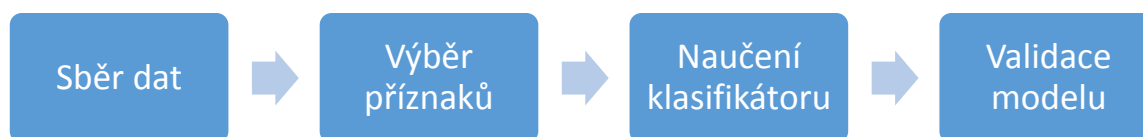


Klasifikace dat pomocí konvoluční neuronové sítě

Konvoluční neuronové sítě (KNS) se v posledních letech staly důležitou metodou zpracování dat. Nejčastěji se s těmito metodami člověk setkává v souvislosti s vývojem samoříditelných automobilů nebo v různých mobilních aplikacích, které rozpoznávají některé druhy zvířete, rostlin nebo hub. Zkrátka, KNS jsou fenomén, kterým by se měl každý matematický biolog, podobně jako student matematických, technických nebo neurovědních oborů, během svého studia do jisté míry zabývat. Tento studijní materiál shrnuje teoretické základy klasifikace dat a konvolučních neuronových sítí a je omezen pouze na architekturu a výpočet konvoluční neuronové sítě. Problematika adaptivních algoritmů bude kompletně vynechána.

Klasifikace obrazových dat

Schéma na obrázku 1 znázorňuje postup při řešení úlohy klasifikace dat, jejímž cílem je zařazení objektů do tříd. Nejprve je potřeba nasbírat dostatečné množství dat. V případě KNS je žádoucí mít velký datový soubor čítající i miliony obrazů. Takový datový soubor je ještě potřeba rozdělit na 2 části – testovací a trénovací data. Na datech trénovacích se učí model KNS klasifikovat objekty do skupin a na testovacích datech se vyhodnocuje kvalita modelu. Druhým bodem je předzpracování dat, nejčastěji jde o selekci příznaků (popsána níže), extrakci příznaků, registraci obrazu na standardní šablonu apod. Konvoluční neuronové sítě se s tímto krokem umí vypořádat po svém, neboť samotná síť kromě klasifikace provádí i předzpracování dat (extrakce příznaků) a povinnost tohoto kroku tedy odpadá. Parametry předzpracování se definují během fáze adaptace KNS. Následuje krok, ve kterém je vybrán a naučen klasifikátor na trénovacích datech. A posledním krokem je vyhodnocení úspěšnosti klasifikace na nezávislém testovacím datovém souboru. Jednotlivé kroky jsou detailněji popsány níže.



Obrázek 1: Postup klasifikace dat

Výběr příznaků: Pro optimální naučení klasifikátoru je výhodné udělat předvýběr příznaků, které jsou důležité pro rozlišení objektů (selekce příznaků). Jako příklad uveďme rozlišení mezi psy a kočkami. Informace o tom, že mají 4 nohy, 1 ocas a 2 oči nám nebude nikterak užitečná, protože je oběma druhům společná. Velikost těla už může být lepším příznakem, protože pes bývá typicky větší, nicméně některá plemena jsou malá a proto je to stále nedostačující informace. Dalšími rozlišujícími znaky mohou být barva, délka srsti nebo tvar hlavy. Pokud v datovém souboru vybereme správné proměnné, máme předpoklad pro naučení úspěšného klasifikátoru. Existuje mnoho metod selekce příznaků, například Fisherovo diskriminační kritérium, dvouvýběrový t-test (v případě 2 tříd) nebo dopředná a zpětná selekce příznaků. Extrakci příznaků potom nazýváme proces předzpracování, kdy kromě výběru podmnožiny příznaků zároveň měníme i příznakový prostor např. PCA, ICA. Zopakujme si, že konvoluční neuronová síť umožňuje tento krok analýzy provádět adaptivně a není proto povinný.

Klasifikátor: Jde o matematickou rovnici, jejímž cílem je přiřazovat datům (např. obrázku zvířete) kategorie (např. pes). Rovnice má zpravidla obor hodnot $\langle 0;1 \rangle$ popř. $\langle -1,1 \rangle$ a její proměnné například reprezentují hodnoty jednotlivých pixelů.

