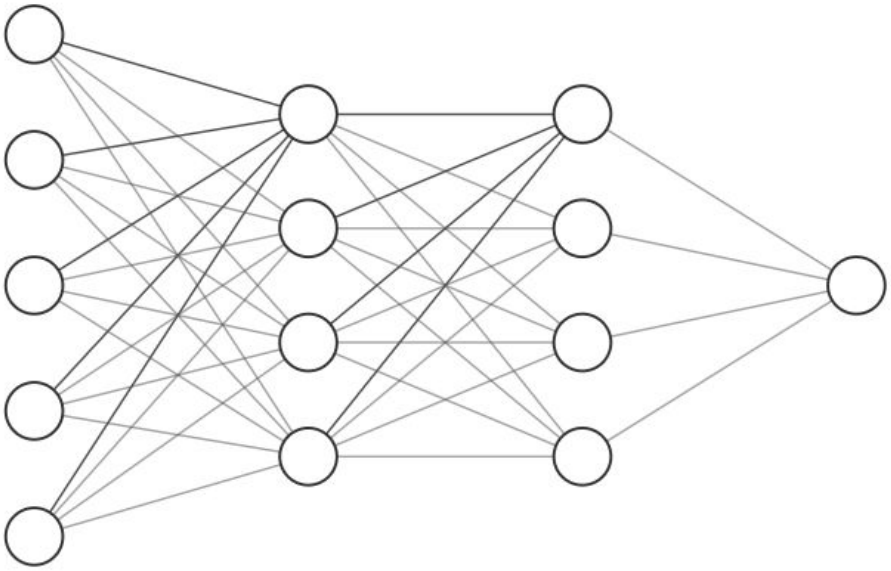
Využití neuronových sítí pro řešení těžkých problémů

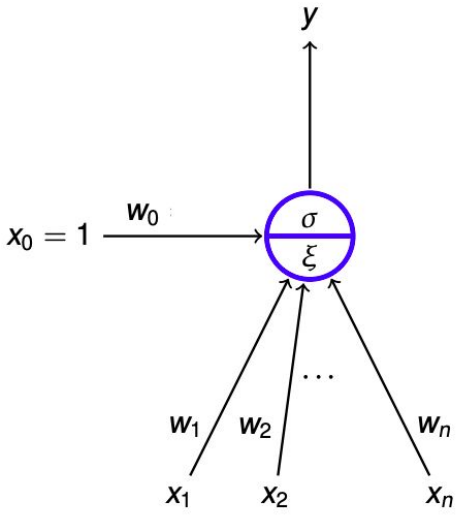


Marek Kadlčík 485294, Michal Starý 485367

Obsah

Teoretický úvod neuronových sítí

Neuronové sítě jsou výpočetní model inspirovaný fungováním mozku. Základní stavební prvek, neuron, se skládá z následujících částí:



$(x_1, x_2, \dots x_n)$ je vstupní vektor R^n

$(w_1, w_2, \dots w_n)$ je vektor vstupujících vah R^n

w_0 je bias neuronu

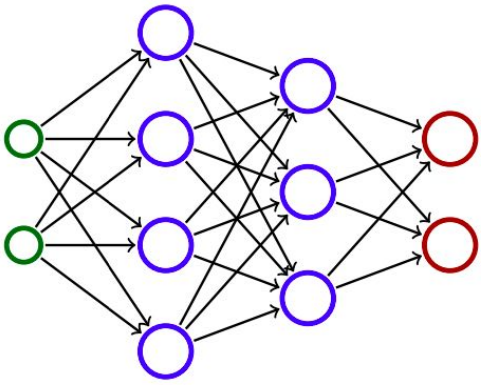
$\xi = (x_0, x_1, \dots x_n) \bullet (w_0, w_1, \dots w_n)$ je potenciál

σ je aktivační funkce

$y = \sigma(\xi)$ je výstup neuronu

Neuronová síť je orientovaný graf neuronů s ohodnocenými hranami, kde se vstupní vektor neuronu skládá z výstupů všech neuronů, ze kterých do něj vede hrana. Vyhodnocení sítě na vstupu $(a_1, a_2, \dots a_p)$ je výpočet, při kterém se vstupní vrstvě neuronů (velikosti p) na výstup dají složky vstupního vektoru. Poté se dokola výběrovým pravidlem vybere podmnožina neuronů a každý vyhodnotí svůj potenciál a aplikací aktivační funkce jeho výstup. Neuronová síť na vstupu zastaví, pokud se pro libovolné další množství iterací nezmění výstupy výstupní vrstvy. Sítě bez cyklů vždy zastaví na libovolném vstupu.

Příklad nejjednodušší architektury neuronové sítě - multilayer perceptron:



