DOI: 10.35191/medsoft_2026_1_38_kofranek_I

10 PRINCIPŮ MODERNÍ AI

Jiří Kofránek

Abstrakt

Moderní umělá inteligence (AI) se v posledních letech dramaticky proměnila. Od rigidních systémů založených na pravidlech, jakými byly lékařské expertní systémy sedmdesátých a osmdesátých let, jsme se posunuli k dynamickým, adaptivním a generativním modelům, které pravidla nedostávají od člověka, ale odvozují je z dat; zlom přinesly grafické procesory, které umožnily neuronové sítě skutečně natrénovat. Autor, který s AI začínal v osmdesátých letech na pravidlových expertních systémech, chce uvést na pravou míru fámy, které kolem AI kolují. Článek je určen lékařům, studentům medicíny a všem, kdo používají nástroje AI, aniž by je vyvíjeli. Srozumitelnou formou a na konkrétních příkladech vysvětluje deset základních pilířů, na nichž stojí nástroje současné AI: neuronové sítě, trénování a používání modelu, rozklad textu na tokeny, číselnou reprezentaci významu, mechanismus pozornosti, architekturu transformeru, kontextové okno, zákony škálování, předtrénování a dotrénování a konečně velké jazykové modely. Každý princip vykládáme pomocí analogie z fyziologie nebo z lékařské praxe a ukazujeme, co z něj plyne pro každodenní práci s AI: proč jazykový model občas sebejistě uvádí nepravdy, proč zapomíná začátek dlouhého rozhovoru a proč se jeho výstupy musí ověřovat.

Klíčová slova

umělá inteligence, velké jazykové modely, neuronové sítě, lékařská praxe, výuka medicíny, bezpečné používání AI

1 Úvod

1.1 Sen starý čtyřicet let

V osmdesátých letech minulého století jsme se s kolegy pokoušeli naučit počítač medicínsky uvažovat. Byla to doba expertních systémů: znalost zkušeného lékaře se měla zapsat do počítače jako soubor pravidel typu „jestliže – pak" a počítač měl z pravidel a z údajů o pacientovi odvodit diagnózu nebo doporučit léčbu. V osmdesátých letech to bylo oblíbeným tématem řady článků a konferencí.

Nebyli jsme sami. Nejznámějším lékařským expertním systémem byl MYCIN, vyvinutý v sedmdesátých letech na Stanfordově univerzitě pro volbu antibiotické léčby bakteriálních infekcí [1.]. Při srovnávacím hodnocení si vedl podobně dobře jako odborníci na infekční nemoci – a přesto se v klinické praxi nikdy nepoužíval. Stejný osud potkal většinu podobných systémů, včetně našich pokusů o jejich vytvoření [2.–9.].

Příčiny se ukázaly být společné. Každé pravidlo musel někdo vymyslet, zapsat a udržovat, a lékaři své znalosti často neumějí vyslovit jako pravidla – zkušený klinik pozná, že je pacient „vážně nemocný", dřív, než dokáže říct proč. Pravidla fungovala v úzké oblasti, pro kterou byla napsaná, a mimo ni selhávala. A s každým novým pravidlem rostla síť vzájemných vazeb, kterou už nikdo nedokázal udržet v hlavě. Tento problém dostal v oboru i vlastní jméno: úzké hrdlo získávání znalostí. Čistě teoreticky jsme snili o „expertních systémech bez experta" [10.], kdy počítač sám z masivních dat bude nacházet zákonitosti a v datech a automaticky vytvářet pravidla.

Sen o počítači, se kterým lze mluvit o pacientovi přirozenou řečí a který pomáhá s odbornou prací, tak na dlouhá desetiletí usnul. Dnes se splnil – ale úplně jinak, než jsme si tehdy představovali.

1.2 Od pravidel k učení z dat

Dnešní umělá inteligence obrátila směr. Pravidla nikdo nepíše: počítač dostane obrovské množství příkladů a pravidla si z nich odvodí sám. Nástrojem, který to umí, jsou umělé neuronové sítě. Není to žádná novinka – první matematický model neuronu popsali McCulloch a Pitts už v roce 1943 [11.] a postup, jak síť učit na příkladech, byl znám od osmdesátých let [12.]. Chyběla ale výpočetní síla: učení sítě vyžaduje miliardy jednoduchých výpočtů a tehdejší počítače na to nestačily.

Zlom přinesly grafické karty. Byly navrženy pro počítačové hry, aby rychle spočítaly barvu milionů bodů obrazovky, a proto umějí provádět obrovské množství jednoduchých výpočtů současně. Ukázalo se, že přesně takové výpočty potřebuje i učení neuronových sítí. V roce 2012 síť natrénovaná na dvou herních grafických kartách vyhrála soutěž v rozpoznávání obrázků s náskokem, který obor zaskočil: chybovala zhruba ve 15 % případů, zatímco nejlepší jiná metoda ve 26 % [13.]. Od té doby se neuronové sítě prosadily v rozpoznávání obrazu, řeči i textu, a v roce 2022 se s nástupem velkých jazykových modelů dostaly do rukou široké veřejnosti.

Cenou za tuto změnu je, že nikdo neumí přečíst pravidla, která si síť odvodila. Expertního systému MYCIN se například dalo zeptat, proč doporučil právě tento lék na léčení bakteriální infekce, a systém pak vypsal řetěz pravidel. Který ho vedl k danému doporučení. Neuronová síť takové vysvětlení nemá – a může se mýlit způsobem, který žádné pravidlo nepředpovídá.

1.3 Proč rozumět principům

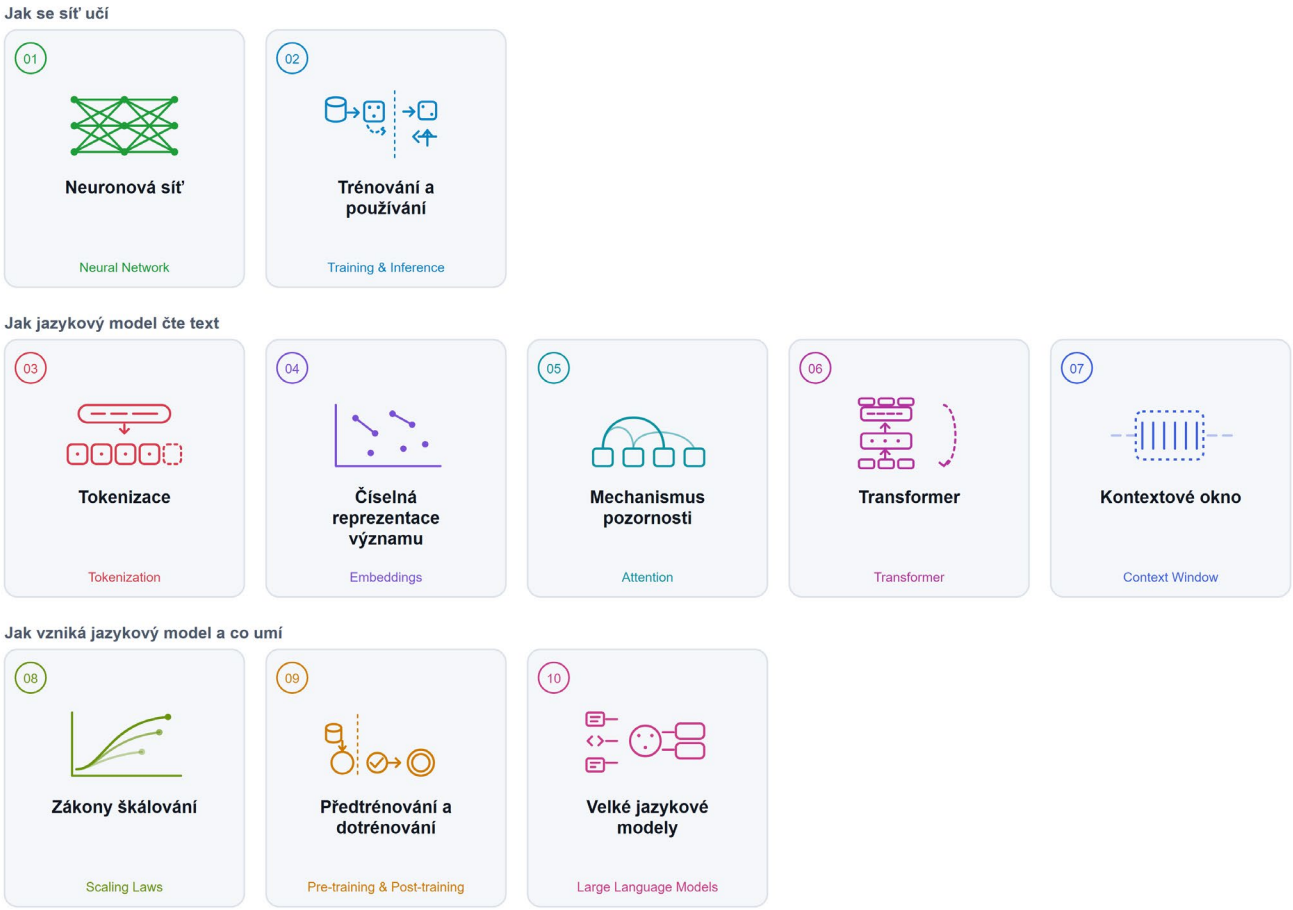
Kolem umělé inteligence dnes koluje řada fám. Podle jedněch AI myslí a rozumí jako člověk, ví všechno a brzy nahradí lékaře. Podle druhých je to jen hloupé doplňování slov, kterému nelze věřit ani slovo. Obě skupiny fám mají společné to, že vznikají z neznalosti toho, jak tyto nástroje fungují.

