VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

IMAGE CAPTIONING WITH RECURRENT NEURAL NETWORKS

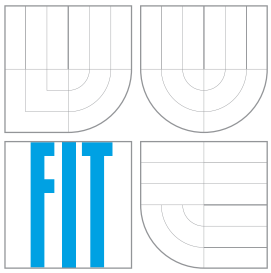
SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE
AUTHOR

Bc. JAKUB KVITA

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

POPIS FOTOGRAFIÍ POMOCÍ REKURENTNÍCH NEURONOVÝCH SÍTÍ

IMAGE CAPTIONING WITH RECURRENT NEURAL NETWORKS

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE
AUTHOR

Bc. JAKUB KVITA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL HRADIŠ, Ph.D.

BRNO 2015

Abstrakt

Výtah (abstrakt) práce v českém jazyce.

Abstract

Výtah (abstrakt) práce v anglickém jazyce.

Klíčová slova

Klíčová slova v českém jazyce.

Keywords

Klíčová slova v anglickém jazyce.

Citace

Jakub Kvita: Image Captioning with Recurrent Neural Networks, semestrální projekt, Brno, FIT VUT v Brně, 2015

Image Captioning with Recurrent Neural Networks

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením pana Michala Hradiše.

.....
Jakub Kvita
January 5, 2016

Poděkování

Zde je možné uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc.

Contents

Chapter 1

Introduction

Klasicky popis toho co se tady bude dit, jak je to dulezite, atd.

Chapter 2

Neural networks

General idea of artificial neural networks emerged after World War II. Perceptron, as a single neuron unit, was created in 1958 by Frank Rosenblatt [25], but became popular only after combination with the backpropagation algorithm [2, 30]. At that time neural nets have not reached massive popularity, not because they are not working, but due to small computing power of machines back then and lack of datasets. Recently (after 2000) neural nets became popular again, under the name of 'Deep Learning', to emphasize the use of several layers stacked on top of each other to create deep architectures, which are far more practical than shallow ones. During this reinvention, neural nets have been successfully applied in multiple fields like computer vision, speech recognition and natural language modeling.

reference, CV,rec,LM

Various useful architectures and algorithms are now introduced almost every month. In this chapter, I will describe only a handful – recurrent neural networks with the LSTM and GRU units, and basics of convolutional neural nets.

2.1 Recurrent neural nets

Feedforward neural nets are extremely powerful models, which can be highly parallelized. Despite that, they can be only applied to problems with inputs and outputs, which have fixed dimensionality (e.g. one-hot encoding vectors). This is a serious drawback, as many of the real-world problems are defined as sequences with lengths that are unknown to us in beforehand. Soon recurrent neural networks were introduced and they proved to be very useful to this kind of task. There is vast amount of recurrent neural networks, many not suitable for sequential tasks like Hopfield network, which are very successful in specific tasks, but nevertheless not useful for us now.

Apart from classification, which can be more precise when using sequences, one of the most important tasks is next value prediction. This core task can be then extended very simply to predict arbitrary number of future values. Prediction problems are all around us, from the weather forecast and stock market prediction to the autocomplete in smartphones or web browsers.

We can understand recurrent neural networks as very deep forward nets with shared weights. It is called RNN unrolling and it is described in figure 1. Layers of this very deep net spread in time, together with the input sequence. This is very innovative idea, which enabled training RNN with backpropagation through time. It also shows that, as very deep networks, they have vanishing or exploding gradient problem, which means that

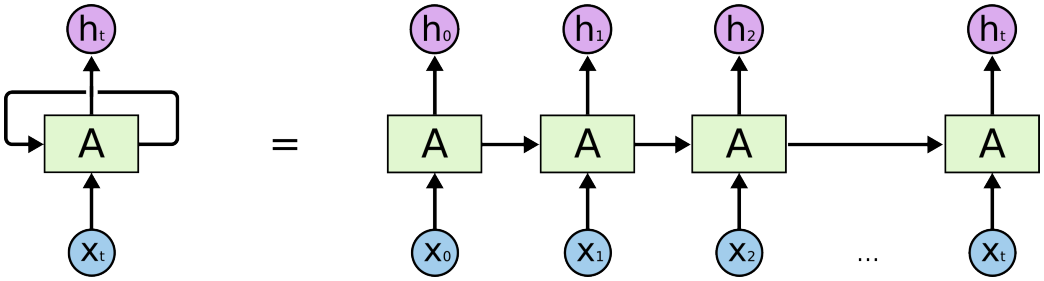


Figure 1: Unrolling of the recurrent neural net. [23]

