VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

STROJOVÝ PŘEKLAD POMOCÍ UMĚLÝCH NEURONOVÝCH SÍTÍ

MACHINE TRANSLATION USING ARTIFICIAL NEURAL NETWORKS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **JONÁŠ HOLCNER**
AUTHOR

VEDOUCÍ PRÁCE **Ing. IGOR SZŐKE, Ph.D.**
SUPERVISOR

BRNO 2018

Abstrakt
Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém (slovenském) jazyce.

Abstract
Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Klíčová slova
Sem budou zapsána jednotlivá klíčová slova v českém (slovenském) jazyce, oddělená čárkami.

Keywords
machine translation, neural machine translation, neural networks, recurrent neural networks, LSTM, encoder, decoder, encoder-decoder model, sequence to sequence, seq2seq, keras, tensorflow, moses, bleu, attention, bi-directional encoder

Citace
HOLCNER, Jonáš. *Strojový překlad pomocí umělých neuronových sítí.* Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szöke, Ph.D.

Strojový překlad pomocí umělých neuronových sítí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Igora Szökeho Další informace mi poskytli... Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

....
Jonáš Holcner
3. prosince 2017

Poděkování

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc (externí zadavatel, konzultant, apod.) napsat neco o metacentru?.

Obsah

Kapitola 1

Úvod

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Kapitola 2

Teorie

2.1 Jazykové modely

[[jazykove modely v kontextu machine translation, generovani jazyku]]

2.1.1 N-gram

2.1.2 log-linear

2.2 Neuronové sítě?

2.3 Rekurentní neuronové sítě

V této kapitole je popsán základní koncept rekurentních neuronových sítí (RNN[1]), jejich srovnání s běžnými neuronovými sítěmi a dále pak popis upravených variant RNN – LSTM 2.3.1 a GRU 2.3.2. Kapitola volně vychází z práce [5].

RNN (Elman [3]) jsou známé již přes dvě desítky let. Úspěšně jsou však používány až v posledních letech a to hlavně díky vyššímu výpočetnímu výkonu a většímu objemu trénovacích dat, který je v současné době dostupný a také, díky většímu výkonu, zpracovatelný. Tento druh neuronových sítí je obzvlášť vhodný například pro rozpoznávání psaného písma, rozpoznávání řeči, v kombinaci s konvolučními neuronovými sítěmi pro generování popisků obrázků a co je nejvíce zajímavé pro tuto práci, pro tvorbu jazykových modelů, generátorů textu a pro překlad.

Jejich hlavních výhodou oproti původním dopředným neuronovým sítím je jejich schopnost držet si vnitřní stav napříč časem. Dopředná neuronová síť pracuje vždy s aktuální hodnotou x na vstupu, pro kterou pomocí vah W získá výstup y (rovnice 2.1).

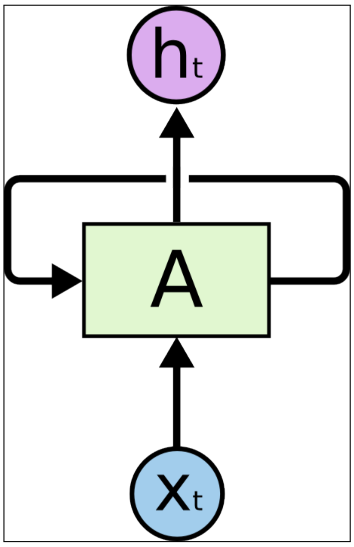
$$y = f(x, W) \tag{2.1}$$

Pokud pak takováto síť pracuje s nějakou sekvencí měnící se v čase, například se slovy v rámci jedné věty, pro každé slovo na vstupu x_t, kde t znázorňuje čas (pozici) slova ve větě, použije stejné váhy pro získání výstupu y_t a nezjistí ani nezachová žádnou úvahu o vzájemném vztahu těchto slov.

[1] z anglického recurrent neural network

RNN tento problém řeší zavedením vnitřního stavu h_t a smyčky (obrázek 2.1). Vstupem dalšího stavu je vždycky výstup ze stavu minulého. Pro každé x_t ze sekvence se tedy nyní může získat výstup y_t pomocí vnitřního stavu h_t z předchozího kroku t (rovnice 2.2. Přičemž počáteční stav h_0 je obvykle nastaven na nulu.

$$h_t = f(x_t, h_{t-1}) \tag{2.2}$$



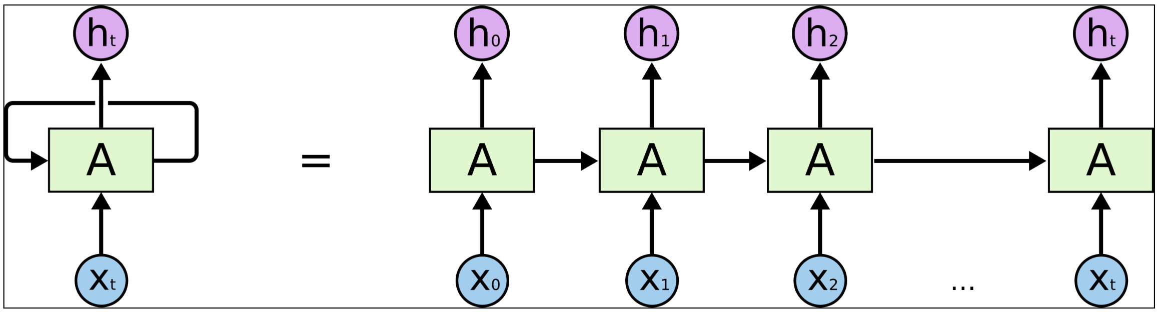
Obrázek 2.1: Recurrent Neural Networks have loops. **[[vlastni obrázek nebo citace `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`]]**