Lékař nemusí umět sestrojit biochemický analyzátor, aby mohl číst laboratorní výsledky. Měl by ale vědět, co měření ruší – třeba že hemolýza zvýší draslík a že lipemické sérum zkreslí řadu stanovení. Stejně tak nemusí umět naprogramovat jazykový model, ale měl by vědět, kdy mu může věřit a kdy ne. K tomu stačí pochopit deset principů, na kterých stojí současná AI.

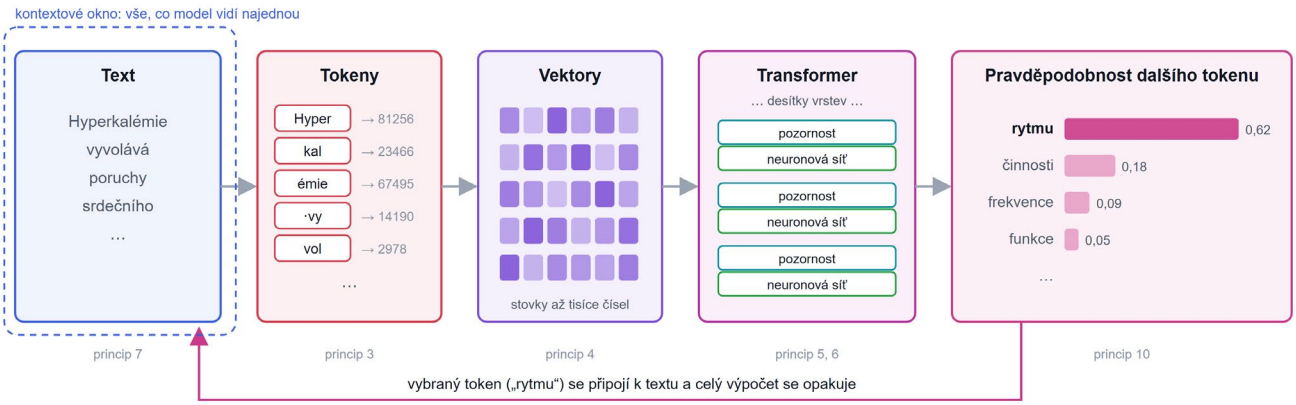
Tento článek původně vznikl jako výklad pro studenty medicíny – pro generaci, která se s umělou inteligencí v nejbližších letech bude setkávat častěji než kterákoli generace před ní. Je určen lékařům, studentům medicíny a všem, kdo používají nástroje AI, aniž by je vyvíjeli. Matematiku vynechávám; každý princip vysvětluji pomocí analogie z fyziologie nebo z lékařské praxe, ukazuji, kde analogie přestává platit, a končím tím, co z principu plyne pro praxi. Čtenáře, které zajímá, co se děje uvnitř, odkazuji na krátké poznámky „pro zvídavé" a na doporučené zdroje na konci článku.

1.4 Deset principů ve třech skupinách

Deset principů tvoří tři skupiny (obr. 1). První dva vysvětlují, jak se neuronová síť učí: co je neuronová síť a jaký je rozdíl mezi jejím trénováním a používáním. Dalších pět popisuje, jak jazykový model čte a zpracovává text: jak se text rozloží na kousky zvané tokeny, jak se jejich význam převede na čísla, jak model pomocí mechanismu pozornosti hledá souvislosti, jak jsou tyto části složené do architektury zvané transformer a kolik textu model najednou „vidí". Poslední tři popisují, jak vzniká jazykový model a co umí: proč větší modely fungují lépe, jak se ze stroje na doplňování textu stane užitečným asistentem a co z toho všeho vychází jako velký jazykový model.



Obrázek 1 – deset principů moderní AI ve třech skupinách: jak se neuronová síť učí (principy 1 a 2), jak jazykový model čte text (principy 3 až 7) a jak jazykový model vzniká a co umí (principy 8 až 10).



Obrázek 2 – cesta textu jazykovým modelem. Text se rozloží na tokeny, tokeny se převedou na vektory, ty projdou desítkami vrstev transformeru a na konci vznikne pravděpodobnost každého možného dalšího tokenu. Vybraný token se připojí k textu a celý výpočet se opakuje. Vše, co model vidí najednou, tvoří kontextové okno. Čísla tokenů jsou skutečná (tokenizátor o200k_base), pravděpodobnosti jsou ilustrační.

DOI: 10.35191/medsoft_2026_1_38_kofranek_I

Těchto deset principů přitom netvoří deset oddělených témat, ale jeden řetěz, kterým projde každý text, který jazykovému modelu napíšeme (obr. 2). Text se rozloží na tokeny, tokeny se převedou na čísla nesoucí význam, ta projdou desítkami vrstev transformeru, a na konci vyjde odhad, které slovo by mělo následovat. To se připojí k textu a celý řetěz se opakuje, dokud nevznikne kompletní odpověď.

2 Jak se neuronová síť učí

2.1 Princip 1: Neuronová síť

Základním stavebním kamenem je **umělý neuron** (obr. 3). Má, podobně jako biologický neuron, několik vstupů („umělých dendritů") a jeden výstup („umělý axon"). Každý vstup umělého neuronu má svou váhu – číslo, které říká, jak silně tento vstup neuron ovlivňuje. Neuron sečte vstupní signály vynásobené jejich vahami a podle výsledného součtu sám vyšle signál dál, k neuronům v další vrstvě. Neurony jsou uspořádány do vrstev: první vrstva přijímá data (například body obrázku), poslední vrstva vydává výsledek (například „na snímku je diabetická retinopatie") a mezi nimi leží skryté vrstvy, ve kterých probíhá vlastní zpracování.

Fyziologovi bude tento popis povědomý, a ne náhodou: umělý neuron je zjednodušeným obrazem biologického neuronu. Ten na svých dendritech přijímá signály z tisíců synapsí. Některé synapse jsou excitační a vyvolají depolarizující postsynaptický potenciál, jiné inhibiční a membránu hyperpolarizují; každá synapse má přitom jinou účinnost. Postsynaptické potenciály se na membráně sčítají v prostoru i v čase, a když součet v oblasti axonového hrbolku překročí práh, neuron vyšle akční potenciál. Váha umělého neuronu odpovídá účinnosti synapse, záporná váha inhibiční synapsi a vážený součet prostorové sumaci. Prahové chování biologického neuronu si čtenář může vyzkoušet na webovém simulátoru Hodgkinova–Huxleyho modelu nervové membrány, který je součástí sady simulátorů popsané v jiném příspěvku tohoto sborníku: podnět 18 mV membránu nevzruší, podnět 19 mV vyvolá plný akční potenciál **(***https://bodylight.physiome.cz/SvelteBodylight.js/medsoft2026/fmu-lab.html?experiment=hodgkin-huxley/prah&lang=cs***).**

Rozhodující vlastností je, že neuron na vstupní součet nereaguje přímo úměrně, ale nelineárně – jako biologický neuron, který pod prahem mlčí a nad ním odpoví. Bez této nelinearity by se síť s libovolným počtem vrstev chovala stejně jako jediná vrstva a nedokázala by se naučit nic složitějšího než přímou úměru. Právě nelinearita umožňuje, aby síť s dostatečným počtem neuronů dokázala napodobit téměř libovolně složitý vztah mezi vstupem a výstupem.

Všechno, co síť „ví", je uloženo ve vahách svých spojení. Síť, která rozpoznává diabetickou retinopatii na fotografii očního pozadí, nikde nemá zapsané pravidlo „mikroaneurysmata a tvrdé exsudáty znamenají retinopatii". Má jen miliony vah, které jsou nastavené tak, že pro snímky s retinopatií vychází na výstupu vysoké číslo a pro zdravé snímky nízké. Takové sítě přitom dosahují senzitivity a specificity srovnatelné s oftalmology [14.] a v roce 2018 byl ve Spojených státech schválen první systém, který diabetickou retinopatii diagnostikuje bez účasti lékaře.

Kde analogie končí. Umělý neuron je velmi hrubé zjednodušení. Nemá akční potenciál, refrakterní fázi ani časový průběh – je to jedno číslo, které se spočítá a předá dál. Biologický mozek obsahuje kolem 86 miliard neuronů se zhruba sto bilionů synapsí, které se během života neustále přestavují. Celý lidský mozek s miliardami těchto neuronů spotřebuje méně energie než slabá žárovka.