the network is not able to learn long-term dependencies, even though in theory it should. This is a serious issue, which is caused by iterating many times over the weights and the activation function with derivatives > 1 (exploding gradient) or < 1 (vanishing gradient). Gradient then dies out and learning stops for distant dependencies. Among others this problem has been solved by the LSTM unit described in part 2.1.1, which is most popular now and following research resulting in GRU described in part 2.1.2.

2.1.1 LSTM – Long Short-Term Memory

Long Short-Term Memory nets are special kind of recurrent network, capable of learning long-term dependencies. This architecture was introduced by Hochreiter & Schmidhuber (1997) [13] after prior research of vanishing gradient problem [12]. Later architecture was refined and popularized by other researchers [9, 10] and nowadays LSTM is most popular RNN architecture used.

The LSTM unit was designed to remember a value for an arbitrary length of time. It contains gates that determine when the input is significant enough to remember, when it should keep or forget the value, and when it should output the value. To understand the flow of data, see the diagram of a simplified LSTM unit on the figure 2. All the gates can be described by the following series of equations.

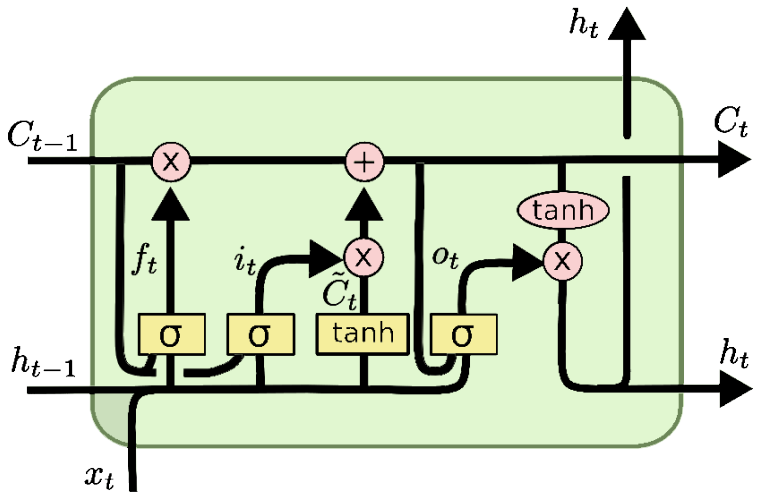


Figure 2: Variation of the LSTM unit. [23]

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) && (1)\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) && (2)\\
z_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) && (3)\\
c_t &= f_t \odot c_{t-1} + i_t \odot z_t && (4)\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) && (5)\\
h_t &= o_t \odot \tanh(c_t) && (6)
\end{aligned}
$$

$$
\sigma(x) = \frac{1}{1 + e^{-x}} \quad (7)
$$

In each time slice the unit is using current input x_t, last stored value c_{t-1} and unit output h_{t-1} to compute next state c_t and output h_t. Variables i_t, f_t, o_t denotes value of input, forget and output gates which are used to control the information flow.

LSTM based on these equations is using total of 11 weight matrices and 4 bias vectors for computations and sigmoid function σ defined in the equation (7) and the operation $\odot$ denotes the element-wise vector product. Equations described in this work are not the only way how to create an LSTM unit, but they will be used later while implementing the proposed model. Some of the versions are omitting 'peephole connections', which allows gates to look at stored value C_{t-1}, C_t or include only some of them.

Training of the LSTM based network can be performed effectively by standard methods like stochastic gradient descend in the form of backpropagation through time. Major problem with vanishing gradients during training described earlier is not an issue as backpropagated error is fed back to each of the gates.