Funkce f z rovnice 2.2 je nelineární funkcí a nejčastěji se používá jedna z funkcí *sigmoid*, *tanh* nebo *relu* 2.2. **[[lepe popsat jednotlivé funkce a jejich výhody/nevýhody]]**

Obrázek 2.2: **[[vedle sebe obrazky funkci relu, tanh, sigmoid, idealne tri ruzny captions]]**

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{2.3}$$

[[doplnit rovnice a vysvětlení jak se to aplikuje dál, stejně tak obrázky s unrolled rnn a popisem toho jak zachovává nějakou informaci (třeba že podstatné jméno je mužské) skrze jednotlivé kroky (i když ne úplně přes vzdálené a tím se dostanu k long term dependencies)]]



Obrázek 2.3: A recurrent neural network and the unfolding in time of the computation involved in its forward computation. [[vlastni obrázek nebo citace http://colah.github.io/posts/2015-08-Understanding-LSTMs/]]

[[trénování rnn, back propagation through time, have difficulties (http://proceedings.mlr.press/v28/pascanu13.pdf) learning long term dependencies]] [[loss computing]] [[gradient computing]] [[související vanishing a exploding gradient problem popsáno v [2]]] [[predchazeni vanish/exploding - lstm a gru, ktery s tim nejak pocitaji. Regularization - vysvětlit co to je]]

[[zvlášť neuronky a to jakým způsobem se učí/optimlizují(gradient..) a pak až konkrétně rekurentní nebo rovnou rekurentní?]] [[deep, bi-directional]]

2.3.1 LSTM

LSTM (Long short term memory [4] je varianta RNN řešící problémy mizejícího/explodujícího gradientu a vzdálených závislostí **[[lepší překlad pro long term dependencies?]]**.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{2.4}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2.5}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{2.6}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{2.7}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{2.8}$$

2.3.2 GRU

2.3.3 Trénování

2.3.4 Global optimization methods viz wiki on RNN

2.4 Word embeddings

[[souvislost s jazykovymy modely]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque

Obrázek 2.4: One image. **[[Napsat pořádný titulek]]**

placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

[[subsection nebo itemize?]]

2.4.1 word2vec

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.4.2 glove

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris.

Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.4.3 fasttext

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.5 Modely seq2seq

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.5.1 Encoder-decoder architektura

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Obrázek 2.5: One image. [[**Napsat pořádný titulek**]]

Obrázek 2.6: One image. [[**Napsat pořádný titulek**]]

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.5.2 Attention/Pozornost

2.6 Frameworky

Chci popsat různé frameworky ze kterých jsem vybíral a vysvětlit proč jsem zvolil Keras. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

[[subsection nebo itemize?, mozna nejake obrazky]]

2.6.1 Tensorflow

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.6.2 Theano

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.6.3 CNTK

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.6.4 Keras

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Kapitola 3

Návrh systému (Praxe, nové myšlenky, které práce přináší)

Rozhodl jsem se. Vymyslel jsem. Rozvrhl jsem. Vypočítal jsem. Odvodil jsem. Zjednodušil jsem. Vylepšil jsem. Navrhl jsem. Zjistil jsem. Vyzkoumal jsem.

3.1 Neformální návrh systému

Vyberu a použiju nějký dataset, který obsahuje zarovnané věty se stejným významem ve dvou různých jazycích. Tyto věty za pomocí nějakého tokenizeru rozdělím na jednotlivé tokeny (slova/značky jako vykřičník) a to bude vstup pro překládací model.

Model bude sestavený z Embedding části, která převádí slova do vektorů s nějakým významem – tedy z toho může neuronová síť něco použít, narozdíl od toho kdyby slova reprezentoval jen index. Pro tento účel použiju přednaučené embeddings od Facebooku (fasttext). Dále v modelu Encoder, který se sestává z vrstvy/vrstev rekurentní neuronové sítě (LSTM). Tímto Encoderem projde celá sekvence embeddings a vznikne tak "thought vector", což je význam dané věty převedený do nějakého velkého vektoru (latent dimension?). Z tohoto vektoru/prostoru pak Dekodér, což je také vrstva/vrstvy LSTM postupně generuje překlad po jednotlivých slovech. Na vstup nejdříve dostane startovací značku a jeho vnitřní stav (memory cell) se inicializuje stavem encoderu. Po každém vygenerovaném slovu dostane toto slovo na vstup a takto generuje tak dlouho, dokud nevygeneruje značku konec sekvence. Výstup z dekoderu je vrstva softmax o velikosti slovníku jazyka do kterého se překládá. Generovaná slova se vyberou buď jednoduchým argmaxem, tedy vybere se vždycky slovo s největší vycházející pravděpodobností a nebo nějakou pokročilejší metodou, jako je beam search.

3.2 Baseline systém v Moses

[[Jak rozlišit návrh a realizaci?]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

3.3 Dataset/y

Jejich struktura, jak je zpracuji a použiji Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Obrázek 3.1: One image. [[**ukázka ze souborů různých jazyků z jednoho datasetu**]]

Kapitola 4

Implementace, experimenty, vyhodnocení

Naprogramoval jsem. Posbíral jsem data. Pustil jsem to. Výsledky jsou takové. Je to tak a tak rychlé.

4.0.1 Bucketing

A padding. Rozdělení sekvencí na skupiny podle délky, abych nepaddingoval zbytečně a tím neplýtval výkon.