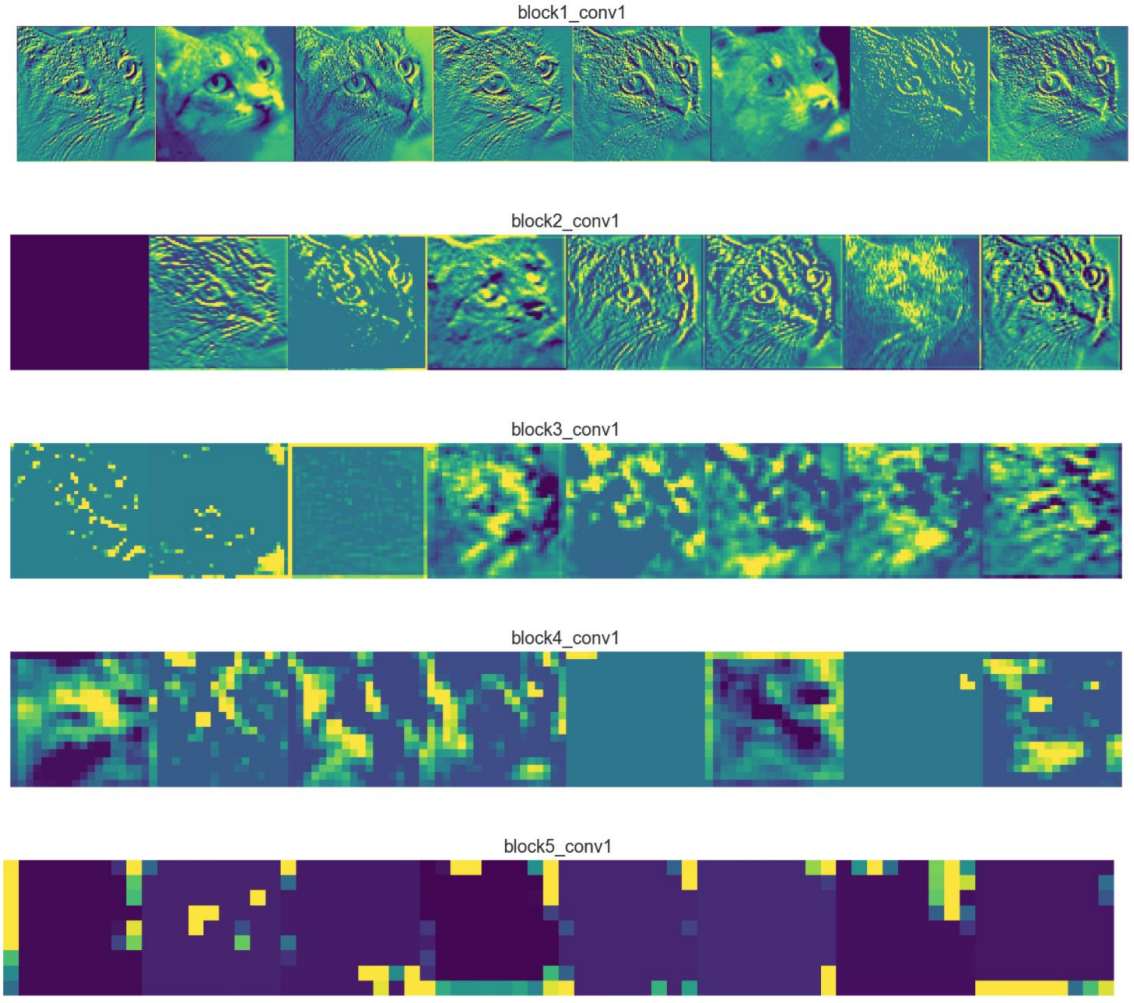
zelená je vstupní vrstva
modré jsou skryté vrstvy
červená je výstupní vrstva

výběrové pravidlo:
vyber v i-tém kroku i-tou vrstvu

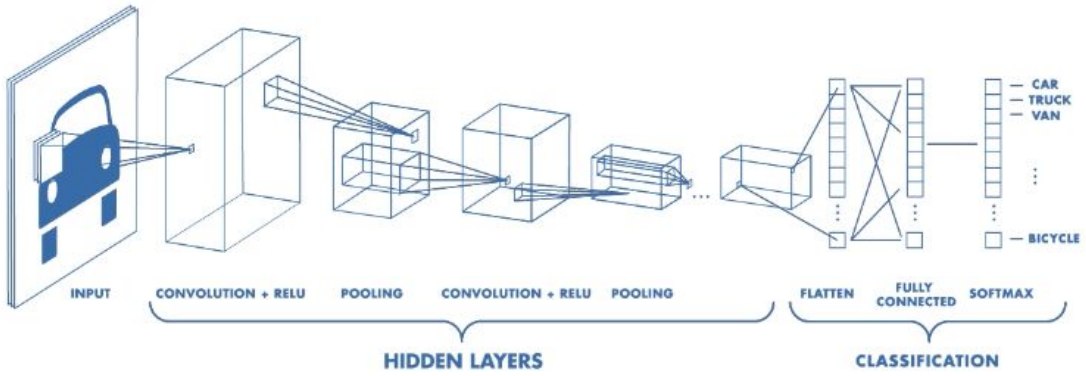
Neuronové sítě umožňují trénování z dat ve tvaru {(vstup, výstup)} upravováním vah mezi neurony. na začátku se vyhodnotí síť na vstupu vzorku, její výstup se porovná s očekávaným výstupem a upravováním vah pomocí gradientního sestupu se minimalizuje její chyba.

Konvoluční neuronové sítě (CNN)

Nejčastější využití konvolučních sítí je v oboru zpracování obrazu (tzv. computer vision). Zde je využito především schopnosti těchto sítí hledat naučené vzory po celé ploše vstupu, včetně brání v potaz dvou dimenzionality obrázku (jsou schopné dívat se na několik sousedních pixelů zároveň). Tyto vzory mohou být od nejzákladnějších typu různých hran, či tvarů až po velmi komplexní, jako jsou například části tváře, tedy nos, ústa, oči, apod. Komplexita těchto vzorů vzrůstá s hloubkou sítě, kde se z jednoduchých vzorů postupně skládají složitější a složitější.

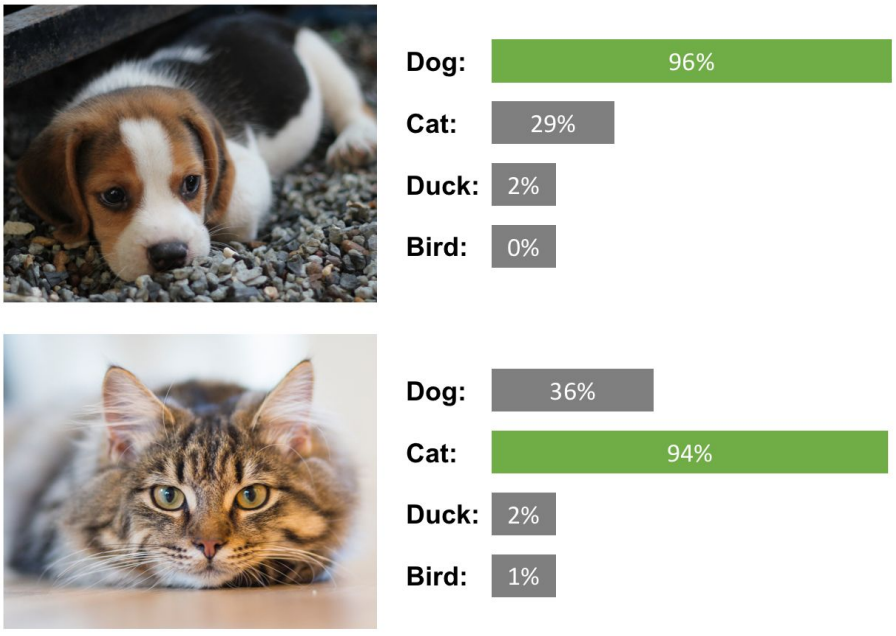


Z technického hlediska je CNN složením 3 druhů vrstev: konvoluční, pooling a fully-connected. Ve většině architektur se nejdříve extrahují vzory pomocí kombinace konvoluční vrstvy, která pomocí filtrů prohledává vstup a následovně pooling vrstva zmenší dimenzionalitu výstupu konvoluční vrstvy. Po několika opakování této kombinace konvoluce a poolingu síť obsahuje již pouze plně propojené vrstvy, kde se tyto vzory dále kombinují, jako by to byli vstupy klasické, plně propojené neuronové sítě (MLP).