- **Např.:** $y = \text{sign}(2x - 1)$
- Když má pixel x hodnotu více než 0.5, pak je $y = 1$ a pixel je modelem klasifikován jako skupina 1
- Když má pixel x hodnotu menší než 0.5, pak je $y = -1$ (pixel klasifikován jako skupina -1)

Koeficienty rovnice se spočítají na základě předložených tréninkových dat. Tj. model se naučí klasifikovat ze zkušenosti. Kromě konvolučních neuronových sítí mezi klasifikátory řadíme např. logistickou regresii, metodu podpurných vektorů, Naivní Bayesův klasifikátor, FLDA nebo neuronové sítě (MLP, LVQ, apod.).

Učení klasifikátoru: Rozhodování klasifikátoru je optimalizováno na předloženém datovém souboru. Děje se tak pomocí adaptivních algoritmů, kterých je celá řada. Například metoda maximální věrohodnosti (logistická regrese) nebo algoritmus zpětného šíření chyby (neuronové sítě). V případě neuronových sítí se v procesu učení počítá hodnota tzv. **chybové funkce** (cost function), která udává, jak dobře klasifikátor rozlišuje objekty tréninkové množiny, které se učí rozpoznávat. Příkladem chybové funkce je střední kvadratická chyba (MSE) nebo cross-entropie. V rámci učení se upravují koeficienty modelu, efektivní učení je takové, kdy se hodnota chybové funkce postupně snižuje.

Validace a vyhodnocení úspěšnosti klasifikace: Úspěšnost klasifikace se testuje na objektech (datech), na kterých se naučený model neadaptoval. V případě velkých datových souborů se na počátku analýzy odloží část dat, která se použije pro testování úspěšnosti klasifikátoru. Pro malé datové soubory se používají složitější validační metody jako leave-one-out křížová validace apod. Na testovacích datech se zjistí, jak úspěšně klasifikátor zařazuje objekty do naučených tříd. Každému testovacímu objektu se přiřadí třída a vypočítá se **matice záměn**. Jako příklad zde uvedeme matici záměn pro 10 testovacích obrázků trojúhelníků a koleček.

- Matice záměn:

	Výsledek klasifikace	
	Kolečko	Trojúhelník
Správná třída	4	2
	1	3

- Celková správnost klasifikace se udává v [%] a jde o poměr správně klasifikovaných obrazů ku všem obrazům. Např. (suma hlavní diagonály matice záměn) / n .
- V tomto případě bude celková správnost 70 % (7 z 10 správně zařazených obrazů)
Pro binární klasifikátory (2 třídy) lze počítat senzitivitu, specificitu, pozitivní a negativní prediktivní hodnotu.

Konvoluční neuronová síť

Motivace

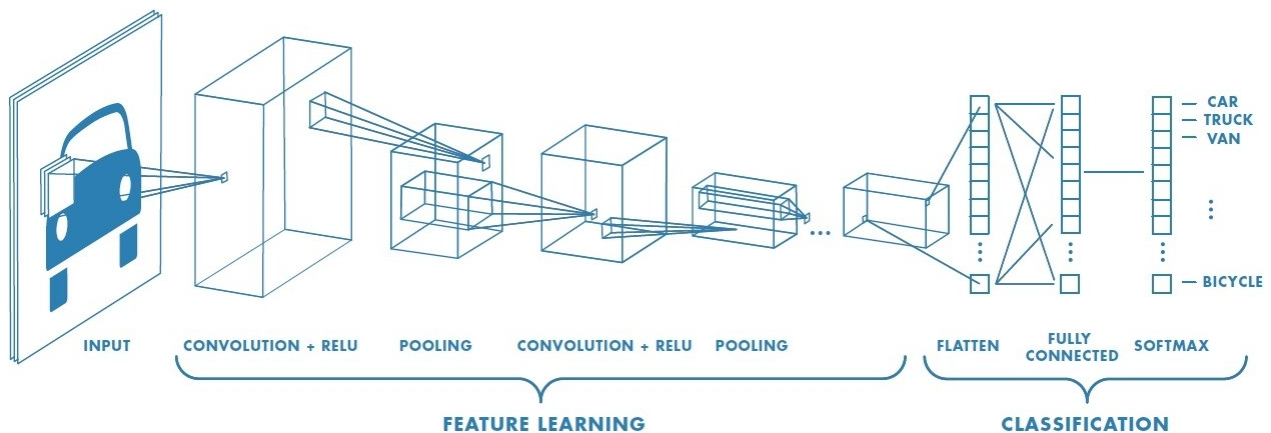
Představme si, že bychom chtěli pomocí KNS rozpoznat číslice. Na obrázku 2 jsou znázorněné 2 číslice „4“ a jejich překryv. V případě, že bychom se zabývaly absolutním umístěním pixelu (jakožto jednotkou obrazu) a považovaly ho za příznak (proměnnou), zcela běžně se stane, že absolutní pozice odpovídajících si pixelů číslic nebudou totožné. Mohli bychom využít některou z metod registrace obrazu, abychom získali správný překryv, ale to celou analýzu zesložituje. Místo toho můžeme využít operaci konvoluce, která v každém místě obrazu hledá nějaký tvar např. svislou nebo šikmou čárku a není třeba vázat se na absolutní polohu pixelu.



