Neural Networks: Architectures and Applications for NLP

Session 01

Julia Kreutzer

25. Oktober 2016

Institut für Computerlinguistik, Heidelberg

Übersicht

1. Deep Learning

2. Neuronale Netze

3. Vom Perceptron zum Neuronalen Netz

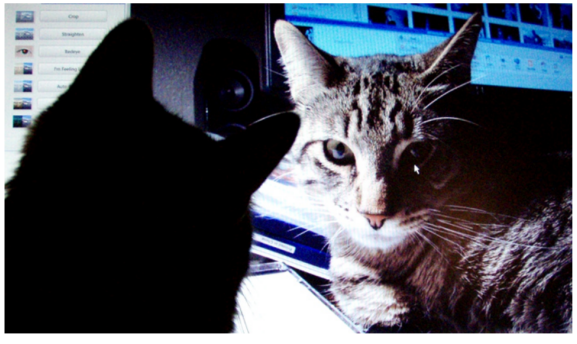
4. Ausblick

Deep Learning (DL)

Deep Learning is hot i

WIRED UK SCIENCE 06.26.12 11:15 AM

GOOGLE'S ARTIFICIAL BRAIN LEARNS TO FIND CAT VIDEOS



BY LIAT CLARK, *Wired UK*

When computer scientists at Google's mysterious X lab built a

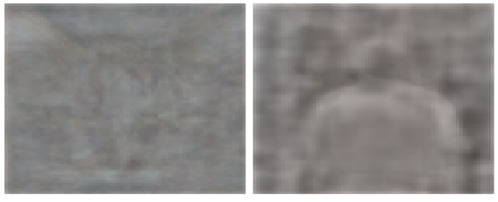


Figure 6. Visualization of the cat face neuron (left) and human body neuron (right).

Abbildung 1: Wired Article, [Le, 2013]

Deep Learning is hot ii

Abbildung 2: Google Research: ”Inceptionism”

