Deep Learning and its Application in Natural Language Processing (DL&NLP)
DA345
Suggested reading materials

Soumitra Samanta

November 28, 2024

Contents

Lecture 1

Motivation, Course overview, Syllabus, Prerequisites and Resources

1.1 Class schedule:

- Tuesday: 03:00AM - 05:00 PM (IH402)
- Thursday: 12:30 - 02:30 PM (IH402)

1.2 Teaching Assistant (TA):

We have a TA in this course:

- TA: Suvajit Patra (2nd yr. PhD student) (IH413)
- Email: `suvajit.patra.cs20@gm.rkmvu.ac.in`

1.3 Prerequisite (s)

Student should have some knowledge in

- Mathematics: *Linear Algebra, Multivariate Calculus, Basis Optimisation and Basic probability*

- Computer programming: Python
- Basic concept in Algorithms and Data Structure
- Introduction to Machine Learning

1.4 Course url:

`https://xlms.rkmvu.ac.in/course/view.php?id=116`

1.5 Credit : 4 (four), approximately 60 credit hours

1.6 Tentative syllabus

Here are is a tentative syllabus:

- Artificial neural network (ANN): Modelling Single neuron activity, different types of activity functions (sigmoid, tanh, ReLU, ELU etc.), how to connect multiple neurons to form a network, Multi-layer perceptron
- Optimization: Back propagation, different loss functions, gradient decent, stochastic gradient decent and different update rules (AdaGrad, RPMSprops, Adam etc.) for network parameters, regularization, dropout, batch normalisation etc.
- Deep learning toolbox: Explore a deep learning toolbox like PyTorch (my personal choice)/ TensorFlow and their autograd functionalities
- Convolutional neural network (CNN): Concept of kernel and convolution, some pooling operation (max, average etc.), some standard CNN architectures like LeNet, AlexNet, VggNet, ResNet etc. and concept of transfer learning
- Recurrent neural network (RNN): Sequential data and how to handle those using neural network, general RNN architecture, some popular RNN architectures like *Long short-term memory (LSTM)*, *Gated recurrent unit (GRU)* and their different variants

- Deep generative models: *Variational Autoencoders (VAE)*, *Generative Adversarial Networks (GAN)*, *Normalizing flows*, *Diffusion models*, etc.

- Neural language model:
  - Introduction to NLP
  - Text preprocessing: tokenisation, stop words, stemming, lemmatisation, etc.
  - Vector representations of text: *Bag of Words*, *TF-IDF*, *word embeddings*, *Word2Vec*, *GloVE*, etc.
  - Sequence modelling: *Recurrent neural network (RNN)*, *Self-Attention network*, etc.
  - Transformers: *Attention, BERT* and its different variants, Encoder-Decoder models
  - Large language model (LLM): *GPT* different variants, *pre-trained language model*, *transfer learning*
  - Application: *text classification*, *sentiment analysis*, *Named Entity Recognition (NER)*, *machine translation*, *text summarization*, *text generation*, etc.

1.7 Related books

We will follow multiple books for different topics. Here are some suggested books will follow in our course :

- Charu C. Aggarwal. *Neural Networks and Deep Learning: A Textbook*, Springer Cham, 2nd edition, 2023.

- Simon Haykin. *Neural Networks and Learning Machines*, Pearson, 3rd edition, 2009.

- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.*, MIT Press, 1st edition, 2016. online

- Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*, Cambridge University Press, 2023. online

- Simon J.D. Prince. *Understanding Deep Learning*, MIT Press, 2023. online
- Eugene Charniak. *Introduction to deep learning*, MIT Press, 2018.
- Michael Nielsen. *Neural Networks and Deep Learning*, online
- Ovidiu Calin. *Deep Learning Architectures: A Mathematical Approach*, Springer Cham, 1st edition, 2020.
- Dan Jurafsky and James H. Martin. *Speech and Language Processing.* draft, 3rd edition, 2024. [online]
- Delip Rao, Brian McMahan. *Natural Language Processing with PyTorch*, O'Reilly Media, Inc, 2019
- Lewis Tunstall, Leandro von Werra, and Thomas Wolf. *Natural Language Processing with Transformers*, O'Reilly Media, Inc, 2022. online code only
- Yoav Goldberg. *A Primer on Neural Network Models for Natural Language Processing.* online

1.8 Evaluation:

Approximate weightage of different components in evaluation are as follows:

Midterm Exam	10%
Final Exam	40%
Assignment and Class test/Quizzes	25%
Project	25%

1.9 Assignments:

There will be some programming assignments. For the programming assignment, we will follow Python programming language for this course. The assignment submission deadline is strict and We will consider 11.59PM as our day end.