4.1 skóre BLEU

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

[[**vysazet hezky vzorce**]]

Kapitola 5

Závěr

- Autor se ohlíží za tím, co udělal: „V práci je. Hlavní úspěchy jsou. Důležitými výsledky jsou. Podařilo se."
- Autor uvede nápady, které nestihl realizovat v podobě možností pokračování: „Ještě by šlo zkusit. Kdybych byl na začátku věděl, co vím teď, dělal bych."
- Autor (ve vlastním zájmu) rekapituluje, jak bylo naplněno zadání práce.

5.0.1 Plány do budoucna

- použití bidirectional první vrstvy encoderu, pro lepší zachování contextu [7] na místo použití obrácených vstupů
- použití wordpieces [7] místo celých slov pro lepší handling rare words
- přidat attention [1]
- přidat beam search [6], sehnat původní článek co přinesl beam search

Literatura

[1] Bahdanau, D.; Cho, K.; Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, ročník abs/1409.0473, 2014, 1409.0473.
URL http://arxiv.org/abs/1409.0473

[2] Bengio, Y.; Simard, P.; Frasconi, P.: Learning Long-term Dependencies with Gradient Descent is Difficult. *Trans. Neur. Netw.*, ročník 5, č. 2, Březen 1994: s. 157–166, ISSN 1045-9227, doi:10.1109/72.279181.
URL http://dx.doi.org/10.1109/72.279181

[3] Elman, J. L.: Finding Structure in Time. *Cognitive Science*, ročník 14, č. 2, 1990: s. 179–211, ISSN 1551-6709, doi:10.1207/s15516709cog1402_1.
URL http://dx.doi.org/10.1207/s15516709cog1402_1

[4] Hochreiter, S.; Schmidhuber, J.: Long Short-Term Memory. *Neural Comput.*, ročník 9, č. 8, Listopad 1997: s. 1735–1780, ISSN 0899-7667, doi:10.1162/neco.1997.9.8.1735.
URL http://dx.doi.org/10.1162/neco.1997.9.8.1735

[5] Luong, M.-T.: *NEURAL MACHINE TRANSLATION*. Dizertační práce, STANFORD UNIVERSITY, 2016.
URL https://github.com/lmthang/thesis

[6] Neubig, G.: Neural Machine Translation and Sequence-to-sequence Models: A Tutorial. *CoRR*, ročník abs/1703.01619, 2017, 1703.01619.
URL http://arxiv.org/abs/1703.01619

[7] Wu, Y.; Schuster, M.; Chen, Z.; aj.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, ročník abs/1609.08144, 2016, 1609.08144.
URL http://arxiv.org/abs/1609.08144

Příloha A

Poznámky

A.1 TODOs

http://www.statmt.org/moses/?n=Moses.Releases A.5

data bych doporucil open subtitles http://opus.lingfil.uu.se/

subtitles projet aligmentem.. oscorovat pary vet a pak vyfiltrovat malo pravdepodobne -> rozumna data
stahnuto, zkusil jsem projet moses => po 24 hodinach zaplnilo cely disk a nebylo hotovo. Filtrovani pomoci -score-options="-MinScore?

wmt vysledky http://www.statmt.org/wmt17/results.html ? BLEU

A.2 Pseudo zadani

chapu to tak, ze: nejdriv projedu kazde slovo na vstupu skrz prepripraveny nauceny veci od facebooku do embeddings a to pak teprve posilam do encoderu. Ten mi z toho pak vyhodi context (thought) vector
pouziju bidirectional LSTM (RNN) + attention + beam search
kazdej vstup (veta) je zarovnana (padding) na nejakou delku (bud se to doplni nebo naopak oreze), potom se z toho udela embeddings s pouzitim pretrained, potom se prozene encoderem, ziskam context, ten se prozedene dekoderem (s pomoci attention)
src input bude reversed sequence, protoze to podle nekterych clanku funguje lip, nevim jestli nestaci pouzit bi-directional LSTM

podle thesis je mozny pouzit hybridni model, generovat znama slova po slovech (ne po jednotlivych characterech) a neznama slova po jednotlivych znacich (takze vystupem muze byt o slovo mimo slovnik ze kteryho se sit ucila)

myslel jsem ze vystupem je embeding, ze kteryho se napr. podle vzdalenosti zjisti vysledne slovo, ale mozna je spis vystupem one hot encoding velikosti output slovniku a embeddings se pouzivaji jenom v prubeznych vrstvach

A.3 Kapitoly

- what is machine learning

- machine learning druhy (odvozovani ze znalosti, feature/representation, deep learning)
- popis ruznych neuronovych siti - CNN (images), structural?/standard NN (?), RNN(audio/translatio
LSTM (translation)?,
- hyperparameters
- popis ruznych zpusobu strojoveho prekladu textu
- popis frameworku? na neuronky
- Tensors
- activation functions
- Tools - moses
- frameworks - google tensorflow, microsoft CNTK, theano, keras (with usage of tensorflow/cntk/theano), tf.contrib.Keras + tf
- pretrained embeddings - facebook, word2vec, glove

A.4 Facebook pretrained word vectors

obsahuji textovou verzi - slovo a jeho vector a binarni verzi ve formatu fastText https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md
https://blog.manash.me/how-to-use-pre-trained-word-vectors-from-facebooks-fasttext-a71e6d55f27
https://www.quora.com/What-is-the-main-difference-between-word2vec-and-fastText

A.5 Moses

- UKAZKA JAK TO POUZIVA GOOGLE TUTORIAL https://github.com/tensorflow/nmt/blob/master/nmt/scripts/wmt16_en_de.sh
- statistical machine translation (SMT) or probably syntax based translation or factored translation
- data preparation
  - tokenisation: This means that spaces have to be inserted between (e.g.) words and punctuation.
  - truecasing: The initial words in each sentence are converted to their most probable casing. This helps reduce data sparsity.
  - cleaning: Long sentences and empty sentences are removed as they can cause problems with the training pipeline, and obviously mis-aligned sentences are removed.
- co zatim zkousim, rucne postupne jednotlive kroky