Deep Learning is hot iii

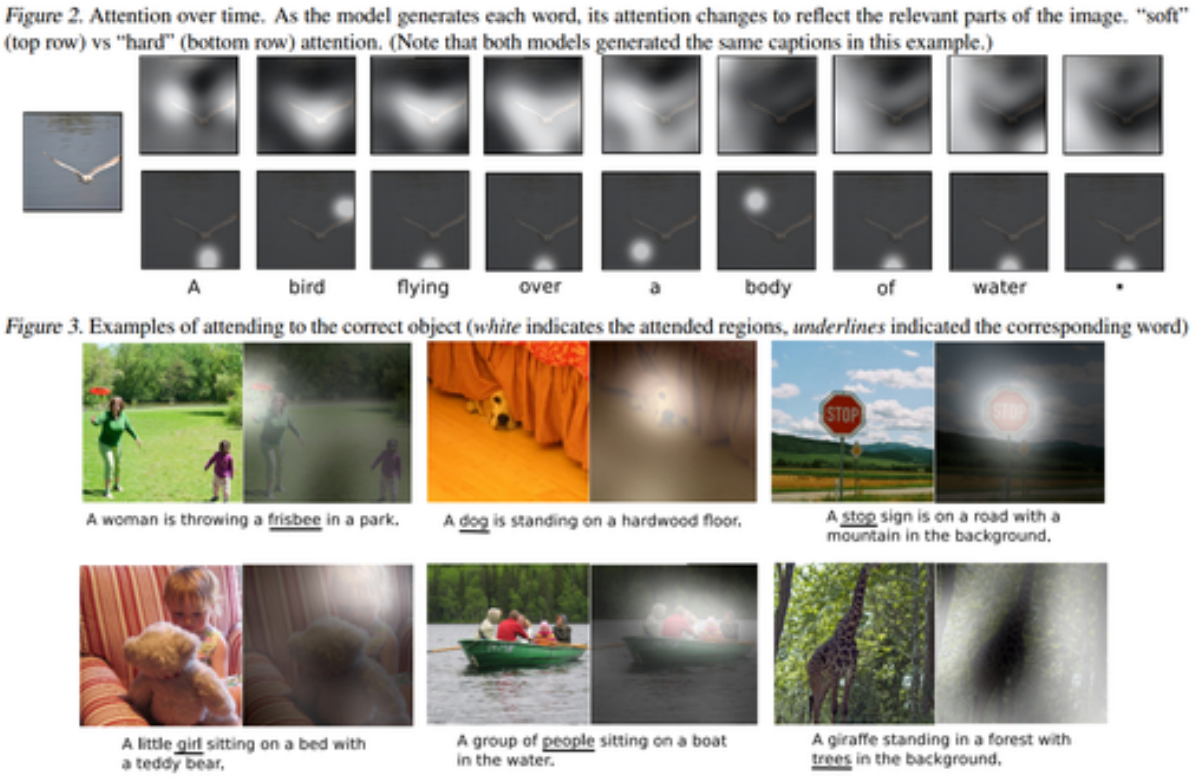


Abbildung 3: Attention for Caption Generation [Xu et al., 2015]

