MSAI-437: Deep Learning

Winter 2025

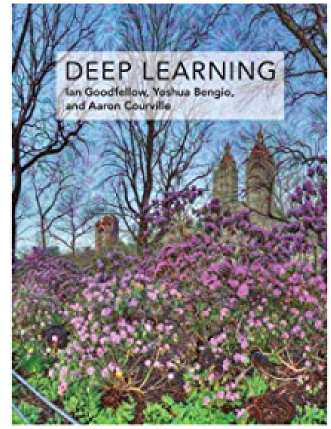
Introduction

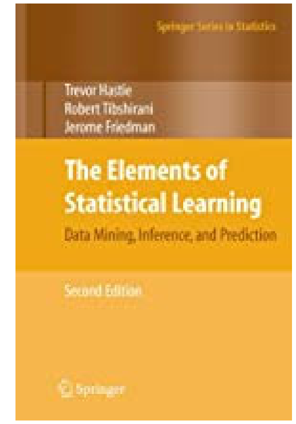
Course Overview: Introduction

Course Instructor:	David Demeter
Teaching Assistant:	Karan Garkel and Jacob John
Meeting Times:	Tuesdays and Thursdays 11:00am to 12:20pm
Location:	Technological Institute, Room #L221
Class Title:	Deep Learning
Format:	Bi-weekly lectures Three Homework Assignments Final Project In-class Final Exam Surveys and Quizzes In-person attendance strongly encouraged Interactive classes preferred
Prerequisites:	Intermediate Python Programming (recursion, data structures, etc.) Basic familiarity with linear algebra and derivatives Mastery of concepts covered in MSAI-349 Training feed-forward neural networks

Course Overview: Resources

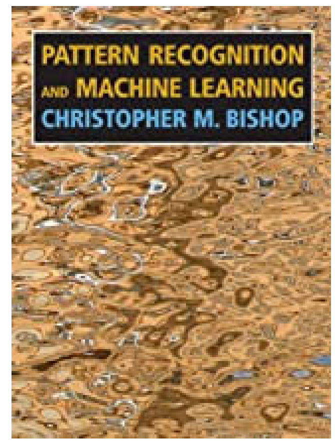
Text Books: None required, some recommended

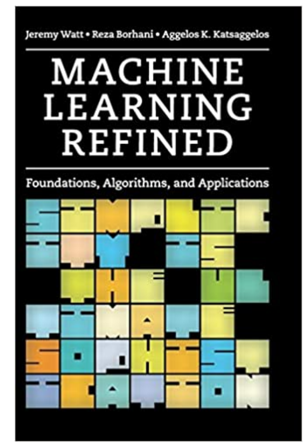




Ian Goodfellow , Yoshua Bengio, Aaron Courville:
Deep Learning (Adaptive Computation and Machine Learning series),
MIT Press, Massachusetts, 2016
https://www.deeplearningbook.org/

Trevor Hastie, Robert Tobshirani, Jerome Friedman:
The Elements of Statistical Learning –
Data Mining, Inference, and Prediction,
2nd edition, Springer, New York, 2009
http://www-stat.stanford.edu/~tibs/ElemStatLearn/





Christopher M. Bishop:
Pattern Recognition and Machine Learning,
Springer, New York, 2006

Jeremy Watt , Reza Borhani, Aggelos K. Katsaggelos
Machine Learning Refined: Foundations, Algorithms, and Applications 1st Edition

Resources: Machines in Wilkinson Lab have GPUs and MSAI students have access
(Note: Students may be also responsible for arranging their own GPU
resources through AWS, GCP, Colab, etc.)

