ECE472, Deep Learning – Syllabus, Fall 2022, Room 104, Th 6-9

tldr: We will introduce the concepts relevant to so called "deep learning" — our fundamental processes are based on computations performed over differentiable graphs, where nodes correspond to operations and edges correspond to operands. We will use the Microsoft Teams site: "ECE-472-1-Deep Learning-2022FA"

Instructor Chris Curro, EE '15, MEE '16; `professor@curro.cc`

Reference Textbook Ian Goodfellow and Yoshua Bengio and Aaron Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`

Assignments There will be a handful of programming assignments — I recommend using Python and TensorFlow for these — each will be due either 1 or 2 weeks after the assigned date. There will be 2 larger projects: a midterm and final.

Citations Plagiarism will not be tolerated. All cases of suspected plagiarism will be submitted to the Dean's office for investigation. Feel free to ask questions of your peers, but please cite them for any help you receive. Cite resources you may utilize from the web and elsewhere.

Quizzes There will be quizzes most weeks. These quizzes will test understanding of assigned research papers. Expect 1-3 papers on most weeks. If you must miss a quiz, please let me know before hand and we will arrange appropriate accommodations, otherwise you receive a zero for that quiz.

Grading Grading breakdown in table at bottom of page. If you fail to submit an assignment you will fail the course. Unexcused late assignments will have a single letter grade deducted per 2 days late. The maximum grade for any tardy assignment is a B.

Attendance We will not take attendance, but it may factor into your participation score. Participation score is multifaceted. We will discuss this during the first class.

Office hours We will arrive at an appropriate schedule during the first class. Expect 1 or 2 hours per week. Additional hours by appointment. Office hours will be conducted remotely on Microsoft Teams.

Grading	
Assignments	30%
Projects	30%
Quizzes	30%
Participation	10%

Boilerplate

Required links

- https://cooper.edu/sites/default/files/uploads/assets/site/files/2020/Cooper-Union-Policy-Upholding-Human-Rights-Title-IX-Protections.pdf
- https://cooper.edu/students/student-affairs/disability
- https://cooper.edu/students/student-affairs/health/counseling

Students Outcomes

- Ability to
  - discuss contemporary research in an intelligent way
  - recognize failings in a given experiment and synthesize follow-up experimentation
  - synthesize hypotheses on ablative and compositional experiments
  - argue in an evidence based way and make conclusions
  - communicate mathematical concepts in a narrative
  - identify situations in which deep learning may or may not be appropriate over other machine learning techniques

We will assess the aformentioned abilities through class discussions, quizzes, and assignment submissions.

Prerequisite Skills

- Knowledge of a programming language (Python preferred)
- Knowledge of differentiation in multivariate calculus
- Knowledge of basic linear algebra and probability (e.g., matrix multiplication, distributions)

Approximate list of topics

Introduction Linear regression. Regression with basis functions. Gradient descent. Automatic differention; reverse mode and forward mode. Affine projection. Multi-layer perceptrons. Activation functions. Cross validation. L1 and L2 regularization. Dropout, bath normalization, and friends. Logistic regression. Binary cross entropy, and other entropy based loss functions. Weight initialization.

Convolutions and friends Convolutional layers. Strided convolutions. Pooling. Residual connections. Transposed convolutions.

Transformers and friends Attention. Multi-head attention. Tokenization. CLIP. Generative-pretraining

Excotica Neural ODEs. Diffusion models. Mixture of experts. Large language models.

Applications and other techniques Autoencoders. Super-resolution. Image inpainting. Speech generation. Speech recognition. Music generation. Image generation. Recommender systems. Text classification. Natural language generation. Reinforcement learning. Style transfer. Content transfer.

Midterm project - due Oct 27.

The goal of the midterm project is to reproduce results from a contemporary research paper.