Obrázek 2: Podobně zapsaná číslice může zabírat různé pixely (modře rozdíl)

Architektura a výpočet

Architektura konvoluční neuronové sítě je znázorněna na obrázku 3. Na vstupu jsou data, v našem případě to bude 2D barevný obraz tj. 3D matice (barvy RGB jsou třetím rozměrem). Obraz je neuronovou sítí zpracováván po krocích (vrstvách). Každá vrstva provádí nějakou matematickou operaci např. konvoluci, podvzorkování nebo násobení vektorů. Parametry těchto operací jsou získávány v procesu učení. U KNS se zpravidla v prvních pár vrstvách střídá konvoluční vrstva s poolingovou (podvzorkovací) vrstvou a to v několika opakováních podle toho, jak chceme mít KNS hlubokou. Na poslední podvzorkovací vrstvu navazují vrstvy, které se používají u klasické vícevrstvé perceptronové neuronové sítě a síť uzavírá tzv. SOFTMAX vrstva. Klasifikace obrazu se provádí tak, že se do KNS vloží obrázek a po výpočtu všech matematických operací je výstupem pravděpodobnost zařazení předloženého obrazu do každé ze tříd.



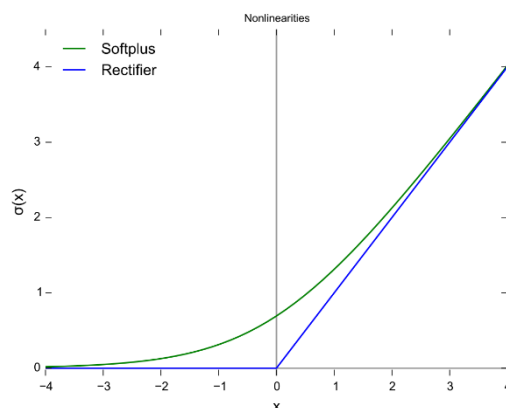
Obrázek 3: Konvoluční neuronová síť (zdroj: www.mathworks.com)

Vrstvy konvoluční neuronové sítě

Konvoluční vrstva: V první vrstvě probíhá operace konvoluce mezi obrázkem a konvolučním jádrem tj. nějakým vzorem, který se v procesu učení vytvoří. Výsledkem konvoluce je matice čísel tzv. příznaková mapa. Výpočet si můžeme představit i jako skalární součin matice konvolučního jádra (obr. 4) a každou možnou oblastí obrazu odpovídající dimenze (máme-li obraz velikosti 5×5 a konvoluční jádro velikosti 3×3, pak takových oblastí obrazu bude 9). Výsledkem tohoto skalárního součinu je jedno číslo příznakové mapy. Toto číslo vstupuje do RELU aktivační funkce (obr. 5), díky které je do modelu zavedena nelinearita, která umožňuje KNS řešit i komplexní nelineární úlohy. Konvolučních jader bývá zpravidla více a jejich počet je volným parametrem modelu, stejně jako dimenze konvolučního jádra a délka posunu. Příznakové mapy se spojují v tensor tak jak je uvedeno na obrázku 3 a konvoluce v dalších vrstvách probíhá nad celým tensorem. Pro lepší představu jak se počítá konvoluce se podívejte na animace pod odkazem: <https://cs.scienceval.com/81012-types-of-convolutions-in-deep-learning-717013397f4d-15>