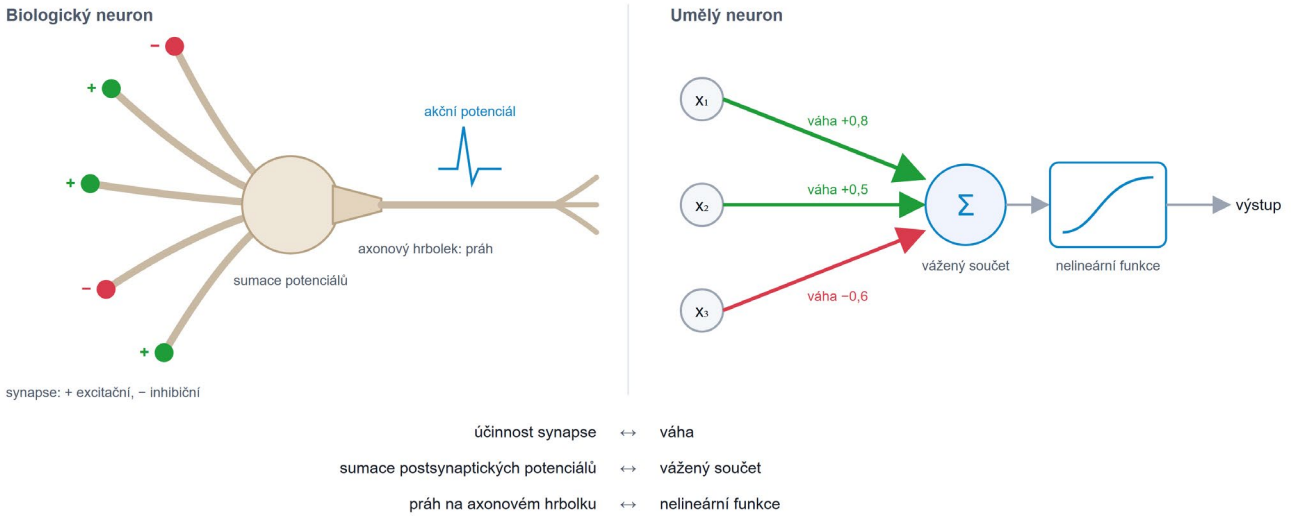
Umělá síť má sice také miliardy vah, ale po skončení učení se už nemění. A hlavně: způsob, jakým se umělá síť učí, nemá v mozku přímý protějšek – o tom je další kapitola.

Pro praxi. Neuronová síť nemá pravidla, která by šla přečíst a zkontrolovat. Její rozhodnutí proto nelze ověřit tím, že se zeptáme na zdůvodnění, ale jen tím, jak dobře funguje na pacientech, které nikdy předtím neviděla. Síť natrénovaná na fotografiích z jednoho typu kamery a jedné populace může na jiné kameře nebo jiné populaci selhat, aniž by to dala jakkoli najevo.

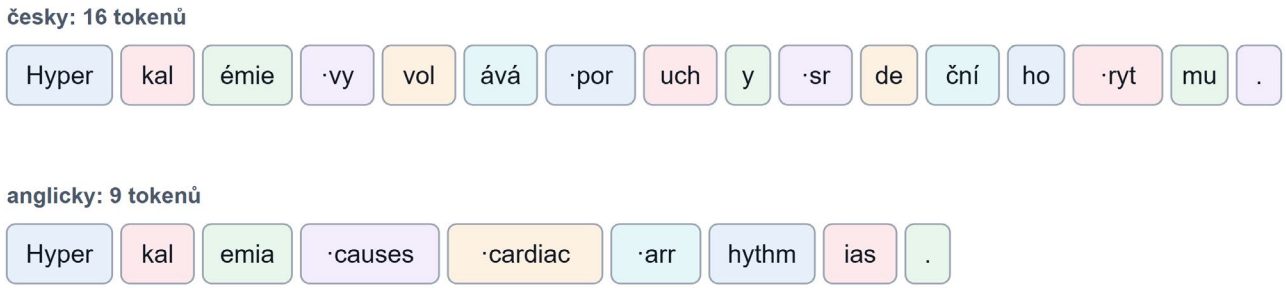
2.2 Princip 2: Trénování a používání modelu

Život neuronové sítě má dvě oddělené fáze. Při **trénování** se síť učí: dostává příklady, u kterých je známá správná odpověď, a své váhy postupně upravuje tak, aby se její odpovědi od správných co nejméně lišily. Při **používání**, které se odborně říká inference, už síť jen odpovídá na nové otázky; její váhy zůstávají takové, jaké byly na konci trénování.

Nejbližší analogií z medicíny je student, který se učí číst elektrokardiogramy. Prochází stovky a tisíce záznamů s popisy a po každé chybě si trochu opraví představu o tom, jak vypadají fibrilace síní, blokáda Tawarova raménka, akutní infarkt a další patologie. Na začátku chybuje často, ale po tisícovce záznamů už pozná na první pohled, o co se jedná, aniž by si musel vybavovat pravidla z učebnice. To je trénování. Když pak jako lékař čte nové EKG u lůžka pacienta, používá to, co se naučil – to je inference.



Obrázek 3 – biologický a umělý neuron. Účinnosti synapse odpovídá váha vstupu, sumaci postsynaptických potenciálů vážený součet a prahu na axonovém hrbolku nelineární funkce. Excitační synapse a kladné váhy jsou zeleně, inhibiční synapse a záporné váhy červeně.



Obrázek 4 – věta „Hyperkalémie vyvolává poruchy srdečního rytmu." a její anglický protějšek rozložené na tokeny tokenizátorem o200k_base, který používají modely GPT-4o. Česká věta má 16 tokenů, anglická 9. Tečka značí mezeru, která patří k tokenu. Hranice tokenů neodpovídají slabikám ani morfémům: „srdečního" = sr | de | ční | ho.

Jak síť pozná, kterým směrem a o jakou hodnotu má váhy upravit? Po každém příkladu se změří, jak moc se její odpověď liší od správné, a pro každou váhu se spočítá, jestli by chybu zmenšilo její zvětšení, nebo zmenšení. Všechny váhy se pak posunou o malý krok ve správném směru. Matematicky se tomu říká gradientní sestup a výpočet, který chybu rozpočítá na jednotlivé váhy od výstupu směrem ke vstupu, se nazývá zpětné šíření chyby [12.]. Lékař zná podobný postup z titrace léku: dávka se upravuje postupně podle změřené odpovědi, a to po malých krocích – příliš velký krok vede k překmitu, příliš malý k tomu, že se cíle nikdy nedosáhne. Totéž platí pro učení sítě, jen se místo jedné dávky léku ladí najednou miliony až miliardy vah a místo jednoho pacienta se postupuje po milionech příkladů.

Obě fáze se liší i nároky. Natrénovat velký jazykový model trvá měsíce na tisících grafických procesorů a stojí podle odhadů desítky až stovky milionů dolarů. Jedna odpověď hotového modelu trvá sekundy. Proto se modely trénují jednou, u několika málo firem s obrovskými výpočetními centry, a pak je používají miliony lidí.

Kde analogie končí. Student se z každého dalšího EKG učí dál, i když už je hotový lékař, a každý pacient jeho zkušenost obohatí. Model se během používání neučí vůbec. Rozhovor s ním jeho váhy nezmění, ať je jakkoli dlouhý a poučný. Když model opravíme a on opravu přijme, platí to jen do konce rozhovoru; v novém rozhovoru udělá tutéž chybu znovu.

Zdánlivě tomu odporuje zkušenost, že si některé chatovací služby „pamatují", co jsme jim řekli včera. Nejde ale o učení. Služba si uloží poznámky z předchozích rozhovorů a při dalším rozhovoru je nenápadně přidá na začátek našeho zadání – model je tedy čte znovu, jako by mu je někdo napsal, a jeho váhy zůstávají stejné. Jak velké množství textu takto model „vidí", popisuje princip kontextového okna v kapitole 3.

Pro praxi. Oprava, kterou model v rozhovoru přijme, platí jen do konce tohoto rozhovoru. Co je důležité, je třeba mu sdělit znovu – nebo si to zapsat do pokynů, které služba ke každému rozhovoru přikládá. A naopak: co modelu v rozhovoru sdělíme, to se nestane součástí jeho znalostí, ale může to zůstat uložené u provozovatele služby. Proto do veřejných služeb nepatří osobní údaje pacientů.

Pro zvídavé. Při trénování musí počítač v paměti držet nejen samotný model, ale i mezivýsledky výpočtu pro každý příklad a údaje potřebné k úpravě vah – proto je trénování tak náročné na paměť i výpočet. Při používání stačí jeden průchod sítí od vstupu k výstupu. Jazykový model ale odpověď píše po jednotlivých slovech a pro každé slovo projde celou sítí znovu, takže i používání je výpočetně náročné.

3 Jak jazykový model čte text

3.1 Princip 3: Tokenizace

Neuronová síť počítá s čísly, ne s písmeny. Než jazykový model začne text zpracovávat, musí ho proto převést na posloupnost čísel. Nedělá to po písmenech ani po celých slovech, ale po kouscích, kterým se říká tokeny. Častá slova tvoří jeden token celá, méně častá slova se rozloží na dva, tři nebo více kousků. Každý token má v slovníku modelu své pořadové číslo a model pak dál pracuje jen s těmito čísly. Slovník velkých jazykových modelů má desítky až stovky tisíc tokenů.

Jak takový rozklad vypadá, ukazuje věta „Hyperkalémie vyvolává poruchy srdečního rytmu." Tokenizátor, který používají modely GPT-4o, ji rozloží na 16 tokenů: Hyper | kal | émie | vy | vol | ává | por | uch | y | sr | de | ční | ho | ryt | mu |. Anglická věta se stejným obsahem, „Hyperkalemia causes cardiac arrhythmias.", vyjde na 9 tokenů: Hyper | kal | emia | causes | cardiac | arr | hythm | ias |. (obr. 4). Hranice tokenů zjevně neodpovídají slabikám ani morfémům – „srdečního" se rozpadne na „sr | de | ční | ho".

Analogií z biologie je genetický kód. Dlouhá posloupnost nukleotidů se nečte po jednotlivých písmenech, ale po trojicích – kodonech – a teprve kodony něco znamenají. Podobně jazykový model nečte text po písmenech, ale po tokenech.

Kde analogie končí. Kodony mají pevnou délku tří nukleotidů a pevný význam daný genetickým kódem. Tokeny mají různou délku a s významem slova nemusejí souviset: vznikly statisticky, z toho, které skupiny znaků se v obrovském množství textů vyskytují nejčastěji. Proto jsou anglická slova, kterých bylo v textech nejvíc, většinou jedním tokenem, kdežto česká slova se ohýbáním mění a rozpadají se na víc kousků.