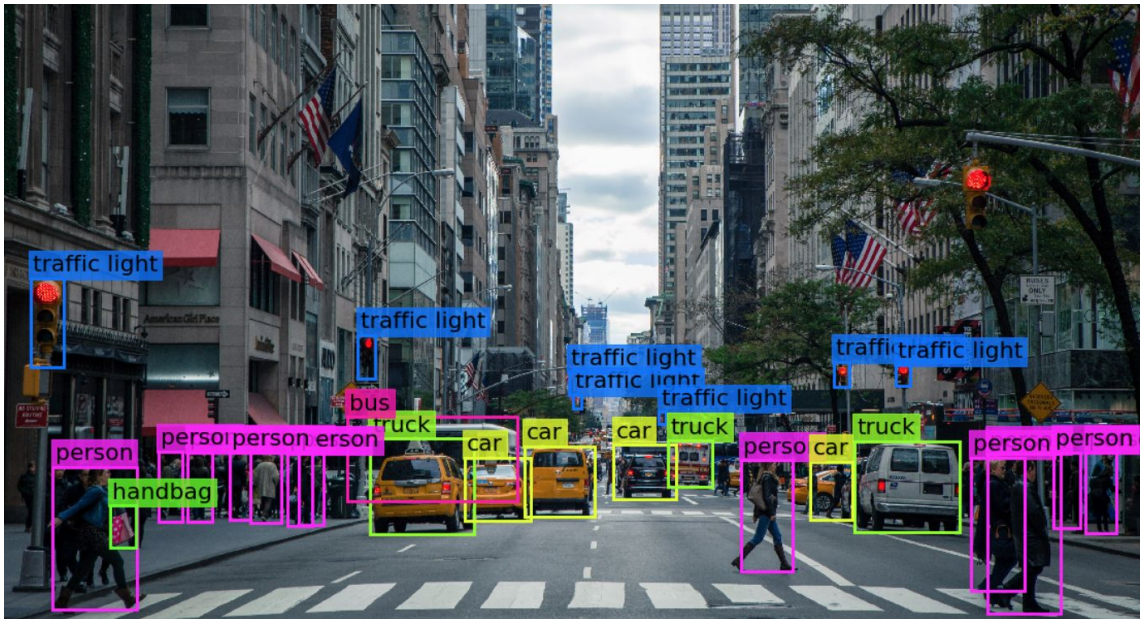
Klasifikace obrázků

Základní problém počítačového zpracování obrazu je klasifikace obrázků, tedy určení toho, co se na obrázku nachází. Nejčastěji je to tedy snaha přidělit obrázku jeden nebo více slovních popisků. Tento problém taky slouží jako benchmark pro nové postupy a algoritmy. Základními datasety, na kterých se tento problém ilustruje bývají MNIST, který obsahuje obrázky ručně psaných číslic, nebo ImageNet, který obsahuje miliony anotovaných obrázků různého druhu.



Detekce objektů z obrázků

Detekce objektů je zobecnění klasifikačního problému, kde se zvolí konečné množství (např. ~100) druhů objektů, které chceme detekovat. Trénovací snímky se anotují pomocí vyznačení rámců, ve kterých se nachází žádané objekty.



Mezi nejznámější a nejúspěšnější algoritmy pro učení NN na detekci objektů se řadí YOLO a RCNN. Oba tyto algoritmy musí řešit jak problém lokalizace objektu na snímku, tak následné rozpoznání, tedy klasificikaci objektu.

Tyto techniky a algoritmy mají velmi široké využití, mezi které se řadí rozpoznávání tváře, lidí, budov. Dále je také rozpoznávání objektů nezbytná součást všech autonomních vozidel.

Pod oblast detekci objektů lze lehce zařadit i rozpoznávání obličejů, či poznávacích značek aut. Systémy založené na rozpoznávání obličejů jsou klíčovou součástí Social Credit System v Číně, kterým vláda monitoruje obyvatelstvo a jeho činy. Díky všudypřítomným kamerám a právě systémům schopných detekce a rozpoznání obličeje je tento systém vysoce soběstačný a efektivní. Stejně tak rozpoznávání poznávacích značek je používána např. i v Brně pro kontrolu parkování v tzv. modrých zónách. Právě toto využití ve zpracování obrazu je jedním z nejvíce stabilních a komerčně dostupných využití neuronových sítí obecně. Jeho masové nasazení také ovšem vyvolává řadu etických otázek, viz Social Credit System v Číně.

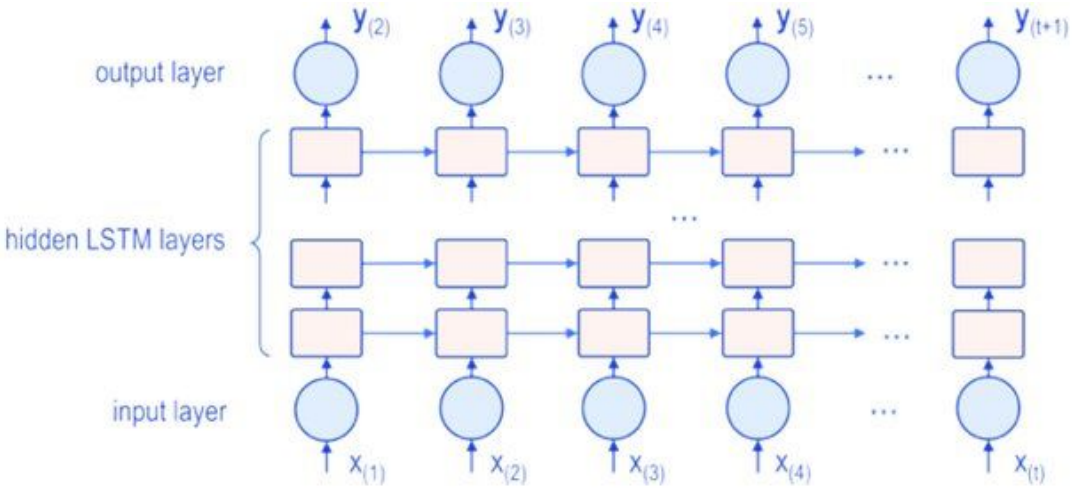
Long Short Term Memory sítě (LSTM)

Long Short Term Memory sítě jsou podtřídou rekurentních sítí. Rekurentní sítě samy o sobě mají schopnost zachovávat informaci o sekvenčnosti dat. Obory ve kterých je tato sekvenčnost jednou z nepostradatelných informací jsou zpracování přirozeného jazyka nebo predikce časových událostí.

