UMĚLÁ INTELIGENCE

Jak funguje, jak s ní pracovat a co vypovídá o nás



RADIM KAUFMANN

RADIM KAUFMANN

Umělá inteligence

Jak funguje, jak s ní pracovat a co vypovídá o nás

ČTENÁŘSKÁ UKÁZKA

Dějiny umělé inteligence a cesta od chyby k učení. První dvě kapitoly připravované knihy.

© 2026 Radim Kaufmann. Pracovní čtenářská ukázka, verze 0.3, listopad 2026. Text a koncepce: Radim Kaufmann, s podporou AI při rešerši a redakční přípravě. Ilustrace a schémata: Kaufmann Creative AI Studio©. Fotografie autora převzata z knihy Elias Thorne. Vydání připravuje radimkaufmann.cz. Konečné PDF a EPUB: 150 Kč.

Obsah ukázky

Úvod autora

1. Dlouhá cesta k umělé inteligenci

2. Jak se chyba stane učením: backpropagation krok za krokem

JAK KNIHU ČÍST

Horní index v textu vede k číslované poznámce v rámečku na konci příslušné kapitoly. Poznámka uvádí pramen, místo, které tvrzení podpírá, a hranice jeho platnosti. Odkaz „Zpět do textu“ vrací čtenáře k prvnímu výskytu.

Obrázky uvnitř kapitol vysvětlují konkrétní vztahy a výpočty. Celostránkové předěly jsou autorské vizualizace. Na konci kapitoly následují Slova autora, oddělená od výkladu a pramenů.

V ÚPLNÉ KNIZE

Na úvodní výklad navážou podrobné dějiny od 50. let, architektury a současné trénování, každodenní použití AI, její chyby a ověřování, otázky rozumění a vědomí i společenské důsledky. Závěr doplní důkladný věcný a jmenný index, bibliografie, slovníček a chronologie AI. Rozsah plánujeme přibližně na 500 a více stran podle potřeb tématu.

ÚVOD AUTORA

Proč chci rozumět stroji

Na umělé inteligenci mě nejvíc zajímá okamžik, kdy se samozřejmá věc přestane zdát samozřejmá. Na obrazovce se objeví odpověď. Je souvislá, někdy překvapivě přesná, jindy sebejistě chybná. Čteme ji stejnýma očima, kterýma čteme dopis od člověka. Právě tady pro mě začíná otázka, kvůli které má smysl psát tuto knihu: co se vlastně stalo mezi naším zadáním a těmi několika větami?



Radim Kaufmann. Fotografie z knihy Elias Thorne.
Kaufmann Creative AI Studio©

Nechci se spokojit s obdivem k výsledku. Zajímá mě práce, která je pod ním. Jak se do výpočtu dostala slova. Co se mění, když se model učí. Proč je někdy užitečný a jindy nás zavede na cestu, po které jsme vůbec nechtěli jít. A také proč nám tak snadno připadá, že za odpovědí musí být někdo, koho si umíme představit.

Píšu pro čtenáře, který se chce ptát a nemusí být programátorem. Nechci po něm, aby technice uvěřil jen proto, že je složitá. Budu raději, když si některý výpočet zkusí na papíře, vrátí se k obrázku nebo si otevře pramen pod číslem v textu. Porozumění se podle mě pozná i podle toho, že dokážeme pojmenovat místo, kde zatím končí.

Proto začínáme historií. Dnešní rozhraní může budit dojem náhlého objevu, ale pod ním leží desítky let otázek, omylů a trpělivé práce. Zajímají mě především chvíle, kdy se původní slib potkal s něčím, co výzkumníci neuměli obejít. Takové chvíle bývají poučnější než dodatečně uhlazený příběh úspěchu. Ukazují, co muselo být skutečně vysvětleno a co jen dobře znělo.

U backpropagation se zastavíme déle. Cizí slovo tu pojmenovává postup, který lze rozebrat na jednotlivé kroky. Nejdřív něco spočítáme, potom změříme chybu a nakonec zjišťujeme, jak na ni působí jednotlivá nastavení. Chci, abychom tuto cestu prošli spolu. Když uvidíme, co se děje s několika čísly, budeme mít pevnější půdu pro otázky kolem systémů, jejichž celé vnitřní chování už tak snadno nepřehlédneme.

Stejně vážně beru otázky, které se do rovnice nevejdou. Komu předáváme rozhodování. Co udělá dostupnost hotové odpovědi s ochotou hledat vlastní. Jak rozlišovat mezi tím, že nám nástroj pomáhá, a tím, že už pouze přebíráme jeho způsob uvažování o úkolu. V těchto věcech nemám zájem čtenáři předkládat jediný povinný postoj. Chci zpřesňovat otázky a dát mu dost podkladů, aby si mohl vytvořit svůj.

Poznámky v této knize proto nejsou ozdobou. Hlavní text má držet příběh a zůstat čitelný; oddělené boxy mají ukázat, z čeho vycházíme, co daná práce opravdu zkoumala a kam už její výsledek nesahá. Tam, kde nabízím vlastní úvahu, ji nechci schovávat za vážnost vědeckého odkazu. Na konci kapitol se k vám proto obracím ještě přímo, několika slovy autora.

Jestli má tahle kniha něco změnit, pak doufám, že způsob, jakým se na odpověď stroje podíváme podruhé. První pohled často stačí k překvapení. Druhý už může vést k porozumění. A právě ten bych si chtěl uchovat.

Radim Kaufmann

KAPITOLA 1

Dlouhá cesta k umělé inteligenci



Otázka položená stroji · autorská vizualizace
Kaufmann Creative AI Studio©

KAPITOLA 1

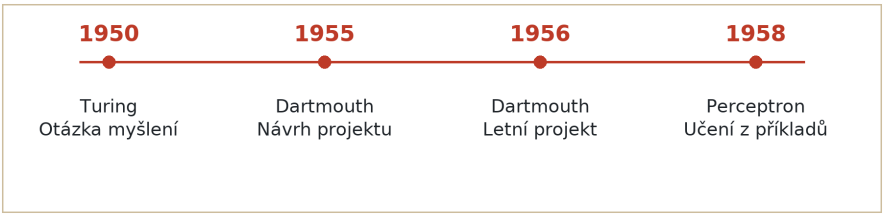
Dlouhá cesta k umělé inteligenci

Otevřeme konverzaci, položíme otázku a za okamžik čteme odpověď. Takto jednoduchý začátek svádí k představě, že podobně jednoduchý musel být i vznik celé technologie. Dějiny AI však ukazují něco jiného: několik souběžných cest, opakované změny představ o inteligenci a dlouhé hledání způsobu, jak přenést úspěch z malé ukázky do skutečného světa. V této kapitole projdeme hlavní rozcestí. Nejde o soutěž o jediného vynálezce. Budeme sledovat, jak se měnila otázka, kterou lidé počítačům pokládali.

1.1 Otázka předchází oboru: 1950-1956

Alan Turing v roce 1950 publikoval v časopise Mind článek Computing Machinery and Intelligence. Nezůstal u sporu o význam slova myšlení: rozpracoval napodobovací hru, v níž posuzovatel hodnotí odpovědi zprostředkované písemnou komunikací. Pozornost přesunul k pozorovatelnému chování stroje. Z takového uspořádání ale neplyne automatická odpověď na otázku, zda stroj něco subjektivně prožívá.[1]

V témže článku Turing uvažoval také o učících se strojích a o možnosti vychovávat zjednodušený výchozí systém. Představa učení tedy v dějinách AI nestojí až na konci, za érou ručního programování. Objevuje se už u jednoho z jejích raných filozofických základů. Turingův návrh zároveň nebyl plánem dnešního jazykového modelu: jeho otázky jsou širší než konkrétní architektura.[1]



Od otázky k výzkumnému programu. Vybrané události; jednotlivé metody mají širší předhistorii. [1] [2] [3] Kaufmann Creative AI Studio©

Dne 31. srpna 1955 se John McCarthy, Marvin Minsky, Nathaniel Rochester a Claude Shannon podepsali pod návrh letního výzkumného projektu v Dartmou-

thu. Plánovali dvouměsíční setkání v létě 1956 a do názvu vložili výraz artificial intelligence. Mezi tématy byly jazyk, abstrakce, neuronové sítě a zlepšování schopností stroje. Jedna zastřešující otázka tak spojovala rozdílné výzkumné programy.[2]

Návrh vycházel z odvážné hypotézy: určité stránky učení a inteligence by mělo být možné popsat natolik přesně, aby je stroj napodobil. Je důležité číst jej jako výzkumnou sázku. Nebyl zprávou, že obecná inteligence už existuje, ani důkazem, že bude snadné ji zkonstruovat. Rozdíl mezi pracovní hypotézou a dosaženým výsledkem nás bude provázet celou knihou.[2]

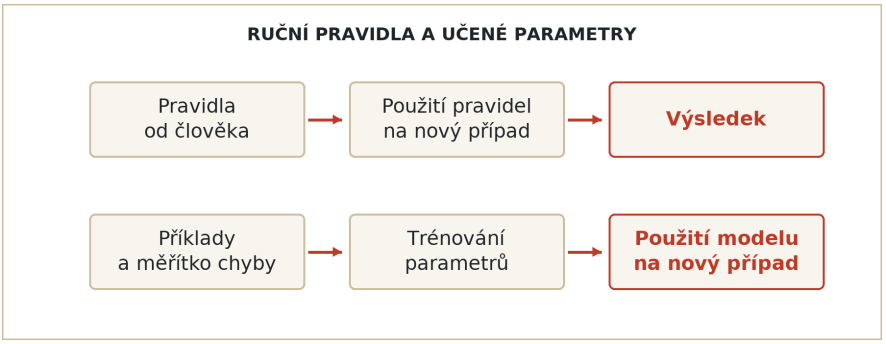
Z dnešního pohledu je užitečné oddělit tři otázky. Co chceme vysvětlit o člověku? Jaký úkol potřebujeme vyřešit? A podle čeho poznáme, že jej systém zvládl? Můžeme postavit výborný nástroj, aniž jsme objasnili lidské vědomí. Stejně tak můžeme předložit zajímavou teorii mysli, aniž máme použitelný výrobek. Jejich směšování vytváří zbytečná nedorozumění.

1.2 Pravidla a zkušenost: dvě cesty k výkonu

Jedna cesta vedla přes symboly a pravidla. Pokud umíme úlohu zapsat pomocí pojmů, vztahů a dovolených operací, mohl by počítač hledat řešení jejich systematickým přetvářením. Allen Newell a Herbert Simon tuto představu výslovně formulovali v článku z roku 1976. Jejich hypotéza fyzikálního symbolového systému byla tvrzením o podmínkách inteligentního jednání, nikoli již uzavřeným výsledkem experimentu.[4]

Představme si jednoduchý plánovací úkol. Zásilka je ve skladu, zákazník je v jiném městě a určité vozidlo má omezenou kapacitu. Symbolický program může pracovat s pojmy zásilka, vozidlo a místo, ověřovat podmínky a sestavovat posloupnost dovolených kroků. Tento příklad je naše didaktická ilustrace: ukazuje, jak lze problém převést na reprezentaci, se kterou se dá počítat.

Druhá cesta se ptala, zda by se potřebné nastavení nemohlo vytvořit z příkladů. Frank Rosenblatt v roce 1958 popsal perceptron: systém inspirovaný otázkami vnímání a organizace informací, v němž se mění účinnost spojení. Dnešní jednoduchý školní model perceptronu vystihuje základní motiv učeného rozhodování, ale není úplnou kopií celé Rosenblattovy původní koncepce.[3]



Dvě cesty k rozhodnutí. Didaktické rozlišení. Skutečné systémy mohou oba postupy kombinovat. [3] [4] Kaufmann Creative AI Studio©

Rozdíl můžeme přiblížit tříděním dopisů. U pravidlového řešení lidé sepíší, podle čeho se má rozhodovat. U učeného řešení předloží příklady a stanoví způsob vyhodnocení, podle kterého se mění parametry. To neznamená, že člověk ve druhém případě zmizel: stále volí úkol, data i kritérium. Přesouvá se část práce, kterou musí výslovně popsat.