Course Overview: Grading

Deliverables:	HW #1: Cost Functions and MLPs	10 pts
	HW #2: CNNs and Autoencoders	10 pts
	HW #3: Recurrent Models and Transformers	10 pts
	Final Project - Proposal	5 pts
	Final Project - Report	25 pts
	Final Exam (Individual)	30 pts
	Reserved	10 pts
	Total	100 pts

Standard scale: 93%-100% = A, 90%-93% = A-,

87%-90% = B+, 83%-87% = B, 80%-83% = B-,

77%-80% = C+, 73%-77% = C, 70%-73% = C-,

60%-70% = D, and less than 60% = F

Course Overview: In-Class Survey

- Expertise: ML Foundations, FFNNs, CNNs, GANs, Autoencoders, RNN, Transformers, Deep Reinforcement Learning, Diffussion Models, Other Topics
- Interests: FFNNs, CNNs, GANs, Autoencoders, RNNs, Transformers, RL, Other Topics
- Domains: Vision, Audio, NLP, Data Science, Other Topics

Course Overview: Schedule

Week #1: Course Overview and Review of Machine Learning Foundations

- Introduction, class policies and survey
- Characterization of deep learning, teaching approach, models covered and domains covered
- Review of regression and classification tasks
- Evaluation metrics, objective functions and learning algorithms
- Limitations and challenges of non-gradient optimization methods

Week #2: Multi-layer Perceptrons and Gradient Descent

- Reframing role of single perceptron, multi-layer perceptrons and architectural representations
- Linear and non-linear activation functions, decision boundaries and deep networks
- Gradient descent applied to ordinary least squares regression and perceptrons
- Illustrative examples of gradient descent in Jupyter Notebooks

Week #3: Backpropagation and Regularization

- Gradient calculations and backpropagation via gradient descent
- Interpretation of under/over fitting in terms of bias/variance trade-offs
- L_1 and L_2 regularization, dropout, data augmentation and batch normalization

Week #4: Convolutional Neural Networks and Adversarial Examples

- Vision tasks, model parameters, image representations and biological motivations
- Building feature maps (kernels), stride, padding, dilation and channels
- Pooling: maximum, minimum, averaging and how to choose
- Convolution architectures and training considerations
- Adversarial examples, fast gradient sign method and potential defenses

Course Overview: Schedule (*Cont.*)

Week #5: Generative Adversarial Networks

- Discriminators (high-to-low dimensional mapping) and generative models (intelligent perturbations)
- Generative methods: inverting a CNN (low-to-high dimensions), gradient sign method, random noise, etc.
- GAN architecture and optimization strategies (generator and discriminator)
- Data distributions, noise sampling, mode collapse and motivation for VAEs
- Mini-max framing of condition generation and selected examples

Week #6: Autoencoders and Recurrent Neural Networks

- PCA framing, potential applications of dense representations and real-world examples Introduction of recurrent architectures, hidden states and backpropagation through time
- Alternative activation functions, including LSTM and GRU cells
- Sequence-to-sequence models, autoencoders and encoder-decoder architectures

Week #7: Transformers

- Preliminaries of language modeling, embedding spaces, context embeddings and task-specific heads
- Introduction of transformer-based models and encoder/decoder stack abstractions
- Attention mechanism, key-query value calculations, multi-head attention and position encoding
- Decoder-only stack, autoregressive training objective and transfer learning of GPT models

Course Overview: Schedule (*Cont.*)

Week #8: Deep Reinforcement Learning

- Compare and contrast reinforcement learning to supervised learning and construction of RL problems
- Exploration and exploitation, reward functions and the one-armed bandit problem
- Q-learning framework and introduction of state spaces, actions, rewards/penalties, discounting and policies
- Enumerating acquired knowledge, policy gradients and approximating q-tables with deep learning
- Deep Q-learning and temporal difference learning.