Z toho plynou dva praktické důsledky. První: čeština je pro jazykový model „dražší" než angličtina. Stejný obsah zabere víc tokenů, a protože se služby účtují a omezují podle počtu tokenů, vejde se české konverzace do stejného limitu méně. Druhý: model nevidí jednotlivá písmena. Úloha, která je pro člověka triviální – spočítat, kolikrát je ve slově „hyperkalémie" písmeno e, nebo napsat slovo pozpátku – je pro model obtížná, protože slovo vidí jako tři kousky „hyper | kal | émie", ne jako dvanáct písmen. Stejně tak čísla: „5 mg" jsou pro model dva tokeny, víceciferná čísla se rozpadají na skupiny číslic a přesné počítání proto není jeho silnou stránkou.

Pro praxi. Úlohy, ve kterých záleží na přesných znacích a číslech – přepočty dávek, opisování hodnot, kontrola kódů diagnóz – jsou slabinou jazykového modelu. Čísla, která model

uvede, je třeba vždy zkontrolovat. Novější modely tuto slabinu obcházejí tím, že si na výpočet samy zavolají kalkulačku nebo napíšou krátký program; i tehdy se ale vyplatí výsledek ověřit.

Pro zvídavé. Slovník tokenů vzniká před trénováním modelu jednoduchým postupem: začne se od jednotlivých znaků a opakovaně se slučuje dvojice, která se v textech vyskytuje nejčastěji, dokud slovník nemá požadovanou velikost [15.]. Tokenizátor ze vzorové věty je volně dostupný a rozklad si lze vyzkoušet na webových stránkách několika poskytovatelů.

3.2 Princip 4: Číselná reprezentace významu

Pořadové číslo tokenu samo o sobě nic neznamená. Token číslo 102 nemá k tokenu číslo 103 žádný vztah, stejně jako nemají žádný vztah dva pacienti se sousedními čísly chorobopisu. Proto se každý token v dalším kroku převede na dlouhou řadu čísel – vektor, který u velkých modelů obsahuje stovky až tisíce hodnot. Tomuto převodu se říká embedding, česky vnoření nebo vektorová reprezentace. Hodnoty vektorů nikdo nenastavuje ručně: síť je nastaví sama při trénování, a to tak, že slova s podobným významem mají podobné vektory. V představě, kde každý vektor je bodem v mnohorozměrném prostoru, leží slova s podobným významem blízko sebe.

Lékař denně pracuje s podobným prostorem, i když mu tak neříká. Pacienta lze popsat řadou laboratorních hodnot – natrémie, kalémie, kreatinin, glykémie, pH, bikarbonát a dalšími – a takový popis je také vektor. Pacienti s podobnou poruchou mají podobné hodnoty a v prostoru laboratorních nálezů tvoří shluky: diabetická ketoacidóza leží jinde než renální selhání a obojí jinde než zdravý člověk. Diferenciální diagnózu pak lze chápat jako otázku, ke kterému shluku má nový pacient nejblíž.

Má to i fyziologický protějšek. Mozek nekóduje informaci tak, že by pro každý pojem měl jeden neuron. Například směr pohybu paže je v motorické kůře zakódován aktivitou celé populace neuronů, z nichž každý je naladěn na trochu jiný směr; teprve vážený součet jejich aktivit určuje výsledný pohyb [16.]. Podobně v embeddingu nenese význam slova jedno číslo, ale vzorec všech hodnot vektoru.

Vektor, na rozdíl od obyčejného čísla (které udává jen velikost, např. *„vážím 80 kg“* nebo *„venku je 20 °C“*), má vždy dvě klíčové vlastnosti: velikost (délku) – „jak moc nebo jak daleko“ – a směr – *„kam to směřuje“*. V dvojrozměrném nebo trojrozměrném prostoru si ho můžeme představit jako šipku s určitou délkou a směrem nebo také jako posloupnost dvou nebo tří čísel (kde ta čísla znamenají koordináty v ploše nebo v prostoru). Prostor může mít ale více dimenzí než 3.

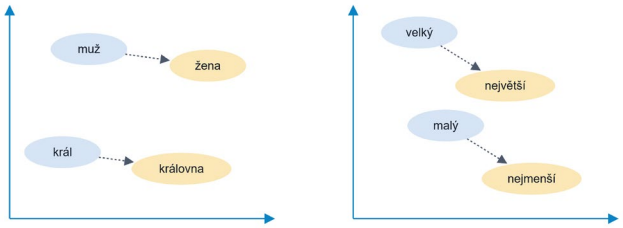
Počítače neumí číst slova jako lidé, ale umí skvěle počítat s čísly. Jedním z největších triků umělé inteligence je to, že se vektory dají použít k pochopení významu slov. AI převádí každé slovo na obrovský vektor (např. seznam 300 čísel), kde každé číslo reprezentuje určitou vlastnost nebo významový kontext. Přesněji řečeno, převádí do vektorové reprezentace ne samotná slova, ale jejich tokeny. Vznikne tak pomyslný „vesmír slov“, kde je každé slovo šipkou – vektorem v 300-rozměrném prostoru (přesněji řečeno, ne slovo, ale každý token).

Když pak natrénujeme AI na ohromné množství textu, pak slova s podobným významem (např. *pes* a *štěně*) mají šipky, které míří téměř stejným směrem, a AI tak snadno pozná, že spolu souvisejí, i když se píší úplně jinak.

Když pak slova reprezentujeme jako vektory, můžeme s nimi provádět skutečnou matematiku:

Např. *Král – Muž + Žena = Královna*

Když od vektoru slova „Král“ odečteme vektor slova „Muž“ a přičteme vektor slova „žena“, dostaneme vektor velmi blízký slovu „Královna“ [17.]. Směr od „Muž“ k „Žena“ tedy v tomto prostoru zachycuje vztah, který mezi oběma slovy je (obr. 5, upraveno podle [18.]).



Obrázek 5 – Slova jako vektory: stejný posun v prostoru významů znamená stejnou změnu významu. Šipka od „muže“ k „ženě“ je rovnoběžná se šipkou od „krále“ ke „královně“ a šipka od „velkého“ k „největšímu“ se šipkou od „malého“ k „nejmenšímu“; proto platí „král“ – „muž“ + „žena“ ≈ „královna“. Ilustrace ve dvou rozměrech – skutečný prostor jich má stovky až tisíce. Upraveno podle Leeho a Trotta [18.].

Kde analogie končí. Osy laboratorního prostoru mají jména: víme, že jedna osa je draslík a druhá kreatinin. Stovky až tisíce os embeddingu žádná jména nemají a nikdo neumí říct, co která znamená. Význam se v nich rozprostírá přes všechny osy najednou.

Příklad. Na embeddingu stojí vyhledávání podle významu, ne podle slov. Dotaz „srdeční selhání“ najde i článek o „insuficienci levé komory“, i když v něm slova „srdeční selhání“ vůbec nejsou, protože oba výrazy mají blízké vektory. Takto pracují i služby, které jazykovému modelu před odpovědí vyhledají vhodné dokumenty – například doporučené postupy – a model pak odpovídá na základě těchto dokumentů (obr. 6).

Pro praxi. Model rozumí parafrázím a synonymům; není třeba trefit přesný odborný termín. Stejná vlastnost má ale i odvrácenou stranu: pojmy, které spolu úzce souvisejí – dva léky téže skupiny, dvě podobné diagnózy – leží blízko sebe a model je může zaměnit.

Pro zvídavé. Podobnost dvou vektorů se obvykle měří úhlem mezi nimi. Vektor, který token dostane na vstupu, je zatím stejný v každé větě; teprve vrstvy pozornosti, o kterých je další kapitola, z něj udělají vektor, který odpovídá významu slova v konkrétní souvislosti.