Deep Learning is hot iv

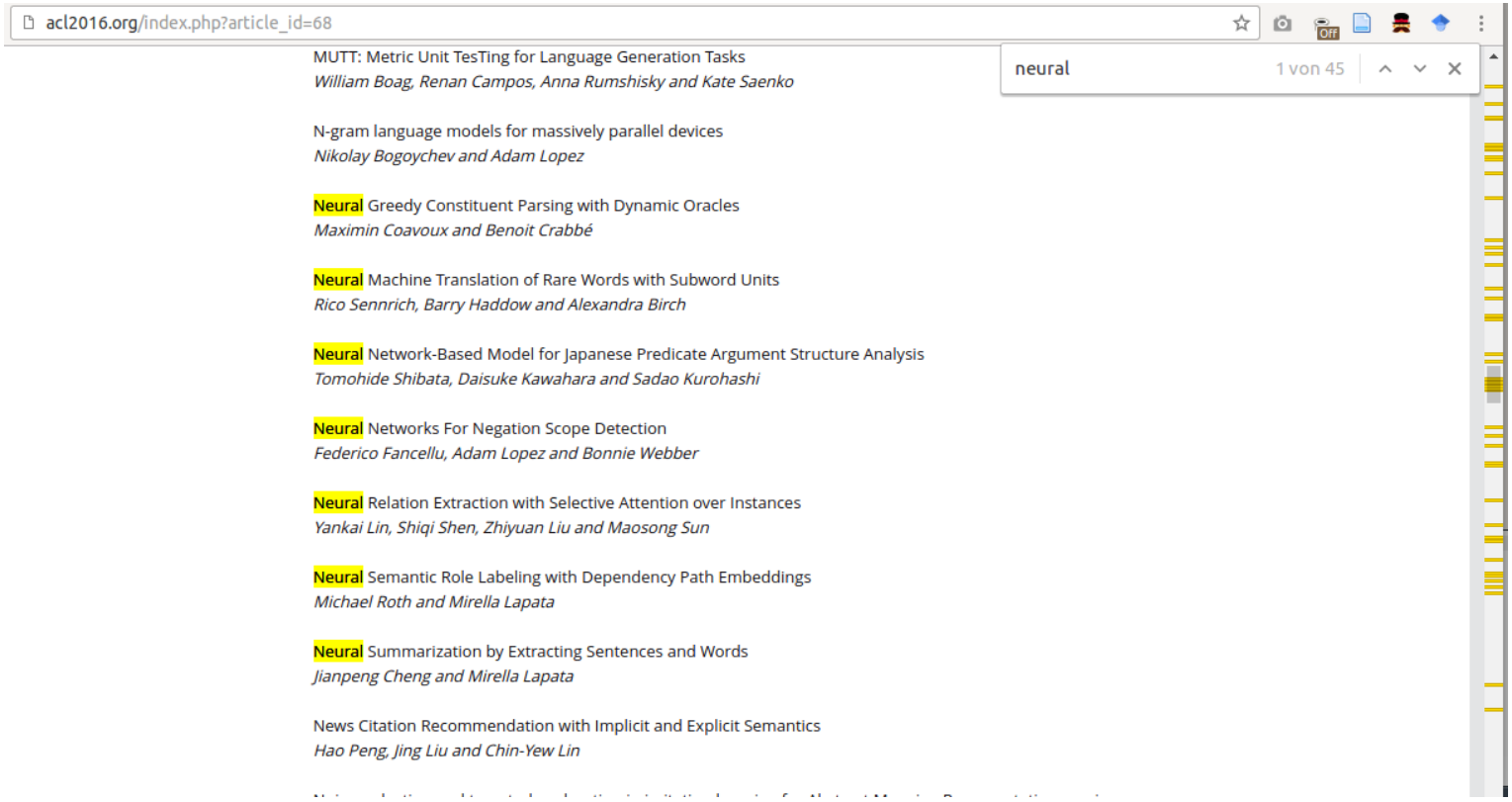


Abbildung 4: ACL'16 accepted papers

Was ist Deep Learning?

- Modelle: tiefe neuronale Netze
- end-to-end Training
- erlernen von Feature-Repräsentationen
- non-lineare Verarbeitung des Inputs

Neuronale Netze (NN)

Geschichte (kurz[1])

- **Perceptron** als Modell für ein einzelnes Neuron [Rosenblatt, 1958]

[1] Lang: [Schmidhuber, 2015]

Geschichte (kurz[1])

- **Perceptron** als Modell für ein einzelnes Neuron [Rosenblatt, 1958]
- **Multi-Layer Perceptron** (MLP) [Ivakhnenko and Lapa, 1966]

[1] Lang: [Schmidhuber, 2015]

Geschichte (kurz[1])

- **Perceptron** als Modell für ein einzelnes Neuron [Rosenblatt, 1958]
- **Multi-Layer Perceptron** (MLP) [Ivakhnenko and Lapa, 1966]
- effizientes Training mit **Backpropagation** [Rumelhart et al., 1986]

[1] Lang: [Schmidhuber, 2015]

Geschichte (kurz[1])

- **Perceptron** als Modell für ein einzelnes Neuron
  [Rosenblatt, 1958]
- **Multi-Layer Perceptron** (MLP) [Ivakhnenko and Lapa, 1966]
- effizientes Training mit **Backpropagation** [Rumelhart et al., 1986]
- **Pre-Training** für bessere lokale Optima
  [Ballard, 1987, Hinton and Salakhutdinov, 2006]

[1] Lang: [Schmidhuber, 2015]

Geschichte (kurz[1])

- **Perceptron** als Modell für ein einzelnes Neuron
  [Rosenblatt, 1958]
- **Multi-Layer Perceptron** (MLP) [Ivakhnenko and Lapa, 1966]
- effizientes Training mit **Backpropagation** [Rumelhart et al., 1986]
- **Pre-Training** für bessere lokale Optima
  [Ballard, 1987, Hinton and Salakhutdinov, 2006]
- **Recurrent Neural Networks** (RNN)
  [Elman, 1990, Williams and Zipser, 1995]

[1] Lang: [Schmidhuber, 2015]

Geschichte (kurz[1])

- **Perceptron** als Modell für ein einzelnes Neuron
  [Rosenblatt, 1958]
- **Multi-Layer Perceptron** (MLP) [Ivakhnenko and Lapa, 1966]
- effizientes Training mit **Backpropagation** [Rumelhart et al., 1986]
- **Pre-Training** für bessere lokale Optima
  [Ballard, 1987, Hinton and Salakhutdinov, 2006]
- **Recurrent Neural Networks** (RNN)
  [Elman, 1990, Williams and Zipser, 1995]
- Lösungen für **explodierende/schwindende Gradienten**
  [Pascanu et al., 2012, Hochreiter and Schmidhuber, 1997]

[1] Lang: [Schmidhuber, 2015]

NN für NLP

- Spracherkennung [Graves et al., 2013, Graves and Jaitly, 2014]
- Sprachmodellierung [Bengio et al., 2003, Schwenk, 2007]
- automatische Übersetzung [Son et al., 2012, Devlin et al., 2014, Sutskever et al., 2014, Sundermeyer et al., 2014]
- beliebige Klassifikationsprobleme, u.a.

  Syntactic Parsing [Chen and Manning, 2014]
  Semantic Role Labeling, POS-Tagging, Chunking [Collobert et al., 2011]
  Event Detection [Nguyen and Grishman, 2015]
  Sentiment Classification [Kalchbrenner et al., 2014]

Warum NN für NLP?