1.10 Project:

- Can be done in a group (max two students)
- Be careful about your project partner!
- If he is auditing the course then you will be in trouble!
- Define your own project
- Submit a one page project proposal- within fixed time (first four weeks)?
- Finished the work within the time-line
- Report submission
- Submission deadline: seven days before the final exam date, is strict and you can adjust your assignment buffer days here
- We will consider 11:59PM as our day end
- Final presentation: 20 min (divided into group members). Five days before the final exam date

1.11 Academic ethics:

We will follow some academic ethics:

- Your grade should reflect your own work.
- Copying or paraphrasing someone's work (code included), or permitting your own work to be copied or paraphrased, even if only in part, is strictly forbidden, and will result in an automatic grade of zero for the entire assignment or exam in which the copying or paraphrasing was done.
- So, ask yourself before copying from others.
- If you are going to have trouble completing an assignment, talk to the instructor and TA before due date.

1.12 DL & NLP related tools

Here are some popular tools:

- Machine Learning in Python - https://scikit-learn.org/stable/
- ML in GPU - https://rapids.ai/
- PyTorch - https://pytorch.org/
- Natural Language Toolkit - https://www.nltk.org/
- NLP for Indian language - https://github.com/AI4Bharat/indicnlp_catalog
- Bangla nlp - https://github.com/sagorbrur/bnlp
- ···

1.13 NLP datasets repository

You can find some datasets to evaluate yous NLP models here:

- https://github.com/niderhoff/nlp-datasets
- https://github.com/sebastianruder/NLP-progress
- https://www.nltk.org/nltk_data/
- https://universaldependencies.org/
- Movie subtitles: https://opus.nlpl.eu/OpenSubtitles-v2018.php
- I am not sure the data can be downloadable or not! But you can try for your application from these sources:
  - Related to Bengali literature: https://nltr.itewb.gov.in/
  - https://nltr.itewb.gov.in/downloads.php
  - https://rabindra-rachanabali.nltr.org/node/1
  - https://nazrul-rachanabali.nltr.org/
  - https://bankim-rachanabali.nltr.org/
  - https://sarat-rachanabali.nltr.org/
  - https://advaitaashrama.org/cw/content.php

1.14 DL & NLP related top tier conference

- International Conference on Machine Learning (ICML) - `https://icml.cc/`
- Neural Information Processing Systems (NeurIPS) - `https://neurips.cc/`
- International Conference on Learning Representations (ICLR) - `https://iclr.cc/`
- Association for the Advancement of Artificial Intelligence (AAAI) - `https://www.aaai.org/`
- Computer Vision Foundation (CVF) - `https://openaccess.thecvf.com/menu`
- Association for Computational Linguistics (ACL)[every year] - papers `https://aclanthology.org/venues/acl//`
- Empirical Methods in Natural Language Processing (EMNLP)[every year] - papers `https://aclanthology.org/venues/emnlp/`
- North American Chapter of the Association for Computational Linguistics (NAACL)[every year] - papers `https://aclanthology.org/venues/naacl/`
- European Chapter of the Association for Computational Linguistics (EACL)[every year] - papers `https://aclanthology.org/venues/eacl/`
- International Conference on Computational Linguistics (COLING) [alternate year (even)] - papers `https://aclanthology.org/venues/coling/`
- Conference on Natural Language Learning (CoNLL)[every year] - papers `https://aclanthology.org/venues/conll/`
- · · ·

1.15 DL & NLP related top journals

- Journal of Machine Learning Research (JMLR) - `https://www.jmlr.org/`
- Journal of Computational Linguistics (JCL) - `https://direct.mit.edu/coli/`
- Transactions of the Association for Computational Linguistics (TACL) - `https://transacl.org/index.php/tacl/index`
- Journal of Information Retrieval (JIR) - `https://www.springer.com/journal/10791`
- · · ·

1.16 For recent updates on ML you can follow the arXiv

You can go to Computer Science (CS) section in arXiv and under that you can find different branches of CS (like ML, CL, AI, IR, etc.).