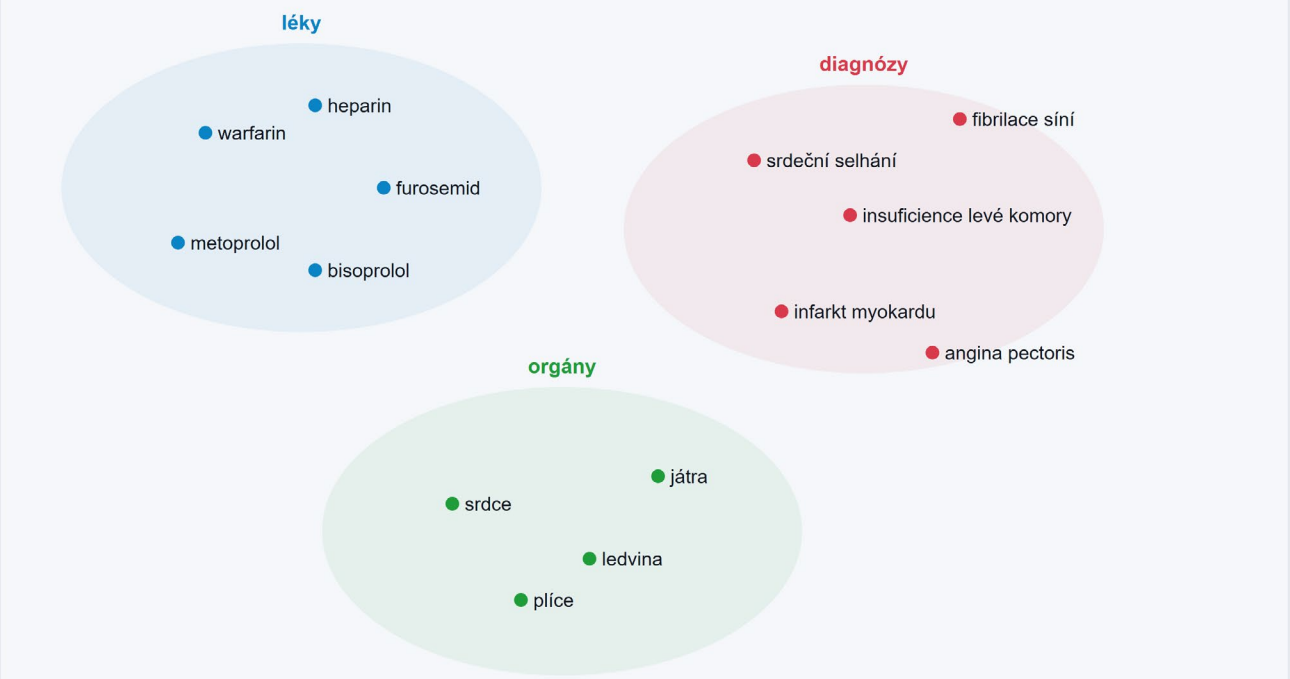
3.3 Princip 5: Mechanismus pozornosti

Význam slova závisí na tom, co stojí kolem něj. Slovo „koruna“ znamená něco jiného ve větě „zaplatil sto korun“, jiného ve větě „koruna stromu“ a ještě jiného v ordinaci zubního lékaře. Starší neuronové sítě četly text slovo po slově a všechno, co už přečetly, musely stlačit do jediné průběžné „paměti“ pevné velikosti. U dlouhých textů pak začátek zapomínaly. Mechanismus pozornosti tento problém vyřešil jinak: při zpracování každého slova se model podívá na všechna ostatní slova textu, ke každému spočítá, jak moc je pro význam zpracovávaného slova důležité, a podle těchto vah z nich poskládá jeho význam v dané souvislosti.

Lékař dělá něco podobného, kdykoli čte zdravotnickou dokumentaci. Tatáž dokumentace obsahuje věk, kouření, rodinnou anamnézu, EKG, nedávný úraz a sportovní aktivity. Je-li důvodem návštěvy bolest na hrudi, padne lékařova pozornost na věk, kouření a EKG; je-li důvodem bolest kolene, na úraz a sport. Stejný text, jiná důležitá místa – podle toho, o co jde. Fyziolog by mluvil o selektivní pozornosti, kterou zná každý z hlučné společnosti: v šumu mnoha hovorů slyšíme ten, který nás zajímá.

Příklad. Ve větách „Pacient užívá warfarin. INR je 5, proto ho vysadíme.“ musí model poznat, k čemu se vztahuje zájmeno „ho“. Gramaticky by mohlo jít o pacienta i o warfarin. Pozornost spojí „ho“ se slovem „warfarin“, protože v souvislosti s INR a vysazováním je právě to slovo pro význam rozhodující (obr. 7).

Kde analogie končí. Lékař se soustředí postupně, na jednu věc po druhé, a ví, proč se soustředí právě na ni. Model počítá



Obrázek 6 – slova jako body v prostoru významů. Slova s podobným významem leží blízko sebe a tvoří shluky – léky, diagnózy, orgány; „srdeční selhání" leží u „insuficience levé komory". Ilustrace: skutečný prostor má stovky až tisíce rozměrů, zde je zjednodušen do roviny a rozmístění slov je jen naznačené.



Obrázek 7 – mechanismus pozornosti: na která slova se při zpracování zájmena „ho" model „dívá". Tloušťka oblouku a číslo pod slovem odpovídají váze pozornosti; největší váhu má slovo „warfarin". Váhy jsou ilustrační; skutečný model jich počítá mnoho najednou, v desítkách „hlav" a vrstev.stovky až tisíce rozměrů, zde je zjednodušen do roviny a rozmístění slov je jen naznačené.

váhy pro všechny dvojice slov v textu najednou a paralelně, a to v mnoha nezávislých „hlavách", z nichž každá sleduje jiný druh vztahu – jedna gramatické vazby, jiná, o kom je řeč, další časovou posloupnost. Nic z toho neprobíhá vědomě.

Pro praxi. Model vnímá souvislosti v celém textu, který dostane. Proto má smysl mu dát celý kontext – věk, anamnézu, medikaci, výsledky – a ne jen holou otázku. Platí to ale i obráceně: nepodstatné podrobnosti mohou model odvést stejně, jako odvedou pozornost člověka.

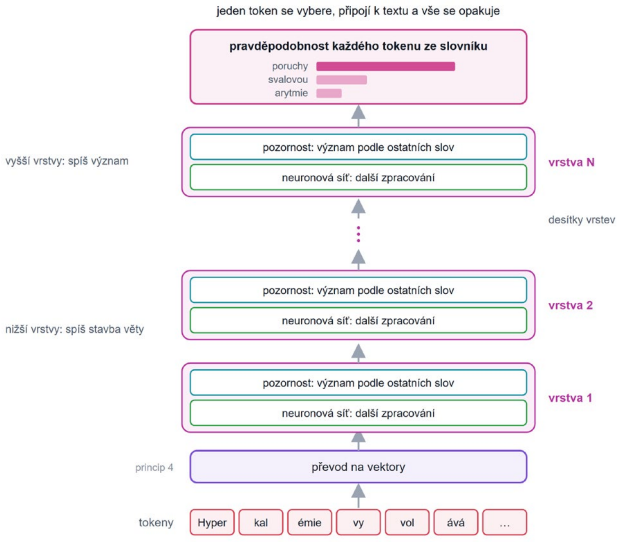
Pro zvídavé. Každé slovo při výpočtu pozornosti klade otázku („co hledám"), každé slovo nabízí štítek („co obsahuji") a obsah („co předám dál"). Čím lépe štítek odpovídá otázce, tím větší váhu dostane obsah. Vzorce lze najít v původní práci [19.] a velmi názorně je vysvětlují videa uvedená v doporučených zdrojích.

3.4 Princip 6: Transformer

Transformer (obr. 8) je stavební plán, podle kterého jsou postavené všechny dnešní velké jazykové modely. Popsala ho v roce 2017 skupina výzkumníků z firmy Google v práci s výmluvným názvem „Attention Is All You Need" – pozornost je vše, co potřebujete [19.]. Transformer se skládá z desítek vrstev naskládaných nad sebe. V každé vrstvě nejprve mechanismus pozornosti upřesní význam každého slova podle ostatních slov textu a pak obyčejná neuronová síť tento význam dál zpracuje. Na vrcholu poslední vrstvy stojí jediná otázka: jaký token bude v textu nejspíš následovat? Výsledkem je pravděpodobnost pro každý token ze slovníku.

Odpověď pak vzniká postupně. Model vybere jeden token, připojí ho k textu a celý výpočet zopakuje, aby určil token následující. Proto odpověď na obrazovce „přibývá" po kouscích. Model přitom nevybírá vždy ten nejpravděpodobnější token; do výběru je záměrně přimíchána trocha náhody, aby text nebyl strojově jednotvárný. Proto tatáž otázka položená dvakrát dostane pokaždé trochu jinou odpověď.

Zpracování textu po vrstvách připomíná zpracování obrazu ve zrakové kůře, jak ho popsali Hubel a Wiesel [20.]: neurony primární zrakové kůry reagují na hrany a čáry určité orientace, vyšší oblasti na tvary a ve spánkovém laloku najdeme oblasti, které reagují na celé tváře. I v transformeru zachycují nižší vrstvy spíš stavbu věty a vyšší vrstvy spíš význam.



Obrázek 8 – transformer: desítky vrstev nad sebou, v každé mechanismus pozornosti a neuronová síť. Nižší vrstvy zachycují spíš stavbu věty, vyšší spíš význam; nahoře vzniká pravděpodobnost každého tokenu ze slovníku. Jeden token se vybere, připojí k textu a vše se opakuje. Při trénování se celý text zpracuje najednou, proto šel transformer trénovat na tisících grafických karet. Pravděpodobnosti jsou ilustrační a místo tokenů jsou zjednodušeně uvedena celá slova.

Kde analogie končí. Zraková dráha má bohaté zpětné vazby, vyšší oblasti ovlivňují nižší a celé zpracování probíhá v čase. Transformer počítá každý token jedním průchodem zdola nahoru. Odpověď nemá napsanou předem: každé slovo volí až ve chvíli, kdy na něj dojde, a jednou napsané slovo už nevezme zpět.

Proč právě transformer znamenal průlom? Starší sítě musely text zpracovávat slovo po slově, protože každý krok potřeboval výsledek kroku předchozího. Transformer při trénování zpracuje celý text najednou – pozornost se pro všechna slova počítá současně. Díky tomu bylo poprvé možné rozdělit trénování mezi tisíce grafických procesorů a naučit model na podstatné části veškerého textu, který je na internetu k dispozici.

Pro praxi. Protože model píše odpověď slovo po slově a nic nevezme zpět, chyba na začátku odpovědi se táhne celou odpovědí. U složitější otázky pomáhá požádat model, aby nejprve postupně rozebral jednotlivé kroky a teprve pak odpověděl. A protože odpověď obsahuje prvek náhody, u důležité otázky se vyplatí zeptat se znovu nebo jinak a odpovědi porovnat: shodné odpovědi nejsou důkazem správnosti, ale rozdílné jsou varováním.

Pro zvídavé. Pozornost sama o sobě nevidí pořadí slov – věta s přeházenými slovy by pro ni vypadala stejně. Proto se do vektorů na vstupu přimíchává i informace o poloze slova v textu. Původní transformer měl dvě části, jednu pro čtení vstupu a druhou pro psaní výstupu; dnešní jazykové modely používají jen tu druhou.

