vrstvě používá *radiální aktivační funkci* závislou na vzdálenosti vstupu $\mathbf{x}$ od centra $\mathbf{c}_j$.
- Výstup h_j j -tého neuronu je dán vztahem

$$h_j(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{c}_j\|), \quad (27)$$

kde $\|\mathbf{x} - \mathbf{c}_j\|$ je eukleidovská vzdálenost mezi vstupem $\mathbf{x}$ a centrem $\mathbf{c}_j$, ϕ je radiální aktivační funkce, například gaussovská

$$\phi(r) = \exp\left(-\frac{r^2}{2\sigma_j^2}\right), \quad (28)$$

kde σ_j je šířka (rozptyl) funkce.

3. Výstupní vrstva – lineární kombinace výstupů ze skryté vrstvy

$$y_k = \sum_{j=1}^m w_{kj} h_j(\mathbf{x}), \quad (29)$$

kde w_{kj} jsou váhové parametry spojující j -tý skrytý neuron s k -tým výstupem, m je počet neuronů ve skryté vrstvě.

Trénink RBF sítí: Trénink RBF sítí probíhá ve dvou hlavních krocích:

1. Určení parametrů skryté vrstvy:

- Centra $\mathbf{c}_j$ jsou často vybírána buď náhodně, nebo pomocí metod shlukování (např. algoritmus k -means).
- Šířky σ_j mohou být nastaveny na základě maximální vzdálenosti mezi centry nebo jako průměrná vzdálenost mezi nejbližšími centry.

2. Optimalizace váhových parametrů:

- Váhy w_{kj} se určují pomocí lineární regrese tak, aby minimalizovaly chybu mezi skutečnými výstupy $\mathbf{y}$ a predikovanými výstupy $\hat{\mathbf{y}}$, tj.

$$\min_W \|\mathbf{y} - \hat{\mathbf{y}}\|_2. \quad (30)$$

4.3.1. Klíčové vlastnosti RBF

RBF sítě stejně jako vícevrstvé perceptrony (MLP) patří mezi univerzální aproximátory [17], což znamená, že jsou schopny aproximovat libovolnou spojitou funkci s libovolnou přesností, pokud mají dostatečný počet neuronů ve skryté vrstvě. U RBF sítí tato schopnost vyplývá z jejich mechanismu modelování lokálních závislostí pomocí radiálních funkcí.

Radiální aktivační funkce mají lokální charakter, což znamená, že každý neuron reaguje pouze na omezenou oblast vstupního prostoru. Tato vlastnost zlepšuje schopnost sítě generalizovat.

Trénink RBF sítí je rychlejší než u vícevrstvých perceptronů, protože váhy výstupní vrstvy lze optimalizovat lineární regresí. RBF sítě dokáží řešit nelineárně separabilní úlohy díky použití nelineárních aktivačních funkcí.

Oproti MLP, které umožňují libovolný počet skrytých vrstev, mají RBF sítě pevnou strukturu (vstupní-skrytá-výstupní vrstva). RBF sítě mají jednodušší a rychlejší trénink, protože optimalizace výstupní vrstvy je lineární. Naproti tomu MLP vyžadují složitější zpětné šíření chyb (backpropagation).

RBF sítě jsou výjimečně užitečné v aplikacích, kde je zapotřebí rychlá funkční aproximace nebo robustní klasifikace s menším počtem datových bodů. Díky svým vlastnostem a flexibilitě zůstávají důležitou architekturou neuronových sítí pro širokou škálu úloh.

4.4. Samoorganizující se mapy (SOM)

Samoorganizující se mapy [10], které navrhl Teuvo Kohonen v roce 1980, představují specifický typ neuronových sítí zaměřených na *učení bez učitele* (unsupervised learning). Na rozdíl od tradičních vícevrstvých perceptronů či konvolučních neuronových sítí, které vyžadují trénink s cílovými hodnotami (supervised learning), SOM fungují samostatně na základě podobnosti mezi datovými vzory. Jejich klíčovou funkcí je schopnost redukovat dimenzionalitu dat a převádět složité, vysokodimenzionální datové prostory do snadno interpretovatelných dvourozměrných nebo trojrozměrných map. Tato vlastnost umožňuje vizualizaci a analýzu dat, která by jinak byla pro lidskou intuici obtížně pochopitelná.