- mam data z http://opus.lingfil.uu.se/ pro cs (Czech)/en (English) pro moses, tzn tri soubory moses.cs-en.cs, moses.cs-en.en, moses.cs-en.ids
- tokenizace

```
~/mosesdecoder/scripts/tokenizer/tokenizer.perl -l en \
< /media/sf_DPbigFiles/OpenSubtitles2016-moses.cs-en.en \
> /media/sf_DPbigFiles/OpenSubtitles2016-moses.cs-en.tokenized.e
```

- nauceni truecaser modelu

```
~/mosesdecoder/scripts/recaser/train-truecaser.perl \
--model /media/sf_DPbigFiles/truecase-model.en \
--corpus /media/sf_DPbigFiles/OpenSubtitles2016-moses.cs-en.toke
```

- truecased

```
~/mosesdecoder/scripts/recaser/truecase.perl \
--model /media/sf_DPbigFiles/truecase-model.en \
< /media/sf_DPbigFiles/OpenSubtitles2016-moses.cs-en.tokenized.e
> /media/sf_DPbigFiles/OpenSubtitles2016-moses.cs-en.tokenized.t
```

- cleaning

```
~/mosesdecoder/scripts/training/clean-corpus-n.perl /media/sf_DP
```

- language model training

```
/media/sf_DPbigFiles/languagemodel$ ~/mosesdecoder/bin/lmplz -o
```

- binarizing for faster loading

```
~/mosesdecoder/bin/build_binary OpenSubtitles2016-moses.cs-en.ar
```

- training - ZKUSIT PUSTIT BEZ PARAMETRU rika to pak ty jednotlivy stepy co chci udelat

```
~/mosesdecoder/scripts/training/train-model.perl -root-dir . --c

with mgiza++
~/mosesdecoder/scripts/training/train-model.perl -root-dir train
and -reordering msd-bidirectional-fe -lm 0:3:$HOME/lm/news-comme
```

OpenSubtitles2016-moses.cs-en.cs/en jsou moc velky, zkusim vzit mensi cast (prvnich x radku) a natrenovat to s tim. Puvodni velka obsahuje 33896950 radku.

- pouziti EMS - Experiment Management System, ktery obdrzi konfiguracni soubor a resi si jednotlive kroky a skripty sam
- nainstaloval jsem xming, potreba pred spustenim skriptu "export DISPLAY=:0"
- /mosesdecoder/scripts/ems/experiment.perl -config config.toy -exec
- spadlo to na step EVALUATION:test:nist-bleu crashed step EVALUATION:test:nist-bleu-c crashed
- takze asi zustanu u rucne postupnych prikazu - vytvoren vlastni skrupt runAll.sh. bash -x ./runAll.sh http://www.statmt.org/moses/?n=Moses.Baseline

A.6 odkazy

https://en.wikipedia.org/wiki/Language_model

- RBMT https://en.wikipedia.org/wiki/Rule-based_machine_translation
- SMT https://en.wikipedia.org/wiki/Statistical_machine_translation
- nejaky dalsi...?
- neuronka (GNMT, Transfomer)

computing BLEU score in python using ntlk lib http://www.nltk.org/_modules/nltk/align/bleu.html
research at google - machine translation articles
https://research.google.com/pubs/MachineTranslation.html
https://en.wikipedia.org/wiki/Google_Neural_Machine_Translation

https://research.googleblog.com/2016/09/a-neural-network-for-machine.html

https://research.googleblog.com/2017/06/accelerating-deep-learning-research.html
https://research.googleblog.com/2017/07/building-your-own-neural-machine.html
https://research.googleblog.com/2017/04/introducing-tf-seq2seq-open-source.html

BUCKETING AND PADDING IN TENSOR FLOW https://www.tensorflow.org/tutorials/seq2seq#bucketing_and_padding
neural machine translation tutorial acl 2016 https://sites.google.com/site/acl16nmt/home
chat bot in keras https://github.com/saurabhmathur96/Neural-Chatbot
https://en.wikipedia.org/wiki/Language_model

A.7 knihovny nad tensorflow

- google.github.io/seq2seq - A general-purpose encoder-decoder framework for Tensorflow, pomoci konfiguraci, snadne vytvoreni komplexnich seq2seq modelu, pouzity pro clanek Massive Exploration of Neural Machine Translation Architectures. NENI UDRZOVANO z https://gitter.im/tensor2tensor/Lobby - google/seq2seq is not maintained. If you're just starting, read this first https://github.com/tensorflow/nmt and read papers, the one for this repo too.
- github.com/tensorflow/tensor2tensor- A library for generalized sequence to sequence models, pomoci konfiguraci (vyber ruznych modulu a moznost vytvoreni vlastni), asi relativne snadny vytvoreni ruznych (obecnych, nejen seq2seq) modelu. VYPADA TO ROZUMNE
- github.com/tensorflow/nmt - zatimco T2T uz je hlavne skladacka predpripravenych veci, tenhle tutorial ukazuje jak vyrobit v tensorflow od pocatku vlastni model

A.8 Neural Machine Translation (seq2seq) Tutorial

https://github.com/tensorflow/nmt podle https://github.com/lmthang/thesis, uzitecnej clanek