- strukturiertes Input
- kein mühsames Feature-Engineering, stattdessen Erlernen von Features
- Feature-Repräsentationen auf höheren Ebenen
- Ausnutzen von großen Textsammlungen ohne Annotationen

Vom Perceptron zum Neuronalen Netz

XOR-Problem i

Beispiel: Lerne XOR!

- Features: x_0, $x_1 \in \{0, 1\}$
- Gold Label:

$$y = \begin{cases} 0 & \text{if } x_0 == x_1 \\ 1 & \text{otherwise} \end{cases}$$

- Vorhersage: $\hat{y} = f_\theta(\mathbf{x})$
- Fehler: $L(f_\theta(\mathbf{x}), y) = (y - f_\theta(\mathbf{x}))^2$
- Minimiere empirisches Risiko auf Trainingsdaten $(\mathbf{x}, y) \in D$:

  Finde $\theta^* = \arg\min R_D(\theta)$; $R_D(\theta) = \frac{1}{|D|} \sum_{i=0}^{|D|} L(f_\theta(\mathbf{x}), y)$

XOR-Problem ii

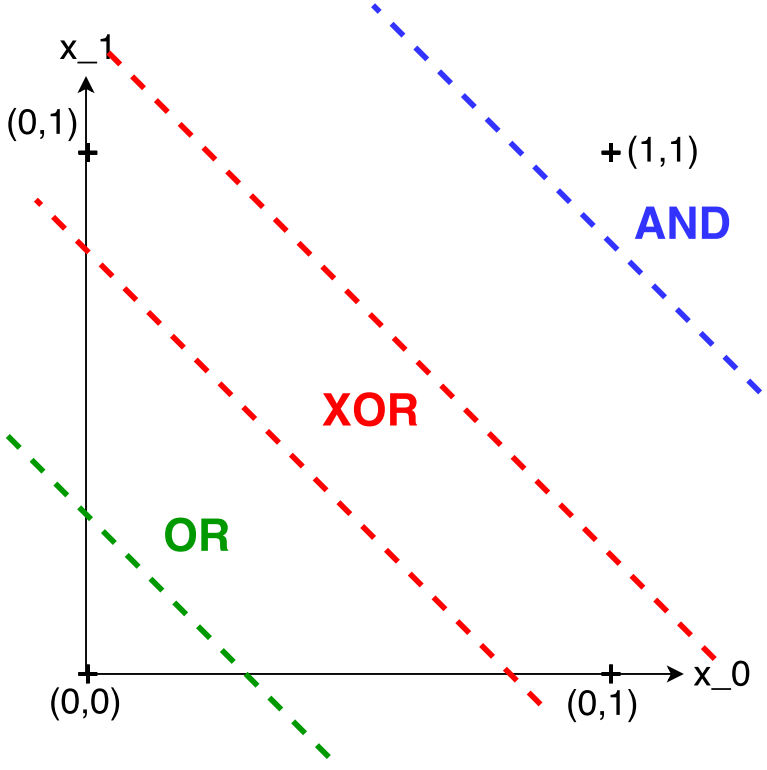


Abbildung 5: Das XOR-Problem

Recap: Perceptron

Einfaches Modell für ein einzelnes Neuron [Rosenblatt, 1958]

- lineare Kombination der Eingabewerte
- Schwellenwertfunktion entscheidet über Ausgabewert (Aktivierungsfunktion)
- Vorhersagen:

$$f_{\mathbf{w}}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

- Update-Regel:
  $\mathbf{w}_{t+1} = \mathbf{w}_t + (y - f_{\mathbf{w}_t}(\mathbf{x}))\mathbf{x}$