Procedure:

1. Find a paper of interest.
2. Pick a reasonable subset of the results to reproduce in the time allotted with present resource constraints.
3. Submit a proposal to me. If approved, continue. Else, go back to step 1 or 2, according to feedback.
4. Write code to reproduce the experiment. Document any necessary assumptions or changes from the paper.
5. Submit code and proof/evidence of reproduction. Submit a ~1-page document explaining your engagement with the work.

Final project, option 1 - due Dec 15.

The goal of the final project is to attempt to produce original research. We define success as a well-demonstrated engagement with the topic of the work.

Procedure:

1. Familiarize yourself with a topic of interest.
2. Propose amendment(s) to or suggest a novel application of an extant methodology.
3. Present the proposal to the class community for feedback. Iterate as necessary.
4. Write code to perform the experiment and produce the results.
5. Write a paper in contemporary conference-style describing your experiments and engagement with the work.
6. Produce a presentation (need not be slides) to present to your peers and guests.

Final project, option 2 - due Dec 15.

Combine open source pre-trained models together in novel way to develop a useful application. For an example of a basic project, see `https://replicate.com/andreasjansson/stable-diffusion-animation`

Procedure:

1. Familiarize yourself with a topic of interest.
2. Develop a proposal.
3. Present the proposal to the class community for feedback. Iterate as necessary.
4. Write code.
5. Write a paper in contemporary conference-style describing your experiments and engagement with the work.
6. Produce a presentation (need not be slides) to present to your peers and guests.

Assignment 1

tldr: Perform linear regression of a noisy sinewave using a set of gaussian basis functions with learned location and scale parameters. Model parameters are learned with stochastic gradient descent. Use of automatic differentiation is required. Hint: note your limits!

Problem Statement Consider a set of scalars $\{x_1, x_2, \ldots, x_N\}$ drawn from $\mathcal{U}(0, 1)$ and a corresponding set $\{y_1, y_2, \ldots, y_N\}$ where:

$$y_i = \sin(2\pi x_i) + \epsilon_i \tag{1}$$

and ϵ_i is drawn from $\mathcal{N}(0, \sigma_{\text{noise}})$. Given the following functional form:

$$\hat{y}_i = \sum_{j=1}^{M} w_j \phi_j \left(x_i \mid \mu_j, \sigma_j\right) + b \tag{2}$$

with:

$$\phi(x \mid \mu, \sigma) = \exp \frac{-(x - \mu)^2}{\sigma^2} \tag{3}$$

find estimates $\hat{b}$, $\{\hat{\mu}_j\}$, $\{\hat{\sigma}_j\}$, and $\{\hat{w}_j\}$ that minimize the loss function:

$$J(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2 \tag{4}$$

for all (x_i, y_i) pairs. Estimates for the parameters must be found using stochastic gradient descent. A framework that supports automatic differentiation must be used. Set $N = 50, \sigma_{\text{noise}} = 0.1$. Select M as appropriate. Produce two plots. First, show the data-points, a noiseless sinewave, and the manifold produced by the regression model. Second, show each of the M basis functions. Plots must be of suitable visual quality.

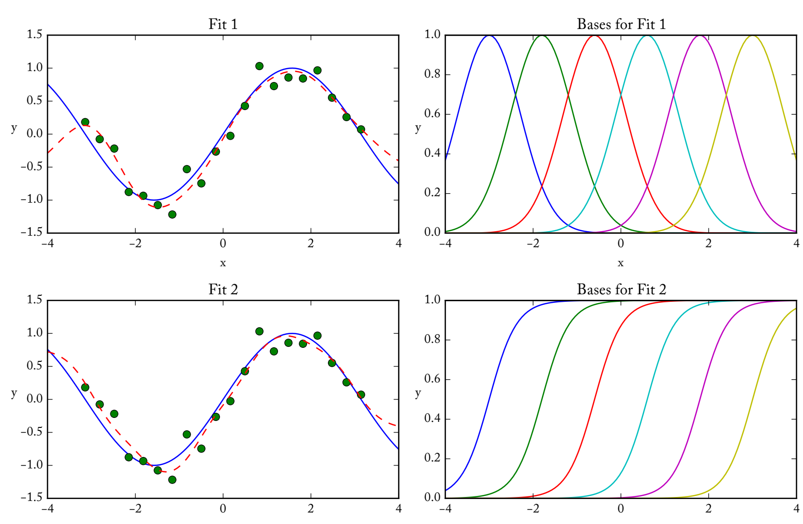

Figure 1: Example plots for models with equally spaced sigmoid and gaussian basis functions.