Kohonenovy mapy vznikly jako odpověď na potřebu nalézt efektivní způsob analýzy a vizualizace datových souborů s vysokou dimenzionalitou. Klasické neuronové sítě, jako jsou MLP, se soustředí na explicitní klasifikaci nebo regresi, ale nejsou přímo navrženy pro explorativní analýzu dat bez známých cílových hodnot. SOM řeší problém zobrazení datových vztahů v nižším rozměru, přičemž zachovávají jejich topologickou strukturu. To znamená, že podobná data jsou mapována na blízké neurony v mřížce, zatímco odlišná data jsou oddělena. Tento přístup je obzvláště užitečný při identifikaci skrytých vzorů nebo shluků v datech.

4.4.1. Princip fungování

SOM se skládají z *mřížky neuronů*, kde každý neuron má svůj váhový vektor, který odpovídá určitému bodu ve vstupním datovém prostoru. Během tréninku se váhy neuronů iterativně přizpůsobují tak, aby co nejlépe odpovídaly vstupním datům. Proces zahrnuje následující klíčové kroky:

1. **Porovnání:** Pro každý vstupní vektor se identifikuje neuron, jehož váhový vektor je nejbližší (tzv. vítězný neuron nebo BMU – Best Matching Unit).
2. **Aktualizace:** Váhy vítězného neuronu i jeho okolních neuronů se přiblíží ke vstupnímu vektoru. Tento krok je řízen funkcí sousedství, která zajišťuje, že blízké neurony se adaptují společně.
3. **Iterace:** Proces se opakuje s různými vstupními vzory, přičemž se postupně snižují parametry, jako je koeficient učení a šířka sousedství.

Na rozdíl od MLP, které se učí minimalizovat chybu pomocí gradientního sestupu, SOM neoptimalizují specifickou ztrátovou funkci. Namísto toho jsou trénovány tak, aby organizovaly data do přirozených struktur a odhalovaly vztahy mezi vzory. Tato samoorganizace poskytuje vhled do datové struktury, aniž by bylo nutné definovat konkrétní cílové výstupy.

4.4.2. Aplikace SOM

Díky své schopnosti analyzovat a vizualizovat složitá data našly SOM široké uplatnění v různých oblastech:

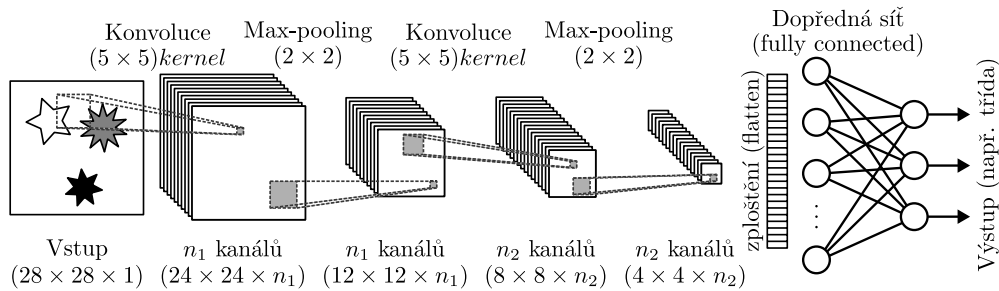
- Analýza genových expresí a zkoumání genetických podobností.
- Segmentace zákazníků a analýza tržních dat.
- Detekce anomálií, poruchová diagnostika, klasifikace patientských dat.
- Komprese obrazů a rozpoznávání vzorů.

Samoorganizující se mapy inspirovaly další pokročilé metody vizualizace dat, jako jsou t-SNE nebo UMAP, které však často neposkytují explicitní interpretaci vztahů mezi daty. Kohonenovy mapy zůstávají důležitou součástí historie neuronových sítí a ukazují, jak mohou biologické principy vést k inovativním výpočetním přístupům. Tento koncept byl přímým přínosem pro aplikace, kde bylo klíčové pochopit a zjednodušit komplexní datové prostory.