3.5 Princip 7: Kontextové okno

Kontextové okno je všechno, co jazykový model vidí najednou: pokyny, které mu dal provozovatel služby, celá dosavadní konverzace, dokumenty, které jsme mu vložili, i jeho vlastní rozepsaná odpověď. Měří se v tokenech. První modely měly okno o velikosti dvou tisíc tokenů, tedy několika stránek textu; dnešní modely pojmou stovky tisíc tokenů, některé i miliony – celou knihu nebo rozsáhlou zdravotnickou dokumentaci. Co se do okna nevejde, pro model neexistuje.

Nejvýstižnější analogií je rozdíl mezi pracovní a dlouhodobou pamětí. Co se model naučil při trénování, je uložené ve vahách: to je jeho dlouhodobá paměť. Obsah kontextového okna je jeho pracovní paměť. Model se tak chová podobně jako nejslavnější pacient v historii neurověd, Henry Molaison, známý jako pacient H. M. Po oboustranném odstranění mediálních částí spánkových laloků v roce 1953 ztratil schopnost ukládat nové vzpomínky [21.]. Ze starých znalostí čerpal, krátce udržel v pozornosti, o čem se právě mluví, ale jakmile rozhovor skončil, nic z něj si neuložil. Jazykový model je na tom stejně: po skončení rozhovoru z něj nic nezůstane. A podobně jako H. M. neznal události po své operaci, model nezná nic, co se stalo po datu, kdy skončilo jeho trénování – pokud mu to někdo nenapíše do kontextu.

Kde analogie končí. Lidská pracovní paměť je malá; udrží několik položek. Pracovní paměť modelu je obrovská – pojme celou dokumentaci pacienta. Ani tak ale model nevěnuje celému oknu stejnou pozornost. Informace na začátku a na konci dlouhého textu využívá lépe než informace uprostřed [22.].

Příklad. V dlouhém rozhovoru model „zapomene" pokyn, který dostal na začátku – například že má odpovídat stručně nebo že pacient je alergický na penicilin. Pokyn sice v okně stále je, ale ztrácí se v množství textu. A když konverzace přeroste velikost okna, služba nejstarší části zahodí nebo je nahradí stručným shrnutím, a co ve shrnutí chybí, je ztraceno. Jak se s tím dá pracovat, ukazuje zkušenost popsaná v kapitole 5.

Pro praxi. Důležité pokyny a údaje zopakovat, nebo je dát na konec zadání. S novým tématem začít nový rozhovor, se stručným shrnutím toho podstatného. U dlouhého vloženého dokumentu se ptát na konkrétní části. A pamatovat, že model nezná nic po datu svého trénování: aktuální doporučený postup mu je třeba vložit.

Pro zvídavé. Protože pozornost porovnává každý token s každým, roste výpočet s druhou mocninou délky textu: dvakrát delší text znamená čtyřikrát víc výpočtů. To je hlavní důvod, proč byla okna dlouho malá a proč je dlouhý kontext dodnes drahý.

4 Jak vzniká jazykový model a co umí

4.1 Princip 8: Zákony škálování

Jedním z nejpřekvapivějších zjištění posledních let je, jak předvídatelně se jazykové modely zlepšují s velikostí. Chyba modelu – měřená tím, jak dobře odhaduje další token v textu, který nikdy neviděl – klesá s počtem parametrů modelu, s množstvím trénovacího textu a s objemem výpočtu podle jednoduchého mocninného vztahu, a to v rozsahu mnoha řádů [23.]. Vyneseno v logaritmických souřadnicích, je to přímka. Kdo zná rozpočet na výpočet, dokáže tedy předem odhadnout, jak dobrý model z něj vznikne.

Fyziologovi tento typ zákonitosti nebude cizí. Kleiberův zákon říká, že bazální metabolismus savců roste přibližně se třemi čtvrtinami mocniny tělesné hmotnosti, a to od myši až po slona [24.]. Ani tady nejde o přímou úměru, ale o mocninný vztah, který v logaritmických souřadnicích dává přímku přes mnoho řádů (obr. 9).

Zákony škálování ukázaly i to, jak výpočet rozdělit. První práce doporučovaly hlavně zvětšovat model [23.]; pozdější výzkum ukázal, že model a množství trénovacího textu mají růst stejným tempem – dvakrát větší model potřebuje dvakrát víc textu, zhruba dvacet tokenů na každý parametr [25.]. Pro představu o měřítku: model GPT-3 z roku 2020 měl 175 miliard parametrů [26.] a dnešní modely se trénují na bilionech tokenů.

S velikostí se navíc objevují schopnosti, které malé modely nemají vůbec: řešit slovní úlohy, překládat mezi jazyky, na které nebyly cíleně trénované, nebo psát funkční programy [27.]. Zda se tyto schopnosti objevují náhle, nebo postupně a jen naše měřítka je ukazují jako skok, je předmětem sporu [28.]; nesporné je, že se objevují.

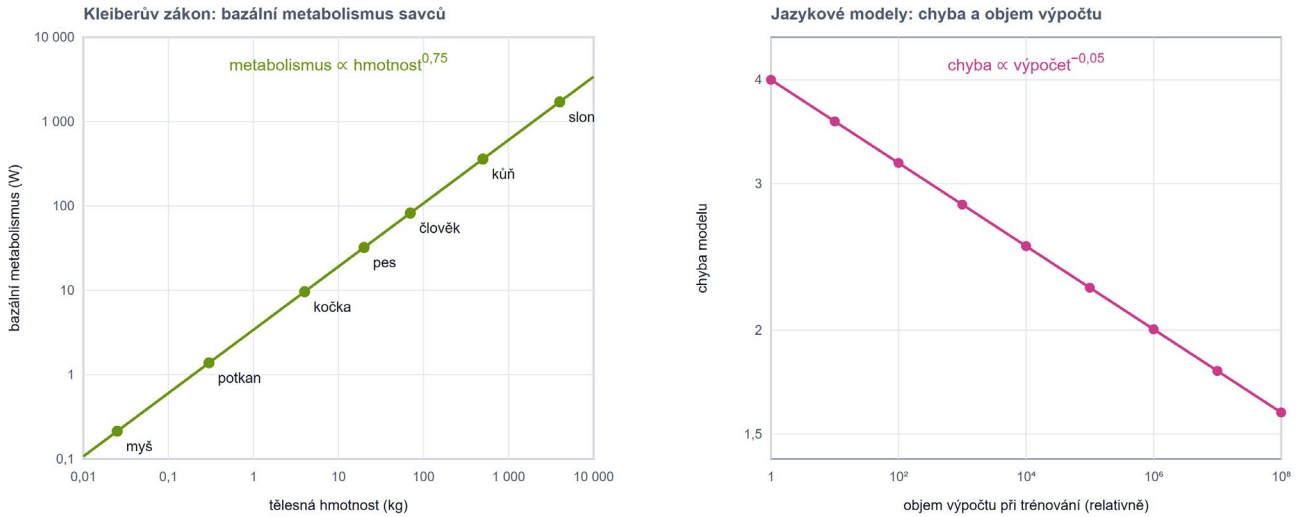
Kde analogie končí. Kleiberův zákon popisuje organismy, které už existují, a vyplývá z fyziky transportu látek v těle. Zákony škálování jsou čistě empirické pozorování o učení a nikdo nezaručuje, že budou platit donekonečna. Už teď narážejí na meze: kvalitních textů je na světě konečně mnoho a výpočetní centra pro trénování spotřebují podle odhadů energii srovnatelnou s menším městem.

Pro praxi. Modely se zlepšují rychle a co nefungovalo loni, může fungovat letos. Zkušenost se starší verzí – v dobrém i ve zlém – proto neplatí napořád. Platí to i pro studie: výsledek srovnání modelu s lékaři z doby před dvěma lety vypovídá o tehdejším modelu, ne o dnešním.

4.2 Princip 9: Předtrénování a dotrénování

Jazykový model vzniká ve dvou fázích. V první, předtrénování, se model na obrovském množství textu – knihách, článcích, webových stránkách, programovém kódu – učí jediné úloze: odhadnout další token. K tomu nepotřebuje žádné lidské popisky; správnou odpověď, tedy skutečně následující slovo, obsahuje sám text. Aby tuto úlohu zvládal dobře, musí se model naučit gramatiku, fakta o světě i to, jak na sebe navazují myšlenky. Výsledkem je takzvaný základní model, který zná jazyk a mnoho faktů, ale neumí vést rozhovor. Na otázku „Jaká je léčba hypertenze?" může odpovědět dalšími otázkami, jako by psal zkušební test, protože v textech často následuje otázka po otázce.

Ve druhé fázi, dotrénování, se ze základního modelu stává asistent. Nejprve se učí na tisících pečlivě připravených ukázko-



Obrázek 9 – Mocninné zákony v logaritmických souřadnicích. Vlevo Kleiberův zákon: bazální metabolismus savců roste se třemi čtvrtinami mocniny tělesné hmotnosti, od myši po slona; zvířata jsou umístěna na přímku zákona, ne vynesena z měření. Vpravo chyba jazykového modelu klesá s objemem výpočtu při trénování, schematicky podle Kaplana a spol. [13.]; každý bod je model trénovaný s jiným rozpočtem výpočtu.

vých rozhovorů, jak má vypadat dobrá odpověď. Potom lidští hodnotitelé porovnávají dvojice odpovědí modelu a určují, která je lepší, a model se učí dávat přednost odpovědím, které lidé hodnotí výš. Účinek je značný: v původní práci dali hodnotitelé přednost odpovědím dotrénovaného modelu s 1,3 miliardy parametrů před odpověďmi stokrát většího modelu, který dotrénovaný nebyl [29.]. V této fázi se model učí i odmítat škodlivé požadavky.