Perceptron – Schematisch

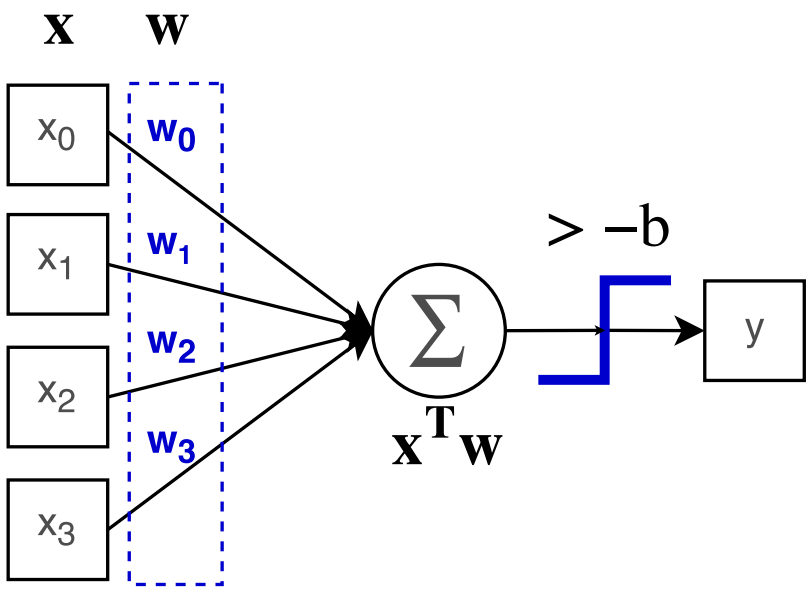


Abbildung 6: Perceptron

XOR: Perceptron

- TensorFlow Playground: ein Perceptron
- nicht linear separierbar!

Multi-Layer-Perceptron

Vernetzung mehrerer Neuronen in Schichten

- Input Layer: Netzeingabe
- Hidden Layer(s): verbundene Schichten von Neuronen, Parameter $\theta = \{\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2\}$
- Output Layer: Netzausgabe
- nicht-lineare, aber differenzierbare Aktivierungsfunktion σ, z.B. `tanh`, `sigmoid`, letzte Schicht meist `softmax`
- Vorhersagen für 2 Hidden Layers: $f_\theta(\mathbf{x}) = \sigma(\mathbf{W}_2 \cdot \sigma(\mathbf{W}_1 \cdot \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2)$

Multi-Layer-Perceptron – Schematisch

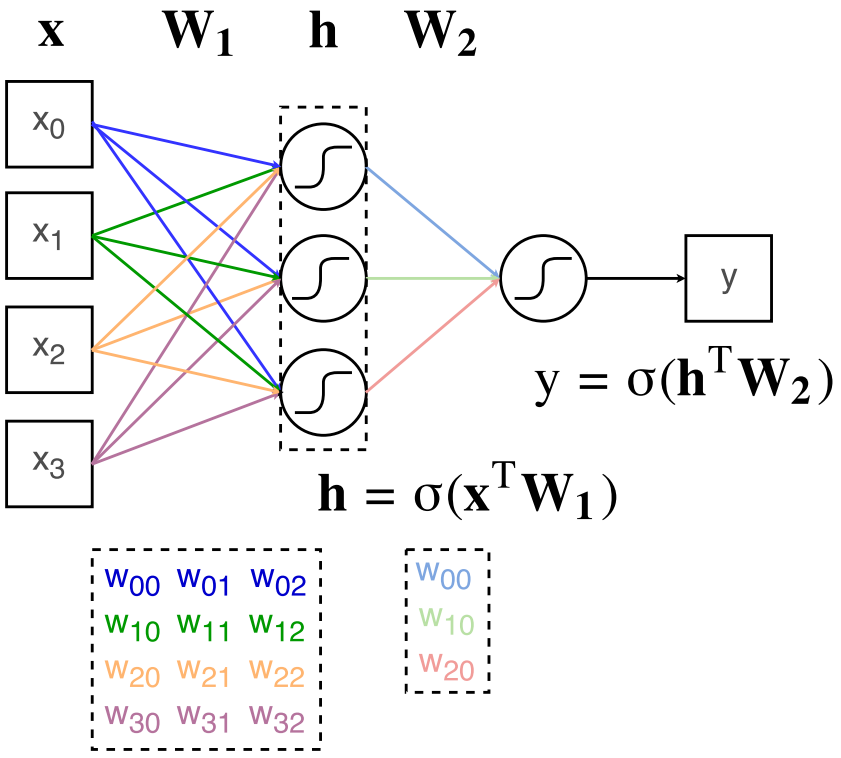


Abbildung 7: Multi-Layer-Perceptron mit 2 Hidden Layers

XOR: Multi-Layer-Perceptron

- TensorFlow Playground: mehrere Schichten
- MLP kann auch nicht-lineare Daten separieren
- *Universality Theorem* [Hornik, 1991]: MLP kann jede kontinuierliche Funktion beliebig genau approximieren, vorrausgesetzt es enthält genügend Neuronen, hier visuell erklärt
- aber: wie viele Schichten, wie viele Neuronen pro Schicht?