- ML - `https://arxiv.org/list/cs.LG/recent`
- CL - `https://arxiv.org/list/cs.CL/recent`
- AI - `https://arxiv.org/list/cs.AI/recent`
- IR - `https://arxiv.org/list/cs.IR/recent`
- · · ·

1.17 Suggested reading

Please go through the class slides.

Lecture 2

Introduction to Artificial neural network

2.1 Suggested reading

Please go through *Chapter 1* of Charu Aggarwal's book [1] (you can find it in our library or you may find it online here but not sure!) or *Chapter 1* of Simon Haykin's book [12] (you can find it in our library or you may find online here but not sure!).

Lecture 3

Introduction to loss function and gradients

3.1 Suggested reading

Please go through *Chapter 2* of Charu Aggarwal's book [1] (you can find it in our library or you may find it online here but not sure!) or *Chapter 7* of Simon J.D. Prince's book [25] (you may find online here but not sure!).

Lecture 4

Introduction to backpropagation

4.1 Suggested reading

Please go through *Chapter 2, section 2.4* of Charu Aggarwal's book [1] (you can find it in our library or you may find it online here but not sure!)

4.2 Assignment-1

Implement a simple two layers neural network (similar to the one discussed in the class for MNIST data) to classify different object in CIFAR-100 dataset. Please download the dataset from here.

Upload your notebook with all the clean data in a folder named (file name also) as your_fullname_your_roll_number_assignment_1. For example, if your name is Amal Das and roll number is 0001, then folder and file name should be amal_das_0001_assignment_1

Submission deadline: 07-10-2024 (11:59 PM)

Lecture 5

Introduction to different activation functions

5.1 Suggested reading

Please go through the class slides.

Please go through *Chapter 4, section 4.4* of Charu Aggarwal's book [1] (you can find it in our library or you may find it online here but not sure!.

Lecture 6

Introduction to parameter initialisation and update rules

6.1 Suggested reading

Please go through the class slides.

Please go through *Chapter 2, section 2.7 and Chapter 4, section 4.5* of Charu Aggarwal's book [1] (you can find it in our library or you may find it online here but not sure!).

Project proposal submission deadline: 27-08-2024 (11:59 PM)

Lecture 7

Convolutional Networks-1

7.1 Suggested reading

Please go through the class slides.

Please go through *Chapter 2, section 2.7 and Chapter 4, section 4.5* of Charu Aggarwal's book [1] (you can find it in our library or you may find it online here but not sure!) OR you can check *Chapter 9* of Ian Goodfellow et al. book [11]. *Chapter 9* is freely downloadable from here.

Lecture 8

Convolutional Networks-2

8.1 Suggested reading

Please go through the class slides.

Please go through *Chapter 2, section 2.7 and Chapter 4, section 4.5* of Charu Aggarwal's book [1] (you can find it in our library or you may find it online here but not sure!) OR you can check *Chapter 9* of Ian Goodfellow et al. book [11]. *Chapter 9* is freely downloadable from here.