Analogií je cesta k lékařskému povolání. Pregraduální studium dá široké znalosti z anatomie, fyziologie, patologie a klinických oborů – to je předtrénování. Specializační příprava pod vedením školitele naučí tyto znalosti použít u konkrétního pacienta, správně komunikovat a nepřekročit své kompetence – to je dotrénování. Hodnocení od lidí je obdobou zpětné vazby od školitele.

Kde analogie končí. Školitel hodnotí, co je správně. Hodnotitelé jazykového modelu hodnotí, co se jim líbí – a to není vždy totéž. Lidé hodnotí výš odpovědi, které s nimi souhlasí, chválí je a znějí sebejistě. Model se tak může naučit přitakávat i tam, kde by měl oponovat [30.]. Na návodnou otázku „Je pravda, že…?“ má sklon odpovědět „ano“.

Pro praxi. Ptát se neutrálně: místo „Je správné nasadit u tohoto pacienta betablokátor?“ se zeptat „Jaké jsou možnosti léčby a jaká jsou jejich rizika?“. Chtít od modelu i argumenty proti vlastnímu názoru. A nebrat souhlas modelu jako potvrzení – model souhlasí snadno.

4.3 Princip 10: Velké jazykové modely

Velký jazykový model spojuje všech předchozích devět principů. Je to neuronová síť s miliardami parametrů, postavená podle plánu transformeru, která čte text rozložený na tokeny převedené na vektory, mechanismem pozornosti hledá souvislosti v celém kontextovém okně, byla předtrénována na obrovském množství textu a dotrénována na užitečného asistenta. Odpověď píše token po tokenu, podle toho, co je v dané souvislosti nejpravděpodobnější. Dnešní modely pracují i s obrázky, zvukem a dokumenty a umějí volat nástroje: vyhledávat na internetu, počítat, psát a spouštět programy.

Z principu, na kterém model pracuje, plyne jeho nejznámější slabina, kterou se vžilo nazývat halucinací. Model píše text, který je pravděpodobný, ne text, který je ověřený. Většinou je pravděpodobný text zároveň pravdivý, protože model se učil z textů, které byly většinou pravdivé. Tam, kde model odpověď nezná, ale vytvoří text, který vypadá stejně přesvědčivě jako pravdivá odpověď – a nijak nedá najevo, že si není jistý.

Lékaři tento jev znají z neurologie a psychiatrie jako konfabulaci, například u Korsakovova syndromu. Pacient s poruchou paměti vyplní mezery ve vzpomínkách plynulým, přesvědčivým a podrobným vyprávěním, které není pravdivé – a sám ho za nepravdivé nepovažuje. Nelže; mozek mu podsune pravděpodobný příběh tam, kde chybí skutečný.

Kde analogie končí. U pacienta je konfabulace projevem poruchy. U jazykového modelu je to přirozený důsledek toho, jak funguje, a nelze ji úplně odstranit. Dá se ale výrazně zmenšit: když model dostane zdroj, ze kterého má odpovídat – vložený dokument, doporučený postup, výsledky vyhledávání –, opírá se o něj místo o to, co je jen pravděpodobné.

Příklad. Typickou halucinací je vymyšlená citace odborné literatury: věrohodní autoři, existující časopis, správný formát, ročník i strany – a článek, který nikdy nevyšel. V jedné studii z roku 2023 bylo z odkazů, které jazykový model uvedl v lékařských textech, téměř polovina vymyšlených a velká část dalších obsahovala chyby [31.]. Novější modely s přístupem k vyhledávání chybují méně, ale ne nikdy.

Pro praxi. Fakta, citace, dávky a čísla ověřovat. Kde to jde, dát modelu zdroj a chtít od něj, aby uvedl, kde přesně v něm odpověď našel. Sebejistý tón odpovědi nic nevypovídá o její správnosti.

5 Zkušenost z praxe: simulátory psané s jazykovým modelem

Principy popsané výše se dají dobře ukázat na vlastní zkušenosti. Webové simulátory, o kterých pojednávají dva další příspěvky tohoto sborníku, vznikaly v roce 2026 ve spolupráci s jazykovým modelem (Claude Code). Model psal program; autor rozhodoval, co se bude dělat, kontroloval výsledky a nesl za ně odpovědnost. Za několik měsíců takové práce se objevily všechny principy z předchozích kapitol – nejvýrazněji principy 2, 7, 9 a 10.

5.1 Model se neučí, učí se projekt

Jedna chyba se opakovala stále znovu: nástroj pro přípravu výukových scénářů při ukládání zahazoval údaje, které sám

DOI: 10.35191/medsoft_2026_1_38_kofranek_l

neupravoval. Model ji opravil, ale v další pracovní relaci ji při jiné úpravě udělal znovu – přesně jak to popisuje princip 2: z rozhovoru se nic nenaučil. Pomohlo teprve pravidlo zapsané do souboru, který model čte na začátku každé práce, a to s výmluvným dovětkem „pošesté se to nesmí opakovat". Znalost projektu tedy nežije v modelu, ale v zápiscích, které mu člověk předkládá.

5.2 Kontextové okno a deník

Dlouhá práce se do kontextového okna nevejde. Když se zaplní, prostředí starší část rozhovoru nahradí stručným shrnutím – a co ve shrnutí chybí, je ztraceno (princip 7). Projekt proto vede pravidla a jejich zdůvodnění v souborech mimo model, podobně jako pacient s poruchou paměti, který si vede deník. Zajímavé je, že tento deník sám narostl tak, že zabíral příliš velkou část okna, a bylo nutné ho zkrátit ze 104 kB na zlomek; podrobnosti se přesunuly do samostatných souborů, které model čte, jen když je potřebuje.

5.3 Kontrola, která nemohla selhat

Jazykový model rád hlásí úspěch (principy 9 a 10). Nejnázornější příklad přinesla kontrola zobrazení simulátorů na mobilním telefonu. Model napsal test, který měřil, zda stránka na displeji telefonu nepřetéká do strany, a test prošel u všech třiceti pohledů. Teprve pohled na snímky obrazovky ukázal, že stránka se na telefonu vykresluje jako na počítači a zmenšuje se na třetinu – chyběl jí jeden řádek, který prohlížeči říká, že se má přizpůsobit šířce displeje. Test měřil proti šířce stránky, a ta byla stejně chybná jako stránka sama. Z podobných zkušeností vzniklo pravidlo projektu: každá kontrola se ověří tím, že se ukáže, že dokáže selhat – pustí se na záměrně pokaženou stránku a musí ji odhalit.

5.4 Čísla se měří, nevymýšlejí

Výukové scénáře obsahují čísla: „tlak na paži je 125/78 mmHg, srdeční výdej 5,1 l/min". Pravidlem projektu je, že každé takové číslo je změřené na běžícím modelu, ne napsané z hlavy, a nástroj porovnává čísla v textu s tím, co student na stránce skutečně uvidí. Právě toto porovnání odhalilo tři scénáře, ve kterých byla čísla změřena za jiných podmínek, než za jakých je pak student uvidí – text sliboval jiné hodnoty než stránka. A výpočet modelu v prohlížeči se ověřuje proti původnímu simulačnímu programu; odchylka je v setinách procenta.

5.5 Dělba práce

Jazykový model píše program rychle a interaktivně přesně podle specifikace zadání. Co je správně, ale rozhoduje člověk, který rozumí fyziologii i modelu: zda je odpověď modelu fyziologicky věrohodná, zda scénář učí to, co má učit, a zda kontrola opravdu něco kontroluje. Tato zkušenost potvrzuje poslední pravidlo závěru tohoto článku.

6 Fámy a co na ně říkají principy

„AI myslí a rozumí jako člověk." Jazykový model píše nejpravděpodobnější pokračování textu (principy 6 a 10). To, že výsledek často vypadá jako porozumění, je důsledek obrovského množství textu, ze kterého se model učil, a toho, jak podrobně musel jazyk zachytit, aby další slovo odhadoval dobře. Zda jde o „porozumění", je spíš filozofická otázka; jisté je, že model nemá vědomí, záměry ani odpovědnost.

„AI ví všechno." Model ví to, co bylo v textech, ze kterých se učil, a jen do data, kdy jeho trénování skončilo (principy 7 a 9). Mezery v znalostech nepozná a vyplní je pravděpodobným textem (princip 10).

„AI se učí z mého rozhovoru." Váhy modelu se při používání nemění (princip 2). To, co si služba z rozhovoru ukládá, není učení, ale může to být uložené u provozovatele – a to je otázka ochrany dat, ne inteligence.

„AI je jen hloupé doplňování slov." Opačná fáma. Model sice opravdu doplňuje slova, ale aby to dělal dobře v obrovském měřítku, musel se naučit mnohem víc než slova: z doplňování textu ve velkém vznikají schopnosti, které malé modely nemají (princip 8). Podceňování je stejně nebezpečné jako přeceňování.

„**AI nahradí programátory**". AI v dohledné době programátory nenahradí, ale zásadně mění jejich roli z pouhých „písařů kódu" na architekty a auditory systémů. Znalost teoretických základů a principů informatiky má dnes naopak **větší smysl než kdy dříve**. Umělá inteligence (jako GitHub Copilot nebo Claude) funguje jako extrémně rychlý asistent. Dokáže bleskově vygenerovat šablonovitý kód, napsat testy nebo najít syntaktickou chybu a výrazně zefektivňuje práci. Zároveň ale roste význam dovedností, jako je návrh architektury, práce s daty nebo schopnost porozumět problému a kriticky vyhodnotit výsledné řešení.