2.1.2 GRU – Gated Recurrent Unit

Gated Recurrent Unit is slightly more dramatic variation on the LSTM theme from 2014 paper [4]. It combines hidden state of the unit h_t with the saved value C_t, merges input and forget gates into one update gate and removes peephole connections. These changes are simplifying standard LSTM models, but not at the expense of performance, and cause rapid growth in popularity. Diagram of the GRU unit is on the figure 3.

$$
\begin{aligned}
r_t &= \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) && (8)\\
z_t &= \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) && (9)\\
\widetilde{h}_t &= \tanh(W_{xh}x_t + W_{hh}(h_{t-1} \odot r_t) + b_h) && (10)\\
h_t &= (1 - z_t) \odot \widetilde{h}_t + z_t \odot h_{t-1} && (11)
\end{aligned}
$$

$$
\sigma(x) = \frac{1}{1 + e^{-x}} \quad (12)
$$

Equations describe a version of GRU unit used in this work, with sigmoid function σ defined in equation (12). The operation $\odot$ again denotes the element-wise vector product.

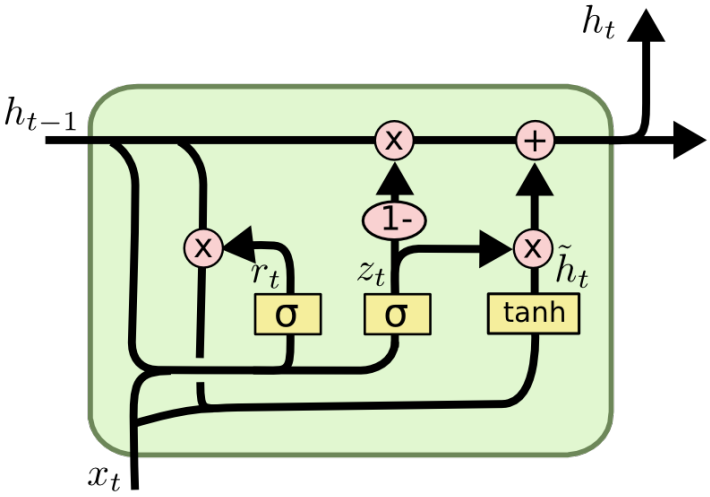


Figure 3: Variation of a GRU unit. [23]

While it is using only 4 weight matrices, 3 biases and just 1 state variable, researchers studied whether this can achieve at least same performance as previous LSTM unit.

Last year, study by Chung [5] was done, where different types of recurrent units were compared on the polyphonic music datasets. In this task LSTM and GRU were significantly better than all the other architectures, with GRU slightly in the lead. Generally, researchers agree that most of the LSTM variations, including GRU, are roughly on the same performance level. In [11] GRU is an average variation, slightly better than vanilla LSTM, with much simpler architecture.

In paper [16], which emphasized variety of tasks and the data, GRU outperformed LSTM unit on all tasks with the exception of language modeling. There are multiple approaches to model languages and in this work I will explore different type than the one mentioned in Jozefowicz's [16] paper. More will be explained in following chapters. Interestingly they also found that LSTM nearly matched the GRU's performance, when its forget gate bias was initialized to 1 and not to naive initialization around 0. It is also worth mentioning that Jozefowicz in his paper discovered several architectures similar to GRU, but with slightly better general performance. They were found by evolutionary algorithm working on candidate architectures represented by the computational graph.

2.1.3 Language modeling and word embeddings

With the addition of LSTM units, recurrent neural nets quickly showed good performance in many different types of sequence processing like speech recognition from sound waves, signal prediction and language modeling. These result were further improved when researchers started stacking LSTMs on top of each other like pancakes.

Text is represented by discrete values and is usually presented to network in form of input vectors with one-hot encoding[1]. If we have a task with K classes, class i will be represented by a vector V of length K. All the entries of V will be switched off to 0, except V_i, which will have the value of 1. Vector V is simultaneously a degenerated multinomial probability distribution of the current input. If the output has the same shape as input,

[1]One-hot encoded vector has exactly one high ('1') value and all the others low ('0').

pridat ukazky vektoru a pravdepodobnostniho rozlozeni

it can be simply created by softmax function at the output layer. Result will be proper multinomial distribution of next value, given current value.