Lecture 9

Introduction to NLP, text document classification (Bag-of-Words model), language model: N-gram

9.1 Suggested reading

For the Bag-of-Words model, you can go through Jacob's [9] book *Chapter 4 (Linguistic applications of classification)*[online]. For further interest, I am encouraging you to go through the paper by *Pang et al.* titled with *Thumbs up?: sentiment classification using machine learning techniques* and the paper by *Zellig S. Harris* titled with *Distributional Structure.* For N-gram language model, you can go through Jurafsky and Martin's [15] book *Chapter 3 (N-gram Language Models)* [online]. For further interest, you can look into the papers referred in *Chapter 3*

Lecture 10

Word embeddings: vector semantics, neural word embedding

10.1 Suggested reading

First go through the word representation in vectorised form in Jurafsky and Martin's [15] book *Chapter 6 (Vector Semantics and Embeddings)* [online]. For *word2vec*, please go through the original paper title with *efficient estimation of word representations in vector space* [20]. A good documentation word2vec parameters title with *word2vec Parameter Learning Explained*. An online demo `https://ronxin.github.io/wevi/`. A word2vec test demo notebook is here: ss_word2vec_demo.ipynb

10.2 Homework

Derive the gradient of *cross-entropy* loss with respect to all the parameters in the *word2vec* model discussed in the class.

10.3 Assignment-2

Implement the vanila *Word2Vec: Skip-gram* model with the following:

- To train your model you use the data from (either one)

  - Swami Vivekananda's complete work: https://archive.org/details/completeworksofswamivivekananda_ninevolumes/SWAMI%20VIVEKANANDA%20COMPLETE%20WORKS%20%28Vol%201%29/ - You can find multiple download options (try txt format)
  - Rabindranath Tagore's work (Gitanjali, Gitabitan): https://nltr.itewb.gov.in/downloads.php - download from eBooks section

- Consider the hidden dimensions (word representation dimension): 10, 50, 100, 200, 300

- Do word clustering based on the above representation into a fixed number of clusters and check how are similar words grouping in your representation?

- Evaluate you representation using WordSim-353 and WiC datasets. Here just consider the intersection words and try to evaluate those words.

- Upload your notebook with all the clean data in a folder named (file name also) as your_fullname_your_roll_number_assignment_2. For example, if your name is Amal Das and roll number is 0001, then folder and file name should be amal_das_0001_assignment_2

Submission deadline: 18-10-2024 (11:59 PM)

Lecture 11

Recurrent Neural Network (RNN)

11.1 Suggested reading

For theoretical understanding of recurrent neural network (RNN), please go through the Goodfellow's [11] book *Chapter 10 (Sequence Modeling: Recurrentand Recursive Nets)* [online] or *Chapter 8, section 8.1 to 8.4* of Charu Aggarwal's book [1] (you can find it in our library or you may find it online here but not sure!) . Use cases of RNN in different NLP problems, please go through Jurafsky and Martin's [15] book *Chapter 8 (RNNs and LSTMs)* [online]. Here are some other useful resources:

- The Unreasonable Effectiveness of Recurrent Neural Networks

11.2 Homework

Derive the gradient of *cross-entropy* loss with respect to all the parameters for a vanilla *RNN* model discussed in the class.

Lecture 12

Long Short-Term Memory (LSTM) and text pre-processing

12.1 Suggested reading

For theoretical understanding of recurrent neural network (RNN), please go through the Goodfellow's [11] book *Chapter 10 section 10.10 (Sequence Modeling: Recurrentand Recursive Nets)* [online].or *Chapter 8, section 85* of Charu Aggarwal's book [1] (you can find it in our library or you may find it online here but not sure!) or Jurafsky and Martin's [15] book *Chapter 8, section 8.5 (RNNs and LSTMs)* [online]. For text pre-processing you can go through the Jurafsky and Martin's [15] book *Chapter 2 (Regular Expressions, Tokenization, Edit Distance)* [online]. Here are some other useful resources:

- The Unreasonable Effectiveness of Recurrent Neural Networks
- Understanding LSTM Networks
- SentencePiece: A good python library for text pre-processing.

12.2 Homework

Derive the gradient of *cross-entropy* loss with respect to all the parameters for a *LSTM* model discussed in the class.