„AI nahradí lékaře." AI zrychluje práci toho, kdo rozumí. Kdo nerozumí, nepozná, kdy se model mýlí – a model se mýlí přesvědčivě (princip 10). Odpovědnost za pacienta zůstává na lékaři.

7 Závěr

Sen, se kterým jsme v osmdesátých letech stavěli expertní systémy, se splnil: s počítačem lze mluvit o odborné problematice přirozenou řečí a on pomáhá s odbornou prací. Splnil se ale jinak, než jsme si představovali. Místo pravidel, která někdo napsal a která lze přečíst, stojí za dnešní AI miliardy vah, které nikdo přečíst neumí; místo spolehlivé logiky pravděpodobnost. Tento sen má proto i jiné slabiny, a kdo s AI pracuje, měl by je znát. Deset principů z tohoto článku se dá shrnout do pěti pravidel pro praxi:

Model neví, kdy neví. Fakta, citace, dávky a čísla ověřovat (principy 3 a 10).

Model vidí jen svůj kontext. Co se mu neřekne, to nezná; co je v dlouhém rozhovoru daleko, to snadno přehlédne (principy 5 a 7).

Model se z rozhovoru neučí. Oprava platí jen do konce konverzace (princip 2).

Model rád souhlasí. Ptát se neutrálně a chtít i argumenty proti (princip 9).

Co se modelu napíše, opouští počítač. Osobní údaje pacientů do veřejných služeb nepatří.

A nad nimi jedno obecné: umělá inteligence nenahrazuje porozumění, ale zrychluje práci toho, kdo rozumí.

TEN PRINCIPLES OF MODERN AI

Abstract

Modern artificial intelligence (AI) has changed dramatically in recent years. From rigid rule-based systems, such as the medical expert systems of the 1970s and 1980s, we have moved to dynamic, adaptive and generative models that do not receive their rules from humans but derive them from data; the turning point came with graphics processors, which made it possible to actually train neural networks. The author, who began working with AI in the 1980s on rule-based expert systems, aims to set straight the myths that circulate about AI. The article is intended for physicians, medical students and everyone who uses AI tools without developing them. In an accessible way and with concrete examples, it explains the ten pillars on which today's AI

tools rest: neural networks, training and using a model, splitting text into tokens, the numerical representation of meaning, the attention mechanism, the transformer architecture, the context window, scaling laws, pre-training and fine-tuning, and finally large language models. Each principle is explained through an analogy from physiology or medical practice, and we show what follows from it for everyday work with AI: why a language model sometimes states falsehoods with confidence, why it forgets the beginning of a long conversation, and why its outputs must be verified.

Keywords

artificial intelligence, large language models, neural networks, medical practice, medical education, safe use of AI

Doporučené zdroje

[1.] Petr Koubský: Nepřehánějme to s antropomorfním pohledem. AI je mašina, jsou to čísla v počítači. Získáno: https://youtu.be/1rHc68t3 × 1o

[2.] 3Blue1Brown (Grant Sanderson): videosérie o neuronových sítích a video „Attention in transformers, visually explained" – nejnázornější vizuální výklad principů 1, 2, 5 a 6, bez nutnosti programovat. Získáno: https://www.youtube.com/@3blue1brown

[3.] Andrej Karpathy: „Intro to Large Language Models" – hodinová přednáška pro širokou veřejnost o tom, jak jazykový model vzniká a co umí. Získáno: https://www.youtube.com/watch?v=zjkBMFhNj_g

[4.] Michael Nielsen: Neural Networks and Deep Learning – volně dostupná online kniha pro čtenáře, který se nebojí trochy matematiky. Získáno: http://neuralnetworksanddeeplearning.com

[5.] 100+ AI Concepts for Developers Explained in 20 Minutes. (Part 01) Video, které bylo inspirací pro definici 10 principů AI. Získáno: https://youtu.be/IauULFe1j-A

Literatura

[1.] Shortliffe EH. Computer-Based Medical Consultations: MYCIN. New York: Elsevier; 1976.

[2.] Valášek M, Bek R, Pokorný Z, Kofránek J. Základy teorie nesubjektivního expertního systému. In: Metody a prostředky ke zkvalitňování řízení a rozhodování v biotechnických systémech. Praha: ČVUT, Knižnice ČSVTS-FEL; 1987. s. 72–78.

[3.] Pokorný Z, Kofránek J, Cirýn J. Model vnitřního prostředí jako součást báze znalostí pro expertní systém. In: Sborník súhrnov IX. patofyziologického kongresu. Košice; 1987. s. 116.

[4.] Valášek M, Bek R, Pokorný Z, Kofránek J. Expert system synthesis on the base of causal system description. In: Abstracts of the 2nd Czechoslovak Conference on Biomedical Engineering. Bratislava; 1987. s. I-1.

[5.] Bek R, Valášek M, Pokorný Z, Kofránek J. Otázka kauzálního popisu v expertních systémech. Acta Polytechnica. 1988;13: 31–62.

[6.] Pokorný Z, Kofránek J, Cirýn J. Model vnitřního prostředí jako součást báze znalostí pro expertní systém. Čs. fyziologie. 1988;37: 360.

[7.] Pokorný Z, Kofránek J, Cirýn J. Požadavky na strukturalizaci báze znalostí pro rozsáhlý medicínský expertní systém. In: APLES '88 – Aplikace expertních systémů. České Budějovice: Dům techniky; 1988. s. 138–142.

[8.] Valášek M, Bek R, Pokorný Z, Kofránek J. The theory of nonsubjective expert system based on causal analysis, new interpretation of fuzzy description and pattern recognition. Computers and Artificial Intelligence. 1988;7(8): 413–428.

[9.] Kofránek J, Popper M. Meze expertních systémů v medicíně. In: APLES '90 – problémy aplikace expertních systémů. České Budějovice: Dům techniky ČSVTS; 1990. s. 45–60.

[10.] Valášek M, Bek R, Pokorný Z, Kofránek J. Expertní systém bez experta? In: Mařík V, Zdráhal Z, editoři. Metody umělé inteligence a expertní systémy. Praha: Knižnice ČSVTS-FEL; 1985. Příloha, s. 1–4.

[11.] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys. 1943;5: 115–133. doi:10.1007/BF02478259

[12.] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;323: 533–536. doi:10.1038/323533a0

[13.] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012;25: 1097–1105.

[14.] Gulshan V, Peng L, Coram M, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA. 2016;316: 2402–2410. doi:10.1001/jama.2016.17216

[15.] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016. s. 1715–1725. doi:10.18653/v1/P16–1162

[16.] Georgopoulos AP, Schwartz AB, Kettner RE. Neuronal population coding of movement direction. Science. 1986;233: 1416–1419. doi:10.1126/science.3749885

[17.] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv:1301.3781. 2013. doi:10.48550/arXiv.1301.3781

[18.] Lee TB, Trott S. Large language models, explained with a minimum of math and jargon. Understanding AI. 27. 7. 2023. Získáno: https://www.understandingai.org/p/large-language-models-explained-with

[19.] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Advances in Neural Information Processing Systems. 2017;30: 5998–6008.

[20.] Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol. 1962;160: 106–154. doi:10.1113/jphysiol.1962.sp006837

[21.] Scoville WB, Milner B. Loss of recent memory after bilateral hippocampal lesions. J Neurol Neurosurg Psychiatry. 1957;20: 11–21. doi:10.1136/jnnp.20.1.11

[22.] Liu NF, Lin K, Hewitt J, et al. Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics. 2024;12: 157–173. doi:10.1162/tacl_a_00638

[23.] Kaplan J, McCandlish S, Henighan T, et al. Scaling laws for neural language models. arXiv:2001.08361. 2020. doi:10.48550/arXiv.2001.08361

[24.] Kleiber M. Body size and metabolism. Hilgardia. 1932;6: 315–353. doi:10.3733/hilg.v06n11p315

[25.] Hoffmann J, Borgeaud S, Mensch A, et al. Training compute-optimal large language models. Advances in Neural Information Processing Systems. 2022;35: 30016–30030. doi:10.52202/068431–2176

[26.] Brown TB, Mann B, Ryder N, et al. Language models are few-shot learners. Advances in Neural Information Processing Systems. 2020;33: 1877–1901.

[27.] Wei J, Tay Y, Bommasani R, et al. Emergent abilities of large language models. Transactions on Machine Learning Research. 2022.

[28.] Schaeffer R, Miranda B, Koyejo S. Are emergent abilities of large language models a mirage? Advances in Neural Information Processing Systems. 2023;36: 55565–55581. doi:10.52202/075280–2425

[29.] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems. 2022;35: 27730–27744. doi:10.52202/068431–2011

[30.] Sharma M, Tong M, Korbak T, et al. Towards understanding sycophancy in language models. International Conference on Learning Representations. 2024.

[31.] Bhattacharyya M, Miller VM, Bhattacharyya D, Miller LE. High rates of fabricated and inaccurate references in ChatGPT-generated medical content. Cureus. 2023;15: e39238. doi:10.7759/cureus.39238

DOI: 10.35191/medsoft_2026_1_38_kofranek_l

Kontakt

doc. MUDr. Jiří Kofránek, CSc.
Oddělení biokybernetiky
Ústav patologické fyziologie 1. LF UK
U Nemocnice 5, 128 53 Praha 2
+420 777 68 68 68
kofranek@gmail.com