Pro základní perceptron si lze představit vážený součet vstupů a rozhodovací práh. Některé kombinace vstupů pak vedou k jedné odpovědi, jiné k druhé. Jeho možnosti závisejí na konkrétní podobě výpočtu. Poučení je širší než jeden historický model: když metoda nezvládne určitý problém, musíme nejprve zjistit, zda omezení patří právě této konstrukci, nebo celé zamýšlené třídě systémů.[3]

Pravidla a učení proto nebudeme líčit jako dvě armády, z nichž jedna definitivně vyhubila druhou. Jsou to různé způsoby, jak do systému dostat strukturu. Už při čtení dalších dějin se vyplatí sledovat jejich spolupráci: někde rozhoduje naučený odhad, jinde prohledávání možností a jinde předem stanovené omezení. Výsledek často vzniká až jejich spojením.

1.3 Když malá ukázka nestačí

Úspěch na několika dobře vybraných příkladech ještě neukazuje, kolik práce bude stát rozšíření systému. Tuto potíž názorně přiblíží strom možností. Jestliže se na každém kroku nabízí deset pokračování, po dvou krocích máme sto posloupností, po šesti milion. Jde o prostý modelový výpočet 10 na šestou, ne o změřené vlastnosti konkrétního programu.

James Lighthill ve zprávě pro britskou Science Research Council z roku 1973 zdůrazňoval právě kombinatorickou explozi. Kritizoval přenos výsledků z omezených úloh do mnohem širšího prostředí. Jeho hodnocení zachycuje významný do-

bový spor. Není však vhodné z něj udělat nadčasový rozsudek nad každou pozdější metodou: posuzoval výzkum a možnosti své doby.[5]

U strojového překladu je důležitým dokumentem americká zpráva ALPAC z roku 1966. Její doporučení se neomezovala na odmítnutí jednoho technického směru. Podporovala také výzkum výpočetní lingvistiky a pomocných nástrojů pro překladatele. Číst dějiny jen jako sled zákazů by tedy znamenalo přehlédnout, kam se měla práce podle samotných dobových dokumentů přesunout.[6]



Jak rychle roste počet možností. Vlastní ilustrace: deset větví na každém kroku, po šesti krocích 10^6 možností. Nejde o měření konkrétního systému.
Kaufmann Creative AI Studio©

Pro období zklamaných očekávání budeme používat označení zimy AI opatrně: jako zkratku pro rozdílné potíže, nikoli jako jednotný světový kalendář. Zpráva o překladu v USA a britské hodnocení výzkumu jsou dva odlišné dokumenty. Samy nedokládají, že by se výzkum všude zastavil, a už vůbec neurčují jedinou příčinu všech pozdějších problémů.[65]

Další odpovědí bylo zúžit úkol a soustředit odbornou znalost do konkrétní oblasti. John McDermott roku 1982 popsal systém R1, který konfiguroval počítačové sestavy VAX-11/780 podle zákaznických objednávek. Článek uvádí jeho pravidelné používání ve výrobní organizaci firmy Digital Equipment Corporation. Byla to praktická práce s omezeními a kompatibilitou součástí, nikoli obecný rozhovor o libovolném tématu.[7]

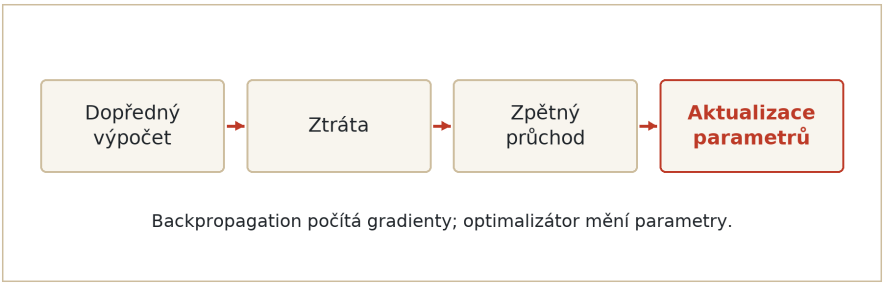
Zde se mění i měřítko úspěchu. Užitečnost programu nemusí spočívat v tom, že připomíná člověka. Může spočívat v tom, že ve vymezené situaci spolehlivě zachytí neslučitelnost, upozorní na chybějící součást nebo pomůže sestavit správné řešení. Pro čtenáře je to důležité vodítko i dnes: ptejme se, za jakých podmínek byl výkon skutečně ověřen.

1.4 Jak se chyba dostala zpátky do sítě

U vícevrstvé sítě vzniká obtížná otázka. Na konci vidíme chybnou odpověď, ale k jejímu vzniku přispělo mnoho mezikroků. Jak rozhodnout, které parametry změnit a jakým směrem? Obecný matematický úkol spočívá ve výpočtu citlivosti výsledku na jednotlivá nastavení. Tady začíná příběh zpětného šíření, označovaného anglickým názvem backpropagation.[8]

Rok 1986 je významným milníkem, nikoli bezpečným datem jediného vynálezu. David Rumelhart, Geoffrey Hinton a Ronald Williams tehdy publikovali v Nature práci Learning representations by back-propagating errors. Představili učení sítí s vnitřními jednotkami prostřednictvím zpětně vypočítávaných informací o chybě. Smyslem nebylo opravovat jen poslední odpověď, ale upravovat také reprezentace uvnitř sítě.[8]

Matematické předchůdce je potřeba odlišit od pozdějšího využití v neuronových sítích. Seppo Linnainmaa publikoval roku 1976 práci o výpočtu vlivu dílčích zaokrouhlovacích chyb na výsledek složeného výpočtu. Paul Werbos ve své disertaci z roku 1974 rozpracoval uspořádané derivace a zpětné výpočty citlivostí. Oba prameny ukazují, proč by bylo nepřesné začínat celou genealogii až článkem z Nature.[9][10]



Dopředu výsledek, zpátky citlivost. Čtyři odlišné části trénovacího kroku. [11]
Kaufmann Creative AI Studio©

Základní myšlenku si můžeme představit jako řetězec závislostí. První mezivýsledek ovlivňuje druhý a ten konečný výsledek. Chceme-li zjistit dopad malé změny na začátku, musíme zohlednit všechny návazné citlivosti. Řetězové pravidlo derivování umožňuje tento vztah počítat; zpětný průchod jej organizuje tak, aby se již získané mezivýsledky mohly opakovaně využít.[9]

Je přitom nutné rozlišovat dvě práce. Backpropagation vypočítává gradient, tedy informace o tom, jak se ztráta mění s parametry. Optimalizační metoda podle těchto informací navrhuje jejich aktualizaci. Výpočet gradientu a samotný krok

učení spolu těsně souvisejí, ale nejsou totožné. Právě toto rozlišení nám později umožní pochopit, co se při trénování skutečně děje.[11]

V názvu zpětné šíření chyby se skrývá užitečná zkratka i možné nedorozumění. Síť nezískává lidský pocit, že něco pokazila. Provádí konkrétní výpočet vzhledem ke zvolené ztrátové funkci. Kvalita tohoto cíle je samostatná otázka. Lze se velmi účinně učit podle nevhodného měřítka; rychlost oprav sama nezaručuje, že systém řeší právě to, co potřebujeme.

1.5 Viditelné triumfy a nenápadná práce

V květnu 1997 porazil systém Deep Blue firmy IBM Garriho Kasparova v šestipartiovém zápase za standardních turnajových podmínek. Šlo o mimořádný výsledek v přesně vymezené hře. Technický popis systému vysvětluje kombinaci specializovaného hardwaru, prohledávání a šachového hodnocení. Převádět tento výkon rovnou na obecné porozumění světu by znamenalo změnit otázku uprostřed hodnocení.[12]

Šachy zde použijme jako pomůcku k přemýšlení o testech. Pravidla jsou známá, tahy lze zaznamenat a konečný výsledek rozpoznáme. Mimo podobně vymezené prostředí bývá už samotné určení dobré odpovědi obtížnější. To nesnižuje šachový úspěch. Učí nás to popsat jeho hranice dříve, než mu přiřadíme širší význam, který nebyl měřen.

Vedle veřejně sledovaných soutěží pokračovalo učení systémů pro konkrétní použití. Yann LeCun a spoluautoři v roce 1998 popsali gradientové metody rozpoznávání dokumentů včetně konvoluční sítě LeNet-5. Jejich studie ukazuje spojení architektury, trénování a praktické úlohy čtení rukopisů. Neuronové sítě proto nepředstavujme jako nápad, který po roce 1986 zmizel a byl znovu nalezen teprve v roce 2012.[13]



Od šachů k obrazu a hře go. Výsledky patří rozdílným úlohám a způsobům hodnocení. [12] [13] [14] [15]
Kaufmann Creative AI Studio©

Roku 2016 následoval výrazný milník v go. Lednový článek týmu Davida Silvera popsal AlphaGo kombinující neuronové sítě s prohledáváním stromu. Síť odhadující vhodné tahy a síť hodnotící pozice pomáhaly řídit hledání. Systém využíval

lidské partie i další učení hraním proti sobě. V tomto konkrétním případě tedy úspěch nevznikl pouhou výměnou prohledávání za učení.[14]

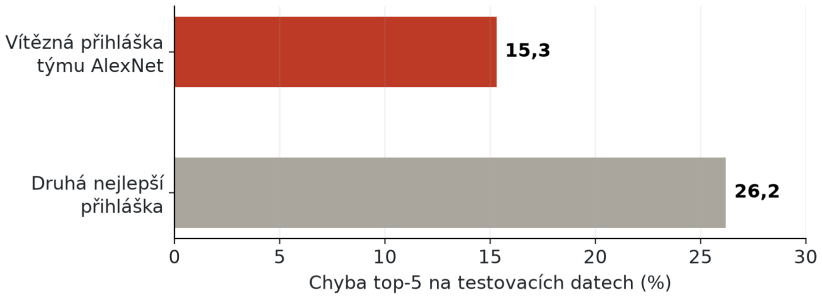
Je třeba oddělit dvě události. Lednový článek referoval o vítězství 5:0 nad evropským šampionem Fan Hui. Zápas s Lee Sedolem se konal až v březnu 2016 v Soulu a skončil 4:1 pro AlphaGo. Jedno datum ani jedna publikace nemohou bez dalšího zastupovat obě. Takové zdánlivé drobnosti rozhodují, zda historické vyprávění drží pohromadě.[14][15]

Z těchto případů si odnáším jednoduché pravidlo čtení dějin: ptejme se vždy, co přesně bylo sestaveno, na čem se to učilo a jak se to zkoušelo. Nálepka inteligence je příliš široká, aby tyto informace nahradila. Šachový systém, čtečka rukopisu a hráč go mohou být všechny pozoruhodné, přesto si jejich schopnosti nemůžeme libovolně přehazovat.