At this point it is necessary to decide what will classes and defined vectors represent. In most cases, text prediction is performed at the word level. K is hence the number of words in the dictionary. This can cause some problems, as in bigger tasks dictionary often exceeds 100 000 records. This many classes require huge amount of training data to properly cover all the cases and high computational cost of the softmax layer is also an issue. This text representation cannot be used for texts not containing separate words, like multi-digit numbers. Nevertheless, state-of-the-art models have been using word-level representation. One of the advantages is no need to teach the net proper forms of the words. The net does not have to remember, how to spell the words properly and can learn other, more useful, features.

To solve the problem with extremely long input vectors, set of techniques called *word embedding* were developed. They map words from the vocabulary to suitable vectors of real numbers in high dimensional space (around 50–1000 dimensions). Chosen vectors cannot be random, they are meaningful in order of performing some following task. For example Skip-gram model from [22] mapped 783 millions words to vectors of 300 real numbers, while creating reasonable relationships between them.

embedding - vice rozebrat, klidne s ukazkama na celou stranku

Character level modeling has been considered and used as an alternative to word-level, but so far had slightly worse performance. Regardless, it is still considered as an option, because it has much simpler representation of input and output. Consider roughly 45 characters in English text and over 50000 words created from them. Character level network is also more suited for Czech or Russian and other fusional languages[2], which heavily use prefixes and suffixes to create new words. This is also an ability, which cannot be overlooked, as it is not available for word level. Character level models have usually smaller vocabulary size and have to be trained longer, as they need to learn spelling of the words on top of the same features of word level. With the properly trained character level model we can benefit from its much greater generative abilities, than we can achieve with word-level.

2.2 Convolutional neural nets

Feed-forward neural nets together with backpropagation algorithm showed very useful for range of tasks and it has even been proven [6, 15] they can approximate any continuous function. However, they were not very good in recognizing objects presented visually. As every unit is connected to large amount of units in the next layer (or all of them in fully-connected layers), the number of weights grows rapidly with the size of the problem and even more with the dimensionality. All these problems are manifesting even in image processing with only two dimensions. Convolutional neural nets (CNN) were introduced as a solution to reduce the number of parameters involved, while exploiting spatial constraints in the input.

Ideas of convolutional nets took inspiration from neurobiology, more precisely from the organisation of neurons in visual cortex of the cat. They were first used in the work of Homma [1] to process temporal signal. Later their design was improved by LeCun et al. [21] and one of the most famous convolutional nets – LeNet – was created. Basic architecture of CNN can be described as the following process:

[2]Fusional language is a type of language distinguished by its tendency to overlay many morphemes to denote grammatical, syntactic, or semantic change.

1. Convolve several small filters on the input image.
2. Subsample this space of filter activations.
3. Repeat steps 1 and 2 until your left with sufficiently high level features.
4. Use a standard a standard FFNN to solve a particular task, using the results features as input.

Ideas of convolutional nets took inspiration from neurobiology, more precisely from the organisation of neurons in visual cortex of the cat. They were first used in the work of Homma [1] to process temporal signal. Later their design was improved by LeCun et al. [21] and one of the most famous convolutional nets – LeNet – was created. Basic architecture of CNN can be described as the following process:

problemy FFNN s vizualnimi daty - zrod konvoluci podle inspirace v mozku, lenet, obrazek lenet, popis konvoluci, popis subsampling