4.5. Konvoluční neuronové sítě

Konvoluční neuronové sítě (CNN) jsou specifickou architekturou neuronových sítí navrženou pro efektivní zpracování dat, která mají prostorovou strukturu, například obrázků. CNN byly poprvé navrženy Kunihiko Fukushima v podobě neocognitronu v roce 1980, ale skutečný průlom přinesl Yann LeCun v roce 1989, kdy použil CNN (model LeNet) pro rozpoznávání rukopisných číslic na poštovních směrovacích číslech [12].

CNN byly zavedeny jako řešení omezení plně propojených sítí při zpracování obrazových dat. Plně propojené sítě mají velké množství parametrů, což vede k vysoké výpočetní náročnosti a riziku přeučení, zejména u dat s prostorovou strukturou, jako jsou obrazy. CNN využívají konvoluční vrstvy, které aplikují filtry na lokální oblasti vstupních dat, čímž zachycují prostorové rysy (např. hrany a textury) s menším počtem parametrů. Díky sdílení vah a hierarchickému zpracování rysů umožňují CNN efektivní extrakci informací a škálovatelnost, což z nich činí ideální nástroj pro úlohy počítačového vidění, jako je např. rozpoznávání objektů.



Obr. 8. Typická architektura konvoluční neuronové sítě. První část typicky pracuje jako tzv. „feature extractor“, na který je následně napojena dopředná (typicky plně propojená) neuronová síť. Na obrázku je vidět jednoduchý příklad klasifikátoru, který má na vstupu monochromatický (jednokanálový) obraz a na výstupu tři třídy

4.5.1. Základní mechanismus a architektura

CNN se skládají z několika vrstev, přičemž klíčové jsou:

1. Konvoluční vrstvy (Convolutional Layers):

- Hlavní funkcí konvoluční vrstvy je extrakce prostorových rysů z dat pomocí aplikace filtru (*kernel*).
- Konvoluce se vypočítává vztahem

$$Y[i, j, k] = X * W_k + b_k = \sum_{m, n, c} X[i + m, j + n, c] \cdot W_k[m, n, c] + b_k, \quad (31)$$

kde $X[i, j, c]$ jsou vstupní data na pozici (i, j) v kanálu c , $W_k[m, n, c]$ je filtr pro kanál c a výstupní kanál k , b_k je posun pro výstupní kanál k , $Y[i, j, k]$ je výstupní hodnota na pozici (i, j) v kanálu k .

- Výsledkem je tzv. matice příznaků (*feature map*), která reprezentuje rysy detekované filtrem.

2. Aktivační funkce (Activation Function): Po konvoluci se na každý výstup aplikuje nelineární aktivační funkce, např. ReLU (*Rectified Linear Unit*)

$$f(x) = \max(0, x). \quad (32)$$

– ReLU přispívá k rychlé konvergenci během trénování tím, že eliminuje saturaci gradientů.

3. Poolingové vrstvy (Pooling Layers):

- Tyto vrstvy redukují dimenzionalitu dat a zachovávají klíčové informace.
- Nejčastější operací je *max-pooling*, definovaný jako

$$Y[i, j] = \max_{m, n} X[i + m, j + n], \quad (33)$$

kde m, n reprezentují velikost poolingového okna.

4. Plně propojené vrstvy (Fully Connected Layers):

- Výstupní feature mapy jsou převedeny na jeden vektor a zpracovány tradiční neuronovou sítí.

4.5.2. Pokročilé varianty CNN

Residual Networks (ResNet) je architektura neuronové sítě, která byla navržena k řešení problémů spojených s trénováním velmi hlubokých sítí, jako je mizení gradientů a zhoršení přesnosti při přidávání dalších vrstev. Klíčovým inovativním prvkem ResNet je zavedení *reziduálních bloků* [6], které umožňují přeskoky (skip connections) přes jednu nebo více vrstev. Tyto přeskoky přenášejí vstupní informace přímo k pozdějším vrstvám, obcházejí některé výpočty a umožňují tak efektivnější učení.

Každý reziduální blok má formu: $y = f(x) + x$, kde $f(x)$ představuje transformaci vstupu x pomocí vrstev sítě (například konvolučních a aktivačních funkcí) a x je přímá cesta (skip connection). Výsledný výstup y kombinuje transformovaný signál $f(x)$ s původním vstupem x . Tento mechanismus usnadňuje učení identického zobrazení ($f(x) = 0$), pokud je to potřeba, což pomáhá při optimalizaci hlubokých sítí.

