- CNNs & Modern CNNs
  - Define and compare canonical image tasks
  - Understand how (color) images are represented for ML tasks
    - Identify key challenges associated with this representation that make traditional ML methods inappropriate for working with images
  - Define a convolutional layer
    - Justify their use for image tasks in terms of translation invariance and locality
    - Compute cross-correlations
    - Define padding/stride and justify their use in convolutional layers
    - Describe how convolutional layers operate on multiple input/output channels
    - Compute parameter counts for convolutional layers given a description of the hyperparameters
  - Define pooling and justify the inclusion of pooling layers in CNNs
  - Define data augmentation and justify its use in training CNNs
  - Define batch normalization and justify its use in training CNNs
    - Describe the relationship between batch normalization and dropout
    - Describe how batch normalization operates during inference
  - Define an inception block and describe how they make use of 1x1 convolutions do reduce the output dimensionality
  - Describe the intuition behind residual connections in a residual network (ResNet)

- Unsupervised Learning
  - Describe the K-means model and objective function
  - Compute the update rule for the K-means algorithm
  - Define K-means++ and describe how it addresses nonconvexity of the K-means objective function
  - Prove that for linear projections, minimizing the reconstruction error and maximizing the variance are equivalent objectives
  - Describe the relationship between eigenvectors of the covariance matrix and principal components of a dataset
  - Visually identify the principal components of a dataset
  - Prove that principal component analysis is an inner product method
  - Describe the architecture of an autoencoder and how they are trained

- Deep Generative Models (VAEs and GANs)
  - Compare & contrast generative and discriminative models
  - Define denoising autoencoders and identify what shortcoming of vanilla autoencoders they address
  - Define sparse autoencoders and identify what shortcoming of vanilla autoencoders they address
  - Describe the primary issue with sampling from an autoencoder
  - Compare & contrast autoencoders and variational autoencoders (VAEs)

    - Describe the role of each term in the VAE training objective from the “neural network perspective”
        - Define the reparameterization trick and describe why it is necessary when training VAEs
    - Prove that the ELBO objective function used to train VAEs is a lower bound of the evidence
    - Describe the roles of a discriminator and a generator in a generative adversarial network (GAN)
    - Identify the role of each term in the GAN objective function
    - Define the issue of mode collapse as it pertains to training GANs
    - Describe the intuition behind an UpConvolutional architecture and justify its use in a GAN
    - Define progressive growing in the context of GANs and identify its primary benefits
    - Compare & contrast the architectures/objective functions of GANs and BEGANs
    - Compare & contrast the architectures/objective functions of GANs and SD-GANs
    - Describe how conditional generation extends the image generation task and how this difference affects the training of the generator in a GAN
        - Formulate image-to-image tasks as conditional generation
    - Describe the architecture and objective function used by contrastive language-image pretraining (CLIP)

- RNNs
    - Given a description of a machine learning task, determine if it is a one-to-one, one-to-many, many-to-one or many-to-many task
    - Define a sequence model
        - Define an autoregressive model in the context of sequence modeling
        - Justify the use of autoregressive modeling for natural language tasks
    - Define a recurrent layer and explain why they are appropriate for sequence data
    - Describe how RNNs are trained via backprop through time
    - Describe how RNNs can be used to generate text by sampling
    - Define vanishing and exploding gradients and explain why RNNs are particularly susceptible to these issues
        - Define gradient clipping
    - Define a long-short term memory (LSTM) cell and describe the role of all the components (internal state, input gate and output gate, forget gate)
    - Compare & contrast an LSTM cell and a gated recurrent unit (GRU)
    - Define a bi-directional RNN (BRNNs) and identify tasks that would be more appropriate for BRNNs over RNNs
    - Define padding and bucketing in the context of training RNNs and identify situations when they would be beneficial/necessary

- Attention & Transformers
    - Define an encoder-decoder architecture for sequence-to-sequence tasks and describe the role of the encoder and the decoder
    - Compare & contrast common metrics for evaluating language models (perplexity, BLEU, ROUGE, METEOR)
    - Identify the two main issues with RNN/LSTM language models (as presented in lecture)
    - Describe the high-level intuition behind neural attention and how it addresses the issue of single vector encodings for sequences
    - Compare & contrast self-attention and attention
    - Define scaled dot-product attention
        - Describe the role of keys, values and queries in scaled dot-product attention
        - Compute the dimensionality of outputs and intermediate quantities for self-attention layers
    - Compare & contrast multi-head attention and attention with a single head
    - Define a positional embedding and justify their use in Transformers
    - Identify the tradeoff between word-based and character-based tokenizations
        - Define subword embedding and describe how they address this tradeoff
    - Define masking and justify its use in Transformer models for language modeling

- Pretraining, Fine-tuning & In-context Learning
    - Describe the architecture and training process of ELMO and explain how it was adapted to different tasks
    - Describe the architecture and training process of BERT and explain how it was adapted to different tasks
    - Compare & contrast language modeling and masked language modeling
    - Define transfer learning and explain how the training process of T5 allows it to achieve SoTA performance on a variety of benchmarks
    - Define prompt engineering
    - Define chain-of-thought prompting
    - Define retrieval augmented generation
    - Define soft prompt tuning
    - Define in-context learning and explain the relationship between few-shot, one-shot and zero-shot prompting
    - Define low rank adaptation and justify its use in fine-tuning large language models
    - Define reinforcement learning from human feedback and justify its use in fine-tuning large language models

- Robustness
  - Compare & contrast targeted and untargeted attacks
  - Define adversarial training
  - Define the domain adaptation problem and identify potential goals in this setting
    - Intuitively explain the relationship between classification error and class distribution at training time vs. test time
    - Define covariate and label shift in terms of conditional probabilities
    - Define a confusion matrix and prove that a classifier's column-normalized confusion matrix is label distribution invariant (under some assumptions).

- Fairness
  - Identify characteristics of a word-embedding that are indicative of different forms of bias
  - Compare & contrast disparate treatment and disparate impact
  - Define the following conditions in the context of algorithmic bias: demographic parity, separation, and calibration.
    - Identify the impossibility relationship between these three conditions
    - Describe the key findings of the Propublica expose of the COMPASS software in terms of these conditions
  - Identify whether some method for achieving a fairness condition is a pre-processing or a training constraints/modification

- Interpretability
  - Describe (at a high-level) how the following entities/methods can be used to perform global feature attribution: linear model feature weights, single-feature ablation, permutation-based importance tests, and Shapley values
    - Describe the primary issue with applying these methods on image data
  - Describe the approach behind LIME and identify a key challenge when using it to interpret a model's behavior
  - Describe the approach behind SHAP and identify a key challenge when using it to interpret a model's behavior
  - Describe the approach behind integrated gradients and identify a key challenge when using it to interpret a model's behavior
  - Describe the approach behind counterfactual explanations and identify potential challenges when using it to interpret a model's behavior