Velkým problémem rekurentních sítí je ovšem tzv. “vanishing gradient” případně “exploding gradient”. Vzhledem ke snaze o zachování sekvencionality má celá síť specifickou architekturu, ve které je ovšem zpětná propagace chyby predikce náchylná k podtečení či přetečení čísel s konečnou přesností, neboť propagovaná chyba se neustále zmenšuje. Pokud k tomuto dojde, síť se téměř nic nenaučí. Tento problém je řešen právě pomocí LSTM sítí, které umožňují takovou zpětnou propagaci chyby, že k podtečení/přetečení nedochází.

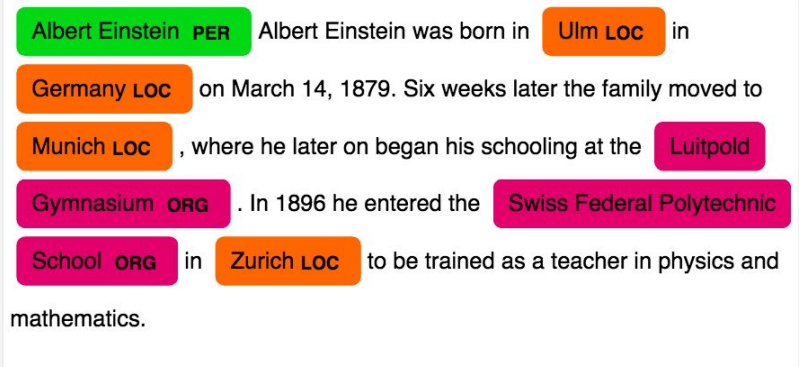
Vzhledem k tomu, že v přirozeném jazyce nezáleží význam slova, či věty pouze na slovech/větách předchozích, ale i následujících, jsou nejpokročilejší modely založené na LSTM sítích obousměrné (tzv. biLSTM). Tyto modely neberou pouze sekvenci předcházejících slov/vět ale i sekvenci následujících. Dívá se tedy do celého okolí slova/věty jejíž význam se snažíme zachytit.

Limitem úspěšnosti LSTM sítí v oblasti přirozeného jazyka se jeví zejména fixní počet kroků (tedy slov) o kolik se síť dívá od zdrojového slova. Proto i přesto, že dokáže úspěšně brát v potaz sémantický význam velmi blízkých slov (přímých sousedů), jakmile má síť brát v potaz sémantický vztah vyjádřený slovem, které je např. 5 slov od zdrojového, stává se tento úkol téměř nesplnitelným, protože se v jiných větách může vyskytnout v úplně jiné vzdálenosti od zdrojového slova.

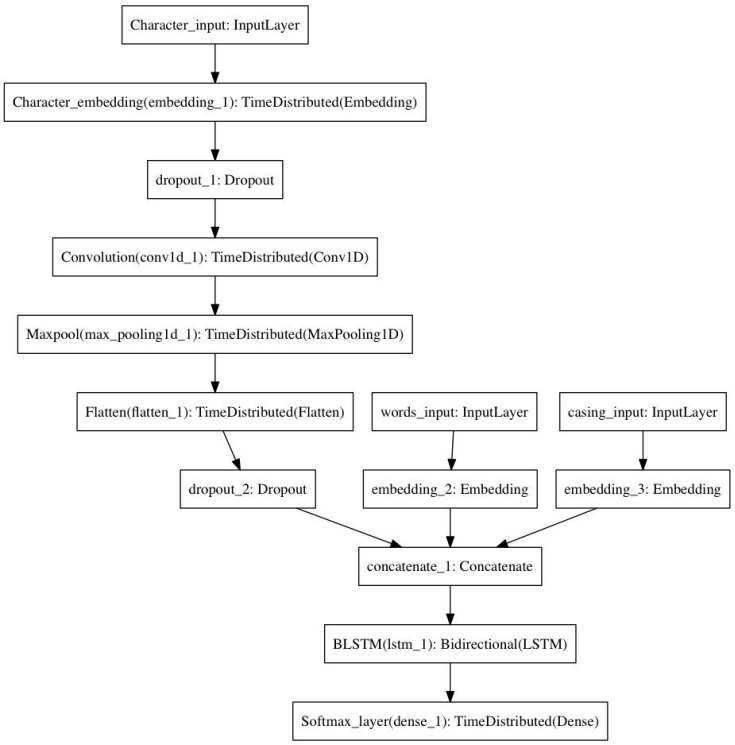


Rozpoznávání pojmenovaných entit (NER)

Rozpoznávání pojmenovaných entit je jeden ze základních problému ve zpracování přirozeného jayzyka. Korektní rozpoznání entit jako jsou jména, datumy, lokality, peněžní částky, atd. je velmi důležitou součástí všech pokročilejších systémů operujících nad jazykem, jako je např. strojový překlad nebo porozumění a sumarizace textu.

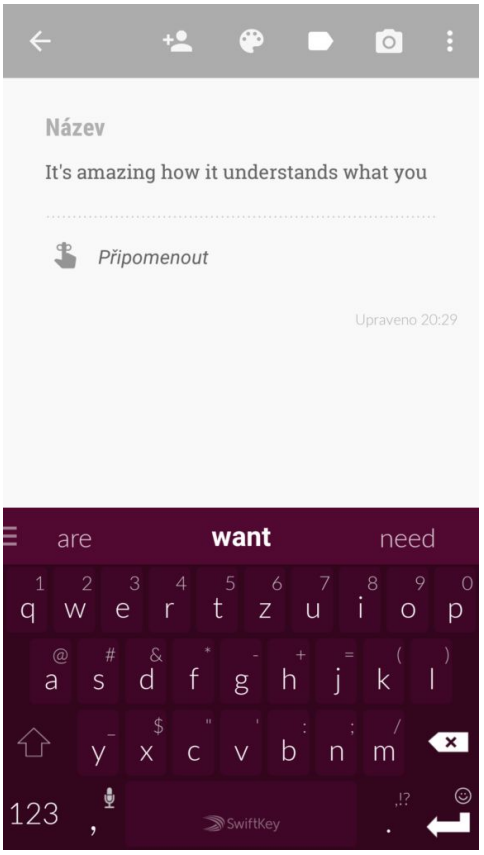


Text se analyzuje na úrovni jednotlivých znaků pomocí konvoluční sítě, vektorizuje na úrovni slov a na vektorizuje základě velkých či malých písmen. Výstupem z každé této části je vektor čísel, který se spojí dohromady. Tento vektor je analyzován LSTM sítí.