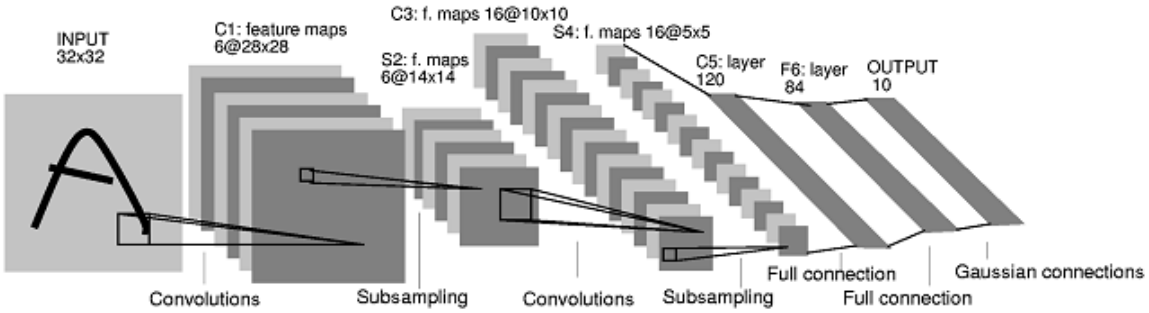


Figure 4: Architecture of LeNet. [21]

Kratky uvod do toho, kde se pouzivaji, popis jak funguji.

Neni potreba davat subsekce na vrstvy, staci popsat jak to funguje vsechno dohromady, jednotlive vrstvy ve vetach v jednom odstavci. Obrazek. V diplomce rozpracovat vic.

Chapter 3

Image caption generation

Scene understanding is one of the fundamental and most difficult tasks of computer vision. Being able to automatically generate image or video captions in regular text could have great effect. However, it is much more complicated than simple classification or object recognition, because the model also need to understand relations between the recognized objects and capture that correctly in the captions.

In this chapter I will do an overview of approaches to this task and more closely describe latest papers on which is this work based (section 3.1). Following parts cover datasets (3.2) and evaluation procedures (3.3) most commonly used for this task.

3.1 Related Work

Currently, neural networks are most heavily used to generate captions. Before them two main approaches were common. The first one used caption templates, which were filled by detected objects and relations. Second was based on retrieving similar captions from database and modifying them to fit current image. Question of similarity ranking has been addressed by many papers, which are based on the idea of joint embedding vector space for both images and captions [19]. Similar descriptions are in this space close to each other.

Both approaches above usually included generalization step to remove information relevant only to current image, for example names. They are quite successful in describing images, but they are heavily hand-designed and their text-generation power is fixed on the database/embedding and is not able to describe previously unseen compositions of objects. Over time these approaches fell out of favor to now dominant neural network methods.

Many of the methods using neural nets are inspired by successes in training of recurrent nets for machine translation. It is worth mentioning Sutskevers work [26], which studied general sequence to sequence mapping by converting input sequence to vector of fixed length. Vector is then decoded to output sequence. This encoder–decoder architecture is closely related to autoencoders and work of Kalchbrenner and Blunsom [17], who were first to map the entire input sequence to vector.

The introduced encoder–decoder architecture is important to the captioning task, because we can interpret image description problem as a translation from an image to a sentence. In this case, encoder part of the model is usually convolutional neural net, as they are excellent in the image classification [27]. Decoder part is similar as in machine translation models – type of a RNN or LSTM, as the output for both tasks is essentially same.

One of the most interesting event in this field is MS COCO Captioning Challenge[1] in which many of the state-of-the-art researchers compete directly against each other. Most of the works described further have participated in this challenge.

3.1.1 Show and Tell: A Neural Image Caption Generator

Clanek z Coco od Googlu.

[29]

3.1.2 Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Clanek z Coco z Montrealu/Toronta

[31]

3.1.3 From Captions to Visual Concepts and Back

Clanek z Coco od Microsoftu, mrknout se i na pokracovani v druhem clanku

[8]

3.1.4 Long-term Recurrent Convolutional Networks for Visual Recognition and Description

Clanek z Coco z berkeley

[7]

3.2 Datasets

Big datasets are necessary requirement in training recurrent neural nets, together with sufficient computing power. As access to machines and hardware suitable for training has been made extremely easy, obtaining enough data become the biggest problem. All the descriptions in the image captioning datasets have to be human generated, which is very expensive. This is one of the reasons, not many specialized datasets are created.