12.3 Related papers

I am encouraging you to read the following papers:

- Learning long-term dependencies with gradient descent is difficult [6] (one of the original vanishing gradient papers)
- On the difficulty of training Recurrent Neural Networks [24] ((proof of vanishing gradient problem))
- A Neural Probabilistic Language Model [5]
- Generating text with recurrent neural networks [33]
- Sequence to sequence learning with neural networks [34]

Lecture 13

Self-Attention

13.1 Suggested reading

For the original *Self-Attention* work, please go through the original paper title with *Attention is All you Need* [37]. Use cases of *transformer* in different NLP task and another explanation, please go through Jurafsky and Martin's [15] book *Chapter 9 (Deep Learning Architectures for Sequence Processing)* [online]. Here are some other useful resources:

- The Illustrated Transformer
- The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

13.2 Homework

Derive the gradient of *cross-entropy* loss with respect to all the parameters for a *transformer* block discussed in the class.

13.3 Assignment-3

Implement a *Transformer* model with two-layers encoder and two-layers decoder for language translation task (English to Bengali/hindi). Consider the following:

- Use the network we have discussed in the class in transformer_exc.ipynb file.
- To train your model you can use the data from OpenSubtitles. For details about the data you can see `https://opus.nlpl.eu/OpenSubtitles/corpus/version/OpenSubtitles`

Submission deadline: 12-11-2024 (11:59 PM)

Lecture 14

Machine Translation and Encoder-Decoder Models

14.1 Suggested reading

Please go through Jurafsky and Martin's [15] book *Chapter 13, Sections 13.1 - 13.4 (Machine Translation)* [online]

14.2 Homework

Derive the gradient of *cross-entropy* loss with respect to all the parameters for a *transformer* one-encoder and one-decoder blocks for language translation task discussed in the class.

Lecture 15

Contextual embeddings and pre-trained language model

15.1 Suggested reading

Please go through Jurafsky and Martin's [15] book *Chapter 11 (Masked Language Models)* [online]. Also, I am encoring you all to please go through the original *BERT* paper [8]

15.2 Homework

Derive the gradient of *cross-entropy* loss with respect to all the parameters for a *BERT* [8] model with two-encoder for both masked language and next sentence prediction task discussed in the class.

- The Illustrated Transformer
- The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
- BERT 101 State Of The Art NLP Model Explained
- GPT-2
- GPT-NeoX, A large scale language model.
- DeBERTa code

- XLNet code
- RoBERTa code and related models
- Hugging Face: a source of pre-trained language models

15.3 Class presentation

For our class presentation, we'll discuss the following papers:

- GPT: Improving Language Understanding by Generative Pre-Training [28], 2018
- GPT2: Language Models are Unsupervised Multitask Learners [29] (can share two groups), 2019
- RoBERTa: A Robustly Optimized BERT Pretraining Approach [19], 2019
- DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter [19], 2019
- BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension [17], 2020
- SpanBERT: Improving Pre-training by Representing and Predicting Spans [14], 2020
- Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer [30] (can share two groups), 2020
- End-to-End Object Detection with Transformers [7], 2020
- ALBERT: A Lite BERT for Self-supervised Learning of Language Representations [16], 2020
- Incorporating BERT into Neural Machine Translation [39], 2020
- Learning Transferable Visual Models From Natural Language Supervision [26] (can share two groups), 2021

- DeBERTa: Decoding-enhanced BERT with Disentangled Attention [13], 2021
- Longformer: The Long-Document Transformer [4], 2021
- Taming Transformers for High-Resolution Image Synthesis [10], 2021
- Training language models to follow instructions with human feedback [22], 2022
- LLaMA: Open and Efficient Foundation Language Models [35] , 2023
- Llama 2: Open Foundation and Fine-Tuned Chat Models [36] , 2023
- Robust speech recognition via large-scale weak supervision [27] (can share two groups), 2023

Lecture 16

Introduction to large language model (LLM)

16.1 Suggested reading

Please go through class slides Also, I am encoring you all to please go through the original papers:

- Training language models to follow instructions with human feedback
- Llama 2: Open Foundation and Fine-Tuned Chat Models

Here is a library for mini-GPT minGPT and Llama .

Lecture 17

Auto-encoders and Variational auto-encoders

Please go through *Chapter 20, Section 20.3* of Murphy's new book [21] . The book is freely downloadable from here.

For more details on Autoencoders, you can go through the *Chapter 14* of Ian Goodfellow et al. book [11]. *Chapter 14* is freely downloadable from here.