Prediktivní klávesnice

Člověk v průměru píše na mobilu o 35% pomaleji než na fyzické klávesnici. Jako pomůcka pro rychlejší psaní lze použít autocomplete a predikci následujících slov.



Klávesnice Swiftkey od Microsoftu začala používat pro predikci rekurentní neuronové sítě v roce 2016 v rámci projektu Neural Alpha. Dříve používali predikci založenou na systému, který nazývají n-gramový model a funguje následovně: z psaného textu vyber posledních n slov a nabídni 3 slova, která se po nich vyskytuje nejčastěji v korpusu.

LSTM architektura je vhodná pro predikci následujícího slova ve větě, kterou píšeme, protože je to sekvence slov, kterou jsou rekurentní sítě schopné zpracovávat. Podle vyjádření SwiftKey neuronová síť umožnila vytvořit model, který bere v potaz větší kontext, do kterého má následující slovo patřit, než předchozí algoritmus.

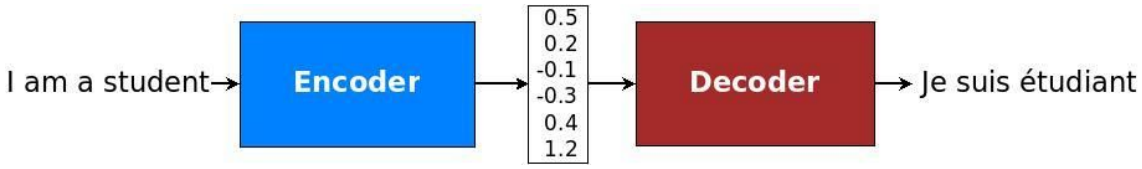
Encoder/Decoder modely

Encoder/Decoder architekury jsou specifickým typem architektur neuronových sítí, které můžou obsahovat libovolnou předchozí specifikaci sítě, tzn MLP, CNN, LSTM (RNN). Snaží se zakódovat vstup do poměrně zhuštěné reprezentace pomocí jedné sítě (Encoderu) a poté z této zhuštěné reprezentace, která by měla co nejlépe zachytit podstatné vlastnosti vstupu dekódovat výstup. Tento výstup může být jiného formátu, ale zachová právě ty podstatné složky vstupu.

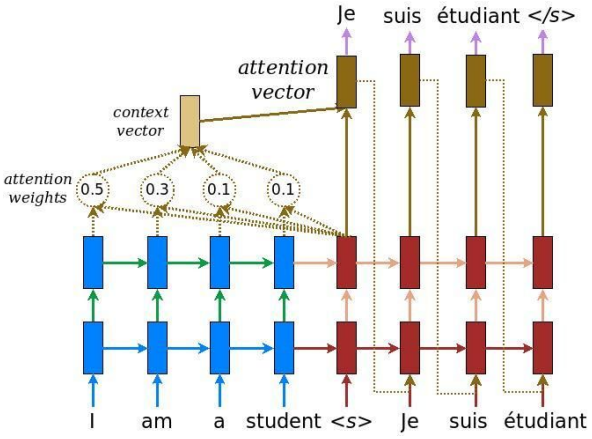
Strojový překlad

Strojový překlad je oblast zpracování přirozeného jazyka, ve které znamenal nástup neuronových sítí obrovský posun kupředu. Modely založené na Encoder/Decoder architektuře překonali dřívější pravidlové i statistické přístupy.

Překlad mezi jednotlivými jazyky se provede tak, že se první jazyk rozdělí na bazické části (dříve slova, nyní již celé věty) a tato věta je pomocí Encoderu zakódována do vnitřní, číselné reprezentace sémantiky věty. Následně se tato číselná reprezentace sémantiky dá Decoderu, který z ní vytvoří větu v druhém jazyku.



Nejpoužívanější architekturou pro Encoder/Decoder jsou rekurentní neuronové sítě a zejména biLSTM. Dokáží brát v potaz sekvenčnost textu, jinak řečeno pracují s informací jaká slova/věty jsou v blízkém okolí. V současnosti nejlepších výsledků dosahují modely používající Attention mechanismus, který vůbec nemusí používat RNN architekturu.

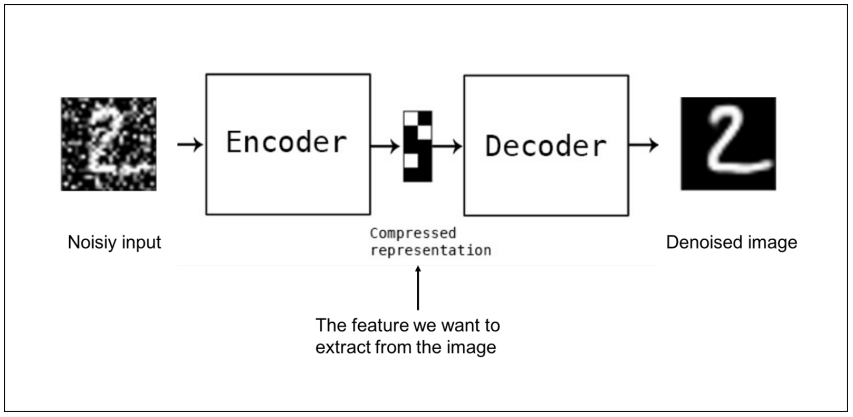


Klasické Encoder/Decoder architekury zakódují přes Encoder vstup do jednoho číselného vektoru, díky čemuž může lehce dojít k ztrátě (průměrování) víceznačnosti

slov/vět. A Decoder následně již nemá šanci zjistit který význam byl v původním textu zmíněn. Při použití Attention mechanismu se Decoder může podívat do stavu všech jednotlivých vrstev Encoderu a proto je schopen rozlišit jaký význam mělo původní slovo/věta. Neboť prakticky vidí celý proces výpočtu skryté reprezentace.