ResNet umožňuje trénování extrémně hlubokých modelů (sítě s více než 100 nebo 1 000 vrstvami) a zlepšuje jejich výkonnost díky stabilnějšímu šíření gradientů a lepší schopnosti generalizace. Tato architektura je široce využívána v počítačovém vidění, zejména v úlohách rozpoznávání objektů, detekce a segmentace obrazů.

Dense Convolutional Network (DenseNet) je architektura neuronové sítě, která zlepšuje efektivitu učení a využití parametrů díky *hustému propojení vrstev*. Na rozdíl od běžných architektur, kde je každá vrstva propojena pouze se sousedními vrstvami, v DenseNet je každá vrstva přímo spojena se všemi předchozími vrstvami. To znamená, že výstup každé vrstvy slouží jako vstup pro všechny následující vrstvy.

Každá vrstva v DenseNet přijímá jako vstup

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]),$$

kde x_l je výstup l -té vrstvy, $[x_0, x_1, \dots, x_{l-1}]$ představuje spojení všech předchozích výstupů (tzv. concatenation), a H_l je nelineární transformace, obvykle se skládající z konvoluční operace, aktivační funkce a normalizace.

DenseNet je známý pro svou výbornou výkonnost v úlohách počítačového vidění, jako je klasifikace obrazů a segmentace, a to při nižším počtu parametrů než srovnatelné modely, jako je ResNet. Architektura je zvláště výhodná v aplikacích, kde je důležité maximalizovat efektivitu výpočtu a využití paměti.

4.5.3. Generativní kontradiktorní sítě (GANs)

GAN (Generative Adversarial Networks) jsou inovativní architekturou neuronových sítí, kterou v roce 2014 představil Ian Goodfellow [4]. GAN sestávají ze dvou sítí: *generátoru*, který vytváří syntetická data, a *diskriminátoru*, který rozlišuje mezi reálnými a generovanými daty. Obě sítě se trénují v procesu, který připomíná soutěž – generátor se snaží „oklamat“ diskriminátor, zatímco diskriminátor se snaží zlepšit

své schopnosti rozpoznat „podvod“. Tento přístup umožňuje generátorům produkovat realistická data, jako jsou obrazy, texty nebo zvukové stopy, a našel široké využití v oblastech od počítačového vidění po generativní umění a syntézu dat.

4.5.4. Konvoluční Long Short-Term Memory (ConvLSTM)

Architektura ConvLSTM spojuje vlastnosti CNN (konvoluční vrstvy) a RNN (paměťové buňky LSTM) [20]. Využívá konvoluční operace pro prostorové zpracování dat v kombinaci s pamětí pro sekvenční zpracování.

Místo tradičních maticových operací ve vrstvách LSTM využívá ConvLSTM konvoluce k zachycení lokálních prostorových rysů. Stav buňky (c_t) i skrytý stav (h_t) v čase t mají tvar tenzorů, což umožňuje modelu efektivněji zachytit prostorovou i časovou dynamiku. ConvLSTM nachází uplatnění v úlohách, jako je předpověď počasí, analýza videí nebo detekce anomálií v časoprostorových datech.

4.5.5. Aplikace CNN

- **Počítačové vidění:** detekce objektů, rozpoznávání obličejů, analýza zdravotnických snímků (CT, MRI).
- **Automatické řízení:** identifikace dopravních značek, detekce překážek.
- **Generativní modely GAN** pro syntetická data.
- **Analýza videí** identifikace aktivit, predikce pohybu.

CNN a jejich varianty, jako jsou ResNet, DenseNet a GANs, přinesly revoluci ve zpracování dat díky své schopnosti efektivně extrahovat a modelovat komplexní prostorové a časové vzory.

4.6. Transformery

Transformery jsou klíčovou architekturou neuronových sítí, která způsobila revoluci v oblasti zpracování sekvenčních dat, zejména přirozeného jazyka. Poprvé byly představeny v práci *Attention is all you need* [23] a jejich hlavní inovací je mechanismus *self-attention*, který umožňuje modelu přiřadit různou váhu jednotlivým částem vstupní sekvence v závislosti na jejich relevanci pro aktuální úlohu. Na rozdíl od rekurentních neuronových sítí (RNN) zpracovávají transformery vstupní data paralelně, což výrazně zlepšuje efektivitu a umožňuje škálování na velké datové soubory. Důležitou součástí architektury je také použití pozičních kódů, které zachovávají informaci o pořadí tokenů v sekvenci.