1.6 Rok 2012: když se sešly metoda, data a výpočet

Alex Krizhevsky, Ilya Sutskever a Geoffrey Hinton v roce 2012 publikovali práci o hluboké konvoluční síti pro rozpoznávání obrazů. Model známý jako AlexNet využíval výpočty na grafických procesorech a několik technických postupů, které usnadňovaly trénování. Článek je důležitý i tím, že pojmenovává více součástí výsledku: architekturu, implementaci, data a omezení přeučení.[16]

V soutěži ILSVRC-2012 dosáhla přihlášená varianta soustavy chyby top-5 ve výši 15,3 procenta; druhý nejlepší soutěžní výsledek činil 26,2 procenta. Top-5 znamená, že se správná třída musí objevit mezi pěti nejvýše hodnocenými návrhy. Není to totožné s chybou jediné nejpravděpodobnější odpovědi. U každého grafu výsledků je proto potřebné uvést, co přesně se počítá.[16]



Soutěžní výsledek roku 2012. ILSVRC-2012. Nižší chyba je lepší; srovnáváme soutěžní přihlášky popsané v článku.[16]
Kaufmann Creative AI Studio©

Samotná křivka výkonu přitom nevypráví celý příběh. Pro vlastní hodnocení potřebujeme znát testovanou úlohu, způsob vytvoření dat i to, zda byly různé systémy porovnávány stejně. Když tyto podmínky zmizí z popisku, číslo působí přesvědčivěji, než kolik skutečně sděluje. S podobnými tabulkami se setkáme i u jazykových modelů; navykněme si číst jejich drobné písmo.

Tato srovnání nejsou důvodem přečíst historii zpětně jako předem napsaný scénář. Při každém úspěchu se můžeme ptát, které podmínky mu pomohly a které části vysvětlení zatím jen předpokládáme. Je to náročnější než vybrat jediné převratné datum, ale čtenáři to dává lepší nástroj k hodnocení dalších oznámení. Datum pojmenuje událost; její vysvětlení musíme teprve sestavit.

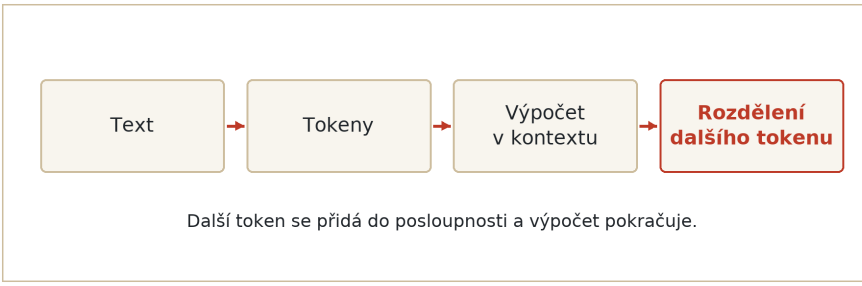
Ani dnešní vývoj nelze zredukovat na heslo, že větší model musí být vždy lepší. Práce o modelu Chinchilla z roku 2022 zkoumala, jak při daném výpočetním rozpočtu rozdělit prostředky mezi počet parametrů a množství tréninkových tokenů. Její výsledky platí pro zkoumané podmínky. Podstatná je samotná otázka hospodárného poměru, nikoli univerzální kouzelné číslo.[17]

1.7 Jazyk jako rozhraní: od Transformeru k základovým modelům

V roce 2017 představili Ashish Vaswani a spoluautoři architekturu Transformer. Jejich původní experimenty se týkaly zejména strojového překladu. Architektura využívala mechanismy pozornosti a umožňovala více paralelizace při trénování než tehdy porovnávané postupy. Nešlo ještě o zveřejnění dnešního konverzačního asistenta, ale o konkrétní návrh, na který mohly další modely navazovat.[18]

Slovo pozornost zde označuje matematický mechanismus vztahů mezi reprezentacemi, nikoli vědomé soustředění. Jako orientační představu si lze vybavit větu, jejíž jednotlivé části potřebujeme vyložit ve vzájemné souvislosti. Přesný výpočet probereme samostatně. Už teď však rozlišujme schopnost mechanismu zpracovávat vztahy od lidského prožitku, který nám jeho název snadno připomene.[18]

Studie GPT-3 z roku 2020 zkoumala velký autoregresivní jazykový model a jeho výkon při zadávání úloh prostřednictvím textu. V popsaném few-shot uspořádání byly ukázky vloženy do kontextu bez změny parametrů gradientovým učením. Tato okolnost je zásadní: následování vzoru v zadání není automaticky totéž jako další trénování modelu.[19]



Slova jako vstup do modelu. Zjednodušené schéma autoregresivního generování. [18] [19]
Kaufmann Creative AI Studio©

Představme si, že do zadání vložíme tři ukázky požadovaného formátu a požádáme o čtvrtou. Může se změnit výsledná odpověď, protože se změnil vstup. Z toho samotného ale nepoznáme, že se upravily váhy sítě. K rozlišení mezi učením, kontextem a případnou pamětí celé aplikace se budeme vracet; v běžném používání se tyto věci často slovně slévají.

Rishi Bommasani a rozsáhlý tým ve zprávě z roku 2021 navrhli označení foundation models, které zde překládáme jako základové modely. Popisují modely trénované na širokých datech, přizpůsobitelné mnoha navazujícím úlohám. Současně upozorňují, že vady společného základu mohou přecházet do odvozených použití. Široká využitelnost a spolehlivost v konkrétní situaci tak zůstávají dvě různé otázky.[20]

Pro čtenáře se tím mění podoba setkání s technologií. Úkol lze často zadat běžnou větou a výsledek také běžnou větou přečíst. V této knize ale budeme trvat na tom, aby příjemné rozhraní nezakrylo potřebu kontroly. Srozumitelný výstup nám pomáhá s nástrojem pracovat; sám o sobě ještě nevysvětluje, jak byl vytvořen ani proč bychom mu měli věřit.

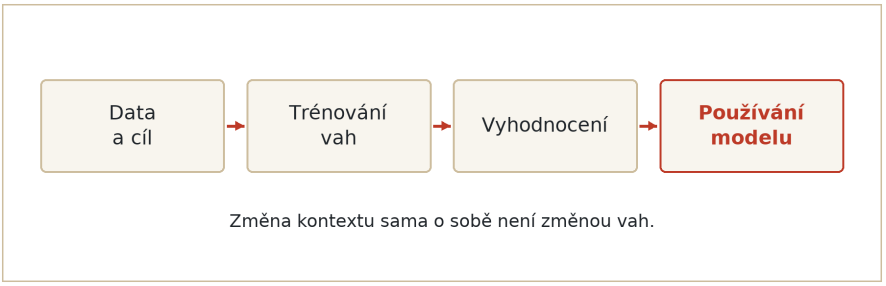
1.8 Současný trénink: více než jeden velký výpočet

Další důležitá otázka zní, jak z modelu předpovídajícího text vytvořit užitečného asistenta. Studie InstructGPT z roku 2022 popsala posloupnost učení na demonstracích, sběru lidských preferencí mezi odpověďmi a dalšího trénování pomocí zpětné vazby. Autoři zároveň uvádějí přetrvávající chyby. Zlepšení podle hodnocení lidí proto není certifikátem neomylnosti.[21]

Je užitečné oddělit to, jak model získává obecné schopnosti, od toho, jak se upravuje jeho chování při zadávání úkolů. Také hodnoticí postupy jsou součástí návrhu: určují, které rozdíly považujeme za zlepšení. Dva systémy se mohou lišit

stylem odpovědi, ochotou přiznat nejistotu nebo úspěšností na konkrétních úlohách. Jedna souhrnná známka tyto rozdíly nemusí zachytit.

Technická zpráva o Llama 3 z roku 2024 dokládá šíři současných vývojových záměrů: vedle jazykového výkonu se zabývá více jazyky, programováním, usuzováním a používáním nástrojů. Jde o doložený příklad jedné modelové rodiny, nikoli univerzální recept pro všechny systémy. Právě tak budeme v knize používat názvy konkrétních produktů: jako datované příklady, jejichž podmínky lze dohledat.[22]



Učení a používání jsou jiné děje. Přehled pro orientaci; konkrétní výrobní postupy se liší. [20] [21] [11]
Kaufmann Creative AI Studio©

Práce DeepSeek-R1, poprvé zveřejněná v lednu 2025 a na arXiv revidovaná v lednu 2026, zkoumá posilované učení pro úlohy s ověřitelnými výsledky, například v matematice a programování. Ukazuje další cestu, jak rozvíjet určité schopnosti modelů pomocí odměny. Ani tento výsledek nedokládá, že by stejně dobře fungoval libovolný cíl nebo že by odpadla potřeba předtrénovaného modelu.[23]

Současnost tedy v této knize nepředstavíme jako hotový konec dějin. Přesný obraz vždy závisí na tom, kterou publikaci, verzi a úlohu posuzujeme. Doložené příklady ukazují různé způsoby spojování architektury, dat, výpočetního rozpočtu a dalšího učení. Jejich společným jmenovatelem není jedna značka. Je jím snaha vytvořit určitý výkon a doložit jej způsobem, který může někdo další přezkoumat.

A právě tady se historická kapitola spojuje s otázkou, kterou si zasloužíme skutečně pochopit. Jak lze převést chybu na změnu parametrů? Co má společného malá síť se dvěma vahami a velký model? Odpověď začíná u derivace, řetězového pravidla a zpětného průchodu. Necháme velká jména na chvíli stranou a projdeme tento výpočet tak pomalu, aby za ním nezůstala magie.

SLOVA AUTORA

Na dějinách umělé inteligence mě přitahuje hlavně to, že narušují pohodlné zkratky. Jakmile si zvykneme na vyprávění o jediném objevu, objeví se předchozí práce. Jakmile máme pocit, že stroj překonal člověka, musíme se zeptat v čem. A když něco působí samozřejmě, stojí za tím obvykle mnoho rozhodnutí, která samozřejmá vůbec nebyla. Nechci vás v této knize přesvědčovat, abyste AI obdivovali nebo se jí báli. Chci, abyste jí lépe rozuměli. Rozumět pro mě znamená dokázat oddělit výsledek od slibu, nástroj od přirovnání a vlastní úsudek od pohodlí hotové odpovědi. Právě proto jsme začali historii. Další krok bude ještě konkrétnější: podíváme se, jak se z vypočtené chyby stane malá změna, kterou může začít velké učení.

Radim Kaufmann

Prameny a souvislosti

1. TURING, Alan M. Computing Machinery and Intelligence. Mind. 1950, 59(236), 433-460.

Lokátor: Oddíly 1-2 (napodobovací hra) a 7 (Learning Machines).

Dokládá: Datování článku, behaviorální uspořádání napodobovací hry a dobové úvahy o učících se strojích.

Meze: Filozofická a metodologická argumentace; nejde o důkaz vědomí ani návrh dnešního LLM.

Otevřít pramen · Zpět do textu

2. McCARTHY, John, Marvin L. MINSKY, Nathaniel ROCHESTER a Claude E. SHANNON. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. 31. srpna 1955.

Lokátor: Datace, úvodní odstavec a seznam výzkumných témat; vydavatelská anotace reedice v AI Magazine 27(4), 2006.

Dokládá: Autoři a datum návrhu, plán na léto 1956 a zastřešení témat označením artificial intelligence.

Meze: Návrh popisuje záměr, nikoli dosažení obecné inteligence. Rok návrhu 1955 a rok setkání 1956 odlišujeme.

Otevřít pramen · Zpět do textu

3. ROSENBLATT, Frank. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review. 1958, 65(6), 386-408.

Lokátor: Úvod a části věnované perceptronovému modelu a změnám spojení.

Dokládá: Učení spojení v perceptronové koncepci a její zaměření na vnímání a organizaci informací.

Meze: Původní koncepce je širší než dnešní zjednodušený jednovrstvý školní perceptron.

Otevřít pramen · Zpět do textu

4. NEWELL, Allen a Herbert A. SIMON. Computer Science as Empirical Inquiry: Symbols and Search. Communications of the ACM. 1976, 19(3), 113-126.

Lokátor: Oddíl Physical Symbol Systems a formulace Physical Symbol System Hypothesis.

Dokládá: Symbolické reprezentace a výslovně hypotetická povaha tvrzení o podmínkách obecného inteligentního jednání.