Week #9: Diffusion Models and Miscellaneous Topics/Catch-up

- Tasks: content generation, representation learning and artistic tools
- Forward diffusion and reverse de-noising processes
- Sampling and conditional generation
- Applying diffusion to selected tasks
- Hebbian learning, restricted Boltzmann machines and deep belief networks

Week #10: Review and In-class Final Exam

- Review of class content and miscellaneous questions/discussion
- Final exam

Course Overview: Format

What's New:

More challenging assignments
More stringent grading
Reading assignments
Jupyter notebooks for illustration (and sample code)
Group dynamics

- maximum group size of 3 people
- up to 1.0 bonus points total for groups of two
- up to 2.0 bonus points total for groups of one
- you can drop from a group, but cannot join another group
- you cannot drop from a group within one week of a deadline

May defer some questions to office hours

What's the Same:

Flexible approach (within reason)
More applied than mathematical/theoretical

Questions?

Course Overview: Policies

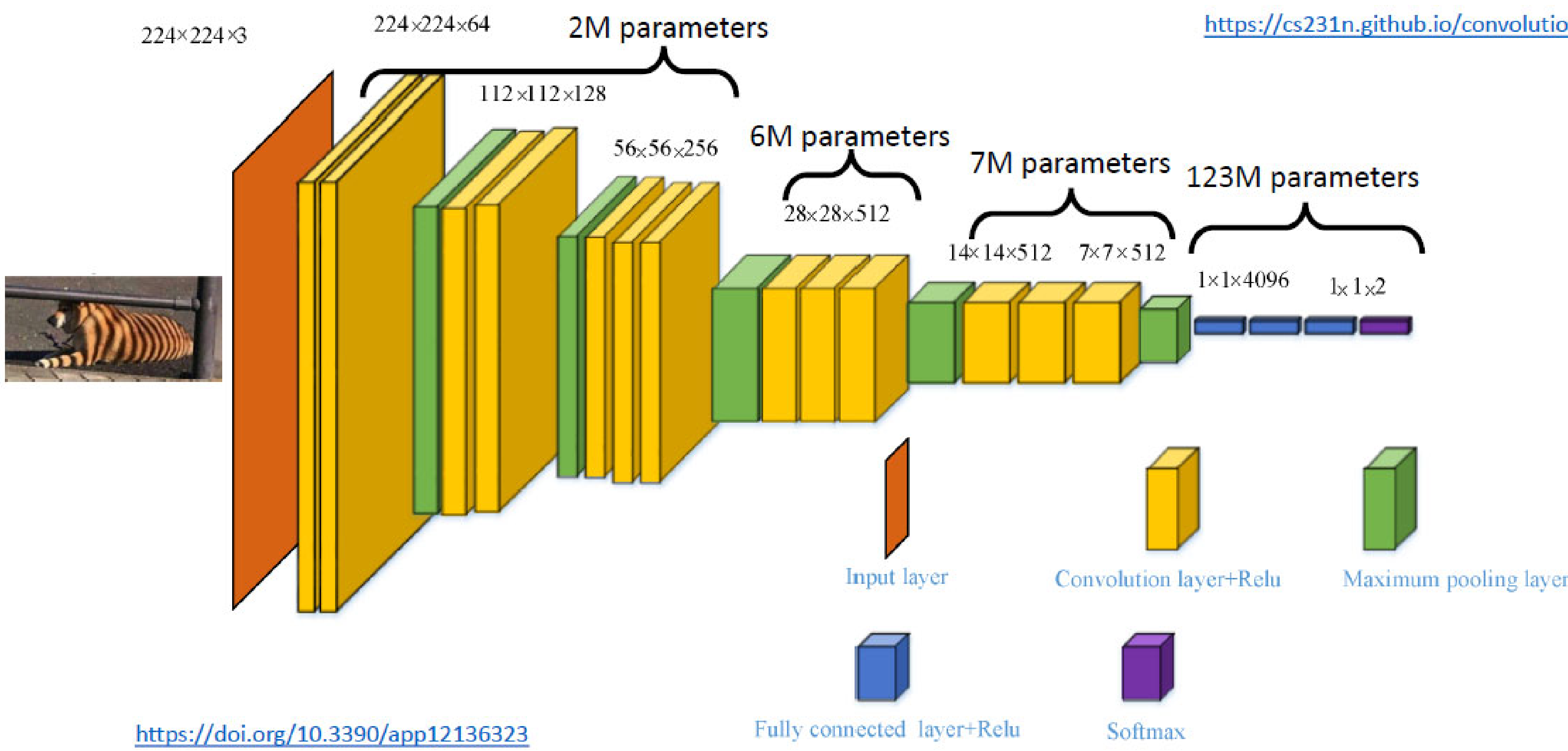
Video Recordings:	Video lectures will be posted to Canvas (for now) May not be shared outside of the class May not be used in lieu of in-person attendance (per University policy)
Communication:	Please use NorthwesternMSAI349@gmail.com
Groups:	A sign-up sheet will be posted in Canvas
Piazza:	Do not ask grading questions or share confidential Please make all posts visible to the entire class
AccessibleNU:	Please make me aware of any accommodations in a timely manner
COVID-19:	Please be aware of and follow University policies
Positive Environment:	Please make me aware of anything that creates an uncomfortable environment.
Other:	Please see the course syllabus for a longer list of University policies.