- nmt model can differ in terms of *directionality* uni/bi, *depth* single/multi layer, *type* vanilla RNN/LSTM/GRU
- In this tutorial, we consider as examples a deep multi-layer RNN which is unidirectional and uses LSTM as a recurrent unit.
- time-major format znamena ze prvni parameter je max_encoder_time a druhy batch-size, u batch-major je to naopak
- DECODER teda funguje tak, ze dostava 1. vysledek z encoderu, tim vi z ceho preklada a k tomu 2. nejdriv znak <s> pro zacatek dekodovani a v dalsich casovych stepech pak pri treningu ty spravne slova prekladu a pri pouziti pak ty slova co sam vygeneruje. Je to dobre popsany v https://github.com/tensorflow/nmt#inference--how-to-generate-translations
- ukazka z tutorialu v gitbashi spadne s encoding problemem, v cmd ne
- ukazka spadne na nedostatku pameti, je potreba zmenit parametry. (po kazde zmene radsi smazat slozku nmt_model).

  ```
  pro cmd
  python -m nmt.nmt ^
      --src=vi --tgt=en ^
      --vocab_prefix=nmt_data/vocab  ^
      --train_prefix=nmt_data/train ^
      --dev_prefix=nmt_data/tst2012  ^
      --test_prefix=nmt_data/tst2013 ^
      --out_dir=nmt_model ^
      --num_train_steps=1000 ^
      --steps_per_stats=100 ^
      --num_layers=2 ^
      --num_units=32 ^
      --batch_size=64 ^
      --dropout=0.2 ^
      --metrics=bleu
  ```

  vyzkousim jak to funguje

  ```
  python -m nmt.nmt ^
      --out_dir=nmt_model ^
      --inference_input_file=nmt_data/my_infer_file.vi ^
      --inference_output_file=nmt_model/output_infer
  ```

A.9 old tensorflow seq2seq tutorial

https://www.tensorflow.org/tutorials/seq2seq

```
old seq2seq tutorial G:\Dropbox\vecicky\python\tensorFlow\RNNtutorial\transl
python translate.py ^
  --data_dir=nmt_data --train_dir=train ^
  --from_vocab_size=100 --to_vocab_size=100 ^
  --from_train_data=nmt_data/OpenSubtitles2016-moses-10000.cs-en-tokenized.t
  --to_train_data=nmt_data/OpenSubtitles2016-moses-10000.cs-en-tokenized.tru
  --size=256 --batch_size=32 --num_layers=1

spadlo to zas na pameti, zkusim mensi size a tak

https://github.com/tensorflow/tensorflow/issues/11157
I suffered exactly the same problem.
I just passed the error by modifying 2 lines of the following seq2seq.py fil

file: Anaconda3\Lib\site-packages\tensorflow\contrib\legacy_seq2seq\python\o
848 #encoder_cell = copy.deepcopy(cell)
849 encoder_cell = core_rnn_cell.EmbeddingWrapper(
850 cell, #encoder_cell,

preklad
python translate.py --decode --data_dir=nmt_data --train_dir=train ^
    --from_vocab_size=100 --to_vocab_size=100
```

A.10 Tensor2Tensor

github.com/tensorflow/tensor2tensor

- po instalaci na windows nefunguji pripravene bin programy podle tutorialu (jako t2t-trainer). Je potreba stahnout si je z repositare a poustet rucne - python t2t-trainer

```
python bin/t2t-trainer.py ^
  --generate_data ^
  --data_dir=walktrough/t2t_data ^
  --problems=translate_enmk_setimes32k ^
  --model=transformer ^
  --hparams_set=transformer_base_single_gpu ^
  --output_dir=walktrough/t2t_train/base ^
  --hparams="batch_size=64"

  a stejne to spadne s
  InternalError (see above for traceback): Blas SGEMM launch failed : m=
  zmenil jsem flags.DEFINE_float("worker_gpu_memory_fraction", 0.35,
                      "Fraction of GPU memory to allocate.") v souboru knih
```

A.11 transformer, tensorFlow

- googls novel neural architecture - Transformer better than GNMT (google neural machine translation) https://research.googleblog.com/2017/08/transformer-novel-neural-network.html
- tensor2tensor https://github.com/tensorflow/tensor2tensor/
- t2t https://research.googleblog.com/2017/06/accelerating-deep-learning-research.html
- Train set for training, validation set for hyperparameters tuning, test set for testing how good

transformer - uses only attention and gets rid of recurrence and convolution. What exactly is and does recurrence and convolution (in context of neural networks)? Probably another thing that could be written in the theoretical part.

A.12 vypisky z deep learning book

- uvod a historie, jak se postupne menily a jaky byly ruzny druhy machine learning
- machine learning basics
  - klasifikace
  - klasifikace s chybejicimi vstupy
  - regrese
  - transription
  - MACHINE TRANSLATION
  - structured output
  - anomaly detection
  - Synthesis and sampling
  - Imputation of missing values
  - Denoising
  - Density estimationorprobability mass function estimation
- Task, Performance measure, Expericen..to co to ma delat, cim a jak se zmeri jak dobre to dela, cinnost na ktere se to nauci
- unsupervised - nema popisky a pocitac se snazi sam z dat urcit nejake zavery - treba clustering, supervised - ty maji label pro data v data setu, takze se nauci pro ktere x je jake y a pak se snazi odvodit y pro dalsi nahodne x, ktere se jim predhodi
- underfitting, overfitting
- train error (chybovost na trainovacim data setu) vs generalization error (chybovost na testovacim datasetu)
- regularization

- hyperparameters
- train vs test vs validation set
- regression - ziskavame nejakou hodnotu na zaklade parametru, klasifikace - rozrazujeme do presne danych trid