Odstraňování šumu

Pro odstranění šumu z obrázků lze použít encoder-decoder architekturu následujícím způsobem: pro dataset čistých obrázků bez šumu vygenerujeme zašumělé snímky. Architektura neuronové sítě má tvar přesýpacích hodin, dimenzionalita vstupní a výstupní vrstvy sedí na velikost obrázků. Síť dostává jako vstup zašumělé snímky a referenční výstup jsou odpovídající čistí. Zúžení architektury v prostředních vrstvách vynucuje, aby se síť při trénování naučila zachovávat podstatnější informace z obrázků. Při práci se snímky je také vhodné použít konvoluční/pooling vrstvy, ale odstraňování šumu lze použít i v jiných doménách než zpracování obrazu.

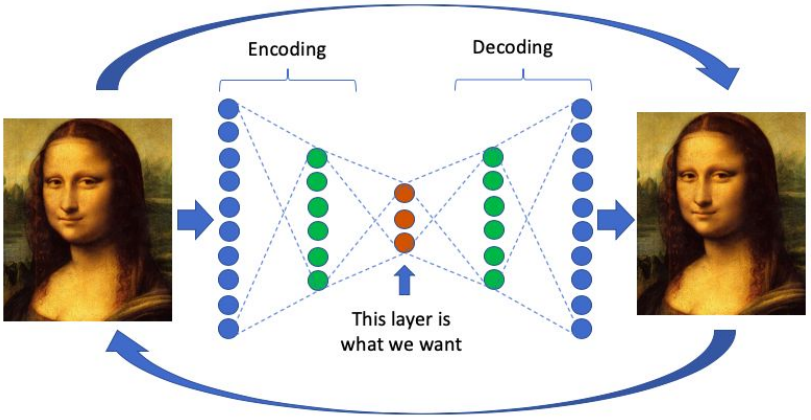


Na následujícím obrázku je vidět vstup a výstup natrénované neuronové sítě pro odstraňování šumu na MNIST datasetu:

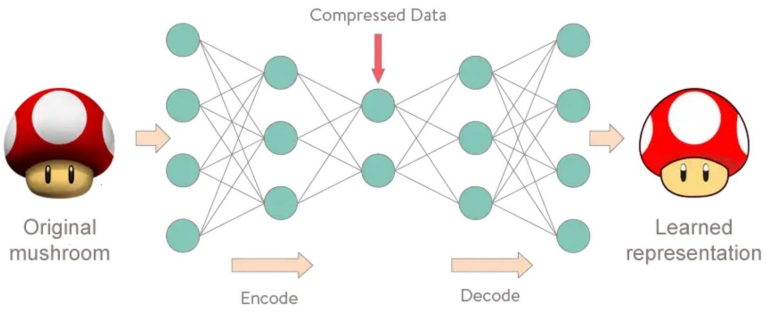
Autoencoders

Autoencoders jsou specifickou podtřídou Encoder-Decoder mechanismu neuronových sítí. Jejich největší výhodou je schopnost extrahovat důležité vlastnosti dat bez téměř žádných vnějších informací. Dokáží tak tedy úspěšně vzít např. obrázek 100x100 který obsahuje milion pixelů a (ztrátově) ho zakóduje do vektoru 100 čísel, které reprezentují pouze důležité vlastnosti obrázku. Tedy nejen, že extrahuje podstatné vlastnosti dat, ale také data komprimuje tím, že zachovává pouze to podstatné.

Princip fungování je v podstatě stejný, stejně jako u obecných encoder-decoder sítí. Prvně zakóduje vstup do vnitřní číselné reprezentace a poté ho zase zpět dekóduje. Zatímco u obecných encoder decoder je výstupem dekódovaná část, výstup autoencoderu je samotná vnitřní reprezentace, která je typicky mnohem menší dimenzionality než vstupní data.

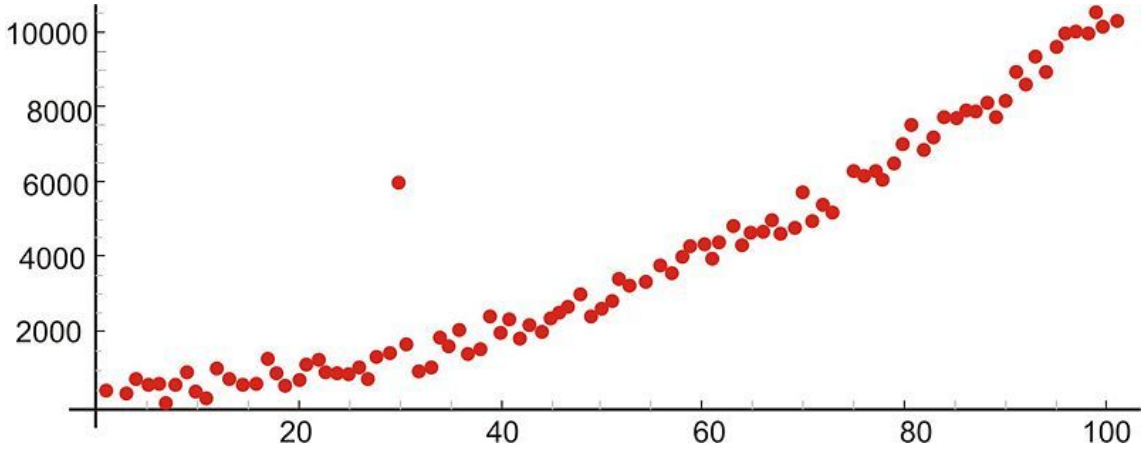


Kouzlo autoencoderu spočívá zejména v chybové funkci celého encoder-decoder modelu. Model se během tréninku snaží o to, aby byl výstup co nejvíce podobný vstupu. Tedy chybová funkce určuje, jak moc je výstup celého modelu stejný jako vstup. Skryté vrstvy musí tedy stále co nejlépe reprezentovat vstup i přesto, že mají mnohem menší dimensionalitu než vstupní data. Tato nejlepší reprezentace ve své podstatě odpovídá výběru podstatných vlastností vstupních dat.