Questions?

Final Project Methods

1. Convolutional Neural Networks

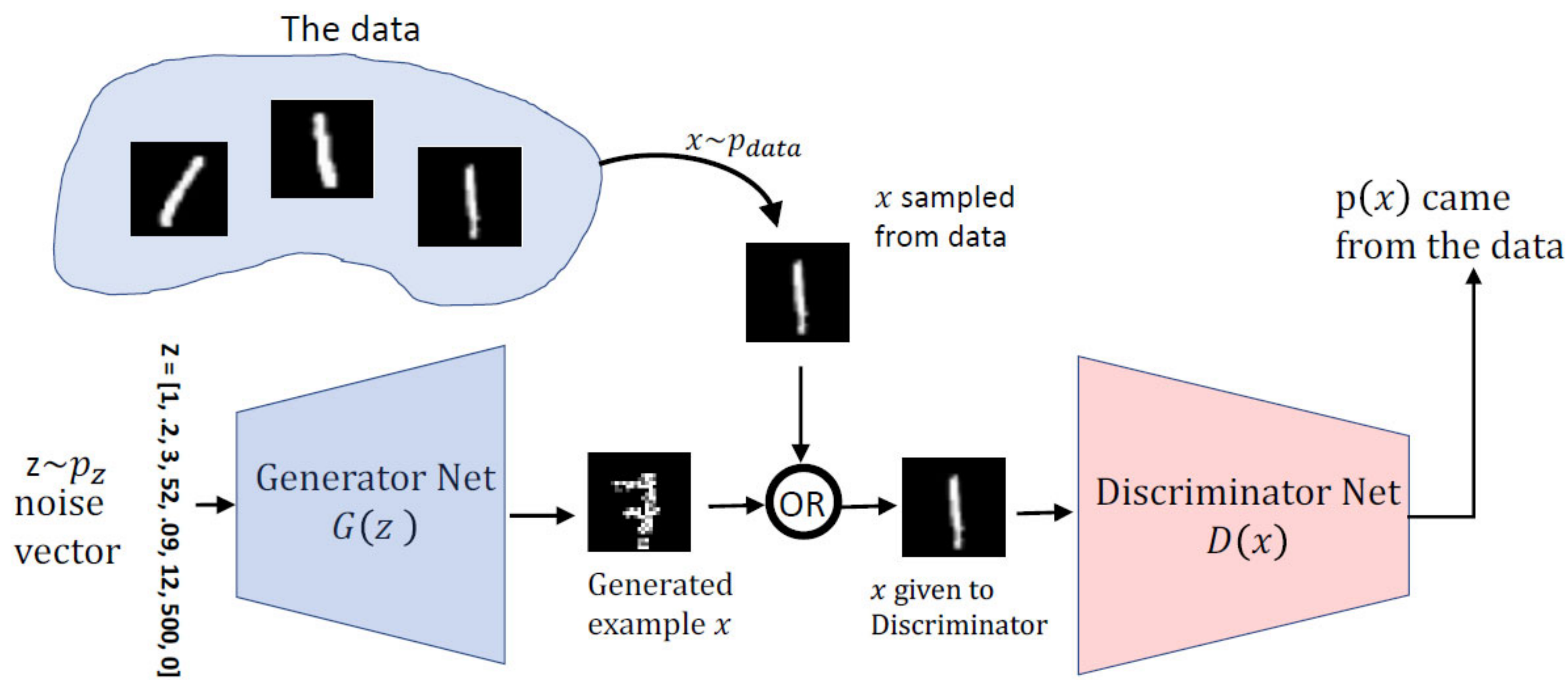
https://cs231n.github.io/convolutional-networks