Assignment 2

tldr: Perform binary classification on the spirals dataset using a multi-layer perceptron. You must generate the data yourself.

Problem Statement Consider a set of examples with two classes and distributions as in Figure 2. Given the vector $x \in \mathbb{R}^2$ infer its target class $t \in \{0, 1\}$. As a model use a multi-layer perceptron f which returns an estimate for the conditional density $p(t = 1 \mid x)$:

$$f : \mathbb{R}^2 \to [0, 1] \tag{5}$$

parametrisized by some set of values θ. All of the examples in the training set should be classified correctly (i.e. $p(t = 1 \mid x) > 0.5$ if and only if $t = 1$). Impose an L^2 penalty on the set of parameters. Produce one plot. Show the examples and the boundary corresponding to $p(t = 1 \mid x) = 0.5$. The plot must be of suitable visual quality. It may be difficult to to find an appropriate functional form for f, write a few sentences discussing your various attempts.

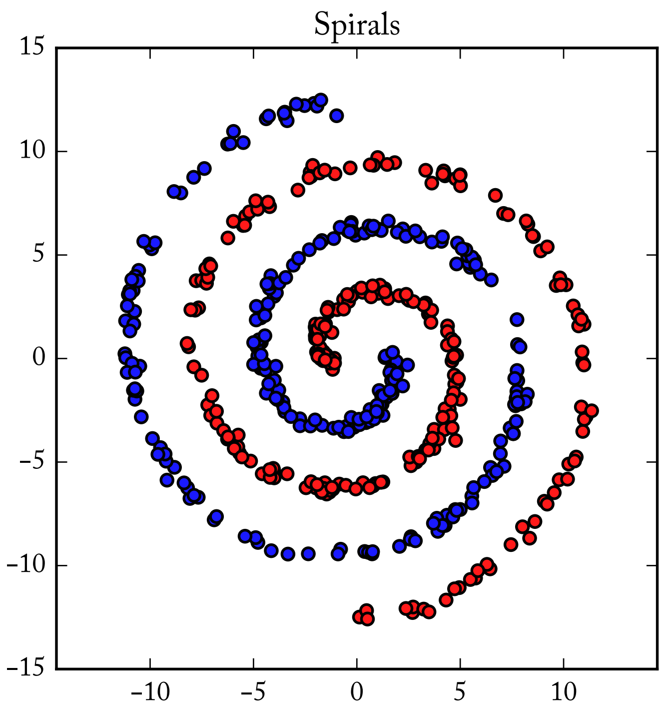

Figure 2: Sample spiral data.

Assignment 3

tldr: Classify MNIST digits with a (optionally convoultional) neural network. Get at least 95.5% accuracy on the test test.

Problem Statement Consider the MNIST dataset consisting of 50,000 training images, and 10,000 test images. Each instance is a 28×28 pixel handwritten digit zero through nine. Train a (optionally convolutional) neural network for classification using the training set that achieves at least 95.5% accuracy on the test set. Do not explicitly tune hyperparameters based on the test set performance, use a validation set taken from the training set as discussed in class. Use dropout and an L^2 penalty for regularization. Note: if you write a sufficiently general program the next assignment will be very easy.

Do not use the built in MNIST data class from TensorFlow.

Extra challenge (optional) In addition to the above, the student with the fewest number of parameters for a network that gets at least 80% accuracy on the test set will receive a prize. There will be an extra prize if any one can achieve 80% on the test set with a single digit number of parameters. For this extra challenge you can make your network have any crazy kind of topology you'd like, it just needs to be optimized by a gradient based algorithm.