Transformery tvoří základ moderních jazykových modelů, jako jsou BERT, GPT nebo T5. Model BERT využívá architekturu transformeru pro dvojstranné porozumění textu, což jej činí ideálním pro úlohy jako klasifikace, odpovídání na otázky nebo pojmenované entity. Na druhé straně modely GPT (např. GPT-3) využívají jednosměrné generování textu a excelují v generativních úlohách, jako je tvorba přirozeného textu, překlad nebo programování.

Transformery se staly standardem nejen ve zpracování přirozeného jazyka (*Natural Language Processing* – NLP), ale také v dalších oblastech, jako je počítačové vidění (Vision Transformers), bioinformatika nebo zpracování hudby. Jejich flexibilita a schopnost efektivně modelovat složité vztahy v datech přinesly zásadní průlom v umělé inteligenci a otevřely cestu k vývoji stále sofistikovanějších jazykových modelů a generativních systémů.

5. Interdisciplinární přístupy a komplexní systémy

5.1. Neuronové sítě a evoluční algoritmy

Evoluční algoritmy (EA) vznikly jako výpočetní technika inspirovaná Darwinovou teorií přirozeného výběru. Jejich principy byly poprvé formálně popsány v 60. a 70. letech 20. století. V roce 1964 John Holland definoval základní rámec genetických algoritmů (GA) [9], který se stal základem pro mnoho dalších evolučních metod. Evoluční algoritmy se postupně rozšířily o varianty, jako jsou evoluční strategie (Rechenberg, 1973), evoluční programování (Fogel, 1966) a genetické programování (Cramer, 1985) [14].

5.1.1. Principy evolučních algoritmů

Evoluční algoritmy napodobují přirozený výběr a adaptaci v populaci jedinců, kde každý jedinec představuje potenciální řešení daného problému:

1. **Populace jedinců** – Každý jedinec je reprezentován genomem, obvykle ve formě binárního, celočíselného nebo reálného vektoru.
2. **Hodnocení** (fitness) – Každý jedinec je ohodnocen podle schopnosti řešit problém pomocí tzv. fitness funkce $f(x)$.
3. **Selekce** – Jedinci s vyšší fitness mají větší šanci předat své vlastnosti další generaci. – Selekcční mechanismy zahrnují turnajovou selekci, ruletový výběr nebo rankování.
4. **Křížení** (crossover) – Kombinace genetické informace dvou nebo více jedinců pro vytvoření potomků.
5. **Mutace** – Náhodná změna některých genů pro udržení genetické diverzity.
6. **Evoluční iterace** – Proces selekce, křížení a mutace se opakuje, dokud není dosaženo optimálního řešení nebo nevyprší stanovený počet iterací.

5.1.2. Spojení s neuronovými sítěmi (neuroevoluce)

Evoluční algoritmy se ukázaly jako efektivní metoda pro optimalizaci neuronových sítí v několika směrech:

1. Optimalizace parametrů

- Evoluční algoritmy nahrazují gradientní sestup při hledání optimálních vah sítě.

- Výhodou je schopnost vyhnout se lokálním extrémům ve složitých optimalizačních problémech

$$\mathbf{w}^* = \arg \max_{\mathbf{w}} f(\mathbf{w}), \quad (34)$$

kde $f(\mathbf{w})$ je fitness funkce² vyjadřující výkonnost sítě na validační sadě.

2. Návrh architektury sítí

- Evoluční algoritmy navrhnou optimální topologie neuronových sítí, včetně počtu vrstev, neuronů a propojení.
- Například NEAT (NeuroEvolution of Augmenting Topologies) [22] kombinuje evoluci topologií a parametrů.

3. Hyperparametrická optimalizace

- Evoluční algoritmy hledají optimální hodnoty hyperparametrů, jako je rychlost učení, velikost dávky nebo typ aktivačních funkcí.

5.2. Genetické programování (GP)

Genetické programování (GP) je varianta evolučních algoritmů, která se zaměřuje na vývoj programů, funkcí nebo pravidel reprezentujících optimální řešení daného problému. Na rozdíl od klasických genetických algoritmů, které pracují s pevně definovanými vektory, GP využívá stromové struktury pro reprezentaci komplexnějších výpočetních procesů. Díky této flexibilitě se GP stalo mocným nástrojem pro návrh adaptivních neuronových sítí.