Detekce outlierů

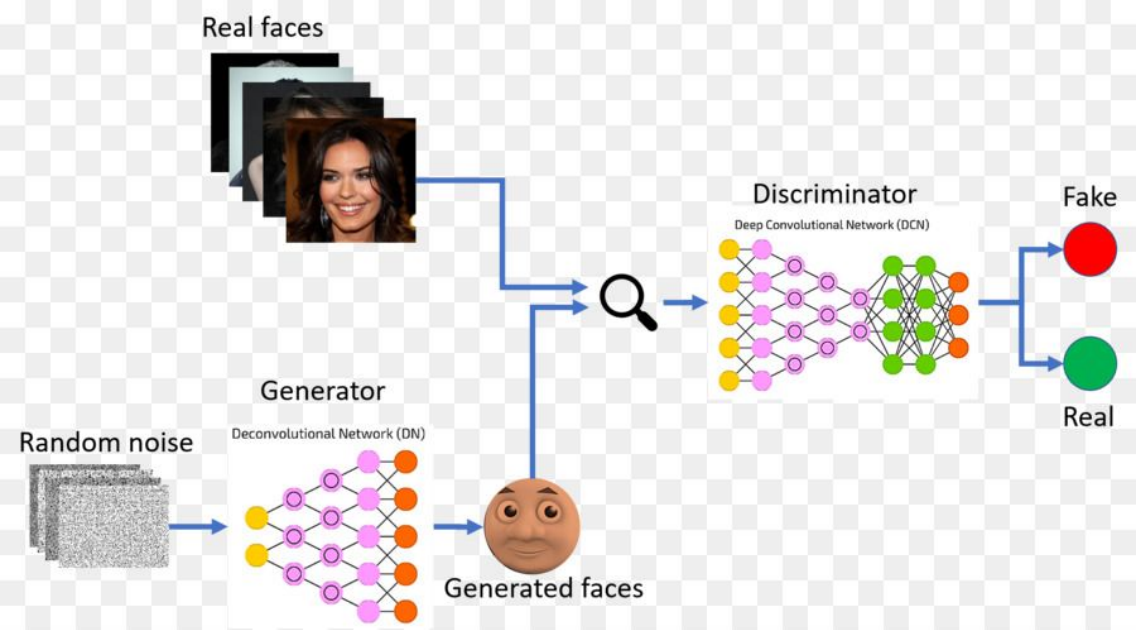
Detekce outlierů je ve své podstatě hledání zvláštních vzorků, vzorků s netypickými vlastnostmi. Outlierem může být například vzorek s chybným měřením, útok v provozu sítě, finanční podvod mezi transakcemi, poruchy v systému, nežádoucí účinky léku na pacienty atd. Existuje mnoho přístupů, jak anomální vzorky detekovat a nejlepší řešení záleží na doméně a konkrétním problému.



K detekci outlierů lze použít i architekturu autoencoderu a metoda spadá do unsupervised učení, tedy nepotřebuje označkované data. Fungování ilustruji na triviálním příkladu: data jsou fotky aut a vzácní outlieři jsou fotky lidí. Při trénování se encoder naučí “dobře” komprimovat fotky aut - ve smyslu reprezentovat tak, aby encoder zase správně vykonstruoval původní fotku, ale tato komprese je specifická pro fotky aut, kterých bylo v datasetu drtivá většina. Rekonstrukční chyba (chyba popisující rozdíl mezi vstupem a výstupem pro autoencoder) je nyní pro fotky aut malá a pro lidi velká. Pomocí velikosti rekonstrukční chyby lze tedy rozlišit outliery od běžných dat.

Generative Adversarial Networks (GANs)

Síť typu GAN je ve své podstatě kombinace dvou neuronových sítí, které spolu navzájem hrají hru. Jedna síť (generátor) generuje data a druhá (diskriminační) síť se snaží určit, zda tyto data jsou vygenerovaná, či opravdová (referenční). Během tohoto procesu se generativní síť zdokonaluje v generování čím dál tím více autentických dat, protože se učí, jak nejlépe oklamat diskriminativní síť. Díky této technice je možné dosáhnout velmi vysoké míry autenticity generovaných dat.



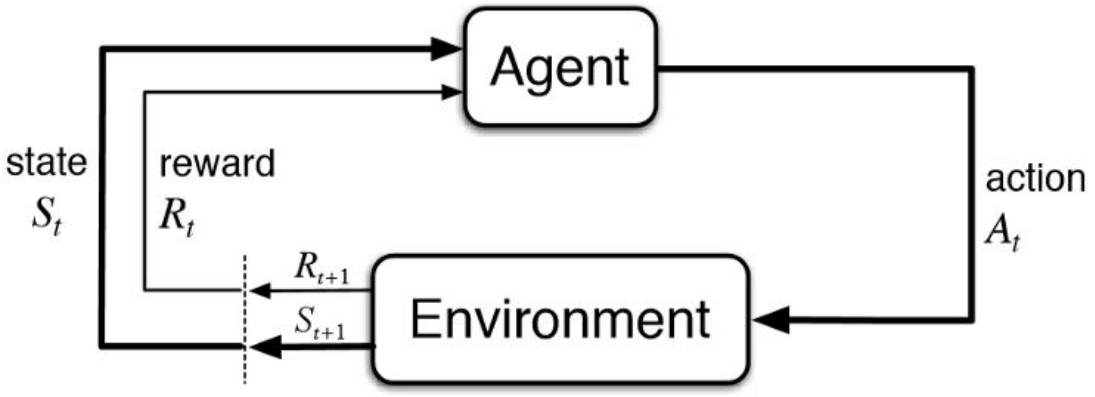
Deepfakes

Deepfakes jsou fotky nebo videa, ve kterých jsou postavy upraveny tak, aby vypadaly jako někdo jiný nahrazením obličejových rysů a v některých případech i hlasu. V praxi jsou zatím podvrhy většinou rozpoznatelné metodou “kouknu a vidím”, ale s větším množstvím trénovacích dat a lepšími algoritmy lze během několika let lze očekávat pro člověka nerozpoznatelné podvrhy. V současné době se používají hlavně pro zábavu (například nahrazení všech postav ve filmech Nicolasem Cagem) ale v brzké době hrozí i využití deepfakes pro tvorbu dezinformačních materiálů nebo nelegální podvody.

Po technické stránce deepfakes kombinují encoder-decoder architekturu a GAN. Encoder slouží k rozpoznávání charakteristických vlastností nahrazovaného obličeje (například náklon hlavy, mimika…). GAN trénuje generátor - v tomto případě decoder na generování “podvodného” obličeje na základě featur extrahovaných encoderem. Diskriminační síť se snaží rozpoznat, zda jde o podvrh nebo ne. Samotná aplikace je dopředné vyhodnocení encoderu a decoderu.