There are two main options how to get images and captions. First, using user-generated data from an online service, most commonly Flicker. However, captions are not made specifically for the task and could be prone to error. Second option is to create captions directly for use in the dataset. Amazon Mechanical Turk[2] is heavily used for this task. All datasets mentioned here are created this way.

Flickr8k [14] was one of the first datasets created for this purpose. It has been later expanded into Flickr30k [32]. MS COCO [3] is dataset created by Microsoft for their captioning challenge. CIDEr [28] datasets PASCAL-50S, ABSTRACT-50S are youngest mentioned, designed specifically for evaluation with the CIDEr metric.

[1] MS COCO Challenge - http://mscoco.org/dataset/#captions-challenge2015

[2] Amazon Mechanical Turk is crowdsourced Internet marketplace to perform tasks that computers are currently unable to do.

Table 1: Image captioning datasets.

Name	Images	Captions per image	Note
Flickr8k[3]	8 092	5	Focused on people or animals (mainly dogs) performing some specific action.
Flickr30k[4]	31 783	5-6	An extension of Flickr8k dataset.
MS COCO[5]	120 000	5	Images are divided - 80 000 for training and 40 000 for testing purposes.
PASCAL-50S[6]	1 000	50	Built upon images from the UIUC Pascal Sentence Dataset.
ABSTRACT-50S[7]	500	50	Built upon images from the Abstract Scenes Dataset. No photos.

3.3 Evaluation

Recent progress in fields like machine translation, which are very similar to image captioning, caused spike of interest in evaluating regular text output accuracy. Although it is sometimes not clear if a description of an image is best option available, some degree of assessment is possible. The best results can be obtained by asking live raters to give a score on the usefulness of each description. Subjective scores can vary, but it can be averaged by giving same description to multiple raters. However this method consumes tremendous amount of time and usually external raters are necessary. Tools like Amazon Mechanical Turk can be used to great extent, but need for automated tools is evident.

3.3.1 Automated metrics

Assuming that one has access to human generated captions, which is ground truth in our case, completely automated metrics exists. Even though all of them compute how alike are generated to human descriptions, different approaches are used. One metric can use several different settings with slight changes in the algorithm. This raises the question, how can we compare results of different works, despite using the 'same' evaluation method. Microsoft group of researchers addresses this issue in [3]. They created an evaluation server[8] which

[3]Flickr8k project page: http://nlp.cs.illinois.edu/HockenmaierGroup/8k-pictures.html
[4]Flickr30k project page: http://shannon.cs.illinois.edu/DenotationGraph/
[5]MS COCO project page: http://mscoco.org/dataset/
[6]PASCAL-50S and ABSTRACT-50S page: http://ramakrishnavedantam928.github.io/cider/
[7]See footnote 6.
[8]MS COCO evaluation server available through http://mscoco.org/dataset/#captions-upload.

have many automated metrics, with several configurations, including all mentioned here. It will serve as a reference point for comparing image captioning models.

The most commonly used metric has been BLEU (Bilingual Evaluation Understudy) [24], which was created in 2002 to evaluate quality of machine translated text from one language to another. Scores are computed on individual segments, usually sentences. BLEU has high correlation with human judgments and is still highly popular even for captioning tasks. However, it is becoming outdated as automatic methods are now outperforming humans. Four different variations of BLEU are used in MS COCO evaluation server.

METEOR (Metric for Evaluation of Translation with Explicit ORdering) [20] is another metric for the evaluation of machine translation from 2007. It was designed to fix some problems of the BLEU metric and it can also look for synonyms and perform stemming on input words.

Last year, metric designed directly to caption evaluation called CIDEr (Consensus-based Image Description Evaluation) [28] was introduced. This is still new metric, but with growing popularity as it correlate well with human judgment. Main idea of this metric is that given enough captions for the same image, metrics perform better. This can be seen in datasets introduced with it (see part 3.2).

Chapter 4

Experiments

Asi kapitola jen na semestralni projekt. V diplomce ji odstranim.

Jak se implementuje deep learning, jake knihovny se pouzivaji - Caffe, Theano, TensorFlow, Torch. Popsat ze Torch bude v tehle kapitole.