5.2.1. Základní principy genetického programování

1. **Reprezentace jedinců** – Každý jedinec v populaci je reprezentován stromem, kde uzel představuje operaci nebo funkci (např. $+$, $-$, $*$, $/$, $\sin$, $\cos$).
2. **Evoluční operátory**
 - **Selekce** – Výběr jedinců na základě fitness funkce.
 - **Křížení** – Kombinace částí dvou stromů pro vytvoření nových jedinců.
 - **Mutace** – Náhodná změna uzlů nebo větví stromu.
3. **Fitness funkce** – Hodnotí kvalitu jedince na základě schopnosti řešit daný problém, například minimalizací chyby předpovědi nebo maximalizací přesnosti klasifikace.

²Poznámka: oproti chybové funkci, kterou se snažíme gradientním sestupem minimalizovat, u fitness funkce hledáme maximum. Někdy lze v kontextu evolučních algoritmů narazit na pojem „cost function“, která je obdobou chybové funkce, a takovou pak opět minimalizujeme.

5.2.2. HyperGP: Evoluční design topologií neuronových sítí

HyperGP [1] je rozšířením přístupu HyperNEAT [21], který kombinuje genetické programování s principy nepřímého kódování (indirect encoding). Cílem HyperGP je generovat efektivní topologie neuronových sítí, které dokáží řešit složité úlohy pomocí evoluční optimalizace.

1. Nepřímé kódování

- Namísto přímého popisu každého spojení mezi neurony se používá pravidel nebo funkcí k odvození celkové struktury sítě.
- Každý jedinec v GP reprezentuje pravidlo, jak se mají propojit neurony, například

$$w_{ij} = f(i, j, d), \quad (35)$$

kde w_{ij} je váha propojení mezi neurony i a j , d je vzdálenost mezi nimi a f je funkce reprezentovaná stromem GP.

2. **Využití geometrie** – HyperGP zahrnuje prostorové uspořádání neuronů, což umožňuje navrhovat topologie s využitím lokálních i globálních pravidel.
3. **Optimalizace aktivací a propojení** – HyperGP umožňuje návrh sítí se specializovanými funkcemi aktivace a propojení, které se přizpůsobují konkrétní úloze.

5.2.3. Konstrukce neuronových sítí pomocí GP a HyperGP

Genetické programování je obzvláště vhodné pro návrh architektur neuronových sítí, které zahrnují:

- **Hluboké sítě s adaptivní topologií** – GP generuje vrstvy s různým počtem neuronů a jejich propojení.
- **Vrstvy se specializovanými aktivacemi** – Stromy mohou obsahovat nelineární aktivační funkce (např. ReLU, tanh, sigmoid) nebo jejich kombinace.
- **Vrstvy s dynamickými parametry** – Například HyperGP může navrhnout dynamické změny v propojení během trénování.

5.2.4. Výhody HyperGP

- **Efektivní design** – Automatizace návrhu topologií šetří čas a eliminuje potřebu manuálního ladění.
- **Flexibilita** – HyperGP může generovat sítě pro široké spektrum aplikací, od jednoduchých klasifikačních úloh po komplexní systémy řízení.
- **Robustnost** – Evolučně navržené sítě často vykazují vyšší odolnost vůči šumu v datech.

Genetické programování, zejména v kombinaci s nepřímým kódováním v HyperGP, představuje pokročilý přístup k automatizovanému návrhu neuronových sítí. Tento směr výzkumu otevírá nové možnosti pro efektivní řešení složitých úloh v různých oblastech.

Spojení neuronových sítí, evolučních algoritmů a komplexních systémů umožňuje řešit problémy, které byly dříve považovány za neřešitelné, a otevírá nové možnosti pro aplikace v průmyslu, vědě i umění.