5.4 Estimators, Bias and Variance

A.13 coursera deeplearning

vypisky z https://www.coursera.org/learn/neural-networks-deep-learning

- RELU - rectified linear unit, nahrazuje sigmoid funkci, protoze se nad ni rychleji uci
- structured data - tabulky informaci, kazdej sloupec je jedna feature (age, bedooorms, price) X unstructured data - obrazky, hudba, text
- logistic regression - jaka je procentualni sance ze x na vstupu = nejake vystupni y (asi jenom jedno konkretni), je to binarni klasifikace. Na rozrazeni do 0-1 pouziva sigmoid funkci
- cost function pro logistickou regresi - prumer loss funkce nad celym trenovacim setem
- vektor v matici je jako jeden sloupec
- (VEKTORIZACE) nasobeni vektoru v maticich misto ve smycce je radove rychlejsi! (numpy.dot), SIMD instrukce (single instruction, multiple data)
- broadcasting (v pythonu) vstupni hodnotu (at uz matici nebo skalar) namnozi takovym zpusobem, aby sla pouzit v operaci s matici
- The main steps for building a Neural Network are:
  - Define the model structure (such as number of input features)
  - Initialize the model's parameters
  - Loop:
  - Calculate current loss (forward propagation)
  - Calculate current gradient (backward propagation)
  - Update parameters (gradient descent)
- $Z_n^{[l](v)}$ l - index of layer (hidden), v - index of training vector, n - index of node in the layer
- activation functions (sigmoid 0-1 better for output layer for binary classification, tanh -1 - 1 better for hidden layers, (leaky) RELU 0/1 even better)
- bias can be initialized to zero but weights must be initialized randomly because all the nodes inside layer would have the same weights and would be calculating the same numbers (they would be identical)

A.14 bridging the gap

vypisky z Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

- encoder (LSTM RNN) - transforms a source sentence into a list of vectors, one vector per input symbol; 8 layers
- decoder (LSTM RNN) - produces on symbol at a time from the vectors; 8 layers
- those two are connected through an attention module - feed forward network with one hidden layer
- attention - koukne se pro kazde slovo na jeho okoli a pri prekladu se rozhodne, ktera slovo s tim danym slovem nejvice souvisi a podle toho vybere spravny preklad
- residual connections enable to train much deeper networks
- neural network model weights can be quantizied to speed up some inference
- BLEU score metric
- pouziti wordpieces (vylepsuje handlig rare slov), coz umoznuje generovani novych slov jako pri pouziti modelu po jednotlivych pismenech, ale je to efektivnejsi jako pri pouziti celych slov
- Using wordpieces gives a good balance between the flexibility of single characters and the efficiency of full words for decoding, and also sidesteps the need for special treatment of unknown words.

A.15 deep learning thesis

vypisky z thesis(nmtTutorialBasedOnThis) https://github.com/lmthang/thesis

- Language modeling is an important concept in natural language processing to allow one to do word prediction, i.e., guessing which word will come next given a preceding context.
- it does so by predicting next words in a text given a history of previous words.
- word embeddings are used instead of one-hot representation for words (long vector, one value for each word in vocabulary, 0 meaning false and 1 meaning true). Word embeddings has the same meaning value but are much smaller matrices.

A.16 clanek sequence to sequence learning with nn

- normal deep neural network models are excellent on many task, but not on mapping sequence to sequence.
- use LSTM to map input sequence to thought vector (encoder) then decoder to map to target sequence

- reversing order of words in all source sentences improves LSTM's performance markedly because of many short term dependencies
- Despite their flexibility and power, DNNs can only be applied to problems whose inputs and targets can be sensibly encoded with vectors of fixed dimensionality.
- The goal of the LSTM is to estimate the conditional probability
- again USES DATASET English to French translation task from the WMT 14 dataset

A.17 clanek unsupervised machine translation using monolingual corpora only

- model that takes sentences from monolingual corpora in two different languages and maps them into the same latent space
- By learning to reconstruct in both languages from this shared feature space, the model effectively learns to translate without using any labeled data.
- TWO WIDELY USED DATASETS and two language pairs - zkusit zjistit ktery jsou widely used datasets a pouzit je taky
- the model has to be able to reconstruct a sentence in a given language from a noisy version of it, as in standard denoising auto-encoders
- The model also learns to reconstruct any source sentence given a noisy translation of the same sentence in the target domain, and vice versa.
- jak vyuziva decoder word embeddings corresponddujici k danemu jazyku?
- The encoder is a bidirectional-LSTM which returns a sequence of hidden states z. At each step, the decoder, which is an LSTM, takes the previous hidden state, the current word and a context vector given by a weighted sum over the encoder states.
- jak funguje naivni inicializace s unsporvised vytvorenym word by word prekladem?
- chapu to tak, ze preklad funguje nasledovne - vezme se source veta v source jazyce, prelozi se translation modelem M (ktery je, nevim co?) a vznikne tak ne uplne povedeny preklad. Protoze se to predtim ucilo autoencodovat z do stejneho jazyka na poskozenych vetach, tak je nasledovne mozne tento poskozeny preklad prohnat autoencoderem a tim dostat spravny preklad (protoze se to predtim ucilo z pozkozenych vet tvorit spravne vety).
- M je nazacatku unsupervised word-by-word translation model using the inferred dictionary
- pouzivaji WMT 14 English-French a WMT 16 English-German
- http://www.statmt.org/wmt14/translation-task.html

A.18 clanek Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation

- The encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable-length target sequence.
- The two networks are trained jointly to maximize the conditional probability of the target sequence given a source sequence.
- RNN Encoder–Decoder learns a continuous space representation of a phrase that preserves both the semantic and syntactic structure of the phrase.
- RNN is neural network with hidden state h and optional output y which takes variable length input x. $h_t = f(h_{t-1}, x_t)$
- After reading the end of the sequence (marked by an end-of-sequence symbol), the hidden state of the RNN is a summary c of the whole input sequence.
- baseline model - popsany co jak vybraly za data, WMT14 english-french, SMT system v moses s default settings