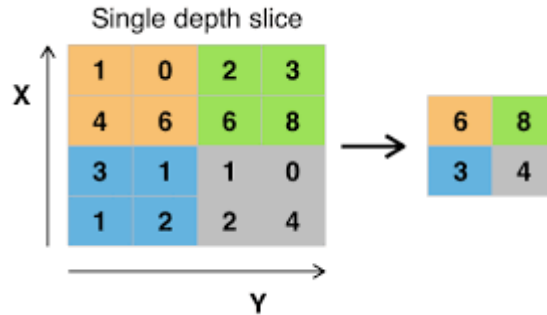


Obrázek 4: Příklad konvolučního jádra



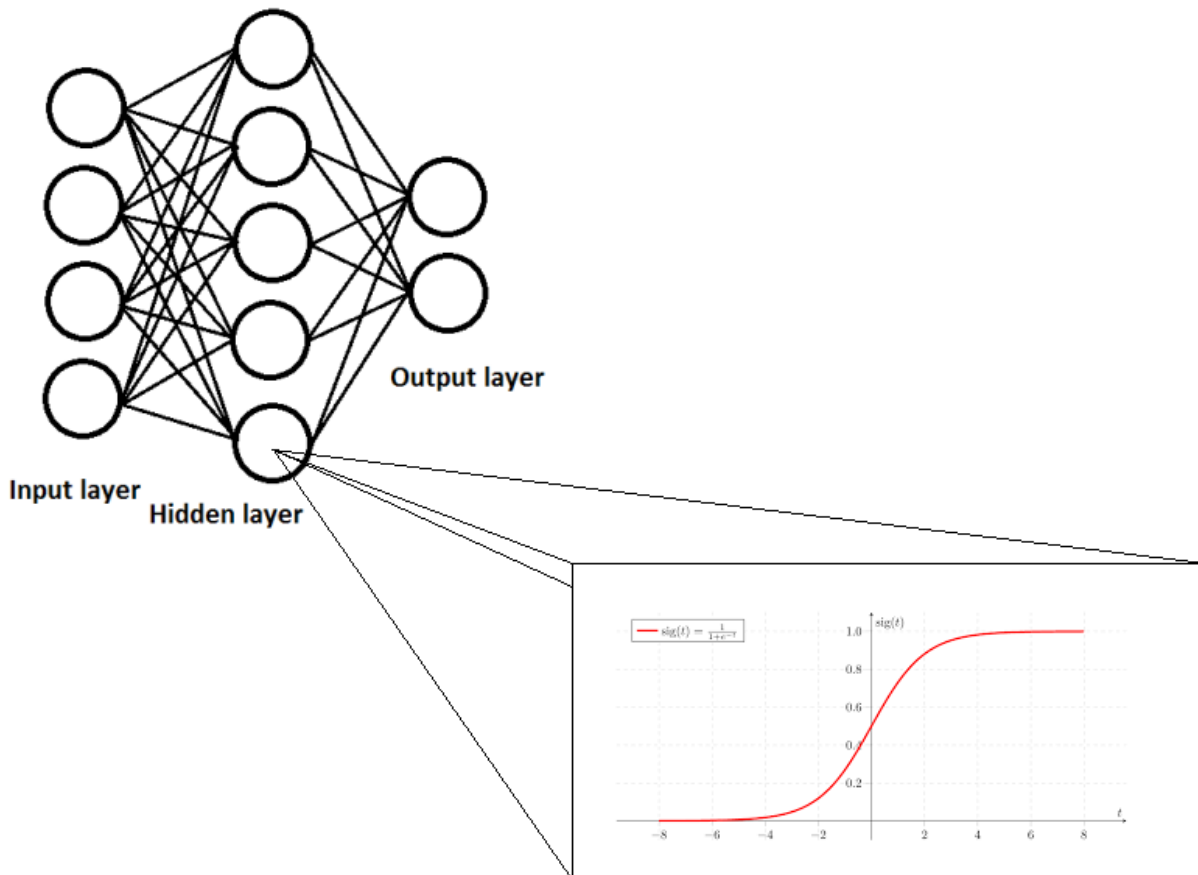
Obrázek 5: Aktivační funkce softplus a ReLU (zdroj: wikipedia)