Meze: Autoři sami připouštějí možnost nepravdivosti hypotézy; nejde o současný konsenzuální důkaz.

Otevřít pramen · Zpět do textu

5. LIGHTHILL, James. Artificial Intelligence: A General Survey. In: Artificial Intelligence: A Paper Symposium. Science Research Council, 1973.

Lokátor: Část Past Disappointments; zejména úvaha o combinatorial explosion, digitální PDF listy 9-11.

Dokládá: Dobová kritika škálování a kombinatorické exploze.

Meze: Hodnocení konkrétního období z pozice autora zprávy; nelze je vydávat za definitivní soud nad budoucími metodami.

Otevřít pramen · Zpět do textu

6. NATIONAL RESEARCH COUNCIL. Language and Machines: Computers in Translation and Linguistics. Washington, DC: National Academy of Sciences - National Research Council, 1966. Publication 1416.

Lokátor: Recommendations, tištěná strana 34 (48. list digitálního PDF).

Dokládá: Doporučení podporovat výpočetní lingvistiku a nástroje pro lidské překladatele.

Meze: Jedna oblast a konkrétní americký institucionální kontext. Dokument sám neprokazuje globální zastaveníAI.

Otevřít pramen · Zpět do textu

7. McDERMOTT, John. R1: A Rule-Based Configurer of Computer Systems. Artificial Intelligence. 1982, 19(1), 39-88.

Lokátor: Abstrakt a vydavatelský bibliografický záznam.

Dokládá: ÚlohaR1, řadaVAX-11/780, pravidlové uspořádání a autorem uváděné provozní používání v DEC.

Meze: Ověřen abstrakt; text neodvozuje další provozní statistiky ani úspory, které v dostupném abstraktu nejsou.

Otevřít pramen · Zpět do textu

8. RUMELHART, David E., Geoffrey E. HINTON a Ronald J. WILLIAMS. Learning Representations by Back-Propagating Errors. Nature. 1986, 323, 533-536.

Lokátor: Abstrakt a popis učicího postupu; strany 533-536.

Dokládá: Vícevrstvé učení a vytváření vnitřních reprezentací prostřednictvím zpětného výpočtu.

Meze: Článek je důležitým milníkem, ne dokladem absence předchozích metod. Digitální autorská kopie je sken.

Otevřít pramen · Zpět do textu

9. LINNAINMAA, Seppo. Taylor Expansion of the Accumulated Rounding Error. BIT. 1976, 16, 146-160.

Lokátor: Abstrakt, oddíly 2-3 a algoritmická část; časopisové strany 146-160.

Dokládá: Výpočet vlivů dílčích chyb ve složeném výpočtu jako matematický předchůdce; využití návazných derivací.

Meze: Článek řeší numerickou analýzu, nikoli demonstrační trénink moderní neuronové sítě. Netvrdíme jediné datum vynálezu.

Otevřít pramen · Zpět do textu

10. WERBOS, Paul John. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Disertační práce. Harvard University, 1974.

Lokátor: Titulní list (srpen 1974); kapitolaII, část(xii) The Ordered Derivative and Dynamic Feedback, digitální PDF listy 108-114.

Dokládá: Dobová formulace uspořádaných derivací a zpětných výpočtů citlivosti.

Meze: Genealogie matematických metod není totéž jako tvrzení o jejich první úspěšné aplikaci na každý typ neuronové sítě.

Otevřít pramen · Zpět do textu

11. KINGMA, Diederik P. a Jimmy BA. Adam: A Method for Stochastic Optimization. ICLR 2015; první preprint 2014. arXiv:1412.6980.

Lokátor: Abstrakt; algoritmus 1 v práci rozlišuje gradienty a následné aktualizace parametrů.

Dokládá: Optimalizátor je postup pracující s gradienty, nikoli synonymum pro jejich výpočet zpětným průchodem.

Meze: Adam je zde příkladem rozlišení rolí. V kapitole nedoporučujeme konkrétní optimalizátor pro libovolnou úlohu.

Otevřít pramen · Zpět do textu

12. CAMPBELL, Murray, A.Joseph HOANE Jr. a Feng-hsiung HSU. Deep Blue. Artificial Intelligence. 2002, 134(1-2), 57-83.

Lokátor: Abstrakt technického článku; výroční archivIBM k zápasu v květnu 1997.

Dokládá: Zápas s Kasparovem v roce 1997, specializovaný hardware, prohledávání a šachové hodnocení.

Meze: Pramen popisuje šachový systém; výsledek nepřenášíme na obecnou inteligenci.

Otevřít pramen · Zpět do textu

13. LECUN, Yann, Léon BOTTOU, Yoshua BENGIO a Patrick HAFFNER. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE. 1998, 86(11), 2278-2324.

Lokátor: Úvod; oddílI.B Gradient-Based Learning; oddílII věnovaný konvolučním sítím a LeNet-5.

Dokládá: Gradientové trénování a konkrétní použití konvolučních sítí v rozpoznávání dokumentů před rokem 2012.

Meze: Doložená případová studie, nikoli úplný přehled všech přístupů devadesátých let.

Otevřít pramen · Zpět do textu

14. SILVER, David, Aja HUANG, Chris J. MADDISON et al. Mastering the Game of Go with Deep Neural Networks and Tree Search. Nature. 2016, 529, 484-489.

Lokátor: Abstrakt, oddíly věnované policy/value networks a Monte Carlo tree search; Obr.1-3.

Dokládá: Spojení neuronových sítí s prohledáváním, lidská data i hraní proti sobě; Fan Hui 5:0.

Meze: Článek z ledna 2016 nedokládá pozdější březnový výsledek proti Lee Sedolovi.

Otevřít pramen · Zpět do textu

15. GOOGLE DEEPMIND. AlphaGo. Oficiální archiv projektu, bez data; část o zápasu s Lee Sedolem v březnu 2016. Přístup 7. listopadu 2026.

Lokátor: Odstavec začínající AlphaGo then competed against legendary Go player Lee Sedol.

Dokládá: Zápas v Soulu v březnu 2016 a skóre 4:1.

Meze: Retrospektivní firemní zdroj; používáme pouze faktický výsledek a datum, nikoli marketingové soudy.

Otevřít pramen · Zpět do textu

16. KRIZHEVSKY, Alex, Ilya SUTSKEVER a Geoffrey E. HINTON. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25. 2012.

Lokátor: Abstrakt; oddíly 3, 4 a 6; Tabulka 2.

Dokládá: GPU implementace, top-5 chyba 15,3% vs 26,2% v ILSVRC-2012.

Meze: Výsledek patří soutěžní přihlášce, nikoli izolované jediné síti. Top-5 se nerovná top-1.

Otevřít pramen · Zpět do textu

17. HOFFMANN, Jordan et al. Training Compute-Optimal Large Language Models. 2022. arXiv:2203.15556.

Lokátor: Abstrakt a formulace optimalizace velikosti modelu a počtu tokenů při daném výpočetním rozpočtu.

Dokládá: Chinchilla jako experiment s poměrem velikosti modelu a množství tréninkových dat.

Meze: Výsledné vztahy závisí na zkoumaném režimu; nejsou univerzálním návodem pro každý rozpočet a účel.

Otevřít pramen · Zpět do textu

18. VASWANI, Ashish et al. Attention Is All You Need. 2017. arXiv:1706.03762.

Lokátor: Abstrakt; první zveřejnění 12. června 2017; architektura v oddílu 3 původní práce.

Dokládá: Transformer, pozornost a strojový překlad jako původní testovaná oblast.

Meze: Výzkum architektury a výkonu; nepodporuje tvrzení o vědomé pozornosti.

Otevřít pramen · Zpět do textu

19. BROWN, Tom B. et al. Language Models Are Few-Shot Learners. 2020. arXiv:2005.14165.

Lokátor: Abstrakt, vymezení few-shot uspořádání bez gradientových aktualizací při zadávání jednotlivých úloh.

Dokládá: Rozdíl mezi ukázkami v kontextu a změnou parametrů u GPT-3 v popisovaném experimentu.

Meze: Tvrzení se vztahuje ke konkrétnímu experimentálnímu uspořádání, nikoli ke všem aplikacím s pamětí nebo doučováním.

Otevřít pramen · Zpět do textu

20. BOMMASANI, Rishi et al. On the Opportunities and Risks of Foundation Models. 2021. arXiv:2108.07258.

Lokátor: Abstrakt a zavedení termínu foundation models; první verze 16. srpna 2021.

Dokládá: Široké předtrénování, přizpůsobení více úlohám a přenos vad společného základu.

Meze: Primární zdroj pro navrženou terminologii a vlastní rámec autorů; nejde o jedinou závaznou definici.

Otevřít pramen · Zpět do textu

21. OUYANG, Long et al. Training Language Models to Follow Instructions with Human Feedback. 2022. arXiv:2203.02155.

Lokátor: Abstrakt; popsaná posloupnost demonstrací, hodnocení preferencí a posilovaného učení.

Dokládá: InstructGPT a rozlišení učení požadovaného chování od samotného zvětšování modelu.

Meze: Lidské preference ani výsledné zlepšení nezaručují pravdivost jednotlivé odpovědi; autoři přetrvávající chyby výslovně uvádějí.

Otevřít pramen · Zpět do textu

22. GRATTAFIORI, Aaron et al. The Llama 3 Herd of Models. 2024. arXiv:2407.21783, verze 3.

Lokátor: Abstrakt; bibliografický záznam a datum verze 3 z 23. listopadu 2024.

Dokládá: Datovaný příklad modelové rodiny se zaměřením na více jazyků, programování, usuzování a používání nástrojů.

Meze: Technická zpráva tvůrců modelu, nikoli nezávislé ověření všech uváděných výsledků; nehodnotíme dnešní pořadí produktů.

Otevřít pramen · Zpět do textu

23. DEEPSEEK-AI et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. 2025. arXiv:2501.12948; verze 2 ze 4. ledna 2026.

Lokátor: Abstrakt a historie verzí (22. ledna 2025; 4. ledna 2026).