17.1 Homework

Derive the gradient of *MSE* loss with respect to all the parameters for a *Variational auto-encoder* model with two-encoder and two-decoder layers for the MNIST digit reconstruction task discussed in the class.

17.2 Assignment-4

Implement an *Auto-encoder* and a *Variational auto-encoder* model for MNIST dataset discussed in the class using the notebook and .py file shared in the xlm under assignment-4.

Submission deadline: 30-11-2024 (11:59 PM)

Lecture 18

Language model evaluation metric

Please go through the class slides and try to read the following papers:

- BLEU: a Method for Automatic Evaluation of Machine Translation [23]
- ROUGE: A Package for Automatic Evaluation of Summaries [18]
- METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments [3]
- Word Error Rate Estimation for Speech Recognition: e-WER [2]
- COMET: A Neural Framework for MT Evaluation [31]
- BLEURT: Learning Robust Metrics for Text Generation [32]
- BERTScore: Evaluating Text Generation with BERT [38]

18.1 Homework

Find out the different pros and cons of language model evaluation metrics like Perplexity, BLEU [23], ROUGE [23], METEOR [3].

Bibliography

[1] Charu C. Aggarwal. *Neural Networks and Deep Learning: A Textbook*. Springer Cham, 2nd edition, 2023.

[2] Ahmed Ali and Steve Renals. Word error rate estimation for speech recognition: e-wer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.

[4] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2021.

[5] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

[6] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the 16th European Conference onComputer Vision (ECCV)*, 2020.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Conference of the North Amer-*

ican Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019.

[9] Jacob Eisenstein. *Introduction to Natural Language Processing.* MIT Press, 1st edition, 2019.

[10] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 1st edition, 2016.

[12] Simon Haykin. *Neural Networks and Learning Machines.* Pearson, 3rd edition, 2009.

[13] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[14] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[15] Dan Jurafsky and James H. Martin. *Speech and Language Processing.* draft, 3rd edition, 2023.

[16] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[17] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[18] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[20] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[21] Kevin Patrick Murphy. *Probabilistic Machine Learning: An introduction.* The MIT Press, 1st edition, 2022.

[22] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[23] 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 of the Association for Computational Linguistics (ACL)*, pages 311–318, 2002.

[24] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks, 2013.

[25] Simon J.D. Prince. *Understanding Deep Learning.* MIT Press, 1st edition, 2023.

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[27] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale

weak supervision. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[28] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[31] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. Comet: A neural framework for mt evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, 2020.

[32] Thibault Sellam, Dipanjan Das, and Ankur Parikh. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7881–7892, 2020.

[33] Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML)*, pages 1017–1024, 2011.

[34] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems(NIPS)*, pages 3104–3112, 2014.

[35] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet Marie-Anne Lachaux , Timothee Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, and Armand Joulin. Llama: Open and efficient foundation language models, 2023.

[36] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton, and Ferrer Moya Chen Guillem Cucurull David Esiobu Jude Fernandes Jeremy Fu Wenyin Fu Brian Fuller Cynthia Gao Vedanuj Goswami Naman Goyal Anthony Hartshorn Saghar Hosseini Rui Hou Hakan Inan Marcin Kardas Viktor Kerkez Madian Khabsa Isabel Kloumann Artem Korenev Punit Singh Koura Marie-Anne Lachaux Thibaut Lavril Jenya Lee Diana Liskovich Yinghai Lu Yuning Mao Xavier Martinet Todor Mihaylov Pushkar Mishra Igor Molybog Yixin Nie Andrew Poulton Jeremy Reizenstein Rashi Rungta Kalyan Saladi Alan Schelten Ruan Silva Eric Michael Smith Ranjan Subramanian Xiaoqing Ellen Tan Binh Tang Ross Taylor Adina Williams Jian Xiang Kuan Puxin Xu Zheng Yan Iliyan Zarov Yuchen Zhang Angela Fan Melanie Kambadur Sharan Narang Aurelien Rodriguez Robert Stojnic Sergey Edunov Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2019.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2017.

[38] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[39] Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. Incorporating bert into neural machine translation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.