Neuronové sítě v reinforcement learning

Reinforcement learning (překládáno jako zpětnovazební učení nebo posílené učení) je druh strojového učení, ve kterém reprezentujeme prostředí (“svět”) pomocí stavů a agenta (nebo agenty), který podle stavu světa může vykonat akce a tím stav světa ovlivnit. Jako zpětnou vazbu poskytuje prostředí agentovi tzv. pozorovatelné vlastnosti světa a odměny (typicky reálné číslo). Účelem agenta je maximalizovat očekávaný zisk z odměn v rámci zvoleného časového rámce, který odhaduje pomocí dedukce a zkušenosti na základě informací, které mu prostředí poskytuje. Dedukce se často modeluje pomocí logických systémů a zkušenost pomocí strojového učení.

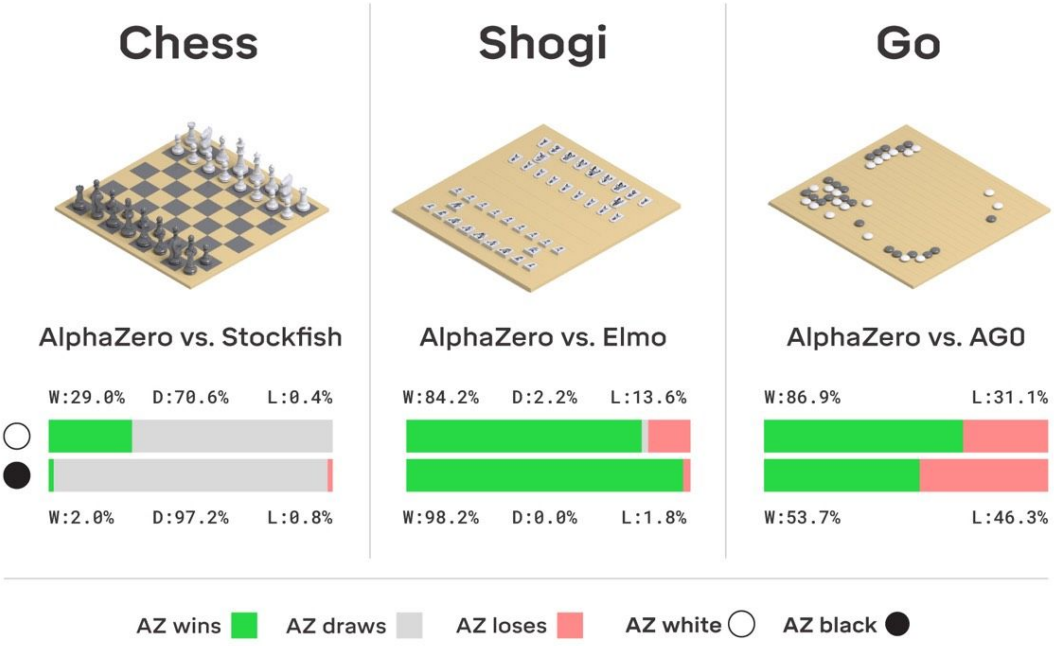


Základním principem učení agenta je aproximace ohodnocení stavů světa, kde hodnota stavu A reprezentuje očekávaný zisk odměn, které dostane, pokud se bude chovat podle svojí strategie počínaje ve stavu A. Optimální strategie je potom taková, která každému stavu přiřazuje akci, pro kterou se agent dostane do sousedního stavu s nejvyšším ohodnocením (lze zobecnit i na nedeterministické/stochastické prostředí).

Pro aproximaci ohodnocení stavů (Q-learning) lze použít regresi pomocí neuronové sítě, která jako vstup bere vektor pozorovatelných (a odvoditelných) proměnných a její výstup je očekávaná hodnota stavu. Ovšem při trénování nejde o standardní supervised učení, protože nemáme k dispozici skutečné hotnoty stavů, pouze zpětnou vazbu ve formě odměn na přechodech mezi stavy.

AlphaZero

AlphaZero je herní algoritmus vyvinutý společností Google DeepMind určený pro deskové hry šachy a go (a shogi). AlphaZero vznikl generalizací algoritmu AlphaGo, specializovaným na hru go. Oba algoritmy využívají reinforcement learning, při kterém program hraje sám proti sobě. AlphaZero neobsahuje žádnou databázi otevíracích strategií nebo historii lidských her, které by mohly urychlit učení, ale zároveň zanést bias.



AlphaStar

AlphaStar je obdobný program AlphaZero, ale jeho úkolem je hrát komplexní online strategickou hru StarCraft 2, které byly pro svou složitost a rozhodovací náročnost až donedávna pokládány za doménu pouze lidí. AlphaStar se také učí hraním proti sobě a v současné době dosáhla úrovně, na které porazí v turnajích průměrně 99.8% hráčů, přestože v sobě obsahuje omezení na rychlost a frekvenci svých reakcí, aby nevyhrávala proti ostatním hráčům pomocí nadlidských reflexů a její vstup jsou pouze informace, které vidí i každý hráč.

Zdroje

https://iq.opengenus.org/types-of-autoencoder/

https://blogs.oracle.com/datascience/fraud-detection-using-autoencoders-in-keras-with-a-tensorflow-backend

https://www.ck12.org/book/Probability-and-Statistics---Advanced-(Second-Edition)/section/9.2/

https://www.svetandroida.cz/swiftkey-neural-alpha-klavesnice/

https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781789957211/20/ch20lvl1sec112/deep-learning-for-computer-vision

https://www.pngfly.com/png-972f1i/

https://towardsdatascience.com/anomaly-detection-with-autoencoder-b4cdce4866a6

https://github.com/tensorflow/nmt

https://medium.com/idealo-tech-blog/zoom-in-enhance-a-deep-learning-based-magnifying-glass-part-2-c021f98ebede

https://www.cnews.cz/alphazero/

https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html

https://mashable.com/2018/01/31/nicolas-cage-face-swapping-deepfakes/?europe=true

https://towardsdatascience.com/deep-learning-for-named-entity-recognition-2-implementing-the-state-of-the-art-bidirectional-lstm-4603491087f1

https://medium.com/dpa-newslab/8-lessons-learned-about-ner-f40b263490db

https://www.researchgate.net/publication/326038469_Deep_Learning_for_Improved_System_Remaining_Life_Prediction/figures?lo=1

https://towardsdatascience.com/basics-of-the-classic-cnn-a3dce1225add

https://github.com/matterport/Mask_RCNN

https://medium.com/udacity/computer-vision-nanodegree-program-what-youll-learn-668dbfc3e5a3

https://is.muni.cz/auth/el/fi/podzim2019/PV021/um/ns.pdf