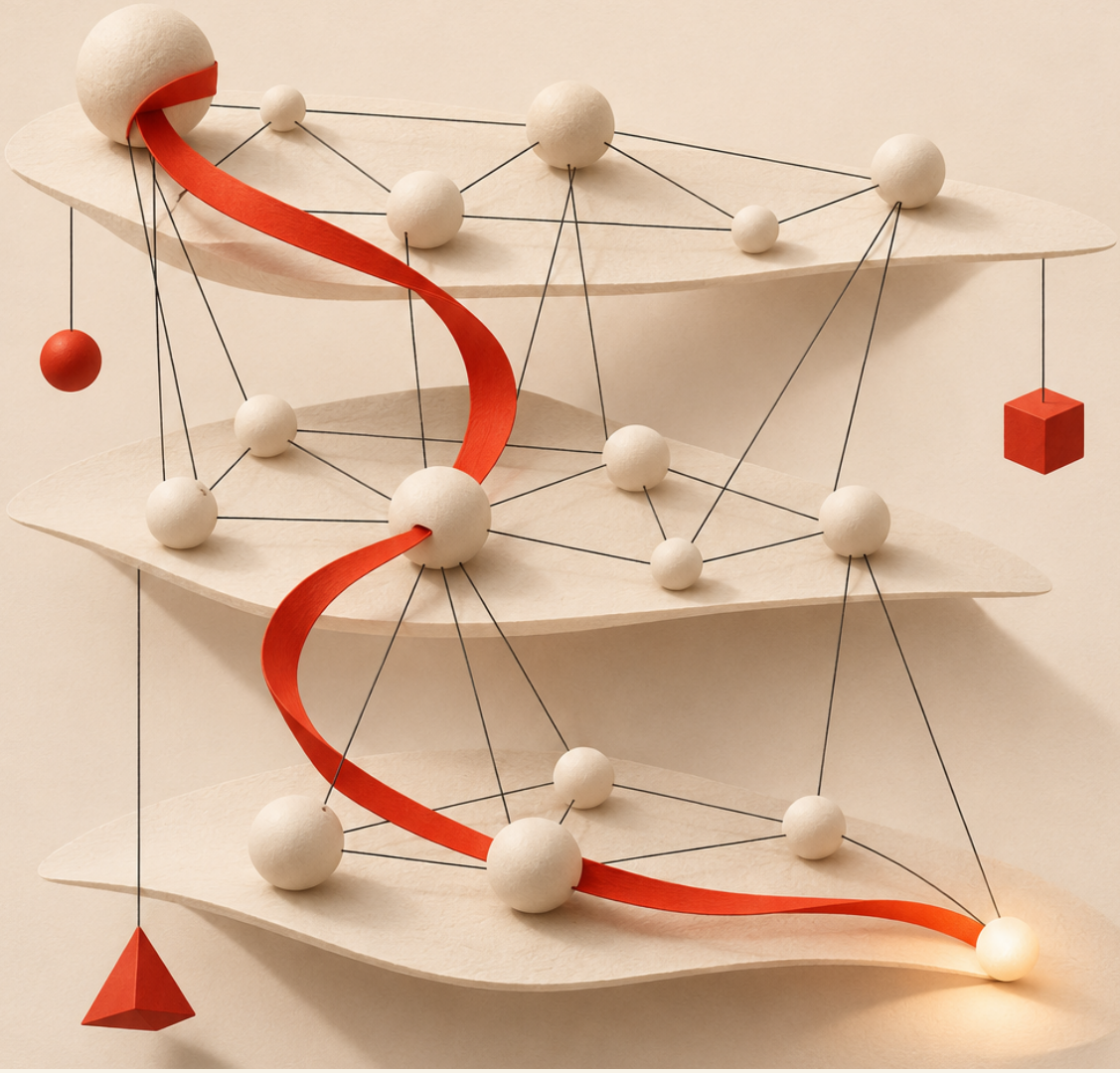
Dokládá: Posilované učení pro ověřitelné úlohy a datování doloženého moderního příkladu.

Meze: Vlastní výsledky vývojářů pro určitá zadání. Absence lidských ukázek postupu v jednom režimu není absence předchozího předtrénování.

Otevřít pramen · Zpět do textu

KAPITOLA 2

Jak se chyba stane učením: backpropagation krok za krokem



Cesta chyby zpátky · autorská vizualizace
Kaufmann Creative AI Studio©

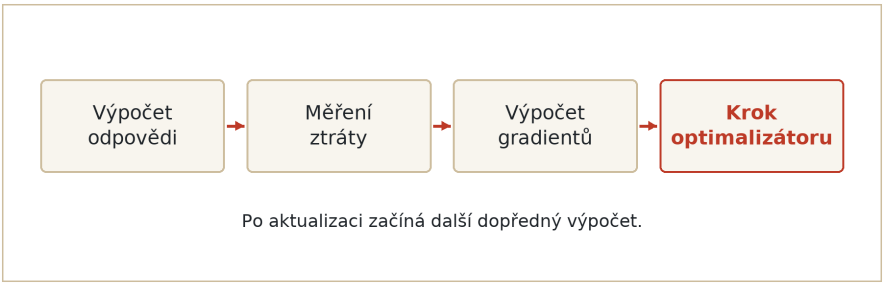
KAPITOLA 2

Jak se chyba stane učením: backpropagation krok za krokem

Síť odpoví špatně. To ještě není učení. Rozhodující je, co se stane potom: podle čeho se mají změnit její parametry, když chybnému výsledku předcházely tisíce, miliony nebo miliardy navzájem propojených výpočtů? Zpětné šíření gradientu, známé jako backpropagation, nabízí na tuto otázku přesnou matematickou odpověď. Abychom ji pochopili, sestavíme vlastní malou síť a provedeme jediný trénovací krok. Každé číslo v příkladu bude možné zkontrolovat tužkou, kalkulačkou nebo několika řádky programu.

2.1 Nestačí vědět, že je odpověď chybná

Představme si jednoduchou předpověď čísla. Správný výsledek je dvě, ale model vypočítá jedna a čtvrt. Rozdíl umíme zjistit okamžitě. Mnohem obtížnější otázka zní: které vnitřní hodnoty máme upravit, o kolik a v jakém směru? Kdybychom měnili každou z nich náhodně a čekali, co se stane, mnoho pokusů bychom promarnili. Potřebujeme informaci o citlivosti výsledku vůči jednotlivým parametrům.



Čtyři kroky jednoho tréninku. Backpropagation je třetí krok; název neoznačuje celý cyklus. [24]
Kaufmann Creative AI Studio©

Právě citlivost vyjadřuje derivace. Říká, jak se sledovaná veličina místně mění při velmi malé změně jiné veličiny. Slovo místně je podstatné: nejde o proroctví pro libovolně velký zásah. Soubor derivací ztráty podle všech parametrů nazýváme

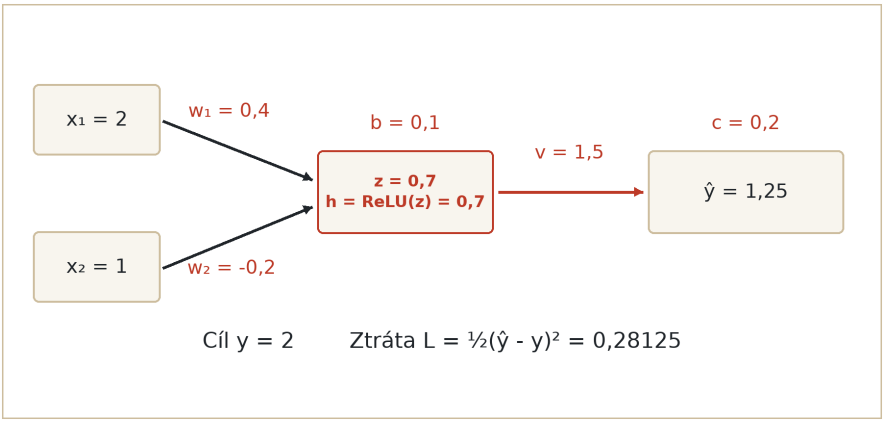
gradient. Backpropagation tento gradient účelně vypočítá pomocí řetězového pravidla. Nevytváří při tom novou definici správné odpovědi.[24]

Je proto užitečné oddělit tři role. Ztrátová funkce stanoví, co budeme v daném úkolu měřit. Backpropagation určí, jak na tuto ztrátu působí drobné změny parametrů. Optimalizátor z gradientů připraví konkrétní aktualizaci. Učení vzniká opakováním celého postupu nad daty, nikoli samotným spuštěním zpětného průchodu. Model přitom může úspěšně zlepšovat zvolené měřítko, a přesto nesplňovat to, co od něj člověk ve skutečnosti potřebuje.

2.2 Malá síť, ve které se nic neschová

Naše síť přijímá dvě čísla: první vstup má hodnotu 2, druhý hodnotu 1. Oba vedou do jednoho skrytého neuronu. Slovo neuron tu označuje matematický výpočet, nikoli biologickou buňku. První vstup násobíme váhou 0,4, druhý váhou -0,2 a k součtu přidáme posun 0,1. Vyjde 0,7. Záporná váha není chyba ani zákaz: při ostatních nezměněných hodnotách zde druhý vstup mezivýsledek snižuje.

Mezivýsledek zpracuje aktivační funkce ReLU. Kladné číslo ponechá, záporné nahradí nulou. Z hodnoty 0,7 tedy dostaneme opět 0,7. Tato aktivace je důležitá pro podobu funkce, kterou síť může vyjadřovat. Kdybychom skládali jen lineární a posunuté lineární operace bez nelinearit, jejich skládáním bychom nezískali obecnou nelineární závislost. V našem konkrétním kroku však zůstáváme v kladné, místně lineární větvi ReLU.[24]



Síť, kterou lze spočítat ručně. Dva vstupy, jeden skrytý neuron a pět trénovatelných parametrů. Vlastní výukový příklad.
Kaufmann Creative AI Studio©

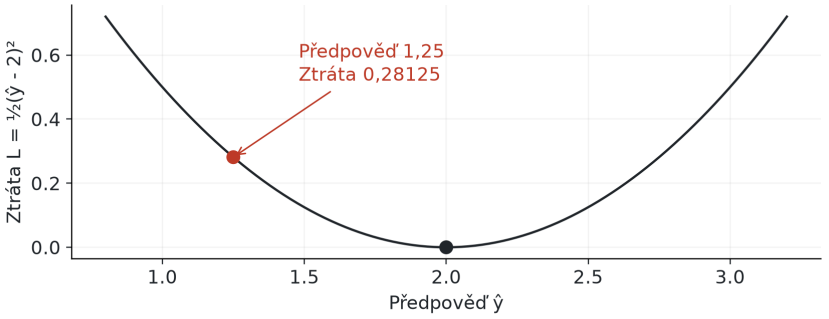
Výstupní část vynásobí aktivaci váhou 1,5 a přičte další posun 0,2. Dostáváme předpověď 1,25. Síť má dohromady pět trénovatelných parametrů: dvě vstupní

váhy, posun skrytého neuronu, výstupní váhu a výstupní posun. Vstupní čísla a správný výsledek jsou pro tuto ukázku zadáním, nikoli hodnotami, které optimalizátor mění.

Malost příkladu není nedostatek. U velkého modelu by se význam jednotlivých kroků schoval za množství operací. Tady vidíme rozdíl mezi vstupem, parametrem a aktivací. Vstup popisuje právě zpracovávaný případ. Parametry určují způsob výpočtu. Aktivace jsou mezivýsledky, které závisí na obojím. Změníme-li vstup, aktivace se mohou změnit i při zcela stejných parametrech. Změníme-li parametry, stejný vstup může vést k jiné předpovědi.

2.3 Chyba potřebuje měřítko

Správný výsledek jsme zvolili jako 2. Rozdíl předpovědi a cíle je proto -0,75. Pro výuku použijeme polovinu druhé mocniny tohoto rozdílu: ztráta vyjde 0,28125. Druhá mocnina odstraňuje znaménko rozdílu; příliš vysoká i příliš nízká předpověď tedy mohou představovat chybu. Faktor jedna polovina pouze zjednodušuje následující derivaci. Nejde o údaj v procentech ani o pravděpodobnost, že má model pravdu.



Chyba má konkrétní měřítko. Kvadratická ztráta pro zvolený cíl y = 2. Tato hodnota není procento ani pravděpodobnost.
Kaufmann Creative AI Studio©

Číselná hodnota ztráty má smysl teprve společně s její definicí a s měřítkem dat. Ztrátu 0,1 u jednoho úkolu nelze bez dalšího prohlásit za lepší než ztrátu 0,2 u jiného. Kdybychom tentýž cíl vyjadřovali v jiné jednotce, změnila by se i velikost kvadratické ztráty. Podobně může vyhodnocování různých případů s různými vahami měnit, které chyby budou trénink nejvíce ovlivňovat.