Assignment 4

tldr: Classify CIFAR10. Acheive performance similar to the state of the art. Classify CIFAR100. Achieve a top-5 accuracy of 90%.

Problem Statement Consider the CIFAR10 and CIFAR100 datasets which contain 32×32 pixel color images. Train a classifier for each of these with performance similar to the state of the art (for CIFAR10). It is your task to figure out what is state of the art. Feel free to adapt any techniques from papers you read. I encourage you to experiment with normalization techniques and optimization algorithms in this assignment. Write a paragraph or two summarizing your experiments. Hopefully you'll be able to resuse your MNIST program.

Assignment 5

tldr: Classify the AG News dataset.

Problem Statement Consider the AG News dataset at `https://huggingface.co/datasets/ag_news` which contains headlines and descriptions for a large set of news articles. Perform proper cross validation. You may use pretrained models; for example, `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

Papers

This paper list is for Fall 2021. The paper list will be be updated by Week 2 of class.

Week 1

1. Atilim Gunes Baydin, Barak A. Pearlmutter, and Alexey Andreyevich Radul. "Automatic differentiation in machine learning: a survey". In: *CoRR* abs/1502.05767 (2015). arXiv: `1502.05767`. URL: `http://arxiv.org/abs/1502.05767`
2. Leon Bottou. "Stochastic Gradient Descent Tricks". In: *Neural Networks, Tricks of the Trade, Reloaded*. Neural Networks, Tricks of the Trade, Reloaded. Vol. 7700. Lecture Notes in Computer Science (LNCS). Springer, Jan. 2012, pp. 430–445. URL: `https://www.microsoft.com/en-us/research/publication/stochastic-gradient-tricks/`

Week 2

3. Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: `1412.6980 [cs.LG]`
4. Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: `1711.05101 [cs.LG]`

Week 3

5. Kaiming He et al. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015. arXiv: `1502.01852 [cs.CV]`
6. Christian Szegedy et al. *Going Deeper with Convolutions*. 2014. arXiv: `1409.4842 [cs.CV]`
7. Kaiming He et al. *Identity Mappings in Deep Residual Networks*. 2016. arXiv: `1603.05027 [cs.CV]`

Week 4

8. Guillaume Alain and Yoshua Bengio. *Understanding intermediate layers using linear classifier probes*. 2018. arXiv: `1610.01644 [stat.ML]`
9. Gabriel Pereyra et al. *Regularizing Neural Networks by Penalizing Confident Output Distributions*. 2017. arXiv: `1701.06548 [cs.NE]`
10. Sergey Ioffe. *Batch Renormalization: Towards Reducing Minibatch Dependence in Batch-Normalized Models*. 2017. arXiv: `1702.03275 [cs.LG]`

Week 5

11. Amir Gholami et al. *SqueezeNext: Hardware-Aware Neural Network Design*. 2018. arXiv: `1803.10615 [cs.NE]`
12. Andrew Howard et al. *Searching for MobileNetV3*. 2019. arXiv: `1905.02244 [cs.CV]`
13. Xiao Sun et al. "Ultra-Low Precision 4-bit Training of Deep Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1796–1807. URL: `https://proceedings.neurips.cc/paper/2020/file/13b919438259814cd5be8cb45877d577-Paper.pdf`

Week 6

14. Chiyuan Zhang et al. *Understanding deep learning requires rethinking generalization*. 2017. arXiv: `1611.03530 [cs.LG]`
15. Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *CoRR* abs/1703.03400 (2017). arXiv: `1703.03400`. URL: `http://arxiv.org/abs/1703.03400`
16. Oren Rippel et al. *Metric Learning with Adaptive Density Discrimination*. 2016. arXiv: `1511.05939 [stat.ML]`