Podvzrokovací (poolingová) vrstva: Tato vrstva slouží ke zjednodušení informace získané pomocí předchozí konvoluční vrstvy. Výsledkem je příznaková mapa s nižší dimenzí. Úplná informace z předchozí vrstvy je sice ztracena, ale hrubá informace o relativním umístění vzoru v původním obraze zůstává. Nejčastější metodou je tzv. **max-pooling**. Principem této metody (obr. 6) je vybrat pouze nejvyšší číslo z vyhraničených podoblastí příznakové mapy (např. dimenze 2×2, 3×3,...). Tato metoda je nejčastější nicméně existují i další jako L2 pooling nebo average pooling.



Obrázek 6: Znárodnění výpočtu metody max-pooling – z každé oblasti se vybere největší číselná reprezentace (zdroj: wikipedia)

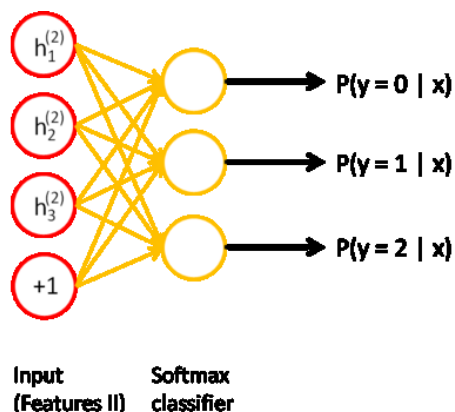
Vrstva s klasickými neurony: V této vrstvě dochází ke zpracování dat pomocí skalárního součinu vstupů (výstupů předchozí vrstvy) a koeficientů, které jsou získány v procesu učení. Výsledek skalárního součinu slouží jako argument funkce logistické sigmoidy (funkce zavádějící nelinearitu do vrstvy modelu), jejíž výsledek je mezi 0 a 1. Místo logistické sigmoidy lze použít i jiné funkce např. hyperbolický tangens. (obr. 7)



Obrázek 7: Znárodnění klasické neuronové sítě (multi-layer perceptron) a její typické aktivační funkce – logistické sigmoidy (zdroj: wikipedia, upraveno)

SOFTMAX: Poslední vrstva (obr. 8) obsahuje tolik neuronů (výstupních čísel), kolik je v datovém souboru kategorií, pak každý neuron reprezentuje jednu třídu (kočka, pes, pták, ...). Výstupní vektor čísel se upraví

tak, aby jeho součet byl 1 a všechna čísla byla mezi 0 a 1, aby se zařazení do tříd mohlo interpretovat jako aposteriorní pravděpodobnost.



Obrázek 8: Poslední vrstva konvoluční neuronové sítě udává pravděpodobnost zařazení klasifikovaného subjektu do každé ze tříd (zdroj: <http://eric-yuan.me/simple-deep-network>)

Počet opakování vrstev konvoluční neuronové sítě může být libovolný. V praxi se běžně opakuje střídání dvojic konvoluční a poolingové vrstvy, na které navazují vrstvy klasické neuronové sítě a na závěr vrstva SOFTMAX. Čím více vrstev je přidáno, tím více vztahů v datech je KNS schopna pojmut v procesu učení, ale také se bude déle adaptovat.

Více se o konvolučních neuronových sítích můžete dočíst například zde:

A. Krizhevsky, I. Sutskever, a G. E. Hinton, „ImageNet Classification with Deep Convolutional Neural Networks“, in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, a K. Q. Weinberger, Ed. Curran Associates, Inc., 2012, s. 1097–1105.

URL: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>

<https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>