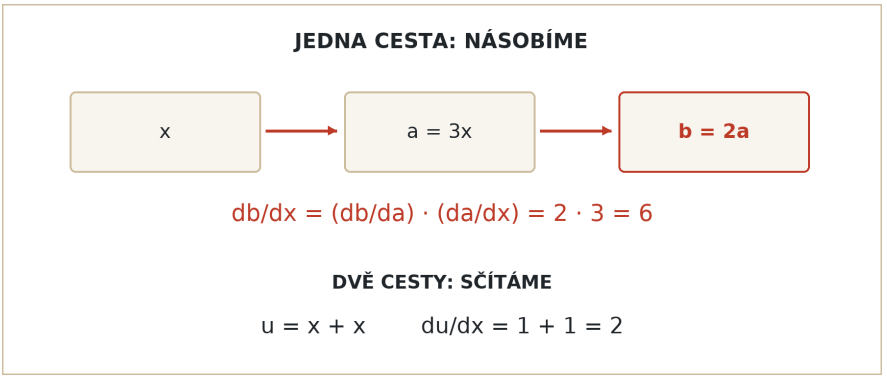
Nejprve tedy rozhodujeme, jaký problém vlastně řešíme. Předpovídáme číslo, vybíráme kategorii, nebo rozdělení pravděpodobností dalších tokenů? Různé cíle

vyžadují odpovídající ztráty. Náš příklad není zmenšeným jazykovým modelem a jeho číselnou ztrátu nelze převést na kvalitu konverzace. Ukazuje obecný princip lokálního výpočtu gradientů na záměrně jednoduchém úkolu. O užitečnosti výsledného systému musíme rozhodovat ještě dalšími měřeními.[25]

2.4 Řetězové pravidlo: sledování závislostí

První váha neovlivňuje ztrátu přímo. Nejprve ovlivní součet ve skrytém neuronu, ten jeho aktivaci, aktivace předpověď a předpověď ztrátu. Chceme-li poznat celkový vliv první váhy, potřebujeme spojit místní citlivosti všech těchto kroků. Na jediné cestě se citlivosti násobí. Pokud se proměnná uplatní v několika větvích, příspěvky příslušných cest se sčítají.

Tuto myšlenku si lze ověřit bez neuronu. Když číslo nejprve vynásobíme třemi a výsledek potom dvěma, drobná změna původního čísla se na konci projeví šestinásobně. U sčítání dvou větví je nutné započítat obě. Taková aritmetika není vysvětlením všech detailů neuronové sítě, ale vystihuje proč při průchodu grafem někdy násobíme a jindy sčítáme.



Citlivosti se skládají po cestách. Vlastní jednoduchý příklad řetězového pravidla a sčítání příspěvků větví. [24] Kaufmann Creative AI Studio©

Zpětný průchod začíná u ztráty a postupuje proti směru závislostí dopředného výpočtu. Označení zpětný neznamená, že běží čas pozpátku nebo že se síť snaží z výsledku rekonstruovat původní vstup. Vypočítáváme derivace. U neuronu, který ovlivnil více dalších výpočtů, se jejich příspěvky musejí správně sejít. Ignorování jedné větve by znamenalo spočítat gradient jiného, neúplného problému.

Úspora spočívá ve sdílení už vypočtených mezivýsledků. Citlivost společné části cesty není nutné znovu odvozovat pro každý parametr zvlášť. Zpětný způsob diferenciace je proto přirozeně vhodný tam, kde máme mnoho parametrů a jednu

skalární ztrátu. Výpočet ale není zdarma: potřebné hodnoty z dopředného průchodu musíme uchovat nebo při zpětném průchodu znovu získat.[26]

Matematický box: od závislosti ke gradientu

Pro jednu cestu mezi parametrem w a ztrátou L používáme řetězové pravidlo. U větvení se příspěvky sčítají. Všechny derivace se vyhodnocují ve stejném aktuálním bodě. Záporný gradient je směr nejrychlejšího místního poklesu vzhledem k běžné eukleidovské geometrii; tvrzení samo neurčuje bezpečnou délku kroku.

$z = w_1x_1 + w_2x_2 + b;\ h = \max(0, z);\ \hat{y} = vh + c$

$L = \tfrac{1}{2}(\hat{y} - y)^2$

$\partial L/\partial w_1 = (\partial L/\partial \hat{y})(\partial \hat{y}/\partial h)(\partial h/\partial z)(\partial z/\partial w_1)$

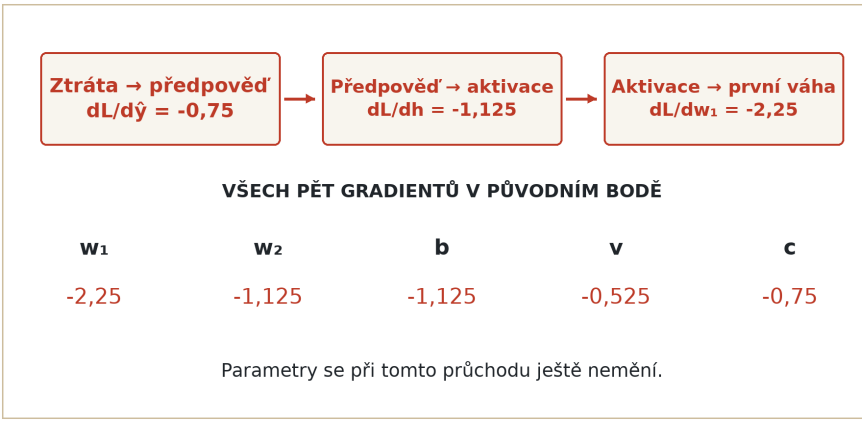
$\partial L/\partial w_1 = (\hat{y} - y) \cdot v \cdot \mathrm{ReLU}'(z) \cdot x_1$

24

2.5 Vracíme se od ztráty k vahám

U naší ztráty je derivace podle předpovědi rovna rozdílu předpovědi a cíle: -0,75. Předpověď je příliš malá. Její nepatrné zvýšení proto ztrátu za současných okolností sníží, což odpovídá zápornému znaménku derivace. To je první údaj zpětného průchodu, nikoli ještě pokyn ke změně všech parametrů stejným způsobem.

Výstupní posun se k předpovědi pouze přičítá. Derivace ztráty podle výstupního posunu je tedy také -0,75. Výstupní váha násobí aktivaci 0,7, takže její derivace vyjde -0,75 krát 0,7, tedy -0,525. Pro pokračování hlouběji do sítě potřebujeme citlivost ztráty vůči aktivaci. Tu získáme násobením -0,75 a dosavadní výstupní váhy 1,5. Výsledek je -1,125.



Gradienty jednoho společného stavu. Všechny hodnoty se vztahují k původním parametrům. ReLU má v použitém kladném bodě derivaci 1.
Kaufmann Creative AI Studio©

Protože vstup do ReLU byl kladný, je její místní derivace jedna. Citlivost se při tomto kroku nezmění. První vstupní váha násobila dvojku, a její gradient je proto -2,25. Druhá váha násobila jedničku, takže dostaneme -1,125. Stejná hodnota náleží posunu skrytého neuronu. Všech pět gradientů jsme získali při jednom uspořádaném návratu přes výpočet.

Všimněme si dvou drobností. Druhá váha byla záporná, ale gradient je rovněž záporný: znaménko váhy samo neurčuje znaménko změny. A když jsme se vraceli přes výstupní váhu, použili jsme její původní hodnotu 1,5. Parametry zatím neměníme. Všechny derivace tohoto kroku musejí odpovídat stejnému stavu sítě, jinak bychom míchali dohromady různé výpočty.

Čtenář si může zkontrolovat alespoň jeden gradient ještě jinak. Zvětšeme pouze první váhu o jednu tisícinu. Skrytý součet vzroste o dvě tisíciny a předpověď o tři tisíciny, protože výstupní váha je stále 1,5. Ztráta se poblíž původního bodu změní přibližně o -0,00225. Dělením velikostí zásahu 0,001 dostáváme očekávanou citlivost -2,25. Přesný konečný rozdíl se nepatrně liší kvůli zakřivení ztráty; derivace popisuje limitu pro zmenšující se zásah.

Výpočetní box: jeden dopředný a jeden zpětný průchod

Autorský didaktický příklad, nikoli měření velkého jazykového modelu. ReLU má v použitém kladném bodě derivaci jedna. Pro kontrolu gradientů lze u každého parametru porovnat ztrátu po jeho nepatrném zvýšení a snížení, zatímco ostatní parametry zůstanou pevné.

$x_1 = 2$; $x_2 = 1$; $y = 2$

$w_1 = 0{,}4$; $w_2 = -0{,}2$; $b = 0{,}1$; $v = 1{,}5$; $c = 0{,}2$

$z = h = 0{,}7$; $\hat{y} = 1{,}25$; $L = 0{,}28125$

$\partial L/\partial w_1 = -2{,}25$; $\partial L/\partial w_2 = -1{,}125$; $\partial L/\partial b = -1{,}125$

$\partial L/\partial v = -0{,}525$; $\partial L/\partial c = -0{,}75$

2.6 Teprve teď se síť změní

Zvolíme nejjednodušší aktualizaci: od každého parametru odečteme jeho gradient násobený rychlostí učení. Tuto rychlost v příkladu nastavíme na 0,05. Záporný gradient po odečtení znamená zvýšení parametru. První váha se proto posune z 0,4 na 0,5125, druhá z -0,2 na -0,14375. Skrytý posun bude 0,15625, výstupní váha 1,52625 a výstupní posun 0,2375.

Uděláme nový dopředný průchod se stejnými vstupy. Skrytý mezivýsledek nyní vyjde 1,0375. Výstup je 1,820984375 a ztráta přibližně 0,0160233. Je nižší než původních 0,28125. Tento pokles jsme nepředpokládali pouze podle obrázku šipky: přepočítali jsme celou síť s novými parametry. Přesné hodnoty i kontrolní program patří k podkladům knihy.

PŘED KROKEM		PO KROKU
Předpověď 1,25		Předpověď 1,820984375
Cíl 2	→	Cíl 2
Ztráta 0,28125		Ztráta 0,0160233

Rychlost učení $\eta = 0{,}05$

Nový výsledek ověřen dalším dopředným výpočtem.

Jeden ověřený krok učení. Pokles na témže výukovém příkladu sám o sobě nic nedokazuje o zobecnění. Kaufmann Creative AI Studio©

Uspěli jsme na jednom uměle zvoleném příkladu. Nemáme tím důkaz, že model rozpozná jiné vstupy, že je vhodně navržen nebo že jsme našli nejlepší možné parametry. Ani správný gradient nezaručuje zlepšení při libovolně velkém kroku. Při větším zásahu už nemusí místní přiblížení vystihovat chování ztrátové funkce. To je důvod, proč rychlost učení není nepodstatné technické nastavení.

Složitější optimalizátory mohou pracovat i s historií gradientů. Adam například udržuje odhady jejich prvního a druhého momentu a podle nich upravuje krok pro jednotlivé parametry. Základní rozdělení práce však zůstává: backpropagation dodává gradienty, optimalizátor je využívá. Adam není jiné jméno pro zpětné šíření a žádný univerzálně nejlepší optimalizátor pro všechny úlohy z tohoto příkladu nevyplývá.[27]

Proč jednoduše nezopakujeme stejné zvýšení parametrů ještě desetkrát? Po prvním kroku už stojíme jinde. Změnil se výstup i citlivosti, a proto je musíme znovu spočítat. Při přiblížení k cíli se může změna zpomalit; po jeho překročení se může směr obrátit. Opakované používání původního gradientu by nebylo opakováním řádného trénovacího kroku. Bylo by slepým pokračováním podle informace, která patřila k předchozímu stavu.

Výpočetní box: společná aktualizace všech pěti parametrů

Používáme gradientní sestup s rychlostí učení $\eta = 0{,}05$. Všech pět nových hodnot vychází z původních parametrů a původních gradientů; neaktualizujeme je postupně uprostřed zpětného průchodu. Pokles je potvrzen novým dopředným výpočtem.