Week 7

17. Léonard Blier, Pierre Wolinski, and Yann Ollivier. *Learning with Random Learning Rates*. 2019. arXiv: `1810.01322 [cs.LG]`
18. Samuel L. Smith et al. *Don't Decay the Learning Rate, Increase the Batch Size*. 2018. arXiv: `1711.00489 [cs.LG]`
19. Leslie N. Smith and Nicholay Topin. *Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates*. 2018. arXiv: `1708.07120 [cs.LG]`

Week 8

20. Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. *A Neural Algorithm of Artistic Style*. 2015. arXiv: `1508.06576 [cs.CV]`
21. Xun Huang and Serge Belongie. *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization*. 2017. arXiv: `1703.06868 [cs.CV]`
22. Tero Karras et al. *Analyzing and Improving the Image Quality of StyleGAN*. 2020. arXiv: `1912.04958 [cs.CV]`
23. Tero Karras et al. "Alias-Free Generative Adversarial Networks". In: *CoRR* abs/2106.12423 (2021). arXiv: `2106.12423`. URL: `https://arxiv.org/abs/2106.12423`

Week 9

24. Aäron van den Oord et al. "WaveNet: A Generative Model for Raw Audio". In: *CoRR* abs/1609.03499 (2016). arXiv: `1609.03499`. URL: `http://arxiv.org/abs/1609.03499`
25. Ron J. Weiss et al. "Wave-Tacotron: Spectrogram-free end-to-end text-to-speech synthesis". In: *CoRR* abs/2011.03568 (2020). arXiv: `2011.03568`. URL: `https://arxiv.org/abs/2011.03568`
26. Aäron van den Oord et al. "Parallel WaveNet: Fast High-Fidelity Speech Synthesis". In: *CoRR* abs/1711.10433 (2017). arXiv: `1711.10433`. URL: `http://arxiv.org/abs/1711.10433`
27. Mikolaj Binkowski et al. "High Fidelity Speech Synthesis with Adversarial Networks". In: *CoRR* abs/1909.11646 (2019). arXiv: `1909.11646`. URL: `http://arxiv.org/abs/1909.11646`

Week 10

28. Manzil Zaheer et al. *Big Bird: Transformers for Longer Sequences*. 2021. arXiv: `2007.14062 [cs.LG]`

29. Andrew Jaegle et al. *Perceiver: General Perception with Iterative Attention*. 2021. arXiv: 2103.03206 [cs.CV]

30. Andrew Jaegle et al. *Perceiver IO: A General Architecture for Structured Inputs and Outputs*. 2021. arXiv: 2107.14795 [cs.LG]

31. Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In: (2019)

Week 11

32. Kai Arulkumaran et al. "A Brief Survey of Deep Reinforcement Learning". In: *CoRR* abs/1708.05866 (2017). arXiv: 1708.05866. URL: http://arxiv.org/abs/1708.05866

33. Julian Schrittwieser et al. "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model". In: *CoRR* abs/1911.08265 (2019). arXiv: 1911.08265. URL: http://arxiv.org/abs/1911.08265

34. Lili Chen et al. "Decision Transformer: Reinforcement Learning via Sequence Modeling". In: *CoRR* abs/2106.01345 (2021). arXiv: 2106.01345. URL: https://arxiv.org/abs/2106.01345

35. Danijar Hafner et al. "Mastering Atari with Discrete World Models". In: *CoRR* abs/2010.02193 (2020). arXiv: 2010.02193. URL: https://arxiv.org/abs/2010.02193

Week 12

36. Kaiming He et al. *Masked Autoencoders Are Scalable Vision Learners*. 2021. arXiv: 2111.06377 [cs.CV]

Week 13

37. Rishabh Agarwal et al. "Neural Additive Models: Interpretable Machine Learning with Neural Nets". In: *CoRR* abs/2004.13912 (2020). arXiv: 2004.13912. URL: https://arxiv.org/abs/2004.13912

38. Muzammal Naseer et al. "Intriguing Properties of Vision Transformers". In: *CoRR* abs/2105.10497 (2021). arXiv: 2105.10497. URL: https://arxiv.org/abs/2105.10497