https://doi.org/10.3390/app12136323

Final Project Methods (*Cont.*)

2. Generative Adversarial Networks

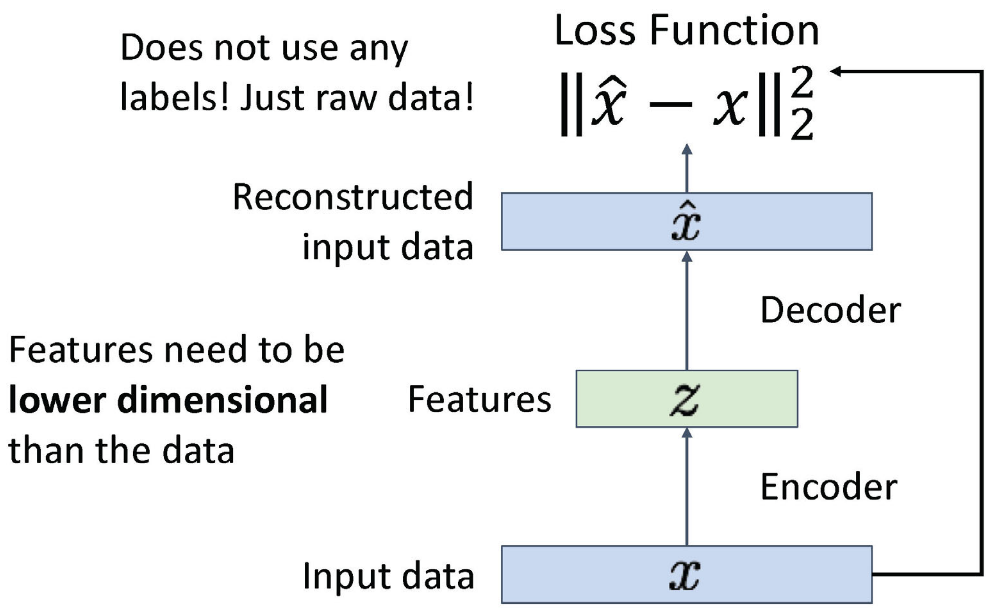


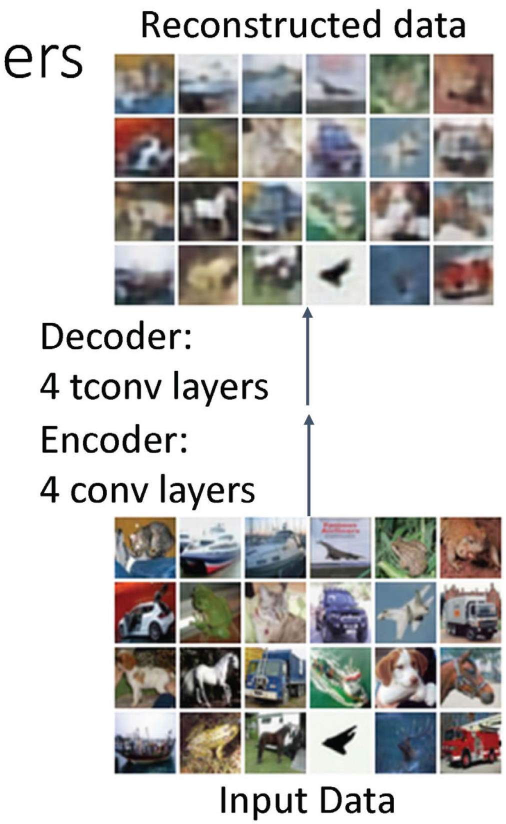
Final Project Methods (*Cont.*)