$\theta_{nové} = \theta_{původní} - \eta \cdot \nabla\theta L$

$w_1 = 0{,}5125$; $w_2 = -0{,}14375$; $b = 0{,}15625$

$v = 1{,}52625$; $c = 0{,}2375$

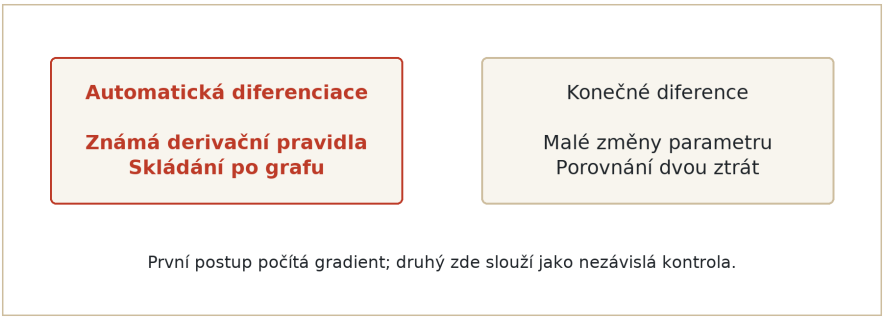
$z_{nové} = h_{nové} = 1{,}0375$; $\hat{y}_{nové} = 1{,}820984375$

$L_{nové} = 0{,}0160232969970703125 < 0{,}28125$

2.7 Jak to zvládne počítač bez ručního odvozování

Pět gradientů jsme spočítali ručně. U velké sítě tuto práci obstarává automatická diferenciace. Program sestaví nebo sleduje výpočet z elementárních operací a využije jejich známá derivační pravidla. V běžném zpětném režimu pak skládá místní derivace v opačném pořadí závislostí. Například dokumentace PyTorch tento postup popisuje pomocí výpočetního grafu a uložených operací.[26]

Automatická diferenciace není totéž co zkoušet každou váhu trochu zvýšit a snížit. Takové zkoušení, nazývané numerická diferenciace pomocí konečných diferencí, se hodí jako kontrola malých výpočtů. Vyžaduje však další vyhodnocování modelu a závisí na velikosti zkušební změny i zaokrouhlování. Naše ověření použilo oba přístupy: analyticky odvozené gradienty a nezávislou numerickou kontrolu pěti parametrů.[28]



Dva způsoby, jak zkontrolovat citlivost. Numerická kontrola závisí na velikosti zásahu a přesnosti výpočtu. 26 28

Kaufmann Creative AI Studio©

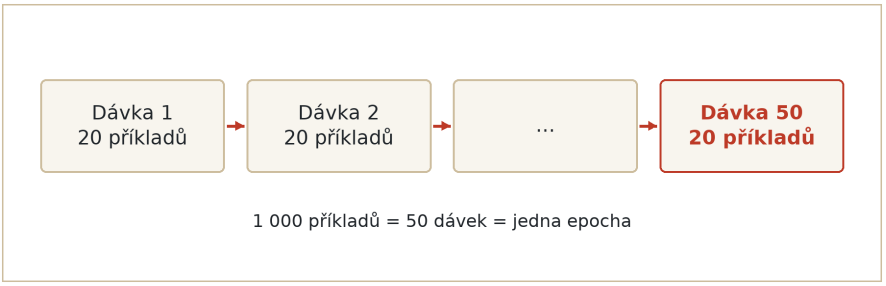
Přesnost matematického postupu nesmíme zaměnit s bezchybností každého programu. Pokud omylem odpojíme část výpočtu, předčasně zaokrouhlíme hodnotu nebo použijeme operaci bez vhodně definovaného gradientu, výsledky mohou být špatné či nepoužitelné. U funkcí s ostrým zlomem, jako je ReLU v nule, navíc pracujeme s výslovně zvoleným pravidlem pro tento bod. V našem příkladu nulu neprotínáme.

U velkého modelu představuje důležitý náklad paměť. Nestačí si představovat jen soubor vah: během tréninku vznikají aktivace, gradienty a často také stav optimalizátoru. Když některé aktivace neuchováme, můžeme je později znovu vypočítat. Tím vyměňujeme část paměťové náročnosti za další početní práci. Větší model tedy není pouze delší seznam čísel uložený na disku; je to i náročnější průběh výpočtu.[29]

2.8 Od jednoho příkladu k celé zkušenosti

Skutečný trénink obvykle nespoléhá na jediný případ. Parametry aktualizuje podle malé dávky příkladů, označované minibatch. Ztráty příkladů lze zprůměrovat a vypočítat gradient tohoto průměru. Jestliže je všech příkladů tisíc a dávka obsahuje dvacet, jeden úplný průchod daty tvoří padesát dávek. Takovému průchodu se říká epocha. Počet dávek, počet aktualizací a počet epoch proto nejsou zaměnitelné údaje.[30]

Jednotlivé příklady mohou požadovat různé směry změny. Dávka jejich vlivy spojí, ale nereprezentuje automaticky dokonale celý svět ani celou trénovací množinu. Záleží na výběru a vážení dat. Kdyby například v sadě převládal jeden snadný typ případu, průměrné zlepšení by mohlo zakrývat potíže u méně častého typu. Výpočet gradientu sám takovou nerovnováhu neoznačí za nespravedlivou nebo nežádoucí.



Dávka, aktualizace a epocha. V tomto zjednodušeném režimu po každé dávce následuje jedna aktualizace. Akumulace gradientů by vztah počtů změnila. [30]
Kaufmann Creative AI Studio©

Trénink tedy potřebuje kontrolu mimo příklady použité k aktualizaci. Validační data pomáhají rozhodovat o nastavení a okamžiku ukončení. Samostatná testovací data slouží k následnému hodnocení; opakované přizpůsobování podle testu by jeho nezávislost oslabovalo. Dělení musí odpovídat úloze. Například téměř totožné kopie na obou stranách by vytvořily příliš snadnou zkoušku.[25]

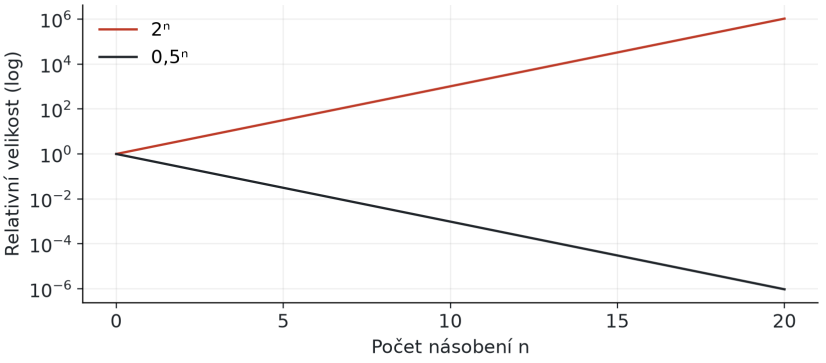
Klesající trénovací ztráta sama o sobě neznamená stále lepší zobecnění. Model může zachycovat i zvláštnosti konkrétní sady. Pravidelné sledování validace, včasné zastavení nebo vhodná regularizace patří k nástrojům, které mají tomuto problému čelit. Žádný z nich nenahrazuje kvalitní data a smysluplné vyhodnocení. Zvládnout příklady, které jsme ukazovali, a obstát v nové situaci jsou dvě odlišné zkoušky.[31]

U naší malé sítě lze tuto omezenost ukázat bez další teorie. Známe jedinou dvojici vstupů a jeden cíl. Nevíme, jaký výsledek by měl náležet například vstupům tři a jedna. Mohli bychom vymyslet mnoho různých pravidel, která souhlasí s původním příkladem, ale na novém se rozcházejí. Úspěšná oprava jedné odpovědi proto sama neurčuje, které pravidlo světa si máme představovat za ní.

2.9 Když gradient zeslábne nebo vybuchne

Zpětný průchod násobí mnoho místních citlivostí. U dlouhých řetězců se mohou výsledné hodnoty stát velmi malými, nebo naopak velmi velkými. Jednoduchý obraz nabízí opakované násobení číslem 0,5 či 2. Po dvaceti násobeních dostaneme přibližně jednu miliontinu, respektive více než milion. U reálných sítí pracujeme s vektory, maticemi a větvením, takže tato ilustrace není úplným popisem jejich dynamiky.

Při mizejících gradientech mohou být některé parametry aktualizovány jen nepatrně. Při explodujících gradientech hrozí nepřiměřené kroky a numerické potíže. Pascanu, Mikolov a Bengio tyto jevy zkoumali v souvislosti s rekurentními sítěmi a popsali omezení normy gradientu jako praktický prostředek proti jeho extrémní velikosti. Gradient clipping nevytváří chybějící signál tam, kde gradient vymizel; řeší jiný problém.[32]



Dlouhá cesta gradientu. Zjednodušená skalární ilustrace. Skutečné sítě mají maticové operace a větvení, nikoli pouze tento řetězec. [32]
Kaufmann Creative AI Studio©

Důležitou roli má také architektura. Reziduální spojení přidává vstup bloku k výstupu jeho transformace. Vzniká tak další cesta výpočtem. He a spoluautoři ukázali, že jejich reziduální sítě umožňují účinně optimalizovat podstatně hlubší modely pro rozpoznávání obrazu. Z tohoto výsledku nelze odvodit, že každé přidání vrstev vždy pomůže nebo že reziduální spojení odstraní všechny problémy tréninku.[33]

Při čtení příběhů o vývoji AI je proto dobré nepřisuzovat úspěch jedinému algoritmu. Samotné pravidlo pro gradienty může zůstat stejné, zatímco se dramaticky zlepší volba aktivací, inicializace, dat, architektury a výpočetních prostředků. V

praxi se musí spojit matematická možnost výpočtu s podmínkami, za nichž se skutečně podaří získat užitečný model.

2.10 Důležitý rok 1986 nebyl počátkem všeho

Článek Davida Rumelharta, Geoffreyho Hintona a Ronalda Williamse z roku 1986 patří ke klíčovým pracím, které ukázaly učení vnitřních reprezentací pomocí zpětného šíření. Jeho význam však není důvodem připsat celou myšlenku jedinému okamžiku nebo jedinému člověku. Samotný článek zmiňuje i nezávisle vzniklé varianty postupu.[34]



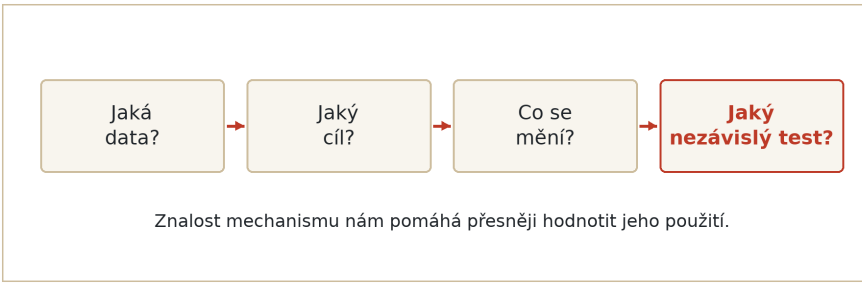
Různé kořeny stejné myšlenky. Tři vybrané prameny; nejde o úplný seznam předchůdců ani o připsání vynálezu jediné osobě. [34] [35] [36]
Kaufmann Creative AI Studio©

Starší kořeny vedou mimo tehdejší hlavní příběh neuronových sítí. Seppo Linnainmaa publikoval v roce 1976 algoritmické zpracování citlivostí výpočetního procesu při analýze zaokrouhlovacích chyb; článek odkazuje i na jeho práci z roku 1970. Paul Werbos v disertaci z roku 1974 rozebíral dynamic feedback jako účelný výpočet derivací použitelný při optimalizaci. Tyto různé kontexty je třeba rozlišovat, nikoli sloučit do legendy o jediném vynálezci.[35][36]

Přesnější otázka tedy není pouze kdo byl první. Zajímá nás, kdo formuloval matematickou metodu, kdo ji spojil s konkrétní třídou modelů, kdo doložil zajímavé výsledky a za jakých podmínek se rozšířila. Takto pojatá historie lépe vysvětluje i dnešek: změna měřítka a dostupnosti může dát starší myšlence nový význam, aniž by se změnil její základní princip.