V této kapitole jsme ukázali, jak se neuronové sítě rozšířily od základních perceptronů až po pokročilé architektury, jako jsou RNN, CNN nebo SOM. Každá z těchto sítí přinesla unikátní přístupy k řešení problémů, čímž přispěla k rozvoji umělé inteligence a její aplikaci v různých oborech. Laureáti Nobelovy ceny Geoffrey Hinton a John Hopfield hráli klíčovou roli v překonání technických výzev a položili základy moderní AI. Budoucnost tohoto oboru slibuje další fascinující objevy, které budou stavět na těchto historických milnících.

6. Závěr

Příběh umělých neuronových sítí, jejich vývoje a přínosů pro současnou společnost je nerozlučně spjat s přínosy Johna Hopfielda a Geoffreyho Hintona, laureátů Nobelovy ceny za fyziku (2024). Oba vědci sehráli klíčovou roli v proměně neuronových sítí z abstraktních matematických modelů v prakticky využitelné technologie, které dnes ovlivňují mnoho oblastí našeho života.

John Hopfield svými průkopnickými pracemi na asociačních pamětech a rekurentních neuronových sítích ukázal, jak mohou sítě modelovat dynamiku a dosáhnout stabilních stavů, což otevřelo cestu k aplikacím v optimalizaci a biologicky inspirovaných výpočtech. Jeho přístup založený na propojení fyziky, biologie a informatiky vytvořil nový rámec pro chápání neuronových sítí jako komplexních systémů. Hopfield také zdůraznil význam interdisciplinarity, což napomohlo vytvoření komunity výzkumníků napříč obory.

Geoffrey Hinton, často označovaný za „kmitra hlubokého učení“, přispěl zásadním způsobem k rozvoji vícevrstvých neuronových sítí, algoritmu zpětného šíření chyb a hlubokého učení. Díky jeho práci se neuronové sítě staly schopnými modelovat složité vztahy v datech a řešit úlohy, které byly dříve mimo dosah tradičních přístupů. Hintonovy obavy ohledně bezpečnosti a regulace AI dnes rezonují vědeckou komunitou a upozorňují na nutnost odpovědného vývoje a využívání těchto technologií.

Budoucnost, kterou oba vědci nastínili, nabízí nespočet příležitostí, ale také výzev. Hopfieldova vize zdůrazňuje důležitost porozumění vztahu mezi strukturou a dynamikou komplexních systémů, zatímco Hinton varuje před riziky spojenými s nekontrolovatelným rozvojem umělé inteligence. Oba zároveň vyzdvihují potřebu interdisciplinárního přístupu a spolupráce, která by mohla vést k vytvoření obecné umělé inteligence (AGI) a nových inovací v oblastech, jako je kvantová fyzika, biologie nebo teorie komplexních systémů.

Jejich práce nejen inspirovala celé generace vědců, ale také výrazně změnila technologický a vědecký svět. Rozpoznávání vzorů, automatický překlad, analýza obrazových dat, predikce časových řad – to vše a mnohem více dnes stojí na základech, které položili tito laureáti. Jak ale sami varují, s velkou silou neuronových sítí přichází i velká

odpovědnost. Tato Nobelova cena není jen oslavou dosažených výsledků, ale také výzvou k tomu, abychom tyto technologie využívali k dobru společnosti a budoucích generací.

Je však důležité zdůraznit, že úspěchy Johna Hopfielda a Geoffreyho Hintonu stojí na širokých základech práce mnoha dalších výjimečných vědců, bez nichž by se obor neuronových sítí a umělé inteligence nemohl vyvinout do současné podoby. Warren McCulloch a Walter Pitts položili teoretické základy modelu neuronu, Donald Hebb přispěl svým pravidlem o učení, Frank Rosenblatt vytvořil perceptron jako první praktickou implementaci neuronové sítě. Další průlom přinesl Yann LeCun, jehož práce na konvolučních sítích umožnila revoluci v počítačovém vidění, Sepp Hochreiter a Jürgen Schmidhuber, kteří navrhli LSTM a umožnili efektivní modelování dlouhodobých závislostí v datech, nebo Ian Goodfellow, jehož generativní kontradiktorní sítě (GANs) otevřely zcela nové možnosti v generování dat. V neposlední řadě je třeba zmínit také vědce jako Marvin Minsky, Seymour Papert nebo David Rumelhart, kteří svými přínosy formovali teoretické a praktické směry výzkumu. Nobelova cena pro Hopfielda a Hintonu je tak nejen uznáním jejich individuálních přínosů, ale také oslavou kolektivního úsilí celé vědecké komunity, která se podílela na formování tohoto fascinujícího oboru.