3. Autoencoders

(Regular, non-variational) Autoencoders

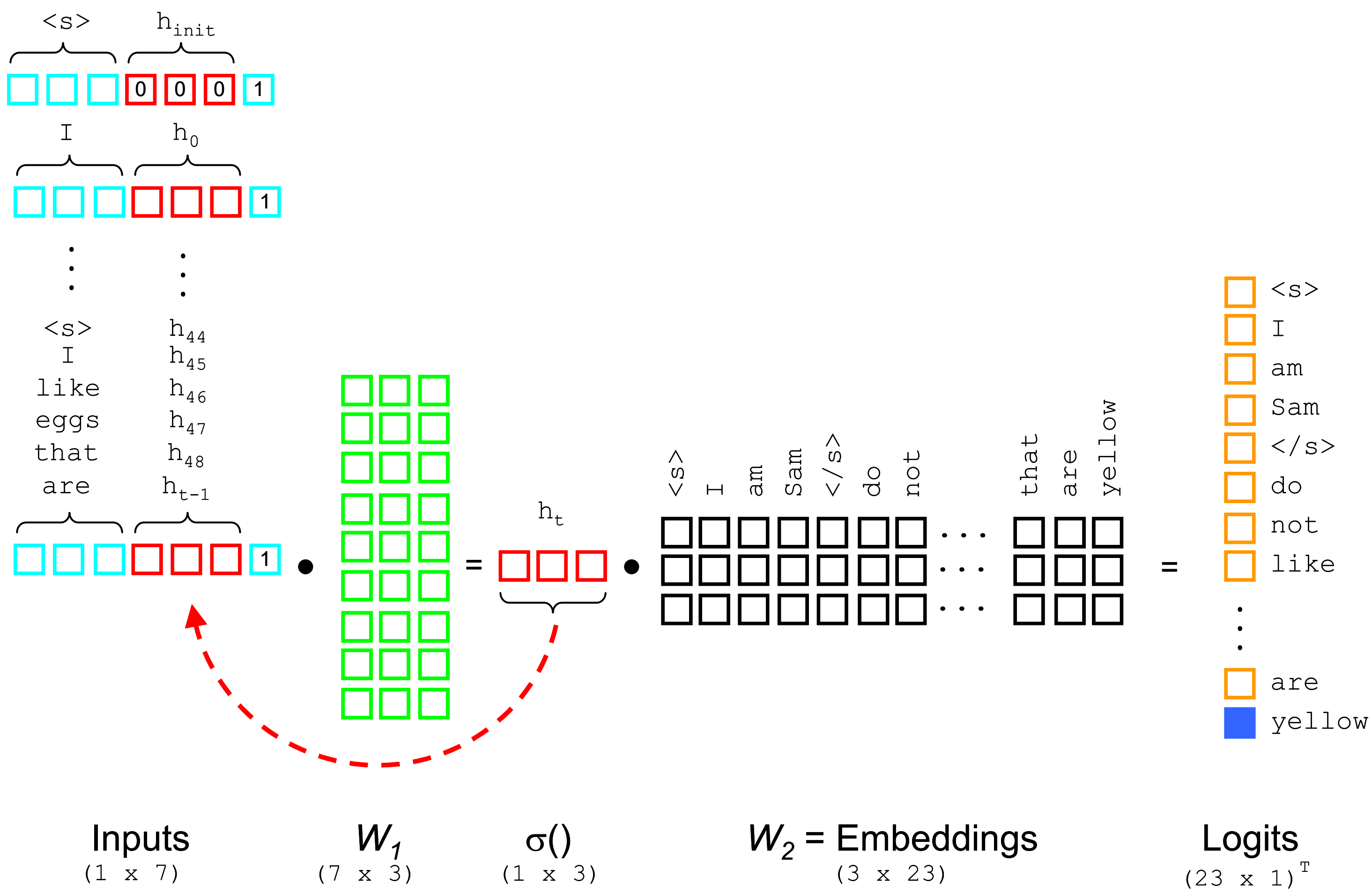
Loss: L2 distance between input and reconstructed data.