2.11 Co se nám podařilo vysvětlit

Vraťme se k původní otázce. Síť se nepotřebovala nejprve rozhodnout, že svou chybu lituje nebo že chce být lepší. Měla definovaný výpočet, cíl a postup aktualizace. Pojmenování učení v tomto technickém kontextu popisuje změnu parametrů podle dat a zvoleného kritéria. Samotný výpočet gradientů neřeší otázku vědomí ani nepřiděluje matematickému systému lidskou zkušenost.



Otázky, které zůstávají. Autorské shrnutí otázek pro čtení tvrzení o trénování. Kaufmann Creative AI Studio©

Na druhou stranu by bylo povrchní odbýt celý proces slovy, že jde jen o násobení. Z jednoduchých operací mohou vznikat složité funkce a užitečné schopnosti. O tom, jaké schopnosti skutečně vznikly, však rozhoduje jejich pečlivé zkoumání. Tato kapitola nám dává možnost sledovat mechanismus. V dalších kapitolách budeme zvlášť posuzovat, co naučený systém dovede, co pouze zdánlivě předvádí a jak spolehlivě to poznáme.

Při příštím setkání s větou, že se model naučil odpovídat, už tedy můžeme žádat konkrétnější vysvětlení. Jaká data viděl? Jaký cíl měl snížit? Které parametry se měnily a které zůstaly pevné? Jak se ověřilo, že zlepšení přežilo setkání s novými případy? Nejsou to drobnosti vyhrazené programátorům. Jsou to otázky, které pomáhají rozlišovat popis postupu od přesvědčivé reklamy na jeho výsledek.

SLOVA AUTORA

Na umělé inteligenci mě zajímá chvíle, kdy se velké slovo promění v něco, čemu lze skutečně porozumět. Backpropagation takovou příležitost nabízí. Nemusíme přijmout obraz tajemného stroje, který se učí způsobem člověku navždy skrytým. Můžeme se zastavit u jediného výpočtu a ověřit každý jeho krok. Pro mě tím úžas nekončí. Naopak získává pevnější základ. Současně si chci uchovat odstup: správně spočítaný gradient ještě neznamená správně položenou otázku. O cílech, datech a přijatelných důsledcích rozhodujeme my. Proto považuji porozumění technice za součást odpovědnosti. Chci vědět, co systém optimalizuje, jak poznáme jeho omyl a kdo ponese následky, když jeho výsledek použijeme. Právě s těmito otázkami má podle mě smysl pokračovat dál.

Radim Kaufmann

Prameny a souvislosti

24. Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. Deep Learning. Chapter 6: Deep Feedforward Networks. 2016.

Lokátor: §6.3 Hidden Units; §6.5.1 Computational Graphs; §6.5.2 Chain Rule of Calculus; algoritmy 6.3-6.4

Dokládá: Nelineární aktivace, definice řetězového pravidla a rozlišení dopředného výpočtu a výpočtu gradientu. Číselný příklad kapitoly je vlastní odvození.

Meze: Učebnicová syntéza, nikoli nový experiment nebo důkaz biologické podobnosti. Nevztahuje se na vlastnosti vědomí.

Otevřít pramen · Zpět do textu

25. Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. Deep Learning. Chapter 5: Machine Learning Basics. 2016.

Lokátor: §5.2 Capacity, Overfitting and Underfitting; §5.3 Hyperparameters and Validation Sets

Dokládá: Rozlišení účelu úlohy a měření výkonu; trénovací, validační a testovací data.

Meze: Neurčuje vhodné metriky konkrétní budoucí aplikace; dělení musí respektovat strukturu jejích dat.

Otevřít pramen · Zpět do textu

26. PyTorch contributors. A Gentle Introduction to torch.autograd. 2026.

Lokátor: Oddíl Computational Graph; text o forward pass, backward pass a DAG; přístup7.11.2026

Dokládá: Automatický záznam výpočtů, zpětný průchod a skládání gradientů podle řetězového pravidla.

Meze: Dokumentace konkrétního nástroje průběžně mění verze. Není obecnou zárukou správnosti uživatelského programu.

Otevřít pramen · Zpět do textu

27. Kingma, Diederik P.; Ba, Jimmy Lei. Adam: A Method for Stochastic Optimization. 2015.

Lokátor: ICLR2015; §2 a algoritmus1, PDFstrana2; preprint poprvé2014

Dokládá: Odhady prvního a druhého momentu gradientů a adaptivní aktualizace parametrů.

Meze: Citujeme mechanismus, nikoli všeobecnou nadřazenost či bezpodmínečnou konvergenci původního algoritmu.

Otevřít pramen · Zpět do textu

28. Baydin, Atilim Gunes; Pearlmutter, Barak A.; Radul, Alexey Andreyevich; Siskind, Jeffrey Mark. Automatic Differentiation in Machine Learning: a Survey. 2018.

Lokátor: JMLR18(153), s.1-43; části Numerical Differentiation a Automatic Differentiation

Dokládá: Odlišení automatické diferenciace, numerických konečných diferencí a symbolické diferenciace.

Meze: Přehledová práce, nikoli benchmark současných verzí knihoven. Kontrolní výpočet naší sítě je vlastní.

Otevřít pramen · Zpět do textu

29. Chen, Tianqi; Xu, Bing; Zhang, Chiyuan; Guestrin, Carlos. Training Deep Nets with Sublinear Memory Cost. 2016.

Lokátor: Abstrakt a autorský preprint v2 z22.4.2016

Dokládá: Výměna paměti za výpočetní práci pomocí opětovného výpočtu mezivýsledků tréninku.

Meze: Konkrétní asymptotické a empirické úspory se týkají podmínek článku; v textu nepřebíráme univerzální číselný slib.

Otevřít pramen · Zpět do textu

30. Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. Deep Learning. Chapter 8: Optimization for Training Deep Models. 2016.

Lokátor: §8.1.3 Batch and Minibatch Algorithms; §8.3.1 Stochastic Gradient Descent, algoritmus8.1

Dokládá: Minibatch, odhad gradientu z dávky a krok gradientního sestupu; rychlost učení.

Meze: Používáme obecné definice, nikoli tvrzení o tom, který optimalizátor dnes vede ve všech aplikacích.

Otevřít pramen · Zpět do textu

31. Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. Deep Learning. Chapter 7: Regularization for Deep Learning. 2016.

Lokátor: Úvod kapitoly; §7.8 Early Stopping, algoritmus7.1

Dokládá: Rozdíl trénovacího výkonu a zobecnění, validační sledování a předčasné zastavení jako regularizace.

Meze: Metody snižují některá rizika, neposkytují obecnou záruku správnosti na odlišných budoucích datech.

Otevřít pramen · Zpět do textu

32. Pascanu, Razvan; Mikolov, Tomas; Bengio, Yoshua. On the difficulty of training recurrent neural networks. 2013.

Lokátor: PMLR28(3), s.1310-1318; §2 a §3.2 Scaling down the gradients, PDFstrana6

Dokládá: Mizející a explodující gradienty; omezení normy gradientu jako prostředek proti explodujícím gradientům.

Meze: Analýza se zaměřuje na rekurentní sítě. Dvacetinásobné násobení0,5 a2 je naše zjednodušená ilustrace, nikoli jejich empirický výsledek.

Otevřít pramen · Zpět do textu

33. He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian. Deep Residual Learning for Image Recognition. 2016.

Lokátor: CVPR2016, s.770-778; §3.1 Residual Learning, rovnice1 a2; preprint2015

Dokládá: Reziduální formulace, identitní spojení a empirické usnadnění optimalizace hlubokých sítí v rozpoznávání obrazu.

Meze: Výsledky konkrétních architektur a dat nejsou zárukou pro libovolnou síť nebo úlohu.

Otevřít pramen · Zpět do textu

34. Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J.. Learning representations by back-propagating errors. 1986.

Lokátor: Nature 323, s.533-536; úvod, rovnice výpočtu derivací a poznámka na s.535 o nezávislých variantách

Dokládá: Učení skrytých reprezentací a význam práce z roku 1986; článek sám připomíná nezávislé varianty postupu.

Meze: Studované sítě a úlohy jsou dobové. Článek nedokládá prvenství celé metody automatické diferenciace.

Otevřít pramen · Zpět do textu

35. Linnainmaa, Seppo. Taylor expansion of the accumulated rounding error. 1976.

Lokátor: BIT 16, s.146-160; §4 algoritmus T, §5.1 paměťové nároky, §6 výpočetní graf; bibliografický odkaz 2 na práci1970

Dokládá: Starší algoritmické kořeny zpětného zpracování citlivostí; reprezentace výpočtu grafem a paměťové nároky.

Meze: Článek řeší zaokrouhlovací chyby. Není dokladem dobového tréninku dnešních neuronových architektur; práci1970 zmiňujeme jako doložený bibliografický předchůdce.

Otevřít pramen · Zpět do textu

36. Werbos, Paul John. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. 1974.

Lokátor: Disertace, Harvard University, srpen1974; kap.II, zejména §iii Nonlinear Regression and Dynamic Feedback a §xii The Ordered Derivative and Dynamic Feedback; úvod kap.I, s.I-7

Dokládá: Dynamic feedback jako účelný výpočet derivací pro optimalizaci před rokem1986.

Meze: Primární text je archivní sken s nedokonalým OCR. Záměrně z něj neodvozujeme jednoduché výlučné prvenství celého oboru.

Otevřít pramen · Zpět do textu

Vydavatelský tým

Umělá inteligence

Vedení a spoluautoři

Autor a vydavatel: Radim Kaufmann

Šéfredaktor: Radim Kaufmann

Spoluautoři: Claire Whitaker, Spencer Reed, Julian Hartwell, Tessa Monroe, Gavin Rhodes, Naomi Fletcher, Wesley Porter

Redakce

Vedoucí redakčního provozu: Erin Caldwell

Akviziční redaktor: Camille Foster

Vedoucí knižní řady: Audrey Nolan

Odpovědný redaktor titulu: Owen Barrett

Vedoucí rubriky: Marissa Cole

Koncepční redaktor: Nora Callahan

Stylistický redaktor: Lydia Shaw

Jazykový redaktor: Maya Chen

Jazykový korektor: Leah Winters

Odborné zpracování a výuka

Rešeršér: Priya Bennett

Ověřovatel faktů: Isaac Merritt

Překladatel: Daniel Kim

Zpracovatel rejstříku: Cora Delaney

Odborný recenzent: Elena Sandoval

Technický redaktor: Andre Wallace

Vydavatelský tým

Umělá inteligence

Obrazová podoba

Výtvarný ředitel: Vera Hollander

Knižní grafik: Jasper Vaughn

Grafik knižních obálek: Roman Blake

Obrazový redaktor: Celeste Rowan

Ilustrátor: Hazel Corbin

Sazba a výroba

Sazeč: Fiona Ashby

Vedoucí výroby: Meredith Sutton

Specialista na elektronické knihy: Devin Cho

Práva, propagace a distribuce

Manažer práv a licencí: Tara Mendoza

Kaufmann Creative AI Studio©

Kreativní ředitel studia: Nolan Pierce

Výtvarnice návrhů obálek a scén: Ivy Carver

Ilustrátorka výukových materiálů: Lucia Vega

Specialista na retuš a obrazovou postprodukci: Felix Archer

Původní ilustrace a výtvarná koncepce © Kaufmann Creative AI Studio. Autorské údaje a licence převzatých fotografií, reprodukcí a dalších děl zůstávají uvedeny u příslušných děl a v obrazových kreditech této knihy.

Tým vydavatelství

Doplněná tiráž · 17. listopadu 2026