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

A.19 understanding LSTM networks and

- potrebuju pochopit co je vystup encoderu/LSTM, rozdil mezi hidden state a cell(memory) state a co presne vechno se pak z toho pouzije v decoderu viz clanek predtim
- urlhttps://www.quora.com/What-is-the-difference-between-states-and-outputs-in-LSTM
- zakladni RNN si neumi pamatovat veci pres delsi casovej usek (single tanh layer)
- LSTM je se reseni - STMs are explicitly designed to avoid the long-term dependency problem
- CELL STATE
  - The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.
  - The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.
  - Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.
  - The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means "let nothing through," while a value of one means "let everything through!" An LSTM has three of these gates, to protect and control the cell state.
  - FIRST forget values based on W_f, then get learn new values based on $W_i and W_C$ and finally get new cell state from it

- OUTPUT h_t is based on cell state and filtered and shifted to -1 and 1 values using another weight W_o
- pochopil jsem cell state jako stav, ke kteremu se dojde patrne jednim pruchodem/-prubehem/iteraci zkrz LSTM vrstvu (takze treba kdyz do toho poslu jednu sequenci), ve kterem se/jaky bunka ma na konci. tzn dostalo to nakou vetu a postupne si to z ni neco bralo a zapominalo a na konci to ma ve svym cell state
- zatimco hidden state je output (ktery v pripade encoderu nevim co je)
- kde jsou v LSTM vahy ktery se uci? patrne uvnitr tech jednotlivych gate a urcuji prave co si to prenasi mezi krokama sekvence INPUT/OUTPUT/FORGET gate. vaha pro forget layer, pro input gate layer a candidate values
- OUTPUT is The vector of outputs from all memory units is the output of the LSTM network.

A.20 The Unreasonable Effectiveness of Recurrent Neural Networks

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

- If training vanilla neural nets is optimization over functions, training recurrent nets is optimization over programs.
- it is known that RNNs are Turing-Complete in the sense that they can to simulate arbitrary programs (with proper weights).
- vanilla RNN (nebo rekneme spis obecne RNN), jeden krok je pronasobeni vah W s vnitrnim hidden stavem h, ten se updatuje kazdy krok pomoci nejake funkce (tanh, v lstm je tam zapominaci a ucici se gate..) a vystupem teda je nasobek W*h
- character-level language model: That is, we'll give the RNN a huge chunk of text and ask it to model the probability distribution of the next character in the sequence given a sequence of previous characters. This will then allow us to generate new text one character at a time.
- docela hezky popsana backpropagace v rnn u obrazku s prikladem "hello"
-

A.21 Language models

https://machinelearningmastery.com/statistical-language-modeling-and-neural-language-models/

- Language modeling is the task of assigning a probability to sentences in a language. [...] Besides assigning a probability to each sequence of words, the language models also assigns a probability for the likelihood of a given word (or a sequence of words) to follow a sequence of words (Page 105 Neural Network Methods in Natural Language Processing, 2017.)

- Language modeling is the art of determining the probability of a sequence of words. This is useful in a large variety of areas including speech recognition, optical character recognition, handwriting recognition, machine translation, and spelling correction
- The use of neural networks in language modeling is often called Neural Language Modeling, or NLM for short.
- Neural Language Models (NLM) address the n-gram data sparsity issue through parameterization of words as vectors (word embeddings) and using them as inputs to a neural network. The parameters are learned as part of the training process. Word embeddings obtained through NLMs exhibit the property whereby semantically close words are likewise close in the induced vector space.
- The neural network approach to language modeling can be described using the three following model properties, taken from "A Neural Probabilistic Language Model", 2003. Associate each word in the vocabulary with a distributed word feature vector. Express the joint probability function of word sequences in terms of the feature vectors of these words in the sequence. Learn simultaneously the word feature vector and the parameters of the probability function.

A.22 clanek Neural Machine Translation and Sequence-to-sequence Models:A Tutorial

- n gram models - The parameters of n gram models consist of probabilities of the next word given n 1 previous words
- has definition of training/development/test data (3.3)
- definition of perplexity
- ngram vs log-linear models - log linear models calculate the same probabilty of word given a context, but use feature vector. Feature function takes a context as input and gives feature vector as output - for example identity of a word is a vector that is why all words has unique id. One hot vectors are used
- dobrej popis LSTM a celkove RNN
- residual connections aby se vyhlo vanishing gradientu
- popis batchingu a vyhod ruznych velikosti
- muzu v kerasu posilat ruzny delky nebo musim mit vsechny stejne dlouhy s paddingem? to je prece plytvani! mohl bych je rozdelit na ruzny velikosti a volat fit takhle s ruznyma, abych to optimalizoval DNESKA- BUCKETING v tensorflow
- dobrej popis encoder-decoder, pouzil bych vzdycky odkaz na puvodni zdroj a podle tohodle pak psal, protoze je to hezky pochopitelny
- pouziti beam search misto argmax, ruzny encodovani a ensembling?
- vyzkouset nebo popsat proc reverse vstupu nebo zkusit misto toho BI-DIRECTIONAL encoder

- takze uz mam do budoucna bidirectional encoder, beam search, bucketing, rare (unknown/oov words)
- popsat ruzny optimizery adam/gradient..

A.23 TENSORFLOW

- NVIDIA GTX 760, CUDA support + cudnn
- tensorboard
- it is often needed to reduce batch size/unit count, because otherwise there is an resourceExhausted error (memory on GPU is too low?)