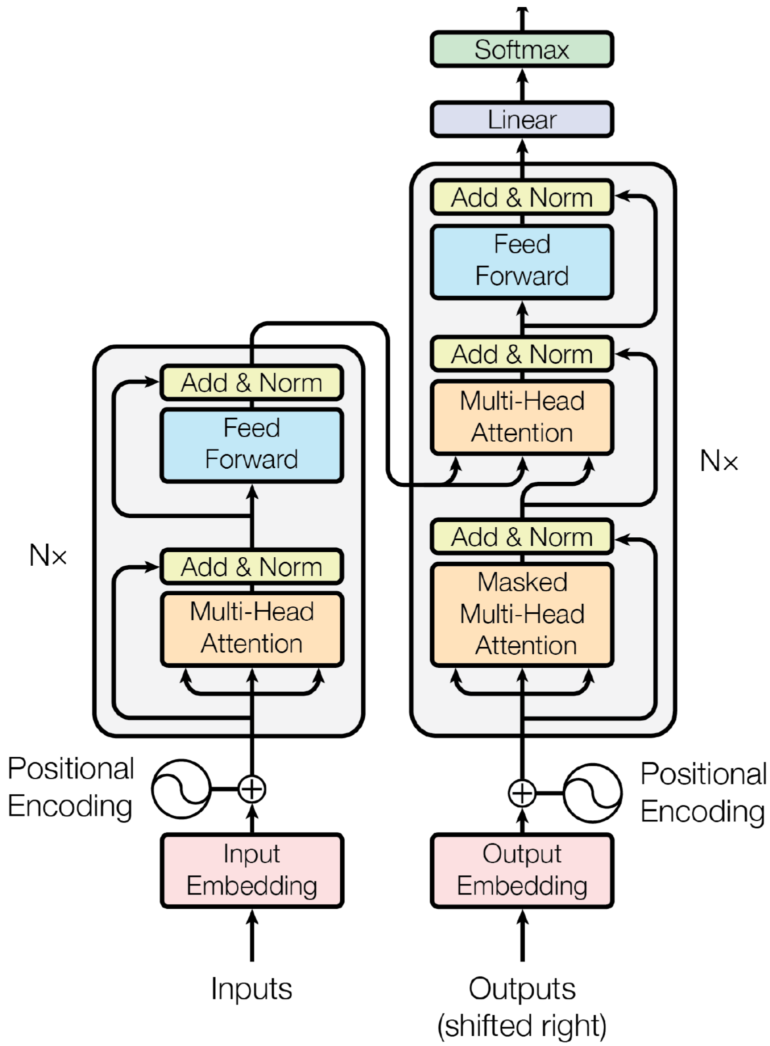
Final Project Methods (*Cont.*)

4. Recurrent Neural Networks



Final Project Methods (*Cont.*)

5. Transformers



Final Project Methods (*Cont.*)

6. Deep Reinforcement Learning

1. Observe state, s_t
2. Decide on an action, a_t
3. Perform action
4. Observe new state, s_{t+1}
5. Observe reward, r_{t+1}
6. Learn from experience
7. Repeat