Budu popisovat veci co jsem zkousel implementovat v Torchi.

4.1 Torch

Torch se zrecykluje do diplomky.

Udelat tady tabulku o ruznych balicich co torch ma

Jak funguji rekurentni site v Torchi.

Nacitani modelu z Caffe, ukladani v Torchi...

4.1.1 nn, nngraph

Linky na knihovny.

4.1.2 rnn

4.1.3 Other packages

loadcaffe, optim,...

4.2 Predicting next character in sequence

Implementace sekce Language modeling, jak se to konkretne dela.

Jak jsem to udelal, co to dela, ukazky.

Karpathyho char-rnn

[18]

Chapter 5

Model

Do semestralniho projektu nebo az na diplomku?

Design modelu, co chci pouzit, jake metody chci zkusit.

Polozit si principialni otazku a zjistit jestli to nejak pomuze, jak to funguje.

5.1 Architecture

Architektura modelu, jake matematicke modely jsem pouzil, bez implementacnich detailu.

5.2 Training details

Popis pomoci jakeho algoritmu jsme trenovali, s jakyma parametrama, minibatches, datasety.

Chapter 6

Conclusion

Udelat jeden zaver pro semestralni projekt, pak ho prepsat pro diplomku.

Bibliography

[1] Les E. Atlas, Toshiteru Homma, and Robert J. Marks II. An Artificial Neural Network for Spatio-Temporal Bipolar Patterns: Application to Phoneme Classification. In *Neural Information Processing Systems*, pages 31–40. American Institute of Physics, 1988. 2.2

[2] Arthur Bryson and Yu-Chi Ho. *Applied Optimal Control: Optimization, Estimation and Control*. Halsted Press book'. Taylor & Francis, 1975. 2

[3] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server. *CoRR*, abs/1504.00325, 2015. 3.2, 3.3.1

[4] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *CoRR*, abs/1406.1078, 2014. 2.1.2

[5] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*, abs/1412.3555, 2014. 2.1.2

[6] G. Cybenko. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989. 2.2

[7] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. *CoRR*, abs/1411.4389, 2014. 3.1.4

[8] Hao Fang, Saurabh Gupta, Forrest N. Iandola, Rupesh K. Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. From Captions to Visual Concepts and Back. *CoRR*, abs/1411.4952, 2014. 3.1.3

[9] Felix A. Gers and Jürgen Schmidhuber. Recurrent Nets that Time and Count. In *IJCNN (3)*, pages 189–194, 2000. 2.1.1

[10] Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000. 2.1.1

[11] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. LSTM: A Search Space Odyssey. *CoRR*, abs/1503.04069, 2015. 2.1.2

[12] Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. *Master's thesis, Technische Universität München*, 1991. 2.1.1

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997. 2.1.1

[14] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013. 3.2

[15] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991. 2.2

[16] Rafal Józefowicz, Wojciech Zaremba, and Ilya Sutskever. An Empirical Exploration of Recurrent Network Architectures. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2342–2350, 2015. 2.1.2

[17] Nal Kalchbrenner and Phil Blunsom. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709. Association for Computational Linguistics, 2013. 3.1

[18] Andrej Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/. [Accessed: 2015-12-30]. 4.2

[19] Andrej Karpathy and Fei-Fei Li. Deep Visual-Semantic Alignments for Generating Image Descriptions. *CoRR*, abs/1412.2306, 2014. 3.1

[20] Alon Lavie and Abhaya Agarwal. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. 3.3.1

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. 2.2, 4

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013. 2.1.3

[23] Christopher Olah. Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 2015-12-20]. 1, 2, 3

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 3.3.1

[25] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. 2

[26] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. *CoRR*, abs/1409.3215, 2014. 3.1

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. *CoRR*, abs/1409.4842, 2014. 3.1

[28] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-Based Image Description Evaluation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 3.2, 3.3.1

[29] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and Tell: A Neural Image Caption Generator. *CoRR*, abs/1411.4555, 2014. 3.1.1

[30] P.J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University Press, 1974. 2

[31] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *CoRR*, abs/1502.03044, 2015. 3.1.2

[32] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 3.2