Herausforderungen

Probleme in der Praxis:

- Over-/Underfitting
- Hyperparametersuche
- Wahl der Fehlerfunktion
- non-konvexe Optimierung
- Initialisierung

Zusammenfassung & Ausblick

Zusammenfassung

Heute gelernt:

- Deep Learning: tiefe Neuronale Netze
- Neuronale Netze können viel,
- ... sind aber nicht so leicht zu trainieren

Nächstes Mal

In zwei Wochen:

- Backpropagation
- Gradientenberechnung
- Optimierung
- Regularisierung

References i


Ballard, D. H. (1987).
Modular learning in neural networks.
In *AAAI*, pages 279–284.


Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003).
A neural probabilistic language model.
The Journal of Machine Learning Research, 3:1137–1155.


Chen, D. and Manning, C. D. (2014).
A fast and accurate dependency parser using neural networks.
In *EMNLP*, pages 740–750.


Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011).
Natural language processing (almost) from scratch.
The Journal of Machine Learning Research, 12:2493–2537.

References ii


Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. M., and Makhoul, J. (2014).
Fast and Robust Neural Network Joint Models for Statistical Machine Translation.
In *ACL (1)*, pages 1370–1380. Citeseer.


Elman, J. L. (1990).
Finding structure in time.
Cognitive science, 14(2):179–211.


Graves, A. and Jaitly, N. (2014).
Towards end-to-end speech recognition with recurrent neural networks.
In *ICML*, volume 14, pages 1764–1772.

References iii


Graves, A., Mohamed, A.-r., and Hinton, G. (2013).
Speech recognition with deep recurrent neural networks.
In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.


Hinton, G. E. and Salakhutdinov, R. R. (2006).
Reducing the dimensionality of data with neural networks.
Science, 313(5786):504–507.


Hochreiter, S. and Schmidhuber, J. (1997).
Long short-term memory.
Neural computation, 9(8):1735–1780.


Hornik, K. (1991).
Approximation capabilities of multilayer feedforward networks.

Neural Networks, 4(2):251–257.

References iv


Ivakhnenko, A. G. and Lapa, V. G. (1966).
Cybernetic predicting devices.
Technical report, DTIC Document.


Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014).
A convolutional neural network for modelling sentences.
In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.


Le, Q. V. (2013).
Building high-level features using large scale unsupervised learning.
In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8595–8598. IEEE.

References v

Nguyen, T. H. and Grishman, R. (2015).
Event detection and domain adaptation with convolutional neural networks.
In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 2, pages 365–371.

Pascanu, R., Mikolov, T., and Bengio, Y. (2012).
Understanding the exploding gradient problem.
arXiv preprint arXiv:1211.5063.

Rosenblatt, F. (1958).
The perceptron: a probabilistic model for information storage and organization in the brain.
Psychological review, 65(6):386.

References vi


Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986).
Learning representations by back-propagating errors.
Nature, 323.


Schmidhuber, J. (2015).
Deep learning in neural networks: An overview.
Neural Networks, 61:85–117.


Schwenk, H. (2007).
Continuous space language models.
Computer Speech & Language, 21(3):492–518.


Son, L. H., Allauzen, A., and Yvon, F. (2012).
Continuous space translation models with neural networks.
In *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 39–48. Association for Computational Linguistics.


Sundermeyer, M., Alkhouli, T., Wuebker, J., and Ney, H. (2014).
Translation Modeling with Bidirectional Recurrent Neural Networks.
In *EMNLP*, pages 14–25.

References viii

Sutskever, I., Vinyals, O., and Le, Q. V. (2014).
Sequence to sequence learning with neural networks.
In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Williams, R. J. and Zipser, D. (1995).
Gradient-based learning algorithms for recurrent networks and their computational complexity.
Back-propagation: Theory, architectures and applications, pages 433–486.


Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015).
Show, attend and tell: Neural image caption generation with visual attention.
In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2048–2057.