A.24 keras

https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html detailni rozbor keras blogu https://machinelearningmastery.com/define-encoder-decoder-sequence-sequence-model-neural-machine-translation-keras/

- pouziva preklad po znacich misto po slovech
- LSTM in keras and time distributed layer https://machinelearningmastery.com/timedistributed-layer-for-long-short-term-memory-networks-in-python/
- inference mode as opposed to learning mode where we insert into decoder start tag and the whole correct translation: 1) Encode the input sequence into state vectors. 2) Start with a target sequence of size 1 (just the start-of-sequence character). 3) Feed the state vectors and 1-char target sequence to the decoder to produce predictions for the next character. 4) Sample the next character using these predictions (we simply use argmax). 5) Append the sampled character to the target sequence 6) Repeat until we generate the end-of-sequence character or we hit the character limit.
- A Dense output layer is used to predict each character. This Dense is used to produce each character in the output sequence in a one-shot manner, rather than recursively, at least during training. This is because the entire target sequence required for input to the model is known during training. The Dense does not need to be wrapped in a TimeDistributed layer.
- we can plot the model to file using keras plot-model, graphviz must be installed (both through pip in python and in windows as binary)
- pro musi byt padding a fixed length? https://danijar.com/variable-sequence-lengths-in-tensorflow/, ale jak teda muze google prekladat libovolne dlouhy vety? rozdeli je na mensi?
- c - cell state, h - hidden state, vysvetleno lip v section A.19 https://machinelearningmastery.com/return-sequences-and-return-states-for-lstms-in-keras/
- https://stackoverflow.com/questions/44515336/how-do-i-show-both-training-loss-and-validation-loss-on-the-same-graph-in-tensor

- **WARNING:** in python set(chars) doesn't return the same set everytime, order isnt given in set!! must sort as a list, otherwise on python close, the loaded model weights wouldn't correspond correctly to it!!!!

- LSTM stateful https://stackoverflow.com/a/46331227, zaver - defaultni stateful false je v pohode a kazda sequence ma vlastni novej C state

- automatic early stopping based on val_loss https://stackoverflow.com/questions/43906048/keras-early-stopping

embeddings in keras https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/

- A word embedding is a class of approaches for representing words and documents using a dense vector representation.

- The position of a word within the vector space is learned from text and is based on the words that surround the word when it is used.

- The position of a word in the learned vector space is referred to as its embedding.

- The keras embedding layer requires that the input data be integer encoded, so that each word is represented by a unique integer. This data preparation step can be performed using the Tokenizer API also provided with Keras.

- similar to https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html

- embedding UNK a ZERO v https://chunml.github.io/ChunML.github.io/project/Sequence-To-Sequence/

- embeddings zobrazitelne v tensorboard pomoci tensorboard callbecku s embeddings freq a embeddings_metadata souboru s metadaty (teoreticky jde pouzit pretrained embedings soubor, ale asi bude praktictejsi vzit z neho jen pouzitej slovnik, kvuli performance). viz slozka machinelearning mastery s embedingsMetadata.txt

attention in keras https://github.com/philipperemy/keras-attention-mechanism

https://chunml.github.io/ChunML.github.io/project/Creating-Text-Generator-Using-Recurrent-Neural-Network/ https://chunml.github.io/ChunML.github.io/project/Sequence-To-Sequence/

-

A.25 numpy

indexing and slicing https://stackoverflow.com/questions/2725750/slicing-arrays-in-numpy-scipy

A.26 leany a herout

- udelat comics verzi

- V textu používejte autorské "my", "já"použijte v úvodu a v závěru
- beran pise (a heroutovi se my taky nelibi) – nepouzivejte MY, "testy byly provedeny"namísto "my jsme provedli testy"
- takze proste v uvodu a zaberu subjektivni veci dam s ja, jinak ne. my taky ne a popisu to nejakym jinym zpusobem
- Úvod je úvod k textu diplomky, ne úvod do problematiky. to je az nasledujici teoreticka cast
- Názvy kapitol ať přesně a jednoznačně vystihují, co kapitola obsahuje. spatne-teorie, detekce, navrh reseni. Dobre- Detekce objektů příznakovými klasifikátory..
- pouzivat prvni osobu (mnoznou nebo jednotnou) jen kdyz pisu o necem co jsem udelal nebo se me tyka. Obecny veci a fakt je jestli to chapu lepsi psat jinak - Kdyz se podivame na vysledky (spatne) / Z vysledku vyplyva (dobre?)
- nepouzivat osloveni ctenare (vy..) - Podivejte se na obrazek (spatne), Obrazek ukazuje (dobre)
- Pište svou diplomku pro studenta, který má na vaše dílo navázat. http://www.herout.net/blog/2016/04/komu-se-pise-diplomka/
- obsah se musi vejit na jednu stranku
- struktura nadpisu by mela mit tri urovne, ctvrta uroven je vetsinou spatne
- lověk, který se v oboru aspoň letmo orientuje, přesně poznat, co se v práci nachází. Dokáže odhadnout, co je cílem práce. Ví, z jakých modulů se celé řešení skládá a k čemu tyto slouží. Řekne, kolik a jakých experimentů řešitel provedl. Dokáže říct, kdo je cílovým „zákazníkem“ práce – komu a k čemu je dobrá.
- pozor na pomlcky a spojovnik. pomlcka je misto carky a je dlouha (–), spojovnik je napr rikam-li, takze vetsinou chci dve carky za sebou
- kazda veta ma sloveso
- nezapomenout na uvody kapitol, kde se popisuje strucne a jasne o cem bude