S dalším rozvojem technologií inspirovaných fungováním mozku se klíčovou výzvou stává dosažení hlubšího porozumění, efektivní kontroly a etického využití umělé inteligence – což jsou hodnoty, které oba laureáti zdůrazňují a předávají budoucím generacím.

L i t e r a t u r a

- [1] BUK, Z., KOUTNÍK, J., ŠNOREK, M.: *NEAT in HyperNEAT substituted with genetic programming*. In: M. Kolehmainen, P. Toivanen, B. Beliczynski (eds.): *Adaptive and Natural Computing Algorithms*, Springer, 2009, 243–252.
- [2] GEOFFREY, H., SEJNOWSKI, T.: *Optimal perceptual inference*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1983.
- [3] GEOFFREY, H. E., SEJNOWSKI, T. J.: *Learning and relearning in Boltzmann machines*. In: D. E. Rumelhart, J. L. McClelland (eds.): *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, MIT Press, 1986, 282–317.
- [4] GOODFELLOW, I. J., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A., BENGIO, Y.: *Generative adversarial networks* [online]. <https://arxiv.org/abs/1406.2661>
- [5] HAYKIN, S. S.: *Neural networks and learning machines*. Third edition, Pearson Education, 2009.
- [6] HE, K., ZHANG, X., REN, S., SUN, J.: *Deep residual learning for image recognition*. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, 770–778.
- [7] HEBB, D.: *The organization of behavior: A neuropsychological theory*. John Wiley, 1949.
- [8] HOCHREITER, S., SCHMIDHUBER, J.: *Long short-term memory*. *Neural Comput.* 9 (1997), 1735–1780.

- [9] HOLLAND, J. H.: *Adaptation in natural and artificial systems*. MIT Press, 1992.
- [10] KOHONEN, T.: *Self-organized formation of topologically correct feature maps*. Biol. Cybernet. 43 (1982), 59–69.
- [11] KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E.: *Imagenet classification with deep convolutional neural networks*. Commun. ACM 60 (2017), 84–90.
- [12] LECUN, Y., BOSER, B., DENKER, J. S., HENDERSON, D., HOWARD, R. E., HUBBARD, W., JACKEL, L. D.: *Backpropagation applied to handwritten zip code recognition*. Neural Comput. 1 (1989), 541–551.
- [13] LEFKOWITZ, M.: *Professor’s perceptron paved the way for AI – 60 years too soon* [online]. <https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon>
- [14] LYNN, C. N.: *A representation for the adaptive generation of simple sequential programs*. In: J. J. Grefenstette (ed.): *Proceedings of the 1st International Conference on Genetic Algorithms*, L. Erlbaum Associates, Inc., 1985, 183–187.
- [15] MCCULLOCH, W. S., PITTS, W.: *A logical calculus of the ideas immanent in nervous activity*. Bull. Math. Biophys. 5 (1943), 115–133.
- [16] MINSKY, M., PAPERT, S.: *Perceptrons*. MIT Press, 1969.
- [17] PARK, J., SANDBERG, I. W.: *Universal approximation using radial-basis-function networks*. Neural Comput. 3 (1991), 246–257.
- [18] ROSENBLATT, F.: *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychol. Rev. 65 (1958), 386–408.
- [19] RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J.: *Learning representations by back-propagating errors*. Nature 323 (1986), 533–536.
- [20] SHI, X., CHEN, Z., WANG, H., YEUNG, D.-Y., WONG, W.-K., WOO, W.-CH.: *Convolutional LSTM network: A machine learning approach for precipitation nowcasting*. In: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (eds.): *Advances in Neural Information Processing Systems*, vol. 28, Curran Associates, Inc., 2015.
- [21] STANLEY, K. O., D’AMBROSIO, D. B., GAUCI, J.: *A hypercube-based encoding for evolving large-scale neural networks*. Artif. Life 15 (2009), 185–212.
- [22] STANLEY, K. O., MIKKULAINEN, R.: *Evolving neural networks through augmenting topologies*. Evol. Comput. 10 (2002), 99–127.
- [23] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., POLOSUKHIN, I.: *Attention is all you need* [online]. <https://arxiv.org/abs/1706.03762>